



Máster Oficial en Tecnologías de la Información y Sistemas Informáticos

**Curso Académico 2010/2011
TESIS FIN DE MASTER**

Cloud Computing: fundamentos, diseño y arquitectura aplicados a un caso de estudio

**Autor: José Manuel Arévalo Navarro
Tutor: Marcos López Sanz**

*Para Almudena,
mi mujer y mi mejor amiga.*

*Gracias por ayudarme a
levantarme en los fracasos
y poner mis pies en la tierra en los éxitos*

<u>ÍNDICE DE ILUSTRACIONES.....</u>	<u>4</u>
<u>LISTA DE TABLAS.....</u>	<u>4</u>
<u>RESUMEN.....</u>	<u>6</u>

1 MOTIVACIÓN.....	7
1.1 PRESENTACIÓN DEL PROBLEMA.....	8
1.2 OBJETIVOS.....	9
1.3 METODOLOGÍA.....	10
1.4 ESTRUCTURA DE LA MEMORIA.....	11
2. CONCEPTOS PREVIOS	13
2.1 INDEPENDENT SERVICE VENDOR (ISV).....	14
2.2 SERVICE ORIENTED ARCHITECTURE (SOA).....	14
2.3 SERVICE LEVEL AGREEMENT (SLA).....	18
3. PRINCIPIOS Y ARQUITECTURA CLOUD.....	21
3.1 ARQUITECTURA DE REFERENCIA EN CLOUD COMPUTING.....	22
3.1.1 Infraestructura como servicio (IaaS).....	23
3.1.2 Plataforma como servicio (PaaS).....	24
3.1.3 Virtualización	26
3.1.4 Software como servicio (SaaS).....	27
3.2 TIPOS DE CLOUD.....	31
3.2.1 Cloud pública.....	32
3.2.2 Cloud privada.....	32
3.2.3 Cloud híbrida.....	33
3.3 IMPLEMENTACIONES.....	34
3.2.1 Amazon Web Services (AWS).....	34
3.2.1.1 Amazon Elastic Compute Cloud (EC2).....	35
3.2.1.2 Amazon Simple Storage Service (S3).....	37
3.2.2 Microsoft Windows Azure.....	38
3.4 CLOUD COMPUTING Y SOA.....	40
4.1 ASPECTOS CLAVES EN LA ELECCIÓN DE IaaS FRENTE A SISTEMAS TRADICIONALES.....	44
4.2 ASPECTOS CLAVES EN LA ELECCIÓN DE PaaS FRENTE A SISTEMAS TRADICIONALES.....	46
4.3 ASPECTOS CLAVES EN LA ELECCIÓN DE SaaS FRENTE A SISTEMAS TRADICIONALES.....	49
4.4 OTROS XaaS: EL CASO DEL VIRTUAL DESKTOP INFRASTRUCTURE (VDI).....	51
4.5 RIESGOS EN CLOUD COMPUTING.....	54
5 INTEGRACIÓN DE UN CASO DE ESTUDIO EN LA NUBE.....	56
5.1 SISTEMA DE INFORMACIÓN INICIAL: GESIMED.....	57
5.2 POSICIONAMIENTO DE GESIMED EN LA NUBE.....	58
5.3 POSICIONAR GESIMED EN UN MODELO SaaS.....	59
5.4 INTEGRACIÓN SOBRE IaaS.....	65
5.5 ARQUITECTURA GLOBAL DEL SISTEMA E IMPLICACIONES DE LA INTEGRACIÓN.....	71
6 CONCLUSIONES Y TRABAJOS FUTUROS.....	75
6.1 CONCLUSIONES.....	76
6.2 TRABAJOS FUTUROS.....	77
REFERENCIAS.....	78

Índice de Ilustraciones

ILUSTRACIÓN 1 MODELO DE METODOLOGÍA TOP-DOWN.....	11
ILUSTRACIÓN 2 SERVICE ORIENTED ARCHITECTURE [12].....	18
ILUSTRACIÓN 3 ACUERDO DE NIVEL DE SERVICIO.....	19
ILUSTRACIÓN 4 MODELOS EN CLOUD COMPUTING.....	23
ILUSTRACIÓN 5 ACCESO GLOBAL EN CLOUD COMPUTING [3].....	28
ILUSTRACIÓN 6 MODELOS DEFINIDOS POR FUNCIONALIDAD [3].....	30
ILUSTRACIÓN 7 MODELOS DE MADUREZ EN SAAS [5].....	31
ILUSTRACIÓN 8 ESQUEMA CLOUD COMPUTING.....	34
ILUSTRACIÓN 9 LOGOTIPO AMAZON WEB SERVICES [9].....	34
ILUSTRACIÓN 10 PLATAFORMA WINDOWS AZURE [14].....	39
ILUSTRACIÓN 11 SERVICIOS EN WINDOWS AZURE [17].....	40
ILUSTRACIÓN 12 SINERGIAS ENTRE SOA Y CLOUD COMPUTING.....	42
ILUSTRACIÓN 13 ARQUITECTURA DE GESIMED [12].....	57
ILUSTRACIÓN 14 MODELO DE NEGOCIO EN FORMULARIO WEB.....	63
ILUSTRACIÓN 15 MASHUP GESIMEDSAAS EN MARKETPLACE.....	64
ILUSTRACIÓN 16 CICLO DE VIDA AMAZON MACHINE IMAGE [9].....	66
ILUSTRACIÓN 17 ARQUITECTURA INTEGRACIÓN IAAS.....	70
ILUSTRACIÓN 18 ARQUITECTURA FINAL INTEGRACIÓN MARKETPLACE.....	73

LISTA DE TABLAS

TABLA 1 EJEMPLOS CAÍDAS DE SERVICIO.....	55
TABLA 2 POSICIONAMIENTO DE GESIMEDSAAS EN LA NUBE.....	58
TABLA 3 PRECIOS GESIMEDSAAS.....	62
TABLA 4 PROCESO DE ADOPCIÓN SAAS.....	65
TABLA 5 INSTANCIAS ESTÁNDAR EN AMAZON WEB SERVICES [9].....	67
TABLA 6 PRECIOS ALMACENAMIENTO EN S3 [9].....	68
TABLA 7 COSTE DEL CASO DE ESTUDIO.....	71
TABLA 8 ANÁLISIS DE COSTES CLOUD COMPUTING FRENTE SOLUCIÓN IN-HOUSE.....	73

Resumen

Este proyecto fin de master plantea la integración y el despliegue de un sistema de información en una infraestructura y plataforma Cloud Computing. Se ha realizado un estudio de este nuevo modelo de servicios, estudiando sus aspectos teóricos y fundamentales, analizando las implementaciones con las que a día de hoy podemos encontrar en el mercado para poder afrontar nuestros retos profesionales. Debemos tener en cuenta los riesgos que deberemos asumir en la adopción del modelo, aportando soluciones para mitigarlos de la forma más eficiente posible. Pero también se analizarán todas las ventajas y se tendrán en cuenta cuando se desarrolle el caso de estudio.

El alcance del proyecto es realizar un detallado análisis sobre el paradigma Cloud Computing mediante un profundo estudio, y una posterior validación mediante la implementación de un caso de estudio detalladamente definido. Se detallarán y justificarán todas las decisiones tomadas, así como los costes ocasionados de la integración frente a los costes que supondría una solución interna. Se detallarán los beneficios introducidos y los obstáculos encontrados a lo largo del caso de estudio. Se analizará que impacto ha tenido partir de un sistema de información orientado a servicios.

Con la realización del proyecto se ha demostrado que se puede contar con nuevas alternativas tecnológicas y nuevos modelos con los que poder ofrecer servicios a nuestros clientes llevándonos a formular preguntas tales como:

¿Qué es y que no es Cloud Computing?

¿Cloud Computing es realmente lo que necesito?

¿Qué necesito para adoptar Cloud Computing?

¿Qué riesgos tiene Cloud Computing?

¿Qué impacto tiene SOA en el Cloud Computing?

Capítulo 1:

Introducción

1.1 Presentación del problema

La realidad profesional en nuestro día a día en el ámbito de las tecnologías de la información actualmente a la hora de afrontar un proyecto, bien sea en organizaciones o en el área de investigación, está basada en un modelo en el cuál un cliente (interno o externo) pide soluciones al departamento TI, (“me hace falta determinado entorno para poner una aplicación en producción”, “necesito determinado entorno porque vamos a abrir una oficina en una determinada geografía”, “el negocio se va a transformar y necesito una plataforma que lo soporte”). Es un escenario en el cuál hay que poner la solución encima de la mesa, generalmente a medida, con orientación al cliente, la cual finalmente se desarrolla después del plazo estimado (en el mejor de los casos).

Cuando tenemos a profesionales (cliente externo o interno) pidiendo una solución a otros profesionales (departamento TI) lo que se genera es una espera, ya que implica una fase de captura de requisitos, análisis y diseño de la solución, por supuesto ese tiempo genera unos costes solo por el simple hecho de que estamos empleando unos recursos tanto humanos como tecnológicos (trabajadores, servidores, módems...etc.) trabajando en unas tareas en el intervalo que duran esas fases. La pregunta que cabe hacerse es: ¿qué alternativas a este modelo tengo?

Una de las alternativas que en los últimos años ha florecido al amparo de la orientación a servicios como paradigma tanto a nivel de negocio como a nivel tecnológico es el paradigma de Cloud Computing. Este paradigma propugna ser capaz de aprovisionarse con recursos TI, de manera directa, instantánea en el tiempo (en tiempo real) y con unos costes en la gestión que sean casi planos. Este proyecto se centra en la realización de un estudio del Cloud Computing como alternativa viable, objetiva y real a los actuales modelos de servicios. *Viable*, porque Cloud Computing está a nuestro alcance; *objetiva*, porque enfrentaremos Cloud Computing a otros modelos desde el punto de vista del negocio y económico; y finalmente, *real*, porque se experimentará sobre un caso de estudio concreto. No se pretende justificar la mejor solución o el mejor modelo sino ofrecer y adoptar el modelo Cloud Computing bajo las premisas que lo hacen realmente útil para poder hacer más sencillo nuestro día a día de negocio, es decir, analizar cuáles son nuestras necesidades de negocio y en base a lo que nos ofrece Cloud Computing decidir si es la mejor solución y el mejor modelo para nuestro proyecto.

1.2 Objetivos

El objetivo principal de este proyecto fin de master es *estudiar en profundidad el paradigma de Cloud Computing mediante la aplicación a un caso de estudio real*, tomando como partida un sistema de información basado en servicios.

La idea subyacente a la consecución de este objetivo es poder demostrar cómo la adopción de Cloud Computing como modelo de servicios representa una alternativa válida bajo ciertas circunstancias. Para conseguir el objetivo anterior se han definido los siguientes objetivos parciales:

- a) **Realizar un estudio detallado del paradigma de Cloud Computing.** Para ello se realizarán las siguientes subtareas:
 - a. Es necesario un análisis de los principios del paradigma de Cloud Computing poniendo un énfasis especial en los aspectos arquitectónicos y de negocio. El resultado de esta tarea será la especificación de lo que se conoce como arquitectura de referencia de Cloud Computing.
 - b. Seguidamente será necesario estudiar las consecuencias de adoptar este paradigma, sus beneficios y riesgos asociados.
- b) **Aplicar los resultados obtenidos del estudio anterior a un caso de estudio particular..** En este caso será necesario completar las siguientes etapas:
 - a. Análisis del caso de estudio: Gesimed. Para poder aplicar un nuevo modelo como es el de Cloud Computing es primordial conocer los detalles del caso de estudio sobre el que se aplicarán sus principios.
 - b. Posicionar Gesimed en un modelo SaaS: Con la intención de expandir las oportunidades de Gesimed es conveniente posicionarlo en un modelo de negocio SaaS. Se implementará una integración real con una tienda de aplicaciones (marketplace), mediante un previo diseño de la arquitectura.
 - c. Integrar Gesimed en una Infraestructura como servicio: Se realizará una integración real de un sistema de información en una IaaS, diseñando la arquitectura como resultado de la integración y se detallarán los costes de uso de

la infraestructura comparándolos frente al coste de una infraestructura tradicional.

1.3 Metodología

Se ha considerado definir una metodología de trabajo. Puesto que ya conocemos el dominio del negocio (proyecto fin de carrera), nos hemos decantado por una aproximación algo más clásica pero con una adaptación particular a nuestras necesidades.

En realidad más que metodología, lo que hemos definido es una *estrategia*. El proceso de desarrollo de proyectos está demasiado encorsetado a una estrategia “**bottom-up**”, tanto desde el punto de vista del desarrollo de los proyectos tecnológicos como desde el punto de vista de la inversión, esta aproximación aunque es válida, correcta, necesaria, consiste en evolucionar y mejorar lo que tengo y lo que soy, es decir, *tengo una realidad tecnológica y lo que quiero es evolucionar, adoptar mejores prácticas*, el problema de todo eso es el tiempo que lleva, ya que obtener un sistema maduro, gobernado y gestionado, implica tiempo, esfuerzo, dinero y, en ocasiones un servicio no puede esperar tanto, ya que conseguir lo mencionado anteriormente es un largo camino a recorrer. Por ello, se cree conveniente analizar para un nuevo servicio, que modelo es el que mayor valor le va aportar a ese proceso de negocio, a esa realidad, a esa carga de trabajo.

Dicho de otro modo, al mismo tiempo que evolucionamos y mejoramos lo que somos, analizar cuáles son mis retos, mis realidades de negocio y ver en cada carga de trabajo cuál es el modelo de servicio es el que mejor aplica. *¿Para esta carga de trabajo aplica Cloud Computing?, entonces vamos a ver con que arquitectura y con qué infraestructura*, pero ya tengo muy claro que Cloud computing aplica, y que sus características están alineadas con nuestro modelo de negocio, que el automatismo que quiero implementar es viable, que voy a obtener un retorno de la inversión, y a partir de ese punto ya hablaremos de tecnología.

Estas conclusiones nos han llevado a adoptar una estrategia basada en Bottom-Up y Top-Down combinadas, es decir hemos evolucionado y mejorado nuestro sistema de partida, pero a la vez hemos analizado que modelo es el que más valor le aporta y lo hemos atacado directamente.



Ilustración 1 Modelo de metodología top-down

Esta metodología está muy orientada al caso práctico de este proyecto. Cada uno de estos hitos se ha llevado a cabo, aunque no ello no sería posible si no hiciéramos un estudio previo y en profundidad del paradigma de Cloud Computing.

En primer lugar hemos detectado las necesidades de nuestro sistema y hemos definido un posicionamiento claro de lo que queremos obtener de Cloud Computing. Posteriormente hemos definido las cargas de trabajo que van tener lugar en este proyecto, en este caso la carga de trabajo es Gesimed. A continuación se han seleccionado los modelos en base a nuestro posicionamiento y hemos estudiado que proveedores nos ofrecen soluciones a nuestras necesidades. Una vez que sabemos el punto de partida y a donde queremos llegar, detectamos los gaps. Posteriormente se ha intentado hacer un cálculo de costes y retorno de la inversión de nuestro nuevo sistema. Por último se ha establecido una hoja de ruta, que en nuestro caso de estudio ha significado el día a día de trabajo en este proyecto.

1.4 Estructura de la memoria

Este Proyecto Fin de Master se estructura en 5 capítulos, cuyo contenido es el siguiente:

- El **presente capítulo** presenta el problema y la motivación que se va a abordar con la realización de este Proyecto Fin de Master, los objetivos que se esperan cumplir y la estrategia y metodología de trabajo seguida.

- El **segundo capítulo** explica los conceptos previos necesarios para la correcta comprensión de las tecnologías utilizadas, tanto los estándares empleados, como las nuevas tecnologías introducidas.

- En el **tercer capítulo** se describen los fundamentos en los que se basa el modelo Cloud Computing, tanto desde el punto de vista teórico como desde el punto de vista tecnológico. Se detalla la arquitectura de referencia y se realiza un análisis de dos de los proveedores de Cloud más importantes en el mercado actualmente, por último en este capítulo se estudia la relación entre SOA y Cloud Computing.

- En el **cuarto capítulo** se explican los aspectos fundamentales para saber tomar decisiones a la hora de elegir este nuevo modelo de servicios, los beneficios frente a sistemas tradicionales y como abordar los riesgos añadidos.

- En el **quinto capítulo** se aplican los conceptos vistos anteriormente a un caso de estudio concreto y real, se justifican los beneficios introducidos en nuestro nuevo sistema, los obstáculos encontrados, así como la arquitectura resultante de haber adoptado Cloud Computing.

- En el **sexto capítulo** se analizan las conclusiones obtenidas en el proyecto y los trabajos futuros propuestos.

- Por último se incluye una **bibliografía** donde se enumeran las fuentes de información empleadas para el desarrollo y documentación del presente proyecto.

Capítulo 2:

Conceptos previos

Para una completa comprensión del trabajo realizado en este proyecto de fin de master es necesario conocer previamente algunas de las características de las tecnologías y aproximaciones utilizadas en el proyecto.

2.1 Independent Service Vendor (ISV)

Un ISV es un término empleado para hacer referencia a empresas especializadas en comercializar con software de distintos segmentos, es decir, hay ISV's para software sanitario, seguridad, ofimática etc. Normalmente un ISV fabrica y distribuye software que funciona en varios sistemas operativos, aportando así mayor valor a su cliente [7]. El motivo de introducir este concepto en este capítulo, es que, anteriormente a estas entidades sencillamente las veíamos en forma de comercial intentando vendernos un producto, es decir una caja con un número determinado de licencias a un coste determinado.

Estas entidades están planteándose actualmente una nueva estrategia de negocio y son pieza clave en el modelo de negocio Software as a Service (SaaS) y aunque vigilan que este modelo les proporcione un verdadero retorno de la inversión (ROI), parece ser la tendencia actual, los ISV's conformarán y aportarán valor a los ecosistemas SaaS, algunas de las características que buscan están alineadas con SaaS.

Los **proveedores de software independientes** (ISV) pueden crear e implantar rápidamente aplicaciones SaaS en alguna plataforma (libre o propietaria), que sea capaz de ofrecer una única plataforma integrada tanto para implantaciones in situ como basadas en la "nube".[7]

La idea es que los ISV también pueden **ofrecer licencias para SaaS de manera mensual aprovechando el modelo de licencias**. El modelo de licencia permite a los ISV escalar sus inversiones en software conforme crezca su negocio.

2.2 Service Oriented Architecture (SOA)

Arquitectura Orientada a Servicios (SOA), es un marco conceptual para integrar procesos

de negocios soportados en tecnología segura a través de componentes desarrollados bajo estándares internacionales que pueden ser re-utilizados y combinados para adaptarse a los cambios de prioridad del negocio. ***SOA es una arquitectura desacoplada de componentes de software que proveen funciones específicas (proveedor) y que pueden ser invocadas por otros componentes (consumidor) independientemente de la plataforma en que se encuentren ambos.*** [según ref. 10]

- Los beneficios que puede obtener una organización que adopte SOA son:
- Facilidad para evolucionar a modelos de negocios basados en tercerización.
- Facilidad para abordar modelos de negocios basados en colaboración con otros entes (socios, proveedores).
- Poder para reemplazar elementos de la capa applicativa SOA sin disrupción en el proceso de negocio.
- Facilidad para la integración de tecnologías disímiles.
- En SOA los proyectos son transversales abarcan distintas áreas, y distintas aplicaciones, siguiendo el flujo transversal de los procesos de negocio de las compañías.

En SOA las soluciones se comparten, y se construyen para que los próximos proyectos se vean beneficiados aplicando una serie de principios que se describen a continuación.

a) LOS SERVICIOS SON REUSABLES

SOA anima a que los servicios sean reusables, a pesar de que la reusabilidad sea propiamente un requisito exigible, aplicando estándares de diseño como los ya comentados (XML, XSD, WSDL etc.) convierten a los servicios en potencialmente reusables y la oportunidad de adaptarse cómodamente a los cambios futuros o cambios en los requisitos con menos esfuerzo. Pero se entiende reusabilidad también en el sentido “entre-aplicaciones” es decir entre entornos heterogéneos, por ello podemos decir que va más allá del concepto de reusabilidad que dicta la orientación a objetos. Los mensajes pueden incluir meta-datos en la cabecera, para ayudar a los servicios a ser más genéricos y por lo tanto más reusables.

b) LOS SERVICIOS COMPARTEN UN CONTRATO FORMAL

El contrato de un servicio consiste en:

- Endpoint del servicio
- Operaciones del servicio
- Entrada y salida de las operaciones de cada servicios
- Reglas características de cada servicios y sus operaciones
- Estos contratos se definen mediante estándares tales como, XSD y WSDL.

c) LOS SERVICIOS TIENEN BAJO ACOPLAMIENTO

Nadie puede predecir como un entorno IT puede evolucionar, como las aplicaciones crecerán, se integrarán o serán reemplazadas, los servicios son capaces de soportar estos cambios debido a que el estar desacoplados es una condición para que un servicio tenga conocimiento de otro servicio, aunque la lógica subyacente, esto se alcanza a través del uso de contratos de servicio que permiten interactuar bajo unos cierto parámetros [10].

d). LOS SERVICIOS ABSTRAEN LA LÓGICA SUBYACENTE

Este principio es el que permite a los servicios a actuar como cajas negras, ocultando los detalles al resto del mundo. El alcance de la lógica representada por un servicio está influenciado significativamente por el diseño de sus operaciones y su posición en un proceso concreto. No hay límite de cuanta lógica puede representar un servicio, este mismo podría estar diseño para realizar una tarea simple u una tarea más compleja o puede ser la puerta de entrada para una solución automática entera. La granularidad de las operaciones está íntimamente relacionada con la naturaleza de la funcionalidad que expone el servicio [10].

e). LOS SERVICIOS SE PUEDEN COMPONER

Un servicio puede representar cualquier rango de lógica de cualquier tipo de fuente, incluyendo otros servicios, la principal razón para implementar este principio es para asegurar que los servicios son diseños para que participen como miembros efectivos en la

composición de otro servicio (si ello fuera requerido). Una extensión de este concepto en SOA, es el concepto de Orquestación, cuando un servicio es controlado por otro servicio padre que compone a los participantes del proceso, aunque sobre este concepto hablaremos más adelante [12].

f). LOS SERVICIOS SON AUTÓNOMOS

La autonomía requiere que el rango de lógica expuesta por un servicio exista dentro de una frontera explícita, esto permite al servicio auto-gobernarse, ya que elimina dependencias con otros servicios, esto no quiere decir que el servicio tenga total permiso y exclusividad sobre la lógica que encapsula, si no que está garantizado que en tiempo de ejecución, el servicio toma el control de la lógica que encapsula [12].

Existen dos niveles de autonomía en los servicios:

Autonomía a nivel de servicio. La frontera del servicio es distinta para cada uno, pero el servicio podría compartir la lógica que subyace y abstrae, por ejemplo, un servicio que encapsula el sistema legado como pueda ser los mainframe de los bancos, o también llamados servicios wrapper, puede que tenga que compartir la lógica con otros clientes del sistema que encapsula.

Autonomía Pura. La lógica encapsulada es de total control del servicio, suele ser habitual cuando esta lógica es creada de cero para dar soporte a las operaciones del servicio.

g). LOS SERVICIOS NO TIENEN ESTADO

Los servicios deben minimizar la cantidad de información que ellos gestionan y la duración que la mantienen, este principio asegura la reusabilidad ya que los servicios serán más generales y promueve la escalabilidad, cuanto mayor estado se mantenga, mayor es el procesamiento que se debe realizar tal y como muestra la ilustración 2.

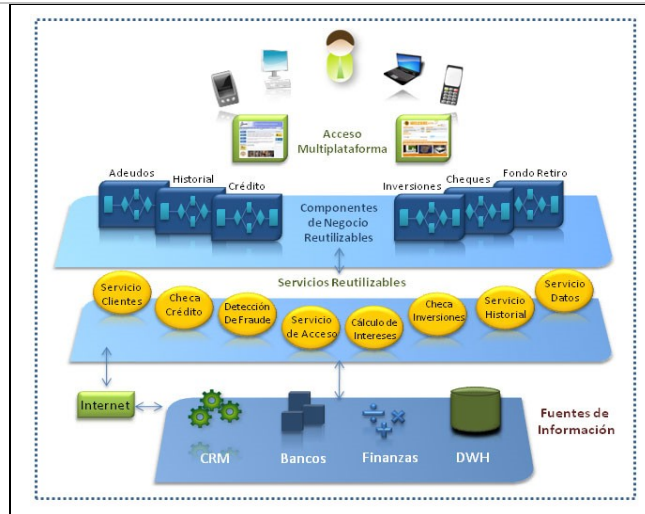


Ilustración 2 Service Oriented Architecture [12]

2.3 Service Level Agreement (SLA)

Un *acuerdo de nivel de servicio* o *Service Level Agreement*, también conocido por las siglas ANS o SLA, es un contrato escrito entre un proveedor de servicio y su cliente con objeto de fijar el nivel acordado para la calidad de dicho servicio [1]. El SLA es una herramienta que ayuda a ambas partes a llegar a un consenso en términos del nivel de calidad del servicio, en aspectos tales como tiempo de respuesta, disponibilidad horaria, documentación disponible, personal asignado al servicio, etc. ***Básicamente el SLA define la relación entre ambas partes: proveedor y cliente.*** Un SLA identifica y define las necesidades del cliente a la vez que controla sus expectativas de servicio en relación a la capacidad del proveedor, proporciona un marco de entendimiento, simplifica asuntos complicados, reduce las áreas de conflicto y favorece el diálogo ante la disputa.

También constituye un punto de referencia para el mejoramiento continuo, ya que el poder medir adecuadamente los niveles de servicio es el primer paso para mejorarlos y de esa forma aumentar los índices de calidad.

Un SLA se negocia entre dos partes donde una de ellas es el cliente y la otra un proveedor de servicios. Estos acuerdos pueden estar vinculados legalmente, o ser un contrato informal (relaciones inter-departamentales). Los contratos entre los proveedores de servicios y una tercera parte son habitualmente y de forma incorrecta, llamadas también SLA, aunque el nivel de servicio ya ha sido definido por el cliente inicial y por lo tanto el acuerdo entre terceras partes no es más que un contrato [1].

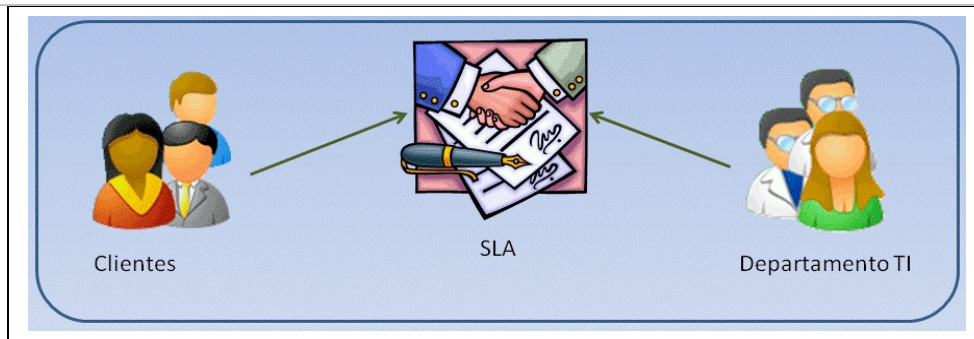


Ilustración 3 Acuerdo de nivel de servicio.

Los ANS definen un punto de entendimiento común sobre servicios, prioridades, responsabilidades y garantías. Cada área de servicio debe tener un SLA definido, que comprenda los niveles de disponibilidad, servicio, rendimiento u otros atributos del servicio, como la facturación. El nivel del servicio también puede ser especificado como *objetivo* y *mínimo*, de forma que los usuarios puedan saber que esperar (mínimo), mientras se ofrece un objetivo que muestra el nivel de rendimiento. En algunos contratos pueden figurar penalizaciones en caso de incumplimiento de los SLA. Es importante remarcar que los acuerdos hacen referencia a los servicios que recibe el usuario, pero no como el proveedor ofrece ese servicio.

Los SLA se han utilizado desde finales de los años 80 por parte de operadores de telecomunicaciones como parte de sus contratos con clientes empresariales. Esta práctica se ha extendido hasta tal punto que actualmente es habitual que un usuario firme un contrato con un proveedor de servicios que incluya una serie de SLA para prácticamente todos los mercados.

Los departamentos de grandes corporaciones han adoptado también el sistema de acuerdos de nivel de servicio respecto a los clientes internos, departamentos de la misma organización ya que mediante este sistema se logra mejorar la calidad del servicio.

Los SLA están, por su naturaleza, basados en los resultados del servicio recibido por el usuario como elemento del acuerdo. Las organizaciones también pueden definir y especificar el sistema por el que el servicio debe ser cumplido mediante una especificación (especificación del nivel de servicio). Este tipo de acuerdo recibe el nombre de *input SLA*, aunque este tipo de acuerdo ha quedado obsoleto ya que las organizaciones permiten a los proveedores seleccionar el método de cumplimiento de los acuerdos.

Los acuerdos de nivel de servicio pueden contener un alto número de parámetros con sus correspondientes objetivos de nivel de servicio. Un caso habitual es un *service desk*. Los parámetros designados habitualmente para estos casos incluyen [2]:

- **ABA** (*Abandonment Rate Agreement* o ratio de abandono): Porcentaje de llamadas abandonadas mientras esperaban recibir atención telefónica.
- **ASA** (*Average Speed to Answer* o tiempo medio de atención): Tiempo medio normalmente medido en segundos, utilizado para que el *service desk* responda la llamada.
- **TSF** (*Time Service Factor* o factor del tiempo de servicio): Porcentaje de llamadas respondidas en un plazo de tiempo determinado, por ejemplo 80% en 20 segundos.
- **FCR** (*First Call Resolution* o resolución en la primera llamada): Porcentaje de llamadas recibidas que pudieron ser resueltas sin necesidad de una segunda llamada.
- **TAT** (*Turn Around Time* o tiempo de respuesta): Tiempo utilizado para completar una tarea determinada.

Los acuerdos de disponibilidad son otro tipo de parámetro muy habitual utilizado en los servicios como servidores dedicados. Algunos acuerdos habituales incluyen un porcentaje, tiempo de operación de la red, tiempos de mantenimiento, etc.

Capítulo 3:

Principios y arquitectura Cloud

3.1 Arquitectura de referencia en Cloud Computing

Cloud Computing es un nuevo modelo de prestación de servicios, no es una nueva tecnología per se, este nuevo modelo está claramente orientado a la ***escalabilidad***, es decir, poder atender una demanda muy fuerte en la prestación de un servicio, ***pero de manera muy directa, inmediata en el tiempo, con un impacto en la gestión y en el coste que es casi plano, esta orientación a la escalabilidad lo que provocará es que el usuario final perciba que todo funciona, todo va rápido, todo es fácil y por lo tanto su experiencia como usuario es mucho más gratificante.***[3] A pesar de que no es una nueva tecnología, es conveniente explicar los fundamentos tecnológicos que los proveedores de Cloud están tomando comúnmente. Como principios tecnológicos es necesaria una fuerte ***capa de virtualización*** de infraestructura (servidores, almacenamiento, comunicaciones etc.). Una ***capacidad muy avanzada en cuanto a aprovisionamiento*** de recursos IT, ***orquestración de esos recursos*** y una ***orientación a servicios***, diría que SOA es el alma de Cloud Computing y nos permitirá dar esa escalabilidad tan agresiva, por ello se implementará también una ***elasticidad***, tanto en el modelo como en la infraestructura. Por último es muy importante destacar la necesidad de una ***estandarización de los servicios***, cuando más estandarizada sea nuestra infraestructura, más sencillo será todo [5].

Como podéis ver, no hay nada nuevo, la única novedad es el nivel de exigencia que le pediremos a ese entorno de computación.

Este capítulo describe algunos de los diferentes modelos de Cloud Computing categorizados como un conjunto de modelos de servicios. Otro modo de verlo sería mediante capas sobre las cuales podrían desplegarse y construirse aplicaciones distribuidas. Estas capas, principalmente son, infraestructura, plataforma y software, con una gran capa de virtualización y protocolos de comunicación.

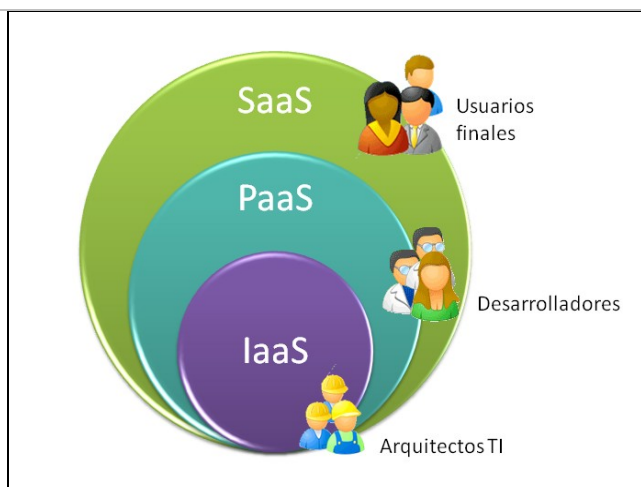


Ilustración 4 Modelos en Cloud Computing

3.1.1 Infraestructura como servicio (IaaS)

Infraestructura como servicio podría definirse como un modelo de servicios de computación, estos servicios se podrían utilizar para resolver nuestras necesidades computacionales sin límites de escalabilidad de nuestros despliegues. Solo pagaríamos por lo que usamos y solo cuando lo necesitemos.

IaaS es un modelo de servicio en el cuál el hardware está virtualizado en la nube [5]. En este particular modelo, el proveedor del servicio provisiona servidores, almacenamiento, redes, y así sucesivamente. Esta manera de ver una infraestructura profesional rompe con todos los moldes, ya que podríamos tener desde un pequeño negocio a una gran empresa en un corto plazo. La adopción de este tipo de servicios está siendo empujada actualmente gracias a una multitud de *startups* que han comenzado a emprender en estos tiempos de crisis y que no se pueden permitir tener su propio *datacenter* o una infraestructura *InHouse*.

Los desarrolladores en este modelo encuentran una manera dinámica y flexible de trabajar, ya que interactúan con la IaaS a través de servidores virtuales, almacenamiento virtual, generalmente se generan instancias de estas máquinas virtuales a golpe de ratón desde un portal, normalmente web, en ese primer momento lo que obtienen es un “*green field*” dispuesto a ser personalizado para que la solución sea completada, el acceso y ***la interacción de la aplicación***

con la IaaS suele ser a través de SOA y contratos de servicios, puede ser mediante WSDL o mediante servicios REST. Un ejemplo de una infraestructura como servicio es Amazon Web Services, cuyos servicios incluyen, almacenamiento, infraestructura de redes, máquinas virtuales, y tienen estos servicios replicados a lo largo del mundo (Irlanda, Singapur, Este y Oeste de EEUU) aunque se entrará en detalle sobre este proveedor más adelante.

IaaS está dirigido a cualquier empresa que desee delegar la implantación de sus sistemas software y aplicaciones en la infraestructura hardware de un proveedor externo (fenómeno conocido tradicionalmente como *hosting*) o que requiera de servicios de almacenamiento externo, copias de seguridad de sus datos, cálculos complejos que requieran software de elevadas prestaciones, etc. El proveedor les permitirá gestionar dichos sistemas en un entorno virtualizado [5].

Así, los *proveedores* de servicios son los *propietarios de las máquinas físicas*, y las ofrecerán como servicio a los usuarios a través de entornos que les permitan gestionarlas, por ejemplo una página Web para el control de las máquinas.

3.1.2 Plataforma como servicio (PaaS)

Una plataforma como servicio (PaaS) es un modelo de servicio que se sitúa por encima de IaaS en cuanto a nivel de abstracción de los recursos IT. *Este modelo propone un entorno software en el cuál un desarrollador puede crear y customizar soluciones dentro de un contexto de herramientas de desarrollo que la plataforma proporciona [3].* La plataforma puede estar basada en un lenguaje específico, varios o frameworks de desarrollo.

En un modelo PaaS los clientes pueden interactuar con el software para introducir o recuperar datos, realizar acciones etc., *pero no tienen responsabilidad de mantener el hardware*, el software o el desarrollo de las aplicaciones, solo se tiene responsabilidad de la interacción con la plataforma [4]. Dicho de otro modo, el proveedor es el responsable de todos los aspectos operacionales. A menudo la plataforma ofrece herramientas de desarrollo y despliegue de aplicaciones como por ejemplos Windows Azure y su integración a través de Visual Studio, es solo un ejemplo, la idea es que se puedan soportar estándares de desarrollo tales como, HTML, CSS, XML, JavaScript etc.

Las *plataformas como servicio* vienen a suponer que el desarrollador de aplicaciones web se olvida de almacenaje de ficheros, de gestión de la base de datos, de balanceo entre máquinas, de ancho de banda, de escalabilidad, de picos de demanda, de estabilidad, de tocar una máquina servidor... en definitiva, la plataforma sobre la que construyes tu aplicación web ya no es cosa tuya, es del servicio que contratas y que pagas religiosamente.

Concentrarte en tu aplicación y ahorrar costes, son las dos ventajas inmediatas de las plataformas como servicio. No sólo porque vendan almacenamiento y ancho de banda más barato que un pequeño proveedor, sino porque también adquirir el conocimiento para montar arquitecturas que escalen cuesta mucho dinero (o muchos años de esfuerzo, aunque *el rol de administrador de sistemas va a cambiar mucho si se impone como tendencia*). Por supuesto tienen sus peros, como comentamos en "[tu aplicación sobre web services](#)": *depender de un único proveedor* (algo con lo que tener mucho cuidado, diseñando la aplicación para tener poco acoplamiento y probándola también siempre en un servidor de toda la vida) y comerte también [sus caídas](#), aunque se antoja improbable que por uno mismo se consiga la disponibilidad de Amazon o Google.

El movimiento de estos dos gigantes de la web (en el mercado de las aplicaciones como servicio también está Salesforce con Force.com o Joyent) es una lucha por el rol por el que siempre ha apostado Microsoft con todas sus fuerzas: *ser la plataforma sobre la que otros construyen sus aplicaciones*, tanto en el escritorio como en el lado del servidor.

Por ello, para saber si realmente estamos ante una auténtica plataforma como servicio se deben dar las siguientes características:

- **Un entorno de desarrollo basado en un navegador** - si tienes que instalar algo en tu computadora para desarrollar aplicaciones, entonces no es PaaS.
- **Despliegue transparente hacia el entorno de ejecución** - idealmente, el desarrollador debería poder desplegar su aplicación PaaS con un solo click. Si hay que hablar con alguna persona para instalar la aplicación, entonces no es PaaS.
- **Herramientas de monitoreo y gestión** - aunque las soluciones basadas en nubes son muy convenientes en cuanto a costos, puede resultar complicado gestionarlas y

escalarlas sin buenas herramientas. Si hay que construir o agregar una herramienta de monitoreo propia para poder escalar la aplicación, entonces no es PaaS.

- **Facturación basada en el uso** - lo que hizo que PaaS fuera popular es que evita pagar por adelantado. Si no puedes pagar con la tarjeta de crédito basándote en el uso que haces de la plataforma, entonces no es PaaS.

3.1.3 Virtualización

Como se ha mencionado anteriormente, es muy importante disponer de una fuerte capa de virtualización en la infraestructura para ser capaces de responder a la demanda con una agresiva escalabilidad. La idea de la virtualización es poder crear servidores virtuales, almacenamiento virtual, redes virtuales y quizás algún día aplicaciones virtuales, es decir un pool de recursos. *Esta abstracción es clave en Cloud Computing ya que permite compartir y acceso ubicuo [4]*. Para crear sistemas de hardware virtual, en ocasiones se usan lo que se denomina como hipervisores,

Un *hipervisor* o *monitor de máquina virtual* (*virtual machine monitor*) es una plataforma que permite aplicar diversas técnicas de control de virtualización para utilizar, al mismo tiempo, diferentes sistemas operativos (sin modificar o modificados en el caso de paravirtualización) en una misma computadora. Es una extensión de un término anterior, “supervisor”, que se aplicaba a kernels de sistemas operativos [3].

La virtualización consiste en asignar un nombre lógico a un recurso físico, estos recursos se gestionan fácil y dinámicamente ya que se realiza un mapeo entre los recursos físicos y lógicos, estos mapeos pueden ser cambiados dinámicamente de una forma muy eficiente, siempre a través de las herramientas y productos que nos ofrecen los proveedores expertos como VMWARE. Algunas de las características de la virtualización son:

- **Acceso:** Un cliente puede acceder a un servicio Cloud desde cualquier geografía.
- **Aplicación:** Un Cloud tiene múltiples instancias de la aplicación y peticiones directas a una de las instancias basadas en condiciones.
- **CPU:** Se pueden crear particiones dentro de los ordenadores como un conjunto de máquinas virtuales que se ejecutan dentro de ese ordenador. Alternativamente se puede introducir un balanceador para repartir cargas.

- Almacenamiento: A menudo se realiza una replicación para conseguir redundancia.

3.1.4 Software como servicio (SaaS)

El modelo de servicio más completo es aquél que ofrece el software y el hardware como un servicio conjunto [3], es decir, SaaS provee la infraestructura, software, solución y toda la pila de aprovisionamiento como un servicio global.

Software as a Service (SaaS) se puede describir como software que está desplegado en un servicio de hosting y puede ser *accedido globalmente* a través de internet mediante navegador, móvil, tablet, etc. Y *donde todos los aspectos que no sean la propia interacción con la aplicación son transparentes al usuario*, En el modelo SaaS, los usuarios *pagan por el uso del servicio mediante cuotas de suscripción*, válidas por un determinado período de tiempo, como en el caso de un alquiler, las características fundamentales de este modelo se pueden resumir en:

- El software está disponible globalmente a través de internet y bajo demanda.
- El modelo de suscripción suele ser mediante licencias o basado en uso y es facturado por mensualidades de forma recurrente.
- Todo lo relativo a operaciones es responsabilidad del proveedor
- Las actualizaciones, mejoras, evoluciones o parches en el aplicativo, debe ser siempre transparente al usuario y por supuesto no debe hacer ningún tipo de configuración.
- SaaS soporta múltiples usuarios generalmente con un modelo multi-tenant.



Ilustración 5 Acceso global en Cloud Computing [3]

Este modelo de servicios normalmente pretende llegar a pequeñas y medianas empresas (PYMES) y a veces a usuarios individuales, dejando para las grandes empresas un modelo basado en VDI (Virtual Desktop Infrastructure) del que hablaremos más adelante. El acceso suele ser a través de un portal o de un CMS web, que puede ser en abierto o bien está sucinto a una suscripción previa de otro servicio en la compañía que ofrece SaaS (ADSL, línea móvil, .etc.). Dicho portal puede ser muy variado, pero generalmente contiene las aplicaciones que has comprado previamente con un acceso mediante mashups. Y por otro lado un catálogo de aplicaciones que se ofrecen, de ahí en adelante el portal puede ser tan avanzado como se quiera pero siempre será el punto de acceso y uso a los servicios.

Los proveedores que desarrollan portales para habilitar SaaS normalmente tienen *tres sistemas básicos* que desarrollar. Por un lado un potente sistema de *billing y facturación*, tanto para soportar el complejo modelo de suscripción por uso, como los modelos de promociones, generación de reportes y business intelligence. Por otro lado, el siguiente subsistema clave, es el *de autenticación, donde generalmente se implementa un sistema basado en single sign on* que se suele comunicar con los sistemas CRM de las compañías para las que se habilita el SaaS o bien con nuestro propio sistema de gestión de cliente si el portal SaaS es propiedad nuestra. Por último es necesario un buen *sistema de integración de ISV's*, el valor al portal SaaS se lo darán las aplicaciones que tengan, por ello cuanto más flexible y dinámica sea nuestra API de integración de aplicaciones, más rápido serán ofrecidas a los clientes y por lo tanto nuestro retorno de la inversión será mayor en menos tiempo.

No se puede acabar este capítulo sin antes explicar algunos aspectos fundamentales de cara al cliente que va a consumir los servicios SaaS. Desde el punto del cliente que va a adquirir los servicios de una aplicación ofrecida como servicio, existen una serie de requisitos mínimos necesarios que una SaaS debe ofrecer:

- **Rendimiento.-** Una SaaS debe ofrecer un rendimiento mínimo y aceptable para que sea atractiva su adquisición. El problema aquí es definir mínimo y aceptable y aunque es un concepto subjetivo puede ser medible en tiempos de respuesta en el acceso a los datos, de ejecución los procesos de negocio, de comunicación a la propia aplicación (delay producido por el alojamiento geográfico de esta), etc.
- **Acuerdo de Nivel de Servicio (en inglés Service Level Agreement) .-** El ISV de la aplicación SaaS debe proveerte de varios niveles de servicio al que los cliente pueda adherirse. Habrá clientes que necesiten su aplicación disponible 8×5 (5 días a la semana, 8 h) y habrá que clientes que necesiten 24 X 7. El ISV deberá instalar en sus sistemas los mecanismos necesarios para poder ofrecer este tipo de acuerdos, esto es, backup, [cluster de alta disponibilidad](#) de datos y aplicación, etc.
- **Privacidad en las comunicaciones.-** Debido a la importancia de los datos que puedan albergar las aplicaciones es necesario que la comunicación que se realiza a través de Internet sea segura, esto es, la comunicación debe realizarse a través de [https](#) u otra forma de comunicación que asegure la privacidad de las comunicaciones.
- **Privacidad de los datos.-** De igual forma el ISV debe asegurar que los datos estén seguros y accesibles única y exclusivamente por el dueño del dato. Esto debe ser especialmente perseguido en la aplicaciones [multitenant](#).(nivel 3 y 4 de maduración).
- **Monitorización de la aplicación.-** El cliente debe saber de alguna forma que es lo que ocurre en su aplicación, por ejemplo: quién accede, a qué procesos, a qué datos, etc. Esto es obligado cuando el pago por el uso de la aplicación se realiza a través de conceptos como horas de utilización de la aplicación, consumo de espacio de disco, o cualquier otra forma que sea variable.
- **Acceso de a los datos.-** El resto de la aplicaciones de la organización deben acceder a través de [APIs](#) o de [Web Services](#), a los datos y lógica de negocio que se utilizan y genera por el uso de la SaaS, sobretodo, en clientes que tengan adoptado la

Cuando nos enfrentamos a elegir qué aplicación SaaS queremos que cubra cierta funcionalidad para nuestra empresa o área funcional, debemos saber qué es lo que nos ofrece el proveedor desde diferentes puntos de vista. Por ejemplo, no es lo mismo que el software te lo ofrezca a un único cliente, o que lo comparta con otros clientes y tampoco es lo mismo que el recurso que comparta sea el código de la aplicación que la base de datos donde lo almacenamos.



Ilustración 6 Modelos definidos por funcionalidad [3]

Para decidir qué es lo que más nos interesa desde el punto de cliente, antes veremos qué es lo que actualmente nos podemos encontrar en el mercado tomando como referencia lo que Microsoft definió y llamó el modelo de madurez de SaaS [4]:

- **Nivel 1 de Madurez.** En este nivel, el cliente tiene su propia versión personalizada de la aplicación alojada. Corre su propia instancia de la aplicación en los servidores del proveedor.
- **Nivel 2 de Madurez.** En este nivel, hay un vendedor (propietario) y un cliente (arrendatario). El proveedor aloja una instancia independiente para cada aplicación de cada cliente. En este nivel todas las implementaciones son del mismo código y el proveedor conoce las necesidades de los clientes, proporcionando opciones detalladas de configuración que permiten al cliente cambiar la forma en la que la aplicación se ve y se comporta para sus usuarios. Los cambios realizados por el arrendatario pueden permitir la disponibilidad de diversas opciones de personalización para sus clientes.
- **Nivel 3 de Madurez.** En este nivel, el vendedor, en lugar de acoger una instancia independiente de la solicitud de cada cliente, alberga una sola instancia. Las políticas de

autorización y de seguridad garantizan que los datos de cada cliente se mantienen separados de la de otros clientes.

- **Nivel 4 de madurez.** En este nivel, el proveedor agrupa a varios clientes en una granja con equilibrio de carga para instancias idénticas, donde los datos de cada cliente que se mantienen separados. Los metadatos configurables proporcionan una experiencia de usuario única y un conjunto de características para cada cliente.

Como se puede ver a medida que aumenta el nivel de madurez se obtiene un mayor aprovechamiento de las economías de escala provenientes de la reducción en cada nivel de los recursos necesarios que componen la solución y por tanto del menor mantenimiento.

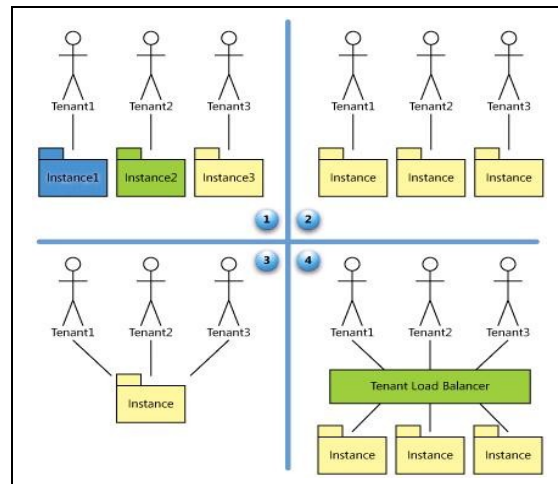


Ilustración 7 Modelos de madurez en SaaS [5]

3.2 Tipos de Cloud

Una vez que se ha valorado que Cloud Computing es un modelo de negocio con unas características que dotarán de mayor valor a nuestros servicios y que obtendremos un retorno de la inversión, hay que estudiar qué tipo de Cloud vamos a adoptar, los tres tipos de Clouds se van a definir en base a quién va a poder acceder a los servicios y quién va a gestionar la infraestructura. Los tipos son: pública, privada e híbrida; también se mencionará como la tendencia actual propone federar las nubes, comunicarlas entre sí y conseguir potenciar más la

3.2.1 Cloud pública

En las nubes públicas, los servicios que se ofrecen se encuentran en servidores externos al usuario, pudiendo tener acceso a las aplicaciones de forma gratuita o de pago, aunque normalmente es de pago y cualquier con una tarjeta de crédito válida puede acceder y consumir rápidamente el servicio, adecuado cuando a la empresa que ofrece el servicio no le importa compartir recursos virtualizados en la nube y donde el despliegue de la aplicación será de manera provisional [5].

La *ventaja* más clara de las nubes públicas es la *capacidad de procesamiento y almacenamiento sin instalar máquinas localmente*, por lo que no tiene una inversión inicial o gasto de mantenimiento en este sentido, si no que se paga por el uso. La carga operacional y la seguridad de los datos (backup, accesibilidad, etc.) recae íntegramente sobre el proveedor del hardware y software, debido a ello, el riesgo por la adopción de una nueva tecnología es bastante bajo. El retorno de la inversión se hace rápido y más predecible con este tipo de nubes.

Como *inconvenientes* se cuenta *con el acceso de toda la información a terceras empresas, y la dependencia de los servicios en línea (a través de Internet)*. También puede resultar difícil integrar estos servicios con otros sistemas propietarios. Es muy importante a la hora de apostar por un servicio en la nube pública, asegurarse de que se puede conseguir todos los datos que se tengan en ella, gratuitamente y en el menor tiempo posible.

3.2.2 Cloud privada

Actualmente existe una importante tendencia en grandes empresas a la implementación, dentro de su estructura y utilizando la red privada de la propia organización, de las llamadas “nubes privadas” [5]. Este concepto, a priori más cercano al de despliegue tradicional de aplicaciones que al de Cloud Computing “estándar”, hace referencia a redes o centros de procesamiento de datos propietarios que utilizan tecnologías características de Cloud Computing, tales como la virtualización. Así, parten de los principios del Cloud Computing tradicional y ofrecen los mismos servicios pero dentro en la propia estructura de la compañía.

Se suelen *diseñar específicamente para un usuario*, proporcionando un control óptimo

de la información gestionada, de su seguridad y de la calidad de servicio ofrecida. Habitualmente, *el usuario es también propietario* de la infraestructura de nube privada, y tiene control total de las aplicaciones desplegadas en ella.

Los principales inconvenientes de este modelo son los analizados para el paradigma tradicional, por ejemplo los relativos a la *ampliación* de los sistemas informáticos. Esto obliga a adquirir nuevos sistemas antes de hacer uso de ellos, contrariamente a lo ofrecido por las nubes públicas, donde ampliar los recursos se reduce a contratarlos con el proveedor de servicios.

Como ventaja de este tipo de nubes, a diferencia de las nubes públicas, destaca la localización de los datos dentro de la propia empresa, lo que conlleva a una mayor seguridad de estos.

3.2.3 Cloud híbrida

El modelo híbrido combina los modelos anteriormente descritos, sobre nubes públicas y privadas, de manera que se aprovecha la ventaja de localización física de la información gestionada por las nubes privadas con la facilidad de ampliación de recursos de las nubes públicas [5]. Las principales cuestiones a vigilar en este modelo son la privacidad y la protección de datos, al igual que en la nube pública.

Las nubes híbridas consisten en combinar las aplicaciones propias de la empresa con las consumidas a través de la nube pública, entendiéndose también como la incorporación de servicios de Cloud Computing a las aplicaciones privadas de la organización. Esto permite a una empresa mantener el control sobre las aplicaciones críticas para su negocio y aprovechar al mismo tiempo las posibilidades ofrecidas por los servicios ofertados por la nube en aquellas áreas donde resulte más adecuado.

Parece que actualmente este tipo de nubes está teniendo buena aceptación en las empresas, por lo que se están desarrollando software de gestión de nube que permita controlar la nube privada e incorporar al mismo tiempo recursos y servicios de proveedores públicos de Cloud Computing.



Ilustración 8 Esquema Cloud Computing

3.3 Implementaciones

A continuación se describen algunos proveedores de servicios que están marcando la tendencia del mercado en la actualidad. Se ha descrito el proveedor de referencia en IaaS (Amazon Web Services) y la referencia en servicios PaaS (Microsoft Windows Azure).

3.2.1 Amazon Web Services (AWS)

Uno de los proveedores de IaaS más sobresalientes en el mercado es *Amazon Web Services*. Este proveedor permite que sus usuarios creen una Imagen de máquina virtual de Amazon (AMI), esto es, una máquina virtual con el sistema operativo Windows o Linux, en la que el usuario instala sus aplicaciones, librerías y datos que necesite.



Ilustración 9 Logotipo Amazon Web Services [9]

Posteriormente, Amazon ejecuta esa máquina en sus sistemas, y le asigna características físicas (como la capacidad de procesamiento máxima disponible, la cantidad de memoria RAM máxima a utilizar, el espacio de almacenamiento máximo disponible, etc.) de acuerdo al contrato suscrito con el usuario. El usuario accede a esa máquina de manera remota de la misma forma en que accedería a un servidor físico tradicional.

Asimismo, el usuario puede indicar a Amazon que amplíe sus sistemas automáticamente según las condiciones que hayan establecido previamente, y puede monitorizar o controlar en todo momento el estado de su máquina virtual.

En cuanto a *precios*, el *coste se factura por hora de utilización y tipo de recursos asignados a cada máquina física* (como la capacidad de procesamiento, la cantidad de memoria RAM, la cantidad de espacio para el almacenamiento secundario, el sistema operativo utilizado o el software adicional necesitado). Para facilitar el cálculo aproximado de la factura mensual, el propio Amazon contiene una calculadora disponible en su Web.

Los servicios más destacables de Amazon son: EC2 y S3. A continuación se explican las características principales de cada uno de ellos.

3.2.1.1 Amazon Elastic Compute Cloud (EC2)

Amazon Elastic Compute Cloud (Amazon **EC2**) es un servicio web que proporciona capacidad informática con tamaño modificable en la nube. Se ha diseñado con el fin de que la informática web resulte más sencilla a los desarrolladores [6].

La sencilla interfaz de servicios web de Amazon EC2 permite obtener y configurar capacidad con una fricción mínima. Proporciona un control completo sobre sus recursos informáticos y permite ejecutarse en el entorno informático acreditado de Amazon. *Amazon EC2 reduce el tiempo necesario para obtener e iniciar nuevas instancias de servidor a cuestión de minutos, lo que permite escalar con rapidez su capacidad (aumentarla o reducirla) cuando cambien los requisitos informáticos [6]*. Amazon EC2 cambia las reglas económicas de la informática al permitirle pagar sólo por la capacidad que realmente utilice. Amazon EC2 proporciona a los desarrolladores las herramientas necesarias para crear aplicaciones resistentes a errores y para aislarse de los casos de error más comunes.

Amazon EC2 presenta un auténtico entorno informático virtual, que permite utilizar interfaces de servicio web para iniciar instancias con distintos sistemas operativos, cargarlas con

su entorno de aplicaciones personalizadas, gestionar sus permisos de acceso a la red y ejecutar su imagen utilizando los sistemas que desee. Para utilizar Amazon EC2, sólo tiene que:

Seleccionar una imagen de plantilla preconfigurada para pasar a estar activo de inmediato. O bien crear una AMI (Amazon Machine Image) que contenga sus aplicaciones, bibliotecas, datos y valores de configuración asociados [9].

Configurar la seguridad y el acceso a red en su instancia de Amazon EC2. Seleccionar los tipos de instancias y sistemas operativos que desee y, a continuación, iniciar, finalizar y supervisar tantas instancias de su AMI como sea necesario, a través de las API de servicio web o la variedad de herramientas de gestión proporcionadas.

Las *características* que nos brinda Amazon con este servicio son:

- **Elástico** – Amazon EC2 permite aumentar o reducir la capacidad en cuestión de minutos, sin esperar horas ni días. Puede enviar una, cientos o incluso miles de instancias del servidor simultáneamente. Dado que todo está controlado mediante las API del servicio web, su aplicación podrá escalarse automáticamente según sus necesidades aumenten o se reduzcan [9].
- **Control total** – Tendrá control total sobre sus instancias. Tiene acceso de usuario raíz a todas ellas, y puede interactuar con ellas como con cualquier otra máquina. Puede detener su instancia y mantener los datos en su partición de arranque, para reiniciar a continuación la misma instancia a través de las API del servicio web. Las instancias se pueden reiniciar de forma remota mediante las API del servicio web. Asimismo, tiene acceso a la emisión de consola de sus instancias [9].
- **Flexible** – Tiene la opción de varios tipos de instancias, sistemas operativos y paquetes de software. Amazon EC2 permite seleccionar una configuración de memoria, CPU, almacenamiento de instancias y el tamaño de la partición de arranque óptimo para su sistema operativo y su aplicación. Por ejemplo, entre sus opciones de sistemas operativos se incluyen varias distribuciones de Linux, Microsoft Windows Server y OpenSolaris [9].
- Con un diseño pensado para su uso con otros Amazon Web Services – Amazon EC2 trabaja con Amazon Simple Storage Service (Amazon S3), Amazon SimpleDB y

Amazon Simple Queue Service (Amazon SQS) para proporcionar una solución completa de computación, procesamiento de consultas y almacenamiento en una gran gama de aplicaciones.

- **Fiable**– Amazon EC2 ofrece un entorno muy fiable en el que las instancias de sustitución se pueden enviar con rapidez y anticipación. El servicio se ejecuta en los centros de datos y la infraestructura de red acreditados de Amazon. El compromiso del contrato a nivel de servicio de Amazon EC2 es de una disponibilidad del 99,95% en cada Región de Amazon EC2 [9].
- **Seguro** – Amazon EC2 ofrece diversos mecanismos para proteger los recursos informáticos [9].

Amazon EC2 incluye interfaces de servicio web para configurar el cortafuegos que controla el acceso de red a grupos de instancias, y el acceso entre estos.

Al iniciar recursos de Amazon EC2 en Amazon Virtual Private Cloud (Amazon VPC), puede aislar sus instancias informáticas especificando el rango de IP que desea utilizar, así como conectarse a su infraestructura de IT existente mediante la red cifrada IPsec VPN estándar del sector.

3.2.1.2 Amazon Simple Storage Service (S3)

Otro de los servicios clásicos de Amazon es S3. **Amazon S3 es el servicio de almacenamiento en la nube.** Está diseñado para realizar el cómputo web a gran escala más fácil. Amazon S3 proporciona una sencilla interfaz de servicios web que pueden ser utilizados para almacenar y recuperar cualquier cantidad de datos, en cualquier momento, desde cualquier lugar en la web. Se permite a cualquier desarrollador acceder a la infraestructura altamente escalable, fiable, seguro, rápido, que Amazon utiliza para ejecutar su propia red mundial de sitios web. El servicio tiene como objetivo maximizar los beneficios de escala y pasar esos beneficios a los desarrolladores. Amazon S3 está creado intencionadamente con un conjunto de funciones mínimas [9].

- **Escriba, lea y elimine objetos** que contengan desde 1 byte hasta 5 gigabytes de datos.

El número de objetos que puede almacenar es ilimitado.

- ***Cada objeto está almacenado en un depósito***, y se recupera por medio de una clave exclusiva asignada por el desarrollador.
- ***Un depósito puede estar almacenado en una de varias Regiones***. Debe escoger una Región cercana para optimizar la latencia, minimizar los costes o afrontar exigencias reguladoras. Amazon S3 está actualmente disponible en las Regiones EE. UU. estándar, UE (Irlanda), oeste EE. UU. (norte de California) y Asia Pacífico (Singapur). La Región EE. UU. estándar redirige automáticamente las solicitudes hacia instalaciones situadas en el norte de Virginia o en el noroeste del Pacífico por medio de asignaciones de red.
- ***Los objetos almacenados en una Región nunca abandonan la misma***, a menos que usted los transfiera. Por ejemplo, los objetos almacenados en la Región UE (Irlanda) nunca salen de la UE.
- ***Se incluyen mecanismos de autenticación*** diseñados para garantizar que los datos se mantienen seguros frente a accesos no autorizados. Los objetos pueden hacerse privados o públicos, y pueden otorgarse derechos a usuarios determinados.
- ***Utiliza interfaces REST y SOAP basadas en estándares*** diseñadas para funcionar con cualquier kit de herramientas de desarrollo en Internet.
- ***Está creado para ser flexible y permitir añadir fácilmente protocolos o capas funcionales***. El protocolo de descarga predeterminado es HTTP. Se proporciona un protocolo BitTorrent para reducir los costes de la distribución a gran escala.

3.2.2 Microsoft Windows Azure

Uno de los proveedores que más ha destacado por el momento es ***Windows Azure***, que ofrece la creación de aplicaciones Web adaptadas a sus sistemas y su despliegue en los mismos con ciertas limitaciones de consumo. Admite varios lenguajes de programación y permite compartir las aplicaciones con todo el mundo o sólo con quien se desee. Asimismo, se puede comenzar a usar gratuitamente y sólo pagar si se necesitan incrementar los límites o los recursos utilizados posteriormente, con un coste inferior al de los sistemas tradicionales.

Otras empresas ***proveedoras de servicios de PaaS*** son ***Velneo*** o ***Azure***. Como ejemplo de esta última destaca ***Windows Azure Platform***, una plataforma que ofrece a los desarrolladores de

aplicaciones un entorno para crear y ejecutar sus aplicaciones en los centros del proveedor. ***Dicho entorno proporciona las funcionalidades necesarias para que las aplicaciones creadas con él puedan realizar diversas tareas de negocio, almacenar información en bases de datos de la “nube” y comunicarse con otras aplicaciones creadas [5].*** con ese o con otros entornos. Los escenarios más comunes donde se emplea esta plataforma abarcan desde la creación de sitios Web para empresas hasta el almacenamiento de grandes cantidades de información de forma más barata y ampliable en bases de datos o sistemas de almacenamiento masivo.

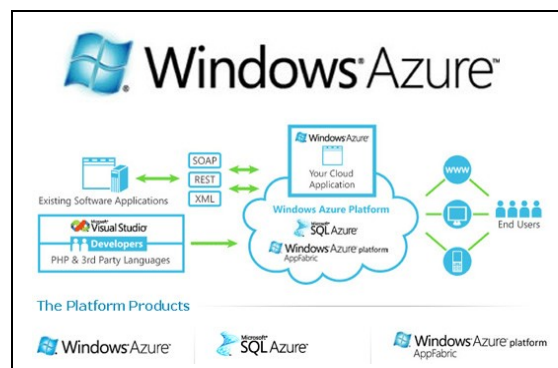


Ilustración 10 Plataforma Windows Azure [14]

Entre las ofertas y servicios que ofrece Windows Azure son:

- **Windows Azure**: sistema operativo como un servicio en línea
- **Microsoft SQL Azure**: solución completa de base de datos Cloud relacional.
- **Plataforma AppFabric de Windows Azure**: conecta servicios Cloud y aplicaciones internas.



Ilustración 11 Servicios en Windows Azure [17]

La plataforma permite la total integración de los servicios mencionados anteriormente con herramientas y aplicaciones que ya existían de Microsoft, como las que todos conocemos, Microsoft Office, Exchange, todas ellas en modalidad SaaS, aunque la novedad reside en su herramienta CRM Microsoft Dynamics, para el seguimiento de clientes y gestión empresarial en general.

3.4 Cloud Computing y SOA

En esta sección se analizarán y evaluarán cómo SOA y Cloud Computing coexisten. La sección discutirá las sinergias entre ambos modelos y se explicará como *el efecto de esta combinación es mayor que la suma de los efectos individuales*.

La mayor parte de las implementaciones e integraciones basadas en Cloud Computing que se realizan hoy en día tienen un propósito común, optimizar las aplicaciones a gran escala.

Estas aplicaciones necesitan ser flexibles, y la adopción de SOA puede proporcionar a los desarrollos basados en Cloud Computing un diseño para el acceso a los servicios a través de un bajo acoplamiento y la habilidad de evolucionar fácilmente que de otro modo sería muy complejo. El papel de SOA en Cloud Computing está comenzando a ser sistémico ya que ayuda a la comprensión de los procesos a nivel de dominio del problema. Este apartado presenta como un buen diseño orientado a servicios proporciona valor añadido a una arquitectura basada en Cloud Computing, así como SOA y Cloud Computing resultan soluciones informáticas complementarias [12].

Cloud Computing es un tipo de solución, una alternativa más con las que afrontar nuestros retos profesionales, una forma de crear un sistema en el que algunos o todos sus

recursos de TI existentes pueden estar en algún nivel de la infraestructura Cloud de terceros, tales como Amazon EC2 o Force.com. Por lo tanto, Cloud Computing es algo que puede implicar parte o la totalidad de una arquitectura. La diferencia principal es que los recursos del sistema van a ser más distribuidos si cabe.

SOA es todo sobre el proceso de definición de una solución informática o la arquitectura, mientras que el Cloud Computing es una alternativa arquitectónica. Por lo tanto, SOA no puede ser sustituido por Cloud Computing. De hecho, la mayoría de soluciones basadas en la nube van a ser definidas a través de SOA. No compiten, son conceptos complementarios [12].

Otro aspecto que debemos considerar, es que SOA, además de ser un patrón arquitectónico, es una estrategia que alinea la tecnología y las realidades del negocio, dejando un entorno listo para sufrir cambios (tanto tecnológicos, como estratégicos) con el menor impacto en costes posible. La decisión de introducir un modelo de servicios basados en Cloud Computing sin duda introduce serios cambios de la tipología comentada anteriormente, veremos en qué medida SOA puede mitigar el gap de adoptar un modelo basado en Cloud.

Una de las principales ventajas de SOA es que está alineado con los procesos de negocio de la organización, por lo tanto podemos delegar los aspectos específicos de la infraestructura al proveedor de servicios Cloud que escojamos. La ventaja de este aspecto reside en que, los profesionales de una organización pueden estar más preocupados en alinear negocio y TI sin preocuparse si la infraestructura lo podrá soportar, dicho de otro modo, se podrá enfocar esfuerzos en las necesidades reales de la organización [12]. En esta línea cabe afirmar que de algún modo SOA proporciona un entorno con las bondades que se han explicado a lo largo del trabajo, y Cloud Computing proporciona alta escalabilidad de forma transparente para el usuario, esta abstracción de infraestructura y escalabilidad evita tener que pensar en balanceos de carga de los servicios, centrándonos en las actividades que realmente proporcionan valor dentro de la organización, es decir, SOA está orientada al negocio y Cloud está orientado a la infraestructura.

Por supuesto, la escalabilidad de Cloud Computing se aplica a todo tipo de estilos arquitectónicos, no sólo a SOA. Sin embargo, una de las ideas centrales en SOA, la reutilización de la funcionalidad y la coherencia, es mucho mayor cuando no hay problemas de escalabilidad,

por lo tanto podemos afirmar que existe una clara sinergia entre reutilización y escalabilidad.

En cuanto a la gestión y gobernanza de los servicios, SOA es en sí mismo un marco de referencia [12], es importante considerar que Cloud Computing va a llevar los servicios SOA fuera del firewall interno donde solían alojarse, se hace necesario un sistema de gestión y gobernanza tremendamente maduros. El tremendo esfuerzo por introducir gobierno SOA se vería recompensando, ya que se le proporcionaría a nuestra arquitectura Cloud características añadidas tales como monitorización para el control de la actividad y cumplimiento de SLAs por parte del proveedor de infraestructura Cloud [15], y un mayor control de la seguridad para mitigar el aumento de riesgo que supone alojar en el exterior los servicios al adoptar Cloud Computing.

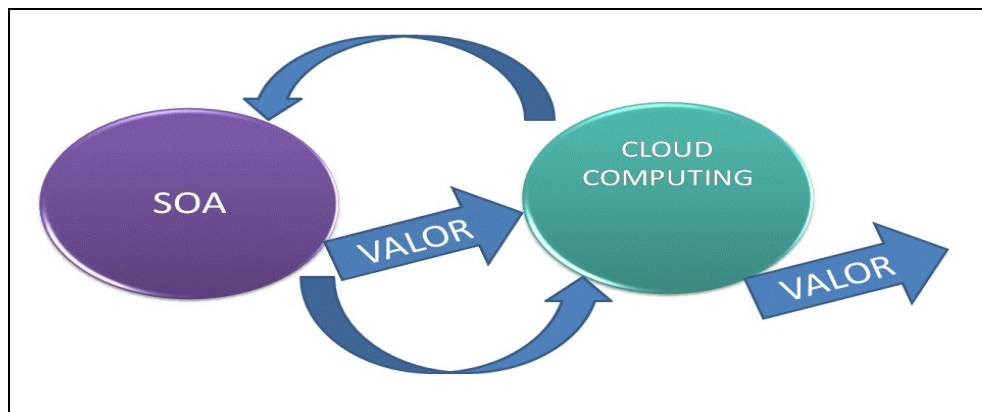


Ilustración 12 Sinergias entre SOA y Cloud Computing

Tal y como se puede observar en la Ilustración 12, las sinergias entre Cloud Computing y SOA aportarán un gran valor añadido a nuestro sistema y a nuestra infraestructura. La combinación no solo nos aporta valor a nuestro negocio, SOA se verá potenciado ya que Cloud Computing extenderá SOA fuera de las fronteras de los firewalls internos de la organización. Y SOA proporcionará a Cloud Computing todo un marco de gestión y gobernabilidad, interfaces débilmente acopladas basadas en estándares y una base de diseño de la arquitectura a través de los principios de orientación a servicios.

Capítulo 4:

Adopción, beneficios y riesgos en Cloud Computing

En este capítulo se explicarán los motivos por los cuales me puede interesar un modelo

basado en la nube. Ya hemos visto que permite el lanzamiento rápido de servicios, el acceso a los mismos desde cualquier lugar, se facilita su difusión y publicidad, es capaz de absorber crecimientos rápidos y picos de carga, facilita la integración con otros servicios, etc. Veamos cuáles son las principales ventajas competitivas que Cloud Computing ofrece y que parámetros son importantes para elegir unos modelos u otros. Por último analizaremos que riesgos debemos asumir y poner esfuerzo para evitarlos.

4.1 Aspectos claves en la elección de IaaS frente a sistemas tradicionales

A continuación se muestra de forma esquematizada una comparación entre el uso de los sistemas informáticos tradicionales requeridos por una organización, y los cambios que puede aportar el uso de IaaS.

En el sistema tradicional se desaprovechan recursos, mientras que en un sistema IaaS se consigue una mayor eficiencia en su utilización (menos sistemas informáticos desaprovechados, menor energía consumida por unidad de información gestionada, sistemas más automatizados).

Asimismo, los recursos físicos se gestionan de manera unificada por parte del proveedor, por lo que el tiempo necesario para adaptar los recursos de un usuario de IaaS a sus necesidades reales en cada momento se reduce notablemente. Así, el proveedor de servicios podrá optimizar el uso en todas sus máquinas, reduciendo así los costes por el servicio.

Las máquinas físicas utilizadas para IaaS son propiedad del proveedor de servicios, con el consiguiente riesgo de que éste podría dejar en algún momento de ofrecer el servicio. Sin embargo, esta característica aporta importantes ventajas, por ejemplo el hecho de que el equipamiento se renueva más fácilmente debido a la economía de escala de estos sistemas y de que se siguen estándares que facilitan la interoperabilidad entre fabricantes.

Por otra parte, aunque en los sistemas tradicionales la infraestructura es propiedad de la organización, tiene el inconveniente de que está asociada a sistemas que se pueden quedar obsoletos o ser incompatibles con otros.

El empleo de servidores virtuales dedicados, que simulan una máquina con un sistema operativo propio, permite separar esta máquina simulada del resto de funcionalidades ofrecidas por el resto de la máquina física. Así, si la máquina física falla, se puede utilizar la máquina simulada en otra máquina física, por lo que las consecuencias de un fallo en alguna de las

máquinas y el tiempo de recuperación se reducen drásticamente.

Además, estas máquinas utilizadas en IaaS se encuentran replicadas, y disponen de centros físicos de almacenamiento y procesamiento con ciertas características (como la refrigeración de las máquinas, su seguridad física, etc.) que, en el caso de ser implantadas en los centros tradicionales, tendrían unos costes demasiado elevados.

Veamos los aspectos clave a tener en cuenta por parte de una empresa a la hora de escoger la implantación de una solución IaaS.

Aspectos técnicos

El proveedor de servicios IaaS ofrece una infraestructura informática para determinados Sistemas Operativos y software (como bases de datos, alojamiento Web, entornos de desarrollo de aplicaciones, servidores de aplicaciones, codificación y *streaming* de vídeo) y la empresa usuaria debe tener en cuenta que no podrá incorporar otros sistemas particulares de su solución.

Aspectos estratégicos

Los usuarios pueden desplegar máquinas virtuales en la infraestructura física de IaaS en muy poco tiempo (en los casos más sencillos, en pocos minutos), por lo que *se reduce significativamente el tiempo y coste asociado de puesta en marcha de nuevos sistemas*. Además, la *capacidad de ampliación de los recursos hardware* es bastante menos costosa y rápida que en el caso tradicional.

Por otro lado, la *disponibilidad y calidad de servicio* ofrecidos en IaaS suelen estar *garantizados* durante casi todo el tiempo de utilización, ofreciendo soluciones alternativas en el caso de falta de servicio. Así, uno de los aspectos estratégicos por los que una empresa podría optar por IaaS sería conseguir una reducción significativa de la inversión en recursos para garantizar la disponibilidad del sistema, que generalmente consiste en la adquisición de sistemas físicos redundantes para evitar pérdidas de servicios que habitualmente no se usan, con el consecuente coste que suponen los recursos desperdiciados.

Otro aspecto estratégico a tener en cuenta es el hecho de que la deslocalización física del hardware utilizado junto con el uso de redes privadas virtuales (VPN) posibilita el *acceso*

simultáneo y seguro de múltiples empleados de la organización a los sistemas con mayor facilidad de disponer de alta velocidad de conexión.

Aspectos económicos

El *coste de utilización* de los servicios IaaS sigue varios modelos:

- En el primer modelo se cobra una *tarifa fija por hora y unidad de recursos utilizados*. Esto suele ser útil para aplicaciones poco probadas en los que el *consumo* sea *impredecible*.

- En el segundo, se ofrece la posibilidad de disponer de *un recurso reservado, con un pequeño coste, y un cobro por el uso posterior*. Suele emplearse en aplicaciones con un uso predecible y que necesiten de capacidad reservada, incluyendo recuperación ante desastres.

- En otros modelos, *se paga en función del uso instantáneo* que se haga de los recursos. Este último caso es adecuado cuando se necesita una alta flexibilidad de los recursos en determinados momentos, por ejemplo, grandes consumos en momentos determinados del día no predecibles.

Habitualmente, se pueden combinar estos modelos para adaptarlos a las necesidades específicas del usuario.

Aspectos legales

El uso de IaaS obliga a sus usuarios a que no exijan la localización en todo momento de la ubicación física de la información gestionada. Otra característica a tener en cuenta es que algunos de los proveedores de servicios IaaS realizan back-ups o copias de la información que gestionan. Estos dos aspectos son importantes si se gestiona información protegida de carácter personal o empresarial.

4.2 Aspectos claves en la elección de PaaS frente a sistemas tradicionales

Al igual que en IaaS, el uso de PaaS aporta ciertas mejoras y facilidades frente a sistemas tradicionales, entre las que destacan:

Calidad final

El importante esfuerzo colaborativo realizado en aplicaciones informáticas creadas con

PaaS hace posible que en la gran mayoría de los casos el usuario perciba una calidad final mayor que la ofrecida por aplicaciones convencionales:

A diferencia del proceso tradicional, donde se desarrolla en un entorno y posteriormente se traslada a otros para su prueba y puesta en marcha, *en PaaS la creación de la aplicación se realiza en un entorno unificado y que será el mismo al que accederán sus usuarios finales*, por lo que se reducirán los errores debidos a las diferencias entre entornos y serán más sencillos de corregir.

Por otra parte, el hecho de gestionar toda la información de manera centralizada permite obtener estadísticas de la información real accedida en cada momento, las cuales podrían reutilizarse para mejorar la aplicación u otras similares.

Interoperabilidad con otros sistemas en línea

Un elevado número de aplicaciones, tales como sistemas de comercio electrónico o sistemas de predicción meteorológica, requieren acceso en tiempo real a información disponible en otros puntos de Internet u otras redes. Trabajar con PaaS facilita la conectividad a esos recursos, ya que ambos estarán diseñados específicamente para trabajar de forma conjunta, y permite actualizar automáticamente las conexiones entre los recursos, lo cual supone una ventaja respecto al desarrollo realizado en los sistemas tradicionales.

Asimismo, PaaS utiliza frecuentemente una infraestructura IaaS, ya descrita anteriormente, beneficiándose de sus ventajas como ampliar o reducir los recursos físicos eficientemente.

A continuación, se citan los aspectos clave a tener en cuenta por parte de una empresa a la hora de escoger la implantación de una solución PaaS.

Aspectos técnicos

A la hora de crear las aplicaciones que posteriormente se situarán en los sistemas PaaS, hay que tener en cuenta que la **tecnología** a usar en las mismas debe ser **compatible** con dichos

sistemas. En general, la tecnología estará basada en estándares internacionales, pero el rango de funciones que ofrece puede ser bastante limitado en ciertos casos. Por ejemplo, en la creación de aplicaciones Web sobre *Google App Engine*, descrito en el apartado 6.4, los lenguajes de programación utilizados únicamente pueden ser *Python* y *Java*. Esto puede reducir el rendimiento de determinadas aplicaciones.

Por otro lado, las plataformas PaaS permiten ampliar fácilmente los recursos disponibles para la aplicación ya que, por ejemplo, se usan sistemas de ficheros y bases de datos específicas para ello. Sin embargo, la gestión de la información en estos sistemas es bastante más compleja, por lo que en la práctica se confía parte de ese control al proveedor de servicios. Se deberá conocer hasta qué punto la información gestionada es crítica, y qué *niveles de seguridad* se establecerán. Esto obliga al proveedor a suministrar información sobre la estructura de los datos.

Finalmente, la *gestión de las aplicaciones* una vez situadas en las máquinas de PaaS suele ser más sencilla que en las instalaciones tradicionales, pero se dispone de menor control de todos los sistemas.

Aspectos estratégicos

Con PaaS se ofrecen soluciones de almacenamiento y computación para los desarrolladores de software accesibles independientemente de la ubicación geográfica, adoptando así economías de escala y flexibilidad de configuración sin que los usuarios de la plataforma necesiten mantener la tecnología subyacente.

Aspectos económicos

Los proveedores PaaS habitualmente ofrecen un periodo de pruebas sin coste en los que los usuarios pueden comprobar las ventajas competitivas que pueden encontrar en PaaS, o pueden experimentar con nuevas aplicaciones adaptadas a ese tipo de sistemas.

Comúnmente, se ofrece un *coste por uso* de los recursos del sistema, es decir, se cobra una cantidad fija por cada GByte de almacenamiento, por cada hora de procesamiento o por cada GByte de información transmitida hacia terceros. Asimismo, para fomentar la implantación de PaaS se tiende a ofrecer un *servicio gratuito limitado a una cantidad diaria de uso*, a partir del cual se realiza el cobro según se ha descrito.

Aspectos legales

Al comenzar a usar los servicios PaaS, se establece un acuerdo entre el proveedor y el usuario en el que se describen las condiciones del servicio ofrecido. Habitualmente, el usuario se compromete a no realizar un uso indebido de los sistemas que se le ofrecen.

Por otro lado, el proveedor señala las condiciones de tarificación del servicio, de garantía de acceso y gestión adecuada de la información, y de las garantías legales en caso de errores o desastres en sus sistemas.

4.3 Aspectos claves en la elección de SaaS frente a sistemas tradicionales

En la práctica, las aplicaciones SaaS se diferencian de las aplicaciones tradicionales en ciertos aspectos fundamentales, varios de ellos ya comentados en las ventajas generales ofrecidas por Cloud Computing:

Coste

Las aplicaciones tradicionales tienen un coste inicial alto basado en la adquisición de las licencias para cada usuario. Estas licencias suelen ser a perpetuidad, es decir, no imponen restricciones temporales a su uso.

En cambio, para las aplicaciones SaaS el coste se basa en el uso, no en el número de usuarios, y el gasto de mantenimiento es nulo, ya que la las aplicaciones las gestiona el propio proveedor. Un modelo más equilibrado entre ambos podría ser el uso de sistemas basados en un uso ilimitado durante un periodo de tiempo.

Administración informática

Las organizaciones que usan software tradicional comúnmente necesitan un departamento de administración o una subcontratación de esas competencias a otras empresas para que se resuelvan problemas asociados a la implantación de la infraestructura informática o la resolución de problemas como la seguridad de los sistemas, la fiabilidad, el rendimiento ofrecido o problemas de disponibilidad. Si se utiliza SaaS, esta administración se ve reducida considerablemente, ya que la realiza el proveedor de servicios basándose en el acuerdo de nivel

Independencia de las mejoras en las aplicaciones

El proveedor de SaaS no sólo se encarga de la administración, como se acaba de comentar, sino que también es el que se encarga de instalar, mantener y actualizar las aplicaciones del cliente, por lo que este último podrá invertir su tiempo en las tareas propias de su negocio, utilizando sus recursos en las áreas más estratégicas.

Existen ciertos aspectos clave a la hora de decidir optar por soluciones SaaS de forma total o parcial en la organización:

Aspectos técnicos

Las aplicaciones informáticas SaaS suelen ofrecer cierta flexibilidad de configuración para su adaptación a las necesidades del cliente. Sin embargo, existen empresas que necesitan aplicaciones muy particulares, cuya **adaptación** a partir de *software* SaaS es demasiado costosa económica o técnicamente para los proveedores de servicios. En esos casos, esas empresas deberán desarrollar un *software* específico.

Otro factor a considerar es el **tipo y la cantidad de datos a transmitir** a las aplicaciones de la empresa. Habitualmente, las redes de comunicaciones ofrecen altas velocidades de transmisión de datos en sus instalaciones, y menores velocidades en su acceso a Internet. Si se utiliza una aplicación SaaS, se ha de considerar que se deberá acceder a Internet para transmitir información. Para paliar la lentitud del sistema al transmitir información, las aplicaciones SaaS sólo transmiten la información estrictamente necesaria (también llamada solución basada en caché) o agrupan la información para transmitirla en el momento óptimo (solución denominada transmisión por lotes).

Aspectos estratégicos

En algunas empresas se presenta cierta **resistencia** a que las funcionalidades de gestión de la empresa se externalicen hacia sistemas en Internet. Sin embargo, se pueden realizar proyectos de prueba en los que se analicen las mejoras que puede aportar a la empresa el uso de estos sistemas SaaS. En consonancia con ello, los proveedores de SaaS ofrecen a menudo periodos de prueba para que las empresas puedan realizar estos análisis.

Aspectos económicos

Para realizar un análisis adecuado se ha de comparar el coste total de propiedad (llamado en términos económicos, TCO) de una aplicación SaaS frente al del software tradicional. Aunque el coste inicial de una aplicación SaaS es habitualmente inferior, el coste a largo plazo se puede llegar a incrementar debido a las tarifas por el uso del servicio. Los factores más destacados que afectan al TCO de una aplicación incluyen el número de licencias de usuario del software necesarias o la cantidad de configuración requerida para integrar la aplicación a la infraestructura de la organización. Asimismo, se ha de tener en cuenta si se han realizado inversiones recientes en infraestructuras de las que se espera un retorno de la inversión en cierto periodo de tiempo.

Aspectos legales

Algunas organizaciones que operan en varios países están sujetas a legislaciones que exigen la obtención de informes que describan cómo gestionan la información. Sin embargo, es posible que los proveedores de SaaS no sean capaces de proporcionar esos informes, o de utilizar sistemas de trazabilidad o seguimiento de la información que gestionan. ***Todo esto debe aparecer claramente especificado en el acuerdo de nivel de servicio (SLA).***

4.4 Otros XaaS: El caso del Virtual Desktop Infrastructure (VDI)

Dada la creciente popularidad que están adquiriendo las distintas soluciones basadas en Cloud Computing, ha surgido una familia de modelos de servicios que se añaden a las ya comentadas (SaaS, PaaS, IaaS) y que se denomina XaaS. Esta familia de servicios tiene como ejemplos, Testing as a Service, Security as a Service, Modeling as a Service, todo como servicio. ***En ocasiones cuando se descontrola la terminología sobre un concepto, esta, deja de aportar valor***, cuando lo que realmente aportar valor añadido es la propia funcionalidad que te ofrece determinado servicio. No obstante, hay un modelo de la familia XaaS y que está siendo adoptando por grandes compañías como por ejemplo el BBVA. ***Este modelo se denomina Desktop as a Service (DaaS), aunque su nombre más popular es Virtual Desktop Infrastructure (VDI).***

La idea que hay detrás de lo que se llama Virtual Desktop Infrastructure (VDI) es

ejecutar sistemas operativos de escritorio y aplicaciones en máquinas virtuales que residen en los servidores del centro de datos [8]. A los sistemas operativos de escritorio dentro de máquinas virtuales también se les conoce como los *escritorios virtuales*. Los usuarios acceden a los escritorios virtuales y aplicaciones desde un cliente PC de escritorio o de *cliente ligero* con un protocolo de visualización a distancia y llegar a casi todas las funciones como si las aplicaciones se cargan en sus sistemas locales, con la diferencia de que las solicitudes son administradas. Al igual que la virtualización de servidores, VDI ofrece muchos beneficios. En concreto, *las tareas administrativas y de gestión se han reducido significativamente [8]*; aplicaciones de forma rápida se pueden agregar, eliminar, actualizar, y parches, la seguridad es centralizada.

VDI tiene algunas similitudes con una arquitectura de infraestructura de aplicaciones para compartir, donde el acceso del usuario es a través de un cliente ligero. Sin embargo, hay diferencias. Por ejemplo, VDI permite a las empresas a aislar a los usuarios entre sí en el caso de un fallo de sesión individual. VDI También puede ejecutar la mayoría de aplicaciones de forma nativa sin modificaciones. VDI soporta aplicaciones que requieren un cliente "pesado". Esta capacidad de soportar toda la gama de tipos de escritorio es esencial, ya que muchos usuarios quieren los beneficios, tales como el espacio de almacenamiento personal, que un PC completo ofrece.

Básicamente, los usuarios desean las características y flexibilidad del escritorio tradicional, pero sin los problemas de la tasa de fallos.

Los cambios como la instalación de una nueva aplicación, la actualización de una aplicación existente, o aplicar un parche se puede hacer sin tener que tocar la PC física del usuario es transparente al usuario [8].

La programación y automatización de parches y actualizaciones tienen una tasa de éxito mayor ya que se puede iniciar / detener las máquinas virtuales de escritorio en el centro de datos [8]. Estas máquinas virtuales son independientes de hardware y se puede ejecutar en cualquier servidor de centro de datos y se puede acceder desde cualquier cliente. Además, los datos asociados con estas aplicaciones se pueden almacenar en el centro de datos, por lo que es más fácil hacer una copia de seguridad de los datos y protegerlo de usuarios no autorizados.

Además de hacer que sea más fácil de desplegar y mantener aplicaciones, *VDI también*

simplifica la resolución de problemas [8]. Por ejemplo, cuando un usuario llama al servicio de asistencia, el personal puede trabajar en el problema en el centro de datos y no tiene que visitar el escritorio.

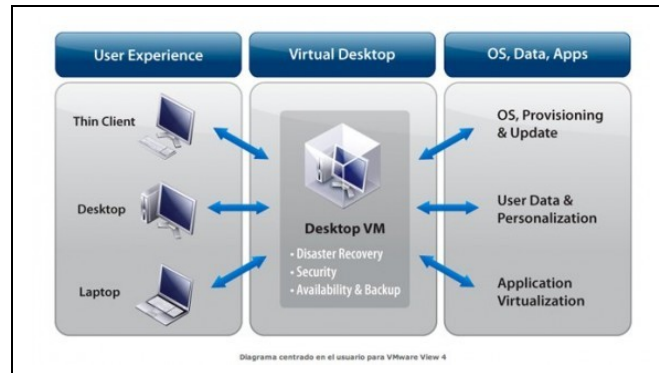


Ilustración 8 Modelo Virtual Desktop Infrastructure [10]

VDI también se puede utilizar para proporcionar acceso a aplicaciones y datos a los usuarios remotos que no están dentro del firewall de la pared de la empresa. Esto es útil cuando un departamento de TI debe apoyar a los usuarios que trabajan desde casa o en otros, oficinas geográficamente dispersas. Apoyar a los usuarios es a menudo una tarea difícil. Cuando hay problemas, el usuario a menudo se necesita enviar su escritorio o portátil a la oficina principal para su reparación. Con VDI, los problemas son más fáciles de solucionar ya que los sistemas virtuales se mantienen en el centro de datos donde hay un personal de TI.

Un beneficio adicional del uso de VDI es que *permite a las empresas a mantener la seguridad y cumplir con las regulaciones de cumplimiento, sin tener que poner como foco tanto en la seguridad del PC* [8] virtuales mediante la asignación de más recursos de CPU y memoria.

VDI reduce el tiempo de inactividad, las velocidades de la resolución de problemas, mejora la capacidad de gestión y control, y ayuda a mantener la seguridad de TI y protección de datos. *El resultado final es una mayor disponibilidad y productividad de los trabajadores mejora.*

4.5 Riesgos en Cloud Computing

Como consecuencia directa de adoptar un nuevo modelo, siempre cabe la posibilidad de que se deban asumir riesgos adicionales, *en Cloud Computing la mayoría de los riesgos son similares y comunes al resto de modelos* y sistemas de información, pero *es conveniente afrontarlos de cara y añadir soluciones*. A continuación se analizan los riesgos más comunes en Cloud Computing.

En secciones anteriores se han mencionado las características y bondades que ofrece Cloud Computing, pero el motivo de este apartado es señalar que este nuevo modelo no debe tomarse a la ligera, que no es fácil adoptarlo y que supone un reto. Existen determinados riesgos, riesgo de seguridad interna, riesgo de seguridad externa, riesgo de caída del servicio, riesgo de protección de datos, riesgo de pérdida de datos. Pero entraremos a detallar los introducidos especialmente por Cloud Computing.

Riesgo de pérdida de datos:

Una de las mayores ventajas de mover nuestros recursos a la nube, es la despreocupación de realizar backups diariamente, resulta una preocupación menos. Pero existe el riesgo de que toda esa información se pueda perder, si estamos hablando de un Cloud público, el SLA debe asegurar la infraestructura está replicada en distintas zonas geográficas y un sistema de recuperación tremendamente maduro y gestionado [1]. Una solución para prevenir pérdida de datos en Clouds públicos es optar por una solución de Cloud híbrida, en la cual solo migraremos a la parte pública nuestros procesos menos críticos. Otra posible solución es optar por una segunda nube pública que sea redundante a la primera y que sirva como backup.

Riesgo de caída del servicio:

Uno de los riesgos que más impacto puede causar en nuestro negocio, es sin duda, la caída del servicio, este riesgo es mayor si estamos haciendo uso de una IaaS pública [1]. La caída de un servicio no solo afecta a que durante un tiempo determinado estemos teniendo pérdidas ya

que nuestra funcionalidad está caída. Si no que además nuestra imagen con respecto a nuestros clientes se ve deteriorada si no lo gestionamos bien.

Servicio	Fecha	Duración
<i>Amazon S3</i>	15/02/2008	2 horas
<i>Google GMail</i>	16/02/2008	30 horas
<i>MS-Azure</i>	13/03/2009	22 horas
<i>Salesforce.com</i>	11/02/2009	6 horas
<i>Microsoft Sidekick</i>	4/10/2009	6 días + pérdida de datos
<i>Google GMail</i>	29/10/2009	2 horas
<i>Amazon EC2</i>	19/04/2011	14 horas

Tabla 1 Ejemplos caídas de servicio

Como solución es necesario solicitar un completo SLA bajo el cual estemos blindados si nuestro proveedor de servicios no cumple con las exigencias negociadas. Algunos ejemplos históricos de caídas de servicio de grandes proveedores son los que aparecen en la figura a continuación.

Capítulo 5:

Integración de un caso de estudio en la nube

A continuación se explicará todo el desarrollo del caso de estudio realizado y que supondrá la consecución de uno de los objetivos de este proyecto fin de master.

En apartados anteriores se han comentado las características y bondades de la computación en la nube, pero es ocasión de llevarlo a la práctica. Es importante definir un claro posicionamiento, describiendo las características del que va a ser nuestro nuevo modelo y posteriormente ejecutarlo sobre nuestro sistema de partida inicial (Gesimed). Analizaremos las ventajas introducidas, así como los obstáculos encontrados de adoptar Cloud Computing y el impacto que ha tenido sobre la arquitectura de nuestro sistema, así como un análisis de costes y del retorno de la inversión (ROI) como consecuencia de adoptar este modelo.

5.1 Sistema de información inicial: *Gesimed*

La integración que realizaremos en este proyecto, tiene como punto de partida la última versión de Gesimed [11] y que supuso el caso de estudio del proyecto fin de carrera hace dos años. *Gesimed se implementó con servicios web y basado en SOA*, la arquitectura final se diseñó de la siguiente manera.

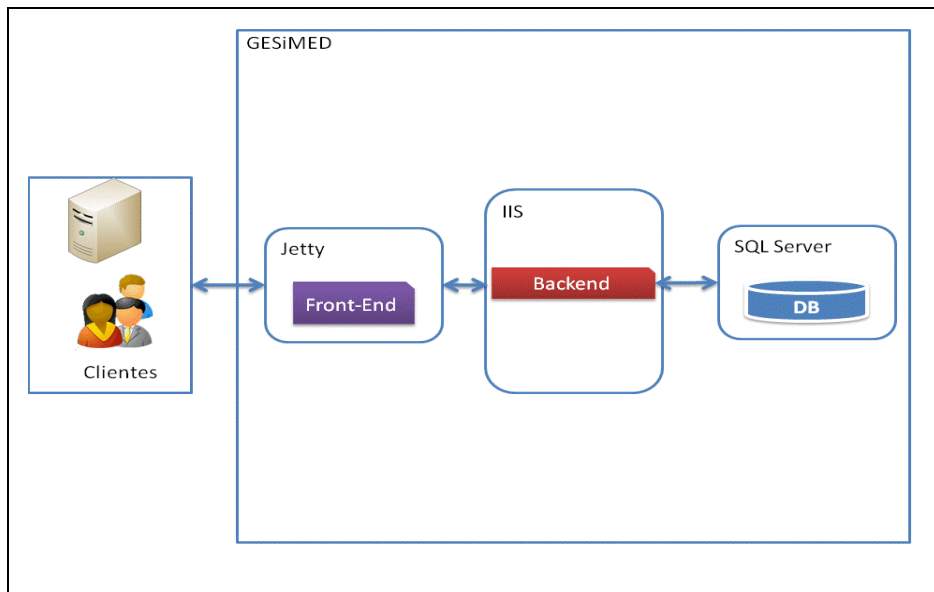


Ilustración 13 Arquitectura de Gesimed [12]

Front-End: Este módulo implementa la interfaz web que se le presenta al usuario mediante servicios web REST. Tiene integrados los servicios SOAP del Back-End y realiza las tareas de orquestador de servicios en función de los eventos que provoca el usuario. Está basado en tecnología Java y alojado en un entorno de máquina local mediante un servidor de aplicaciones Jetty.

Back-End: Este subsistema implementa mediante servicios web compuestos basados en WSDL todo el núcleo de Gesimed, se diseñaron e implementaron todos los servicios necesarios para cubrir toda la funcionalidad (Login, gestión de usuarios, gestión de imágenes, etc.). Este módulo accede a los datos alojados en la base de datos, incluso las imágenes están alojadas en la base de datos dada su naturaleza objeto relacional, asumimos la base de datos como un sistema legacy. Este Back-End está implementado con tecnología .Net y Windows Communication Foundation (WFC).

5.2 Posicionamiento de Gesimed en la nube

Una vez realizado el estudio sobre el paradigma de Cloud Computing, ya sabemos que alternativas y ventajas nos ofrece. *En esta sección se explicará de qué manera vamos a posicionar Gesimed en la nube*, como ya se ha visto los modelos pueden ser SaaS, PaaS, IaaS. Por ello se detallará cuál va a ser el posicionamiento exacto de Gesimed en la nube. *Los motivos se justificarán en detalle en apartados posteriores*, ya que esta sección tan solo nos permitirá tener una idea global y de alto nivel de la estrategia seleccionada para nuestro caso de estudio.

	IaaS	PaaS	SaaS
Gesimed	✗	✗	✗
AmazonWS	✓	✗	✗
MS-Azure	✓	✓	✗
GesimedSaaS	✓	✗	✓
WaveMaker	✗	✓	✗
Force.com	✓	✓	✓

Tabla 2 Posicionamiento de GesimedSaaS en la nube

La tabla muestra cuál será nuestra posición en la nube con respecto al resto del mercado. Dada la funcionalidad de Gesimed y puesto que esta la que marca la categorización de un sistema en alguno de los tipos de modelos que ofrece Cloud Computing, ***se ha considerado que el modelo que le aplica es el de SaaS***, la justificación a esta decisión se puede encontrar en el siguiente capítulo. Por otro lado vamos a ***hacer uso de una infraestructura como servicio*** para conseguir las características que rigen un modelo SaaS, ***se realizará un estudio de costes (sección 5.4) y del ROI (sección 5.5) frente a diseñar una infraestructura con un modelo tradicional.***

5.3 Posicionar Gesimed en un modelo SaaS

Cuando en alguna ocasión nos planteamos una migración, la adopción de un nuevo modelo, en definitiva, un cambio, la pregunta que debe dar sentido a la motivación de esa iniciativa es ¿Por qué?. Llegados a este punto y a pesar de haber descrito características muy beneficiosas del Cloud Computing, la pregunta razonable de este punto es ¿Por qué adoptar Cloud Computing?, ¿Por qué posicionar Gesimed en otro modelo SaaS?.

Este apartado no solo pretende explicar cómo se ha realizado la transición hacia un nuevo modelo, si no también justificar las razones que nos han llevado a hacerlo. Parece muy obvio, pero lo cierto es que, Cloud Computing ofrece tantas alternativas y oportunidades, que el abanico de casos de estudios posibles se multiplican, y las razones, objetivos u enfoques de cada uno de ellos puede ser muy distintos.

Siendo concisos, los motivos que nos han llevado a tomar la decisión de migrar nuestro sistema de información hacia un modelo SaaS son:

- ***Motivos de integración***

Cada vez es más típico encontrar a usuarios que acuden a portales de contenidos web donde se les ofrece adquirir aplicaciones. Dichos portales suelen tener la característica de ser agregadores de servicios o ISV's, es muy común que los ISV's deban implementar un contrato de servicios o interfaz de servicios web previamente, generalmente un WSDL.

Nuestro sistema está basado en SOA. Gracias a que partimos de un sistema orientado a servicios la integración con agregadores de servicios, marketplaces, apps-stores, etc., podría resultar muy sencilla y potenciaría comercialmente la aplicación.

- **Motivos Comerciales**

El modelo de Software as a Service, tiene como característica el pago por uso, mediante un modelo de facturación basado en licencias o por tiempo de uso, en nuestro caso podríamos incluso definir por intercambio en megas del tamaño de las imágenes médicas (mayor calidad, mayor precio). Nuestro sistema suele llegar con mayor frecuencia al sector de la investigación, durante periodos muy concretos de tiempo, como es la duración de un proceso de investigación o lo que dure una asignatura universitaria.

Debemos definir detalladamente un modelo de suscripción concreto, cuota de alta, cuota mensual, promociones, etc. Por otro lado el proceso de baja del servicio debe ser dinámico y flexible, “a golpe de ratón”. Cualquier tipo de suscripción que obliguemos al usuario más allá del tiempo que lo va a usar puede resultar que es dinero desperdiciado o una molestia para realizar la baja del servicio. A continuación se detalla formalmente la lista de procesos necesarias para realizar una migración hacia SaaS.

a) Asegurarse de que se entiende porque queremos implementar SaaS

En la introducción de este capítulo hemos dado respuesta a esta cuestión que se considera debe ser la primera. Es muy importante conocer nuestros actuales procesos y funcionalidades, analizar hacia donde queremos llegar, detectar los riesgos y establecer la hoja de ruta del proyecto.

b) Solicitar un SLA antes de firmar ningún contrato

Tanto desde el punto de vista del cliente como del proveedor, deberemos confeccionar un SLA, con todo el tipo de soporte que ofreceremos, tiempos de resolución de incidencias, etc. Pero también se lo solicitaremos a nuestros proveedores de servicios, en nuestro caso servicios de infraestructura (AWS).

c) **Planificar un equipo de soporte TI**

En ocasiones surgirán incidencias de distinto tipo y gravedad, nuestro SLA tendrá una serie de características que definen la forma en la que se responderá a dichas incidencias. Será necesario contar con un equipo técnico y de soporte para dar respuesta a esos requisitos bajo los cuales nos comprometemos con el cliente.

d) **Definir un modelo de negocio para las aplicaciones**

Este es uno de los apartados más importantes, consiste en definir bajo qué modelo el usuario puede hacerse con nuestro servicio, es decir, las opciones que se le van a ofrecer, puede ser bajo licencias de uso, individuales o por paquetes. Es muy común establecer una política de promociones, del tipo trial, bienvenida, por tramos, etc. A continuación se define nuestro modelo de negocio.

Puesto que nuestro sistema está muy ligado al ámbito investigador, es muy probable que esté motivado a usarse en grupos (grupo universitario, grupo médico), por ello vamos a ofrecer nuestra aplicación SaaS mediante ***paquetes de diez usuarios, la idea es que quién las compré (profesor, jefe de departamento) asigne cada una de las diez licencias diez personas.***

Para compensar la idea de que alguien quiera hacer uso individual, hemos definido una ***promoción de degustación o trial de dos meses de duración y una sola licencia***, bajo la cual estará libre de pagar nada. Cuando la promoción de degustación finalice, el usuario debe comprar la aplicación si no quiere perder todos sus datos.

Para todos los usuarios que quieran disfrutar de la aplicación en modo de pago, les proporcionamos una ***promoción de bienvenida de un mes de duración***, esta se aplicará independientemente de haber disfrutado de la de degustación, ***esta promoción se obtiene por valor de diez licencias y cuando caduque se empezarán a realizar los cobros mensuales.***

En cuanto al número de licencias que se compren fuera de las promociones anteriormente

mencionadas, se han establecido una política de precios por tramos como muestra la figura a continuación.

Tramo por licencias	Precio de alta	Precio mensual
[10 - 30]	10,90 euros	20,50 euros
[40 - 60]	7,50 euros	17,50 euros
[70 - +]	5,50 euros	15,50 euros

Tabla 3 Precios GesimedSaaS

La idea es que estos precios no sean nuestra única fuente de ingresos, se estudiará la opción de introducir publicidad segmentada por tipo de usuario en las versiones de degustación y bienvenida, así como la integración no solo con portales de aplicaciones SaaS si no la posibilidad de colaborar con portales de centros de investigación. Por último se muestra un prototipo de formulario, en el proceso de compra de la aplicación, cada MarketPlace lo implementará a su manera pero recomendamos seguir este modelo para mayor facilidad de integración.

GESIMED SAAS

Tipo de Servicio a contratar:

Paquete degustación (1 usuario y 1 proyecto)

Paquete de 10 usuarios (primer mes GRATIS)

Número de paquetes de 10 usuarios: 1

Datos de contacto

Nombre de la persona: José Manuel

Dirección de correo: jmanuel.arevalo@gmail.com

Número de teléfono: 606698576

Detalles de Usuario

Administrador: Seleccionar...

Refrescar carrito Volver Finalizar Compra

Ilustración 14 Modelo de negocio en formulario web

e) Desarrollo de las aplicaciones basado en WEB 2.0

La web 2.0 no es una tecnología actual pero dado que la mayoría de las aplicaciones basadas en SaaS son web, se hace necesario facilitarle al cliente una mejor experiencia como usuario. Nos permitirá introducir notificaciones multicanal (Email, SMS, RSS), llevar un control de actividad con herramientas como Google Analytics, etc. Aunque el aspecto más importante en este contexto es la implementación basada en mashups. Estos pueden ser fácilmente implementables mediante algún framework basado en Ajax (Dojo, Mootools) y servicios web de tipo Rest o SOAP, el framework realiza peticiones asíncronas con formato Java Script Object Notation (JSON) a los servicios, recoge la información y la dibuja en la página, normalmente en tiempo de carga de la página [16]. Así se podrán dibujar diferentes módulos independientes y funcionales, un claro ejemplo es iGoogle. ***En un MarketPlace*** de ejemplo, ***podríamos tener las aplicaciones como Mashups, justo antes de iniciar el proceso de compra o para añadir las al carrito como muestra la imagen.***



Ilustración 15 mashup GesimedSaaS en marketplace

f) Arquitectura de integración con un marketplace

Los MarketPlaces a modo de ecosistemas SaaS, son una de las motivaciones por las cuales los ISV migran a un modelo SaaS. Los clientes potenciales acudirán a estos portales de autoservicio de aplicaciones y software como servicio para adquirir los distintos servicios que se les ofrece. Si queremos potenciar las oportunidades de nuestro sistema y llegar al máximo número de gente posible nos integraremos con un Marketplace. Sin duda tenemos la ventaja de que partimos de un sistema basado en SOA, aunque en este proceso se detectan algunas tareas que pueden causar mayor o menor impacto en función de la arquitectura de partida.

Las tareas detectadas necesarias en este proceso son:

- ***Implementación del contrato de integración del Marketplace***

El Marketplace contendrá una interfaz de integración estandarizada y definida para poder entender el modelo de negocio de cada aplicación, este será diferente en cada caso y por ello una de las maneras más óptimas de integrar aplicaciones con un modelo de negocio esta, así podrán conocer los precios, si se ofrecen licencias unitarias o por paquetes, si tiene promociones y sus características, etc.

- ***Proporcionar nuestras interfaces al Marketplace***

Esta tarea va a resultar más o menos traumática en función de nuestro diseño y arquitectura de partida. Como mínimo debemos proporcionar al MarketPlace la interfaz de SSO ya que deberá hacer login a los usuarios contra nuestro sistema, pero también puede solicitarnos servicios relativos a la gestión de usuarios (CRM) o facturación (Billing). Cabe destacar que partimos de interfaces WSDL debido a nuestro diseño SOA y por lo tanto esta tarea no debe causar impacto en nuestra implementación.

g) Definir el RoadMap de las aplicaciones

GesimedWS pasará a ser un ISVs basado en SaaS, como tal cualquier actualización, parche o mejora debe ser transparente para el usuario. Por ello es se considera muy útil para causar buena imagen al usuario. Si en cortos periodos de tiempo el cliente ve mejoras, notará que la aplicación no está abandonada y por tanto el miedo a perder los datos algún día será menor.

Todos los procesos analizados anteriormente, se consideran una lista no cerrada para cada caso de estudio, pero es lo suficientemente genérica como para tomarla en cualquier desarrollo basado en SaaS. Dado que este proyecto es de carácter investigador y algunos aspectos comentados pueden quedar fuera del alcance se muestra una tabla con las actividades que se llevaron a cabo.

Proceso	Estado	Explicación
<i>a) ¿Por qué SaaS?</i>	✓	Definido en 5.3
<i>b) Solicitar SLA</i>	✗	Fuera del alcance del TFM
<i>c) Planificar equipo TI</i>	✗	Fuera del alcance del TFM
<i>d) Definir modelo de negocio</i>	✓	Definido en 5.3
<i>e) Web 2.0</i>	✓	Definido en PFC [ref. 11]
<i>f) Integración marketplace</i>	✓	Definido en 5.5
<i>g) RoadMap de aplicaciones</i>	✗	Fuera del alcance del TFM

Tabla 4 Proceso de adopción SaaS

5.4 Integración sobre IaaS

Una vez hemos conseguido posicionar GesimedSaaS, vamos a hacer uso de una infraestructura como servicio, hemos escogido Amazon Web Services como proveedor de dicho servicio por varios motivos.

- Ofrece servicios para distintas plataformas (Linux y Windows).
- Ofrece servicios para una variedad de lenguajes de programación (java, .net, php..etc.).
- Tiene una comunidad activa de desarrolladores y ofrece gran documentación.
- Sus servicios tienen una gran reputación y llevan años en el mercado.

El motivo principal de adoptar IaaS es poder atender a la fluctuante demanda de peticiones de servicio a GesimedSaaS, para ello se considera *necesario un gran esfuerzo en conseguir una alta escalabilidad con unos costes de gestión prácticamente planos. Gracias a obtener todos los recursos hardware como un servicio, podremos olvidarnos de todos los procesos de mantenimiento de infraestructura* y centrarnos en la tarea que nos ocupa, evolucionar GesimedSaaS. Los servicios y características computacionales que hemos usado del proveedor de servicios de infraestructura son:

- ***EC2 (Elastic Cloud)***¹:

Este servicio nos proporcionará elasticidad en nuestra infraestructura gracias a las máquinas virtuales que nos proporciona, también llamadas Amazon Machine Images (AMIs). El ciclo de vida de cada una de las AMIs es el siguiente.

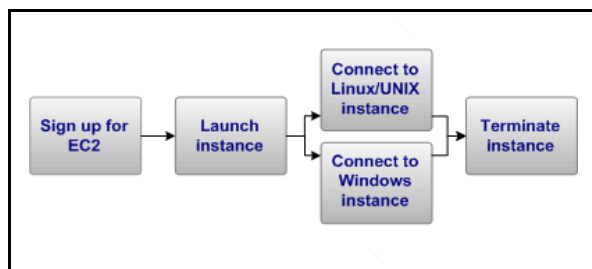


Ilustración 16 Ciclo de vida Amazon Machine Image [9]

Entre los distintos tipos de instancias hemos escogido el modelo intermedio que ofrece Amazon pero que se adapta a nuestras necesidades, las características son las siguientes:

¹ <http://s3.amazonaws.com/ec2-downloads/ec2.wSDL>

Instancia	windows	linux	Ram	#cpu	almacenamiento	Rendimiento E/S	Plataforma	
<i>Pequeña</i>	\$0.064 hora	\$0.042	1.7gb	1	160gb	<i>Moderado</i>	32bits	✘
<i>Grande</i>	\$0.257 hora	\$0.155	7.5gb	4	850 gb	<i>Alto</i>	64 bits	✔
<i>X-Grande</i>	\$0.534 hora	\$0.307	15gb	8	1690gb	<i>Alto</i>	64bits	✘

Tabla 5 Instancias estándar en Amazon Web Services [9]

Existen máquinas de propósito específico pero para nuestro caso y la demanda que esperamos se ajusta mejor el modelo estándar grande.

En cuanto al SLA, es importante analizar el que nos proporciona Amazon, las características son las siguientes [9]:

- “Año de servicio” son los 365 días anteriores a partir de la fecha de una reclamación de SLA.
- El “Porcentaje de tiempo de actividad anual” se calcula restando a 100% el porcentaje de periodos de 5 minutos durante el Año de servicio en el que Amazon EC2 se encontró en el estado de “Región no disponible”. Si ha estado utilizando Amazon EC2 durante un periodo de tiempo inferior a 365 días, su Año de servicio seguirán siendo los 365 días anteriores, pero se considerará que los días anteriores a su uso del servicio han tenido una Disponibilidad de región del 100%. Todo tiempo de inactividad que tenga lugar antes de una reclamación de Crédito de servicio con éxito no podrá utilizarse para futuras reclamaciones. Las mediciones de Porcentaje de tiempo de actividad anual excluyen los tiempos de inactividad derivados de forma directa o indirecta de cualquier Exclusión de SLA de Amazon EC2 SLA (definido a continuación).
- “Región no disponible” y “No disponibilidad de la región” significan que más de una Zona de disponibilidad en la que está ejecutando una instancia, dentro de la misma

Región, está “No disponible” para usted.

- “No disponible” significa que todas sus instancias en ejecución no tienen conectividad externa durante un periodo de cinco minutos, y que no podrá ejecutar instancias de sustitución.
- Un “Crédito de servicio” es un crédito en dólares, calculado tal y como se establece a continuación, que podríamos abonar a una cuenta Amazon EC2 que cumpliera determinadas condiciones.

Si el Porcentaje de tiempo de actividad anual de un cliente de Amazon cae por debajo del 99,95% durante el año de servicio, dicho cliente optará a recibir un crédito de servicio equivalente al 10% de su factura (excluyendo los pagos únicos realizados por Instancias reservadas) durante el Periodo de crédito apto.

- ***S3 (Simple Storage)***²:

Se ha creído conveniente hacer uso de este servicio de almacenamiento para alojar las distintas imágenes médicas de nuestro sistema, son imágenes muy pesadas y este servicio nos proporcionará escalabilidad en este aspecto.

En cuanto a los precios, Amazon cuenta con las siguientes tarifas:

Tamaño	Almacenamiento	Redundancia
Primer 1 TB	\$0.140 /GB	\$0.0950 /GB
Siguiente 49 TB/Mes	\$0.125 /GB	\$0.083 /GB
Siguiente 49 TB/Mes	\$0.110 /GB	\$0.073 /GB
Siguiente 49 TB/Mes	\$0.0950 /GB	\$0.063 /GB
Siguiente 49 TB/Mes	\$0.080 /GB	\$0.053 /GB
Siguiente 49 TB/Mes	\$0.055 /GB	\$0.037 /GB

Tabla 6 Precios almacenamiento en S3 [9]

En principio vamos a hacer uso ilimitado del almacenamiento ya que cuánto más imágenes debemos cargar, mayor es el uso que hacen los usuarios de nuestros servicio y

² <http://s3.amazonaws.com/s3-downloads/s3.wSDL>

obtendremos un alto retorno de la inversión. A continuación se esquematiza las instancias que levantaremos de todos los servicios para GesimedSaaS.

Por último cabe destacar algunos cambios importantes en el Back-End de nuestro sistema de partida. Como se ha explicado y se ha podido observar, el Back-End aglutinaba toda la funcionalidad en un mismo módulo, si queremos integrarnos con otros sistemas y adoptar una IaaS, es necesario modularizar las funcionalidades de los servicios para desacoplarlos aún más, esto proporcionará una gran flexibilidad de integración ya que debemos tener en cuenta que nuestro sistema de partida es SOA y eso nos dará una gran ventaja.

Las imágenes han sido sacadas de la base de datos y se han introducido en el servicio de almacenamiento para ganar en escalabilidad. Los módulos en los que se ha dividido son:

- **Billing:** Sistema de facturación por uso de la aplicación
- **SSO:** Protocolo de autenticación al sistema
- **CRM:** Funcionalidad de gestión de usuarios
- **Images Repository:** Almacenamiento de las imágenes médicas
- **Backend:** Procesamiento de imágenes.

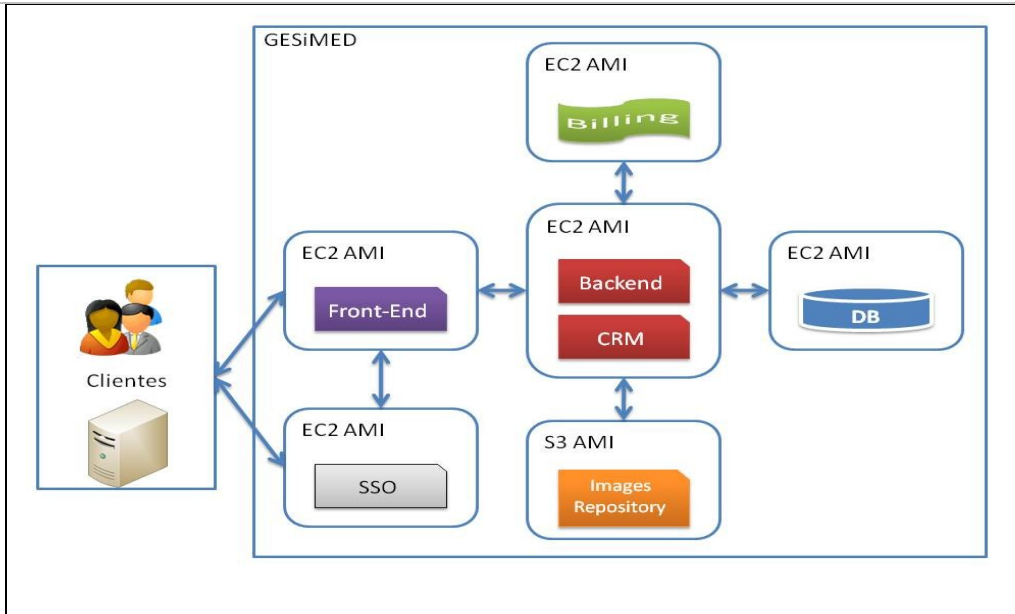


Ilustración 17 Arquitectura integración IaaS

Por último, en la tabla 7 detallaremos el coste de cada una de estas instancias mediante la calculadora que nos proporciona Amazon [9]. En la última sección de este capítulo se definirá un análisis entre esta solución basada en IaaS frente a diseñar nuestra propia solución para obtener un entorno dedicado *InHouse*.

Módulo	Tipo Instancia	S.O	Coste
<i>Back-End y CRM</i>	Ec2-grande	Windows Server R2	\$352,86/mes
<i>Billing</i>	Ec2-grande	Linux	\$281,33/mes
<i>SSO</i>	Ec2-grande	Linux	\$281,33/mes
<i>Front-End</i>	Ec2-grande	Linux	\$281,33/mes
<i>BD</i>	Ec2-grande	Win +SQLserver	\$794,96/mes
<i>Images repository</i>	S3	Linux	\$238/mes
TOTAL			\$2.229,81/mes

Tabla 7 Coste del caso de estudio

Cabe destacar que algunos precios pueden verse incrementados con respecto a las tablas anteriores, pero debemos tener en cuenta por ejemplo el caso de la base de datos, la cual viene con la licencia y la instalación del SQL-Server de Microsoft. También hemos añadido dos ip-elásticas para compra de nuestro dominio en internet, así como detalles en el ancho de banda de entrada y salida con el datacenter situado en Irlanda de Amazon.

5.5 Arquitectura global del sistema e implicaciones de la integración

Como resultados de las secciones anteriores, ya disponemos de un posicionamiento claro dentro de Cloud Computing, de una definición clara de lo que va a ser nuestro modelo de negocio y de una fuerte infraestructura que nos proporcionará elasticidad y escalabilidad a nuestro sistema. Pero se han detectado algunos obstáculos en la integración del sistema con la IaaS y con el MarketPlace.

Obstáculos con la integración del MarketPlace

Como se ha comentado en los primeros capítulos, los proveedores de servicios basados en Cloud Computing, bien sea, SaaS, PaaS, IaaS, necesitan estandarizar su oferta y sus interfaces, de lo contrario no podrían responder a la demanda del servicio con las bondades que ofrece la nube. Por ello debemos asumir que cualquier ISV que quiera integrarse con un

MarketPlace debe hacer un desarrollo adicional que consiste en implementar su contrato de integración (WSDL). Gracias a nuestro diseño de partida SOA y posterior modularización del sistema este impacto es menor de lo que podría ser con un sistema tradicional. Esta implementación se requiere para poder transmitir nuestro modelo de negocio al Marketplace, debemos tener en cuenta que cada ISV tendrá su propio modelo de negocio. Por ello el Marketplace solicita a través de una interfaz única y estandarizada el modelo de negocio concreto de cada aplicación en el momento que el usuario desee comprarla.

El último obstáculo encontrado en esta parte del proceso de integración es la de proporcionar las interfaces de autenticación, debemos tener en cuenta que cuando un usuario quiera comprar la aplicación y usarla, el Marketplace debe autenticar al usuario contra nuestro sistema. Pero este obstáculo no ha sido tal, ya que partíamos de un sistema basado en SOA, con interfaces claramente definidas, tan solo es necesario proporcionar los endpoints al equipo de desarrollo del Marketplace, por otro lado si fuera necesario podemos proporcionar más interfaces como Billing y CRM, por lo comentado anteriormente.

Obstáculos con la integración la IaaS

El principal problema encontrado en la integración de la IaaS es el de adaptación a sus interfaces, esto se ve muy claro con la migración de las imágenes desde la base de datos al servicio de almacenamiento S3. Ahora debemos acceder a ellas mediante servicios REST y ello ha causado impacto en el Front-End ya que la interfaz de comunicación es totalmente diferente.

Este hecho es bastante sintomático de lo que es Cloud Computing, ya que si hubiera sido la IaaS la que se hubiera tenido que adaptar a nosotros, no estaríamos realmente ante un servicio en la nube, no sé si estaríamos ante algo mejor o peor, pero no sería Cloud Computing. Debemos tener en cuenta que en este caso Amazon Web Services tiene todo un reto por delante debido al tipo de servicio que ofrece y esto no sería posible si no estandariza su oferta e interfaces tal y como pasaba con el Marketplace. A continuación se muestra la arquitectura global del sistema.

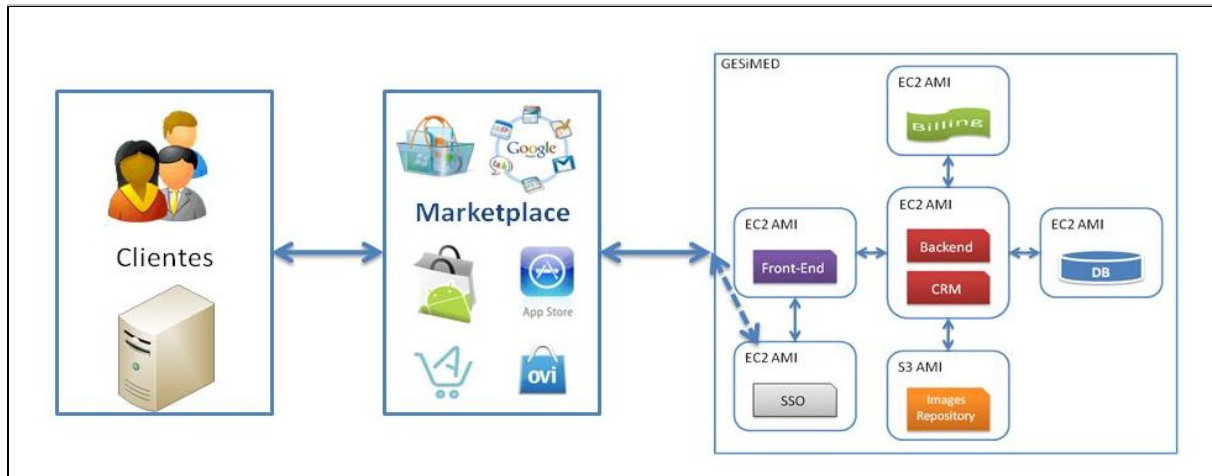


Ilustración 18 Arquitectura final integración Marketplace

Por último se ha realizado un análisis comparativo de los costes de toda la solución llevada a cabo, tanto la adopción de un modelo SaaS como la integración con una IaaS, frente a los costes que supondría no haber escogido Cloud Computing y haber optado por una solución dedicada in-house.

	Solución interna (inicial)	Solución Cloud (inicial)	Solución interna (mensualidad)	Solución Cloud (mensualidad)
<i>Racks</i>	\$3000	\$0	\$0	\$0
<i>Switches</i>	\$2000	\$0	\$0	\$0
<i>Servers</i>	\$33.000	\$0	\$0	\$2229,81
<i>Firewalls</i>	\$1500	\$0	\$0	\$0
<i>Gestión sw + mantenimiento</i>	\$1000	\$1000	\$100	\$2000
<i>24/7 Soporte</i>	\$0	\$0	\$0	\$1700
<i>Degradación</i>	\$0	\$0	\$100	\$0
<i>Consumo energético</i>	\$0	\$0	\$200	\$0
TOTALS	\$40.500	\$1000	\$400	\$5229,81

Tabla 8 Análisis de costes Cloud Computing frente solución in-house

De esta tabla se pueden sacar varias conclusiones. La primera está claramente que se puede observar el alto coste inicial de los recursos hardware en mi infraestructura interna frente a los costes planos de una IaaS y un coste mensual de ese gasto energético que provocan mis instalaciones. Sin embargo los riesgos serán mucho menores, debido a esa alta inversión en firewalls y seguridad que no proporciona una IaaS pública.

La segunda gran conclusión de esta tabla, es el alto coste que ha ocasionado migrar nuestro modelo de negocio a SaaS, ya que ahora debemos afrontar soporte y mantenimiento, ya que estamos ofreciendo un servicio. Sin embargo en un modelo de software tradicional esto no tiene por qué ser así. Cabe destacar comparando inversiones, se tardaría unos ocho años en llegar al gasto que tiene una solución interna, sin embargo esta, estaría diseñada y pensada totalmente por el cliente, no como en la IaaS en la que el usuario más que diseñar su solución, escoge entre lo que le ofrecen.

Capítulo 6:

Conclusiones y trabajos futuros

A continuación se entrará a describir las conclusiones obtenidas del actual trabajo, objetivos conseguidos (o no conseguidos), así como ventajas que se han observado y obstáculos encontrados. Por último se realizará una serie de propuestas de proyectos como posibles trabajos futuros.

6.1 Conclusiones

Este proyecto de fin de master proponía realizar un estudio detallado acerca de la situación real del paradigma de Cloud Computing tanto desde un punto de vista tecnológico, para lo que se ha incluido un estudio de sus principios centrados en las configuraciones arquitectónicas más habituales; como desde un punto de vista de negocio, incluyendo referencias y comentarios acerca de las plataformas de implementación ofrecidas por algunas de las compañías que actualmente más están apostando por este paradigma.

Por otro lado, se estableció como objetivo complementario, la implementación y despliegue de un supuesto práctico (Gesimed) sobre una arquitectura de Cloud real. Para ello, se ha posicionado Gesimed en un modelo SaaS, con el fin de potenciar sus posibilidades de negocio. A tal fin, se ha realizado un análisis de que procesos y tareas hacen falta en cualquier integración en la nube, lo que nos ha resultado muy útil como hoja de ruta en nuestro caso de estudio.

Además, con la realización de este proyecto hemos obtenido una infraestructura ***con unos costes de gestión y mantenimiento que son prácticamente planos y que hemos aumentado drásticamente la escalabilidad de nuestro sistema***. Sin embargo, en cuanto a la integración para integrar Gesimed en ***SaaS, nos ha introducido una carga de trabajo*** que antes no teníamos, ***toda la parte de operaciones*** que nos ahorrábamos con la IaaS, ahora como somos el proveedor del servicio recae sobre nosotros.

Por el contrario hemos tenido que pagar un precio muy caro, ***las interfaces de nuestro sistema actual han debido ser adaptadas a la IaaS***, de no haber sido así, no habiéramos estado ante una IaaS real, si la IaaS es la que se tiene que adaptar a tus sistemas entonces no estamos hablando de un entorno basado en Cloud computing. ***Gracias a que nuestro sistema de partida estaba basado en SOA el esfuerzo de esta integración ha sido mínimo***.

6.2 Trabajos futuros

Se ha conseguido un nuevo hito en este proyecto que se inició con el proyecto fin de carrera cuya evolución ha dado lugar a esta tesis fin de master. Pero estudiar Cloud Computing no ha hecho si no dar más ideas y oportunidades para seguir aumentando nuestros conocimientos y explorar nuevas vías de desarrollo.

Esta sección propone una serie de proyectos para quién le pudiera gustar proponerlo a sus alumnos como posibles proyectos fin de carrera o investigación interna.

- **Migración de herramientas de modelado hacia una aproximación PaaS.** La primera propuesta es poder aprovechar todas las características de PaaS, migrando herramientas desarrolladas internamente en el Kybele, basadas MDA como por ejemplo Archimedes [13] hacia un modelo de PaaS.
- **Desarrollo de un sistema de ficheros distribuido en la nube (KybeleBox).** La idea es poder simular una herramienta como DropBox, teniendo claros los conceptos vistos en este trabajo y definiendo un alcance del proyecto se puede desarrollar este proyecto con frameworks de desarrollo de sistemas de ficheros como Dokan (<http://dokan-dev.net/en/>) o Hadoop (<http://hadoop.apache.org/>).

Referencias

- [1] *Anthony T. Velte, Toby J. Velte, Cloud Computing a practical approach, 2009.*
- [2] *Bob Lozano, Executive's guide to Cloud Computing, 2010*
- [3] *Dr Mark I Williams, A quick start guide to Cloud Computing, moving your business into the cloud 2010*
- [4] *Barrie Sosinsky, Cloud Computing bible, 2011.*
- [5] *Cloud Application Architectures: Building Applications and Infrastructure in the Cloud (Theory in Practice (O'Reilly)) by George Reese (Paperback - Apr 10, 2009)*
- [6] *Programming Amazon Web Services: S3, EC2, SQS, FPS, and SimpleDB by James Murty (Paperback - Apr 1, 2008)*
- [7] *Cloud Computing SaaS And Web Applications Specialist Level Complete Certification Kit - Software As A Service Study Guide Book And Online Course - Second Edition by Ivanka Menken and Gerard Blokdijk (Paperback - Sep 7, 2010)*
- [8] *Demystifying the Virtual Desktop: Starting with Desktop Virtualization by Michael Fox (Paperback - Oct 30, 2010)*
- [9] *Amazon Web Services, <http://aws.amazon.com/documentation/>. último acceso 20/04/2011*
- [10] *Thomas Erl. SOA Concepts, technology and design, 2008*
- [11] *José Manuel Arévalo, Migración de un sistema de información basada en SOA con un aproximación basada en MDA.*
- [12] *Marcos Lopez Sanz, Archimedes: A Service Oriented Framework for Model Driven Development of Software, 2011*
- [13] *Thomas Earl, SOA with .NET and Windows Azure, 2009*
- [14] *David Lithium Cloud Computing and SOA Convergence in Your Enterprise: A Step-by-Step Guide, 2008*
- [15] *Jefrrey Hanson, Mashups Strategies for the modern Enterprise, 2007*
- [16] *Microsoft Windows Azure <http://msdn.microsoft.com/en-us/library/gg465396.aspx>, ultimo acceso 10/03/2011*
- [17] *Steve Weber, Sell on Amazon: A Guide to Amazon's Marketplace, Seller Central, and Fulfillment by Amazon Programs*
- [18] *ServidoresHP <http://h10010.www1.hp.com/wwpc/es/es/sm/WF06a/3709945-3709945-3710102-3722789-3722789-4186432.html> último acceso 21/05/11*
- [19] *Almacenamiento <http://h10010.www1.hp.com/wwpc/es/es/sm/WF02a/12169-3798502-3954626.html> último acceso 21/05/11*
- [20] *Resto de productos HP <http://www8.hp.com/es/es/home.html> último acceso 21/05/11*