

Grado Universitaria en Ingeniería Informática  
2022-2023

*Trabajo Fin de Grado*

# "Desarrollo de una plataforma central para la gestión de servicios cloud de múltiples proveedores"

---

Rafael Zhu Zhou

Tutor/es

José Antonio Iglesias Martínez

Leganés, 05 de septiembre de 2023



Esta obra se encuentra sujeta a la licencia Creative Commons **Reconocimiento - No Comercial - Sin Obra Derivada**



## RESUMEN

La computación en la nube ha sido una de las tecnologías emergentes de la pasada década. Junto a este crecimiento también han aparecido una gran cantidad de proveedores con sus propias ventajas y desventajas. Debido a esto, existe la posibilidad de crear un sistema con servicios provenientes de distintos proveedores, haciendo la gestión y administración de estos más complicados. El objetivo de este trabajo es la creación de una plataforma capaz de sincronizar los servicios de estos proveedores en un mismo lugar, aumentando la productividad del personal encargado de la gestión de estos servicios.

Para conseguir esto, primero se ha realizado un análisis del mercado y estado del arte sobre el tema. Se ha continuado con el análisis de requisitos y diseño del sistema y su correspondiente desarrollo e implementación. Tras finalizar el desarrollo, se ha realizado la evaluación de las pruebas. Para finalizar, se han añadido unas conclusiones generales y personales y un listado de trabajos futuros relacionados con el tema.

Como resultado, se ha conseguido desarrollar satisfactoriamente el objetivo, consiguiendo una aplicación de escritorio capaz de sincronizar el servicio de máquinas virtuales de Amazon Web Services y Azure con una pequeña cantidad de acciones posibles.

## ABSTRACT

Cloud computing has been one of the emerging technologies of the past decade. Along with this growth has come a multitude of providers with their own advantages and disadvantages. Because of this, there is the possibility of creating a system with services coming from different providers, making the management and administration of these more complicated. The aim of this work is to create a platform capable of synchronising the services of these providers in one place, increasing the productivity of the staff in charge of managing these services.

To achieve this, a market analysis and state of the art on the subject was carried out first. This was followed by the requirements analysis and design of the system and its corresponding development and implementation. After the development was completed, the evaluation of the tests was carried out. Finally, some general and personal conclusions and a list of future work related to the topic have been added.

As a result, the objective has been successfully developed, achieving a desktop application capable of synchronising the Amazon Web Services and Azure virtual machine service with a small number of possible actions.

**Palabras clave:** cloud computing, multi-cloud, AWS, Azure, cloud services, cloud management, desktop app development



## **DEDICATORIA**

Me gustaría dar las gracias a todas esas personas que me han ayudado y que me han apoyado durante la realización de este proyecto. Sin ellos, no hubiese podido finalizar este trabajo. En especial, me gustaría destacar la ayuda del profesor y tutor del proyecto, José Antonio Iglesias Martínez, que me ha ayudado mucho en todo lo referente a la memoria y ha sido sabido mantener la paciencia en cuanto a todas las dudas que le he preguntado.



## ÍNDICE GENERAL

1. INTRODUCCIÓN. . . . .	1
1.1. Motivación . . . . .	1
1.2. Objetivos . . . . .	2
1.3. Resumen del documento. . . . .	2
2. ESTADO DEL ARTE. . . . .	4
2.1. Proveedores de servicios en la nube . . . . .	4
2.2. Arquitecturas multi-cloud y sus ventajas e inconvenientes . . . . .	5
2.3. Trabajos académicos . . . . .	6
3. ANÁLISIS Y DISEÑO . . . . .	9
3.1. Alcance del proyecto. . . . .	9
3.2. Análisis de requisitos y casos de uso de la plataforma . . . . .	10
3.2.1. Plantilla de requisitos . . . . .	10
3.2.2. Requisitos funcionales. . . . .	11
3.2.3. Requisitos no funcionales . . . . .	16
3.2.4. Plantilla casos de uso . . . . .	16
3.2.5. Casos de uso . . . . .	18
3.2.6. Matriz de trazabilidad . . . . .	26
3.3. Diseño de arquitectura . . . . .	27
3.3.1. Interfaz visual . . . . .	27
3.3.2. Controlador del listado . . . . .	28
3.3.3. Confirmar acción. . . . .	28
3.3.4. Inicio de sesión. . . . .	28
3.3.5. Conector JavaScript . . . . .	28
3.3.6. Funcionalidades Python . . . . .	28
3.3.7. Diagrama de clases. . . . .	29
3.4. Interfaz visual . . . . .	30
3.4.1. Prototipo visual. . . . .	30

4. DESARROLLO E IMPLEMENTACIÓN . . . . .	34
4.1. Herramientas y lenguajes de programación . . . . .	34
4.2. Componentes . . . . .	35
4.3. Desarrollo de las funcionalidades. . . . .	36
4.3.1. Funcionalidad: comprobar credenciales . . . . .	36
4.3.2. Funcionalidad: crear máquina virtual. . . . .	37
4.3.3. Funcionalidad: recoger máquinas virtuales . . . . .	39
4.3.4. Funcionalidad: recoger regiones . . . . .	42
4.3.5. Funcionalidad: iniciar máquina virtual. . . . .	43
4.3.6. Funcionalidad: parar máquina virtual . . . . .	44
4.3.7. Funcionalidad: terminar máquina virtual. . . . .	46
5. EVALUACIÓN . . . . .	48
5.1. Pruebas unitarias . . . . .	48
5.2. Cumplimiento de requisitos . . . . .	49
6. GESTIÓN DEL PROYECTO . . . . .	51
6.1. Planificación . . . . .	51
6.1.1. Fase: estado del arte . . . . .	51
6.1.2. Fase: diseño. . . . .	51
6.1.3. Fase: desarrollo. . . . .	52
6.1.4. Fase: post-procesado. . . . .	52
6.1.5. Diagrama de Gantt . . . . .	52
6.2. Presupuesto . . . . .	54
6.2.1. Coste del personal . . . . .	54
6.2.2. Coste material . . . . .	54
6.3. Marco legal . . . . .	55
6.4. Impacto socio-económico . . . . .	55
7. CONCLUSIONES Y TRABAJO FUTURO . . . . .	57
7.1. Conclusiones generales . . . . .	57
7.2. Conclusiones personales. . . . .	58
7.3. Trabajo futuro . . . . .	59
BIBLIOGRAFÍA . . . . .	61





## ÍNDICE DE FIGURAS

3.1	Diagrama de clases . . . . .	29
3.2	Prototipo visual: asociación de cuentas . . . . .	31
3.3	Prototipo visual: listado de máquinas virtuales . . . . .	32
3.4	Prototipo visual: creación de máquinas virtuales . . . . .	32
3.5	Prototipo visual: visualización de parámetros . . . . .	33
4.1	Diagrama de flujo: comprobación de credenciales . . . . .	37
4.2	Diagrama de flujo: crear máquina virtual (AWS) . . . . .	38
4.3	Diagrama de flujo: crear máquina virtual (Azure) . . . . .	39
4.4	Diagrama de flujo: recoger máquinas virtuales (AWS) . . . . .	40
4.5	Diagrama de flujo: recoger máquinas virtuales (Azure) . . . . .	41
4.6	Diagrama de flujo: recoger regiones . . . . .	43
4.7	Diagrama de flujo: iniciar máquina virtual . . . . .	44
4.8	Diagrama de flujo: parar máquina virtual . . . . .	45
4.9	Diagrama de flujo: terminar máquina virtual(AWS) . . . . .	46
4.10	Diagrama de flujo: terminar máquina virtual(Azure) . . . . .	47
6.1	Diagrama de Gantt . . . . .	53



## ÍNDICE DE TABLAS

3.1	Plantilla de requisito . . . . .	11
3.2	Requisito funcional 1: Integración de proveedores de servicio . . . . .	11
3.3	Requisito funcional 2: Tutoriales para asociar cuentas . . . . .	12
3.4	Requisito funcional 3: Bloqueo de la aplicación . . . . .	12
3.5	Requisito funcional 4: Listado de máquinas virtuales . . . . .	12
3.6	Requisito funcional 5: Información resumen . . . . .	13
3.7	Requisito funcional 6: Cambio de la información resumen . . . . .	13
3.8	Requisito funcional 7: Origen de la máquina virtual . . . . .	13
3.9	Requisito funcional 8: Apagar la máquina virtual . . . . .	14
3.10	Requisito funcional 9: Eliminar la máquina virtual . . . . .	14
3.11	Requisito funcional 10: Reanudar la máquina virtual . . . . .	14
3.12	Requisito funcional 11: Información en tiempo real . . . . .	15
3.13	Requisito funcional 12: Crear máquina virtual . . . . .	15
3.14	Requisito funcional 13: Una sola cuenta por proveedor . . . . .	15
3.15	Requisito funcional 14: Visualizar todos los atributos . . . . .	16
3.16	Requisito no funcional 1: Compatibilidad con Linux y Windows . . . . .	16
3.17	Requisito no funcional 2: Conexión a internet . . . . .	16
3.18	Plantilla de caso de uso . . . . .	17
3.19	Caso de uso 1: Asociar cuenta con AWS . . . . .	18
3.20	Caso de uso 2: Asociar cuenta con Azure . . . . .	19
3.21	Caso de uso 3: Creación de una máquina virtual . . . . .	20
3.22	Caso de uso 4: Apagado de una máquina virtual . . . . .	21
3.23	Caso de uso 5: Eliminar una máquina virtual . . . . .	22
3.24	Caso de uso 6: Reanudar una máquina virtual . . . . .	23
3.25	Caso de uso 7: Visualizar todas las máquinas virtuales . . . . .	24
3.26	Caso de uso 8: Visualizar todos los parámetros . . . . .	25
3.27	Caso de uso 9: Modificar la información resumen . . . . .	26
3.28	Matriz de trazabilidad . . . . .	27

4.1	Tabla de entradas y salidas: comprobar credenciales . . . . .	37
4.2	Tabla de entradas y salidas: crear máquina virtual . . . . .	38
4.3	Tabla de entradas y salidas: recoger máquinas virtuales . . . . .	40
4.4	Tabla de entradas y salidas: recoger regiones . . . . .	42
4.5	Tabla de entradas y salidas: iniciar máquina virtual . . . . .	44
4.6	Tabla de entradas y salidas: parar máquina virtual . . . . .	45
4.7	Tabla de entradas y salidas: terminar máquina virtual . . . . .	46
5.1	Tabla de pruebas unitarias . . . . .	49
5.2	Listado de verificación de requisitos . . . . .	50



# 1. INTRODUCCIÓN

En este primer capítulo, se realiza un esbozo general del objetivo del trabajo realizado, la razón por la cual surgió la idea y para qué sirve. Así, la finalidad de este capítulo es explicar en detalle el porqué se está realizando este trabajo y qué objetivos se establecen al iniciar el mismo.

Tras realizar una introducción completa del trabajo y sus objetivos, se menciona la estructura y organización en la que se ha dividido el documento para aclarar en qué orden se explican las cosas y el flujo de la lectura para facilitar la navegación del lector.

## 1.1. Motivación

La solución e idea de crear una plataforma central para todos los servicios en la nube disponibles surge de la experiencia del autor en sus prácticas empresariales, donde se vio obligado a trabajar e interactuar con varios proveedores de servicios en la nube distintos (en concreto, Amazon Web Services(AWS) y Microsoft Azure), siendo esto un impedimento a la hora de trabajar, ya que disponía de varias pestañas para una sola función.

Fuera de esta experiencia también surgen otras razones por las cuales esta solución es una propuesta de gran valor y puede llegar a ser muy importante en el futuro. A continuación, se enuncian las más destacadas.

En primer lugar, es necesario destacar que el uso de servicios multi-cloud se ha convertido en una tendencia cada vez más popular en el mundo de la tecnología debido a las ventajas que ofrece en términos de flexibilidad, escalabilidad y reducción de costos. Todos los clientes de servicios en la nube tienen la posibilidad de seleccionar los servicios más adecuados en función de a sus necesidades y combinarlos de manera eficiente para lograr sus objetivos de negocio al menor coste.

Además, con la creciente demanda de servicios de nube en diferentes regiones geográficas y la necesidad de cumplir con regulaciones locales, contar con una herramienta que permita la utilización de varios proveedores de servicios de nube se vuelve crucial. De esta manera, todos los clientes de servicios en la nube pueden evitar la dependencia de un solo proveedor y tener la capacidad de migrar de una nube a otra según sea necesario.

Otra de las mejoras que puede aportar la implementación de una plataforma multi-cloud es el aumento de la disponibilidad. Al tener disponibles varios servicios en la nube, si en algún momento el proveedor actual que se está utilizando se vuelve inaccesible o presenta reiteradas interrupciones, facilita la creación en otro proveedor de forma fácil y sencilla.

## **1.2. Objetivos**

El objetivo principal de este trabajo es diseñar, desarrollar e implementar una plataforma multi-cloud que permita al usuario final acceder a los recursos y servicios de múltiples proveedores de nube sin necesidad de tener conocimientos previos de programación. En general se va a buscar un enfoque más técnico que de diseño o profesional.

La plataforma multi-cloud que se plantea en este trabajo permite al usuario final acceder a sus servicios de distintos proveedores en una única interfaz, intuitiva y fácil de usar. Esto significa que el usuario final no necesita conocer los detalles técnicos de cada proveedor de nube o aprender a programar para utilizar los servicios de la nube. En lugar de eso, se les permite acceder a los servicios y recursos que necesiten a través de una única plataforma que simplifique el proceso y ahorre tiempo y esfuerzo. Para lograr esto se han definido los siguientes subobjetivos:

- Realizar un estudio de la importancia de los entornos multi-cloud mediante aplicaciones similares y estudios académicos.
- Identificar los requisitos y necesidades de los usuarios finales para la utilización de una plataforma multi-cloud.
- Diseñar la estructura y funcionamiento de la plataforma.
- Desarrollar la plataforma multi-cloud teniendo en cuenta las funcionalidades y requisitos identificados en los objetivos anteriores.
- Realizar pruebas y evaluaciones de la plataforma multi-cloud para asegurar su correcto funcionamiento por parte del usuario final.
- Enunciar todos los trabajos pendientes por hacer.

Además, se plantea un objetivo secundario a lo largo del proyecto, pero no son prioritarios como los mencionados anteriormente:

- Implementación de algún modelo de Inteligencia Artificial dentro de la plataforma.

## **1.3. Resumen del documento**

El primer capítulo de este proyecto es esta introducción, dónde se explica la motivación del tema elegido. Adicionalmente, se han planteado todos los objetivos necesarios para dar por completo el trabajo y un resumen de la estructura de la memoria de este trabajo.

El segundo capítulo consiste en un estudio del estado del arte del tema. En él, se explican algunos conceptos básicos necesarios para el completo entendimiento del trabajo,



como la definición de un entorno multi-cloud. También se ha realizado una recopilación de trabajos académicos relacionados al tema con el fin de enmarcar el estado actual del tema en el mundo académico.

El tercer capítulo consiste en el estudio de las necesidades de la idea planteada para su posterior diseño. Se enuncian todos los requisitos y casos de uso extraídos y se realiza un diagrama de clases. También se ha añadido un prototipo visual.

El cuarto capítulo explica el desarrollo y la implementación de toda la idea. Primero se comenta que herramientas han sido utilizadas durante el desarrollo. Para el desarrollo, se ha elaborado un listado de componentes y funcionalidades con una explicación detallada de cada uno.

El quinto capítulo recoge todas las pruebas realizadas para determinar y controlar la calidad del programa desarrollado. También se ha realizado una recopilación de todos los requisitos tratados en el tercer capítulo para determinar si se han cumplido todos tras el desarrollo.

El sexto capítulo consiste en una recopilación de todas las conclusiones, tanto personales como generales, de todo el proyecto. Tras esto, se enuncian todos los trabajos que se han quedado pendientes por realizar e ideas surgidas durante el desarrollo que no se han podido abordar.

El último capítulo recogen información sobre dos temas importantes durante la elaboración del proyecto. El primero es el presupuesto, donde se realiza una estimación del dinero gastado durante todo el proyecto. El segundo es el marco legal, dónde se realiza un análisis del marco legal sobre el que se encuentra el proyecto.

Finalmente, tras este último capítulo, se encuentra la bibliografía y los anexos del proyecto.

## 2. ESTADO DEL ARTE

A lo largo de este capítulo se explica de forma resumida el estado actual de los servicios en la nube para posteriormente continuar con las ventajas y desventajas de utilizar una arquitectura multi-cloud, así como algunos ejemplos de servicios multi-cloud disponibles actualmente en el mercado. Se finaliza con algunas anotaciones y trabajos académicos relacionados con el tema encontrados para situar mejor el estado actual dentro del mundo académico y no exclusivamente comercial.

### 2.1. Proveedores de servicios en la nube

Para comenzar este epígrafe primero se debe explicar la definición de la nube. La nube es un concepto abstracto que engloba una enorme red de servidores mundiales conectados para formar un único entorno[1]. Dependiendo de la disponibilidad de esta nube, se puede clasificar en nube pública (ofrecido por una empresa externa), privada (creada de forma interna en la infraestructura de la empresa) e híbrida (parte de la nube es pública y parte privada).

Existen un gran número de beneficios al usar una infraestructura basada en la nube. IBM en concreto menciona las siguientes[2]:

- **Flexibilidad:** los servicios en la nube se solicitan con el modelo «on-demand» (se solicita cuando se necesita) que permite escalar en cargas de trabajos variables. Se pueden utilizar modelos públicos, privados o híbridos en función de la necesidad. Muchas de las empresas que ofrecen servicios en la nube también ofrecen diferentes herramientas y servicios que se adaptan a las necesidades.
- **Eficiencia:** dado que la parte computacional se encuentran en servidores externos, su acceso, tanto a datos como al sistema, se puede realizar desde cualquier sitio siempre y cuando se disponga de conexión a internet. Se acelera el lanzamiento de aplicaciones, ya que no es necesario planear y gestionar la infraestructura. Se optimizan los costes debido a la metodología de pago por uso, donde únicamente se cobra por los recursos utilizados.
- **Valor estratégico:** debido a que la mayoría de decisiones relacionadas con la infraestructura las toman los proveedores de servicios en la nube, la empresa que adquiere la arquitectura en la nube se puede centrar en desarrollar las aplicaciones u otras prioridades, lo que supone una ventaja competitiva frente a las empresas que deben gestionar sus propias infraestructuras. También permite la colaboración desde cualquier lugar del mundo, favoreciendo la reciente modalidad de teletrabajo.

Existen muchos proveedores de servicios en la nube, cada uno con sus propias fortalezas y debilidades. Sin embargo, el mercado se podría resumir en un pequeño listado de proveedores de servicios en la nube, ya que ellos poseen casi el 80 % de todo el mercado de servicios en la nube durante el último cuatrimestre de 2020. Estos proveedores son: Amazon Web Services (32 %), Microsoft Azure(20 %), Google Cloud Platform (9 %), Alibaba Cloud(6 %), IBM Cloud (5 %), Salesforce (3 %), Tencent Cloud (2 %) y Oracle Cloud (2 %) [3].

## **2.2. Arquitecturas multi-cloud y sus ventajas e inconvenientes**

Una arquitectura multi-cloud hace referencia al uso de servicios en la nube de diferentes proveedores. Por ejemplo, si una organización aloja su base de datos en Amazon Web Services, pero utiliza máquinas virtuales de Azure se considera que está utilizando una arquitectura multi-cloud. Esta arquitectura presenta múltiples beneficios:

- Un aumento de la fiabilidad y el tiempo de actividad. Al distribuir servicios en muchos proveedores de servicios en la nube, el tiempo de inactividad disminuye. Si un proveedor experimenta una interrupción, los demás proveedores pueden sustituirle, garantizando que los servicios críticos del sistema sigan funcionando.
- Mejora en el rendimiento. Al usar varios proveedores con puntos fuertes en distintas áreas, se puede aprovechar en cada ocasión el mejor proveedor, creando así una mejora directa al rendimiento del sistema.
- Reducción de costes. Como se dispone de varios proveedores de servicios en la nube, a la hora de contratar servicios se pueden comparar los precios de todos los proveedores y elegir la opción que mejor se ajuste a las necesidades específicas.
- Mejora en la seguridad. Al repartir datos entre distintas nubes, se reduce el riesgo de pérdida de datos. También puede proporcionar redundancia y opciones variadas de copias de seguridad, mejorando aún más la seguridad.

Sin embargo, también presenta algunas desventajas que hay que tener en cuenta:

- Aumento de la complejidad. Al tener que manejar distintos servicios en la nube, su administración y control pueden ser más complicados y por ello dedicar más tiempo a estas actividades. Se necesitan más conocimientos, habilidades y recursos para manejar un entorno multi-cloud.
- Dificultad a la hora de integrar. El manejo de datos, comunicaciones y otro tipo de flujo de datos puede ser un desafío importante, especialmente si presentan diferentes API (Interfaz de Programación de Aplicaciones, conjunto de reglas y protocolos que permite que diferentes aplicaciones informáticas se comuniquen entre

sí, compartan datos y funcionen juntas de manera eficiente), protocolos o formatos de archivos. Esto supone desarrollo y personalización adicionales para adaptar la implementación con todos los proveedores.

- Incremento de costes. A pesar de que se ha mencionado que reduce los costes, estos costes son exclusivos de la infraestructura. En este caso aumentarían los costes en otras áreas como la integración, el desarrollo, la administración y la supervisión del entorno multi-cloud.
- Riesgos de seguridad. Aunque aparezca en el apartado de ventajas, esto solo se considera una ventaja si se maneja bien. Si se maneja de manera incorrecta, el entorno multi-cloud, al contar con distintos protocolos de seguridad entre distintos proveedores de servicios en la nube, puede presentar más vulnerabilidades y desafíos a la hora de manejar todos, aumentando los riesgos en la seguridad.

En el mercado actual, existe una amplia variedad de soluciones que permiten la creación y gestión de plataformas multi-cloud, cada una con sus propias características y funcionalidades. Algunos ejemplos de estas soluciones son: CloudBolt Software, Morpheus, Terraform o Ansible.

### **2.3. Trabajos académicos**

Dentro del mundo académico, se está empezando a investigar de manera exhaustiva todas las utilidades de la computación en la nube en muchas áreas y ámbitos.

Por ejemplo, Q. Cao et al. [4] realizaron un estudio donde se analizaba la utilidad de usar entornos en la nube en la gestión de cadenas de suministros. En él, descubren que el mayor problema en este ámbito era el intercambio de información con otras organizaciones, provocados por las limitaciones de sus entornos tecnológicos. Tras esto, analizan la implementación de un entorno en la nube y concluyen que tendría un impacto positivo en el intercambio de información. Como posee un impacto positivo en el intercambio de información, de manera indirecta, un entorno en la nube impacta de forma positiva en el rendimiento en la gestión de cadenas de suministros.

S.A. Bello et al. [5] realizaron un trabajo de investigación donde analizaban la utilidad de servicios en la nube en el área de la construcción. En dicho trabajo, se sacaron las siguientes ideas:

- La computación en la nube está ganando terreno en la industria de la construcción, con importantes oportunidades de crecimiento. Las pequeñas y medianas empresas en el sector están utilizando cada vez más servicios en la nube para digitalizar sus procesos, especialmente a través de aplicaciones de Modelado de Información de la Construcción.

- Los casos de uso principales de la computación en la nube en la construcción incluyen sistemas de seguridad, sistemas de minimización de residuos, gestión de la cadena de suministro, gestión de la energía e informática de construcción. La computación en la nube proporciona potencia informática para el análisis de datos y ofrece un acceso seguro, rentable y conveniente a los datos de construcción para todas las partes interesadas.
- La computación en la nube se utiliza con mayor frecuencia durante las etapas de estudio de viabilidad, diseño y construcción del ciclo de vida del proyecto. Sin embargo, su uso en las etapas de entrega y operaciones sigue siendo menos común. Los sistemas posteriores a la ocupación, como la gestión de instalaciones y la demolición, aún no han aprovechado completamente las tecnologías de la nube.
- El estudio identifica desafíos que obstaculizan una mayor adopción de la computación en la nube en la industria de la construcción y sugiere estrategias para superarlos.
- Se anticipan futuros beneficios de una mayor adopción de la computación en la nube en la industria de la construcción.

Además, también hay una gran cantidad de estudios tratando de mejorar los sistemas en la nube. J. Gao et al. [6] presentaron un modelo de predicción de fallas conocido como Bi-LSTM (Bidirectional Long Short-Term Memory) de múltiples capas, el cual demostró ser más preciso en la predicción de la finalización de tareas y trabajos en comparación con enfoques anteriores. Los resultados revelaron una alta precisión en la predicción de fallas, con un 93 por ciento de precisión en tareas y un 87 por ciento en trabajos, junto con puntajes F1 (medida que combina la precisión, proporción de predicciones positivas correctas, y el recall, proporción de elementos positivos reales correctamente clasificados) igualmente destacados. Además, este método de predicción tiene la ventaja de una baja tasa de falsos positivos, lo que sugiere la eficacia de la gestión proactiva de fallas basada en estas predicciones. Además, cabe destacar que no conlleva una sobrecarga significativa de tiempo en comparación con otros enfoques.

A nivel de gestión de multi-cloud, H.K. Palani et al. [7] estudia los beneficios de utilizar la inteligencia artificial para cualquier operación de la nube. Entre estos beneficios se pueden encontrar, por ejemplo, la detección de anomalías de forma proactiva, visibilidad y monitorización mejorada, automatización, toma de decisiones inteligente y optimización de recursos y costes.

P. Schmidt [8] realizó una revisión del estado actual de los entornos multi-cloud. Entre sus conclusiones, destaca que gracias a las ventajas que ofrece, como evitar un bloqueo por el proveedor o una mejor disponibilidad, los entornos multi-cloud seguirán creciendo y expandiéndose por el mercado. Sin embargo, actualmente se crece a un ritmo lento debido a la gran complejidad de la gestión de entornos multi-cloud. Para ello, el autor

prevé la intervención del uso de la inteligencia artificial como solución a la gestión de estos entornos.

De forma casi idéntica, J. Hong et al. [9] mencionan que el “multi-cloud computing” es una tecnología creciente gracias a las grandes ventajas que aporta en contra parte del tradicional «cloud computing». También menciona la aplicación del Big Data y el Machine Learning para resolver los problemas actuales que presenta la adopción de un entorno multi-cloud.

Apoyando la idea de la aplicación de inteligencia artificial en los entornos multi-cloud. X.Wang et al. [10] introduce un modelo de selección de servicios en la nube denominado CBRSM (cloud service broker-based service selection model) que hace uso de intermediarios de servicios en la nube. A partir de este modelo, se desarrolla una estrategia dinámica de selección de servicios en la nube denominada DCS(dynamic cloud service). Tanto el modelo como la estrategia hacen uso de un mecanismo de aprendizaje adaptativo implementado por los intermediarios de servicios en la nube agrupados, los cuales incorporan un conjunto de algoritmos de selección de servicios en la nube dinámicos. Los resultados de experimentos de simulación detallados demuestran que esta aproximación presenta ventajas y una eficiencia superior en comparación con los enfoques existentes en esta área. Además, es importante destacar que esta estrategia posee una amplia aplicabilidad y puede ser aprovechada en la selección automática de otros servicios web.

Como se puede observar, una importante parte de los trabajos relacionados con los entornos multi-cloud presentan e investigan la idea de utilizar inteligencia artificial para mejorar y facilitar todos los aspectos de este tema, sin necesidad de perder ventajas otorgadas por la misma. Sin embargo, en el caso de este trabajo, la idea es facilitar la gestión de los recursos de un entorno multi-cloud basado en la creación de una herramienta intuitiva y sencilla para realizar estas tareas.

### 3. ANÁLISIS Y DISEÑO

En este capítulo se explica toda la información relacionada con el análisis y al diseño de la solución a desarrollar. Para ello, se comienza explicando el alcance del proyecto, explicando las limitaciones y restricciones impuestas en el proyecto. Tras esto se enuncian todos los requisitos y casos de usos detectados.

Teniendo en cuenta estos puntos, se realiza el diseño de la arquitectura mediante un diagrama de clases. Para finalizar, se enuncian algunas anotaciones de cara a la interfaz visual acompañada de un prototipo visual.

#### 3.1. Alcance del proyecto

En este epígrafe se enuncian las limitaciones impuestas en el proyecto. Estos límites nacen para reducir el gran tamaño que supone este proyecto si se desea que sea completo.

En primer lugar, los entornos multi-cloud poseen infinitud de funciones y servicios, desde crear bases de datos y copias de seguridad hasta automatizar tareas y ejecutar código sin necesidad de tener una máquina virtual.

En segundo lugar, el estudio y el diseño de la interfaz visual también es un tema muy grande donde hay que estudiar los clientes potenciales, la competencia en el mercado y la viabilidad del producto. Además, se debería crear una marca e imagen, las cuales traen una gran cantidad de ventajas en el mercado.

A estos hay que añadir también que durante los últimos años, uno de los temas más importantes a tratar es la seguridad de los datos que manejan las aplicaciones de software. Sin embargo, conseguir esto con desarrolladores no expertos en el tema puede llevar una gran cantidad de tiempo.

Por estas razones, como se menciona en el capítulo 1, se ha tomado un enfoque más técnico, basado en la arquitectura y el desarrollo del software. Para reducir el tamaño del proyecto, se han definido las siguientes limitaciones:

- Elegir únicamente uno de los servicios en la nube para poder gestionar desde la plataforma. De entre todos, se ha decidido que el servicio elegido va a ser el aprovisionamiento de máquinas virtuales. Aun dentro de las máquinas virtuales, se limita aún más, dejando únicamente los atributos de las máquinas virtuales más básicos.
- Desarrollar la aplicación únicamente para Windows. No se va a realizar la compatibilidad con otros sistemas operativos como Linux o MacOS.
- Ignorar el tema de la ciberseguridad. Esto se ha decidido debido a la naturaleza de una aplicación de escritorio local, donde no se tendrá comunicación con el exte-

rior. La única información sensible a mandar y recibir se procesarán mediante los protocolos implantados por los proveedores de servicios en la nube o las librerías utilizadas. Tampoco se va a guardar ningún tipo de información sensible en ningún sitio.

- Se va a implementar una interfaz visual poco elaborada y muy básica para centrar toda la atención del proyecto en la parte técnica. Este tema se detalla en el epígrafe 3.4.
- Se va a elegir únicamente dos proveedores de servicios en la nube. Los dos elegidos han sido Amazon Web Services (AWS) y Azure por ser los dos proveedores más grandes del mercado, comentado durante el epígrafe 2.1.

Con este alcance limitado definido, el proyecto será posible realizarlo con los recursos disponibles para poder cumplir con los objetivos propuestos durante el capítulo 1.

### **3.2. Análisis de requisitos y casos de uso de la plataforma**

En este epígrafe se enuncian todos los requisitos establecidos teniendo en cuenta las limitaciones impuestas en el epígrafe 3.1. A continuación del listado de requisitos, se recopilan todos los casos de uso encontrados en el proyecto y se finaliza con la creación de una matriz de trazabilidad.

#### **3.2.1. Plantilla de requisitos**

Los requisitos se han dividido en dos tipos: funcionales y no funcionales. Los requisitos funcionales son declaraciones detalladas que describen una funcionalidad o capacidad del sistema para obtener el objetivo del usuario. Los requisitos no funcionales son declaraciones detalladas sobre una característica o propiedad del sistema, software o entorno que no está directamente relacionada con una funcionalidad.

Para su identificación se utiliza la notación RF/RNF-00. De esta manera se tienen los requisitos funcionales RF-01 y RF-02; y de requisitos no funcionales los RNF-01 y RNF-02. Además del identificador, cada requisito está acompañado de otros muchos campos:

- Nombre: identificador descriptivo del requisito para describir su funcionalidad de forma resumida.
- Versión: número que identifica de forma indirecta la cantidad de veces que se ha modificado el requisito.
- Tipo: valor que indica si el requisito es funcional o no funcional.
- Dependencias: relación con otros requisitos, los cuales necesitan ser cumplidos antes de comenzar este.



- Descripción: explicación detallada de la funcionalidad que se obtiene al cumplir el requisito.

Se ha decidido ignorar otros campos usuales en las definiciones de requisitos como la necesidad, la prioridad o la importancia del requisito. Esto se debe a la limitada funcionalidad que dispone la aplicación, siendo así necesario el cumplimiento de todos los requisitos, sin dar más importancia a ninguno sobre los demás.

TABLA 3.1. PLANTILLA DE REQUISITO

Identificador	
Nombre	
Versión	
Tipo	
Dependencias	
Descripción	

Todos estos campos de información del requisito son recogidos en tablas a lo largo del documento usando como plantilla la tabla 3.1.

### 3.2.2. Requisitos funcionales

TABLA 3.2. REQUISITO FUNCIONAL 1: INTEGRACIÓN DE PROVEEDORES DE SERVICIO

Identificador	RF-01
Nombre	Integración de proveedores de servicio
Versión	1.0
Tipo	Funcional
Dependencias	
Descripción	La aplicación debe ser capaz de poder asociar las cuentas de AWS y Azure.

TABLA 3.3. REQUISITO FUNCIONAL 2: TUTORIALES PARA ASOCIAR CUENTAS

Identificador	RF-02
Nombre	Tutoriales para asociar cuentas
Versión	1.0
Tipo	Funcional
Dependencias	RF-01
Descripción	Cada proveedor de servicio en la nube debe proporcionar un tutorial detallado sobre cómo asociar la cuenta con la aplicación.

TABLA 3.4. REQUISITO FUNCIONAL 3: BLOQUEO DE LA APLICACIÓN

Identificador	RF-03
Nombre	Bloqueo de la aplicación
Versión	1.1
Tipo	Funcional
Dependencias	RF-01. RF-02
Descripción	Si la aplicación no está asociada a ninguna cuenta de ningún proveedor, la aplicación no debe mostrar nada, salvo el botón para asociar cuenta.

TABLA 3.5. REQUISITO FUNCIONAL 4: LISTADO DE MÁQUINAS VIRTUALES

Identificador	RF-04
Nombre	Listado de máquinas virtuales
Versión	1.0
Tipo	Funcional
Dependencias	RF-01
Descripción	Al asociar una cuenta de un proveedor de servicio, en la pantalla inicial, se debe mostrar un listado con todas las máquinas virtuales de esa cuenta.

TABLA 3.6. REQUISITO FUNCIONAL 5: INFORMACIÓN  
RESUMEN

Identificador	RF-05
Nombre	Información resumen
Versión	1.1
Tipo	Funcional
Dependencias	RF-01. RF-04
Descripción	El sistema debe mostrar información resumen de cada máquina virtual sin necesidad de acceder a ella, directamente desde el propio listado. Esta información resumen consta de tres campos de la máquina virtual.

TABLA 3.7. REQUISITO FUNCIONAL 6: CAMBIO DE LA  
INFORMACIÓN RESUMEN

Identificador	RF-06
Nombre	Cambio de la información resumen
Versión	1.0
Tipo	Funcional
Dependencias	RF-01. RF-04. RF-05
Descripción	El usuario debe poder cambiar la información que se muestra como resumen en el listado.

TABLA 3.8. REQUISITO FUNCIONAL 7: ORIGEN DE LA  
MÁQUINA VIRTUAL

Identificador	RF-07
Nombre	Origen de la máquina virtual
Versión	1.0
Tipo	Funcional
Dependencias	RF-01. RF-04.
Descripción	El sistema debe mostrar de forma clara a que proveedor pertenece cada máquina virtual.

TABLA 3.9. REQUISITO FUNCIONAL 8: APAGAR LA MÁQUINA VIRTUAL

Identificador	RF-08
Nombre	Apagar la máquina virtual
Versión	1.0
Tipo	Funcional
Dependencias	RF-01. RF-04.
Descripción	El usuario debe poder apagar la máquina virtual desde el listado. Siempre se debe pedir una confirmación de la acción.

TABLA 3.10. REQUISITO FUNCIONAL 9: ELIMINAR LA MÁQUINA VIRTUAL

Identificador	RF-09
Nombre	Eliminar la máquina virtual
Versión	1.0
Tipo	Funcional
Dependencias	RF-01. RF-04.
Descripción	El usuario debe poder eliminar definitivamente la máquina virtual desde el listado. Siempre se debe pedir una confirmación de la acción.

TABLA 3.11. REQUISITO FUNCIONAL 10: REANUDAR LA MÁQUINA VIRTUAL

Identificador	RF-10
Nombre	Reanudar la máquina virtual
Versión	1.0
Tipo	Funcional
Dependencias	RF-01. RF-04. RF-08
Descripción	El usuario debe poder reanudar una máquina virtual apagada desde el listado. Siempre se debe pedir una confirmación de la acción.

TABLA 3.12. REQUISITO FUNCIONAL 11: INFORMACIÓN EN  
TIEMPO REAL

Identificador	RF-11
Nombre	Información en tiempo real
Versión	1.0
Tipo	Funcional
Dependencias	RF-01. RF-04.
Descripción	Toda la información que muestre el sistema debe ser actualizada en tiempo real. Se considera tiempo real siempre y cuando el intervalo entre actualizaciones no sea mayor a cinco minutos.

TABLA 3.13. REQUISITO FUNCIONAL 12: CREAR MÁQUINA  
VIRTUAL

Identificador	RF-12
Nombre	Crear máquina virtual
Versión	1.1
Tipo	Funcional
Dependencias	RF-01. RF-04.
Descripción	El usuario debe tener a disposición un botón para crear una máquina virtual. Al crear la máquina, el usuario debe poder elegir los atributos básicos y el proveedor de la máquina.

TABLA 3.14. REQUISITO FUNCIONAL 13: UNA SOLA CUENTA  
POR PROVEEDOR

Identificador	RF-13
Nombre	Una sola cuenta por proveedor
Versión	1.0
Tipo	Funcional
Dependencias	RF-01.
Descripción	El usuario solo podrá conectar una única cuenta por proveedor de servicios.

TABLA 3.15. REQUISITO FUNCIONAL 14: VISUALIZAR TODOS  
LOS ATRIBUTOS

Identificador	RF-14
Nombre	Visualizar todos los atributos
Versión	1.0
Tipo	Funcional
Dependencias	RF-01. RF-04.
Descripción	El sistema debe permitir al usuario la posibilidad de visualizar todos los parámetros disponibles en la aplicación de una máquina visual si se desea.

### 3.2.3. Requisitos no funcionales

TABLA 3.16. REQUISITO NO FUNCIONAL 1: COMPATIBILIDAD  
CON LINUX Y WINDOWS

Identificador	RNF-01
Nombre	Compatibilidad con Windows 10
Versión	1.0
Tipo	No funcional
Dependencias	
Descripción	La aplicación se debe poder utilizar en el sistema operativo Windows 10.

TABLA 3.17. REQUISITO NO FUNCIONAL 2: CONEXIÓN A  
INTERNET

Identificador	RNF-02
Nombre	Conexión a internet
Versión	1.0
Tipo	No funcional
Dependencias	
Descripción	El sistema solo funcionará si el usuario está conectado a internet.

### 3.2.4. Plantilla casos de uso

Tras definir los requisitos, se procede a explicar la parte de los casos de uso. Un caso de uso es una representación del uso del sistema en una situación concreta y específica. Para su identificación se utiliza la notación C00, donde 00 representa el número del caso.

Por ejemplo, los dos primeros casos de uso se identifican como C01 y C02. El formato de la definición de cada caso de uso posee los siguientes campos:

- **Identificador:** código identificador del caso de uso.
- **Nombre:** identificador descriptivo del caso de uso para describir su funcionalidad de forma resumida.
- **Versión:** número que identifica de forma indirecta la cantidad de veces que se ha modificado el caso de uso.
- **Descripción:** explicación detallada sobre el caso de uso.
- **Objetivo:** meta final que desea obtener el usuario o actor.
- **Condiciones previas:** listado de condiciones que se deben cumplir para que el caso de uso se lleve a cabo.
- **Condiciones posteriores:** listado de cambios o resultados esperados tras la secuencia de pasos.
- **Secuencia:** listado de pasos ordenados para obtener el objetivo.
- **Requisitos:** listado de los requisitos a los cuales afecta el caso de uso.

TABLA 3.18. PLANTILLA DE CASO DE USO

Identificador	
Nombre	
Versión	
Descripción	
Objetivo	
Condiciones previas	
Condiciones posteriores	
Secuencia	
Requisitos	

De la misma manera que en los requisitos, toda la información viene recogida en una tabla, la cual sigue el formato que se observa en la plantilla que se muestra en la tabla 3.18.

### 3.2.5. Casos de uso

TABLA 3.19. CASO DE USO 1: ASOCIAR CUENTA CON AWS

Identificador	C01
Nombre	Asociar cuenta con AWS.
Versión	1.0
Descripción	El usuario dispone de una cuenta de AWS con máquinas virtuales y quiere empezar a usar la aplicación. Por ello, desea asociar su cuenta de AWS con la aplicación.
Objetivo	Conseguir asociar la cuenta de AWS del usuario con la aplicación.
Condiciones previas	1. El usuario debe poseer una cuenta de AWS.
Condiciones posteriores	1. Si el usuario posee alguna máquina virtual en la cuenta, esta debería aparecer en la pantalla.
Secuencia	<ol style="list-style-type: none"><li>1. Abrir la aplicación.</li><li>2. Pulsar el botón de «Asociar Cuentas».</li><li>3. Elegir AWS.</li><li>4. Introducir todos los datos que se solicitan.</li><li>5. Pulsar el botón de «Asociar Cuentas».</li></ol>
Requisitos	RF-01. RF-02. RF-03. RF-13.



TABLA 3.20. CASO DE USO 2: ASOCIAR CUENTA CON AZURE

Identificador	C02
Nombre	Asociar cuenta con Azure.
Versión	1.0
Descripción	El usuario dispone de una cuenta de Azure con máquinas virtuales y quiere empezar a usar la aplicación. Por ello, desea asociar su cuenta de Azure con la aplicación.
Objetivo	Conseguir asociar la cuenta de Azure del usuario con la aplicación.
Condiciones previas	1. El usuario debe poseer una cuenta de Azure.
Condiciones posteriores	1. Si el usuario posee alguna máquina virtual en la cuenta, esta debería aparecer en la pantalla.
Secuencia	<ol style="list-style-type: none"> <li>1. Abrir la aplicación.</li> <li>2. Pulsar el botón de «Asociar Cuentas».</li> <li>3. Elegir Azure.</li> <li>4. Introducir todos los datos que se solicitan.</li> <li>5. Pulsar el botón de «Asociar Cuentas».</li> </ol>
Requisitos	RF-01. RF-02. RF-03. RF-13.

TABLA 3.21. CASO DE USO 3: CREACIÓN DE UNA MÁQUINA VIRTUAL

Identificador	C03
Nombre	Creación de una máquina virtual.
Versión	1.0
Descripción	Con la cuenta asociada, el usuario desea crear una máquina virtual en uno de los proveedores.
Objetivo	Crear una máquina virtual en el proveedor seleccionado.
Condiciones previas	1. Debe haber al menos una cuenta asociada a uno de los proveedores de servicio. (Ver C01 y C02).
Condiciones posteriores	1. La nueva máquina virtual debe aparecer en el listado de máquinas virtuales tanto en la aplicación como en el proveedor.
Secuencia	<ol style="list-style-type: none"> <li>1. Abrir la aplicación.</li> <li>2. Comprobar si hay una cuenta asociada.</li> <li>3. Pulsar el botón de «Crear».</li> <li>4. Seleccionar todos los parámetros según la necesidad.</li> <li>5. Confirmar la creación de la máquina virtual</li> </ol>
Requisitos	RF-01. RF-02. RF-03. RF-12.

TABLA 3.22. CASO DE USO 4: APAGADO DE UNA MÁQUINA VIRTUAL

Identificador	C04
Nombre	Apagado de una máquina virtual.
Versión	1.0
Descripción	El usuario dispone de una máquina virtual para alojar un servicio que solo quiere tener activo por los días. Ahora desea apagarla cada día antes de acabar.
Objetivo	Apagar la máquina virtual deseada.
Condiciones previas	<ol style="list-style-type: none"> <li>1. Debe haber al menos una cuenta asociada a uno de los proveedores de servicio. (Ver C01 y C02).</li> <li>2. Disponer de una máquina virtual encendida asociada a la cuenta.</li> </ol>
Condiciones posteriores	<ol style="list-style-type: none"> <li>1. La máquina virtual seleccionada deberá aparecer como apagada tanto en la aplicación como en el proveedor.</li> </ol>
Secuencia	<ol style="list-style-type: none"> <li>1. Abrir la aplicación.</li> <li>2. Comprobar si hay una cuenta asociada.</li> <li>3. Buscar la máquina virtual que se desea apagar.</li> <li>4. Pulsar el botón de «Apagar».</li> <li>5. Confirmar el apagado.</li> </ol>
Requisitos	RF-01. RF-02. RF-03. RF-04. RF-08.

TABLA 3.23. CASO DE USO 5: ELIMINAR UNA MÁQUINA VIRTUAL

Identificador	C05
Nombre	Eliminar una máquina virtual.
Versión	1.0
Descripción	El usuario ya no necesita más la máquina virtual seleccionada. Por ello desea eliminarla para dejar de gastar dinero.
Objetivo	Eliminar la máquina virtual deseada.
Condiciones previas	<ol style="list-style-type: none"> <li>1. Debe haber al menos una cuenta asociada a uno de los proveedores de servicio. (Ver C01 y C02).</li> <li>2. Disponer de una máquina virtual, ya sea encendida o apagada, asociada a la cuenta.</li> </ol>
Condiciones posteriores	<ol style="list-style-type: none"> <li>1. La máquina virtual seleccionada deberá aparecer como eliminada o, directamente no aparece más, tanto en la aplicación como en el proveedor.</li> </ol>
Secuencia	<ol style="list-style-type: none"> <li>1. Abrir la aplicación.</li> <li>2. Comprobar si hay una cuenta asociada.</li> <li>3. Buscar la máquina virtual que se desea apagar.</li> <li>4. Pulsar el botón de «Eliminar».</li> <li>5. Confirmar el apagado.</li> </ol>
Requisitos	RF-01. RF-02. RF-03. RF-04. RF-09.

TABLA 3.24. CASO DE USO 6: REANUDAR UNA MÁQUINA  
VIRTUAL

Identificador	C06
Nombre	Reanudar una máquina virtual.
Versión	1.0
Descripción	El usuario quiere reanudar una máquina virtual que había apagado por la noche.
Objetivo	Reanudar una máquina virtual.
Condiciones previas	<ol style="list-style-type: none"> <li>1. Debe haber al menos una cuenta asociada a uno de los proveedores de servicio. (Ver C01 y C02).</li> <li>2. Disponer de una máquina virtual apagada asociada a la cuenta.</li> </ol>
Condiciones posteriores	<ol style="list-style-type: none"> <li>1. La máquina virtual seleccionada deberá aparecer encendida tanto en la aplicación como en el proveedor.</li> </ol>
Secuencia	<ol style="list-style-type: none"> <li>1. Abrir la aplicación.</li> <li>2. Comprobar si hay una cuenta asociada.</li> <li>3. Buscar la máquina virtual que se desea reanudar.</li> <li>4. Pulsar el botón de «Reanudar».</li> <li>5. Confirmar la reanudación.</li> </ol>
Requisitos	RF-01. RF-02. RF-03. RF-04. RF-10.

TABLA 3.25. CASO DE USO 7: VISUALIZAR TODAS LAS  
MÁQUINAS VIRTUALES

Identificador	C07
Nombre	Visualizar todas las máquinas virtuales
Versión	1.0
Descripción	El usuario quiere saber cuántas máquinas virtuales tiene ligadas a sus cuentas.
Objetivo	Ver todas las máquinas, tanto apagadas como encendidas, asociadas a sus cuentas.
Condiciones previas	<ol style="list-style-type: none"> <li>1. Debe haber al menos una cuenta asociada a uno de los proveedores de servicio. (Ver C01 y C02).</li> <li>2. Disponer de una o más máquinas virtuales asociadas a las cuentas.</li> </ol>
Condiciones posteriores	En la pantalla principal de la aplicación, se mostrarán todas las máquinas virtuales de las cuentas asociadas.
Secuencia	<ol style="list-style-type: none"> <li>1. Abrir la aplicación.</li> <li>2. Comprobar si hay una cuenta asociada.</li> <li>3. Navegar por la pantalla principal para visualizar todas las máquinas virtuales.</li> </ol>
Requisitos	RF-01. RF-02. RF-03. RF-04. RF-07. RF-11.

TABLA 3.26. CASO DE USO 8: VISUALIZAR TODOS LOS  
PARÁMETROS

Identificador	C08
Nombre	Visualizar todos los parámetros.
Versión	1.0
Descripción	El usuario no encuentra la información que necesita de la máquina. Por lo que desea visualizar todos los parámetros disponibles de la misma.
Objetivo	Visualizar todos los parámetros.
Condiciones previas	<ol style="list-style-type: none"> <li>1. Debe haber al menos una cuenta asociada a uno de los proveedores de servicio. (Ver C01 y C02).</li> <li>2. Disponer de una máquina virtual asociada a la cuenta.</li> </ol>
Condiciones posteriores	<ol style="list-style-type: none"> <li>1. Aparecerá un listado con todos los atributos por pantalla de la máquina virtual seleccionada.</li> </ol>
Secuencia	<ol style="list-style-type: none"> <li>1. Abrir la aplicación.</li> <li>2. Comprobar si hay una cuenta asociada.</li> <li>3. Buscar la máquina virtual.</li> <li>4. Pulsar el botón de «Más info».</li> </ol>
Requisitos	RF-01. RF-02. RF-03. RF-04. RF-14.

TABLA 3.27. CASO DE USO 9: MODIFICAR LA INFORMACIÓN  
RESUMEN

Identificador	C09
Nombre	Modificar la información resumen.
Versión	1.0
Descripción	El usuario desea ver un parámetro que actualmente no se encuentra visible desde el listado y quiere cambiarlo.
Objetivo	Modificar la información resumen.
Condiciones previas	<ol style="list-style-type: none"> <li>1. Debe haber al menos una cuenta asociada a uno de los proveedores de servicio. (Ver C01 y C02).</li> <li>2. Disponer de una máquina virtual asociada a la cuenta.</li> </ol>
Condiciones posteriores	<ol style="list-style-type: none"> <li>1. El nuevo parámetro se podrá visualizar desde el listado.</li> </ol>
Secuencia	<ol style="list-style-type: none"> <li>1. Abrir la aplicación.</li> <li>2. Comprobar si hay una cuenta asociada.</li> <li>3. Pulsar el botón de «Asociar Cuentas».</li> <li>4. Ir a la sección de «Información Resumen».</li> <li>5. Modificar según necesidad los parámetros.</li> <li>6. Pulsar en «Cambiar».</li> </ol>
Requisitos	RF-01. RF-02. RF-03. RF-04. RF-05. RF-06.

### 3.2.6. Matriz de trazabilidad

Para garantizar la integridad y completitud del proyecto, se ha elaborado una matriz de trazabilidad. En esta matriz tiene como objetivo comprobar si todos los requisitos son tratados en los casos de uso, pero también es útil para saber qué grupo de requisitos forman un caso de uso en específico.

En esta matriz de trazabilidad el eje de ordenadas representan los requisitos mediante sus identificadores y el eje de abscisas representan los casos de uso mediante sus identificadores. Si el requisito se relaciona con el caso de uso, la casilla con sus coordenadas se



marca con una 'X'.

TABLA 3.28. MATRIZ DE TRAZABILIDAD

	C01	C02	C03	C04	C05	C06	C07	C08	C09
RF-01	X	X	X	X	X	X	X	X	X
RF-02	X	X	X	X	X	X	X	X	X
RF-03	X	X	X	X	X	X	X	X	X
RF-04				X	X	X	X	X	X
RF-05									X
RF-06									X
RF-07							X		
RF-08				X					
RF-09					X				
RF-10						X			
RF-11							X		
RF-12			X						
RF-13	X	X							
RF-14								X	

En la tabla 3.28 se observa como todos los requisitos son tratados mediante los casos de uso.

### 3.3. Diseño de arquitectura

En este epígrafe se recoge toda la arquitectura diseñada para implementar y desarrollar los casos de uso detectados. Para conseguir este objetivo, se identifican todas las clases del sistema y se elabora un diagrama de clases en función de este listado de clases. En cada clase se explica detalladamente con todos sus atributos y métodos. En total se han detectado doce clases en la aplicación:

#### 3.3.1. Interfaz visual

Esta clase es la encargada de mostrar visualmente todos los aspectos de la aplicación al usuario. También se encarga de recibir todas las entradas que introduzca el usuario. Como atributos, dispone del listado de máquinas virtuales y de la región seleccionada por el usuario. No dispone de ningún método.

### **3.3.2. Controlador del listado**

Esta clase es la encargada de realizar cualquier tratamiento de información del listado de máquinas virtuales. Se encarga tanto de solicitar la información de todas las máquinas virtuales como de mandar estos datos a la interfaz visual. Como atributos, dispone de dos variables que indican si la cuenta de AWS o Azure se encuentra conectada. Como métodos, posee el método para mandar la información y otro método para solicitar la información de todas las máquinas virtuales.

### **3.3.3. Confirmar acción**

Esta clase, recibiendo los datos introducidos por el usuario, se encarga de realizar la llamada correspondiente en todas las acciones relacionadas con el estado de una máquina virtual. Como atributos posee el identificador de la máquina virtual que desea cambiar su estado, la acción que se desea realizar y el proveedor de la máquina que se desea cambiar. Como método, dispone de uno que se encarga de tratar todos los datos necesarios para realizar la acción deseada y contactar con el conector de JavaScript.

### **3.3.4. Inicio de sesión**

Esta clase se encarga de comprobar si las credenciales introducidas por el usuario son válidas o no. Como atributos, posee un vector de cadenas de texto representando las credenciales aún no verificadas. Como método, dispone de uno que se encarga de tratar la información para realizar la comprobación de credenciales y otro para recoger las regiones disponibles.

### **3.3.5. Conector JavaScript**

Esta clase se encarga de actuar como unión entre la parte visual y las llamadas a los proveedores de servicios. Como atributos, no posee ninguno. Como métodos, posee uno por cada llamada necesaria al proveedor de servicios. Cada método se encarga de realizar la llamada al código correspondiente que realice la acción que se desee ejecutar.

### **3.3.6. Funcionalidades Python**

En este caso, no es una única clase, sino un conjunto. Cada clase corresponde a una llamada al proveedor de servicios en la nube y su única función es llamar y devolver la información deseada. No posee ningún atributo y su único método es la ejecución de la llamada.

### 3.3.7. Diagrama de clases

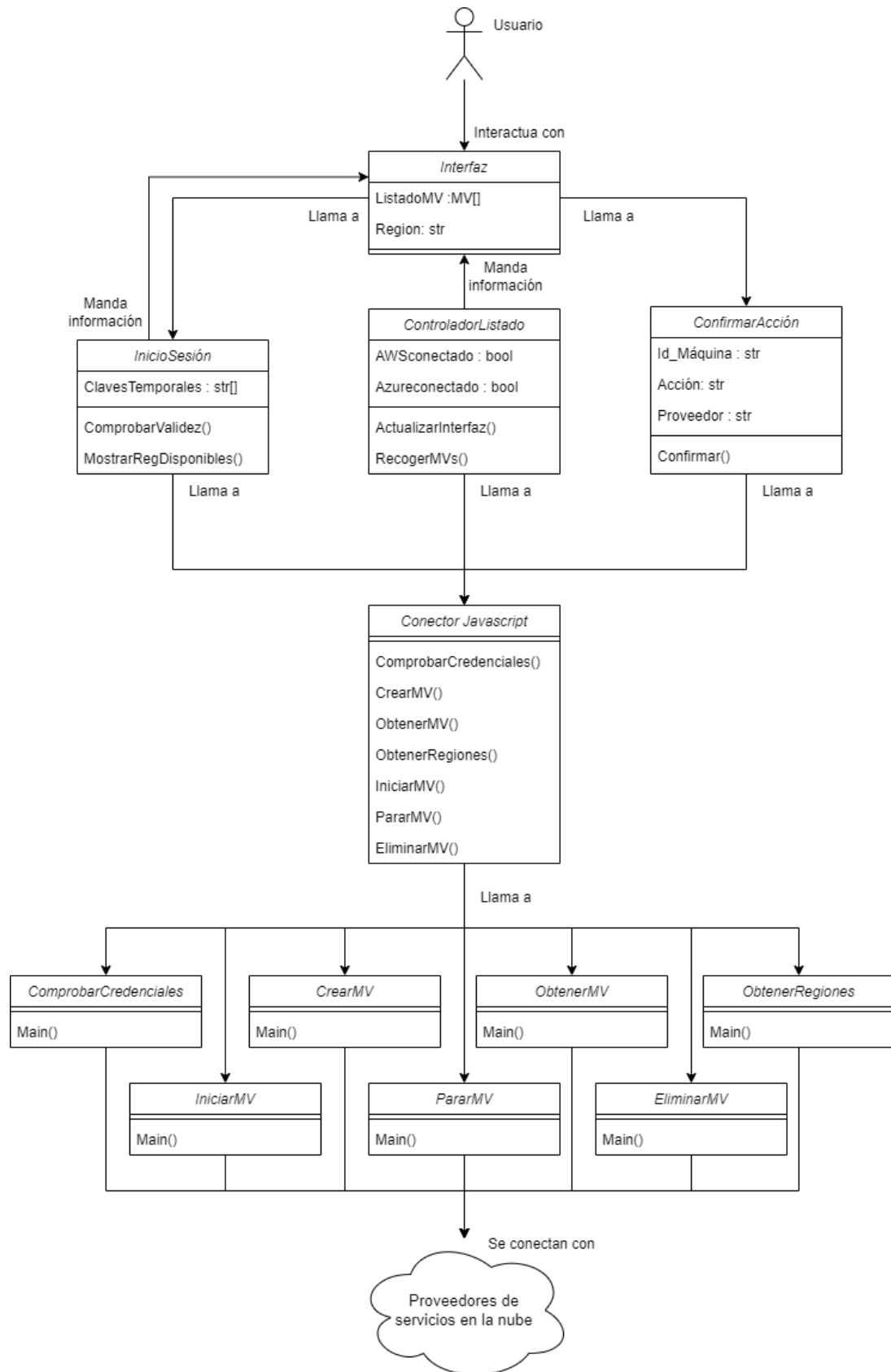


Fig. 3.1. Diagrama de clases

Utilizando estas clases, se ha elaborado el diagrama de clases para facilitar la comprensión de la arquitectura del código. El resultado se muestra en la figura 3.1. Este diagrama ha sido creado utilizando la herramienta en línea Draw.io, que permite realizar diagramas de manera gratuita e intuitiva.

### **3.4. Interfaz visual**

A nivel de interfaz visual, se busca una interfaz simple, intuitiva y cómoda. Para conseguir esto, se realizó un diseño muy básico sin ningún tipo de análisis ni estudio de experiencia de usuario. El resultado final consiste de cuatro pantallas principales:

- Pantalla de inicio de sesión: pantalla para asociar cuentas de proveedores en la nube. Se muestra siempre y cuando no se encuentre asociada ninguna cuenta. En la parte inferior, se encuentra el botón para asociar cuentas y también indicadores para saber si se ha asociado alguna cuenta.
- Pantalla principal: pantalla que muestra todas las máquinas virtuales contratadas en las cuentas asociadas. Se muestra siempre y cuando haya alguna cuenta asociada. En la parte inferior, aparece un nuevo botón para crear una máquina virtual.
- Creación de máquinas virtuales: pantalla que se muestra al crear una nueva máquina virtual. Se muestra tras pulsar el botón de crear una máquina virtual en la parte inferior de la interfaz. En esta pantalla se muestra el formulario que hay que rellenar para crear una nueva máquina virtual.
- Datos de la máquina virtual: pantalla que se muestra cuando se desea observar todos los atributos de una máquina virtual. Se muestra al solicitar toda la información de una máquina virtual. En esta pantalla se muestran todos los parámetros de la máquina virtual que se ha solicitado.

A nivel profesional se debería realizar un estudio mucho más elaborado, teniendo en cuenta el público objetivo, los recursos que tienen, aplicaciones similares y otros muchos factores. Sin embargo, como se ha mencionado previamente, se ha tomado un enfoque más técnico, por lo que no se ha realizado este análisis exhaustivo para realizar una interfaz visual de alto nivel. Del mismo modo se han utilizado colores básicos, sin diseñar un icono de aplicación ni marca y con formas primitivas.

#### **3.4.1. Prototipo visual**

Para obtener un modelo visual de la interfaz propuesta, se ha desarrollado un prototipo visual de la aplicación. Se ha utilizado la herramienta en línea Uizard. La utilidad de un prototipo visual reside en su bajo coste y rápido desarrollo, habilitando una retroalimentación temprana por parte del interesado sin invertir muchos recursos. Gracias a esto, se

puede llegar a ahorrar mucho tiempo y recursos plasmando de manera inicial el estado final del producto.

Este prototipo posee la interfaz final deseada de cada pantalla con explicaciones breves sobre cada parte y sirve como guía para el desarrollo. Sin embargo, no es el resultado definitivo, aunque lo más correcto sería que se pareciese a él.

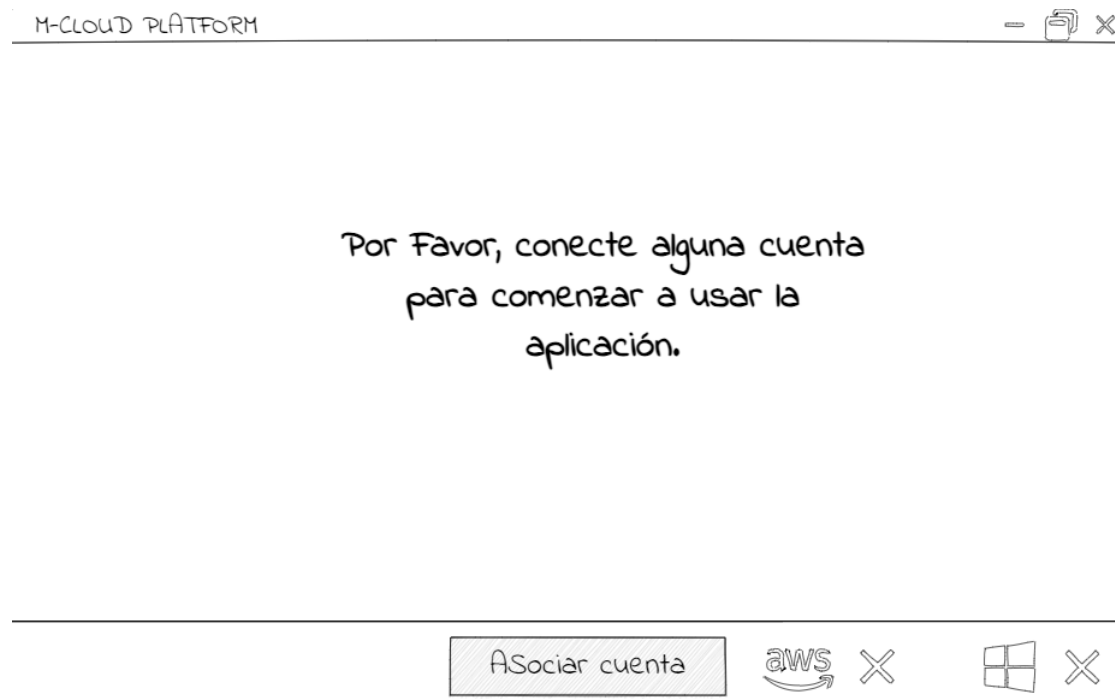


Fig. 3.2. Prototipo visual: asociación de cuentas

En la figura 3.2, se representa la pantalla del escenario donde el usuario no ha asociado ninguna cuenta todavía. En esta pantalla, no se muestra nada, salvo un mensaje que pide asociar alguna cuenta y la parte inferior común a todas las pantallas.

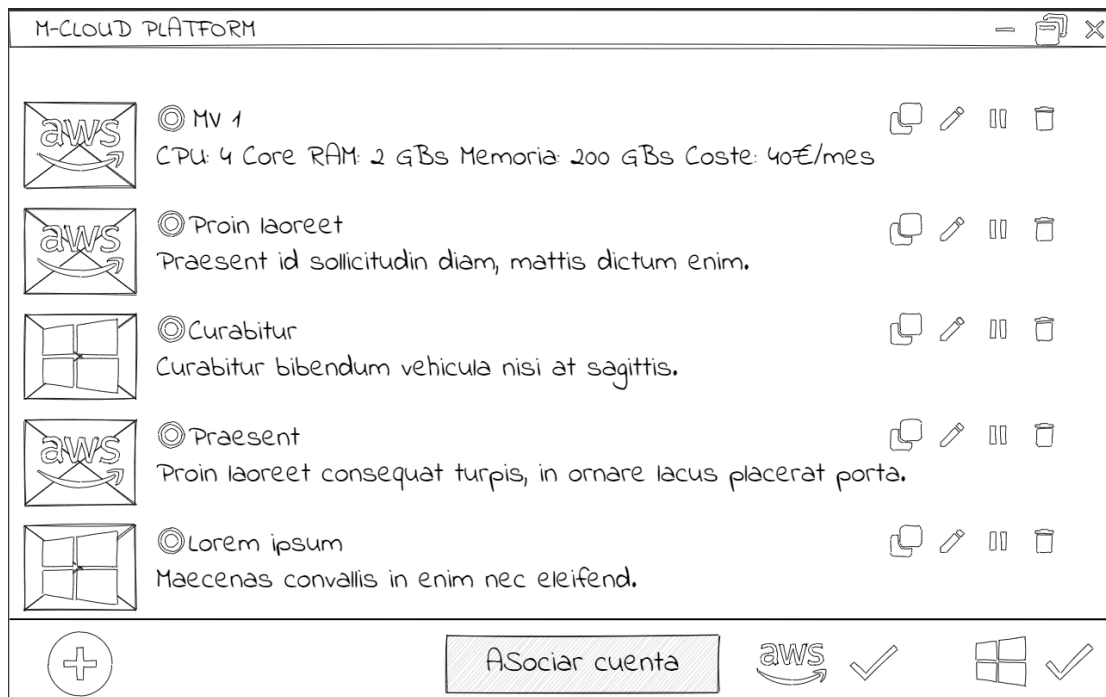


Fig. 3.3. Prototipo visual: listado de máquinas virtuales

En la figura 3.3, tras asociar una o varias cuentas, se muestran todas las máquinas virtuales con la información, resumen de cada una y sus botones correspondientes a cada acción. En la parte inferior izquierda se observa también el botón de crear máquinas virtuales.

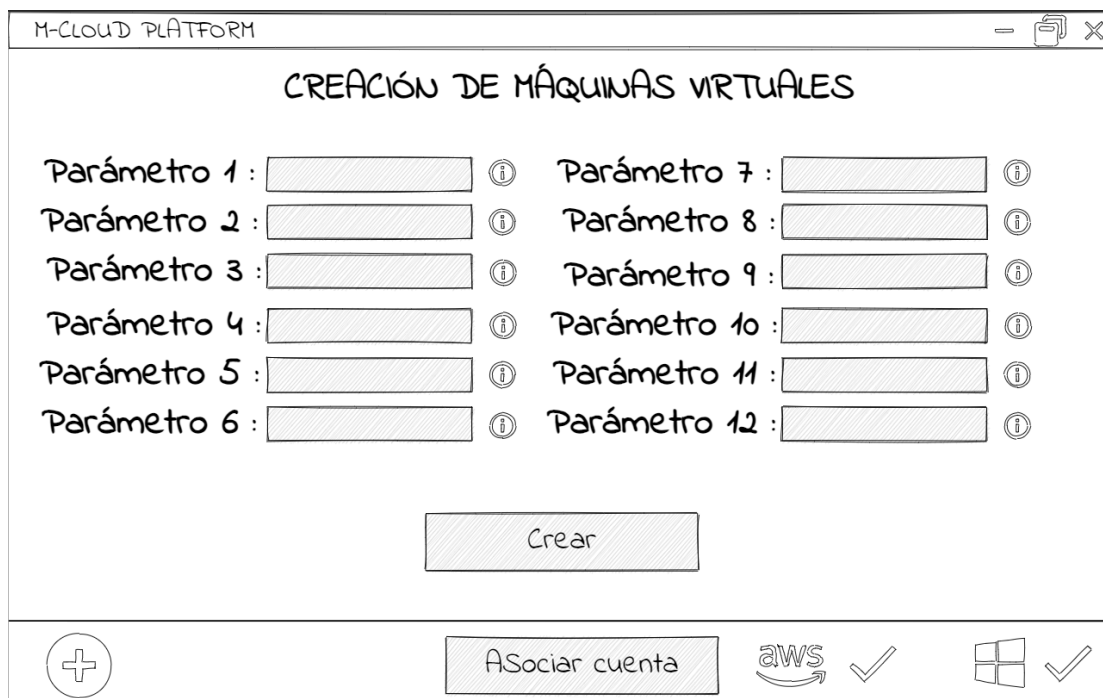


Fig. 3.4. Prototipo visual: creación de máquinas virtuales

Si pulsamos el botón de la parte inferior izquierda, el sistema cambiará a la panta-

lla de creación de máquinas virtuales, mostrada en la figura 3.4. Esta pantalla posee los parámetros editables a la hora de crear una máquina virtual.

The image shows a visual prototype of a web application window titled "M-CLOUD PLATFORM". The main heading inside the window is "Nombre máquina virtual (AWS)". Below this heading, there are 12 parameter labels, each followed by a text input field and an information icon (i in a circle). The parameters are arranged in two columns: "Parámetro 1" through "Parámetro 6" on the left, and "Parámetro 7" through "Parámetro 12" on the right. Below the parameter fields is a button labeled "Editar". At the bottom of the window, there is a navigation bar containing a plus icon in a circle, a button labeled "ASociar cuenta", the AWS logo, a checkmark, a four-pane window icon, and another checkmark.

Fig. 3.5. Prototipo visual: visualización de parámetros

Si en vez de crear una máquina virtual, se pulsa en el botón de solicitar información de una máquina virtual, el sistema muestra la pantalla de visualización de parámetros, cuya disposición es muy similar a la de creación. Esta disposición se muestra en la figura 3.5.

## 4. DESARROLLO E IMPLEMENTACIÓN

En este capítulo se comentan todas las decisiones relacionadas con el desarrollo y la implementación de la solución. Primero, se explican todas las herramientas que se utilizan durante el desarrollo, justificando todas las decisiones tomadas. Tras esto se realiza un listado de los componentes desarrollados, así como las funciones más importantes del sistema. 3

### 4.1. Herramientas y lenguajes de programación

Durante este epígrafe, se explican todas las herramientas, los lenguajes de programación escogidos y las librerías utilizadas durante el desarrollo del proyecto.

Para realizar el control de versiones se planteó dos opciones principales. Ambas son herramientas basadas en Git. La primera era utilizar Bitbucket, debido a la experiencia adquirida por el autor en sus prácticas en empresas. La otra opción era utilizar la herramienta Github, una de las más famosas a nivel mundial de repositorios públicos y con la cual el autor también poseía experiencia debido a todos los trabajos realizados durante la carrera universitaria.

Finalmente, se decidió utilizar Github debido a uno de los deseos de dejar el proyecto como un repositorio público donde todo el mundo sea capaz de desarrollar la idea y continuar su evolución.

Para elegir el lenguaje de programación también se planteó dos opciones. La primera era utilizar Python, el lenguaje utilizado durante casi toda la carrera universitaria del autor. La segunda era utilizar C# y el framework .NET, el lenguaje utilizado durante las prácticas empresariales del autor.

Esta decisión fue más complicada, ya que lo más cómodo era utilizar Python gracias a la gran comunidad que existe y a la gran cantidad de librerías que existen. Sin embargo, en C# ya se poseía experiencia sobre la gestión de recursos en la nube como AWS mediante código. Finalmente, se decidió utilizar Python debido a las librerías que facilitaban en gran medida el desarrollo de las funcionalidades.

A nivel de interfaz visual, se ha escogido utilizar ElectronJS, el cual se basa en NodeJS. Este marco de trabajo está especializado en el desarrollo de aplicaciones de escritorio utilizando tecnologías web. Por esta razón, los lenguajes de programación utilizados son HTML, CSS, JavaScript y Python. Los tres primeros apoyan e intervienen la parte visual del sistema, Python para desarrollar las funciones relacionadas con los servicios en la nube y JavaScript como conexión entre la parte visual y las funciones.

Debido al uso de diferentes lenguajes de programación, el entorno de desarrollo pro-



gramado (IDE - Integrated Development Environment) más adecuado y más flexible era “Visual Studio Code”. Además, es posible conectar directamente el entorno de trabajo con el repositorio del código, agilizando las acciones relacionadas con el control de versiones.

En cuanto a librerías utilizadas, se ha utilizado un paquete en NodeJS denominado “PythonShell” que permite ejecutar código de Python desde una llamada de JavaScript. A nivel funcional, en Python, se han utilizado las librerías proporcionadas por los propios proveedores de servicios en la nube. En el caso de AWS, ponen a disposición la librería Boto3 y toda la documentación[11]. En el caso de Azure, tienen una gran cantidad de bibliotecas para el manejo de los recursos mediante código Python (Azure SDK for Python)[12]. También se ha utilizado la librería “sys”, nativa de Python, para permitir el paso de parámetros en las llamadas desde JavaScript.

## 4.2. Componentes

A nivel de estructura, se pueden diferenciar cuatro componentes principales en el proyecto. Estos se han dividido según la funcionalidad y el objetivo de su presencia. En algunos de los componentes, existen una gran cantidad de archivos que conforman el componente en sí, esto quiere decir que cada componente no es un único fichero. Los cuatro componentes son:

- Interfaz visual: está compuesta por el archivo .html y todos los .css que existen para dar una estructura y formato a toda la interfaz visual. El archivo .html se encarga de poner la estructura y el esqueleto de la interfaz, mientras que los archivos .css indican la apariencia y la posición de cada elemento de la interfaz.
- Funciones de la interfaz visual: está compuesta por los archivos .js encargados de otorgar procesamiento y funcionalidades a la interfaz visual. Además, también sirve como almacén de todas las variables necesarias para el correcto funcionamiento de la aplicación. Las funcionalidades que se han implementado en este componente son las siguientes:
  - Otorgar lógicas y acciones a los botones de la interfaz visual.
  - Comprobar si se ha asociado alguna cuenta.
  - Solicitar la información de las máquinas virtuales.
  - Otorgar la funcionalidad de las pestañas en la aplicación.
  - Otorgar el comportamiento de un modal a los elementos deseados.
  - Cargar la información recogida de los proveedores de servicios en los correspondientes campos.
- Funciones relacionadas con los proveedores de servicio: está compuesta por todos los archivos ejecutables programados en Python para permitir todas las acciones

deseadas en el proveedor. Cada funcionalidad posee su archivo correspondiente a AWS y Azure. Estos archivos realizan las siguientes funciones:

- Comprobar si las credenciales introducidas son correctas.
  - Crear una máquina virtual con los datos solicitados.
  - Recoger todas las máquinas virtuales de una región.
  - Recoger todas las regiones disponibles en la cuenta asociada.
  - Iniciar una máquina virtual parada.
  - Parar una máquina virtual en marcha.
  - Terminar (borrar) una máquina virtual.
- Conexión entre las funciones de la interfaz visual y las funciones relacionadas con los proveedores de servicios: es el único componente compuesto por un único archivo (nombrado como `python.js`) el cual se encarga de crear las funciones que llaman a las funciones relacionadas con los proveedores de servicios y recogen los datos que mandan estos archivos.

### **4.3. Desarrollo de las funcionalidades**

En este epígrafe, se comenta el desarrollo de las funcionalidades implementadas relacionadas con el proveedor de servicios. Solo se va a comentar estas funcionalidades, ya que todo el desarrollo relacionado con la interfaz visual y sus funciones pueden variar en función de la interfaz que se desee implementar y de la misma manera la conexión entre las funcionalidades de interfaz con las funcionalidades del proveedor de servicio tampoco se comentan. Las funcionalidades elegidas se encuentran listadas en el epígrafe 4.2.

#### **4.3.1. Funcionalidad: comprobar credenciales**

Esta funcionalidad tiene como responsabilidad recibir todos los datos de la cuenta necesarios para realizar posteriormente las acciones sobre las máquinas virtuales y comprobar si estas credenciales son correctas o inválidas.

En AWS se reciben dos parámetros: la clave pública y la clave secreta; y en Azure se reciben cuatro parámetros: el identificador del cliente, el identificador del inquilino, el identificador de la subscripción y la clave secreta del cliente. Como salida, en ambos casos, se manda el mensaje “Buena” en caso de ser válidas las credenciales. En caso contrario, se manda el mensaje “Mala”. Si ha ocurrido algún error, manda el mensaje “Error: ” seguido del error producido. Toda esta información viene recogida en la tabla 4.1.

	AWS	Azure
Entrada	Clave pública Clave secreta	ID del cliente ID del inquilino ID de la subscripción Clave secreta del cliente
Salida	Buena (si las credenciales son validas) Mala (si las credenciales son inválidas) Error: ... (si ha saltado algún error)	

TABLA 4.1. TABLA DE ENTRADAS Y SALIDAS: COMPROBAR CREDENCIALES

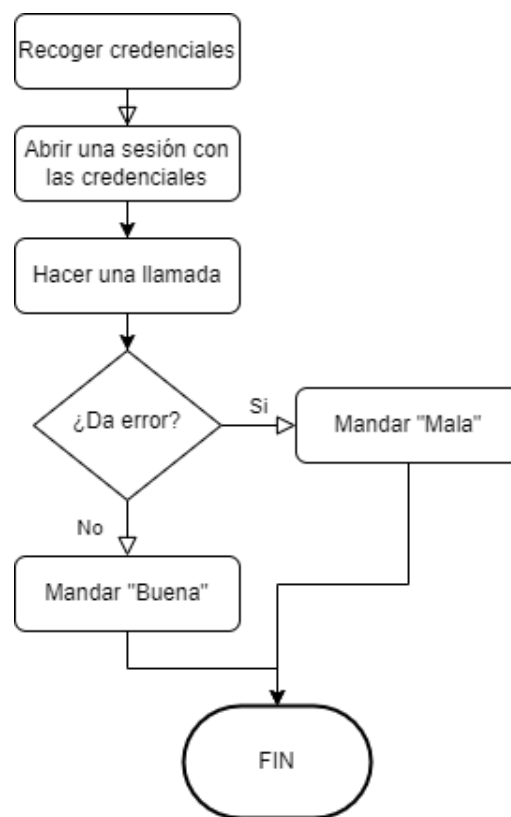


Fig. 4.1. Diagrama de flujo: comprobación de credenciales

Para la explicación del código, en ambos proveedores se mantiene la misma lógica. Esta lógica se viene representada en el diagrama de flujo mostrado en la figura 4.1.

#### 4.3.2. Funcionalidad: crear máquina virtual

Esta funcionalidad tiene como responsabilidad crear una máquina virtual según los datos introducidos. Se espera que esta acción se vea reflejada en la página del proveedor en la que se ha creado la máquina virtual.

En ambos proveedores, aparte de las credenciales de sesión, se pasan como parámetros el nombre de la máquina, la región, la imagen y el tipo de máquina de la máquina que se desea crear. Como salida, en este caso solo se manda un mensaje de “Máquina creada” en caso de crearse correctamente. Si da algún error durante el proceso de creación de la máquina, se manda un mensaje, el cual comienza con “Error: ” y con el error ha saltado. Toda esta información viene recogida en la tabla 4.2.

	AWS	Azure
Entrada	Clave pública Clave secreta Región ID de la imagen Tipo de máquina Nombre de la máquina virtual	ID del cliente ID del inquilino ID de la subscripción Clave secreta del cliente Nombre de la máquina virtual Tipo de máquina Región Imagen
Salida	Máquina creada (si se crea correctamente) Error (si da algún error durante el proceso)	

TABLA 4.2. TABLA DE ENTRADAS Y SALIDAS: CREAR MÁQUINA VIRTUAL

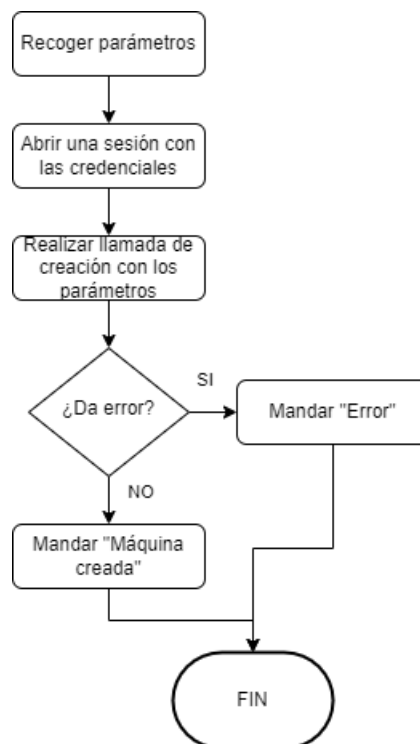


Fig. 4.2. Diagrama de flujo: crear máquina virtual (AWS)

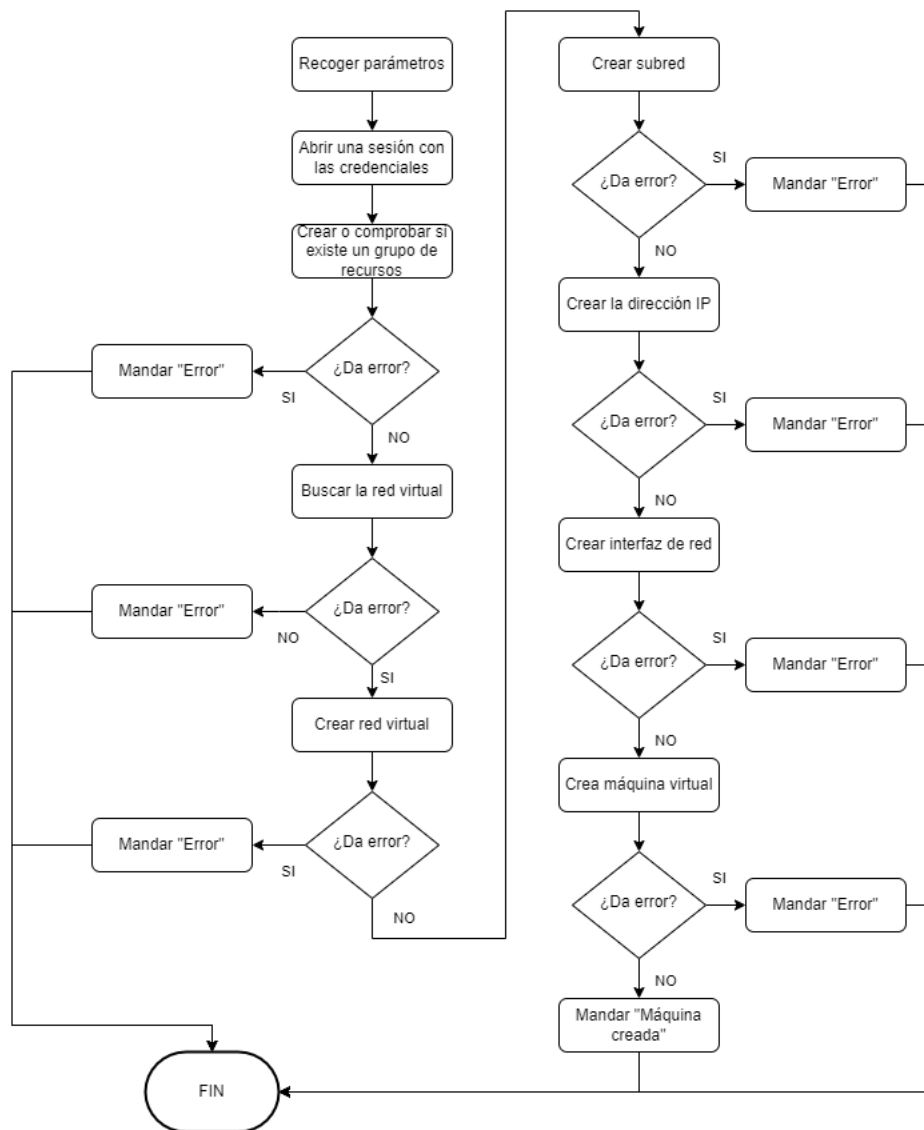


Fig. 4.3. Diagrama de flujo: crear máquina virtual (Azure)

Para la explicación de código, en esta funcionalidad el diagrama de flujo difiere un poco entre los proveedores. En AWS, la creación de máquinas virtuales se simplifica en una sola llamada con los parámetros seleccionados. Sin embargo, en Azure, es necesario previamente tener un grupo de recursos, una red virtual, una subred, una dirección IP (Internet Protocol) y una interfaz de red para poder permitir la creación de una máquina virtual. Estos diagramas de flujo se pueden observar en las figuras 4.2 y 4.3.

#### 4.3.3. Funcionalidad: recoger máquinas virtuales

Esta funcionalidad tiene como responsabilidad recoger todos los datos de todas las máquinas virtuales que existan en la región indicada.

	AWS	Azure
Entrada	Clave pública Clave secreta Región	ID del cliente ID del inquilino ID de la subscripción Clave secreta del cliente Región
Salida	Por cada máquina virtual y cada atributo de cada una: Manda el nombre del atributo y su valor Error: ... (si ha saltado algún error))	

TABLA 4.3. TABLA DE ENTRADAS Y SALIDAS: RECOGER MÁQUINAS VIRTUALES

En ambos casos, aparte de los credenciales, como entrada se recibe como parámetro también la región. Como salida, en ambos casos, por cada máquina virtual existente, se manda una línea por cada atributo con el nombre del atributo y el valor del mismo. Si ha ocurrido algún error, manda el mensaje “Error: ” seguido del error producido. Toda esta información viene recogida en la tabla 4.3.

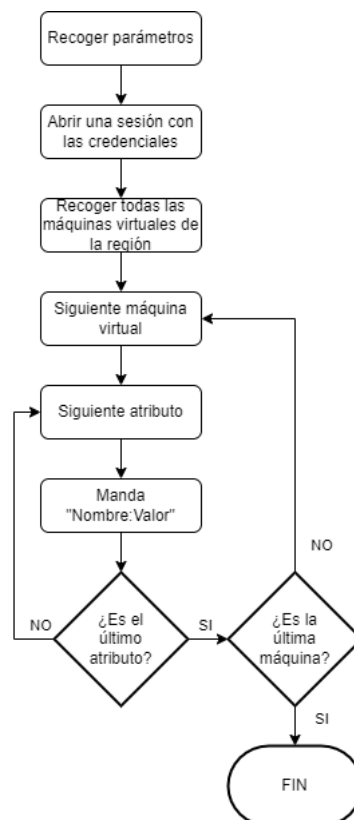


Fig. 4.4. Diagrama de flujo: recoger máquinas virtuales (AWS)

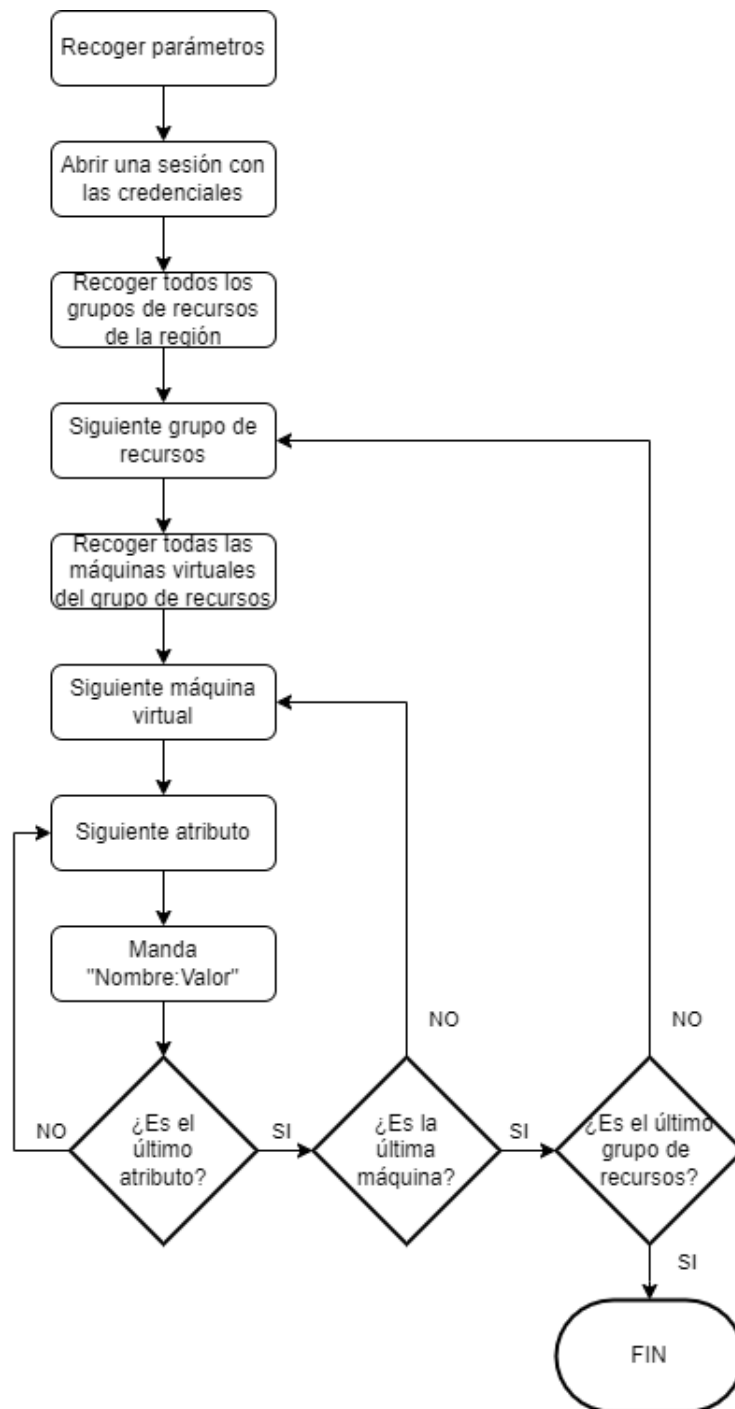


Fig. 4.5. Diagrama de flujo: recoger máquinas virtuales (Azure)

Para la explicación del código, en AWS existe una llamada que recoge todas las máquinas virtuales de la región. Sin embargo, en Azure, hay que hacerlo por cada grupo de recursos, por ello en este caso hay que hacer otro bucle por cada grupo de recursos de la región seleccionada. Esto viene representado en los diagramas de flujo mostrados en la figura 4.4 y la figura 4.5.

#### 4.3.4. Funcionalidad: recoger regiones

Esta funcionalidad tiene como responsabilidad recoger todas las regiones disponibles en la cuenta asociada. Esto es necesario para poder elegir una región en la que sea posible todas las demás funcionalidades.

	AWS	Azure
Entrada	Clave pública Clave secreta	ID del cliente ID del inquilino ID de la subscripción Clave secreta del cliente
Salida	Por cada región: Manda el nombre de la misma Error: ... (si ha saltado algún error))	

TABLA 4.4. TABLA DE ENTRADAS Y SALIDAS: RECOGER REGIONES

En esta funcionalidad, como entrada, ambos reciben únicamente las credenciales. Como salida, también comparten la misma, donde por cada región, se imprime una vez su nombre. Si ha ocurrido algún error, manda el mensaje “Error: ” seguido del error producido. Toda la información viene recogida en la tabla 4.4.



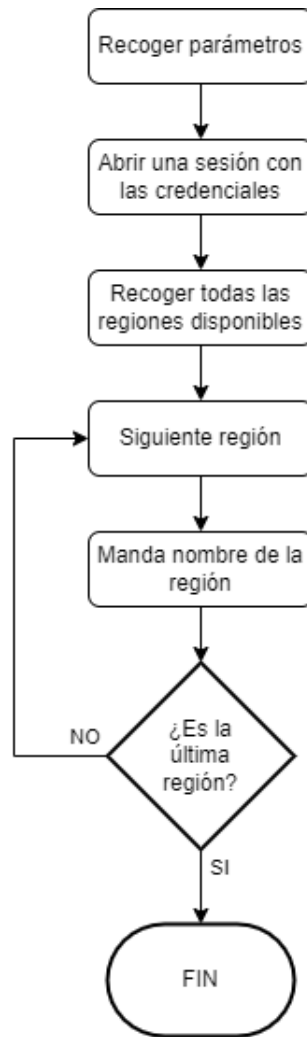


Fig. 4.6. Diagrama de flujo:  
recoger regiones

Para la explicación del código, en esta funcionalidad ambos siguen la misma lógica. Esta lógica viene representada en el diagrama de flujo mostrado en la figura 4.6.

#### 4.3.5. Funcionalidad: iniciar máquina virtual

Esta funcionalidad tiene como responsabilidad iniciar o encender una máquina virtual deseada que se encuentre parada.

En esta funcionalidad, como entrada, en AWS se recogen la región y el identificador de la máquina virtual para poder poner en marcha la máquina virtual. Mientras tanto, en Azure, se necesita el nombre de la máquina virtual y el grupo de recursos al que está asociado. Como salida, ambos tienen la misma, si todo se ha realizado sin ningún tipo de problema manda el mensaje “Máquina virtual reanudada”. Si ha ocurrido algún error, manda el mensaje “Error: ” seguido del error producido.

	AWS	Azure
Entrada	Clave pública Clave secreta Región ID de la máquina virtual	ID del cliente ID del inquilino ID de la suscripción Clave secreta del cliente Nombre de la máquina virtual Grupo de recursos
Salida	Máquina virtual reanudada (si todo ha ocurrido bien) Error: ... (si ha saltado algún error)	

TABLA 4.5. TABLA DE ENTRADAS Y SALIDAS: INICIAR MÁQUINA VIRTUAL

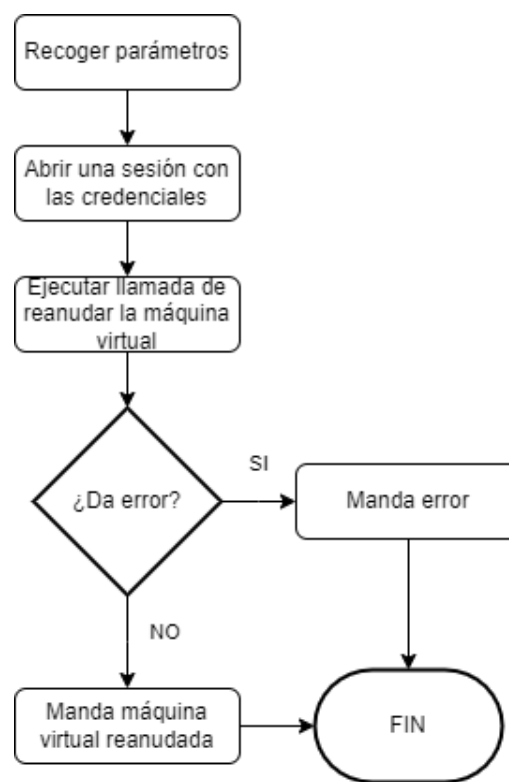


Fig. 4.7. Diagrama de flujo: iniciar máquina virtual

Para la explicación del código, en esta funcionalidad ambos siguen la misma lógica. Esta lógica viene representada en el diagrama de flujo mostrado en la figura 4.7.

#### 4.3.6. Funcionalidad: parar máquina virtual

Esta funcionalidad tiene como responsabilidad parar o apagar una máquina virtual deseada que se encuentre encendida.

En esta funcionalidad, como entrada, en AWS se recogen la región y el identificador

	AWS	Azure
Entrada	Clave pública Clave secreta Región ID de la máquina virtual	ID del cliente ID del inquilino ID de la suscripción Clave secreta del cliente Nombre de la máquina virtual Grupo de recursos
Salida	Máquina virtual parada (si todo ha ocurrido bien) Error: ... (si ha saltado algún error)	

TABLA 4.6. TABLA DE ENTRADAS Y SALIDAS: PARAR MÁQUINA VIRTUAL

de la máquina virtual para poder parar la máquina virtual. Mientras tanto, en Azure, se necesita el nombre de la máquina virtual y el grupo de recursos al que está asociado. Como salida, ambos tienen la misma, si todo se ha realizado sin ningún tipo de problema manda el mensaje “Máquina virtual parada”. Si ha ocurrido algún error, manda el mensaje “Error: ” seguido del error producido.



Fig. 4.8. Diagrama de flujo: parar máquina virtual

Para la explicación del código, en esta funcionalidad ambos siguen la misma lógica. Esta lógica viene representada en el diagrama de flujo mostrado en la figura 4.8.

#### 4.3.7. Funcionalidad: terminar máquina virtual

Esta funcionalidad tiene como responsabilidad borrar o eliminar una máquina virtual deseada.

	AWS	Azure
Entrada	Clave pública Clave secreta Región ID de la máquina virtual	ID del cliente ID del inquilino ID de la suscripción Clave secreta del cliente Nombre de la máquina virtual Grupo de recursos
Salida	Máquina virtual eliminada (si todo ha ocurrido bien) Error: ... (si ha saltado algún error)	

TABLA 4.7. TABLA DE ENTRADAS Y SALIDAS: TERMINAR MÁQUINA VIRTUAL

En esta funcionalidad, como entrada, en AWS se recogen la región y el identificador de la máquina virtual para poder eliminar la máquina virtual. Mientras tanto, en Azure, se necesita el nombre de la máquina virtual y el grupo de recursos al que está asociado. Como salida, ambos tienen la misma, si todo se ha realizado sin ningún tipo de problema manda el mensaje “Máquina virtual parada”. Si ha ocurrido algún error, manda el mensaje “Error: ” seguido del error producido.



Fig. 4.9. Diagrama de flujo: terminar máquina virtual(AWS)

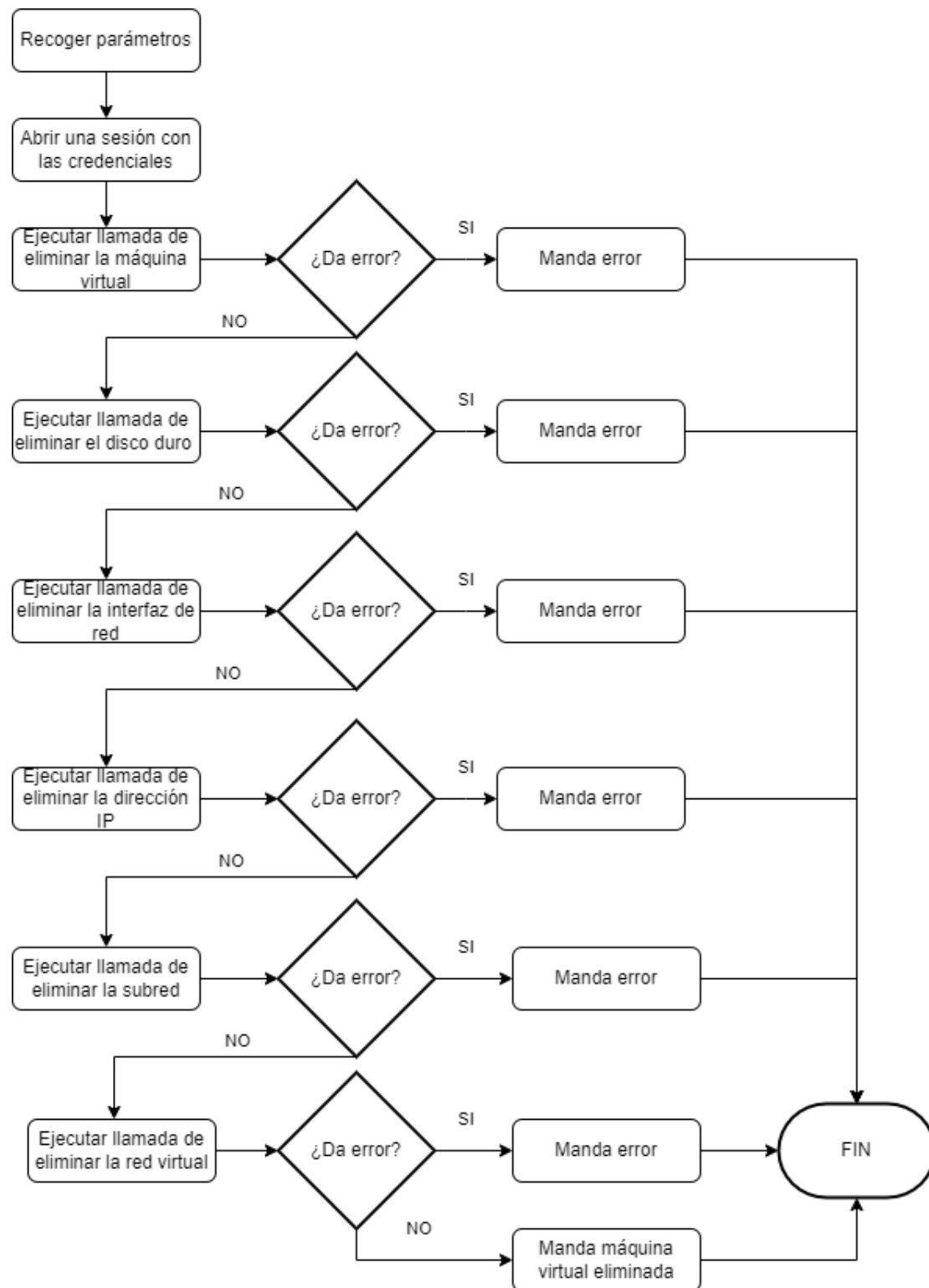


Fig. 4.10. Diagrama de flujo: terminar máquina virtual(Azure)

Para la explicación del código, en AWS, se sigue una lógica muy parecida a la de reanudar o pausar la máquina virtual pero con la llamada de terminar. Sin embargo, en Azure, hay que borrar todos los recursos ligados a la máquina virtual de forma ordenada para evitar consumir créditos de la suscripción. Ambas lógicas se pueden observar en los diagramas mostrados en la figura 4.9 y la figura 4.10.

## 5. EVALUACIÓN

En este capítulo se recoge toda la información sobre las pruebas y controles sobre la plataforma. Por un lado, tenemos los controles de calidad mediante pruebas unitarias que aseguran la calidad y se determina si se ha conseguido lo que se deseaba obtener. Por otro lado, se ha llevado un control sobre los requisitos para determinar si han sido cumplidos o no y poner fin el ciclo de desarrollo del proyecto.

### 5.1. Pruebas unitarias

Para determinar el correcto funcionamiento de todas las funcionalidades vistas en el epígrafe 4.3, se han desarrollado un conjunto de pruebas unitarias. Las pruebas unitarias son una parte importante en un proyecto de desarrollo, ya que aporta muchos beneficios. Entre ellos podemos destacar:

- **Detección de errores:** si al introducir unos datos no se obtiene la salida esperada, significa que hay algún error en el desarrollo.
- **Facilita el mantenimiento:** las pruebas unitarias contribuyen a la creación de código, ya que cuando se realizan cambios o mejoras en el futuro, las pruebas unitarias pueden asegurar que las modificaciones no rompan las funcionalidades existentes.
- **Documentación:** entender las pruebas unitarias ayuda al desarrollador a comprender mejor la funcionalidad que debe tener el sistema en ese estado concreto.
- **Facilita la depuración:** si ocurre algún error, es fácil localizarlo y dar con la raíz del problema, ya que se sabe perfectamente qué unidad de la batería de pruebas ha dado error.

En total se han desarrollado treinta pruebas para verificar el desarrollo de las todas las funcionalidades, donde cada uno tiene un objetivo en específico. Cada prueba tiene un identificador asociado, que sigue el formato TNúmero de la pruebaAWS o AZR, por ejemplo T01AWS o T01AZR.

Identificador	Descripción
T01AWS	Inserción correcta de credenciales en AWS.
T02AWS	Inserción incorrecta de credenciales en AWS.
T03AWS	Caso de error en inserción de credenciales en AWS.
T04AWS	Recogida de máquinas virtuales en AWS.
T05AWS	Caso de error en la recogida de máquinas virtuales en AWS.
T06AWS	Recogida de regiones en AWS.
T07AWS	Caso de error en la recogida de regiones en AWS.
T08AZR	Inserción correcta de credenciales en Azure.
T09AZR	Inserción incorrecta de credenciales en Azure.
T10AZR	Caso de error en inserción de credenciales en Azure.
T11AZR	Recogida de máquinas virtuales en Azure.
T12AZR	Caso de error en la recogida de máquinas virtuales en Azure.
T13AZR	Recogida de regiones en Azure.
T14AZR	Caso de error en la recogida de regiones en Azure.
T15AWS	Encendido de una máquina virtual en AWS.
T16AWS	Caso de error en el encendido de una máquina virtual en AWS.
T17AWS	Apagado de una máquina virtual en AWS.
T18AWS	Caso de error en el apagado de una máquina virtual en AWS.
T19AZR	Encendido de una máquina virtual en Azure.
T20AZR	Caso de error en el encendido de una máquina virtual en Azure.
T21AZR	Apagado de una máquina virtual en Azure.
T22AZR	Caso de error en el apagado de una máquina virtual en Azure.
T23AWS	Creación de una máquina virtual en AWS.
T24AWS	Caso de error en la creación de una máquina virtual en AWS.
T25AZR	Creación de una máquina virtual en Azure.
T26AZR	Caso de error en la creación de una máquina virtual en Azure.
T27AWS	Eliminación de una máquina virtual en AWS.
T28AWS	Caso de error en la eliminación de una máquina virtual en AWS.
T29AZR	Eliminación de una máquina virtual en Azure.
T30AZR	Caso de error en la eliminación de una máquina virtual en Azure.

TABLA 5.1. TABLA DE PRUEBAS UNITARIAS

Todas las pruebas unitarias desarrolladas se encuentran descritas en la tabla 5.1.

## 5.2. Cumplimiento de requisitos

Para finalizar el ciclo de desarrollo es necesario asegurarse de que se han cumplido con todas las expectativas y necesidades planteadas. Para ello, se va a realizar un listado de verificación de requisitos que ayuda a evaluar exhaustivamente si se cumple con todos los

requisitos establecidos. A través de este listado se puede confirmar que todos los aspectos claves han sido considerados y abordados de manera efectiva.

Es recomendable que este procesado de verificación lo haga una persona o un grupo de personas fuera del equipo de desarrollo. Sin embargo, en este caso, lo va a realizar la misma persona que realizó el desarrollo de la aplicación.

ID	Estado	Comentarios
RF-01	Completado	
RF-02	Completado	
RF-03	Completado	
RF-04	Completado	
RF-05	Completado	
RF-06	Completado	
RF-07	Completado	
RF-08	Completado	
RF-09	Completado	
RF-10	Completado	
RF-11	Completado	
RF-12	Completado	
RF-13	Completado	
RF-14	Completado parcialmente	Se han omitido muchos atributos y se han dejado los más básicos.
RNF-01	Completado	Se ha realizado la compilación y hay versiones para Linux y Windows
RNF-02	Completado	Las librerías no funcionan sin conexión a internet.

TABLA 5.2. LISTADO DE VERIFICACIÓN DE REQUISITOS

El listado de verificación de requisitos se puede observar en la tabla 5.2.



## **6. GESTIÓN DEL PROYECTO**

Durante este capítulo final, se comentan aspectos generales sobre el proyecto. Estos aspectos forman el proceso previo y posterior, a la vez de independiente, al desarrollo de la aplicación. En total se tratan cuatro temas principales: la planificación, el presupuesto, el marco legal y el impacto socio-económico.

A nivel de la metodología utilizada, se ha utilizado una metodología tradicional en cascada, donde tras cada fase viene la siguiente sin ciclos ni repeticiones. Se podría haber utilizado una metodología ágil, sin embargo, al ser una sola persona actuando como cliente, mánager y trabajador, no tendría mucho sentido utilizarla.

### **6.1. Planificación**

Para realizar una correcta planificación se realiza una división del proyecto en fases. En total se han detectado cuatro fases: estado del arte, el diseño, desarrollo y post-procesado. Se planea comenzar el proyecto el 1 de junio y acabar el 1 de septiembre. También se ha creado un diagrama de Gantt con las duraciones estimadas de cada fase.

#### **6.1.1. Fase: estado del arte**

En esta fase, se recoge todo el tiempo invertido en estudiar las definiciones de servicios en la nube, plataformas multi-cloud, competidores y trabajos académicos sobre el tema. Para realizar esta fase, se ha estimado una duración de cuatro semanas. Esta fase contiene las siguientes tareas:

- Definiciones generales del proyecto (Una semana).
- Estudio de aplicaciones similares (Una semana).
- Lectura y análisis de trabajos académicos similares (Dos semanas).

#### **6.1.2. Fase: diseño**

En esta fase, se recoge todo el tiempo que se va a dedicar en el proceso de extracción de requisitos y casos de uso y en el diseño de la arquitectura de la aplicación. Para realizar esta fase, se ha estimado una duración de cuatro semanas. Esta fase contiene las siguientes tareas:

- Extracción de requisitos (Una semana).

- Extracción de casos de uso y matriz de trazabilidad (Una semana).
- Diseño de la arquitectura e interfaz visual (Dos semanas).

### **6.1.3. Fase: desarrollo**

En esta fase, se recoge todo el tiempo que se va a dedicar en el desarrollo de todas las funcionalidades de la aplicación. Para realizar esta fase, se ha estimado una duración de dos semanas. Esta fase contiene las siguientes tareas:

- Preparación de las herramientas y el entorno de trabajo (Dos días)
- Estudio de las librerías a utilizar (Dos días).
- Desarrollo de la aplicación (Una semana).
- Depuración y pruebas de las funcionalidades (Tres días).

### **6.1.4. Fase: post-procesado**

En esta fase, se recoge todo el tiempo que se va a dedicar en las revisiones tras el desarrollo y la escritura de esta memoria. Para realizar esta fase, se ha estimado una duración de dos semanas. Esta fase contiene las siguientes tareas:

- Comprobación de cumplimientos de requisitos (Un día).
- Compilación y exportación de la aplicación (Dos días).
- Escritura de la memoria (Once días).

### **6.1.5. Diagrama de Gantt**

Con todas las fases definidas y limitadas en el tiempo, se ha elaborado un diagrama de Gantt que representa de manera visual la planificación del proyecto. El diagrama resultante se muestra en la figura 6.1. Este diagrama ha sido creado utilizando la herramienta “Microsoft Project” mediante el aula virtual proporcionada por la Universidad Carlos III de Madrid.

Se puede observar como sobra una semana hasta el fin del proyecto, la cual se planea utilizar para comunicar y revisar el documento con el tutor.

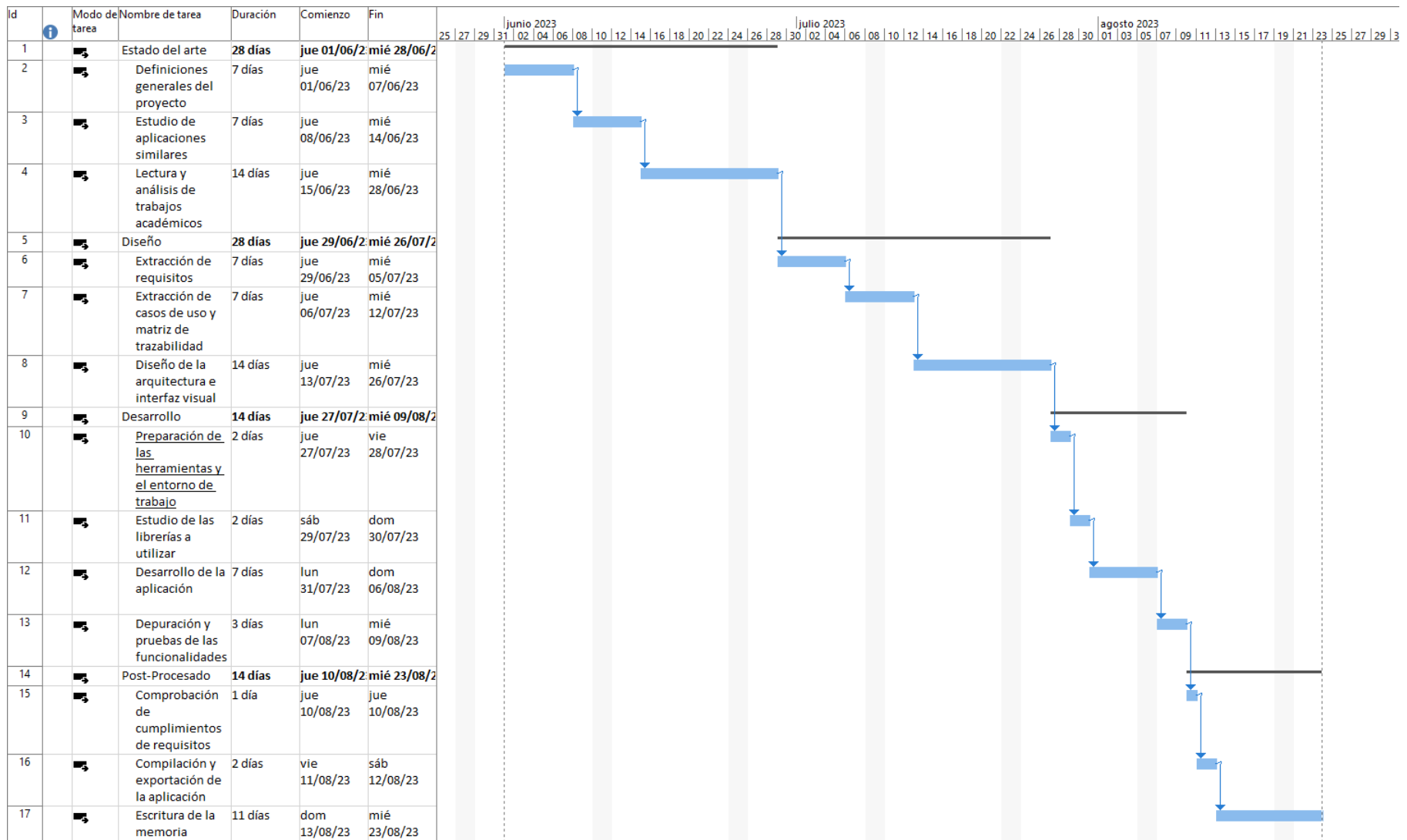


Fig. 6.1. Diagrama de Gantt.

## **6.2. Presupuesto**

Para calcular el presupuesto utilizado durante el proyecto, se ha desglosado en dos costes principales: el coste del personal y el coste del material. Sumando ambos se obtiene el coste total del proyecto estimado.

En total, el presupuesto estimado para el proyecto es 8.650,50 €.

### **6.2.1. Coste del personal**

Para realizar este cálculo, se ha establecido tres meses (30 días por mes) como tiempo trabajado (estimado en el epígrafe 6.1). Suponiendo una dedicación de 5 horas diarias los siete días de la semana, se obtiene un total de 600 horas en total.

Una vez calculadas las horas invertidas en el proyecto, se procede a calcular el coste del personal. Para conseguir esto, se necesita determinar el sueldo por hora. El sueldo se establece en función de la situación laboral actual del autor, siendo este un valor representativo del estudiante en este momento. Este valor es de 13 € brutos por hora.

Uniando el dato de las horas totales y el sueldo por hora, se obtiene un total de 7.800 € brutos en sueldos del personal.

### **6.2.2. Coste material**

En este apartado, se deben tener en cuenta muchos costes derivados del material utilizado para el desarrollo del proyecto. Para ello se han dividido en los siguientes:

- Coste de los servicios en la nube durante las pruebas: debido al uso de los servicios de la capa gratuita de AWS y utilizando el crédito gratuito prestado por Azure, se estima un coste de 0 €. Sin embargo, se va a añadir un coste estimado de 10 € al total en concepto de servicios en la nube. Este valor ha sido obtenido utilizando los costes estimados por los propios servicios de la nube en el caso de no tener crédito gratuito.
- Coste de licencias: todo el software utilizado es de uso gratuito, como Visual Studio Code o Github. La memoria se ha escrito utilizando el portal Overleaf, por lo que no hay un gasto real de licencia. Se podría incluir en este apartado el coste de la licencia del Windows 10 utilizado en la computadora que se va a utilizar para programar, sin embargo, también se ha obtenido de manera gratuita por ser estudiante de la Universidad Carlos III de Madrid. Por ello, el coste estimado es de 0 €.
- Coste de los dispositivos: Para realizar el proyecto se ha utilizado la computadora personal del autor, estimada en unos 800 €. Sin embargo, para el desarrollo de la aplicación no es necesario tener una computadora igual de potente que la usada. A pesar de ello, se añadirán los 800 € íntegros al presupuesto total de la aplicación.

- Gastos menores: en este grupo se incluyen gastos como la electricidad y el internet utilizados durante la elaboración del proyecto. Este valor es muy dependiente de la localización y las compañías contratadas, por ello se estima como un aumento del 5 % del coste total del proyecto.

En total, el coste material se estima en 850.50 €.

### **6.3. Marco legal**

En este epígrafe, se examinan los aspectos legales que influyen en el proyecto desarrollado. Se va a separar en dos puntos principales: la propiedad intelectual y la protección de datos.

En relación con la propiedad intelectual, todo ha sido desarrollado de manera propia, salvo las librerías, siendo estas de carácter público. El único problema que puede llegar a existir es el uso de los logos de AWS y Azure para identificar el origen de la máquina virtual. En el caso de publicar la aplicación se debería pedir permiso específico a dichas empresas para utilizar su logo en la aplicación. Por todo lo demás, no existen ningún riesgo de violar la propiedad intelectual de ninguna manera.

De cara a la protección de datos, el riesgo reside en el trato de contraseñas y claves de acceso a las cuentas de los usuarios en los proveedores de servicios. Sin embargo, como no se dispone de ningún sistema de ciberseguridad, la sesión no se guarda entre usos de aplicación. Con esto, se quiere recalcar que no se almacena ningún dato sensible de nadie en ningún momento. Únicamente se utiliza durante el uso de la aplicación, que al ser de carácter local no se conecta con el exterior. Al conectarse con el exterior, se utilizan los estándares de protección internos de las librerías otorgadas por los mismos proveedores de servicios. Por estas razones, no existe ningún riesgo que violar la “Ley Orgánica de Protección de Datos Personales y garantía de los derechos digitales” (LOPDGDD).

### **6.4. Impacto socio-económico**

A continuación se realiza el análisis del impacto socio-económico del proyecto. Para ello, se va a dividir en cuatro ámbitos principales: el económico, el social, el medioambiental y el ético. El impacto socio-económico que posee el proyecto actual con limitaciones es muy escaso y poco interesante. Por esta razón, el análisis se realiza teniendo en cuenta el proyecto al completo, sin limitaciones.

- Impacto económico: desde una perspectiva económica, el proyecto presenta la oportunidad de generar beneficios significativos. La implementación exitosa de esta plataforma multi-cloud podría optimizar la utilización de recursos y reducir costos operativos para las empresas que dependen de múltiples proveedores de servicios

en la nube. Esto podría liberar recursos financieros y mejorar la eficiencia empresarial. Además, al simplificar la gestión de múltiples plataformas, se podrían liberar recursos humanos y técnicos, permitiendo a las empresas centrarse en su área de negocio principal.

- **Impacto social:** en el ámbito social, el proyecto también tiene el potencial de causar un impacto significativo. Al unificar los recursos de diferentes proveedores en una única aplicación, se podrían experimentar mejoras palpables en la accesibilidad y la usabilidad de los servicios en la nube. Esto podría llevar a una mayor adopción de tecnologías basadas en la nube por parte de empresas y usuarios, lo que a su vez podría impulsar la colaboración y la innovación en una escala más amplia. Además, al simplificar la experiencia del usuario, el proyecto podría contribuir a la reducción de barreras de entrada tecnológicas.
- **Impacto medioambiental:** en lo que respecta al impacto medioambiental, el proyecto podría desempeñar un papel crucial en la promoción de la sostenibilidad y la conservación del entorno. Al permitir una administración más eficiente de los recursos en la nube, se podrían anticipar posibles reducciones en el consumo energético y en la huella ecológica asociada a la infraestructura digital. La consolidación de recursos también podría inspirar a proveedores y usuarios a adoptar prácticas menos contaminantes para el medio-ambiente en sus operaciones.
- **Consideraciones éticas:** finalmente, es esencial analizar el impacto ético del proyecto. La presencia de una aplicación como la del proyecto podría ejercer un impacto ético positivo en la industria de servicios en la nube. Al fomentar una mayor competencia entre los proveedores, la plataforma podría motivar mejoras en la calidad de los servicios y la innovación en general. Además, esta competencia podría impulsar una mayor transparencia en las políticas de los proveedores y mejorar la atención al cliente, generando un ambiente más equitativo y ético para los usuarios. La presión por mantener la confianza de los clientes y ganar una reputación positiva en el mercado podría llevar a una mayor responsabilidad en las prácticas empresariales, en línea con las expectativas éticas y las necesidades de los usuarios.

A todas estas consideraciones, hay que añadir que de manera indirecta fomenta el uso de servicios en la nube, haciéndolos más accesibles a cualquier usuario o empresa. Por esto, a estas consideraciones vistas, se podrían añadir todas las derivadas del uso de un sistema en la nube.

## 7. CONCLUSIONES Y TRABAJO FUTURO

En este capítulo se recogen todas las conclusiones sobre el trabajo, tanto generales como personales. Además, se añade un epígrafe sobre trabajos futuros, que consiste en todas las funcionalidades que no se ha llegado a abordar en este proyecto o expansiones para mejorar el mismo y se pueden realizar en algún futuro trabajo.

En el epígrafe de conclusiones general se repasan los objetivos del proyecto y si se han logrado completar satisfactoriamente o no. Por otro lado, en el epígrafe de conclusiones personales, se incluyen anotaciones personales, logros y conocimientos aprendidos durante el desarrollo del proyecto.

### 7.1. Conclusiones generales

Para empezar, antes de realizar cualquier tipo de conclusión o análisis, se debe comprobar si el proyecto ha cumplido todos los objetivos propuestos inicialmente en el epígrafe 1.2. Para poder realizar esta comprobación se comentan uno por uno cada objetivo:

1. Realizar un estudio de la importancia de los entornos multi-cloud mediante aplicaciones similares y estudios académicos. Estado: completado

A lo largo del capítulo 2, se habla de la importancia y ventajas y desventajas que poseen los entornos multi-cloud, una variedad de aplicaciones que encajan en el contexto y problema planteado y una recolección de trabajos académicos con sus correspondientes anotaciones.

2. Identificar los requisitos y necesidades de los usuarios finales para la utilización de una plataforma multi-cloud. Estado: completado parcialmente.

Se ha elaborado un listado de requisitos en el epígrafe 3.2. Sin embargo, este listado solo corresponde a una pequeña parte de las necesidades de un usuario final de la plataforma. Esta decisión fue tomada con el fin de limitar el alcance que posee el proyecto.

3. Diseñar la estructura y funcionamiento de la plataforma. Estado: completado.

Durante los epígrafes 3.2.5 y 3.3, se ha diseñado la estructura y los casos de uso de la plataforma, lo que ha facilitado en gran medida la etapa de desarrollo de esta.

4. Desarrollar la plataforma multi-cloud teniendo en cuenta las funcionalidades y requisitos identificados en los objetivos anteriores. Estado: completado.

A pesar de que esta información está ligada al desarrollo que se comenta durante el capítulo 4, la mejor manera de demostrar que se ha cumplido este objetivo se

localiza en el epígrafe 5.2, donde se realiza una revisión de todos los requisitos y funcionalidades para determinar si han sido implementados o no.

5. Realizar pruebas y evaluaciones de la plataforma multi-cloud para asegurar su correcto funcionamiento por parte del usuario final. Estado: completado.

Este objetivo se ha cumplido gracias a todas las pruebas unitarias desarrolladas que aseguran el correcto funcionamiento de la aplicación. Estas pruebas unitarias han sido comentadas en el epígrafe 5.1.

6. Enunciar todos los trabajos pendientes por hacer. Estado: completado.

Este objetivo ha sido completado en el epígrafe 7.3, donde se comenta todas las funcionalidades y trabajos no abordados en este proyecto.

Además de estos, también se planteó un objetivo secundario:

1. Implementación de algún modelo de Inteligencia Artificial dentro de la plataforma. Estado: no completado.

Se llegó a plantear en un inicio, pero no se ha conseguido desarrollar nada relacionado con la inteligencia artificial en la plataforma. Se ha planteado las ideas que se propusieron en el epígrafe 7.3.

Con este resumen del estado de los objetivos planteado, se puede confirmar que se ha conseguido obtener los objetivos del proyecto de forma satisfactoria.

## **7.2. Conclusiones personales**

A lo largo de este proceso de desarrollo, he tenido la oportunidad de explorar a fondo el mundo de los servicios en la nube, y quiero aprovechar este espacio para compartir mis reflexiones personales y algunas valiosas lecciones que he extraído de esta experiencia. Este proyecto no solo ha ampliado mi conocimiento en el campo, sino que también ha contribuido a mi crecimiento académico y profesional.

Para empezar, quiero volver a recalcar la gran importancia que supone la presencia de los servicios en la nube, el poder tener a disposición un servidor encendido todos los días a todas horas sin tener conocimientos de infraestructuras ni electrónica permite a cualquier empresa con un pequeño departamento de TIC ser capaz de mantener su propio servidor. Su potencial puede llegar a ser inmenso si se continúa desarrollando.

A nivel educativo he tenido la posibilidad de experimentar de primera mano la creación de una aplicación desde 0 hasta la creación del ejecutable o instalador. Por el camino se han reforzado muchos conocimientos relacionados con Python, como por ejemplo, las pruebas unitarias; a HTML/CSS, con la gestión de formularios y modales; a JavaScript,



permitiendo conectar una interfaz visual con los programas desarrollados en Python. También se ha ganado experiencia navegando y buscando información por una documentación de empresas como Amazon y Microsoft.

También se han dado situaciones y experiencias negativas, como la necesidad de limitar el proyecto, ya que se había estimado mal el tamaño del tema tratado. Aun así, de todas ellas se han obtenido lecciones importantes para en un futuro poder evitarlas, como por ejemplo realizar un análisis más exhaustivo del tema antes de realizar nada.

Todas estas lecciones van a ser de mucha utilidad en mi futuro, tanto profesional como personal, donde planeo desarrollar aplicaciones y videojuegos de manera constante en paralelo a mi trabajo y, de forma interna en el trabajo, aplicar los conocimientos aprendidos sobre la programación.

Una de las cosas que me hubiese gustado profundizar es en el tema de la inteligencia artificial, ya que me parece un tema apasionante y con mucho trayecto en el futuro. Sin embargo, no he conseguido llegar a abordar este tema en el tiempo planificado para el desarrollo de este proyecto.

### **7.3. Trabajo futuro**

Este epígrafe es uno de los más importantes de todo el proyecto. Gracias a este epígrafe el tema tratado no se queda estancado y esboza los caminos por los que se puede continuar avanzando para hacer evolucionar la idea. A nivel de trabajos posibles futuros, se pueden dividir en tres grupos principales: dentro del servicio de máquinas virtuales, otros servicios en la nube y las mejoras independientes a los servicios en la nube.

Dentro del servicio de máquinas virtuales, existen una gran cantidad de funcionalidades no implementadas debido a las limitaciones tanto de personal como de tiempo en este proyecto. Se van a enunciar las funcionalidades que hubiese sido deseable implementar:

1. Clonar una máquina virtual.
2. Editar los parámetros de una máquina virtual creada.
3. Implementar una terminal que conecte automáticamente con la máquina virtual creada.
4. Permitir personalizar el disco asociado a la máquina virtual.
5. Gestionar las IP asociadas a la máquina virtual.
6. Generar claves de acceso para poder conectarse con la máquina virtual de forma remota.
7. Ampliar el tipo de máquina y las imágenes para poder seleccionar cualquiera que se encuentre en el portal de AWS.

8. Gestionar los grupos de seguridad para bloquear o permitir ciertas IP.
9. Permitir realizar copias de seguridad.
10. Gestionar los gastos actuales y estimados de la máquina virtual.

El proyecto actual está limitado a únicamente el servicio de máquinas virtuales, pero este servicio es una minúscula parte de entre todos los servicios que ofrece un proveedor de servicio en la nube. En este epígrafe solo se va a enunciar algunos de los más importantes:

1. Bases de datos y derivados.
2. Contenedores.
3. Almacenes de datos.
4. Ejecución de código sin servidor.

De forma independiente de los servicios en la nube, se pueden aplicar también un gran número de mejoras al proyecto. Entre todas las posibilidades, se han destacado las siguientes:

1. Implementación de ciberseguridad y sesiones. (Actualmente se debe iniciar sesión cada vez que se abre la aplicación).
2. Ampliar el catálogo de proveedores de servicios en la nube a otros importantes como Google Cloud Platform.
3. Mejorar la interfaz visual para facilitar el uso de la aplicación.

La última mejora, incluida también en el grupo de mejoras independientes de los servicios en la nube, es la de aplicar modelos de inteligencia artificial. Se ha separado de la enumeración previa, ya que se desea ampliar su explicación. Al ser un objetivo secundario del proyecto, se llegó a plantear la idea, pero no se desarrolló. Esta idea consistía en, según el gasto que se produce cada día de una máquina virtual, calcular el gasto mensual aproximado mediante una predicción calculada mediante una serie temporal.

Otra idea fue analizar el consumo de una máquina virtual y que un modelo de inteligencia artificial analice y determine si hay algún tipo de máquina entre los disponibles de todos los proveedores que suponga un ahorro de gasto sin pérdida de potencia.

Tras recoger todas las ideas para trabajos futuros, se puede observar como existe una gran capacidad de avance y evolución en este tema. Tras numerosos trabajos enfocados en el mismo camino, se podrá llegar a un proyecto completo y con una importancia relevante en el mundo profesional.

## BIBLIOGRAFÍA

- [1] Microsoft. «¿Qué es la nube?» (2017), [En línea]. Disponible en: <https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-the-cloud> (Acceso: 10-06-2023).
- [2] IBM. «Ventajas del cloud computing.» (2019), [En línea]. Disponible en: <https://www.ibm.com/es-es/cloud/learn/benefits-of-cloud-computing> (Acceso: 10-06-2023).
- [3] J. Cohen. «4 Companies Control 67 % of the World's Cloud Infrastructure.» (2021), [En línea]. Disponible en: <https://www.pcmag.com/news/four-companies-control-67-of-the-worlds-cloud-infrastructure> (Acceso: 07-06-2023).
- [4] Q. Cao, D. G. Schniederjans y M. Schniederjans, «Establishing the use of cloud computing in supply chain management,» *Operations Management Research*, vol. 10, pp. 47-63, 2017.
- [5] S. A. Bello et al., «Cloud computing in construction industry: Use cases, benefits and challenges,» *Automation in Construction*, vol. 122, p. 103 441, 2021.
- [6] J. Gao, H. Wang y H. Shen, «Task failure prediction in cloud data centers using deep learning,» *IEEE transactions on services computing*, vol. 15, n.º 3, pp. 1411-1422, 2020.
- [7] H. K. Palani, R. Shelke y S. Pagi, «AIOps with Multi Cloud,» *Available at SSRN 4473251*, 2023.
- [8] P. Schmidt, «Literature Review on Multi Cloud Management,»
- [9] J. Hong, T. Dreibholz, J. A. Schenkel y J. A. Hu, «An overview of multi-cloud computing,» en *Web, Artificial Intelligence and Network Applications: Proceedings of the Workshops of the 33rd International Conference on Advanced Information Networking and Applications (WAINA-2019) 33*, Springer, 2019, pp. 1055-1068.
- [10] X. Wang, J. Cao e Y. Xiang, «Dynamic cloud service selection using an adaptive learning mechanism in multi-cloud computing,» *Journal of Systems and Software*, vol. 100, pp. 195-210, 2015.
- [11] AWS. «AWS SDK para Python (Boto3).» (), [En línea]. Disponible en: <https://aws.amazon.com/es/sdk-for-python/> (Acceso: 11-08-2023).
- [12] Azure. «Uso de las bibliotecas de Azure (SDK) para Python.» (), [En línea]. Disponible en: <https://learn.microsoft.com/es-es/azure/developer/python/sdk/azure-sdk-overview> (Acceso: 11-08-2023).

- [13] E. Nogueira, A. Moreira, D. Lucrédio, V. Garcia y R. Fortes, «Issues on developing interoperable cloud applications: definitions, concepts, approaches, requirements, characteristics and evaluation models,» *Journal of Software Engineering Research and Development*, vol. 4, n.º 1, pp. 1-23, 2016.

## **ANEXO A. REPOSITORIO GITHUB**

Enlace: <https://github.com/RafaelZhZh/MultiCloud>

En este repositorio se encuentra todo el código desarrollado durante el proyecto, con el historial de versiones correspondiente.

## ANEXO B. APLICACIÓN PORTABLE EN WINDOWS

Para usar la aplicación, se ha habilitado un enlace para la descarga del archivo: [https://drive.google.com/file/d/1n8o\\_k5QLfnGmdix0i5Az\\_BL4QcslAJye/view?usp=drive\\_link](https://drive.google.com/file/d/1n8o_k5QLfnGmdix0i5Az_BL4QcslAJye/view?usp=drive_link)

Para poder utilizar la aplicación es necesario tener Python 3 instalado. Además, se debe instalar de forma manual las siguientes librerías de Python mediante el comando “pip install {nombre de la librería}”:

- boto3
- azure.mgmt.resource
- azure.identity
- azure.mgmt.compute
- azure.mgmt.network
- azure.mgmt.resource
- azure.mgmt.subscription