

DESARROLLO DE APLICACIONES PARA CLOUD COMPUTING

María Murazzo¹, Flavia Millán, Nelson Rodríguez¹, Daniela Segura², Daniela Villafañe²

maritemurazzo@gmail.com, flavia.millan@gmail.com, nelson@iinfo.unsj.edu.ar,
danycec_16@hotmail.com, jacky1552@hotmail.com

Universidad Nacional de San Juan, Facultad de Ciencias Exactas, Físicas y Naturales

RESUMEN: La cloud computing es vista por algunos autores como la Cuarta Generación de aplicaciones y se espera que sea la forma en que en estos próximos años se consumirán los recursos informáticos.

Lograr un Cloud abierto, que soporte la diversidad de equipos y que funciones adecuadamente además para plataformas heterogéneas es el desafío más importante que se presenta para los diseñadores de sistemas.

El objetivo que se pretende alcanzar con este trabajo de investigación es la construcción de una plataforma para construir Cloud Públicas y Abiertas, que sean accedidas desde cualquier tipo de dispositivo y se comporte conforme a las características del cliente.

Palabras Claves: Cloud Computing, Web Service, Windows Azure.

¹ Miembros del Proyecto de investigación Arquitectura de acceso a servicios Web desde dispositivos móviles heterogéneos (21 E 830)

² Alumnos tesistas de la Carrera de Licenciatura en Ciencias de la Información.

1 Introducción

En los últimos años se ha visto evolucionar tecnologías vitales para el mundo empresarial en lo que a TIC (Tecnologías de Información y Comunicaciones) se refiere, tales como los servicios de telefonía, las telecomunicaciones, los datacenter, etc.

Así que la pregunta es, ¿por qué no conectarme a Internet y que alguien suministre todos los servicios de computación que la empresa necesita de manera simple y se facture mensualmente por ello?, de esta forma todo lo que sea computación se convierta en una utility más.

Esa idea no es nueva, se viene trabajando en este concepto desde hace algunos años, conceptos precursores son *utility computing*, *computación bajo demanda*, *computación elástica* o *grid computing* [1].

Internet usualmente se visualiza y conceptualiza como una gran nube donde todo está conectado y donde al conectarse se suministran todos los servicios requeridos. A este esquema de trabajo se lo denomina *Cloud Computing* o *Computación en Nube*, la cual es similar a todos los esquemas antes nombrados, pero potenciada con las tecnologías de virtualización [2].

2 ¿Qué es el Cloud Computing?

El concepto de Cloud Computing tiene como principal característica, la transformación de los modos tradicionales, de cómo las empresas utilizan y adquieren los recursos de TI (Tecnología de la Información).

Cloud Computing, representa un nuevo tipo del valor de la computación en red. Entrega mayor eficiencia, escalabilidad masiva y más rápido y fácil desarrollo de software. Los nuevos modelos de programación y la nueva infraestructura de TI permitirán que surjan nuevos modelos de negocios.

La Cloud Computing es un modelo de aprovisionamiento de recursos TI que potencia la prestación de servicios TI y servicios de negocio, facilitando la operativa del usuario final y del prestador del servicio.

Una de las principales ventajas para las empresas que deciden incorporar a sus actividades servicios prestados a través de Internet es la posibilidad de reducir sus gastos de personal técnico, instalaciones, software y, sobre todo, de tareas de mantenimiento; de esta manera el retorno de la inversión es inmediato, ya que no es necesaria preinstalación ni configuración alguna.

Todo ello se realiza de manera fiable y segura, con una escalabilidad elástica, que es capaz de atender fuertes cambios en la demanda no previsible a priori, sin que esto suponga un incremento en los costos de gestión.

La característica básica de este modelo es que los recursos y servicios informáticos, tales como infraestructura, plataforma y aplicaciones, son ofrecidos y consumidos como servicios a través de Internet sin que los usuarios tengan que tener ningún conocimiento de lo que sucede detrás.

La Cloud Computing es un esquema del tipo aaS o as a Service y que a veces se expresa como XaaS o EaaS para significar Everything as a Service. En general cualquier cosa como un servicio.

Se puede dividir al Cloud Computing en tres capas: *Software como Servicio (SaaS)*, *Plataforma como Servicio (PaaS)* e *Infraestructura como Servicio (IaaS)* [3] [4].

Cualquier organización que desee servicios de TICs podrá implementar un esquema XaaS y eliminar todos sus requerimientos internos y contratar sus necesidades en estas áreas externamente a cambio de un pago mensual, sin inversiones de capital.

2.1 Características del Cloud Computing

Una de las principales características del Cloud Computing es que no hay necesidad de conocer la infraestructura detrás de esta, pasa a ser “una nube” donde las aplicaciones y servicios pueden fácilmente crecer (escalar), funcionar rápido y casi nunca fallan, sin conocer los detalles del funcionamiento de esta “nube”.

Este tipo de servicio se paga según alguna métrica de consumo, no por el equipo usado en sí, sino por ejemplo en el consumo de electricidad o por uso de CPU/hora. Entre otras características podemos mencionar: *auto reparable*, *escalable*, *regido por un acuerdo de nivel de servicio (SLA)*, *virtualizado* y *multipropósito*.

La consultora IDC estima que en los próximos cinco años el gasto en Cloud Computing se triplicará alcanzando la cifra de 42 mil millones de dólares, contabilizando el 9% de los ingresos en cinco segmentos del mercado claves. Pero lo más importante es que el gasto en el período se acelerará hasta capturar el 25% del gasto en IT en el 2012 y casi un tercio en el 2013.

Frank Gens, Senior VP y Analista jefe en IDC dice: “Un reciente estudio entre Ejecutivos de IT, CIOs y los colegas en las líneas de negocio muestra que el Cloud Computing está ‘cruzando el abismo’ y entrando en un período de amplia adopción.

Más aún, la crisis económica amplificará la adopción del cloud. Este modelo ofrece una manera más barata para que el negocio use y adquiera tecnología. Esta ventaja es

verdaderamente importante para los pequeños y medianos negocios, un sector que será clave en cualquier plan de recuperación [5].

3 Planteo del problema

Las exigencias y requerimientos de los usuarios tanto a nivel profesional como social han cambiado y se han ampliado. Las principales características que se requieren de las aplicaciones actualmente son tres: *ubicuidad, disponibilidad y seguridad*.

Sin embargo, las necesidades de los usuarios cambian al ritmo de la tecnología y se esta imponiendo las necesidades de *localización, inmediatez y personalización* [6]. Debido a estas exigencias, se hace necesario depender de la cloud para la distribución de los servicios.

Las aplicaciones comenzarán a almacenar los datos en la cloud en lugar de en la PC y serán más poderosas debido a que el poder de procesamiento también es descargado de la cloud.

Los entornos cloud simplifican el trabajo de desarrollo de aplicaciones y también su utilización. Se trabaja sobre una plataforma agnóstica en cuanto a sistema operativo y donde una misma aplicación corre sobre diferentes plataformas de usuarios finales.

Las aplicaciones cloud sacan a la computación y al almacenamiento de datos fuera del escritorio para ubicarlos en el cloud. Así, esas aplicaciones se pueden usar en una variedad de plataformas de hardware, que incluye: *PC y dispositivos móviles* [7].

En función de lo mencionado anteriormente, este trabajo tiene como objetivo principal el estudio de una arquitectura de desarrollo de aplicaciones ubicuas basadas en servicios en el cloud.

El principal objetivo de las aplicaciones ubicuas es el establecimiento de entornos, donde cualquier dispositivo de acceso, puedan comunicarse en forma consistente y transparente al usuario.

El concepto de ubicuidad que se pretende en este trabajo, es lograr que el diseñador de una aplicación que estará almacenada en la cloud, pueda ser accedida por múltiples usuarios, independientemente del dispositivo de acceso que se este usando.

4 Arquitectura Propuesta

Para lograr el desarrollo de aplicaciones ubicuas, es necesario independizar la interface de las aplicaciones del código. Esto se puede lograr mediante el uso de la Arquitectura MVC (Modelo – Vista – Controlador). MVC es un patrón de arquitectura de aplicación que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

La arquitectura MVC fue introducida como parte de la versión Smalltalk-80 del lenguaje de programación Smalltalk. Fue diseñada para reducir el esfuerzo de programación necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos.

Sus características principales son que el Modelo, las Vistas y los Controladores se tratan como entidades separadas; esto hace que cualquier cambio producido en el Modelo se refleje automáticamente en cada una de las Vistas

Este modelo de arquitectura presenta varias ventajas:

- Hay una clara separación entre los componentes de un programa; lo cual permite implementarlos por separado
- Hay una API muy bien definida; cualquiera que use la API, podrá reemplazar el Modelo, la Vista o el Controlador, sin aparente dificultad.
- La conexión entre el Modelo y sus Vistas es dinámica; se produce en tiempo de ejecución, no en tiempo de compilación

El Modelo es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El Modelo no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas, y notificar a las Vistas cuando cambia el Modelo.

La Vista es el objeto que maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa con el Modelo a través de una referencia al propio Modelo.

El Controlador es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo a través de una referencia al propio Modelo.

El lenguaje de programación Java proporciona soporte para la arquitectura MVC mediante dos clases: Observer y Observable. El Modelo es un subtipo de Observable y la Vista es un subtipo de Observer.

Por otro lado, el Framework Microsoft's ASP.NET MVC, ofrece un nivel de separación de la capa presentación que permite soportar múltiples dispositivos sin modificar ni duplicar la mayoría del código del sitio.

Por todo lo mencionado, haciendo uso de la Arquitectura MVC, se podrá desarrollar una única lógica de aplicación y será la interface de usuario la que se deberá adaptar al dispositivo de acceso que se este usando.

En función de lo analizado, se pretende la migración del concepto de Aplicaciones basadas en Web a Aplicaciones basadas en Cloud donde la virtualización es un aspecto fundamental junto con la plataforma de desarrollo usada.

Teniendo en cuenta la Arquitectura MVC que se sugiere usar, se propone una implementación basada en Web Service. Esta elección se debe a que los Web Service son la evolución de las tradicionales aplicaciones cooperativas.

En términos simples, los Web Service proponen convertir a los recursos de software en servicios disponibles para quien los requiera. Conceptualmente, la idea es crear componentes reutilizables, fáciles de emplear y de mantener, en lugar de aplicaciones complejas.

El consorcio W3C define los Servicios Web como sistemas software diseñados para soportar una interacción interoperable maquina a maquina sobre una red. Los Servicios Web, suelen ser APIs Web que pueden ser accedidas dentro de una red (principalmente Internet) y son ejecutados en el sistema que los aloja.

La definición de Servicios Web propuesta se refiere a clientes y servidores que se comunican mediante mensajes XML (Extensible Markup Language) que siguen el estándar SOAP (Simple Object Access Protocol).

Los Servicios Web han evolucionado desde una primera generación que no explotaba todos sus beneficios, al no integrarse con otros servicios Web, hasta una nueva generación de servicios Web en la cual todos los servicios son típicamente compuestos, es decir construidos a partir de otros servicios más primitivos Web.

Lo servicios Web presentan grandes ventajas como:

- Aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen.
- Fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.
- Permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados.
- Permiten la interoperabilidad entre plataformas de distintos fabricantes por medio de protocolos estándar y abiertos.

En los últimos años, se ha popularizado un estilo de arquitectura Software REST (Representational State Transfer). Los Servicios Web que funcionan bajo REST (servicios Web RESTful) se presentan como una alternativa prometedora distinta a los servicios basados en SOAP por su simplicidad y naturaleza liviana, además de la capacidad de transmitir datos directamente sobre HTTP.

¿Por qué surge el debate entre los Servicios Web basados en REST y SOAP? Muchos diseñadores de Servicios Web han llegado a la conclusión que SOAP es demasiado

complicado. Por tanto, están comenzando a utilizar Servicios Web basados en REST para mostrar cantidades de datos masivos. Este es el caso de grandes empresas como eBay y Google.

Las ventajas de REST recaen en la potencial escalabilidad de este tipo de sistemas, así como el acceso con escaso consumo de recursos a sus operaciones debido al limitado número de operaciones y el esquema de direccionamiento unificado [8] [9].

Para lograr implementar esto, los desarrolladores deben usar un PaaS capaz de soportar el desarrollo de aplicaciones para cloud y que sea accedido desde dispositivos heterogéneos.

4.1 Propuesta de Implementación

Un PaaS, ofrecen la posibilidad de desplegar aplicaciones sin tener que ocuparte del mantenimiento de la infraestructura. Las distintas plataformas suelen especializarse en un determinado lenguaje de programación, servidor de aplicaciones y base de datos, como por ejemplo, *Google App Engine* que ofrece la posibilidad de desplegar aplicaciones Phyton o *Windows Azure* que permite implementaciones basadas en .NET, Java o Phyton.

Un PaaS, debe ser capaz de contener todos los componentes necesarios (servidor de aplicaciones, base de datos, frameworks, etc.) para poder desplegar y ejecutar las aplicaciones. Debe tener un rendimiento óptimo, deber ser escalable y ofrecer seguridad y privacidad de la información que almacena. El propietario de la aplicación solo se debe preocupar del buen funcionamiento de la aplicación, pues la escalabilidad, el balanceo de carga, y todo aquello que suponga el mantenimiento de la plataforma corre a cargo del proveedor PaaS.

Para este trabajo se opto por usar como PaaS a Windows Azure [10]. Este PaaS, es un sistema operativo de servicios de la cloud que sirve como entorno de desarrollo, servicio de hosting y administración de servicios para la plataforma de servicios Azure. Ofrece a los desarrolladores con necesidades de computación y almacenamiento bajo demanda, a desplegar, escalar y administrar sus aplicaciones Web en Internet a través de los datacenter de Microsoft. Estos datacenter estan, distribuidos por todo el mundo, garantizando la disponibilidad del servicio, seguridad de acceso e integridad de los datos. Ese alojamiento, tendrá un costo, pero las empresas no tendrán que montar y, sobre todo, mantener su propio centro de datos ni preocuparse por aspectos como la seguridad del hardware y el software. En función de esto se puede decir que Windows Azure es un PaaS y un IaaS.

Azure es una plataforma flexible e interoperable, que permite múltiples protocolos de Internet como HTTP, REST, SOAP y XML. Los entornos de desarrollo soportados pueden ser Microsoft Visual Studio, .NET Framework y Eclipse. Los lenguajes con los que se pueden escribir las aplicaciones son .NET (C# y Visual Basic), C++; PHP, Ruby, Python y Java.

Los tres componentes principales de esta plataforma son:

- *Windows Azure*: es el sistema operativo utilizado por la plataforma.
- *SQL Azure* (anteriormente conocido como Microsoft SQL Services): es el motor de base de datos en la plataforma Windows Azure.
- *AppFabric* (anteriormente conocido como .NET Services): es el componente de middleware que consiste en servicios como ServiceBus y Access Control.

4.2 Un Caso Práctico

La aplicación desarrollada esta destinada a niños de edad escolar (véase la figura 1). La cual incluye juegos interactivos, posibilitando el desarrollo biológico, psicológico y social del niño. Los juegos que se destacan son de Agilidad (juegos que permiten cambiar de posición en el espacio y hacer recorridos con varias posiciones corporales), Auditivos (juegos que ayudan a perfeccionar la función sensorial auditiva), e Intelectuales (hacen intervenir la comparación de fijar la atención de dos o más cosas para descubrir sus relaciones, como el dominio, el razonamiento la reflexión y la imaginación creadora).

Además la aplicación está orientada a permitir la interacción entre el docente y el alumno, reafirmando de esta manera los conocimientos adquiridos en clase.



Figura 1: Capturas de pantallas de la Aplicación corriendo en el Cloud

5 Conclusiones

Como ya se ha mencionado en el documento, las empresas están haciendo una importante inversión en Cloud Computing y cada vez se esta popularizando mas los dispositivos móviles y las aplicaciones capaces de hacer uso de todo su potencial.

De esta manera será necesario el desarrollo de aplicaciones, almacenadas en el cloud y accedidas en forma transparente por los usuarios sin importan el tipo de dispositivo que estén usando.

En función de todo lo analizado, el objetivo de este trabajo es sentar las bases para la construcción de un Cloud Publico con capacidad de brindar servicios ubicuos, que cumplan con los estándares de usabilidad, a dispositivos heterogéneos.

6 Bibliografía

- [1] Jinzy Zhu, Xing Fang, Zhe Guo, Meng Hua Niu, Fan Cao, Shuang Yue and Qin Yu Liu, IBM Cloud Computing Powering a Smarter Planet, Libro Cloud Computing, Volumen 5931/2009, Páginas 621-625.
- [2] Gansen Zhao, Jiale Liu, Yong Tang, Wei Sun, Feng Zhang, Xiaoping Ye and Na Tang, Cloud Computing: A Statistics Aspect of Users, Libro Cloud Computing, Volumen 5931/2009, Páginas 347-358.
- [3] Andrew Weiss, Computing in the clouds. netWorker 11, 4 (Dec. 2007), 16-25. DOI= <http://doi.acm.org/10.1145/1327512.1327513>
- [4] Srinivasa Rao, Nageswara Rao, Kusuma Kumari, Cloud Computing: An Overview. Queue 7, 5 (Jun. 2009), 3-4. DOI= <http://doi.acm.org/10.1145/1538947.1554608>.
- [5] IDC, IDC Finds Cloud Computing Entering Period of Accelerating Adoption and Poised to Capture IT Spending Growth Over the Next Five Years, <http://www.idc.com/getdoc.jsp?containerId=prUS21480708>
- [6] Canaleta y Vernet, TIC versus mTIC, Actas de las V Jornadas de Informática y Sociedad (JIS'2004). Marzo 2004, pagina 61-67
- [7] Herman Mehling, Cloud-based Mobile Applications On the Rise, <http://www.devx.com/wireless/Article/44394?trk=DXRSS>.
- [8] William Brogden, REST versus SOAP – the REST story, http://searchwebservices.techtarget.com/tip/0,289483,sid26_gci1227190,00.html.
- [9] William Brogden, REST versus SOAP – the SOAP story, http://searchwebservices.techtarget.com/tip/0,289483,sid26_gci1231889,00.html.
- [10] Microsoft. “Azure Academic Pilot”. www.azurepilot.com