



Memoria

Análisis de Jugadores en la Nube

Alumno: M^a de la Vega Rodrigálvarez Chamarro

Consultor: Juan Carlos González Martín

2º Ciclo de Informática

Curso 2011- 2012 Q1

*A mis padres,
por sufrir en silencio estos dos años y medio,
por su generosidad,
por su ayuda*

*A todos mis amigos,
por su apoyo incondicional,
por no abandonarme en el camino*

*A todos ellos,
simplemente por tener paciencia y estar ahí*

Contenido

1. INTRODUCCIÓN.....	6
2. DESCRIPCIÓN DEL PROYECTO	6
2.1. Objetivos del proyecto	6
2.2. Justificación del Proyecto	7
2.3. Retorno de la inversión.....	8
2.4. Organización del Proyecto.....	9
2.4.1. Paquetes de trabajo	9
2.4.2. Planificación Inicial y Real del proyecto	11
2.5. Resultados esperados	16
3. ESTUDIO: ANÁLISIS DE JUGADORES EN LA NUBE	17
3.1. Cloud Computing.....	17
3.1.1. Concepto	17
3.1.2. Arquitectura.....	19
3.1.3. Servicios y Paradigmas de <i>Cloud Computing</i>	20
3.1.4. Tipos de <i>Nubes</i>	24
3.1.5. Ventajas y Desventajas	26
3.1.6. Recomendaciones migración a la <i>nube</i>	30
3.2. Análisis de Jugadores en la Nube	32
3.2.1. IBM.....	32
3.2.2. Salesforce.....	32
3.2.3. Google	33
3.2.4. Amazon.....	40
3.2.5. Microsoft	43
3.2.6. Comparativa de Servicios Ofrecidos por los diferentes proveedores	64
4. PROTOTIPO. TORMENTA DE IDEAS	66
4.1. Análisis de Requisitos.....	66
4.1.1. Requisitos Funcionales	66
4.1.2. Requisitos No Funcionales	68
4.1.3. Requisitos No Contemplados	68
4.2. Diagramas de casos de Uso.....	68
4.2.1. Actores.....	69
4.3. Diseño	70
4.3.1. Diagrama de la Arquitectura de Software.....	70
4.3.2. Diseño de la Base de Datos	71
4.4. Despliegue en Windows Azure	73
4.4.1. Creación de una BD en SQL Azure	73
4.4.2. Creación de una nueva Base de Datos en SQL Azure.....	76
4.4.3. Despliegue de un Web Role en Windows Azure	78

5. OBJETIVOS CONSEGUIDOS.....	82
6. EVALUACIÓN DE COSTES	83
7. PUNTOS DE MEJORA.....	84
8. CONCLUSIONES.....	85
9. SIGLAS Y ACRÓNIMOS.....	87
10. BIBLIOGRAFÍA.....	89

Tabla de Ilustraciones

Figura 1. Plan de trabajo. Dedicación de Recursos.	12
Figura 2. Plan de Trabajo Inicial. Diagrama de Gantt.	14
Figura 3. Plan de Trabajo Real. Diagrama de Gantt.	15
Figura 4. Concepto de Cloud Computing.	17
Figura 5. Arquitectura de Cloud Computing.	20
Figura 6. Capas Cloud Computing.	20
Figura 7. Tipos de Nubes.	24
Figura 8. Google App Engine.	37
Figura 9. Arquitectura de Google DataStore.	38
Figura 10. Modelo de Cloud Computing de AWS.	40
Figura 11. Infraestructura de Amazon.	41
Figura 12. Plataforma Windows Azure.	45
Figura 13. Distribución Servidores Plataforma Windows Azure (2010).	46
Figura 14. Distribución de componentes en la nube.	47
Figura 15. Arquitectura de la infraestructura Windows Azure.	48
Figura 16. Servicios de Computación o Proceso (Compute Services).	49
Figura 17. Servicio de almacenamiento de Windows Azure (Storage Service).	50
Figura 18. Servicio de objetos grandes binarios (blobs).	51
Figura 19. Servicio de Tablas (tables).	51
Figura 20. Servicio de colas (queues).	52
Figura 21. SQL Azure.	56
Figura 22. Sincronización de Bases de Datos.	59
Figura 23. Windows Azure AppFabric.	59
Figura 24. Funcionamiento del bus de servicio.	60
Figura 25. Servicio de control de acceso.	62
Figura 26. Servicio de Caché.	62
Figura 27. Tabla comparativa Google App Engine, Microsoft Windows Azure, Amazon.	65
Figura 28. Diagrama general de casos de uso.	69
Figura 29. Arquitectura Software Sistema.	70
Figura 30. Diseño de Bases de Datos.	71
Figura 31. Diseño Base de Datos. Tablas de Autenticación de Usuarios.	72
Figura 32. Portal Windows Azure.	73
Figura 33. Crear un nuevo servidor de BD.	74
Figura 34. Seleccionar región.	74
Figura 36. Establecer rango de IPs.	75
Figura 35. Configurar inicio de sesión.	75
Figura 37. Establecer rango de IPs de Windows Azure.	75
Figura 38. Servidor de BD creado.	76
Figura 39. Creación de la BD sobre el servidor. Opción Administrar.	77
Figura 40. Acceso al portal de SQL Azure.	77
Figura 42. Portal de Administración de Windows Azure.	79
Figura 41. Empaquetar aplicación.	79
Figura 43. Nuevo servicio hospedado.	80
Figura 44. Despliegue aplicación en Windows Azure.	81
Figura 45. Advertencia número de instancias.	82

1. Introducción

El objetivo del presente documento es realizar la memoria del Proyecto Final de Carrera de 2º ciclo de *Ingeniería Informática* en el área de .NET de la *Universitat Oberta de Catalunya*. En este documento se pretende recoger, de forma sintetizada, toda la información recopilada a lo largo del proyecto denominado *Análisis de Jugadores en la nube*.

2. Descripción del Proyecto

De las tres propuestas ofertadas como posibles proyectos, se escogió el enunciado denominado **Análisis de jugadores en la nube** donde había que analizar las propuestas que los diferentes proveedores del mercado ofrecen en la *nube*, explicando las posibilidades que proponían en cuanto a desarrollo sobre este tipo de plataformas y qué servicios reales se ejecutaban sobre cada una de ellas.

El estudio realizado ha sido más profundo en la propuesta que *Microsoft* ofrece en la *nube*, con el objeto de centrar el proyecto dentro del área seleccionada para elaborar el Proyecto Final de Carrera.

También se ha realizado una prueba de concepto, sobre la plataforma que *Microsoft* ofrece, donde se demuestra cómo desarrollar una aplicación para la *nube* y las prestaciones que determinados servicios de la *nube* poseen.

2.1. Objetivos del proyecto

Los principales objetivos del proyecto son:

- Explicar el paradigma de *Cloud Computing* y su evolución.
- Definir el tipo de servicios que ofrece la *nube* y su utilidad.
- Ventajas y desventajas de trabajar en la *nube*.
- Identificar los principales proveedores que ofrecen servicios en la *nube* profundizando en aquellos que se consideran los grandes jugadores de la misma.
- Realizar una comparativa y destacar los valores diferenciales de las propuestas realizadas por los proveedores más importantes de la *nube*.
- Realizar un prototipo sobre la plataforma propuesta por *Microsoft*, o bien a nivel de desarrollo o bien en cuanto a servicios implementados.

Como objetivos derivados de los principales, se pretende:

- Obtener un amplio conocimiento de los servicios que se ofrecen en la *nube* para conocer cuál es el proceso de adaptación a la misma y las implicaciones que ello conlleva.

- Saber desarrollar una aplicación en la *nube* usando la tecnología que *Microsoft* ofrece.
- Evolucionar hacia las nuevas tendencias en el mercado, para adaptarse a ellas y no quedarse atrás dentro del mundo de las TI.

2.2. Justificación del Proyecto

Se puede decir que la *nube* es la evolución natural de Internet. En los comienzos, Internet era una red basada en protocolos TCP/IP, poco a poco, fue evolucionando y se crearon aplicaciones orientadas a la comunicación tales como, el correo electrónico y la transferencia de ficheros. Más tarde, con la llegada de la *World Wide Web*, Internet paso a ser una fuente enorme de información y contenido. Este punto, llevo a las empresas a introducirse en la red donde, comenzaron a apoyarse en Internet para todo tipo de aplicaciones *e-business*.

El incremento de las aplicaciones basadas en web, y el mayor número de usuarios que eran capaces de acceder a estas aplicaciones hizo que las empresas tuvieran que desarrollar una importante infraestructura. Esta infraestructura tenía que poseer potentes servidores escalables y fiables y, mejores sistemas de gestión que permitieran dar cabida a los servicios que se ofrecían. Con el incremento de la demanda de estas aplicaciones por parte de los usuarios, se debía mantener la calidad de los servicios disponibles y la seguridad de la información que se estaba tratando. El mantenimiento de esta infraestructura se hacía muy difícil y costoso para las empresas, especialmente para las PYMES, haciendo que, en algunos casos, se perdiese de vista el objetivo principal de su negocio, no pudiendo desarrollar correctamente sus productos.

Poco a poco, fueron surgiendo nuevas iniciativas dentro de Internet que permitían un mejor acceso a los recursos informáticos y a las aplicaciones de Internet tales como, la virtualización, el *grid computing*, las arquitecturas orientadas a servicios... Otras propuestas se centraron en hacer Internet más accesible a todo el mundo a través de todo tipo de dispositivos (*smartphones*, dispositivos móviles, sensores...). En los últimos tiempos, ha aparecido el paradigma *Cloud Computing* que permite que Internet se convierta en una plataforma de desarrollo mayor, que amplía y mejora las tecnologías y las capacidades introducidas por las iniciativas anteriormente mencionadas, convirtiendo a la *nube* en la plataforma base para aplicaciones, información y servicios.

Los diferentes proveedores de *Cloud Computing* ofrecen rápidamente a las empresas el acceso a los recursos, tanto de hardware como de software necesarios, sin tener que realizar un desembolso previo. Tener los servicios alojados en la *nube* permite una infraestructura de gran escalabilidad y flexibilidad transparente a las empresas que hace que estas se dejen de preocupar por la infraestructura técnica, el hardware a utilizar, las licencias de software disponibles, de las posibles actualizaciones..., permitiendo centrarse en su negocio, en lugar de hacerlo en los aspectos tecnológicos. Esto facilitará a las empresas la generación y la implementación de nuevas ideas con el consecuente impulso de la innovación.

Con este proyecto se ha pretendido exponer las diferentes opciones existentes en la *nube* y el desarrollo de un pequeño prototipo que permita ver, de forma más objetiva, lo que le puede suponer a un empresa traspasar su modelo de negocio a la *nube* así como, exponer las posibles ventajas y desventajas que supone tener toda la información en la *nube* y la conveniencia o no de pasar los servicios ofrecidos por la empresa a la *nube*.

2.3. Retorno de la inversión

El proceso de transición a la *nube* de los servicios que una empresa dispone, supone definir con detalle la estrategia de transición y elaborar un plan de negocio sólido antes de realizar ningún paso hacia delante. Se deberían identificar las actividades que se consideran críticas para el negocio y valorar el papel que los sistemas de información tienen dentro de estas actividades. La transición a la *nube* de estas aplicaciones requerirá de un análisis detallado que justifique el cambio, la inversión y el sobreesfuerzo inicial que esto puede suponer.

Sin embargo, una vez iniciado el proceso de transición a la *nube*, el retorno de la inversión es apreciable a corto plazo debido a que:

- La empresa elimina todo el gasto de que dedicaba al mantenimiento de la infraestructura donde estaban implementados los diferentes servicios. Esto producirá un ahorro de costes ya que, únicamente dispondrá de los recursos informáticos que, en un momento determinado, necesite y pagará solamente por aquellos recursos que este consumiendo.
- Si los servicios ofrecidos presentan un fuerte crecimiento, el modelo ayuda a alcanzar una mayor escala de operaciones sin grandes esfuerzos ni inversiones añadidas.
- Se disminuirán los costes energéticos debido a que los centros de procesamiento de datos y de almacenamiento ya no se encuentran dentro de la empresa y son compartidos con otras empresas que también tienen sus servicios alojados en la *nube*.
- Se reducen los costes dedicados a la innovación ya que se dispondrá de los recursos informáticos en función de sus necesidades, de esta forma, la empresa se puede centrar únicamente en el desarrollo y mejora de sus productos dándoles un valor añadido y poniendo en práctica ideas innovadoras al disponer de los recursos necesarios para hacerlo sin que ello suponga una inversión fuerte de dinero.

El *Cloud Computing* se presenta como un modelo que puede sostener las operaciones de una empresa con una menor financiación.

2.4. Organización del Proyecto

2.4.1. Paquetes de trabajo

Para una correcta organización del trabajo, el proyecto se ha dividido en paquetes de trabajo, cuya finalización de cada uno se ha hecho coincidir con el plan de entregas de Pruebas de Evaluación Continua definidos en el aula. A continuación se explica con más detalle cada uno de los paquetes de trabajo.

2.4.1.1. PT1. Plan de Trabajo

Objetivos

Este paquete incluye las actividades relacionadas con la definición inicial del proyecto y su planificación a lo largo del tiempo. Los objetivos principales son:

- Definir el alcance del proyecto.
- Establecer los objetivos que se desean alcanzar con la elaboración del proyecto.
- Organizar el proyecto en tareas y desglosarlas a lo largo del tiempo.

Tareas

- T1. Elaboración del PDT. Redacción del Plan de Proyecto donde se incluye la descripción y los objetivos a desarrollar a lo largo de todo el proyecto.
- T2. Elaboración del cronograma. Elaborar el *diagrama de Gantt* con el desglose de todas las tareas a lo largo del tiempo.

2.4.1.2. PT2. Análisis de Jugadores en la nube

Objetivos

Este paquete es la parte principal del proyecto, en él se realiza una investigación profunda de los sistemas de *Cloud Computing* que completa los estudios realizados al inicio del proyecto.

Este estudio contiene un análisis detallado de los grandes proveedores de la *nube* y los servicios que ofertan así como una comparativa entre ellos.

Tareas

- T1. Formación inicial sobre el paradigma de *Cloud Computing*. Formación previa sobre el paradigma de *Cloud Computing*, los nuevos conceptos que aparecen asociados a esta temática, los diferentes tipos de *nubes* y los servicios que pueden llegar a ser ofrecidos.
- T2. Análisis de los diferentes jugadores en la *nube*. Investigación sobre los sistemas de *Cloud Computing* que los diferentes proveedores están ofreciendo.

- T3. Comparativa de servicios. Una vez que se tiene claro lo que ofrece cada proveedor, se realizará una comparativa entre los mismos. Los objetivos principales de esta tarea son establecer las capas en las que se mueve cada jugador, determinar las tendencias de futuro que poseen y, explicar las ventajas y desventajas de trabajar en la *nube* con cada uno de ellos.

2.4.1.3. PT3. Desarrollo Prototipo

Objetivos

En este paquete se profundiza en los servicios que uno de los grandes jugadores ofrece, más concretamente, se estudia los servicios que *Microsoft* ofrece en la *nube*.

En base a lo estudiado, se define un prototipo que sirve como muestra de que tipo de aplicaciones se pueden ofrecer en la *nube* que *Microsoft* proporciona. Para ello, será necesario obtener una formación base en *Windows Azure* para poder desarrollarlo.

Tareas

- T1. Estudio de la Plataforma de Microsoft en la nube. Con esta tarea se pretende profundizar en las posibilidades que *Microsoft* tiene en la *nube* y de esa forma, poder plantear el contenido del prototipo a desarrollar.
- T2. Formación en Windows Azure. Antes de comenzar con el desarrollo del prototipo es necesario estudiar las diferentes herramientas existentes en *Windows Azure*.
- T3. Instalación y Configuración del entorno de desarrollo. Instalación y configuración del entorno de desarrollo que permita construir el prototipo usando la plataforma *Windows Azure*.
- T4. Análisis de requisitos del prototipo. Una vez que se ha definido que es lo que se quiere hacer dentro del prototipo se confecciona el análisis de requisitos para el mismo.
- T5. Diseño del prototipo. Realizar el diseño del prototipo dentro de la *nube*.
- T6. Implementación del prototipo. Desarrollo del prototipo en base a los casos de uso.
- T7. Verificación y pruebas. Efectuar pruebas finales para verificar que el prototipo funciona con se espera.
- T8. Manual de Usuario. Manual donde se indica cómo funciona el prototipo desarrollado.

2.4.1.4. PT4. Difusión de resultados

Objetivos

Este paquete de trabajo tiene como objetivo realizar la memoria final del proyecto que contendrá la descripción del trabajo realizado.

En la memoria se incluye un apartado de conclusiones donde se hace una breve valoración de los resultados obtenidos a lo largo del proyecto.

Por otra parte, se creará un video de presentación en el que se resume el estudio realizado sobre los diferentes jugadores en la *nube* y se muestra una demo del prototipo desarrollado.

Tareas

- T1. Memoria. Elaboración del documento donde se explica el trabajo realizado y las conclusiones a las que se han llegado durante la realización del proyecto.
- T2. Presentación. Creación de un video donde se resume todas las tareas que se han realizado a lo largo del proyecto y las conclusiones a las que se han llegado.

2.4.1.5. PT5. Gestión global del proyecto

Objetivos

Este paquete de trabajo pretende llevar un seguimiento del proyecto a lo largo del mismo para poder detectar rápidamente cualquier desviación que pueda surgir y de esa forma, llevar a cabo las medidas necesarias para que el impacto sobre la totalidad del proyecto sea la menor posible.

Tareas

- T1. Seguimiento del proyecto. Revisión del avance del proyecto a medida que pasa el tiempo.

2.4.2. Planificación Inicial y Real del proyecto

A continuación, se muestra una tabla resumen que contiene las horas que se planificaron al principio del proyecto y las horas invertidas en el mismo.

Nombre de la Tarea	Trabajo por recurso	Trabajo	Trabajo real por recurso	Trabajo Real
PT1. Plan de Trabajo		36 h		24 h
<i>T1. Elaboración del PDT</i>		<i>30,3 h</i>		<i>20 h</i>
Jefe de Proyecto	30,3 h		20 h	
<i>T2. Elaboración del cronograma</i>		<i>5,7 h</i>		<i>4 h</i>
Jefe de Proyecto	5,7 h		4 h	
PT2. Análisis de jugadores en la <i>nube</i>		77,1 h		75,5 h
<i>T1. Formación Paradigma Cloud Computing</i>		<i>21,9 h</i>		<i>28,5 h</i>

Documentador	7,3 h		8,5 h	
Analista	14,6 h		20 h	
<i>T2. Análisis jugadores en la nube</i>		19,2 h		22 h
Desarrollador	6,4 h		5.5 h	
Analista	12,8 h		16.5 h	
<i>T3. Comparativa de servicios</i>		36 h		25 h
Documentador	18 h		15 h	
Analista	18 h		10 h	
PT3. Desarrollo del prototipo		133,2 h		128 h
<i>T1. Estudio Plataforma Microsoft</i>		14,55 h		19 h
Analista	14,55 h		19 h	
<i>T2. Formación en Windows Azure</i>		14,55 h		15 h
Desarrollador	4,85 h		9 h	
Analista	4,85 h		3 h	
Diseñador	4,85 h		3 h	
<i>T3. Instalación y Configuración entorno desarrollo</i>		9,6 h		3 h
Desarrollador	9,6 h		3 h	
<i>T4. Análisis de requisitos</i>		8,4 h		10 h
Analista	8,4 h		10 h	
<i>T5. Diseño del prototipo</i>		26,1 h		10 h
Diseñador	26,1 h		10 h	
<i>T6. Implementación</i>		38,7 h		45 h
Desarrollador	38,7 h		45 h	
<i>T7. Verificación y pruebas</i>		10,2 h		6 h
Desarrollador	10,2 h		6 h	
<i>T8. Manual de Usuario</i>		19,5 h		20 h
Documentador	13 h		15 h	
Desarrollador	6,5 h		5 h	
PT4. Difusión de resultados		56,4 h		67 h
<i>T1. Memoria</i>		28,2 h		30 h
Documentador	28,2 h		30 h	
<i>T2. Presentación</i>		28,2 h		37 h
Documentador	18,8 h		30 h	
Jefe de Proyecto	9,4 h		7 h	
PT5. Gestión del proyecto		24,23 h		10 h
<i>T1. Seguimiento del proyecto</i>		24,43 h	10 h	
Jefe de Proyecto	24,43 h			
Total		335,53 h		304,5 h

Figura 1. Plan de trabajo. Dedicación de Recursos.

Tal y como se puede observar en la tabla, existen actividades tales como la *implementación* del prototipo y la generación de la documentación donde ha sido necesario invertir más horas de las previstas por lo tanto suponía retrasos en las tareas que dependían de ellas. También se puede observar que existen tareas que su dedicación ha sido menor de la esperada, como por ejemplo la instalación del entorno de desarrollo, lo que ha supuesto que se hayan podido compensar unas tareas con otras.

El riesgo más importante que ha afectado al desarrollo del proyecto ha sido el compaginarlo con la elaboración de trabajos de otras asignaturas. Esto ha obligado a cambiar la distribución del tiempo de forma que ha habido semanas completas que no se ha podido trabajar en el proyecto y en otras semanas se ha tenido que incrementar la dedicación al mismo para llegar a cumplir objetivos.

Los hitos principales del proyecto, coincidentes con la entrega de prácticas, se han finalizado en las flechas planificadas tal y como se muestra en los siguientes diagramas de Gantt.

Diagrama de Gantt. Planificación Inicial

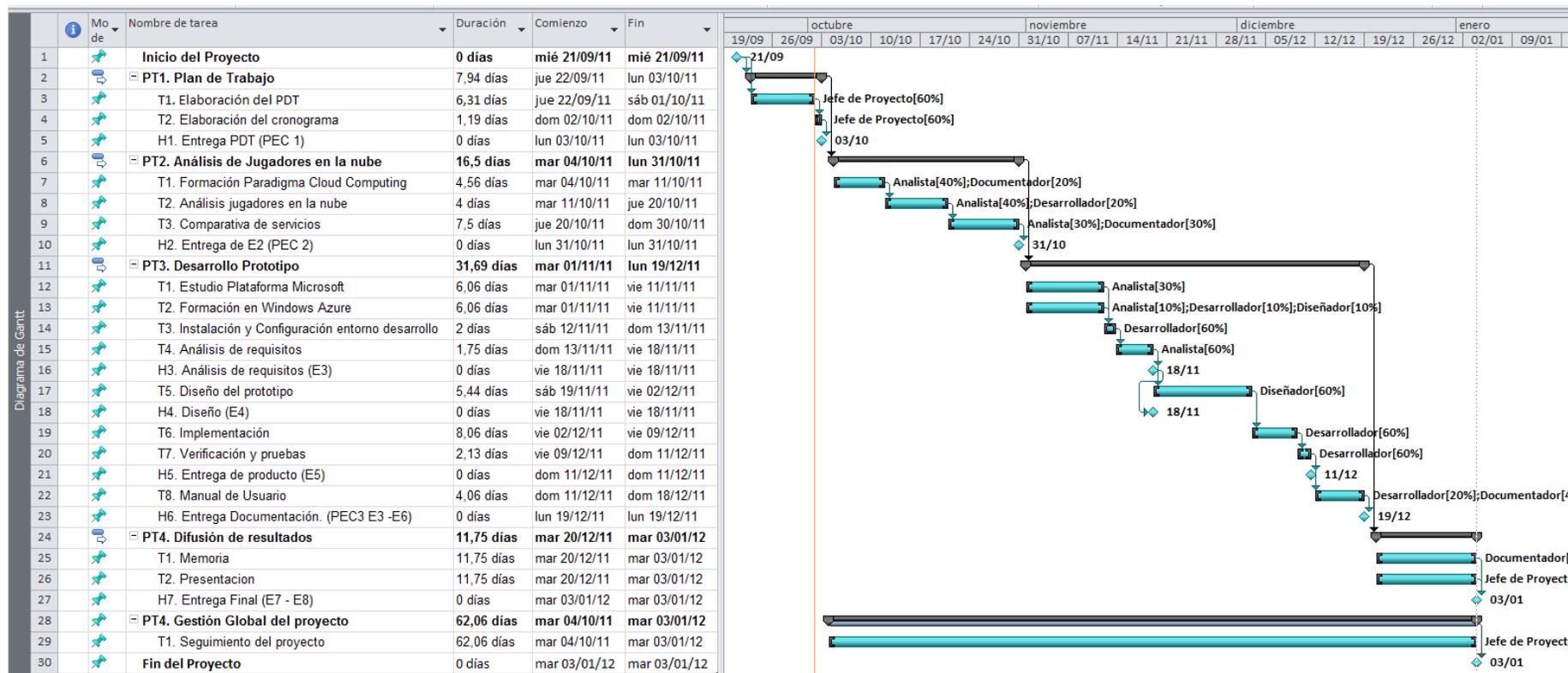


Figura 2. Plan de Trabajo Inicial. Diagrama de Gantt.

Diagrama de Gantt. Planificación Real

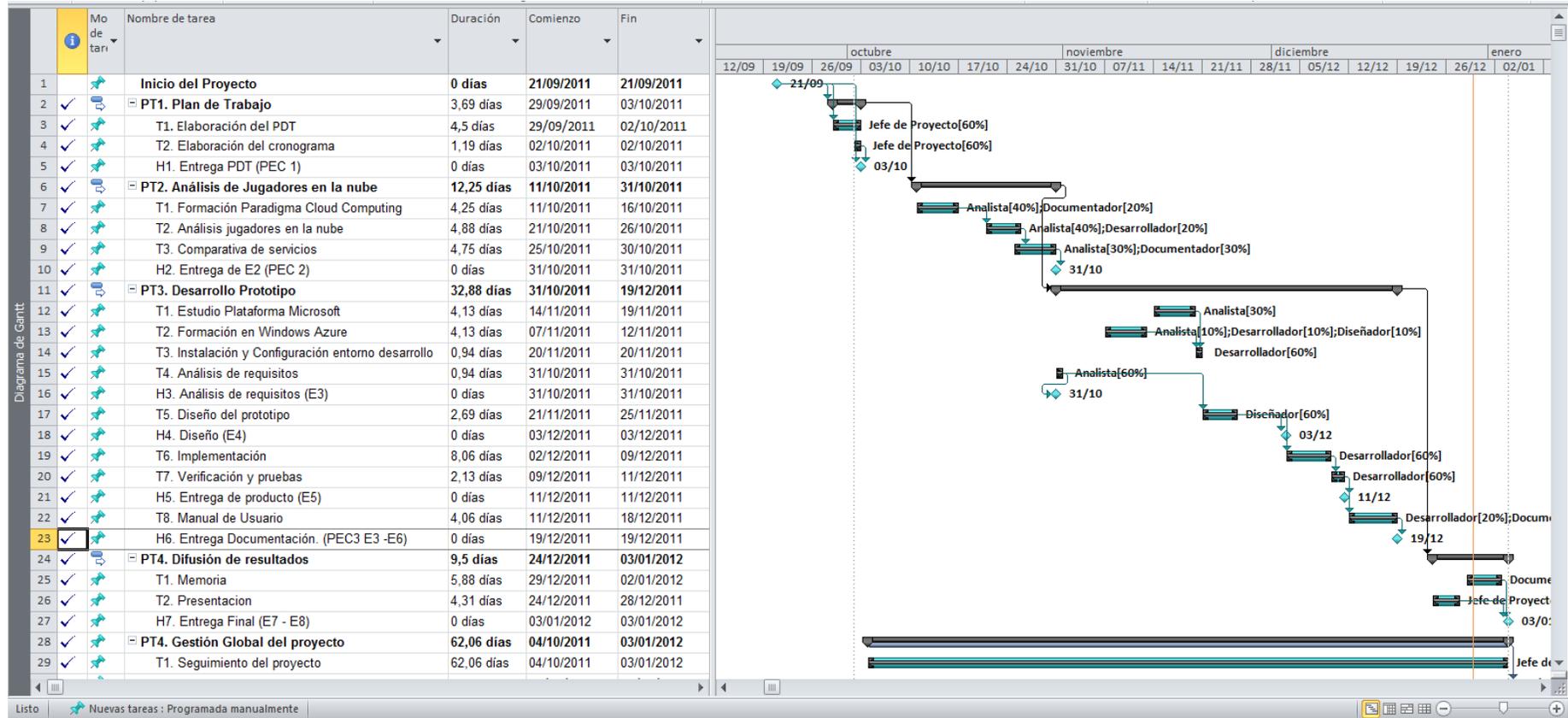


Figura 3. Plan de Trabajo Real. Diagrama de Gantt.

2.5. Resultados esperados

Al finalizar el proyecto se pretende conseguir los siguientes resultados:

- Un informe donde se defina el concepto de *Cloud Computing* y se enumeran las ventajas y desventajas que supone trabajar dentro de la *nube*. También contendrá una evaluación de los grandes proveedores que actualmente están ofreciendo sus servicios en la *nube*. El informe contendrá una explicación de la categoría en la que se están moviendo los diferentes jugadores, los servicios que cada uno de ellos ofrece y una comparativa entre las propuestas que cada proveedor está haciendo disponibles, así como las limitaciones que poseen.
- Un prototipo desarrollado dentro de la *nube*, usando los servicios de *Windows Azure*, de forma que una empresa pueda ver de forma práctica la utilización de los servicios dentro de la *nube*.

3. Estudio: Análisis de Jugadores en la *nube*

3.1. Cloud Computing

En este punto se profundiza sobre el concepto de *Cloud Computing* y sobre todos los servicios y paradigmas que aparecen asociados a esta tecnología.

Por otra parte, se definen los diferentes tipos de *nubes* existentes, cuando se recomienda trabajar con cada una de ellas y los diferentes servicios que se pueden ofrecer.

3.1.1. Concepto

Según el Laboratorio de Tecnologías de la información, dentro del *National Institute of Standards and Technology* (NIST) perteneciente al Departamento de Comercio del gobierno Federal de los Estados Unidos, se define *Cloud Computing* de la siguiente forma:

“*Cloud Computing* es un modelo que permite el acceso bajo demanda y a través de la red a un conjunto de recursos compartidos y configurables (como redes, servidores, dispositivos de almacenamiento, aplicaciones y servicios) que pueden ser rápidamente asignados y liberados con una mínima gestión por parte del proveedor del servicio”¹.

Cloud Computing se puede definir como un nuevo paradigma de computación donde los recursos se ofrecen de forma dinámica, escalable y virtualizada como servicios a través de Internet. Por medio de esta tecnología, los clientes o usuarios finales, sin tener que preocuparse del mantenimiento y la infraestructura subyacente utilizada para prestar los diferentes servicios, accederán a diferentes plataformas de *hardware* y *software*, servidores virtualizados, dispositivos de almacenamiento, aplicaciones de *software* y plataformas para el desarrollo de aplicaciones, usando los servicios ofrecidos por los diferentes proveedores de *Cloud Computing*. El acceso podrá ser realizado desde una gran variedad de dispositivos tales como ordenadores personales, portátiles, *smartphones*, PDAs...

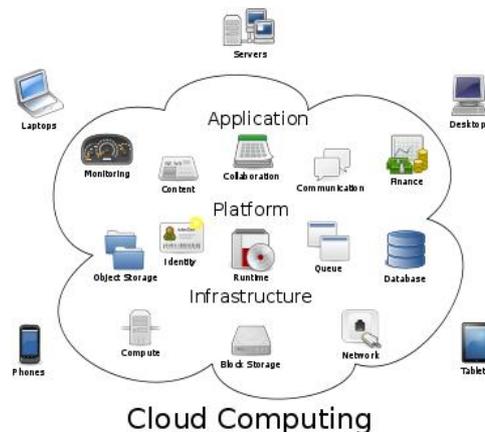


Figura 4. Concepto de *Cloud Computing*.

¹ (Cierco, 2011)

El proveedor de los servicios es quien gestiona el coste de los servidores, la infraestructura necesaria para mantener todos los servicios, las actualizaciones de *software* necesarias, la virtualización de las máquinas y el alojamiento en sus servidores de las aplicaciones que los clientes pueden utilizar de forma que, el cliente no tiene que preocuparse por todos los gastos que ello implica (mantenimiento, electricidad, compra de servidores, formación de personal para el mantenimiento de los sistemas...), pagando únicamente por aquellos servicios que está utilizando (*pay-per-use*).

Las principales características que ofrece el modelo de *Cloud Computing* son:

- *Autoservicio bajo demanda*: El usuario accede a los servicios de la *nube* únicamente cuando sean necesario sin necesidad de interactuar con el proveedor de los servicios. Se disponen tanto de servicios proporcionados por terceros como de desarrollos propios. Los servicios ofrecidos están escalados sobre diversos centros de datos.
- *Sistemas autónomos*: Los sistemas de *Cloud Computing* son sistemas autónomos gestionados de forma transparente al usuario. El *software* y los datos existentes dentro de la *nube* pueden ser reconfigurados según las necesidades de los diferentes usuarios. Los recursos son asignados de forma granulada sin que tenga que ser necesaria la intervención humana.
- *Flexibilidad*: Los recursos se asignan y se liberan de forma rápida, dando la sensación al usuario que dispone de recursos ilimitados y que siempre se encuentran a su disposición.
- *Compartición de recursos*: Los recursos (sistemas de almacenamiento, memoria, ancho de banda, capacidad de procesamiento, máquinas virtuales,...) son compartidos por varios usuarios, a los cuales se les asignan las capacidades de forma dinámica según sus peticiones o necesidades. Para los usuarios es transparente el lugar donde se encuentran estos recursos.
- El *acceso a los recursos es a través de la red* y por medio de mecanismos estándar que son utilizados tanto por teléfonos móviles como por ordenadores personales PDAs...
- El acceso a través de Internet también requerirá que el proveedor tengan configurados *sistemas de seguridad y de calidad de servicio*.
- El acceso a los diferentes servicios proporcionados por los diferentes proveedores no requiere un perfil de usuario muy especializado y se puede realizar a través de diversos navegadores o APIs que proporcionan un marco de trabajo estándar para acceder e integrar los servicios existentes en la *nube*.
- *Multiusuario*: Permite usar los recursos a varios usuarios de forma simultánea compartiendo los costes entre todos ellos.
- Por parte de los clientes finales no requiere de una gran inversión.

- El proveedor es capaz de medir el uso que cada cliente ha hecho del sistema lo que implica que se *puede pagar únicamente por aquellos que se está utilizando*. El precio se basa en las capacidades de uso.
- *Calidad de servicio (QoS)*: Los proveedores de la *nube* garantizarán la calidad del servicio en función de los rendimientos de la CPU, el *hardware*, el ancho de banda y la capacidad de la memoria.
- *Fiabilidad*: Dispone de redundancia de recursos proporcionando una alta fiabilidad, siendo muy adecuada para situaciones de continuidad de negocio y recuperación de desastres.
- *Seguridad*: La seguridad es igual de buena o mejor que en los sistemas tradicionales, ya que los proveedores del servicio son capaces de manejar determinados problemas para los que los clientes no están preparados.
- El *mantenimiento es menor* debido a que el *software* que utiliza el cliente final no se encuentra instalada en cada uno de los equipos.

El cambio más grande que provoca el *Cloud Computing* se encuentra en el diseño de los servicios y aplicaciones en sí, que los hará más sencillos a la hora de consumirlos y de interactuar con ellos desde cualquier lugar e incluso desde dispositivos móviles. La *nube* supone la evolución natural de Internet.

3.1.2. Arquitectura

El entorno general sobre el que está montado el sistema de *Cloud Computing* puede dividirse en dos partes: la parte del núcleo y la parte de gestión.

El núcleo se divide en tres capas:

- *Capa de recursos*: Es la infraestructura del sistema y está formada por los servidores, dispositivos de almacenamiento y redes tanto físicos como virtualizados. Todos estos elementos se pueden encontrar geográficamente distribuidos de forma transparente a los clientes. La forma en la que estará montada esta infraestructura depende del proveedor y de la manera en la que haya decidido implementar la *nube*.
- *Capa de plataforma*: Es la capa más compleja y puede ser dividida en múltiples subcapas como por ejemplo: un marco de computación donde se gestione el envío de transacciones y la programación temporal de las tareas, una subcapa de almacenamiento que proporciona almacenamiento ilimitado y capacidades de cacheo. El servidor de aplicaciones y otros componentes apoyan la lógica general de aplicaciones usando las capacidades de la plataforma bajo demanda o a través de una gestión flexible de forma que, ningún componente sea un cuello de botella para el sistema general.

- *Capa de aplicaciones:* Basado en los recursos y en los componentes que dispone, las aplicaciones pueden soportar transacciones distribuidas de gran tamaño y gestionar grandes volúmenes de datos.

Todas las capas proporcionan servicios externos de gestión a través de servicios web o interfaces abiertos. Estos servicios pueden ser de configuración, monitorización, programación de tareas, registro...

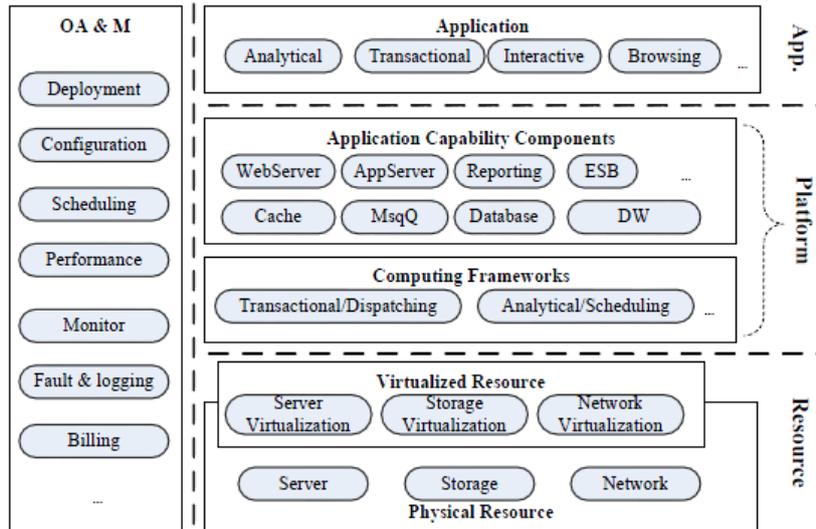


Figura 5. Arquitectura de Cloud Computing.

3.1.3. Servicios y Paradigmas de Cloud Computing

Cloud Computing se puede ver como una colección de servicios ofrecidos a los clientes que se distribuyen en tres capas: Infraestructura como servicio (IaaS), Plataforma como servicio (PaaS) y Software como servicio (SaaS).



Figura 6. Capas Cloud Computing.

Los clientes accederán a la nube y harán uso de los diferentes servicios que se ofrecen en las diferentes capas como por ejemplo:

- Alquilar necesidades básicas como CPU, memoria, capacidad de almacenamiento, servidores de aplicaciones, bases de datos... con el objeto de construir centros de datos virtuales (IaaS).
- Desarrollar y alojar aplicaciones de software o servicios que hayan construido (PaaS).
- Acceder a servicios web a través de interfaces estándar o a diversas aplicaciones desde un portal web (SaaS).

Todos estos servicios se engloban en lo que se conoce como *Tecnología de la información como Servicio* (ITaaS) que engloba todo lo que comprende desde los usuarios finales hasta los centros de datos virtuales.

A continuación, se explicará con un poco más de detalles las diferentes capas.

3.1.3.1. Infraestructura como Servicio (IaaS)

Se considera la primera capa de *Cloud Computing* y comprende los activos físicos de la *nube* tales como servidores, dispositivos de almacenamiento, switches, racks, *software* de configuración... En este nivel, simplemente se ofrece a la organización el *hardware* para que pueda instalar todo lo que quieran sobre él. Cada proveedor de la *nube* organiza los diferentes elementos en la manera que cree más conveniente.

En este nivel principalmente se alquila capacidad de almacenamiento y computación básica. La unidad base que se proporciona es el servidor y esto incluye CPU, memoria, soporte de almacenamiento, sistema operativo y software de monitorización.

La automatización de la plataforma IaaS es una pieza fundamental, de esta manera, el cliente irá ampliando los recursos que necesita de forma automática y escalable sin que el proveedor esté involucrado en el proceso. El usuario irá obteniendo más recursos (nuevos servidores, espacio de almacenamiento...) conforme sus necesidades así lo requieran. La infraestructura necesaria por el cliente será escalable dinámicamente hacia arriba o hacia abajo en función de recursos necesitados en un momento determinado.

La virtualización es otra pieza clave dentro de esta tecnología ya que, se puede maximizar la utilización de los recursos de forma eficiente y se puede reducir el coste de la plataforma IaaS debido a que los diferentes usuarios podrán compartir los recursos.

Los clientes podrán monitorizar y controlar el hardware que están utilizando a través de Internet.

El cliente se despreocupa de la gestión de los servidores físicos siendo el proveedor el que se tiene que preocupar por ello y además debe garantizar un cierto nivel de rendimiento del sistema.

El cliente pagará por consumo de recursos: espacio en disco utilizado, tiempo de CPU, espacio en base de datos, transferencia de datos...

Un ejemplo de IaaS son los servicios ofrecidos por *Amazon*.

3.1.3.2. Plataforma como Servicio (PaaS)

Es la siguiente capa que forma parte de la *nube*. En esta capa, el proveedor de servicios ofrece capacidades de TI paquetizadas o recursos lógicos tales como bases de datos, sistemas de ficheros, entornos de aplicación.

En este nivel se encuentran involucradas dos tecnologías base:

- Desarrollo de *software* que será creado, testeado y ejecutado en la *nube*. Gracias a la mejora del ancho de banda, los desarrolladores pueden crear aplicaciones de forma remota accediendo a través del navegador web. También se da la posibilidad de integrar herramientas de desarrollo local sobre *Cloud Computing*, lo que significa, que una vez que las aplicaciones han sido creadas de forma local, se desplegarán directamente en la *nube* a través de las herramientas de integración. Incluye aplicaciones de diseño, desarrollo, test, despliegue y alojamiento.
- Entorno operativo de aplicaciones distribuidas a larga escala. Se ofrece el acceso a aplicaciones de *middleware*, bases de datos y sistemas de ficheros que se encuentran distribuidos en gran cantidad de servidores. De esta forma se permite el acceso de millones de usuarios a través de Internet. Estos entornos permiten la colaboración en equipo, la integración de servicios web, de bases de datos, escalabilidad, gestión de estado y versionado.

Estas herramientas suelen ofrecer soporte de ayuda en la creación de interfaces de usuario que suelen estar basados en HTML o *JavaScript*. Los protocolos de comunicación utilizados para realizar llamadas a los diferentes servicios web son SOAP y REST.

Los servicios ofertados en esta capa permiten el desarrollo de aplicaciones sin tener que invertir dinero en servidores específicos, licencias, software especializado...La infraestructura bases tanto el *software* como el *hardware*, es transparente para el desarrollador.

Por el contrario, en esta capa suele existir un problema, cada proveedor la ha desarrollado según sus políticas y puede resultar difícil el traspasar aplicaciones desarrolladas con un proveedor a otro, creando una fuerte dependencia con los mismos.

Algunos de los proveedores que ofrecen servicios dentro de esta capa son *Microsoft (Windows Azure Platform)*, *Google (Google App Engine)* o *Salesforce.com (Force.com)*.

3.1.3.3. Software como Servicio (SaaS)

Esta capa se encuentra en el nivel más alto y es en el que la mayoría de los usuarios que no se encuentran dentro del entorno de TI verán y consumirán.

En este nivel los proveedores ofrecen aplicaciones que los usuarios finales o las organizaciones utilizan con una mínima personalización.

En este modelo, la aplicación es alojada como un servicio de cara a los clientes que acceden a través de Internet. El cliente no necesita mantener la aplicación, es el proveedor quien decide cuando se actualizan versiones e instala los parches necesarios para mantener la infraestructura funcionando.

Para los proveedores esta capa supone una fuente continua de ingresos ya que, los clientes pagarán de forma periódica por el uso de las aplicaciones y no únicamente cuando deseen comprar una nueva actualización. Para el cliente supone una inversión menor. En menos tiempo es capaz de valorar si usando esa herramienta es capaz de mejorar su productividad y le supone menos costes de licencias y un ahorro en los gastos de instalación y mantenimiento tanto de *hardware* como de *software*.

Se considera que existen dos categorías de aplicaciones:

- Soluciones de negocios orientadas a empresas tales como aplicaciones para la cadena de suministro, CRM...
- Servicios orientados al cliente o al público en general que pueden ser de pago o libres (web mail, oficina virtual de un banco,...)

Las tecnologías involucradas en esta capa son *Web 2.0*, *Mashup*, *SOA* y multiusuario.

3.1.3.4. Otras categorías de servicios en la *nube*

Existen otros servicios ofertados en la *nube* que derivan de los modelos mencionados con anterioridad.

Database as a Service

Este servicio deriva del nivel *PaaS*. Se basa en la idea de evitar el coste y complejidad que al cliente le supone mantener sus propias bases de datos. La organización no se preocuparía del mantenimiento de los servidores donde se encuentran las bases de datos ni de toda la seguridad de la mismas (redundancia de datos, sistemas de *backup*...) La organización sólo se preocuparía de crear y gestionar la base de datos.

Las bases de datos creadas por la organización podría integrarse con otros servicios tales como calendarios, emails... que diesen más valor a la misma.

Software más Servicio (S+S)

En este modelo, los servicios que se ofrecen a través de Internet, son reforzados con *software* que se ejecuta localmente en la máquina. Esto proporciona la flexibilidad de usar un proveedor en la *nube* y la confiabilidad de tener la información almacenada en la propia organización.

Este servicio permite mejorar la experiencia de usuario combinando *software* cliente, que proporcionan características específicas, con la capacidad de conectarse a Internet. Permite trabajar *offline* y sincronizar la información cada cierto tiempo con la que hay en la *nube*. Por otra parte, también permite mantener los datos más sensibles en la organización y el resto de la información subirla a la *nube*.

Microsoft ofrece este tipo de servicios dentro de su plataforma ya que permite el desarrollo de aplicaciones de forma local y una vez creadas se pueden integrar en la *nube* a partir de herramientas instaladas en las máquinas locales.

Otro ejemplo serían las aplicaciones de sincronización de los dispositivos móviles como por ejemplo *iTunes*...

Cloud Storage (Storage as a Service)

Es un servicio de almacenamiento de datos en la *nube*, el proveedor alquila espacio a los clientes. Este servicio permite acceder a los datos desde cualquier sitio donde exista conexión a Internet. Los datos se almacenan en diferentes servidores proporcionando redundancia para asegurar que los clientes dispongan en cualquier momento de sus datos.

Las ventajas que ofrece este servicio es la protección de los datos en caso de desastre.

Los problemas con los que se puede encontrar este servicio son los posibles ataques a la seguridad y, que determinados datos no puedan ser almacenados en la *nube* por temas de *copyright* o sensibilidad de los mismos.

3.1.4. Tipos de Nubes

Basándose en la localización de los servicios y de las aplicaciones, las *nubes* se pueden clasificar de la siguiente forma:

- *Nubes* públicas o externas.
- *Nubes* privadas o internas.
- *Nubes* Híbridas.

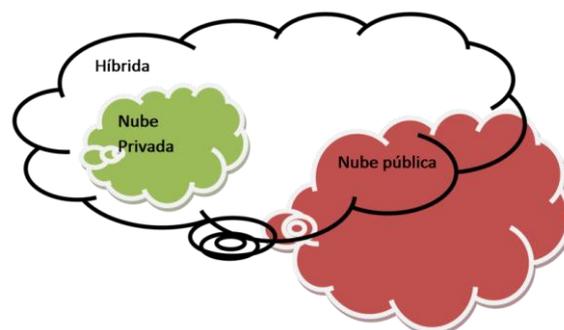


Figura 7. Tipos de Nubes.

3.1.4.1. Nubes públicas

En una *nube* pública los recursos de computación son proporcionados dinámicamente sobre Internet al público en general a través de aplicaciones o servicios *web*.

Estas *nubes* suelen ser manejadas por terceras partes y las aplicaciones de diferentes clientes están ejecutándose a la vez sobre los mismos servidores mientras comparten dispositivos de almacenamiento o redes.

En este modelo el proveedor hace sus recursos de TI escalables de forma masiva. La *nube* pública es el primer modelo de desarrollo de los servicios en la *nube*.

Los usuarios finales no son conscientes de las aplicaciones que pueden estar ejecutándose a la vez en el mismo servidor.

La *nube* pública ofrece a los usuarios finales un modelo limpio y con menos infraestructura, pero para entornos empresariales existen aplicaciones que no pueden ser introducidas en ella por temas de seguridad y confidencialidad de la información, rendimiento, fiabilidad, cuestiones legales.

3.1.4.2. Nube privada

Son aquellas *nubes* cuya infraestructura opera únicamente sobre una sola organización. Estas *nubes* son construidas para uso exclusivo de un único cliente, proporcionando un control total sobre los datos, la seguridad, la calidad y el servicio.

Representa un modelo de desarrollo donde las empresas ofrecen servicios en la *nube* sobre las redes corporativas a sus usuarios internos detrás de un entorno totalmente protegido por cortafuegos. Está orientada a grandes corporaciones que debido a su gran tamaño pueden beneficiarse de los servicios que ofrecen la *nube*.

Las *nubes* privadas pueden ser creadas y gestionadas por los propios grupos de TI de la organización o por un tercero que puede ser el proveedor de la *nube*.

El hecho de que la *nube* sea privada hace que se pierda alguna de las cualidades principales de la misma ya que hay que comprarla, construirla, gestionarla y mantenerla.

3.1.4.3. Nube Híbrida

Esta *nube* combina múltiples modelos de *nubes* privadas y públicas. Es un modelo de desarrollo donde una organización proporciona servicios en la *nube* gestionados a nivel interno y accede a otros que son proporcionados de forma externa. Una organización puede almacenar datos de clientes dentro de sus propios centros de datos y conectarse a un servicio en la *nube* de procesamiento de datos cuando sea necesario.

Las *nubes* híbridas presentan la complejidad de determinar cómo distribuir las aplicaciones por las diferentes *nubes* y suelen ser más difíciles de mantener.

Por otra parte las nubes híbridas representan el modelo despliegue idóneo para aquellas organizaciones que poco a poco van traspasando sus servicios a la *nube*.

3.1.5. Ventajas y Desventajas

La computación en la *nube* es una tecnología que todavía está en vías de desarrollo y, al igual que otras tecnologías o servicios, tiene sus ventajas y presenta algunos riesgos que es necesario tener en cuenta para poder resolverlos.

Por otra parte, el hecho de traspasar información a *nube* y la deslocalización de la información en la misma, hace que se tengan que tener en cuenta diversos aspectos legales relativos a la seguridad de la información y a la ley de protección de datos (LOPD) que habrá que revisar cuando se firme un acuerdo con un proveedor de servicios.

3.1.5.1. Ventajas

Cloud Computing es una estrategia ganador-ganador para el proveedor de servicios y para el consumidor de los mismos y ofrece las siguientes ventajas:

- *Escalabilidad.* Satisface los requisitos de negocio bajo demanda redimensionando los recursos ocupados por las aplicaciones para satisfacer los requisitos cambiantes de los clientes. La disposición de los mismos es en función de sus necesidades y en base a un modelo automático.
- El espacio de almacenamiento va incrementándose conforme el cliente lo necesite, esto da la sensación de disponer de un sistema de almacenamiento ilimitado.
- Tiene un coste bajo y permite a las organizaciones ahorrar energía. Esto es debido a que las organizaciones podrán usar ordenadores de bajo coste para conectarse a los diferentes servicios y no necesitarán mantener en su organización grandes dispositivos de *hardware*.
- No es necesario mantener una plantilla completa de TI para mantener los sistemas ya que ahora será tarea del proveedor. Los recursos humanos de la empresa podrán estar dedicados a los aspectos básicos de su negocio.
- Mejora la eficiencia de la gestión de recursos a través de la programación dinámica de los mismos.
- *Movilidad.* Permite el acceso a los servicios alojados en la *nube* desde cualquier punto donde se disponga una conexión a Internet. Este es un aspecto muy beneficioso para aquellas compañías que dispone muchos trabajadores con perfil de movilidad.
- El pago de los servicios es por consumo (*pay-per-use*).
- Las aplicaciones que se desarrollan en el entorno de la *nube* pueden ser desplegadas y testeadas con mayor facilidad y su mantenimiento económico es mucho menor.

- Las actualizaciones de las aplicaciones que se ofrecen dentro de la *nube* son realizada automáticamente por parte del proveedor, sin tener que preocuparse el cliente de su mantenimiento.

3.1.5.2. Riesgos

A la hora de migrar el núcleo de negocio de una organización, es necesario tener en cuenta determinados riesgos que pueden existir para intentar mitigarlos en la medida de lo posible.

Los riesgos que pueden darse en el entorno de *Cloud Computing* son:

- *Seguridad de la información.* El usuario final pierde el control de la gestión de la seguridad en la *nube* y se preocupa por el acceso a su información por parte de terceros no autorizados o por la pérdida de los datos.

Hay que tener especial cuidado con aquellas aplicaciones que se pueden considerar críticas y con aquellos datos sensibles que no puedan ser publicados en Internet.

En este caso los proveedores usan las mismas técnicas y herramientas que las usadas en los centros de datos tradicionales aplicando fuertes perímetros de seguridad en la red.

Por otra parte para evitar que existan ataques entre máquinas virtuales que están ubicadas dentro de mismo servidor, el proveedor despliega mecanismos de seguridad que incluye *firewalls*, detección y prevención de intrusos, monitorización de integridad e inspección de registros.

También será necesario adoptar medidas de seguridad por parte de los usuarios y de los proveedores para la recuperación en caso de desastres.

Los datos que se encuentran en los servidores de la *nube* deberían estar encriptados.

- *Conectividad.* El acceso a los recursos en la *nube* depende de la conexión a Internet, no siempre se puede asegurar que el nivel de conexión a la *nube* sea del 100% y esto puede dar lugar a la paralización de una empresa durante un tiempo indeterminado. También hay que tener en cuenta posibles interrupciones en el suministro eléctrico.

El proveedor a través de técnicas de redundancia, deslocalización de datos entre otras deberá intentar asegurar el 100% de disponibilidad de sus servicios en la *nube*.

Además el incremento de las necesidades de ancho de banda puede provocar una subida de los costes.

- *Falta de estandarización:* La no estandarización de los servicios ofrecidos por los diferentes proveedores de la *nube* hace que se incremente el riesgo de cautividad hacia un proveedor de servicios, perdiendo la posibilidad de poder migrar a otro proveedor de servicios que puede resultar de mayor interés a la organización de forma sencilla.

Tampoco hay una regulación firme sobre acuerdos de estandarización sobre la interfaz externa de *Cloud Computing* lo cual puede dificultar la migración de los datos.

Esta falta de estándares también supone un riesgo para los proveedores ya que comprometen sus inversiones y desde el punto de vista de los clientes se disminuye la competencia efectiva del mercado.

Existen iniciativas dirigidas a promover una *nube* abierta donde se creen determinados estándares que permitan a una organización elegir su proveedor libremente sin que le resulte una dificultad añadida.

- Dificultades de integración de determinadas aplicaciones que se encuentran geográficamente dispersas.
- *Identidad en la red.* Cada proveedor de servicios y cada aplicación tiene una manera diferente de identificar sus usuarios y de ver a que partes tienen acceso, esto puede crear dificultades a la hora de interconectar servicios de diferentes proveedores. Actualmente existe una aplicación estándar de identificación abierta y descentralizada, denominada *OpenId*, que permite a los usuarios conectarse a diferentes servicios usando una misma identidad digital.
- Otro riesgo que puede aparecer es la resistencia al cambio por parte de los departamentos de TI de aquellas organizaciones grandes que vean como su perfil y peso específico en la empresa puede ser reducido.

3.1.5.3. Aspectos Legales

Cuando una organización decide dar el paso y migrar su negocio en la *nube* debe tener en cuenta ciertas consideraciones legales que surgen dependiendo de la información que este manejando. Los problemas suelen surgir a partir de la deslocalización de los datos y la protección de los mismos según las leyes existentes en cada país.

El cliente deberá revisar el acuerdo legal a firmar con el proveedor de los servicios para no incurrir en posibles ilegalidades. Por norma general, los contratos que ofrecen los diferentes proveedores no suelen cubrir todas las responsabilidades y necesidades legales al respecto².

Los diferentes aspectos a tener en cuenta son:

- La protección de datos en la *nube* no es la misma que la existente en los servidores privados. Determinadas instituciones gubernamentales pueden acceder libremente a la información existente en la *nube*, sin necesidad de solicitar ninguna autorización mientras que para acceder a los servidores privados sí que la necesitan.
- También hay que tener en cuenta el lugar geográfico donde va a encontrarse la información en la *nube*. Existen determinados gobiernos que no permiten que los datos se encuentre en determinados países: Algunos ejemplos pueden ser:
 - El gobierno canadiense no permite que la información ubicada en servidores externos se ubique dentro del límite geográfico de Estados Unidos.

² <http://politube.upv.es/play.php?vid=47980>

- Según la legislación española, ciertos tipos de datos, como por ejemplo los datos bancarios, deben estar ubicados en España, no pueden estar ubicados en otro país.
 - Hay que ser muy cuidadoso con la ubicación de los datos y como se regula la protección de los mismos ya que, la legislación no es la misma en todos los países.
 - La deslocalización de los servidores puede dar lugar a transferencias internacionales de datos no autorizadas si se traslada la información a servidores ubicados en el extranjero o si se produce acceso a la información desde un país extranjero. Será necesario formalizar contratos para la transferencia internacional de los datos.
 - Esta deslocalización puede conllevar que determinados servicios se presten en países donde, por jurisdicción, está prohibido su uso o no se encuentra autorizado.
 - Se puede incurrir en una cesión no consentida de los datos por parte del proveedor a terceros, luego hay que asegurarse quienes van a ser las organizaciones que van a poder acceder a los datos y firmar cláusulas de confidencialidad con todos ellos.
 - Las organizaciones, están obligadas por ley a realizar copias de seguridad de determinada información con el objeto de impedir su pérdida. Habrá que establecer contratos con las organizaciones para que se encarguen de realizarlas.
 - Es necesario controlar quien realiza la custodia de los datos.
 - Se deben cumplir las LOPD del lugar al que pertenezca la información, se aplica la legislación del país de la organización que es responsable de los datos. El cliente deberá asegurarse que el proveedor del servicio se encuentra contractualmente obligado a proteger dicha información en los niveles que necesite el cliente.
 - Es necesario garantizar en el contrato los niveles de desempeño del servicio y los niveles de calidad necesarios. En ese escenario debe quedar claro cuáles son las garantías a las que se compromete contractualmente el proveedor del servicio con el fin de que alguna disminución de estos niveles no genere perjuicios en el cliente y sus negocios.
 - También será necesario asegurar las responsabilidades e indemnizaciones que los diferentes proveedores ofrecen ante posibles interrupciones del servicio o pérdidas de datos, ya que son temas que no suelen estar en los contratos y pueden causar grandes perjuicios a una organización
 - Otro tema que deben tener en cuenta las organizaciones a la hora de firmar con un proveedor es los aspectos relativos a la terminación del contrato, sin que queden cláusulas vagas a la hora de desprenderse de los datos sensibles. Así, debe quedar claro en el contrato cuánto tiempo tarda el proveedor del servicio en entregar la información, cómo se entrega al cliente esta información y en qué condiciones de seguridad y confiabilidad.
-

- El cliente debe prever que los contratos de *Cloud Computing* que realice contengan cláusulas arbitrales con mecanismos efectivos para solucionar cualquier tipo de controversia que surja del contrato o su terminación, deslocalizando al contrato para eliminar los riesgos jurisdiccionales.

Si la organización que suscribe los servicios en la *nube* es española se tiene que regir a las leyes españolas, por lo que deberá tener en cuenta que la información esté ubicada en lugares que la protección de datos sea igual. Hay que revisar muy bien los diferentes contratos que ofrecen los diferentes proveedores.

3.1.6. Recomendaciones migración a la *nube*

Uno de los atractivos más importantes de *Cloud Computing* es la oferta de servicios en la red que antes suponían una gran inversión inicial para las organizaciones que no estaban dispuestas a realizar o que no podían asumir.

Otro de los atractivos de trabajar en la *nube* es el liberarse de las responsabilidades que supone mantener el *hardware*, el *software*, las redes y el personal necesario para mantener todos los sistemas funcionando correctamente. Además el coste de una suscripción en la *nube* frente al coste de mantenimiento en el sistema tradicional es bastante importante y supone un gran ahorro para las organizaciones.

Pese a estas ventajas no siempre es recomendable que una organización traspase todo su entorno de trabajo a la *nube*. Antes de pasar sus servicios a la *nube* y elegir el proveedor que más le convenga, una organización debe tener en cuenta los siguientes factores:

- *Funcionalidad*: La solución del proveedor debe integrar una suite de negocio que sirva, casi en su totalidad, a toda la organización, personalizable y de gran disponibilidad.
- *Disponibilidad y calidad de servicio*: El proveedor debe asegurar una disponibilidad casi del 100% y calidad de servicio elevada de forma que no se produzcan cortes durante el acceso a la *nube*.
- *Tiempos de respuesta*: Una organización debe asegurarse que los niveles de servicio cumplan sus expectativas.
- *Seguridad y privacidad*.
- *Capacidad de recuperación y asistencia*: El proveedor debe asegurar que es capaz de recuperar datos ante posibles pérdidas y que estarán accesibles siempre aunque haya caídas en algunos de los servidores donde se alojen.
- *Capacidad de personalización*: Las aplicaciones deben permitir la personalización de la interfaz de usuario de forma que sean fáciles de manejar y puedan ofrecer imágenes corporativas.

- *Capacidad de integración:* Identificar las APIs de servicio que dispone el proveedor y ver si se pueden integrar con las aplicaciones propietarias de la organización.
- *Capacidad de adaptación en otros entornos.* Habría que estudiar si es sencilla la migración de los servicios y las aplicaciones desarrolladas para la *nube* de un proveedor a otro.
- *Mantenimiento de los sistemas por parte del proveedor,*
- *Alojamiento de los datos:* Saber donde estarán los datos físicamente para no incurrir en problemas legales.
- *Monitorización:* Averiguar si el proveedor tiene servicios de monitorización para que la organización pueda controlar sus propios sistemas.
- *Establecimiento de conexiones seguras a la nube*
- *Soporte que ofrece el proveedor.*
- *Aspectos contractuales:* Antes de formalizar un contrato con un proveedor, revisar detenidamente el contrato e incluir todos aquellos aspectos relativos a la protección de los datos, copias de seguridad, responsabilidades e indemnizaciones ante interrupciones de servicio o pérdida de datos...

Por otra parte, las empresas deberían plantearse la viabilidad de migrar sus aplicaciones a la *nube* y si realmente es rentable para ellos o no. También deberían mirar temas legales en cuanto al alojamiento y la protección de los datos.

Es mucho más sencillo que un particular o un pequeño negocio introduzcan sus sistemas en la *nube*. Para las grandes empresas la migración a la *nube* de todas sus aplicaciones puede resultar ardua.

Actualmente, la opción más viable por la que las empresas suelen optar es por la utilización de nubes híbridas donde los datos sensibles son almacenados dentro de una *nube* privada y los servicios que proporcionan son alojados en nubes públicas.

3.2. Análisis de Jugadores en la Nube

3.2.1. IBM

IBM es uno de los grandes proveedores de servicios en la *nube* y sus objetivos se centran principalmente dentro de los entornos de *SaaS* y *IaaS*.

3.2.1.1. LotusLive iNotes

Dentro del área de *SaaS*, *IBM* ofrece una versión online de *Lotus Notes* denominada *LotusLive iNotes*.

LotusLive iNotes es un servicio de email basado en web que proporciona servicios de mensajería y calendario a las organizaciones. Es especialmente útil para los usuarios que acceden a su correo de forma remota ya que solo necesita un navegador web y una conexión a Internet para acceder a *iNotes*.

LotusLive iNotes compite con *Google Gmail* y con *Microsoft Exchange Online*.

3.2.1.2. Smart Business Storage Cloud

Dentro del área de *IaaS*, *IBM* ofrece dentro de sus centros de datos un servicio de almacenamiento en la *nube*. Este servicio es denominado *Smart Business Storage Cloud* y proporciona espacio de disco escalable en la *nube*. El producto compite con *Amazon S3*.

IBM también ofrece sobre la *nube* *Compute-as-a-Service*, este servicio es denominado *Computing on Demand (CoD)* y proporciona recursos de computación sobre los centros de datos de *IBM*.

3.2.2. Salesforce

Salesforce.com inició su andadura en la *nube* en el año 2008 y es líder dentro de la plataforma *SaaS*. Fue la primera compañía en proporcionar un producto de éxito dentro de la *nube*.

3.2.2.1. Salesforce.com CRM

*Salesforce.com*³ proporciona un software de gestión de relaciones de clientes (CRM) en la *nube*. Las principales características de este producto incluyen campañas de marketing, gestión de líderes, información de seguimiento relativa a los contactos realizados. Todas las características se encuentran en el modelo creado para la *nube* y tanto las aplicaciones como los datos de clientes se encuentran alojados en los servidores de *Salesforce.com*.

³ <http://www.salesforce.com/es/>

3.2.2.2. Force.com

Dentro del entorno de *PaaS*, *Salesforce.com* proporciona un entorno de desarrollo denominado *force.com*⁴. Al igual que otras plataformas de desarrollo de *Cloud Computing*, *force.com* permite a los usuarios acceder a la plataforma de desarrollo de aplicaciones y de ejecución a través del navegador. Esto permite a las organizaciones desarrollar aplicaciones y productos sin necesidad de realizar grandes inversiones en *software* y *hardware* a dedicar al desarrollo de sus productos.

La plataforma de *force.com* incluye una base de datos relacional, un motor de flujos de trabajo y un interfaz de diseño de herramientas. La plataforma también incluye un IDE basado en *Eclipse* y un lenguaje propietario denominado *Apex* cuya sintaxis es similar a la ofrecida por *Java*. Sobre la plataforma de *force.com* no es posible ejecutar aplicaciones construidas en *Java* o *.Net*, los desarrolladores se ven obligados a usar *Apex* para construir las aplicaciones.

Force.com también proporciona una herramienta denominada *builder* para construir aplicaciones web de forma rápida. *Builder* proporciona una interfaz de usuario para crear objetos, campos dentro de objetos y relaciones entre campos. Una vez que el usuario ha creado los objetos, *builder* crea de forma automática una interfaz web con operaciones de creación, actualización y borrado de elementos. La plataforma también proporciona un entorno rico de generación de informes donde se pueden usar diferentes tipos de gráficos.

Además *Force.com* incluye un marco de trabajo de desarrollo de interfaces de usuario denominados *VisualForce* el cual, permite a los desarrolladores construir interfaces web para las aplicaciones estándar.

Los grandes obstáculos de *force.com* son el uso de un lenguaje propio no integrable con ningún otro y la falta de paquetes de integración.

Google se ha asociado con *Salesforce* para hacer más fácil que las compañías de todos los tamaños puedan ejecutar sus negocios en la *nube*. La alianza de *Salesforce.com* y *Google* ha creado la plataforma más ancha del mundo de *Cloud Computing* para construir y ejecutar aplicaciones.

3.2.3. Google

Google lanzó su plataforma de *Cloud Computing* en 2008, desde entonces, *Google* ha ofrecido servicios orientados principalmente a las capas de *Software como Servicio* y *Plataforma como Servicio*.

⁴ <http://www.salesforce.com/es/platform/>

3.2.3.1. Google Apps

Dentro de la capa de *Software como Servicio (SaaS)*, Google ha creado *Google Apps*⁵. Dependiendo del perfil de acceso (individual, formación – estudiantes, o negocio) pueden variar las características de los productos ofrecidos. Google ha sido pionero este tipo de servicios dentro de la *nube*.

Proporciona diversas aplicaciones web con funcionalidades similares al software ofimático y además permite a los usuarios comunicarse de forma fácil y eficiente. Todas las aplicaciones son mantenidas en la web y se accede a ella a partir de navegadores y los usuarios pueden acceder a sus cuentas desde cualquier ordenador conectado a Internet sin necesidad de instalar nada localmente.

La información en esta plataforma está dividida en múltiples servidores y discos, lo que hace complicado que sea leída si alguien entrase a único servidor.

Los servicios proporcionados por Google dentro de esta estructura son:

- Aplicaciones de mensajería dentro de las cuales se encuentran *Google Gmail*, *Google Talk*, *Google Groups* y *Google Calendar*. Estas aplicaciones son ofertadas tanto para usuarios individuales como para organizaciones o entornos educativos. El precio varía dependiendo del grupo en el que se encuentren los usuarios.
 - *Google Gmail*: Ofrece a los usuarios alojar sus correos electrónicos sobre los servidores de Google y además, proporciona una interface para acceder a ellos. El servicio enfocado a las organizaciones proporciona también la adaptación del entorno de trabajo al formato corporativo de la empresa y soporte de atención 24 horas del día. Provee herramientas para migrar desde *Lotus Notes* o *Microsoft Exchange*.
 - *Google Talk*: Servicio de mensajería instantánea y de llamadas de voz.
 - *Google Calendar*: Es un navegador basado en una agenda. Dispone de múltiples vistas (diaria, semanal, mensual) y pueden visualizarse múltiples calendarios. También ofrece la posibilidad de compartir calendarios con otros usuarios. Este servicio es accesible desde dispositivos móviles y permite la sincronización con *Microsoft Outlook*.
 - *Google Groups*: Proporciona a los usuarios alojar foros de discusión. Los mensajes se alojan en los centros de datos de Google.
- Aplicaciones de colaboración dentro de las cuales se encuentran, entre otras, *Google Docs*, *Google Sites*, *Google vVdeos* para empresas. Este grupo de servicios también está disponible para usuarios individuales, entornos educativos y organizaciones.

⁵ <http://www.google.com/apps/intl/es/business/index.html>

- *Google Docs*: Software de construcción de documentos basado en web. Se pueden construir tanto documentos de texto, hojas de cálculo, presentaciones. Permite que varios usuarios accedan al mismo documento y puedan editarlo y modificarlo a la vez. También es posible visualizar el histórico de cambios de los documentos. Presenta la limitación de que no se puede acceder a *Google Docs* sin estar conectado a Internet. Los usuarios puede continuar trabajando en sus documentos desde cualquier ubicación.
- *Google Sites*: Permite la creación y personalización de páginas web y blogs.
- Otras aplicaciones de *Google* disponibles en la web para todo tipo de usuarios tales como *Google Reader*, *Blogger*, álbumes web de *Picasa*, *AdWords* ...
 - *Picasa*: Es una herramienta gratuita de *Google* para editar imágenes donde se permite subir y compartir imágenes y videos que se encuentran sobre los centros de datos de *Google*.
- *Google* también ofrece a las organizaciones y a los entornos educaciones funciones propias tales como 25GB de almacenamiento de correo electrónico por usuario, interoperabilidad con *Blackberry* y *Microsoft Office* entre otras.
- También proporciona un servicio de asistencia las 24 horas del día, los 7 días de la semana y garantiza su tiempo de actividad el 99.9% tanto a empresas como a entornos educacionales.

Google Apps cuenta con varias características importantes:

- Proporciona un manejable panel de control que facilita al usuario las tareas de administración tales como la gestión de cuentas, la personalización de interfaces, la activación y desactivación de aplicaciones.
- Aunque la información se encuentre alojado en los servidores de *Google*, los usuarios pueden personalizar el dominio de sus correos, sus páginas web con los colores corporativos de su empresa...
- Los administradores pueden integrar estos servicios con plataformas existentes así como extender las funcionalidades del núcleo de aplicaciones con la API que *Google* ofrece. Existen APIs para suministro, para creación de informes, migración, manipulación de datos e integración con sistemas de SSO.

3.2.3.2. Google App Engine

Dentro de la capa de la *Plataforma como servicio (PaaS)*, *Google* ha desarrollado *Google App Engine*⁶. *Google App Engine* es una plataforma de desarrollo alojada en la *nube* de *Google*. Todo el *hardware*, sistemas operativos y *software* requerido es proporcionado por el proveedor a los

⁶ <http://code.google.com/intl/es/appengine/>

clientes, por lo tanto, los clientes únicamente se tienen que centrar en la creación o instalación de sus propios servicios mientras que *App Engine* ejecuta el servicio sobre los servidores de *Google*.

App Engine está muy limitado en cuanto a los lenguajes que pueden ser utilizados para construir los servicios. Inicialmente el lenguaje de desarrollo soportado fue *Python* pero actualmente también soporta *Java*. Si el usuario no está familiarizado con ninguno de los lenguajes soportados, el cliente tiene que aprenderlos antes de construir sus propios servicios, además existen aplicaciones que no pueden ser simplemente colocadas sobre *App Engine*, sólo los servicios escritos completamente en *Java* y *Python* son soportados. El despliegue del código y su ejecución es gestionado por *Google*.

Google App Engine permite a las aplicaciones alojadas que se escalen dinámicamente según el tráfico existente y la capacidad de almacenamiento se incremente o decremente según se requiera. La arquitectura que dispone *App Engine* es notable ya que permite el auto escalado en tiempo real sin virtualización para muchos tipos de aplicaciones web, el balanceo de la carga y la ejecución distribuida de aplicaciones. Este auto escalado es dependiente del desarrollo y usa un conjunto limitado de las APIs nativas de cada plataforma y en algunas instancias, será necesario especificar APIs de *Google* como *URLFetch*, *DataStore* en lugar de las llamadas nativas. Por ejemplo, Una aplicación de *App Engine* no podrá escribir directamente sobre el sistema de ficheros si no que deberá usar *Google Datastore* o abre un socket o acceder a través de un servicio de *Google*.

Los servicios de datos son proporcionados a través de *Google Datastore* que es una base de datos distribuida no relacional y puede ser accedida a través de GQL que es un lenguaje similar a SQL.

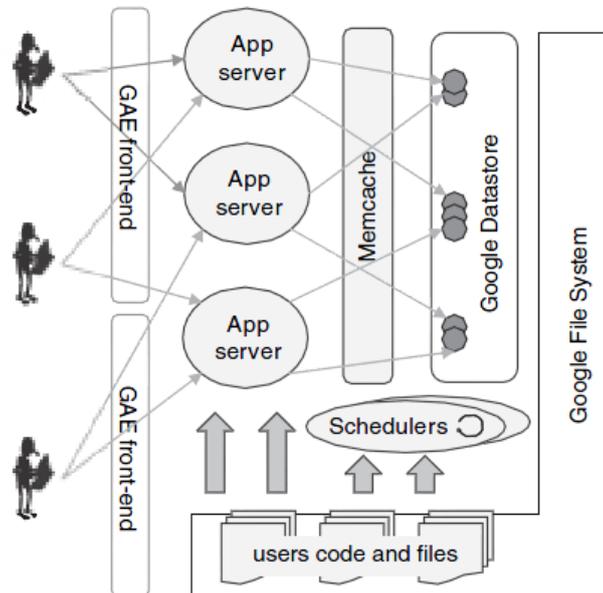


Figura 8. Google App Engine.

El uso de recursos para una aplicación desarrollada en esta plataforma es medido en términos de solicitudes web servidas y horas de CPU consumidas ejecutando solicitudes o procesos de *batch*.

El despliegue de aplicaciones en *Google* es gratuito, esto es debido a que las aplicaciones no se están ejecutando sobre máquinas virtuales dedicadas. Las aplicaciones son servidas por un gran número de servidores web que se encuentran dentro de los centros de datos de *Google* y que ejecutan las solicitudes de usuarios finales de cualquier parte del mundo. El servidor web carga el código desde los servidores de ficheros de *Google* en la memoria y sirve las diferentes solicitudes. Cada solicitud de una aplicación particular es servida por cualquiera de los servidores web de *Google App Engine*. No hay garantías de que un mismo servidor sirva dos veces la misma solicitud.

Mientras que la arquitectura es capaz de garantizar el escalado natural y el incremento de la carga, también significa que el código de las aplicaciones no puede confiar en los datos que tiene en la memoria. Una caché distribuida denominada *Memcache* está disponible y dirige parcialmente el problema, en particular, en las sesiones HTTP, de forma que si se ha realizado una solicitud sobre la misma sesión y se ha realizado en diferentes servidores, pueda traer los datos de la sesión (al menos la mayoría de las veces).

Google App Engine permite a los usuarios construir aplicaciones web básicas rápidamente. La configuración e instalación de la aplicación es rápida y sencilla. Sin embargo, en la actualidad existen diversos problemas que preocupan a la comunidad de desarrolladores de *Google App*

Engine sobre el desarrollo de aplicaciones para organizaciones de gran escala y plataformas de alojamiento⁷.

3.2.3.3. Google DataStore

Los datos de las aplicaciones persisten en *Google Datastore*. *Datastore* permite que las aplicaciones definan tipos estructurados (*kinds*) y almacenen sus instancias (*entities*) de forma distribuida sobre el sistema de ficheros de GFS. Aunque existen importantes diferencias entre los modelos de datos relacionales y *Datastore*, los tipos estructurados (*kinds*) pueden ser vistos como tablas estructuradas y las entidades como registros. En este sistema, todas las entidades de un tipo de datos no tienen por qué tener las mismas características, esto hace que sea bastante flexible a la hora de almacenar conjuntos de datos.

Google Datastore permite consultas simples con condiciones a partir del uso de GQL que, esencialmente, sigue las mismas bases que SQL pero con ciertas restricciones.

Google Datastore es un almacén de datos distribuidos donde los objetos (*entities*) de todas las aplicaciones de GAE son mantenidos usando un gran número de servidores y sistemas distribuidos de ficheros. Es importante asegurar que a pesar de compartir un esquema de datos distribuidos con muchos otros usuarios, los datos de la aplicación son recogidos eficientemente y actualizados de forma atómica. *Google Datastore* proporciona un mecanismo para agrupar entidades de diferentes *kinds* de forma jerárquica.

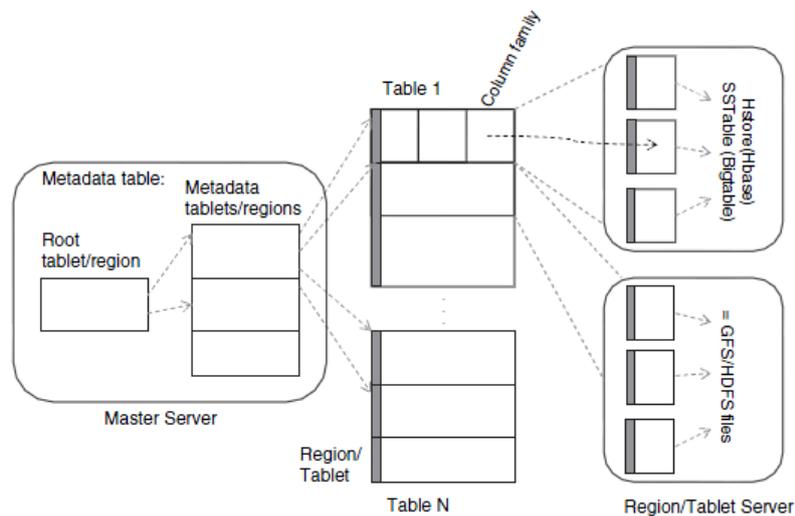


Figura 9. Arquitectura de Google DataStore.

3.2.3.4. Google Cloud SQL

Google ha creado su propia Base de Datos relacional para su plataforma de desarrollo tal y como se cita en el siguiente artículo⁸ de la revista *Cloud Computing*.

⁷ <http://aralbalkan.com/1504>

⁸ <http://www.revistacloudcomputing.com/2011/10/google-crea-su-base-de-datos-para-la-nube/>

“Google ha desarrollado Cloud SQL, una base de datos relacional para su plataforma de desarrollo y alojamiento de aplicaciones basada en cloud *App Engine*.

Así, se dispone de un integrado en *Google App Engine* de un servicio de datos de alta disponibilidad. Entre los beneficios de esta nueva base, cabe destacar que es de alta escalabilidad y disponibilidad, totalmente familiar a *MySQL* y muy intuitiva.

Además, el servicio incluye una funcionalidad de importación y exportación que nos permita trasladar nuestras bases de datos *MySQL* a *Google Cloud SQL*. Actualmente, esta base de datos de *Google* para *Cloud Computing* se halla en fase beta y gratuita para posteriormente pagar una cuota mensual.”

*Google Cloud SQL*⁹ permite disponer de un servicio de datos de alta disponibilidad integrado con *Google App Engine* donde los datos se replicarán de forma sincronizada en múltiples centros de datos, escalándose de forma automática. Para conectarse desde una aplicación se usará JDBC para *Java* y DB-API en *Python*.

Para administrar las diferentes bases de datos existe una interfaz gráfica y además, permite exportar bases de datos de *MySQL* a la *nube* usando *App Engine*.

3.2.3.5. Google Web Toolkit

*Google Web Toolkit*¹⁰ junto con el complemento de *Google* para *Eclipse* permite a los desarrolladores crear aplicaciones sobre sus ordenadores personales y a continuación desplegarlos y depurarlos dentro de la *nube*. Esta utilidad se podría encontrar dentro de los servicios de S+S.

Los desarrolladores pueden desarrollar y depurar aplicaciones web usando el lenguaje de programación *Java* y después pueden desplegarlos con una alta optimización en *JavaScript*. Con esta herramienta los desarrolladores evitan muchas horas intentando hacer compatible la aplicación con otros navegadores, introduciendo AJAX de forma que se producen ganancias de rendimiento y productividad.

Incluye las capacidades de *Java 5* el cual introduce tipos enumerados, anotaciones, *auto boxing*, listas de parámetros variables... El compilador de *Google Web Toolkit 1.5* produce código más rápido, mejorando el rendimiento que los usuarios finales notan. También proporciona un conjunto de librerías muy amplio que ayuda a los desarrolladores a construir clases de AJAX, lanzar test, librerías reusables para el diseño de interfaces, estructuras de datos, comunicación cliente servidor, internacionalización, testeo, accesibilidad.

⁹ <http://code.google.com/intl/es-ES/apis/sql/>

¹⁰ <http://code.google.com/intl/es-ES/webtoolkit/>

3.2.4. Amazon

Amazon inició su oferta dentro del mundo de *Cloud Computing* en el año 2006 y es pionera y líder de mercado dentro del área de *IaaS*. El conjunto de productos de Amazon dentro de la *nube* es denominado *Amazon Web Services*¹¹ (AWS).

Existen importantes diferencias entre la *nube* de Amazon y los tradicionales proveedores de alojamiento y son las que se enumeran a continuación:

- El grado de automatización disponible para los usuarios finales se ofrece como un servicio web para controlar el número de instancias virtuales que se están ejecutando en cualquier momento.
- La capacidad que se ofrece a los usuarios de paquetizar y guardar sus propias configuraciones de máquinas virtuales como imágenes de máquinas de Amazon (AMI)
- El pago de los servicios es por hora de uso real del sistema frente al pago de cuotas mensuales o anuales de los servicios típicos de *hosting*.

Amazon dispone un gran número de productos y servicios que permite a muchos usuarios de la *nube* construir complejas aplicaciones sin tener que confiar el despliegue y la configuración de sus servicios a tradicionales *middleware* y a productos de bases de datos. Algunos de los servicios más importantes se describen en el siguiente apartado.

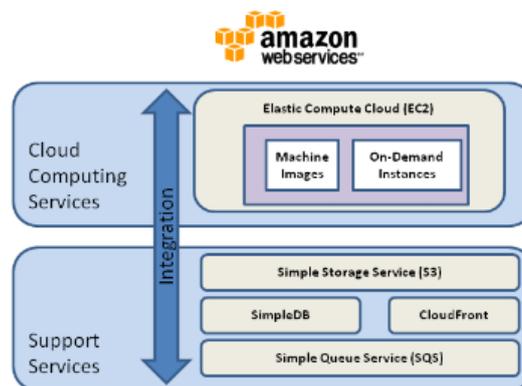


Figura 10. Modelo de *Cloud Computing* de AWS.

3.2.4.1. Arquitectura y funcionamiento de los servicios de Amazon

La *nube* de Amazon proporciona una *Infraestructura como Servicio (IaaS)* por lo tanto, la infraestructura de computación como servidores, dispositivos de almacenamiento o puntos finales de red de una capacidad deseada son virtualmente proporcionados en pocos minutos a través de una consola de gestión automatizada basada en web. Este servicio es denominado *Elastic*

¹¹ <http://aws.amazon.com/es/>

Compute Cloud (EC2). EC2 es uno de los servicios ofrecidos por la nube de Amazon pero es habitual usarlo para describir la oferta completa de la nube.

La siguiente figura se muestra la infraestructura de la nube de Amazon desde el punto de vista del usuario.

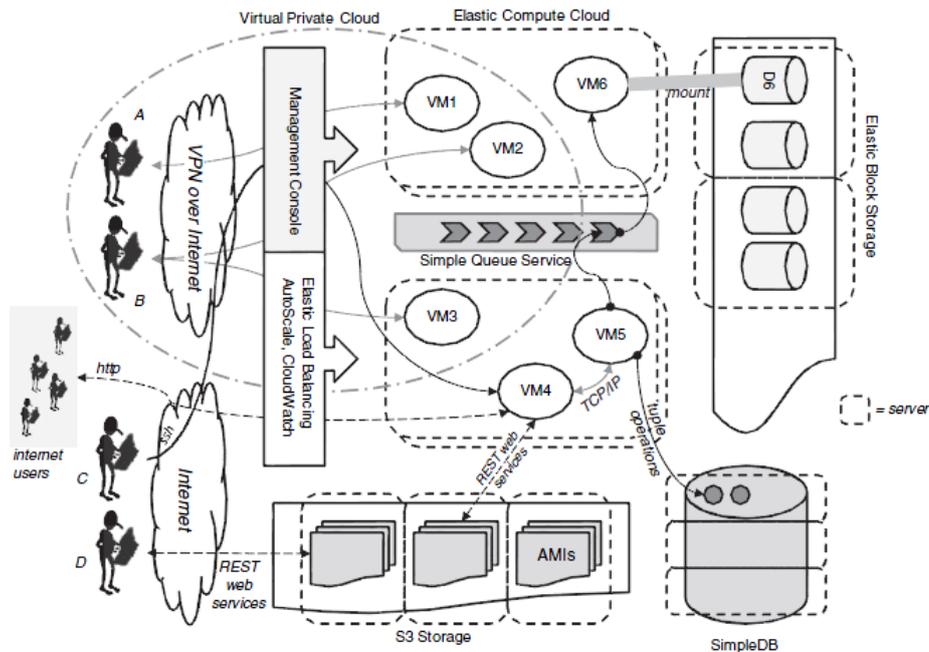


Figura 11. Infraestructura de Amazon.

Los servicios son implementados sobre una gran red de servidores. El servicio EC2 proporciona a los usuarios acceso a las máquinas virtuales dedicadas que tienen la capacidad deseada por el usuario y que son suministradas sobre la red de servidores físicos cuyos detalles (ubicación, capacidad...) son transparentes al usuario final.

A través de la consola de gestión, el usuario genera pares de claves PKI (claves RSA y certificados X.25) los cuales serán usados para conectarse de forma segura a los diferentes servidores virtuales a través de Internet. Los usuarios pueden elegir entre diferentes modelos de máquinas virtuales (AMI) desde las cuales suministrar sus servicios. Todas las AMIs son almacenadas sobre el Amazon Simple Storage Service (S3) y se usan para iniciar la configuración deseada de los servidores virtuales.

Los usuarios de la nube tienen permisos de administrador sobre sus servidores y por lo tanto tienen un control total sobre ellos. Por ejemplo, puede desarrollar aplicaciones y hacerlas públicas sobre Internet. Los usuarios pueden acceder a diferentes servidores a través de la red interna de la nube.

Otro de los servicios proporcionados en Amazon es Amazon SimpleDB que dispone que un objeto sea almacenado siendo identificado por un par clave-valor y que permitirá actualizarlo y recuperarlo de forma eficiente. SimpleDB no es una base de datos relacional. Los servidores

virtuales también pueden utilizar sistemas de bases de datos relacionales que vendrán preinstalados como partes de una AMI o habrán sido instalados por los usuarios sobre el servidor.

Es importante remarcar que los servidores virtuales no tienen un sistema de almacenamiento persistente, por lo que cualquier dato sobre el sistema de ficheros se perderá cuando se apague el servidor virtual. Para almacenar datos de forma persistente se necesita configurar *Elastic Block Storage (EBS)* sobre el servidor virtual. EBS mantiene la persistencia de los datos a través de todos los usuarios sobre grandes conjuntos de servidores de datos. Una vez que los servidores han sido reiniciados, se deberá adjuntar los datos de usuario desde los EBS como un volumen de almacenamiento lógico montado como un disco.

Otro servicio disponible de Amazon es el *Simple Queue Service (SQS)*. El SQS es una cola de mensajes persistentes que es útil para el almacenamiento temporal de los datos que necesitan ser obtenidos para ser procesados por un servidor. SQS es normalmente usado para gestionar transferencias de datos asíncronas entre servidores de procesos que están orientados a flujos de trabajo en *batch*.

Los datos en S3 pueden ser ficheros de cualquier tipo (generalmente *blob – binary large object*). Los usuarios acceden y modifican los objetos que se encuentran en S3 vía URI usando servicios web REST. Los objetos almacenados en S3 son accesibles a través de internet y a través de los servidores virtuales que se encuentran en la *nube*. S3 es especialmente útil para almacenar grandes colecciones de datos no estructurados que necesitan estar disponibles para muchas aplicaciones clientes. Todos los datos almacenados aquí se encuentran replicados al menos tres veces para asegurar la tolerancia a fallos.

El almacenamiento en S3 también es usado para almacenar imágenes de máquinas virtuales definidas por los usuarios. Estas imágenes pueden estar disponibles a otros usuarios.

Un objetivo importante de cualquier servicio web es aislar a los usuarios de la demanda variables gestionando de forma automática la escalabilidad de los recursos reservados en las aplicaciones de la *nube*.

En *Amazon EC2* el usuario necesita definir explícitamente una arquitectura que permita escalar los recursos usando las herramientas de gestión proporcionadas por *Amazon*. Parámetros de rendimiento, ciclos de CPU, la información de entrada/salida puede ser monitorizada desde la *nube (Amazon Cloud Watch)*, estos datos podrán ser usados para añadir o borrar servidores virtuales de un clúster de aplicaciones (*Amazon Auto Scale*). *Elastic Load Balancing* permite a un grupo de servidores ser configurados dentro de un conjunto a través de los cuales las solicitudes de entrada (conexiones HTTP) se irán balanceando. Las estadísticas de rendimiento también podrán ser monitorizadas por *Amazon Cloud Watch* y tratadas por *Auto Scale* para añadir o quitar servidores del conjunto de servidores que se usan para balancear la carga. Usando estas herramientas, los usuarios pueden configurar una arquitectura escalable que puede ser ajustado en función de los recursos consumidos, CPU.

Las empresas que contratan los servicios en la *nube*, también requieren que los servidores virtuales se encuentren protegidos y usaran las mismas políticas de salvaguarda que aplican a cualquiera de sus recursos sobre sus centros de datos.

Amazon EC2 proporciona un servicio privado virtual en la *nube*, por el que los servidores virtuales pueden ser conectados a la red interna de una organización a través de VPN. Los servidores tendrán una dirección IP privada controlada por el centro de operaciones de la organización. El acceso a estas máquinas puede ser controlado por el mismo conjunto de políticas, *firewalls* y *proxies* que el resto de la organización.

3.2.5. Microsoft

Microsoft, como todos los grandes proveedores de *software* tiene marcado el *Cloud Computing* como su gran objetivo.

Microsoft lanzo sus servicios en la *nube* en el año 2009 y, al igual que *Google*, ha ofrecido servicios orientados principalmente a las capas de *Software como Servicio* y *Plataforma como Servicio*. *Microsoft* fue el proveedor que más tarde entró en el mundo de *Cloud Computing* y ya se le puede considerar líder en este tipo de servicios.

3.2.5.1. Office 365

*Office 365*¹² es la herramienta que *Microsoft* ofrece dentro de la capa de SaaS y une los servicios que antes ofrecía dentro de *Office Web Apps* y *Microsoft Business Productivity Suite*. *Office 365* integra las herramientas que ofrece online con las herramientas instaladas en el escritorio.

Los servicios que se ofrece desde *Office 365* son:

- *Office Profesional Plus*: Versión de *Microsoft Office* suministrada como servicio flexible, contiene las aplicaciones más recientes de *Office* y *Office Web Apps*.
- *Exchange Online*: Gestor de correo electrónico, calendario y contactos. Es la versión online de *Outlook*. Los correos se encuentran alojados en los servidores de *Microsoft* en la *nube*. *Microsoft* garantiza una disponibilidad del 99.9% e incluye un paquete de copias diarias y pre-escaneo de mensajes.

Los servidores de *Exchange Online* están dispersados geográficamente y tienen como objetivo facilitar las tareas de gestión de los departamentos de TI evitando la necesidad de configurar, desplegar y monitorizar soluciones de correo dentro de la organización.

- *SharePoint Online*: Es un repositorio documental donde todos los documentos que se encuentran en él son accesibles para el resto de usuarios dependiendo de los permisos que tengan concedidos. Se pueden crear calendarios y definir flujos de

¹² <http://www.microsoft.com/es-es/office365/online-services.aspx>

trabajo que deben cumplir los documentos. También se permite crear la página web de una organización.

Puede escalarse a multitud de sitios dentro de una organización, soporta carga balanceada y *clústeres* de bases de datos.

- *Lync Online*: Herramienta que facilita la mensajería instantánea, comunicaciones de vídeo y voz y la posibilidad de enviar archivos entre usuarios.
- *Office Web Apps*: Versiones en línea de *Microsoft Word*, *Microsoft Excel*, *Microsoft Power Point* y *Microsoft OneNote* que permiten ver y editar documentos directamente desde un navegador web.

3.2.5.2. Windows Azure Platform

Dentro de la capa de *Plataforma como servicio (PaaS)*, *Microsoft* ha desarrollado *Windows Azure Platform*¹³. *Azure* proporciona un sistema operativo y una plataforma de desarrollo para acceder y desarrollar aplicaciones sobre los centros de datos de *Microsoft*.

Windows Azure Platform es una plataforma interoperable, que permite el desarrollo de aplicaciones en diferentes lenguajes así como la comunicación con cualquier entorno externo. *Visual Studio* es la herramienta con la que los desarrolladores crean las aplicaciones a través de todos los lenguajes que soporta (C#, VB, ASP...). Una extensión de *Azure* al conjunto de herramientas estándar permite que las aplicaciones puedan ser desarrolladas y desplegadas en la *nube* de *Microsoft*. Además dispone de otras herramientas y SDKs para que puedan desarrollarse aplicaciones no nativas de *Microsoft*.

. La plataforma *Windows Azure* está compuesta por una serie de bloques que son utilizados por los desarrolladores para crear y alojar servicios dentro de la *nube*. Los desarrolladores pueden hacer uso de todos los componentes que se ofrecen o sólo de aquellos que son necesarios para la creación de sus aplicaciones. Cada uno de los bloques que la plataforma desempeña una función diferente.

La plataforma de *Windows Azure* es un conjunto de servicios basados en la *nube* que permiten que:

- Los desarrolladores utilicen sus habilidades actuales y sus herramientas conocidas para desarrollar aplicaciones *cloud*.
- Los ISVs y los integradores de sistemas accedan al mercado con rapidez.
- Los administradores de TI obtengan acceso a un nuevo conjunto de recursos sin agregar complejidad.
- Las empresas de cualquier tamaño sean capaces de responder con rapidez a medida que cambian las necesidades comerciales.

¹³ <http://www.microsoft.com/windowsazure/>

- En *Windows Azure* está soportado el desarrollo directo sobre la plataforma *.net* pero también con *PHP, Java, Rubi* desde diferentes entornos tales como *Eclipse*.

Azure ofrece grandes similitudes con *App Engine* en la forma de desarrollar y desplegar aplicaciones dentro de la *nube* y, también posee similitudes con el entorno de *Amazon* en cuanto al uso de la virtualización, el control del usuario sobre la reserva de servidores virtuales para una aplicación y el control de usuario sobre la elasticidad de la plataforma. Pero, a diferencia de *Google* y *Amazon*, *Microsoft* proporciona servicios de almacenamiento relacional.

En la figura que se muestra a continuación se puede ver la estructura de la plataforma *Azure* desde el punto de vista del usuario.

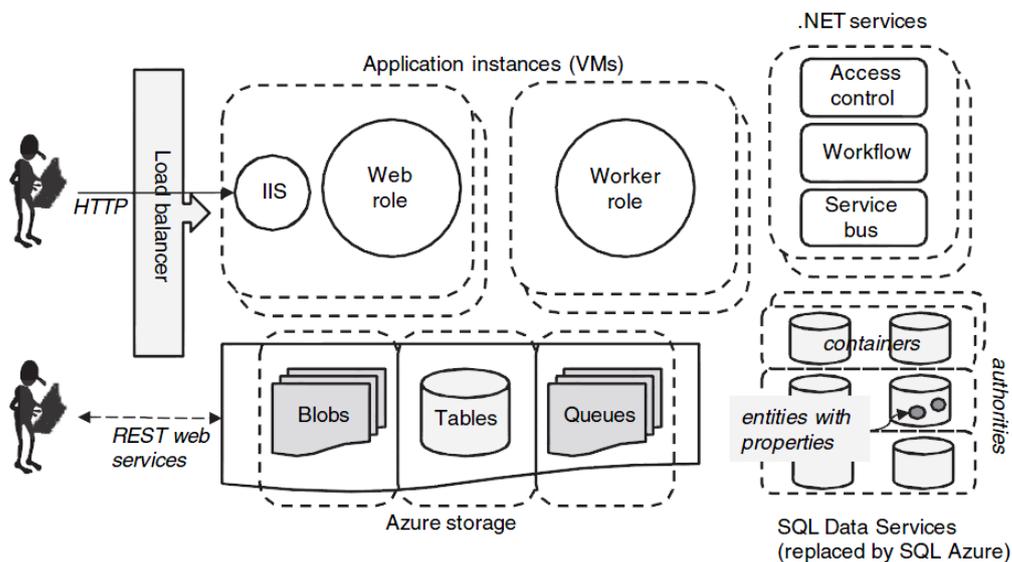


Figura 12. Plataforma Windows Azure.

Microsoft Azure disfruta de las ventajas de que existen muchas aplicaciones en las empresas que usan como base la plataforma de *Microsoft*. Desde que *Azure* está disponible su popularidad se ha incrementando exponencialmente llegando a ser uno de los proveedores más importantes.

Los diferentes componentes que forman la plataforma *Windows Azure* se ejecutan en centros de datos que están localizados alrededor de todo el mundo:

- Norte América: Norte y Centro (Chicago, IL), Centro y Sur (San Antonio, TX).
- Asia: Este de Asia (Hong Kong, China), Sureste de Asia (Singapur)
- Europa: Norte de Europa (Ámsterdam, Holanda) y Oeste de Europa (Dublín, Irlanda).

Cada centro de datos tiene servidores agrupados dentro de contenedores, donde cada contenedor tiene entre 1800 y 2500 servidores. Los centros de distribución de contenidos (CDN) donde se encuentran las copias de los servidores se distribuyen a lo largo de 24 países.

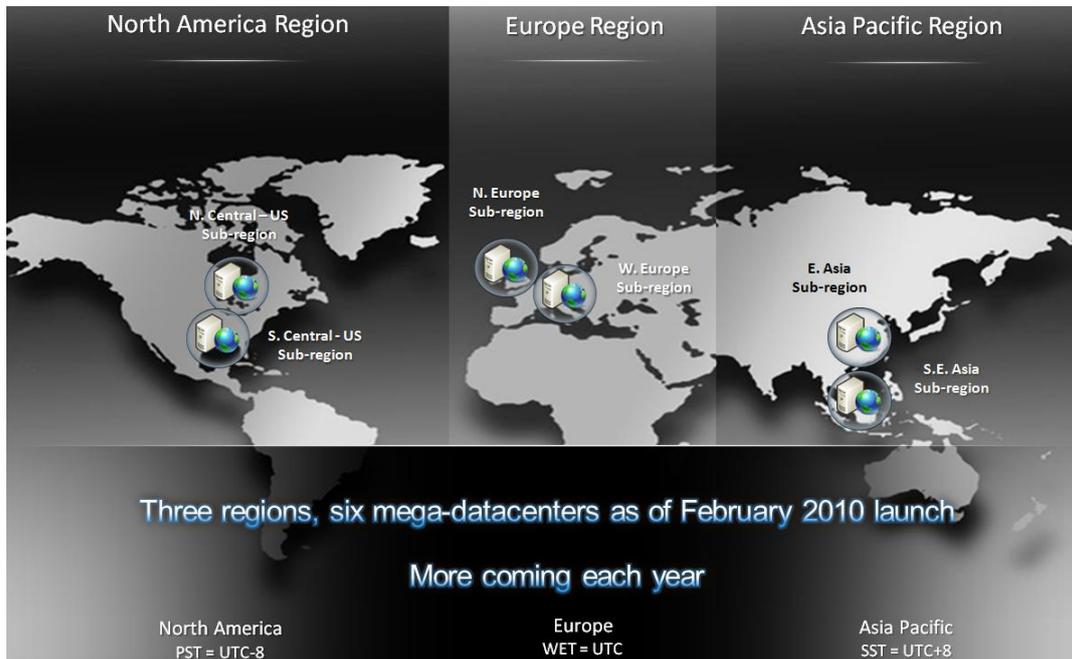


Figura 13. Distribución Servidores Plataforma Windows Azure (2010).

Los desarrolladores que usan la plataforma pueden controlar en que centros de datos quieren que sus aplicaciones sean ejecutadas y sus datos almacenados, dando la posibilidad de situarlos lo más cerca posible de los usuarios.

La plataforma de *Windows Azure* está dividida en cuatro grandes partes:

- *Windows Azure*: Ofrece un entorno de *Windows* para ejecutar aplicaciones y almacenar datos en los servidores que *Microsoft* aloja en sus centros de datos.
- *SQL Azure*: Servicio de datos relacional en la *nube* basado en *SQL Server*.
- *Windows Azure AppFabric*: Ofrece servicios *Cloud* para conectar aplicaciones que se ejecutan en la *nube* o de forma interna.
- *Windows Azure Marketplace*: Un servicio online para comprar de datos y aplicaciones basados en la *nube*.

La siguiente figura muestra la *nube* y todos sus componentes.

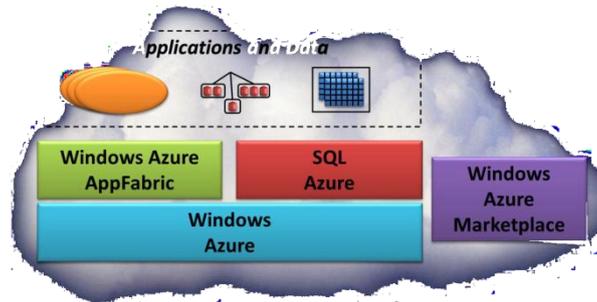


Figura 14. Distribución de componentes en la *nube*.

A continuación se explican cada una de las tecnologías que forman parte de la plataforma de *Windows Azure*.

3.2.5.2.1. *Windows Azure*

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* y hace disponible a los desarrolladores un entorno de computación bajo demanda y almacenamiento alojado en los centros de datos de *Microsoft* para aplicaciones web.

Es un entorno de ejecución que incluye servidores web, servicios computacionales, sistemas de almacenamiento básicos, gestión de colas, gestión de servicios y almacenamiento básico.

Sobre *Windows Azure* se pueden utilizar tantos los conocimientos de .NET como de otros lenguajes ajenos a esta plataforma como por ejemplo PHP, C/C++, Java. *Windows Azure* es una plataforma abierta a lenguajes que no son nativos de *Microsoft* mediante el uso de estándares.

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 basada en *framework .NET 4.0* 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* permite la ejecución de servicios *Windows Communication Foundation (WCF)* y llamar a código nativo desde las aplicaciones *Azure* basadas en NET.

Windows Azure también proporciona servicios de almacenamiento no relacional y colas con acceso no autenticado que son accesibles mediante una interfaz *REST* que se puede utilizar desde cualquier lenguaje que tenga la capacidad de realizar peticiones *HTTP*.

A continuación se puede ver cuál es la arquitectura básica de *Windows Azure*.

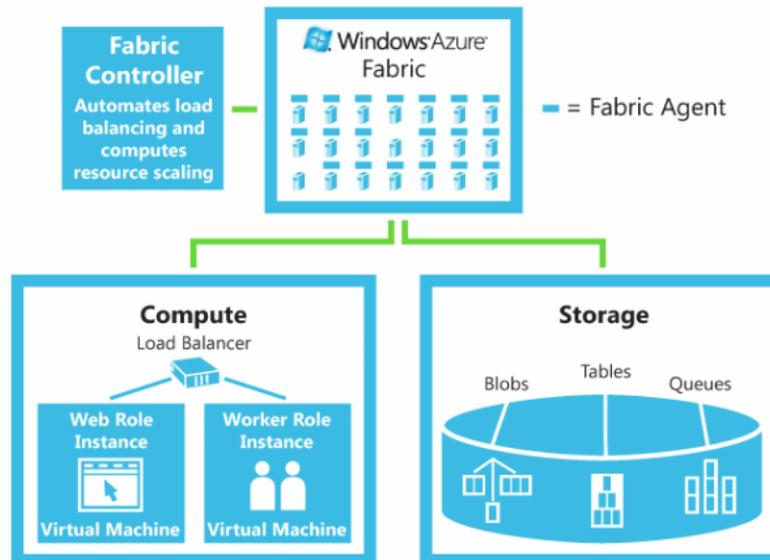


Figura 15. Arquitectura de la infraestructura *Windows Azure*.

Seguidamente se explican las cinco partes de las que consta *Windows Azure*.

Servicio de Computación o Proceso (Compute Services)

Es la forma en la que interactúan todos los componentes de la plataforma *Azure*. Ofrece una alta disponibilidad y se encarga de distribuir el trabajo entre las diferentes máquinas virtuales y de levantar tantas instancias como se considere necesario para dar la mejor respuesta posible. Cada instancia es una máquina virtual que se encuentra aislada del resto de clientes.

Se ejecuta sobre *Internet Information Server (IIS)* versión 7 que se encuentra habilitado para las aplicaciones web de ASP.NET. Los desarrolladores pueden escribir tanto servicios gestionados como no gestionados, sin tener que preocuparse de la infraestructura de los sistemas operativos donde alojan los servicios.

Una aplicación de *Windows Azure* utiliza recursos de computación a través de uno o varios componentes denominados *roles*. Los roles pueden ser de tres tipos:

- **Web Role:** Esta dedicado principalmente a ejecutar aplicaciones basadas en web. Cada instancia de rol Web tiene pre-configurado un IIS para ejecutarse dentro de él, por lo que la creación de aplicaciones con ASP.NET, WCF u otras tecnologías web es sencilla. También es posible crear aplicaciones en PHP, Java y otras tecnologías que no sean de Microsoft.

Los protocolos de entrada que soporta son HTTP y HTTPS a través de los puertos 89 y 443 y los protocolos de salida pueden ser cualquiera de los protocolos de TCP.

Se encuentran balanceadas entre las diferentes máquinas virtuales de la plataforma.

- *Worker Role*: Diseñado para ejecutar una gran variedad de código. Las aplicaciones alojadas dentro de roles de trabajo pueden ejecutarse de forma asíncrona, a lo largo de un periodo continuado de tiempo de forma independiente a las interacciones o a los datos de los usuarios. Puede exponer puntos extremos internos y externos y realizar llamadas a interfaces externas. También puede comunicarse con los servicios de almacenamiento que posee *Windows Azure*.

El uso más común de los roles de trabajo es hospedar procesos en segundo plano que se encuentran por debajo de los roles web. Es muy común que una aplicación interactúe con los usuarios a través de un rol web y que el rol de trabajo se ocupe del procesamiento. El rol web usa el sistema de almacenamiento en cola para enviar solicitudes al rol de trabajo y éste comprueba la cola de forma periódica y procesa las solicitudes. La salida del proceso se hace disponible a través de los diferentes dispositivos de almacenamiento que ofrece *Windows Azure*. Estas aplicaciones no pueden ser accedidas desde redes externas pero pueden realizar solicitudes externas de tipo *HTTP* o a otras instancias de tipo *worker role*.

Aunque ambos roles se encuentre debajo del mismo servicio una instancia de un rol de trabajo es independiente de una instancia donde se ubica un rol web.

- *Virtual Machine role*: Permite desplegar una imagen de un servidor personalizado *Windows Server 2008 R2* sobre *Windows Azure*. Este rol puede ser usado cuando la aplicación desarrollada requiera una elevada personalización del sistema operativo y no puede ser automatizada. Permite un control total sobre el entorno de la aplicación y permite migrar aplicaciones ya existentes a la *nube*. Todavía se encuentra en su versión Beta.

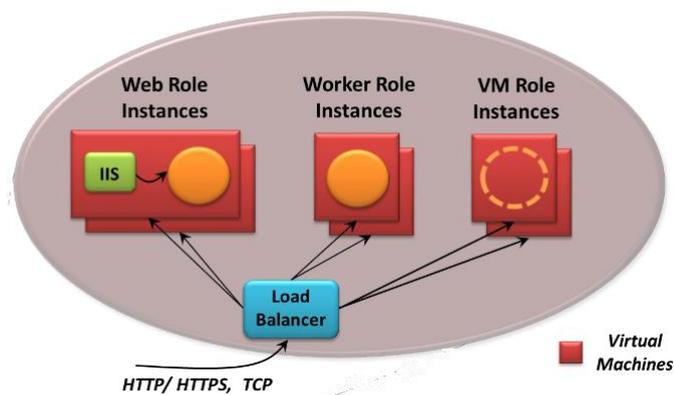


Figura 16. Servicios de Computación o Proceso (Compute Services)

Para los desarrolladores, construir aplicaciones en Windows Azure es similar a la construcción de aplicaciones en Windows. Microsoft proporciona plantillas en Visual Studio para ayudar a la creación de aplicaciones. También se proporciona un Kit de Desarrollo de Software (SDK) que simula el entorno de la *nube* localmente y permite su ejecución y depuración de forma local antes de ser desplegado en la *nube*.

Cuando se despliega una aplicación dentro de *Windows Azure* también se transmite la información de configuración, para cada rol se indica el número de instancias sobre las que se va a ejecutar. Cada instancia se crea una máquina virtual diferente.

Debido a esta escalabilidad, los roles de *Windows Azure* no deben mantener sus estado entre solicitudes ya que no se garantiza que múltiples solicitudes a una misma aplicación sean ejecutadas sobre la misma instancia. Para poder mantener una cierta persistencia, los datos específicos del cliente deben ser escritos sobre los diferentes sistemas de almacenamiento que se encuentran disponibles en la *nube*.

Los desarrolladores también pueden elegir el tamaño de las máquinas virtuales así como el número de procesadores y la cantidad de memoria que van a necesitar. También pueden solicitar que se incremente o decremente el número de instancias necesarias para uno o más roles.

Algunas de las ventajas que ofrece la centralización de las aplicaciones en la *nube* a los usuarios son:

- *Administración*: Se encarga automáticamente de los aspectos relacionados con el balanceo de la carga y la conmutación por error. Esto supone un menor coste y esfuerzo en el proceso de administración de la aplicación.
- *Disponibilidad*: Las aplicaciones se encuentran siempre disponibles siendo inmunes a las diferentes actualizaciones de *software* y a los errores de *hardware*.
- *Escalabilidad*: Se puede crear aplicaciones escalables que se ejecutan dentro de los centros de datos de *Microsoft*. Se puede ampliar o reducir las instancias necesarias en función de los recursos que en cada momento necesite la aplicación.

Servicios de Almacenamiento (Storage Services)

Las aplicaciones pueden trabajar con los datos de diferentes maneras. El servicio de almacenamiento de *Windows Azure* permite realizar el tratamiento de la información mediante cuatro servicios básicos diferentes que proporcionan un almacenamiento persistente y perdurable en la *nube*. Es un sistema de almacenamiento no relacional que no tiene tablas, ni vistas ni procedimientos. Estos servicios son escalables y de fácil acceso.

Los diferentes tipos de almacenamiento son accesibles desde varias aplicaciones o instancias de forma simultánea y se ofrecen como un servicio de acceso dedicado.

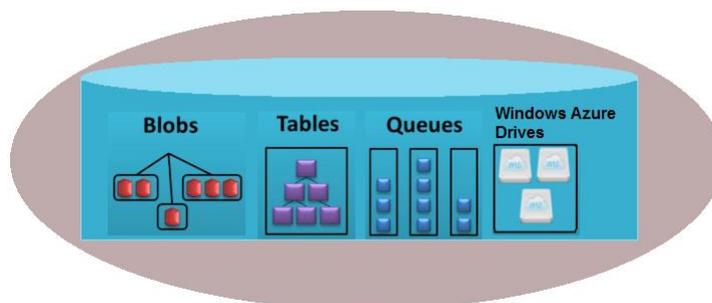


Figura 17. Servicio de almacenamiento de *Windows Azure* (Storage Service).

Los diferentes sistemas de almacenamiento son los que se explican a continuación:

- *Servicio de objetos binarios grandes (BLOBS)*: Es la manera más simple para almacenar grandes cantidades de texto no estructurado o datos binarios tales como imágenes, sonidos, videos... El servicio de blobs ofrece una jerarquía de almacenamiento simple: Una cuenta de almacenamiento puede poseer uno o más contenedores y en cada contenedor puede existir uno o varios blobs.

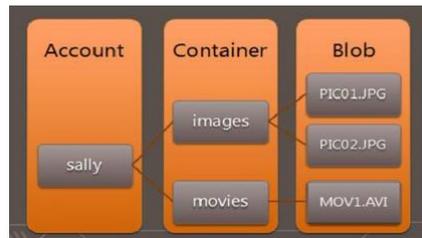


Figura 18. Servicio de objetos grandes binarios (blobs).

Existen dos tipos de *blobs*:

- Blobs de bloque (*Block Blobs*) los cuales están optimizados para *streaming*.
- Blobs de página (*Page Blobs*) que están optimizados para el acceso aleatorio de lectura y escritura en lugares arbitrarios del blob.

Para poder distinguir los blobs unos de otros se permite la asociación de metadatos una vez que han sido almacenados.

Para realizar la transferencia de blobs más eficiente, pueden dividirse en bloques, y si un error ocurre durante la transmisión, esta se reanuda desde el último bloque transferido. El API de los blobs está basado en REST.

- *Servicio de tablas (tables)*: Este servicio se utiliza para guardar grandes cantidades de datos de una forma estructurada no relacional. Es perfecto para trabajar con aplicaciones que necesitan trabajar con los datos a través de consultas. Las consultas se pueden realizar a través de la API REST o a través de las librerías de ADO.NET e incluso LINQ.

Dentro de una cuenta de almacenamiento de *Windows Azure* se pueden crear múltiples tablas, las cuales vienen diferenciadas por su nombre. Dentro de las tablas se almacenan entidades que son representadas como clases. Las entidades tienen un conjunto de propiedades.

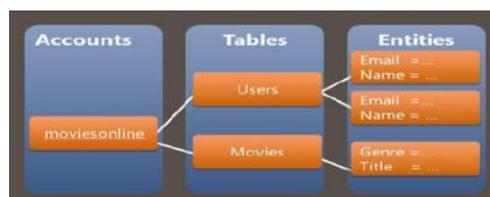


Figura 19. Servicio de Tablas (tables).

Las tablas no tienen un esquema definido e incluso las propiedades pueden ser de diversos tipos.

- *Servicio de colas (queue)*: Son procedimientos asíncronos que funcionan de manera distribuida. Ofrecen un servicio de mensajería fiable y persistente para comunicar las instancias de los roles web y los roles de trabajo. Se usan para transferir mensajes entre aplicaciones o servicios de *Windows Azure*. El API de las colas también está basado en REST y está basado en dos tipos de abstracciones: colas y mensajes.

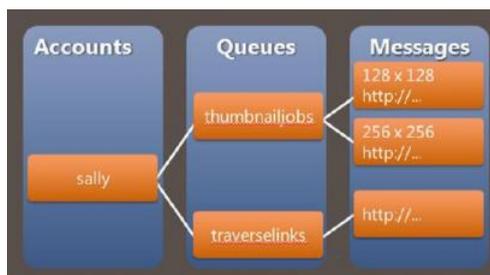


Figura 20. Servicio de colas (queues).

Las colas soportan atributos que se especifican como claves pares valor, la cuales permiten asociar metadatos a las colas de manera que se puedan 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. El tamaño máximo de los mensajes es de 8Kb.

Cuando un mensaje se lee desde una cola, el consumidor del mensaje es responsable de eliminarlo una vez que lo ha procesado. Cuando el mensaje ha sido leído, no estará disponible para otros consumidores durante cierto tiempo. Si el mensaje no es borrado en ese tiempo, su visibilidad se restablece para que otros consumidores puedan hacer uso de él.

- *Windows Azure Drive*: La unidad de *Windows Azure* permite a las aplicaciones montar un blob en páginas, que es un disco duro virtual (VHD) NTFS de un solo volumen. Permite la transferencia de datos entre nubes públicas y privadas. Es posible disponer de un sistema persistente que pueda ser compartido y accesible a través de la API de NTFS como si se accediera a una unidad de disco.

Este sistema de almacenamiento facilita la migración de aplicaciones a la *nube* que tienen dependencias de una unidad de disco.

Las principales características de los servicios de almacenamiento de *Windows Azure* son:

- Independientemente de cómo han sido almacenados los datos, estos son replicados tres veces. De esta manera, se posee una tolerancia a fallos que previene la pérdida de datos en caso de desastre. Los datos son replicados en centros de datos separados unos de otros por cientos de kilómetros dentro del mismo continente.

- Garantiza la consistencia de la información.
- Inserción y actualización de tablas (*Table Upsert*). Permite el envío de una solicitud única a Tablas de *Windows Azure*, Inserta una entidad si no existe o la actualiza o reemplaza si ya existía.
- A través de consultas *Select* se permite recuperar un subconjunto de datos dentro de una entidad, mejorando de esta forma el rendimiento y reduciendo el coste de serialización y deserialización, así como el ancho de banda utilizado.
- Permite a los clientes obtener una concesión sobre un mensaje y renovar esa concesión mientras se procesa, así como actualizar el contenido del mensaje para realizar un seguimiento de su procesamiento.
- Un mensaje insertado recientemente permanece invisible en la cola hasta que se agota el tiempo de espera de visibilidad.

Controlador (*Fabric Controller*)

Se considera los cimientos sobre los que se soporta la plataforma *Azure* y se le denomina más comúnmente *Fabric*.

Es el servicio que monitoriza, mantiene y proporciona máquinas para alojar las aplicaciones que los desarrolladores crean y almacenan en la *nube*. Toda la gestión que realiza el controlador es transparente al resto de servicios de *Windows Azure*.

Es el componente de la arquitectura que se encarga de proporcionar los servicios base de *Windows Azure* de manera transparente respecto a la infraestructura de TI. El desarrollador no sabe nunca en qué máquina concreta del centro de datos de *Microsoft* se está ejecutando la aplicación *Azure*. Se encarga de asegurar que la aplicación recibe tiempo de ejecución, ancho de banda y recursos para su ejecución, balanceando la carga a las máquinas virtuales de forma transparente a la aplicación.

El controlador se comunica con los servidores a través de un servicio especial que se ejecuta en los servidores. El servicio rastrea el estado actual de los servidores y el estado objetivo. El controlador y el servicio gestionan como el sistema puede alcanzar el estado objetivo desde el estado actual, como por ejemplo, pasar de un estado inactivo a ejecutar un rol de trabajo. Cuando un error ocurre, el servicio detecta el fallo y cambia el estado actual y lo marca como que algo ha fallado, entonces, el controlador y el servicio tienen que gestionar que es lo que se requiere para volver al estado objetivo.

Uno de los roles principales del controlador es proporcionar los recursos necesarios basándose en las necesidades de las aplicaciones que han sido escritas por el desarrollador. Para gestionar esto, tiene un modelo de servicio declarativo que define exactamente lo que la aplicación necesita. Este modelo define los roles que la aplicación ejecuta y como se comunican entre ellos, cuales son los requisitos del sistema operativo, la cantidad de memoria que se necesita, el ancho de banda requerido... Cuando llega el momento de proporcionar recursos a una instancia, el

controlar examina los requisitos definidos, y a través de su propio inventario busca los recursos que se ajustan a las características.

El controlador es altamente redundante y se pueden llegar a encontrar de 5 a 7 réplicas disponibles en un mismo instante de tiempo. Los estados de todos los nodos son replicados en el *fabric*, y todas estas réplicas aseguran que nada suceda cuando el controlador está gestionando un nodo particular.

El controlador monitoriza todas las instancias que se está ejecutando. Cuando una instancia falla, el controlador inicia una nueva instancia para mantener el número total de instancias activas que la aplicación requiere.

También orquesta el acceso a los recursos de almacenamiento y colas de la plataforma *Windows Azure* para la aplicación sin necesidad de preocupar al desarrollador.

Centro de distribución de datos (CDN)

El centro de distribución de datos de *Windows Azure* permite mejorar el rendimiento y la fiabilidad del sistema colocando copias de los datos cerca de donde se encuentran los usuarios. Los datos a los que acceden los usuarios están almacenados en caché en lugares estratégicamente situados, de esa forma, se permite un acceso más rápido a los mismos.

La CDN de *Windows Azure* es utilizada para garantizar el rendimiento del sistema y proporcionar una forma rápida de acceder a los datos a aquellos usuarios finales que se encuentran lejos de los contenidos. Colocando los datos en lugares más cercanos se evitan viajes innecesarios por la red para cargar los contenidos.

La CDN también permite administrar cargas elevadas de contenido que se distribuyen a gran escala de manera instantánea.

La entrega de los contenidos se puede realizar desde una gran variedad de recursos de *Windows Azure*, incluidos los servicios de almacenamiento y los roles. Se puede distribuir cualquier tipo de contenido tal como objetos web (jpg, css, javascript,..), objetos descargables, aplicaciones, *streaming*...

Redes Virtuales

Windows Azure Connect proporciona mecanismos para configurar la conectividad de red IP entre los recursos locales y los recursos que se encuentran en la *nube*, de esta forma, se permite a los desarrolladores compilar fácilmente aplicaciones basadas en la *nube* y que se puedan conectar con seguridad a la infraestructura local.

Se puede establecer una conectividad directa y segura entre los roles web, roles de trabajo o los roles de VM hospedados en la *nube* y los equipos locales detrás de un *firewall*, lo que permite administrar y solucionar problemas de forma remota.

Permitirá a una organización mover una aplicación a la *nube* de forma que los datos se mantuviesen en las máquinas locales de la organización.

Las ventajas que ofrece el servicio de *Windows Azure Connect* son:

- Instalación sencilla basada en agente. Se instala un agente en cada equipo que debe conectarse a la *nube*.
- No es necesario conocer la topología de la red.
- Funciona según la directiva del *firewall* corporativo. Sólo requiere acceso a los puertos 80 y 443.
- Administración a través del Portal de *Windows Azure*.
- Rápida configuración.
- Estándares abiertos seguros a través del uso de protocolos estándar del sector IPSec de un extremo a otro, para establecer conexiones seguras.
- Control pormenorizado de la conectividad. La directiva de red del portal de *Windows Azure* permite especificar que conexiones deben configurarse en función de los equipos.

3.2.5.2.2. **SQL AZURE**

SQL Azure es el servicio de datos relacional que *Microsoft* oferta en la *nube* y que está construido sobre la tecnología de *SQL Server*. Proporciona servicios de bases de datos altamente escalables y con alta disponibilidad alojados por *Microsoft* en la *nube*.

Facilita en gran medida el despliegue de bases de datos sobre la *nube* ya que, los desarrolladores y el personal asociado al departamento de TI no tienen que instalar, actualizar y gestionar la infraestructura de bases de datos siendo transparente a ellos.

La alta disponibilidad y la tolerancia a errores ya se encuentran integradas. Al igual que otros servicios de la plataforma de *Windows Azure*, *SQL Azure* está funcionando en los centros de datos de *Microsoft*. La infraestructura de los centros de datos proporciona a los servicios de *SQL Azure* el balanceo de la carga, la tolerancia a fallos y la capacidad de replicación.

El servicio de *SQL Azure* está compuesto de las siguientes capas:

- *Infraestructura*: Es la capa que proporciona la administración del hardware y los sistemas operativos. Es el núcleo de los centros de datos y se comparte con muchos otros servicios.
- *Plataforma*: Esta formada por las instancias de *SQL Server*, el *SQL Azure fabric* y el gestor de servicios. Las instancias de *SQL* representan las bases de datos desplegadas, sus réplicas y las instancias de sistemas operativos que alojan las diferentes instancias de *SQL*.

- *SQL Azure fabric* es el marco de trabajo que automatiza el despliegue, replicación, tolerancia a fallos y el balanceo de la carga de los servidores de bases de datos. Al igual que en los servicios de almacenamiento ofrecidos en *Windows Azure*, *SQL Azure fabric* mantiene tres réplicas disponibles de las bases de datos, en el caso que una réplica falle, se crea otra inmediatamente para mantener siempre las tres réplicas activas.
- *Capa de servicios*: Es la capa que hace de enlace entre la plataforma y el cliente. Expone el protocolo TDS sobre el puerto 1433. Esta capa es la responsable de enrutar las conexiones a la instancia de base de datos primaria sobre la plataforma.

También se lleva a cabo en esta capa la monitorización de los servicios de facturación para realizar el seguimiento de uso de la base de datos y los costes que ello conlleva.

- *Capa de cliente*: Es la única capa que se encuentra fuera de los centros de datos de Microsoft. No incluye ningún componente específico de *SQL Azure*. Usa las herramientas locales del cliente para obtener un acceso completo a las bases de datos que se encuentran en la *nube*.

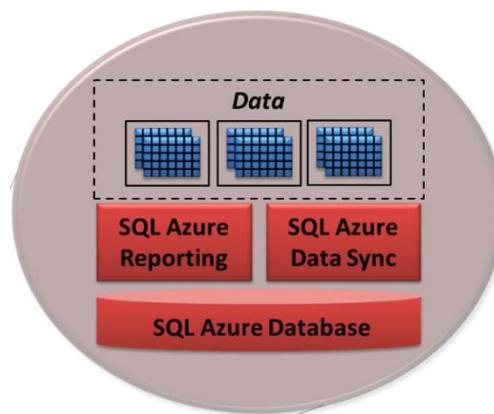


Figura 21. SQL Azure.

Junto con el servicio básico de bases de datos, se proporcionan otros dos más complementarios como son la generación de informes (*Reporting*) y la sincronización de los servicios.

SQL Azure Database

SQL Azure Database es una base de datos relacional en la *nube* construida bajo las tecnologías de *SQL Server*. Permite desplegar soluciones de bases de datos relacionales sobre la *nube* y obtener las ventajas que un centro de datos proporciona, tales como disponibilidad, escalabilidad y seguridad sin necesidad de tener que configurarlos de forma explícita.

Las ventajas que se ofrecen por trabajar con bases de datos sobre la *nube* son:

- *Mantenibilidad:* Ofrece la escala y las funcionalidades de un centro de datos de empresa sin los gastos administrativos asociados al mantenimiento de instalaciones necesarias para mantener toda la infraestructura. Esto permite que no haya que dedicar recursos específicos a su mantenimiento.
- *Alta disponibilidad:* Se basa en la tecnología ya probada de *Windows Server*. El servicio hace copias redundantes de los datos en diversos servidores físicos para mantener la disponibilidad de datos en caso de fallo del sistema. El sistema es más robusto que los sistemas de bases de datos habituales.
- *Escalabilidad:* Conforme la base de datos va creciendo, el sistema crece proporcionalmente. Los usuarios sólo pagan por el espacio que usan.
- *Modelo de desarrollo familiar:* Al igual que en los sistemas de bases de datos locales, *SQL Azure* proporciona los mismos interfaces y protocolos de comunicación (TDS). Se usan las mismas herramientas cliente y librerías para construir aplicaciones cliente que utilizan bases de datos almacenadas en la *nube*.

Utiliza el mismo protocolo que para acceder a las bases de datos locales y por lo tanto pueden ser usadas por cualquier cliente de SQL Server como puede ser *Entity Framework*, ADO .NET, ODBC, PHP...

- *Modelo de datos relacional:* Los datos son almacenados en los servidores SQL usando Transact-SQL. Dentro de cada servidor de *SQL Azure* se pueden crear múltiples bases de datos que tienen tablas, vistas, procedimientos almacenados, índices... Este modelo de datos se basa en el modelo relacional y simplifica el proceso de migración de bases de datos a la *nube*.

Los servidores y las bases de datos de *SQL Azure* son objetos virtuales que no corresponde con los servidores y bases de datos físicos.

Hay características de SQL Server que todavía no se encuentran pero que serán implementadas en un futuro.

- Transacciones distribuidas.
- Consultas a servidores remotos
- Acceso desde tecnologías antiguas como OLEDB.
- Servicio de mensajes de SQL Server.
- *SQL Common Language Runtime*
- Búsqueda completa de texto.

SQL Azure Reporting

Es la tecnología capaz de genera informes ricos sin tener que mantener una infraestructura propia. Usa las mismas herramientas que hasta ahora han sido utilizadas en local para subir informes a la *nube*. Está diseñado para trabajar con datos almacenados en *SQL Azure Database*.

No es necesario administrar o mantener una infraestructura separada para la generación de informes, lo que conlleva costes más bajos y menor complejidad. Los clientes pueden acceder a los informes desde el portal de *SQL Azure*, a través del navegador o de sus propias aplicaciones.

Las características principales que posee son:

- Desarrollo de informes a través de *Business Intelligence Design Studio* o *Report Builder*, permitiendo realizar informes con características avanzadas: mapas, gráficos...
- Exportación a varios formatos como Excel, Word, HTML, PDF...
- Escalabilidad y flexibilidad bajo demanda. Permite aumentar o disminuir la capacidad de la plataforma según los requisitos.
- Alta disponibilidad y tolerancia a fallos.

Al igual que en *SQL Azure Database* existen funcionalidades que existen en el entorno local pero que todavía no se han implementados en la *nube*.

SQL Azure DataSync

El almacenamiento de datos en *SQL Azure Database* hace que puedan ser accesibles desde cualquier aplicación usando únicamente una conexión a Internet. Pero, existen situaciones donde tiene sentido mantener copias de los datos basados en la *nube* en otros lugares.

SQL Azure Data Sync es un servicio de sincronización de datos basado en la *nube* que permite la sincronización bidireccional de los datos de forma que puedan ser compartidos de forma sencilla entre distintas bases de datos de *SQL Azure* o bases de datos locales.

Los escenarios de sincronización que se proporcionan son los siguientes:

- Sincronización de datos entre bases de datos de *SQL Azure* y bases de datos locales de SQL Server. El sentido puede ser tanto unidireccional como bidireccional.
- Sincronización de datos entre bases de datos de *SQL Azure* alojadas en diferentes centros de datos.

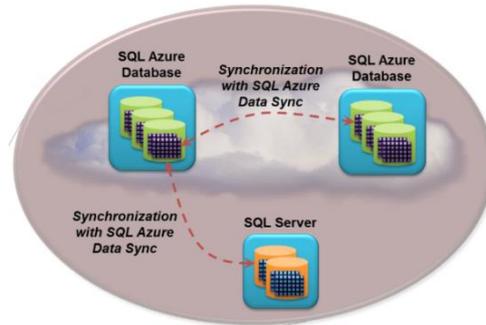


Figura 22. Sincronización de Bases de Datos.

Al igual que el resto de servicios ofertados en la *nube*, es un servicio totalmente administrado, por lo que no es necesario escribir una lógica compleja para realizar la sincronización. Dentro del portal se puede configurar rápidamente la sincronización seleccionando los elementos y haciendo clic.

El control de la sincronización es total, se puede especificar de forma detallada las tablas y columnas que se quieren sincronizar, los filtros que se desean aplicar, las directivas de resolución de conflictos y la frecuencia con la que se quiere llevar a cabo la sincronización.

3.2.5.2.3. *App Fabric*

Windows Azure AppFabric proporciona la infraestructura para las aplicaciones. A través de sus componentes, permite la integración de servicios y aplicaciones que se ejecutan en la *nube* en proveedores de acceso tradicionales y en la propia empresa basándose en estándares de interoperabilidad.

AppFabric permite a los usuarios de la *nube* conectar servicios locales a los servicios alojados en la *nube*, asegurar los diferentes servicios con marcos de seguridad nuevos o ya existentes (basados en identidades, directorio activo,...) y permite construir y activar la integración de aplicaciones compuesta.

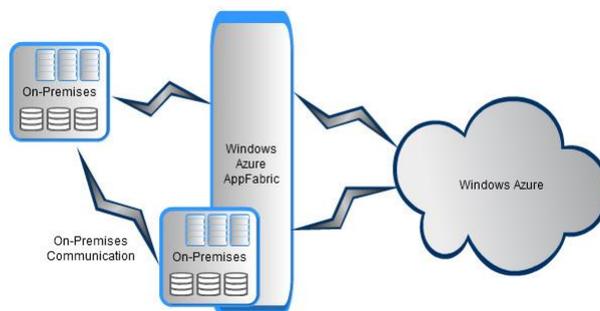


Figura 23. Windows Azure AppFabric.

Los tres componentes de *AppFabric* hoy en día son el bus de servicio (*Service Bus*), el control de acceso (*Access Control*) y el servicio de Caché (*Caching*).

Bus de Servicio (Service Bus)

Es un servicio que permite orquestar la conectividad segura entre diferentes servicios y aplicaciones a través de cortafuegos y redes utilizando diversos patrones de comunicación. Los servicios se registran en el *bus* de forma que pueden ser accedidos fácilmente desde diferentes tipologías de red.

Mediante el *bus de servicios*, una aplicación que se conecta a servicios controlados por terceros, puede olvidarse de detalles como la autenticación, autorización, protocolos de comunicación, cortafuegos delegando estas cuestiones al *bus* de servicios.

Facilita la labor de conexión entre aplicaciones que se ejecutan sobre *Windows Azure* o contra *SQL Azure* con aplicaciones que se están ejecutando en una infraestructura propia contra servidores de datos convencionales. Ayuda a la creación de aplicaciones compuestas mediante la integración de servicios ya existentes y otros nuevos que se ejecutan en la plataforma de *Azure*.

El funcionamiento del bus de servicio es el que se muestra en la siguiente figura:

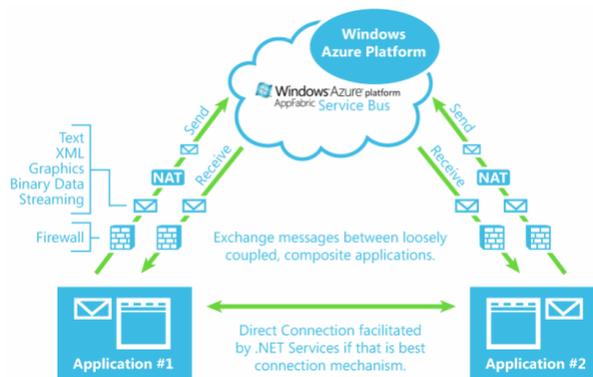


Figura 24. Funcionamiento del bus de servicio.

El mecanismo de comunicación preferido entre aplicaciones en la *nube* es *Windows Communication Foundation (WCF)*.

El bus de servicio proporciona funciones de mensajería y conectividad segura que permite crear aplicaciones distribuida en la *nube*, así como aplicaciones híbridas.

Las principales características del bus de servicio son:

- Proporciona una infraestructura hospedada, segura y de gran disponibilidad que permite una comunicación extensa, ofreciendo funcionalidades de mensajería, nomenclatura y publicación de servicios.
- En base a la mensajería desacoplada entre remitente y receptor permite el intercambio de mensajes asíncrono sin necesidad de que ambos estén conectados en el mismo momento.

- Proporciona una dirección URL estable, accesible desde cualquier punto de Internet, sin tener en cuenta la ubicación real.
- Ofrece espacios de nombres jerárquicos globales que son independientes del DNS y del transporte.
- El acceso anónimo a los servicios sólo se permite si se conceden los permisos adecuados.
- El servicio de mensajería centralizado con equilibrio de carga que se ejecuta en la *nube*, admite varios protocolos de transportes y estándares de servicios web tales como SOA, WS-* y REST. También admite la comunicación bidireccional y proporciona conectividad directa punto a punto a través de NAT.
- El servicio de mensajería está negociado con colas, temas y suscripciones. La clasificación de mensajes está basada en FIFO con colas y se admiten varios remitentes y varios receptores.
- Incluye API de cliente .NET, compatibilidad con el acceso HTTP y REST desde plataformas distintas de .NET.

Las principales ventajas del bus de servicio son:

- Exposición de aplicaciones y servicios a través de firewalls, puertas de enlace NAT y otros límites de red problemáticos.
- Menos complicaciones para crear aplicaciones de correspondencia imprecisa mediante la exposición sencilla de extremos.
- Modelo de programación sencillo que admite protocolos estándar y amplía enlaces estándares similares para WCF.
- Ayuda a bloquear el tráfico malintencionado y a proteger a los servicios de las intrusiones o ataques de negación de servicios.

Control de acceso (Access Control)

Permite genera una autorización federada entre aplicaciones y servicios, sin la programación complicada que se requiere para proteger aplicaciones que sobrepasan los límites de una organización. Permite configurar reglas de autenticación y autorización de forma fácil y flexible cubriendo diferentes necesidades de seguridad y distintas infraestructuras de administración de identidades. Este servicio es de gran utilidad para orquestar servicios y comunicarlos entre sí.

Admite un modelo sencillo de reglas y *claims* que pueden configurarse con facilidad y flexibilidad cubriendo diferentes necesidades de seguridad e infraestructuras de administración de identidades. Puede integrar de forma sencilla las aplicaciones de la *nube* con proveedores de identidad como *Windows Live*, *Google*, *Yahoo* y *Facebook*, así como con los directorios de empresa basados en los estándares como *Active Directory*.

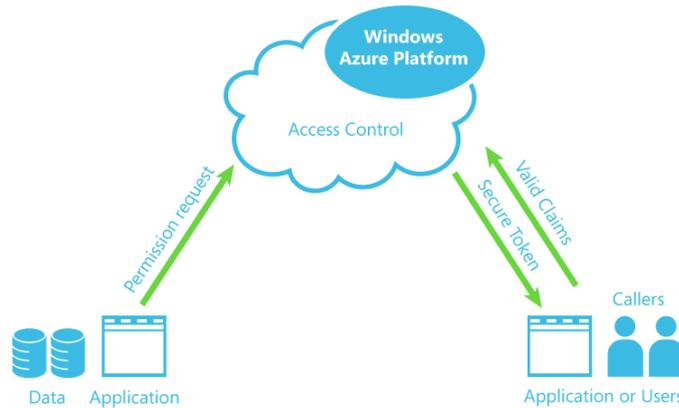


Figura 25. Servicio de control de acceso.

Las ventajas que ofrece el control de acceso son:

- La autorización basada en reglas permite a las aplicaciones responder como si las cuentas de usuario se administrasen localmente.
- Proporciona la flexibilidad necesaria para integrarse de forma rápida con los proveedores de identidad.
- Se basa en *Microsoft .NET Framework* y WCF.

Servicio de Caché (Caching)

Para muchas aplicaciones, uno de los caminos más eficientes para mejorar el rendimiento es a través del acceso a datos en la caché.

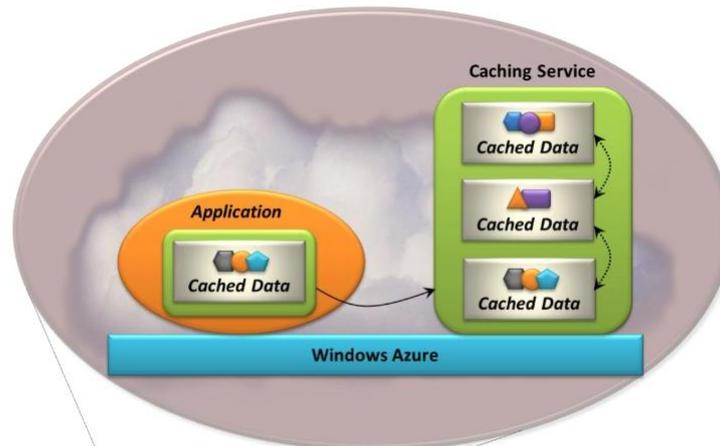


Figura 26. Servicio de Caché.

El servicio de caché proporciona una caché distribuida para las aplicaciones de *Windows Azure*. El servicio incluye una caché local que puede mantener una copia reciente de los conjuntos de datos a los que se ha accedido desde cada una de las instancias. Si un elemento de la aplicación no es encontrado en la caché local, la biblioteca de caches, automáticamente contacta con la caché compartida la cual es proporcionada por la biblioteca del servicio de caché. La caché

se extiende sobre un gran número de instancias de *Windows Azure* y cada una de ellas mantiene cacheados diferentes datos.

El servicio de caché suele ser útil para:

- El almacenamiento en caché de datos de referencia a los cuales se debe acceder de forma rápida y frecuente desde una aplicación.
- El almacenamiento en caché de datos orientados a actividades que se generan como parte de la actividad de la aplicación (ejem.: cesta de la compra).
- El almacenamiento en caché de datos orientados a recursos.

Las ventajas que ofrece este servicio son:

- Se proporciona como servicio y no requiere una administración e instalación de las instancias. Sólo es necesario configurar la región donde se desea ubicar el servicio de caché y el tamaño de memoria que la caché necesita.
- Posee una administración flexible, se ajusta el tamaño de la memoria caché de acuerdo con las necesidades de uso en cada momento.
- Es fácilmente integrable con las aplicaciones web ASP.NET.

3.2.5.2.4. Market Place

Windows Azure Marketplace es un mercado en línea global donde los proveedores de servicios y los publicadores de datos pueden publicar y vender aplicaciones, servicios, componentes y bases de datos de *Windows Azure*.

Microsoft ofrece un canal global de ventas online que permite acelerar sus inversiones a los socios dando a conocer sus servicios en la *nube*.

AppMarket

El propietario de las aplicaciones es quien decide el tipo de suscripciones que quiere hacer disponibles, las condiciones de uso para utilizar sus aplicaciones y el precio que los usuarios deberían pagar por su usuario.

Las aplicaciones se publican en el *marketplace* de forma que los clientes las pueden encontrar fácilmente y suscribirse a ellas.

Las aplicaciones que se exponen son expuestas como Software como servicio (SaaS).

Tiene una orientación específica para los desarrolladores. No sólo ofrece una plataforma para la venta de productos finales. Los desarrolladores pueden compartir, encontrar, vender o comprar plantillas, componentes, servicios, paquetes de datos o código para construir aplicaciones.

DataMarket

Al igual que en las aplicaciones, el usuario será quien decida qué datos son accesibles para el resto de las personas, las condiciones de uso y el precio que los clientes deberían pagar.

Los datos son expuestos en el *marketplace* haciendo que los datos puedan ser consumidos con herramientas tales como *Excel* o a través de APIs estándar. Soporta múltiples tecnologías cuando los datos son publicados; *SQL Azure*, REST y SOAP basados en servicios web y OData servicios.

Está orientada a compartir, comprar o vender conjuntos de datos.

3.2.6. Comparativa de Servicios Ofrecidos por los diferentes proveedores

Entre los muchos proveedores de servicios existentes en la *nube*, existen tres que son los que dominan el mercado de *Cloud Computing*, esto proveedores son *Amazon*, *Google* y *Microsoft*.

Las principales diferencias entre ellos son el área de la *nube* sobre el que se mueven y cuando empezaron a ofertar sus servicios en la *nube*.

- *Amazon* es el proveedor más antiguo y basa sus servicios dentro del área de *IaaS* aunque también esta introduciéndose en otras áreas tales como *PaaS*.
- *Google* se mueve en torno al área de *PaaS* a través de su plataforma *App Engine* y en el área de *SaaS* a través de la oferta realizada dentro de *Google*.
- *Microsoft* fue el proveedor que más tarde en ofrecer sus servicios en la *nube* y se mueve en las mismas áreas que *Google*. Dentro del área de *PaaS* ha desarrollado la plataforma *Windows Azure* y dentro del área de *SaaS* ha desarrollado *Office 365*.
- La oferta de *Microsoft* y de *Google* son muy similares y ambos enfocan su negocio tanto a organizaciones de diferentes tamaños como a usuarios particulares. Pese a su reciente incorporación a la *nube*, *Microsoft* es el líder indiscutible del mercado dentro de la capa de *PaaS*. *Google* sigue liderando la capa de *SaaS* aunque es una cuestión más bien económica.
- La arquitectura que ofrecen *Google* y *Microsoft* en el área de *PaaS* es similar.

A continuación se muestra una tabla comparativa de lo que cada servicio ofrece:

	Google AppEngine	Microsoft Windows Azure	Amazon
Objetivo	Desarrollo de aplicaciones web	Orientar su negocio a las organizaciones y desarrollo de aplicaciones web	Ofrecer flexibilidad a las empresas en el ámbito de las TI
SO Soportados	Google Chrome OS	Windows Azure	Windows Server, Linux, Open Solaris
Lenguajes de Programación soportados	Python, Java	Principalmente .NET pero también puede integrar otros lenguajes como PHP, Java, Ruby...	Java, PHP, Python, Ruby, .NET
Herramientas de desarrollo	IDE basado en Eclipse	Herramientas Azure para Microsoft Visual Studio, IDE de Eclipse.	IDEs de desarrollo de los lenguajes mencionados con anterioridad
Modelos de programación	Solicitudes basadas en web	No registrado	No registrado
Almacenamiento	BlobStore	Unidades NTFS en blobs. Azure Datastore (blobs, tables, queues)	Amazon Simple Storage Service (S3) y Amazon Elastic Blob Store (EBS)
Persistencia de datos	Datastore Cloud SQL	SQL Azure	Amazon Simple DB
Colas	No soportado	Windows Azure Queue	Amazon Simple Queue Service
Escalado automático	Sí	Sí	Semi-automatico.
Proveedores de Infraestructura	Sus propios centros	Sus propios centros.	Sus propios centros

Figura 27. Tabla comparativa *Google App Engine, Microsoft Windows Azure, Amazon.*

4. Prototipo. Tormenta de Ideas

El objetivo del prototipo es la creación de un portal donde los usuarios de un grupo de trabajo tengan un lugar donde generar ideas y poder valorarlas.

El funcionamiento básico del prototipo será:

- Una persona genera una nueva idea y la introduce en el portal. En el portal explica brevemente en qué consiste su idea y solicita a un determinado grupo de personas que aportarán comentarios a su idea y la valorarán antes de la fecha que se ha puesto como límite.
- Las personas que el propietario ha asignado para valorar la idea, accederán al portal y podrán puntuar y comentar la idea sobre la cual se les ha pedido su opinión.
- Dentro del portal se mostrará un ranking de las ideas mejor valoradas.

En los siguientes apartados se va a mostrar brevemente los requisitos especificados para el desarrollo del prototipo, la arquitectura establecida y el diseño de la base de datos.

También se explicará cómo se crea una base de datos dentro de *SQL Azure* y el despliegue de una aplicación, creada usando un rol web, dentro de la *nube*.

4.1. Análisis de Requisitos.

4.1.1. Requisitos Funcionales

Los requisitos funcionales (RF) son utilizados para establecer los objetivos y características que debe poseer la aplicación en base a su comportamiento (qué debe hacer el sistema).

A continuación se listan los requisitos funcionales que se detectan inicialmente y que el portal de ideas deberá satisfacer.

Se han agrupado los requisitos por módulos a partir de funciones globales que debe tener la aplicación.

Modulo 1, Acceso a la aplicación

- RF 1.** Todos los usuarios que accedan a la aplicación deberán identificarse a partir de su nombre de usuario y una contraseña.
 - RF 2.** La gestión de usuarios permitirá crear nuevos usuarios, modificar la información de los mismos y listarlos y darlos de baja.
 - RF 3.** Existirán tres grupos de usuarios: administrador, moderador, colaborador.
 - RF 4.** Solo podrán realizar la gestión de usuarios aquellas personas que tengan permisos de administrador dentro de la organización.
-

- RF 5.** Dar de baja un usuario implica que perderá el acceso a la aplicación pero que no se eliminará del sistema.

Modulo 2, Gestión de temas

- RF 6.** La gestión de temas permitirá crear nuevos temas, listarlos, consultarlos y darlos de baja.
- RF 7.** Los colaboradores son las personas encargadas de dar de alta nuevos temas en la aplicación.
- RF 8.** Los temas se utilizan para clasificar las ideas. Dentro de un tema puede haber muchas ideas.
- RF 9.** Un tema tendrá un código de tema y un nombre.
- RF 10.** Un tema vendrá identificado por su código.

Modulo 3, Gestión de ideas

- RF 11.** La gestión de ideas permitirá crear nuevas ideas, listarlas, editarlas, valorarlas, comentarlas y archivarlas.
- RF 12.** Cualquier usuario puede crear una nueva idea. Por cada idea el usuario introducirá un título, una descripción y una fecha límite de valoración. Además tendrá que asignar los temas dentro de los cuales se engloba una idea (al menos deberá pertenecer a un tema) y deberá asignar a la idea la persona o personas que pueden hacer comentarios y valoraciones sobre la misma. El código de la idea se generará de forma automática.
- RF 13.** Una idea únicamente podrá ser modificada por el propietario de la misma.
- RF 14.** Cada vez que se genere una idea o el propietario de la misma lo considere necesario, se enviará un correo electrónico al grupo de usuarios asignado a la idea para informarle que se ha generado una nueva idea donde deben colaborar o que esa idea ha sufrido modificaciones.
- RF 15.** Los usuarios asignados a la idea podrán introducir comentarios sobre la misma y valorarla siempre y cuando no se haya superado la fecha límite de valoración.
- RF 16.** La valoración de una idea vendrá dada por un valor numérico entre 1 y 10.
- RF 17.** Una idea únicamente podrá ser archivada por un moderador aunque no sea el propietario de la misma. La persona que archiva la idea deberá introducir la razón por la que la archiva (finalizada, obsoleta, repetida...)
- RF 18.** Los colaboradores podrán ver un listado de todas las ideas, pero sólo podrán aportar su opinión sobre aquellas tareas abiertas en las cuales se encuentren asignados.

RF 19. Las ideas podrán ser visualizadas en un listado. Los criterios de búsqueda dentro del listado serán: por temas, por palabras, según su estado (abiertas, activas, archivadas...)

RF 20. Existirá un apartado donde se visualizará un listado de todas las ideas activas ordenadas según la valoración que hayan obtenido. El ranking de ideas podrá ser filtrado por temáticas.

4.1.2. Requisitos No Funcionales

Los requisitos no funcionales (RNF) establecen la naturaleza de la aplicación en base a su rendimiento, volumen de datos, frecuencia de tratamiento, seguridad, implantación y otros (como debe operar el sistema).

RNF 1. La aplicación a desarrollar será sobre un entorno web sobre la plataforma de *Microsoft* en la *nube* (*Windows Azure Platform*).

RNF 2. Para el desarrollo de la aplicación se utilizará Visual Studio 2010 y el SDK de Windows Azure para poder integrar la aplicación en *nube*.

RNF 3. El lenguaje de programación base para desarrollar la herramienta será C#.

4.1.3. Requisitos No Contemplados

Los requisitos no contemplados (RNC) son aquellos que no se encuentran dentro del alcance del proyecto.

RNC 1. Generación de informes.

RNC 2. Generación de gráficas.

4.2. Diagramas de casos de Uso

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

El diagrama de casos de uso se ha dividido en varios paquetes según las principales funcionalidades que va a tener el sistema.

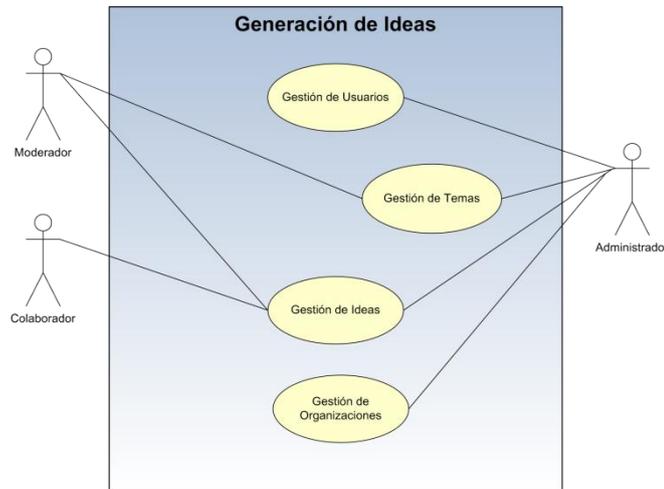


Figura 28. Diagrama general de casos de uso.

El caso de uso de **Gestión de Usuarios** contiene todos aquellos casos relacionados con la autenticación de usuarios y la gestión de los mismos.

El caso de uso de **Gestión de Temas** contiene todos los casos relacionados con la gestión de los temas dentro del portal.

El caso de uso **Gestión de Ideas** contiene todos los casos relacionados con la gestión y valoración de ideas.

El caso de uso **Gestión de Organizaciones** contiene todos los casos relacionados con la gestión básica de organizaciones dentro del portal.

4.2.1. Actores

El grupo de actores que va a interactuar con el sistema es el que se describe a continuación:

- *Administrador*: Representa el rol del sistema que se encarga de dar de alta a los usuarios, introducir sus datos personales y adjudicarles el rol que les corresponde.
- *Moderador*: Representa el rol del sistema encargado de gestionar los temas y archivar aquellas ideas que ya no se consideran útiles.
- *Colaborador*: Representa el rol del sistema encargado de generar ideas y de aportar comentarios o valoraciones sobre las mismas.

4.3. Diseño

4.3.1. Diagrama de la Arquitectura de Software

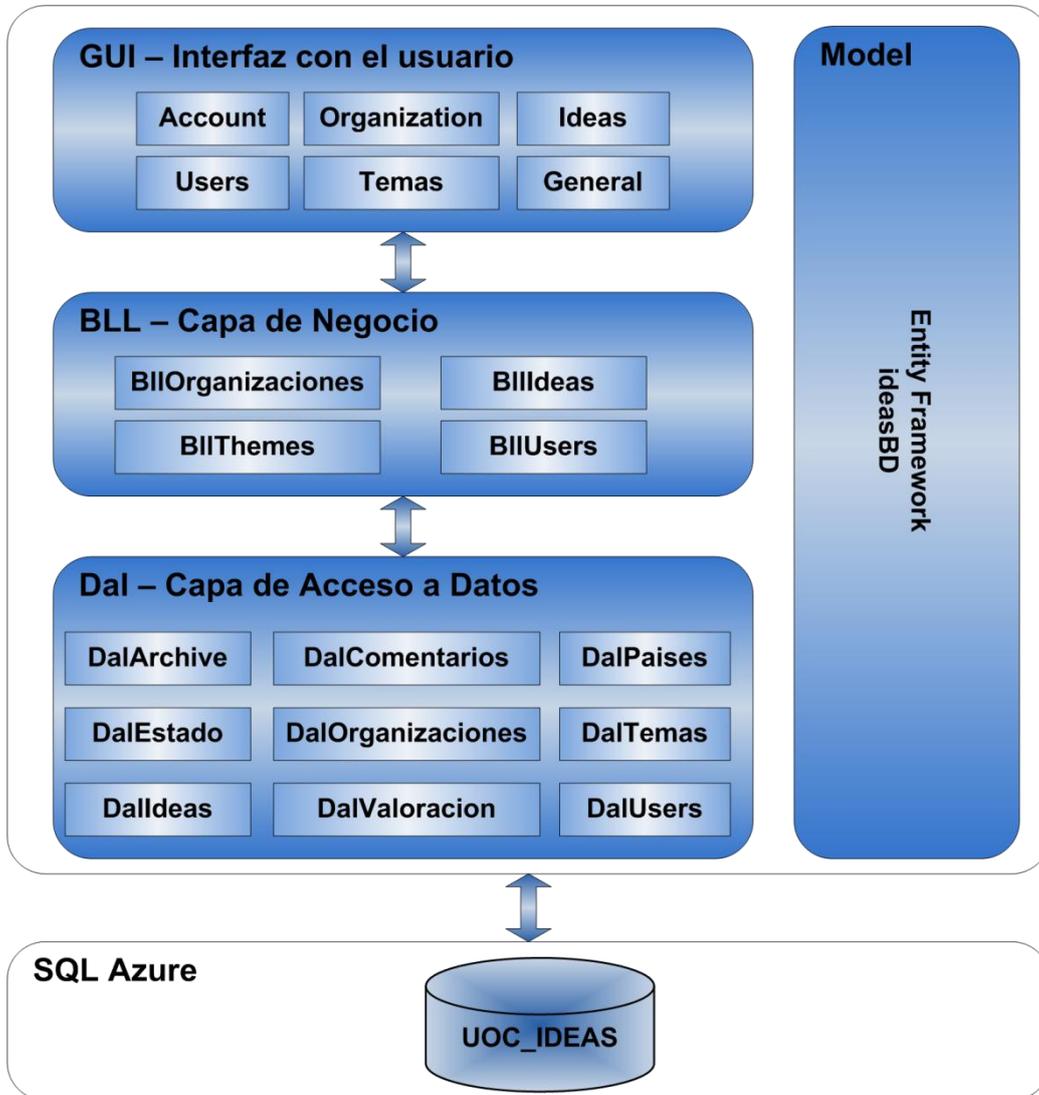


Figura 29. Arquitectura Software Sistema

4.3.2. Diseño de la Base de Datos

Estructura de las tablas de la base de datos que contienen la información de la aplicación.

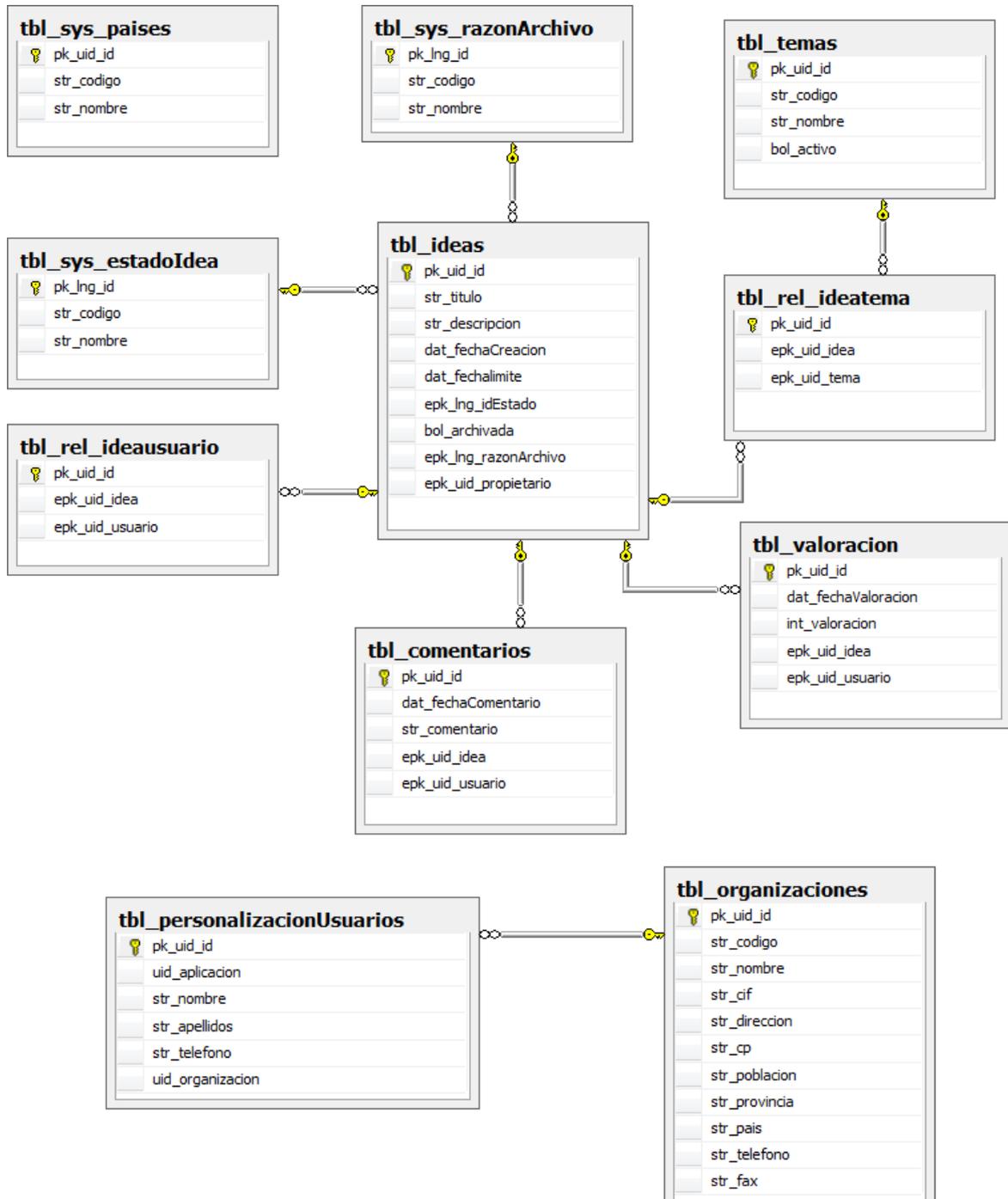


Figura 30. Diseño de Bases de Datos.

Diseño de las tablas que contienen toda la información acerca de la autenticación de usuarios.

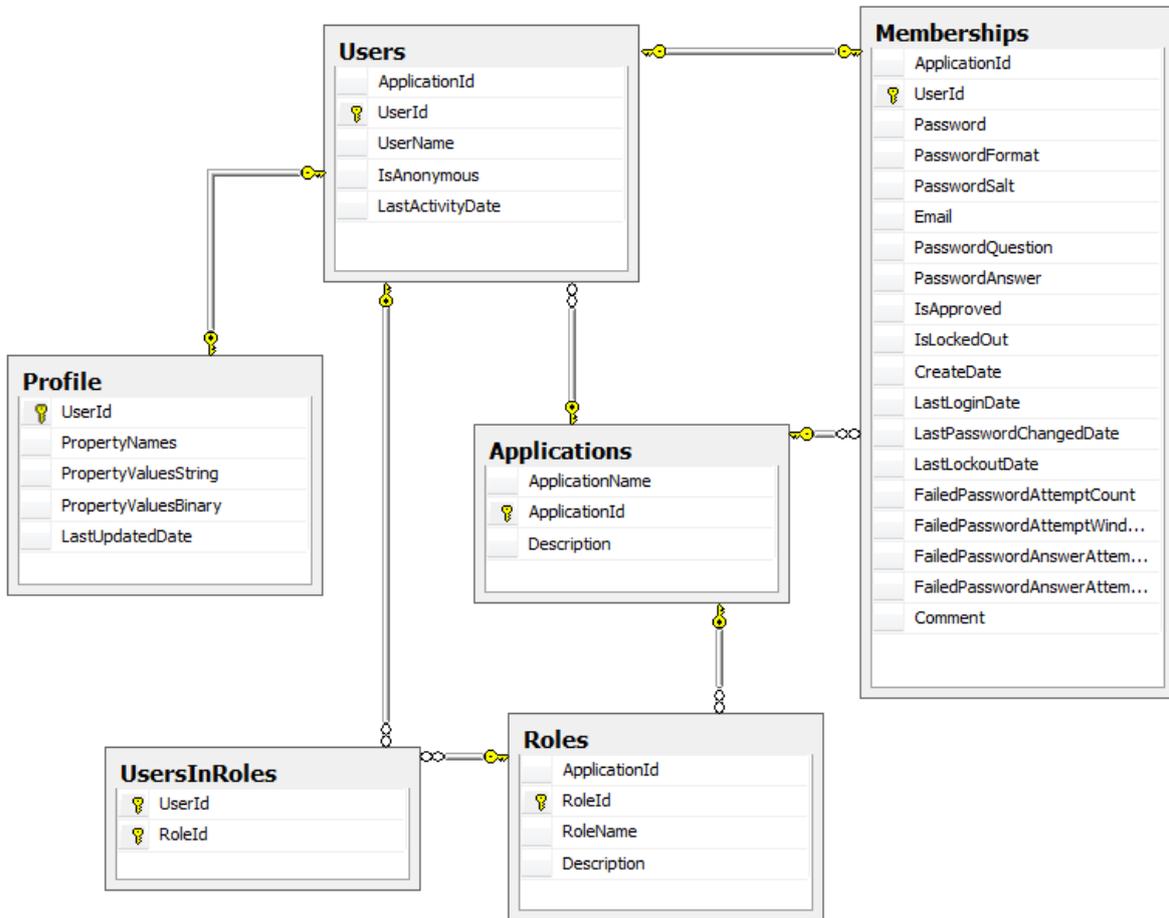


Figura 31. Diseño Base de Datos. Tablas de Autenticación de Usuarios.

4.4. Despliegue en Windows Azure

A continuación se explican los pasos necesarios para hacer pública tanto la aplicación como la base de datos.

Dentro de este apartado se presenta:

- La creación y configuración de la Base de Datos dentro de *SQL Azure*.
- La publicación de la aplicación dentro de *Windows Azure*.

Hay que hacer notar que para poder alojar bases de datos en *SQL AZURE* y desplegar aplicaciones en *Windows Azure* es necesario disponer de una cuenta en la plataforma *Windows Azure* (<http://www.windows.azure.com>).

4.4.1. Creación de una BD en SQL Azure

Dentro de este apartado se van a explicar todos los pasos necesarios para disponer de una base de datos en *SQL Azure*.

4.4.1.1. Creación de un nuevo Servidor en SQL Azure.

Los pasos a seguir para crear un nuevo servidor de bases de datos dentro de *SQL Azure* son los que se muestran a continuación:

1. Acceder al portal de gestión de la plataforma *Windows Azure*.

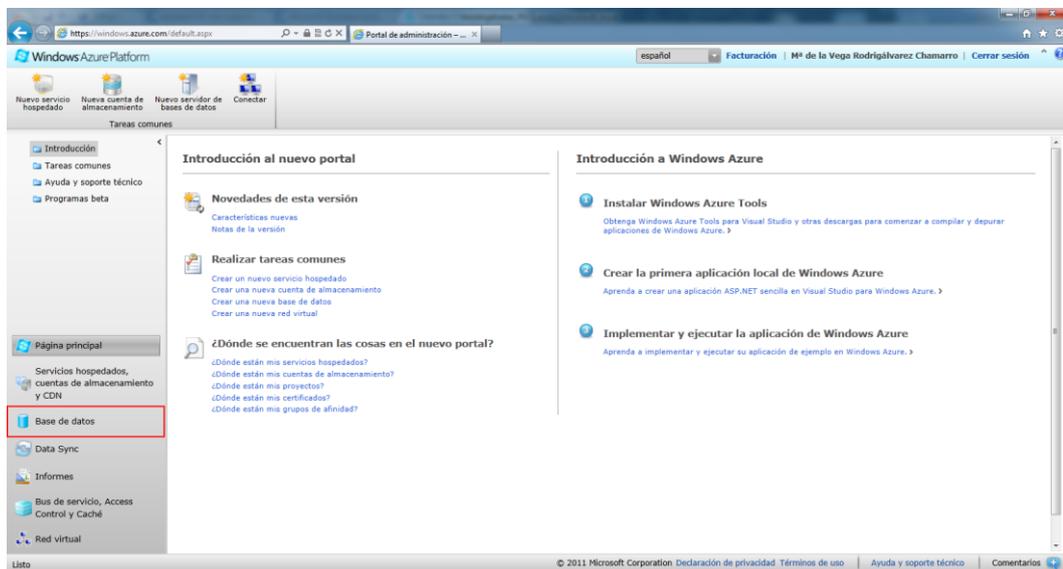


Figura 32. Portal Windows Azure.

2. Seleccionar la opción Base de Datos. Una vez seleccionada esta opción, aparece en pantalla la página de Bases de Datos. En primer lugar, será necesario crear un nuevo servidor donde se alojará la base de datos, para ello, pulsar el botón Crear.

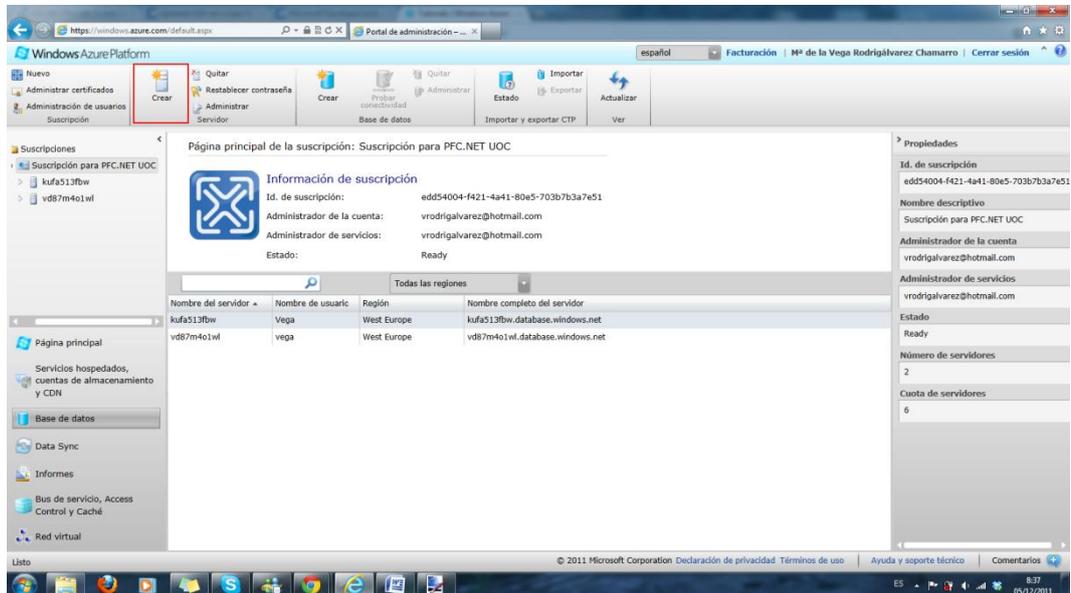


Figura 33. Crear un nuevo servidor de BD.

3. A continuación, aparece una ventana donde se deben introducir los datos del nuevo servidor que va a ser creado. En primer lugar, habrá que seleccionar una región (donde los datos van a ser almacenados). Es recomendable escoger una región cercana al lugar desde el cual se va a acceder (Ejem.: Oeste de Europa).



Figura 34. Seleccionar región.

4. Después, habrá que definir el inicio de sesión del administrador, para ello, será necesario introducir un login y una contraseña en base a las normas de seguridad establecidas por Microsoft.

Crear servidor

Crear un nuevo servidor

Especifique el inicio de sesión y la contraseña de la entidad de seguridad a nivel de servidor del servidor SQL Azure.

Inicio de sesión de administrador: admBD

Contraseña:

Repetir contraseña:

< Anterior Siguiente > Cancelar

Figura 35. Configurar inicio de sesión.

5. El siguiente paso consiste en la definición de las reglas de Firewall para el servidor. Las reglas de *Firewall* identifican las direcciones IP específicas que son capaces de comunicar directamente con el servidor *SQL Azure*. Para cada regla introducida es necesario especificar un nombre y el rango de direcciones IP a las que afecta. Pueden agregarse tantas reglas como se consideren necesarias.

Agregar regla de firewall

Agregar una nueva regla de firewall.

Nombre de regla: MyIPAddress

Inicio de intervalo IP: . . .

Final de intervalo IP: . . .

Su dirección IP actual: . . .

Aceptar Cancelar

Figura 36. Establecer rango de IPs.

6. Antes de finalizar, añadir una nueva regla, que permita a otros servicios de *Windows Azure* tener acceso a este servidor y pulsar finalizar para terminar la creación del servidor.

Crear servidor

Crear un nuevo servidor

Especifique una o más reglas de firewall que permitan el acceso al servidor SQL Azure. De lo contrario, no podrá conectarse a las bases de datos de este servidor ni administrarlo.

Nombre de regla	Inicio de intervalo IP	Final de intervalo IP
MicrosoftServices	0.0.0.0	0.0.0.0
MyIPAddress		

Agregar Actualizar Eliminar

Permitir que otros servicios Windows Azure tengan acceso a este servidor

< Anterior Finalizar Cancelar

Figura 37. Establecer rango de IPs de Windows Azure

- Una vez completados todos los pasos, se podrá ver que se ha creado un nuevo servidor dentro de la página de bases de datos. Las propiedades del servidor se pueden ver en la parte derecha de la ventana, estas propiedades se utilizarán para poder acceder a él desde cualquier aplicación que desee acceder a las bases de datos que se encuentre dentro del servidor.

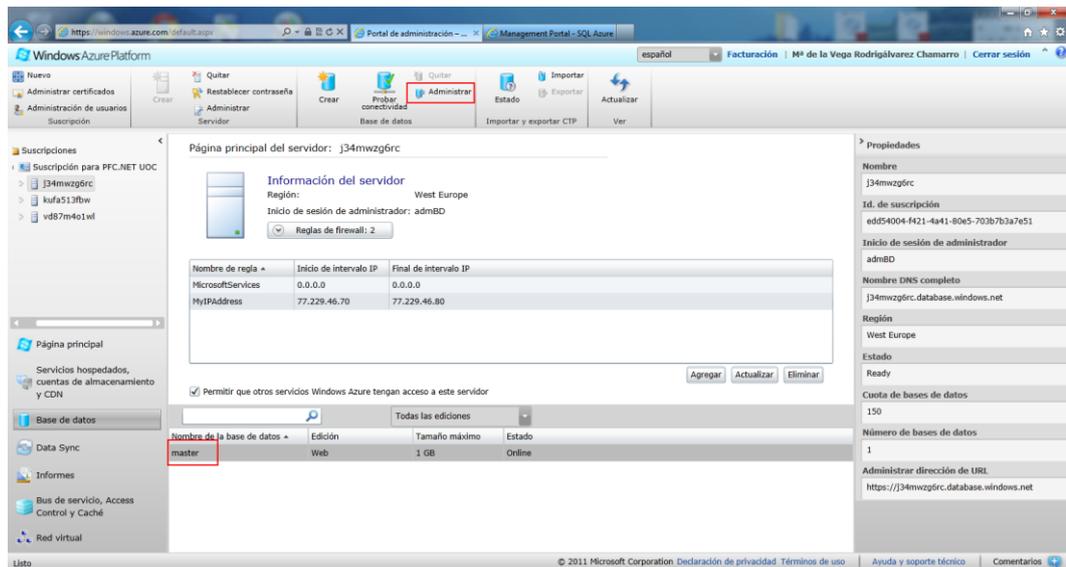


Figura 38. Servidor de BD creado.

- Para poder conectarse a la base de datos de SQL Azure desde la aplicación .NET será necesario configurar la cadena de conexión. El nombre del servidor que se debe utilizar en la cadena de conexión es la cadena que aparece debajo de la etiqueta Nombre DNS completo en la parte derecha de la ventana bajo el grupo de propiedades.

4.4.2. Creación de una nueva Base de Datos en SQL Azure

Para poder crear una nueva base de datos, utilizando scripts SQL, dentro de *SQL Azure* se ejecutarán los siguientes pasos:

- Desde el portal de gestión de *SQL Azure*. Para ello habrá que seleccionar la base de datos *master* que se encuentra debajo del nuevo servicio creado y pulsar la opción *Administrar*.

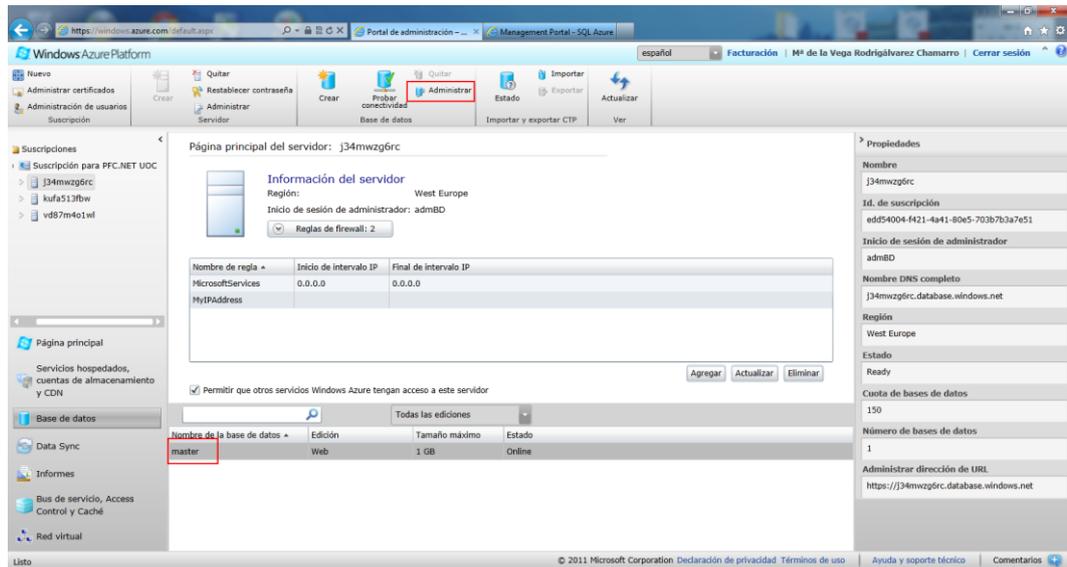


Figura 39. Creación de la BD sobre el servidor. Opción Administrar.

Se abre una nueva página donde el usuario deberá introducir, además de la BD a la que quiere acceder, el *login* de usuario y su contraseña.

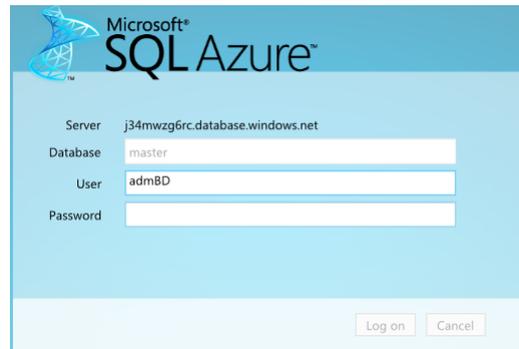


Figura 40. Acceso al portal de SQL Azure.

Si el *login* y la contraseña han sido correctamente introducidos, se accede al portal de administración de la base de datos seleccionada.

Si todos los datos se han ejecutado correctamente se puede proceder a ejecutar los diferentes scripts de base de datos.

- Abrir los diferentes ficheros que contienen los scripts de configuración de base datos y ejecutarlos uno a uno. Dentro del portal de gestión de *SQL Azure*, habrá que pulsar la opción *Abrir* que permite abrir ficheros que se encuentran localmente en el ordenador y, a continuación, seguir los pasos siguientes:
 1. Conectarse a la base de datos *master* y ejecutar el script `00_InstallAzure_UOC_IDEAS.sql`

2. Cambiar la conexión actual a la nueva base de datos creada (UOC_IDEAS). El usuario es el mismo que en la ocasión anterior.

Salir del portal y volver a acceder seleccionando en la pantalla de entrada el nombre de la nueva base de datos generada.

3. Ejecutar el script para crear las tablas relativas a la autenticación de usuarios *01_InstallAuthenticationTables.sql*.
4. Ejecutar el script para la generación de las tablas principales de las bases de datos *02_CreateTables.sql*.
5. Ejecutar el script para la generación de vistas *03_CreateView.sql*
6. Ejecutar los scripts necesarios para introducir los valores por defecto de la base de datos *04_InsertDefaultValues.sql*.

Una vez que las tablas han sido creadas sobre la base de datos y se han insertado los valores por defecto dentro de ella, la base de datos estará disponible para trabajar sobre ella desde cualquier aplicación.

Para poder conectarse desde una aplicación local a la base de datos alojada en SQL Azure, hay que indicar dentro de la plataforma de *Windows Azure* el rango de direcciones desde el cual se va a realizar la conexión.

4.4.3. Despliegue de un Web Role en Windows Azure

Lo primero que hay que hacer para poder desplegar una aplicación en la *nube* dentro de la plataforma *Windows Azure* será empaquetar la aplicación, para ello hay que seguir los siguientes pasos:

1. Situarse sobre el proyecto denominado **BraStor**, pulsar el botón derecho del ratón y seleccionar la opción *Package*.
2. Sobre la ventana que aparece en pantalla elegir las siguientes opciones y pulsar *Package*.
 - a. Service Configuration: *Cloud*.
 - b. Build Configuration: *Release*

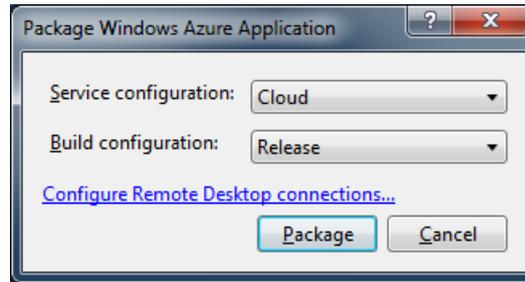


Figura 41. Empaquetar aplicación.

3. Cuando ha finalizado compilación y empaquetado de la aplicación, abre una carpeta donde se han generado los siguientes archivos:
 - a. BraStor.cspkg
 - b. ServiceConfiguration.Cloud.cscfg

Estos archivos serán los que se desplegarán dentro de *Windows Azure*, por lo tanto es importante recordar donde han sido almacenados.

Una vez empaquetada la aplicación se puede realizar el despliegue sobre la plataforma *Windows Azure*. Los pasos a seguir son:

1. Ir a la dirección <http://windows.azure.com> e iniciar sesión con el usuario asociado a la suscripción que se haya creado. Una vez iniciada la sesión se accede al portal de administración de *Windows Azure* donde se pueden administrar los diferentes servicios.

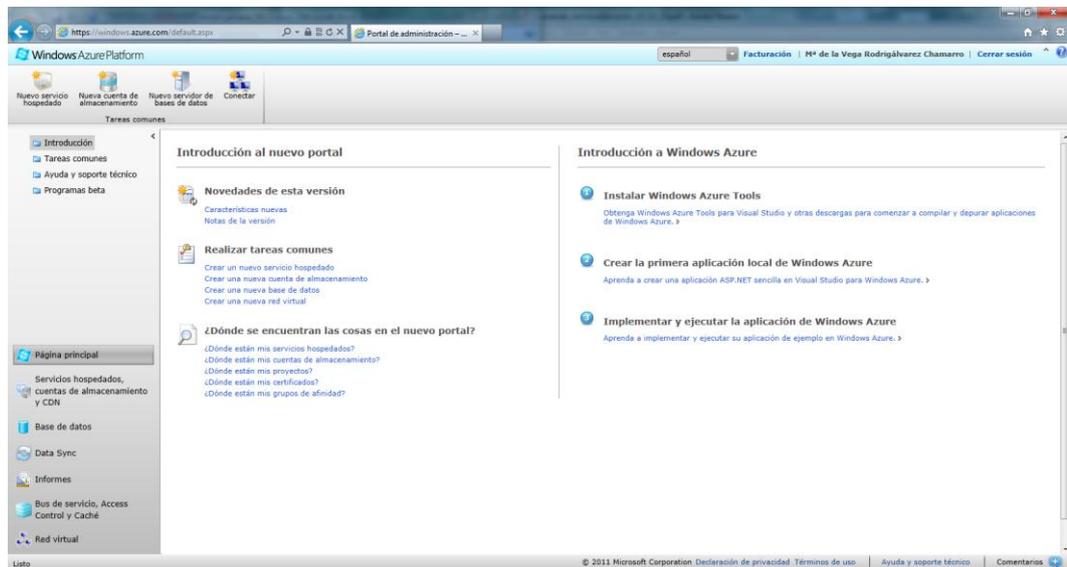


Figura 42. Portal de Administración de Windows Azure.

- Lo primero que hay que hacer es crear un nuevo servicio hospedado. Para ello, seleccionar sobre el menú inferior izquierdo la opción *Servicios hospedados, cuentas de almacenamiento y CDN*, sobre el menú que aparece en la parte superior elegir la opción denominada *Servicios hospedados* y hacer clic sobre el botón que aparece en la zona superior “*Nuevo servicio hospedado*”.

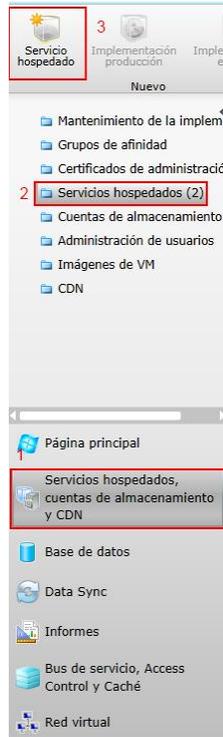


Figura 43. Nuevo servicio hospedado.

- Al presionar el botón, aparece una pantalla donde se introducen los parámetros necesarios para crear el nuevo servicio.

Hay que elegir la suscripción donde se quiere desplegar la aplicación y se le asignará un nombre al servicio (ejem.: *BrainStorming*).

A continuación hay que especificar la URL desde la cual se va a poder acceder al servicio. Esta URL debe ser única. (ejem.: *BrainStorming.cloudapp.net*).

Hay que elegir la región donde se quiere alojar el servicio, en este caso se elegirá el oeste de Europa por proximidad a la ubicación actual.

Seleccionar la opción *Implementar en entorno de producción* y se le dará un nombre a la implementación.

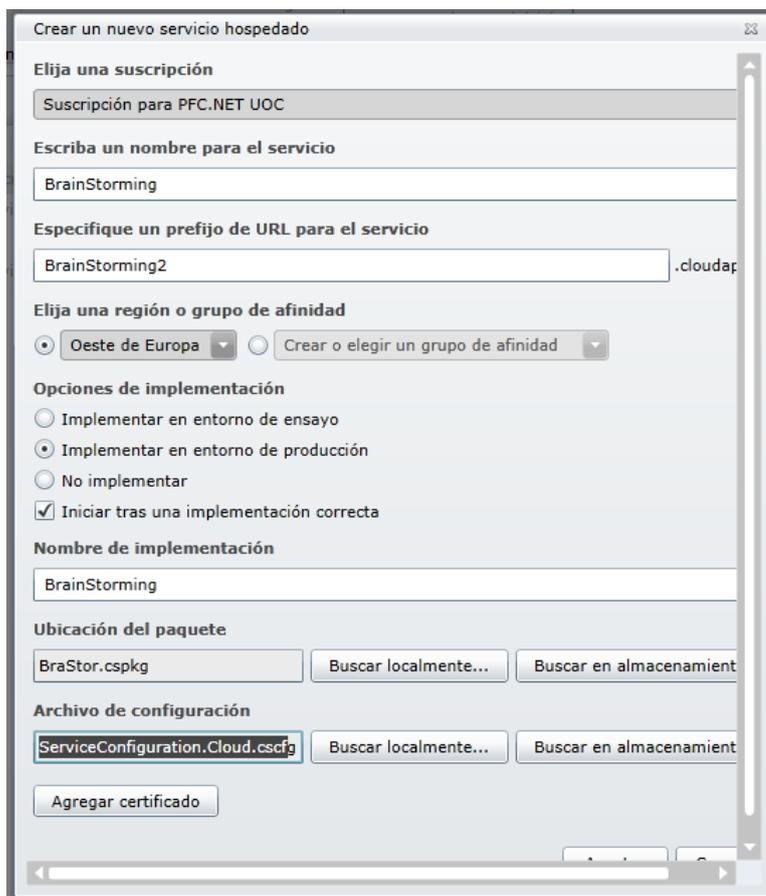


Figura 44. Despliegue aplicación en Windows Azure.

4. Sobre la misma ventana, introducir el archivo donde se ha empaquetado la aplicación y el archivo de configuración. Estos archivos son los que se han generado durante la paquetización de la aplicación. Para seleccionarlos desde el equipo local pulsar los botones *Buscar localmente*.
5. Una vez que se han introducido todos los datos correctamente pulsar el botón *Aceptar*.
6. En algunas ocasiones, aparece un mensaje de aviso, esto es porque únicamente se ha escogido una instancia para desplegar el role web, y suele ser recomendable configurar más de una. Pulsar *Sí* para continuar.

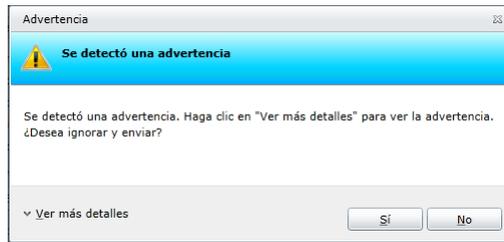


Figura 45. Advertencia número de instancias.

7. *Windows Azure* inicia el proceso de despliegue en la *nube*. Este proceso puede tardar varios minutos.
8. Una vez que se ha desplegado la aplicación, se puede acceder a la misma desde la dirección URL asignada con anterioridad. (ejem.: *BrainStorming.cloudapp.net*)

5. Objetivos Conseguidos

Los objetivos establecidos al inicio del proyecto se encuentran definidos en el apartado 2.1 del presente documento.

En vista de los resultados obtenidos, los objetivos principales del proyecto han sido alcanzados:

- El **principal objetivo** era realizar un análisis de los principales jugadores en la *nube* donde se definiera las capas en las que se movía cada uno y los servicios que cada uno de ellos proporcionaba para poder realizar una comparativa entre ellos. El estudio realizado sobre los principales proveedores aparece detallado en esta memoria en el apartado 3.2, donde además de analizar uno por uno los diferentes proveedores, se hace un estudio comparativo de los tres jugadores principales.
- Un paso previo al cumplimiento del objetivo principal del proyecto era explicar el paradigma de *Cloud Computing*, identificar los tipos de servicio que se podían ofrecer e identificar ventajas y posibles riesgos de trabajar en la *nube*. Este punto también ha sido cumplido satisfactoriamente y se ve reflejado en el apartado 3.1 del presente documento.
- El **objetivo final** del proyecto era la implementación de un prototipo utilizando alguno de los servicios o herramientas proporcionados por la *nube*. En esta ocasión, se ha realizado una aplicación básica utilizando las herramientas de programación que *Windows Azure* proporciona para *Visual Studio 2010*. Además se han utilizado los servicios de bases de datos relacional (*SQL Azure*) proporcionados por *Microsoft* en la *nube* y se ha mostrado como se una aplicación puede ser desplegada en la *nube*. Por el hecho de ser un prototipo, la implementación del mismo es muy básica y no está depurada al cien por cien, pero se ha conseguido interactuar con los servicios que ofrece *Microsoft* en la *nube*, conectando la aplicación con el servicio de datos

relacional que ofrece en SQL Azure, y se ha demostrado lo sencillo que puede llegar a ser desplegar una aplicación sobre la misma.

6. Evaluación de Costes

En el plan de proyecto se presentó el presupuesto del mismo. Para calcular los costes de las personas asignadas al proyecto se tomaron como referencia los valores de la siguiente tabla:

Tabla de precios	
Recurso	Coste/Hora
Jefe de proyecto	48 €
Analista	36 €
Analista Programador	24 €
Técnico de Sistemas	35 €

Figura 46. Tarifas Vigentes.

A continuación, se muestra una tabla con el coste previsto inicial del proyecto y el coste real del mismo.

Nombre de la Tarea	Trabajo por recurso	Trabajo	Coste Estimado	Trabajo real por recurso	Trabajo Real	Coste Real
PT1. Plan de Trabajo		36 h	1.728,00 €		24 h	1.152 €
<i>T1. Elaboración del PDT</i>		<i>30,3 h</i>			<i>20 h</i>	
Jefe de Proyecto	30,3 h		1.454,40 €	20 h		960 €
<i>T2. Elaboración del cronograma</i>		<i>5,7 h</i>			<i>4 h</i>	
Jefe de Proyecto	5,7 h		273,60 €	4 h		192 €
PT2. Análisis de jugadores en la nube		77,1 h	2.395,20 €		75,5 h	2.370 €
<i>T1. Formación Paradigma Cloud Computing</i>		<i>21,9 h</i>			<i>28.5 h</i>	
Documentador	7,3 h		175,20 €	8,5 h		204 €
Analista	14,6 h		525,60 €	20 h		720 €
<i>T2. Análisis jugadores en la nube</i>		<i>19,2 h</i>			<i>22 h</i>	
Desarrollador	6,4 h		153,60 €	5.5 h		132 €
Analista	12,8 h		460,80 €	16.5 h		594 €
<i>T3. Comparativa de servicios</i>		<i>36 h</i>			<i>25 h</i>	
Documentador	18 h		432,00 €	15 h		360 €
Analista	18 h		648,00 €	10 h		360 €
PT3. Desarrollo del prototipo		133,2 h	4.103,40 €		128 h	3.612 €
<i>T1. Estudio Plataforma Microsoft</i>		<i>14,55 h</i>			<i>19 h</i>	
Analista	14,55 h		523,80 €	19 h		684 €
<i>T2. Formación en Windows Azure</i>		<i>14,55 h</i>			<i>15 h</i>	

Desarrollador	4,85 h		116,40 €	9 h	216 €
Analista	4,85 h		174,60 €	3 h	108 €
Diseñador	4,85 h		174,60 €	3 h	108 €
T3. Instalación y Configuración entorno desarrollo		9,6 h		3 h	
Desarrollador	9,6 h		230,40 €	3 h	72 €
T4. Análisis de requisitos		8,4 h		10 h	
Analista	8,4 h		302,40 €	10 h	360 €
T5. Diseño del prototipo		26,1 h		10 h	
Diseñador	26,1 h		939,60 €	10 h	360 €
T6. Implementación		38,7 h		45 h	
Desarrollador	38,7 h		928,80 €	45 h	1080 €
T7. Verificación y pruebas		10,2 h		6 h	
Desarrollador	10,2 h		244,80 €	6 h	144 €
T8. Manual de Usuario		19,5 h		20 h	
Documentador	13 h		312,00 €	15 h	360 €
Desarrollador	6,5 h		156,00 €	5 h	120 €
PT4. Difusión de resultados		56,4 h	1.579,20 €	67 h	1.776 €
T1. Memoria		28,2 h		30 h	
Documentador	28,2 h		676,80 €	30 h	720 €
T2. Presentación		28,2 h		37 h	
Documentador	18,8 h		451,20 €	30 h	720 €
Jefe de Proyecto	9,4 h		451,20 €	7 h	336 €
PT5. Gestión del proyecto		24,23 h	1.172,40 €	10 h	480 €
T1. Seguimiento del proyecto		24,43 h		10 h	
Jefe de Proyecto	24,43 h		1.172,40 €	10 h	480 €
Total		335,53 h	10.978,20 €	304,5 h	9.390 €

7. Puntos de Mejora

Aunque los objetivos del proyecto han sido alcanzados satisfactoriamente, siempre quedan opciones para poder seguir trabajando y profundizando en el tema. Como posibles líneas de trabajo que continuarán el trabajo comenzado en este proyecto podrían ser:

- Como el entorno del *Cloud Computing* es un mundo cambiante y que avanza a pasos agigantados, se podría continuar y profundizar el estudio de los diferentes proveedores analizando si ofrecen nuevos servicios, si han ampliado su oferta a nuevas capas, si han variado las técnicas de trabajar en la nube... También se podría estudiar si surgen nuevos proveedores en la *nube* capaces de marcar la diferencia con respecto a los que ahora existe.
- A lo largo del proyecto, se ha visto que un punto que marca mucha polémica en estos momentos son los acuerdos legales y la problemática existente en cuanto a la

ubicación de los datos, seguridad de la información que se almacena en la nube, quienes son los responsables de la información alojada en la *nube*, que sucede cuando hay interrupciones del servicio, quien se hace responsable, etc. Por lo tanto, se podría realizar un estudio más en profundidad para ver qué pasos debe tener en cuenta una empresa antes de pasar sus datos y sus servicios a la *nube*.

- En España, aquellos datos que se consideran críticos, tales como datos bancarios..., no pueden estar alojados fuera del territorio nacional, esto hace complicado que ciertas organizaciones puedan integrar sus servicios en la *nube*. Un posible trabajo podría ser estudiar y desarrollar una *nube* híbrida que permita que los datos se encuentren dentro de una *nube* privada y los servicios que proporciona la organización se encuentren en la *nube* pública, de forma que se cumplan todos los requisitos legales que marca la ley española.
- Un paso adelante del presente proyecto, sería la implementación del prototipo usando las herramientas de desarrollo que ofrecen el resto de proveedores y de esa forma, poder hacer una comparación más práctica entre los diferentes proveedores.
- Una gran preocupación en el mundo del *Cloud Computing* es el tema de la seguridad y el acceso a la información por parte de personas no autorizadas. Se podría realizar un estudio que buscara formas más seguras de autenticación en la nube: modos de acceso, políticas de restricción, permisos,...
- En cuanto a la implementación del prototipo se podrían realizar las siguientes mejoras:
 - Utilización de controles de acceso federados, de esta forma, se podría ver el funcionamiento de otro servicio ofrecido por *Microsoft* en la *nube*.
 - Implementación de una aplicación usando *worker role* y así comparar ambos roles y ver cómo se comunican entre ellos.
 - Trabajar con sistemas de almacenamiento no relacionales.

8. Conclusiones

La elaboración de este proyecto me ha permitido acceder a un mundo que actualmente no había visto con demasiada profundidad, en base a ello, he obtenido las siguientes conclusiones:

- El entorno del *Cloud Computing* es un mundo muy amplio que toca un gran número de aspectos de la informática.
- Para usuarios particulares, servicios como los proporcionados en la capa de *SaaS*, hace que sea mucho más sencillo acceder a ellos sin necesidad de un desembolso económico importante. Servicios ofrecidos por *Google* o *Microsoft* hace que cualquier persona pueda tener una cuenta de correo con gran cantidad de espacio, compartir sus fotos con otras personas...

- Para las organizaciones también supone un beneficio poder trabajar en la *nube*, ya que dependiendo del tamaño de la organización, hace que puedan desentenderse de costes que antes les suponía un gran esfuerzo mantener en la propia empresa. Ahora existen proveedores externos que pueden mantener la infraestructura de *hardware* y *software* sin que la organización tenga que tener recursos específicos para ello, de esa forma, se pueden dedicar al núcleo de su negocio.
- Como punto negativo, el tema de la seguridad o las interrupciones en el servicio. Aunque los diferentes proveedores aseguren en su oferta que está controlado en un 99.9%, no es algo que esté totalmente controlado y aunque raras veces sucede una interrupción del servicio, con la consecuente no disponibilidad de los datos, cuando ocurre, se han dado casos de que las empresas no han podido acceder a sus datos durante varios días (*Amazon* debido a la caída de un rayo en su centro de datos de Dublín)¹⁴, puede causar grandes pérdidas a las organizaciones y la compensación por parte de los proveedores es demasiado baja. Esto hace que grandes organizaciones tengan serias dudas de dar el salto a la *nube*.
- Otro punto a destacar, es la localización de los datos. Cada país, tiene su propia ley de protección de datos y hay que llevar mucho cuidado a la hora de subir datos de cierta naturaleza a la nube.
- También es necesario revisar con mucho detalle los contratos que se firman con los diferentes proveedores.

Cloud Computing es una tecnología que ha supuesto un cambio radical con lo que hasta ahora se conocía y la forma tradicional de trabajar de las empresas, pero antes de considerarla una tecnología robusta y afianzada, donde muchas organizaciones están uniéndose a ella, será necesario afianzar más temas como la seguridad y disponibilidad de la información.

¹⁴ <http://www.genbeta.com/web/amazon-cloud-sin-servicio-en-la-zona-europea-debido-a-un-rayo>

9. Siglas y Acrónimos

Acrónimo	Descripción
ADO	ActiveX Data Objects
AJAX	Aysnchronous JavaScript and XML
AMI	Amazon Machine Image
API	Application Programming Interface
AWS	Amazon Web Services
BLOBS	Binary Large Objects
CDN	Content Distribution Network
CoD	Compute on Demand
CPU	Central Processing Unit
CRM	Customer Relationship Management
DB	DataBase
DNS	Domain Name System
EC2	Amazon Elastic Compute Cloud
EBS	Elastic Block Storage
FIFO	First Input First Output
GAE	Google App Engine
GFS	Google File System
GQL	Google Query Language
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IaaS	Infraestructure as a Service
IDE	Integrated Development Environment
IIS	Internet Information Service
IP	Internet Protocol
IPSec	Internet Protocol Security
ISV	Independent Software Vendor
ITaaS	Information Technologies as a Service
JDBC	Java Database Connectivity
LOPD	Ley de Protección de Datos.
NAT	Network Address Translation
NIST	National Institute of Standards and Technology
NTFS	New Technology File System
ODBC	Open Database Connectivity
PaaS	Platform as a Service
PDA	Personal Digital Assistant
PDT	Plan de Proyecto
PFC	Proyecto Final de Carrera

PHP	Hypertext Preprocessor
PKI	Public Key Infrastructure
PYMES	Pequeñas y Medianas Empresas
QoS	Quality of Service
REST	Representation State Transfer
RSA	Rivest, Shamir y Adleman
S3	Amazon Simple Storage Service
S+S	Software plus Service
SaaS	Software as a Service
SDK	Software Development Kit
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
SQS	Simple Queue Service
SSO	Single Sign On
TCP	Transmission Control Protocol
TDS	Tabular Data Stream
TI	Tecnologías de la Información
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
VHD	Virtual Hard Disk
VM	Virtual Machine
VPN	Virtual Private Network
WCF	Windows Communication Foundation

10. Bibliografía

- Alarcon, J. (2011, Noviembre 11). *Proveedores universales de ASP.NET: Membership y Roles fuera de SQL Server*. Retrieved Noviembre 2011, from <http://geeks.ms/blogs/jalarcon/archive/2011/11/11/proveedores-universales-de-asp-net-membership-y-roles-fuera-de-sql-server.aspx>
- Ariz Lopez de Castro, E. (10 de 2010). *Aspectos Legales Cloud Computing*. Recuperado el 10 de 2011, de <http://www.slideshare.net/enekoariz/aspectos-legales-cloud-computing>
- Astrea It Services. (s.f.). *Cloud computing Tutorial*. Recuperado el 10 de 2011, de <http://thecloututorial.com/index.html>
- Chappel, D. (2010). *Introducing the Windows Azure Platform*. San Francisco: David Chappell & Associates.
- Cierco, D. (2011). *Cloud Computing: Retos y oportunidades*. Madrid: Fundacion Ideas.
- Furht, B., & Escalante, A. (2010). *HandBook of Cloud Computing*. Springer.
- Fundación Bankinter,Accenture. (s.f.). *Cloud Computing: la tercera ola de las tecnologías de la información*. Recuperado el 01 de 10 de 2011, de <http://www.fundacionbankinter.org/es/publications/cloud-computing>
- Hall, A. (2001, Febrero 17). *A short introduction to Windows Azure AppFabric*. Retrieved Diciembre 2011, from <http://www.cloudave.com/10053/a-short-introduction-to-windows-azure-appfabric/>
- John Wiley & Sons. (2011). *Cloud Computing. Principles and Paradigms*. Canada: Wiley.
- Krasis. (s.f.). *¿Qué es Cloud Computing?* Recuperado el 01 de 10 de 2011, de <http://msdn.microsoft.com/es-es/windowsazure/gg318630.aspx>
- Krasis. (s.f.). *¿Qué es Windows Azure Platform?* Recuperado el 01 de 10 de 2011, de
- Microsoft. (s.f.). Obtenido de <http://msdn.microsoft.com/es-es/windowsazure/gg318632.aspx>
- Landa Matín, I., & Zorilla Castro, U. (2011, Noviembre). *Almacenamiento en Windows Azure*. Retrieved Diciembre 2011, from <http://es.scribd.com/doc/72993459/46/WINDOWS-AZURE-DRIVE>
- Microsoft. (2009, Noviembre). *Updated ASP.net scripts for use with Microsoft SQL Azure*. Retrieved Noviembre 2011, from <http://support.microsoft.com/kb/2006191/es>
- Microsoft. (2011, Diciembre). *Windows Azure*. Retrieved Diciembre 2011, from www.windowsazure.com
- Nagy, S. (2009, Marzo). *The Azure Fabric Controller*. Retrieved Noviembre 2011, from <http://azure.snagy.name/blog/?p=89>
- Panzano, L. (2010, Julio 18). *La Plataforma Windows Azure*. Retrieved Noviembre 2011, from <http://blogs.msdn.com/b/luispanzano/archive/2010/07/19/la-plataforma-windows-azure.aspx>

- Plain Concepts. (2011). *Qué es Windows Azure Fabric*. Retrieved Noviembre 2011, from <http://www.estoyenlanube.com/recursos/windows-azure-appfabric/que-es-windows-azure-appfabric/>
- Qian, L., Luo, Z., Du, Y., & Guo, L. (2009). Cloud Computing: An overview. Lecture Notes in Computer Science, (pág. 6).
- Redkar, T. (2009). *Windows Azure Platform*. Apres.
- Rodriguez, D. (2011, Marzo). *Primer vistazo a SQL Azure Reporting CTP*. Retrieved Diciembre 2011, from <http://geeks.ms/blogs/davidjrh/archive/2011/03/22/primer-vistazo-a-sql-azure-reporting-ctp.aspx>
- Shroff, G. (2010). *Enterprise Cloud Computing. Technology, Architecture, Applications*. Cambridge: Cambridge.
- Varios. (s.f.). *Cloud Platforms Vendors*. Recuperado el 1 de 10 de 2011, de <http://geekswithblogs.net/gotchass/archive/2010/08/09/cloud-platform-vendors.aspx>
- Varios. (s.f.). *Computación en nube*. Recuperado el 1 de 10 de 2011, de Wikipedia: http://es.wikipedia.org/wiki/Computaci%C3%B3n_en_nube
- Velte, A. T., Velte, T. J., & Elsenepeter, R. (2010). *Cloud computing. A practical Approach*. McGraw-Hill.
- Villegas Carrasquilla, L. (s.f.). *Cloud computing y sus aspectos legales*. Recuperado el 10 de 2011, de <http://www.mercadodedinero.com.co/Tecnologia/cloud-computing-y-sus-aspectos-legales.html>
- Wikipedia. (2010, Febrero). *Azure Services Platforms*. Retrieved Noviembre 2011, from http://en.wikipedia.org/wiki/Azure_Services_Platform