



Universidad
de La Laguna

Escuela Superior de
Ingeniería y Tecnología
Sección de Ingeniería Informática

Trabajo de Fin de Grado

Computación en la nube. Oportunidades, retos y caso de estudio

*Cloud Computing. Opportunities, Challenges and Case
Study.*

Sara Báez García

La Laguna, 8 de julio de 2015

D. **Elena Sánchez Nielsen**, con N.I.F. 42.848.599-J profesor de la Escuela Superior de Ingeniería y Tecnología adscrito al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como tutor

C E R T I F I C A

Que la presente memoria titulada:

“Computación en la nube. Oportunidades, retos y caso de estudio.”

ha sido realizada bajo su dirección por D. **Sara Báez García**, con N.I.F. 78.647.701-Y.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 8 de julio de 2015

Agradecimientos

Me gustaría agradecer a mi tutora, Elena Sánchez Nielsen, por su labor y su ayuda en la gestión y durante el desarrollo de este trabajo.

A mis compañeros y amigos de clase, por compartir conmigo estos años de estudio y de crecimiento tanto intelectual como personal.

Pero sobre todo, gracias a Raúl y a mi familia, porque siempre están apoyándome aunque me caiga cien veces.

Licencia



© Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-SinObraDerivada 4.0 Internacional.

Resumen

El objetivo de este trabajo ha sido el estudio de las oportunidades y retos del cloud computing, así como aspectos como sus antecedentes, estado actual y legislación aplicable. Para esto se ha realizado un análisis de las soluciones actuales disponibles.

Además, se realizó un caso de estudio, que consistía en el desarrollo e implementación de una aplicación web. Este caso de estudio permitió analizar y estudiar de forma real la flexibilidad, la rapidez pero también las lagunas y otros muchos aspectos de este nuevo paradigma.

En los capítulos que componen este trabajo se detalla cada uno de los puntos que brevemente se acaban de nombrar. Asimismo, se incluye un apartado de conclusiones, posibles líneas de trabajo futuras y una sección de presupuesto.

Palabras clave: Computación en la nube, retos, oportunidades, caso de estudio, análisis, desarrollo, soluciones, web, aplicación.

Abstract

The main goal of this project has been the analysis and study of the opportunities and challenges of cloud computing, as well as its technological background, current state and applicable legislation. To do this, a research of the actual solutions has been made.

In addition, a case study has been made, too. Its purpose was the development and implementation of a web application. This case study allowed to study in a real way, not only the flexibility and the quickness but the lack and many other characteristics of this new paradigm.

In the chapters on this project it is detailed every briefly appointed characteristic and points. Furthermore, it is included a conclusions section, future works and a budget section.

Keywords: *Cloud computing, challenges, opportunities, case study, analysis, development, solutions, web, application.*

Índice general

1. Introducción	1
2. Estado del arte	2
2.1. Antecedentes, origen del <i>cloud computing</i>	2
2.2. Concepto de <i>cloud computing</i>	4
2.3. Tipos de <i>cloud</i>	6
2.3.1. Según el modelo de desarrollo	6
2.3.2. Según el modelo de servicio	8
2.4. Soluciones tecnológicas actuales	8
2.4.1. Amazon Web Services	9
2.4.2. Google Cloud Platform	10
2.4.3. Microsoft Azure	11
2.4.4. Docker	11
3. Ventajas y retos	13
3.1. Ventajas	13
3.1.1. Ventajas para las empresas	14
3.1.1.1. Ventajas estratégicas	14
3.1.1.2. Ventajas técnicas	14
3.1.1.3. Sostenibilidad	15
3.1.2. Ventajas para las administraciones públicas	15
3.1.3. Ventajas para innovación e investigación	16
3.1.4. Ventajas para los ciudadanos	16
3.2. Retos	17
3.2.1. Disponibilidad del servicio	18
3.2.2. Seguridad y privacidad de los datos	18
3.2.3. Falta de estandarización	19
3.2.4. Adaptabilidad a la nube	19
4. Estado actual	20
4.1. Situación actual en España	20
4.1.1. Dentro del sector servicios	21
4.1.2. Dentro de las administraciones públicas	21
4.2. Situación actual en el mundo	22

4.3.	Ley aplicable	22
4.3.1.	Ley Orgánica de Protección de Datos - LOPD	23
4.3.2.	Ley de Servicios de la Sociedad de Información y de Comercio Electrónico - LSSI	23
4.3.3.	Ley de Acceso Electrónico de los Ciudadanos a los Servicios Públicos - LAECSP	24
4.3.4.	Ley de Firma Electrónica	24
5.	Caso de estudio	25
5.1.	Especificaciones	25
5.2.	Planteamiento	26
5.2.1.	Entorno de desarrollo	26
5.2.1.1.	Google App Engine	27
5.2.2.	Herramientas a utilizar	27
5.2.2.1.	Python/Django	27
5.2.2.2.	MySQL	27
5.2.2.3.	Google App Engine SDK	28
5.2.2.4.	PyCharm	28
5.2.2.5.	Linux Mint Debian	28
6.	Desarrollo del caso de estudio	29
6.1.	Metodología	29
6.2.	Desarrollo	30
6.2.1.	Estructura general	30
6.2.2.	Perfiles de usuario	31
6.2.3.	Sitios y parámetros	32
6.2.4.	<i>Background</i>	33
6.2.5.	Pruebas y detección de errores	34
6.3.	Funcionalidades	35
6.4.	Problemas encontrados	36
6.4.1.	Versiones	36
6.4.2.	Base de datos	36
6.4.3.	<i>Third-part libraries</i>	37
6.4.4.	Ficheros estáticos	37
6.4.5.	Tareas en <i>background</i>	37
7.	Conclusiones y líneas futuras	38
7.1.	Conclusiones	38
7.2.	Trabajos futuros	39
7.2.1.	Interfaz	39
7.2.2.	Añadir parámetros	39
7.2.3.	Configuración del tiempo de actualización	39
7.2.4.	Estudio de nuevo entorno <i>cloud</i>	39

8. Summary and Conclusions	40
8.1. Conclusions	40
8.2. Future work	40
8.2.1. Interface	41
8.2.2. New parameters	41
8.2.3. Configurable update time	41
8.2.4. New cloud environment	41
9. Presupuesto	42
9.1. Coste por trabajador	42
9.2. Coste por uso del entorno	43
9.3. Coste total	43
Bibliografía	43

Índice de figuras

2.1. Evolución de las arquitecturas y la tecnología hasta el <i>cloud computing</i>	4
2.2. Sistema informático: <i>hardware</i> , <i>software</i> y usuarios.	5
2.3. Tipos de nube según su modelo de desarrollo: privada, híbrida, comunitaria y pública.	7
4.1. Soluciones <i>cloud</i> por sector.	22
6.1. Relación de los modelos de datos	31
6.2. Formulario de registro de nuevo usuario	32
6.3. Usuario logueado con un sitio definido	32
6.4. Creación de un nuevo sitio	33
6.5. Parámetros actualizados de un sitio	33
6.6. <i>Cron Jobs</i> activos en <i>Google Cloud Platform</i>	34

Índice de tablas

9.1. Coste por el trabajo de programación	42
9.2. Coste por el trabajo de análisis	42
9.3. Coste por el entorno cloud	43
9.4. Coste total	43

Capítulo 1

Introducción

El *cloud computing* o computación en la nube es un paradigma que ofrece tanto a usuarios individuales, como a empresas y como a entidades públicas, una nueva forma de prestación de los servicios de tratamiento de la información, como ya hemos dicho, es válido para cualquier tipo de usuario.

Las soluciones y servicios del *cloud computing* aportan nuevas ventajas que no se consiguen con los sistemas tradicionales de las tecnologías de la información. Sobre todo, podemos destacar el ahorro en costes y la facilidad para escalar los recursos de los que dispone la empresa.

El objetivo del presente trabajo de fin de grado es estudiar y analizar tanto las oportunidades como los retos que ofrece el paradigma del *cloud computing*. Así como desarrollar e implementar un servicio de aplicación en la nube, con el fin de evaluar la rapidez, la flexibilidad, la mejora de la gestión tecnológica, los riesgos asociados, y la legislación aplicable para los clientes que contraten los servicios de un entorno *cloud*.

Capítulo 2

Estado del arte

En este capítulo se introducen los aspectos necesarios para la comprensión del proyecto y sus objetivos. De esta forma, se describen varios términos, como computación en la nube y las tecnologías disponibles para su uso. Asimismo, se hace un recorrido desde los orígenes del *cloud computing* hasta las soluciones actuales.

2.1. Antecedentes, origen del *cloud computing*

Actualmente y desde los últimos años, nos encontramos ante un proceso de deslocalización e internacionalización de las grandes empresas. Su necesidad de cómputo y procesamiento de datos ha aumentado más rápido que la capacidad de cómputo de la que disponen los ordenadores personales. Esto ha hecho que las arquitecturas de cálculo hayan tenido que evolucionar hacia métodos de ejecución simultánea y colaborativa entre varios equipos informáticos.

Gracias a la creación de Internet, ha sido posible la aparición de muchos paradigmas cuya finalidad es ofrecer la tecnología como un servicio. Internet permite la descentralización, que equipos individuales puedan conectarse con otros que se encuentran en un lugar del mundo diferente. Esta forma de interconexión hace que podamos obtener una mayor cantidad de recursos computacionales y de almacenamiento de forma distribuida. Los nuevos paradigmas, tales como la *Web 2.0*, computación P2P, *data centers*, clúster, *grid*, computación ubicua, etc, pretenden explotar todo este potencial.

Sin embargo y para centrarnos más en el origen de lo que hoy conocemos como *cloud computing*, vamos a detenernos solo en las definiciones de dos paradigmas de computación bastante usados o estudiados, como son la computación en clúster y en *grid*.

Greg Pfister (gurú en *clusters*) define un clúster como un tipo de sistema paralelo que consiste en la interconexión de equipos individuales para actuar como uno solo [1]. Considera que un equipo individual es aquel que es independiente y

puede funcionar por sí mismo, es lo que se conoce como un nodo. Por lo tanto, un clúster es una configuración de un grupo de nodos, que aparecen en la red como una sola máquina y actúan como tal. Puede ser administrado como un solo sistema y además está diseñado para tolerar fallos en los componentes, de manera que si uno falla, el usuario no se percate de ello. Además, existen varios tipos de clúster; puede ser que todos los equipos sean similares en cuanto a *hardware* y *software*, por lo que estaremos hablando de clúster homogéneo; si difieren en rendimiento pero tienen similitudes, se trata de un clúster semi-homogéneo y en cambio, si tanto el *hardware* como el *software* es diferente, se conoce como clúster heterogéneo. Este último tipo es el más sencillo y económico.

Podemos considerar un *grid* como una especialización de un clúster, los centros de investigación y las universidades que disponían de estos sistemas clúster, comenzaron a ofrecer los servicios de cálculo y almacenamiento a terceros, a través de protocolos estándar, dando lugar a la que hoy en día se denomina computación *grid*. Por lo tanto, la computación *grid*, o también conocida como computación distribuida, consiste en un gran número de equipos organizados en clústeres y conectados mediante una red de comunicaciones. Está orientada al procesamiento en paralelo y al almacenamiento de grandes cantidades de información. Permite compartir una amplia variedad de recursos que están distribuidos geográficamente. Entre estos recursos se pueden encontrar supercomputadoras, sistemas de almacenamiento, etc.

A pesar de que este tipo de computación ofrece muchas ventajas, tales como la potencia, escalabilidad, integración de sistemas heterogéneos, etc, solo tuvieron éxito dentro del ámbito académico y de investigación. Esto fue debido a la complejidad para utilizar la infraestructura y los problemas de portabilidad entre *grids*.

El siguiente paso tras los clústeres y la computación en *grid*, fue la virtualización. Esta nueva tecnología consiste en la creación virtual de algún recurso, que puede ser un sistema operativo, un servidor, un dispositivo de almacenamiento, recursos de red, etc. Esto tiene muchas ventajas, debido a que es posible replicar un entorno sin necesidad de instalar y configurar todo el software que requieren las aplicaciones. La virtualización simula una plataforma de *hardware* autónoma y ejecuta el *software* como si este estuviera instalado.

Esta nueva tecnología permite distribuir la carga de trabajo de una manera mucho más sencilla que en la computación *grid*. Y también permite, como la anterior, la integración de entornos heterogéneos. Así es como surge el nuevo paradigma de computación, el *cloud computing*. Las plataformas de *cloud computing* tienen características de ambas tecnologías, los clústeres y los *grids*, pero añade sus propias ventajas.

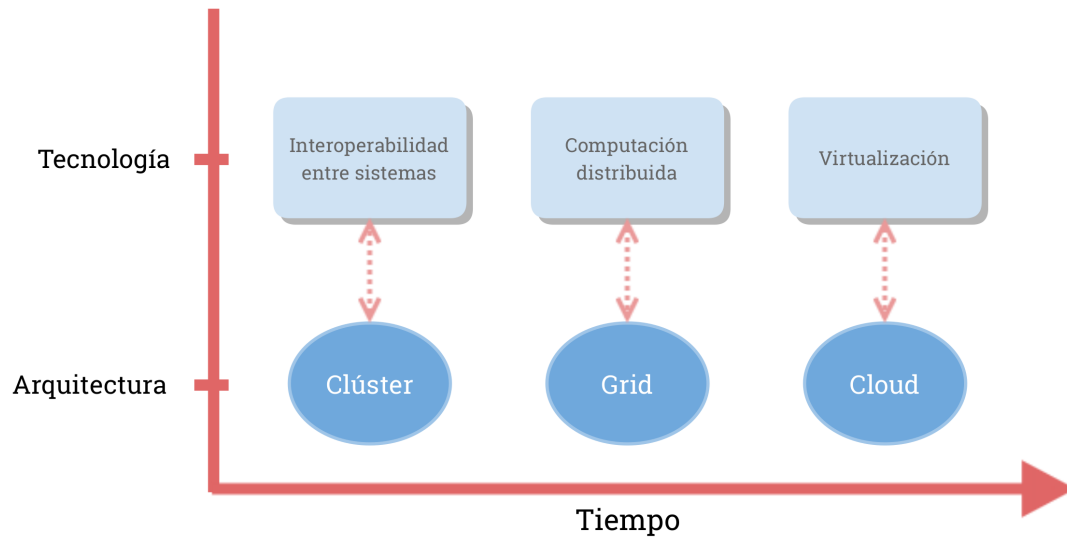


Figura 2.1: Evolución de las arquitecturas y la tecnología hasta el *cloud computing*.

2.2. Concepto de *cloud computing*

La computación en la nube también se conoce como: *cloud computing* o simplemente, *the cloud*, es un nuevo paradigma de computación, una nueva forma de prestación de servicios. Pero previamente, antes de definir la computación en la nube, debemos conocer algunos conceptos.

Un sistema informático es un conjunto de partes interrelacionadas que permiten almacenar y procesar información. Las partes que conforman un sistema informático son: el *hardware*, el *software* y los usuarios.

- El *hardware* proporciona los recursos computacionales del sistema.
- El *software*, que a su vez lo podemos dividir en el sistema operativo y en los programas de aplicación.
 - El sistema operativo se encarga de controlar y coordinar el uso del *hardware* por parte de las diferentes aplicaciones. Proporciona un entorno adecuado para que los programas puedan realizar un trabajo útil.

- Los programas de aplicación definen las diferentes formas en las que los recursos de un sistema son utilizados, de manera que sean capaces de resolver los problemas de los usuarios.
- Los usuarios son el conjunto tanto de los administradores, desarrolladores, etc, como de los usuarios finales.

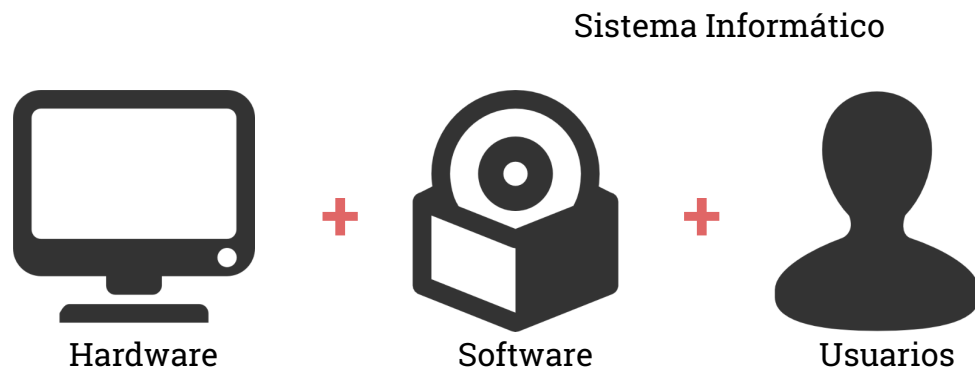


Figura 2.2: Sistema informático: *hardware*, *software* y usuarios.

Una vez definido el concepto de sistema informático, podemos pensar en la computación en la nube, como un sistema informático como servicio. Es un tipo de computación que se basa en Internet para ofrecer diferentes servicios. Se trata de un modelo de *pay-per-use* o bajo demanda, en el que el usuario solo paga por los servicios que necesita.

Según el NIST (*National Institute of Standards and Technology*)[2], el *cloud computing* se define como un modelo tecnológico que permite el acceso ubicuo, adaptado y bajo demanda en red a un conjunto compartido de recursos de computación configurables, tales como redes, servidores, equipos de almacenamiento, aplicaciones y servicios, que pueden ser rápidamente aprovisionados y liberados con un esfuerzo de gestión reducido o interacción mínima con el proveedor del servicio.

Una nube es un tipo de sistema paralelo y distribuido que consiste en la colección de equipos interconectados y virtualizados de tal forma que representan una cola máquina o un conjunto de recursos computacionales unificados. Las características son establecidas entre el proveedor de servicios y el consumidor.

Las principales características de la computación en la nube son las siguientes:

- Pago bajo demanda por el servicio, lo que hace posible que el usuario pueda ampliar o reducir los recursos que necesita de manera rápida y automática y sin necesidad de negociar con el proveedor de servicios. Esto conlleva a una reducción de costos, ya que únicamente se paga por los servicios utilizados.
- Accesibilidad a los recursos a través de la red, tanto desde mecanismos estándares como desde plataformas heterogéneas tales como dispositivos móviles, ordenadores portátiles, *tablets*, etc.
- Escalabilidad y elasticidad, los recursos pueden ser rápidamente liberados o adquiridos según la demanda, de esta forma se escala hacia adentro o hacia afuera de manera automática. Se puede decir que podemos disponer de recursos ilimitados.
- Medición del servicio. Con estos sistemas es posible controlar y optimizar automáticamente el uso de los recursos, como por ejemplo el ancho de banda, la cantidad de almacenamiento, etc.
- *Resource pooling*, es decir, que los recursos de los proveedores están compartidos de manera que varios consumidores puedan acceder a ellos. Este acceso se hará dependiendo de las asignaciones de cada consumidor según su demanda. El consumidor no conoce la localización física de los recursos que está utilizando, pero puede especificar una zona como por ejemplo un país, una ciudad o un centro de datos.

2.3. Tipos de *cloud*

Las soluciones actuales de computación en la nube se pueden clasificar en varios grupos, dependiendo siempre del parámetro que elijamos para realizar la clasificación.

2.3.1. Según el modelo de desarrollo

Una de las clasificaciones que se puede realizar se hace atendiendo a la privacidad. De esta manera, encontramos cuatro modelos de desarrollo para la computación en la nube: nube privada, pública, comunitaria e híbrida.

- Nube privada: una sola organización hace la gestión y la administración de sus servicios en la nube, esta organización puede ser el mismo proveedor de servicios, la empresa contratante o un tercero. En caso de que se trate de un tercero, este actuará bajo las necesidades de la organización.

Normalmente se elige este tipo de nube cuando la información es crítica, se necesita centralizar los recursos informáticos y además se quiere tener

flexibilidad a la hora de disponer de los mismos. Esto es posible ya que la solución adquirida está adaptada a las necesidades de la empresa contratante y se puede realizar el control y la supervisión de la seguridad y la protección de la información.

- Nube pública: en este tipo de nube, no se conoce la ubicación de la información. El proveedor de servicios ofrece sus recursos de forma gratuita a diferentes entidades. No se debe pensar que una nube pública es una nube insegura.

En este tipo de nube se dispone de plazos de tiempo reducidos para la disponibilidad del servicio. Además de que los costes por su uso, en caso de que los haya, son bastante bajos.

- Nube comunitaria (*community cloud*): los servicios se comparten en una comunidad cerrada de entidades que tienen los mismos objetivos, de manera que colaboran entre ellas. En este caso la nube es gestionada y administrada por una o más entidades de la comunidad.
- Nube híbrida: estas nubes pueden estar formadas por dos o más tipos de nubes de los que ya hemos hablado: pública, privada o comunitaria. Algunos servicios se ofrecen de manera pública, como por ejemplo las herramientas de desarrollo, y otros de manera privada, como es el caso, por ejemplo de la infraestructura.

Una entidad que opte por esta soluciones puede beneficiarse de las ventajas que el resto de nubes tenían por separado. Por ejemplo, se dispone de una gran flexibilidad a la hora de adquirir servicios, pero además es posible tener control sobre ellos.

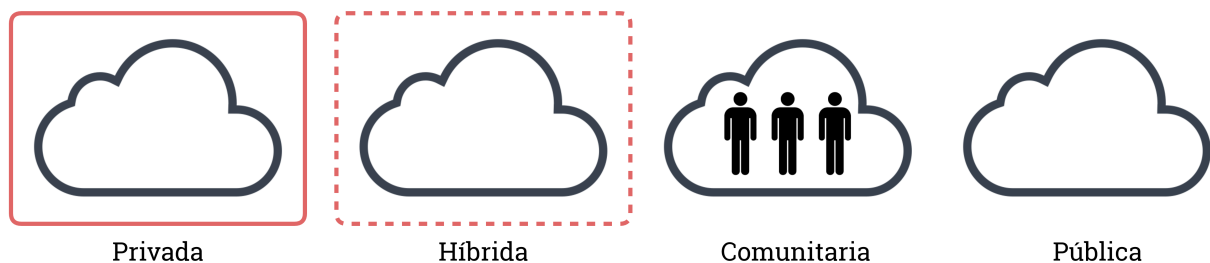


Figura 2.3: Tipos de nube según su modelo de desarrollo: privada, híbrida, comunitaria y pública.

2.3.2. Según el modelo de servicio

Si atendemos al modelo de servicio ofrecido, encontramos tres familias, que hoy en día son las que se conocen como *XaaS*, que significa *anything as a service*, o simplemente como *As a service*. A pesar de que hoy en día las soluciones que se ofrecen suelen ser mixtas, es decir, combinando varios modelos de servicio, vamos a realizar la siguiente clasificación para entender mejor los conceptos.

- *Software as a Service (SaaS)*: en este grupo lo que se ofrecen son productos finales como un servicio, de manera que la entidad pueda disponer de estas aplicaciones para el desarrollo de su propia actividad. Se puede acceder a las aplicaciones desde diferentes dispositivos y lo pueden hacer varios clientes a la vez.

Se provee al cliente con las licencias necesarias para el uso de un *software* como demanda. Además, puede hacerse de dos formas: el proveedor dispone de las aplicaciones en sus servidores y el cliente accede por el navegador, o se instala la aplicación en los sistemas del cliente y la licencia expira cuando acabe el periodo contratado.

- *Platform as a Service (PaaS)*: este modelo de servicio consiste en ofrecer al usuario herramientas con las que pueda desarrollar, hacer el testeo, despliegue, mantenimiento y *hosting* de sus propias aplicaciones informáticas, sin la necesidad de que tenga que instalarlas en sus equipos locales. Las grandes ventajas de este modelo son que el usuario no tiene que pagar las licencias de las herramientas y además está exento del mantenimiento y actualización de las mismas.
- *Infrastructure as a Service (IaaS)*: se caracteriza porque provee capacidad de almacenamiento, y recursos computacionales que el usuario utilizará para desarrollar su propio *software*. Es decir, se pone a disposición del cliente el uso de una infraestructura informática como servicio.

Se elige este método como alternativa para no tener que adquirir todos los servidores, el espacio de almacenamiento y los equipos de red necesarios para desarrollar la actividad del cliente.

2.4. Soluciones tecnológicas actuales

En apartados anteriores hemos hablado de la clasificación de los diferentes tipos de nube que existen hoy en día atendiendo a diferentes parámetros. Por lo tanto, las soluciones tecnológicas actuales van a pertenecer a una de estos tipos o, de forma más real, van a pertenecer a varios de ellos a la vez. Si hacemos un recorrido general por cada uno de los tipos de cloud según su modelo de servicio, encontramos algunas soluciones generales como las que se listan a continuación:

- *SaaS*:
 - Paquetes de software de oficina
 - Gestión de proyectos y portfolios
 - Mensajería
 - Gestión de contenidos
 - ...
- *PaaS*:
 - Integración de datos
 - Sistemas de gestión de bases de datos
 - Aplicaciones de seguridad
 - Portales de aplicaciones
 - ...
- *IaaS*:
 - Servicios de computación
 - Servicios de almacenamiento
 - Servicios de copia de seguridad

Para ser más específicos vamos a nombrar y explicar alguna de las soluciones actuales propuestas por grandes empresas, como son Amazon, Google y Microsoft. Estas empresas han sabido adaptarse al nuevo paradigma que es el cloud computing y cada una de ellas ofrece algunas soluciones cloud, tanto *PaaS*, como *IaaS* y *SaaS*.

2.4.1. Amazon Web Services

Amazon fue la empresa pionera en esta nueva tecnología, ofrece sus propios servidores a los clientes, para que estos puedan aprovechar los recursos de los que dispone la empresa. Así en el año 2006 surge Amazon Web Services. Las soluciones que ofrecen son tanto plataforma como infraestructura como servicio, las empresas que lo contratan pueden modificar la potencia informática, la cantidad de almacenamiento, etc. Dispone de una gran flexibilidad, ya que se puede utilizar cualquier plataforma o modelo de programación.

Tiene varios centros de datos repartidos por el mundo, de manera que los usuarios cercanos a cada unas de las regiones mantienen sus datos y recursos en esa zona en concreto.

Además de esto, dispone de servicios adicionales que se pueden incorporar a las aplicaciones existentes o simplemente utilizarse de forma independiente. Algunos como Amazon CloudFront, Amazon EC2, Amazon SimpleDB, etc. La mayoría de estos servicios no están expuestos directamente a los usuarios finales, ya que su fin es que sean utilizados para que los clientes puedan desarrollar *software*, que sí será el que esté pensado para ser utilizado por usuarios finales. A estos servicios se accede mediante HTTP haciendo uso de protocolos REST y SOAP y en todos ellos se paga por uso.

Amazon EC2 o Amazon Elastic Cloud Compute es una de las partes centrales de AWS. Con este servicio, los usuarios pueden pagar por equipos virtuales para alojar sus aplicaciones en vez de tener que comprar equipos dedicados. Esta plataforma se basa en el principio de la virtualización, de este modo, es posible utilizar diferentes sistemas operativos y personalizarlos. Además de todas las características básicas que ofrece el cloud computing, de las cuales hemos hablado anteriormente, Amazon EC2 dispone de herramientas de recuperación de datos y proporciona un gran aislamiento frente al resto de procesos que se realizan en sus máquinas.

Esta plataforma funciona con lo que se conocen como instancias y dependiendo de las características, rendimiento, capacidad de cómputo, etc, el usuario puede elegir entre todos los tipos de instancia existentes. Por ejemplo, hay instancia de uso general, optimizadas para informática, memoria o almacenamiento e instancias de GPU.

2.4.2. Google Cloud Platform

Google Cloud Platform es la plataforma de cloud computing que ofrece Google a sus clientes y se trata de un servicio tanto de *PaaS* como de *IaaS*. Esta misma infraestructura es la utilizada internamente por Google para algunos de sus productos, como Google Search y Youtube. En esta plataforma, los usuarios pueden desarrollar y alojar sus aplicaciones haciendo uso de un amplio rango de programas que permiten crear desde sencillos sitios web hasta complejas aplicaciones.

Esta plataforma es un conjunto de servicios modulares basados en la nube con múltiples herramientas de desarrollo. Encontramos servicios de *hosting* y computación, de almacenamiento en la nube, *big data* y APIs específicas, como por ejemplo de traducción y predicción. Cada uno de estos productos dispone de interfaz web, herramienta para línea de comandos y de una REST API. Los productos que ofrecen actualmente son los siguientes: App Engine y Compute Engine como plataforma e infraestructura como servicio respectivamente. Como servicio de almacenamiento disponen de Cloud Storage, Cloud Datastore y Cloud SQL. Además ofrecen *Big Data* y APIs de traducción y predicción como habíamos nombrado anteriormente.

La plataforma como servicio, o *PaaS* de Google, es conocida como Google App Engine y gracias a ella y a las herramientas de las que dispone, es posible crear un *SaaS* y alojarlo en esta plataforma. Tiene soporte para múltiples lenguajes de programación y *frameworks*. Además, es posible desarrollar de manera local la aplicación, gracias al SDK que tiene disponible cada lenguaje.

App Engine permite a los usuarios no tener que preocuparse por la administración, ni por la configuración, ni por el balanceo, etc, de sus servidores o bases de datos. Incluso es posible comparar dos versiones de la misma aplicación.

2.4.3. Microsoft Azure

Microsoft Azure es tanto la plataforma como la infraestructura como servicio que ofrece Microsoft a sus clientes, anteriormente era conocida como Windows Azure o como Azure Services Platform. Esta plataforma se encuentra en los centros de datos de la empresa, en concreto dispone de nodos en 24 países diferentes, repartidos por todo el mundo. Dispone tanto de alojamiento para aplicaciones, como de servicios propios e incluso comunicaciones seguras. Según lo describen en Microsoft, Windows Azure es una capa en la nube, que se encuentra funcionando sobre servidores Windows Server. Esta capa es la que se encarga de escalar los recursos y manejar la información de la aplicación web del usuario, de manera que le pueda asignar la memoria necesaria, entre otras cosas.

La plataforma también dispone de un sistema de copias de seguridad automático, para así proteger la información de usuario en caso de pérdidas. Las copias de seguridad se almacenan de forma cifrada y el usuario puede acceder a ellas para recuperarlas. En Windows Azure están soportados algunos lenguajes de programación y *frameworks*, así como bases de datos relacionales y no relacionales, *blobs* y colas de mensajes, entre otras muchas funcionalidades. Además, es posible desplegar máquinas virtuales tanto con Windows Server como con máquinas con distribuciones de Linux.

Al igual que las plataformas de las que hemos hablado anteriormente, Windows Azure está formada por una serie de componentes, que proporcionan: capacidad de cómputo, almacenamiento, bases de datos, servicios para las aplicaciones, mercado de aplicaciones, etc.

2.4.4. Docker

Docker significa contenedor y es una plataforma abierta para construir, empaquetar y lanzar aplicaciones distribuidas de forma automática. El concepto, como su nombre indica, es empaquetar las aplicaciones con todas sus dependencias dentro de un contenedor. Esta plataforma empezó como un *framework*, una

herramienta de alto nivel de LXC, pero ahora dispone de una librería propia, llamada *libcontainer*.

Es una herramienta que está pensada tanto para programadores como para administradores de sistemas, debido a que ofrece muchas ventajas, tales como que permite lanzar las aplicaciones tanto en entornos Linux, OS X y Windows, sin que importe el lenguaje utilizado, ya que crea una capa de abstracción del sistema operativo, eliminando los problemas derivados de dependencias y versiones.

Docker implementa una API de alto nivel, la diferencia principal de Docker frente a cualquier máquina virtual, es que no necesita de la inclusión de un sistema operativo, sino que se basa en las funcionalidades del *kernel* y aísla los recursos tales como la CPU, la memoria, los dispositivos de entrada/salida, la red, etc. Gracias al uso de los contenedores, es posible aislar recursos pueden ser aislados, restringir los servicios y además, se puede hacer que los procesos tengan una vista privada del sistema operativo. Los contenedores pueden compartir el mismo *kernel*, pero cada uno de ellos estará restringido a usar cierta cantidad de recursos.

La plataforma es fácilmente integrable con multitud de infraestructuras, tales como Amazon Web Services, Google Cloud Platform, Jenkins, Puppet, Vagrant, entre otras muchas.

Capítulo 3

Ventajas y retos

Como ya hemos adelantado en el capítulo anterior, la computación en la nube es una tecnología en auge desde los últimos años debido a la multitud de ventajas que presenta. Sin embargo, actualmente también se enfrenta a muchos retos que debemos conocer para poder hacer una valoración global de esta nueva tecnología.

En este capítulo se profundiza en las ventajas y oportunidades y en las desventajas y retos que presenta esta nueva tecnología, tanto para las empresas privadas, como para las empresas públicas y los ciudadanos.

3.1. Ventajas

Las ventajas de este nuevo paradigma son notables tanto para empresas privadas, como para organizaciones públicas y de investigación y para los ciudadanos, esto es gracias a su rapidez, flexibilidad, disponibilidad, etc. Es decir, el *cloud computing* ofrece un conjunto de ventajas de tipo económico, tecnológico, ambiental y social. De entre todas las ventajas, las más notables para los usuarios son el ahorro en costes y la facilidad para aumentar los recursos disponibles.

Los ahorros en costes son debidos a que es posible evitar los gastos tanto en *hardware*, como en *software*, soporte y seguridad. Por otro lado, la flexibilidad y la escalabilidad de los recursos se hace de una manera muy sencilla y en el momento que el cliente lo requiera, de forma que puede aumentar o disminuir los recursos que está utilizando en cualquier momento y además pagando solo por lo que usa. Otra de las ventajas más atrayentes es la capacidad de recuperación ante problemas, o desastres.

Podemos atrevernos a decir que el paradigma del *cloud computing*, gracias a todas las ventajas que ofrece frente a los métodos tradicionales, está haciendo que aumente la productividad de las empresas, se mejore en los servicios públicos, la calidad de vida y estemos avanzando hacia lo que se conoce como Sociedad de la Información y del conocimiento.

La Sociedad de la Información y del Conocimiento, es un término que hace referencia a la capacidad de utilizar la tecnología actual de forma que se pueda crear, obtener, o compartir cualquier información de manera rápida desde cualquier lugar y de la forma que se prefiera.

3.1.1. Ventajas para las empresas

Las principales ventajas competitivas que el cloud computing ofrece a las empresas privadas, las podemos clasificar en tres grupos: ventajas estratégicas, ventajas técnicas y, como ya hemos dicho, ventajas económicas aunque también existen ventajas para la sostenibilidad.

3.1.1.1. Ventajas estratégicas

La ventaja que este paradigma ofrece a las empresas desde un punto de vista estratégico es una de las más importantes, ya que las organizaciones pueden centrar todos sus esfuerzos en su negocio, debido a que ahora es el proveedor el que se encarga de gestionar las competencias asociadas a las tecnologías de la información.

Entre las ventajas de tipo estratégico encontramos:

- Creación de nuevos productos y servicios: esto es posible debido a la reducción de costes, que hace que sea posible que las empresas creen nuevos productos y/o servicios, que antes no resultaban rentables.
- Trabajo colaborativo: la computación en la nube permite que muchas personas a la vez puedan trabajar sobre la misma herramienta, aplicación o documento, de esta manera se fomenta la productividad, comunicación y colaboración entre empleados.
- Mejora de la productividad: como los recursos están disponibles para acceder a ellos desde cualquier ubicación física, se puede trabajar sobre los recursos de forma online, desde cualquier lugar, haciendo que aumente la flexibilidad de la empresa para trabajar a distancia y la productividad de sus empleados.
- Innovación: como ya hemos dicho, el ahorro en costes y que ahora la empresa pueda centrar sus esfuerzos en desarrollar su actividad de negocio, hace posible que la empresa tenga más posibilidades de invertir en innovación.

3.1.1.2. Ventajas técnicas

- La nube es una plataforma que permite a los usuarios disponer de la tecnología más actual, lo que hace que no haya riesgo de pérdida de competitividad por obsolescencia tecnológica. Además de esto el tiempo de adopción de nuevos servicios, infraestructuras o tecnologías es mucho menor.

- Los proveedores de *cloud computing*, ofrecen también soporte y redundancia en los sistemas que sus clientes contratan, de manera que existe una gran resistencia a desastres y buena capacidad de recuperación ante fallos. Esto es debido a que, a diferencia de los proveedores de servicios tradicionales, los equipos que usan los clientes son los propios del proveedor, por lo que les resulta más fácil acceder a ellos y resolver los problemas de una manera más rápida y eficiente, ya que conocen perfectamente la infraestructura.

3.1.1.3. Sostenibilidad

La reducción en el consumo de energía es notable, debido a que la empresa necesita de menos equipamiento propio, ya que lo contrata al proveedor. Esto es posible porque la empresa no dispone de un exceso de recursos informáticos, sino que la plataforma que contrata se adapta a las necesidades de su entidad. Los centros de datos utilizan diseños de infraestructuras avanzados, de forma que los sistemas de refrigeración y de acondicionamiento de energía se aprovechen bien y no haya pérdidas.

3.1.2. Ventajas para las administraciones públicas

Una administración pública es similar en muchos aspectos a una empresa privada, ya que ambas lo que buscan es prestar servicios, gestionar recursos, relacionarse con los proveedores, etc. Entonces, es lógico pensar que estas entidades también pueden optar por una solución *cloud* para desempeñar su actividad y así beneficiarse de las ventajas que ofrece esta tecnología.

Asimismo, hablando en términos de las administraciones públicas en España, existen algunas disposiciones y decretos que indican que los ciudadanos deben poder relacionarse con estas entidades por medios electrónicos [3], también existe lo que se conoce como "gobierno abierto", que es una directiva que se lleva a cabo en las administraciones públicas a fin de que desarrollen su gestión de forma transparente, participativa y colaborativa con los ciudadanos. Para realizar esto, es necesario implementar modelos de cooperación, reutilización e intercambio de información y servicios entre entidades.

Además de las ventajas obvias que este paradigma aporta a este tipo de entidades, tales como el ahorro en costes tecnológicos, la flexibilidad y la escalabilidad, el ahorro energético, existen otras muchas ventajas específicas para las administraciones públicas:

- Facilita las tareas de soporte tecnológico intensivo, ya que es el proveedor el que se encarga de esto y por lo tanto no se incurre en grandes gastos en este aspecto.

- Generalización de todos los servicios transversales de la Administración y por lo tanto un aprovechamiento y reutilización de las infraestructuras tecnológicas.
- Modernización de entidades pequeñas, locales o municipales, que no disponen de recursos necesarios para modernizar sus procesos y equipos de la forma tradicional.
- Investigación y colaboración en entidades con carácter educativo, tales como universidades, fundaciones, centros de investigación, etc. Incluso la cooperación entre estos centros.

La tendencia actual en las administraciones públicas está siendo la utilización de la nube privada para la migración de sus servicios y la utilización también de herramientas de la nube pública para desarrollar su actividad.

3.1.3. Ventajas para innovación e investigación

En estos términos, es totalmente esencial que exista un ambiente de colaboración e interoperabilidad entre entidades dedicadas a la investigación, además de la existencia de tecnologías avanzadas, por lo que la nube, tanto privada como pública, puede favorecer en muchos aspectos al desarrollo de estas actividades. Algunas de las ventajas más notables del *cloud computing* en estas áreas son las siguientes:

- Plataformas de colaboración entre entidades, de manera que la realización de investigaciones y proyectos de forma conjunta es mucho más rápida y eficaz.
- Estandarización de sistemas, procesos y datos entre empresas que participan en el mismo proyecto.
- Disposición de entornos grandes e intensivos de procesamiento de datos, de manera que las tareas se realicen más ágilmente y ahorrando en costes.

3.1.4. Ventajas para los ciudadanos

Anteriormente hablábamos de la Sociedad de la Información y el Conocimiento, que es la vertiente hacia la que se mueve la sociedad actual. Esto, es en gran medida debido a la capacidad que tiene la sociedad para disponer de información, compartirla y crearla. Las redes sociales han tenido un gran impacto y aceptación en la sociedad y su uso ya es ampliamente extendido. Ahora es posible acceder a la información desde cualquier localización gracias a la tecnología *cloud*.

Las características de este paradigma no son visibles para los usuarios, pero gracias a ellas, son capaces de acceder a gran variedad de servicios de forma gratuita o a precios muy bajos y lo más importante, sin necesidad de disponer de equipos especializados para ello. Algunos de estos servicios más típicos y conocidos son los gestores de correo electrónico, buscadores, enciclopedias, álbumes de fotografías, etc.

Las principales ventajas para los ciudadanos, que aporta la computación en la nube son las que nombramos a continuación:

- Amplia oferta de servicios y productos tecnológicos similares, debido a la competitividad existente, que permite a los ciudadanos poder elegir entre las soluciones que le parecen más estables, económicas y seguras.
- Variedad en los servicios disponibles para que los ciudadanos realicen sus tareas cotidianas, desde ocio, hasta trabajo, gestión del hogar, educación, etc. Gracias a los dispositivos móviles, la utilización de estos servicios es mucho más sencilla.
- Los ciudadanos pueden acceder a un mayor número de servicios, gracias a la administración electrónica. Lo hacen a través de Internet y de esta forma pueden realizar de manera más sencilla, ágil y efectiva muchos trámites de la administración.
- Disponibilidad de un "gobierno abierto" que permita que los ciudadanos puedan acceder a la información sobre las actividades realizadas por el gobierno, sus gastos, datos que genera, etc. Además de fomentar la participación ciudadana para diseñar políticas públicas.
- Las redes sociales permiten que los ciudadanos compartan experiencias, conocimientos, que hagan negocios o que demanden bienes y servicios.

3.2. Retos

La computación en la nube presenta un gran número de ventajas, que han hecho que su popularidad y aceptación en la actualidad haya aumentando considerablemente. Sin embargo, es necesario conocer los retos o riesgos específicos ante los que se enfrenta este paradigma, para poder valorar el conjunto de esta tecnología y poder decidir cual es la solución que más nos conviene adoptar.

Los principales retos del *cloud computing* están relacionados con la seguridad, más concretamente con la confidencialidad y privacidad de los datos y con la disponibilidad e integridad de los servicios y los datos.

Según la Universidad de Berkley [4], existen diez retos a los que se enfrenta el *cloud computing* y son los siguientes:

- Disponibilidad del servicio
- Bloqueo de los datos
- Confidencialidad de los datos y auditabilidad
- Cuellos de botella en la transferencia de datos
- Rendimiento del servicio impredecible
- Escalabilidad de la capacidad de almacenamiento
- Tolerancia a fallos en sistemas distribuidos a gran escala
- Rápida escalabilidad tecnológica
- Prejuicio de reputación
- Licencias de software

A continuación, en las siguientes secciones, vamos a desarrollar algunos de los retos más notables que afectan a este paradigma.

3.2.1. Disponibilidad del servicio

Como se trata de una tecnología reciente, existen dudas sobre si el proveedor será capaz de cumplir los niveles de servicio acordados en todo momento. Sobre todo en los procesos más críticos para el cliente.

Para subsanar estas dudas o problemas, hay que analizar el impacto que la pérdida de servicio pueda suponer y buscar alternativas. Además, en los acuerdos se pueden fijar cláusulas de penalización.

3.2.2. Seguridad y privacidad de los datos

La seguridad y la privacidad de los datos es uno de los mayores retos del *cloud computing* que hace que muchas empresas privadas y organizaciones públicas se planteen si implantar la tecnología. Este problema aparece por el hecho de que tanto los datos como los sistemas tecnológicos se encuentran fuera de la empresa.

Además de esto, dependiendo del tipo de datos que maneje la empresa, el lugar de desarrollo de su actividad, etc, es necesario tener en cuenta la Ley Orgánica de Protección de Datos. Esta ley, establece las medidas que hay que tomar a la hora de tratar con datos de carácter personal. Así como aspectos de transferencia internacional de datos, subcontratación, derechos ARCO, etc.

En esta situación, en la que los datos son críticos, es el cliente el responsable de ellos y, por lo tanto, el responsable de ellos y se debe encargarse de que el proveedor cumpla las medidas de seguridad pertinentes.

Algunos de los problemas comunes suelen aparecer a la hora de hacer uso de un sistema de computación en la nube son los siguientes:

- Fuga de información por ataques a la plataforma *cloud*
- Borrado no seguro de la información
- Incidencias no comunicadas
- Pérdida de disponibilidad del servicio
- Pérdida de información por parte del proveedor

3.2.3. Falta de estandarización

Todavía surgen problemas en los entornos *cloud* debidos a problemas de estandarización. El problema es que al, muchas veces, no existir un estándar definido, las migraciones de datos y aplicaciones y la integración con otros subsistemas de negocio, suele ser complejo. La mayor ventaja de la estandarización es la posibilidad de migrar las aplicaciones a un proveedor diferente, o a otro entorno con mayor facilidad.

A pesar de que es un campo en el que todavía queda mucho trabajo, ya existe un buen número de estándares para trabajar en entornos en la nube.

3.2.4. Adaptabilidad a la nube

Este término se refiere a que para hacer un buen uso de los entornos en la nube y sacarles su mayor provecho y beneficiarse de todas sus ventajas, es necesario adaptar las aplicaciones a esta nueva tecnología. Se ha de tener en cuenta que esta infraestructura está formada por sistemas muy paralelizables, por lo que para maximizar el rendimiento, las aplicaciones que se migren a la nube deben adaptarse a esta forma de trabajo.

Esto puede hacer que las aplicaciones tengan que ser rediseñadas para trabajar en entornos altamente paralelos y además para que se haga uso de formatos estándar, de forma que si en un futuro hay que adaptarse a nuevas versiones, el cambio no sea muy grande y se realice de forma fluida.

Capítulo 4

Estado actual

Tras analizar en el capítulo anterior tanto las ventajas como los retos del *cloud computing*, se realiza en este capítulo un análisis de la situación actual de esta tecnología. Se hace de manera más concreta en España y de forma general en el resto del mundo. Además se incluyen las leyes que afectan directamente a las empresas que utilizan este paradigma.

4.1. Situación actual en España

La sociedad actual cada vez se muestra más abierta a adoptar los servicios que ofrece la nube. Es una tecnología que va evolucionando para ofrecer cada vez más ventajas y posibilidades.

La economía española, está orientada sobre todo al sector servicios, de forma que este nuevo paradigma de el *cloud computing* se adapta perfectamente. La gestión de la información y el trato con los clientes son dos factores muy importantes y que pueden verse beneficiados con la adopción de esta tecnología. Asimismo, como ya hemos visto en capítulos anteriores, las ventajas que proporciona la nube son especialmente favorables para las pequeñas y medianas empresas, precisamente el tipo de empresa predominante en España, ya que estas no son capaces de afrontar grandes inversiones en infraestructuras de datos. La nube les permite ponerse a la altura de empresas más grandes en cuanto a tecnología e infraestructura, haciendo que las pymes se vuelvan más competitivas en el mercado.

La crisis económica en el país, también ha sido un impulsor para la implantación de la tecnología *cloud* en España. Las empresas disponen de menos capital para invertir en tecnología, y muchas de ellas no pueden permitirse adquirir equipos propios y dedicados, por lo que optan cada vez más por el *cloud computing*.

Existe un estudio [5] en el que se identifica a las ubicaciones más propicias para implantar servicios *cloud* y España se encuentra entre ellos. Alguna de las razones son los costes frente al resto de países europeos (norte y oeste), las infraestructuras de transporte y la relación con Sudamérica. Por el momento solo se elige a España

como ubicación para los centros de prestación de servicios, pero no como ubicación para los *data centers* de las empresas líderes en *cloud*. Esto se debe en gran medida al clima del territorio español y a la poca facilidad para acceder a energías limpias de forma barata.

Para ser más específicos a continuación se listan las oportunidades que ofrece España para el desarrollo de servicios de *cloud*:

- Costes laborales
- Cualificación
- Posición geográfica
- Mayoría de pymes

4.1.1. Dentro del sector servicios

Como ya hemos comentado, el sector servicios es el más predominante en la economía española. Es un sector que se centra en las personas y por este motivo debe centrarse en su actividad de negocio y estar atento para captar nuevas oportunidades. El *cloud computing* ayuda mucho a desempeñar esto, ya que elimina el problema de tener que encargarse de la gestión de infraestructuras, aplicaciones, etc, y pasa a hacerse cargo el proveedor *cloud*.

Dentro de este sector, las entidades financieras son las que más fácil han tenido la adopción de esta tecnología. Debido a la crisis económica, muchas cajas de ahorro se vieron obligadas a fusionarse con otras para reducir gastos. El problema era que muchas tenían tanto modelos de gestión diferentes como modelos y sistemas informáticos y ante este problema es cuando se plantea la posibilidad de utilizar un entorno *cloud*, para que ambas entidades fusionaran sus sistemas en uno común.

4.1.2. Dentro de las administraciones públicas

La aparición y fomento del *cloud computing* en las administraciones públicas, se debe también a la crisis económica. Las administraciones se han visto obligadas a reducir gastos y para esto se ha decidido, en muchas ocasiones implantar un entorno *cloud*. Además de esto, es necesario que exista comunicación entre las entidades repartidas por todo el territorio, de manera que puedan compartir datos, información y sistemas.

La mayor ventaja de disponer de computación en la nube es para los usuarios, ya que muchos servicios se pueden usar de forma *online*, tales como, la solicitud de documentos, subvenciones, tributos, etc. Utilizando una plataforma *cloud*, el usuario no debe preocuparse de desplazamientos, ni de horarios, ni de las competencias de las que se encarga cada organismo público.

4.2. Situación actual en el mundo

Entre las soluciones favoritas se encuentra la nube privada, debido a la cierta desconfianza que sigue existiendo en torno a la seguridad que pueden proporcionar las nubes públicas. Y por sectores nos encontramos con que son las administraciones públicas y las entidades financieras las que más apuestan por esta tecnología para desarrollar su actividad [6].

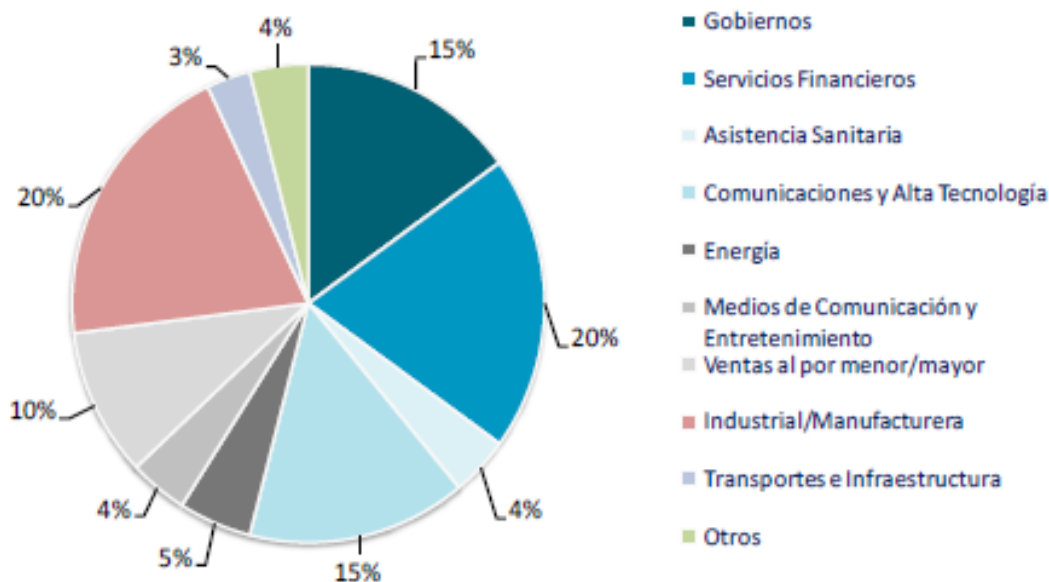


Figura 4.1: Soluciones *cloud* por sector.

Las grandes empresas del sector se encuentran en Estados Unidos y disponen de un mercado fuerte y consolidado. Sin embargo, en la Unión Europea se está apostando fuerte para impulsar el uso de la computación en la nube.

En Europa se apuesta por el uso de la tecnología *cloud* en las administraciones públicas para, sobre todo, ahorrar en costes. En países como Alemania, Francia y Gran Bretaña, se hicieron grandes apuestas por este nuevo paradigma, con inversiones para comenzar proyectos de implantación.

4.3. Ley aplicable

En este apartado nombraremos la legislación vigente, que se aplica a esta nueva tecnología, tanto dentro del territorio español, como en el resto del mundo. Como ya hemos dicho en los apartados anteriores, es el cliente el responsable del tratamiento de los datos personales que se van a usar en el desarrollo de su actividad empresarial. A pesar de que el contrato que firme indique que la responsabilidad es del proveedor, esta no se desplaza y seguirá siendo del cliente.

Si la empresa trata con datos personales, en un entorno en la nube, es muy

importante conocer la ubicación de estos, debido a que las garantías que se exigen para su protección son distintas dependiendo del país en el que se encuentren alojados. Si el traspaso de datos se realiza entre los países de la Unión Europea, Islandia, Liechtenstein y Noruega, no se considera que se esté realizando una transferencia internacional de datos, pero entre el resto de países, sí, por lo que se deben proporcionar garantías jurídicas adecuadas para el tratamiento y la transferencia de estos datos. Actualmente existen varios países de fuera de la Unión Europea, con un nivel adecuado de protección.¹

Algunas de las leyes vigentes que afectan a cualquier entorno *cloud* son las que se indican y explican en los siguientes apartados.

4.3.1. Ley Orgánica de Protección de Datos - LOPD

En el territorio español, se aplica la Ley Orgánica de Protección de Datos [7] y su reglamento de desarrollo (RLOPD [8]) que establecen medidas para la regulación de la seguridad y las técnicas a la hora de tratar con datos de carácter personal. Esta ley también regula la transferencia internacional de datos, la subcontratación y la atención de los derechos ARCO, que son los de acceso, rectificación, cancelación y oposición.

Para poder considerar que se está cumpliendo esta ley, se deben tener en cuenta los siguientes aspectos:

- Si el contrato con el proveedor incluye datos o activos de carácter personal, el proveedor debe garantizar e indicar la ubicación de los mismos, y en caso de que la ubicación o sea reconocida como un emplazamiento de puerto seguro, se deberá solicitar una autorización a la agencia.
- Deben establecerse las condiciones de acceso a los datos, las medidas de seguridad a implementar, las reglas de devolución en caso de cierre del contrato y las subcontrataciones a terceros.
- Se debe garantizar el borrado de los datos tras el fin del contrato.
- Cláusulas de penalización, responsabilidad civil y procedimientos y directrices de seguridad.

4.3.2. Ley de Servicios de la Sociedad de Información y de Comercio Electrónico - LSSI

La Ley de Servicios de la Sociedad de Información y de Comercio Electrónico [9], o LSSI, es una ley de origen español, que surge en 2002. Tiene como obje-

¹https://www.agpd.es/portaleswebAGPD/canalresponsable/transferencias_internacionales/index-ides-idphp.php

tivo la regulación del régimen jurídico de la sociedad de la información y de la contratación por vía electrónica, como su nombre indica, el comercio electrónico y los servicios de intermediación. Esta ley se aplicará solo en caso de que la actividad constituya una actividad económica o lucrativa para el prestador. De esta manera, se considerará prestador tanto a los operadores de red y servicios de comunicaciones, ISP y empresas y ciudadanos con página *web* propia.

4.3.3. Ley de Acceso Electrónico de los Ciudadanos a los Servicios Públicos - LAECSP

Es la ley [3] que reconoce el derecho que tienen los ciudadanos para poder relacionarse con las administraciones públicas por medios electrónicos. Además de esto, es la que regula los aspectos básicos de la utilización de las tecnologías de la información en la actividad administrativa de las entidades y en las relaciones entre entidades, entre otras cosas.

Gracias a esta ley, se facilita el acceso de los ciudadanos a las diferentes plataformas públicas y se aumenta la transparencia administrativa. Asimismo, se mejora la comunicación, el intercambio de datos y de servicios entre entidades administrativas.

4.3.4. Ley de Firma Electrónica

La firma electrónica [10] es un concepto jurídico, es el equivalente electrónico a la firma manuscrita. Algunos ejemplos de firma electrónica son la firma digital, usuario y contraseña, firma con lápiz electrónico, etc.

La ley que se aplica a este concepto tiene como objeto regular la firma electrónica, su eficacia jurídica y la prestación de servicios de certificación. Según esta ley, la firma electrónica es el conjunto de datos en forma electrónica, consignados junto a otros o asociados con ellos, que pueden ser utilizados como medio de identificación del firmante.

Capítulo 5

Caso de estudio

Los objetivos de este trabajo son estudiar y analizar las oportunidades y retos que ofrece el paradigma de *cloud computing*, para ello, en capítulos anteriores se realiza un estudio teórico de las ventajas y los riesgos de esta tecnología, así como se estudia la legislación aplicable y el estado actual del *cloud*. En este capítulo, realizaremos el estudio desde el punto de vista práctico, con un ejercicio real, que nos permitirá evaluar aspectos como la rapidez, la flexibilidad, la gestión tecnológica y otros muchos aspectos. Se define el caso de estudio propuesto así como los requerimientos que debe cumplir el mismo.

Como ya hemos visto, la computación en la nube se puede clasificar, según su modelo de servicio, en *SaaS*, *PaaS* e *IaaS*; *software*, plataforma e infraestructura, respectivamente. En este caso de estudio lo que se pretende es desarrollar una aplicación en un entorno *cloud*. Por lo tanto, lo que realmente se está haciendo es desarrollando un *software* como servicio.

5.1. Especificaciones

El caso de estudio consiste en el desarrollo e implementación de una aplicación en un entorno *cloud*. Es tarea del alumno elegir y evaluar las herramientas a utilizar, tales como el lenguaje de programación, *framework*, base de datos, etc. En este caso, la aplicación permitirá la monitorización de aspectos esenciales en servidores. Esto es muy útil en la sociedad actual, ya que la mayoría de las empresas dispone de página *web*, e incluso muchas personas disponen de sitio *web* personal.

Los usuarios de la aplicación podrán definir sitios web para monitorizarlos, seleccionando para ello los parámetros que más le interesen.

Los requisitos iniciales definidos para la aplicación son los que se definen a continuación:

- La aplicación dispondrá de tareas paralelas para aprovechar al máximo las

características de las plataformas de computación en la nube, por lo tanto será una aplicación pensada y desarrollada para un entorno *cloud computing*.

- Debe permitir la definición de perfiles de usuario, de forma que cada uno de los usuarios pueda definir los parámetros a monitorizar y en caso de que se requiera los umbrales permitidos según su criterio.
- En el supuesto de que las pruebas definidas por el usuario para cada uno de sus sitios definidos fallen, se enviará una notificación de correo electrónico, indicándole el problema.
- Varios parámetros disponibles para monitorizar, como por ejemplo el estado del sitio, o cambios de contenido.

5.2. Planteamiento

Una vez definidos los requisitos para la aplicación se toman las decisiones necesarias para proceder con el desarrollo de la misma. En esta sección se detallan tanto las herramientas utilizadas para el desarrollo de la aplicación, como el entorno *cloud* decidido.

5.2.1. Entorno de desarrollo

La elección del entorno *cloud* de desarrollo es una de las decisiones más importantes para el desarrollo de una aplicación en la nube. Será lo que marque las directrices para la implementación y posterior alojamiento y puesta en funcionamiento de la aplicación.

En la sección 2.4 se definen tres de las soluciones *cloud* actuales más populares, las que se tuvieron en cuenta como opción: *Google Cloud Platform*, *Amazon Web Services* y *Microsoft Azure*.

La plataforma elegida para el desarrollo de la aplicación fue *Google Cloud Platform*. Una de las razones principales es la madurez del servicio de *PaaS* de la empresa *Google*. A pesar de que *Amazon Web Services* dispone ya de plataforma como servicio (*Beanstalk*), empezó como *IaaS*. Por esto, para alojar una aplicación el usuario debía realizar tareas de administración de sistemas de forma que configurara de forma adecuada la instancia de la que disponía, así como la base de datos, etc. *Google Cloud Platform*, en cambio, empezó como *PaaS*, con lo que se conoce como *Google App Engine*.

Uno de los posibles problemas de GAE (*Google App Engine*) era que usaba *Big Table* como base de datos, pero ya dispone de soporte para bases de datos relacionales. *Big Table* es un sistema de bases de datos creado por la empresa,

que se caracteriza por ser distribuido y muy eficiente, pero es una base de datos de tipo no relacional.

5.2.1.1. Google App Engine

Google App Engine o GAE, es la plataforma que permite alojar aplicaciones web incluida en *Google Cloud Platform*. Tiene soporte para lenguajes de programación como Go, Java, PHP y Python y para algunos *frameworks* de estos lenguajes.

Una de sus principales ventajas es que dispone de un SDK para cada uno de ellos, tanto para Linux como para Windows y OS X. Otra de sus ventajas más notables es el panel de administración, que dispone de una interfaz limpia y clara, con muchísimas herramientas para realizar de manera más sencilla la monitorización de tus aplicaciones. Así como un control de versiones para la aplicación e integración con herramientas de control de versiones como GitHub o BitBucket.

5.2.2. Herramientas a utilizar

Tras definir los requisitos de la aplicación y el entorno *cloud* sobre el que se va a desarrollar el caso de estudio, es necesario escoger las herramientas que se van a utilizar.

5.2.2.1. Python/Django

Python es un lenguaje de programación interpretado, *open source*, de propósito general, multiplataforma, de tipado dinámico y multiparadigma. Lo que significa que permite tanto programación orientada a objetos, programación funcional y programación imperativa. Está diseñado con una filosofía que favorece a que se escriba código legible.

Para este lenguaje existen muchas implementaciones diferentes, así como *frameworks*, pero para el desarrollo de esta aplicación web se elige hacer uso de Django.

El *framework* Django se utiliza para desarrollo web, es *open source* y sigue el patrón de diseño de modelo-vista-controlador (MVC). Está pesado sobre todo para facilitar la creación de sitios o aplicaciones web muy complejos. Para ello utiliza una filosofía basada en la reutilización y la conectividad de componentes, el desarrollo rápido y el principio de no repetición (que viene del inglés "DRY", *don't repeat yourself*).

5.2.2.2. MySQL

MySQL es un sistema gestor de bases de datos relacional, multihilo y multiusuario. Este sistema está disponible para una gran cantidad de plataformas y de

sistemas y además permite elegir entre diferentes mecanismos de almacenamiento entre los que varía la velocidad de operación, las transacciones, etc. Permite replicación y es un amplio subconjunto del lenguaje SQL.

Google Cloud SQL, se basa en este sistema gestor y permite crear bases de datos e instancias para asignarlas a las aplicaciones web. Además elimina el problema del mantenimiento y la gestión de la base de datos.

5.2.2.3. Google App Engine SDK

GAE dispone de un kit de desarrollo de *software* propio y para cada uno de los lenguajes para los que tiene soporte. Si usamos como sistema operativo de desarrollo, Windows o Mac OS X, dispondremos de interfaz gráfica del SDK.

Se trata de una herramienta muy útil, ya que incluye una aplicación de un servidor web, que simula el entorno de *App Engine*, de manera que es muy sencillo hacer pruebas a la aplicación sin necesidad de desplegarla en la nube.

5.2.2.4. PyCharm

PyCharm es un entorno de desarrollo integrado (IDE) usado para programar en Python. Dispone de depurador, análisis de código, test unitarios y control de versiones. Además de esto soporta desarrollo con Django, Google App Engine, Flask, Web2Py, y muchos otros.

Tiene disponibles varios tipos de licencia, comunitaria, profesional y licencias para uso educativo.

5.2.2.5. Linux Mint Debian

Linux Mint es una distribución del sistema operativo GNU/Linux, y está basada en Debian. La mayoría de los paquetes *software* que incluye se encuentran bajo licencia de *software* libre.

Este el sistema operativo utilizado para el desarrollo de la aplicación. Para este sistema, no existe aplicación gráfica del GAE SDK.

Capítulo 6

Desarrollo del caso de estudio

En el capítulo anterior, el capítulo 5, se define el caso de estudio y los requerimientos que se deben cumplir para su correcta finalización. En este capítulo, se explica la metodología de trabajo elegida y el desarrollo seguido para la creación de la aplicación web. Asimismo, se indican las funcionalidades finales de la misma y los problemas encontrados durante la realización del caso de estudio.

6.1. Metodología

El desarrollo de la aplicación web se hecho bajo la metodología de desarrollo en cascada. En este tipo de metodología se ordenan las tareas de forma descendente y secuencialmente, de manera que el inicio de cada etapa debe esperar a la finalización de la etapa anterior. Al final de cada una de las etapas debe hacerse una valoración o revisión para concluir si queda finalizada la etapa y se puede pasar a la siguiente.

Las tareas definidas para el desarrollo del caso de estudio fueron las siguientes:

1. Análisis y decisión del entorno *cloud*
2. Definición de herramientas a utilizar
3. Configuración del entorno
4. Creación de perfiles de usuario
5. Creación de sitios asociados a los usuarios
6. Implementación de parámetros a monitorizar
7. Actualizaciones en *background*
8. Pruebas y detección de errores
9. Cierre

6.2. Desarrollo

Una vez decidida la metodología de trabajo, las herramientas a utilizar y se han definido claramente los requisitos que debe satisfacer la aplicación, es posible comenzar con el desarrollo de la misma. Comenzaremos haciendo una introducción sobre los módulos que componen la aplicación, para luego especificar más el desarrollo de cada uno de ellos.

6.2.1. Estructura general

El *framework* Django, genera automáticamente un árbol de directorios. Si hacemos uso de un IDE con soporte para GAE, además conseguiremos generar los ficheros específicos para que nuestra aplicación funcione en la plataforma.

La aplicación final está compuesta por tres aplicaciones de Django. Se separa en aplicaciones, debido a que se trata de modelos de datos diferentes y de este modo queda mejor estructurado el proyecto y es más fácil añadir funcionalidades o eliminarlas. El esquema o árbol de directorios es el siguiente:

```
TFG
├── myproject
├── profiles
├── mysites
└── controller
```

Bajo el directorio *myproject* se encuentran los ficheros de configuración del proyecto, son comunes para cada una de las aplicaciones que conformen la aplicación final. Los siguientes directorios *profiles*, *mysites* y *controller*, son los directorios en los que se alojan las aplicaciones, cada una con sus propios modelos de datos y vistas.

La aplicación *profiles* se encarga de la creación y el almacenamiento de los perfiles de usuario que posteriormente van a utilizar la aplicación. Cada uno de los usuarios introduce sus datos de identificación y su correo electrónico para que pueda recibir las notificaciones.

En *mysites* se encuentra la gestión de los sitios asociados a cada uno de los usuarios. Esta aplicación permite agregar un nuevo sitio para monitorizar, así como los parámetros y umbrales. Se permite tanto la modificación como el borrado de los mismos.

Por último, *controller* es la aplicación encargada de realizar las actualizaciones de los parámetros que el usuario definió para cada sitio. Se definen las funciones a realizar en cada uno de los casos. Esta es la aplicación que realiza el trabajo real, y cada una de sus funciones es llamada para la actualización de los parámetros de de forma paralela.

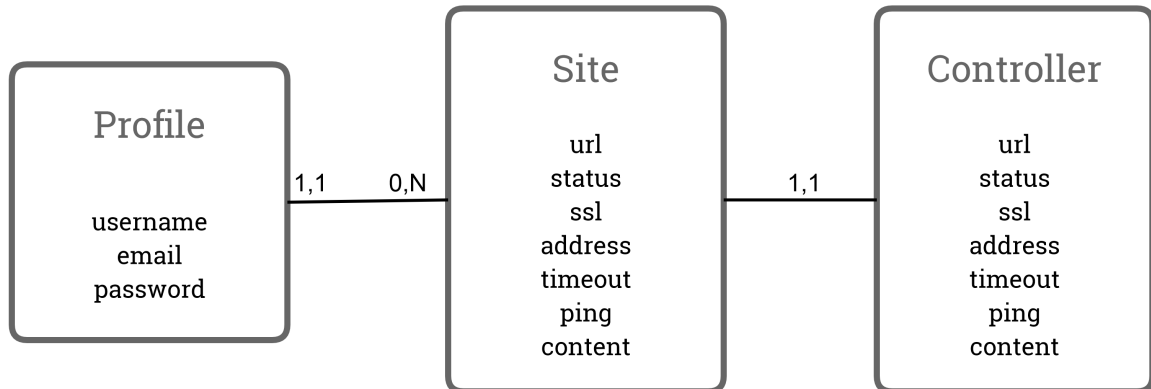


Figura 6.1: Relación de los modelos de datos

6.2.2. Perfiles de usuario

En la primera fase de desarrollo de la aplicación se implementaron los perfiles de usuario. Django dispone de un sistema de autenticación de usuarios propio, que es capaz de gestionar las cuentas de usuario, los grupos, permisos y las *cookies* de las sesiones. Además, es posible modificar los perfiles de usuario a nuestro gusto y de manera que se adapten mejor a las necesidades de la aplicación.

En este caso se creó un perfil de usuario muy sencillo, con una serie de campos básicos, tales como *username*, *email* y contraseña. La vista que se utiliza para la creación de los usuarios es una vista basada en clase. Una vista es una función que recibe un *request* y devuelve un *response*. En Django, es posible utilizar una clase como una vista, que es lo que se conoce como vistas basadas en clases, entonces deja de ser una función y pasa a ser un objeto. De esta manera se reutiliza código aprovechando la herencia y herencia múltiple de clases.

Este modelo de usuario lo vamos a denominar *Profile* y posteriormente tendrá asociados tantos *Sites* como el propio usuario defina.

TFG Application [Index](#) [Login](#) [Register Here](#)

User registration.

Please, register here!

Username: Required. 30 characters or fewer.

Email address:

Password:

First name:

Last name:

Figura 6.2: Formulario de registro de nuevo usuario

TFG Application [Index](#) [New Site](#) [Logout](#)

TFG Application.

Hello rulox!

ID	Site URL	Configure	Delete	Details
1	http://www.google.com/	Update	Delete	Details

Figura 6.3: Usuario logueado con un sitio definido

6.2.3. Sitios y parámetros

Tras la creación de los perfiles de usuario, se realiza la implementación de los sitios y de los parámetros que se van a asociar a ellos. Cada usuario define su propios sitios y los parámetros que quiere que se monitoreen.

El modelo de datos para cada sitio, llamado *Site*, tiene un único *Profile* asociado, asimismo, se definen los parámetros disponibles para monitorizar. Para la creación de los sitios, también se utiliza una vista basada en clase. Además es posible modificar la configuración del sitio, para cambiar los umbrales o los parámetros definidos, y eliminarlo.

TFG Application [Index](#) [New Site](#) [Logout](#)

Url:

Status OK:

SSL Cert Verification:

Timeout Limit: 0 Disabled.

Address:

Ping Limit (ms): 0 Disabled. Fails if ping > your parameter.

Content Changed: Check if web content has been updated.

Type of content: Content you want to track for changes

Figura 6.4: Creación de un nuevo sitio

TFG Application [Index](#) [New Site](#) [Logout](#)

<http://www.google.com/>

Status	Address	SSL	Ping	Timeout	Changed	Last Updated
200	74.125.195.99	✓	306.0	OK	False	July 8, 2015, 1:19 a.m.

Figura 6.5: Parámetros actualizados de un sitio

6.2.4. *Background*

Para almacenar los resultados de las tareas a monitorizar, se crea un nuevo modelo de datos que llamaremos *SiteController*. Este modelo de datos se asocia a un *Site* existente y que previamente estaba asociado a un *Profile*.

En la sección 6.2.1, vemos que la aplicación final está formada por tres módulos, el módulo encargado de la actualización de los parámetros y del envío de las notificaciones por correo electrónico al usuario es *controller*. En este módulo se definen las funciones que van a actualizar cada uno de los parámetros. Se definen de forma individual, para luego poder ejecutarlas de manera concurrente.

Las actualización concurrente se hace mediante el servicio *cron* disponible en la plataforma *cloud*. Este servicio permite definir tareas y programar su ejecución a horas determinadas o incluso en intervalos de tiempo. Estas tareas son más conocidas como *cron jobs*. Estas tareas invocan una URL, haciendo uso de una petición GET.

Push Queues Pull Queues **Cron Jobs**

A cron job is a scheduled task that runs at a specific time or at regular intervals.

Cron job ^	Description	Frequency	Last run	Status	
/content_changed/	content changed checker job	every 5 minutes (GMT)	2015-07-07 (16:39:00) 1 sec late	Success	Run now
/ping/	ping checker job	every 5 minutes (GMT)	2015-07-07 (16:39:00) On time	Success	Run now
/ssl/	ssl checker job	every 5 minutes (GMT)	2015-07-07 (16:39:00) 1 sec late	Success	Run now
/status/	status checker job	every 5 minutes (GMT)	2015-07-07 (16:40:02) 2 sec late	Success	Run now
/timeout/	timeout checker job	every 5 minutes (GMT)	2015-07-07 (16:40:02) 2 sec late	Success	Run now

Figura 6.6: *Cron Jobs* activos en *Google Cloud Platform*

6.2.5. Pruebas y detección de errores

Cada una de las etapas definidas en la sección 6.1 incluye una subetapa de corrección de errores, debido a que al tratarse de un desarrollo en cascada, es mejor probar la aplicación y corregir los errores a medida que surgen.

Para hacer pruebas con los parámetros implementados se han utilizado varios tipos de direcciones web:

- Páginas estáticas
- Blogs
- Páginas dinámicas
- Páginas con más de una dirección *host*

Se ha hecho esta clasificación, ya que es necesario tener en cuenta el tipo de página que estamos monitorizando para preveer el resultado de la actualización del parámetro.

Al testear el parámetro de *cambio de contenido*, es importante tener en cuenta el parámetro que utilicemos. Actualmente es posible elegir entre el cuerpo de la web, los *links*, la cabecera o los párrafos. Si elegimos una página estática, el contenido no cambiará, sin importar el parámetro que elijamos para hacer la comprobación. En cambio, con páginas dinámicas, el resultado será muy diferente. Este tipo de páginas (un ejemplo de ellas es *google.com*), tienen parámetros que cambian constantemente, como por ejemplo *timestamps*. Entonces, hay que ser cuidadosos a la hora de la elección del parámetro, para no recibir notificaciones de cambio de contenido en cada comprobación.

Otro parámetro sensible al tipo de sitio elegido es el de la comprobación de la dirección de *hosting*, ya que hay sitios que disponen de balanceador, y no siempre se sirven al usuario desde la misma dirección IP.

6.3. Funcionalidades

Finalmente, y según lo indicado en la sección 5.1, la aplicación cumple todos los requisitos previstos. Esta aplicación web permite realizar tareas de monitorización de servidores.

- Dispone de perfiles de usuario
- El usuario define sus propias páginas web e indica los parámetros a monitorizar
- Modificación y eliminación de un sitio de usuario
- En caso de error con alguno de los parámetros se notifica por correo electrónico al usuario

Actualmente en la aplicación se pueden monitorizar seis parámetros diferentes, y son los siguientes:

- Estado del *site*: mediante una petición HTTP, se comprueba si el *site* está operativo. En cambio, si no se encuentra disponible por un fallo de servidor, el usuario será notificado.
- Certificado SSL: es posible comprobar si el certificado SSL de un sitio se encuentra disponible.
- *Timeout*: para este parámetro es necesario configurar un umbral. El umbral que coloque el usuario indicará cuanto tiempo como máximo puede tardar su sitio en responder. Si el sitio tarda más del tiempo indicado en responder, se enviará una notificación.
- Dirección de alojamiento del sitio: comprueba si la dirección ip sobre la que se sirve la página indicada ha cambiado. Si el sitio se sirve desde varias direcciones diferentes, el parámetro cambiará y devolverá un error. Solo será útil para páginas que se sirvan desde la misma dirección siempre.
- *Ping*: este parámetro indica el tiempo en el que debe responder la página.
- Cambio en el contenido: es posible marcar este parámetro para comprobar si el contenido del sitio web ha cambiado. Lo que se hace es extraer del código fuente algún o algunos campos que el usuario indique, como por

ejemplo, *body*, *links*, cabecera, etc. Una vez extraído se le pasa un algoritmo de reducción criptográfico (*hash*²), el *md5*.

6.4. Problemas encontrados

Durante el desarrollo de una aplicación suelen surgir problemas, es muy útil tenerlos en cuenta para futuras líneas de trabajo o ampliación del proyecto.

6.4.1. Versiones

Para desarrollar la aplicación web se ha utilizado Python/Django como lenguaje y *framework*, sobre Google App Engine, como se indica en la sección 5.2. Actualmente, Python va por su versión 3.4 y Django se encuentra en la 1.8, en cambio, para trabajar con estas herramientas en GAE, debe hacerse con versiones anteriores, estas son 2.7 y 1.5 para Python y Django respectivamente.

El problema aparece debido a que muchas de las librerías actuales para Django, no funcionan en una versión tan anterior a la actual, por lo que hay que buscar soluciones alternativas.

6.4.2. Base de datos

Este resulta ser un problema derivado del problema de las versiones desactualizadas. A partir de Django 1.7, se sustituye el comando *syncdb*, utilizado para crear las tablas en la base de datos de la aplicación por nuevos comandos que facilitan la modificación de tablas.

El comando *syncdb*, realiza la creación de tablas para los modelos que aún no se habían creado, es decir que este comando no ejecuta en ningún caso un *ALTER TABLE*. En caso de que se modifique la estructura o campos de un modelo, hay que modificar la tabla de forma manual o borrar entera la base de datos y migrala de nuevo. En local puede usarse una aplicación externa para realizar los cambios en las tablas.

En el entorno *cloud*, en cambio, no se puede usar la aplicación externa, sino que hay que crear una nueva instancia, asignarla al proyecto actual y modificar el fichero de configuración de la aplicación Django, para que ahora apunte a la nueva instancia de la base de datos.

Además de esto, Cloud SQL está limitado al uso de MySQL como sistema gestor de la base de datos, mientras que en otros entornos *cloud*, como puede ser AWS, el abanico de posibilidades es mucho mayor.

²Una función hash es aquella que dada una determinada entrada, la convierte en una cadena de logitud fija.

6.4.3. *Third-part libraries*

GAE incluye por defecto la librería estándar de Python y algunas librerías externas [11]. Estas últimas se pueden agregar al proyecto de manera sencilla incluyéndolas en uno de los ficheros de configuración de GAE. Sin embargo, el resto de librerías hay que incluirlas en un directorio dentro del directorio de proyecto.

El código de las librerías debe subirse a la nube y por lo tanto contarán como ficheros que pueden llenar o hacer que aumente la cuota de ficheros considerablemente. Además de esto, para que sean visibles por GAE, debe incluirse un fichero de configuración en el que se indique cual es el directorio que contiene las librerías.

6.4.4. Ficheros estáticos

Normalmente, en Django, los ficheros estáticos se incluyen en el directorio del proyecto, y se añade la ruta de ubicación de este directorio al fichero de configuración del proyecto. En cambio, si queremos que estos ficheros sean encontrados en la nube, tenemos que indicarlo en el fichero específico del entorno.

Cuando estamos utilizando el administrador de Django, App Engine no es capaz de encontrar los ficheros estáticos de estilo y accedemos a las páginas sin formato. Esto es porque por defecto Python busca los ficheros en el *path* de la librería Django y no puede acceder a ellos por problemas de permisos. Para solucionarlo, debemos copiar el directorio de estáticos del administrador (ubicado en la librería) en nuestro directorio de estáticos del proyecto.

6.4.5. Tareas en *background*

Existen múltiples aplicaciones que permiten ejecutar tareas en *background* para Python y Django, pero todas ellas utilizan sus propios sistemas de bases de datos y deben ser lanzadas de forma independiente. Algunos ejemplos de estas aplicaciones son Celery o RabbitMQ. Es posible usar estas dos aplicaciones en la nube, pero para ello habría que crear una máquina virtual en Google Compute Engine donde pudiéramos instalar la aplicación para ponerla a funcionar.

App Engine proporciona un servicio de *cron*, con el que es posible ejecutar tareas de forma automática en ciertos intervalos de tiempo o a horas específicas.

Capítulo 7

Conclusiones y líneas futuras

Anteriormente, se define tanto el caso de estudio, como las herramientas utilizadas y el proceso de desarrollo seguido para su finalización y una vez acabado, se definen los problemas que han surgido durante la implementación. En este capítulo se añaden unas pautas para una posible ampliación del trabajo en el futuro y se realizan las conclusiones del trabajo.

7.1. Conclusiones

Personalmente me ha parecido un trabajo muy interesante, por muchos factores. El primero de ellos es que se trata de un proyecto que incluye tanto trabajo de análisis como de desarrollo, por lo que, bajo mi punto de vista hace que quede un estudio mucho más completo y real. Los apartados de análisis pretenden dar a conocer en que consiste el *cloud computing*, así como las ventajas y los retos reales que se presentan actualmente. Mientras que el caso de estudio permite evaluar de forma real si estas ventajas y retos son tales.

Otra de las razones es que me ha permitido familiarizarme con una tecnología que no conocía, y que, gracias al desarrollo de este trabajo he descubierto que me gusta mucho. Además de esto, la elección del lenguaje de programación y del *framework* ha sido ambiciosa, ya que son herramientas con las que no estaba muy familiarizada y ahora me siento cómoda trabajando con ellas.

Me habría gustado poder desarrollar la aplicación en otro entorno *cloud* a la vez, para poder comparar dos plataformas de manera real, pero estoy muy contenta con el resultado del trabajo realizado. Enfrentarme a un trabajo de esta magnitud me ha permitido crecer y además disfrutar haciéndolo, ya que he descubierto que el *cloud computing* es una tecnología que tiene mucho que ofrecer y me gustaría profundizar más con ella.

7.2. Trabajos futuros

A pesar de que el proyecto haya quedado finalizado, siempre existen mejoras que se pueden realizar sobre el mismo, incluso líneas de trabajo que se basan en la idea principal pero que luego se bifurcan para obtener nuevos resultados. Algunas de las opciones de trabajo futuro que se pueden realizar sobre el proyecto son las que se nombran en los siguientes apartados.

7.2.1. Interfaz

La interfaz actual es una interfaz muy básica, que carece de un diseño atractivo e intuitivo para el usuario. En caso de que se pretenda seguir desarrollando y dando soporte a la aplicación, la mejora de la interfaz sería un punto a tener en cuenta.

7.2.2. Añadir parámetros

Actualmente es posible monitorizar una gran cantidad de parámetros de un servidor. Algunos de estos parámetros ya están implementados en la aplicación, según se indica en la sección 6.3, sin embargo sería interesante tener en cuenta aspectos como las redirecciones, DNS, *web scraping*³, etc.

Además, ya que se trata de una aplicación modular, la modificación y adición de parámetros se realizaría de manera muy sencilla.

7.2.3. Configuración del tiempo de actualización

Una de las mejoras que se pueden realizar sobre la aplicación es permitir que el usuario pueda decidir la frecuencia con la que se actualiza cada uno de los parámetros que asoció a sus sitios.

Esto resulta útil porque hay parámetros que necesitan ser actualizados con más frecuencia que otros, como por ejemplo el estado de la página o el tiempo de respuesta.

7.2.4. Estudio de nuevo entorno *cloud*

Quizás la línea de trabajo futuro más interesante sería de elegir un nuevo entorno de *cloud computing* para desplegar la aplicación. De esta manera se podría hacer no solo un estudio y posterior valoración del desarrollo web en la nube, sino que se podría realizar una comparación de varias plataformas para ver los puntos fuertes y las debilidades de una y de otra.

³Web Scraping es una técnica que consiste en la extracción de información de sitios web con el uso de *software*.

Capítulo 8

Summary and Conclusions

In the previous chapter, the case study, the used tools and the development processes were defined, as well as the problems found during the application's implementation. This chapter shows the project's conclusions and some guidelines to future work lines.

8.1. Conclusions

For many reasons, I consider that this project has been very interesting for me. First and foremost, this is a project that includes analysis and development tasks, so that, from my point of view, it makes the project a real, comprehensive study. The purpose of the analysis sections is to make known the cloud computing, as well as its real advantages, opportunities and challenges. While the case study let us know if these are really as the analysis shows.

Another reason is that I could become familiar with a technology I did not know and that I have found that I like the cloud. Moreover, the choice of the programming language and framework has been very ambitious, because they were tools I was not familiar with.

I would have liked to develop the web application in another cloud environment simultaneously, in order to compare both platforms. But in spite of this, I am very proud and happy with the results I have got. To face a project like this has allowed me to grow as a person and I have enjoyed during the process. I am eager to keep practising and developing new applications in the cloud.

8.2. Future work

In spite of the ending of the project, some improvements or new work lines could be developed. Some of these are explained in the following sections.

8.2.1. Interface

The current interface is a really basic one. It lacks of an attractive and user-friendly design. In case of future support and development, the improvement of the interface could be a point to consider.

8.2.2. New parameters

Currently, it is possible to monitor an enormous amount of parameters into a single server. There are some of them already developed in the web application, just like the section 6.3 says. However, considering parameters such as redirects, DNS, web scraping⁴, etc, could be interesting.

Moreover, as the application is modular, it is very easy the addition and modification of parameters.

8.2.3. Configurable update time

One of the improvements that could be made in the web application is to allow the configuration of the parameters' update time. In this way, the user could specify the frequency of refresh.

This could be very useful, because there are parameters whose update time should be shorter than another parameter's update time, such as the status of the page and the time of response.

8.2.4. New cloud environment

Perhaps the most interesting future work line to consider is to choose a new cloud environment to study and to deploy the web application. As a result of that, it would be possible to study and evaluate the development of applications in the cloud and to contrast and compare different platforms, in order to analyse the strengths and weaknesses of each one.

⁴Web scraping is a computer software technique of extracting information from websites.

Capítulo 9

Presupuesto

En este capítulo se indica el coste estimado del desarrollo del trabajo. Cabe destacar que no se ha incurrido en gastos por adquisición de licencias, ya que el IDE utilizado dispone de licencia gratuita para estudiantes.

Se medirá tanto el costo del trabajo de análisis, como el trabajo de desarrollo y el coste por el uso de la plataforma en la nube.

9.1. Coste por trabajador

Se estiman los costes tanto de un programador como de un analista. El trabajo del programador está estimado para un mes y medio y un mes para realizar el trabajo de análisis. Se ha estimado que en un mes, las horas de trabajo son de ciento sesenta.

Concepto	Cantidad
Coste por hora	10 euros
Horas de trabajo	240 horas
Total	2400 euros

Tabla 9.1: Coste por el trabajo de programación

Concepto	Cantidad
Coste por hora	15 euros
Horas de trabajo	160 horas
Total	2400 euros

Tabla 9.2: Coste por el trabajo de análisis

9.2. Coste por uso del entorno

En la siguiente tabla se desglosa el coste por el uso de la plataforma *cloud* y sus distintos componentes y herramientas. No es posible calcular en total del coste, ya que es relativo y depende muchos factores, algunos como el número de peticiones, usuarios registrados, ficheros almacenados, etc.

Concepto	Cantidad
Cloud SQL por hora	0.092 euros
App Engine Instance (hora)	0.045 euros
Tráfico saliente (GB)	0.11 euros
Almacenamiento (GB/mes)	0.16 euros

Tabla 9.3: Coste por el entorno cloud

9.3. Coste total

El coste total estimado para el desarrollo del proyecto es el que se muestra en la siguiente tabla. Queda excluido el coste del entorno en la nube.

Concepto	Cantidad
Programador	2400 euros
Analista	2400 euros
Total	4800 euros

Tabla 9.4: Coste total

Bibliografía

- [1] Greg F. Pfister. *In Search of Clusters*. 1998.
- [2] Peter Mell and Timothy Grance. The NIST definition of cloud computing. *Special Publication 800-145*, 2011.
- [3] Ley de Acceso Electrónico a los Ciudadanos a los Servicios Públicos. <http://www.boe.es/buscar/doc.php?id=BOE-A-2009-18358>.
- [4] Above the Clouds: A Berkeley View of Cloud Computing, UCB/EECS-2009-28. Technical report, 2009.
- [5] Ian Marriot. Gartners 30 Leading Locations for Offshore Services. *Gartner*, 2009.
- [6] Forecast: Public Cloud Services, Worldwide and Regions, Industry Sectors, 2010-2015. Technical report, 2011.
- [7] Ley Orgánica de Protección de Datos de Carácter Personal. <https://www.boe.es/buscar/act.php?id=BOE-A-1999-23750>.
- [8] Real Decreto. <https://www.boe.es/buscar/act.php?id=BOE-A-2008-979>.
- [9] Ley de Servicios de la Sociedad de Información y de Comercio Electrónico. <http://www.lssi.gob.es/paginas/Index.aspx>.
- [10] Ley de Firma Electrónica. <http://www.boe.es/buscar/act.php?id=BOE-A-2003-23399>.
- [11] Librerías externas incluidas en Google App Engine. <http://cloud.google.com/appengine/docs/python/tools/libraries27>.
- [12] Google Cloud Platform. <https://cloud.google.com/>.
- [13] Amazon Web Services. <http://aws.amazon.com/es/>.
- [14] Microsoft Azure. <http://azure.microsoft.com/es-es/>.
- [15] Docker. <https://www.docker.com/>.

- [16] Google App Engine - GAE. <https://cloud.google.com/appengine/>.
- [17] Django. <https://www.djangoproject.com/>.
- [18] Python. <https://www.python.org/>.
- [19] MySQL. <https://www.mysql.com/>.
- [20] PyCharm - JetBrains. <https://www.jetbrains.com/pycharm/>.