

UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR



Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

**CLOUD COMPUTING: FUNDAMENTOS Y DESPLIEGE DE
UN SERVICIO EN LA NUBE**

Sergio Sánchez Prado
Tutora: Ana María González Marcos

JUNIO 2021

CLOUD COMPUTING: FUNDAMENTOS Y DESPLIEGE DE UN SERVICIO EN LA NUBE

AUTOR: Sergio Sánchez Prado
TUTOR: Ana María González Marcos

Dpto. Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Junio de 2021

INDICE DE CONTENIDOS

1 INTRODUCCIÓN	1
1.1 MOTIVACIÓN.....	2
1.2 OBJETIVOS	3
2 PARADIGMA DEL CLOUD COMPUTING	5
2.1 PROPIEDADES BÁSICAS DE LA METODOLOGÍA CLOUD COMPUTING.....	5
2.2 MODELOS DE SERVICIO EXISTENTES.....	7
2.2.1 DIVISIÓN POR MODELOS DE SERVICIO	9
2.2.2 DIVISIÓN POR MODELOS DE IMPLEMENTACIÓN	11
2.2.3 DIVISIÓN POR AGENTES INTERVINIENTES.....	13
2.3 PRINCIPALES VENTAJAS E INCONVENIENTES	13
3 PRINCIPALES TECNOLOGÍAS SOBRE LAS CUALES SE ASIENTA LA METODOLOGÍA CLOUD COMPUTING	15
3.1 EQUILIBRADO DE CARGA.....	15
3.2 VIRTUALIZACIÓN	16
3.3 BASES DE DATOS NO RELACIONALES	18
4 SEGURIDAD Y ASPECTOS LEGALES	21
5 MERCADO ACTUAL DEL CLOUD COMPUTING	25
6 CASO PRÁCTICO: DESARROLLO Y DESPLIEGUE DE UN SERVICIO REAL EN LA NUBE.....	29
6.1 ARQUITECTURA	29
6.2 IMPLEMENTACIÓN DE UN RECURSO WEB ESTÁTICO.....	33
6.2.1 OBJETIVO.....	33
6.2.2 DESARROLLO	33
6.3 DESARROLLO DE UNA FUNCIÓN LAMBDA	34
6.3.1 OBJETIVO.....	34
6.3.2 DESARROLLO.....	34
6.4 INTEGRACIÓN DE LA APLICACIÓN WEB Y LA FUNCIÓN LAMBDA	34
6.4.1 OBJETIVO.....	34
6.4.2 DESARROLLO	35
6.5 CREACIÓN DE LA BASE DE DATOS NOSQL	35
6.5.1 OBJETIVO.....	35
6.5.2 DESARROLLO	35
6.6 DESPLIEGUE DE LA APLICACIÓN WEB EN LA NUBE	36
6.6.1 OBJETIVO.....	36
6.6.2 DESARROLLO.....	36
7 PRINCIPALES CORRIENTES EMERGENTES DERIVADAS DEL CLOUD COMPUTING: EDGE COMPUTING Y FOG COMPUTING.....	37
8 CONCLUSIONES.....	41
BIBLIOGRAFÍA	I
ANEXOS	IX
A BALANCEADORES DE CARGA	IX
B VIRTUALIZACIÓN.....	XV
C CASO PRÁCTICO: DESARROLLO Y DESPLIEGUE DE UN SERVICIO REAL EN LA NUBE – ANEXO EVIDENCIAS DE IMPLEMENTACIÓN	XVII
GLOSARIO.....	XXXIII

INDICE DE FIGURAS

FIGURA 1-1: EVOLUCIÓN DE LAS ARQUITECTURAS DE CALCULO COMPUTACIONAL	2
FIGURA 2-1: PRINCIPALES CARACTERÍSTICAS DE LA METODOLOGÍA CLOUD COMPUTING	6
FIGURA 2-2: MODELO DE CLASIFICACIÓN DE SOLUCIONES CLOUD COMPUTING	8
FIGURA 2-3: PRINCIPALES DIFERENCIAS ENTRE LOS MODELOS DE SERVICIO IAAS, PAAS Y SAAS	10
FIGURA 3-1: PRINCIPALES DIFERENCIAS ENTRE VIRTUALIZACIÓN Y CLOUD COMPUTING	17
FIGURA 5-1: GRÁFICA DEL CRECIMIENTO EN LA ADOPCIÓN DE SOLUCIONES CLOUD COMPUTING EN EMPRESAS (2011 – 2020)	26
FIGURA 6-1: CASO PRÁCTICO: DESARROLLO Y DESPLIEGUE DE UN SERVICIO REAL EN LA NUBE – DESCRIPCIÓN DE LA ARQUITECTURA DEL SISTEMA.....	30
FIGURA 6-2: REPRESENTACIÓN DEL MODELO LÓGICO FUNCIONAL DE LAS ARQUITECTURAS SERVERLESS	31
FIGURA 7-1: PRINCIPALES CARACTERÍSTICAS EN TORNO A LAS CORRIENTES EMERGENTES DERIVADAS DEL CLOUD COMPUTING: FOG COMPUTING Y EDGE COMPUTING	37
FIGURA 7-2: ESQUEMA DE LA PILA DE PROCESAMIENTO DE DATOS INDUSTRIALES IoT	39
FIGURA ANEXO A - 1: REPRESENTACIÓN GRÁFICA DE LA LÓGICA FUNCIONAL ASOCIADA AL MÉTODO DE BALANCEO - ROUND ROBIN.....	X
FIGURA ANEXO A - 2: REPRESENTACIÓN GRÁFICA DE LA LÓGICA FUNCIONAL ASOCIADA AL MÉTODO DE BALANCEO - WEIGHTED ROUND ROBIN	XI
FIGURA ANEXO A - 3: REPRESENTACIÓN GRÁFICA DE LA LÓGICA FUNCIONAL ASOCIADA AL MÉTODO DE BALANCEO - LEAST CONNECTION ¹²	XI
FIGURA ANEXO A - 4: REPRESENTACIÓN GRÁFICA DE LA LÓGICA FUNCIONAL ASOCIADA AL MÉTODO DE BALANCEO - WEIGHTED LEAST CONNECTION	XII
FIGURA ANEXO C - 5: IMPLEMENTACIÓN DE UN RECURSO WEB ESTÁTICO – FRAGMENTO DE CÓDIGO ASOCIADO A LA CODIFICACIÓN DEL RECURSO WEB ESTÁTICO	XVII
FIGURA ANEXO C - 6: IMPLEMENTACIÓN DE UN RECURSO WEB ESTÁTICO (II) – CREACIÓN DEL BACKEND PARA EL DESPLIEGUE DEL RECURSO WEB ESTÁTICO	XVII
FIGURA ANEXO C - 7: IMPLEMENTACIÓN DE UN RECURSO WEB ESTÁTICO (III) – CARGA Y DESPLIEGUE DEL RECURSO WEB EN AWS AMPLIFY	XVIII
FIGURA ANEXO C - 8: IMPLEMENTACIÓN DE UN RECURSO WEB ESTÁTICO (IV) – CARGA Y DESPLIEGUE DEL RECURSO WEB EN AWS AMPLIFY [2].....	XVIII
FIGURA ANEXO C - 9: IMPLEMENTACIÓN DE UN RECURSO WEB ESTÁTICO (V) – CARGA Y DESPLIEGUE DEL RECURSO WEB EN AWS AMPLIFY [3].....	XIX
FIGURA ANEXO C - 10: IMPLEMENTACIÓN DE UN RECURSO WEB ESTÁTICO (VI) –DESPLIEGUE COMO APLICACIÓN DEL RECURSO WEB IMPLEMENTADO MEDIANTE AWS AMPLIFY.....	XIX
FIGURA ANEXO C - 11: DESARROLLO DE UNA FUNCIÓN LAMBDA (I)- PANEL DE CREACIÓN DE UNA FUNCIÓN LAMBDA.....	XX
FIGURA ANEXO C - 12: DESARROLLO DE UNA FUNCIÓN LAMBDA (II)- FRAGMENTO DE CÓDIGO ASOCIADO A LA CODIFICACIÓN DE LA FUNCIÓN LAMBDA	XX

FIGURA ANEXO C - 13: DESARROLLO DE UNA FUNCIÓN LAMBDA (III)- CONFIGURACIÓN DEL EVENTO DE PRUEBA DE LA FUNCIÓN LAMBDA.....	XXI
FIGURA ANEXO C - 14: DESARROLLO DE UNA FUNCIÓN LAMBDA (IV)- RESULTADO DE LA EJECUCIÓN DEL CASO DE PRUEBA	XXI
FIGURA ANEXO C - 15: INTEGRACIÓN DE LA APLICACIÓN WEB Y LA FUNCIÓN LAMBDA - INTERFAZ PRINCIPAL DEL MENÚ DEL SERVICIO AMAZON API GATEWAY	XXII
FIGURA ANEXO C - 16: INTEGRACIÓN DE LA APLICACIÓN WEB Y LA FUNCIÓN LAMBDA (II) – CONFIGURACIÓN E INTEGRACIÓN DEL MÉTODO POTS DE LA API CON LA FUNCIÓN LAMBDA	XXII
FIGURA ANEXO C - 17: INTEGRACIÓN DE LA APLICACIÓN WEB Y LA FUNCIÓN LAMBDA (II) – CONFIGURACIÓN E INTEGRACIÓN DEL MÉTODO POTS DE LA API CON LA FUNCIÓN LAMBDA [2]	XXIII
FIGURA ANEXO C - 18: INTEGRACIÓN DE LA APLICACIÓN WEB Y LA FUNCIÓN LAMBDA (II) – CONFIGURACIÓN E INTEGRACIÓN DEL MÉTODO POTS DE LA API CON LA FUNCIÓN LAMBDA [3]	XXIII
FIGURA ANEXO C - 19: INTEGRACIÓN DE LA APLICACIÓN WEB Y LA FUNCIÓN LAMBDA (V) – PANEL DE CONFIGURACIÓN DE LA API	XXIII
FIGURA ANEXO C - 20: INTEGRACIÓN DE LA APLICACIÓN WEB Y LA FUNCIÓN LAMBDA (VI) – EJECUCIÓN DE UN CASO DE PRUEBA SOBRE LA API CREADA	XXIV
FIGURA ANEXO C - 21: CREACIÓN DE LA BASE DE DATOS NoSQL – PANEL PRINCIPAL DE GESTIÓN DE LA BBDD NoSQL	XXV
FIGURA ANEXO C - 22: CREACIÓN DE LA BASE DE DATOS NoSQL (II) – DETALLE DE LA CONSULTA DE VALORES/ELEMENTOS DE LA BBDD NoSQL.....	XXV
FIGURA ANEXO C - 23: CREACIÓN DE LA BASE DE DATOS NoSQL (III) – DETALLE DE LA CONSULTA DE VALORES/ELEMENTOS DE LA BBDD NoSQL [2]	XXVI
FIGURA ANEXO C - 24: CREACIÓN DE LA BASE DE DATOS NoSQL (IV) – PANEL DE GESTIÓN DE ROLES Y PERMISOS DE LA BBDD.....	XXVI
FIGURA ANEXO C - 25: CREACIÓN DE LA BASE DE DATOS NoSQL (V) – PANEL DE GESTIÓN DE ROLES Y PERMISOS DE LA BBDD [2]	XXVI
FIGURA ANEXO C - 26: CREACIÓN DE LA BASE DE DATOS NoSQL (VI) – REGISTRO DE POLÍTICA DE ACCESO A LA BBDD	XXVII
FIGURA ANEXO C - 27: CREACIÓN DE LA BASE DE DATOS NoSQL (VII) – DETALLE DE LA POLÍTICA DE ACCESO GENERADA SOBRE LA BBDD.....	XXVII
FIGURA ANEXO C - 28: DESPLIEGUE DE LA APLICACIÓN WEB EN LA NUBE - FRAGMENTO EXTENDIDO DE CÓDIGO ASOCIADO A LA CODIFICACIÓN DE LA FUNCIÓN LAMBDA v2..	XXVIII
FIGURA ANEXO C - 29: DESPLIEGUE DE LA APLICACIÓN WEB EN LA NUBE (II) - FRAGMENTO DE CÓDIGO ASOCIADO A LA LLAMADA DE LA API COMO PARTE DEL RECURSO WEB	XXVIII
FIGURA ANEXO C - 30: DESPLIEGUE DE LA APLICACIÓN WEB EN LA NUBE (III) - CARGA Y DESPLIEGUE DEL RECURSO WEB v2 EN AWS AMPLIFY	XXIX
FIGURA ANEXO C - 31: DESPLIEGUE DE LA APLICACIÓN WEB EN LA NUBE (IV) – EJECUCIÓN DE LA APLICACIÓN WEB DESPLEGADA	XXIX
FIGURA ANEXO C - 32: DESPLIEGUE DE LA APLICACIÓN WEB EN LA NUBE (V) - EJECUCIÓN DE LA APLICACIÓN WEB DESPLEGADA [2].....	XXIX

FIGURA ANEXO C - 33: DESPLIEGUE DE LA APLICACIÓN WEB EN LA NUBE (VI) – EJECUCIÓN DE LA APLICACIÓN WEB DESPLEGADA [3].....	XXX
---	-----

INDICE DE TABLAS

TABLA 7-1: RESUMEN COMPARATIVO DE CARACTERÍSTICAS PRINCIPALES DE CORRIENTES EMERGENTES: CLOUD COMPUTING.....	39
TABLA 7-2: RESUMEN COMPARATIVO DE CARACTERÍSTICAS PRINCIPALES DE CORRIENTES EMERGENTES: FOG COMPUTING	40
TABLA 7-3: RESUMEN COMPARATIVO DE CARACTERÍSTICAS PRINCIPALES DE CORRIENTES EMERGENTES: EDGE COMPUTING	40

Agradecimientos

A mi familia por haber estado siempre ahí, confiando en mí en todo momento y dándome todo el cariño y fuerza que necesitaba para sacar este trabajo adelante.

Mención especial a mi hermana, Libertad. Quién siendo la pequeña de la casa, es todo un ejemplo a seguir y una figura en la cual reflejarme.

A Carla. Contigo empezó esto, y no podía acabar sin agradecerte todo lo que has hecho por mí durante todo este largo tiempo.

A mis amigos, por ponerse de mi parte en todo lo que hago sin importar las circunstancias, gracias.

Y por supuesto, a mi tutora Ana, quién jamás perdió la esperanza ni en mí, ni en mi trabajo. Sin lugar a dudas, este TFG no hubiera sido posible sin ti. Jamás podré dejar de agradecértelo.

Resumen

El origen de este Trabajo Fin de Grado (en adelante TFG) viene motivado por la inquietud por conocer qué es lo que rodea el paradigma emergente del Cloud Computing desde un punto de vista conceptual. Cuál es, y cuál ha sido su evolución desde su origen, y sobre qué tipo de tecnologías subyacentes recae esta novedosa metodología.

En este sentido, y como parte del presente TFG, se realizará un profundo análisis bibliográfico en torno a las principales propiedades y modelos existentes, así como las ventajas e inconvenientes que derivan del uso de esta metodología en cuestiones que tienen que ver con la eficiencia, flexibilidad y/o disponibilidad de los datos y servicios manejados por los proveedores en la nube.

De este modo, a continuación, y tras analizar desde un punto de vista técnico las tecnologías más relevantes que permiten el despliegue e implementación del modelo de servicios que representa el paradigma del Cloud Computing, se ahondará en el marco legal que engloba la privacidad de la información manejada por los proveedores de servicios, como principal falla de seguridad identificada en consecuencia del intercambio de información constante en la red.

Posteriormente, el conjunto de definiciones y conceptos presentados a lo largo de la presente memoria, serán llevados a la práctica a través de la implementación y despliegue de un caso de uso real en la nube, que permitirá afianzar la teoría aplicada a la operativa de un proveedor de servicios en la red, como es Amazon Web Services (AWS). Cuyos procesos embebidos, posibilitarán el desarrollo de una arquitectura de tipo serverless, como forma de entender la estructura funcional y operativa que rodea la metodología bajo estudio.

Para finalizar, y analizando desde un punto de vista de mercado el auge que ha alcanzado el Cloud Computing en el marco tecnológico actual, se indagará sobre la evolución que ha tomado el paradigma en los últimos años, así como las principales vertientes que han derivado de él, dando lugar a nuevas interpretaciones del modelo tales como son Edge Computing y el Fog Computing entre otros.

Palabras clave

Paradigma, Cloud Computing, Nube, AWS, Fog Computing, Edge Computing, Serverless

Abstract

The origin of this Final Degree Project (hereinafter TFG) is motivated by the concern to know what surrounds the emerging paradigm of Cloud Computing from a conceptual point of view. What it is, and what has been its evolution since its origin, and what kind of underlying technologies this new methodology relies on.

In this sense, and as part of this TFG, an in-depth bibliographic analysis will be carried out on the main existing properties and models, as well as the advantages and disadvantages derived from the use of this methodology in matters related to the efficiency, flexibility and/or availability of the data and services handled by cloud providers.

Thus, after analysing from a technical point of view the most relevant technologies that enable the deployment and implementation of the service model that represents the Cloud Computing paradigm, we will then delve into the legal framework that encompasses the privacy of the information handled by service providers, as the main security flaw identified as a result of the constant exchange of information on the network.

Next, the set of definitions and concepts presented throughout this report will be put into practice through the implementation and deployment of a real use case in the cloud, which will allow the theory applied to the operation of a network service provider, such as Amazon Web Services (AWS), to be reinforced. Whose embedded processes will enable the development of a serverless architecture, as a way of understanding the functional and operational structure that surrounds the methodology under study.

Finally, and analysing from a market point of view the boom that Cloud Computing has reached in the current technological framework, the evolution that the paradigm has taken in recent years will be investigated, as well as the main aspects that have derived from it, giving rise to new interpretations of the model such as Edge Computing and Fog Computing, among others.

Keywords

Paradigm, Cloud Computing, Cloud, AWS, Fog Computing, Edge Computing, Serverless.

1 Introducción

En la última década, los continuos procesos de deslocalización e internacionalización de las grandes empresas, unidos al auge en el uso de las nuevas tecnologías de la información para el tratamiento y distribución de datos, han hecho que las necesidades de cómputo de las empresas hayan crecido exponencialmente a un ritmo muy por encima de las capacidades de cálculo albergadas por los ordenadores personales. Ante dicha necesidad, se ha producido un cambio evolutivo en las arquitecturas de cálculo, centrándose estas, en la ejecución simultánea de procesos en múltiples equipos informáticos de una forma integral e independiente al mismo tiempo, lo que hoy en día es conocido como Cloud Computing.

Antes de continuar, es importante destacar el uso generalizado del término “Cloud Computing” a lo largo del presente trabajo, y no el de “Computación en la Nube” como registro traducido al idioma castellano por la Unión Internacional de Telecomunicaciones (UIT). Dicha elección viene dada por la gran aceptación que ha tenido el primer término en el uso cotidiano englobado dentro del ámbito tecnológico bajo el cual se encuadra la presente memoria.

Así pues, el concepto o metodología del Cloud Computing representa el resultado de evolución tecnológica derivada de la tendencia por lograr maximizar la capacidad de ejecución de diferentes procesos en múltiples computadores de una forma concurrente.

Durante este proceso evolutivo y conforme queda plasmado dentro de la Figura 1-1, han sido múltiples las soluciones planteadas por el camino, desde el uso de SO tipo UNIX que operaban mediante pools de ordenadores que se comportaban como un único sistema gracias a la parametrización de clústers; hasta el uso de tecnologías basadas en la virtualización, donde la implementación de máquinas virtuales hacían posible la independencia remota entre Hardware y Software, lo que permitía a su vez la distribución eficiente de cargas de trabajo gracias al proceso de homogeneidad logrado dentro de un conjunto de ordenadores heterogéneos.

A lo largo de este evolutivo, y anteriormente al uso de la tecnología de virtualización como solución más avanzada hasta el momento, surgiría una corriente intermedia basada en Grid como arquitectura de cálculo orientada al procesamiento en paralelo de grandes cantidades de información. No obstante, la complejidad de estas, aunado a los problemas de integridad que presentaba, le hicieron rápidamente pasar a un segundo plano en cuanto a su uso como método de cálculo y procesamiento lógico computacional.

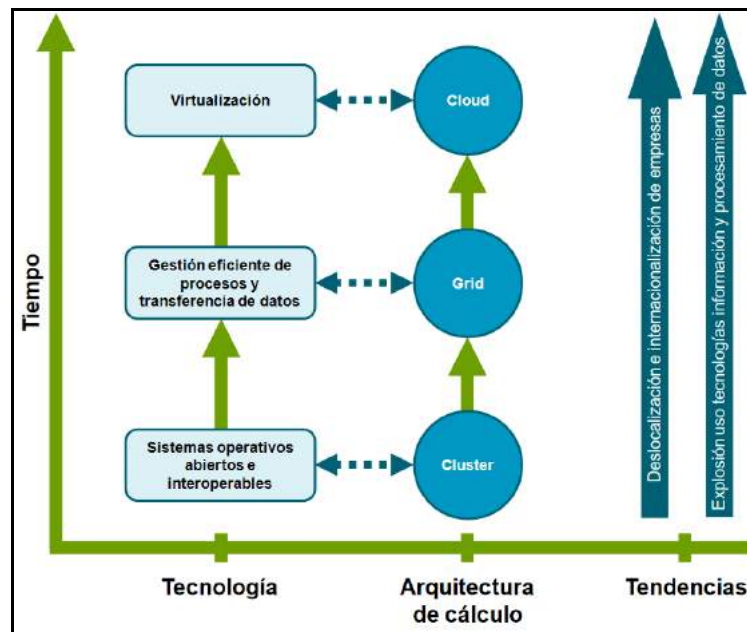


Figura 1-1: Evolución de las Arquitecturas de Cálculo Computacional ¹

De este modo, el Cloud Computing se presenta como un nuevo modelo emergente de prestación de servicios, y no una tecnología en sí misma, donde el gran avance logrado en cuanto al aprovisionamiento y gestión de recursos, le convierte en un nuevo paradigma capaz de solventar las crecientes necesidades requeridas por los proveedores de servicios en Internet, donde las grandes capacidades de cómputo han dejado prácticamente obsoletas las tecnologías conocidas hasta el momento en cuanto a la gestión de recursos concurrentes y prestación de servicios bajo demanda.

1.1 Motivación

Debido a la innovación objetiva que supone la inclusión de un nuevo paradigma en el entorno tecnológico actual, unido a la incertidumbre y desconocimiento generalizado creado en torno a la revolución derivada con respecto al concepto Cloud Computing, el presente trabajo surge de la necesidad por generar una base de conocimiento que permita comprender la metodología emergente presentada, desde un punto de vista tanto técnico, como operativo de estudio.

Así, y con el objetivo en mente de generar una bibliografía completa que pudiera servir como base a la hora de impartir una posible docencia a futuro en torno a esta nueva metodología, previamente se ha llevado a cabo un estudio pormenorizado alrededor de diferentes guías docentes impartidas actualmente en diversas organizaciones universitarias, tanto públicas como privadas, con el principal objetivo de recopilar los aspectos inherentes que debería recoger la presente memoria para cumplir así, con el propósito principal dispuesto bajo un prisma didáctico de definición.

¹ Extraído de: Santamaría-Buitrago, F. A., Ballesteros-Ricaurte, J. A., & González-Sanabria, J. S. (2015). Plataforma cloud computing como infraestructura tecnológica para laboratorios virtuales, remotos y adaptativos -Cloud computing as technologic infrastructure for virtual, remote and adaptive labs. Revista Científica, n° 23, BOGOTÁ, D.C. https://www.researchgate.net/publication/289389459_plataforma_cloud_computing_como_infraestructura_tecnologica_para_laboratorios_virtuales_remotos_y_adaptativos

1.2 Objetivos

El objetivo principal del presente trabajo es el de analizar y comprender el marco tecnológico que envuelve al paradigma del Cloud Computing como metodología emergente en el contexto tecnológico actual, tanto bajo un punto de vista teórico, como práctico de definición. Bajo esta premisa, se deberá entrar en detalle en torno al modelo de servicio que supone el hecho de operar bajo esta nueva metodología, lo que llevará inevitablemente, a concretar sus características y propiedades principales, así como sus ventajas e inconvenientes más notorios.

Así, en primer lugar, algunas de las preguntas más relevantes son, ¿En qué consiste este nuevo paradigma?, ¿En base a qué necesidades se plantea? ¿Qué propiedades presenta con respecto al estado del arte actual?, ¿Qué beneficios e inconvenientes aporta en cuanto a las arquitecturas y modelos más tradicionales? y ¿Bajo qué división se pueden categorizar los servicios englobados en la nube?; a estas y otras preguntas, se les dará respuesta como inicio del planteamiento del presente análisis bibliográfico realizado dentro del Capítulo 2: “*Paradigma del Cloud Computing*”.

A continuación, tras analizar técnicamente en el Capítulo 3: “*Principales Tecnologías sobre las cuales se asienta la metodología Cloud Computing*” las principales tecnologías que permiten el desempeño operativo del modelo de servicios en la nube propuesto por el Cloud Computing, el siguiente objetivo focalizado del presente trabajo se centrará en exponer un caso de uso real de implementación y despliegue sobre el paradigma manejado como parte del Capítulo 6: “*Caso Práctico: Desarrollo y Despliegue de un Servicio Real en la Nube*” descrito. En este punto, el objetivo principal recaerá sobre la comprensión de los diferentes elementos y componentes involucrados en la gestión de un servicio, suministrado por un proveedor de servicios en la nube como es Amazon Web Services (AWS). Así, el propósito real de este apartado diferirá de la dificultad que entraña la implementación y desarrollo de la propia aplicación a desplegar, ya que, conforme se ha expuesto, la intención formal de este capítulo será la de comprender la arquitectura, funcionalidades y operativa que posibilita el intercambio y despliegue de un servicio en la nube a través de un proveedor de servicios real.

Por último, y ligado a la intención de hacer comprender el marco evolutivo que envuelve a esta nueva metodología, se abordarán diferentes perspectivas en torno al análisis de ciertos aspectos más relevantes asociados al uso del Cloud Computing. Entre ellos, los aspectos legales que lo rodean en el Capítulo 4: “*Seguridad y Aspectos Legales*”, la creciente demanda de su uso en el mercado actual, como parte del Capítulo 5: “*Mercado Actual del Cloud Computing*”, o las nuevas corrientes emergidas en base al paradigma que representa esta nueva metodología en el séptimo y último capítulo de la presente memoria “*Principales Corrientes Emergentes Derivadas del Cloud Computing: Edge Computing y Fog Computing*”.

2 Paradigma del Cloud Computing

A nivel formal, el NIST (*National Institute of Standards and Technology*), define Cloud Computing como *“Un modelo tecnológico que permite el acceso ubicuo, adaptado y bajo demanda en red a un conjunto compartido de recursos de computación configurables compartidos (por ejemplo: redes, servidores, equipos de almacenamiento, aplicaciones y servicios), que pueden ser rápidamente provisionados y liberados con un esfuerzo de gestión reducido o interacción mínima con el proveedor del servicio.”*

Tomando como base la definición anterior, la tecnología sobre la cual se implementa el modelo tecnológico del Cloud Computing, recae sobre los principios de virtualización como método de abstracción entre el Hardware y el Software, haciendo posible la simulación de entornos e infraestructuras configurables, a través de los cuales, el usuario adquiere la posibilidad de distribuir cargas de trabajo de un modo sencillo dando lugar a un nuevo paradigma, el cloud computing.

En este sentido, el objetivo perseguido por el presente capítulo es el de afianzar el concepto de Cloud Computing como modelo tecnológico emergente, a través de la definición teórica de las principales propiedades y características de la metodología Cloud, así como de las distintas tipologías y roles que desempeñan los proveedores de servicios en cuanto a la categorización de los modelos operativos que giran en torno al paradigma del Cloud Computing.

Finalmente, se detallarán brevemente las principales ventajas e inconvenientes derivados del uso de la metodología del Cloud Computing como nuevo paradigma asociado al contexto tecnológico actual. Aspecto que se profundizará como parte del capítulo 4: *“Seguridad y Aspectos Legales”* de la presente memoria.

2.1 Propiedades Básicas de la Metodología Cloud Computing

En primer lugar, para comprender las principales propiedades en torno al uso y avances presentados por la metodología del Cloud Computing, a continuación, se detalla una por una las principales características que la diferencia de las metodologías y arquitecturas más tradicionales usadas hasta el momento. En este sentido, y conforme se muestra en la Figura 2-1, dichos avances pueden categorizarse en tres grandes grupos de acuerdo a las principales propiedades e innovaciones que presenta: Modelo de Servicio, Acceso y Propiedades de la Arquitectura.

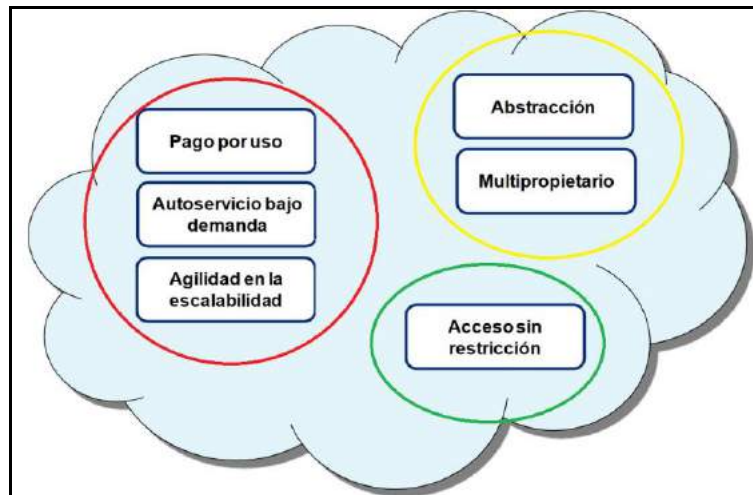


Figura 2-1: Principales Características de la Metodología Cloud Computing ²

MODELO DE SERVICIO

- **Pago por Uso:** El modelo de facturación presentado por la metodología Cloud Computing gira en torno al concepto de “pago por consumo”, es decir, el cliente únicamente debe asumir los costes derivados del uso de los servicios contratados. De este modo, el usuario paga únicamente por los servicios que requiere en cada momento concreto, ofreciéndole la posibilidad de ampliar los recursos contratados a medida que sus necesidades se incrementan.
- **Agilidad en la Escalabilidad:** De la mano del apartado anterior, la escalabilidad en el sistema resulta inminente conforme el crecimiento en las necesidades del mismo. El usuario únicamente deberá realizar una ampliación del servicio contratado, lo cual automáticamente, se verá reflejado en la mejora objetiva del sistema sin que el usuario deba preocuparse de la escalabilidad del mismo.

Dicha propiedad, conlleva una serie de ventajas objetivas en relación a posibles riesgos inherentes a la planificación de cualquier proyecto, como pueden ser errores a la hora de estructurar las necesidades de un sistema, los recursos, o el consumo entre otros.

- **Autoservicio Bajo Demanda:** El autoservicio bajo demanda como propiedad ligada a la flexibilidad presentada por el Cloud Computing, permite al usuario acceder a los servicios contratados sin la necesidad de interacción constante con el proveedor de servicios en la nube.

PROPIEDADES DE LA ARQUITECTURA

- **Abstracción:** Los recursos manejados por los diferentes servicios contratados al proveedor, permanecen aislados de los equipos informáticos durante el marco temporal de procesamiento que engloba el servicio. Este hecho, es llevado a cabo gracias al uso de la virtualización como tecnología subyacente, gracias a la cual, el

² Extraído de: Uruña, A. et al. (2012) *Cloud Computing. Retos y Oportunidades*. Ontis (Madrid) pp.14 https://www.ontsi.red.es/sites/ontsi/files/1-estudio_cloud_computing_retos_y_oportunidades_vdef.pdf

conjunto de tareas ligado al mantenimiento de la infraestructura, actualizaciones, control de errores, escalado, etc. Permanecen en el lado del proveedor, abstrayendo de este modo la lógica del sistema al entorno de procesamiento en la nube, y evitando que sea el usuario quien deba preocuparse por este tipo de inconvenientes y tareas asociadas.

- **Multipropietario:** El avance logrado en cuanto al modelo de servicios gestionado en la nube, permite a los usuarios acceder al total de los recursos alojados en una misma Plataforma de manera concurrente, indistintamente de su ubicación, permitiendo así la optimización del proceso de concurrencia a la hora de acceder y consultar a la información compartida alojada en la nube.

ACCESO

- **Acceso sin Restricción:** El único requisito de acceso a los servicios contratados en la nube, es el de disponer de un dispositivo con conexión a Internet. Esta propiedad aumenta considerablemente el rango de dispositivos capaces de operar con los servicios ofrecidos por la nube, maximizando así los criterios de accesibilidad al servicio y operatividad de uso bajo la presente metodología.

En base a los distintos avances y propiedades recogidas en el presente apartado, queda evidenciado como el Cloud Computing representa una metodología de uso revolucionaria en cuanto a las innovaciones presentadas con respecto a las arquitecturas más tradicionales, donde las empresas y usuarios deben realizar cuantiosas inversiones en recursos, personal, infraestructuras, actualizaciones, etc.

Además, bajo los modelos más tradicionales de uso, la computación bajo demanda, así como la disponibilidad y flexibilización de acceso a los recursos y servicios ofrecidos, se vuelven más complejos a medida que crecen las necesidades por parte del usuario, lo que inevitablemente aumenta el grado de complejidad en cuanto a costes asociados, accesibilidad al sistema y capacidad de procesamiento entre otros contrapuntos, llegando a desencadenar riesgos inherentes a la hora de procesar grandes volúmenes de información.

2.2 Modelos de Servicio Existentes

Conforme ha quedado detallado en el apartado anterior, y dadas las cuantiosas ventajas que representa el uso de esta metodología en cuanto a valores de eficiencia, accesibilidad, costes y optimización de procesos de cómputo, el modelo representado por el Cloud Computing se ha vuelto imprescindible tanto para empresas de base tecnológica, como para usuarios finales englobados dentro del ámbito de las TIC, donde la innovación presentada por la conceptualización del modelo de servicios en la nube, ha abierto el abanico de posibilidades ofrecidas en relación al alcance de Pymes y particulares, cuyas limitaciones albergadas por las arquitecturas tradicionales de computación, hacían prácticamente imposible el acceso a este tipo de avances.

Fruto de esta popularización, desde su nacimiento, la conceptualización del término bajo el cual se engloba el Cloud Computing ha ido evolucionando con el tiempo, ofreciendo un inmenso abanico de nuevas posibilidades en cuanto a las diferentes opciones, fórmulas y maneras de entender la metodología del Cloud Computing como proceso de provisión de

servicios en la nube. En este sentido, y conforme al avance de la tecnología, se han ido generando nuevos modelos de implementación, comúnmente conocidos como tipos de Cloud Computing.

Inicialmente, estos modelos se dividían en dos grandes categorías: nubes privadas y nubes públicas. Sin embargo, estos modelos han dado lugar nuevos tipos de entender la metodología enfundada tras el término de Cloud Computing, donde la combinación de los modelos originales, han dado pie a nuevos conceptos bajo los cuales entender el modelo de servicios ofrecido.

En base a ello, y tomando como referencia los informes publicados por Deloitte “*Cloud Computing: Forecasting change. Market Overview and Perspective*” y el NIST (*National Institute of Standards and Technology*) “*Cloud Computing Standards Roadmap*”, se definen tres propiedades fundamentales que delimitan la categorización de las soluciones y modelos basados en Cloud Computing: familias, formas de implementación y agentes intervinientes.

Este desglose categórico, junto con sus diferentes tipos de soluciones asociadas, se pueden representar de manera tridimensional como una visión global que representa la interacción entre los tipos de nubes y modelos de servicio existentes en la actualidad. Dicho sistema queda representado por la siguiente Figura 2-2 :

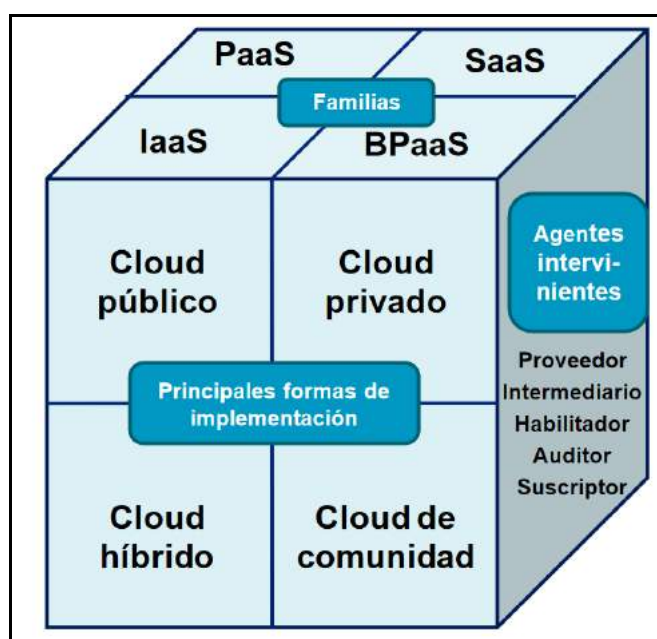


Figura 2-2: Modelo de Clasificación de Soluciones Cloud Computing³

³ Extraído de: Urueña, A. et al. (2012) *Cloud Computing. Retos y Oportunidades – Resumen Ejecutivo*. Ontis (Madrid) pp.14 https://www.ontsi.red.es/sites/ontsi/files/1-estudio_cloud_computing_retos_y_oportunidades_vdef.pdf

2.2.1 División por Modelos de Servicio

INFRASTRUCTURE AS A SERVICE (IaaS)

En primer lugar, el servicio ofrecido como IaaS o “*Infraestructura como Servicio*”, ofrece al usuario la posibilidad de gestión integral de una infraestructura informática completa, lo que incluye la gestión de recursos subyacentes tales como la capacidad de computación, el manejo de bases de datos, el espacio en disco, la configuración de redes o la gestión de firewalls entre otros muchos. Bajo este modelo, el usuario posee un mayor control de la información, ya que, tras una capa de virtualización, es el propio usuario el último responsable de la configuración de la infraestructura virtual generada. No obstante, se debe hacer hincapié en que verdaderamente, la infraestructura real que se ejecuta por detrás de la capa virtualizada, sigue perteneciendo al proveedor de servicios como máximo responsable y gestor de la misma.

El uso del Cloud Computing bajo el modelo IaaS descrito, va directamente ligado al uso del mismo por los administradores de sistemas. Entre los proveedores de servicios IaaS más populares se encuentran: AWS (Amazon), Microsoft Azure (Microsoft) y Google Cloud Platform (Google).

PLATFORM AS A SERVICE (PaaS)

Este servicio implica un nivel de abstracción superior al detallado como parte del modelo IaaS. En él, el proveedor facilita en forma de servicio al usuario, el conjunto de herramientas informáticas (Platform), orientadas al desarrollo, despliegue y testeo de cualquier aplicación informática, como pueden ser el SO, las librerías o los lenguajes de programación entre otros.

Basándose en la política de modelado de servicios descrita, así como en las principales ventajas aparentes que ofrece este modelo de servicios en términos de escalabilidad y mantenibilidad, PaaS es un servicio orientado principalmente a los desarrolladores software. No obstante, bajo este modelo, es el propio proveedor de servicios quien se encarga de administrar la infraestructura subyacente, por lo que el usuario únicamente debe preocuparse por el código de la aplicación a desarrollar. Ejemplos de este tipo de modelo de servicios y proveedores de estos son: AWS Elastic Beanstalk (Amazon), Azure App Service (Azure), Google App Engine (Google) o Red Hat OpenShift (Red Hat) entre otros.

SOFTWARE AS A SERVICE (SaaS)

Como su propia traducción literal indica, este modelo de servicio gira en torno al concepto del encapsulamiento de un software o aplicación como un servicio integral ofrecido por un proveedor externo. Dicho servicio se basa en el despliegue completo de una aplicación en la infraestructura del mismo proveedor. De este modo, el usuario prescinde de las labores de gestión y/o control sobre la infraestructura, y logra el acceso a la aplicación para su uso como un servicio bajo demanda.

Algunos ejemplos populares de proveedores que ofrecen SaaS como modelo de servicio en la nube son: Microsoft Office 365, las aplicaciones web de Google o el servicio de mensajería Slack entre otros.

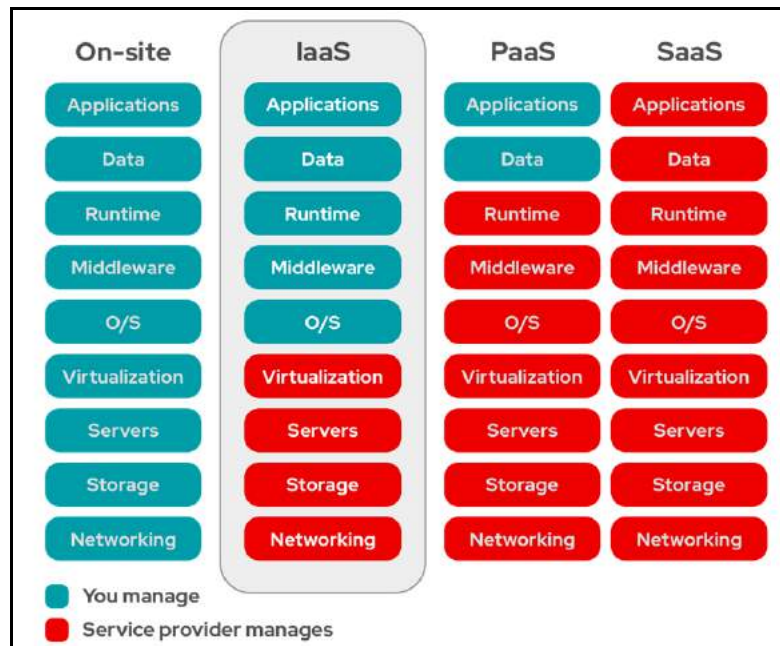


Figura 2-3: Principales Diferencias entre los Modelos de Servicio IaaS, PaaS y SaaS ⁴

En la Figura 2-3 anterior, y a modo resumen de los modelos de servicios anteriormente descritos, se representa gráficamente el alcance relativo a las tareas y actividades incluidas en la parte del proveedor de servicios, y aquellas en las que es el propio usuario quien tiene acceso a la gestión y administración de los componentes indicados.

BUSINESS PROCESS AS A SERVICE (BPaaS)

Finalmente, y como modelo de servicio más reciente incorporado al mercado, se encuentra el BPaaS, donde los recursos utilizados mediante dicho modelo para la ejecución de procesos de negocio, son compartidos entre diferentes clientes del proveedor. Dicho modelo se encuentra directamente ligado al concepto *end-to-end* como esquema de provisión estandarizada de compartición de procesos de negocio bajo demanda entre empresas principalmente.

No obstante, tras la gran aceptación recibida por este tipo de modelos de provisión de servicios, y de la mano de la constante innovación que envuelve la metodología Cloud Computing, durante los últimos años han ido surgiendo nuevos modelos de negocio de la mano de los grandes proveedores que persiguen explotar el valor inherente al paradigma del procesamiento en la nube. De este modo, a continuación, se aporta el detalle de algunos de los modelos más recientes que han tenido una mayor acogida por los usuarios de este tipo de servicios:

- **Seguridad como Servicio (SECaaS):** bajo este modelo, la seguridad de una infraestructura es ofertada al cliente como un servicio establecido en la nube a cambio de una suscripción o pago. Este tipo de modelos de servicio incluyen diferentes componentes en materia de seguridad como pueden ser: métodos de autenticación, detección de intrusiones, antivirus o antimalware entre otros.

⁴ Extraído de: RedHat. ¿What is IaaS? <https://www.redhat.com/es/topics/cloud-computing/what-is-iaas>

- **Funciones como Servicio (FaaS):** este modelo de servicios se fundamenta bajo lo que se conoce como arquitectura serverless o “sin servidor”. De este modo, dentro del alcance expuesto por FaaS, los usuarios implementan diferentes funciones Lambda que desencadenan eventos en la parte del servidor, bajo una estructura basada en contenedores efímeros que se ejecutan únicamente tras la llamada de la parte del usuario, multiplicando así la escalabilidad del sistema, a la vez que se reduce el coste de las tarifas de pago por uso como una de las principales ventajas presentadas por el Cloud Computing.

Bajo el contexto derivado del modelo FaaS presentado, y como parte del capítulo “6. Caso Práctico: Desarrollo y Despliegue de un Servicio Real en la Nube” incluido en la presente memoria, se implementa un caso real basado en el uso y codificación de este tipo de servicio, donde se detalla la metodología y herramientas que engloba el uso de la tecnología serverless y FaaS, de la mano de un proveedor real de servicios como es AWS de Amazon.

Asimismo, además de los modelos descritos anteriormente, y tal como se ha comentado en el párrafo inicial del presente apartado, la creciente evolución del mercado en este punto hace que este tipo de modelos desempeñe un desarrollo incipiente en cuanto a nuevas formas de explotar la metodología Cloud como, por ejemplo: Plataforma de integración como servicio (iPaaS), Mobile "backend" como servicio (MBaaS) o Identidad como servicio (IDaaS) entre otros muchos.

2.2.2 División por Modelos de Implementación

En segundo lugar, y tomando en consideración la manera de integrar y explotar los distintos servicios ofrecidos por la nube, los diferentes modelos se pueden clasificar en:

NUBE PÚBLICA

Modelo de implementación basado en la oferta de servicios de computación virtualizados por parte de los proveedores para múltiples usuarios. El presente modelo de servicio es entendido como un servicio bajo demanda e inmediato, donde los usuarios acceden a él a través de Internet o de redes privadas virtuales (VPNs). El total de los recursos ligados a la implementación de la infraestructura física son propiedad del proveedor, quien se encarga de su gestión, administración y mantenimiento integral, abstrayendo completamente al usuario de la capacidad de ejecución de dichas tareas.

NUBE PRIVADA

En el extremo contrario de lo expuesto como parte de la definición de nube pública, el modelo de nube privada se caracteriza por el suministro por parte del proveedor, del conjunto de recursos e infraestructuras requeridas por el cliente de una manera extensible a los elementos subyacentes que lo componen. Bajo esta visión de modelo de servicios, los recursos ofrecidos pasan a ser propiedad de una sola institución, manteniendo la infraestructura en una red privada de uso, y llegando incluso a ofrecer la posibilidad de alojar los servicios en las propias instalaciones del cliente, siendo este el máximo administrador de la infraestructura contratada y contando con la potestad mandataria en cuanto a decisiones que atañen a valores de seguridad, privacidad o configuración.

NUBE HÍBRIDA

La nube híbrida nace como una composición de ambos modelos anteriormente descritos. Bajo esta perspectiva, ambas nubes, tanto pública como privada, componen dos entidades independientes, pero mantienen una interrelación basada en principios de comunicación y portabilidad de aplicaciones y datos.

Este concepto surge de la necesidad de algunas empresas de lograr aumentar su capacidad de producción alojada en su propia nube privada a través de los servicios ofrecidos por los proveedores de nubes públicas, obteniendo de este modo, un modelo híbrido de implementación de servicios que da lugar a la presente categorización. En este sentido, el modelo expuesto persigue el objetivo de lograr beneficiarse de las ventajas individualizadas de cada modelo subyacente, no obstante, presentan ciertos inconvenientes en materia de sincronización de los datos manejados entre ambas nubes.

NUBE DE LA COMUNIDAD

Finalmente, y bajo el modelo englobado por concepto de nube de la comunidad, diferentes organizaciones con objetivos y servicios comunes, ofrecen sus servicios bajo un modelo colaborativo de uso, los cuales, son administrados por las organizaciones constituyentes o terceras partes.

Las fortalezas y debilidades de este modelo de implementación, se sitúan entre las de la nube pública y la privada, maximizando valores relativos a las nubes privadas, como pueden ser la elasticidad, disponibilidad o seguridad de la información, pero reduciendo el número de recursos y servicios ofrecidos por una nube pública.

Además de la clasificación anterior, y conforme se describía en el primer apartado relativo a los modelos por servicio, los modelos basados en la implementación, han desarrollado nuevas variantes compuestas por los principios anteriormente descritos, dando lugar a nuevas formas de interpretar la categorización de servicios ofrecidos en este punto. Un ejemplo de ello, son los modelos expuestos a continuación:

- **Intercloud:** Basado en un modelo implementado por el concepto abstracto de “nube de nubes”, el modelo Intercloud implementa un complejo entramado de red en el cual, todos los usuarios están interconectados globalmente y cuyo objetivo es la de fomentar la interoperabilidad directa entre proveedores públicos de servicios en la nube.
- **Multicloud:** El modelo Multicloud parte de la premisa funcional presentada por el modelo heterogéneo de nube híbrida. Sin embargo, el alcance de este modelo se extiende al uso de una gran variedad de servicios, lo que permite implementar diferentes políticas de acceso, a la vez que maximiza su flexibilidad y reduce la dependencia entre proveedores.

2.2.3 División por Agentes Intervinientes

En lo relativo a la clasificación por agentes intervinientes, los modelos de servicio dependen del propósito final ligado al rol del usuario principal contratante del servicio. Bajo esta perspectiva, el modelo se puede dividir en: Habilitador, Proveedor, Intermediario, Auditor o Suscriptor.

Finalmente, y de manera independiente a la categoría perteneciente o al tipo de modelo de servicio establecido en torno a la conceptualización del Cloud Computing como metodología, se ha de destacar la propiedad común de todos ellos de extraer, agrupar y compartir recursos informáticos escalables en una red, ya sea pública o privada. Ejecutando diferentes cargas de trabajo que incluyen la gestión parcial de algún tipo de infraestructura común, así como el manejo de aplicaciones e interfaces de programación de aplicaciones API para la interacción con los servicios ofrecidos por un proveedor.

2.3 Principales Ventajas e Inconvenientes

Analizados de manera detallada las diferentes propiedades, modelos y tipologías bajo los cuales se engloban el conjunto de servicios alojados en la nube como parte del paradigma Cloud Computing, a continuación, en el presente apartado se estudiarán las principales ventajas e inconvenientes del uso de esta metodología desde un punto de vista global objetivo.

En primer lugar, y en lo que respecta a las ventajas que ofrece el uso del modelo de servicios basado en la nube, la mayoría de ellas quedan evidenciadas por las propias características inherentes al paradigma presentado. No obstante, y entre los principales beneficios que conlleva el uso de dicha metodología, se han de destacar las facilidades que otorga su operativa, en cuanto a la capacidad por parte del usuario o cliente de escalar rápidamente en función de sus necesidades concretas, a lo que se le suman propiedades como la flexibilidad que brinda el acceso dinámico bajo demanda presentado por los servicios presentados en la nube, la disponibilidad constante de los servicios contratados o la capacidad de procesar y almacenar grandes volúmenes e información que, en cualquier otro caso, exigiría disponer de cuantiosas infraestructuras tecnológicas con los correspondientes gastos que ello conlleva en materia de adquisición, mantenimiento o actualización entre otros.

Además, el modelo de uso detrás del paradigma presentado por el Cloud Computing, permite una equidad entre las grandes y pequeñas empresas, indistintamente de su índole o sector que representen, gracias a la liberación de las tareas que suponen el mantenimiento, actualización y amortización de grandes inversiones tecnológicas en sistemas y recursos tecnológicos, y facilitando el acceso a los servicios en la nube a través de un modelo basado en el pago por uso, como única limitación de acceso al mismo. Ampliando así, el alcance global de los servicios prestados.

Bajo las innumerables ventajas que presenta a priori el uso e implementación bajo el modelo de servicios en la nube propuesto por la metodología Cloud Computing, se expone a continuación un breve resumen de con algunos de los principales beneficios que esta aporta a nivel global:

- Permite la optimización en las labores de implementación y desarrollo gracias a las propiedades que presenta en cuanto a alta disponibilidad, escalabilidad, flexibilidad, eficiencia y ahorro de costes en infraestructura.
- Modelo dinámico de pago por uso basado en el concepto de servicio bajo demanda, donde los costes son proporcionales a las necesidades individuales de cada usuario.
- Amplía el alcance existente en relación a la innovación empresarial, gracias a la minimización de riesgos que conlleva el uso de dicha metodología en materia de costes y recursos. Además, el carácter descentralizado que presenta, permite extender los beneficios que conlleva a una escala global, accesible para todo el mundo.

En contrapunto a los beneficios que otorga al usuario la adopción del paradigma que plantea el Cloud Computing, este se enfrenta a una serie de riesgos que paradójicamente derivan de una de sus principales virtudes, la abstracción y el procesamiento de datos por agentes externos o proveedores de servicio.

Este proceso conlleva el riesgo inherente de tener que confiar a la nube los datos sensibles que manejan los servicios contratados, lo cual, supone una serie de desventajas en lo que respecta a la integridad de información manejada, la gestión de datos sensibles y la gestión de la propiedad intelectual entre otros aspectos que tienen que ver con la privacidad del usuario, el valor comercial de las compañías, o la repercusión que esto pueda suponer a efectos legales y/o jurídicos.

Así, las principales desventajas o inconvenientes del uso del paradigma planteado por el Cloud Computing en la actualidad, recaen sobre tres puntos concretos de acción principalmente, todos ellos, relacionado con el manejo y tratamiento de la información requerida en cuanto a la gestión de los servicios en la nube:

- Riesgo que conlleva la externalización de servicios a proveedores externos.
- Tratamiento de la información fuera de los límites manejados por el usuario.
- Dependencia inherente de un proveedor de servicios que provea al usuario de la infraestructura de cómputo.

A pesar de estas desventajas, la protección de datos es probablemente el inconveniente que genere una reticencia mayor entre los usuarios de la nube. Por ello, y como parte del capítulo “4. Seguridad y Aspectos Legales” de la presente memoria, se adjunta un breve análisis sobre las premisas y aspectos legales que rodean este nuevo concepto en materia de privacidad y protección de la integridad de los datos se refiere.

3 Principales Tecnologías sobre las cuales se asienta la metodología Cloud Computing

Tras haber profundizado teóricamente sobre el paradigma que plantea el Cloud Computing como metodología emergente relativa a la prestación de servicios en la nube, así como de los distintos modelos de negocio que lo rodean y las ventajas y desventajas que su uso conlleva. En el presente capítulo se hará un breve recorrido en torno a las principales tecnologías en las cuales se apoya la metodología manejada, como manera de entender desde un punto de vista técnico el comportamiento de este nuevo paradigma en cuanto al uso y manejo de las herramientas existentes en el contexto global tecnológico bajo el cual se desarrolla el Cloud Computing.

3.1 Equilibrado de Carga

Uno de los principales problemas derivados de la alta disponibilidad ofrecida por los servicios alojados en la nube como parte de la metodología Cloud Computing, es el reto que supone el hecho de atender y gestionar la gran cantidad de solicitudes originadas por el alto número de usuarios activos en cada momento del tiempo. Se trata de un problema directamente ligado a la escalabilidad del sistema, ocasionado por el continuo crecimiento exponencial del número de usuarios que utilizan una tecnología.

En este punto, una de las posibles soluciones manejadas, ya presentes en el estado del arte actual tecnológico de los sistemas basados en peticiones web, hace referencia a los balanceadores o equilibradores de carga. A este respecto, y como parte del “*ANEXO A: Balanceadores de Carga*” adjunto a la presente memoria, se aporta una breve caracterización del proceso implementado por este tipo de tecnología, así como el detalle pormenorizado de algunos de los principales métodos de balanceo sobre los cuales pende la lógica implicada en los procesos de distribución subyacentes.

Así pues, y conforme queda evidenciado tanto en el Anexo mencionado como en el presente apartado, el balanceo de carga es un factor determinante para aquellas redes en las que es difícil predecir el número de solicitudes que se enviarán a un servidor, factor que, unido al exponencial número de conexiones concurrentes existentes en un determinado momento del tiempo, le hacen un método indispensable en las labores de redireccionamiento y manejo de solicitudes para sistemas y redes basados en Cloud Computing.

De este modo, y a través del uso de este tipo de herramientas, se logra maximizar, y al mismo tiempo, reducir, la carga existente sobre un mismo servidor, maximizando la capacidad operativa de un determinado servicio. Además, un nodo balanceador de carga ofrece una serie de ventajas añadidas que hacen de su uso una herramienta indispensable en la gestión del tráfico concurrente para servicios con un alto tráfico de conexiones, como su uso como *firewall* en el control de acceso a la red como principal medida de seguridad del sistema o servicio, o la capacidad de tolerancia a fallos en cualidad de maximizar la disponibilidad del servicio gracias a su infraestructura descentralizada.

3.2 Virtualización

Bajo el contexto que engloba el ámbito de la informática, la virtualización es entendida como el conjunto de procedimientos orientados a la abstracción de recursos dentro de una misma CPU. Dicha abstracción, ha ido evolucionando a lo largo de la historia del término, haciendo referencia desde la simulación de sistemas computacionales completos, hasta la abstracción de recursos concretos o componentes individuales. No obstante, todas ellas comparten un nexo en común que hace referencia al fin en sí mismo como uno de los principales objetivos de la virtualización, y es la de ocultar o abstraer los diferentes recursos o características técnicas a través de la encapsulación y simulación gracias a una combinación simbiótica de hardware y software.

Como parte del “ANEXO B: *Virtualización*” adjunto a la presente memoria, se incluye un breve análisis pormenorizado entorno a la tecnología que envuelve a los procesos de virtualización, así como un desglose técnico de los principales métodos de virtualización utilizados actualmente.

En este sentido, y ligado la tecnología bajo la cual se enmarcan los procesos de virtualización mencionados, cabe destacar el estrecho margen de diferenciación tecnológica existente entre los procedimientos de virtualización, y la definición y operativa que engloba el Cloud Computing. Bajo este contexto, y conforme queda expuesto a lo largo de la presente memoria, ambos conceptos hacen referencia a la separación de recursos abstractos, ya sean software o hardware, para generar un entorno útil paralelo, en el cual, se puedan ejecutar procedimientos remotos de manera concurrente. No obstante, se ha de recalcar la principal diferencia entre ambos términos, y es que, si bien la virtualización fundamenta una tecnología en sí misma, el Cloud Computing hace referencia a una metodología de uso mediante el aplicativo y ejecución de diferentes recursos o servicios específicos.

De manera sencilla, y bajo la perspectiva específica de infraestructura *IaaS* contenida dentro de los modelos de servicio ofrecidos dentro del Cloud Computing, la cual provee al usuario de la independencia de espacios simulados tanto públicos, como privados e híbridos; el proceso de virtualización haría referencia a la tecnología encargada de generar las diferentes “nubes” o entornos de procesamiento remoto. En este punto, el software de virtualización se ejecuta sobre el hardware de origen de la CPU y se encarga de extraer los diferentes recursos de la máquina. Estos recursos pueden ser el potencial de procesamiento en bruto capacitado por los servicios del cloud, como el almacenamiento o las aplicaciones basadas en la nube, las cuales contienen el código de ejecución y los recursos necesarios para implementarlos.

No obstante, el alcance delimitado por los procesos de virtualización finalizaría en este punto. A continuación, los recursos recogidos deberán asignarse a los diferentes grupos y servicios centralizados, antes denominado “nubes”, y esas nubes a su vez, deberán coordinarse con el software de gestión y automatización provisto por el servicio *IaaS* en este caso, fundamentando así, el claro ejemplo que se concibe al describir el Cloud Computing como una metodología, y no como tecnología en sí misma.

Entre otras ventajas, las nubes proporcionan el acceso independiente o capacidad de autoservicio, así como la creación de grupos de recursos dinámicos como principales beneficios que marcan la diferencia con respecto a los procedimientos de virtualización más tradicional.

En la siguiente Figura 3-1, se puede apreciar las principales diferencias existentes entre los conceptos de Virtualización y Nube, en las cuales y a pesar de su aparente similitud a nivel operativo, se observan grandes diferencias en cuanto a, por ejemplo, el propósito, el uso o la tenencia entre otros.

	Virtualización	Nube
Definición	Tecnología	Metodología
Propósito	Crear múltiples entornos simulados desde un solo sistema de hardware físico	Agrupar y automatizar recursos virtuales para su uso a pedido
Uso	Entregar recursos en paquetes a usuarios específicos para un fin determinado	Entregar variedad de recursos a grupos de usuarios para varios fines
Configuración	Basado en imágenes	Basado en plantillas
Duración	Años (largo plazo)	De horas a meses (corto plazo)
Costo	Gastos de capital (CAPEX) elevados, gastos operativos (OPEX) bajos	Nube privada: CAPEX elevados, OPEX bajos Nube pública: CAPEX bajos, OPEX elevados
Escalabilidad	Escalabilidad vertical	Escalabilidad horizontal
Carga de trabajo	Con estado	Sin estado
Tenencia	Usuario único	Varios usuarios

Figura 3-1: Principales Diferencias entre Virtualización y Cloud Computing ⁵

Así, el NIST (*National Institute of Standards and Technology*) menciona cinco características o beneficios destacables del cloud computing frente a los procesos de virtualización en un entorno operativo de gestión de recursos: *“Servicios ofrecidos a través de la red, independencia de recursos agrupados entre sí, adaptación específica de los servicios a una interfaz de usuario concreta, capacidad de aprovisionamiento y automatización del control y asignación de recursos optimizada.”* Además, hay que tener en cuenta que el proceso de virtualización crea la red y agrupa los recursos, pero requiere de otro software adicional de gestión, como es el SO, para generar una interfaz de usuario que permita construir las VM requeridas en el proceso de control y asignación de los recursos manejados.

Como conclusión derivada de las palabras de Mike Adams, director de marketing de producto en VMware, se establece que, *“La principal diferencia entre la virtualización y el cloud computing radica en que, a través de la virtualización, es el software quien se encarga de simular o manipular el hardware, mientras que la segunda tecnología es obtenida como consecuencia del servicio dispuesto de dicha simulación. Dicho de otro modo, la virtualización es la infraestructura sobre la cual recaen los procesos albergados dentro del cloud computing, mientras que esta última, consolida los servicios prestados a través de los distintos entornos virtualizados.”*

⁵ Extraído de: RedHat. *¿En qué se diferencian cloud computing y virtualización?* <https://www.redhat.com/es/topics/cloud-computing/cloud-vs-virtualization>

No obstante, y a pesar de las evidentes diferencias operativas de ejecución y concepto entre ambos términos, el conjunto formado por los procesos de virtualización y el cloud computing conforman en muchos casos un mismo sistema integral en multitud de proyectos de transformación digital y servicios tecnológicos actuales.

3.3 Bases de Datos No Relacionales

El auge en el enfoque de las nuevas tecnologías orientadas a la demanda de servicios en la nube como base de la metodología Cloud Computing, ha generado diferentes problemas en relación al elevado volumen y tipología de datos a manejar como principal precursor de lo que hoy en día se conoce como Big Data, lo cual ha desentrañado una serie de nuevos retos para los desarrolladores y administradores de bases de datos a la hora de optimizar los procesos de gestión y manejo de este tipo de recursos, a la vez que se logre mantener la velocidad operativa de los servicios ofrecidos.

Frente a este fenómeno, las bases de datos relacionales (NoSQL) son consideradas como una solución capaz de ofrecer las características esenciales dispuestas por los servicios en la nube, gracias a que estas han sido diseñadas en base a una serie de principios fundamentales orientados a las características que estos ofrecen, tales como, proveer rapidez, escalabilidad, alta disponibilidad y elasticidad, a la vez que se centran en aspectos esenciales como son la administración de datos como objetivo principal de cualquier tipo de base de datos. En este sentido, y tras revisar técnicamente el concepto de base de datos no relacional, a continuación, se profundizará en torno a las principales características que representa esta novedosa tecnología.

A pesar de que este tipo de estructuras fueron inventadas en la década de los 60 del pasado siglo, el auge en este tipo de tecnología proviene de su reciente adopción por los principales proveedores de Internet como Amazon, Google o Facebook entre otros. Su uso se hace inherente a la evolución de la tecnología y de los retos a nivel tecnológico que supone el tratamiento y procesamiento de grandes volúmenes de información de una inmensa cantidad de usuarios activos en un determinado momento, siendo un factor determinante, el tiempo de respuesta y procesamiento de las peticiones entrantes.

De este modo, y a grandes rasgos, las bases de datos no relacionales, también conocidas como bases de datos NoSQL, son sistemas de almacenamiento de información que rompen con el esquema entidad-relación tradicional de las bases de datos relacionales. En ellas, los datos no se almacenan en tablas, y la información tampoco se organiza en registros o campos como venía siendo habitual hasta ahora, sino que proveen un esquema mucho más flexible, con modelos de consistencia relajados, lo cual, incide positivamente en valores relacionados con la disponibilidad y el rendimiento a la hora de almacenar información.

Así pues, y tras analizar formalmente la base fundamental sobre la cual se asientan este tipo de bases de datos, a continuación, se exponen las principales características y propiedades definidas sobre dicha tecnología como forma de entender su rol operativo y las ventajas que su uso presenta con respecto a su operativa conjunta al paradigma del Cloud Computing como metodología:

- **Flexibilidad:** El modelo iterativo sobre el cual se estructuran las bases de datos no relacionales favorece un alto grado de flexibilidad y libertad para el desarrollador a lo hora de llevar a cabo las funciones de almacenamiento de manera eficiente. Esta característica, además, favorece su idoneidad para albergar tanto datos estructurados, como no estructurados, lo cual optimiza su rendimiento como tecnología principal de almacenamiento de datos. Además, y como parte de esta propiedad, se ha de destacar su capacidad de almacenar datos de los cuáles no se tiene una estructura clara y que serían casi imposibles de organizar en una base de datos tradicional del tipo relacional.
- **Escalabilidad:** La base iterativa presentada en el apartado anterior, hace que este tipo de estructuras sean fácilmente escalables horizontalmente a través del uso e implementación de diferentes clústers distribuidos, lo que repercute favorablemente sobre el coste que conllevaría el escalado de almacenamiento a través del uso de servidores físicos.
- **Disponibilidad:** A través del uso de arquitecturas de tipo distributivo, las bases de datos NoSQL se alejan de las estructuras típicas relacionales basadas en el uso de nodos primarios y secundarios, maximizando la disponibilidad en este punto, y garantizando el acceso a los procedimientos de lectura y escritura del dato incluso en caso de fallo o error en un nodo determinado.

El proceso de replicación automática del dato realizada en los distintos clústeres que conforman el sistema (nubes, servidores, centros de gestión de datos, etc.) permite al usuario minimizar la latencia de acceso, aumentando la eficiencia del sistema distribuido y garantizando la disponibilidad sea cual sea el punto de acceso de origen a la información.

- **Alto Rendimiento:** La cualidad de adaptación a tipos de datos específicos presentada por parte de las bases de datos de tipo NoSQL, así como el uso de patrones de acceso únicos en cada caso, permiten optimizar su rendimiento respecto a las bases de datos relacionales convencionales conocidas hasta el momento.
- **Altamente Funcional:** Gracias al uso compartido de API de desarrollo, las bases de datos de tipo NoSQL proporcionan diferentes funcionalidades específicas que se adaptan al modelo de datos manejado, incrementando el nivel funcional del conjunto y maximizando la operatividad y eficiencia en cada caso.

Asimismo, y al igual que ocurre con el resto de tecnologías, las bases de datos no relacionales también presentan una clasificación en cuanto al rol concreto que desempeñan y el fin operativo hacia el cual van orientadas. En este sentido, existen diferentes tipologías de acuerdo al enfoque concreto que presentan. A continuación, se adjunta un breve detalle de aquellas más características:

- **Bases de datos clave/valor:** El primer modelo de base de datos no relacional se apoya en la simplicidad de uso, y van orientadas a maximizar las velocidades de lectura/escritura para grandes volúmenes de datos nos transaccionales. Los valores almacenados no representan ninguna restricción, pueden ser de cualquier tipología (texto, vídeo, JSON, etc.) y se accede a ellos a través de una clave, motivo por el cual, se considera uno de los modelos más flexibles a la hora de almacenar datos.

- En cuanto a la escalabilidad y disponibilidad del sistema, las bases de datos NoSQL de clave-valor, comparten y replican sus datos en diferentes clústers, lo que le impide operar bajo modelos basados en transacciones.

Entre los sistemas más conocidos en este punto, se encuentran: Google BigTable, Amazon Dynamo o Cassandra entre otros.

- **Bases de datos de documentos:** El segundo modelo presentado radica de la operativa mostrada por las bases de datos anteriores de clave-valor, entre los sistemas más usados en este punto, se encuentran CouchDB, MongoDDB o SimpleDB. En este caso, la principal diferencia reside en que el valor almacenado, corresponde a un documento únicamente, indistintamente del tipo que represente (XML, JSON, HTML, etc.), el cual, se encarga de almacenar todos los datos relacionados con la clave que representa. Además, bajo este modelo, es posible indexar los campos más populares contenidos en los documentos como forma de proporcionar una recuperación rápida de los mismos sin necesidad de conocer la clave.
- **Bases de datos de grafos:** Como su propio nombre indica, este tipo de bases de datos se fundamenta bajo estructuras de tipo grafo que interactúan entre sí a la hora de almacenar, correlacionar y consultar relaciones. Su peculiaridad reside en la adyacencia libre de índice, es decir, los elementos adyacentes se unen entre sí sin la necesidad de usar un índice. Esta propiedad le hace una tipología idónea para los sistemas basados en recomendaciones.

Vistas las principales propiedades que presentan las Bases de Datos NoSQL, así como algunas de las estructuras y tipologías más usadas hasta la fecha, cabe destacar brevemente algunas de sus desventajas con respecto a las Bases de Datos relacionales más tradicionales:

- Difieren de las propiedades de atomicidad, consistencia, integridad y durabilidad presentadas por las bases de datos relacionales como características más relevantes.
- No son compatibles con determinadas consultas del tipo SQL. Además, este tipo de base de datos no dispone de un lenguaje de consulta declarativo, por lo que requiere de un esfuerzo mayor en tareas de implementación para la manipulación de los datos contenidos.
- Al conformar una tecnología actualmente considerada como emergente, carecen de un sistema de estandarización, por lo que cada una de ellas es implementada en base a los objetivos y funciones específicas requeridas.
- Actualmente poseen un equipo de soporte bastante más limitado que en el caso de las bases de datos relacionales.

En cuanto a la relación que este tipo de estructuras mantiene con la revolución acontecida bajo la metodología del Cloud Computing, quedan evidenciadas las propiedades comunes que comparten en cuanto a su fin por la gestión de grandes volúmenes de información con el hándicap objetivo que supone el mantener la velocidad y eficiencia en las operaciones manejadas, así como por la amplia diversidad en cuanto a la variedad y/o esquemas presentados por la estructura de la información a gestionar en cada momento del tiempo.

4 Seguridad y Aspectos Legales

Tras haber asentado las bases teóricas en torno a la tecnología que rodea el paradigma del Cloud Computing, y una vez analizadas las principales ventajas y desventajas presentadas por esta nueva metodología tecnológica, en el presente capítulo se hará un breve repaso acerca de los términos legales a nivel nacional que envuelven al uso del Cloud Computing en materia de privacidad y tratamientos de datos sensibles, así como los aspectos más relevantes que tienen que ver con la seguridad y la integridad de los mismos.

Debido al gran alcance teórico concebido en el presente aparatado en cuanto a la legislación, normativa y aspectos jurídicos que envuelven el uso de servicios ofrecidos en la nube, el presente capítulo tiene por objetivo centrarse en el análisis de aquellos puntos más relevantes en torno a las principales debilidades y desventajas presentadas por el uso y gestión del paradigma asociado al Cloud Computing en materia de legislación general y tratamiento de datos de carácter sensible.

Así, según los datos concluidos por la compañía Cloud Security Alliance, *“Las tres principales amenazas en materia de seguridad en la nube, tienen que ver con la API e Interfaces inseguras, la pérdida de datos y las fugas de Hardware. Amenazas que en el último año representaron el 29%, 25% y 10% respectivamente dentro del cómputo global de los cortes de seguridad detectados en la nube.”*

De este modo, antes de continuar, se ha de profundizar en torno a los principales pilares que rodean a la integridad y seguridad de cualquier sistema o modelo de gestión, adquiriendo una visión concreta de las vulnerabilidades que atañen al uso fraudulento de la metodología Cloud Computing, así como de los datos que se almacenan en la nube. De este modo, los tres pilares fundamentales en materia de seguridad serían:

- **Amenazas:** Bajo esta categoría se engloban el conjunto de problemas que atañen a una organización desde dentro, es decir, aquellas que inciden directamente a los datos de una organización. Bajo este alcance se sitúan: violaciones y secuestros de datos, amenazas externas, pérdida de información, negación del servicio, problemas en la autenticación, etc.
- **Vulnerabilidades:** Conformadas por los puntos débiles y factores de riesgo que rodean al sistema de una organización. En este punto, es importante tener un conocimiento ampliado en torno a los factores de riesgo y debilidades tecnológicas que presenta una organización para poder asegurar la integridad de los datos en la nube.
- **Responsabilidades:** Este apartado es quizá el más abstracto de entre los tres, también el más crítico en valores de integridad y compromiso de confidencialidad de los datos gestionados. Hace referencia a la definición minuciosa del alcance con respecto a las competencias que asume el proveedor de servicios en la nube, y el que va directamente ligado a la protección de seguridad como responsabilidad final del usuario a nivel individual con acceso a los servicios en la nube.

En este sentido, y a pesar de que las leyes y normativas varían según el lugar en el que se alojen los servidores contratados, cabe hacer hincapié en la inexistencia actual de una normativa específica que regule el uso del Cloud Computing en materia de seguridad y privacidad de la información tanto a nivel nacional como internacional de aplicación. No obstante, y como puede intuirse, es un hecho que el uso de esta metodología conlleva una serie de implicaciones jurídicas a la hora de proveer los servicios ofrecidos por las grandes y pequeñas compañías del sector.

Sin embargo, sí que existe un marco común de normas y leyes vigentes a nivel nacional que regulan el uso y tratamiento de datos e información de carácter personal que, si bien no corren jurisprudencia dentro del ámbito concreto del Cloud Computing, se aplican al resto de procedimientos comunes en materia de privacidad y seguridad de la información y que, por tanto, podrían aplicar al paradigma que supone la gestión de datos en la nube. Dichas premisas, quedan recogidas principalmente en los siguientes documentos concretos:

- Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal.
- Ley 34/2002, de Servicios de la Sociedad de la Información y del Comercio Electrónico (LSSI).
- Ley 32/2003 General de Telecomunicaciones.
- Código Penal para delitos de estafa a través de medios electrónicos.
- Gestión de la Propiedad Intelectual.

Bajo la normativa marcada a nivel nacional por las leyes y decretos expuestos anteriormente, tendría cabida la integración de un conjunto de propiedades contractuales sobre los cuales intentar definir un marco común que regule los aspectos directamente ligados al uso de servicios en la nube. Dichos aspectos, incidirían directamente en las siguientes áreas operativas concretas:

- **Ámbito de aplicación de procesamiento de datos:** conjunto de normativas dirigidas a regir qué tipo de datos pueden ser procesado y con qué propósitos exactos.
- **Outsourcing:** Para el caso de que el proveedor de servicios requiera de la subcontratación de terceros, se ha de definir el alcance de estos en cuanto al acceso a los datos manejados.
- **Acuerdo de niveles de servicio (SLA):** Medidas orientadas a la disponibilidad y recuperación de la información.
- **Pérdida de la información:** Se ha de definir un plan concreto que defina las responsabilidades, penas y prejuicios inmediatos en cuanto a situaciones derivadas de la pérdida o robo de la información tanto en el lado del usuario, como en el del proveedor del servicio.

- **Medidas de seguridad de la información:** La responsabilidad de los proveedores de servicio por salvaguardar la información gestionada como parte de las garantías del servicio ofrecido, le obliga a implantar políticas internas de encriptamiento y/o seguridad en la mantenibilidad de los datos manejados.
- **Transferencia y/o borrado de la información:** Se debe asegurar mediante un sistema de garantías, que el proveedor del servicio elimine de manera íntegra todos los datos recopilados de un usuario tras la finalización de los servicios prestados.
- **Ubicación de la información:** La descentralización y flexibilidad ofrecida por el uso del paradigma del Cloud Computing, obliga a definir políticas que aseguren que los datos de un usuario se mantengan y sean procesados siempre desde un lugar fijo en el mapa. En caso contrario, se deberán implementar políticas de consentimientos expresos que hagan conocedor al usuario del servicio del movimiento de sus datos.
- **Segregación de la información:** La información de cada usuario deberá ser gestionada de manera individual e independiente a la de cualquier otro usuario.

Bajo este contexto, y tras haber analizado los principales retos y aspectos contractuales implicados en la posible regulación del marco de privacidad y seguridad de los datos gestionados en la nube, la pregunta es, ¿Qué mínimo legal tienen que incluir las condiciones generales de contratación en materia de protección de datos para un servicio en la nube? A continuación, se exponen las principales características que debería recoger cualquier acuerdo contractual dentro del marco regulatorio que rodea la seguridad e integridad de la información en la nube en base a las particularidades presentadas:

- Debe quedar correctamente reflejado el alcance dispuesto por el proveedor de servicios en cuanto al uso y tratamiento de los datos recopilados de un usuario.
- Se ha de dejar constancia de que el proveedor de servicios no comunicará los datos manejados de un usuario a agentes externos del proceso. Y bajo el contexto específico bajo el cual el usuario acepte estos términos, ha de ser el propio proveedor quien se asegure de manera contractual que los servicios ofrecidos por terceros, operen de manera vigente al marco de garantías jurídicas impuesto en el EEE (Espacio Económico Europeo).
- El acuerdo contractual debe incluir que, una vez finalizada la prestación de los servicios por parte de cualquiera de las dos partes, el proveedor de servicios en la nube se hará responsable de devolver la propiedad íntegra de los datos al cliente. Ya sea mediante un traslado, a sus propios sistemas o a los sistemas de otro proveedor de servicios en la nube diferente.

Además de los aspectos regulatorios listados anteriormente en materia contractual, se debe destacar la necesidad añadida de incluir las cláusulas requeridas en cuanto al tratamiento y garantía de cumplimiento de la privacidad con respecto a los datos manejados, para aquellos proveedores que presten sus servicios fuera del EEE (Espacio Económico Europeo) y queden, por tanto, exentos del ámbito regulatorio identificado.

No obstante, y más allá del propio conjunto de normas y leyes presentadas, existen otras formas a nivel operativo vinculadas directamente a maximizar la integridad de los datos como forma inequívoca de mejorar aspectos que atañen a la privacidad del uso de los servicios en la nube. Así, por ejemplo, y a sabiendas de que el modelo de nube pública presenta un componente mayor en cuanto a índices de flexibilidad e inversión por parte del usuario se refiere, el uso de los servicios a través del modelo de nube privada otorga al usuario una mayor capacidad de control sobre la infraestructura de servicios manipulada, lo cual, repercute favorablemente a salvaguardar la integridad de los datos e información manejada en todo momento. Aspecto que no quita que existan otra serie de riesgos y amenazas persistentes tal y como se ha detallado a lo largo del presente capítulo.

En definitiva, en base a la información contenida en el presente capítulo, queda más que justificada la necesidad inherente de agregar nuevos mecanismos jurídicos y legales que acoten los derechos y deberes de los usuarios y proveedores de servicios en la nube, manteniendo así un escenario autoritario que regule los principales pilares que rodean a la integridad y seguridad de los datos manejados por los mismos. No obstante, y al tratarse de una innovación reciente en cuanto a la manera de entender el paradigma que supone el uso del Cloud Computing como metodología relativamente reciente y en constante evolución, este punto inevitablemente exige un sobreesfuerzo en la coordinación de equipos legales y actores involucrados en el proceso, especialmente los proveedores de servicios en la nube y los usuarios y empresas que los contratan.

5 Mercado Actual del Cloud Computing

Como prólogo en la definición del caso de uso práctico implementado como parte del capítulo 6: “*Caso Práctico: Desarrollo y Despliegue de un Servicio Real en la Nube*” de la presente memoria, a lo largo del presente capítulo se expondrá un breve recorrido en torno de la evolución que ha sufrido el paradigma del Cloud Computing hasta convertirse en el modelo existente en la actualidad, así como del crecimiento observado en cuanto a las cifras de negocio manejadas por las grandes compañías del sector de las nuevas tecnologías.

A continuación, se contextualizará los principales usos actuales del Cloud Computing como metodología emergente, a través del detalle comparativo ofrecido por los grandes proveedores de servicios existentes en la nube, entre ellos, Amazon, Microsoft o IBM.

De este modo, conforme se expone a lo largo de la presente memoria, y en base a las propiedades y ventajas objetivas que rodean el uso del Cloud Computing como metodología de servicios en la nube, la irrupción presentada por este nuevo paradigma dentro del contexto tecnológico actual ha roto con las perspectivas de soluciones *on premise* que se venían utilizando hasta la fecha, donde el usuario era poseedor de su propia infraestructura, normalmente alojada en un espacio físico o entorno determinado, y era el último responsable de asegurar el estado, control y mantenibilidad de la misma.

En este sentido, durante el segundo trimestre de 2018, los ingresos generados por infraestructuras de tipo Cloud Computing igualaron las cifras presentadas por las infraestructuras más tradicionales. La tendencia ascendente presentada en torno a la inversión en este punto, hizo que, durante el segundo semestre de ese mismo año, los niveles de facturación repercutidos en infraestructuras en la nube, superasen a los presentados por las arquitecturas tradicionales que se venían utilizando hasta la fecha, dejando a la cabeza el paradigma del Cloud Computing como líder indiscutible en cuanto a modelos de servicio de infraestructuras en el mundo TI.

En la siguiente Figura 5-1 adjunta, se representan los resultados de la encuesta realizada en 2019 por la prestigiosa editorial estadounidense International Data Group (IDG) a más de 500 profesionales del sector de las nuevas tecnologías, como forma de medir las tendencias de la computación en la nube entre los responsables de la toma de decisiones tecnológicas. En dicha encuesta, se les preguntó a los usuarios acerca del número de aplicaciones o servicios que tenían alojados en la nube en ese mismo momento.

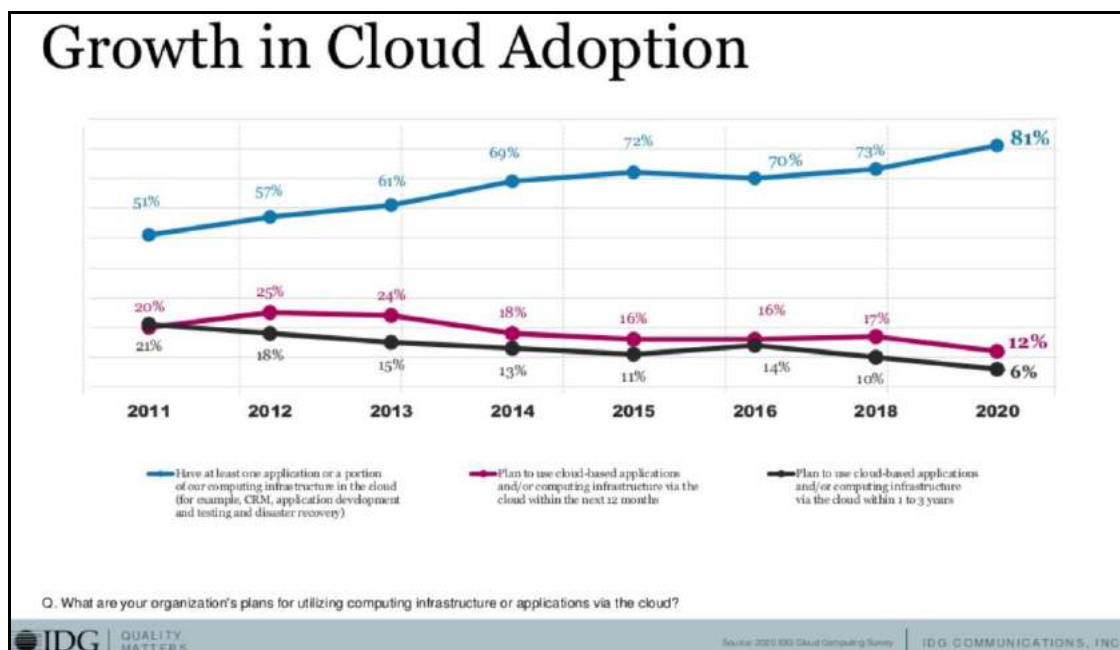


Figura 5-1: Gráfica del Crecimiento en la Adopción de Soluciones Cloud Computing en Empresas (2011 – 2020) ⁶

Los resultados observados muestran como la gran mayoría (81%) de las organizaciones y usuarios encuestados, poseían al menos un elemento de su infraestructura en la nube, lo que representaba un crecimiento del 73% con respecto a los datos manejados en 2018. Así, y de la mano de este último dato, creció también la inversión producida en este tipo de infraestructuras, donde los resultados obtenidos, mostraron como se estimaba un gasto total de más de 73,8 millones de dólares de inversión, valor que por supuesto, se mantendría al alza en comparación con los de años atrás.

Además, el estudio realizado presentado anteriormente, permitió obtener una visión global del crecimiento en la adopción del Cloud Computing como metodología emergente dentro del marco tecnológico presentado en el contexto actual. Crecimiento que quedaría avalado por las cifras resultantes siguientes:

- Únicamente el 8% de los encuestados admitió que, en la actualidad, el 100% de sus infraestructuras en el entorno informático seguían estando presentes en sus propias instalaciones frente al 92% restante.
- Alrededor de un 32% del presupuesto de las empresas y organizaciones asociado al contexto de las nuevas tecnologías, iría destinado a la mejora e integración de infraestructuras en la nube.
- Mientras que un total del 54% de las aplicaciones manejadas por los usuarios presentes en la nube provenían de migraciones de aplicaciones tradicionales basadas en un marco operativo local, el 46% restante fueron aplicaciones diseñadas y creadas específicamente bajo entornos nativos de Cloud Computing para trabajar en la nube.

⁶ Extraído de: Ruiz, F. J. (2020, 10 junio). Publicado estudio sobre Cloud Computing en empresas 2020. Blog de Datapius. <https://blog.datapius.com/index.php/2020/06/10/publicado-estudio-sobre-cloud-computing-en-empresas-2020/>

- Aproximadamente un 55% de los encuestados operaban bajo infraestructuras compuestas por varias nubes de carácter público.

De la mano de estas cifras, son muchas las organizaciones y empresas que recientemente han ido incorporando este paradigma a su operativa diaria dentro del entorno TI que las rodea. Así, por ejemplo, el reconocido periódico norteamericano The New York Times, logró convertir a PDF más de 11 millones de artículos e imágenes tomadas entre 1851 y 1980 en 24h, gracias al uso de instancias de Amazon EC2 (Amazon Elastic Compute Cloud); compromiso que, si se hubiera planificado mediante el uso de arquitecturas clásicas, le hubiera llevado alrededor de 7 semanas de trabajo continuo.

Dadas las cifras reflejadas en torno al rápido crecimiento en el volumen de negocio aparente en el sector, y a raíz de las nuevas oportunidades que supone el hecho de maximizar el alcance de prestaciones en cuanto a la manera de conceptualizar los servicios en la nube como eje central del Cloud Computing, son muchas las compañías que se han incorporado al mercado como proveedores de servicios en la nube. A continuación, se adjunta un breve repaso en torno a algunos de los proveedores a nivel europeo que manejan un mayor volumen tanto de usuarios, como de facturación y servicios ofrecidos, y que, por tanto, se conciben como las más representativas en el sector:

MICROSOFT

A través de su plataforma comercial Microsoft Azure, desde 2014 Microsoft es uno de los pocos proveedores en el mercado en alojar entre sus servicios el alojamiento completo de sistemas, es decir, es capaz de ofrecer la posibilidad de adquirir las tres capas completas que conforman los modelos basados en servicios: IaaS, PaaS y SaaS.

Entre sus principales propiedades, además, se ha de destacar el uso de entornos basados en nubes híbridas, que, junto a su compromiso por la implementación bajo código abierto, le convierten en uno de los principales proveedores de máquinas virtuales basadas en SO, tanto en Linux, como en Windows, lo que le otorga una cuota de mercado que ronda el 15% con respecto a sus competidores.

AMAZON

Con casi la mitad de la cuota actual en el mercado de infraestructuras de nube pública, Amazon se sitúa como uno de los grandes líderes como proveedor de servicios en la nube. Bajo la marca comercial de Amazon Web Services (AWS), desde su nacimiento, la compañía americana se considera como una de las pioneras en el sector de servicios en la nube.

Entre su catálogo de servicios, cuenta con una oferta variada que incluye el aprovisionamiento de las tres capas del modelo: IaaS, PaaS, SaaS, esta última menos desarrollada, a las que incorpora, además, una interfaz de gestión que optimiza las labores de accesibilidad a sus servicios, y que le otorga una gran capacidad de escalado y flexibilidad a la operativa que ofrece. Todo ello, a un precio muy competitivo dentro del sector.

Además, actualmente AWS es la única nube que ofrece una red Ethernet de 100 Gbps para instancias informáticas. Hecho que, sumado a la implementación de su plataforma Amazon EC2 (Amazon Elastic Compute Cloud), le convierten en uno de los proveedores idóneos para aquellas tareas computacionales que requieran una mayor velocidad de procesamiento, tales como desarrollo de videojuegos, o machine Learning entre otras.

IBM

Aprovechando su amplia experiencia en tecnología de software, IBM ha dado pasos de gigante en los últimos años y ha conseguido escalar posiciones frente a otros proveedores cloud del mercado. Watson, que es como se llama su plataforma, ofrece servicios IaaS, PaaS y SaaS, con entornos de nube pública y nube híbrida, y aunque su oferta no cuenta con el extenso catálogo de soluciones y productos de otros proveedores, tiene sus puntos fuertes en IA, Big Data, Bots y Analytics, apostando claramente por el futuro que se vislumbra en torno al desarrollo de la IA.

De este modo, en los últimos años IBM ha escalado posiciones en el mercado con respecto a sus competidores más cercanos tras haber superado ya a otros proveedores como Salesforce o SAP entre otros. Así, tras adquirir al proveedor RED HAT, por la cifra de 34.000 millones de dólares en 2019, IBM se ha provisto de su experiencia en el desarrollo software para implementar su propia plataforma Watson de servicios en la nube.

Dicha plataforma ofrece entre sus servicios al usuario diferentes entornos basados tanto en nube pública, como en nube híbrida, y al igual que los proveedores vistos hasta ahora, es capaz de ofrecer la virtualización de un sistema integro gracias a los modelos de servicio: IaaS, PaaS y SaaS. En este sentido, y aunque el alcance de su catálogo de servicios sea mucho menor en comparación a sus rivales más directos, la plataforma Watson incorpora diferentes métodos basados en las nuevas tecnologías como son la IA, el Big Data o Analytics, que lo convierten en un claro competidor con respecto al marco evolutivo presente en el desarrollo del sector TI.

Además de los ejemplos de compañías citadas anteriormente, y dado el gran impacto económico mundial que ha tenido la gestión y oferta de servicios en la nube, han sido numerosas las empresas que se han lanzado a abordar este nuevo reto desde un punto de vista de provisión de este tipo de servicios. Así, bajo el paradigma que envuelve el Cloud Computing, grandes compañías internacionales como Alibaba, Google u Oracle, componen diferentes alternativas para el usuario dentro de un mercado cada vez más competitivo.

En base a la información proporcionada en el presente capítulo, queda evidenciado como el Cloud Computing ha pasado a formar parte del contexto actual en materia tecnológica, como una de las principales herramientas del mercado capaces de ofrecer grandes índices de rendimiento en cuanto a la gestión y optimización de tareas bajo el contexto IT.

En este sentido, el conjunto de ventajas objetivas que presenta el uso e implementación de este paradigma, suponen actualmente un gran aliciente para las empresas, lo que se traduce en no solo un crecimiento de los beneficios derivados de los costes que suponía el mantenimiento y escalabilidad de las arquitecturas tradicionales, sino de una nueva manera de hacer negocios en torno a la creación y gestión de nuevos modelos de servicio, suponiendo así, una mayor presión competitiva para implementar nuevas oportunidades en base al cambio que ello supone entre los grandes, y no tan grandes, proveedores de servicios la nube.

6 Caso Práctico: Desarrollo y Despliegue de un Servicio Real en la Nube

Continuando con la hoja de ruta definida en cuanto a los objetivos específicos planteados al inicio de la presente memoria, y una vez analizados teóricamente los principales apartados que rodean el paradigma del Cloud Computing como metodología emergente, dentro del presente capítulo se describe paso a paso la ejecución y desarrollo de un caso práctico real, como forma de evidenciar de manera empírica el uso de la metodología Cloud Computing en un entorno real operativo, a través de la codificación y despliegue de una aplicación web sencilla, que implementará una serie de llamadas y subrutinas a los servicios ofrecidos por AWS de Amazon como principal proveedor de servicios en la nube, bajo el modelo de servicios FaaS presentado.

En este sentido, es importante destacar el hecho de que el verdadero objetivo perseguido en el presente capítulo no reside en la dificultad comprendida en cuanto a la codificación funcional de la propia aplicación web presentada, ya que, el verdadero propósito es, tal y como se introduce en los objetivos iniciales de la memoria, la comprensión, conceptualización y familiarización con un entorno real de uso de la metodología Cloud Computing, a través de la gestión unificada de servicios ofrecidos por un proveedor de servicios real en la nube bajo un modelo FaaS, cuyo manejo y configuración de herramientas internas permitirán la comprensión de conceptos clave en cuanto a la gestión de servicios alojados en la nube, como pueden ser el desarrollo y despliegue de una arquitectura tipo serverless, la administración de recursos alojados en la nube mediante el manejo de bases de datos del tipo NoSQL o la integración del carácter interactivo mediante el uso de API específicas.

6.1 Arquitectura

En primer lugar, y como parte de la Figura 6-1, se aporta el detalle entorno a la representación gráfica de la arquitectura de servicios que intervendrán a lo largo del caso práctico, así como de las interrelaciones establecidas entre cada una de ellos. Cada uno de los servicios incluidos, implementará una serie de funcionalidades integradas entre sí como requisito inherente en cuanto al correcto desarrollo de la aplicación serverless objetivo bajo el modelo de servicio FaaS utilizado.

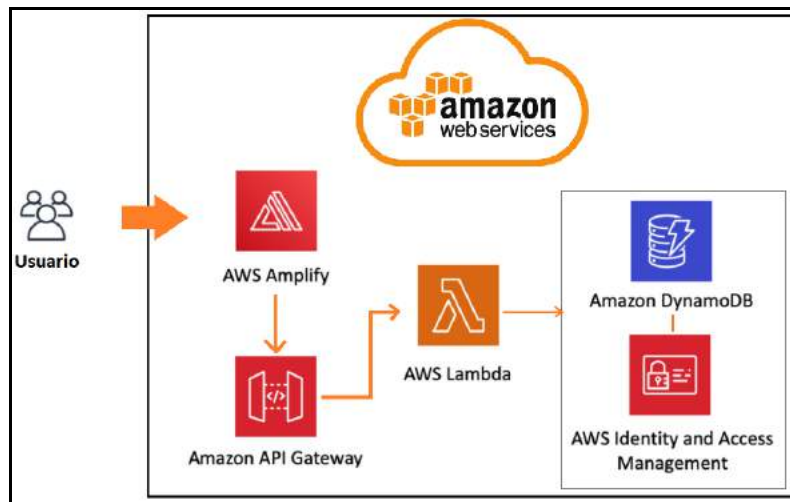


Figura 6-1: Caso Práctico: Desarrollo y Despliegue de un Servicio Real en la Nube – Descripción de la Arquitectura del Sistema

Como ya se ha mencionado, la arquitectura planteada girará en torno al concepto serverless, también conocido como FaaS (*Functions as a Service*), que, al contrario de lo que pueda hacer intuir su traducción literal como “sin servidor”, la realidad es que este tipo de arquitecturas incorporan servicios del tipo BaaS (*Backend as a Service*), a través de los cuales, y mediante un concepto de “computación efímera”, se encarga de resolver peticiones concretas en base al uso de contenedores momentáneos y sin estado, apoyándose en las ventajas de la metodología Cloud Computing, lo que en ningún momento quiere decir que no exista un servidor detrás de la lógica global de la arquitectura.

Sin embargo, este modus operandi permite al usuario abstraer el concepto “servidor”, lo que implica prescindir de las tareas relativas a la gestión, administración y mantenimiento de las infraestructuras físicas utilizadas hasta el momento, permitiendo así al usuario beneficiarse de una serie de ventajas añadidas, y siendo el proveedor de servicio quien se encargue de realizar la gestión del autoescalado del desarrollo concreto. Este hecho, conlleva a su vez ciertas ventajas añadidas, tales como:

- **Alta Escalabilidad y Disponibilidad:** El proceso de escalado de una aplicación o servicio se realiza prácticamente de manera automática a través del aumento de las unidades de consumo del servicio contratado al proveedor. En este caso, el usuario dispondrá siempre de los servicios que necesite de una forma que se evite el aprovisionamiento desmedido de recursos, al contrario de las arquitecturas tradicionales, donde de manera inherente, más escalabilidad, requiere un mayor número de servidores físicos.
- **Evita costes asociados a la administración de un servidor:** Las tareas de gestión, mantenimiento y administración del servidor, corren por parte del proveedor de servicios elegido, por lo que el usuario no deberá hacerse cargo de este tipo de tareas subyacentes.
- **Pago únicamente por Consumo:** Los costes del servicio se adecúan al uso exacto precisado por el usuario, evitando así posibles sobrecostes, y ciñéndose estrictamente al rendimiento y capacidad de cómputo requeridos por cada usuario en cada momento determinado.

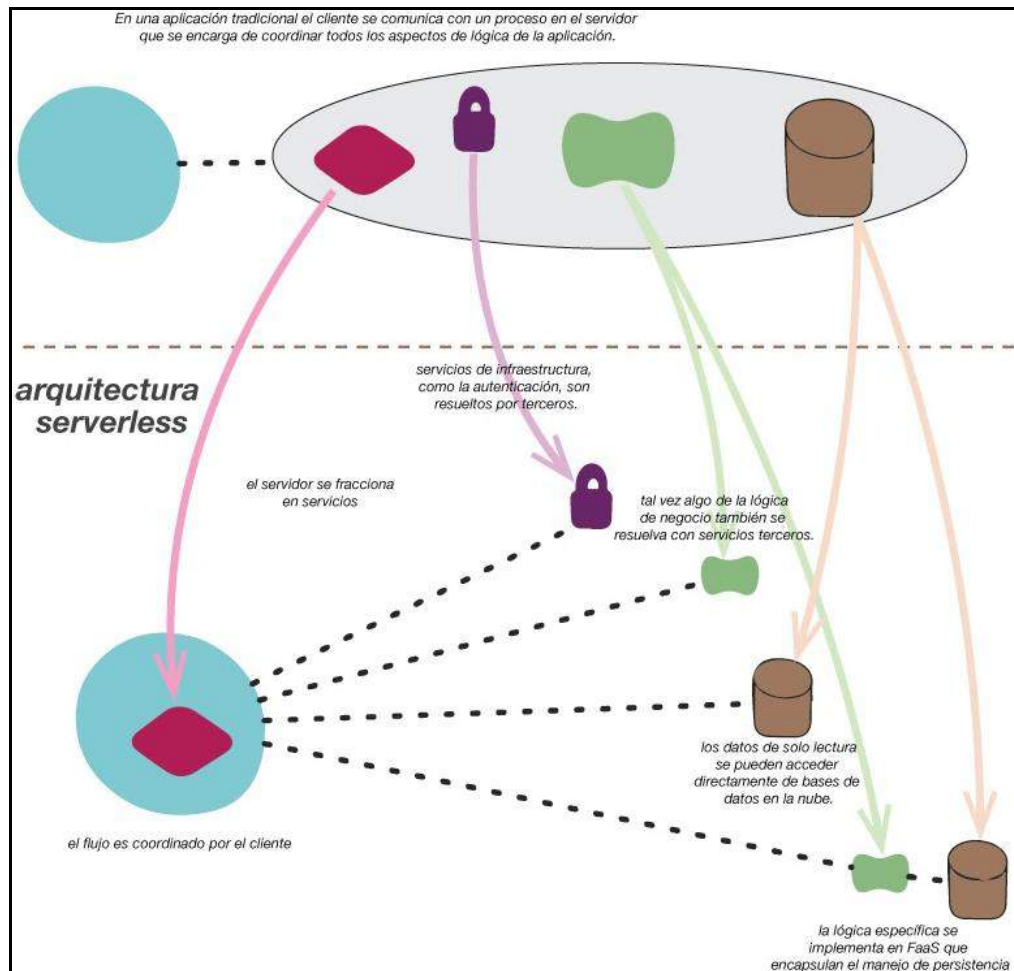


Figura 6-2: Representación del Modelo Lógico Funcional de las Arquitecturas Serverless ⁷

Así, y conforme se puede apreciar en la Figura 6-2 anterior, en el primer caso, en el cual se implementa la lógica asociada a una arquitectura tradicional, el conjunto de servicios concurrentes permanece encapsulados dentro de un mismo servidor, mientras que, en la segunda parte de la figura, vemos como cada servicio es implementado por un tercero de manera independiente (arquitectura basada en microservicios), quien se encarga de la gestión, procesamiento y mantenimiento de cada uno de ellos.

Debido a sus enormes ventajas aparentes, y dentro de la creciente demanda de nuevos servicios ofrecidos por los principales proveedores en la nube, el uso de este tipo de arquitecturas serverless como modelo de servicios FaaS, se ha extendido entre las principales compañías del sector, como pueden ser: Amazon AWS (*Amazon Lambda*), Google Cloud (Google Cloud Functions) o Microsoft Azure (Microsoft Azure Functions). En estos casos, los diferentes servicios son gestionados y servidos por un mismo proveedor.

En este sentido, y tras la caracterización realizada en torno la arquitectura sobre la cual irá desplegada la aplicación a implementar, a continuación, se detalla brevemente cada uno de los servicios involucrados en el desarrollo y despliegue de la aplicación como contextualización técnica sobre el caso práctico implementado posteriormente:

⁷ Extraído de: Galván, P. Un Vistazo a la Arquitectura Serverless. <https://sg.com.mx/revista/52/un-vistazo-la-arquitectura-serverless>

- **AWS Amplify:** En primer lugar, AWS Amplify se presenta como un conjunto de herramientas y servicios orientados al desarrollo de aplicaciones web y móvil dentro de la infraestructura AWS. En este sentido, y bajo el marco de código abierto como principal característica en este punto, el usuario posee la capacidad de administrar un conjunto de diferentes componentes, interfaces y servicios, todos ellos basados en la política de uso en la nube, a través de los cuales, se ofrece la posibilidad de acelerar el desarrollo de aplicaciones web y móviles dentro de AWS, lo que incluye el acceso a diferentes funciones y servicios específicos tales como: análisis, notificaciones, autenticación o datos sin conexión a gran escala entre otros.
- **Amazon API Gateway:** Junto a AWS Lambda, API Gateway conformará el conjunto de servicios orientados a implementar el componente serverless del caso práctico presentado. Concretamente, su rol dentro de la arquitectura planteada será el de ejercer de nexo para que el resto de la infraestructura backend implementada pueda acceder al conjunto de servicios alojados en AWS, así como a los datos almacenados en la nube. Lo que incidirá directamente en la gestión y optimización de las tareas de despliegue de la aplicación planteada.

Para ello, y orientado al caso práctico propuesto, el presente servicio ofrecerá la posibilidad de desarrollar una API del tipo RESTful, basada en protocolos HTTP, que habilitará la comunicación entre el cliente y el servidor sin estado. Además, y de manera inherente, incluirá una serie de tareas añadidas a la propia gestión del canal de comunicación como son el control de acceso y autorización, el monitoreo del servicio, o la administración del tráfico en la propia aplicación. Tareas que serán ejecutadas de manera totalmente autónoma, y que irán implícitas en el propio uso del servicio.

- **Amazon DynamoDB | AWS Identity and Access Management (IAM):** Amazon DynamoDB es un servicio de creación y gestión de bases de datos no relacionales del tipo NoSQL, que destaca por su rapidez y flexibilidad en cuanto a la gestión de datos en servicios de aplicaciones que requieren de latencias constantes a cualquier escala. Su principal característica, es la administración integral del dato en la nube, haciéndola compatible con prácticamente todos los modelos de almacenamiento de datos clave – valor, y automatizando las labores de gestión y administración de infraestructura de manera autónoma, por lo que el usuario no debe preocuparse en este punto por las labores de aprovisionamiento, instalación o configuración de la base de datos.

En cuanto a los principales beneficios y ventajas presentadas por este servicio, la mayoría de ellas concuerdan con las ya mencionadas relativas al uso y gestión del dato mediante bases de datos del tipo NoSQL, entre las cuales destacan: Alta disponibilidad y durabilidad del dato, eficiencia y desempeño, escalabilidad, administración autónoma, programación basada en eventos, flexibilidad o control de acceso minucioso entre otras.

Además, y gracias a la integración conjunta con el servicio AWS Identity and Access Management (IAM), la arquitectura propuesta para la aplicación, permitirá desarrollar un control verificado de acceso basado en la autenticación de identidades, mediante el cual, se llevará a cabo el registro de autorizados y permisos de acceso a los datos y recursos contenidos en la base de datos.

- **AWS Lambda:** AWS Lambda compondrá la base de ejecución entorno a la cual gire el resto de la infraestructura a implementar. Su principal función como servicio integrado, es el de ejecutar fragmentos de código basado en eventos, sin la necesidad de disponer de un servidor físico como tal. En este sentido, y a través de las políticas de autoescalado ofrecidas, el presente servicio asegura la respuesta de la parte del servidor, indistintamente de la carga que maneje en cada caso.

De este modo, AWS Lambda permite la ejecución de código sin la necesidad añadida de aprovisionar ni administrar servidores, crear una lógica de escalado de clústeres basada en la carga de trabajo, mantener integraciones de eventos o administrar tiempos de ejecución. Esta serie de principios, compondrán la base de la tecnología serverless a implementar bajo el marco de metodología Cloud Computing identificada.

Dentro del caso de uso propuesto, y como parte del esquema de la arquitectura presentada en la Figura 6-1, el servicio AWS Lambda se encargará de manejar y procesar las peticiones entrantes generadas por el servicio AWS Amplify, y conmutadas a través del servicio Amazon API Gateway. Además, integrará la lógica para la administración de la información manejada dentro de la base de datos NoSQL generada a partir del servicio Amazon DynamoDB.

Llegados a este punto, y tras haber analizado funcionalmente cada uno de los servicios que intervendrán en la consecución de la infraestructura conjunta que implementará el caso práctico de uso propuesto, se detalla a continuación, el detalle paso a paso de la ejecución de cada una de las etapas comprendidas como parte del proceso pormenorizado de desarrollo y despliegue llevado a cabo. En este sentido, y debido al gran volumen de capturas existentes, las distintas evidencias derivadas del proceso de implementación y despliegue quedan recogidas como parte del “*Anexo C: Caso Práctico: Desarrollo y Despliegue de un Servicio Real en la Nube – ANEXO Evidencias de Implementación*”.

6.2 Implementación de un Recurso Web Estático

6.2.1 Objetivo

El objetivo de este primer punto, será el de llevar a cabo la implementación de los recursos estáticos que compondrán la aplicación web en la nube. Para ello, se hará uso de la Consola de AWS Amplify, que, junto con el servicio respectivo, se encargarán de alojar los recursos web estáticos manejados, así como de implementar el punto de acceso mediante la URL facilitada por el propio servicio.

En este punto, y dado que el presente caso práctico no comprende el desarrollo de una aplicación web real, se optará por el uso de un dominio genérico, ofrecido por el propio proveedor de servicios AWS.

6.2.2 Desarrollo

En primer lugar, se llevará a cabo la codificación del recurso web mediante el desarrollo de un fichero de tipo HTML sencillo, que muestre por pantalla un texto sin formato ninguno. A continuación, se deberá crear una nueva instancia dentro de la consola de servicio de AWS Amplify que permitirá el despliegue del entorno bajo el cual se ejecutará el HTML generado, eligiendo para ello la opción “*Deploy without Git provider*”.

Bajo estas condiciones, se procederá a realizar el despliegue y ejecución del recurso estático implementado a través del dominio (URL) generado por la propia consola de AWS Amplify, dando como resultado, la ejecución del recurso web estático codificado.

6.3 Desarrollo de una Función Lambda

6.3.1 Objetivo

A través de la consecución del presente apartado, y bajo la codificación del servicio AWS Lambda, se logrará obtener y configurar una función Lambda, basada en la arquitectura serverless que, gracias a la integración conjunta con el apartado anterior, permitirá realizar la invocación de un servicio bajo demanda concreto mediante la implementación de eventos del tipo Lambda Trigger.

Este tipo de funciones serverless, tienen por objetivo eliminar la necesidad de administrar tanto el Software como el Hardware que implica el manejo de un Servidor. De este modo, la operativa de este tipo de funciones se reduce a la invocación de servicios bajo demanda especificados dentro del código manejado y alojados en la propia nube de servicios ofrecidos por el proveedor.

6.3.2 Desarrollo

Una vez validado el recurso estático, el siguiente hito en el proceso será el de llevar a cabo la implementación de la función Lambda como eje central de la arquitectura serverless objetivo. Para ello, se deberá acceder a la consola provista por el servicio AWS Lambda, y a continuación, generar una nueva función desde cero. Para este caso concreto, se seleccionará Python 3.8 como lenguaje de programación de la función. Una vez generada la función, se deberá definir computacionalmente el recurso que se usará como respuesta a la petición de servicio al proveedor. En este caso, la respuesta comprenderá la unión de [*“Hello from Lambda”* + *nombre* + *apellido*]. Donde *nombre* y *apellido* serán los valores de entrada que recibirá el servicio para construir la respuesta y que se deberán implementar como petición desde el recurso web a modo formulario.

Una vez codificada la función Lambda, se generará un evento de prueba dentro de la misma consola de AWS, en la cual, se valide la operativa de la misma a través de la inserción manual de un fichero JSON con los argumentos de entrada que espera el servicio requerido. Es decir, un *nombre* y un *apellido*.

6.4 Integración de la Aplicación Web y la Función Lambda

6.4.1 Objetivo

Como resultado de la ejecución del presente apartado, y a través del uso del servicio ofrecido Amazon API Gateway se llevará a cabo la codificación integral de una API de tipo RESTful, que implementará las funciones de llamada a la función Lambda desde cualquier navegador web como elemento intermedio entre el backend interpretado por la función Lambda y el recurso HTML desarrollado en el primer apartado. Dicho proceso, requerirá, además, de la configuración adicional de los métodos HTTP específicos de la API como método de validación del desarrollo logrado.

6.4.2 Desarrollo

Implementada tanto la función Lambda, como el recurso web que ejercerá las labores de formulario de entrada de los argumentos requeridos por la primera función, se deberá implementar la integración entre ambos apartados a través del servicio Amazon API Gateway.

Dicho proceso, será llevado a cabo mediante la configuración de una API RESTful específica, sobre la que se deberá llevar a cabo la construcción de un método POST que dé respuesta a la petición entrante (Función Lambda). En este punto, es importante destacar la necesidad de activar el CORS a la hora de establecer los criterios definidos para el método POST, ya que, sin esta propiedad, el servicio carecería de los permisos requeridos a la hora de leer los argumentos de entrada introducidos en el HTML como recurso estático en la parte del usuario.

Finalmente, y al igual que en los apartados anteriores, se generará un pequeño caso de prueba para testar de un modo simulado, la validez del método POST generado dentro de la propia consola de Amazon API Gateway.

6.5 Creación de la Base de Datos NoSQL

6.5.1 Objetivo

Mediante el despliegue y parametrización de la consola de administración AWS, y a través del uso del servicio DynamoDB proporcionado por el mismo proveedor, el principal objetivo del presente apartado será el de generar un servicio clave-valor bajo una estructura de base de datos del tipo NoSQL, que permita almacenar la información manejada por la aplicación web.

Para ello, además de generar la estructura de almacenamiento mencionada, será necesario implementar los permisos de acceso a la base de datos mediante la llamada al servicio AWS Identity and Access Management (IAM), de manera que la función Lambda, pueda acceder y registrar la información manejada como parte de la estructura clave-valor alojada en la nube.

6.5.2 Desarrollo

Conforme así se indica en el objetivo del presente apartado, para el desarrollo en este punto se hará uso del servicio DynamoDB a través del cual, se generará un nuevo recurso o tabla, donde se albergarán los valores de llamada a la función Lambda como único par de clave-valor a albergar, y que, por tanto, conformará el ID de acceso a los elementos de la tabla.

A continuación, y a través del servicio IAM, se le concederán los permisos requeridos de escritura, lectura y modificación a la función Lambda generada a través de la integración entre ambos componentes mediante la creación de una política de acceso específica. De este modo, se evitará que otras funciones creadas por terceros, puedan interactuar con la base de datos generada para este caso.

6.6 Despliegue de la Aplicación Web en la Nube

6.6.1 Objetivo

Una vez implementados todos los apartados que conformarán el servicio a desplegar, el último paso en el proceso será el de agregar interacción a la aplicación web, para ello, se deberá actualizar el resultado obtenido tras la primera interacción del proceso “6.2 Implementación de un Recurso Web Estático” de manera que éste invoque la API REST resultante del tercer apartado.

Este proceso permitirá agregar al servicio la capacidad de invocar a la API generada, y con ello, desplegar la aplicación web que permitirá al usuario mostrar el texto introducido en el primer paso de la interacción.

6.6.2 Desarrollo

Codificados de manera independiente los distintos recursos intervinientes en el proceso operativo objetivo: Formulario Web Estático, Función Lambda, Base de Datos NoSQL y API; el siguiente paso previo al despliegue final, será el de interconectar todos los elementos entre sí. Para ello, en la función Lambda, y después de haber generado la política de acceso respectiva, se deberá implementar el proceso que le permita escribir los valores de entrada sobre la Base de Datos construida. De tal forma, que estos queden registrados como parte del procedimiento de la aplicación objetivo.

A continuación, y dentro de la parte del cliente implementada por el recurso web, se codificará la llamada a la API creada en el tercer paso. Dicha llamada, deberá implementar el procesado previo de los argumentos de entrada correspondientes, a tipo JSON, los cuales, serán distribuidos por la API a la función Lambda para su posterior procesamiento y registro en la base de datos.

Realizadas las integraciones requeridas, se desplegará nuevamente la versión extendida del recurso web conforme se realizó en el primer paso del proceso. En este caso, el recurso web implementará un formulario en el que, tras introducir el usuario un nombre y un apellido en los campos correspondientes, la llamada a la API ejecutará el evento Lambda Trigger de la función, y permitirá completar el proceso implementado en cuanto al procesamiento y registro de los valores en la base de datos creada, permitiendo a su vez al usuario, visualizar los valores introducidos por pantalla como respuesta generada por la función Lambda.

A modo conclusión, y través del desarrollo y despliegue de la aplicación web presentada como caso práctico de uso, se logrará obtener una arquitectura de tipo serverless, basada en los principios del Cloud Computing como metodología descentralizada de computación, a través de la cual, y gracias a la integración de los servicios IaaS en la nube ofrecidos por un proveedor real como es AWS de Amazon, queda evidenciado empíricamente, como se puede lograr implementar una aplicación en la nube, 100% funcional, sin la necesidad de aprovisionamiento de un servidor físico real, como una de las principales virtudes presentadas por el manejo del paradigma del Cloud Computing.

7 Principales Corrientes Emergentes Derivadas del Cloud Computing: Edge Computing y Fog Computing

Dentro del contexto innovador que supone la evolución constante vislumbrada dentro del sector de las nuevas tecnologías, unido a la incesante demanda por descubrir nuevas metodologías capaces de procesar un mayor volumen de datos en un tiempo cada vez más reducido, muchas empresas comienzan a cuestionarse si el paradigma del Cloud Computing, o mejor dicho, la nube, será capaz de aguantar el ritmo del crecimiento en las necesidades mostradas por los usuarios y las nuevas tecnologías emergentes (5G, IoT, Industria 4.0, etc.).

Internet ha pasado de ser una mera fuente de información estática, a un mecanismo de alimentación de datos que ayuda a mejorar el rendimiento en cuanto a la potenciación de las capacidades de procesamiento y computación de alto nivel. Bajo este contexto, y entendiendo que no todos los datos transmitidos a la nube son procesados como parte de un servicio, surgen las siguientes cuestiones: ¿Es necesario que todos los datos sean enviados a la nube para su posterior procesamiento del servicio? ¿Existe alguna forma de almacenar selectivamente los datos en la nube?

Bajo estas dos premisas, recientemente han emergido diversas corrientes innovadoras cuya base conceptual deriva de las diferentes maneras de entender el paradigma del Cloud Computing. En el presente capítulo, se analizarán desde un punto de vista técnico-conceptual, las dos corrientes que han tomado una posición privilegiada en cuanto a su uso y adopción por los usuarios y las empresas que rodean al mundo de las nuevas tecnologías: Fog Computing y Edge Computing.

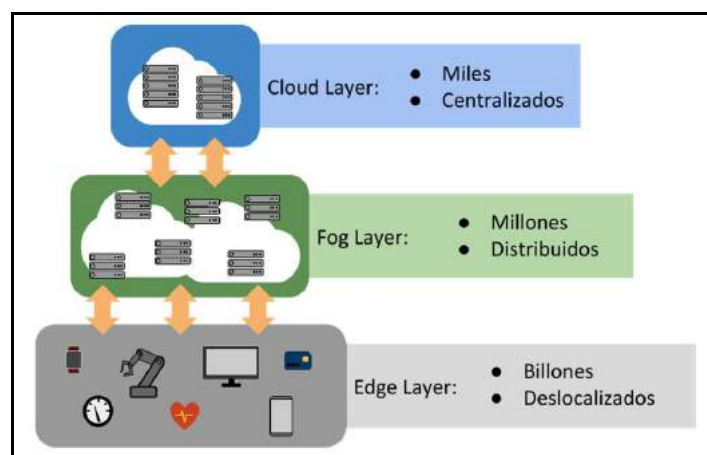


Figura 7-1: Principales Características en torno a las Corrientes Emergentes derivadas del Cloud Computing: Fog Computing y Edge Computing ⁸

⁸ Extraído de: Gradiant. (2018, 4 mayo). Edge/Fog Computing: del Cloud hacia la computación en los dispositivos. <https://www.gradiant.org/blog/edge-fog-computing-cloud/>

Tal y como se puede observar en la Figura 7-1, las tres metodologías componen una pirámide jerárquica en torno al alcance y política del procesamiento asociado a la cantidad de dispositivos manejados. La primera capa, referenciada en la Figura 7-1 como “*Cloud Layer*”, representa el uso de la metodología Cloud Computing bajo los esquemas conceptuales definidos a lo largo de la presente memoria. A continuación, se detalla brevemente la consistencia de las otras dos capas, bajo el modelo funcional que representan:

FOG COMPUTING

El concepto que engloba la metodología en torno al Fog Computing, hace referencia a una arquitectura de red descentralizada, en la que tanto los recursos, como los datos y aplicaciones, se alojan en un lugar lógico (Comúnmente denominado nodo o Gateway) entre la nube, y el dispositivo raíz que genera la información. Propone, por tanto, un modelo bajo el cual la distribución de la capacidad de cómputo se acerca hacia los extremos de la red.

Se entiende que bajo esta operativa las fuentes de datos estarían más próximas a los sistemas de procesado, minimizando así la distancia que recorren los datos en la red, lo que indirectamente repercute favorablemente respecto a los niveles de latencia y tiempo efectivo de procesamiento de la infraestructura de red.

Esta metodología se enfoca a servicios que requieran del procesamiento de información en tiempo real, y que, a su vez, su alto consumo de ancho de banda les permita beneficiarse de la baja latencia ofrecida por este paradigma. Además, al presentar un carácter distribuido, se logran maximizar ciertos aspectos que tienen que ver con la seguridad y la resiliencia que ofrecen de manera inherente este tipo de arquitecturas.

EDGE COMPUTING

El Edge Computing como metodología, permite la recopilación y procesamiento de datos en un mismo dispositivo inteligente situado al borde de la red gracias al uso de dispositivos PACs (Controladores de Automatización Programables). Estos, permiten realizar un pre-procesado de los datos que requieren del servicio, filtrando aquellos que no tienen necesidad de ser trasladados a la nube para que sean procesados, y sí únicamente los que necesiten ser almacenados. Dicha visión, conforma un planteamiento eficiente del procesamiento y gestión de la información en la nube, que, además, disminuye el número de posibles conflictos y errores durante el servicio.

Este tipo de arquitecturas suelen ir vinculadas al concepto del Internet de las Cosas (IoT), escenario en el cual resulta necesario que la latencia del procesamiento fluctúe entre valores mínimos, a la vez que se asegure la integridad y privacidad de los datos intercambiados a través de la red.

La manera de entender estas dos nuevas corrientes metodológicas, queda representada de manera gráfica en la siguiente Figura 7-2. En ella, se muestran las diferentes propiedades de cada capa, así como el modelo y rol que desempeñan dentro de la pirámide jerárquica conformada, en función del alcance operativo de procesamiento que presenta cada paradigma:

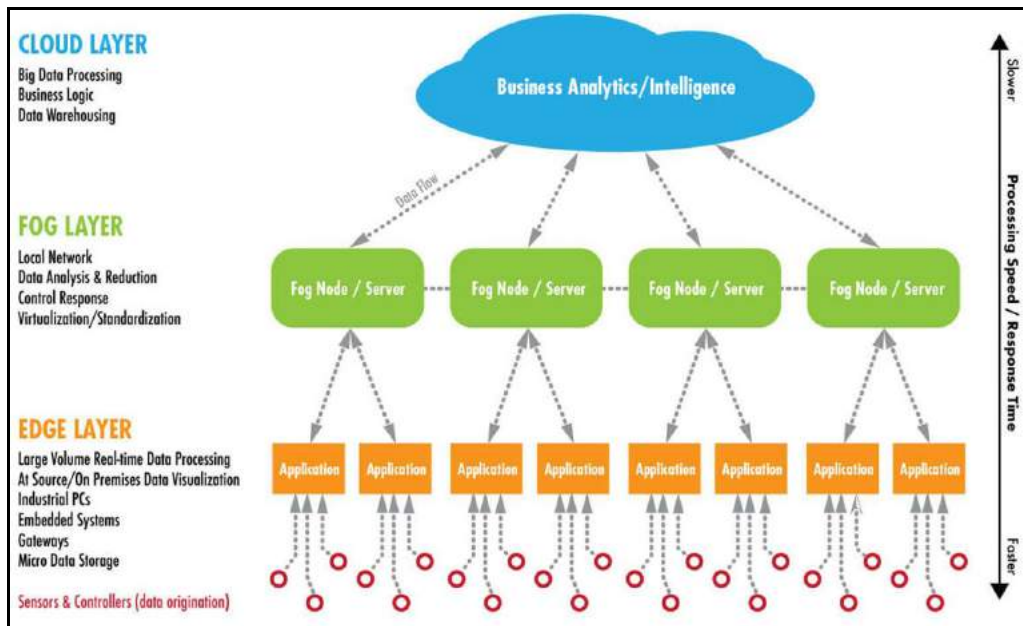


Figura 7-2: Esquema de la Pila de Procesamiento de Datos Industriales IoT ⁹

Para finalizar, se adjunta como parte de las tablas: Tabla 7-1, Tabla 7-2 y Tabla 7-3, un breve resumen comparativo en torno a los puntos de incidencia de cada una de las corrientes innovadoras descritas a lo largo del presente capítulo. En ellas, se puede observar las principales características de cada arquitectura dispuesta que rodean a cada metodología emergente, así como los beneficios e inconvenientes que aporta el uso de cada uno de los paradigmas presentados dentro del contexto actual tecnológico:

Tabla 7-1: Resumen Comparativo de Características Principales de Corrientes Emergentes: Cloud Computing

CLOUD COMPUTING	
ARQUITECTURA	<ul style="list-style-type: none"> • Modelo basado en el procesamiento centralizado de los datos. • Permite acceder a grandes cantidades de datos con alta flexibilidad, disponibilidad y rapidez.
VENTAJAS	<ul style="list-style-type: none"> • Fácil de escalar. • Bajo coste de almacenamiento. • La transferencia de datos está basada en protocolos de red robustos (TCP/IP).
DESVENTAJAS	<ul style="list-style-type: none"> • Latencia/Tiempo de Respuesta. • Altos costes del ancho de banda y consumo de energía. • Requiere de conexión a Internet. • Puede llegar a plantear problemas de privacidad, seguridad y legales.

⁹ Extraído de: Jenkins, J. (2019, 8 abril). Cloud, Fog and Edge Computing – What’s the Difference? WINSYSTEMS. <https://www.winsystems.com/cloud-fog-and-edge-computing-whats-the-difference/>

Tabla 7-2: Resumen Comparativo de Características Principales de Corrientes Emergentes: Fog Computing

FOG COMPUTING	
ARQUITECTURA	<ul style="list-style-type: none"> • Extiende el Cloud al extremo de la red – Computación descentralizada. • Cualquier dispositivo computable, con almacenamiento y con conexión a la red puede ejercer de nodo. • Procesa los datos en el mismo nodo, o en un dispositivo Gateway. Por tanto, no es necesario acceder a la propia red local de la arquitectura.
VENTAJAS	<ul style="list-style-type: none"> • Análisis de datos en tiempo real (STR). • Los datos sensibles permanecen dentro de la red mientras que los menos sensibles, se envían a la nube para su posterior procesamiento. • Ahorro de costes en almacenamiento y red. • Alta escalabilidad. • El despliegue de nodos puede hacerse de distintos puntos de vista de acuerdo a los niveles de privacidad requeridos: privado, compartido, público o híbrido.
DESVENTAJAS	<ul style="list-style-type: none"> • Depende de un gran número de nodos a la hora de transferir los datos del conjunto de dispositivos a la nube, lo que supone una clara desventaja en este punto.

Tabla 7-3: Resumen Comparativo de Características Principales de Corrientes Emergentes: Edge Computing

EDGE COMPUTING	
ARQUITECTURA	<ul style="list-style-type: none"> • No requiere trabajar conectado a la nube ni a ningún nodo de acceso. La inteligencia se traslada al borde de la red, donde los dispositivos físicos se conectan por primera vez, y donde tienen origen los datos del IoT. • Su uso queda delimitado a un número menor de dispositivos. • Aloja los servicios de procesamiento, almacenamiento y comunicación en los dispositivos PACs.
VENTAJAS	<ul style="list-style-type: none"> • Simplifica la comunicación interna entre dispositivos, a través de la conexión física de los PACs, lo que permite aumentar la velocidad de recopilar y procesar los datos manejados. • Son los propios PACs quienes determinan que información debe almacenarse localmente o cual enviarse a la nube para su posterior análisis.
DESVENTAJAS	<ul style="list-style-type: none"> • Modelo menos escalable que Fog Computing (Reduce la capacidad de hacer pooling de recursos) • La conexión se realiza a través de redes propietarias con seguridad personalizada y poca interoperabilidad. • Trabaja de manera independiente, sin tener constancia de la existencia de una nube.

8 Conclusiones

A modo conclusión, y en base a la información dispuesta a lo largo de la presente memoria, el paradigma generado en torno al Cloud Computing podría definirse como un nuevo estilo de computación, basado en una metodología que permite a los usuarios acceder a diferentes recursos bajo demanda, como aplicaciones, datos o infraestructuras complejas, a través de la virtualización de servicios dispuestos por proveedores en la nube. Lo que conlleva una gran serie de beneficios con respecto a las arquitecturas más tradicionales que se venían utilizando hasta la fecha, en materia de disponibilidad, escalabilidad, flexibilidad, eficiencia y ahorro de costes entre otras muchas ventajas.

Bajo esta premisa, el concepto sobre el cual recae el peso innovador de esta metodología es el uso del modelo de servicios bajo demanda en la nube generado alrededor de la misma. Así, los diferentes recursos ofrecidos, son provistos desde la nube por un proveedor, permitiendo al usuario hacer uso de los mismos a través de las diferentes capas o modelos de servicios contratados, entre las más comunes: IaaS, PaaS y SaaS. Asimismo, el carácter presentado por la infraestructura como pública, privada o híbrida como combinación de los dos anteriores, otorga diferentes posibilidades de privacidad en cuanto al acceso a los recursos ofrecidos. Lo cual, genera una arquitectura de servicios que maximiza el alcance en las infraestructuras generadas por, y para el usuario.

Tal y como ha quedado evidenciado, son muchas las ventajas y beneficios que presenta este nuevo paradigma dentro del sector de las nuevas tecnologías. Sin embargo, también presenta grandes riesgos en materia de privacidad como principal inconveniente. En este sentido, la información procesada en la nube viaja a través de la red, desencadenando una serie de diversos conflictos aún no resueltos, que tienen que ver principalmente con la protección de la privacidad del usuario y la integridad de la información sensible manejada por los proveedores de servicio.

No obstante, y dados los cuantiosos beneficios que presenta la metodología en materia de eficiencia y optimización en torno a los procesos de computación, son muchas las compañías que han decidido sumarse al mercado como proveedores de servicio en la nube. Un mercado que actualmente no deja de crecer, generando grandes inversiones y nuevas corrientes emergentes que no cesan de evolucionar constantemente hacia la mejora y optimización de los servicios ofrecidos al cliente.

Ya para finalizar, tras la profundización realizada sobre el estudio que enmarca el desarrollo de la presente memoria, y vistos los apartados que conciernen a este nuevo paradigma, opino que el Cloud Computing es sin duda una gran revolución en cuanto a los niveles de computación que se venían utilizando hasta el momento. La capacidad de escalar un sistema en la nube, unido a la flexibilidad que aporta el modelo de servicios provistos en la red, sin duda, maximizará el alcance en los niveles de computación para los usuarios “de a pie”, lo que ineludiblemente repercutirá favorablemente en el crecimiento exponencial previsto para las tecnologías emergentes, las cuales, ya de por sí venían presentando un carácter evolutivo constante hacia la optimización y eficiencia. No obstante, y bajo mi criterio, pienso que aún quedan por resolver grandes conflictos en torno a la privacidad del usuario en la nube como medida prioritaria, lo que inherentemente, debería suponer una evolución global en el marco legal y judicial en torno a este nuevo paradigma que es el Cloud Computing.

Bibliografía

Dado el carácter bibliográfico presentado en su mayor parte por el presente trabajo, ha sido necesario llevar a cabo un profundo estudio en torno al análisis detallado de cada uno de los aspectos involucrados en la conceptualización del paradigma Cloud Computing, en cuanto a los distintos apartados y subapartados bajo los cuales se divide la presente memoria.

De este modo, a continuación, se presenta la bibliografía que ha permitido construir la base de conocimiento aplicada en la consecución del presente TFG, sobre la que, debido a su gran extensión, se ha decidido realizar una división en función del apartado bajo el cual aplica cada una de las fuentes utilizadas, para facilitar así el manejo, consulta y accesibilidad a cada uno de los recursos manipulados:

2. Paradigma del Cloud Computing

1&1 IONOS España S.L.U. (2021, 28 mayo). Cloud computing. IONOS Digital guide. <https://www.ionos.es/digitalguide/servidores/know-how/cloud-computing/> [Última Consulta: junio 2021]

Akamai. Arquitectura de cloud computing. <https://www.akamai.com/es/es/resources/cloud-computing-architecture.jsp> [Última Consulta: junio 2021]

Arévalo Navarro, J. M. (2011) Cloud Computing: fundamentos, diseño y arquitectura aplicados a un caso de estudio. Tesis fin de Máster, Universidad Rey Juan Carlos (Madrid) <https://burjcdigital.urjc.es/bitstream/handle/10115/5945/MemoriaTFMFINAL-2.pdf;jsessionid=B076112EFD941D2023B231C0A8FB65EB?sequence=1> [Última Consulta: junio 2021]

BeServices (2018, 27 junio). Tipos de Cloud Computing. <https://www.beservices.es/cloud-computing-tipos-n-5324-es> [Última Consulta: junio 2021]

Citelia (2020, 9 abril). Cloud computing: qué es y qué tipos existen. <https://citelia.es/blog/tipos-cloud-computing-computacion-nube/> [Última Consulta: junio 2021]

Computación en la nube. En Wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/Computaci%C3%B3n_en_la_nube [Última Consulta: junio 2021]

Conasa. (2019, 7 noviembre). Cloud Computing: El futuro de las empresas está en la nube. <https://conasa.grupocibernos.com/blog/cloud-computing-el-futuro-de-las-empresas-esta-en-la-nube> [Última Consulta: junio 2021]

Garzas, J. (2016, 15 enero). Cloud Computing: IaaS, PaaS y SaaS para muy Dummies. <https://www.javiergarzas.com/2016/01/que-es-cloud-computing-iaas-paas-y-saas-para-muy-dummies.html> [Última Consulta: junio 2021]

Lastras Hernansanz, J., Lázaro Requejo, J. & Mirón García, J.D. (2009) Arquitecturas de red para servicios en Cloud computing. Universidad Complutense de Madrid [https://eprints.ucm.es/id/eprint/9452/1/Arquitectura de red para servicios en Cloud Computing- Jorge Lastras Hernansanz, Javier L%C3%A1zaro Re.pdf](https://eprints.ucm.es/id/eprint/9452/1/Arquitectura_de_red_para_servicios_en_Cloud_Computing-_Jorge_Lastras_Hernansanz,_Javier_L%C3%A1zaro_Re.pdf) [Última Consulta: junio 2021]

Martin, D. (2020, 11 junio). Cloud computing: Uso y beneficios. Velneo. <https://velneo.es/cloud-computing-uso-beneficios/> [Última Consulta: junio 2021]

Modelos de servicio en la nube | Tipos de cloud computing | AWS. Amazon Web Services, Inc. https://aws.amazon.com/es/types-of-cloud-computing/?WICC=tile&tile=types_of_cloud [Última Consulta: junio 2021]

PowerData. Cloud: definiciones, servicios, despliegue, su seguridad y privacidad. <https://www.powerdata.es/cloud> [Última Consulta: junio 2021]

RedHat. ¿Qué es IaaS? <https://www.redhat.com/es/topics/cloud-computing/what-is-iaas> [Última Consulta: junio 2021]

Urueña, A. et al. (2012) Cloud Computing. Retos y Oportunidades – Resumen Ejecutivo. Ontis (Madrid) https://www.ontsi.red.es/sites/ontsi/files/2-_resumen_ejecutivo_cloud_computing_vf.pdf [Última Consulta: junio 2021]

Urueña, A. et al. (2012) Cloud Computing. Retos y Oportunidades. Ontis (Madrid) https://www.ontsi.red.es/sites/ontsi/files/1-_estudio_cloud_computing_retos_y_oportunidades_vdef.pdf [Última Consulta: junio 2021]

3. Principales Tecnologías sobre las cuales se asienta la metodología Cloud Computing

1&1 IONOS España S.L.U. (2016, 30 mayo). Balanceo de carga: conoce a fondo sus ventajas. IONOS Digitalguide. <https://www.ionos.es/digitalguide/servidores/know-how/balanceo-de-carga-conoce-a-fondo-sus-ventajas/> [Última Consulta: junio 2021]

Admin, A. (2009, 8 enero). ¿se puede virtualizar todo? Tipos de virtualización. La nube del Ciberespacio. <https://www.cibercloud.es/index.php/2009/01/08/se-puede-virtualizar-todo-tipos-de-virtualizacion/> [Última Consulta: junio 2021]

Araujo, Á. (2016, 19 abril). ¿Qué es una Base de Datos NoSQL? Oracle España. <https://blogs.oracle.com/spain/qu-es-una-base-de-datos-nosql> [Última Consulta: junio 2021]

Balance de carga. En Wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/Balance_de_carga [Última Consulta: junio 2021]

Bases de datos en AWS: la herramienta correcta para el trabajo adecuado. Amazon Web Services, Inc. <https://aws.amazon.com/es/nosql/> [Última Consulta: junio 2021]

De la Hoz, J. (2015). Big data y NoSQL: su rol en la revolución del cloud computing y sus retos hacia la estandarización. I+D Revista de Investigaciones, 6(2), 111-124.

<https://pdfs.semanticscholar.org/02df/c9121e4aacaf9e620496b9b981fb0f5f9b42.pdf>

[Última Consulta: junio 2021]

Enetic (2017, 21 noviembre). Virtualización vs cloud computing ¿En qué se diferencian?

<https://enetic.es/actualidad/virtualizacion-vs-cloud-computing-en-que-se-diferencian/>

Equilibrador de carga. En Wikipedia, la enciclopedia libre.

https://es.wikipedia.org/wiki/Equilibrador_de_carga [Última Consulta: junio 2021]

Flores, F. (2021, 22 marzo). Cloud Computing: Tipos de nubes, servicios y proveedores.

OpenWebinars.net. <https://openwebinars.net/blog/tipos-de-cloud-computing/> [Última

Consulta: junio 2021]

Fractal. (2018, 13 febrero). Cloud Computing - Nuevas tendencias paradigmáticas.

<https://www.fractal.com/es/blog/2018/02/13/cloud-computing-nuevas-tendencias-paradigmaticas>

[Última Consulta: junio 2021]

García Grijalva, L.A. (2013) VIRTUALIZACIÓN. Universidad Tecnológica de Puebla (México) TÓPICOS SELECTOS DE T.I.

https://docs.google.com/presentation/d/1PexF2TyvyZaSR-rBjJUWY6lbyBk6eZq3GXfXip_RX2c/htmlpresent?hl=es

[Última Consulta: junio 2021]

Hurwitz, J., Bloor, R., Kaufman, M., & Halper, F. (2009). Cloud Computing For Dummies (1.a ed.). IBM For Dummies. <https://www.ibm.com/cloud-computing/files/cloud-for-dummies.pdf>

[Última Consulta: junio 2021]

IONOS Ayuda. Vista general: Balanceador de Carga.

<https://www.ionos.es/ayuda/servidores-cloud/balanceador-de-carga/vista-general-balanceador-de-carga/>

[Última Consulta: junio 2021]

Martín, A. E. et al. (2013, 7 junio). Bases de datos NoSQL en cloud computing. SEDICI.

<http://sedici.unlp.edu.ar/handle/10915/27121> [Última Consulta: junio 2021]

NoSQL. En Wikipedia, la enciclopedia libre.

<https://es.wikipedia.org/wiki/NoSQL#Arquitectura> [Última Consulta: junio 2021]

Pérez, A. (2014, 26 enero). ‘Cloud computing’ y virtualización, ¿cuál es la diferencia?

TICbeat. <https://www.ticbeat.com/cloud/cloud-computing-virtualizacion-cual-es-la-diferencia/>

[Última Consulta: junio 2021]

RedHat. ¿En qué se diferencian cloud computing y virtualización?

<https://www.redhat.com/es/topics/cloud-computing/cloud-vs-virtualization>

[Última Consulta: junio 2021]

RedHat. VENTAJAS DE LA VIRTUALIZACIÓN | Virtualización

<https://www.redhat.com/es/topics/virtualization> [Última Consulta: junio 2021]

Tejada, Z. (2018, 12 febrero). Datos no relacionales y NoSQL - Azure Architecture Center. Microsoft Docs. <https://docs.microsoft.com/es-es/azure/architecture/data-guide/big-data/non-relational-data> [Última Consulta: junio 2021]

Wacker, H. (2000, 09 junio). ¿Qué es el balanceo de carga? IDG Communications S.A.U. <https://www.computerworld.es/tendencias/que-es-el-balanceo-de-carga> [Última Consulta: junio 2021]

4. Seguridad y Aspectos Legales

AGENCIA ESPAÑOLA DE PROTECCIÓN DE DATOS (2018) GUÍA para clientes que contraten servicios de Computing. AEPD. <https://www.aepd.es/es/documento/guia-cloud-clientes.pdf-0> [Última Consulta: junio 2021]

Axarnet (2019) Diferencias entre Virtualización y Cloud Computing. <https://axarnet.es/blog/virtualizacion-o-cloud-computing> [Última Consulta: junio 2021]

Durà, P. (2016, 28 abril). EN LA NUBE: ASPECTOS LEGALES A TENER EN CUENTA A LA HORA DE CONTRATAR LOS SERVICIOS DE CLOUD COMPUTING. Procoden. <https://www.procoden.es/blog/en-la-nube-aspectos-legales-a-tener-en-cuenta-a-la-hora-de-contratar-los-servicios-de-cloud-computing/> [Última Consulta: junio 2021]

GraphEverywhere. (2019, 29 junio). Bases de datos NoSQL vs SQL | Bases de datos no relacionales vs relacionales. <https://www.grapheverywhere.com/nosql-vs-sql/> [Última Consulta: junio 2021]

Hablando de ceros y unos. (2014, 13 mayo). Balanceo de Carga. <https://laredinfinita.wordpress.com/2014/05/11/balanceo-de-carga/> [Última Consulta: junio 2021]

Lopez, M. A. (2019, 5 marzo). Aspectos legales en Cloud Computing. MYL Abogados & Asesores. <https://mylabogados.es/aspectos-legales-en-cloud-computing/> [Última Consulta: junio 2021]

Riva, Á. (2020, 3 agosto). Aspectos legales a considerar al contratar servicios en la nube. PYMEactiva.es. <https://pymeactiva.es/aspectos-legales-a-considerar-al-contratar-servicios-en-la-nube/> [Última Consulta: junio 2021]

Tablado, F. (2020, 10 septiembre). Base de datos no relacional. ¿Qué es? Características y ejemplos. Ayuda Ley Protección Datos (LOPDGDD). <https://ayudaleyprotecciondatos.es/bases-de-datos/no-relacional/> [Última Consulta: junio 2021]

Treviño, G. J. A. (2011, 25 octubre). Aspectos Legales del Cloud Computing. <https://es.slideshare.net/joelgomezmx/aspectos-legales-del-cloud-computing-9882972> [Última Consulta: junio 2021]

5. Mercado Actual del Cloud Computing

Ayuda Ley Protección Datos (LOPDGDD). (2020, 28 octubre). Los mejores proveedores en Cloud Computing del mercado. <https://ayudaleyprotecciondatos.es/cloud-computing/proveedores/> [Última Consulta: junio 2021]

García, V. (2020, 14 julio). Estos han sido los principales avances del cloud computing. Revista Byte TI. <https://revistabyte.es/actualidad-it/cloud-computing/linke-cloud/> [Última Consulta: junio 2021]

García, V. (2020, noviembre 16). Estas son las tendencias cloud que marcarán el 2021. Revista Byte TI. <https://revistabyte.es/actualidad-it/tendencias-cloud-2021/> [Última Consulta: junio 2021]

IDG. (2020, agosto 6). 2020 Cloud Computing Study. <https://www.idg.com/tools-for-marketers/2020-cloud-computing-study/> [Última Consulta: junio 2021]

Jesuïtes Educació (2020, 01 noviembre). Los 5 mejores proveedores de Cloud Computing del 2020. <https://fp.uoc.fje.edu/blog/los-5-mejores-proveedores-en-cloud-computing/> [Última Consulta: junio 2021]

Jones, E. (2021, 29 abril). Cuota de mercado de la nube - una mirada al ecosistema de la nube en 2021. Kinsta. <https://kinsta.com/es/blog/cuota-de-mercado-de-la-nube/> [Última Consulta: junio 2021]

Ruiz, F. J. (2020b, junio 10). Publicado estudio sobre Cloud Computing en empresas 2020. Blog de Dataprius. <https://blog.dataprius.com/index.php/2020/06/10/publicado-estudio-sobre-cloud-computing-en-empresas-2020/> [Última Consulta: junio 2021]

STAMFORD, C. (2019, 2 abril). Gartner Forecasts Worldwide Public Cloud Revenue to Grow 17.5% in 2019. Gartner. <https://www.gartner.com/en/newsroom/press-releases/2019-04-02-gartner-forecasts-worldwide-public-cloud-revenue-to-g> [Última Consulta: junio 2021]

ThreePoints (2021, 08 junio) Cloud Computing: Principales proveedores y casos de éxito. <https://www.threepoints.com/int/cloud-computing-principales-proveedores-y-casos-de-exito> [Última Consulta: junio 2021]

6.Caso Práctico: Desarrollo y Despliegue de un Servicio Real en la Nube

¿Qué es Amazon API Gateway? | Amazon Web Services, Inc.
https://docs.aws.amazon.com/es_es/apigateway/latest/developerguide/welcome.html
[Última Consulta: junio 2021]

¿Qué es Amazon DynamoDB? | Amazon Web Services. Amazon Web Services, Inc.
https://docs.aws.amazon.com/es_es/amazondynamodb/latest/developerguide/Introduction.html [Última Consulta: junio 2021]

Alvarez, F., Gallego, C. & Evgeniev, M. (2019, 11 abril). Serverless. BBVA NOTICIAS.
<https://www.bbva.com/es/serverless/> [Última Consulta: junio 2021]

Amazon DynamoDB Servicio de base de datos NoSQL rápido y flexible para cualquier escala | Amazon Web Services. Amazon Web Services, Inc.
<https://aws.amazon.com/es/dynamodb/> [Última Consulta: junio 2021]

Amazon DynamoDB. En Wikipedia, la enciclopedia libre.
https://es.wikipedia.org/wiki/Amazon_DynamoDB [Última Consulta: junio 2021]

Amplify Framework Docs. (2020). Amplify Framework Documentation.
<https://docs.amplify.aws/> [Última Consulta: junio 2021]

Arquitecturas sin servidor. Amazon Web Services, Inc.
<https://aws.amazon.com/es/lambda/serverless-architectures-learn-more/> [Última Consulta: junio 2021]

AWS Identity and Access Management. Guía del usuario | Amazon Web Services. Amazon Web Services, Inc.
https://docs.aws.amazon.com/es_es/IAM/latest/UserGuide/iam-ug.pdf [Última Consulta: junio 2021]

AWS Lambda. Ejecute código sin tener que pensar en los servidores o los clústeres. Pague solo por lo que utiliza. | Amazon Web Services. Amazon Web Services, Inc.
<https://aws.amazon.com/es/lambda/> [Última Consulta: junio 2021]

AWS Lambda. En Wikipedia. https://en.wikipedia.org/wiki/AWS_Lambda [Última Consulta: junio 2021]

Becerro, R. G. (2021, 15 marzo). AWS Lambda, arquitectura serverless para implementar APIs. Paradigma. <https://www.paradigmadigital.com/dev/aws-lambda-arquitectura-serverless-implementar-apis/> [Última Consulta: junio 2021]

Benedet, M. (2019, 7 agosto). Arquitectura serverless: qué es y qué no es. #ADN CLOUD.
<https://blog.mdcloud.es/arquitectura-serverless/> [Última Consulta: junio 2021]

Características de Amazon API Gateway | Amazon Web Services, Inc.
<https://aws.amazon.com/es/api-gateway/features/> [Última Consulta: junio 2021]

Características de AWS Lambda | Amazon Web Services. Amazon Web Services, Inc.
<https://aws.amazon.com/es/lambda/features/> [Última Consulta: junio 2021]

Colina, D. (2021, 15 marzo). Profundizando en API Gateway de AWS. Paradigma.
<https://www.paradigmadigital.com/dev/profundizando-api-gateway-aws/> [Última Consulta: junio 2021]

Galván, P. Un Vistazo a la Arquitectura Serverless. <https://sg.com.mx/revista/52/unvistazo-la-arquitectura-serverless> [Última Consulta: junio 2021]

González Trejo, L. A. ¿Qué es la computación sin servidor (serverless)? | ASIMOV Ingeniería S. de R.L. de C.V. <https://asimov.mx/blog/el-poder-de-la-nube-aws-6/post/que-es-la-computacion-sin-servidor-serverless-20#wrap> [Última Consulta: junio 2021]

González, B. (2020, 16 julio) Desarrollo de aplicaciones con AWS Amplify. Tribalyte Technologies. <https://tech.tribalyte.eu/blog-apps-aws-amplify> [Última Consulta: junio 2021]

Gracia, L. (2016, 22 junio). ¿Qué es una Arquitectura Serverless (y AWS Lambda)? Un poco de Java y +. <https://unpocodejava.com/2016/06/22/que-es-una-arquitectura-serverless-y-aws-lambda/> [Última Consulta: junio 2021]

Identity and Access Management in Amazon DynamoDB | Amazon Web Services, Inc. https://docs.aws.amazon.com/es_es/amazondynamodb/latest/developerguide/authentication-and-access-control.html [Última Consulta: junio 2021]

InBestCloud. Amazon Dynamodb <https://www.inbest.cloud/amazon-dynamodb> [Última Consulta: junio 2021]

J. (2019, 25 noviembre). ¿Qué es una Arquitectura Serverless? Tú Programas. <https://tuprogramas.com/que-es-una-arquitectura-serverless/> [Última Consulta: junio 2021]

Preguntas frecuentes de AWS Amplify | Servicios de front-end web y móviles | Amazon Web Services, Inc. <https://aws.amazon.com/es/amplify/faqs/?nc=sn&loc=5> [Última Consulta: junio 2021]

Preguntas frecuentes sobre Amazon DynamoDB | Base de datos de clave-valor NoSQL | Amazon Web Services. Amazon Web Services, Inc. <https://aws.amazon.com/es/dynamodb/faqs/> [Última Consulta: junio 2021]

7.Principales Corrientes Emergentes Derivadas del Cloud Computing: Edge Computing y Fog Computing

A. (2020, marzo 12). ¿Qué es Fog Computing y en qué se diferencia del Edge Computing? Blog de arsys.es. <https://www.arsys.es/blog/fogcomputing/> [Última Consulta: junio 2021]

ARSYS. (2021) Principales tendencias Cloud para 2021. <https://www.arsys.es/blog/file/uploads/2020/12/tendencias-cloud-2021-1.pdf> [Última Consulta: junio 2021]

Coremain. (2018, 30 julio). Cloud Computing, Edge Computing, Fog Computing ¿En qué se diferencian? <https://www.coremain.com/diferencias-cloud-computing-edge-computing-fog-computing/> [Última Consulta: junio 2021]

Gradiant. (2018, 4 mayo). Edge/Fog Computing: del Cloud hacia la computación en los dispositivos. <https://www.gradiant.org/blog/edge-fog-computing-cloud/> [Última Consulta: junio 2021]

Interempresas. Cloud computing, cuna de los desarrollos tecnológicos de la nueva era. <https://www.interempresas.net/TIC/Articulos/196938-Cloud-computing-cuna-de-los-desarrollos-tecnologicos-de-la-nueva-era.html> [Última Consulta: junio 2021]

IT Digital Media Group. (2020, 11 noviembre). Así evolucionará cloud computing en 2021. Cloud | IT User. <https://www.ituser.es/cloud/2020/11/asi-evolucionara-cloud-computing-en-2021> [Última Consulta: junio 2021]

Miller, V. (2019, 4 noviembre). Cloud, Fog, and Edge Computing: 3 Differences That Matter. Dzone.Com. <https://dzone.com/articles/cloud-vs-fog-vs-edge-computing-3-differences-that> [Última Consulta: junio 2021]

Mistretta, M. (2021, 14 enero). Edge es el nuevo cloud y otras siete tendencias en cloud computing para 2021. IT Masters Mag. <https://www.itmastersmag.com/noticias-analisis/edge-es-el-nuevo-cloud-y-otras-siete-tendencias-en-cloud-computing-para-2021/> [Última Consulta: junio 2021]

Pastor, J. (2018, 19 julio). Edge Computing: qué es y por qué hay gente que piensa que es el futuro. Xataka. <https://www.xataka.com/internet-of-things/edge-computing-que-es-y-por-que-hay-gente-que-piensa-que-es-el-futuro> [Última Consulta: junio 2021]

Anexos

A Balanceadores de Carga

Un balanceador de carga puede implementarse tanto como un dispositivo software, como es el caso de los manejadores internos instalados en la parte backend de los servidores, como un dispositivo hardware, tales como routers o switches que se encargan de gestionar y enrutar las peticiones salientes y entrantes del par usuario-servidor. En ambos casos, la estrategia a seguir a la hora de gestionar las distintas peticiones queda definida por una serie de reglas o algoritmos predeterminados previamente por el usuario, en base a los cuales, y atendiendo a la carga de trabajo presente en un momento determinado del tiempo, el Servidor es capaz de gestionar de manera óptima las múltiples peticiones existentes.

Principales Métodos de Balanceo de Carga

Descrita brevemente la tecnología implementada detrás de los balanceadores de carga, cabe incidir en que estos se pueden clasificar principalmente en tres grupos atendiendo a la naturaleza de las prioridades establecidas: de petición, basado en sesión, o de métodos; y estos iterativamente, se dividen de acuerdo a los algoritmos estratégicos de reparto aplicados. Se detallan a continuación, los métodos de balanceo de carga más representativos:

- **Round-Robin:** En primer lugar y conforme se puede observar en la siguiente figura adjunta Figura Anexo A - 1, este algoritmo se encarga de distribuir las peticiones entrantes una a una de una forma equitativa entre los diferentes clústers o servidores, atendiendo a una estrategia de carácter cíclico, es decir, comienza en la posición 1 hasta N, donde $N+1 = 1$.

El principal problema que presenta este método radica en que las peticiones son distribuidas de forma equitativa, sin embargo, el algoritmo no tiene en cuenta la carga asociada a cada petición, por lo que, cabe la posibilidad de generar una descompensación a la hora de procesar las peticiones.

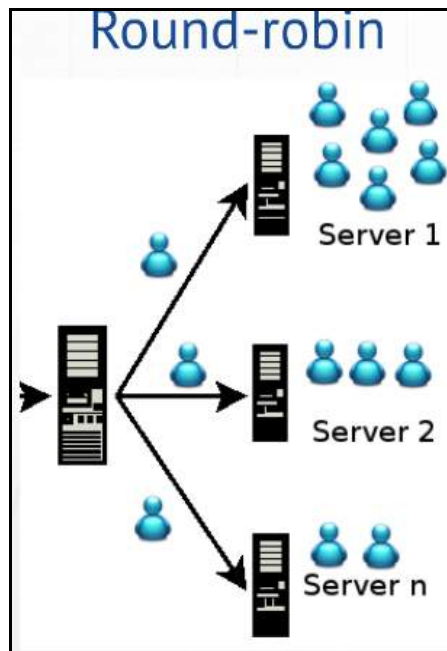


Figura Anexo A - 1: Representación Gráfica de la Lógica Funcional asociada al Método de Balanceo - Round Robin¹⁰

- Weighted Round-Robin:** Este algoritmo resuelve la problemática existente en el método Round-Robin anterior. Basándose en la estructura cíclica como base de actuación, Weighted Round-Robin define un peso ponderado para cada uno de los servidores en relación al nivel de carga y poder de procesamiento presentado, a continuación, en base a estos pesos calculados, distribuye las peticiones de una manera ponderada, repartiendo la carga de trabajo de una manera proporcional en función del peso.

En la siguiente Figura Anexo A - 2, se puede apreciar cómo, el servidor 1 puede manejar 4 veces la carga que soporta el servidor 2, por lo que el balanceador de carga en este punto, redirigirá un mayor número de peticiones a dicho servidor.

¹⁰ Extraído de: *Equilibrador de carga*. En Wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/Equilibrador_de_carga

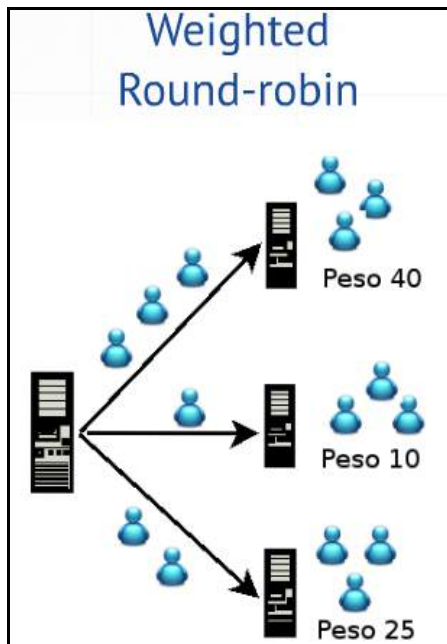


Figura Anexo A - 2: Representación Gráfica de la Lógica Funcional asociada al Método de Balanceo - Weighted Round Robin¹¹

- **Least Connection:** Tomando como base de ejemplo la Figura Anexo A - 3, el reparto de las peticiones entrantes se redistribuye tomando como base el número de conexiones activas presentes en cada servidor en el momento de la llegada de la petición. De este modo, sin tener en cuenta la carga de procesamiento asociada, el balanceador de carga asignará la petición entrante a aquel servidor que presente un menor número de conexiones concurrentes.

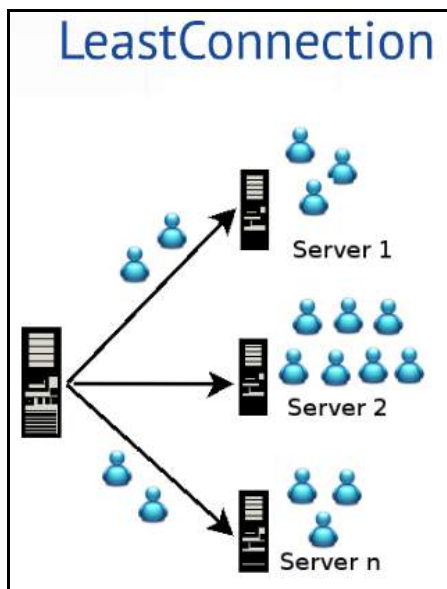


Figura Anexo A - 3: Representación Gráfica de la Lógica Funcional asociada al Método de Balanceo - Least Connection¹¹

¹¹ Extraído de: *Equilibrador de carga. En Wikipedia, la enciclopedia libre.*
https://es.wikipedia.org/wiki/Equilibrador_de_carga

- **Weighted Least Connection:** Basándose en el sistema de balanceo anterior, Weighted Least Connection distribuye las peticiones entrantes dependiendo del número de conexiones activas existentes en cada servidor en ese momento, dicho comportamiento queda ilustrado en la Figura Anexo A - 4. No obstante, la principal diferencia de este método radica en la ponderación de las cargas manejadas por cada servidor a la hora de realizar el redireccionamiento de la petición entrante, por lo que, el balanceo en este punto, se realiza de una manera más óptima que el anterior.

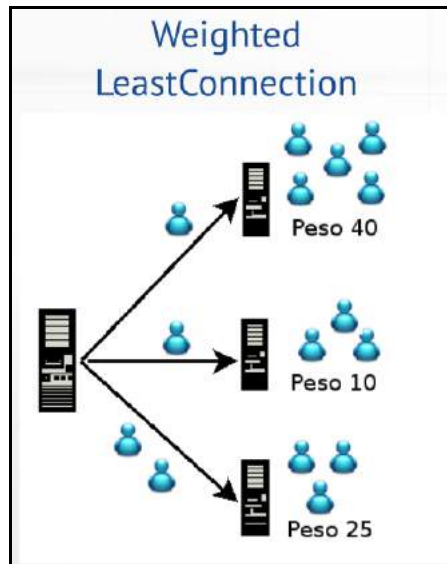


Figura Anexo A - 4: Representación Gráfica de la Lógica Funcional asociada al Método de Balanceo - Weighted Least Connection¹²

- **Random:** Este algoritmo se basa en un método de reparto aleatorio, en el cual, tras la llegada de una petición entrante, esta se reparte de una forma aleatoria entre uno de los servidores activos en ese momento determinado escogido al azar. No obstante, su utilidad como método balanceador es cuestionable, dado que incluso existe la posibilidad de fallar en el intento de asignar una petición de carga cuando alguno de los servidores se encuentre caído.
- **Fastest:** La petición entrante se asigna al servidor en cuestión que menor tiempo de respuesta ofrezca en ese momento. Esto es, al más rápido, tal y como indica el propio nombre del método.
- **Observed:** El método Observed plantea un algoritmo combinado fruto de la unión de la lógica implantada en los métodos anteriores, Least Connections y Fastest para realizar el equilibrado de carga. En este contexto, se genera un listado a modo ranking de acuerdo a los criterios preestablecidos de: número actual de conexiones y tiempo de respuesta del servidor. A continuación, el servidor que encabece el listado y, por tanto, tenga un menor número de conexiones presentes y un tiempo de respuesta menor (más rápido) recibirá la petición entrante en cuestión.

¹² Extraído de: *Equilibrador de carga*. En Wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/Equilibrador_de_carga

- **Predictive:** El algoritmo Predictive es considerado como una versión mejorada del método Observed anteriormente descrito, dado que, basándose en la misma lógica del uso de rankings, el presente método realiza un análisis del seguimiento y tendencia de cada uno de los servidores identificados, determinando qué servidores pueden mejorar o empeorar su desempeño en un corto periodo de tiempo, es decir, aquellos servidores que tengan una tendencia a la mejora, recibirán mayor cantidad de conexiones. Al ser este un método dinámico, con un alcance predictivo operacional, se presenta como una solución óptima en cualquier contexto habilitado de equilibrado de carga.
- **Ip-Hash:** La selección del Servidor que atenderá una petición X de un usuario Y, se realiza en base a algún parámetro fijo relacionado con Y, como, por ejemplo, la dirección IP del usuario. De este modo, todas las peticiones de un mismo usuario son atendidas siempre por un mismo servidor. Su lógica interna, tal y como su nombre indica, está basada en las tablas Hash de funcionamiento.

Tal y como ha quedado introducido anteriormente, estas son algunas de las principales estrategias de distribución de recursos en torno a las cuales se implementan las estrategias que definen la lógica subrogada en los balanceadores o equilibradores de carga, sin embargo, cada contexto específico requerirá del aplicativo de una estrategia concreta, que implemente la distribución de recursos de una manera eficiente en cuanto al resto del sistema global acontecido.

B Virtualización

De manera genérica, la virtualización se puede definir como una tecnología capaz de gestionar y administrar de manera concurrente diferentes entornos y recursos dedicados (tanto software, como hardware), desde una sola unidad física operativa o CPU, abstrayendo y aislando entre si cada uno de ellos, de manera que el conjunto de los mismos sea ejecutado en paralelo de una forma opaca y totalmente independiente.

Durante este proceso, el software de virtualización se ejecuta por debajo de la capa hardware origen o CPU, dando como resultado una división del Sistema en diferentes entornos autónomos e independientes. Cada entorno generado es definido comúnmente bajo las siglas VM (Virtual Machine), y depende directamente de los límites definidos dentro del Sistema origen CPU, así como de la capacidad de gestión del software de virtualización, el cual, se hace cargo de repartir y gestionar paralelamente los diferentes recursos requeridos por cada entorno en base a sus necesidades operativas específicas.

Gracias a ello, la virtualización permite al usuario llevar a cabo diferentes procedimientos que, mediante la operativa y gestión de una única unidad física o CPU, serían inconcebibles a nivel técnico, como por ejemplo, la configuración de entornos y migraciones específicas en los procesos de validación de nuevo hardware, o la creación de diferentes bancos de pruebas en los procesos de testeo regresivo, sin la necesidad añadida de utilizar un hardware específico para cada una de las pruebas o servidores de desarrollo.

En materia de seguridad e integridad de la información y los datos manejados, la virtualización ha de integrarse de tal forma que los entornos generados repliquen las medidas preventivas necesarias en la mitigación de ataques maliciosos de una forma común. En lo que se refiere a un entorno de desarrollo, cada usuario posee su propio espacio aislado, lo que evita que el código malicioso afecte de manera global al resto de entornos concurrentes que estén siendo simulados en la misma unidad física de origen.

Principales Métodos de Virtualización

Respecto a los tipos de virtualización, existen múltiples enfoques y métodos asociados a las diferentes características y recursos simulados en cada situación concreta, lo cual, plantea un gran abanico de posibilidades en relación a la combinatoria obligada de software y hardware. No obstante, a continuación, se presentan algunos de los principales tipos de virtualización desde un punto generalizado de uso:

- **Emulación:** El proceso de simulación se realiza para un hardware completo, admitiendo en este punto un nuevo SO totalmente virtualizado, el cual, dote al usuario de la experiencia de obtener una CPU completamente diferente. Este método está orientado a la creación y desarrollo de software en máquinas que no se encuentren físicamente disponibles.
- **Virtualización Nativa/Completa:** La VM simula las distintas características hardware requeridas por un SO, de tal forma que se pueda ejecutar un nuevo SO de forma aislada, sin modificar los parámetros y configuraciones de la CPU original.

- **Virtualización Parcial o Address Space Virtualization:** En este punto, la VM simula instancias concretas del hardware que se desea virtualizar, concretamente, las llamadas Address Spaces o espacios de direccionamiento. Este método dota al usuario de la capacidad de compartir recursos y aislar procesos dentro de un marco virtualizado sin modificar las configuraciones del SO original.
- **Virtualización de SO:** Este método permite virtualizar de manera completa un SO bajo un mismo servidor, permitiendo así crear múltiples servidores virtuales aislados y ejecutarlos de manera concurrente dentro de una misma CPU física. Las aplicaciones que corren dentro de cada uno de los diferentes SO virtuales disponen de independencia y opacidad entre sí, conformando para ellas un sistema autónomo en sí mismo por cada servidor generado.
- **Virtualización de Aplicaciones:** La virtualización de aplicaciones está basada en la ejecución de un software o programa en un entorno local generado dentro de la misma CPU, con recursos locales a su vez. Mediante esta operativa, el usuario posee la capacidad de generar un pequeño entorno virtual con los componentes mínimos requeridos por las rutinas de la aplicación simulada, tales como entradas a registros, archivos u objetos globales de referencia. La principal ventaja que aporta este método es la eliminación de los posibles conflictos existentes entre las aplicaciones a ejecutar, y aplicaciones nativas del sistema físico original.

C Caso Práctico: Desarrollo y Despliegue de un Servicio Real en la Nube – ANEXO Evidencias de Implementación

Como parte del presente Anexo, se adjuntan las diferentes interfaces y fragmentos de código correspondientes al proceso de desarrollo y despliegue del caso práctico expuesto en el capítulo “6.Caso Práctico: Desarrollo y Despliegue de un Servicio Real en la Nube”, de la presente memoria. De este modo, a continuación, se muestra paso a paso la implementación llevada a cabo en cada uno de los apartados referidos, como forma de ilustrar y evidenciar la ejecución del caso práctico mencionado:

6.2 Implementación de un Recurso Web Estático

```
index.html
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="UTF-8">
5      <title>TFG_2021_SSP</title>
6  </head>
7
8  <body>
9      6.- Caso Práctico: Desarrollo y Despliegue de un Servicio Real en la Nube
10 </body>
11 </html>
```

Figura Anexo C - 5: Implementación de un Recurso Web Estático – Fragmento de Código Asociado a la Codificación del Recurso Web Estático

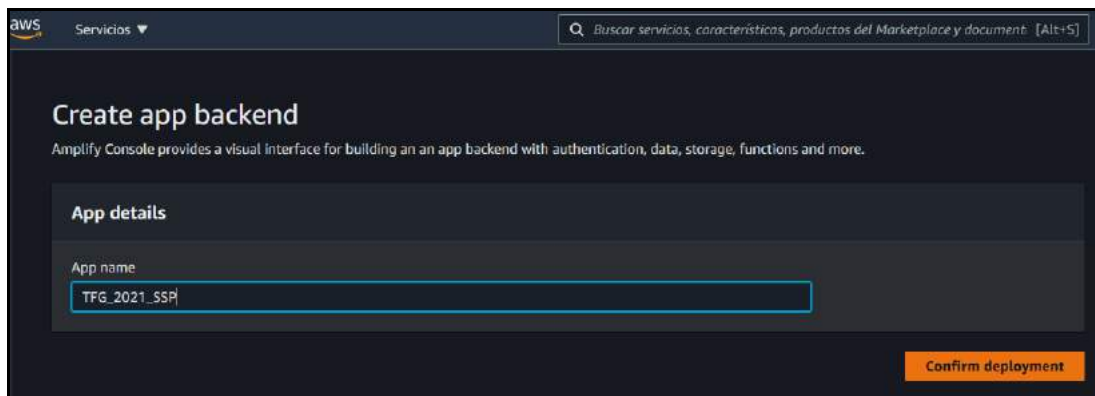


Figura Anexo C - 6: Implementación de un Recurso Web Estático (II) – Creación del Backend para el Despliegue del Recurso Web Estático

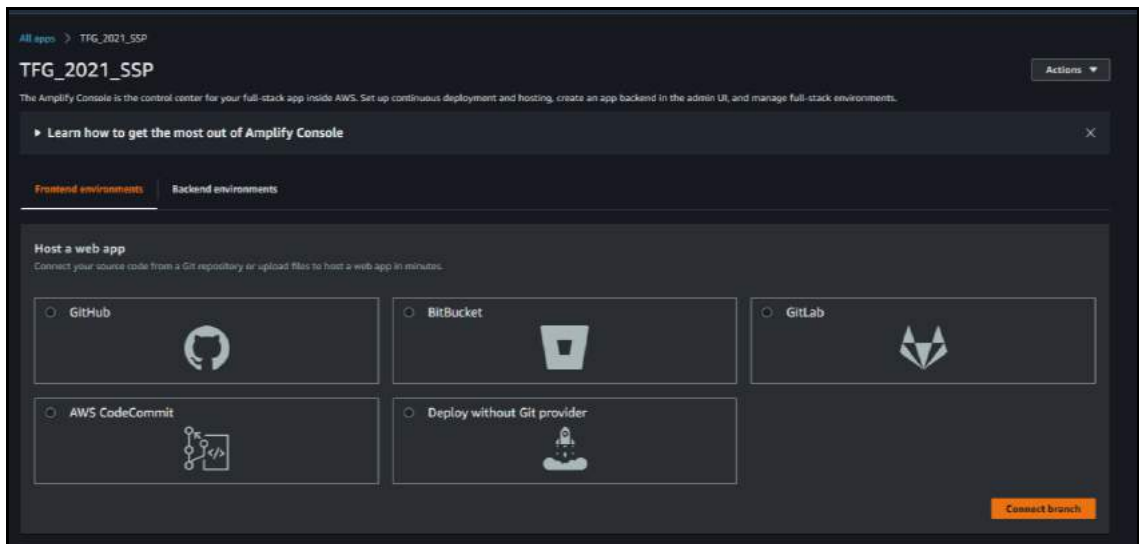


Figura Anexo C - 7: Implementación de un Recurso Web Estático (III) – Carga y Despliegue del Recurso Web en AWS Amplify

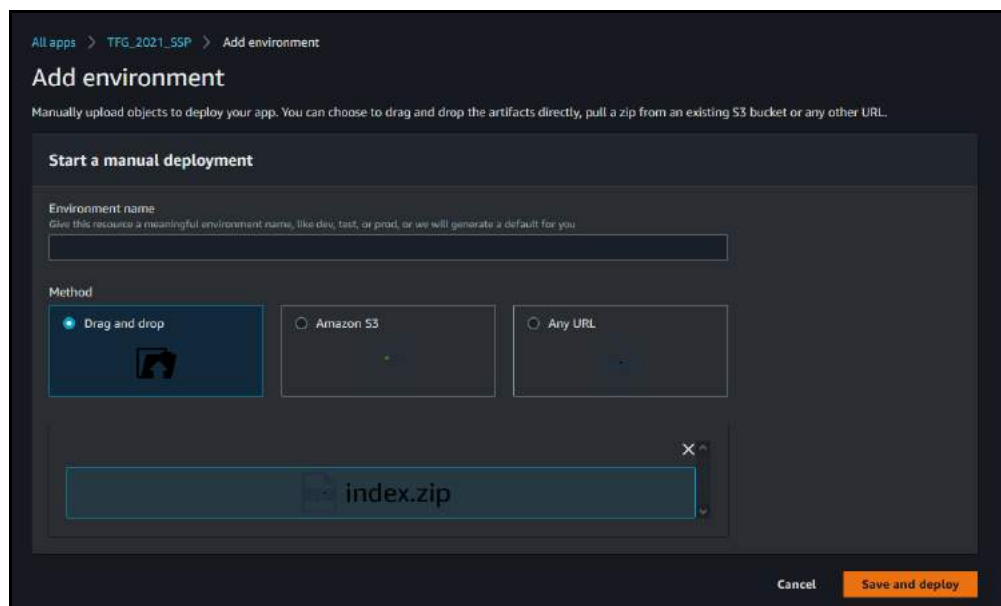


Figura Anexo C - 8: Implementación de un Recurso Web Estático (IV) – Carga y Despliegue del Recurso Web en AWS Amplify [2]

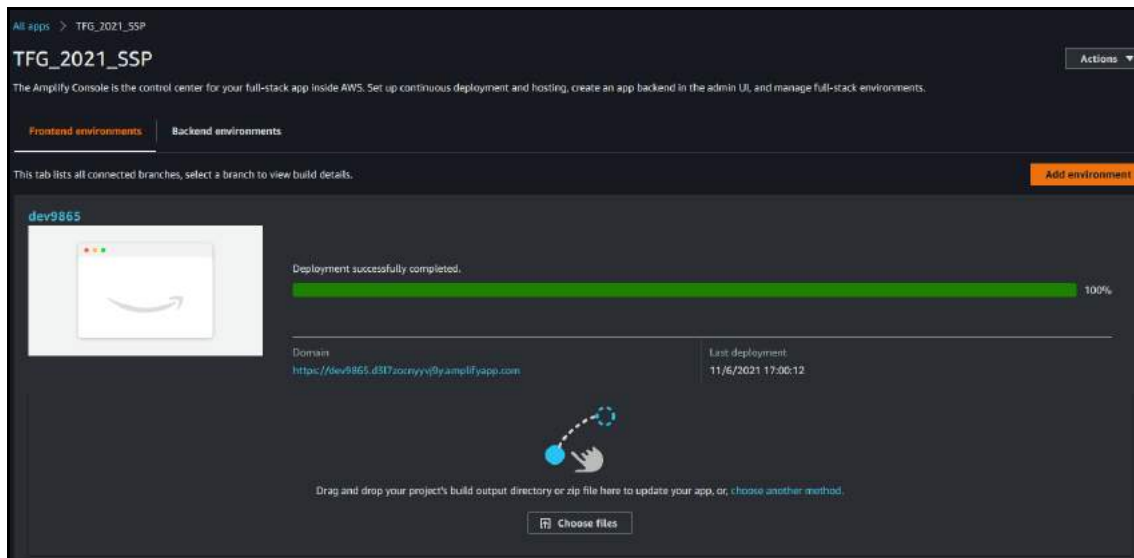


Figura Anexo C - 9: Implementación de un Recurso Web Estático (V) – Carga y Despliegue del Recurso Web en AWS Amplify [3]

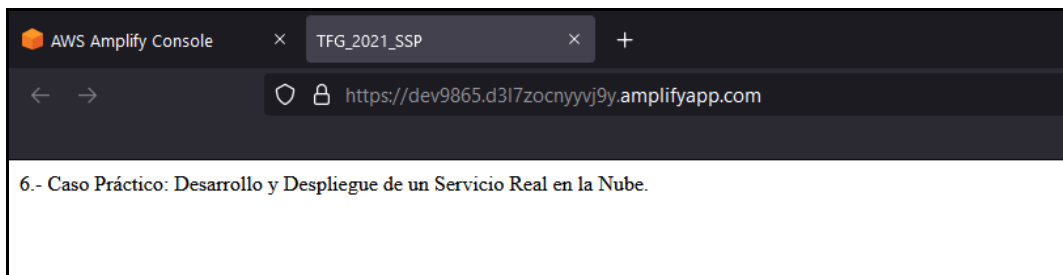


Figura Anexo C - 10: Implementación de un Recurso Web Estático (VI) –Despliegue como Aplicación del Recurso Web Implementado mediante AWS Amplify

6.3 Desarrollo de una Función Lambda

Lambda > Funciones > Crear una función

Crear una función [Info](#)

Seleccione una de las siguientes opciones para crear la función.

- Crear desde cero** Empezar con un sencillo ejemplo "Hello World".
- Utilizar un proyecto** Cree una aplicación Lambda utilizando un código de muestra y los ajustes de configuración predefinidos de casos de uso comunes.
- Imagen del contenedor** Seleccione una imagen de contenedor para implementar para la función.

Información básica

Nombre de la función
Escriba un nombre para describir el propósito de la función.

Utilice exclusivamente letras, números, guiones o guiones bajos. No incluya espacios.

Tiempo de ejecución [Info](#)
Elija el lenguaje que desea utilizar para escribir la función. Tenga en cuenta que el editor de código de la consola solo admite Node.js, Python y Ruby.

Permisos [Info](#)
De forma predeterminada, Lambda creará un rol de ejecución con permisos para cargar registros en Amazon CloudWatch Logs. Puede personalizar este rol predeterminado más adelante al agregar los disparadores.

- ▶ **Cambiar el rol de ejecución predeterminado**
- ▶ **Configuración avanzada**

Figura Anexo C - 11: Desarrollo de una función Lambda (I)- Panel de Creación de una Función Lambda

```
Tools Window Test | Deploy Changes not deployed  
lambda_funcionior x (+)  
1 # Importación de las librerías correspondientes al manejo de objetos JSON  
2 import json  
3  
4 # Definimos la función que tomará como valor de entrada  
5 def lambda_handler(event, context):  
6 # Extraemos los valores del objeto del servicio Lambda  
7     name = event['firstName'] + ' ' + event['lastName']  
8 # return del objeto creado en base a los valores de entrada  
9     return {  
10         'statusCode': 200,  
11         'body': json.dumps('Hello from Lambda, ' + name)  
12     }  
13
```

Figura Anexo C - 12: Desarrollo de una función Lambda (II)- Fragmento de Código Asociado a la Codificación de la Función Lambda



Figura Anexo C - 13: Desarrollo de una función Lambda (III)- Configuración del Evento de Prueba de la Función Lambda

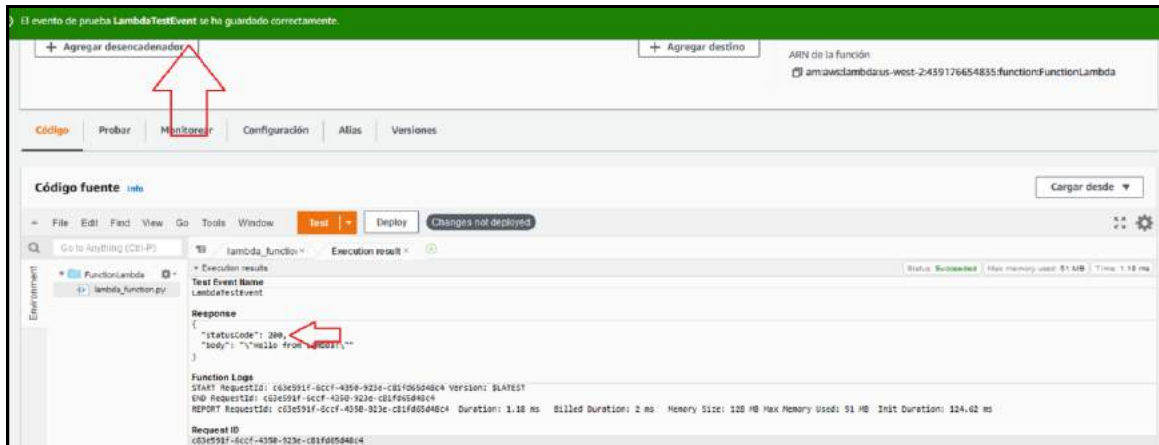


Figura Anexo C - 14: Desarrollo de una función Lambda (IV)- Resultado de la Ejecución del Caso de Prueba

6.4 Integración de la Aplicación Web y la Función Lambda

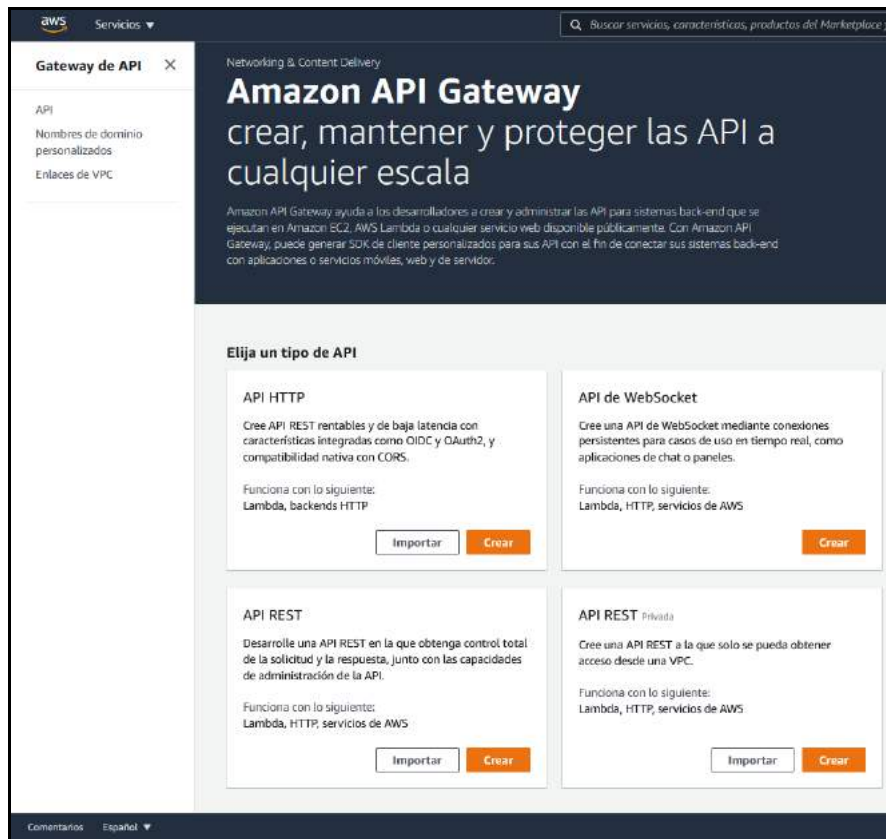


Figura Anexo C - 15: Integración de la Aplicación Web y la Función Lambda - Interfaz Principal del Menú del Servicio Amazon API Gateway

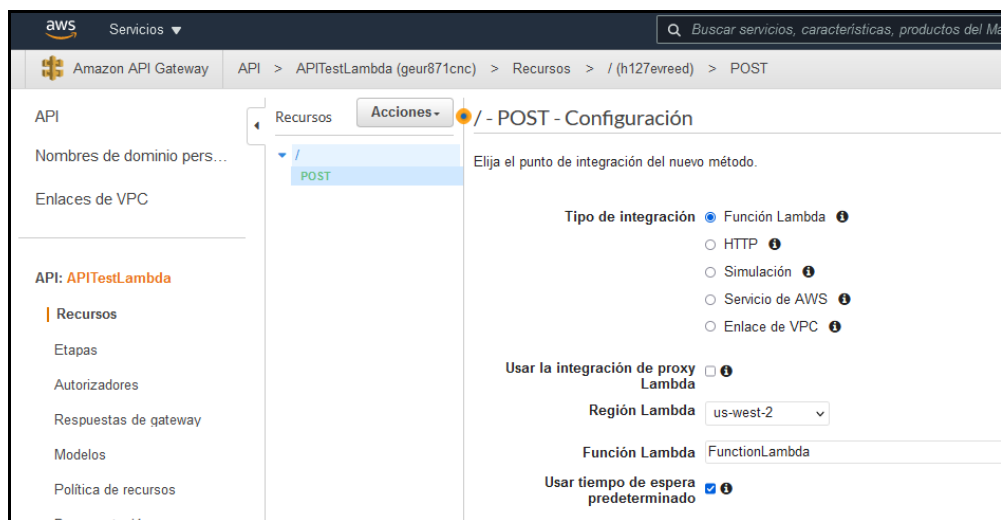


Figura Anexo C - 16: Integración de la Aplicación Web y la Función Lambda (II) – Configuración e Integración del Método POTS de la API con la Función Lambda

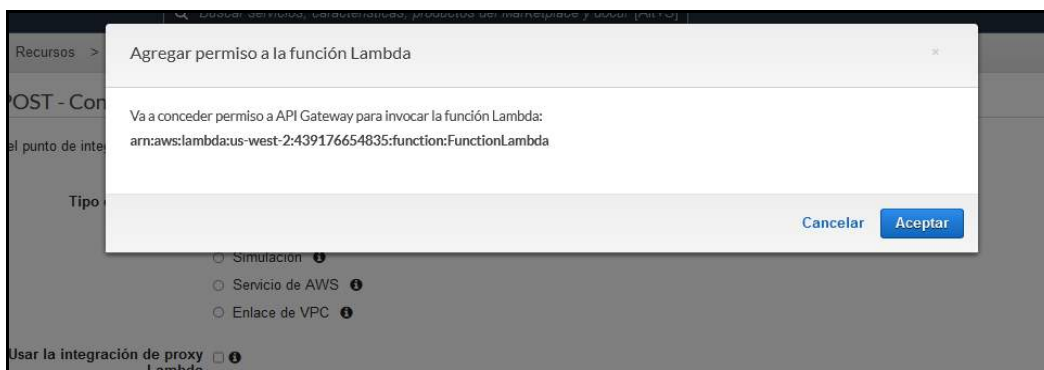


Figura Anexo C - 17: Integración de la Aplicación Web y la Función Lambda (II) – Configuración e Integración del Método POTS de la API con la Función Lambda [2]



Figura Anexo C - 18: Integración de la Aplicación Web y la Función Lambda (II) – Configuración e Integración del Método POTS de la API con la Función Lambda [3]

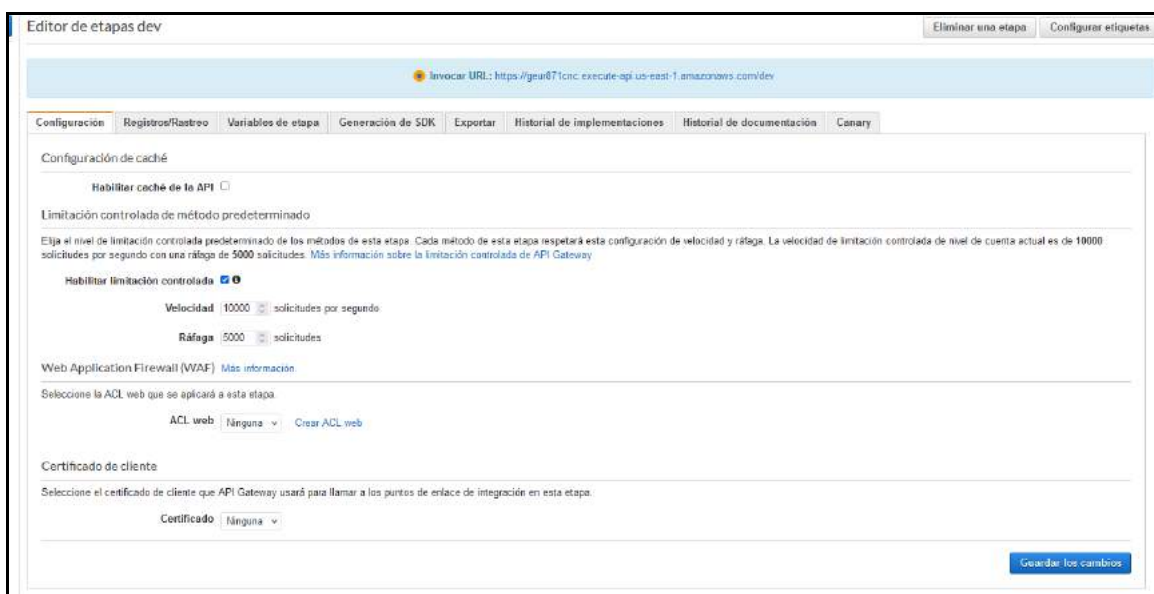


Figura Anexo C - 19: Integración de la Aplicación Web y la Función Lambda (V) – Panel de Configuración de la API

ejecución de método / - POST - Prueba de método

Realice una llamada de prueba al método. Cuando realiza una llamada de prueba, API Gateway omite la autorización e invoca directamente al método.

Ruta
No hay ningún parámetro de ruta para este recurso. Puede definir los parámetros de ruta mediante la sintaxis {rePathParam} en una ruta de recurso.

Cadenas de consulta
No hay ningún parámetro de cadena de consulta para este método. Puede agregarlos a través de la solicitud de método.

Encabezados
No hay ningún parámetro de encabezado para este método. Puede agregarlos a través de la solicitud de método.

Variables de etapa
No hay ninguna **Variable de etapa** para este método.

Cuerpo de la solicitud

```

{
  "method": "POST",
  "headers": {}
}

```

Solicitud /
Estado: 200
Latencia: 549 ms
Cuerpo de respuesta

```

{
  "statusCode": 200,
  "body": "{\"Hello from Lambda!\""}
}

```

Encabezados de respuesta

```

{"Access-Control-Allow-Origin":"*", "X-Base-Trace":"2d","Access-Control-Allow-Headers":"x-amz-date","Content-Type":"application/json"}

```

Registros

```

Execution log for request 18ca966d-92da-4881-a220-2a1347622940
Fri Jun 11 15:24:30 UTC 2021 : Starting execution for request: 18ca966d-92da-4881-a220-2a1347622940
Fri Jun 11 15:24:30 UTC 2021 : HTTP Method: POST, Resource Path: /
Fri Jun 11 15:24:30 UTC 2021 : Method request path: {}
Fri Jun 11 15:24:30 UTC 2021 : Method request query string: {}
Fri Jun 11 15:24:30 UTC 2021 : Method request headers: {}
Fri Jun 11 15:24:30 UTC 2021 : Method request body before transformations:
{
  "method": "POST",
  "headers": {}
}

```

Figura Anexo C - 20: Integración de la Aplicación Web y la Función Lambda (VI) – Ejecución de un Caso de Prueba sobre la API Creada

6.5 Creación de la Base de Datos NoSQL

TestLambdaDataBase [Cerrar](#)

Información general | Elementos | Métricas | Alarmas | Capacidad | Índices | Tablas globales | Copias de seguridad | Contributor Insights | Desencadenadores | Cont

Alertas recientes

No se han activado alarmas de CloudWatch para esta tabla.

Detalles de Kinesis Data Stream

Utilice Amazon Kinesis Data Streams para DynamoDB para capturar los cambios a nivel de elemento de la tabla como un flujo de datos de Kinesis. [Más información](#)

Flujo habilitado No
Nombre del flujo -

[Administrar el streaming a Kinesis](#)

Detalles del flujo de DynamoDB

Flujo habilitado No
Ver tipo -
ARN del flujo más reciente -

[Administrar el flujo de DynamoDB](#)

Detalles de la tabla

Nombre de la tabla	TestLambdaDataBase
Clave de partición principal	ID (Cadena)
Clave de ordenación principal	-
Recuperación a un momento dado	DESABILITADO Habilitar
Tipo de cifrado	PREDETERMINADO Administrar el cifrado
ARN de clave principal de KMS	No aplicable
Estado de cifrado	-
CloudWatch Contributor Insights	DESABILITADO Administrar Contributor Insights NUEVO
Atributo de tiempo de vida	DESABILITADO Administrar TTL
Estado de la tabla	Activo
Fecha de creación	11 de junio de 2021, 17:26:55 UTC+2
Modo de capacidad de lectura/escritura	Aprovisionado
Último cambio al modo bajo demanda	-
Unidades de capacidad de lectura aprovisionada	5 (Auto Scaling Deshabilitados)
Unidades de capacidad de escritura aprovisionada	5 (Auto Scaling Deshabilitados)
Tiempo de disminución más reciente	-
Tiempo de aumento más reciente	-
Tamaño de almacenamiento (en bytes)	0 bytes
Recuento de elementos	0 Administrar recuento dinámico
Región	US West (Oregon)
Nombre de recurso de Amazon (ARN)	am:aws:dynamodb:us-west-2:439176654835:table/TestLambdaDataBase

El tamaño de almacenamiento y el recuento de elementos no se actualizan en tiempo real. Se actualizan de forma periódica, alrededor de cada seis horas.

Figura Anexo C - 21: Creación de la Base de Datos NoSQL – Panel Principal de Gestión de la BBDD NoSQL

aws Servicios [Cerrar](#)

Buscar servicios, características, productos del Marketplace y docu. [Alt+S]

Crear tabla | Eliminar tabla

Filtrar por nombre de tabla

Elija un grupo ... Acciones

Nombre

TestLambdaDataBase

TestLambdaDataBase [Cerrar](#)

Información general | **Elementos** | Métricas | Alarmas | Capacidad | Índices | Tablas globales | Copias de seguridad

Crear elemento | Acciones

Examen: [Tabla] TestLambdaDataBase: ID

Examen [Tabla] TestLambdaDataBase: ID

Añadir filtro

Iniciar búsqueda

ID	LatestGreetingTime
Sergio Sanchez	Fri, 11 Jun 2021 15:45:06 +0000

Figura Anexo C - 22: Creación de la Base de Datos NoSQL (II) – Detalle de la Consulta de Valores/Elementos de la BBDD NoSQL

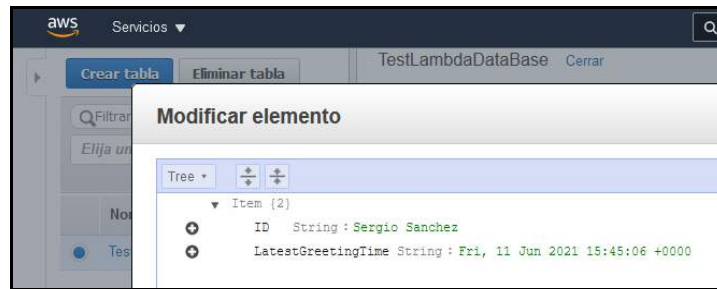


Figura Anexo C - 23: Creación de la Base de Datos NoSQL (III) – Detalle de la Consulta de Valores/Elementos de la BBDD NoSQL [2]



Figura Anexo C - 24: Creación de la Base de Datos NoSQL (IV) – Panel de Gestión de Roles y Permisos de la BBDD

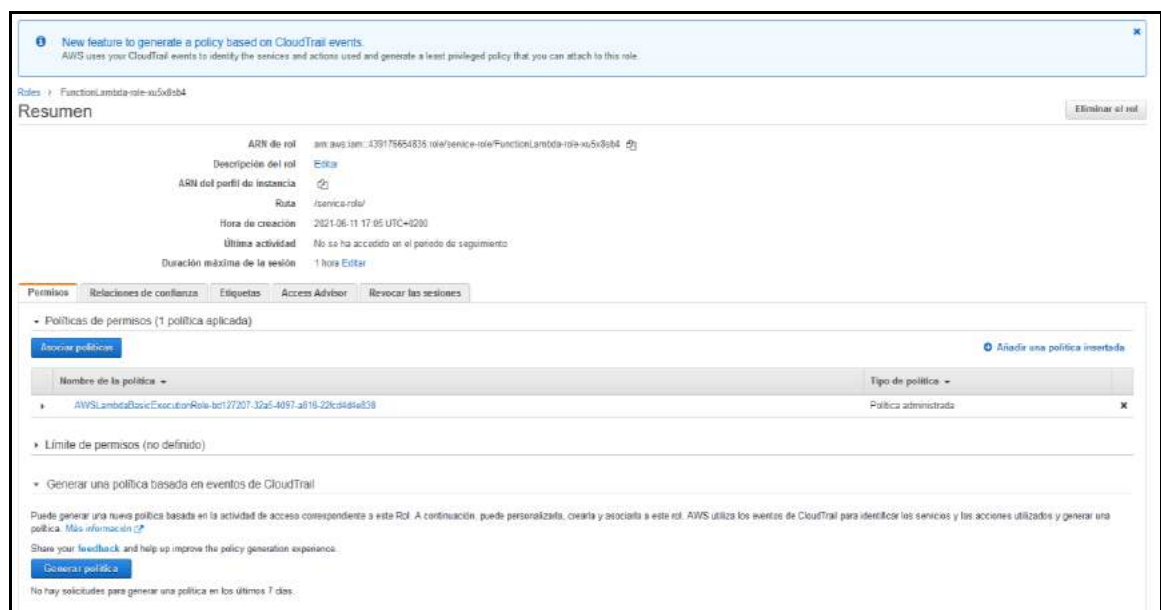


Figura Anexo C - 25: Creación de la Base de Datos NoSQL (V) – Panel de Gestión de Roles y Permisos de la BBDD [2]



Figura Anexo C - 26: Creación de la Base de Datos NoSQL (VI) – Registro de Política de Acceso a la BBDD

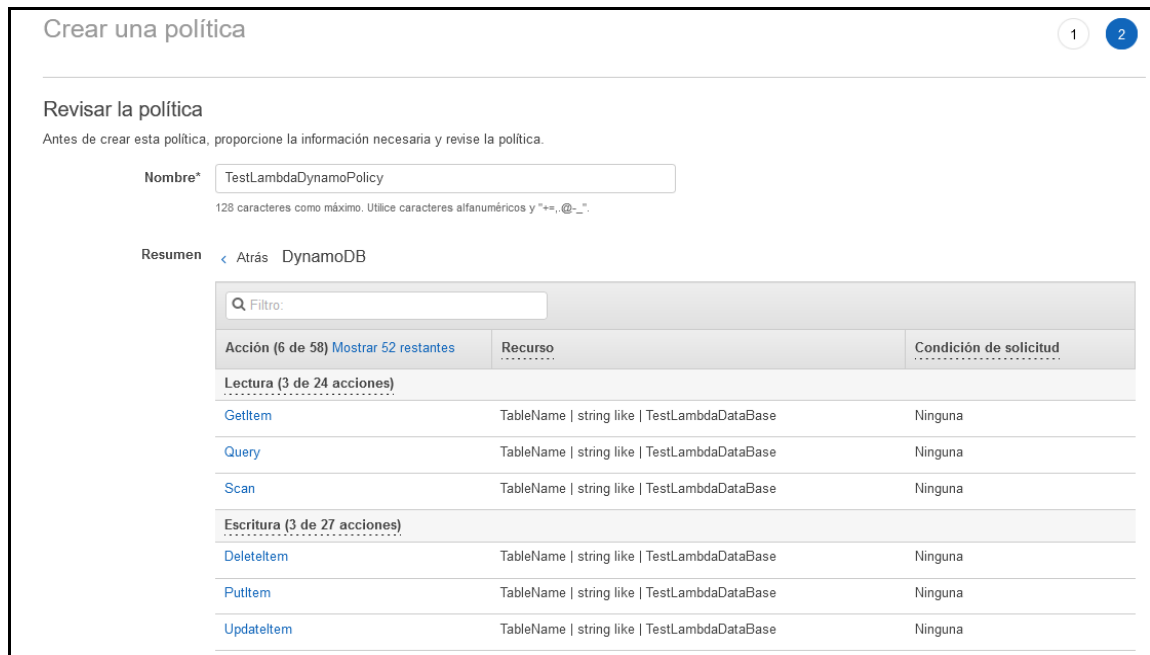
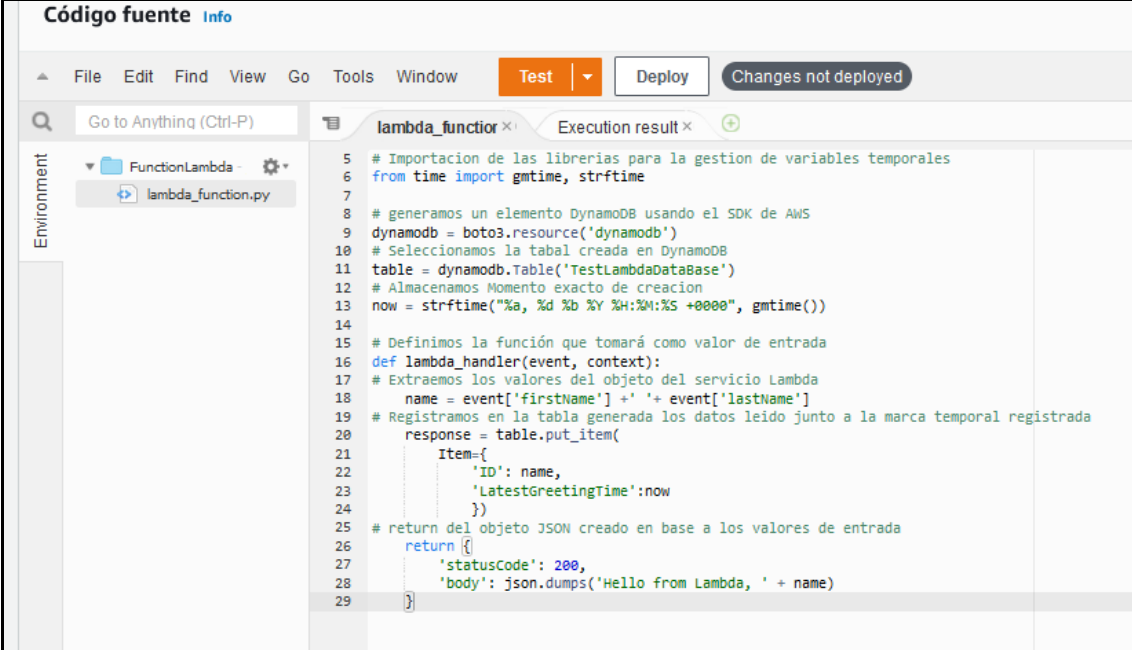


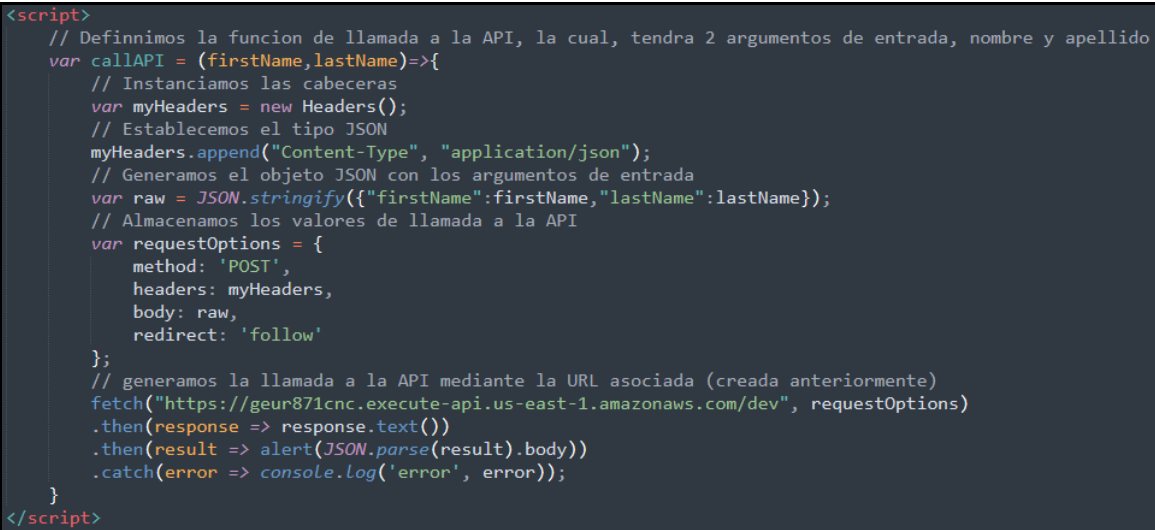
Figura Anexo C - 27: Creación de la Base de Datos NoSQL (VII) – Detalle de la Política de Acceso Generada sobre la BBDD

6.6 Despliegue de la Aplicación Web en la Nube



```
Código fuente Info
File Edit Find View Go Tools Window Test Deploy Changes not deployed
Go to Anything (Ctrl-P)
Environment
FunctionLambda -
lambda_function.py
5 # Importacion de las librerias para la gestion de variables temporales
6 from time import gmtime, strftime
7
8 # generamos un elemento DynamoDB usando el SDK de AWS
9 dynamodb = boto3.resource('dynamodb')
10 # Seleccionamos la tabla creada en DynamoDB
11 table = dynamodb.Table('TestLambdaDataBase')
12 # Almacenamos Momento exacto de creacion
13 now = strftime("%a, %d %b %Y %H:%M:%S +0000", gmtime())
14
15 # Definimos la función que tomará como valor de entrada
16 def lambda_handler(event, context):
17 # Extraemos los valores del objeto del servicio Lambda
18 name = event['firstName'] + ' ' + event['lastName']
19 # Registramos en la tabla generada los datos leído junto a la marca temporal registrada
20 response = table.put_item(
21     Item={
22         'ID': name,
23         'LatestGreetingTime':now
24     })
25 # return del objeto JSON creado en base a los valores de entrada
26 return {
27     'statusCode': 200,
28     'body': json.dumps('Hello from Lambda, ' + name)
29 }
```

Figura Anexo C - 28: Despliegue de la Aplicación Web en la Nube - Fragmento Extendido de Código Asociado a la Codificación de la Función Lambda v2



```
<script>
// Definimos la función de llamada a la API, la cual, tendrá 2 argumentos de entrada, nombre y apellido
var callAPI = (firstName,lastName)=>{
// Instanciamos las cabeceras
var myHeaders = new Headers();
// Establecemos el tipo JSON
myHeaders.append("Content-Type", "application/json");
// Generamos el objeto JSON con los argumentos de entrada
var raw = JSON.stringify({"firstName":firstName,"lastName":lastName});
// Almacenamos los valores de llamada a la API
var requestOptions = {
method: 'POST',
headers: myHeaders,
body: raw,
redirect: 'follow'
};
// generamos la llamada a la API mediante la URL asociada (creada anteriormente)
fetch("https://geur871cnc.execute-api.us-east-1.amazonaws.com/dev", requestOptions)
.then(response => response.text())
.then(result => alert(JSON.parse(result).body))
.catch(error => console.log('error', error));
}
</script>
```

Figura Anexo C - 29: Despliegue de la Aplicación Web en la Nube (II) - Fragmento de Código Asociado a la Llamada de la API como parte del Recurso Web

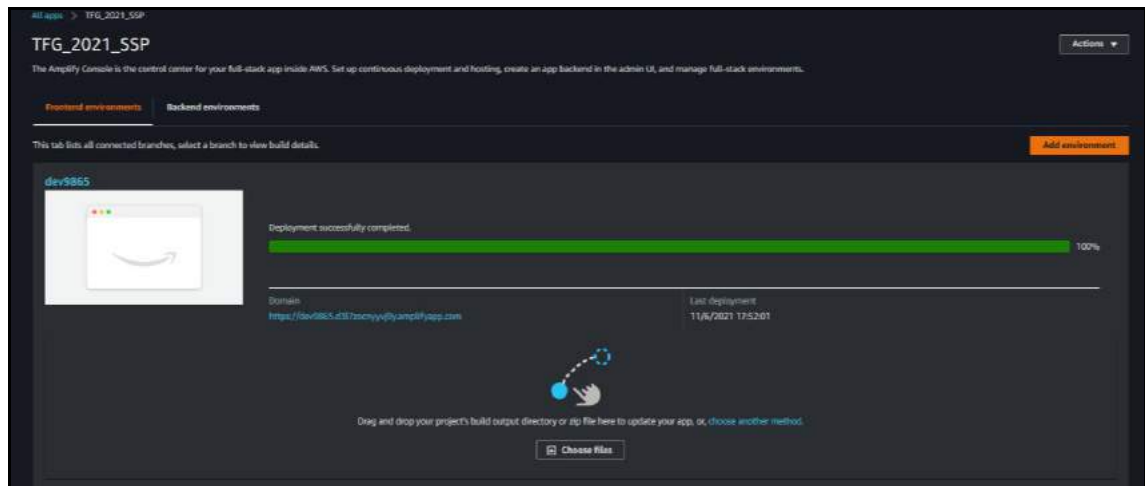


Figura Anexo C - 30: Despliegue de la Aplicación Web en la Nube (III) - Carga y Despliegue del Recurso Web v2 en AWS Amplify

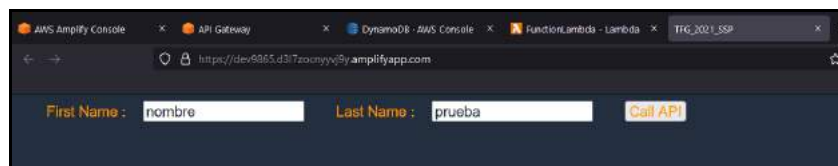


Figura Anexo C - 31: Despliegue de la Aplicación Web en la Nube (IV) – Ejecución de la Aplicación Web Desplegada

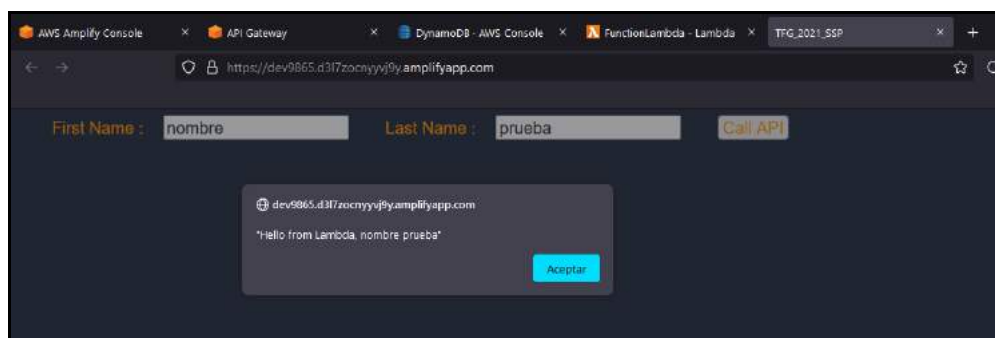


Figura Anexo C - 32: Despliegue de la Aplicación Web en la Nube (V) - Ejecución de la Aplicación Web Desplegada [2]

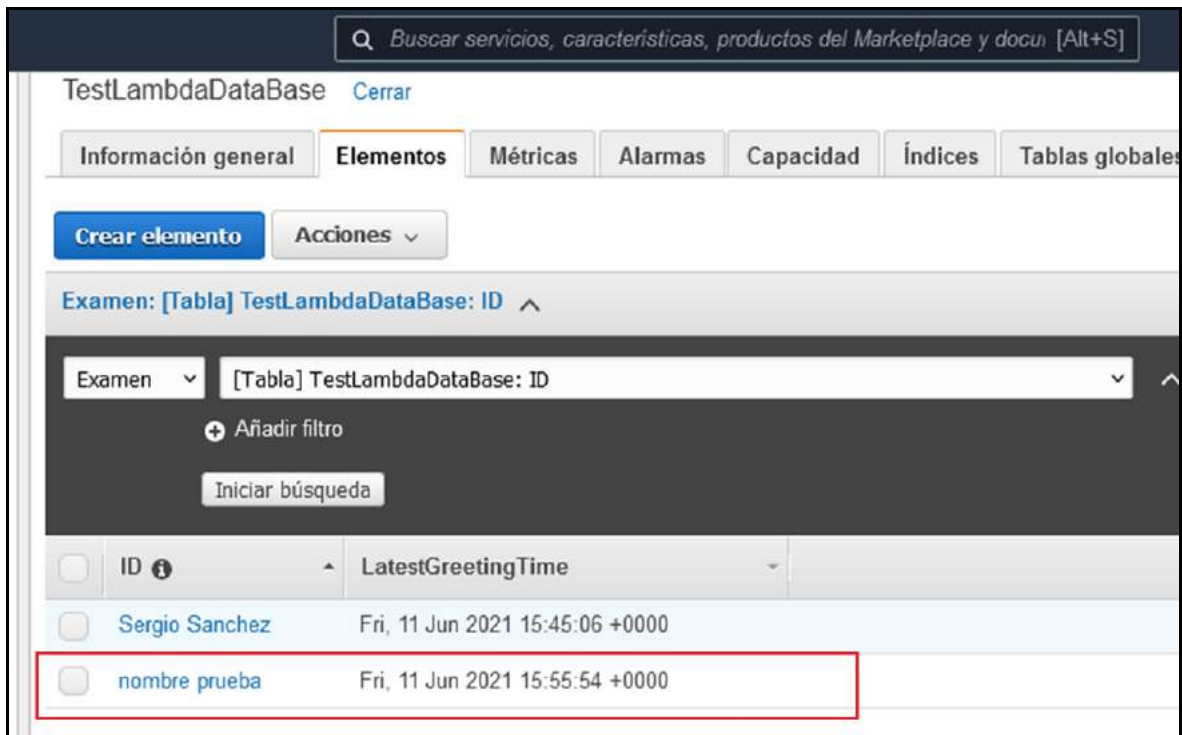


Figura Anexo C - 33: Despliegue de la Aplicación Web en la Nube (VI) – Ejecución de la Aplicación Web Desplegada [3]

Glosario

API	Application Programming Interface
AWS	Amazon Web Services
CPU	Central Processing Unit
CTO	Chief Technical Officer
CORS	Intercambio de Recursos de Origen Cruzado
EEE	Espacio Económico Europeo
Gbps	Gigabyte per Second
HTML	HyperText Markup Language
IAM	AWS Identity and Access Management
JSON	JavaScript Object Notation
LSSI	Servicios de la Sociedad de la Información y del Comercio Electrónico
NIST	National Institute of Standards and Technology
NoSQL	Bases de Datos no Relacionales
PACS	Controladores de Automatización Programables
SLA	Acuerdo de Niveles de Servicio
SO	Sistema Operativo
SQL	Structured Query Language
TIC	Tecnologías de la Información y la Comunicación
UIT	Unión Internacional de Telecomunicaciones
UNIX	Uniplexed Information and Computing Service
VM	Virtual Machine (Máquina Virtual)
VPN	Red Privada Virtual
XML	Extensible Markup Language

