



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA
Máster en Computación Paralela y Distribuida
Departamento de Sistemas Informáticos y Computación

***Aplicación de las Tecnologías Cloud
en el Análisis de Estructuras de
Edificación***

TRABAJO FIN DE MÁSTER

Presentado por:

Pedro de la Fuente Aragón

Dirigido por:

Ignacio Blanquer Espert

Codirigido por:

José Miguel Alonso Ábalos

Valencia, Septiembre de 2013

Resumen

El presente Trabajo Fin de Máster describe el desarrollo de un servicio Cloud seguro, fiable y de alta productividad para el análisis de estructuras de edificación. Dicho servicio se apoya en la plataforma Cloud desarrollada en el proyecto europeo VENUS-C y se ejecuta sobre la infraestructura Cloud de Windows Azure. Las simulaciones en el Cloud se ejecutan por medio de Architrave, un entorno software avanzado para el diseño, análisis y visualización de edificios y estructuras de ingeniería civil. La migración al Cloud se ha llevado a cabo por medio del Generic Worker, una implementación Web Role de Windows Azure que gestiona la ejecución de los trabajos remotos. El servicio Cloud emplea el estándar CDMI para el envío y la descarga de datos. Además, el trabajo incluye el desarrollo de un cliente gráfico, encargado de la gestión de las simulaciones remotas, del envío y la recepción de los datos y de informar al usuario sobre el estado y el progreso de las simulaciones. Para validar la implementación, se ha ejecutado un caso de estudio que consiste en el diseño de una estructura real de gran envergadura. Los resultados de la ejecución muestran las ventajas del sistema Cloud respecto a la aproximación secuencial, en términos de incremento de velocidad, eficiencia y cantidad de soluciones estructurales analizadas por unidad de tiempo.

Abstract

The current Master's Thesis describes the development of a high throughput, secure and reliable Cloud service for the analysis of building structures. This service relies on the platform developed at the European VENUS-C project, and it is deployed over the Windows Azure Cloud Infrastructure. The simulations in the Cloud are performed by Architrave, an advanced software environment for the design, 3D analysis and visualisation of buildings and civil engineering structures. The migration to the Cloud has been carried out by means of the Generic Worker, a Windows Azure Web Role which manages the execution of the remote tasks. The Cloud service employs the CDMI standard for uploading and downloading data. Also, this work includes the development of a GUI client, in charge of managing the remote simulations, sending and receiving the input and output data and informing the user about the status and the progress of the simulations. In order to validate the implementation, a use case consisting on the design of a real large building has been executed. The execution results shows the advantages of the Cloud system with respect to the sequential approach, in terms of speed-up, efficiency and the amount of structural solutions analysed per time unit.

Agradecimientos

*A José Miguel Alonso, por todo el esfuerzo dedicado al presente trabajo,
y por brindarme la oportunidad de trabajar juntos y haber confiado en mí.*

*A Ignacio Blanquer, por su dedicación y predisposición,
a Pau Lozano, por su colaboración y su importante aportación a los desarrollos,
y a mis compañeros del GRyCAP, por considerarme y tratarme como uno más entre ellos.*

*A Geraldine Ros, por su ánimo en los momentos más difíciles,
su apoyo incondicional, su comprensión, y por ser capaz de sacar lo mejor de mí.*

*Y a mi familia, por todo el apoyo que me han ofrecido durante todo este tiempo,
y por saber que siempre podré contar con ellos.*

Índice de contenidos

1	Introducción	1
1.1	Objetivos	2
2	Análisis de Estructuras	5
2.1	Análisis estructural en el proceso de diseño de un edificio	5
2.2	Clasificación de las cargas que actúan sobre una estructura	7
2.3	Tipos de análisis estructural.....	8
2.4	Métodos matriciales para el análisis estático de estructuras.....	9
2.4.1	Análisis estático de estructuras.....	11
2.4.2	Análisis dinámico de estructuras	11
3	Tecnologías de las Aplicaciones Distribuidas	13
3.1	Arquitecturas Distribuidas.....	13
3.1.1	Arquitectura de un nivel	14
3.1.2	Arquitectura de dos niveles	14
3.1.3	Arquitectura de tres niveles	15
3.1.4	Arquitectura de N niveles.....	16
3.1.5	Arquitectura P2P	16
3.2	Estado del Arte en las Tecnologías para el Desarrollo de Aplicaciones Distribuidas ..	17
3.2.1	Remote Procedure Call.....	17
3.2.2	Transaction Processing Monitor	19
3.2.3	Object Request Broker	20
3.2.4	Middleware orientados a mensajes.....	23
3.2.5	Servicios web	24
4	Computación Cloud	29
4.1	Definición de Cloud computing.....	29
4.2	Modelos de servicio Cloud	30
4.2.1	Infraestructura como Servicio (IaaS).....	31
4.2.2	Plataforma como Servicio (PaaS).....	32
4.2.3	Software como Servicio (SaaS).....	33
4.3	Modelos de despliegue Cloud	34
4.3.1	Cloud público	34
4.3.2	Cloud privado	35
4.3.3	Cloud comunitario	36
4.3.4	Cloud híbrido.....	37
4.4	Tecnologías del Cloud computing	37
4.4.1	Virtualización	37
4.4.2	Arquitecturas Orientadas a Servicios	38
4.5	Beneficios del Cloud computing.....	39

4.6	La plataforma Windows Azure	40
4.6.1	Modelos de ejecución.....	41
4.6.2	Gestión de datos.....	42
4.6.3	Interconexión.....	42
4.6.4	Análisis empresarial.....	42
4.6.5	Comunicación entre aplicaciones.....	43
4.6.6	Cacheado de información.....	43
4.6.7	Otros servicios.....	43
4.7	El proyecto VENUS-C	44
4.7.1	Servicio de Gestión de Trabajos.....	45
4.7.2	Gestión de los datos.....	47
4.7.3	Servicios de Monitorización, Contabilidad y Facturación.....	47
4.7.4	Eliminación de Tráfico Redundante.....	48
5	<i>Computación Cloud aplicada al Análisis de Estructuras de Edificación</i>	49
5.1	Estado del arte del uso de Computación Distribuida y Cloud en el Análisis de Estructuras	50
5.2	Servicio Cloud de Análisis de Estructuras de Edificación basado en el software Architrave	51
5.2.1	La herramienta de cálculo estructural Architrave.....	53
5.2.2	El Gestor Cliente de Simulaciones Remotas.....	54
5.2.3	El Servicio Cloud de Análisis de Estructuras de Edificación.....	58
5.2.4	Ejemplo de uso del Servicio Cloud de Análisis Estructural.....	60
5.3	Caso de Estudio de Análisis Estructural en el Cloud	63
5.3.1	Descripción del caso de estudio.....	63
5.3.2	Ejecución del caso de estudio.....	64
5.3.3	Análisis de prestaciones.....	65
6	<i>Conclusiones Finales y Trabajos Futuros</i>	71
6.1	Trabajos Futuros	74
7	<i>Publicaciones Realizadas</i>	77
8	<i>Bibliografía</i>	79

Índice de figuras

<i>Figura 1: Diagrama de flujo del proceso de diseño de un edificio.</i>	7
<i>Figura 2: Clasificación de acciones según el Código Técnico de la Edificación.</i>	8
<i>Figura 3: Tipos de análisis estructural.</i>	8
<i>Figura 4: Modelo alámbrico de estructura plana de ejemplo.</i>	10
<i>Figura 5: Arquitectura de un nivel.</i>	14
<i>Figura 6: Arquitectura de dos niveles.</i>	15
<i>Figura 7: Arquitectura de tres niveles.</i>	15
<i>Figura 8: Funcionamiento básico de RPC.</i>	18
<i>Figura 9: Funcionamiento de una llamada RPC transaccional.</i>	20
<i>Figura 10: Enlace estático y dinámico de interfaces en CORBA.</i>	22
<i>Figura 11: Modelo de colas de mensajes en MOM.</i>	23
<i>Figura 12: Esquema y ejemplo de mensaje SOAP.</i>	25
<i>Figura 13: Esquema de especificación de servicio WSDL.</i>	26
<i>Figura 14: Esquema de entrada de registro UDDI.</i>	26
<i>Figura 15: Responsabilidad de gestión en los diferentes modelos de servicio Cloud.</i>	31
<i>Figura 16: Representación de un Cloud público.</i>	35
<i>Figura 17: Representación de un Cloud privado.</i>	36
<i>Figura 18: Representación de un Cloud comunitario.</i>	36
<i>Figura 19: Representación de Cloud híbrido.</i>	37
<i>Figura 20: Servicios de Windows Azure.</i>	41
<i>Figura 21: Arquitectura de VENUS-C.</i>	45
<i>Figura 22: Arquitectura software del sistema Cloud.</i>	52
<i>Figura 23: Captura de la definición de una simulación remota en Architrave.</i>	54
<i>Figura 24: Captura del Gestor Cliente de Simulaciones Remotas.</i>	56
<i>Figura 25: Arquitectura física y ejemplo de uso del Servicio Cloud de Análisis Estructural.</i>	61
<i>Figura 26: Modelo de la estructura del Banco Nórdico en Architrave.</i>	64
<i>Figura 27: Tiempo de respuesta en horas obtenido en la ejecución del caso de uso completo.</i>	68
<i>Figura 28: Speed-up obtenido en la ejecución del caso de estudio completo.</i>	68
<i>Figura 29: Eficiencia obtenida en la ejecución del caso de estudio completo.</i>	69

Índice de tablas

<i>Tabla 1: Características principales de los modelos de servicio Cloud.</i>	<i>34</i>
<i>Tabla 2: Resultados de la simulación de una de las soluciones estructurales en el Cloud.</i>	<i>65</i>
<i>Tabla 3: Tiempos de ejecución de las distintas fases de la simulación en el Cloud (configuración B).</i>	<i>66</i>
<i>Tabla 4: Tiempos de ejecución de las distintas fases de la simulación en el Cloud (configuración A).</i>	<i>67</i>

1 Introducción

El análisis de estructuras de edificación es el proceso que determina la respuesta de una estructura, por medio de los resultados de esfuerzos, tensiones y desplazamientos que experimenta, ante una serie de cargas que actúan sobre ella. Este proceso, que ha sido automatizado para su resolución mediante la computación, normalmente requiere la resolución de grandes sistemas de ecuaciones lineales o no lineales, así como la resolución de problemas de valores propios, exigiendo una elevada potencia de cálculo y gran cantidad de volumen de almacenamiento, e incrementándose aún más en varios órdenes de magnitud cuando se trata del análisis dinámico en el tiempo. Además la precisión de los resultados en el análisis de estructuras es un aspecto fundamental, por lo que requiere la aplicación de métodos numéricos de análisis precisos y eficientes sobre modelos estructurales tridimensionales con elevado detalle y nivel de realismo.

Durante el diseño de una estructura de edificación, el ingeniero o arquitecto responsable suele evaluar diferentes diseños y soluciones estructurales en base a diferentes criterios como la seguridad, la funcionalidad o el coste económico de la estructura, así como posibles limitaciones de ejecución de la obra. Cada una de estas alternativas estructurales incluye variaciones en las dimensiones y las propiedades de sus elementos estructurales, así como en las cargas que actúan sobre la misma. Todas estas alternativas deben ser evaluadas con el fin de obtener aquella que cumpla con todos los requisitos estructurales y de seguridad, y además presente el menor coste económico. Por otro lado, la Normativa de Construcción Sismorresistente (NCSE-02) requiere la simulación de cada una de las soluciones estructurales bajo la influencia de al menos cinco terremotos representativos distintos. Por tanto, la fase de evaluación analítica de las diferentes soluciones estructurales puede llegar a ser excesivamente costosa en términos temporales y computacionales.

Debido a la complejidad del proceso de análisis durante el diseño de una estructura, el cálculo estructural tradicionalmente se ha llevado a cabo introduciendo una serie de simplificaciones con el fin de reducir el tamaño del problema y el volumen de resultados generado, y obtener los resultados de cálculo en tiempos de respuesta razonables, haciendo de este modo viable el análisis de estructuras mediante computadores personales. En cambio, este tipo de simplificaciones que son válidas para estructuras sencillas y regulares, resultan inadecuadas para estructuras singulares o de gran envergadura.

El uso de técnicas de computación avanzadas como la Computación de Altas Prestaciones o *High Performance Computing* (HPC) dentro de las herramientas de cálculo permite abordar de manera realista el análisis estático y dinámico de estructuras complejas y de gran dimensión, y reducir considerablemente los tiempos de respuesta. Sin embargo, a pesar de las capacidades que ofrecen los entornos de Computación de Altas Prestaciones, los estudios de arquitectura o ingeniería raramente poseen o hacen uso de este tipo de infraestructuras.

Afortunadamente, la tecnología de Computación en la Nube o *Cloud Computing* aparece como una solución a los requisitos computacionales de las organizaciones, permitiendo el uso de recursos remotos de cómputo y de almacenamiento para satisfacer la demanda computacional de los análisis dinámicos en el tiempo de estructuras de gran envergadura.

En este contexto, el presente Trabajo Fin de Máster se enmarca dentro del proyecto del 7º Programa Marco de la Unión Europea titulado “Virtual Multidisciplinary Environments Using Cloud Infrastructures (VENUS-C)”, con referencia EC 261565, y que persigue el desarrollo de una plataforma Cloud orientada a la investigación, que dé soporte al despliegue y a la ejecución de aplicaciones de carácter científico y permita a la comunidad científica su uso de manera sencilla.

La participación dentro del proyecto VENUS-C tuvo lugar mediante la consideración de las aplicaciones de arquitectura e ingeniería civil como uno de los siete escenarios de usuario diferentes. Dichos escenarios fueron los encargados de determinar los requisitos que debía proporcionar dicha plataforma y de validar su comportamiento mediante el despliegue de aplicaciones Cloud científicas. En este escenario se ha desarrollado un servicio Cloud que permite llevar a cabo el análisis estático y dinámico de estructuras de edificación en dicho entorno de computación. A través de dicho servicio, los arquitectos, los ingenieros de cálculo y la comunidad científica pueden realizar la ejecución, de manera simultánea, de múltiples análisis estáticos y dinámicos de estructuras complejas en recursos computacionales remotos, de manera transparente y mediante un modelo de pago por uso, logrando reducir notablemente los tiempos de la fase análisis dentro del proceso de diseño de un edificio.

El trabajo ha dado lugar, además, a dos publicaciones en congresos científicos, así como diversas participaciones en conferencias y referencias en artículos de prensa.

1.1 Objetivos

El principal objetivo que se plantea dentro del presente Trabajo Fin de Máster consiste en el diseño, la implementación y el despliegue de un servicio Cloud de análisis de estructuras de edificación, multiusuario, fiable, seguro, de alta productividad y que permita a los arquitectos y a la comunidad científica estructural llevar a cabo el cálculo de sus estructuras en entornos de computación en la Nube, de manera transparente y mediante un modelo de pago por uso. Gracias a este modelo de ejecución, se pretende evitar la inversión por parte de los ingenieros de cálculo y los estudios de arquitectura en software especializado y en potentes recursos hardware propios o costosas infraestructuras hardware que normalmente tienden a estar infrautilizados. De este modo, el Servicio Cloud de Análisis Estructural pondrá a disposición de los usuarios un software eficiente además de la misma potencia computacional que ofrecen dichas infraestructuras, permitiendo así satisfacer los elevados requisitos de tiempo de cómputo y de almacenamiento de las simulaciones dinámicas en el tiempo de estructuras de gran envergadura, y pagando únicamente por el software y por los recursos computacionales y de almacenamiento empleados.

La ejecución de las simulaciones estructurales mediante el servicio Cloud debe realizarse a través de una herramienta de análisis estructural. Para dicho cometido se ha escogido Architrave, un entorno que ofrece el soporte completo al proceso de diseño y análisis de estructuras de edificación. Este entorno de análisis estructural es el resultado de los trabajos de investigación desarrollados en la Universitat Politècnica de València por el grupo CiD del Departamento de Mecánica de Medios Continuos y Teoría de Estructuras y por el grupo GRyCAP del Instituto de Instrumentación para Imagen Molecular. De este modo, se pretende integrar de manera transparente el envío de simulaciones al servicio Cloud a través del componente llamado Architrave Cálculo, encargado de invocar el Simulador Estructural para su ejecución local y visualizar los

resultados obtenidos; así como el uso del Simulador Estructural de Architrave, que integra técnicas HPC basadas en MPI, para la ejecución de las simulaciones estructurales dentro del servicio Cloud.

Previamente al desarrollo del servicio Cloud, es importante llevar a cabo un análisis del estado del arte de las tecnologías disponibles para el desarrollo de aplicaciones distribuidas. Este análisis permitirá conocer las diferentes alternativas existentes, así como las características que ofrece cada una de ellas.

Asimismo, para obtener el máximo partido de la solución Cloud dedicada al análisis estructural, resulta básico conocer las características del Cloud Computing, las posibilidades que ofrece en sus distintos modelos de servicio y de despliegue y los beneficios que aportan las distintas alternativas. Además, puesto que el Servicio Cloud de Análisis Estructural se basa en la plataforma VENUS-C, resulta imprescindible estudiar los diferentes componentes que incluye la arquitectura de la misma, a fin de ser empleados.

Por otro lado, el Servicio Cloud de Análisis Estructural a desarrollar debe ofrecer ciertas garantías, de modo que permita ser explotado por la comunidad de arquitectos e ingenieros de cálculo dentro de un entorno de producción. Por ello, dicho servicio debe incluir mecanismos de tolerancia a fallos, de modo que todas las peticiones de cálculo puedan ser atendidas satisfactoriamente. Además, debido a que el servicio está disponible a través de la red, debe incluir mecanismos de seguridad que garanticen que pueda ser utilizado únicamente por usuarios autorizados, preservando además la privacidad de sus datos referentes a sus proyectos estructurales, tanto dentro del propio almacenamiento del servicio Cloud como durante el tránsito de dichos datos entre la máquina cliente y el servicio.

Además, debido a que el servicio Cloud consistirá en un sistema multiusuario, la escalabilidad y la productividad son aspectos fundamentales a considerar. En este sentido, el servicio a desarrollar deberá poder ajustar su capacidad computacional en función de la carga de trabajos del sistema, así como permitir la ejecución simultánea de simulaciones tanto estáticas como dinámicas, sin que las primeras se vean penalizadas por el elevado coste computacional de las segundas. Por ello, resultará necesario llevar a cabo un análisis de prestaciones de la solución desarrollada a través de la ejecución de un caso de uso realista, que permita evaluar tanto la capacidad computacional y las prestaciones de la herramienta como los requisitos de usabilidad de la misma.

2 Análisis de Estructuras

El procedimiento completo que se lleva a cabo durante el diseño de una estructura de edificación conlleva una serie de fases, donde primero se define un modelo estructural y posteriormente se analiza su comportamiento para evaluar si cumple con las condiciones de seguridad, funcionalidad y durabilidad asociadas al tipo de estructura a edificar y al uso para el cual se ha diseñado. Este análisis permite determinar, de manera simulada, la respuesta de dicho modelo ante un conjunto de cargas externas de diferente naturaleza, que actúan de manera independiente o simultáneamente sobre el mismo.

El modelo estructural, que como se menciona previamente se genera durante las primeras fases del proceso de diseño, incluye las propiedades geométricas y estructurales del edificio, así como las distintas cargas o acciones externas aplicadas sobre el mismo. Este conjunto de propiedades que se definen permiten obtener, durante la fase de análisis estructural, una serie de características para poder determinar el comportamiento de dicha estructura, como son su rigidez, en el caso de un análisis de tipo estático, y la masa y el amortiguamiento asociados a la misma, en el caso de un análisis dinámico.

El análisis estructural permite obtener el comportamiento de la estructura ante las cargas externas [1]. Debido a que la geometría consta de un conjunto de elementos estructurales continuos, este proceso requeriría determinar la respuesta de la estructura ante un número infinito de puntos que la componen y, además, ante un número infinito de instantes de tiempo en el caso de un análisis de tipo dinámico en el tiempo. Esta aproximación resulta inabordable, por lo que en la práctica se realiza la discretización tanto del dominio espacial como del temporal, permitiendo obtener una solución numérica del problema. En el caso del dominio espacial, se define un número finito de puntos de la estructura, predeterminados de manera adecuada, en los cuales se obtendrá la respuesta de la estructura. Dicha estructura queda entonces dividida en barras, que se corresponden con las vigas o pilares de la estructura, y en elementos finitos, los cuales discretizan elementos continuos tales como losas de cimentación o de forjado y muros. Por otro lado, en el caso del dominio temporal, se define un conjunto de instantes de tiempo en los cuales se obtendrá la respuesta de la estructura.

A través del análisis estructural es posible obtener, para el conjunto de puntos de la estructura definido, los resultados de deformaciones que experimenta la misma y los esfuerzos y tensiones internas que sufren sus elementos estructurales. En el caso de un análisis estático, se obtienen un conjunto de resultados de deformaciones y esfuerzos finales como consecuencia de las cargas externas constantes que actúan sobre la misma. En cambio, debido a la naturaleza variable de las cargas dinámicas, en el análisis dinámico a lo largo del tiempo se obtiene un conjunto de resultados para cada paso de tiempo, que dependerá del estado de la estructura en el instante anterior y del valor de la carga en el instante actual, por lo que tanto la complejidad computacional como el coste espacial es mucho más elevado.

2.1 Análisis estructural en el proceso de diseño de un edificio

El proceso de diseño de un edificio da lugar a una serie de etapas o fases previas a la ejecución en obra de dicha estructura, tal y como muestra la Figura 1. Inicialmente, el arquitecto o ingeniero de estructuras considera un conjunto de diseños o de alternativas estructurales diferentes que tratan de dar solución al mismo problema. Estos diseños

estructurales determinan la disposición general y las propiedades físicas de los elementos estructurales, como su geometría y dimensiones, así como las condiciones de contorno de la misma. En este proceso se determinan además las hipótesis de carga o combinaciones de acciones que debe soportar la estructura, de modo que se simulen los efectos más desfavorables de la misma dentro de las condiciones de diseño, de seguridad y de uso de dicha estructura. El objetivo de diseñar diferentes alternativas estructurales consiste en poder seleccionar la solución que mejor cumpla los requisitos de diseño y ofrezca además una mejor respuesta a las acciones o cargas externas consideradas. La evaluación con elevado detalle de diferentes diseños estructurales puede llegar a ser una labor muy costosa en términos temporales, por lo que suelen emplearse modelos simplificados. En cambio, aún con el diseño de dichos modelos simplificados, esta fase es una de las que más tiempo requiere dentro del proceso completo de diseño de una estructura.

Tras seleccionar el diseño estructural más apropiado, éste se estudia con más detalle en las siguientes fases de diseño. Para ello, se evalúan a continuación diferentes configuraciones o soluciones estructurales procedentes del mismo diseño original, que consisten en variaciones de las propiedades de los diferentes elementos estructurales. Dentro de esta etapa se analiza la respuesta de las distintas soluciones estructurales en base a las tensiones, esfuerzos y deformaciones obtenidos a partir de cada hipótesis de carga, donde se obtiene el comportamiento de la estructura ante las cargas y las tensiones ejercidas sobre los elementos estructurales. Este proceso trata de obtener la solución estructural más eficiente, de modo que cumpla con los requisitos estructurales y las normativas de seguridad, presentando además el menor coste económico.

Seguidamente, una vez que ha sido seleccionada la solución estructural más eficiente, normalmente son necesarias la aplicación de una serie de pequeñas modificaciones en el modelo estructural, como por ejemplo ligeras variaciones en las dimensiones de las columnas, o la armadura interna de las vigas, de modo que el modelo estructural completo cumpla con todas las condiciones de las normativas de seguridad. En esta fase se comprueba por tanto que los distintos elementos estructurales sean capaces de resistir las cargas que se han configurado en las distintas hipótesis, y los resultados de sus esfuerzos queden dentro de los límites de seguridad impuestos por las normativas.

Finalmente, cuando el modelo estructural ha sido definido por completo, únicamente queda generar la documentación asociada a la solución estructural en forma de listados y planos de obra, para llevar a cabo la ejecución de la misma.

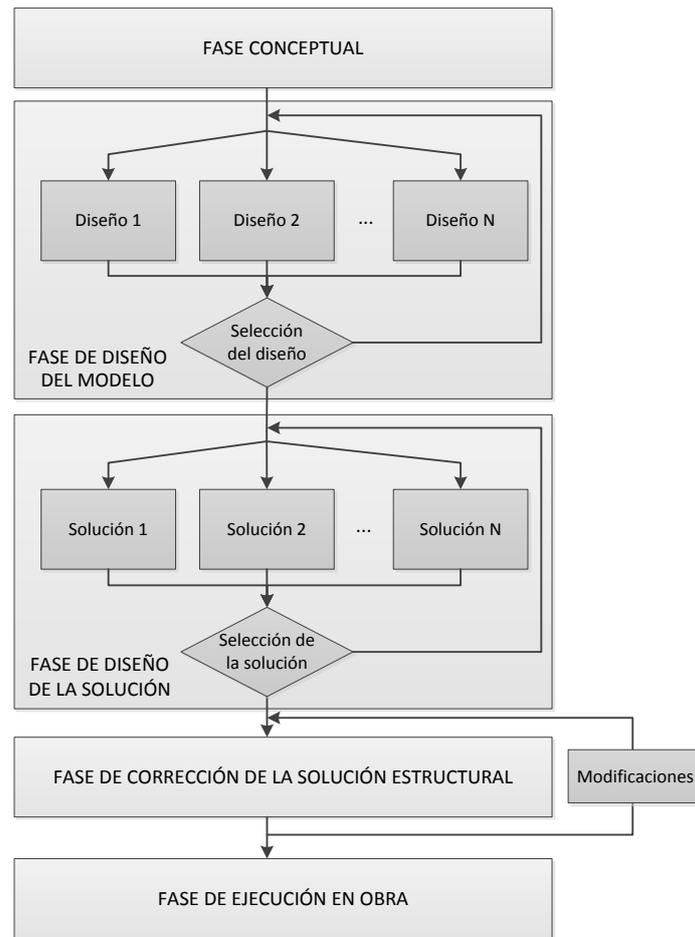


Figura 1: Diagrama de flujo del proceso de diseño de un edificio.

2.2 Clasificación de las cargas que actúan sobre una estructura

Las cargas se definen como acciones directas de una fuerza que actúa sobre uno o varios elementos de una estructura y que son capaces de producir estados tensionales en los mismos. A continuación se describe la clasificación de las distintas acciones que establece el Código Técnico de la Edificación [2], como se puede observar en la Figura 2. Dentro de ella existen tres grupos principales: las acciones permanentes, las variables y las accidentales.

Las cargas permanentes son las acciones o cargas propias de la edificación que actúan en todo momento y son constantes en posición y magnitud. Dentro de las cargas permanentes se distinguen las cargas del peso propio de la estructura, las de pretensado y las propias del terreno. Por otro lado, las cargas variables son ajenas a la estructura y que pueden actuar o no. Se subdividen en cinco categorías: las de sobrecarga de uso, las acciones sobre barandillas y elementos divisorios, las de viento, las acciones térmicas y las de nieve.

Finalmente, las cargas accidentales comprenden las acciones que acontecen de manera accidental e inciden sobre la estructura, y presentan una probabilidad de actuación pequeña pero de gran importancia. Este grupo de cargas se subdivide en las acciones sísmicas, las de incendio, las de impacto, y otras posibles acciones específicas del uso de la estructura (como por ejemplo fábricas químicas, laboratorios o almacenes de materiales explosivos).

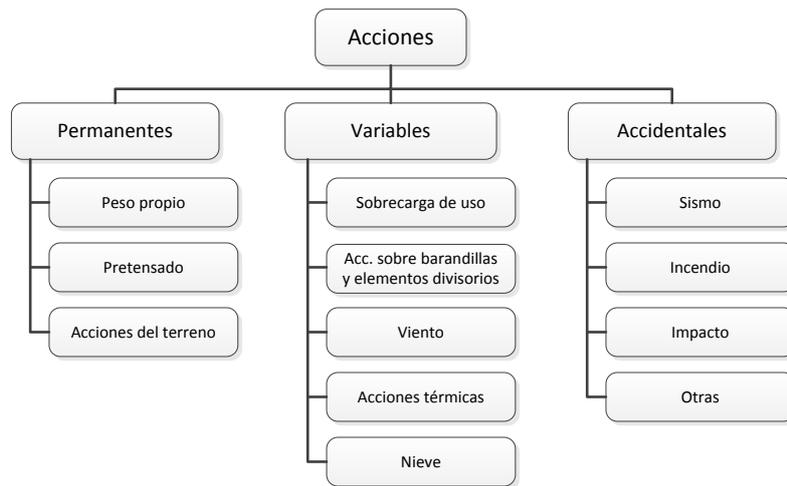


Figura 2: Clasificación de acciones según el Código Técnico de la Edificación.

2.3 Tipos de análisis estructural

Existen diferentes métodos o alternativas para determinar la respuesta de una estructura ante un conjunto de cargas. A continuación se establece una clasificación general que describe brevemente cada uno de los tipos de cálculo, como se muestra en la Figura 3.

Los distintos tipos de cálculo se clasifican en dos grandes grupos, en función de cómo se considera que se comporta la rigidez de los elementos estructurales. Dentro de ellos se distinguen dos clases de análisis diferentes: el lineal y el no lineal.

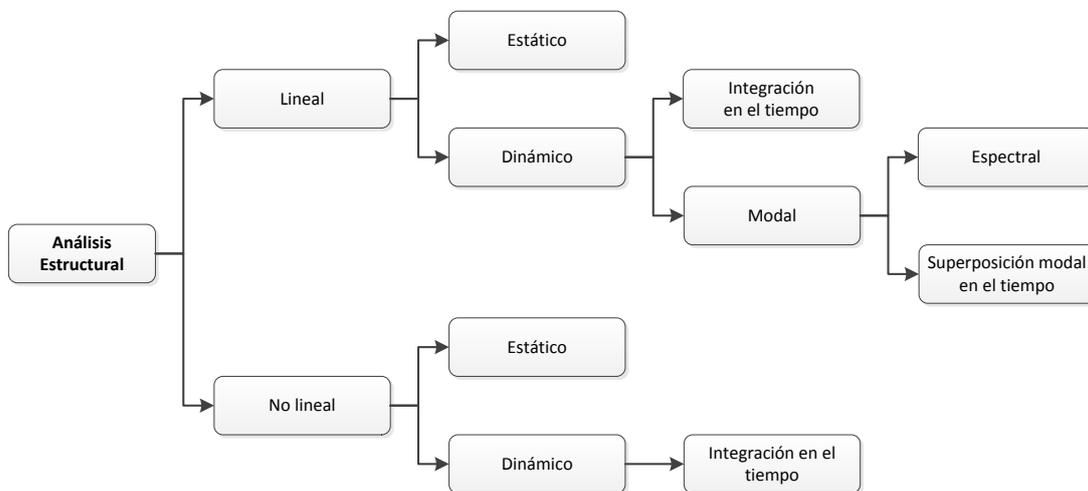


Figura 3: Tipos de análisis estructural.

El análisis estructural de tipo lineal considera que la variación de la rigidez de los elementos estructurales ante su deformación es suficientemente pequeña como para prescindir de su influencia durante el proceso de cálculo. De este modo, asume que ni las propiedades de la forma ni las del material de dichos elementos estructurales cambiarán durante el proceso de deformación, generando así una respuesta lineal en función del valor de las cargas aplicadas sobre ella. Dentro de este tipo de análisis se distingue el análisis estático y el análisis dinámico de estructuras. Mediante el análisis estático, es posible determinar la respuesta de la estructura ante cargas que no varían

con el tiempo, como por ejemplo ante las acciones directas. En cambio, el análisis dinámico permite obtener el comportamiento de la estructura ante cargas que varían su valor en el tiempo, como son las acciones sísmicas.

Dentro del análisis lineal existen dos métodos diferentes que permiten calcular la respuesta dinámica de una estructura: los métodos de integración en el tiempo, que como se describe más adelante, obtienen el comportamiento de la estructura ante cargas dinámicas mediante un procedimiento numérico paso a paso, y el análisis modal, que determina las frecuencias naturales y los modos de vibración de los elementos de dicha estructura mediante un procedimiento de cálculo de valores propios. Asimismo, existen dos técnicas que permiten determinar el comportamiento de una estructura ante cargas dinámicas por medio del análisis modal: el análisis espectral, que determina la respuesta máxima de una estructura ante una carga sísmica en forma de espectro de respuesta, y el análisis por superposición modal, que calcula la respuesta de la estructura a lo largo del tiempo mediante la superposición de los movimientos de los distintos modos de vibración de la estructura.

Por otro lado, el análisis estructural de tipo no lineal considera que la rigidez cambia durante el proceso de deformación, por lo que el valor de la rigidez de los elementos estructurales debe actualizarse debido a que el método de resolución no lineal progresa a través de un proceso de solución iterativa. Dentro del análisis no lineal cabe distinguir dos causas diferentes del comportamiento no lineal: la no linealidad geométrica, que considera que los cambios en la rigidez provienen de los cambios de forma de los elementos estructurales, y la no linealidad de material, que supone que los cambios en la rigidez ocurren a causa de cambios en las propiedades del material bajo condiciones operativas. Además, dentro del análisis no lineal se puede determinar la respuesta de una estructura ante cargas constantes mediante el análisis estático, o la respuesta de la misma a lo largo del tiempo ante la actuación de cargas dinámicas o variables mediante el análisis dinámico por medio de los métodos de integración en el tiempo.

2.4 Métodos matriciales para el análisis estático de estructuras

El gran avance tecnológico que ha experimentado la informática ha ofrecido un nuevo abanico de posibilidades en el mundo de la investigación, permitiendo abordar problemas extremadamente complejos de manera más sencilla. Este hecho ha permitido el desarrollo de procedimientos numéricos apropiados para su uso en computadores. Por ello, dentro del campo del análisis de estructuras, la incorporación de la computación ha conducido al desarrollo de métodos basados en el álgebra matricial [1].

El empleo de la notación matricial presenta diversas ventajas dentro del cálculo de estructuras. La primera de ellas es que permite, desde el punto de vista teórico, utilizar métodos de cálculo de una forma más compacta, precisa y al mismo tiempo completamente general. De este modo, los principios fundamentales del cálculo no se ven oscurecidos ni por las operaciones de cálculo ni por las diferencias geométricas en las topologías de las estructuras analizadas. La segunda ventaja consiste en que dicha notación proporciona, desde el punto de vista práctico, un sistema adecuado de análisis, determinando las bases idóneas para el desarrollo de algoritmos de cálculo.

Además, los métodos matriciales requieren gran cantidad de cálculo sistemático, por lo que el cálculo de estructuras por computador a través de dichos métodos resulta un problema fácilmente adaptable al entorno computacional. Por tanto, la aplicación de los métodos matriciales en entornos computacionales dentro del campo del análisis de

estructuras permite abordar estructuras grandes y complejas, donde los métodos manuales convencionales requieren un esfuerzo humano excesivo.

Asimismo, las operaciones del álgebra matricial son fácilmente programables en un computador, por lo que su aplicación requiere únicamente un conocimiento básico del cálculo matricial, o el uso de rutinas desarrolladas por especialistas que están, generalmente, incorporadas en los propios métodos de cálculo.

Los métodos de análisis estructural que se presentan en este apartado consisten en implementaciones del *Método de los Elementos Finitos* (MEF). Este método numérico permite aproximar soluciones de ecuaciones diferenciales parciales a través de sistemas de ecuaciones lineales mediante la división de medios continuos en un número elevado de subdominios no intersectables entre sí, llamados elementos finitos.

Los métodos de análisis estructural se basan en una serie de principios que coinciden con los establecidos para la Resistencia Clásica de Materiales. Algunos de estos principios pueden ser el de linealidad, que indica los desplazamientos y los esfuerzos de los elementos estructurales son funciones lineales de las cargas aplicadas; el de superposición, que establece que los esfuerzos producidos por cargas que actúan simultáneamente pueden obtenerse como la suma de los efectos producidos de manera aislada; el principio de equilibrio, que determina que tanto la estructura global como sus elementos están en equilibrio bajo la acción de las fuerzas exteriores e interiores que actúan sobre la misma; o el de unicidad de soluciones, que establece que no existen soluciones diferentes para los problemas de análisis estructural. Además, dichos métodos hacen uso de diferentes condiciones básicas que deben satisfacer los resultados, como son las condiciones de compatibilidad de deformaciones, que establecen que los movimientos en los extremos de barras deben ser compatibles con los de los nudos a los que están unidas dichas barras, y las condiciones de equilibrio de fuerzas, que indican que las fuerzas que actúan en los extremos de las barras deben mantener el equilibrio.

Como ha sido mencionado previamente, existen diferentes métodos o procedimientos para llevar a cabo el análisis de estructuras, donde la elección del más adecuado depende de aspectos como el tipo de problema a resolver, el nivel de aproximación requerido, la complejidad computacional que implica y la facilidad para detectar y corregir errores cometidos. Pero, previamente a su descripción, cabe indicar que existe una relación simple y única entre las deformaciones y los esfuerzos de la estructura, de modo que la obtención de un conjunto de resultados implica la determinación directa del otro.

La Figura 4 muestra una estructura aporticada de ejemplo.

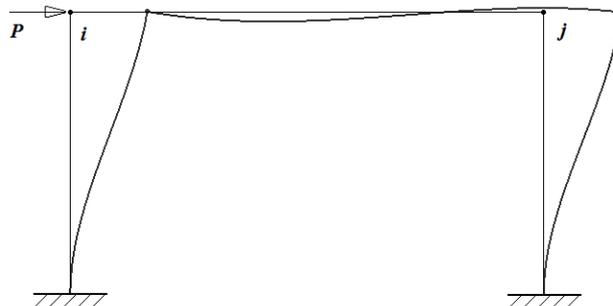


Figura 4: Modelo alámbrico de estructura plana de ejemplo.

Considerando dicha estructura, y suponiendo que han sido calculados los desplazamientos de los nudos de la misma, entonces puede determinarse la forma deformada de la estructura, así como los esfuerzos o tensiones en cualquier punto de dicha estructura, partir de las relaciones fuerza-desplazamiento. Por ejemplo, la siguiente relación determina los esfuerzos que tienen lugar en el extremo i de la barra horizontal de la estructura de ejemplo en función de los desplazamientos de los nudos i y j . En dicha relación, las submatrices K_{ii} , K_{ij} , K_{ji} y K_{jj} forman la matriz de rigidez de la barra que une los nudos i y j , y cuyos valores son funciones de la elasticidad del material y de la longitud, del área y de la inercia de su sección.

$$\begin{pmatrix} P_i \\ P_j \end{pmatrix} = \begin{pmatrix} K_{ii} & K_{ij} \\ K_{ji} & K_{jj} \end{pmatrix} \cdot \begin{pmatrix} d_i \\ d_j \end{pmatrix}$$

2.4.1 Análisis estático de estructuras

El Método de la Rigidez, también conocido como Método de las Deformaciones, es un procedimiento de análisis que determina los movimientos de los nudos que componen una estructura al aplicar sobre ellos una serie de cargas exteriores conocidas. Tras conocer el movimiento de dichos nudos de la estructura, se puede por tanto analizar por separado el comportamiento de cada uno de los elementos de la estructura y determinar las tensiones, los esfuerzos y las deformaciones que experimentan a través de las relaciones fuerza-desplazamiento.

El análisis mediante el Método de la Rigidez requiere la ejecución de una serie de etapas o pasos para determinar la respuesta de la estructura:

1. Generación de la matriz de rigidez K que expresa las reacciones de los nudos que delimitan cada barra o elemento finito en función de su movimiento.
2. Generación de la matriz de cargas externas p que establece las cargas exteriores que actúan sobre los nudos de la estructura.
3. Imposición de las condiciones de iniciales de cálculo relacionadas con los elementos de contorno de la estructura.
4. Determinación, mediante en el principio de equilibrio, de los resultados de desplazamientos d en los nudos mediante la resolución del sistema de ecuaciones lineales $K \cdot d = p$.
5. Cálculo de las fuerzas en los extremos de las barras a partir de las relaciones fuerza-desplazamiento.
6. Cálculo de los esfuerzos, las tensiones y las deformaciones en todos los puntos de la estructura.

2.4.2 Análisis dinámico de estructuras

El análisis dinámico de estructuras permite obtener el comportamiento de una estructura cuando actúan cargas dinámicas sobre el mismo. En este tipo de análisis, la respuesta de la estructura se caracteriza mediante un sistema de ecuaciones diferenciales parciales que expresan el movimiento de sus elementos estructurales.

A través de la semidiscretización del problema mediante el Método de Elementos Finitos, el comportamiento de la estructura a lo largo del tiempo se puede expresar como un sistema de ecuaciones diferenciales ordinarias de segundo orden de la forma

$M\ddot{d} + C\dot{d} + Kd = R$, donde M , C y K son respectivamente las matrices de masas, de amortiguamiento y de rigidez de la estructura, R es el vector de fuerzas externas aplicadas, y d , \dot{d} y \ddot{d} son respectivamente los vectores desplazamiento, velocidad y aceleración de la estructura. Además, cuando actúa un terremoto sobre la estructura, el vector de fuerzas que se aplica sobre la misma se obtiene mediante la expresión $R = -MI\ddot{x}_g$, donde I es un vector compuesto por unos y ceros y \ddot{x}_g indica la aceleración sísmica del suelo.

Existen diferentes técnicas que permiten resolver el sistema anterior de ecuaciones diferenciales en el tiempo, como son los métodos de integración directa o el análisis por superposición modal [3].

Los métodos de integración directa en el tiempo no realizan ninguna transformación de la ecuación de movimiento anterior, sino que la integran mediante un procedimiento numérico paso a paso, logrando la discretización del tiempo. De este modo, se asume la variación de los desplazamientos, velocidades y aceleraciones dentro del intervalo de tiempo Δt , aunque este hecho determinará aspectos como la precisión de los resultados, el coste computacional o la estabilidad del método concreto de integración empleado.

Existe una gran variedad de métodos de integración diferentes. Algunos de los más comunes son el método de las Diferencias Centradas, el método de Newmark o el método de Houbolt. Como ejemplo, el método de las Diferencias Centradas aproxima el valor de la velocidad y de la aceleración mediante las siguientes expresiones:

$${}^t\dot{d} = \frac{1}{2\Delta t} ({}^{t+\Delta t}d - {}^{t-\Delta t}d)$$

$${}^t\ddot{d} = \frac{1}{\Delta^2} ({}^{t+\Delta t}d - 2{}^td + {}^{t-\Delta t}d)$$

Como puede observarse, el valor de ${}^{t+\Delta t}d$ se puede deducir de las relaciones anteriores. De este modo, el error que se obtiene en ambas expresiones es del orden de $O(\Delta t^2)$. Sin embargo, este método de integración es condicionalmente estable, de modo que si el valor de Δt es inferior a un valor umbral que depende de la masa y la rigidez de la estructura a analizar, la respuesta del método es inestable.

El procedimiento de análisis dinámico es similar al proceso que se realiza mediante el Método de la Rigidez, donde la secuencia de etapas que requieren ambos métodos es prácticamente semejante. Las diferencias consisten en que, para determinar la respuesta de la estructura a lo largo del tiempo ante la actuación de cargas dinámicas se requiere, para cada paso de tiempo, la resolución del sistema de ecuaciones que representa el comportamiento de la estructura, expresado en función de las relaciones que determina el método de integración. Tras la resolución de dicho sistema de ecuaciones, la forma de obtener los resultados de esfuerzos de la estructura es similar a como se realiza en el método citado.

Como puede observarse, el coste computacional en el análisis dinámico aumenta en función del número de instantes de tiempo calculado. Por tanto, el tiempo de respuesta en un cálculo dinámico dependerá, además de la complejidad y del tamaño de la estructura a analizar, de la duración que se desee simular del terremoto o de las cargas dinámicas que actúan sobre dicha estructura, y de la precisión de la discretización del tiempo, o lo que es igual, del intervalo o incremento de tiempo entre cálculos consecutivos de la respuesta de la estructura.

3 Tecnologías de las Aplicaciones Distribuidas

El mundo de la computación, desde sus orígenes hasta la actualidad, ha sido partícipe de una constante evolución, experimentando continuos cambios en los modelos de programación y en las arquitecturas de los sistemas.

Antes de 1980, los ordenadores existentes, denominados *mainframes*, eran muy voluminosos y extremadamente caros. Por ello, las grandes empresas y organizaciones solían hacer uso de un conjunto de terminales interactivos, mucho más económicos, que se conectaban a dichos ordenadores e interactuaban con ellos por medio de comandos para hacer uso de sus recursos computacionales y de almacenamiento.

En cambio, a partir de los años 80, dos grandes avances tecnológicos iniciaron los cambios en el mundo de la computación. El primero de ellos consistió en el desarrollo de potentes microprocesadores, muchos de ellos con la capacidad computacional de los costosos *mainframes*, pero con un precio enormemente inferior. Este hecho dio lugar a la popularización de los ordenadores personales, que junto con las pobres prestaciones de las redes de comunicaciones en aquel momento, provocó un cambio en el desarrollo de software hacia arquitecturas monolíticas.

El segundo avance tecnológico fue el desarrollo de las redes de computación de altas prestaciones. Las redes de área local o *LANs* permitían interconectar cientos de máquinas dentro de un mismo edificio o zona e intercambiar información con velocidades y tiempos mucho menores que hasta aquel momento. Además, la popularidad que adquirieron las redes de interconexión o *WANs*, junto con la evolución de sus prestaciones, fomentó el desarrollo de nuevas tecnologías para el desarrollo de aplicaciones. Asimismo, el avance tecnológico en las redes de computación dio lugar a lo que hoy conocemos como Internet, o la gran red global de alcance mundial de redes heterogéneas interconectadas a través de la familia de protocolos TCP/IP.

Este conjunto de cambios tecnológicos provocó de nuevo un cambio en el desarrollo de software hacia arquitecturas no monolíticas de más de un nivel para el desarrollo de sistemas distribuidos. Los sistemas distribuidos son sistemas software donde un conjunto de componentes independientes se encuentran interconectados y ofrecen a los usuarios la imagen de un sistema único. Dichos sistemas se clasifican en diferentes tipos: los sistemas de información distribuidos, los sistemas de computación ubicua y los sistemas de computación distribuidos. Además de los sistemas de computación distribuidos se distinguen los sistemas cluster, los sistemas de computación Grid y más recientemente los sistemas de computación Cloud, siendo estos últimos objeto de estudio en el presente trabajo. Por ello, este capítulo analiza las diferentes arquitecturas y tecnologías que dan soporte a los sistemas distribuidos

3.1 Arquitecturas Distribuidas

La arquitectura de un sistema distribuido hace referencia a la forma en que se organizan y distribuyen los distintos componentes que integran el sistema, donde dichos componentes se encuentran interconectados y se comunican para coordinar acciones.

La clasificación de las diferentes arquitecturas distribuidas guarda una estrecha relación con los diferentes niveles conceptuales o capas que componen un sistema de información. Normalmente, los sistemas de información se diseñan alrededor de tres capas conceptuales: la capa de presentación, responsable de la comunicación con

entidades externas (sean usuarios u otras máquinas) y de presentar la información con el formato apropiado; la capa de lógica de negocio, quien lleva a cabo el procesamiento de datos como consecuencia de las operaciones solicitadas por medio de la capa de presentación; y la capa de datos, encargada del acceso a las diferentes fuentes de datos.

A continuación se presentan los diferentes tipos básicos de arquitecturas distribuidas en base a la combinación y distribución de las capas lógicas de los sistemas de información.

3.1.1 Arquitectura de un nivel

Tal y como muestra la Figura 5, la arquitectura de un nivel se corresponde con las aplicaciones monolíticas, donde las tres capas anteriores (presentación, lógica de negocio y datos) residen y se ejecutan en una única máquina, de modo que no ocurre distribución ni a nivel lógico ni a nivel físico.

Como ventajas, esta arquitectura permite una comunicación entre capas inmediata, sin necesidad de realizar transformaciones de datos para la comunicación entre las mismas, así como integrar optimizaciones de bajo nivel, dando lugar a sistemas extremadamente eficientes. En cambio, esta arquitectura presenta un inconveniente principal: no proporciona ningún punto de entrada desde el exterior, de modo que los sistemas se perciben como cajas negras que no permiten integrarse con otros sistemas.

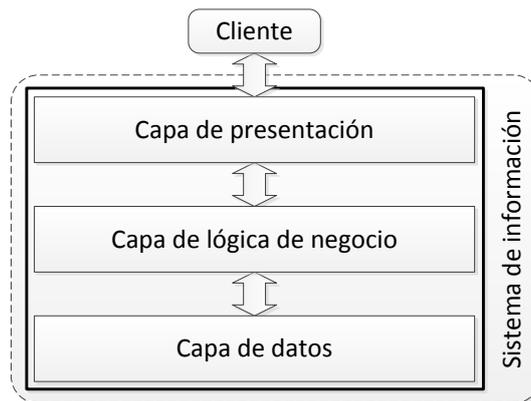


Figura 5: Arquitectura de un nivel.

3.1.2 Arquitectura de dos niveles

La arquitectura de dos niveles, obsérvese la Figura 6, traslada la capa de presentación a la máquina cliente, permitiendo liberar la carga computacional que requiere la gestión de dicha capa de la máquina servidora, y la interacción con distintos servidores por parte de un cliente. Esta arquitectura fomentó el desarrollo de interfaces de programación de aplicaciones (*Application Programming Interfaces* o *APIs*), definiendo la manera de interactuar con los servicios y estableciendo la antesala de la integración ente sistemas.

La arquitectura de dos niveles presenta numerosas ventajas. A parte de las ya mencionadas, conserva la ventaja principal de la arquitectura de un nivel: debido a que las capas de lógica de negocio y datos residen en la misma máquina, la interacción entre estos dos componentes se realiza de manera muy eficiente. Además, permite el desarrollo de sistemas portables entre diferentes plataformas a través del desarrollo de diferentes tipos de clientes para cada plataforma, pues la capa de presentación es

independiente del servidor. Por otro lado, el inconveniente principal de esta arquitectura consiste en que la escalabilidad suele ser muy limitada. Los servidores soportan un número limitado de clientes, de modo que su funcionamiento puede verse afectado o incluso interrumpido ante una carga que no pueden gestionar. Además, esta arquitectura no permite la integración de distintos sistemas, puesto que únicamente sería posible por medio del cliente, que obligaría a este último a conocer el API de todos los servicios con los que necesitara interactuar, siendo inmanejable conforme aumenta el número de servicios empleados.

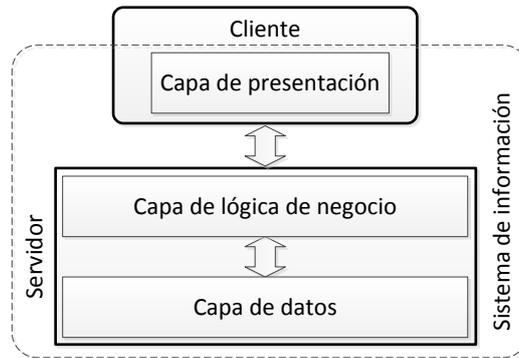


Figura 6: Arquitectura de dos niveles.

3.1.3 Arquitectura de tres niveles

La arquitectura de tres niveles presenta la solución a la integración entre diferentes servicios. Tanto a nivel lógico como físico, esta arquitectura define una clara separación entre cada una de las capas, tal y como muestra la Figura 7: la capa de presentación reside en el cliente, la capa de lógica de negocio reside en un nivel intermedio llamado *middleware* que permite la integración con la capa de datos, y la capa de datos comprende los servidores que la arquitectura permite integrar. De este modo, desde la perspectiva de los servidores de la capa de datos, el código o las funciones que se ejecutan en la capa de lógica de negocio y que acceden a los servidores de la capa de datos se comportan como meros clientes de manera similar a una arquitectura cliente/servidor.

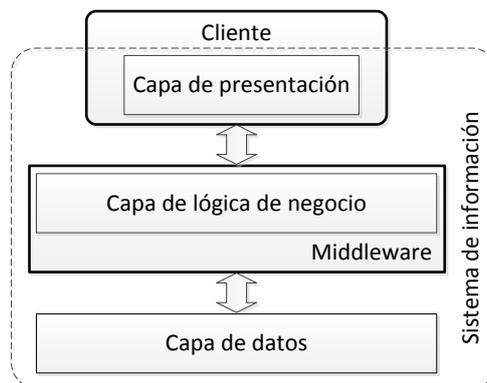


Figura 7: Arquitectura de tres niveles.

La principal ventaja que presenta esta arquitectura, además de permitir la integración de diferentes servicios, consiste en habilitar la distribución del nivel intermedio de la capa de lógica de negocio entre diferentes nodos o máquinas, solucionando los problemas de escalabilidad de la arquitectura de dos niveles. Por otro lado, algunos de los

inconvenientes que presenta esta arquitectura son la pérdida de prestaciones como consecuencia de la transformación de datos necesaria para la comunicación entre capas, así como la necesidad de emplear estándares bien definidos para la integración entre diferentes sistemas.

3.1.4 Arquitectura de N niveles

La arquitectura de N niveles no es más que el resultado de aplicar la flexibilidad que ofrece la arquitectura de tres capas, la cual permite la creación de aplicaciones flexibles mediante componentes reusables. Esta arquitectura tiene lugar mediante dos esquemas concretos. El primero de ellos consiste en una arquitectura donde la capa de datos puede incluir desde una simple base de datos a un sistema completo de dos o tres niveles. El segundo es un sistema donde la capa de presentación consta de un servidor Web completo empleado para el tratamiento y la conversión de la información procedente de la capa de lógica de negocio.

La principal ventaja de este tipo de arquitectura consiste en la modularidad de la misma, que facilita la replicación de componentes y distribución de capas a lo largo de múltiples máquinas para la obtención de alta disponibilidad y reducción de los tiempos de respuesta. En cambio, esta arquitectura modular presenta también un inconveniente importante: el incremento del número de niveles conlleva un aumento importante de la complejidad y los costes de desarrollo, configuración y mantenimiento del sistema.

3.1.5 Arquitectura P2P

La arquitectura *peer-to-peer* o P2P [4] emplea un modelo de aplicación distribuida totalmente diferente al modelo cliente/servidor de las arquitecturas anteriores. El modelo cliente/servidor establece una clara separación en dos roles diferentes, donde los servidores son proveedores de determinados servicios o recursos y los clientes se encargan de solicitarlos. En cambio, la arquitectura P2P no establece este tipo de distinción, ya que los nodos que forman la red son proveedores y consumidores de recursos, actuando simultáneamente como clientes y servidores. En este tipo de arquitecturas, también llamadas redes punto a punto o redes entre iguales, los nodos son equipotentes y tienen privilegios idénticos.

Dentro de la arquitectura P2P se establecen diferentes clasificaciones, por ejemplo en base a su grado de centralización, donde se pueden distinguir los sistemas centralizados, los cuales dependen plenamente de un nodo central; los sistemas mediados, que requieren un servidor central para localizar a otros nodos de la red; y los sistemas descentralizados, donde la ejecución de los nodos es completamente independiente. También se han establecido clasificaciones en base al tipo de recurso compartido, donde se distinguen sistemas de distribución de contenido, de uso de procesador, de mensajería, etcétera. Además, dentro de los sistemas descentralizados se definen diferentes arquitecturas en función del grado de descentralización, como son las arquitecturas puramente descentralizadas, donde todos los nodos tienen el mismo papel, y las arquitecturas parcialmente descentralizadas, donde algunos de los nodos realizan labores de coordinación. Otra clasificación importante de la arquitectura P2P se establece en base a la estructura de la red superpuesta que forman los nodos. Dentro de ella se distinguen los sistemas estructurados, donde se establece una correspondencia entre el recurso compartido y su ubicación en la red lógica, y los sistemas no estructurados, donde no existe esta relación.

Este tipo de arquitectura presenta numerosas ventajas frente a las arquitecturas cliente/servidor. Entre ellas, se puede destacar la alta escalabilidad y la robustez: la capacidad del sistema aumenta conforme se incorporan nuevos nodos a la red y la ocurrencia de fallos en alguno de los nodos no compromete ni daña el correcto funcionamiento de la red. Por el contrario, el esquema cliente/servidor suele presentar una escalabilidad limitada y la parada u ocurrencia de fallos en un número reducido de servidores puede provocar la caída del sistema. En cambio, las arquitecturas P2P también presentan diversos inconvenientes. Por ejemplo, mientras que los esquemas cliente/servidor ofrecen un mayor control sobre el acceso a los recursos a través de los servidores, proporcionando mayor control sobre el acceso a los mismos, el acceso descentralizado a través de las arquitecturas P2P introduce un mayor riesgo y da lugar a mayores vulnerabilidades debido a la falta de administración de los servicios. Además, la actualización de la información en las arquitecturas cliente/servidor es mucho más sencilla que en las arquitecturas P2P, donde los cambios o actualizaciones en la red deben ser propagadas por todos los nodos de la red, de modo que el tiempo de actualización del sistema es mucho más elevado, lo que en determinadas arquitecturas podría provocar inundaciones en la red.

3.2 Estado del Arte en las Tecnologías para el Desarrollo de Aplicaciones Distribuidas

El *middleware* es el componente de los sistemas informáticos que facilita la interacción entre aplicaciones en plataformas heterogéneas y permite integrar servicios bajo, una interfaz común, tanto en aplicaciones monolíticas como en entornos distribuidos.

A lo largo del tiempo han ido apareciendo y evolucionado diferentes tecnologías que implementan el componente *middleware* para adaptarse a los requisitos de los diferentes escenarios y arquitecturas software. Por ello, en este apartado se analiza la arquitectura y las características de los diferentes tipos de *middleware*, así como las diferentes plataformas o tecnologías más comunes que implementan dichos tipos de *middleware*.

3.2.1 Remote Procedure Call

El mecanismo de Llamada a Procedimiento Remoto (*Remote Procedure Call* o RPC) [5] es el tipo de *middleware* más básico. Tras surgir en los años 80, se presenta como un mecanismo para invocar procedimientos ubicados en otras máquinas. Curiosamente, este mecanismo se emplea como base en la mayoría de tipos de *middleware* y tiene un papel importante en gran parte de las implementaciones de *middleware* actuales, como RMI o SOAP.

El funcionamiento de RPC se basa en una metodología bien definida. Para la comunicación de aplicaciones mediante RPC, primero es necesario definir el interfaz del procedimiento. Para ello se emplea el Lenguaje de Definición de Interfaces (*Interface Definition Language* o IDL) que proporciona una representación abstracta del procedimiento en cuanto a la identificación del mismo y de sus parámetros de entrada y salida. A continuación se compila la definición IDL mediante el compilador asociado, generando los siguientes componentes:

- *Stub cliente o proxy*: Este código, que se compila junto con la aplicación cliente, ofrece un procedimiento con la signatura definida en el fichero IDL, siendo invocado localmente y encargándose de enmascarar la llamada al procedimiento

remoto. Cuando se invoca dicho procedimiento por parte del cliente, el stub se encarga de localizar al servidor, empaquetar y serializar los datos de manera apropiada, comunicarse con el servicio remoto, obtener la respuesta y devolverla como parámetro de salida.

- *Stub servidor*: Este componente es similar al stub cliente, pero implementa el procesamiento de las invocaciones remotas en el lado del servidor. Por ello, debe ser compilado junto con la aplicación del servidor. El stub servidor se encarga de atender las invocaciones del stub cliente, deserializar y desempaquetar los datos de entrada, invocar la implementación real del procedimiento y redirigir el resultado de la invocación al stub cliente.
- *Plantillas de código y referencias*: Además de los dos componentes anteriores, el compilador IDL se encarga de generar los ficheros auxiliares necesarios para la compilación (como por ejemplo, ficheros de cabecera) y las plantillas con el código básico del servidor, donde únicamente es necesario añadir el código que implementa la funcionalidad del procedimiento.

El esquema de funcionamiento de un mecanismo de Llamada a Procedimiento Remoto se muestra en la Figura 8.

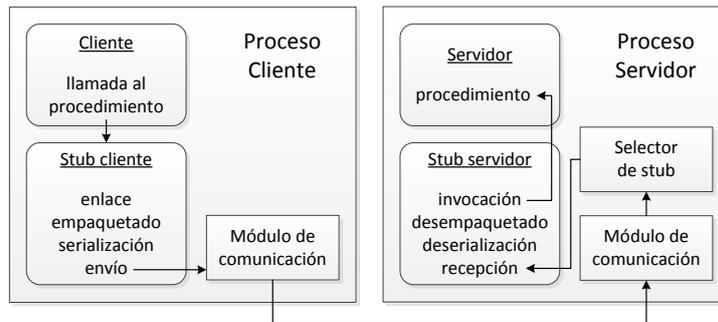


Figura 8: Funcionamiento básico de RPC.

RPC permite al usuario decidir el nivel de control sobre cómo se gestionan las invocaciones, generando interfaces con mayor nivel de configuración. Por ejemplo, RPC permitiría configurar detalles de gestión de errores en la red, del control de temporizadores y de retransmisiones de paquetes.

Antes de efectuar una Llamada a Procedimiento Remoto, el cliente necesita ubicar y enlazar con el servidor de procedimiento. Este proceso se denomina *binding* y puede ser de dos tipos: estático y dinámico. El binding o enlace estático es un mecanismo sencillo y eficiente: la ubicación del servidor del procedimiento remoto se especifica en el código fuente, de modo que no requiere ningún servicio adicional para localizar los servidores de procedimientos. Este tipo de enlace tiene en cambio importantes inconvenientes: los clientes y los servidores quedan fuertemente acoplados, de modo que un cambio en la ubicación del servidor requeriría recompilar el código del cliente. El binding o enlace dinámico, en cambio, emplea lo que denominan un *servicio de nombres y directorios* que proporciona la ubicación de los servidores de procedimientos en base a la signatura del procedimiento invocado. Este tipo de enlace, aunque introduce cierta sobrecarga como el registro de procedimientos y el descubrimiento de la ubicación de los mismos a través del servicio de nombres y directorios, se presenta como un mecanismo mucho más flexible que soluciona los problemas del enlace estático, soportando además otras características como por ejemplo balanceo de carga.

Distributed Computing Environment (DCE) es un ejemplo representativo de RPC de la Open Software Foundation (OSF), del cual existen diversas distribuciones comerciales por parte de IBM y HP. A día de hoy se sigue empleando en algunos middleware o tecnologías populares como DCOM (*Distributed Component Object Model*) y ODBC (*Open Data Base Connectivity*).

La plataforma DCE fue el resultado del intento de estandarización de RPC, e incluye diversos servicios adicionales que resultan muy importantes en el desarrollo de aplicaciones distribuidas: un servicio de nombres y directorios mejorado que proporciona creación y gestión de distintos dominios RPC; un servicio de tiempo que ofrece sincronización de relojes; un servicio que facilita la gestión multihilo y multiproceso; un servicio de ficheros distribuidos que permite a las aplicaciones compartir datos; y un servicio de seguridad que gestiona la autenticación y la protección de las operaciones dentro del entorno DCE.

3.2.2 Transaction Processing Monitor

Los Monitores de Procesamiento de Transacciones (*Transaction Processing Monitors* o TP Monitors) [6] son uno de los tipos de middleware más antiguos, siendo importantes por ser unos de los más eficientes y empleados a lo largo del tiempo. Son por ello el concepto básico de muchos de los middleware actuales.

TP Monitor ofrece como principal característica el soporte de ejecución de transacciones distribuidas. Este tipo de middleware tiene como base el RPC Transaccional, y su objetivo consiste en ofrecer un comportamiento transaccional a conjuntos de llamadas a procedimientos remotos que con un RPC convencional se tratarían como independientes y podrían originar comportamientos no deseados en determinadas arquitecturas, como pueden ser las de 3 o de N niveles.

El middleware TP Monitor, de manera análoga a los Sistemas de Gestión de Bases de Datos transaccionales, pretende garantizar al programador que todos los procedimientos invocados se han ejecutado si todos ellos han finalizado correctamente, y que ninguno de ellos se ha ejecutado si alguno ha abortado su ejecución. Para la gestión de transacciones, TP Monitor emplea las órdenes o instrucciones de Comienzo de Transacción (CDT) y Fin de Transacción (FDT). Es el módulo denominado *Gestor de Transacciones* quien se encarga de coordinar las interacciones entre clientes y servidores en el contexto de las transacciones, tratando todas las llamadas a procedimiento remoto entre estas dos instrucciones como una unidad.

Cuando un cliente ejecuta la instrucción CDT para iniciar una transacción, ver Figura 9, el stub cliente contacta con el Gestor de Transacciones para crear un nuevo contexto de transacción y obtener el identificador asociado a la misma. A continuación, el cliente lleva a cabo las invocaciones a procedimientos remotos, que llevan asociado el ID de transacción obtenido. Cuando los stub servidores reciben las invocaciones, notifican al Gestor de Transacciones su participación en dicha transacción, y procesan la invocación de manera similar a RPC. Finalmente, cuando el cliente ejecuta la instrucción FDT, el stub cliente notifica al Gestor de Transacciones, quien inicia el Protocolo Commit de Dos Fases (*Two Phase Commit* o 2PC) con los servidores que intervienen en dicha transacción para determinar el resultado de la misma. Cuando el protocolo finaliza, el Gestor de Transacciones notifica al stub cliente el resultado de la transacción, quien se encarga de devolver el resultado al código cliente.

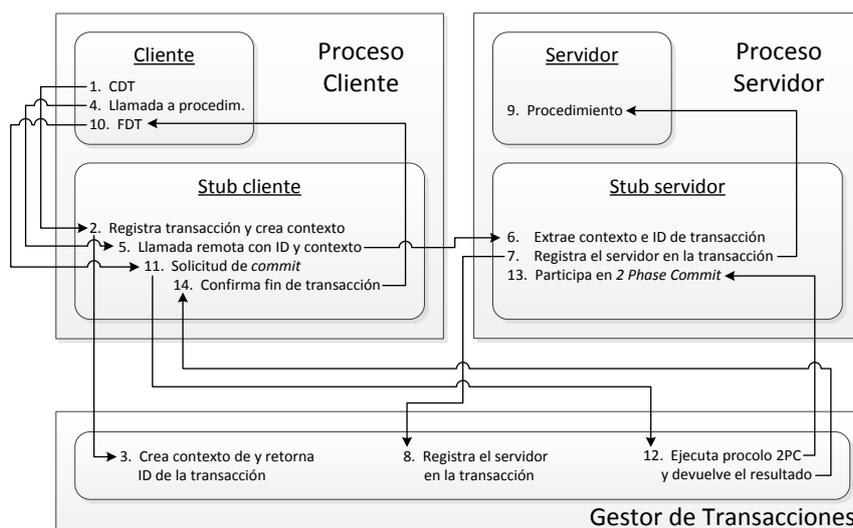


Figura 9: Funcionamiento de una llamada RPC transaccional.

TP Monitor proporciona las herramientas para desarrollar, ejecutar, gestionar y mantener sistemas de información transaccionales distribuidos. El middleware TP Monitor incluye principalmente:

- Todos los componentes mencionados que dan soporte a RPC (servicio de nombres y directorios, seguridad, stubs, etc).
- Abstracciones para la gestión de transacciones mediante las instrucciones CDT y FDT y mecanismos de callback.
- Un Gestor de Transacciones que ofrece el soporte transaccional así como herramientas de *logging*, recuperación, cerrojos, etc.
- Un Monitor que tiene como funciones la planificación de hilos, el balanceo de la carga, la gestión de prioridades, etc.

Además, TP Monitor soporta cargas elevadas de usuarios, llegando a gestionar cientos o incluso miles de clientes concurrentemente a través del uso de hilos y del balanceo de la carga entre diferentes máquinas del sistema.

Uno de los TP Monitor más antiguos, pero que aún sigue siendo clave en numerosas entidades financieras, administraciones y grandes empresas es el *Customer Information and Control System* (CICS) de IBM. Este TP Monitor de finales de los años 60 se ejecuta principalmente en mainframes de IBM, y se hizo popular debido a su capacidad de crear y gestionar hilos con un orden de magnitud más rápido que los sistemas operativos de aquella época. Actualmente, CICS ofrece soporte para invocación de transacciones vía HTTP, y puede actuar como proveedor o cliente de servicios Web incluyendo, entre otros, soporte para SOAP, WS-Atomic Transaction y Cifrado y Firma XML.

3.2.3 Object Request Broker

Object Request Broker (ORB) [7] es un tipo de infraestructura middleware que permite la interoperabilidad entre objetos. Este tipo de middleware surge al principio de los años 90 como evolución de RPC y tras la popularidad que adquirieron los lenguajes de programación orientados a objetos. El propósito de ORB es similar a RPC: ocultar la

complejidad de las invocaciones remotas, siendo en este caso invocaciones a métodos de objetos y proporcionar la misma apariencia que las invocaciones locales desde la perspectiva del programador. Además de la gestión de invocaciones a métodos de objetos remotos, ORB incluye otras características como son:

- La independencia del sistema operativo y soporte de numerosos lenguajes de programación.
- La posibilidad de interacción entre diferentes ORBs de diferentes tecnologías.
- La transparencia de ubicación y de distribución.
- La integración con software ya existente, incluso con sistemas heredados.
- Otras, tales como el enlace dinámico, la gestión del ciclo de vida de los objetos y la persistencia.

Asimismo, muchas de las implementaciones ORB incluyen características heredadas de RPC y TP Monitor, como el uso del lenguaje IDL para la descripción de invocaciones remotas, el soporte para transacciones distribuidas, el servicio de directorios, etcétera.

Existen numerosas implementaciones de ORB, algunas de ellas muy populares y con un uso muy extendido, como por ejemplo Java RMI o .NET Remoting. En cambio, probablemente el ORB más popular es la especificación *Common Object Request Broker Architecture* (CORBA). CORBA [8] es una arquitectura y una especificación para la creación y gestión de aplicaciones orientadas a objetos distribuidas en una red. CORBA está desarrollado por el *Object Management Group* (OMG), un consorcio formado por más de 800 empresas, y consiste en una especificación estándar en vez de una implementación concreta, de modo que es independiente del lenguaje de programación empleado en la aplicación y del sistema operativo sobre el que se ejecuta. CORBA fue el ORB más popular de mediados y finales de los años 90, donde numerosos distribuidores de software implementaron la especificación CORBA y donde multitud de aplicaciones fueron desarrolladas sobre esta arquitectura.

Los sistemas que cumplen la especificación CORBA están formados por tres componentes principales:

- **Object Request Broker:** Es el componente principal de la arquitectura y proporciona las funciones básicas de interoperabilidad entre objetos.
- **CORBA services:** Proporciona, a través de un API estándar, una serie de servicios necesarios para los objetos, como gestión del ciclo de vida y seguridad.
- **CORBA facilities:** Proporciona una serie de servicios de alto nivel necesarios para las aplicaciones, como la gestión de la documentación, la internacionalización o el soporte para agentes móviles.

El funcionamiento de CORBA es muy similar a como ocurre en RPC (ver Figura 10). Primero, se declara el interfaz de un objeto a través del lenguaje IDL para que los clientes puedan ser conscientes de los métodos que proporciona y, a continuación, se compila la definición para generar los stubs cliente y skeletons o stubs servidor asociados. A diferencia del IDL de RPC, el IDL de CORBA soporta conceptos de la programación orientada a objetos, como herencia y polimorfismo. Para la interacción entre clientes y servidores, únicamente es necesario conocer el interfaz IDL del servidor. En cambio, el cliente debe ser consciente de otros aspectos propios de las

aplicaciones que no están formalizados en CORBA, como la semántica de las invocaciones u otras restricciones que puedan imponer las aplicaciones.

Además de la interacción con servicios definidos a través del lenguaje IDL, CORBA permite descubrir de manera dinámica nuevos objetos y servicios, obtener su interfaz y construir invocaciones a los mismos durante la ejecución, sin necesidad de generar ni enlazar los stubs cliente previamente. Esta funcionalidad se ofrece a través dos componentes: el *interface repository* o repositorio de interfaces, que almacena las definiciones de todos los objetos conocidos por el ORB y el *dynamic invocation interface* o interfaz de invocaciones dinámicas que proporciona operaciones para explorar el repositorio y construir dinámicamente invocaciones basadas en nuevas interfaces descubiertas. Asimismo, para obtener las referencias a los objetos que ofrecen los servicios, son necesarios dos componentes adicionales: el *naming service*, que permite obtener referencias a objetos en base al nombre del servicio necesario; y el *trader service*, que ofrece a los clientes la búsqueda de servicios en base a unas propiedades o características especificadas. Aunque la capacidad de interactuar con nuevos servicios descubiertos de manera dinámica es una característica interesante y potente, realmente es una funcionalidad raramente empleada debido a su complejidad técnica (resulta más sencillo el acceso a través de los stubs) y semántica (comprender que un servicio realmente cumple con la funcionalidad esperada, así como interactuar con el servicio de la manera esperada por el mismo).

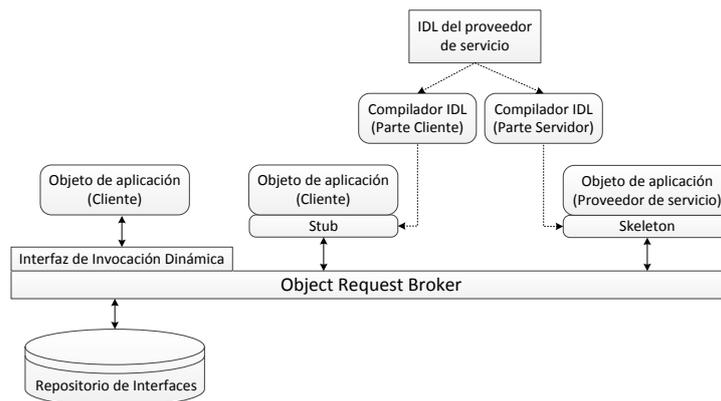


Figura 10: Enlace estático y dinámico de interfaces en CORBA.

Una de las características principales de CORBA es la encapsulación, que se proporciona de múltiples modos. Por ejemplo, CORBA ofrece independencia de ubicación, ofreciendo el referenciado de objetos mediante identificadores lógicos, sin relación a la dirección física del objeto llamado CorbaLoc, y permitiendo que un objeto servidor pueda cambiar de ubicación, e incluso de ORB. Además, dicho referenciado de objetos proporciona un mecanismo de identificación interpretable y editable por el ser humano. Por otro lado, CORBA también ofrece interoperabilidad entre ORBs a través de GIOP o *General Inter-ORB Protocol*. GIOP es un protocolo abstracto de comunicación que permite la interacción entre ORBs, e incluye otros protocolos de comunicación subyacentes, como Internet InterORB Protocol (IIOP) para su uso en Internet, SSL InterORB Protocol (SSLIOP) que proporciona autenticación y cifrado o HyperText InterORB Protocol (HTIOP) que funciona sobre HTTP.

3.2.4 Middleware orientados a mensajes

Los middleware orientados a mensajes (*message-oriented middleware* o MOM) [9] ofrecen, a diferencia de los tipos de middleware anteriores, una forma de comunicación asíncrona entre clientes y servidores o entre componentes del sistema distribuido. En cambio, cabe señalar que existen también versiones asíncronas tanto de RPC como de TP Monitor.

Este tipo de middleware se basa en la interoperabilidad a través de mensajes, donde clientes y proveedores de servicios se comunican mediante el intercambio de los mismos. Los mensajes son conjuntos de datos estructurados, normalmente caracterizados por un *tipo*, y un conjunto de pares clave-valor que constituyen los parámetros del mensaje. Actualmente, la mayoría de aplicaciones con comunicación basada en mensajes emplean tipos de datos XML.

Para que ocurra la comunicación por medio de mensajes, clientes y servidores deben coincidir en los tipos de mensaje empleados. Entonces, puede iniciarse la comunicación, donde las aplicaciones cliente envían mensajes a los proveedores de servicios, para solicitar dichos servicios, y los servidores procesan los mensajes, llevan a cabo las acciones apropiadas en función del contenido y envían mensajes de respuesta a los clientes con la información asociada. Como se observa, en los middleware orientados a mensajes, la distinción entre clientes y servidores es puramente conceptual, ya que no existen comportamientos distinguidos como en los tipos de middleware anteriores, y desde el punto de vista del middleware únicamente ocurre un intercambio de mensajes.

Las ventajas de los MOM frente a otros tipos de middleware se deben al uso de colas de mensajes, tal y como se muestra en la Figura 11. En el modelo de colas de mensajes, los mensajes enviados por los clientes se encolan en la cola específica del servidor, quien recupera dicho mensaje a través de funciones del MOM y se encarga de su procesamiento. Los mensajes permiten otras características como la inclusión de caducidad de los mismos, o prioridades. Además, si una aplicación que se ejecuta sobre MOM falla o es incapaz de procesar los mensajes que recibe, el middleware almacena dichos mensajes en la cola de la aplicación. Por tanto, MOM presenta un comportamiento más robusto ante fallos, permitiendo de un modo más sencillo diseñar arquitecturas tolerantes a fallos, e incluso incluir el balanceo de la carga permitiendo, a múltiples aplicaciones que proporcionan el mismo servicio procesar los mensajes de la misma cola.

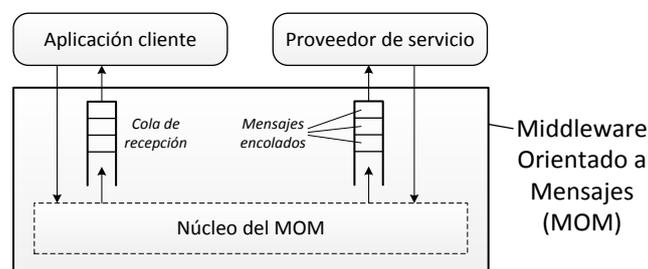


Figura 11: Modelo de colas de mensajes en MOM.

Algunos de los MOM más conocidos son IBM WebSphere MQ o Microsoft Message Queuing (MSMQ). Además, CORBA también proporciona su propio servicio de mensajes. MSMQ permite comunicar aplicaciones que se ejecutan en procesos o servidores distintos, de manera fiable. MSMQ garantiza fiabilidad en la entrega de mensajes entre máquinas diferentes ejecutando Windows, aunque éstas no se encuentren

siempre en ejecución: cuando un mensaje no alcanza su destino, MSMQ encola el mensaje y lo reenvía cuando el destino es alcanzable. Además, MSMQ soporta seguridad y gestión de prioridades en los mensajes. Asimismo, MSMQ ofrece soporte para transacciones, permitiendo agrupar múltiples operaciones sobre múltiples colas en una transacción, asegurando que se ejecuten o todas o ninguna de las operaciones.

3.2.5 Servicios web

Los servicios web (*Web services*) [10] son un tipo de tecnología desarrollada para la comunicación de aplicaciones ampliamente aceptada para la implementación de aplicaciones distribuidas. Su gran aceptación y uso se debe principalmente al gran desarrollo de la Web y al uso de sus estándares en la arquitectura y las comunicaciones de dicha tecnología.

El World Wide Web Consortium (W3C), la comunidad encargada del desarrollo de estándares para el crecimiento de la Web, ofrece la siguiente definición:

A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.

De la definición se puede extraer el protagonismo que presenta el uso de estándares abiertos. Esta característica aporta a las aplicaciones que emplean esta tecnología total interoperabilidad, independientemente de las propiedades de las aplicaciones o de las plataformas sobre las que se ejecuten. Además, gracias al uso de dichos estándares, los servicios web permiten combinar e integrar servicios de diferentes compañías de manera más sencilla, así como el desarrollo de componentes software más versátiles y reutilizables.

La arquitectura de los servicios web contiene cuatro niveles principales: el nivel de comunicaciones, que define el protocolo de transporte por el que se transmiten los mensajes; el nivel de mensajes, que indica el protocolo de intercambio de información de los mensajes; el nivel de descripción, que especifica la forma en que se describen los servicios web; y el nivel de procesos, que incluye otras características de nivel superior como descubrimiento, agregación etc.

En el nivel de transporte, los protocolos más comunes son los estándares web HTTP (*Hypertext Transfer Protocol*) y SMTP (*Simple Mail Transfer Protocol*), entre otros. En el nivel de transporte, comúnmente se emplea SOAP (*Simple Object Access Protocol*), aunque pueden emplearse otros protocolos. En el nivel de descripción se emplea normalmente WSDL (*Web Service Description Language*) y para el descubrimiento de servicios la tecnología más empleada es UDDI (*Universal Description Discovery and Integration*). Además, REST es una técnica de arquitectura software que surge en el año 2000 y ha adquirido una notable popularidad dentro del desarrollo de servicios web. A continuación se analizan con más detalle los conceptos fundamentales de SOAP, WSDL, UDDI y REST.

3.2.5.1 SOAP

Simple Object Access Protocol (SOAP) [11] es un protocolo estándar basado en el formato de la especificación XML para el intercambio de información estructurada en

servicios web. En su especificación, SOAP se define como un protocolo ligero para el intercambio de información en entornos distribuidos, que funciona sobre una variedad de protocolos subyacentes, diseñado para ser independiente de cualquier plataforma, modelo de programación y semántica de implementaciones específicas. Sus características principales son la simplicidad (se basa en el estándar conocido XML que almacena datos en un formato legible), además de ser extensible (permite incorporar nuevas características, como por ejemplo seguridad), neutral (se puede emplear sobre una gran variedad de protocolos de transporte), e independiente.

SOAP es un protocolo unidireccional y sin estado, diseñado y creado para soportar aplicaciones débilmente acopladas que se comunican mediante intercambio de mensajes. Los mensajes SOAP, ver Figura 12, tienen una estructura bien definida, donde un bloque principal o “envoltorio” (*envelope*) incluye toda la información del mensaje. Dentro del envoltorio se distinguen dos partes: una cabecera (*header*), cuya presencia es opcional, y un cuerpo (*body*) obligatorio. Ambas partes pueden contener además subpartes, formadas por conjuntos de elementos señalados mediante etiquetas.



Figura 12: Esquema y ejemplo de mensaje SOAP.

3.2.5.2 WSDL

Web Service Description Language (WSDL) [12] es un lenguaje de descripción de interfaces basado en el estándar XML que permite describir servicios de red como colecciones de *endpoints* capaces de intercambiar mensajes. Esta tecnología permite crear descripciones, interpretables por un ordenador, que indican cómo invocar un servicio, qué parámetros se esperan y qué estructuras de datos se devuelven.

WSDL fue creado originalmente por Microsoft, IBM y Ariba, y tiene como propósito principal proporcionar una forma unificada de describir servicios, así como las directrices para automatizar ciertos aspectos de la comunicación entre aplicaciones.

Como se puede observar en la Figura 13, la estructura de los interfaces WSDL está formada por una parte *abstracta* y una parte *concreta*. La parte abstracta es conceptualmente análoga al IDL y define la forma abstracta de interacción con el servicio. Ésta se produce mediante la definición de un conjunto de tipos de puerto (*port types*) o interfaces, un conjunto de *operaciones* de cada interfaz, un conjunto de mensajes que se intercambian en cada operación y un conjunto de *tipos* de definición de datos. La parte concreta, en cambio, define los protocolos y los formatos de datos de los interfaces así como los endpoints del servicio.

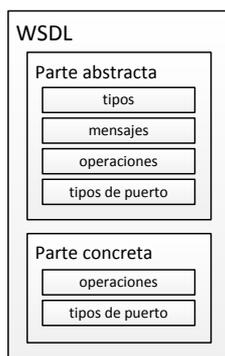


Figura 13: Esquema de especificación de servicio WSDL.

3.2.5.3 UDDI

Universal Description, Discovery and Integration (UDDI) [13] es una especificación abierta que permite descubrir negocios o proveedores de servicios web, así como registrar y dichos servicios. UDDI es análogo al concepto del servicio de nombres y directorios, siendo básicamente un registro de servicios web que gestiona información sobre proveedores, implementaciones y metadatos de los servicios. De este modo UDDI permite, a través de las APIs y las estructuras de datos que define, que los proveedores de servicios pueden anunciar los servicios que ofrecen y que los clientes pueden descubrir servicios que cubran sus necesidades, así como obtener la información necesaria para usar dichos servicios.

Los registros UDDI contienen tres entidades que incluyen diferente información, tal y como muestra la Figura 14:

- **businessEntity**: Similar al concepto de páginas blancas, incluye la información de contacto del proveedor de servicios web.
- **businessService**: Similar al concepto de páginas amarillas, describe un grupo de servicios clasificado que ofrece una entidad.
- **bindingTemplate**: Similar al concepto de páginas verdes, describe la manera de invocar los servicios web.

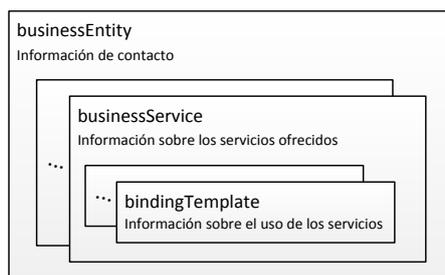


Figura 14: Esquema de entrada de registro UDDI.

3.2.5.4 REST

Representational State Transfer (REST) [14], a diferencia de los protocolos o tecnologías anteriores, es una técnica de arquitectura software para sistemas hipertexto distribuidos. Esta técnica de arquitectura, que trata de proporcionar una semántica de interfaz uniforme para el acceso a los recursos frente al uso de interfaces arbitrarias o

específicas de la aplicación, se ha convertido en la actualidad en el modelo de diseño de APIs web predominante.

Los objetivos principales que REST logra cumplir son:

- Escalabilidad de la interacción con los componentes, a través del uso de la Web.
- Generalidad de interfaces, mediante la interacción con los servidores a través del protocolo HTTP.
- Despliegue de componentes independiente, permitiendo interactuar a servidores antiguos con clientes nuevos y viceversa, gracias a la extensibilidad de HTTP.
- Compatibilidad con componentes intermedios para reducir la latencia, incrementar la seguridad y encapsular sistemas heredados. La Web es compatible con estos componentes como proxys, firewalls o cachés.

REST propone el uso de HTTP para la invocación de llamadas remotas entre diferentes máquinas, sin la necesidad de uso de otros middleware como CORBA o RPC. La gestión de los objetos o recursos se basa las operaciones HTTP POST, GET, PUT y DELETE como operaciones CRUD (Create, Read, Update, Delete) para el manejo de los recursos. De este modo, REST se presenta como una alternativa sencilla y más ligera que los servicios web basados en SOAP y WSDL.

4 Computación Cloud

Cloud computing, o computación en la Nube, es un término que describe una amplia colección de servicios. Como ha ocurrido con otras tecnologías, muchas empresas tratan de utilizar este término en sus productos y soluciones para otorgarles un carácter innovador, causando en numerosas ocasiones confusión o creando una falsa idea de lo que realmente significa dicho concepto. Por ello, resulta importante comprender qué es el Cloud computing, qué posibilidades o servicios ofrece y qué ventajas aporta sobre las tecnologías tradicionales.

El Cloud computing es un nuevo paradigma donde los recursos informáticos de bajo nivel (hardware) o alto nivel (software) se ofrecen como servicio, de modo que los usuarios pueden acceder y utilizar los mismos sin necesidad de preocuparse sobre su gestión o mantenimiento, mediante un modelo de pago por uso [15].

El concepto fundamental del Cloud computing ya se empleaba en la década de los 50, donde la capacidad computacional de las empresas o entidades se concentraba en lo que se denominan *mainframes* u ordenadores centrales, accesibles a través de *thin clients* o terminales sin capacidad computacional. De este modo, el acceso compartido al mainframe, a través de múltiples terminales, permitía compartir la capacidad computacional del ordenador central, reduciendo de este modo los periodos de inactividad e incrementando así el aprovechamiento de la inversión. Por otra parte, ya en 1961, en una ponencia en el centenario del MIT, el profesor John McCarthy presentó la idea de ofrecer los recursos computacionales, como por ejemplo capacidad de procesamiento o de almacenamiento, como si de un servicio básico se tratara (similar al suministro de electricidad, agua o gas) [16].

En cambio, han sido varios los factores decisivos para la aparición y actual adopción del Cloud computing, como son el progresivo incremento en las capacidades de las tecnologías de comunicación, el continuo aumento de las necesidades de capacidad computacional y, principalmente, el desarrollo y la evolución de Internet. Como consecuencia, la disponibilidad ubicua de redes de gran capacidad, de ordenadores y dispositivos de almacenamiento de bajo coste, así como la gran adopción de las tecnologías de virtualización y de arquitecturas orientadas a servicios han permitido el despegue de la computación en la Nube.

4.1 Definición de Cloud computing

En 2009, Peter Mell y Tim Grance, dos miembros del *National Institute of Standards and Technology* (NIST), acuñaron la definición de Cloud computing más ampliamente aceptada [17]:

Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable and reliable computing resources (e.g., networks, servers, storage, applications, services) that can be rapidly provisioned and released with minimal consumer management effort or service provider interaction.

Esta definición de Cloud computing pone énfasis en cinco características esenciales:

- *Autoservicio de aprovisionamiento de recursos bajo demanda:* Los usuarios o clientes deben poder ajustar sus necesidades de recursos computacionales de

manera unilateral, automática y sin intervención humana entre los clientes y los proveedores de servicios Cloud. Para facilitar la autogestión de los recursos, los proveedores de servicios Cloud deben ofrecer dicha gestión a través de una interfaz simple, intuitiva y efectiva. De este modo, la facilidad de autogestión de los recursos computacionales y la automatización del proceso de aprovisionamiento sin interacción humana constituye un ahorro tanto para los clientes como los proveedores de servicios Cloud.

- *Acceso de red general:* Los recursos o servicios alojados en el Cloud deben permitir el acceso a una gran variedad de dispositivos diferentes, como pueden ser ordenadores, tabletas electrónicas o teléfonos inteligentes, a través de protocolos de acceso estándar. Además, estos recursos o servicios deben estar interconectados mediante conexiones de gran ancho de banda y baja latencia.
- *Agrupamiento de recursos:* En el Cloud, los recursos computacionales deben ser agrupados para dar servicio a múltiples usuarios, siguiendo un modelo de compartición multiusuario. De este modo, los proveedores de Cloud consiguen, por un lado, ofrecer y reasignar dinámicamente recursos físicos y computacionales para cubrir las demandas de los usuarios y, por otro lado, aumentar y optimizar el aprovechamiento de los recursos computacionales físicos utilizados y desconectar aquellos recursos no utilizados. Este modelo ofrece una sensación de independencia en la ubicación de los recursos, donde los usuarios normalmente no pueden escoger la ubicación exacta a bajo nivel de sus recursos computacionales (por ejemplo, a nivel de cluster o de nodo de cómputo), pero sí pueden especificar ubicaciones de mayor nivel de abstracción, generalmente a nivel geográfico (país, región o incluso *datacenter* concreto).
- *Elasticidad rápida:* El Cloud debe permitir alojar o liberar recursos computacionales de manera rápida y eficiente, para cubrir las necesidades o requisitos de los usuarios. Además, los usuarios deben tener la sensación de que los recursos ofrecidos son ilimitados, es decir, la capacidad de aprovisionamiento parece ilimitada tanto en cantidad de recursos como en tiempo de uso de los mismos. Para que esta característica tenga sentido y pueda ser aprovechada, resulta necesario el desarrollo e implementación de servicios débilmente acoplados que escalen de manera independiente a otros servicios.
- *Servicio medido:* Los proveedores Cloud deben tener la capacidad de monitorizar y medir el uso de los recursos ofrecidos en el servicio. De este modo, pueden llevar a cabo un control y optimización automatizada de los recursos empleados, así como facturar a los clientes en base a los recursos consumidos, siguiendo un modelo de pago por uso.

Además de las cinco características anteriores, la definición de Cloud computing del NIST [18] incluye la distinción de tres modelos de servicio, así como la definición de cuatro modelos de despliegue diferentes. Estos modelos de servicio y de despliegue se analizan en los siguientes apartados.

4.2 Modelos de servicio Cloud

Los proveedores de la computación en la nube ofrecen diferentes tipos y niveles de servicio, abarcando desde servicios más básicos y de bajo nivel, como son el aprovisionamiento de recursos hardware y de red, hasta los servicios más complejos y

de alto nivel, como el aprovisionamiento de aplicaciones o soluciones software completas. De este modo, dentro del Cloud computing se definen diferentes modelos en función de los servicios o el tipo de recursos proporcionados por los proveedores.

De acuerdo a la Figura 15, los tres modelos principales de servicio que han sido ampliamente reconocidos son la Infraestructura como Servicio, la Plataforma como Servicio y el Software como Servicio [19]. Sin embargo, cabe añadir que, desde la aparición del Cloud computing y las clasificaciones iniciales de los modelos de servicio, ha habido una tendencia a incluir cualquier concepto o solución software como servicio [20] [21] [22], incluso partes fundamentales de alguno de los modelos de servicio principales como el Almacenamiento como Servicio o la Red como Servicio.

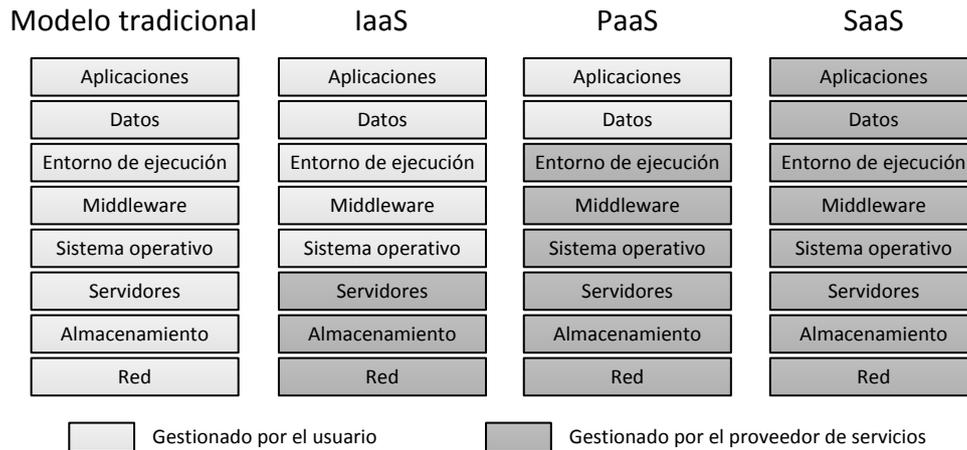


Figura 15: Responsabilidad de gestión en los diferentes modelos de servicio Cloud.

4.2.1 Infraestructura como Servicio (IaaS)

El modelo de Infraestructura como Servicio (*Infrastructure as a Service* o *IaaS*) ofrece aprovisionamiento de recursos hardware (bien físicos o más frecuentemente virtuales) de cómputo, de almacenamiento y de red, donde el usuario puede desplegar y ejecutar sus aplicaciones. En este modelo, el usuario alquila recursos computacionales en vez de comprarlos, instalarlos y mantenerlos como se ha realizado tradicionalmente. De este modo, el usuario evita la administración y el mantenimiento del hardware, tarea de la que se encarga el proveedor de servicios, pero en cambio posee el control sobre el sistema operativo y el software que se ejecuta sobre dicho hardware, con la misma libertad de configuración, gestión y manipulación como si de una máquina suya se tratara. Por tanto, este modelo ofrece la disposición bajo demanda de los requisitos hardware necesarios para la construcción de cualquier tipo de arquitectura software para cualquier tipo de aplicación.

Algunos ejemplos de Infraestructuras como Servicio son Rackspace Cloud Servers [23] o Amazon Elastic Compute Cloud (Amazon EC2) [24], siendo esta última la más común y empleada. Amazon EC2 es un servicio web que proporciona acceso a recursos computacionales a través de un interfaz Web interactivo, proporcionando además un completo API que ofrece la misma funcionalidad a través de sus interfaces REST y SOAP, aunque este último ha sido calificado como obsoleto y dejará de ofrecer servicio a partir de 2014. Los usuarios solicitan los recursos computacionales bajo petición, con un sistema operativo entre los múltiples que se ofrecen. Esto significa que, por sí mismo, el servicio proporcionado no es dinámicamente escalable, aunque Amazon posee gran variedad de servicios que complementan el servicio de EC2, como por

ejemplo servicios de autoescalado, balanceo de carga, almacenamiento, entrega de contenido, etc., que cubren estos aspectos e inclusive llegan a ofrecer características más propias de una Plataforma como Servicio.

4.2.2 Plataforma como Servicio (PaaS)

En el modelo de Plataforma como Servicio (*Platform as a Service* o *PaaS*), los proveedores de Cloud computing ofrecen una plataforma de cómputo y un conjunto de soluciones, a través de lenguajes de programación, librerías, servicios y herramientas soportadas por los proveedores, mediante las cuales los desarrolladores crean sus soluciones software que despliegan sobre dicha plataforma de cómputo. De este modo, los usuarios no se encargan de la gestión ni del control del hardware, ni del sistema operativo, cuya responsabilidad queda al cargo del proveedor de servicios. En cambio, poseen el control y la posibilidad de configurar las aplicaciones desplegadas sobre dicha plataforma.

Como puede observarse, este modelo ofrece el entorno para la gestión completa del ciclo de vida de las soluciones Cloud. Sin embargo, aunque este tipo de modelo facilita el desarrollo de aplicaciones sobre el Cloud, este modelo obliga al desarrollador a emplear un entorno de desarrollo y una pila de componentes software concreta. La solución software pasa a denominarse como *vendor lock-in* o dependiente del proveedor, siendo particular de un entorno concreto. Por tanto, un cambio de proveedor de Plataforma como Servicio implicaría cierto grado de reescritura de la aplicación desarrollada para su adaptación al entorno y al conjunto de componentes software del nuevo proveedor, lo que requeriría una nueva inversión de recursos económicos para la adaptación de dicha solución Cloud. Como consecuencia, ha surgido una nueva variedad de Plataforma como Servicio que ofrece el desarrollo de aplicaciones Cloud independientes del proveedor. Este nuevo tipo se conoce como *Open Platform as a Service*, el cual ofrece el mismo enfoque pero evita las restricciones de desarrollo dependiente del proveedor.

Algunos ejemplos de Plataforma como Servicio son Windows Azure [25], VENUS-C [26] o CodeCloud [27]. Windows Azure ofrece las herramientas para el desarrollo, implementación, despliegue y administración de aplicaciones en una infraestructura computacional propia de Microsoft. La gestión de los recursos y aplicaciones se realiza de manera similar a los servicios de Amazon, tanto de manera interactiva, a través de un portal Web, como por medio de un interfaz REST. Windows Azure ofrece tanto recursos computacionales como servicios de caching, gestión de redes, almacenamiento y bases de datos, mensajería, etcétera.

Por otro lado, VENUS-C es un proyecto europeo que ofrece los servicios y herramientas para el desarrollo o adaptación al Cloud de aplicaciones de carácter científico. VENUS-C no posee infraestructura propia, sino que emplea la infraestructura de Microsoft a través de Windows Azure o la del Barcelona Supercomputing Center, a través de COMPSs [28]. Inclusive, se proporcionan las herramientas para el despliegue de servicios Cloud en recursos hardware locales de manera sencilla. En el siguiente capítulo se describen, de manera más detallada, los servicios ofrecidos por estas dos plataformas.

Por último, CodeCloud es una arquitectura software que permite la creación y el uso de infraestructuras computacionales para la ejecución de aplicaciones o tareas mediante el aprovisionamiento de recursos de diferentes proveedores Cloud de Infraestructura como

Servicios. CodeCloud consta de diversos componentes que llevan a cabo diferentes labores como la descripción de la infraestructura computacional para la ejecución de la aplicación, la búsqueda de la imagen de máquina virtual más apropiada para la infraestructura a crear, el despliegue de máquinas virtuales en recursos locales o de proveedores cloud remotos, y la contextualización o instalación y configuración de las dependencias software para la ejecución de las aplicaciones. Además, ofrece acceso homogéneo a diferentes middleware Cloud, como OpenNebula [29] u OpenStack [30].

4.2.3 Software como Servicio (SaaS)

El modelo de Software como Servicio (*Software as a Service* o *SaaS*) ofrece el uso de aplicaciones que se ejecutan en una infraestructura Cloud. Estas aplicaciones son accesibles de manera global a través del navegador Web o un programa cliente. En este modelo de servicio, la única interacción de los usuarios con el servicio se realiza mediante el uso del software, de modo que el resto de aspectos de la arquitectura del servicio son abstraídos. De este modo, los usuarios no se tienen que preocupar de la gestión ni del mantenimiento de la infraestructura hardware, del sistema operativo, o de la aplicación que usan, puesto que toda esta labor queda al cargo del proveedor de servicios.

Por tanto, el modelo de Software como Servicio ofrece la infraestructura completa para el uso de soluciones software. En comparación con el modelo de ejecución de software tradicional, el coste de distribución y de mantenimiento del software es mucho menor, lo que normalmente implica que el coste de las aplicaciones SaaS sea muy inferior. Además, este tipo de aplicaciones posee una barrera de entrada mucho menor que las aplicaciones tradicionales, las cuales son adquiridas e instaladas por los usuarios. En cambio, como aspecto negativo del Software como Servicio, los usuarios dependen completamente de su conexión de red, de manera que un fallo en la conexión imposibilitaría el uso del servicio, mientras que, en el modelo tradicional, la ejecución de las aplicaciones únicamente depende del estado de la máquina local. Por ello, las aplicaciones más comunes y apropiadas para el Software como Servicio son las aquellas sin requisitos “a medida” (por ejemplo, aplicaciones de email), aplicaciones con una importante carga de acceso móvil o web (por ejemplo, aplicaciones de ventas), o aplicaciones cuya demanda es transitoria o cíclica (por ejemplo, aplicaciones de facturación).

Algunos ejemplos de software como servicio son Google Apps for Business [31] y Salesforce Sales Cloud [32]. Google Apps for Business proporciona los servicios tradicionales de email y aplicaciones de ofimática. Sales Cloud, por su parte, es la aplicación de ventas o *Customer Relationship Management* (CRM) más popular y pionera en la introducción del SaaS en el mundo de los negocios.

A modo de resumen, la Tabla 1 recoge las características principales de los tres modelos de servicio Cloud descritos.

<i>Características</i>	<i>IaaS</i>	<i>PaaS</i>	<i>SaaS</i>
Categoría de servicio	Máquinas, almacenamiento y red	Entorno de ejecución, servicios web, bases de datos	Software

<i>Características</i>	<i>IaaS</i>	<i>PaaS</i>	<i>SaaS</i>
Nivel de monitorización	Recursos físicos	Recursos lógicos	Aplicación
Optimización de recursos	Virtualización de recursos de servidores, red y almacenamiento	Sistemas de ficheros distribuidos a gran escala, bases de datos, etc.	Aplicaciones multi-usuario
Medición de uso	Uso de recursos físicos	Uso de recursos lógicos	Uso de la aplicación
Capacidad de gestión	Alta (nivel de Sistema Operativo)	Media (nivel de servicio)	Muy baja (nivel de software)
Ventajas	Mayor libertad y flexibilidad	Mayor facilidad de desarrollo y menor nivel de administración	Barrera de entrada muy pequeña, nivel de administración muy bajo
Inconvenientes	Administración completa del Sistema Operativo y mayor complejidad de desarrollo	Menor libertad, soluciones <i>vendor lock-in</i>	Dependencia del estado de la red

Tabla 1: Características principales de los modelos de servicio Cloud.

4.3 Modelos de despliegue Cloud

Los modelos de despliegue Cloud hacen referencia al conjunto de usuarios o clientes que es capaz de interactuar con una infraestructura Cloud determinada. Este concepto no guarda ninguna relación con el modelo de servicio, de modo que cualquiera de los modelos de servicio puede existir en cualquiera de los mencionados modelos de despliegue.

Aunque el uso de dichas infraestructuras esté restringido a determinados usuarios, según el modelo de despliegue, estos modelos deben mantener los principios fundamentales del Cloud computing. De este modo, los modelos deben proporcionar un escalado dinámico de recursos virtuales y pueden ser accedidos a través de una conexión de red, cuyos usuarios no tienen el control ni se encargan de la administración de la infraestructura que emplean.

El NIST define cuatro modelos de despliegue Cloud, que se describen a continuación: Cloud público, Cloud privado, Cloud comunitario y Cloud híbrido [18].

4.3.1 Cloud público

En el modelo de despliegue público, el aprovisionamiento de recursos se realiza para el uso abierto del público en general (cualquier usuario, entidad o empresa). En este modelo de despliegue, las infraestructuras Cloud pueden ser propiedad y

responsabilidad de una empresa, entidad académica u organización gubernamental, y se encuentran en las instalaciones del proveedor Cloud.

Típicamente, en el modelo de despliegue público, el Cloud es gestionado en un *datacenter* o centro de datos propiedad de un proveedor que da servicio a múltiples clientes y emplea aprovisionamiento dinámico (ver Figura 16). De este modo, la eliminación de los costes de administración y mantenimiento de los recursos informáticos, a través del uso del modelo de Cloud público, puede significar un ahorro económico para el sector de la investigación. En cambio, algunos aspectos como la seguridad de los datos, las políticas de empresa, temas legislativos o bien requisitos de rendimiento o de fiabilidad suponen una barrera actual que impide a parte de empresas del sector de las TIC migrar al uso de infraestructuras Cloud públicas.

Amazon EC2 [24], Windows Azure [25] y Google Drive [33] son ejemplos de Cloud público en los tres modelos diferentes de servicio: el primero de ellos como Infraestructura como Servicio, el segundo de ellos como Plataforma como Servicio, y el último de ellos como Software como Servicio.

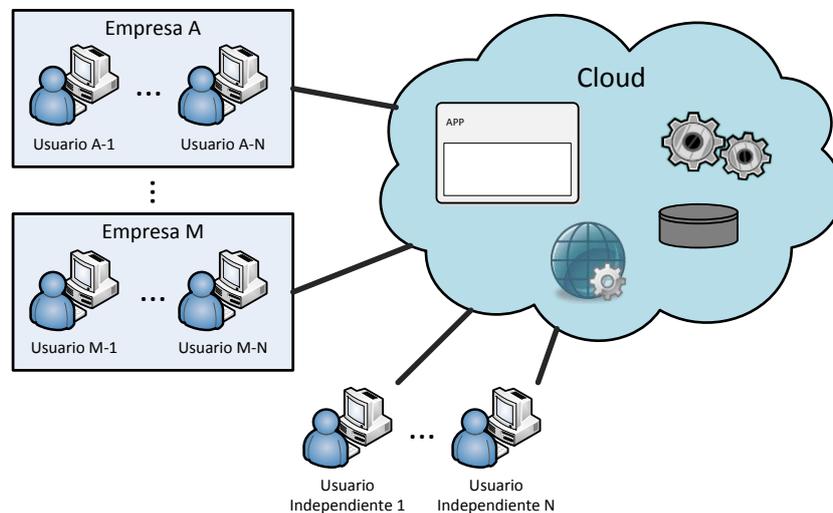


Figura 16: Representación de un Cloud público.

4.3.2 Cloud privado

En el modelo de despliegue privado, a diferencia del público, el aprovisionamiento de la infraestructura Cloud se realiza para uso exclusivo de una única organización que comprende múltiples clientes, como muestra la Figura 17. Dicha infraestructura puede ser propiedad y responsabilidad de dicha organización o de un tercero, y se puede encontrar en instalaciones locales (*on premises*) o externas (*off premises*).

De este modo, se considera que este modelo de despliegue ofrece un nivel mayor de seguridad que el modelo de Cloud público, en tanto en cuanto dicha organización es propietaria de la infraestructura Cloud y puede ejercer el control sobre qué aplicaciones son desplegadas y cómo son desplegadas en dicha infraestructura Cloud. Evidentemente, el modelo de despliegue privado no es inherentemente más seguro que el modelo de despliegue público si no se emplean las políticas de seguridad necesarias.

Algunos ejemplos de Cloud privado son VMware Private Cloud Computing [34] como modelo de Infraestructura como Servicio y ActiveState Stackato [35] como modelo de Plataforma como Servicio.

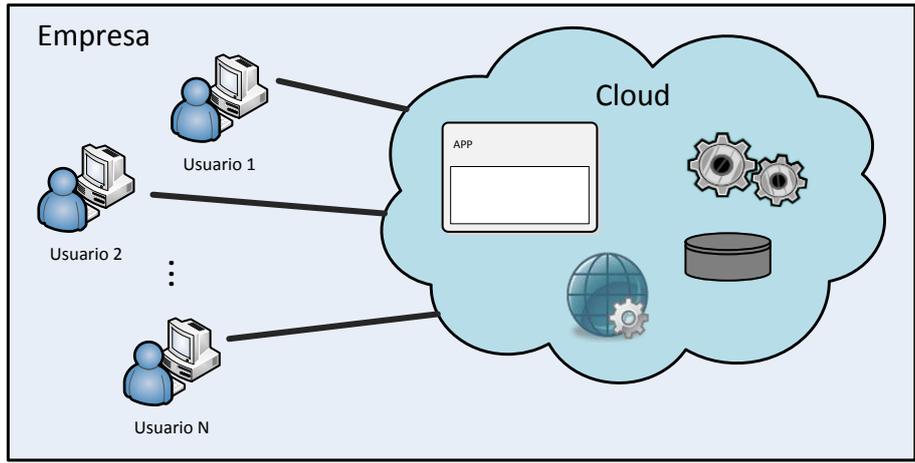


Figura 17: Representación de un Cloud privado.

4.3.3 Cloud comunitario

En el modelo de despliegue comunitario, ver Figura 18, el aprovisionamiento se realiza para uso exclusivo de una comunidad específica de clientes o usuarios de organizaciones que comparten requisitos (por ejemplo, propósito, políticas de seguridad). Dicha infraestructura puede ser propiedad y responsabilidad de una o más organizaciones de la comunidad o de un tercero, y puede se puede encontrar en instalaciones locales o externas.

Conceptualmente, el modelo de despliegue comunitario se ubica entre el modelo de despliegue público y el modelo privado. Este modelo de despliegue pretende obtener las ventajas que ofrecen los modelos de despliegue privado y público, es decir, un mayor control sobre la infraestructura, y por tanto mayor nivel de seguridad, con un esquema de compartición de los recursos acotado y una administración y un mantenimiento de los recursos compartido. En cambio, diferentes aspectos propios de la compartición de la infraestructura entre organizaciones, como pueden ser los cambios de propietario o responsabilidades en la administración de la misma, pueden complicar la gestión de despliegues comunitarios y afectar de manera directa a sus requisitos.

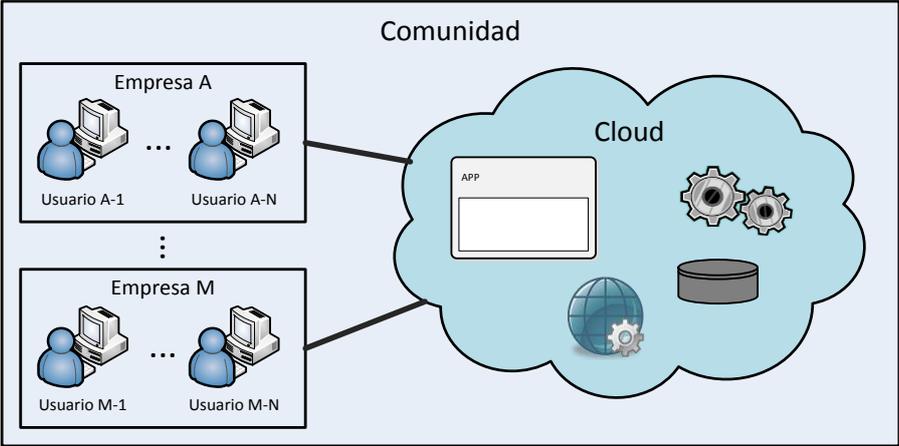


Figura 18: Representación de un Cloud comunitario.

4.3.4 Cloud híbrido

En el modelo de despliegue híbrido, obsérvese la Figura 19, la infraestructura Cloud es una composición de dos o más infraestructuras Cloud (privadas, comunitarias o públicas) que permanecen como entidades únicas pero se unen mediante tecnologías estándar o propietarias que permiten portabilidad de datos y de aplicaciones.

Este modelo de despliegue resulta útil en diferentes escenarios. Por ejemplo, una organización podría optar por desplegar y ofrecer aplicaciones software no críticas en un Cloud público, mientras sus aplicaciones críticas se despliegan en un Cloud privado, en sus instalaciones locales. En cambio, otros entornos híbridos aprovechan la naturaleza dinámica del Cloud computing para emplear lo que se conoce como *cloudbursting*, que se refiere al despliegue dinámico de una aplicación que, mientras se ejecuta predominantemente en un despliegue privado, puede ser también desplegada en un Cloud público para atender picos de demanda eventuales.

Como ejemplo de Cloud híbrido se incluye Rackspace Hybrid Cloud [36], que permite combinar en una misma arquitectura Clouds públicos, privados y hardware dedicado.

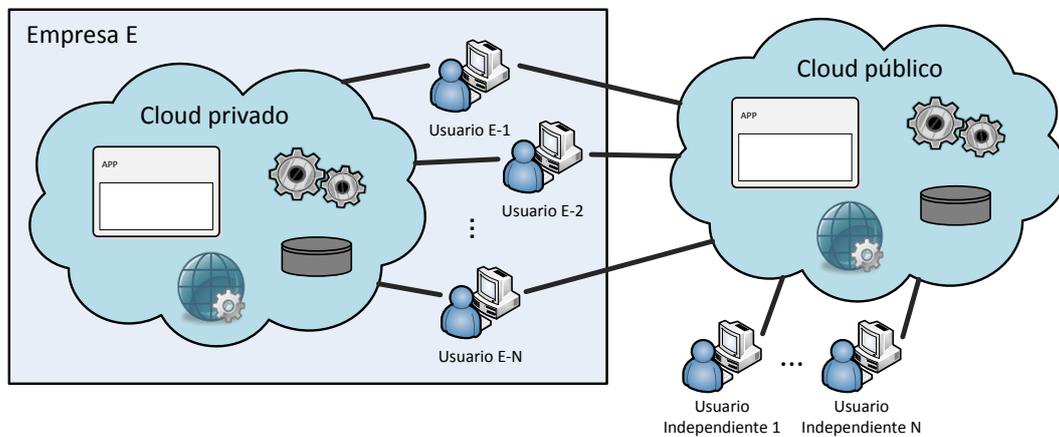


Figura 19: Representación de Cloud híbrido.

4.4 Tecnologías del Cloud computing

A continuación se describen brevemente algunas de las tecnologías software que son clave en el Cloud computing. Entre estas tecnologías se incluye la virtualización y las arquitecturas orientadas a servicios.

4.4.1 Virtualización

La virtualización es una tecnología que permite crear una simulación del hardware y del software de una máquina que se ejecuta sobre una máquina física concreta. Esta tecnología permite ocultar las características físicas de la máquina, de modo que únicamente proporciona un medio para crear, gestionar y hacer uso de los recursos virtualizados por medio de un elemento software denominado “hipervisor”.

La virtualización ofrece varias características que dan lugar a que se haya convertido en una tecnología imprescindible en el Cloud computing. Entre estas características destaca el particionado, que da soporte a múltiples aplicaciones y sistemas operativos en un mismo hardware; el aislamiento, de modo que recursos virtualizados permanecen aislados de la máquina física y del resto de recursos virtualizados; y la encapsulación,

donde los recursos virtualizados se representan y se almacenan como ficheros independientes.

La tecnología de virtualización proporciona un nuevo modelo de aprovisionamiento de recursos a través de la virtualización de los mismos, aportando homogeneidad de los recursos para los clientes, independencia de la plataforma física empleada, y seguridad y aislamiento entre diferentes recursos. De este modo, la virtualización garantiza numerosas ventajas, como son:

- Un mayor aprovechamiento de los recursos computacionales.
- Una mayor flexibilidad, en cuanto a la facilidad de realizar copias de respaldo, migrar, replicar o recuperar recursos virtualizados, etcétera.
- Tolerancia a fallos, en tanto en cuanto un fallo en un recurso virtual no afecta ni a la máquina física ni al resto de recursos virtualizados.

Existen diversas herramientas de aprovisionamiento de recursos virtuales que permiten la creación de entornos Cloud mediante virtualización, como por ejemplo OpenNebula [29] y EMOTIVE Cloud [37]. OpenNebula es un conjunto de herramientas *open-source* para la gestión de infraestructuras de centros de datos heterogéneos distribuidos. Este conjunto de herramientas permite gestionar la infraestructura de los centros de datos para la creación de Infraestructuras como Servicio de tipo pública, privada o híbrida. OpenNebula tiene como responsabilidad la orquestación del almacenamiento, la red, la monitorización y la seguridad para el despliegue máquinas virtuales en infraestructuras distribuidas. Además, OpenNebula soporta diferentes interfaces Cloud, como Amazon EC2 Query [38], OGF Open Cloud Computing Interface (OCCI) [39] y vCloud [40], así como diferentes hipervisores, como por ejemplo Xen [41], KVM [42] y VMware [43]. Por otro lado, EMOTIVE Cloud es un conjunto de herramientas que proporciona, a los usuarios, entornos virtualizados a través de los hipervisores Xen, KVM o VirtualBox [44]. EMOTIVE Cloud ofrece gestión de las máquinas virtuales con diferentes políticas de planificación. Además, tiene la capacidad de emplear recursos externos, permitiendo la creación de Clouds públicos, privados e híbridos.

4.4.2 Arquitecturas Orientadas a Servicios

Software Oriented Architecture o SOA es un tipo de arquitectura software para la creación de aplicaciones que implementan procesos de negocio mediante un conjunto de componentes débilmente acoplados, con interfaces bien definidas, que interactúan mediante intercambio de mensajes para ofrecer un servicio concreto. Además, las SOA incluyen un sistema que permite descubrir y utilizar nuevos servicios en base a ciertos metadatos.

Los patrones de diseño de las SOA se apoyan sobre una serie de principios:

- Reusabilidad, a través del diseño de componentes usables y evitando duplicidad de funciones u operaciones entre los mismos.
- Interoperabilidad, mediante el uso de tecnologías estándar.
- Granularidad, donde los diferentes componentes o servicios deben tener relevancia suficiente en el sistema.
- Modularidad, a través de componentes bien definidos que implementan un conjunto de operaciones sencillas.

- Abstracción, de modo que los interfaces de los servicios sean genéricos y oculten los detalles de la implementación subyacente.
- Composición, permitiendo el uso conjunto de diferentes componentes para la creación de servicios de complejidad creciente.

Aunque las SOA pueden implementarse mediante una gran variedad de tecnologías para el desarrollo de aplicaciones distribuidas, normalmente se apoyan en el uso de estándares para favorecer la interoperabilidad y reusabilidad de componentes. Por ello, las SOA suelen emplear estándares como SOAP, para el intercambio de mensajes, WSDL, para la descripción de interfaces, y UDDI, para la publicación y descubrimiento de servicios.

4.5 Beneficios del Cloud computing

El paradigma del Cloud computing ofrece una serie de ventajas a desarrolladores y usuarios sobre las formas tradicionales de ofrecer servicios software. Entre ellas, se encuentran las siguientes:

- *Flexibilidad y tolerancia a fallos:* Las implementaciones en Cloud ofrecen mayor flexibilidad, en tanto en cuanto permiten escoger el grado de libertad de configuración del sistema en función del modelo de servicio, así como capacidad de ajustar los recursos en base a las necesidades del servicio y de seleccionar las capacidades de los recursos computacionales empleados. Asimismo, ponen a disposición mecanismos sencillos para emplear redundancia e incrementar así la fiabilidad y robustez de los sistemas. Los mecanismos de recuperación ante fallos son además inherentes al modelo de Plataforma como Servicio.
- *Reducción de costes:* El Cloud computing ofrece un modelo de pago por uso, de modo que la facturación se realiza únicamente en base a los recursos consumidos. Este nuevo modelo puede suponer un ahorro tanto para nuevas empresas de perfil innovador (*start-ups*) como para empresas consolidadas: para las *start-ups*, el Cloud computing supone tener a disposición los recursos computacionales necesarios sin la necesidad de realizar grandes inversiones iniciales en infraestructuras computacionales, evitando así los costes derivados de las mismas, como los costes de personal de mantenimiento y los costes energéticos, reduciendo de esta manera las barreras financieras de entrada al negocio. En cuanto a las empresas consolidadas, éstas pueden poseer los recursos computacionales necesarios para atender sus niveles usuales de demanda, pero hacer uso de los recursos Cloud para atender picos de demanda transitorios y puntuales. Se evitan de esta manera realizar nuevas inversiones en infraestructura hardware para atender dichos picos de demanda y poseer recursos computacionales infrutilizados durante la mayor parte del tiempo.
- *Centralización del almacenamiento de datos:* El Cloud computing ofrece gran cantidad de recursos de almacenamiento que los usuarios, clientes o empresas tradicionalmente incorporaban en sus instalaciones locales y que ahora pueden solicitar, adaptar o liberar en función de sus necesidades, ajustando de este modo los costes operacionales. La centralización de la infraestructura de almacenamiento ofrece diferentes ventajas, como por ejemplo rentabilizar el coste en aspectos como bienes inmobiliarios, servicios básicos y personal de gestión, tanto para los proveedores Cloud, que ofrecen estos servicios de

almacenamiento, como para las empresas que hacen uso de los mismos, eliminando los costes de administración de las infraestructuras tradicionales. Además, en un sistema centralizado resulta más sencillo aplicar medidas de protección de datos y monitorizar y controlar el acceso a los mismos que en un conjunto de sistemas geográficamente distribuidos en diferentes partes de una misma organización.

- *Tiempo reducido de despliegue:* El Cloud computing permite, a costes relativamente bajos, poner a disposición gran cantidad de recursos computacionales y de almacenamiento en muy poco tiempo, sin necesidad de inversiones iniciales en hardware, software y personal de gestión y mantenimiento del mismo. De este modo, las empresas pueden comenzar a ofrecer sus servicios de manera mucho más rápida y sencilla reduciendo su tiempo salida al mercado.
- *Escalabilidad:* El Cloud computing ofrece los medios para que los clientes puedan aprovisionar o liberar recursos computacionales de manera rápida y sencilla, conforme aumenta o disminuye su demanda computacional. En cambio, aunque los proveedores Cloud deben ofrecer la apariencia de aprovisionamiento infinito de recursos, éstos deben establecer ciertos límites en el aprovisionamiento a los clientes para controlar demandas extremas de recursos y asegurar la suficiente capacidad para el resto de clientes Cloud.

4.6 La plataforma Windows Azure

Windows Azure [25] es una plataforma de computación en la Nube que se identifica como PaaS, dentro de la clasificación de modelos de servicio Cloud, y como Cloud público, dentro de los modelos de despliegue. Como plataforma, Windows Azure ofrece el entorno y las herramientas necesarias para llevar a cabo el diseño, la implementación, validación y puesta en producción de aplicaciones Cloud dentro de la infraestructura computacional de Microsoft, permitiendo gestionar de manera sencilla el ciclo de vida completo de las aplicaciones desarrolladas sobre esta plataforma.

La plataforma Windows Azure ofrece soporte a tres modelos de ejecución diferentes, además de proporcionar una gran cantidad de componentes o servicios para diferentes propósitos tales como la gestión de datos, de identidades, interconexión, mensajería, multimedia, etcétera. Además, Azure proporciona SDKs (*source development kits*) para diferentes lenguajes, como .NET, Java, PHP, Node.js y Python, además de un SDK de soporte básico. Dichos SDKs pueden integrarse en diferentes entornos de desarrollo como Visual Studio o Eclipse, facilitando y flexibilizando el desarrollo de aplicaciones para Windows Azure. La Figura 20 muestra los componentes o servicios que ofrece Windows Azure organizados en diferentes categorías.

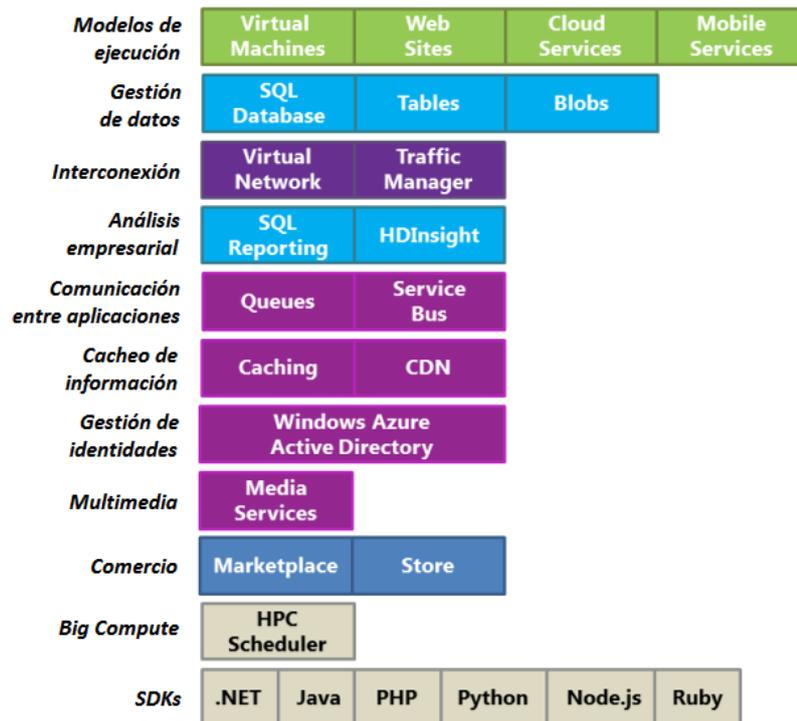


Figura 20: Servicios de Windows Azure.

A continuación se analizan brevemente los servicios ofrecidos por la plataforma Windows Azure [45]:

4.6.1 Modelos de ejecución

Windows Azure soporta tres modelos de ejecución diferentes:

- **Máquinas Virtuales:** Windows Azure Virtual Machines proporciona los servicios propios de las Infraestructuras como Servicio o IaaS, permitiendo crear máquinas virtuales a partir de una de las imágenes disponibles en el catálogo o a partir de imagen de una máquina virtual creada por el usuario, y con el tamaño de recursos computacionales deseado de entre cinco opciones disponibles.
- **Sitios Web:** Windows Azure Web Sites ofrece un entorno web autogestionado que emplea Internet Information Services (IIS). Este servicio permite ajustar dinámicamente el número de instancias que sirven los sitios web, llevando a cabo además la función de balanceo de la carga.
- **Servicios Cloud:** Windows Azure Cloud Services incluye las ventajas de los dos servicios anteriores, proporcionando la capacidad de administración y de instalación de software propio en las máquinas virtuales, así como las capacidades de ajuste dinámico de los recursos computacionales y gestión del balanceo de la carga. Además, estas instancias presentan capacidad de autogestión, encargándose de las labores de gestión e instalación de actualizaciones del sistema operativo. Las instancias en este modelo de ejecución pueden ser del tipo *Web Role* o *Worker Role*, donde las primeras incluyen la ejecución del servidor web IIS, para aceptar las peticiones de los usuarios, y las segundas se destinan a labores de procesamiento.

4.6.2 Gestión de datos

La plataforma Windows Azure ofrece tres opciones diferentes de almacenamiento, con diferente propósito: Base de datos SQL, Tablas y Blobs. Además, Windows Azure proporciona alta disponibilidad y tolerancia a fallos en sus servicios de almacenamiento mediante la replicación de la información almacenada en tres nodos diferentes de uno de los centros de datos. Además, permite acceder a la información almacenada tanto desde dentro de Azure como desde cualquier aplicación ejecutándose en cualquier dispositivo y lugar. A continuación se analizan los diferentes tipos de almacenamiento:

- Base de datos SQL: SQL Azure Database proporciona un sistema de almacenamiento relacional con soporte de consultas SQL ANSI. Además, este servicio se encarga de la gestión de la infraestructura hardware subyacente, así como de la actualización del software, liberando al usuario de las tareas administrativas.
- Tablas: Windows Azure Tables ofrece un sistema de almacenamiento no relacional para datos estructurados siguiendo una aproximación *clave-valor* (o *key-value store*). Este servicio permite almacenar conjuntos de propiedades de diferentes tipos, como información binaria, valores numéricos, cadenas de caracteres, fechas, etcétera, donde se accede a la información a través de una clave única para cada conjunto de propiedades. Además, se trata de sistema de almacenamiento masivamente escalable, siendo capaz de gestionar, de manera eficiente, tablas de millones de entidades de hasta un terabyte de tamaño.
- Blobs: Windows Azure Blobs almacena datos binarios no estructurados de hasta un terabyte de tamaño. Los blobs se organizan en contenedores de un único nivel de anidamiento (es decir, no permite crear subcontenedores dentro de un contenedor) que se pueden definir como públicos o privados. Además, a través de Windows Azure Drives, Azure ofrece almacenamiento persistente a las instancias en ejecución mediante sistemas de ficheros que se almacenan como un blob.

4.6.3 Interconexión

La plataforma ofrece tres alternativas de conexión con los recursos de Windows Azure:

- Redes Virtuales: Windows Azure Virtual Networks interconecta máquinas virtuales creadas en la red de Windows Azure con recursos de otras instalaciones a través de una VPN, pudiendo acceder a dichas máquinas como si se encontraran en la propia red local.
- Traffic Manager: Windows Azure Traffic Manager permite definir reglas para especificar cómo se deben encaminar las peticiones de los usuarios a las aplicaciones de Windows Azure, como por ejemplo dirigir las en base al centro de datos más cercano o redirigirlas en caso de superar un tiempo máximo.

4.6.4 Análisis empresarial

Windows Azure posee dos servicios diferentes para análisis empresarial:

- Generación de informes: SQL Reporting ofrece un servicio de generación de informes, de información almacenada en bases de datos SQL, para integrar tanto

en aplicaciones de Azure como aplicaciones en recursos locales. Soporta diferentes formatos, como HTML, XML o PDF.

- **HDInsight:** Windows Azure proporciona un servicio basado en Apache Hadoop para problemas de “Big Data” o gestión y análisis de grandes volúmenes de datos. Su funcionamiento se basa en la distribución de la información entre múltiples máquinas virtuales y la propagación de la lógica de los trabajos de tipo MapReduce sobre dichas máquinas para su procesamiento en paralelo.

4.6.5 Comunicación entre aplicaciones

Azure posee dos servicios para comunicar aplicaciones:

- **Colas:** Windows Azure Queues ofrece un servicio donde un conjunto de aplicaciones o componentes de una aplicación pueden encolar y desencolar mensajes de hasta 64KB, ofreciendo persistencia y acceso concurrente.
- **Service Bus:** Mediante este servicio, las aplicaciones se comunican a través de un sistema de colas, pero siguiendo un modelo de *publicación y suscripción* que permite realizar difusiones a través del canal de suscripción. Además, permite a las aplicaciones comunicarse directamente a través de firewalls.

4.6.6 Cacheado de información

La plataforma Windows Azure proporciona los siguientes servicios de *caching* o cacheado de información:

- **Caching:** Azure Caching permite hospedar cachés de información tanto en las instancias donde se encuentran desplegadas las aplicaciones (topología colocalizada) como en máquinas virtuales dedicadas (topología dedicada).
- **CDN:** Azure CDN emplea una red de entrega de contenidos (*content delivery network*) para conservar copias de blobs o contenidos accedidos cerca de los usuarios, obteniendo una menor latencia y mayor ancho de banda.

4.6.7 Otros servicios

Windows Azure ofrece, además de los previamente descritos, los siguientes servicios:

- **Gestión de identidades:** Azure Identity incluye dos servicios para la gestión de los usuarios: un directorio activo (Azure Active Directory) con mecanismos de registro y autenticación en aplicaciones SaaS y un servicio de control de acceso (Azure Active Directory Access Control) para autenticar usuarios empleando identidades de otros proveedores.
- **High-Performance Computing:** Azure HPC Scheduler da soporte para aplicaciones HPC que emplean el estándar MPI (Message Passing Interface), permitiendo comunicar los procesos de una aplicación que se ejecutan simultáneamente sobre diferentes máquinas virtuales de Azure.
- **Multimedia:** Azure Media Services ofrece una completa gestión de contenidos multimedia mediante un modelo de pago por uso, permitiendo la codificación y conversión entre formatos, protección de contenidos, inserción de publicidad y distribución vía *streaming*.

- **Comercio:** El Azure Marketplace y el Azure Store permiten a los desarrolladores poner a disposición tanto aplicaciones SaaS como *datasets* con distintos tipos de información (demográfica, financiera, etc.), los cuales podrán ser empleados por los usuarios, previo pago.c

4.7 El proyecto VENUS-C

VENUS-C (Virtual multidisciplinary EnviroNments USing Cloud Infrastructures) [26] es un proyecto financiado por el 7º Programa Marco de la Comisión Europea donde proveedores Cloud y comunidades científicas han colaborado para el desarrollo de una plataforma de computación Cloud orientada a la ciencia, la investigación y las empresas de carácter innovador, lo que comúnmente se denomina como “Cloud científico” o *scientific Cloud*.

El objetivo principal del proyecto VENUS-C consiste en el desarrollo de una plataforma Cloud basada en virtualización y orientada a servicios, de fácil uso, que dé soporte a los requisitos de las aplicaciones científicas y que permita a los investigadores y PyMEs el desarrollo, validación y despliegue de infraestructuras Cloud flexibles y altamente escalables de manera sencilla. Dicho proyecto ha contado con los recursos computacionales de la infraestructura de Windows Azure de Microsoft, del Centro de Supercomputación de Barcelona (BSC), del Royal Institute of Technology (KTH) y de la empresa Engineering Group. La plataforma VENUS-C ofrece sus servicios por medio de los componentes Generic Worker [46] y COMPSs [28], aunque el capítulo actual se centra en el estudio de la solución sobre Generic Worker, la cual se basa a su vez sobre Windows Azure.

Dentro del desarrollo del proyecto, VENUS-C contempló 7 escenarios de diferentes áreas de investigación y necesidades computacionales, como por ejemplo biomedicina, arquitectura e ingeniería civil y protección civil y emergencias, para el estudio de los componentes y requisitos necesarios para dar soporte a los despliegues Cloud en ámbitos de investigación. Además, para la validación de la arquitectura y la funcionalidad de la plataforma, VENUS-C se apoyó de 15 pilotos y 5 experimentos que abarcan otras áreas como biología, química, ciencias de la tierra, matemáticas o medicina.

La plataforma VENUS-C se asocia al concepto de PaaS, donde se prepara, instala y ejecuta una aplicación en una máquina virtual con un entorno de ejecución apropiado para ofrecer un servicio. Como ya se mencionó previamente, el desarrollador no necesita mantener el sistema operativo o el entorno de ejecución de la máquina virtual en este modelo computación, pudiendo centrarse únicamente en la funcionalidad de su propia aplicación. Además, a través de este modelo, se puede proporcionar a dicha aplicación de manera sencilla escalabilidad, seguridad y alta disponibilidad. Otras plataformas como Windows Azure o Google App Engine proporcionan estas mismas características. En cambio, esta iniciativa presenta dos importantes novedades sobre las plataformas de computación Cloud comunes: la facilidad de portar una aplicación existente al Cloud y la interoperabilidad entre diferentes infraestructuras sin la necesidad de tener que adaptar dicha aplicación. En este sentido, VENUS-C expone un interfaz de servicio web que cumple con el estándar OGF BES/JSDL [47], junto con librerías para Java y .NET para el uso de la plataforma.

La plataforma VENUS-C integra diversos componentes dentro de su arquitectura que proporcionan diferentes servicios, como se observa en la Figura 21. Los dos

componentes principales son el de Gestión de Datos o *Data Management*, encargado de la transferencia de los datos de entrada al Cloud y de la recuperación de los resultados de ejecución, y el Servicio de Lanzamiento (*Programming Model Enactment Service* o PMES), que permite a los usuarios asignar recursos computacionales en el Cloud, enviar trabajos y gestionar su ciclo de vida, monitorizar su estado o cancelar trabajos. Además de los dos servicios mencionados, VENUS-C proporciona servicios de Monitorización (*Monitoring*), Contabilidad (*Accounting*) de uso de recursos, Facturación (*Billing*) y Eliminación de Tráfico Redundante (*Traffic Redundancy Elimination* o TRE). Estos servicios que componen la plataforma VENUS-C se describen en los siguientes apartados.

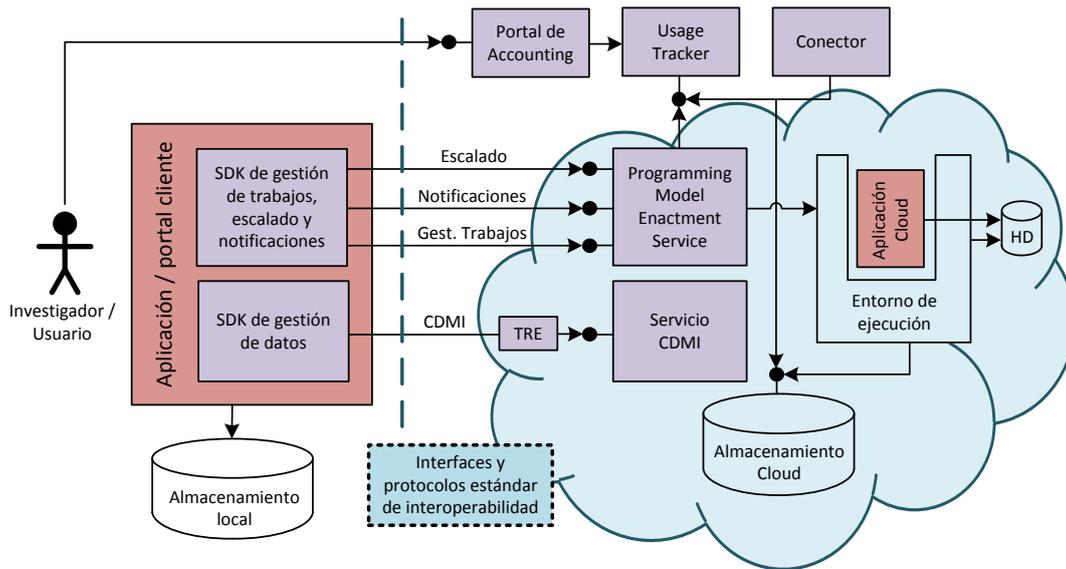


Figura 21: Arquitectura de VENUS-C.

4.7.1 Servicio de Gestión de Trabajos

Dentro del contexto de VENUS-C se han desarrollado dos componentes que ofrecen la funcionalidad para el lanzamiento y la gestión de trabajos [48], pero que emplean un modelo de programación diferente: el Generic Worker [46], del Microsoft Research, y el PMES-COMPSs [49], del Barcelona Supercomputing Center (BSC). El presente apartado describe con mayor detalle el Generic Worker, el cuál ha sido empleado para la implementación del servicio de cálculo estructural, objeto de esta Tesis de Máster. Este componente forma parte de la arquitectura del servicio, el cual se ha puesto en producción sobre la plataforma Windows Azure y se ejecuta sobre máquinas virtuales basadas en el sistema operativo Windows Server 2008 y .NET Framework.

4.7.1.1 El componente Generic Worker

El Generic Worker [46] es el componente que ejerce como Gestor de Trabajos en despliegues Cloud sobre la plataforma Windows Azure. La implementación de este componente está basada en un Web Role de Azure que implementa el API y la funcionalidad de PMES y se encarga de atender peticiones de registro de nuevos trabajos, de atender la solicitud del estado de un trabajo concreto, de la cancelación de trabajos, etcétera. Además, su implementación mediante Web Role incluye un proceso interno que comprueba si existen nuevos trabajos a procesar y ejecutar dichas tareas. Debido a su naturaleza, los servicios basados en el Generic Worker permiten el

despliegue de múltiples instancias de este Web Role para trabajar de manera coordinada.

Para el uso del Generic Worker se han desarrollado herramientas de integración para que los desarrolladores puedan migrar sus aplicaciones a Windows Azure de manera sencilla. La diferencia respecto a un servicio Cloud convencional sobre Windows Azure consiste en que los servicios basados en el Generic Worker ejecutan trabajos “genéricos”, es decir, trabajos acompañados de una descripción que especifica la aplicación a ejecutar, los ficheros de entrada y de salida asociados a su ejecución y los parámetros de invocación al fichero ejecutable.

Cuando una instancia de un despliegue Cloud basado en el Generic Worker atiende y ejecuta una petición de trabajo, realiza la siguiente secuencia de acciones:

1. Descarga, del almacenamiento Cloud, un paquete de aplicación que contiene el ejecutable a lanzar y una plantilla de descripción de dicha aplicación.
2. Descarga, del almacenamiento citado, los ficheros de entrada de la aplicación.
3. Lanza la aplicación con los parámetros de invocación indicados en la descripción del trabajo y modifica el estado del trabajo a “running”.
4. Cuando la ejecución de la aplicación finaliza correctamente, copia los ficheros de resultados en el almacenamiento Cloud y modifica el estado del trabajo a “finished”. Si falla la ejecución, modifica el estado del trabajo a “failed”.

Cuando se selecciona un trabajo, si los ficheros de entrada no están disponibles en el lugar indicado, el estado del trabajo pasa a “pending” y se selecciona un nuevo trabajo. De este modo, el Generic Worker permite la ejecución de trabajos interdependientes siguiendo un modelo dirigido por datos (o *data-driven model*). En cuanto a la subida o descarga de datos, el Generic Worker soporta almacenamiento en blobs tanto del estándar CDMI [50] como del propio de Windows Azure.

Las interacciones con el Generic Worker pueden ser autenticadas mediante un mecanismo de usuario y contraseña, y controladas mediante políticas de autorización de acceso. Estas políticas se basan en la definición de un conjunto de roles de usuario y una tabla que establece qué roles posee cada uno de los usuarios del servicio. Además, el Generic Worker hace uso del *Security Token Service* [51], que consiste en un servicio que protege las comunicaciones mediante el uso de WS-Security por medio de *tokens* de seguridad.

El Generic Worker expone también un interfaz adicional al rol de administrador a través del cual puede modificar el tamaño del despliegue Cloud. Este interfaz permite aumentar o disminuir el número de instancias en ejecución, permitiendo así asegurar la demanda de recursos computacionales de los usuarios. De este modo, a través de este interfaz, las decisiones y las acciones de escalado del servicio se podrían automatizar de acuerdo a diferentes criterios. Además, incluye también un interfaz de notificaciones que permite crear subscripciones a trabajos ejecutados sobre la plataforma VENUS-C, a través de las cuales se puede conocer el estado interno de la ejecución de un trabajo, así como el progreso de su ejecución, los mensajes generados durante la misma, etcétera.

4.7.1.2 El componente PMES-COMPSs

PMES-COMPSs [49] es el componente del Barcelona Supercomputing Center que ejerce como Gestor de Trabajos en despliegues Cloud sobre las plataformas EMOTIVE [37] y OpenNebula [29]. Este componente emplea un modelo de programación diferente

al del Generic Worker. Así, por ejemplo, el entorno de ejecución de COMPSs tiene la capacidad, tanto de manera automática en función de la carga del sistema como en base a los requisitos especificados para una tarea, de crear dinámicamente un conjunto de máquinas virtuales para el procesamiento de trabajos.

Este servicio se compone de un módulo Gestor de Trabajos o *Job Manager* y un módulo Procesador de Trabajos o *Job Dispatcher*, encargados de la ejecución del trabajo y la gestión de su ciclo de vida. Su mecánica de ejecución de trabajos es diferente a la empleada por el Generic Worker. Cuando se envía un nuevo trabajo al servicio, se reenvía la descripción del trabajo al Job Manager, quien se encarga de crear y encolar dicho trabajo en el Job Dispatcher. Entonces, se solicita una máquina virtual al proveedor Cloud para desplegar el entorno de COMPSs, la aplicación y las dependencias sobre dicha máquina. Cuando finaliza el despliegue del entorno de ejecución de COMPSs, se ejecutan los trabajos en un conjunto de máquinas creadas bajo petición. Finalmente, cuando se completa la ejecución del trabajo, los ficheros de salida indicados en la descripción del trabajo se suben al servicio de almacenamiento Cloud especificado y se detiene el entorno de ejecución de COMPSs.

4.7.2 Gestión de los datos

El componente Data Manager [52] proporciona un API de acceso unificado a diferentes servicios de almacenamiento de datos a través del protocolo estándar CDMI (*Cloud Storage Management Interface*) [50] de la SNIA (*Storage Networking Industry Association*). Este componente está formado por el módulo de almacenamiento, que consiste en un servicio que cumple con el estándar CDMI a través de un interfaz REST para el acceso a datos, y un conjunto de librerías de acceso a dicho servicio.

El módulo de almacenamiento soporta un subconjunto de operaciones de la especificación CDMI que permiten la gestión de contenedores y de datos o *blobs* mediante operaciones de creación, lectura, actualización y eliminación. Este componente presenta las siguientes características principales:

- Sirve como *proxy* de acceso a datos para acceso continuo tanto local como a sistemas de almacenamiento Cloud.
- Proporciona etiquetado de datos y almacenamiento de metadatos.
- Ofrece acceso protegido mediante Listas de Control de Acceso (ACLs), estableciendo permisos de lectura, escritura y/o eliminación.
- Se integra con el servicio de Accounting, ofreciendo el control y el seguimiento del consumo de recursos.
- Se integra con el servicio TRE, que se describe más adelante, para la mejora de la eficiencia de las descargas.

4.7.3 Servicios de Monitorización, Contabilidad y Facturación

Los servicios de Monitoring, Accounting y Billing [53] son responsables de gestionar el consumo de recursos Cloud y los costes financieros asociados al mismo, así como de poner a disposición dicha información a los usuarios.

Los servicios de Accounting y Billing se apoyan en un componente llamado Recopilador de Consumo o *Usage Tracker*. ese trata de un servicio web que tiene como objetivo recopilar y almacenar la información de consumo que publica el resto de

servicios, además de ponerla a disposición de los servicios de Accounting y Billing para mostrar la información relativa. Este servicio expone un conjunto de operaciones de lectura, escritura, actualización y eliminación de registros de consumo.

Debido a que los proveedores Cloud compatibles con VENUS-C proporcionan la información de consumo de diferente modo, la recuperación de esta información se realiza a través de unos módulos específicos para cada proveedor, llamados Conectores. Estos módulos recuperan la información de consumo almacenada en los servicios de los diferentes proveedores Cloud, procesan dicha información para homogeneizar los datos de consumo y registran la información resultante en el Usage Tracker.

Por otro lado, el servicio de Gestión de Facturación o *Billing Manager* presenta como funciones la gestión de las políticas de facturación y asociación de costes al consumo de recursos, y la aplicación de dichas políticas a los datos de consumo de los usuarios.

Por último, estos servicios ponen la información accesible a través de un portal web, donde los usuarios, en función de su nivel de privilegios, pueden consultar su consumo y facturación en base a diferentes criterios, bien sea por usuario, por recurso computacional, etc.

4.7.4 Eliminación de Tráfico Redundante

VENUS-C incluye un servicio de Eliminación de Tráfico Redundante (*Traffic Redundancy Elimination* o TRE) [54] que permite mejorar la velocidad y el tiempo de acceso en entornos donde se requiere continuamente acceso, descarga, distribución y/o modificación de datos. Este servicio permite eliminar la transmisión de contenido redundante y reducir significativamente el tráfico de red.

El funcionamiento del TRE se fundamenta en la capacidad de predicción para eliminar el tráfico redundante. Los clientes del servicio observan el flujo de datos para encontrar un fragmento de datos en una descarga previa o en un fichero local, envían al servicio información de predicción y notifican al mismo cuando se encuentra una coincidencia en los datos de descarga.

El uso del servicio TRE se basa en la redirección de mensajes HTTP, donde el cliente invoca al proxy cliente TRE mediante peticiones HTTP GET, siendo transparente tanto para el usuario como para el servicio Cloud.

5 Computación Cloud aplicada al Análisis de Estructuras de Edificación

El análisis de estructuras de edificación de gran dimensión es considerado un problema costoso debido tanto a su complejidad computacional como al volumen de datos que requiere. Además, el análisis dinámico en el tiempo de estructuras incrementa significativamente el orden de complejidad, debido a que este tipo de análisis requiere obtener el comportamiento de la estructura a lo largo del tiempo ante cargas dinámicas, multiplicando considerablemente tanto el tamaño del problema a resolver como el volumen de datos que genera. Por este motivo, numerosas herramientas comerciales de cálculo estructural incluyen una serie de simplificaciones que les permite abordar el problema tiempos de cálculo razonables, logrando reducir el tamaño del problema y el volumen de resultados generado. En cambio, este tipo de simplificaciones, que resultan aptas para edificios regulares, no resultan adecuadas para edificios con una elevada complejidad en altura y planta.

La Computación de Altas Prestaciones o *High Performance Computing* (HPC) proporciona las herramientas necesarias para simular estructuras de gran envergadura de manera eficiente, sin necesidad de incluir simplificaciones, y con tiempos de respuesta muy razonables. En este sentido, el programa de cálculo estructural Architrave permite llevar a cabo el diseño y el análisis, tanto estático como dinámico, de estructuras 3D de manera realista gracias a su simulador HPC.

A pesar de las posibilidades que ofrecen las aplicaciones HPC, los estudios de arquitectura e ingeniería raramente adquieren o emplean sistemas computacionales para llevar a cabo la ejecución de sus simulaciones estructurales en paralelo. Normalmente, este tipo de usuarios instala y ejecuta su software de cálculo estructural en su ordenador personal. Como consecuencia, el tamaño y la complejidad de las estructuras a analizar, el tipo de análisis empleado y el número de soluciones estructuras diferentes evaluadas suele estar limitado por las características de sus ordenadores personales.

Gracias al modelo de computación en la Nube, aparece una nueva vía que permite poner a disposición los recursos computacionales necesarios para abordar este tipo de problemas de manera eficiente. Como ya se comentó anteriormente, el Cloud computing ofrece acceso a través de la red a una variedad de recursos, tanto software como hardware, que permite disponer de los mismos a demanda, de manera rápida y sin intervención humana. Por tanto, este modelo permite a los desarrolladores de software diseñar aplicaciones escalables y fiables de manera más sencilla, suponiendo además un ahorro en el mantenimiento de los recursos empleados. Por otro lado, el uso del modelo de computación Cloud aporta una ventaja adicional a sus usuarios debido a que las aplicaciones se ejecutan de manera remota. En este modelo, los usuarios acceden a las aplicaciones Cloud mediante el navegador o mediante una aplicación ligera, liberando a los usuarios de la ejecución de la carga computacional de la aplicación principal.

Por ello, con el objetivo de superar las limitaciones que impone el análisis de estructuras en los ordenadores convencionales, y haciendo uso del potencial y las ventajas que ofrece el modelo de computación en el Cloud, se ha implementado un Servicio Cloud de Análisis Estructural, basado en el Generic Worker y los componentes de la plataforma VENUS-C, que permite a la comunidad de arquitectos e ingenieros de estructuras llevar a cabo el análisis de sus diseños estructurales en el Cloud.

5.1 Estado del arte del uso de Computación Distribuida y Cloud en el Análisis de Estructuras

Gracias al avance de las tecnologías distribuidas y a la popularidad que están adquiriendo las tecnologías Cloud, surge un nuevo escenario para la comunidad de arquitectos y de ingenieros estructurales que les ofrece el acceso a entornos computacionales potentes de manera muy sencilla. De este modo, el uso de tecnologías Cloud aplicadas al análisis de estructuras y obras de ingeniería civil permitiría abordar, de manera sencilla y cómoda, el diseño de estructuras que resultan inabordables en una máquina local, de manera más eficiente, y sin tener que soportar la carga computacional de la resolución del problema.

El uso de las tecnologías distribuidas para el análisis de estructuras es un tema que no se inicia con las tecnologías Cloud. De hecho, antes de surgir el Cloud, se desarrollaron diferentes implementaciones basadas en tecnologías Web para el cálculo remoto de estructuras. Por ejemplo, Peng y Law [55] presentaron la implementación de un entorno colaborativo basado en tecnologías Web para el desarrollo de aplicaciones de cálculo de estructuras. En la misma línea, Dolenc [56] presenta una plataforma SOA que proporciona un conjunto de aplicaciones específicas para el análisis de estructuras mediante elementos finitos.

Por otro lado, las tecnologías Grid, que fueron antecesoras de las tecnologías Cloud, también tenían por objetivo poner a disposición, tanto a comunidades científicas como a otros usuarios, numerosos recursos computacionales para la resolución de problemas complejos, aunque con menor facilidad que la que ofrece el Cloud. De hecho, dentro del grupo GRyCAP del Instituto de Instrumentación para Imagen Molecular, se desarrolló un Servicio Grid de Análisis de Estructuras [57] basado en el middleware Globus Toolkit 4 y en las tecnologías SOA, que permitía el análisis estático y dinámico de estructuras en este tipo de entorno a través de un sencillo cliente. Por otro lado, Wei, Zheng y Zhang [58] presentaron una plataforma Grid, basada también en Globus Toolkit, que ofrece un entorno integrado para el desarrollo de aplicaciones de ingeniería.

En relación a las tecnologías Cloud para el análisis estructural, están resultando numerosos los esfuerzos para adaptar esta aplicación a la computación en la nube. Por ejemplo, Kumar y Cheng [59] proponen un framework para la implementación de servicios Cloud para el ámbito de la construcción. Este framework sugiere el alojamiento de las herramientas software para el procesamiento de las estructuras como aplicaciones Cloud independientes en el Cloud público, y el almacenamiento y la gestión de los datos y modelos estructurales en el Cloud privado. Por otro lado, Beach, Rezgui, y Rana [60] presentan un prototipo de implementación Cloud para la gestión del ciclo de vida completo de los datos de los modelos estructurales basados en el proceso de modelado denominado *Building Information Modelling* (BIM)¹.

En cuanto a la implementación del proceso de análisis estructural en el Cloud, Xin, Xiao-qun, Hong y Zhen-li [61] proponen la combinación de técnicas de computación paralela basadas en MPI con el modelo de programación MapReduce en entornos Cloud para el análisis de mecánica de sólidos basado en el método de elementos finitos. Además, en el contexto del UberCloud Experiment [62], se evalúa el uso del HPC en el

¹ Building Information Modelling (BIM) es el proceso de generación y gestión de datos del edificio durante su ciclo de vida mediante software dinámico de modelado 3D de edificios y en tiempo real.

Cloud a través del cálculo no lineal de estructuras complejas mediante Abaqus. Por su parte, García y Bayo [63] presentan el desarrollo de una herramienta basada en el modelo de Software como Servicio para el análisis de estructuras 3D basadas en celosías. La arquitectura de esta aplicación Cloud consta de tres componentes: la aplicación cliente, que consiste en una aplicación web que hace uso de las tecnologías WebGL [64] y HTML5 [65] para el renderizado y la gestión de las estructuras desde el navegador; la aplicación servidora, que atiende las peticiones de cálculo a través del puerto 80 y consiste en un solver escrito en C++ bajo un preprocesador PHP; y el protocolo de comunicación, basado en XML HTTP Request (XHR) [66] con mensajes basados en JavaScript Object Notation format (JSON) [67].

Por otro lado, Ari y Muhtaroglu [68] presentan el diseño y la implementación de un servicio Cloud para el análisis de elementos finitos. La arquitectura de este servicio se compone de un portal web, encargada de la gestión de los usuarios y de los modelos estructurales empleados por los mismos, un componente solver basado en CalculiX [69] que lleva a cabo el análisis, y un gestor de trabajos que planifica la ejecución de los mismos y selecciona los recursos computacionales apropiados para su ejecución. Asimismo, Arita, Nozaki y Demizu [70] presentan una plataforma para diseño de mecánica sobre el Engineering Cloud [71] de Fujitsu. Por último, Simmhan et al. presentan en [72] el Generic Worker, un framework que permite el despliegue y la ejecución de aplicaciones científicas en el Cloud sobre la infraestructura de Microsoft, y que posteriormente formaría parte de la plataforma VENUS-C.

Además, como herramienta comercial disponible, existe un plug-in para Autodesk Revit Structure [73], llamado Autodesk 360 Structural Analysis [74], que surge tras el éxito del componente Project Storm para Revit Structure de Autodesk Labs. Este producto permite llevar a cabo análisis estáticos en la nube, sin interferir en el flujo de trabajo del diseño de la estructura y permitiendo además integrar los resultados como parte del proceso BIM.

5.2 Servicio Cloud de Análisis de Estructuras de Edificación basado en el software Architrave

El apartado actual presenta los diferentes componentes o módulos que componen el Servicio Cloud de Análisis Estructural (*Structural Analysis Cloud Service*), o también denominado SCAE, desarrollado en el marco del presente Trabajo Fin de Máster. Para la implementación del SCAE se han considerados los siguientes requisitos:

- El usuario debe poder definir las simulaciones remotas del mismo modo que las locales, a través de la misma herramienta.
- Las simulaciones deben ejecutarse de manera transparente al usuario.
- El servicio debe estar disponible únicamente para usuarios autorizados.
- La información de las simulaciones únicamente debe estar disponible para sus propietarios.
- El usuario debe poder simular diferentes estructuras al mismo tiempo.
- La ejecución de las simulaciones dinámicas no debe penalizar el tiempo de respuesta de las simulaciones estáticas.
- El usuario debe estar informado continuamente del estado de sus ejecuciones.

- El servicio debe poder ajustar sus recursos computacionales en base a la carga del sistema.

La arquitectura completa incluye los componentes del sistema que residen tanto en la parte de la máquina cliente como en la parte Cloud, como se muestra en la Figura 22. Dentro de la arquitectura se distinguen tres partes principales: el componente de cálculo estructural de Architrave, el Gestor Cliente de Simulaciones Remotas, y el Servicio Cloud de Análisis Estructural.

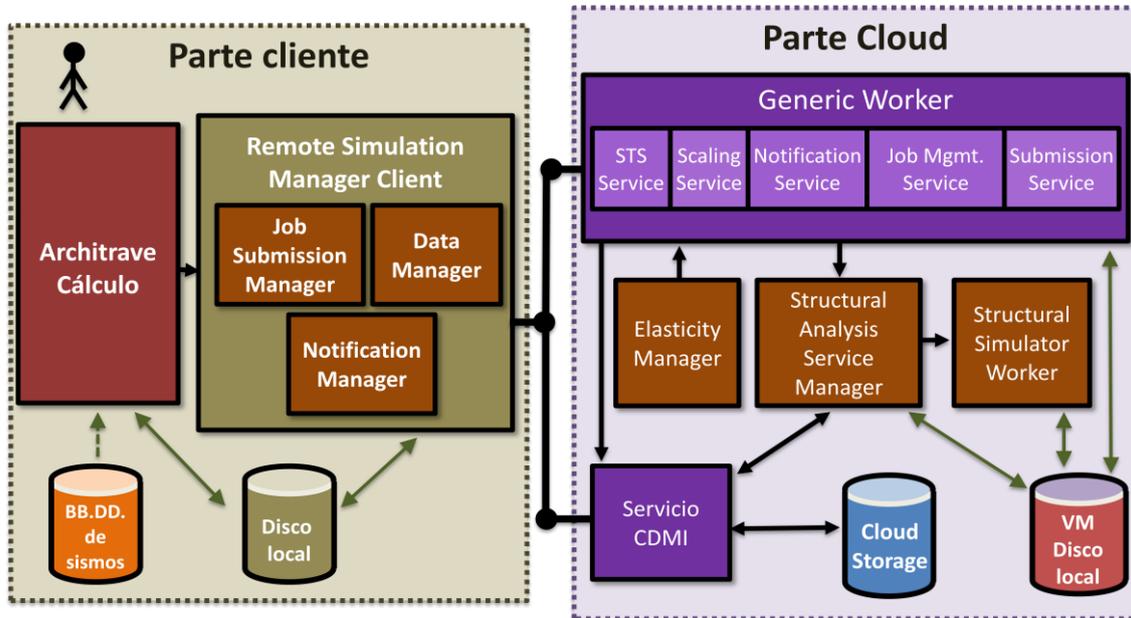


Figura 22: Arquitectura software del sistema Cloud.

En la parte del cliente, Architrave Cálculo es la aplicación Windows que permite gestionar el modelo estructural, lanzar el análisis local o definir las opciones de análisis en el Cloud y visualizar los resultados. Esta aplicación se encarga de lanzar el Gestor Cliente de Simulaciones Remotas y de solicitarle el envío de nuevos trabajos al servicio Cloud. El Gestor Cliente de Simulaciones Remotas (*Remote Simulation Manager Client*) es el responsable del envío y la gestión de las simulaciones en el Cloud, así como de la recuperación de los resultados.

Por otro lado, en la parte Cloud, el Servicio Cloud de Análisis Estructural es quien lleva a cabo la ejecución de las simulaciones en el Cloud. El servicio Cloud está basado en el Generic Worker, a través del cual ofrece protección de las comunicaciones mediante tokens de seguridad, transferencias a través del estándar CDMI, actualizaciones del estado de las simulaciones basadas en notificaciones y autoescalado del servicio en función de la carga del sistema. Dentro del servicio Cloud se ejecuta el Gestor del Servicio de Análisis Estructural (*Structural Analysis Service Manager*), quien se encarga de lanzar y encapsular los resultados generados por el Simulador Estructural (*Structural Simulator Worker*), responsable de llevar a cabo el cálculo de la estructura.

A continuación se analizan con mayor detalle todos los componentes de la arquitectura del sistema.

5.2.1 La herramienta de cálculo estructural Architrave

Architrave [75] es un sistema informático para entornos Windows orientado al diseño y al cálculo estático y dinámico de estructuras de edificación y obra civil mediante el método de los elementos finitos. Esta herramienta surge como el resultado de los trabajos de investigación desarrollados en la Universitat Politècnica de València por el grupo CiD del Departamento de Mecánica de Medios Continuos y Teoría de Estructuras y por el grupo GRyCAP del Instituto de Instrumentación para Imagen Molecular.

Architrave se compone de tres módulos independientes que interactúan entre sí: Architrave Diseño, Architrave Cálculo y el Simulador Estructural.

5.2.1.1 Architrave Diseño

El componente Architrave Diseño consiste en una interfaz gráfica de usuario basada en VisualLISP/OpenDCL e implementada como un plugin de AutoCAD. Este componente permite definir el modelo estructural mediante barras, elementos finitos, apoyos teóricos, zapatas y otros elementos de contorno, así como las cargas externas que se aplican al modelo estructural y las diferentes hipótesis de carga. Además, el componente incluye un generador de modelos basado en prototipos que permite crear celosías, pórticos 3D, losas y muros. Las superficies estructurales se mallan dinámicamente y de manera automática mediante la triangulación de Delaunay.

5.2.1.2 Architrave Cálculo

Architrave Cálculo es una aplicación gráfica desarrollada en .NET para la visualización y la gestión del modelo estructural. Este componente incluye un motor gráfico 3D de alto rendimiento que permite visualizar en tiempo real la vista alámbrica y sólida de modelos estructurales complejos. Architrave Cálculo se encarga también de invocar el Simulador Estructural para llevar a cabo el proceso de análisis, además de gestionar y visualizar los resultados obtenidos, como por ejemplo los desplazamientos, los esfuerzos o las tensiones que sufren los elementos estructurales. Asimismo, este componente permite editar y modificar las propiedades de los elementos del modelo estructural, así como dimensionar y peritar el comportamiento de los mismos. Finalmente, gracias a este componente es posible generar planos de la estructura y listados con los resultados del análisis estructural y del dimensionado de los elementos. Architrave Cálculo es por tanto el componente empleado para definir el tipo de análisis a realizar sobre un modelo estructural, lanzar dicho análisis en la máquina local o definir las opciones asociadas al cálculo de la estructura mediante el Servicio Cloud de Análisis Estructural.

En relación al análisis mediante el servicio Cloud, Architrave Cálculo permite configurar ciertas opciones de cálculo remoto, como puede apreciarse en la Figura 23. Por ejemplo, es posible indicar la cantidad de resultados que se generarán en el Cloud y que se recibirán en la máquina local, y la forma en la que serán descargados. De este modo, el Gestor Cliente de Simulaciones Remotas, el componente encargado de gestionar el lanzamiento de simulaciones al servicio Cloud, podría llevar a cabo la descarga automática de los resultados cuando estuvieran disponibles, o bien podrían descargarse bajo petición del usuario a través de dicha herramienta, y en función de las opciones seleccionadas. Además, en el caso de análisis dinámico de estructuras, Architrave Cálculo permite al usuario solicitar la generación remota de un vídeo con la respuesta de la estructura, así como escoger entre almacenar en el Cloud los resultados de todos los instantes de tiempo de la simulación o únicamente los resultados de los tres

instantes de tiempo más desfavorables, de acuerdo al valor de los cortantes basales² de la estructura en los ejes X, Y y al valor del módulo de ambos ejes. También, el usuario puede configurar también la cantidad de resultados a descargar, en función de la cantidad de resultados generados y almacenados en el Cloud. Asimismo, Architrave Cálculo permite indicar al Gestor Cliente de Simulaciones Remotas la eliminación automática de los resultados de la simulación una vez que han sido descargados en la máquina local.

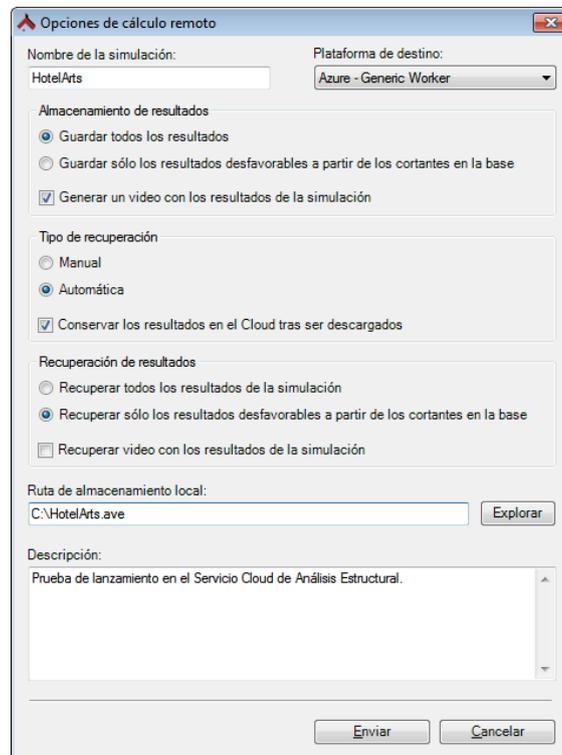


Figura 23: Captura de la definición de una simulación remota en Architrave.

5.2.1.3 Simulador Estructural

El Simulador Estructural consiste en una aplicación paralela de consola basada en el estándar MPI que simula la respuesta de la estructura mediante el método de los elementos finitos. Este componente de Computación de Altas Prestaciones o *High Performance Computing* permite realizar análisis estáticos, o también dinámicos mediante métodos de integración directa en el tiempo [76], análisis modal espectral [77] o análisis por superposición modal en el tiempo. Como se describe más adelante, este componente está presente en el Servicio Cloud de Análisis Estructural, tomando el nombre de *Structural Simulator Worker*, para llevar a cabo la simulación de los modelos estructurales en el Cloud.

5.2.2 El Gestor Cliente de Simulaciones Remotas

El Gestor Cliente de Simulaciones Remotas (*Remote Simulation Manager Client*), o también GCSR, es una aplicación gráfica desarrollada en .NET que se ejecuta de

² Los cortantes basales son una estimación de la máxima fuerza lateral que ocurre debido a un movimiento sísmico en la base de la estructura, y se emplean como criterio para evaluar la bondad de una solución estructural ante la acción de un terremoto.

manera independiente a Architrave Cálculo y que permite al usuario gestionar las simulaciones lanzadas al Servicio Cloud de Análisis Estructural y los resultados asociados a las mismas. Concretamente, esta aplicación, cuya captura de pantalla se muestra en la Figura 24, ofrece las siguientes características:

- Controla el acceso al servicio Cloud a través de un mecanismo de autenticación mediante usuario y contraseña.
- Informa en tiempo real del estado de las simulaciones lanzadas al servicio Cloud, así como el progreso de las ejecuciones en curso.
- Muestra la cantidad de resultados disponible de las distintas ejecuciones y la cantidad de los resultados descargados en la máquina local del cliente.
- Visualiza además otro tipo de información asociada a las simulaciones, como el tipo de análisis estructural (estático o dinámico) o la manera en que se descargarán los resultados (de forma manual o automática, descargando todos los resultados o únicamente los instantes más desfavorables).
- Permite cancelar los trabajos en ejecución enviados al servicio Cloud y los trabajos existentes en la cola de envío de nuevos trabajos, así como las descargas de resultados en curso y las descargas en la cola de espera de descarga.
- Ofrece, en el caso de simulaciones dinámicas en el tiempo con almacenamiento de todos los resultados, la posibilidad de iniciar la descarga de resultados una vez que el trabajo ha entrado en ejecución. De este modo, la ejecución de la simulación remota y la fase de recuperación de resultados se solapan en el tiempo, logrando reducir drásticamente el tiempo total empleado.
- Posibilita la eliminación los resultados generados por las simulaciones para liberar el espacio ocupado por los mismos y reducir así costes de almacenamiento, así como la eliminación de los propios registros o entradas de las simulaciones lanzadas anteriormente.
- Incluye una utilidad que muestra los resultados de los cortantes basales para cada uno de los instantes de tiempo de las diferentes simulaciones dinámicas, permitiendo al usuario conocer cuáles han sido los instantes de tiempo más desfavorables de una simulación en base a dicho criterio.
- Habilita la descarga de manera selectiva de los resultados de los instantes de tiempo deseados, e informa de los pasos de tiempo que han sido descargados anteriormente.

El Gestor Cliente de Simulaciones Remotas se lanza a través de Architrave Cálculo como un ejecutable independiente, de modo que puede continuar su ejecución, por ejemplo para descargar los resultados de las ejecuciones, aunque Architrave Cálculo se cierre. Dentro del RSMC se distinguen tres módulos internos diferentes: el Gestor de Envío de Trabajos, el Gestor de Datos, y el Gestor de Notificaciones.

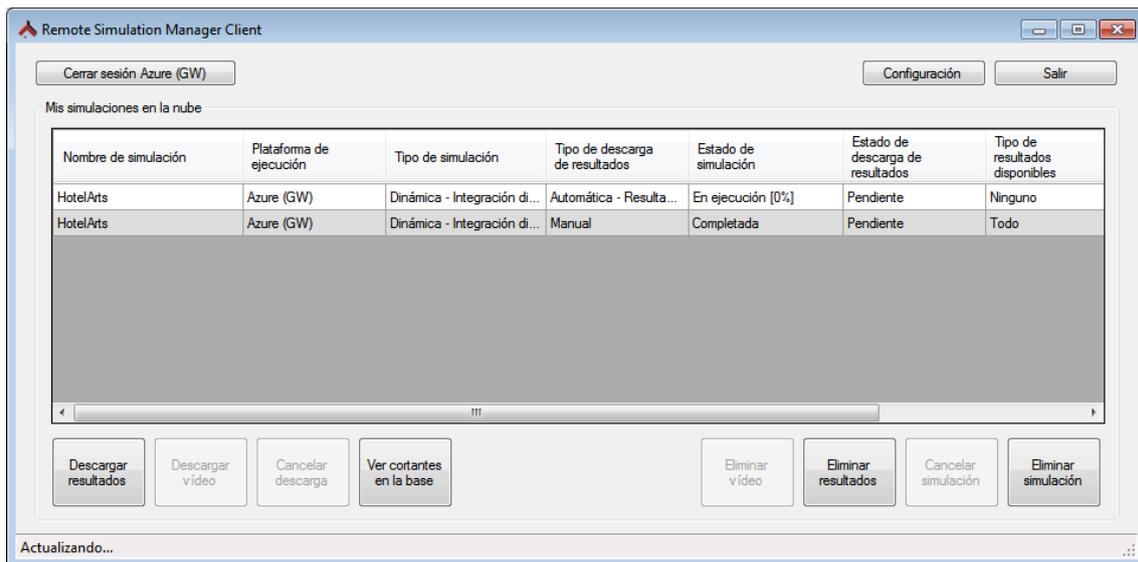


Figura 24: Captura del Gestor Cliente de Simulaciones Remotas.

5.2.2.1 El Gestor de Envío de Trabajos

El Gestor de Envío de Trabajos o *Job Submission Manager* es el componente que interactúa con el servicio Cloud para gestionar los trabajos remotos. Concretamente, cuando se lanza el Gestor Cliente de Simulaciones Remotas, este componente se encarga de comprobar el estado actual de las simulaciones lanzadas previamente al servicio Cloud e informar al usuario de dicho estado. Las simulaciones remotas se pueden encontrar en uno de los siguientes estados: Enviando, Pendiente, En ejecución, Finalizada, Fallida, Cancelando o Cancelada. Cuando acaba de comprobar el estado de las simulaciones, el Gestor de Envío de Trabajos lleva a cabo el envío al servicio Cloud de peticiones de nuevas simulaciones remotas recibidas por medio de Architrave. Además, ofrece el soporte para la cancelación de trabajos o la eliminación de registros de simulación.

El Gestor de Envío de Trabajos es el responsable de generar el documento JSDL (*Job Submission Description Language*) que se envía al Generic Worker con cada nueva petición de trabajo. El documento JSDL es un fichero XML que incluye la siguiente información asociada a cada trabajo:

- Información general, mediante las etiquetas *JobName* y *CustomerJobID*, a modo de cadenas de caracteres que indican el nombre del trabajo y su identificador.
- Referencia a la aplicación a ejecutar, mediante las etiquetas *AppPkgReference* y *AppDescReference*. *AppPkgReference* incluye una referencia al blob del servicio de almacenamiento Cloud con la aplicación, la cual consiste en un paquete ZIP con el ejecutable y todas sus dependencias. Por otro lado, *AppDescReference* almacena una referencia al blob con la plantilla que describe la invocación a la aplicación, con el conjunto de parámetros obligatorios y opcionales esperados por la misma.
- Valor de los parámetros de invocación de la aplicación. Para ello, define diferentes tipos de etiquetas, entre ellos: *LiteralArgument*, para guardar parámetros literales; *SwitchArgument*, para incluir parámetros booleanos; y *SingleReferenceInputArgument* y *SingleReferenceOutputArgument*, para incluir referencias al servicio de almacenamiento Cloud de los ficheros de entrada y de

salida de la aplicación, que pueden ser tanto referencias al almacenamiento de Azure como referencias al servicio CDMI. *SingleReferenceInputArgument* permite definir los ficheros de entrada de la aplicación, los cuales deberán ser descargados del Cloud Storage previamente al lanzamiento de la aplicación, mientras que *SingleReferenceOutputArgument* determina los ficheros que genera la aplicación y que son enviados al Cloud Storage al finalizar su ejecución.

Internamente, el Gestor de Envío de Trabajos consiste en una implementación de la interfaz *IjobManager* a la que el Gestor Cliente de Simulaciones Remotas accede y que interactúa con el servicio Cloud basado en el Generic Worker. Esta interfaz define el conjunto de operaciones que debe implementar un módulo para la interacción con un *Programming Model Enactment Service* o Servicio de Lanzamiento de Trabajos concreto. Por ello, dicha interfaz contiene, entre otras, las siguientes operaciones: *GetAllSimulations*, para obtener la lista de simulaciones existentes en el Cloud; *SubmitSimulation*, para el envío de nuevos trabajos; *CancelSimulation*, para la cancelación de trabajos; y *DeleteSimulation*, para la eliminación de los registros de la simulación.

5.2.2.2 El Gestor de Datos

El componente denominado Gestor de Datos o *Data Manager* es el responsable de enviar los datos de entrada de las simulaciones y controlar la descarga de los resultados generados por las mismas, bien sea por medio del servicio CDMI o bien directamente mediante el servicio de almacenamiento de Windows Azure. Internamente, el Gestor de Datos interactúa con cualquier componente que implemente la interfaz *ITransferer*. Este componente incluye, entre otras, las siguientes funciones: *RetrieveSimulationResults*, para la recuperación de los resultados de las simulaciones; *DownloadSimulationVideo*, para la descarga del video con la respuesta de la estructura; *DownloadBaseShears*, para la descarga de los resultados de los cortantes en la base; y *DeleteSimulationResults*, para la eliminación de los resultados de las simulaciones.

Los componentes que implementan la interfaz *ITransferer* son quienes realmente se comunican con los distintos servicios de almacenamiento. Esta interfaz define el conjunto de operaciones que debe implementar un módulo para la interacción con un servicio de almacenamiento Cloud concreto. De este modo, la interfaz *ITransferer* contiene, entre otras, las siguientes operaciones: *UploadFile* y *DownloadFile*, para el envío y la descarga de datos al servicio de almacenamiento Cloud; *ListContainer*, para listar el contenido de un contenedor del Cloud; y *DeleteBlob*, para la eliminación de ficheros del servicio de almacenamiento.

Para la interacción con el Cloud Storage se han desarrollado dos implementaciones de *ITransferer*: *CDMITransferer* y *AzureTransferer*. *CDMITransferer* es el encargado de interactuar con el CDMI proxy y llevar a cabo el envío y la descarga de datos al Cloud a través de dicho servicio. Para la interacción a través de CDMI, ha sido también necesario el desarrollo de un módulo que cumple con el interfaz REST del servicio CDMI y permite la comunicación con el mismo. Por otro lado, la implementación *AzureTransferer* permite interactuar directamente con el servicio de almacenamiento de Windows Azure, gracias al uso de las librerías incluidas en su SDK.

5.2.2.3 Gestor de Notificaciones

El componente *Notification Manager* o Gestor de Notificaciones se encarga de obtener los cambios de estado de las simulaciones, así como los cambios en el estado del

progreso de su ejecución, y de actualizar la información asociada. El esquema de notificaciones se basa en una implementación mediante colas de almacenamiento de mensajes de Windows Azure. El Servicio Cloud de Análisis Estructural emplea dos colas por cada usuario: una para almacenar las notificaciones de cambio de estado de las simulaciones y otra para almacenar las notificaciones del progreso de las simulaciones en ejecución.

Periódicamente, el Gestor Cliente de Simulaciones Remotas solicita refrescar el estado de las simulaciones que se encuentran en un estado no finalizado. Entonces, el Gestor de Notificaciones procesa los mensajes de cambio de estado y de progreso de ejecución existentes en las colas. La implementación del Gestor de Notificaciones incluye las siguientes operaciones: *CreateJobSubscription* y *DeleteJobSubscription*, para crear o eliminar la suscripción a las notificaciones de una simulación, y *UpdateFromNotifications*, que permite actualizar el estado de las simulaciones en base a las notificaciones existentes en las colas.

5.2.3 El Servicio Cloud de Análisis de Estructuras de Edificación

El Servicio Cloud de Análisis Estructural (*Structural Analysis Cloud Service*) o SCAE es el conjunto de componentes que se encarga de atender, registrar y procesar las peticiones de cálculo estructural en el Cloud.

Es importante matizar que el Servicio Cloud de Análisis Estructural implementado está basado en un modelo de Computación de Alta Productividad o *High Throughput Computing*, donde múltiples clientes emplean concurrentemente el servicio y diversas soluciones estructurales se analizan al mismo tiempo. Por ello, teniendo en cuenta además que las simulaciones dinámicas consumen mucho más tiempo que las simulaciones estáticas, el SCAE consta de dos servicios diferentes de análisis estructural para el procesamiento de simulaciones: un servicio para la gestión de simulaciones estáticas, llamado *static-sacs*, y otro para el procesamiento de simulaciones dinámicas, llamado *dynamic-sacs*. De este modo, el tiempo de respuesta de las simulaciones estáticas no se ve penalizado por el elevado coste computacional de las simulaciones dinámicas, que podrían llegar a acaparar todas las instancias disponibles del servicio. Además, de este modo se logra incrementar de manera notable la productividad del sistema.

Como se ha mencionado anteriormente, los dos servicios de análisis estructural están basados en el Generic Worker y ofrecen un conjunto de características interesantes, como protección de las comunicaciones, transferencias de datos mediante el estándar CDMI y ajuste automático del número de instancias en ejecución en función de la carga del sistema. El SCAE está formado por los siguientes componentes: el Generic Worker, el Gestor del Servicio de Análisis Estructural, el Simulador Estructural, el Servicio CDMI y el Gestor de Elasticidad.

5.2.3.1 El Generic Worker

El Generic Worker, como ya se mencionó anteriormente, consiste en un Web role para Windows Azure que se encarga de gestionar la ejecución de las simulaciones remotas. Dentro del Generic Worker, se distinguen los siguientes módulos internos que desempeñan diferentes funciones: el Security Token Service, que protege las comunicaciones con el servicio mediante tokens de seguridad; el Scaling Service, que incrementa o decrementa el número de instancias en ejecución del propio Generic

Worker; el Submission Service, que gestiona la recepción de trabajos al servicio Cloud, atendiendo y registrando las nuevas peticiones que llegan al servicio; el Notification Service, que envía notificaciones de los cambios de estado de los trabajos y del progreso de los trabajos en ejecución; y el Job Management Service, que proporciona información asociada al estado de los trabajos, como pueden ser el estado de la ejecución, la salida estándar y de error de la aplicación, y la hora de envío, de comienzo y de finalización del trabajo.

5.2.3.2 El Gestor del Servicio de Análisis Estructural

El Gestor del Servicio de Análisis Estructural (*Structural Analysis Service Manager*), o también denominado GSAE, es el componente que lleva a cabo la invocación del Simulador Estructural y el postprocesamiento de los resultados generados por el mismo. Este componente realiza cuatro tareas principales: la lectura de los ficheros de geometría estructural y de parámetros de cálculo, la generación de los ficheros de entrada necesarios para la ejecución del Simulador Estructural, la invocación del Simulador Estructural con los parámetros adecuados en función del tipo de cálculo seleccionado y la encapsulación en un único fichero de resultados, en el caso de ejecuciones de análisis dinámico con respuesta en el tiempo, por cada paso de tiempo, de los múltiples ficheros generados por el Simulador Estructural.

El Gestor del Servicio de Análisis Estructural permite, en función de las opciones seleccionadas por el usuario en la definición de la simulación remota, guardar en el servicio de almacenamiento Cloud todos los resultados de paso de tiempo de la simulación, o bien únicamente los instantes de tiempo más desfavorables de acuerdo al valor de los cortantes basales de la estructura. Además, también lleva a cabo la generación de un video con la respuesta de la estructura al finalizar la ejecución del Simulador Estructural y su almacenamiento en el Cloud, si dicha opción ha sido seleccionada. Asimismo, en el caso de la ejecución de análisis dinámico en el tiempo, si se selecciona el almacenamiento de todos los resultados de la simulación, el Gestor del Servicio de Análisis Estructural permite solapar la ejecución del Simulador Estructural con la encapsulación de los ficheros de resultados y su envío al servicio de almacenamiento Cloud, de modo que el usuario puede comenzar la descarga los resultados de una simulación mientras ésta se ejecuta.

5.2.3.3 El Simulador Estructural

Este componente, también llamado *Structural Simulator Worker*, es el mismo componente que forma parte de Architrave. Como ya se indicó anteriormente, se trata de un módulo HPC basado en el estándar MPI que lleva a cabo la ejecución del análisis estructural.

5.2.3.4 El Servicio CDMI

El Servicio CDMI se corresponde con el componente Data Manager de la plataforma VENUS-C, y es el encargado de enviar y descargar los datos a través del protocolo estándar CDMI. Este componente no forma parte del Generic Worker, y se ejecuta en una instancia de Azure aparte, junto con el Gestor de Elasticidad. Como ya se mencionó en el capítulo correspondiente, el Servicio CDMI ofrece otras características, como la función de proxy de acceso a datos o el control de acceso mediante ACLs.

5.2.3.5 El Gestor de Elasticidad

El Gestor de Elasticidad o *Elasticity Manager* es responsable de monitorizar la carga del sistema Cloud y ajustar automáticamente el número de instancias en ejecución del servicio en función de la carga del sistema. Este componente se ejecuta en una instancia de Azure junto al Servicio CDMI e interactúa con el Servicio de Escalado (Scaling Service) del Generic Worker para ajustar el número de recursos computacionales del servicio Cloud.

La labor del Gestor de Elasticidad consiste en consultar, cada cierto período de tiempo establecido, el número de instancias ociosas mediante la diferencia entre el número de instancias del despliegue y el número de trabajos en ejecución, y tomar las acciones apropiadas en función de dicho valor. Si el número de instancias ociosas obtenido es inferior a cierto umbral preconfigurado, el Gestor de Elasticidad aumenta el tamaño del despliegue en un número de instancias definido, siempre que no se haya alcanzado el tamaño máximo configurado. En cambio, si detecta que no existe ningún trabajo en ejecución, se reduce el número de instancias desplegadas al mínimo establecido.

El Gestor de Elasticidad define diferentes valores de configuración para el servicio de cálculo estático y para el servicio de análisis dinámico. Concretamente, en el caso del servicio de cálculo estático, establece un tamaño mínimo de 2 instancias en ejecución y un tamaño máximo de 8, ya que el tipo de simulaciones que ejecuta requiere menor complejidad computacional. En cambio, debido a que las simulaciones dinámicas tienen mayor coste computacional y requieren un mayor tiempo de ejecución, el Gestor de Elasticidad establece para el servicio de análisis dinámico un mínimo de 4 instancias en ejecución y un máximo de 28. Para ambos servicios, tanto el umbral de instancias ociosas permitido antes de desplegar más instancias como el tamaño de incremento cuando se requiere el aumento de los recursos computacionales es de 2 instancias.

El valor del intervalo de tiempo entre eventos de monitorización de la carga del sistema es de 15 minutos. Este valor debe ajustarse apropiadamente para permitir la puesta en funcionamiento de nuevas instancias en ejecución. Si este intervalo fuera menor al tiempo de despliegue de nuevas instancias, aunque se mantuviera carga del sistema tras un incremento del número de instancias, el Gestor de Elasticidad podría volver a solicitar un nuevo incremento del tamaño del servicio en determinados casos, debido a que se detectaría que el número de trabajos pendientes de ejecución no se ha reducido pese al incremento del número de instancias. En cambio, si el valor del intervalo fuera muy grande, el Gestor de Elasticidad no respondería con rapidez ante el aumento de la carga del sistema, penalizando el tiempo de respuesta de las simulaciones.

5.2.4 Ejemplo de uso del Servicio Cloud de Análisis Estructural

La Figura 25 detalla las distintas etapas que se suceden para llevar a cabo el envío de nuevas simulaciones al SCAE, el proceso de ejecución de las mismas y la recuperación y visualización de los resultados asociados. En el ejemplo incluido se describe la labor que realizan los distintos componentes, así como la comunicación y la transferencia de datos que ocurre entre los mismos.

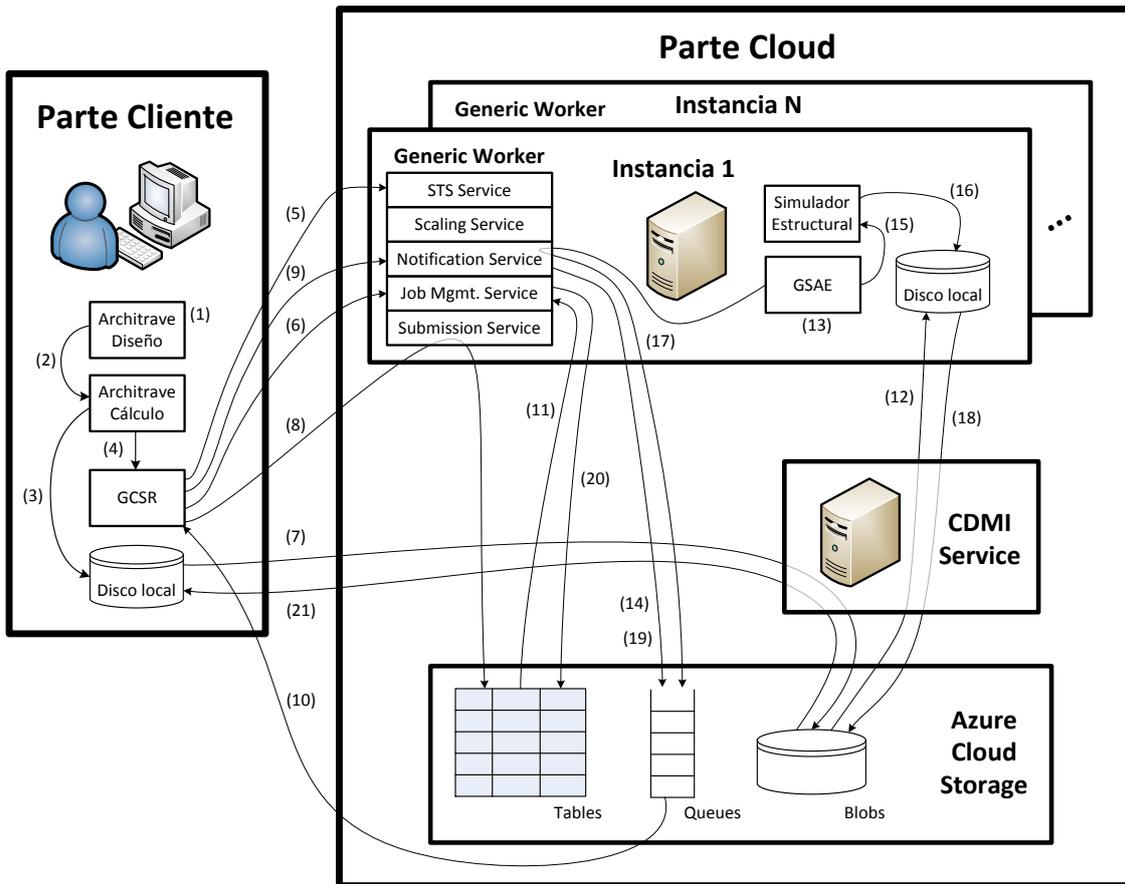


Figura 25: Arquitectura física y ejemplo de uso del Servicio Cloud de Análisis Estructural.

Inicialmente, el usuario define el modelo estructural en Architrave Diseño, incluyendo la geometría y las acciones externas que actúan sobre la estructura (1). A continuación, desde Architrave Cálculo, el usuario escoge el tipo de análisis a realizar (por ejemplo, dinámico en el tiempo) y selecciona la ejecución de la simulación en el Cloud con las opciones deseadas, como la descarga automática de los resultados o la generación del vídeo (2). Como consecuencia, Architrave Cálculo lanza el Gestor Cliente de Simulaciones Remotas, escribe un fichero binario con la geometría estructural y otro fichero con los parámetros de cálculo (3), y envía un mensaje al GCSR que contiene la ruta de ambos ficheros, junto con las opciones de configuración de la simulación (4).

Cuando se inicia el Gestor Cliente de Simulaciones Remotas, se solicitan las credenciales del usuario. Una vez que el usuario se ha autenticado y se comprueba que está autorizado para el uso del SCAE por medio del Security Token Service (5), el GCSR obtiene la lista de simulaciones lanzadas previamente al servicio Cloud y el estado actual de las mismas por medio del Job Management Service (6). A partir de ese momento, el usuario puede enviar nuevas simulaciones al servicio, descargar los resultados de las simulaciones existentes, cancelar las simulaciones que se encuentran en ejecución, etcétera.

Tras obtener la lista de simulaciones, los diferentes componentes del GCSR proceden con el envío de nuevos trabajos al SCAE. Así, cuando se envía una nueva tarea, el Gestor de Datos copia primero los ficheros binarios de geometría estructural y los parámetros de entrada en el servicio de almacenamiento Cloud, por medio del servicio CDMI (7). A continuación, el Gestor de Envío de Trabajos envía el trabajo al servicio

Cloud en forma de documento JSDL y registra la simulación a través del Submission Service (8), donde una de las instancias del servicio Cloud, determinada por el balanceador de carga de Windows Azure, registra la entrada en la tabla de Azure correspondiente. Seguidamente, el Gestor de Notificaciones se suscribe a los cambios de estado significativos de la tarea a través del Notification Service (9) y, finalmente, se informa al usuario del éxito del envío. Además, en el caso del envío de una simulación con análisis dinámico en el tiempo y con descarga de todos los resultados de cálculo, el Gestor de Datos inicia inmediatamente la recuperación de los resultados, siguiendo un modelo *data-driven*. A partir de ese momento, el Gestor Cliente de Simulaciones Remotas consulta periódicamente, por medio del Gestor de Notificaciones, los cambios de estado y el progreso de la ejecución del trabajo, e informa al usuario (10).

Las instancias desplegadas del servicio que se encuentran ociosas consultan periódicamente la tabla correspondiente en busca de trabajos pendientes de ser ejecutados. De este modo, una vez que el trabajo ya ha sido registrado en el SCAE, una de las instancias se dispondrá a procesarlo. Para ello, inicialmente obtiene de dicha tabla la información asociada al trabajo (11) y descarga los ficheros binarios de entrada al cálculo desde el almacenamiento Cloud al disco local de la máquina virtual (12).

Una vez que los ficheros están disponibles localmente, la instancia encargada de la ejecución del trabajo lanza el Gestor del Servicio de Análisis Estructural (13) y notifica el cambio de estado del trabajo mediante el Notification Service del Generic Worker (14). Entonces, lee los ficheros de entrada y genera un fichero de entrada interpretable por el Simulador Estructural para la ejecución de la simulación. Seguidamente, el GSAE pone en marcha el Simulador Estructural, que es invocado con los parámetros adecuados (15), dependiendo del tipo de análisis definido por el usuario. En ese instante, comienza el cálculo estructural, donde los ficheros de resultados se generan periódicamente (en caso de tratarse de un análisis dinámico en el tiempo) o al finalizar la simulación y se almacenan en el disco local (16). Mientras se ejecuta el Simulador Estructural, el GSAE informa al Notification Service del progreso de la ejecución del trabajo, quien actualiza la información en la cola correspondiente (17). Además, en caso de ejecución de un análisis dinámico con respuesta a lo largo tiempo, el GSAE se encarga de encapsular, por cada instante de tiempo de la simulación, en un único fichero los múltiples ficheros de resultados que genera el Simulador Estructural. Asimismo, si el usuario ha escogido la opción de almacenamiento de todos los resultados de tiempo, el Gestor del Servicio de Análisis Estructural procedería con la copia de resultados al servicio Cloud de almacenamiento (18) mientras la simulación continúa su ejecución en el Cloud.

Finalmente, cuando termina la ejecución del Simulador Estructural, tiene lugar el envío de los resultados de cálculo pendientes al Cloud Storage (18). Cuando se trata de una simulación dinámica en el tiempo, el Gestor del Servicio de Análisis Estructural genera y envía al servicio de almacenamiento Cloud un fichero binario que contiene los resultados de los cortantes basales en el eje X, Y y en el módulo de ambos ejes, y seguidamente genera un video con la respuesta de la estructura y copia en el almacenamiento Cloud si dicha opción ha sido seleccionada. Por el contrario, si el usuario desea almacenar únicamente los resultados desfavorables, envía los tres pasos de tiempo más desfavorables al servicio de almacenamiento Cloud. Si en cambio la simulación no consiste en un análisis dinámico en el tiempo, los resultados del Simulador Estructural serán enviados por parte del Gestor del Servicio de Análisis Estructural al servicio Cloud de almacenamiento cuando haya concluido la simulación. Una vez finaliza la ejecución del Gestor del Servicio de Análisis Estructural, la instancia

del Generic Worker encargada de ejecutar el trabajo notifica del fin de la ejecución mediante el Notification Service (19) y registra en la tabla de Azure que la tarea ha sido completada (20).

Entonces, en la parte del cliente, el Gestor Cliente de Simulaciones Remotas, tras consultar periódicamente el estado de sus simulaciones por medio de la cola de notificaciones correspondiente, detecta una notificación de finalización de trabajo (10) e inicia la descarga de los resultados correspondientes (21) en caso de que el usuario haya escogido la descarga automática de resultados. En caso de tratarse de una simulación dinámica en el tiempo con descarga de todos los resultados, el Gestor Cliente de Simulaciones Remotas ya habrá iniciado previamente la descarga de resultados para llevarla a cabo de manera solapada con la ejecución del trabajo, por medio del Gestor de Datos. Por tanto, el Gestor descargará finalmente los resultados de los cortantes en la base y el video con la respuesta de la estructura, si ha sido previamente seleccionado. En cambio, el Gestor Cliente de Simulaciones Remotas descargará únicamente los tres instantes de tiempo más desfavorables y los resultados de los cortantes basales si dicha opción ha sido escogida por el usuario. De igual modo, si la simulación ejecutada es de tipo estático, dicho Gestor descargará los resultados de la respuesta de la estructura.

Finalmente, cuando el Gestor Cliente de Simulaciones Remotas detecta que la simulación ha finalizado todos los ficheros han sido descargados en la máquina del cliente, informa al usuario de que los resultados de la ejecución se encuentran disponibles. Entonces, el usuario puede abrir los ficheros asociados a la simulación desde Architrave Cálculo y visualizar dichos resultados.

5.3 Caso de Estudio de Análisis Estructural en el Cloud

A continuación se describe el caso de estudio representativo que ha sido aplicado para validar el sistema Cloud. Para ello, primero se describe la estructura seleccionada como el caso de estudio, justificando el interés estructural y computacional del mismo. A continuación, se presentan los resultados obtenidos en las distintas ejecuciones realizadas, y se exponen las conclusiones extraídas de los resultados, así como las mejoras o los beneficios esperados mediante el uso de una solución computacional basada en el Cloud.

5.3.1 Descripción del caso de estudio

La estructura seleccionada se corresponde con una reinterpretación, según usos y normativas actuales, de la estructura original del Banco Nórdico ubicado en Helsinki, obra del arquitecto finlandés Alvar Aalto en 1962 (ver Figura 26). Concretamente, se trata de una estructura aporricada, resuelta mediante una losa maciza sobre pilares de hormigón interiores y de acero en fachada. Los núcleos de comunicación vertical se materializan como pantallas de hormigón armado, funcionando como estructura vertical y arriostramiento lateral. Las luces son moderadas, excepto en algunas zonas en plantas bajas donde se refuerza la losa con vigas de cuelgue de hormigón. Los pilares metálicos de fachada se disponen a luces mitad, debiendo utilizarse vigas de desvío a nivel de primer forjado. La cimentación es de losa, dada la presencia de nivel freático. El modelo se ha realizado con barras para pilares y vigas y elementos finitos de tamaño medio para losas y muros. Para las losas se ha utilizado un mallado Delaunay, mientras que los muros se resuelven con mallado simple.

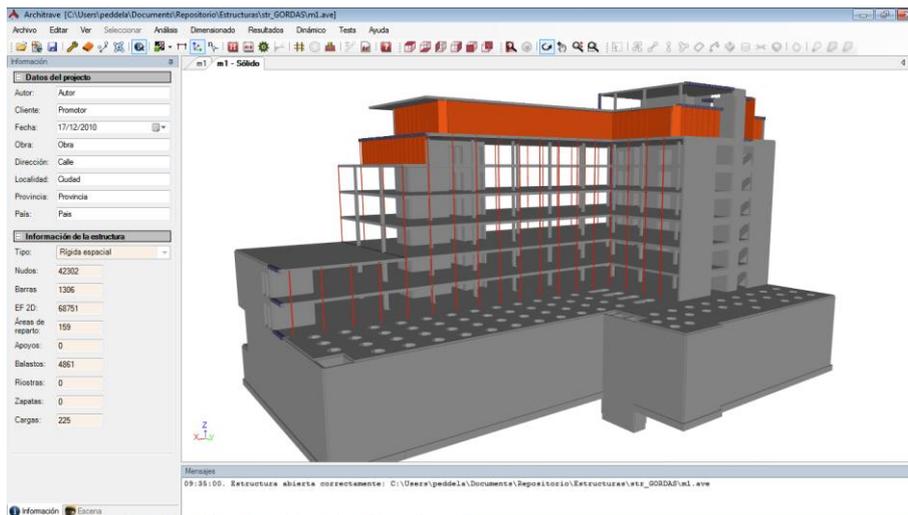


Figura 26: Modelo de la estructura del Banco Nórdico en Architrave.

El caso de estudio presentado consiste en el diseño del edificio, mediante el análisis del comportamiento de diez soluciones estructurales diferentes. Cada una de dichas soluciones consiste en una variación de algunos de los elementos estructurales, como la dimensión de las columnas, o el grosor de las losas o de los muros. Además, cada solución ha sido analizada bajo la influencia de cinco terremotos representativos, de acuerdo a las exigencias de la Normativa Española de Construcción Sismorresistente de Edificación (NCSE-02) [78]. Por tanto, el objetivo del caso de estudio consiste en seleccionar la mejor de las diez soluciones estructurales, cumpliendo con los requisitos estructurales y las normativas de seguridad, y además presentando el menor coste económico.

5.3.2 Ejecución del caso de estudio

Para la ejecución del caso de estudio, se ha definido por tanto un conjunto 50 de simulaciones dinámicas en el tiempo que han sido ejecutadas en dos configuraciones diferentes: la configuración A, donde los resultados de todos los instantes de tiempo fueron almacenados en el Cloud y descargados por el cliente; y la configuración B, donde únicamente los tres pasos de tiempo más desfavorables fueron almacenados en el Cloud y descargados por el usuario. Además, las dos configuraciones fueron lanzadas sobre diferentes tamaños del servicio de análisis estructural, compuestos por 1, 10, 25 y 50 instancias de Windows Azure de tamaño “*Medium*”, que disponen de una CPU de doble núcleo de 1,60 GHz y 3,5 GBytes de memoria RAM. Cada una de las soluciones estructurales contiene 253.812 grados de libertad, con 1.306 barras y 68.751 elementos finitos 2D. La respuesta dinámica de la estructura fue analizada bajo la influencia de una carga sísmica de 12 segundos de duración, con un incremento de tiempo de simulación de 0,01 segundos. Los resultados fueron almacenados cada 0,05 segundos, de modo que 240 instantes de tiempo fueron almacenados en disco, frente a los 1200 pasos de tiempo calculados.

Cada una de las simulaciones dinámicas tardó alrededor de 120,03 minutos empleando un ordenador convencional, con una CPU Intel core i5 de 3,20 GHz y 4 GBytes de memoria RAM, y requirió 2,48 GBytes para el almacenamiento de los 240 instantes de tiempo de los resultados. En cambio, cuando únicamente se consideraron los tres pasos de tiempo más desfavorables, necesitó sólo 117,84 MBytes de almacenamiento. El

fichero de entrada con la geometría y la carga sísmica a emplear en cada simulación ocupaba 5,43 MBytes de espacio de almacenamiento.

Por tanto, para la ejecución en la máquina local del caso de estudio completo, ejecutando las 50 simulaciones una tras otra, fueron necesarias alrededor de 100,03 horas, y se generó un volumen de resultados del orden de 124,04 GBytes cuando se almacenaron los 240 pasos de tiempo, o un volumen de 5,75 GBytes cuando únicamente se almacenaron los tres instantes de tiempo más desfavorables. Además, la cantidad de de los datos de entrada, es decir, del conjunto de los 50 ficheros con las diferentes soluciones estructurales, fue de 271,5 MBytes.

5.3.3 Análisis de prestaciones

En el presente apartado se presentan los resultados de las pruebas de validación que demuestran el comportamiento de la plataforma Cloud de acuerdo al caso de estudio descrito. Para ello, se ha llevado a cabo un estudio cuantitativo, donde se ha analizado la eficiencia de la plataforma por medio del tiempo de respuesta y del incremento de velocidad o *speed-up* que ha obtenido la aproximación Cloud respecto a la aproximación local y respecto al uso de una instancia de ejecución del servicio de Azure en el Cloud.

Previamente a la ejecución del caso de estudio completo, se ha analizado el comportamiento obtenido en el Cloud por parte de una de estas soluciones estructurales ante un terremoto. La Tabla 2 muestra el tiempo de respuesta, el tamaño de los datos de entrada y de salida y la cantidad de datos descargados por el cliente local para las dos configuraciones de ejecución descritas, junto con el tiempo de respuesta y el tamaño de los datos cuando se ejecutó dicha simulación en la máquina local del usuario.

<i>Tipo de ejecución</i>	<i>Tiempo de respuesta (minutos)</i>	<i>Datos de entrada (MBytes)</i>	<i>Datos de salida (GBytes)</i>
Ejecución local – Almacenamiento de los resultados de los 240 pasos de tiempo	120,03	5,43	2,48
Configuración A: Ejecución remota – Almacenamiento y recuperación de los resultados de los 240 pasos de tiempo	279,48	5,43	2,48
Configuración B: Ejecución remota – Almacenamiento y recuperación de los resultados de los 3 pasos de tiempo más desfavorables	277,46	5,43	0,12

Tabla 2: Resultados de la simulación de una de las soluciones estructurales en el Cloud.

Antes de analizar los resultados es importante aclarar que, para la configuración A, los movimientos de datos de la máquina virtual al almacenamiento Cloud, y del almacenamiento Cloud a la máquina del usuario, se sucedieron de manera solapada con la ejecución de la simulación. Sin embargo, para la configuración B, los movimientos de los datos tuvieron lugar cuando la simulación había finalizado y por tanto los resultados de tiempo más desfavorables habían sido calculados.

Los resultados de la Tabla 2 muestran una diferencia destacable entre el tiempo de respuesta local y el tiempo de respuesta en el Cloud. Por ello, para poder analizar de manera detallada dicha sobrecarga del tiempo, se ha analizado el consumo de tiempo de las diferentes fases que se suceden en una simulación en el Cloud. La Tabla 3 muestra el tiempo empleado en cada una de las fases de la simulación de una única solución estructural donde únicamente los resultados más desfavorables se guardan en el Cloud y se descargan en la máquina local (configuración B). Como se muestra en la tabla, el tiempo consumido en la ejecución de la simulación en el Cloud dio lugar a un retraso de 155,28 minutos respecto a la ejecución local.

	<i>Envío de datos de entrada y del trabajo</i>	<i>Descarga de la aplicación y lanzamiento del trabajo</i>	<i>Ejecución del Simulador Estructural y encapsulación de resultados</i>	<i>Envío de resultados al Cloud Storage</i>	<i>Descarga de resultados en la máquina local</i>
Ejecución local	-	-	120m 02s	-	-
Ejecución remota	6s	39s	275m 19s	50s	34s
Tiempo de sobrecarga	6s	39s	155m 17s	50s	34s

Tabla 3: Tiempos de ejecución de las distintas fases de la simulación en el Cloud (configuración B).

Los valores de la Tabla 3 reflejan que la mayor parte de la sobrecarga reside en la ejecución del Simulador Estructural y del Gestor del Servicio de Análisis Estructural, responsables del análisis de la estructura y de la encapsulación de los resultados generados respectivamente. Para la configuración B, la complejidad computacional es similar a la ejecución local. Por tanto, se puede asumir que la principal sobrecarga se debe a diferencias en el hardware, concretamente en las características de la CPU, o en las velocidades de acceso al disco para operaciones de lectura o escritura.

Los tiempos de sobrecarga en la configuración A muestran un comportamiento similar a la configuración B, como se refleja en la Tabla 4. Sin embargo, en la configuración A, durante la ejecución del Simulador Estructural se sucede también la transferencia de los resultados al Cloud y a la máquina cliente. Por ello, la quinta columna de la tabla muestra el tiempo consumido para el envío de los datos al almacenamiento Cloud y la descarga de los resultados en el cliente de los últimos pasos de tiempo, es decir, cuando la ejecución del Simulador Estructural ha finalizado. De nuevo, los valores de la Tabla 4 muestran que la principal sobrecarga ocurre durante la simulación de la estructura, con un retraso de 154,88 minutos respecto a la ejecución local.

	<i>Envío de datos de entrada y del trabajo</i>	<i>Descarga de la aplicación y lanzamiento del trabajo</i>	<i>Ejecución del Simulador Estructural y encapsulación de resultados</i>	<i>Envío de últimos resultados al Cloud Storage y descarga en la máquina local</i>
Ejecución local	-	-	120m 02s	-
Ejecución remota	6s	39s	274m 55s	3m 49s
Tiempo de sobrecarga	6s	39s	154m 53s	3m 49s

Tabla 4: Tiempos de ejecución de las distintas fases de la simulación en el Cloud (configuración A).

Tras el análisis de los tiempos de ejecución de una única solución estructural con un solo terremoto, se ha llevado a cabo la ejecución del caso de estudio completo, donde las 50 simulaciones fueron enviadas al Cloud mismo tiempo. La evaluación compara el tiempo de respuesta de la ejecución local con el tiempo de respuesta en remoto, para las dos configuraciones descritas, y sobre un conjunto de despliegues Cloud de diferente tamaño compuestos de 1, 10, 25 y 50 instancias Azure Web Role de tamaño “Medium”. Cada una de las pruebas de caso de estudio ha sido lanzada tres veces diferentes al Cloud, por lo que los resultados obtenidos se pueden considerar fiables. Los resultados mostrados se han obtenido como el promedio de tres ejecuciones diferentes. Los tiempos de respuesta fueron medidos como la diferencia de tiempo entre el comienzo del primer envío de trabajo y la descarga del último resultado de la última simulación.

La Figura 27 muestra los tiempos de respuesta para la ejecución del caso de uso completo, para ambas configuraciones y para distinto número de instancias del servicio Cloud. Como puede observarse, para la configuración B, el tiempo de respuesta se reduce gradualmente conforme aumenta el número de instancias. En el caso de la configuración A, los resultados muestran inicialmente una tendencia similar. En cambio, cuando el número de instancias del servicio es superior a 25, los resultados adquieren un valor de alrededor de 17 horas. Este hecho se debe al efecto “cuello de botella” que ocurre en la red durante la recogida del elevado volumen de datos por parte del cliente, debido a que el tiempo necesario para la descarga de todos los resultados generados es muy superior al tiempo necesario para la ejecución de las simulaciones. Por tanto, el tiempo total invertido estará determinado por el tiempo consumido en la recepción, en la máquina local, de los 124,04 GBytes de resultados generados.

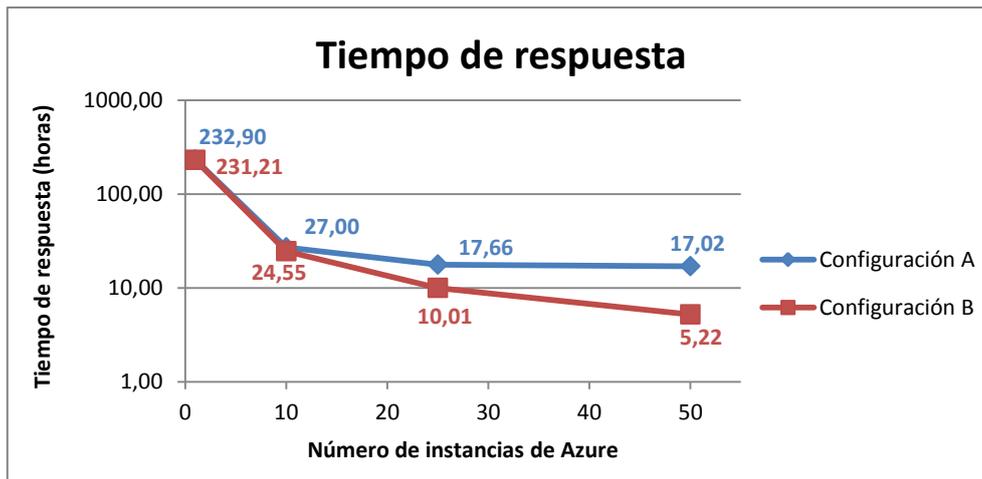


Figura 27: Tiempo de respuesta en horas obtenido en la ejecución del caso de uso completo.

En cualquier caso, cabe destacar la drástica reducción del tiempo que se ha obtenido en el cálculo mediante el uso servicio Cloud. Mientras que para la aproximación tradicional se necesitan más de 4 días para llevar a cabo el análisis del caso de estudio completo, únicamente fueron necesarias 17,02 horas para el caso de la configuración A y 5,22 horas para la configuración B cuando las simulaciones se ejecutaron en el Cloud.

La Figura 28 muestra el comportamiento del sistema Cloud en términos de incremento de velocidad o *speed-up*, cuando se aumenta el número de instancias en ejecución en el servicio para el cálculo del caso de uso completo, para ambas configuraciones, respecto a la aproximación secuencial y respecto a la ejecución sobre una instancia del servicio.

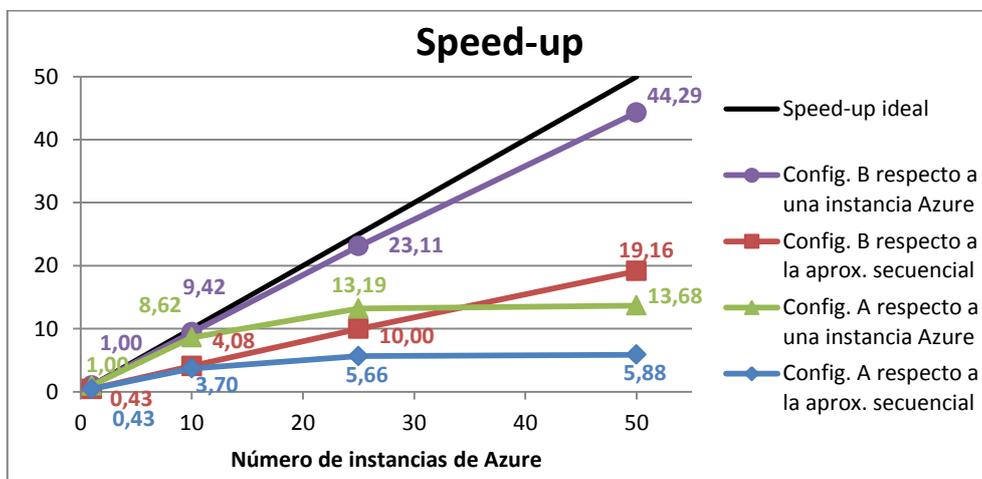


Figura 28: Speed-up obtenido en la ejecución del caso de estudio completo.

En la ilustración se puede apreciar cómo los resultados de speed-up, en comparación con la aproximación tradicional, son mucho menores que respecto a la ejecución sobre una instancia del servicio, para ambas configuraciones. Esta diferencia en los resultados se debe principalmente a la diferencia entre el hardware de la máquina local y de las instancias de Azure, como ya ha sido mencionado previamente. Mientras que el speed-up en la configuración B respecto a una instancia de Azure obtiene valores cercanos al caso ideal (44,24 para el caso de 50 instancias), el speed-up con respecto a la aproximación secuencial se encuentra lejos de estos valores (19,16 en el caso con 50 instancias). En el caso de la configuración A, se puede apreciar cómo el valor del speed-up no aumenta de manera apreciable cuando el número de instancias de Azure es

superior a 25, debido a las razones mencionadas anteriormente sobre el efecto de “cuello de botella” que ocurre en la red durante la recuperación de los resultados.

El comportamiento del sistema Cloud en términos de eficiencia se detalla en la Figura 29. Esta Figura muestra de nuevo los resultados respecto a la aproximación secuencial y respecto a una instancia de Azure, para ambas configuraciones. Como se esperaba, se obtuvieron muy buenos valores de eficiencia para el caso de la configuración B (0,89 respecto a una instancia de Azure para el caso con 50 instancias), pero se obtuvieron peores resultados de eficiencia para la configuración A. Claramente se puede observar cómo el decremento de la eficiencia es muy reducido para el caso de la configuración B cuando se aumenta el número de instancias en ejecución en el servicio Cloud, mientras que la tendencia del decremento de la eficiencia para el caso de la configuración A es mucho más pronunciada.

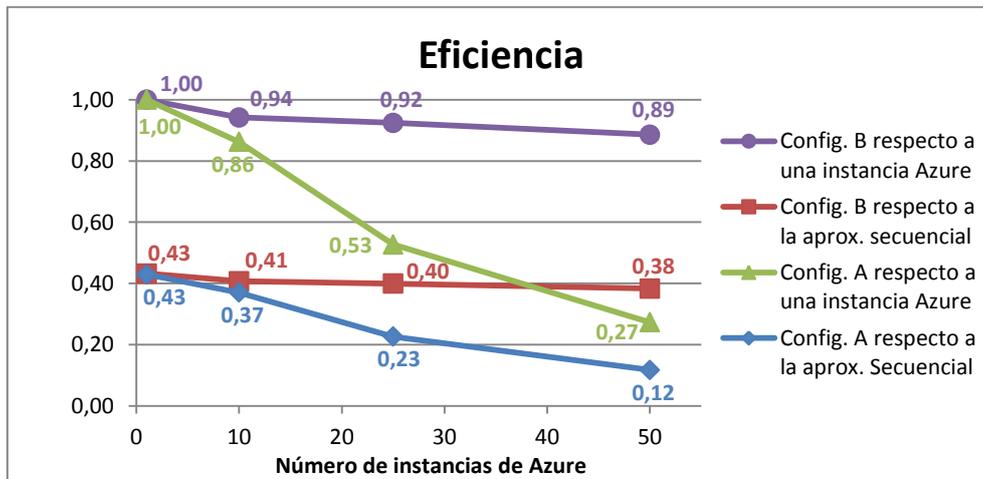


Figura 29: Eficiencia obtenida en la ejecución del caso de estudio completo.

En todo caso, se corrobora cómo el Servicio Cloud de Análisis Estructural permite reducir los tiempos de cálculo y, por consiguiente, reducir el tiempo total requerido en el diseño de estructuras de gran envergadura.

6 Conclusiones Finales y Trabajos Futuros

El presente Trabajo Fin de Máster muestra el uso de las tecnologías Cloud, apoyándose en el modelo de Plataforma como Servicio, para ofrecer una solución dedicada a la resolución de un problema complejo en el mundo de la ingeniería, como es el análisis dinámico en el tiempo de estructuras de edificación. Para ello se presenta el análisis, diseño e implementación de un Servicio Cloud de Análisis Estructural, basado en el componente Generic Worker de la plataforma VENUS-C, y desplegado sobre la infraestructura computacional de Microsoft a través de Windows Azure. El servicio permite a la comunidad de arquitectos e ingenieros llevar a cabo un análisis estructural, tanto de tipo estático como dinámico, de estructuras de gran envergadura, de manera eficiente, transparente, y en recursos computacionales remotos a través de la red, por medio de la herramienta Architrave. El SCAE lleva a cabo el cálculo de estructuras a través de un simulador basado en Computación de Altas Prestaciones que, junto con el modelo de Computación de Alta Productividad en el que está basado el servicio de análisis, obtiene buenas prestaciones en términos de tiempos de respuesta.

El diseño y la implementación del servicio Cloud han estado dirigidos a satisfacer los requisitos de usabilidad de la solución, como la sencillez o la transparencia en el uso del servicio. La definición de los parámetros de las simulaciones por medio de Architrave Cálculo se realizan del mismo modo que las ejecuciones locales, donde únicamente resulta necesario especificar de manera adicional algunas opciones relativas al cálculo remoto como la cantidad de resultados que se desean almacenar en el almacenamiento Cloud, la selección de la descarga automática o manual de los resultados, y la cantidad de resultados que se desean descargar (todos los resultados o únicamente los más desfavorables). Esto supone el análisis de estructuras por medio del SCAE de manera sencilla y natural. Por otro lado, el Gestor Cliente de Simulaciones Remotas desarrollado, cuyo funcionamiento se integra con Architrave, permite gestionar de manera sencilla las simulaciones enviadas al servicio Cloud, manteniendo en todo momento informado al usuario del estado y del progreso de las mismas. El Gestor ofrece también la posibilidad de cancelar los trabajos remotos cuando se encuentran en ejecución o en espera, además de descargar y eliminar los resultados remotos de análisis, tanto de manera manual como automática, en base a las opciones definidas.

Junto con los requisitos anteriores, la implementación del servicio Cloud ha tenido en consideración otros requerimientos no funcionales de carácter relevante, como la productividad, la elasticidad, la fiabilidad y la seguridad. El SCAE se basa en un modelo de Computación de Alta Productividad, donde múltiples clientes emplean de manera concurrente el servicio y diversas soluciones estructurales se analizan al mismo tiempo. Dicho servicio incluye además un componente que permite ajustar automáticamente el número de instancias de cómputo en ejecución, en base a la carga de trabajo del sistema. El servicio es por tanto capaz de soportar picos de demanda o incrementos transitorios de la carga, pero sin la necesidad de hacer un uso continuo de una gran infraestructura computacional infrutilizada, reduciendo los costes de consumo de recursos en base al uso real de los mismos. Asimismo, el despliegue de un servicio basado en Windows Azure, como es el caso, incluye de manera inherente un mecanismo de balanceo de la carga, así como mecanismos de tolerancia y recuperación ante fallos que garantizan una considerable calidad del servicio. Por último, el SCAE incluye diferentes mecanismos de seguridad, permitiendo el acceso únicamente a los usuarios autenticados y autorizados al uso del mismo, protegiendo las comunicaciones entre la máquina cliente y el servicio Cloud, preservando la integridad y la privacidad de las

mismas, y permitiendo a los usuarios del servicio únicamente el acceso a los datos del Cloud Storage de los cuales son propietarios.

Para evaluar la capacidad del sistema, se ha ejecutado un caso real que ha consistido en la evaluación del comportamiento de diez soluciones estructurales diferentes procedentes del mismo modelo estructural, ante la actuación de cinco terremotos distintos sobre cada una de ellas, dando lugar en total a 50 simulaciones diferentes e independientes entre sí. Este análisis ha permitido analizar, ante diferentes tamaños o número de instancias en ejecución del servicio de análisis estructural, las prestaciones del sistema en cuanto al tiempo de respuesta de la ejecución del caso de estudio completo en el Cloud, el incremento de velocidad frente a la ejecución en la máquina local de manera secuencial y la eficiencia del mismo. Para ello, se han evaluado dos configuraciones diferentes. En la primera de ellas, todos los resultados de cálculo se almacenan en el Cloud Storage y se descargan en la máquina local del cliente. En la segunda, únicamente se almacenan en el Cloud y se descargan en la máquina cliente los resultados correspondientes a los tres instantes de tiempo más desfavorables.

Los resultados del análisis muestran cómo, en el caso de la configuración donde únicamente se consideran los tres instantes más desfavorables, el comportamiento del sistema muestra una mejora progresiva conforme aumenta el número de instancias en ejecución. Esto no ocurre así, sin embargo, en la configuración con descarga de todos los resultados a partir de un número de instancias determinado, debido principalmente al elevado volumen de datos generado y a la capacidad de la red. En cualquier caso, es importante considerar la reducción del tiempo de respuesta que se obtiene mediante el uso del servicio Cloud frente a la ejecución secuencial en la máquina local, gracias a la ejecución de simulaciones de manera concurrente mediante el uso de múltiples instancias de cómputo.

Por lo tanto, las tecnologías Cloud se muestran como una alternativa apropiada para satisfacer las necesidades computacionales y de almacenamiento de las herramientas de cálculo estructural. El diseño de soluciones para el análisis dinámico de estructuras de edificación basadas en este tipo de tecnologías proporciona una serie de ventajas importante. Desde el punto de vista de los desarrolladores de software, el Cloud permite la implementación de entornos robustos de alta productividad de manera relativamente sencilla. Por otro lado, los proveedores de soluciones Cloud pueden disponer de un nuevo servicio que ajusta de manera automática el número de recursos computacionales empleados, evitando de este modo costes innecesarios. Por último, se ofrece a los usuarios un nuevo modelo de servicio basado en el pago por el uso realizado, sin la necesidad de emplear sus propios recursos computacionales y facilitándoles el acceso a soluciones software más avanzadas y eficientes que no adquieren en propiedad.

Concretamente, en el caso de las soluciones para el análisis de estructuras, las tecnologías Cloud ofrecen nuevas posibilidades a la comunidad de arquitectos e ingenieros de estructuras. Por ejemplo, la comunidad científica puede llevar a cabo un mayor número de experimentos por unidad de tiempo, pudiendo acelerar así la generación de resultados de investigación, o llevar a cabo análisis más completos y exhaustivos para mejorar sus resultados. Asimismo, la comunidad de ingenieros de estructuras y arquitectos podrá abordar estructuras más complejas y llevar a cabo un mayor número de simulaciones dinámicas, incluso más precisas, sin la necesidad de emplear simplificaciones, facilitando así también el diseño de estructuras más fiables y seguras. Además, la simulación de múltiples soluciones estructurales simultáneamente permite reducir el tiempo empleado durante el proceso de diseño de estructuras de

edificación, de modo que las empresas de ingeniería y los estudios de arquitectura pueden incrementar fácilmente su productividad y su volumen de negocio.

Dentro del trabajo realizado, cabe destacar la labor de documentación realizada para el desarrollo del presente Trabajo Fin de Máster. La elaboración del mismo ha requerido el estudio de los conceptos básicos sobre análisis de estructuras, así como las diferentes técnicas para su implementación en entornos computacionales. Además, ha sido necesario el análisis de múltiples tecnologías alternativas para el diseño de sistemas distribuidos, y de las características que presenta cada una de ellas. Asimismo, con el objeto del diseño de una solución Cloud para el análisis de estructuras, ha resultado imprescindible el estudio de las características que ofrecen los diferentes modelos de servicio y de despliegue Cloud, las ventajas e inconvenientes que aporta cada uno de ellos, las tecnologías en que se basa el Cloud, y los beneficios que aporta sobre las soluciones tradicionales. También, con el fin de diseñar una solución Cloud innovadora para el análisis de estructuras de edificación, ha sido necesario el estudio del estado del arte del uso de soluciones distribuidas para análisis estructural. Por último, como se menciona anteriormente, la solución implementada para el análisis de estructuras se basa en el middleware VENUS-C y en la plataforma Windows Azure. Por ello, resulta importante denotar la labor de estudio de los diferentes módulos o servicios que componen ambas herramientas y de las características que ofrecen cada uno de ellos, así como la labor de aprendizaje necesario del uso de dichas herramientas para el desarrollo de servicios Cloud.

Finalmente, es importante matizar las aportaciones realizadas durante el desarrollo del presente trabajo. Como se menciona previamente, este trabajo ha requerido una importante labor de documentación y formación en campos tan diversos como el análisis estructural, las tecnologías para el desarrollo de aplicaciones distribuidas y computación Cloud, así como de estudio de las características y de los servicios que ofrecen los componentes en los que basa la solución Cloud desarrollada. El trabajo realizado ha dado lugar al servicio Cloud mencionado para el análisis de estructuras, lo que ha implicado el desarrollo de diferentes componentes para poder ofrecer dicho servicio a los usuarios. Dicho desarrollo incluye la implementación del Gestor del Servicio de Análisis Estructural, encargado de la invocación del Simulador Estructural y el postprocesamiento de los resultados generados por el mismo, y del Gestor de Elasticidad, responsable de ajustar automáticamente el número de instancias en ejecución del servicio en función de la carga del sistema. Además, con el fin de facilitar a los usuarios la gestión de las simulaciones remotas y la interacción con el servicio Cloud desarrollado, se ha llevado a cabo el diseño y la implementación del Gestor Cliente de Simulaciones Remotas, de los diferentes componentes internos de los que consta, y del mecanismo de comunicación con Architrave Cálculo para el envío de simulaciones al servicio Cloud desde dicha herramienta. Asimismo, dicho trabajo ha comprendido también la ejecución de un caso de uso real para verificar el correcto funcionamiento del servicio y evaluar el comportamiento del mismo ante diferente número de instancias de cómputo en ejecución. Por último, el trabajo desarrollado ha dado lugar a dos publicaciones en congresos científicos [79] [80], como se menciona en el siguiente capítulo, además de diferentes apariciones en diversas conferencias [81] [82] [83] y noticias de prensa [84] [85] [86].

6.1 Trabajos Futuros

La solución Cloud presentada para el análisis de estructuras de edificación se basa en el uso del componente Generic Worker de la plataforma VENUS-C y se ejecuta sobre la infraestructura de Microsoft por medio de Windows Azure. En esta implementación, cuando finaliza la ejecución de todos los trabajos las instancias se encuentran a la espera de atender otros nuevos, y es el componente Elasticity Manager quien se encarga de monitorizar la carga del sistema y aumentar o disminuir en consecuencia el número de instancias del servicio en ejecución. Una alternativa a este modelo de ejecución de trabajos sucede mediante el uso de los desarrollos del proyecto CodeCloud.

El entorno de ejecución de CodeCloud tiene la capacidad, tanto de manera automática en función de la carga del sistema como en base a los requisitos especificados para una tarea, de crear dinámicamente un conjunto de máquinas virtuales para el procesamiento de trabajos, a través por ejemplo de las plataformas OpenStack y OpenNebula. De este modo, CodeCloud permite ajustar de manera más precisa la cantidad de recursos computacionales necesarios, creando y desplegando nuevas instancias al recibir peticiones de nuevos trabajos cuando la cantidad de recursos computacionales libres es insuficiente, y eliminando dichos recursos nada más finalizar la ejecución de las tareas cuando no existen trabajos para procesar. En cambio, debido a la rápida capacidad de adaptación de los recursos computacionales del sistema, este tipo de solución podría ver penalizado el tiempo de respuesta de los trabajos en determinados escenarios como consecuencia del tiempo empleado en el despliegue de nuevos recursos computacionales, como por ejemplo en aquellos donde la carga de trabajos es transitoria con periodos relativamente bajos. Por ello, una posible línea de trabajo consiste en la implementación del SCAE basándose en el uso de CodeCloud, comparando el comportamiento de esta nueva implementación frente a la solución basada en el Generic Worker en términos de tiempo de respuesta de los trabajos, rendimiento del sistema y coste total de los recursos computacionales empleados, ante un modelo de carga real donde diferentes usuarios emplean el servicio para la ejecución de múltiples simulaciones estructurales.

Por otro lado, el modelo de Plataforma como Servicio que proporcionan los proveedores de servicios Cloud facilita la labor de diseño y desarrollo de soluciones Cloud y libera de determinadas responsabilidades de gestión del servicio. En cambio, este modelo introduce cierta sobrecarga en el comportamiento del sistema y convierte la solución desarrollada en *vendor lock-in* o dependiente del proveedor. Por ello, otra nueva línea de trabajo consistiría en el desarrollo de una implementación nativa del servicio Cloud para análisis de estructuras sobre la plataforma Windows Azure, que permitiera evaluar la sobrecarga que introduce el uso de la plataforma VENUS-C sobre Windows Azure y la posible mejora en el rendimiento de la misma frente a la implementación anterior.

Siguiendo la misma línea, el desarrollo de una solución Cloud propia basándose en el modelo de Infraestructura como Servicio, como el ofrecido por Amazon EC2, permitiría determinar la sobrecarga que introduce el uso de plataformas Cloud en el caso del análisis dinámico de estructuras de edificación, así como la posible conveniencia o no del uso de las mismas para la resolución de dicho problema de ingeniería. Además, esta propuesta permitiría evaluar y comparar entre las diferentes alternativas otros aspectos importantes en el diseño de soluciones Cloud, como la curva de aprendizaje del uso de las herramientas Cloud, la facilidad y el esfuerzo en desarrollo que supone el proceso de migrar al Cloud aplicaciones ya desarrolladas, y el tiempo de comercialización o *“time to market”* que supone cada una de las alternativas.

Por último, una nueva línea de trabajo que resulta de gran interés consistiría en el diseño e implementación de una solución Cloud para el análisis de estructuras basada en la migración de la funcionalidad de la herramienta Architrave Cálculo para su funcionamiento en un entorno web, por medio de tecnologías como HTML5, para la gestión de las acciones y las interacciones con el usuario, y WebGL para el renderizado de manera eficiente de los modelos estructurales por medio del navegador web. La integración de dicha herramienta en el entorno Cloud dentro del SCAE ofrece numerosas ventajas. Por ejemplo, la implementación SaaS consistiría en sí mismo en una solución multiplataforma, independiente del sistema operativo o de la arquitectura del dispositivo que hace uso de dicho servicio. Además, esta solución permitiría a los usuarios acceder y gestionar sus proyectos estructurales desde cualquier ubicación y por medio de cualquier dispositivo que soporte las tecnologías mencionadas, incrementando sus posibilidades de trabajo. Asimismo, este tipo de soluciones favorece la compartición de datos entre la comunidad de usuarios y facilita las labores de trabajo en equipo.

En cambio, una solución basada completamente en el modelo SaaS a través introduce importantes inconvenientes. Por ejemplo, desde la perspectiva de los usuarios, tanto el uso del servicio como la experiencia del usuario se ven fuertemente condicionadas por el estado y la capacidad de la red, impidiendo la posibilidad de trabajar cuando no se dispone de acceso a la misma, a diferencia de la aproximación tradicional basada en la instalación de la aplicación en la máquina local. Además, las interacciones del usuario con la aplicación a través del navegador suelen presentar un rendimiento y una fluidez mucho menores que en una aproximación tradicional. Por otro lado, desde la perspectiva del desarrollador, el proceso de portabilidad de una aplicación compleja a un entorno web puede convertirse en una labor extremadamente complicada y costosa. Por lo tanto, para optar por una solución SaaS completa para el análisis de estructuras de edificación en la Nube, y por tanto llevar a cabo el proceso de desarrollo y de migración al Cloud de la aplicación de análisis estructural, resulta necesario llevar a cabo previamente un estudio para determinar las posibilidades, la viabilidad y la conveniencia de la implementación frente a la aproximación Cloud actual.

7 Publicaciones Realizadas

Los desarrollos del presente Trabajo Fin de Máster han dado lugar a las siguientes publicaciones científicas:

1. J.M. Alonso, A. Alonso, P. de la Fuente, F. Gómez, V. Hernández, P. Lozano, A. Pérez. “A Cloud System Implementation for the Analysis of Civil Engineering Structures”, in: Proceedings of the 2012 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA12), CSREA Press, 2012, pp. 210—216.
2. M. Caballer, P. de la Fuente, P. Lozano, J.M. Alonso, C. de Alfonso. “Easing the Structural Analysis Migration to the Cloud”, in: Proceedings of the 7th Iberian Grid Infrastructure Conference (IBERGRID 2013), Universitat Politècnica de València, 2013, pp. 159—171.

Del 16 al 19 de julio de 2012 se celebró en Las Vegas (EE.UU) el congreso “The 2012 International Conference on Parallel and Distributed Processing Techniques and Applications” (PDPTA12). Este congreso internacional, indexado como tipo B según [87], y cuyo ratio de aceptación fue del 28% de los artículos enviados, aceptó el artículo “A Cloud System Implementation for the Analysis of Civil Engineering Structures”. Esta publicación presentó los requisitos, la arquitectura y las características de un servicio Cloud para el análisis de estructuras de edificación basado en el componente Generic Worker perteneciente a la plataforma VENUS-C, donde además se analizó el comportamiento de dicha solución mediante la ejecución de un caso de estudio.

Además, los días 19 y 20 de septiembre tuvo lugar en Madrid (España) el congreso “7th Iberian Grid Infrastructure Conference” (IBERGRID 2013), donde fue aceptado para su publicación el artículo “Easing the Structural Analysis Migration to the Cloud”. Esta publicación describió la migración de una aplicación científica de análisis estructural al Cloud por medio de CodeCloud, donde además se comparó, en términos de tiempo de respuesta, el rendimiento de dicha aproximación frente a la del artículo anterior.

El trabajo desarrollado ha dado lugar, además, a diferentes participaciones en conferencias:

1. I. Blanquer. “User Scenarios in VENUS-C - Focus on Structural Analysis”, in: Cloudscape III – Taking European Cloud Infrastructure Forward. Brussels, Belgium. Marzo, 2011.
2. J.M. Alonso. “VENUS-C - Architrave: Un Sistema para el Diseño y Análisis de Estructuras en la Nube”, in: 6^o Reunión Plenaria de la Red Española de e-Ciencia. Centro Superior de Investigaciones Científicas, Madrid. Mayo, 2012
3. I. Blanquer. “From research-to-market - turning scientific results into business opportunities VENUS-C success stories”, in: Cloudscape V – Cloud for savings, Cloud for quality. Brussels, Belgium. Marzo, 2013.

Por último, la participación en el proyecto VENUS-C como escenario de análisis estructural ha sido referenciada en diversas ocasiones por la prensa nacional [84] [85] e internacional [86].

8 Bibliografía

- [1] R.K. Livesley, *Métodos matriciales para el cálculo de estructuras.*: H. Blume Ediciones, 1970.
- [2] Documento Básico: Seguridad Estructural - Acciones en la Edificación. Código Técnico de la Edificación. [Online]. http://www.codigotecnico.org/cte/export/sites/default/web/galerias/archivos/DB_SE-AE_abril_2009.pdf
- [3] D.E. Beskos, S.A. Anagnostopoulos, *Computer Analysis and Design of Earthquake Resistant Structures: A Handbook.*: Computational Mechanics Publications, 1997.
- [4] The Internet Engineering Task Force (IETF). (2009, Noviembre) RFC 5694 - Peer-to-Peer (P2P) Architecture: Definition, Taxonomies, Examples, and Applicability. [Online]. <http://tools.ietf.org/html/rfc5694>
- [5] B. J. Nelson, A. D. Birrel, "Implementing remote procedure calls," *ACM Trans. Comput. Syst.*, vol. 2, no. 1, pp. 39-59, 1984.
- [6] A. Reuter, J. Gray, *Transaction Processing: Concepts and Techniques.*: Morgan Kaufmann Publishers Inc., 1992.
- [7] J. Chacko et al, "Object Request Broker," US 2011/0271288 A1, 2011.
- [8] Object Management Group (OMG). Common Object Request Broker Architecture (CORBA), version 3.3. [Online]. <http://www.omg.org/spec/CORBA/3.3/>
- [9] E. Curry, "Message-Oriented Middleware," in *Middleware for Communications.*: John Wiley and Sons, 2004, pp. 1-28.
- [10] The World Wide Web Consortium (W3C). Web Services Architecture. [Online]. <http://www.w3.org/TR/ws-arch/>
- [11] The World Wide Web Consortium (W3C). SOAP Version 1.2 Part 1: Messaging Framework (Second Edition). [Online]. <http://www.w3.org/TR/soap12-part1/>
- [12] The World Wide Web Consortium (W3C). Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language. [Online]. <http://www.w3.org/TR/wsdl20/>
- [13] Organization for the Advancement of Structured Information Standards (OASIS). UDDI Version 3.0.2. [Online]. <https://www.oasis-open.org/committees/uddi-spec/doc/spec/v3/uddi-v3.0.2-20041019.htm>

- [14] R.T. Fielding, R.N. Taylor, "Principled design of the modern Web architecture," *ACM Trans. Internet Technol.*, vol. 2, no. 2, pp. 115-150, 2002.
- [15] R. Buyya, J. Broberg, and A. M. Goscinski, *Cloud computing: Principles and Paradigms.*: Willey, 2011.
- [16] H. Garfinkel and S.L. Abelson, *Architects of the Information Society: 35 Years of the Laboratory for Computer Science at MIT.*: MIT Press, 1999.
- [17] P. Mell and T. Grance. (2009) Effectively and Securely Using the Cloud Computing Paradigm - NIST, Information Technology Laboratory. [Online]. http://csrc.nist.gov/organizations/fissea/2009-conference/presentations/fissea09-pmell-day3_cloud-computing.pdf
- [18] T. Grance and P. Mell. (2011, September) The NIST Definition of Cloud Computing. [Online]. <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>
- [19] CloudU. Understanding the Cloud Computing Stack: SaaS, PaaS, IaaS. [Online]. http://broadcast.rackspace.com/hosting_knowledge/whitepapers/Understanding-the-Cloud-Computing-Stack.pdf
- [20] R. L. Krutz and R. D. Vines, *Cloud Security: A Comprehensive Guide to Secure Cloud Computing.*: John Wiley & Sons, 2010.
- [21] A. S. Tanenbaum and M. van Steen, *Distributed Systems: Principles and Paradigms*, 2nd ed.: Prentice-Hall, Inc., 2006.
- [22] B. Sosinsky, *Cloud Computing Bible*, 1st ed.: Wiley Publishing, 2011.
- [23] Rackspace, US Inc. (2013) Cloud Server Infrastructure Hosting - Rackspace Cloud Servers. [Online]. <http://www.rackspace.com/cloud/servers/>
- [24] Amazon Web Services, Inc. (2013) Amazon Elastic Compute Cloud (Amazon EC2), Cloud Computing Servers. [Online]. <http://aws.amazon.com/ec2/>
- [25] Microsoft. Windows Azure: Plataforma en la nube de Microsoft | Hospedaje en la nube | Servicios en la nube. [Online]. <http://www.windowsazure.com/>
- [26] VENUS-C - Virtual multidisciplinary Environments USing Cloud Infrastructures. [Online]. <http://www.venus-c.eu/Pages/Home.aspx>
- [27] C. de Alfonso, M. Caballer, F. Alvarruiz, G. Moltó, V. Hernández, "Infrastructure deployment over the Cloud," *2011 Third IEEE International Conference on Cloud Computing Technology and Science*, pp. 517-521, 2011.
- [28] Barcelona Supercomputing Center. (2013) COMP Superscalar | BSC-CNS. [Online]. <http://www.bsc.es/computer-sciences/grid-computing/comp-superscalar>

- [29] OpenNebula Project. OpenNebula - Open Source Data Center Virtualization. [Online]. <http://opennebula.org/>
- [30] OpenStack project. (2013) OpenStack Open Source Cloud Computing Software. [Online]. <http://www.openstack.org/>
- [31] Google Inc. (2013) Google Apps for Business. [Online]. <http://www.google.com/enterprise/apps/business/>
- [32] Salesforce.com, Inc. (2013) Sales Cloud: Sales Force Automation Tools - Salesforce.com. [Online]. <http://www.salesforce.com/sales-cloud/overview/>
- [33] Google Inc. (2013) Google Drive. [Online]. <http://www.google.com/drive/about.html>
- [34] VMware Inc. (2013) Private Cloud Computing with VMware Virtualization. [Online]. <http://www.vmware.com/solutions/cloud-computing/private-cloud/>
- [35] ActiveState Software Inc. (2013) Stackato | The secure cloud application platform for the enterprise | ActiveState. [Online]. <http://www.activestate.com/stackato>
- [36] Rackspace, US Inc. (2013) Hybrid Cloud - Secure & Scalable Converged Infrastructure by Rackspace. [Online]. <http://www.rackspace.com/cloud/hybrid>
- [37] EMOTIVE Cloud (Elastic Management Of Tasks In Virtualized Environments). [Online]. <http://autonomic.ac.upc.edu/emotive/>
- [38] Amazon Web Services, Inc. Query Requests - Amazon Elastic Compute Cloud. [Online]. <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-query-api.html>
- [39] (2013) Open Cloud Computing Interface | Open Standard | Open Community. [Online]. <http://occi-wg.org/>
- [40] VMware, Inc. (2013) VMware vCloud Suite for Cloud Computing & Cloud Management. [Online]. <http://www.vmware.com/products/datacenter-virtualization/vcloud-suite/overview.html>
- [41] Xen Project. (2013) The Xen Project, the powerful open source industry standard for virtualization. [Online]. <http://www.xenproject.org/>
- [42] (2013) Main Page - KVM. [Online]. <http://www.linux-kvm.org/>
- [43] VMware Inc. (2013) VMware Virtualization for Desktop & Server, Public & Private Clouds. [Online]. <http://www.vmware.com/>
- [44] ORACLE. (2013) Oracle VM VirtualBox. [Online]. <https://www.virtualbox.org/>

- [45] Microsoft. Intro to Windows Azure - Windows Azure fundamentals. [Online]. <http://www.windowsazure.com/en-us/develop/net/fundamentals/intro-to-windows-azure/>
- [46] C. Geuer-Pollmann, "The VENUS-C Generic Worker," in *1st anual SICS Cloud Day*, 2011.
- [47] I. Foster et al. (2007) OGSA Basic Execution Service Version 1.0. Technical report, Open Grid Forum. [Online]. <http://www.ogf.org/documents/GFD.108.pdf>
- [48] VENUS-C. Deliverable 6.4 – Programming Models - Prototypes. [Online]. http://www.venus-c.eu/deliverables_year2/VENUS-C_D6.4.pdf
- [49] D. Lezzi, R. Rafanell, F. Lordan, E. Tejedor, R.M. Badia, "COMPSs in the VENUS-C Platform: enabling e-Science applications on the Cloud," *5th Iberian Grid Infrastructure Conference (IBERGRID 2011)*, June 2011.
- [50] Storage Networking Industry Association (SNIA). (2012, June) Cloud Data Management Interface (CDMI), Version 1.0.2. [Online]. <http://snia.org/sites/default/files/CDMI%20v1.0.2.pdf>
- [51] VENUS-C. Deliverable 6.5 - Application Security – Prototypes. [Online]. http://www.venus-c.eu/deliverables_year2/VENUS-C_D6.5.pdf
- [52] VENUS-C. Deliverable 6.3 - Data Management – Prototypes. [Online]. http://www.venus-c.eu/deliverables_year2/VENUS-C_D6.3.pdf
- [53] VENUS-C. Deliverable 6.6 - Accounting, Monitoring and Billing – Prototypes. [Online]. http://www.venus-c.eu/deliverables_year2/VENUS-C_D6.6.pdf
- [54] VENUS-C. Deliverable 6.7 - Networking & Security – Prototypes (TRE). [Online]. http://www.venus-c.eu/deliverables_year2/VENUS-C_D6.7.pdf
- [55] K.H. Law, J. Peng, "Building Finite Element Analysis Programs in Distributed Services Environment," *Computers & Structures*, 2004.
- [56] M. Dolenc, "Web Services for Finite Element Analysis," *Proceedings of the Fourth International Conference on Engineering Computational Technology*, 2004.
- [57] J.M. Alonso, V. Hernández, R. López, G. Moltó, "A Service Oriented System for on Demand Dynamic Structural Analysis over Computational Grids," *High Performance Computing for Computational Science - VECPAR*, 2006.
- [58] G. Wei, Y. Zheng, J. Zhang, *Grid Service-Based Parallel Finite Element Analysis.*: Springer Berlin Heidelberg, 2004, pp. 123-130.

- [59] B. Kumar and J.C.P. Cheng, "Cloud Computing and its Implications for Construction IT," *Proceedings of the 13th International Conference on Computing in Civil and Building Engineering (ICCCBE XIII)*, July 2012.
- [60] T.H. Beach, Y. Rezgui, O.F. Rana, "CloudBIM: Management of BIM Data in a Cloud Computing Environment," *Proceedings of the 28th International Conference of CIB W78*, October 2011.
- [61] Z. Xin, L. Xiao-qun, F. Hong, C. Zhen-li, "Structural Finite Element Method Based on Cloud Computing," *Proceedings of the 2012 International Conference on Computer Science and Electronics Engineering (ICCSEE '12)*, vol. 3, pp. 98-101, 2012.
- [62] (2013) The UberCloud HPC Experiment: Compendium of Case Studies. [Online]. <https://intranet.birmingham.ac.uk/it/teams/infrastructure/fm/bear/documents/public/the-hpc-experiment-.pdf>
- [63] J. Garcia, E. Bayo, "Integrated 3D Web Application for Structural Analysis Software as a Service," *Journal of Computing in Civil Engineering*, pp. 159-166, March 2013.
- [64] Khronos Group. (2013) WebGL - OpenGL ES 2.0 for the Web. [Online]. <http://www.khronos.org/webgl/>
- [65] World Wide Web Consortium (W3C). (2013) HTML 5.1 Nightly - A vocabulary and associated APIs for HTML and XHTML. [Online]. <http://www.w3.org/html/wg/drafts/html/master/>
- [66] (W3C), World Wide Web Consortium. (2012, December) XMLHttpRequest. [Online]. <http://www.w3.org/TR/XMLHttpRequest/>
- [67] D. Crockford. (2006) RFC 4627 - The application/json Media Type for JavaScript Object Notation (JSON). [Online]. <http://tools.ietf.org/html/rfc4627>
- [68] I. Ari, N. Muhtaroglu, "Design and implementation of a cloud computing service for finite element analysis," *Advances in Engineering Software*, vol. 60-61, pp. 122-135, November 2012.
- [69] G. Dhondt, K. Wittig. (1998) CALCULIX: A Three-Dimensional Structural Finite Element Program. [Online]. <http://www.calculix.de/>
- [70] Y. Arita, N. Nozaki, K. Demizu, "Mechanical Design Platform on Engineering Cloud," *FUJITSU Scientific & Technical Journal (FSTJ)*, vol. 48, no. 4, pp. 422-427, October 2012.
- [71] M. Yasuda, "Fujitsu's Engineering Cloud," *FUJITSU Scientific & Technical Journal (FSTJ)*, vol. 48, no. 4, pp. 404-412, October 2012.

- [72] T. Simmhan, C. van Ingen, G. Subramanian, J. Li, "Bridging the Gap between the Cloud and an eScience Application Platform," *Proceedings of the 2010 IEEE 3rd International Conference on Cloud Computing*, pp. 474-481, 2009.
- [73] Autodesk Inc. (2013) Autodesk Revit. [Online]. <http://www.autodesk.com/products/autodesk-revit-family/overview>
- [74] Autodesk, Inc. Structural Analysis for Autodesk® Revit®. [Online]. <https://structuralanalysis.360.autodesk.com/>
- [75] Architrave. [Online]. <http://www.architrave.es/>
- [76] J.M. Alonso, V. Hernández, "Three-dimensional structural dynamic analysis using parallel direct time integration methods," *Proceedings of the Fifth International Conference on Engineering Computational Technology*, no. paper 120, 2006.
- [77] J.M. Alonso, V. Hernández, "A parallel implementation of 3D modal analysis of building structures," *Proceedings of the Tenth International Conference on Civil, Structural and Environmental Engineering Computing*, no. paper 232, 2005.
- [78] Ministerio de Fomento, Gobierno de España. Norma de Construcción Sismorresistente: Parte general y edificación (NCSE-02). [Online]. <http://www.fomento.es/NR/rdonlyres/949FF672-CB56-4332-BD7B-C408C2FCC05A/81030/0820200.pdf>
- [79] J.M. Alonso, A. Alonso, P. de la Fuente, F. Gómez, V. Hernández, P. Lozano, A. Pérez, "A Cloud System Implementation for the Analysis of Civil Engineering Structures," *Proceedings of the 2012 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA12)*, p. 210—216, 2012.
- [80] M. Caballer, P. de la Fuente, P. Lozano, J.M. Alonso, C. de Alfonso, "Easing the Structural Analysis Migration to the Cloud," *Proceedings of the 7th Iberian Grid Infrastructure Conference (IBERGRID 2013)*, p. 159—171, 2013.
- [81] I. Blanquer. (2011) User Scenarios in VENUS-C - Focus on Structural Analysis. Cloudscape III Presentations & videos. [Online]. <http://www.sienainitiative.eu/Repository/FileScaricati/a13a6df3-879f-4897-85ad-59defec702d3.pdf>
- [82] J.M. Alonso, "VENUS-C - Architrave: Un Sistema para el Diseño y Análisis de Estructuras en la Nube," in *6ª Reunión Plenaria de la Red Española de e-Ciencia*, Centro Superior de Investigaciones Científicas, Madrid, Mayo, 2012.
- [83] I. Blanquer. From research-to-market - turning scientific results into business opportunities VENUS-C success stories | Cloudscape. [Online]. <http://www.cloudscapeseries.eu/Content/CloudscapeSeries.aspx?id=268>

- [84] Crean una nueva "nube" para combatir incendios y lograr mejores medicamentos - ABC.es. [Online]. <http://www.abc.es/agencias/noticia.asp?noticia=1266277>
- [85] Noticia UPV | El proyecto VENUS-C, en el que participa la Universitat Politècnica de València, consiste en el desarrollo de una nueva plataforma de tecnología cloud que contribuirá a mejorar la competitividad de la investigación científica de la UE. [Online]. <http://www.upv.es/noticias-upv/noticia-4322-tecnologia-clou-es.html>
- [86] Von Felix Knoke, "Microsoft Labs: Die Wissensmacher," *CHIP.de*, Oktober 2012.
- [87] Computing Research and Education (CORE). Conference Rankings. [Online]. <http://core.edu.au/index.php/categories/conference%20rankings>