



Universidad Internacional de La Rioja  
Escuela Superior de Ingeniería y Tecnología

Grado en Ingeniería Informática

# Evaluación del uso de la tecnología de contenedores para la creación de un laboratorio virtual

Web del proyecto: <https://github.com/hpereirab/TFG>

Trabajo fin de estudio presentado por:	Pereira Bogó, Héctor
Línea de investigación:	Ingeniería del Software
Director/a:	Soltero Domingo, Francisco José
Fecha:	29 de junio de 2021

## Resumen

Este Trabajo Fin de Grado tiene como objetivo evaluar el uso de Contenedores como alternativa para crear un entorno de formación, pongamos un aula virtual basada en esta tecnología, con el objetivo de poder ofrecer un entorno ligero, versátil y dinámico.

Tradicionalmente para este tipo de entornos se han utilizado equipos físicos ubicados en un aula de una institución educativa, o bien un entorno virtualizado de múltiples máquinas virtuales que ofrezcan un entorno controlado para la realización de las diversas unidades formativas.

A lo largo de este trabajo se evaluarán las distintas opciones existentes para finalmente seleccionar las más adecuadas al proyecto en virtud de las conclusiones que se puedan obtener del estudio de dichas alternativas. Finalmente se elaborará un caso de estudio en el que se puedan verificar las conclusiones obtenidas.

**Palabras clave:** Virtualización Contenedores Docker Linux Laboratorio Virtual

## Abstract

This Bachelor Project aims to evaluate the use of Containers as an alternative to create a training environment, let's put a virtual classroom based on this technology, with the aim of being able to offer a light, versatile and dynamic environment.

Traditionally, for this type of environment, physical equipment located in a classroom of an educational institution has been used, or a virtualized environment of multiple virtual machines that offered a control environment for the completion of the various training units.

Throughout this work, the different existing options will be evaluated to select the most appropriate for the project by virtue of the conclusions that can be obtained from the study of said alternatives. Finally, a case study will be developed in which it can verify the conclusions obtained.

**Keywords:** Virtualization Containers Docker Linux Virtual Laboratory

## Índice de contenidos

1. Introducción .....	9
1.1. Justificación del tema elegido.....	10
2. Estado del arte .....	12
2.1. Definiciones .....	12
Virtualización.....	12
Contenedores.....	14
Cloud Computing.....	15
2.2. Tipos de entornos .....	16
Entorno físico .....	17
Entorno virtualizado (MV).....	17
Entorno utilizando contenedores .....	17
Entorno virtualizado vs contenedores (Docker). .....	19
2.3. Trabajos similares .....	20
2.4. Tecnologías empleadas.....	21
Docker .....	21
Sistema Operativo Linux.....	21
Apache Guacamole .....	22
Nginx.....	22
Github.....	22
PostgreSQL .....	22
2.5. Resumen del estado del arte .....	23
3. Diseño de la propuesta .....	24
3.1. Objetivos .....	24

Objetivo principal .....	24
Objetivos secundarios .....	24
3.2. Metodología.....	24
Metodología de Desarrollo: Scrum .....	24
Historias de Usuario .....	25
Sprints.....	25
3.3. Diseño de la propuesta .....	27
Introducción .....	27
Arquitectura del Sistema.....	28
Casos de uso: Introducción .....	29
Casos de uso: Actores .....	30
Casos de uso y diagramas de Secuencia .....	31
Desarrollo de los casos de uso .....	31
Imágenes Docker .....	32
Comandos predefinidos .....	34
Portal de acceso Apache Guacamole .....	34
Política de usuarios .....	38
Persistencia de datos .....	38
Funcionalidades del Laboratorio.....	39
3.4. Pruebas .....	40
Introducción .....	40
Equipo de Pruebas.....	40
Entorno de pruebas.....	41
Pruebas de rendimiento del laboratorio virtual .....	42
Pruebas de consumo de recursos del laboratorio virtual .....	43
Evaluación del uso de la tecnología de contenedores para la creación de un aula virtual	

Pruebas de consumo de rendimiento de una máquina virtual .....	44
Pruebas de consumo de recursos de una máquina virtual.....	46
4. Comparativa y evaluación .....	48
4.1. Comparativa.....	48
4.2. Evaluación .....	51
4.3. Evaluación en entorno real .....	51
5. Conclusiones y trabajo futuro .....	56
5.1. Conclusiones .....	56
5.2. Líneas de trabajo futuro .....	57
Migración del sistema a un servicio en la nube. ....	57
Uso de LDAP para autenticación de usuarios .....	57
Integración del sistema en una plataforma de orquestación .....	58
Comparativa del sistema con diferentes sistemas operativos .....	58
Implementación de un sistema piloto en un entorno educativo. ....	58
Integración del sistema en una plataforma educativa online. ....	58
Referencias bibliográficas.....	59
Índice de acrónimos .....	65
Anexo A. Tablas de Historias de Usuario .....	66

## Índice de figuras

Estructura de una Virtualización Nativa .....	13
Estructura de una Virtualización Alojada o “Hosted Virtualization” .....	13
Tendencia de las aplicaciones bajo “contenedores” .....	18
Tecnologías utilizadas bajo Docker .....	18
Tiempo de respuesta medio por usuarios entre VM y Docker. ....	19
Gráfico de previsión de desarrollo del Proyecto (Burn down).....	25
Arquitectura del laboratorio virtual. ....	28
Diagrama de casos de uso. ....	30
Pantalla inicial al entrar en el Sistema Operativo. ....	34
Página de inicio del portal Guacamole .....	35
Home del usuario tipo “alumno” .....	35
Formulario de creación de Conexiones.....	36
Formulario de creación de Usuarios.....	36
Listado de usuarios .....	37
Listado de conexiones. ....	37
Sesiones activas .....	38
Historial.....	38
Ejemplo de compilación en C .....	40
Tiempo requerido para levantar el entorno.....	42
Tiempo requerido para apagar el entorno .....	42
Consumo de recursos en reposo. ....	43
Consumo de recursos con 4 puestos activos por entorno gráfico .....	43
Consumo de recursos con 4 puestos por protocolo SSH .....	44

Oracle VirtualBox con máquinas virtuales Ubuntu .....	45
Consumo de recursos durante el encendido del sistema .....	45
Consumo de recursos en reposo .....	46
Consumo de recursos de las máquinas virtuales con carga de trabajo .....	47
Gráfico comparativo de tiempo de encendido .....	48
Gráfico comparativo de tiempo de apagado.....	49
Gráfico comparativo de consumo de RAM en reposo .....	49
Gráfico comparativo de tiempo de CPU en reposo.....	49
Gráfico comparativo de consumo de RAM en carga.....	50
Gráfico comparativo de consumo de CPU en carga .....	50
Gráfico comparativo de reserva de espacio en Disco .....	51
Consumo de CPU en servidor del entorno piloto.....	52
Consumo de Memoria en servidor del entorno piloto .....	53
Consumo de CPU en laboratorio virtual del entorno piloto .....	54
Consumo de memoria RAM en laboratorio virtual del entorno piloto.....	55

## Índice de tablas

Comparativa de los Distintos Modelos de Cloud Computing.....	16
Características del equipo de pruebas .....	41
Requisitos mínimos de SO Ubuntu .....	41
Historias de Usuario Relacionadas con el Laboratorio.....	67
Historias de Usuario Relacionadas con las Pruebas .....	68



## 1. INTRODUCCIÓN

En todo proceso de aprendizaje la práctica y aplicación de los conocimientos es un elemento fundamental, ya que además de afianzar los conocimientos, demuestra que se han interiorizado.

Esto es de vital importancia en los conocimientos técnico-científicos, y más concretamente en la informática, donde todas las instituciones educativas incorporan en sus planes de estudio las prácticas como parte de sus métodos de enseñanza.

Por esto las instituciones educativas cada vez incorporan más elementos TIC al método de estudio, estos métodos crearán nuevas generaciones con una menor brecha tecnológica digital, que aprovecharán las oportunidades que la tecnología aporta a su formación. Para esto por supuesto la labor de docente es primordial, ya que él debe ser el que incluya en su metodología este tipo de laboratorios técnico-prácticos, tal y como concluye (Moya López, 2013) donde afirma que incluir las TIC en el proceso educativo:

[...]favorecerá la reducción de la brecha digital existente entre las distintas generaciones, contribuirá al desarrollo de la competencia digital, tanto de los docentes como de los alumnos [...]

También y en un aspecto más práctico, en el estudio realizado por (Espinosa-Ríos et al., 2016) donde se estudia la importancia de las habilidades prácticas a la hora de elaborar un conocimiento científico aplicado concretamente a las reacciones químicas y se concluye que:

[...]es posible concluir que las prácticas de laboratorio como estrategia didáctica en la enseñanza y el aprendizaje de las reacciones químicas y su ejecución, teniendo en cuenta los niveles de abertura, lograron desarrollar y fortalecer diversas habilidades científicas en los estudiantes [...]

Por ello y por el constante aumento de la digitalización de la sociedad que año tras año aumenta tal y como puede comprobarse en informes como el de la “Sociedad Digital en España 2019” (Fundación Telefónica, 2020b) o “El futuro digital de Europa” (Fundación Telefónica, 2020a), es necesario contemplar soluciones que aporten nuevas herramientas a los métodos de estudio, y entre ellas tal y como se ha comentado, están las aulas/laboratorios prácticos virtuales.

## 1.1. JUSTIFICACIÓN DEL TEMA ELEGIDO

Como parte del aprendizaje, las competencias prácticas son una de las labores más importantes a interiorizar. Por ello, y más concretamente en los estudios de carácter técnico-científico, es importante contar con unas correctas herramientas para que los alumnos puedan practicar sus conocimientos.

En el ámbito de la informática, este tipo de competencias se practican en laboratorios o prácticas más o menos guiadas, que plantean a los alumnos diversas situaciones que han de resolver en un entorno controlado.

En cuanto a las características necesarias para llevar a cabo estos estudios, pueden efectuarse bien de forma aislada con los materiales de los que cada alumno disponga (su equipo informático) o bien en las instalaciones de la institución educativa presencialmente, en aulas preparadas para ello o telemáticamente, en laboratorios virtuales. Existen múltiples estudios sobre la elaboración de este último tipo de laboratorios, como por ejemplo en el estudio realizado por (Andújar Márquez & Mateo Sanguino, 2010), donde se realiza un recorrido por esta herramienta de aprendizaje, partiendo de los laboratorios tradicionales hasta los más recientes laboratorios virtuales, tomando como ejemplo en este estudio un laboratorio para el ensayo en robots. Otros autores como (Luengas et al., s. f.) incluso estudian aplicar los últimos avances en la realidad virtual aplicándolos a la enseñanza, en cuanto a su utilidad práctica en los laboratorios prácticos se refiere.

Como se ha mencionado anteriormente, la variante virtual o remota de estos complementos a la enseñanza está cada vez más extendida e incluso en la enseñanza superior de nuestro país tenemos ejemplos claros de aplicación, tal y como se puede ver en el documentos del autor (Domínguez et al., 2010), donde se plantea y desarrolla la estructura de un laboratorio remoto para la enseñanza de la automática en la Universidad de León.

Tal y como se ha presentado, los laboratorios remotos son una herramienta que se utiliza en diversos ámbitos y a todos los niveles educativos. La Ingeniería Informática no es una excepción, y también existen diversas propuestas a nivel nacional como por ejemplo la de (Londoño Salazar & Alvarez Córdoba, 2015), donde se realiza un estudio de las necesidades y estructura que debiera tener un laboratorio virtual para un determinado plan de estudios a desarrollar de forma remota.

En cuanto al costo de la infraestructura, también existen estudios al efecto. (Calderón et al., 2015), donde desde la perspectiva de mejorar la calidad educativa en zonas con un nivel de desarrollo económico menor, se estudian las alternativas necesarias para incorporar las bondades de los avances en las TIC a los modelos educativos, sacándoles el mayor rendimiento.

Teniendo en cuenta todo lo anteriormente citado, queda de manifiesto la necesidad de la enseñanza a través de la práctica en todos los niveles educativos, pero si cabe en las relacionadas con la ciencia y la tecnología toman incluso más relevancia.

## 2. ESTADO DEL ARTE

En este apartado se analizan los distintos conceptos que son necesarios para el desarrollo de este trabajo, con el objetivo de que se pueda comprender el estado de la cuestión. Para ello primeramente se van a definir distintos conceptos que son de importancia a la hora de entender el trabajo, como son los conceptos de virtualización, contenedores y computación en la nube (Cloud Computing), posteriormente se lleva a cabo una comparativa de las distintas soluciones tecnológicas disponibles a la hora de realizar un entorno virtualizado, para finalmente concretar una solución con un valor añadido, que se desarrollará a lo largo de este Trabajo Fin de Grado.

### 2.1. DEFINICIONES

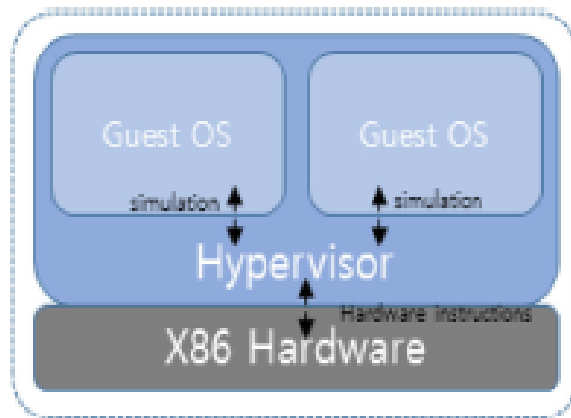
#### Virtualización

La virtualización es una tecnología que permite mediante el uso de un software simular un sistema o un recurso tecnológico, que se manifiesta en lo que se denomina máquina virtual (MV). Esto se utiliza para múltiples objetivos, desde los más sencillos como puede ser trabajar temporalmente con un sistema operativo distinto al nativo, hasta los más complejos como pueden ser las aplicaciones relacionadas con la computación de alto rendimiento o la virtualización de diversos servicios. Por ello se pueden distinguir dos tipos de virtualización.

#### Virtualización nativa

Este tipo de virtualización el software de virtualización (Hypervisor) es también el Sistema Operativo, en adelante SO. En la Figura 1 puede verse la arquitectura de una virtualización nativa.

**Figura 1**



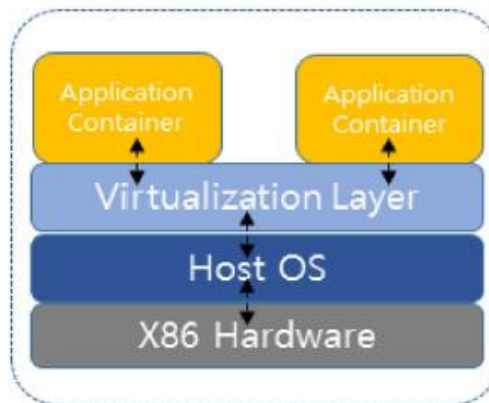
#### *Estructura de una Virtualización Nativa*

*Nota: Tomado de A performance comparison of linux containers and virtual machines using Docker and KVM (p. 2), por M. Chae, H. Lee & K. Lee, 2019, Cluster Computing, Vol. 22, 1.*

#### Virtualización alojada o Hosted

En esta segunda variante el software de virtualización se ejecuta sobre el SO de la propia máquina, tal y como puede observarse en la Figura 2.

**Figura 2**



#### *Estructura de una Virtualización Alojada o “Hosted Virtualization”*

*Nota: Tomado de A performance comparison of linux containers and virtual machines using Docker and KVM (p. 3), por M. Chae, H. Lee & K. Lee, 2019, Cluster Computing, Vol. 22, 1.*

Esta tecnología lleva aplicándose desde los años 60 y ha ido evolucionando durante todo este tiempo, quedando de manifiesto sus beneficios como pueden ser: (INCIBE, 2020)

- Reducción de costes: El hecho de poder disponer de un servidor virtualizado que de servicio a distintas necesidades es una ventaja que además aporta un plus de eficiencia.
- Escalabilidad: El software de virtualización está pensado para ser fácilmente escalables, permitiendo el crear o eliminar distintas máquinas virtuales según las necesidades.
- Compatibilidad: Gracias a la posibilidad que ofrece la virtualización de crear distintas máquinas virtuales, es posible ejecutar diferente software que requieran un entorno específico para ser ejecutados.
- Seguridad: La virtualización permite crear entornos y máquinas aisladas del resto de una red, o bien de otras MVs.

## Contenedores

Esta tecnología permite aislar aplicaciones junto con todo lo que necesitan para ejecutarse, por ejemplo, archivos o configuraciones particulares. Tal y como se aproxima esta idea en (Wang, 2016), la primera implementación de esta tecnología se denomina Contenedores Linux, a partir de ahora LXC, y no es más que un método de virtualización a nivel de SO que permite ejecutar varios sistemas Linux en un mismo servidor.

Al tratarse de un componente aislado, una de las ventajas de esta tecnología es que se pueden migrar los contenedores entre distintas máquinas, siempre que estas permitan la ejecución de contenedores. Esto es posible debido a que un contenedor incorpora en su interior un SO llevado a su mínima expresión, con el objetivo de ejecutar la aplicación para la cual se ha creado, por lo que no incorpora componentes innecesarios.

Docker ha sido un avance en este campo, ya que lleva al extremo lo anterior, siendo un proyecto concebido exclusivamente para ejecutar aplicaciones aisladas, lo que las convierte en portables, dotando a esta tecnología de más flexibilidad. Tal y como se afirma en (Wang, 2016):

[...] Docker ofrece la flexibilidad de la nube a cualquier infraestructura capaz de trabajar con contenedores [...]

Como puede observarse se trata de tecnologías con similitudes, ya que una es un refinamiento de la otra. Si bien también existen múltiples diferencias, las cuales se enumeran a continuación:

- Mientras que LXC está pensado para trabajar con múltiples procesos, Docker lo está para trabajar en proceso único, esto es así para permitir que por ejemplo detener o actualizar ciertos servicios sin que afecten al resto, lo cual es beneficioso para, por ejemplo, para los microservicios.
- Los contenedores Docker están diseñados para trabajar sin Estados, es decir que carecen de persistencia de datos, aunque existen posibilidades para que esto no sea definitivo, como la figura de los “Volúmenes”.
- Las Imágenes Docker, configuraciones del servicio, son archivos de sólo lectura. Esto implica que cada vez que hagamos un cambio en la Imagen se generará una nueva, que pasará a ser una versión superior de la anterior.
- Docker también ofrece más portabilidad, esto es debido a que con respecto a LXC, Docker ofrece la posibilidad real de trasladar los contenedores de una máquina a otra garantizando que el entorno será el mismo, esto incluye la configuración de red, almacenamiento o el SO sobre el que se ha creado el contenedor.

## Cloud Computing

Una de las tendencias que está cobrando importancia en los últimos años es el concepto de cloud computing, el cual se basa en ofrecer distintos tipos de servicios gestionados por terceros en los cuales es posible pagar por uso, algunos ejemplos de compañías que ofrecen estos servicios son Amazon AWS, Microsoft Azure o Google Cloud. Se distinguen principalmente 3 modelos de servicio, los cuales son:

### Infraestructura como servicio (IaaS)

Este modelo ofrece como ventaja el no tener que adquirir físicamente la infraestructura en la cual vamos a correr nuestro software, esto es que el modelo se basa en un alquiler de la

infraestructura necesaria a un tercero, teniendo como particularidad que la gestión de dichas máquinas corre a cargo del cliente del servicio.

#### Plataforma como servicio (PaaS)

En este modelo el cliente no gestiona física ni administrativamente las máquinas sobre las que ejecuta el software, este modelo es muy utilizado cuando no se desea diseñar o administrar la infraestructura y simplemente se hace uso de esta, donde se ejecutará el software sin necesidad de controlar o prever la escalabilidad de los sistemas, algo que el proveedor del servicio hará automáticamente.

#### Software como servicio (SaaS)

Es el último nivel de abstracción de los servicios en la nube, en él se hace uso de un determinado software, sin necesidad de diseñarlo, implementarlo o actualizarlo. Simplemente el proveedor ofrecerá el software y se hará uso de este.

#### Comparativa de los servicios

A continuación, se muestra una tabla en la que pueden observarse las diferencias entre los distintos tipos de servicio en la nube, quedando reflejado en ellos el nivel de administración que posee el cliente y el proveedor.

**Tabla1**

*Comparativa de los Distintos Modelos de Cloud Computing*

	IaaS	PaaS	SaaS
<b>Aplicaciones</b>	Cliente	Cliente	Proveedor
<b>Datos</b>	Cliente	Cliente	Proveedor
<b>Middleware</b>	Cliente	Proveedor	Proveedor
<b>SO</b>	Cliente	Proveedor	Proveedor
<b>Virtualización</b>	Proveedor	Proveedor	Proveedor
<b>Servidores</b>	Proveedor	Proveedor	Proveedor
<b>Almacenamiento</b>	Proveedor	Proveedor	Proveedor
<b>Redes</b>	Proveedor	Proveedor	Proveedor

## 2.2. TIPOS DE ENTORNOS

Dependiendo del ámbito en el que se desarrolle una actividad informática, existen diferentes posibilidades a la hora de crear el entorno de esta, pudiendo incluso entrelazarse las



posibilidades que se relatarán a continuación, es decir que un entorno físico también podría tener partes de este virtualizadas.

#### Entorno físico

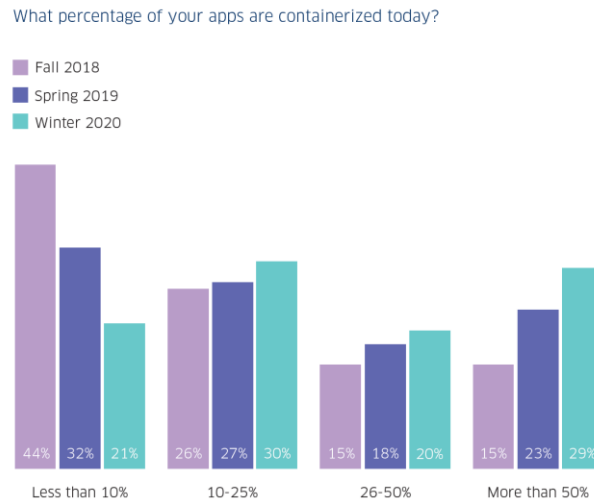
Se entenderá por entorno físico, aquel en el que todos los componentes de este se encuentran formados por hardware no virtualizado, como por ejemplo un puesto de trabajo en una oficina o bien los equipos que pueden encontrarse en un cibercafé.

#### Entorno virtualizado (MV)

En esta variante, los equipos dan servicio a máquinas virtuales, que estarán gestionadas por los administradores mediante el hipervisor. Este tipo de entorno puede encontrarse de forma física en un Centro de Proceso de Datos (CPD) o bien con sus servicios en la Nube, siendo esta última opción una de las que se encuentran en auge hoy en día, con la aparición de los servicios en la nube y los entornos de “pago por uso”, con lo que la empresa ya no tiene la necesidad de disponer de un CPD propio, pudiendo hacer uso de estos servidores remotos cuando lo considera necesario.

#### Entorno utilizando contenedores

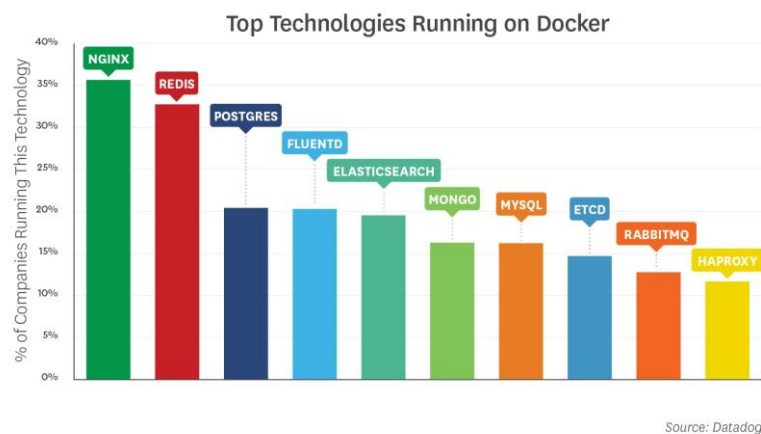
Por último, se presenta un entorno con la tecnología de contenedores, aunque no se puede definir como un entorno claro, sí es una tendencia que facilita y optimiza los recursos Hardware, por ello cada vez más aplicaciones se encuentran bajo esta tecnología, tal y como puede observarse en la Figura 3, donde se aprecia una tendencia al alza.

**Figura 3**

### *Tendencia de las aplicaciones bajo “contenedores”*

*Nota: Adaptado de 6 Container Adoption Trends of 2020, por A. Kohgadai, 2020, stackrox.com (<https://www.stackrox.com/post/2020/03/6-container-adoption-trends-of-2020/>)*

Un ejemplo de aplicaciones de uso común que se pueden encontrar adaptadas a la tecnología de contenedores (Docker) puede observarse en la Figura 4, donde se puede apreciar una estadística de uso por aplicaciones tan conocidas como nginx o Elasticsearch.

**Figura 4**

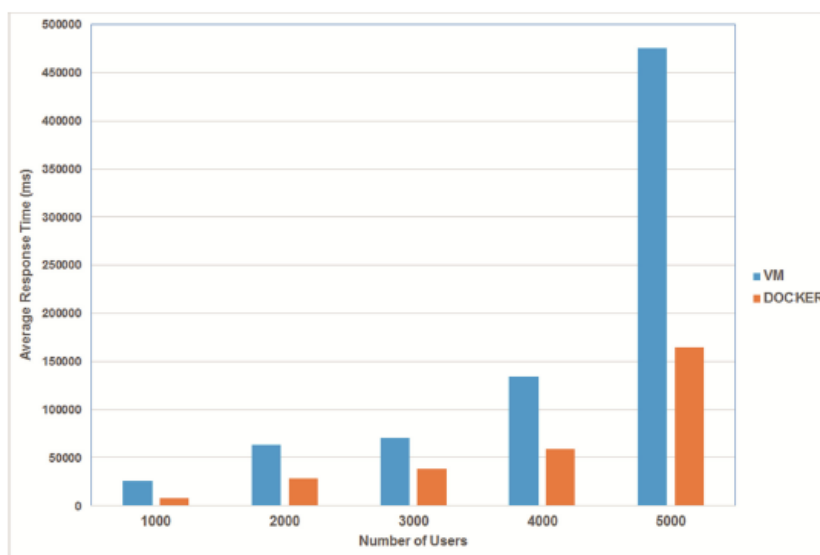
### *Tecnologías utilizadas bajo Docker*

*Nota: Adaptado de 8 Surprising Facts About Real Docker Adoption, por Datadog, 2018, datadoghq.com (<https://www.datadoghq.com/docker-adoption/>)*

### Entorno virtualizado vs contenedores (Docker).

Desde la aparición de los contenedores se han estudiado y analizado sus características y ventajas con respecto a la virtualización tradicional, por ello se han publicado multitud de artículos en los que se compara el rendimiento de ambos sistemas, como pueden ser por ejemplo los publicados por (Chae et al., 2019),(Felter et al., 2015),(Kozhirbayev & Sinnott, 2017) o (Kavitha & Varalakshmi, 2018). De entre otras conclusiones publicadas, se puede observar cómo en un entorno con contenedores, el consumo de CPU y memoria es menor que en un entorno con máquinas virtuales, así como sus tiempos de respuesta, tal y como se puede observar en la Figura 5, correspondiente a peticiones HTTP.

**Figura 5**



*Tiempo de respuesta medio por usuarios entre VM y Docker.*

*Nota: Adaptado de “Performance Analysis of Virtual Machines and Docker Containers” (p. 109), por B. Kavitha & P. Varalakshmi, 2018, Data Science Analytics and Applications*

Estos estudios tienen como punto común resaltar la buena optimización recursos/rendimiento que ofrecen los contenedores, en campos como puede ser el relativo al Internet de las cosas (IoT) (Ramalho & Neto, 2016), los aplicados al aprendizaje máquina (Xu et al., 2017) o las bases de datos (Felter et al., 2015).

### 2.3. TRABAJOS SIMILARES

En la actualidad existen diversos estudios sobre entornos de enseñanza virtuales que abarcan distintos tipos de aproximaciones, como por ejemplo los trabajos publicados por (Muradova, 2020) o (Smolyaninov et al., 2021) donde se plantea la necesidad de disponer de entornos software para la enseñanza a distancia, en concreto en el trabajo de Smolyaninov se menciona a que con la aparición de la COVID-19 y su impacto a nivel mundial, este tipo de enseñanza ha manifestado su importancia, mostrando incluso beneficios en la calidad de la educación.

En la línea de lo anterior, existen trabajos sobre el diseño que han de tener este tipo de herramientas para que sean útiles en la enseñanza y que los recursos empleados se conviertan en una herramienta que cumpla la función para la cual se han diseñado, por ejemplo el trabajo de (Londoño Salazar & Alvarez Córdoba, 2015) plantea distintos modelos sobre los que basar el diseño de un laboratorio virtual, estos diseños servirán de base para definir el objetivo y el método de aprendizaje que se defina para el laboratorio virtual.

Siguiendo con la definición y diseños de las características de un laboratorio virtual, el trabajo de (Andújar Márquez & Mateo Sanguino, 2010) plantea un enfoque práctico para el diseño de un laboratorio remoto orientado a la robótica.

En cuanto a laboratorios remotos, el autor (Domínguez et al., 2010) se estudia la creación de un laboratorio remoto para la enseñanza de la automática en el ámbito educativo de la Universidad de León.

Existen también estudios sobre la diferencia tecnológica en el uso de virtualización y entornos en la nube, los anteriormente citados (Chae et al., 2019),(Felter et al., 2015),(Kozhirkbayev & Sinnott, 2017) o (Kavitha & Varalakshmi, 2018) son clara muestra de ello, pero también existen estudios más recientes como por ejemplo el trabajo de (Slamnik-Krijes'torac et al., 2021) donde se plantea la necesidad de disponer de laboratorios de bajo costo, estudiándose el uso de dispositivos Raspberry Pi para el montaje de un aula física.

También en el ámbito de la creación de laboratorios de bajo costo existe la propuesta de (Calderón et al., 2015), donde se estudian distintas posibilidades de creación de actividades educativas utilizando las tecnologías de la información y las comunicaciones (TIC).

## 2.4. TECNOLOGÍAS EMPLEADAS

Para el desarrollo de la propuesta de este Trabajo Fin de Grado se han examinados las distintas tecnologías existentes, con el objetivo de evaluar su utilidad y adecuación para su utilización como punto de partida para la creación de un entorno educativo virtual. Por ello y habiendo evaluado dichas alternativas, se ha decidido utilizar las siguientes tecnologías para el desarrollo de este

### Docker

Docker es una tecnología Open Source (código abierto) que, como se ha comentado anteriormente, utiliza las capacidades del Kernel de Linux tales como el uso de Cgroups<sup>1</sup> y namespaces<sup>2</sup>, para lograr que los procesos se ejecuten independientemente. Esto es importante para lograr una buena optimización de los recursos además de incrementar la seguridad al mantener los sistemas separados. En el caso de este Trabajo, resulta muy interesante la posibilidad de optimizar los recursos a la hora de compartimentar los procesos, ya que haciendo uso de esta propiedad se pretende lograr que la arquitectura propuesta sea lo más liviana posible, logrando así un entorno funcional con el menor gasto de hardware posible.

Otra de las características interesantes es la posibilidad de crear distintas imágenes y controlarlas con el control de versiones de Docker, esta capacidad permite moldear los contenedores con facilidad y rapidez, ya que la propia tecnología incluye un control de versiones que optimiza el tamaño, reutilizando lo que se puede e incluyendo los cambios en las imágenes Docker cuando es preciso.

### Sistema Operativo Linux

Aunque la tecnología de contenedores permite su utilización bajo ciertas condiciones en el Sistema Operativo Windows, se ha optado por ejecutar el motor de Docker bajo SO Linux debido a que su funcionamiento y posibilidades están optimizados para estos sistemas operativos. Para ello se ha optado por la distribución Ubuntu 20.04, una de las más conocidas

---

<sup>1</sup> Cgroups: Grupos de control, utilizados para la asignación de recursos a los procesos.

<sup>2</sup> Namespace: Funcionalidad utilizada para agrupar procesos y segregar recursos.

y utilizadas, que es compatible al completo con la tecnología Docker y además existe gran documentación al respecto de cómo utilizar dicha tecnología bajo este sistema Operativo.

### Apache Guacamole

Apache Guacamole es una tecnología que permite acceder a equipos mediante el uso de VNC, RDP o SSH sin la necesidad de un cliente, haciendo uso simplemente de la tecnología HTML5 y un navegador web. Se trata de una tecnología Open Source y con una buena documentación. Es una tecnología muy interesante para este TFG debido a que permite el acceso a distintas máquinas sin necesidad de depender de un cliente concreto, lo que ofrece una posibilidad de conectividad multiplataforma, incluso con dispositivos móviles.

Esta tecnología ofrece además la posibilidad de llevar una gestión de usuarios, lo que permitirá la asignación de recursos de forma eficiente.

### Nginx

Nginx es un proxy/servidor web Open Source, su funcionamiento es similar a Apache, pero se trata de un servicio más liviano, ya que consume menos recursos y es capaz de gestionar más solicitudes por minuto. En este TFG se utilizará como proxy de cara a facilitar el acceso al portal que proporciona entrada al sistema de laboratorio virtual.

### Github

Github es un portal creado para alojar proyectos de cualquier desarrollador, el portal utiliza Git como sistema de control de versiones, por lo que facilita la gestión y administración de los proyectos. El portal es gratuito, pero siempre y cuando los proyectos alojados sean código abierto (Open Source). Esta herramienta se utilizará en el proyecto para almacenar al código del proyecto, así como los archivos de configuración y la documentación generada durante el desarrollo de este TFG.

### PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos relacional orientado a objetivos y código abierto. Su uso en esta propuesta es para la gestión del portal de entrada al laboratorio virtual, gestionado a través de usuarios con Apache Guacamole.

## 2.5. RESUMEN DEL ESTADO DEL ARTE

Habiendo desarrollado el estado de la cuestión previamente, en este Trabajo fin de Grado, se propone realizar un estudio de arquitectura centrada en la figura del laboratorio virtual que permita aprovechar algunos de los últimos avances y tendencias tecnológicas. Además, se buscará que la infraestructura propuesta sea fácilmente replicable, escalable y sostenible en el tiempo.

A lo largo de este documento se han visto diferentes ejemplos y aplicaciones prácticas para los distintos tipos de laboratorios, desde los clásicos ubicados físicamente en un aula hasta los contruidos sobre virtualización en mayor o menor medida. También se han valorado propuestas más recientes como la tecnología basada en los contenedores, que han ido ganando terreno por su capacidad de optimización de los recursos hardware.

Por ello desde este trabajo, se propone utilizar como elemento principal para el diseño de la propuesta la tecnología de contenedores (Docker). Se busca con esta opción aprovechar las características de rendimiento, portabilidad y escalabilidad que los contenedores ofrecen para la elaboración de un laboratorio virtual que sea económica y tecnológicamente equilibrado, intentando sacar el máximo partido al hardware disponible, y permitiendo que el modelo y la arquitectura propuesta sean lo suficientemente elásticos para que permitan amoldarse a distintos proyectos con los mínimos cambios. Además, esta tecnología permite una eventual migración a los servicios de Cloud Computing, lo que le otorga a la propuesta una gran flexibilidad a la hora de escalar en necesidad, pudiendo desarrollar la propuesta en un pequeño servidor para unos pocos alumnos o bien en un servicio en la nube adaptable a la cantidad de alumnos que se necesiten, siendo pues una opción viable para un particular o una institución educativa de mayor relevancia.

Este laboratorio tendrá un acceso multiplataforma, a través de cualquier navegador con capacidad para ejecutar HTML5, por lo que proporcionará al usuario un gran abanico de posibilidades a la hora del acceso y utilización.

Finalmente, y como prueba piloto se definirá un pequeño laboratorio de 5 puestos, donde se podrán testar las posibilidades y rendimiento de la arquitectura propuesta.

## 3. DISEÑO DE LA PROPUESTA

### 3.1. OBJETIVOS

#### Objetivo principal

Estudio de las soluciones tecnológicas para la creación de entornos de formación virtuales en contenedores.

#### Objetivos secundarios

Realización de un caso de estudio donde se implementa una de las soluciones evaluadas.

Utilización de metodologías ágiles para el desarrollo de la propuesta.

Utilización de la tecnología HTML5 para posibilitar el acceso desde el mayor número de dispositivos.

Evaluación del consumo de recursos de la solución propuesta.

### 3.2. METODOLOGÍA

A lo largo de este apartado se desarrolla el método de trabajo elegido para desarrollar el contenido del trabajo, así como las secciones en las que se ha dividido el mismo para llegar a su completa elaboración.

#### Metodología de Desarrollo: Scrum

Se ha escogido la metodología ágil para el desarrollado de este trabajo debido a que se gran parte de este se basa en investigación y pruebas, entre las disponibles, se ha seleccionado la metodología SCRUM, debido a que además de ser una metodología ampliamente utilizada, permite flexibilidad e inmediatez a la hora de afrontar el proyecto y su desarrollo.

Esta metodología permite que el proyecto se pueda desglosar en pequeñas tareas que permitan conocer el avance de este a lo largo del tiempo, con lo que se consigue una visualización de la progresión en el mismo acorde con el avance del trabajo.

Para ello se ha considerado la estructura de proyecto que se relata en los siguientes puntos.



## Historias de Usuario

Para la definición de los requisitos y funcionalidades que debe disponer el laboratorio virtual, así como la documentación generada, se definen historias de usuario clasificadas por los diversos temas (documentación, laboratorio y pruebas). Estas Historias definen las necesidades identificadas y las clasifican por temática y momento de su desarrollo, incluyéndolas en el correspondiente Sprint de los 4 identificados, tal y como se detalla a continuación. En el Anexo A puede consultarse el listado completo de historias de usuario, así como el Sprint en el que se han desarrollado y la duración estimada de las mismas.

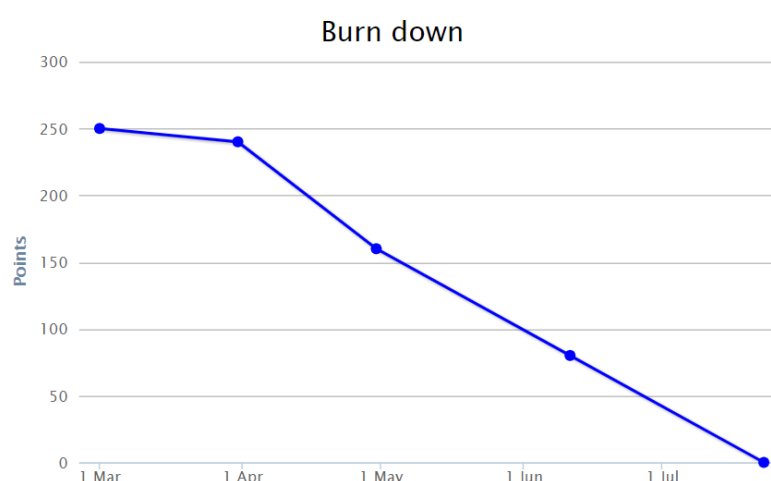
## Sprints

Un sprint es una agrupación de historias que se establecen para su resolución en un periodo de tiempo que en el caso de este proyecto será de 30 días.

Se han establecido 4 Sprints, cada uno con una duración aproximada de 30 días, los cuales se corresponden con la duración estimada del proyecto, 4 meses.

En la Figura 6, se representa visualmente el desarrollo esperado para el desarrollo de los distintos Sprints, desde su inicio en Marzo hasta su finalización en Julio. A continuación, se detallan los distintos Sprints, para la identificación de las tareas se ha incluido en el Anexo A un listado completo de las mismas con su referencia e identificador.

**Figura 6**



*Gráfico de previsión de desarrollo del Proyecto (Burn down).*

*Fuente: Imagen propia.*

*Nota: En el Gráfico puede apreciarse que en la parte izquierda se representan los puntos de historia del proyecto al inicio (250), así como su descenso a medida que avanza el desarrollo del proyecto.*

#### SPRINT 1, marzo-abril de 2021

Durante este primer sprint, se ha realizado la mayor parte de documentación sobre la temática del proyecto, así como la recopilación bibliográfica del mismo. Al final de este debe disponerse de un modelo de documentación de Memoria, así como unos primeros apartados redactados, como son los primeros apartados introductorios de la documentación.

Este primer Sprint consta de las tareas MEM1, MEM2 y MEM3, con un total de 37 puntos de Historia.

#### SPRINT 2, abril-mayo de 2021

El segundo Sprint está orientado a afianzar los conocimientos necesarios para iniciar los trabajos del laboratorio, así como seguir completando la documentación del Proyecto. Para ello se realiza una labor de documentación técnica acerca de las tecnologías a emplear, así como la preparación de un entorno de desarrollo y pruebas para la realización del piloto de laboratorio virtual contemplado en este TFG.

Este Sprint consta de las siguientes Historias de Usuario:

- Tareas MEM4, MEM5, MEM6 y MEM7, relativas a diversos apartados de la documentación.
- Tareas LAB1, LAB2, LAB3 y LAB4, relativas a la puesta en marcha de un entorno de trabajo y la documentación de este.
- Tareas PRU1 y PRU2, relativas a las pruebas de funcionamiento de las tecnologías empleadas.

El Sprint consta de un total de 90 Puntos de Historia.

#### SPRINT 3, mayo-junio de 2021

Durante el tercer Sprint se realiza la mayor parte del trabajo de documentación técnica, así como de labores de desarrollo sobre el entorno de trabajo. Se realizan las labores de puesta

en marcha de las últimas tecnologías, así como el desarrollo de los elementos necesarios para que todos estos componentes puedan interactuar correctamente entre ellos. Se realizan también las pruebas relacionadas con estos últimos aspectos.

El Sprint consta de las siguientes Historias de Usuario:

- Tarea MEM8, relativa a la documentación de los contenidos y arquitectura y funcionamiento del laboratorio.
- Tareas LAB5 y LAB6, relativas a la configuración de los últimos componentes y al desarrollo de los componentes necesarios para su funcionamiento.
- Tarea PRU3, constando de las pruebas relacionadas con las anteriores tareas.

El Sprint consta de un total de 63 puntos de historia.

#### SPRINT 4, junio-julio de 2021

El último Sprint consta de la parte relativa a la recopilación de datos de funcionamientos del laboratorio, así como la redacción de las conclusiones y vías de trabajo futuro sobre la propuesta. También se realizan las pruebas de rendimiento y funcionamiento en el laboratorio.

El Sprint consta de las siguientes Historias de Usuario:

- Tarea MEM9, relativa a las conclusiones y trabajo futuro de la documentación
- Tarea LAB7, relativa a la puesta en marcha definitiva del laboratorio de pruebas.
- Tareas PRU4 y PRU5, relativas a las pruebas finales y de rendimiento sobre el laboratorio.

El Sprint consta de un total de 60 puntos de historia.

### 3.3. DISEÑO DE LA PROPUESTA

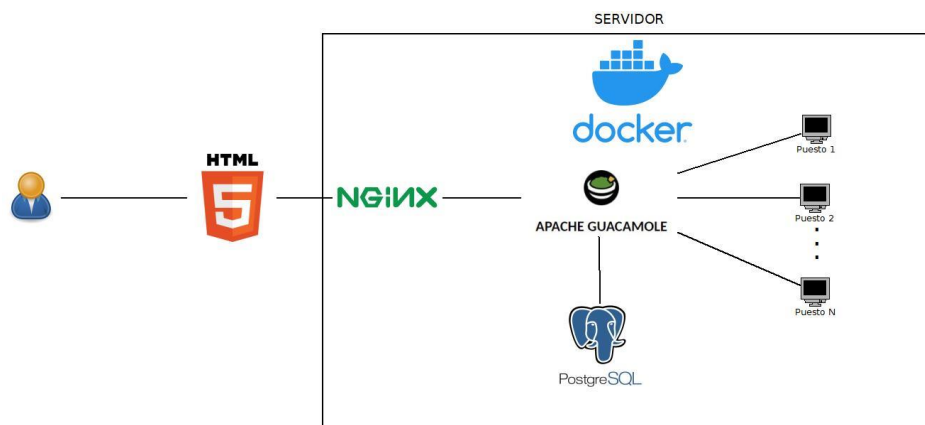
#### Introducción

Teniendo en cuenta las conclusiones anteriores, a continuación, se elaborará la propuesta de arquitectura y funcionamiento de un laboratorio virtual basado en contenedores Docker. El objetivo de este es ofrecer una funcionalidad similar a la de un laboratorio virtualizado o físico bajo una demanda de hardware mucho menor, a la vez que pueda disponerse de disponibilidad de este cuando se necesite.

## Arquitectura del Sistema

Tal y como puede observarse en la Figura 7, la arquitectura del sistema consta de un servidor que contiene en su interior todo lo necesario para que el sistema funcione.

**Figura 7**



*Arquitectura del laboratorio virtual.*

*Fuente: Imagen propia.*

El sistema está basado en la tecnología Docker para que genere todo el entorno (puestos de trabajo, portal de acceso y base de datos), los distintos puestos de trabajo, cuyas imágenes han sido previamente generadas y puestas a disposición del sistema. A estos distintos puestos se accede a través de la tecnología Apache Guacamole, que funciona como un portal de acceso que habilita a cada usuario el acceso a las máquinas a las cuales deba tener acceso. Estos accesos se definen por el Administrador del Sistema en el propio Guacamole, que almacena todos estos perfiles en su base de datos, que, en el caso de esta propuesta, será PostgreSQL.

El sistema dispone también de un Proxy bajo la tecnología NGINX que puede ser configurado según las distintas necesidades, en el caso del piloto simplemente redireccionará el acceso por el puerto 80 al puerto del Apache Guacamole, que por defecto es el 8080.

Finalmente, los usuarios acceden al sistema por la URL del portal que crea Apache Guacamole, para ello el único requisito es acceder al sistema con un navegador que permita utilizar la tecnología HTML5, bajo la que podrá accederse a este desde cualquier dispositivo que permita usar un navegador compatible.

Esta última característica ofrece al usuario la posibilidad de conectarse desde cualquier Sistema Operativo o cualquier dispositivo (Tablet, Smartphone, Notebooks, etc.) siempre y cuando disponga de un navegador que pueda correr esta tecnología.

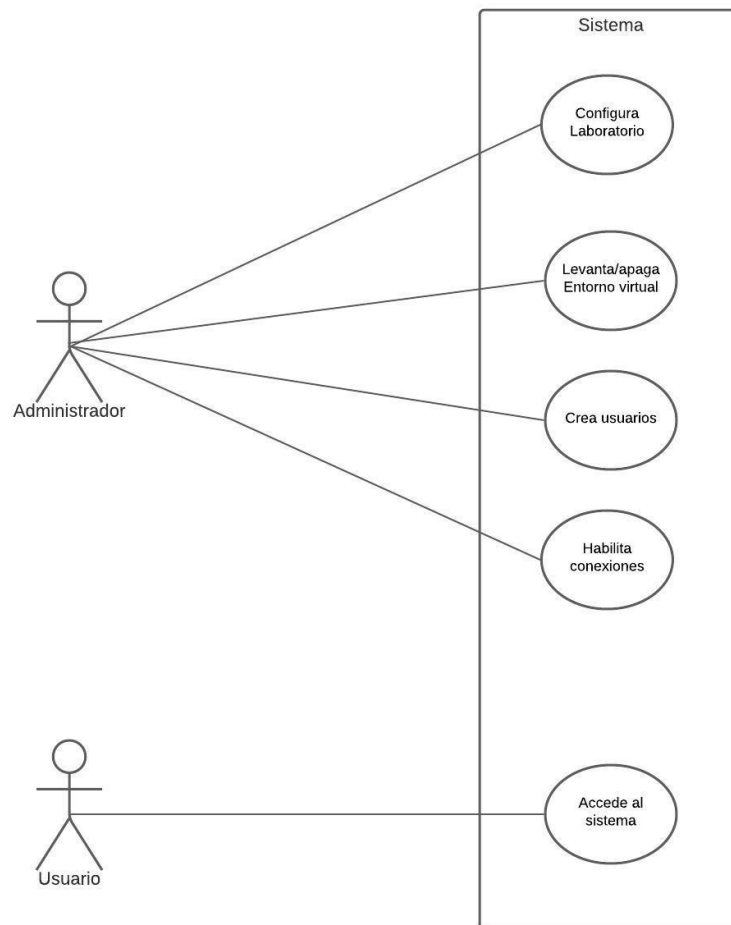
#### Casos de uso: Introducción

Dentro de las funcionalidades del laboratorio se identifican dos actores bien diferenciados, cuyos casos de uso se basan en las distintas posibilidades que el sistema ofrecerá a cada uno de ellos, siendo estos el Administrador y el Usuario.

Esta distribución de tareas se ha planteado para que el Sistema pueda ser administrado de forma sencilla e incluso unipersonalmente, pensando siempre en un sistema de administración eficaz que siga la filosofía planteada en la tecnología base que lo conforma, los contenedores.

El usuario dispondrá de la posibilidad de conectarse al sistema y operar con el mismo, para ello hará uso de las credenciales proporcionadas por el Administrador y de trabajar con el sistema utilizando para ello un dispositivo que pueda ejecutar un navegador con posibilidad de interpretar HTML5, como por ejemplo Google Chrome o bien Mozilla Firefox.

Tal y como puede verse en la Figura 8, el Administrador será el encargado de configurar tanto el laboratorio como de establecer los distintos usuarios/contraseñas y recursos asignado a cada usuario, así como asegurarse de que el entorno está operativo cuando así se precise. Todo esto lo realizará a través de un mismo portal, tal y como se definirá más adelante, siendo necesaria su intervención a nivel de servidor únicamente cuando se deban hacer cambios en la arquitectura del laboratorio o bien para resetear o depositar recursos en el Sistema.

**Figura 8**

*Diagrama de casos de uso.*

*Fuente: Imagen propia.*

#### Casos de uso: Actores

**Administrador:** Como se desprende de la anterior figura, el Administrador es el actor que se encargará de la gestión del Sistema, así de asegurarse de su disponibilidad. También será el encargado de habilitar las conexiones y credenciales del resto de usuarios para que puedan acceder al sistema.

**Usuario:** Se trata del destinatario final del Sistema, dentro del mismo se utilizará los recursos y herramientas asignados para su perfil.

## Casos de uso y diagramas de Secuencia

En virtud de los casos de uso identificados, se incluye como documentación adicional en el Github del proyecto el conjunto de los diagramas de secuencia, así como los casos de uso descritos y desarrollados en un formato homogéneo.

### Desarrollo de los casos de uso

A partir de los casos de uso para cada una de las operaciones anteriores, se procede a resumir cada una de esas operaciones:

#### **Configurar entorno:**

El Administrador se conectará al servidor, normalmente mediante protocolo SSH y editará los archivos de configuración del Sistema, donde se podrán configurar el número de puestos, así como las carpetas persistentes o el resto de las particularidades del entorno. Una vez realizada esta operación se enviará la orden de iniciar el entorno a Docker, el cual emite una respuesta por defecto de cómo ha ido la operación. Finalmente, el usuario se desconectará del servidor.

#### **Levantar entorno virtual:**

El Administrador se conectará al Servidor y ejecutará el comando definido para levantar el entorno, este comando es un script en BASH que se limita a iniciar el entorno y mostrar una salida por pantalla de cómo ha ido la operación.

#### **Apagar entorno virtual:**

El Administrador se conectará al Servidor y ejecutará el comando definido para apagar el entorno, este comando es un script en BASH que se limita a apagar el entorno y mostrar una salida por pantalla de acerca del resultado de la operación.

#### **Habilitar conexiones:**

El Administrador se conectará al portal Apache Guacamole a través de su Navegador utilizando sus credenciales. Una vez en el portal creará la conexión con los datos de red relativos al puesto hacia el que se desea crear y Guacamole guardará dichos datos en su BBDD PostgreSQL. Finalmente ofrece una confirmación visual de la operación realizada.

#### **Crear usuarios:**

El Administrador se conectará al portal Apache Guacamole a través de su Navegador utilizando sus credenciales. Una vez en el portal creará el usuario con los datos de este y le asignará una o varias conexiones hacia los puestos de trabajo virtual según las necesidades y Guacamole guardará dichos datos en su BBDD PostgreSQL. Finalmente ofrece una confirmación visual de la operación realizada.

**Acceso al sistema:**

El usuario accederá al portal Apache Guacamole haciendo uso de un navegador que sea compatible con la tecnología HTML5 y utilizando sus credenciales asignadas. Una vez dentro accederá a una de las conexiones asignadas y Guacamole redirigirá esta petición a través de VNC o SSH dependiendo del tipo de conexión hacia el puesto asignado.

**Imágenes Docker**

Para la creación del entorno virtual, es necesario el uso de imágenes Docker. En el presente apartado se realiza una descripción de cada una de ellas, así como el uso que se hace de las mismas en el entorno virtual. Todas las imágenes a excepción de la que crea el SO de cada uno de los puestos de usuario se han obtenido del portal oficial de documentación y recursos de Docker<sup>3</sup>.

**Imagen de Apache Guacamole:**

Para el desarrollo del entorno virtual se ha utilizado la versión 1.3.0, la cual es la versión más reciente disponible. Esta imagen levantará un contenedor que albergará un Apache Guacamole que dará acceso de gestión al Administrador del Sistema, así como acceso funcional a los usuarios de este. Estos accesos estarán personalizados en el propio Guacamole, así como las conexiones a los distintos puestos de trabajo virtuales.

**Imagen de PostgreSQL:**

Para el desarrollo del entorno se ha utilizado la versión 13 de PostgreSQL, al ser la versión más reciente compatible con la imagen de Apache Guacamole seleccionada. Esta imagen levantará

---

<sup>3</sup> <https://hub.docker.com/>



un contenedor que albergará un servidor de PostgreSQL cuya base de datos dará funcionalidad y persistencia al contenedor que alberga al Apache Guacamole.

### **Imagen de NGINX:**

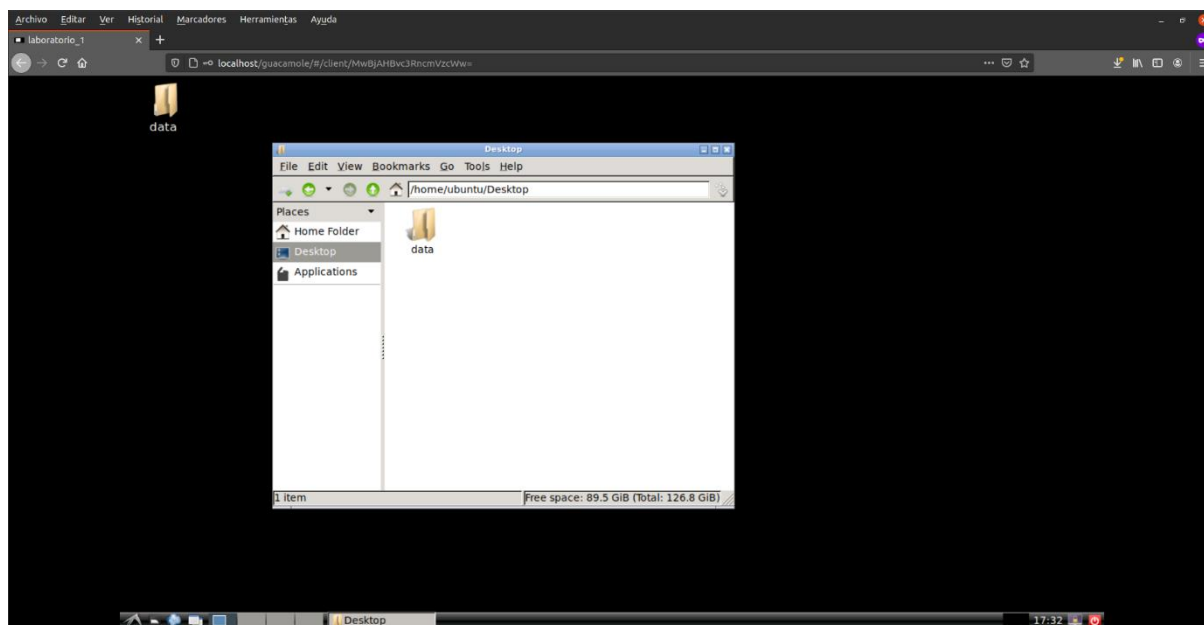
En el entorno virtual se hace uso de la versión 1.19, al ser la más reciente disponible. El contenedor que alberga a NGINX proporcionará una capa extra a la hora de actuar como balanceador, aunque para el piloto de laboratorio, simplemente hará una redirección del puerto 80 al 8080, que es el puerto que utiliza Guacamole por defecto.

### **Imagen del Sistema Operativo:**

Esta última imagen se basa en una imagen por defecto de Ubuntu 20.04. Se ha seleccionado este sistema operativo por ser uno de los más extendidos en su uso. Además, la abundante documentación y paquetería disponible en sus repositorios convierten a este SO en un entorno ideal para este laboratorio virtual.

Esta imagen se ha personalizado para el uso, incluyendo un entorno gráfico ligero que permita operar con el Sistema de forma amigable. Entre otras características se incluyen un servidor de SSH para su uso y conexión por medio de una consola, diversos editores de texto que puedan facilitar el desarrollo, así como un conjunto de compiladores y herramientas de programación que convierten el entorno en un sistema apto para su uso como plataforma de desarrollo.

El contenido de la imagen puede editarse al gusto del Administrador del Sistema incluyendo o eliminando la paquetería que se considere, estado disponible una vez levantado de nuevo el laboratorio. Puede observarse una pantalla ejemplo en la Figura 9, donde se aprecia la interfaz básica del sistema corriendo bajo un navegador Chrome.

**Figura 9**

*Pantalla inicial al entrar en el Sistema Operativo.*

*Fuente: Imagen propia.*

Se ha incluido en el Github del proyecto la imagen editada para su consulta o modificación.

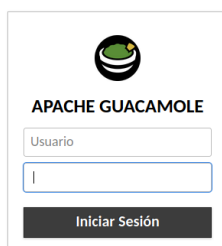
### Comandos predefinidos

Para facilitar el uso del entorno se han desarrollado dos scripts en BASH, el primero de ellos para levantar completamente el entorno y el segundo para apagarlo. Estos scripts se ubicarán en el servidor en el que corre Docker y su uso tal y como se indica en los diferentes casos de uso, se encuentra limitado al Administrador, ya que su ejecución podría dejar a los usuarios sin acceso al Sistema.

### Portal de acceso Apache Guacamole

El portal Guacamole es el punto de acceso al Sistema tanto para los usuarios como para el Administrador. Para ello todos han de acceder mediante un navegador web utilizando la dirección proporcionada a cada uno de ellos.

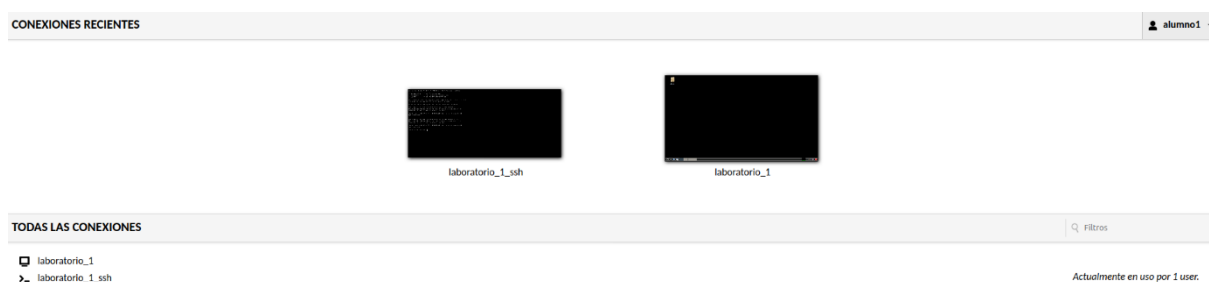
Una vez introducida el portal mostrará una pantalla de bienvenida en la que se ha de introducir usuario y contraseña (Figura 10). Estas credenciales se habrían facilitado previamente a cada interesado. Una vez dentro dispondrá de acceso a los recursos que el Administrador disponga para su uso. En el piloto se tratará de dos conexiones, una por VNC y otra a través de SSH.

**Figura 10**

*Página de inicio del portal Guacamole*

*Fuente: Imagen propia.*

Una vez en el interior de este, el usuario observará en su pantalla de inicio los siguientes datos, tal y como puede observarse en la Figura 12.

**Figura 11**

*Home del usuario tipo “alumno”*

*Fuente: Imagen propia.*

En la Figura 11 puede observarse como el usuario tipo “alumno” puede acceder a los dos tipos de conexión bien en el apartado inferior de la misma o, si previamente ha hecho uso de ellas, a través del acceso que proporciona en “conexiones recientes”. En la parte superior derecha figurará el nombre del usuario, así como unos datos básicos.

En el caso del administrador, además de observar lo anterior tendrá habilitadas las opciones correspondientes a su rol, donde podrá crear usuario y/o conexiones para dar funcionalidad al sistema, estas distintas pantallas se muestran las Figuras 12 y 13.

Figura 12

## CONFIGURACIONES

Sesiones Activas
Historial
Usuarios
Groups
**Conexiones**
Preferencias

Haga clic o toque en una de las conexiones de abajo para gestionar esa conexión. Dependiendo de

+ Nueva Conexión
+ Nuevo Grupo

Filtros

+

laboratorio\_1

+

laboratorio\_1\_ssh

+

laboratorio\_2

+

laboratorio\_2\_ssh

+

laboratorio\_3

+

laboratorio\_3\_ssh

+

laboratorio\_4

+

laboratorio\_4\_ssh

+

laboratorio\_5

+

laboratorio\_5\_ssh

Formulario de creación de Conexiones.

Fuente: Imagen propia.

Figura 13

### EDITAR USUARIO

Nombre de usuario: alumno1

Contraseña:

Validar Contraseña:

### PERFIL

Nombre completo:  Alumno 1

Correo electrónico:  alumno1@correo.es

Organización:  universidad

Puesto:  alumno

Formulario de creación de Usuarios.

Fuente: Imagen propia.

Además de lo anterior, el Administrador dispone también de diversas pantallas informativas donde puede obtenerse información acerca de los distintos elementos generados en Guacamole o bien del uso que se les está dando al Sistema, en las Figuras 14, 15, 16 y 17 pueden observarse los distintos paneles de información, que incluyen el listado de usuarios creados, las conexiones habilitadas, las sesiones activas y el historial de acceso.

**Figura 14**

Usuario	Organización	Nombre completo	Last active
alumno1	universidad	Alumno 1	2021-04-08 10:56:07
alumno2	universidad	Alumno 2	2021-04-07 16:55:47
alumno3	universidad	Alumno 3	
alumno4	universidad	Alumno 4	
alumno5	universidad	Alumno 5	
guacadmin			2021-05-03 08:46:03

*Listado de usuarios*

*Fuente: Imagen propia.*

**Figura 15**

Nombre completo	Last active	Gestionar
laboratorio_1		
laboratorio_1_ssh		
laboratorio_2		
laboratorio_2_ssh		
laboratorio_3		
laboratorio_3_ssh		
laboratorio_4		
laboratorio_4_ssh		
laboratorio_5		
laboratorio_5_ssh		

*Listado de conexiones.*

*Fuente: Imagen propia.*

Figura 16



### Sesiones activas

Fuente: Imagen propia.

Figura 17

CONFIGURACIONES				
Sesiones Activas	Historial	Usuarios	Groups	Conexiones
<p>Aquí se lista el historial de las últimas conexiones y se pueden ordenar haciendo clic en los encabezados de la columna. Para buscar un registro específico, introduzca la cadena de texto a filtrar y haga clic en "Buscar". Solo se listarán los registros que coincidan con el filtro introducido.</p>				
Usuario	Activo Desde	Duración	Nombre de conexión	Host Remoto
guacadmin	03-05-2021 08:43:18	13.7 seconds	laboratorio_5	10.0.0.10
guacadmin	03-05-2021 08:43:13	18.3 seconds	laboratorio_4	10.0.0.10
guacadmin	03-05-2021 08:43:07	24.4 seconds	laboratorio_3	10.0.0.10
alumno1	08-04-2021 10:56:08	20 minutes	laboratorio_1	10.0.0.7
alumno1	08-04-2021 10:54:26	22.8 seconds	laboratorio_1	10.0.0.7
alumno1	08-04-2021 10:49:58	29.6 seconds	laboratorio_1	10.0.0.7
guacadmin	07-04-2021 22:00:24	9.7 minutes	laboratorio_2_ssh	10.0.0.7

### Historial.

Fuente: Imagen propia.

### Política de usuarios

El Sistema se basa en dos tipos de usuarios con funciones y funcionalidades diferenciadas. El primero de ellos, denominado Administrador, realiza las funciones de mantenimiento y configuración del entorno virtual, así como dar de alta/baja usuarios y recursos en el Sistema. El segundo tipo de usuario es el denominado genéricamente como Usuario, este actor se conectará al Sistema y utilizará los recursos que el Administrador le asigne. No dispondrá de ninguna capacidad de configuración y/o modificación en el sistema más allá de su carpeta personal en el recurso asignado.

### Persistencia de datos

El Sistema incluye dos sistemas de persistencia de datos diferenciados. Por un lado, se dispone de una carpeta común a todos los puestos virtuales que se monta en el sistema en modo "sólo lectura" y en cuyo interior el Administrador podrá depositar diferentes recursos documentales

para que tengan acceso todos los puestos virtuales. Esta carpeta únicamente puede ser modificada desde el Servidor, por lo que el único con acceso de escritura a la misma será el Administrador del Sistema.

En segundo lugar, se dispone de una copia persistente de la carpeta “home” del usuario en cada uno de los sistemas virtuales. Esto es así para que exista persistencia en los datos y ficheros que los usuarios crearán en los recursos asignados de su máquina.

Estas dos carpetas se almacenan físicamente en el Servidor que alberga el Sistema, por lo que el acceso y modificación de estas es realizada por el Administrador.

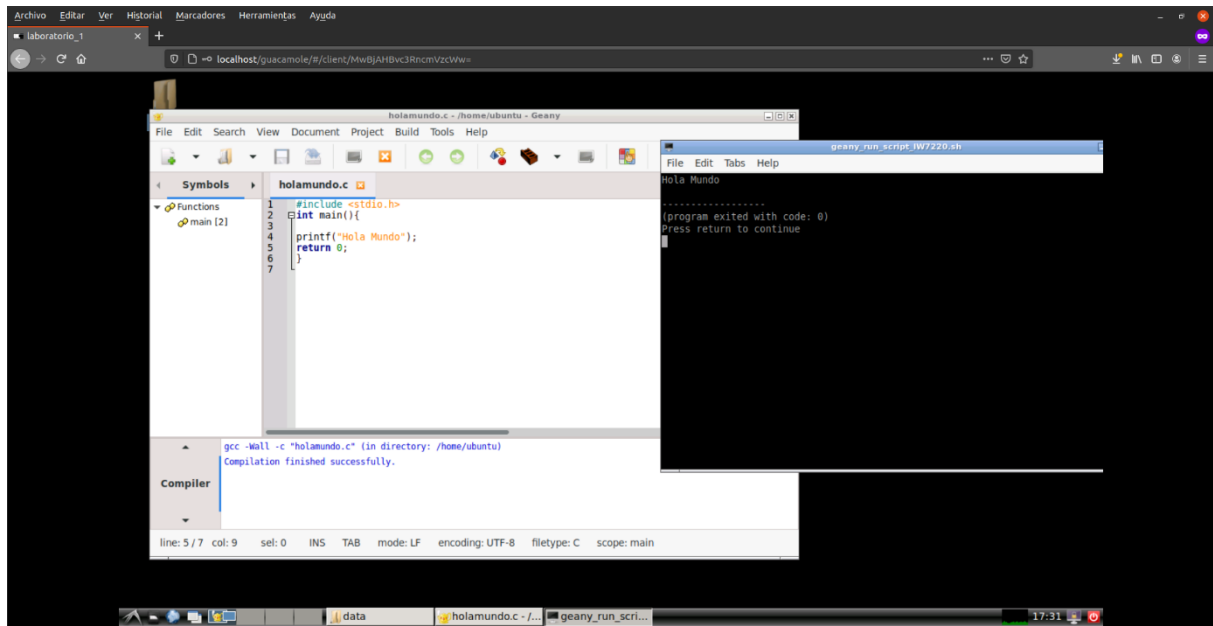
### Funcionalidades del Laboratorio

En la versión piloto del laboratorio se han establecido unas funciones básicas de cara a probar la funcionalidad y rendimiento del entorno. Estas funciones básicas aportan al usuario dos tipos de acceso a cada máquina, uno por SSH y otro mediante VNC al contenedor que ejecuta UBUNTU 20.04 utilizando para ello el entorno gráfico LXDE, que es un entorno especialmente ligero y que aporta una funcionalidad suficiente para que el usuario pueda trabajar con el entorno.

El entorno dispone de un usuario sin privilegios, denominado “ubuntu” que sirve como usuario base para realizar las labores deseadas con el Sistema. Este usuario no está personalizado ya que no es necesario al ser cada puesto de trabajo un nuevo contenedor independiente del resto.

Este entorno incluye para la versión piloto a evaluar un conjunto limitado de paquetes software, entre los que se incluyen diversos compiladores y el editor de código Geany (Figura 18), con el objetivo de aportar cierta funcionalidad de entorno de desarrollo. También se incluye el lector de documentos Evince.

Todo este software es configurable a nivel de imagen Docker, pudiendo disponer de una imagen para cada tipo de laboratorio que se desee implementar, o bien una imagen más completa que permita utilizarse para diferentes funciones, teniendo en cuenta que, a mayor número de aplicaciones, se requerirá un mayor número de recursos, por lo que será preciso valorar en cada caso la necesidad de incluir dichas funcionalidades.

**Figura 18**

*Ejemplo de compilación en C*

*Fuente: Imagen propia.*

### 3.4. PRUEBAS

#### Introducción

A lo largo de todo este TFG se ha desarrollado el argumento de que un entorno virtualizado con contenedores ofrecería ventajas de rendimiento y optimización de recursos frente al tradicional laboratorio físico o bien virtualizado. Por ello se ha planteado una serie de pruebas al entorno para evaluar el rendimiento, usando como servidor una máquina no optimizada para un entorno profesional. Estas pruebas tienen como objetivo evaluar la eficiencia del Sistema, así como el consumo de recursos en tiempo real de un entorno con una infraestructura definida previamente.

Se realizarán pruebas de tiempo de respuesta, de consumo de CPU y de consumo de memoria en tiempo real.

#### Equipo de Pruebas

El equipo de pruebas se trata de un portátil con las características técnicas que pueden observarse en la Tabla 2.



**Tabla 2***Características del equipo de pruebas*

Disco	SSD 250GB
CPU	Intel Core i5 8250U
Frecuencia	1,80Ghz
SO	Ubuntu 20.04
RAM	8 GB DDR3

En dicha máquina se ha instalado el SO Ubuntu 20.04 así como el motor Docker que permitirá ejecutar el entorno basado en contenedores.

**Entorno de pruebas**

Para la realización de las pruebas se ha optado por una arquitectura de servidor y 5 puestos de trabajo, todos ellos accesibles mediante SSH y VNC . Estos puestos han estado activos en todo momento de las pruebas y se ha accedido simultáneamente a 4 de los mismos con diferentes cargas de trabajo, con el objetivo de simular un entorno de trabajo lo más cercano a un uso real del sistema.

Además de lo anterior, se utilizarán cinco máquinas virtuales Ubuntu 20.04 creadas con el software Oracle VirtualBox para realizar pruebas de rendimiento y consumo de recursos, para evaluar posteriormente el rendimiento de ambos entornos. Como equipo anfitrión se utilizará el mismo que en las pruebas de rendimiento del laboratorio virtual.

Según la documentación oficial de Ubuntu, los requisitos mínimos para que el Sistema Operativo funcione correctamente son los que figuran en la Tabla 3.

**Tabla 3***Requisitos mínimos de SO Ubuntu*

	<b>Mínimo</b>	<b>Recomendado</b>
<b>Espacio HDD</b>	8 GB	20 GB
<b>CPU</b>	AMD 64 bits o superior	Intel Dual Core 1,5Ghz o Superior
<b>Memoria RAM</b>	512 MB	2 GB

Para la realización de las pruebas se ha optado con una configuración intermedia, utilizando para ello máquinas virtuales de 1 GB de memoria RAM y 10 GB de espacio en el disco duro.

## Pruebas de rendimiento del laboratorio virtual

Dentro de la categoría de “rendimiento”, se realiza una medición del tiempo de puesta en marcha y apagado del entorno. Para el primer caso, tal y como se observa en la Figura 19, el tiempo de encendido es de 8,543 segundos. En cuanto al apagado del entorno, el tiempo asciende hasta los 13,052 segundos, como se observa en la Figura 20.

**Figura 19**

```

hector@hector-Mi:~/tfg$ time sh levantar_entorno.sh
Levantando el servicio de guacamole...

Creating network "tfg_lab" with driver "bridge"
Creating lab3      ... done
Creating lab4      ... done
Creating guacamole-db ... done
Creating lab5      ... done
Creating lab1      ... done
Creating lab2      ... done
Creating guacamole-daemon ... done
Creating guacamole-web ... done
Creating guacamole-proxy ... done

Levantando los puestos de trabajo...

Entorno de trabajo en funcionamiento.

real    0m8,543s

```

*Tiempo requerido para levantar el entorno*

*Fuente: Imagen propia.*

**Figura 20**

```

hector@hector-Mi:~/tfg$ time sh apagar_entorno.sh
apagando el servicio de guacamole y los puestos de trabajo...

Stopping guacamole-proxy ... done
Stopping guacamole-web ... done
Stopping guacamole-daemon ... done
Stopping lab4 ... done
Stopping lab1 ... done
Stopping lab2 ... done
Stopping lab5 ... done
Stopping guacamole-db ... done
Stopping lab3 ... done
Removing guacamole-proxy ... done
Removing guacamole-web ... done
Removing guacamole-daemon ... done
Removing lab4 ... done
Removing lab1 ... done
Removing lab2 ... done
Removing lab5 ... done
Removing guacamole-db ... done
Removing lab3 ... done
Removing network tfg_lab

Entorno de trabajo apagado.

real    0m13,052s

```

*Tiempo requerido para apagar el entorno*

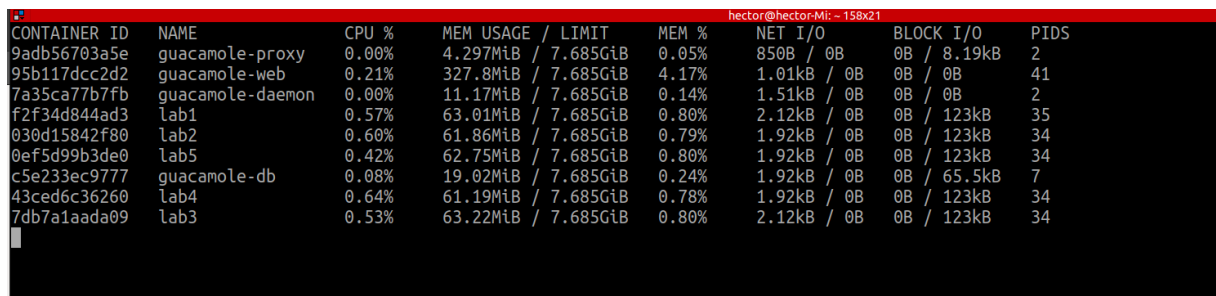
*Fuente: Imagen propia.*

## Pruebas de consumo de recursos del laboratorio virtual

Para medir el rendimiento del sistema en uso y su capacidad de optimización de recursos, se realizan diferentes pruebas con diferentes cargas de trabajo para evaluar el rendimiento del sistema.

En la primera de ellas, representada en la Figura 21, se muestra el consumo de recursos del sistema en reposo, siendo este de 671 MB de memoria y un uso de CPU del 3%.

**Figura 21**



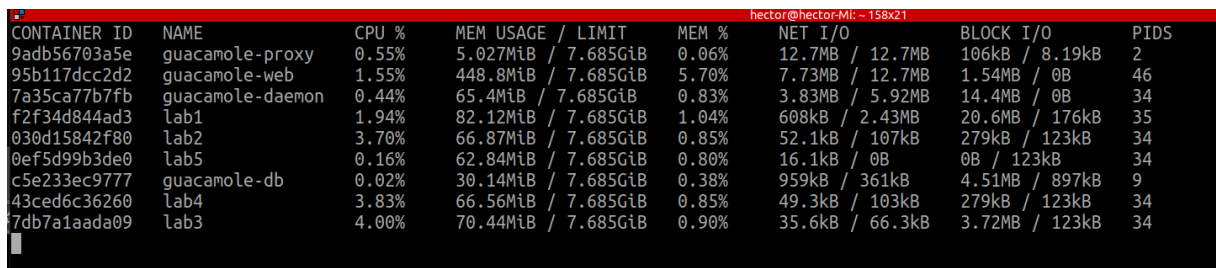
CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O	BLOCK I/O	PIDS
9adb56703a5e	guacamole-proxy	0.00%	4.297MiB / 7.685GiB	0.05%	850B / 0B	0B / 8.19kB	2
95b117dcc2d2	guacamole-web	0.21%	327.8MiB / 7.685GiB	4.17%	1.01kB / 0B	0B / 0B	41
7a35ca77b7fb	guacamole-daemon	0.00%	11.17MiB / 7.685GiB	0.14%	1.51kB / 0B	0B / 0B	2
f2f34d844ad3	lab1	0.57%	63.01MiB / 7.685GiB	0.80%	2.12kB / 0B	0B / 123kB	35
030d15842f80	lab2	0.60%	61.86MiB / 7.685GiB	0.79%	1.92kB / 0B	0B / 123kB	34
0ef5d99b3de0	lab5	0.42%	62.75MiB / 7.685GiB	0.80%	1.92kB / 0B	0B / 123kB	34
c5e233ec9777	guacamole-db	0.08%	19.02MiB / 7.685GiB	0.24%	1.92kB / 0B	0B / 65.5kB	7
43ced6c36260	lab4	0.64%	61.19MiB / 7.685GiB	0.78%	1.92kB / 0B	0B / 123kB	34
7db7a1aada09	lab3	0.53%	63.22MiB / 7.685GiB	0.80%	2.12kB / 0B	0B / 123kB	34

*Consumo de recursos en reposo.*

*Fuente: Imagen propia.*

Tras la prueba en reposos se mide el consumo del sistema con un uso intenso del mismo, esto es utilizando cuatro de los puestos de trabajo puestos de trabajo, primeramente, en entorno gráfico, este consumo se representa en la Figura 22 y da como resultado un consumo de 894 MB de memoria y un uso de CPU del 16%.

**Figura 22**



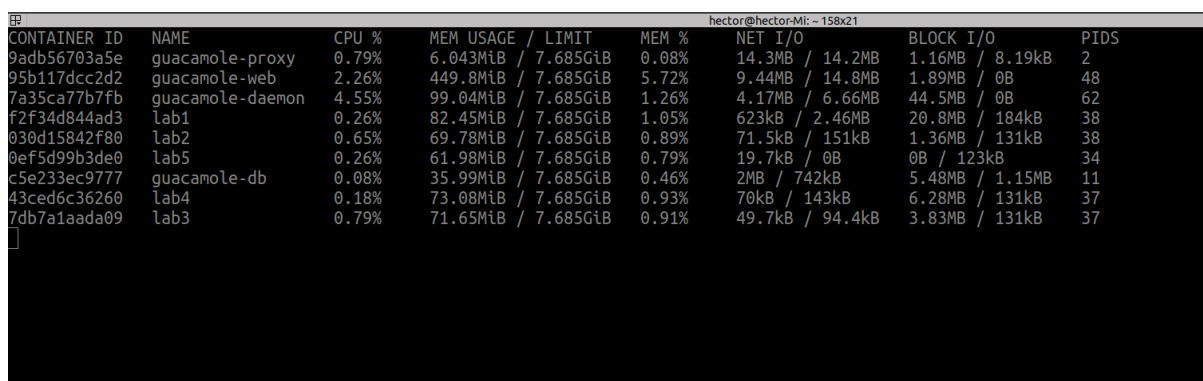
CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O	BLOCK I/O	PIDS
9adb56703a5e	guacamole-proxy	0.55%	5.027MiB / 7.685GiB	0.06%	12.7MB / 12.7MB	106kB / 8.19kB	2
95b117dcc2d2	guacamole-web	1.55%	448.8MiB / 7.685GiB	5.70%	7.73MB / 12.7MB	1.54MB / 0B	46
7a35ca77b7fb	guacamole-daemon	0.44%	65.4MiB / 7.685GiB	0.83%	3.83MB / 5.92MB	14.4MB / 0B	34
f2f34d844ad3	lab1	1.94%	82.12MiB / 7.685GiB	1.04%	608kB / 2.43MB	20.6MB / 176kB	35
030d15842f80	lab2	3.70%	66.87MiB / 7.685GiB	0.85%	52.1kB / 107kB	279kB / 123kB	34
0ef5d99b3de0	lab5	0.16%	62.84MiB / 7.685GiB	0.80%	16.1kB / 0B	0B / 123kB	34
c5e233ec9777	guacamole-db	0.02%	30.14MiB / 7.685GiB	0.38%	959kB / 361kB	4.51MB / 897kB	9
43ced6c36260	lab4	3.83%	66.56MiB / 7.685GiB	0.85%	49.3kB / 103kB	279kB / 123kB	34
7db7a1aada09	lab3	4.00%	70.44MiB / 7.685GiB	0.90%	35.6kB / 66.3kB	3.72MB / 123kB	34

*Consumo de recursos con 4 puestos activos por entorno gráfico*

*Fuente: Imagen propia.*

Para medir el rendimiento del entorno “SSH” se mide el consumo de recursos en los mismos puestos que en la prueba anterior, dando como resultado los datos que se muestran en la Figura 23, siendo un consumo total de 950 MB de memoria y un 10% de consumo de CPU.

**Figura 23**



CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O	BLOCK I/O	PIDS
9adb56703a5e	guacamole-proxy	0.79%	6.043MiB / 7.685GiB	0.08%	14.3MB / 14.2MB	1.16MB / 8.19kB	2
95b117dcc2d2	guacamole-web	2.26%	449.8MiB / 7.685GiB	5.72%	9.44MB / 14.8MB	1.89MB / 0B	48
7a35ca77b7fb	guacamole-daemon	4.55%	99.04MiB / 7.685GiB	1.26%	4.17MB / 6.66MB	44.5MB / 0B	62
f2f34d844ad3	lab1	0.26%	82.45MiB / 7.685GiB	1.05%	623kB / 2.46MB	20.8MB / 184kB	38
030d15842f80	lab2	0.65%	69.78MiB / 7.685GiB	0.89%	71.5kB / 151kB	1.36MB / 131kB	38
0ef5d99b3de0	lab5	0.26%	61.98MiB / 7.685GiB	0.79%	19.7kB / 0B	0B / 123kB	34
c5e233ec9777	guacamole-db	0.08%	35.99MiB / 7.685GiB	0.46%	2MB / 742kB	5.48MB / 1.15MB	11
43ced6c36260	lab4	0.18%	73.08MiB / 7.685GiB	0.93%	70kB / 143kB	6.28MB / 131kB	37
7db7a1aada09	lab3	0.79%	71.65MiB / 7.685GiB	0.91%	49.7kB / 94.4kB	3.83MB / 131kB	37

*Consumo de recursos con 4 puestos por protocolo SSH*

*Fuente: Imagen propia.*

Finalmente, en cuanto al espacio en disco del sistema, y debido a que por la arquitectura de Docker no se asigna un espacio concreto a cada contenedor, sino que se usa dinámicamente, no se realizan cálculos de este.

### Pruebas de consumo de rendimiento de una máquina virtual

Para comparar el rendimiento del laboratorio con una máquina virtual, se realizan las pruebas de encendido y apagado de la misma, calculando el tiempo que le lleva a cada una de las tareas. Estas tareas se realizan utilizando la interfaz del software Oracle VirtualBox, puede observarse esta distribución en la Figura 24.

Figura 24



Oracle VirtualBox con máquinas virtuales Ubuntu

Fuente: Imagen propia.

Tiempo de encendido del Sistema: 1min 29seg.

Tiempo de apagado del Sistema: 9,28 seg.

Se destaca también que el consumo de recursos durante el encendido de las máquinas virtuales se ha disparado ralentizando el sistema anfitrión por completo, tal y como puede observarse en la Figura 25.

Figura 25

Nombre	Estado	95% CPU	88% Memoria	28% Disco	0% Red	10% GPU	Motor de GPU	Consumo de energía	Tendencia de ...
Microsoft Word (2)		0,4%	86,3 MB	0 MB/s	0 Mbps	0%	GPU 0 - 3D	Muy baja	Muy baja
VirtualBox Virtual Machine		21,0%	85,3 MB	10,4 MB/s	0 Mbps	0%		Muy alta	Bajo
VirtualBox Virtual Machine		0%	1,2 MB	0 MB/s	0 Mbps	0%		Muy baja	Muy baja
VirtualBox Virtual Machine		0%	1,6 MB	0 MB/s	0 Mbps	0%		Muy baja	Muy baja
VirtualBox Virtual Machine		15,6%	91,2 MB	7,5 MB/s	0 Mbps	0%		Muy alta	Bajo
VirtualBox Virtual Machine		0%	1,2 MB	0 MB/s	0 Mbps	0%		Muy baja	Muy baja
VirtualBox Virtual Machine		0%	1,6 MB	0 MB/s	0 Mbps	0%		Muy baja	Muy baja
VirtualBox Virtual Machine		24,1%	92,6 MB	15,6 MB/s	0 Mbps	0%		Muy alta	Bajo
VirtualBox Virtual Machine		0%	1,2 MB	0 MB/s	0 Mbps	0%		Muy baja	Muy baja
VirtualBox Virtual Machine		0%	1,6 MB	0 MB/s	0 Mbps	0%		Muy baja	Muy baja
VirtualBox Virtual Machine		20,3%	92,4 MB	14,3 MB/s	0 Mbps	0%		Muy alta	Bajo
VirtualBox Virtual Machine		0%	1,2 MB	0 MB/s	0 Mbps	0%		Muy baja	Muy baja
VirtualBox Virtual Machine		0%	1,6 MB	0 MB/s	0 Mbps	0%		Muy baja	Muy baja

Consumo de recursos durante el encendido del sistema

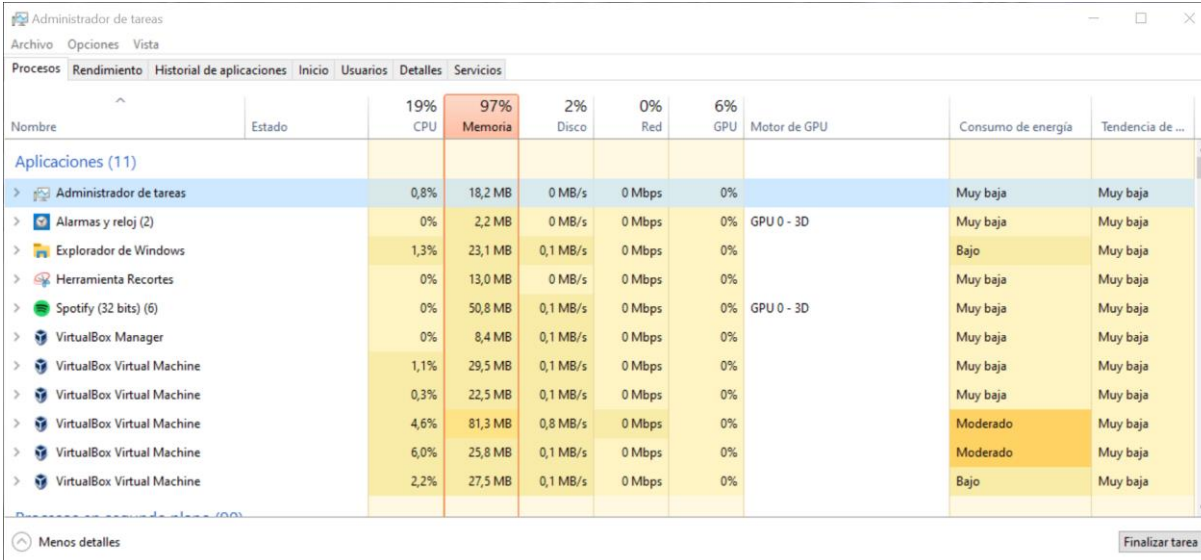
Fuente: Imagen propia.

## Pruebas de consumo de recursos de una máquina virtual

Se realizan mediciones de consumo de recursos con las máquinas virtuales en distinto estado de funcionamiento, esto es en reposo y en carga de trabajo, dando como resultado los siguientes datos:

En reposo, teniendo en cuenta como tal el estado en el que la máquina se encuentra en la pantalla de inicio sin haber ejecutado o abierto ningún tipo de recursos, el entorno compuesto por máquinas virtuales ofrece un consumo de CPU del 14,2%, tal y como se aprecia en la Figura 26. En cuanto a la memoria RAM su consumo es de 5GB, ya que se ha reservado 1024MB para cada una de las máquinas virtuales.

**Figura 26**



The screenshot shows the Windows Task Manager Performance tab. The 'CPU' section shows 19% usage. The 'Memory' section shows 97% usage. The 'Disk' section shows 2% usage. The 'Network' section shows 0% usage. The 'GPU' section shows 6% usage. The 'GPU Motor' section shows 'GPU 0 - 3D' and 'GPU 0 - 3D'. The 'Energy' section shows 'Muy baja' and 'Muy baja'. The 'Tendencia de...' section shows 'Muy baja' and 'Muy baja'.

Nombre	Estado	19% CPU	97% Memoria	2% Disco	0% Red	6% GPU	Motor de GPU	Consumo de energía	Tendencia de ...
<b>Aplicaciones (11)</b>									
Administrador de tareas		0,8%	18,2 MB	0 MB/s	0 Mbps	0%		Muy baja	Muy baja
Alarmas y reloj (2)		0%	2,2 MB	0 MB/s	0 Mbps	0%	GPU 0 - 3D	Muy baja	Muy baja
Explorador de Windows		1,3%	23,1 MB	0,1 MB/s	0 Mbps	0%		Bajo	Muy baja
Herramienta Recortes		0%	13,0 MB	0 MB/s	0 Mbps	0%		Muy baja	Muy baja
Spotify (32 bits) (6)		0%	50,8 MB	0,1 MB/s	0 Mbps	0%	GPU 0 - 3D	Muy baja	Muy baja
VirtualBox Manager		0%	8,4 MB	0,1 MB/s	0 Mbps	0%		Muy baja	Muy baja
VirtualBox Virtual Machine		1,1%	29,5 MB	0,1 MB/s	0 Mbps	0%		Muy baja	Muy baja
VirtualBox Virtual Machine		0,3%	22,5 MB	0,1 MB/s	0 Mbps	0%		Muy baja	Muy baja
VirtualBox Virtual Machine		4,6%	81,3 MB	0,8 MB/s	0 Mbps	0%		Moderado	Muy baja
VirtualBox Virtual Machine		6,0%	25,8 MB	0,1 MB/s	0 Mbps	0%		Moderado	Muy baja
VirtualBox Virtual Machine		2,2%	27,5 MB	0,1 MB/s	0 Mbps	0%		Bajo	Muy baja

### Consumo de recursos en reposo

*Fuente: Imagen propia.*

En estado de carga de trabajo, entendiendo como tal el uso de aplicaciones ofimáticas o de cierta carga de trabajo en el sistema, las máquinas virtuales ofrecen un consumo de recursos de 90% de CPU, así como los 5GB de memoria RAM reservadas para el conjunto de máquinas virtuales, tal y como se observa en la Figura 27.

**Figura 27**

The screenshot shows the Windows Task Manager Performance tab. The top bar indicates overall system usage: CPU 100%, Memory 96%, Disk 33%, Network 0%, GPU 6%, and GPU Motor 0%. The main table lists 11 applications and their resource consumption. The 'VirtualBox Virtual Machine' entries show high CPU usage (17.4% to 19.3%) and high energy consumption ('Muy alta'), while other applications like 'Administrador de tareas' and 'Alarmas y reloj' show low usage and energy consumption ('Muy baja').

Nombre	Estado	100% CPU	96% Memoria	33% Disco	0% Red	6% GPU	Motor de GPU	Consumo de energía	Tendencia de ...
<b>Aplicaciones (11)</b>									
Administrador de tareas		1,0%	18,3 MB	0 MB/s	0 Mbps	0%		Muy baja	Muy baja
Alarmas y reloj (2)		0%	1,9 MB	0,1 MB/s	0 Mbps	0%	GPU 0 - 3D	Muy baja	Muy baja
Explorador de Windows		0,3%	13,9 MB	0,1 MB/s	0 Mbps	0%		Muy baja	Muy baja
Herramienta Recortes		0,8%	2,6 MB	0,1 MB/s	0 Mbps	0%		Muy baja	Muy baja
Spotify (32 bits) (6)		0,5%	68,1 MB	0,4 MB/s	0 Mbps	0%	GPU 0 - 3D	Muy baja	Muy baja
VirtualBox Manager		0%	12,8 MB	0,1 MB/s	0 Mbps	0%		Muy baja	Muy baja
VirtualBox Virtual Machine		17,4%	28,2 MB	14,6 MB/s	0 Mbps	0%		Muy alta	Bajo
VirtualBox Virtual Machine		17,7%	23,3 MB	2,3 MB/s	0 Mbps	0%		Muy alta	Bajo
VirtualBox Virtual Machine		18,6%	94,7 MB	17,8 MB/s	0 Mbps	0%		Muy alta	Bajo
VirtualBox Virtual Machine		19,3%	29,2 MB	20,1 MB/s	0 Mbps	0%		Muy alta	Bajo
VirtualBox Virtual Machine		17,0%	23,0 MB	9,3 MB/s	0 Mbps	0%		Muy alta	Bajo

*Consumo de recursos de las máquinas virtuales con carga de trabajo*

*Fuente: Imagen propia.*

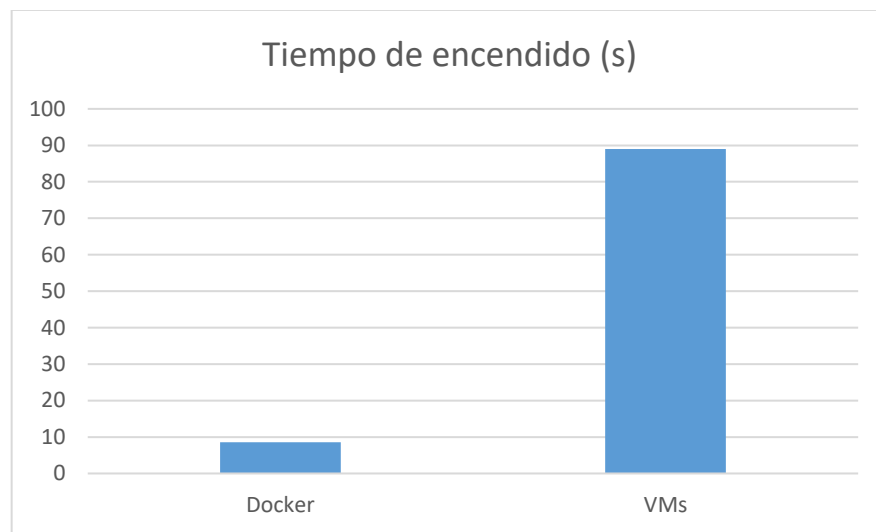
Finalmente, teniendo en cuenta que cada máquina virtual tiene asignado un espacio en disco de 10GB, se consumen permanentemente 50GB de disco duro al tener el sistema activo.

## 4. COMPARATIVA Y EVALUACIÓN

### 4.1. COMPARATIVA

Para comparar el rendimiento de ambas configuraciones de sistema se han tenido en cuenta los datos obtenidos de las pruebas de rendimiento y consumo de recurso de ambos entornos, dando como resultado los siguientes datos que se exponen a continuación (Figuras 28 a 34).

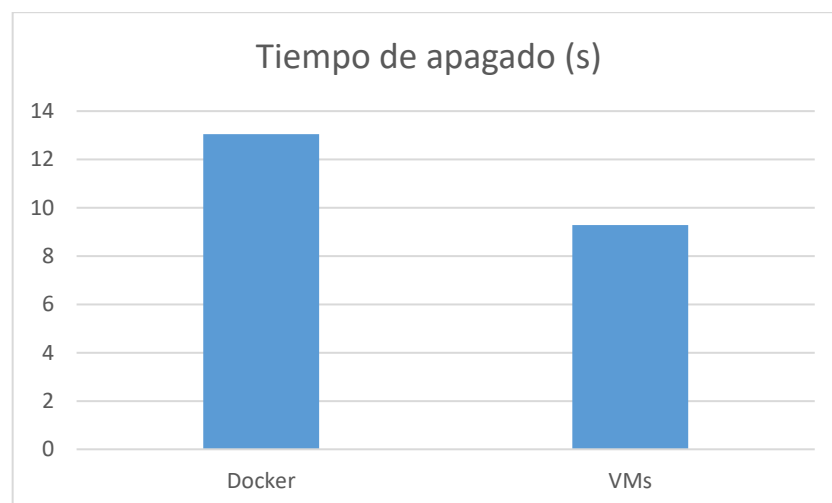
**Figura 28**



*Gráfico comparativo de tiempo de encendido*

*Fuente: Imagen propia.*

**Figura 29**



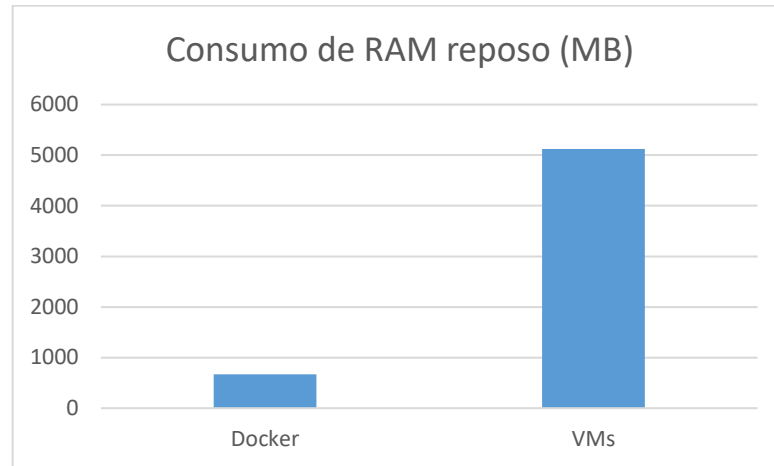
Evaluación del uso de la tecnología de contenedores para la creación de un aula virtual



*Gráfico comparativo de tiempo de apagado*

*Fuente: Imagen propia.*

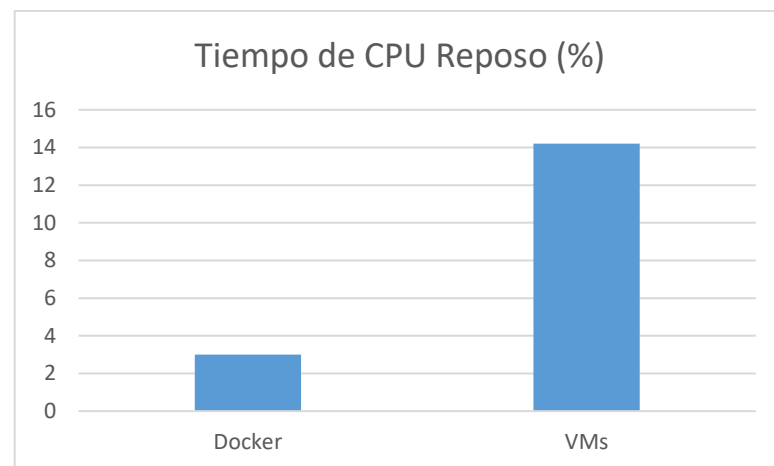
**Figura 30**



*Gráfico comparativo de consumo de RAM en reposo*

*Fuente: Imagen propia.*

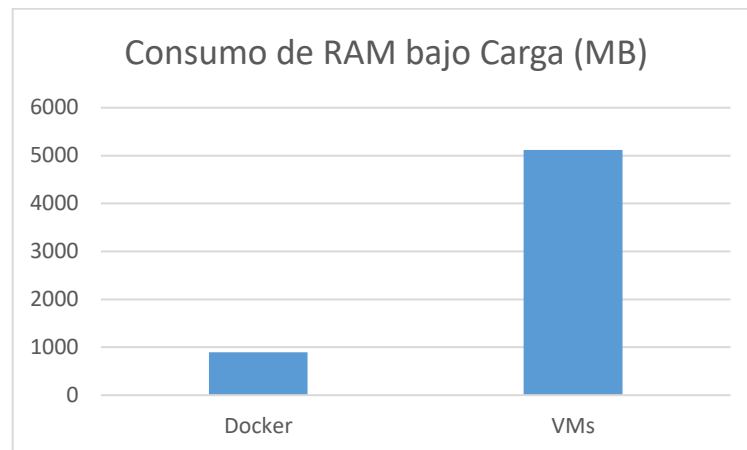
**Figura 31**



*Gráfico comparativo de tiempo de CPU en reposo*

*Fuente: Imagen propia.*

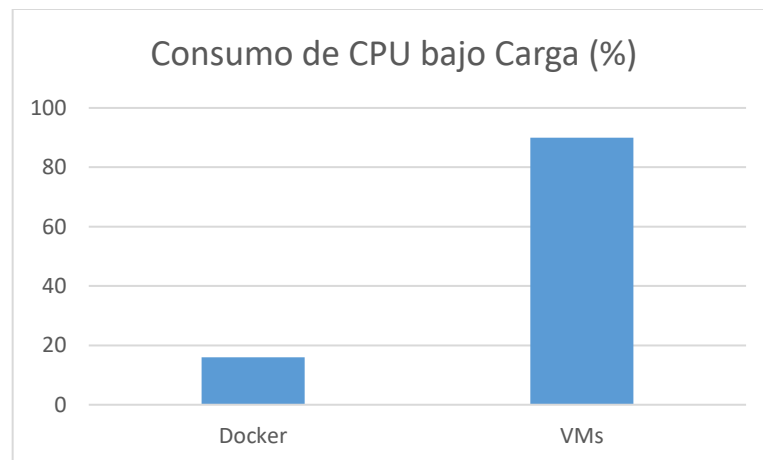
**Figura 32**



*Gráfico comparativo de consumo de RAM en carga*

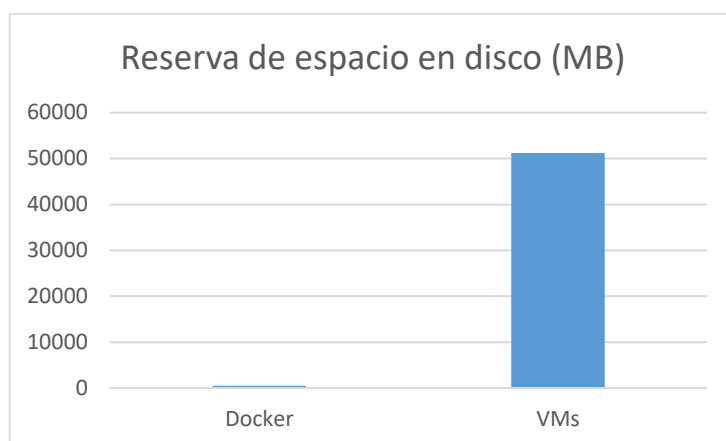
*Fuente: Imagen propia.*

**Figura 33**



*Gráfico comparativo de consumo de CPU en carga*

*Fuente: Imagen propia.*

**Figura 34***Gráfico comparativo de reserva de espacio en Disco**Fuente: Imagen propia.*

## 4.2. EVALUACIÓN

El sistema dispone de un rendimiento y puesta en marcha/apagado muy rápido, siendo este menor a los 15 segundos, aunque los datos de apagado del sistema son mejores en un entorno virtualizado la comparativa de este último dato es parejo en ambas configuraciones.

El consumo de memoria del sistema por puesto de trabajo es muy bajo, menor de 100MB por puesto, lo que facilita el escalado del sistema y permite su implementación sin la necesidad de disponer de una gran infraestructura.

El consumo de CPU es bajo, por debajo del 20% bajo carga de trabajo, lo que al igual que en el punto anterior es una ventaja de cara a la escalabilidad y aprovechamiento de los recursos disponibles.

El gasto de espacio en disco es dinámico en función de las necesidades del sistema, por lo que no es necesario disponer de un espacio reservado específicamente para cada entorno, por lo que favorece de nuevo la facilidad para escalar el sistema y dimensionarlo de manera adecuada.

## 4.3. EVALUACIÓN EN ENTORNO REAL

Para la evaluación del entorno en un entorno controlado, pero con datos reales, se ha configurado una pequeña infraestructura piloto que pueda alojar un número reducido de

laboratorios para efectuar una práctica consistente en una instalación en un sistema operativo Ubuntu un servidor XAMPP y WordPress. Esta sirve para evaluar el propósito de la infraestructura estudiada a lo largo de este documento.

Para las pruebas en entorno real se ha utilizado un servidor ubicado en la nube con disponibilidad para 4 CPU y 4 GB de memoria RAM, compartiendo estos recursos para toda la infraestructura, consistente en 3 laboratorios con 4 usuarios para cada uno de los mismos.

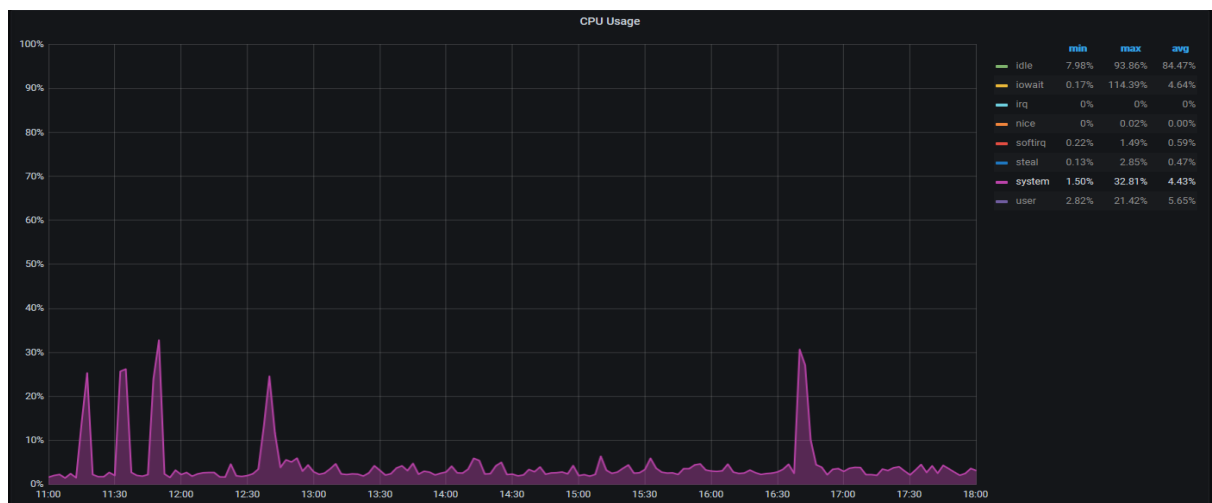
Una vez realizadas las pruebas con el entorno debidamente desplegado se han obtenido unos resultados satisfactorios en su funcionamiento, los que se muestran a continuación.

### Rendimiento de la máquina servidor

Se ha realizado un estudio de rendimiento de la máquina servidor, este estudio consiste en la monitorización del consumo de CPU y memoria RAM, obteniéndose los siguientes datos.

Tal y como puede observarse en la Figura 35, se aprecia un consumo de CPU bajo, teniendo sus mayores picos en los periodos de trabajo con los laboratorios, alcanzando picos de 33% de uso.

**Figura 35**



*Consumo de CPU en servidor del entorno piloto*

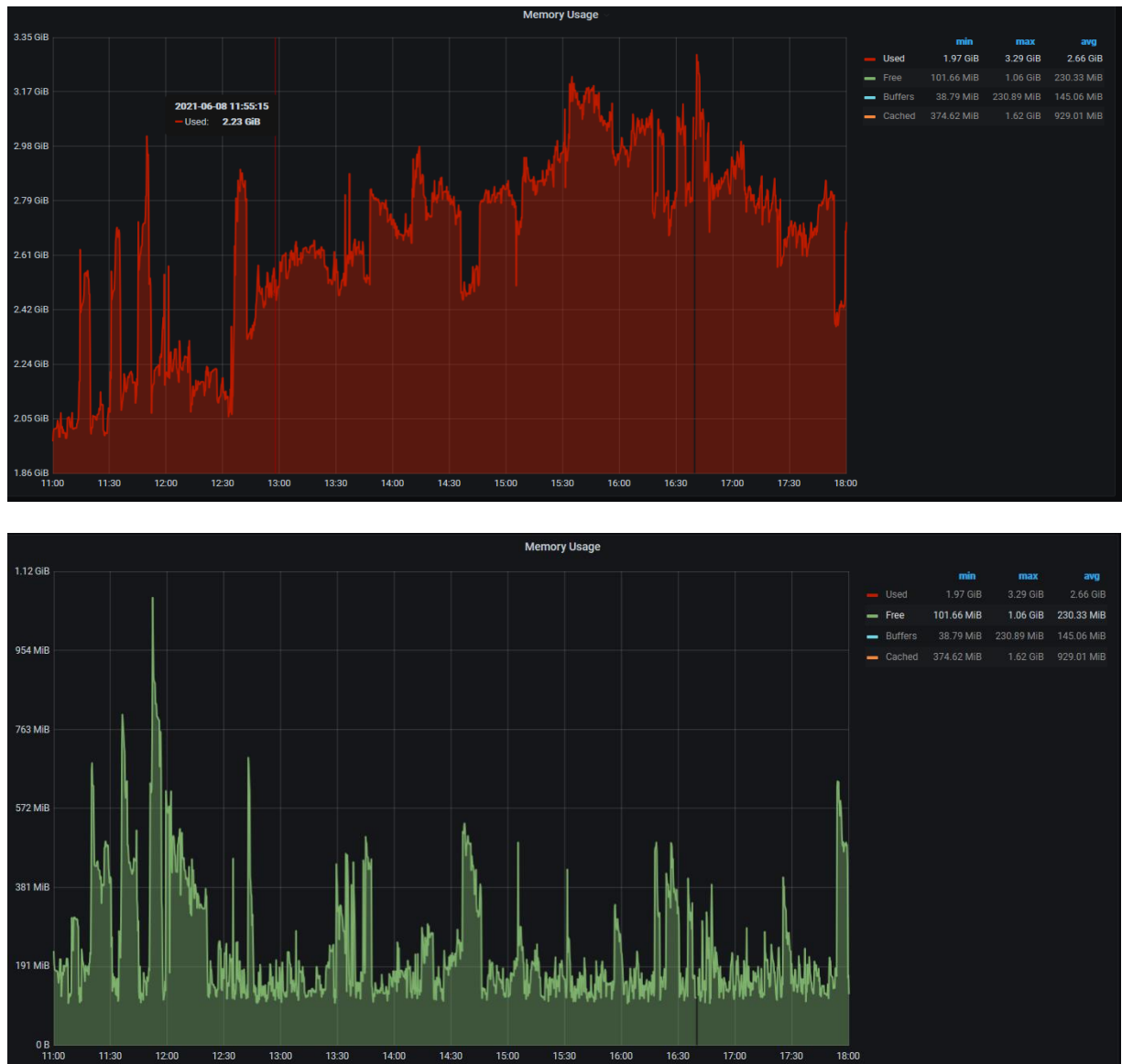
*Fuente: Imagen propia.*

En cuanto al uso de la memoria, se aprecia como a medida que los laboratorios están en funcionamiento y se emplean para instalar software, la memoria requerida para su funcionamiento va en aumento, reduciendo su uso una vez finalizan las tareas de instalación

Evaluación del uso de la tecnología de contenedores para la creación de un aula virtual

que requieren las tareas piloto, este uso puede apreciarse en la Figuras 36, donde se relaciona la memoria usada con la libre a lo largo de la prueba piloto.

**Figura 36**



### *Consumo de Memoria en servidor del entorno piloto*

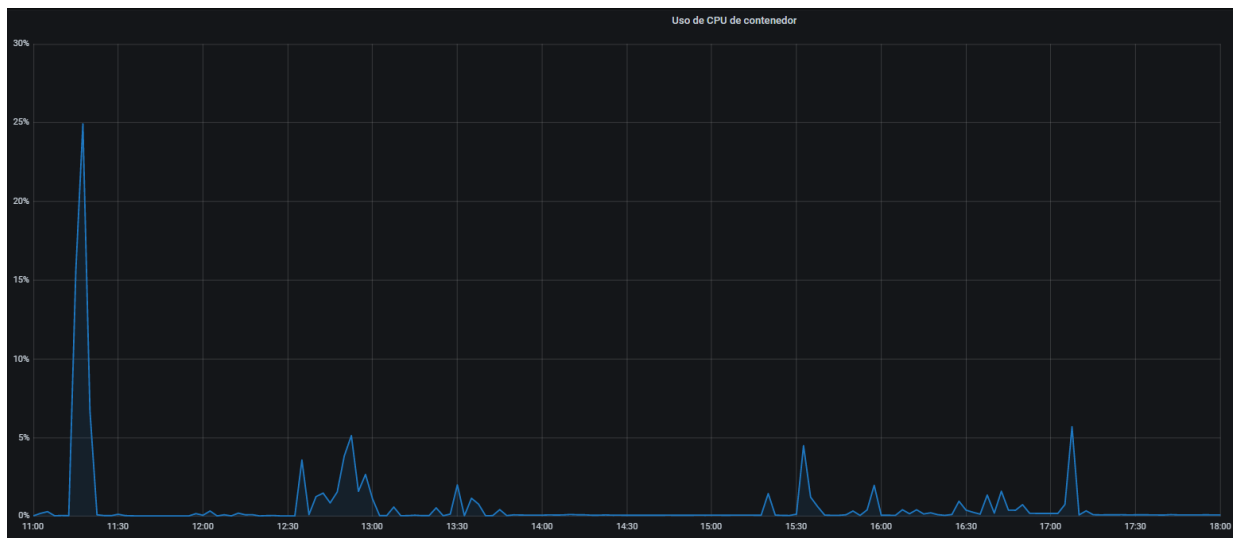
*Nota: En la parte superior, se muestra el consumo, mientras que en la inferior se aprecia la memoria libre en cada momento.*

*Fuente: Imagen propia.*

### Rendimiento de los laboratorios virtuales

También se ha realizado una monitorización de los laboratorios virtuales a lo largo de la prueba piloto, obteniéndose datos parejos para los mismos. Dentro de los consumos de CPU puede observarse en la Figura 37 como se experimenta un pico al inicio de la prueba piloto, donde se procede a la instalación el software necesario (XAMPP y WordPress), mientras que posteriormente el consumo es más moderado.

**Figura 37**



*Consumo de CPU en laboratorio virtual del entorno piloto*

*Fuente: Imagen propia.*

En cuanto al consumo de memoria, se han apreciado picos directamente relacionados al consumo de CPU, utilizando una media de aproximadamente 1GB por laboratorio virtual de media, esto puede apreciarse en la Figura 38. Estos picos debieran de ser controlados en caso de despliegue en un entorno de producción en un entorno real debido a que se corre el riesgo de superar la memoria disponible. Para ello se limitaría la RAM disponible por laboratorio, con el objetivo de impedir que un contenedor pueda reclamar más memoria de la disponible para sí mismo, dificultando el funcionamiento del resto de la infraestructura.

**Figura 38**



*Consumo de memoria RAM en laboratorio virtual del entorno piloto*

*Fuente: Imagen propia.*

## 5. CONCLUSIONES Y TRABAJO FUTURO

### 5.1. CONCLUSIONES

Tras las pruebas y comparativa de datos de ambos entornos se extraen las siguientes conclusiones del sistema:

De las conclusiones anteriores se desprende que una arquitectura basada en contenedores para un laboratorio virtual es una opción para tener en cuenta para entornos donde sea útil mantener una infraestructura limitada o bien sea necesario contener el uso de recursos. Esta optimización de recursos es importante ya que limita en gran medida la necesidad de disponer de un gran potencial hardware para montar un entorno formativo.

Se ha comprobado también que las características que ofrecen los contenedores en un entorno formativo son de gran interés, ya que se puede disponer de distintas estructuras o configuraciones sin necesidad de grandes esfuerzos. También puede procederse a una limpieza o actualización del entorno con gran facilidad.

En comparación con el uso de máquinas virtuales, ha quedado patente que, aunque en tiempos de ejecución las soluciones son similares en tiempos de respuesta, en sus requisitos y consumo los contenedores muestran un mejor comportamiento en cuanto al consumo de estos, esto es debido a que por sus características no usan más que los recursos que precisan en cada momento, pudiendo incluso limitar el uso de estos de forma fija en la definición del entorno.

Más concretamente se ha experimentado una mejora del 82% en cuanto al consumo de memoria RAM en tiempo de ejecución, esta mejora es debido a que no se requiere una asignación previa de memoria para la ejecución del sistema, sino que utilizará toda aquella que precise y tenga disponible.

Se experimenta también un consumo del 82% menor en lo que se refiera a exigencia de CPU durante la ejecución de los laboratorios en su conjunto.

En cuanto a la capacidad de disco se nota la eficiencia del sistema en cuanto a que no necesita una asignación fija del espacio, sino que utilizará aquel que esté disponible en el servidor, de



modo que no se dispondrá de espacio asignado no utilizado, al igual que ocurría con la asignación de memoria RAM.

Tras la implementación del laboratorio modelo se ha comprobado la viabilidad del sistema para su uso en entornos formativos, ya que sus bajos consumos de recursos y su capacidad de adaptación y escalabilidad le ofrecen una gran capacidad de adaptación.

El uso de HTML5 en el sistema permite su uso con multitud de dispositivos distintos siempre que estos sean compatibles con tal tecnología, por lo que otorga una gran variedad de posibilidades a la hora de conectarse al Sistema. Estas posibilidades son muy importantes hoy en día ya que permitiendo la conectividad desde cualquier dispositivo que cumpla con los requisitos el usuario podrá conectarse y trabajar desde distintos dispositivos.

Se ha llevado a cabo una prueba exitosa en un entorno real mediante el despliegue de la infraestructura en un servidor ubicado en la nube, en el que se han podido verificar los datos de consumo de recursos obtenidos en el entorno de pruebas local. Estos resultados han dispuesto que el uso de los contenedores para el despliegue de un entorno virtual es posible con resultados positivos para entornos formativos reales.

De estos resultados se desprende que, para el entorno utilizado para la prueba piloto, cada laboratorio ha consumido de media 1 GB de RAM, mientras que la CPU no ha superado picos de uso de un 33%, lo que da muestra de su eficiencia.

## 5.2. LÍNEAS DE TRABAJO FUTURO

**Migración del sistema a un servicio en la nube.**

Debido al avance de las posibilidades en la nube, se propone como futura línea de trabajo el albergar el entorno de trabajo en sistemas Online, como puede ser por ejemplo Amazon AWS. Estudiando la escalabilidad del sistema en dicho entorno.

**Uso de LDAP para autenticación de usuarios**

Parte de la carga de trabajo del Administrador es dar de alta todos los usuarios del Sistema. Se propone como mejora el uso de un sistema de gestión de usuarios centralizado de usuarios, como puede ser LDAP.

### Integración del sistema en una plataforma de orquestación

La tecnología basada en contenedores no para de avanzar y entre sus mejoras se encuentran las plataformas de orquestación. Se propone como mejora a esta propuesta la migración de esta a un entorno como Kubernetes.

### Comparativa del sistema con diferentes sistemas operativos

La tecnología basada en contenedores permite la implementación de diferentes sistemas operativos. Para el proyecto se ha trabajado con Ubuntu por ser una de las distribuciones más conocidas, pero sería posible elaborar diferentes arquitecturas con otros sistemas operativos.

### Implementación de un sistema piloto en un entorno educativo.

Para evaluar la funcionalidad práctica de la propuesta se propone la implantación de la arquitectura en un entorno educativo real, para ello se elaboraría un piloto de arquitectura y prácticas para un entorno de explotación real.

### Integración del sistema en una plataforma educativa online.

Las plataformas educativas online son una herramienta muy utilizada actualmente, por lo que se propone el estudio de una posible integración en alguna plataforma online, que pueda aprovechar las funcionalidades que ofrece esta infraestructura para la generación de laboratorios virtuales en los que realizar prácticas de laboratorio.

## Referencias bibliográficas

- Andújar Márquez, J. M., & Mateo Sanguino, T. J. (2010). Diseño de Laboratorios Virtuales y/o Remotos. Un Caso Práctico. *Revista Iberoamericana de Automática e Informática Industrial RIAI*, 7(1), 64-72. [https://doi.org/10.1016/S1697-7912\(10\)70009-1](https://doi.org/10.1016/S1697-7912(10)70009-1)
- Bernstein, D. (2014). Containers and Cloud: From LXC to Docker to Kubernetes. *IEEE Cloud Computing*, 1(3), 81-84. <https://doi.org/10.1109/MCC.2014.51>
- Bora, U. J., & Ahmed, M. (2013). E-learning using cloud computing. *International Journal of Science and Modern Engineering*, 1(2), 9-12.
- Bora, U. J., & Ahmed, M. (2013). *E-Learning using Cloud Computing*. 1(2), 5.
- Calderón, S. E., Núñez, P., Di Laccio, J. L., & Iannelli, L. M. (2015). Aulas-laboratorios de bajo costo, usando TIC. *Revista Eureka sobre Enseñanza y Divulgación de las Ciencias - 2015*, 12 (1) pp. 212-226. <https://rodin.uca.es/xmlui/handle/10498/16934>
- Caminero, A. C., Ros, S., Hernández, R., Robles-Gómez, A., Tobarra, L., & Granjo, P. J. T. (2016a). VirTUAL remoTe labORatories Management System (TUTORES): Using Cloud Computing to Acquire University Practical Skills. *IEEE Transactions on Learning Technologies*, 9(2), 133-145. <https://doi.org/10.1109/TLT.2015.2470683>
- Caminero, A. C., Ros, S., Hernández, R., Robles-Gómez, A., Tobarra, L., & Granjo, P. J. T. (2016b). VirTUAL remoTe labORatories Management System (TUTORES): Using Cloud Computing to Acquire University Practical Skills. *IEEE Transactions on Learning Technologies*, 9(2), 133-145. <https://doi.org/10.1109/TLT.2015.2470683>
- Casalicchio, E., & Iannucci, S. (2020). The state-of-the-art in container technologies: Application, orchestration and security. *Concurrency and Computation: Practice and Experience*, 32(17), e5668. <https://doi.org/10.1002/cpe.5668>
- Chae, M., Lee, H., & Lee, K. (2019). A performance comparison of linux containers and virtual machines using Docker and KVM. *Cluster Computing*, 22(1), 1765-1775. <https://doi.org/10.1007/s10586-017-1511-2>
- Coronel, G. A., & Vallejos, O. (2018). *Rendimiento de aplicaciones web en plataformas de contenedores de código abierto*. XX Workshop de Investigadores en Ciencias de la Evaluación del uso de la tecnología de contenedores para la creación de un aula virtual

- Computación (WICC 2018, Universidad Nacional del Nordeste).  
<http://sedici.unlp.edu.ar/handle/10915/67217>
- Domínguez, M., Reguera, P., & Fuertes, J. J. (2010). Laboratorio Remoto para la Enseñanza de la Automática en la Universidad de León (España). *Revista Iberoamericana de Automática e Informática industrial*, 2(2), 36-45. <https://riunet.upv.es/handle/10251/146530>
- Espinosa-Ríos, E. A., González-López, K. D., & Hernández-Ramírez, L. T. (2016). Las prácticas de laboratorio. *Entramado*, 12(1), 266-281.  
<https://doi.org/10.18041/entramado.2016v12n1.23125>
- Felter, W., Ferreira, A., Rajamony, R., & Rubio, J. (2015). An updated performance comparison of virtual machines and Linux containers. *2015 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, 171-172.  
<https://doi.org/10.1109/ISPASS.2015.7095802>
- Fox, G. C., Ishakian, V., Muthusamy, V., & Slominski, A. (2017a). Status of Serverless Computing and Function-as-a-Service (FaaS) in Industry and Research. *arXiv:1708.08028 [cs]*.  
<https://doi.org/10.13140/RG.2.2.15007.87206>
- Fox, G. C., Ishakian, V., Muthusamy, V., & Slominski, A. (2017b). Status of Serverless Computing and Function-as-a-Service(FaaS) in Industry and Research. *arXiv:1708.08028 [cs]*.  
<https://doi.org/10.13140/RG.2.2.15007.87206>
- Fundación Telefónica. (2020a). *El futuro digital en Europa*. Taurus.  
<https://www.fundaciontelefonica.com/cultura-digital/publicaciones/el-futuro-digital-de-europa/714/>
- Fundación Telefónica. (2020b). *Sociedad Digital en España 2019*. Taurus.  
<https://www.fundaciontelefonica.com/cultura-digital/publicaciones/sociedad-digital-en-espana-2019/699/>
- Gil, S., Calderón, S., Núñez, P., Di Laccio, J. L., & Iannelli, L. M. (2014). Aulas-laboratorios de bajo costo, usando TIC. *Revista Eureka sobre Enseñanza y Divulgación de las Ciencias*, 12(1), 212-226. <https://doi.org/10.2526/638>

- Großmann, M., & Klug, C. (2017). Monitoring Container Services at the Network Edge. *2017 29th International Teletraffic Congress (ITC 29)*, 1, 130-133. <https://doi.org/10.23919/ITC.2017.8064348>
- Hendrickson, S., Sturdevant, S., Harter, T., Venkataramani, V., Arpaci-Dusseau, A. C., & Arpaci-Dusseau, R. H. (2016). *Serverless Computation with OpenLambda*. 8th {USENIX} Workshop on Hot Topics in Cloud Computing (HotCloud 16). <https://www.usenix.org/conference/hotcloud16/workshop-program/presentation/hendrickson>
- INCIBE. (2020, enero 9). *La virtualización puede ser la solución a tus problemas*. INCIBE. <https://www.incibe.es/protege-tu-empresa/blog/virtualizacion-puede-ser-solucion-tus-problemas>
- Kaewkasi, C. (2018). *Docker for Serverless Applications: Containerize and orchestrate functions using OpenFaas, OpenWhisk, and Fn*. Packt Publishing Ltd.
- Kavitha, B., & Varalakshmi, P. (2018). Performance Analysis of Virtual Machines and Docker Containers. En S. R & M. Sharma (Eds.), *Data Science Analytics and Applications* (pp. 99-113). Springer. [https://doi.org/10.1007/978-981-10-8603-8\\_9](https://doi.org/10.1007/978-981-10-8603-8_9)
- Kozhირbayev, Z., & Sinnott, R. O. (2017). A performance comparison of container-based technologies for the Cloud. *Future Generation Computer Systems*, 68, 175-182. <https://doi.org/10.1016/j.future.2016.08.025>
- Lingayat, A., Badre, R. R., & Gupta, A. K. (2018). Performance Evaluation for Deploying Docker Containers On Baremetal and Virtual Machine. *2018 3rd International Conference on Communication and Electronics Systems (ICCES)*, 1019-1023. <https://doi.org/10.1109/CESYS.2018.8723998>
- Lloyd, W., Ramesh, S., Chinthalapati, S., Ly, L., & Pallickara, S. (2018). Serverless Computing: An Investigation of Factors Influencing Microservice Performance. *2018 IEEE International Conference on Cloud Engineering (IC2E)*, 159-169. <https://doi.org/10.1109/IC2E.2018.00039>
- Londoño Salazar, J. E., & Alvarez Córdoba, A. A. (2015). *Modelo para la implementación de laboratorios en programas bajo modalidad virtual – caso aplicado a la Ingeniería Informática*. <https://repositorial.cuaieed.unam.mx:8443/xmlui/handle/20.500.12579/3900>

- Luengas, L. A., Guevara, J. C., & Sánchez, G. (s. f.). *¿Cómo desarrollar un laboratorio virtual? Metodología de diseño*. 6.
- Morabito, R. (2017). Virtualization on Internet of Things Edge Devices With Container Technologies: A Performance Evaluation. *IEEE Access*, 5, 8835-8850. <https://doi.org/10.1109/ACCESS.2017.2704444>
- Moya López, M. (2013). De las TICs a las TACs: La importancia de crear contenidos educativos digitales | DIM: Didáctica, Innovación y Multimedia. *Revista DIM*, N° 27. <https://www.raco.cat/index.php/DIM/article/view/275963>
- Nazareno, G. (2018). *Virtualización de servidores. Conceptos básicos*. <http://148.202.167.116:8080/xmlui/handle/123456789/2283>
- Pérez, A., Moltó, G., Caballer, M., & Calatrava, A. (2018a). Serverless computing for container-based architectures. *Future Generation Computer Systems*, 83, 50-59. <https://doi.org/10.1016/j.future.2018.01.022>
- Pérez, A., Moltó, G., Caballer, M., & Calatrava, A. (2018b). Serverless computing for container-based architectures. *Future Generation Computer Systems*, 83, 50-59. <https://doi.org/10.1016/j.future.2018.01.022>
- Performance Evaluation of Docker Container and Virtual Machine. (2020). *Procedia Computer Science*, 171, 1419-1428. <https://doi.org/10.1016/j.procs.2020.04.152>
- Pessolani, P. A. (2018). *Un modelo de arquitectura para un sistema de virtualización distribuido* [Tesis, Universidad Nacional de La Plata]. <https://doi.org/10.35537/10915/71044>
- Piraghaj, S. F., Dastjerdi, A. V., Calheiros, R. N., & Buyya, R. (2015). Efficient Virtual Machine Sizing for Hosting Containers as a Service (SERVICES 2015). *2015 IEEE World Congress on Services*, 31-38. <https://doi.org/10.1109/SERVICES.2015.14>
- Ramalho, F., & Neto, A. (2016). Virtualization at the network edge: A performance comparison. *2016 IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 1-6. <https://doi.org/10.1109/WoWMoM.2016.7523584>
- Redhat. (2021). *Diferencias entre IaaS, PaaS y SaaS*. <https://www.redhat.com/es/topics/cloud-computing/iaas-vs-paas-vs-saas>

- Robles-Gómez, A., Tobarra, L., Pastor, R., Hernández, R., Duque, A., & Cano, J. (2019). Analyzing the Students' Learning within a Container-based Virtual Laboratory for Cybersecurity. *Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality*, 275-283. <https://doi.org/10.1145/3362789.3362840>
- Sianipar, J. H., Willems, C., & Meinel, C. (2016). A Container-Based Virtual Laboratory for Internet Security e-Learning. *International Journal of Learning and Teaching*, 2(2), 121-128.
- Soltesz, S., Pötzl, H., Fiuczynski, M. E., Bavier, A., & Peterson, L. (2007). Container-based operating system virtualization: A scalable, high-performance alternative to hypervisors. *Proceedings of the 2nd ACM SIGOPS/EuroSys European Conference on Computer Systems 2007*, 275-287. <https://doi.org/10.1145/1272996.1273025>
- Tobarra, L., Robles-Gómez, A., Pastor, R., Hernández, R., Duque, A., & Cano, J. (2020). Students' Acceptance and Tracking of a New Container-Based Virtual Laboratory. *Applied Sciences*, 10(3), 1091. <https://doi.org/10.3390/app10031091>
- Turnbull, J. (2014). *The Docker Book: Containerization Is the New Virtualization*. James Turnbull.
- Villamizar, M., Garcés, O., Ochoa, L., Castro, H., Salamanca, L., Verano, M., Casallas, R., Gil, S., Valencia, C., Zambrano, A., & Lang, M. (2017). Cost comparison of running web applications in the cloud using monolithic, microservice, and AWS Lambda architectures. *Service Oriented Computing and Applications*, 11(2), 233-247. <https://doi.org/10.1007/s11761-017-0208-y>
- Wang, C. (2016). *Containers 101: Linux containers and Docker explained - ProQuest*. InfoWorld.com. <http://www.espaciotv.es:2048/referer/secretcode/trade-journals/containers-101-linux-docker-explained/docview/1791594210/se-2?accountid=142712>
- Watts, T., Benton, R., Glisson, W., & Shropshire, J. (2019, enero 8). *Insight from a Docker Container Introspection*. <https://doi.org/10.24251/HICSS.2019.863>
- Xu, P., Shi, S., & Chu, X. (2017). Performance Evaluation of Deep Learning Tools in Docker Containers. *2017 3rd International Conference on Big Data Computing and Communications (BIGCOM)*, 395-403. <https://doi.org/10.1109/BIGCOM.2017.32>
- Zhang, J., Lu, X., & Panda, D. K. (2016). Performance Characterization of Hypervisor-and Container-Based Virtualization for HPC on SR-IOV Enabled InfiniBand Clusters. *2016 IEEE International*

*Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, 1777-1784.  
<https://doi.org/10.1109/IPDPSW.2016.178>

Docker Inc. (2021). *Documentación de Docker*. Docker docs. <https://docs.docker.com/>

Liang, N., Wang, C., Liu, P., & Zhang, Y. (2021). Research of College Computer Laboratory Based on Cloud Computing Technology. *2021 IEEE International Conference on Information Communication and Software Engineering (ICICSE)*, 1-4.  
<https://doi.org/10.1109/ICICSE52190.2021.9404093>

Muradova, F. R. (2020). Virtual Laboratories in Teaching and Education. *Theoretical & Applied Science*, 2 (82). <https://doi.org/10.15863/TAS.2020.02.82.18>

Slamnik-Krijes̃torac, N., Gil, M. C., & Marquez-Barja, J. M. (2021). Cloud-based virtual labs vs. Low-cost physical labs: What engineering students think. 8.

Smolyaninov, A., Pocebneva, I., Fateeva, I., & Singur, K. (2021). Software implementation of a virtual laboratory bench for distance learning. *E3S Web of Conferences*, 244, 11009.  
<https://doi.org/10.1051/e3sconf/202124411009>

Yusuf, I., & Widyaningsih, S. W. (2020). Implementing E-Learning-Based Virtual Laboratory Media to Students' Metacognitive Skills. *International Journal of Emerging Technologies in Learning (IJET)*, 15(05), 63. <https://doi.org/10.3991/ijet.v15i05.12029>



## Índice de acrónimos

<b>CPD</b>	Centro de Proceso de Datos
<b>IaaS</b>	Infrastructure as a Service, Infraestructura como servicio
<b>IoT</b>	Internet of Things, Internet de las Cosas
<b>MV</b>	Máquina Virtual
<b>PaaS</b>	Platform as a Service, Plataforma como servicio
<b>SaaS</b>	Software as a Service, Software como servicio
<b>SO</b>	Sistema Operativo
<b>TFG</b>	Trabajo Fin de Grado
<b>TIC</b>	Tecnologías de la Información
<b>UNIR</b>	Universidad Internacional de la Rioja
<b>URL</b>	Uniform Resource Locator, Localizador de Recursos Uniforme

## Anexo A. Tablas de Historias de Usuario

### Historias de Usuario Relacionadas con la Memoria del TFG

Tema	ID	Historia de Usuario	Criterio de aceptación	Sprint
MEMORIA	MEM1	Como Autor Quiero Elaborar una documentación bibliográfica Para poder Crear una bibliografía y disponer de datos suficientes para la elaboración del proyecto	a) Se dispone de una documentación extensa sobre contenedores, virtualización y entornos virtuales. b) Se dispone de una bibliografía suficiente c) Se conocen los conceptos de Virtualización d) Se profundiza en la tecnología de Contenedores e) Se dispone de comparativas entre virtualización y contenedores	Sprint 1
MEMORIA	MEM2	Como Autor Quiero Elaboración de un modelo de Memoria Para poder Iniciar la redacción de la memoria del TFG	a) Se dispone de un modelo .doc o .docx b) Se configuran los distintos tipos de letra, títulos y grafías c) Se conocen los criterios de citación APA 7ª edición	Sprint 1
MEMORIA	MEM3	Como Autor Quiero Haber redactado el apartado "Introducción" Para poder Completar dicho apartado	a) Se elabora la introducción. b) Queda suficientemente clara la motivación del TFG	Sprint 1
MEMORIA	MEM4	Como Autor Quiero Haber redactado el apartado "Estado del Arte" Para poder Completar dicho apartado	a) Se elabora el "Estado del Arte" b) Los conceptos quedan suficientemente claros c) Tiene una extensión acorde al tamaño de la memoria d) Se profundiza en los conceptos de Virtualización, contenedores y Cloud computing e) Se diferencian los distintos tipos de entorno de laboratorio informático f) Queda claro el valor añadido del TFG	Sprint 2
MEMORIA	MEM6	Como Autor Quiero Haber redactado el apartado "Objetivos" Para poder Completar dicho apartado	a) Se detalla el objetivo primario b) Se detalla el objetivo secundario	Sprint 2
MEMORIA	MEM7	Como Autor Quiero Haber redactado el apartado "Tecnologías empleadas" Para poder Completar dicho apartado	a) Se desglosan las tecnologías empleadas en el diseño b) Queda claro el objetivo de cada tecnología empleada	Sprint 2
MEMORIA	MEM5	Como Autor Quiero Haber redactado el apartado "Metodología" Para poder Completar dicho apartado	a) Se elabora el apartado "Metodología" b) Queda claro el concepto de metodología ágil c) Se desglosan las historias d) Se desglosan los Sprints	Sprint 2
MEMORIA	MEM8	Como Autor Quiero Haber redactado el apartado "Contenidos" Para poder Completar dicho apartado	a) Se detalla el contenido del Proyecto b) Queda definido el objetivo principal del proyecto c) Se detalla la arquitectura del laboratorio d) Se detallan las características del laboratorio	Sprint 3
MEMORIA	MEM9	Como Autor Quiero Haber redactado el apartado "Conclusiones y trabajo futuro" Para poder Completar dicho apartado	a) Se detallan los resultados b) Se definen una conclusiones c) Se indican las líneas de trabajo futuro	Sprint 4

### Historias de Usuario Relacionadas con el Laboratorio

Tema	ID	Historia de Usuario	Criterio de aceptación	Sprint	Puntos
LABORATORIO	LAB1	Como Autor Quiero Realizar labores de documentación Para poder Disponer de los conocimientos necesarios	a) Adquirir los conocimientos necesarios de Linux b) Adquirir los conocimientos necesarios de Docker c) Adquirir los conocimientos necesarios de Guacamole d) Adquirir los conocimientos necesarios de PostgreSQL	Sprint 2	21.0
LABORATORIO	LAB2	Como Autor Quiero Poner en marcha el entorno de trabajo Para poder Comenzar la elaboración del laboratorio virtual	a) Se dispone de un equipo con capacidad para ejecutar Docker b) Se instalan los componentes software necesarios	Sprint 2	8.0
LABORATORIO	LAB3	Como Autor Quiero Poner en marcha Docker Para poder Realizar pruebas y ejecutar contenedores	a) Se instala Docker en el equipo de pruebas b) Se comprueba que funciona correctamente	Sprint 2	8.0
LABORATORIO	LAB4	Como Autor Quiero Poner en marcha PostgreSQL Para poder Disponer de una base de datos para Guacamole	a) Se instala PostgreSQL en el equipo de pruebas b) Se comprueba que funciona correctamente	Sprint 2	5.0
LABORATORIO	LAB5	Como Autor Quiero Poner en marcha Guacamole Para poder Realizar pruebas con dicha tecnología	a) Se instala Guacamole b) Se configura Guacamole c) Se comprueba que es accesible y funcional	Sprint 3	8.0
LABORATORIO	LAB6	Como Autor Quiero Elaborar las imágenes de Docker a utilizar en el TFG Para poder Disponer de las imágenes necesarias para el proyecto	a) Se descargan las imágenes necesarias del repositorio de Docker b) Se elaboran las imágenes propias necesarias para el proyecto	Sprint 3	21.0
LABORATORIO	LAB7	Como Autor Quiero Poner en marcha el laboratorio de prueba Para poder Realizar pruebas de rendimiento	a) Se dispone de todos los contenedores corriendo correctamente b) La arquitectura funciona correctamente c) Se dispone de al menos 2 contenedores de prueba	Sprint 4	21.0

*Historias de Usuario Relacionadas con las Pruebas*

Tema	ID	Historia de Usuario	Criterio de aceptación	Sprint	Puntos
PRUEBAS	PRU1	Como Autor Quiero Realizar pruebas iniciales con Docker Para poder Comprobar que funciona correctamente	a) Docker se ejecuta b) Pueden descargarse imágenes c) Pueden crearse imágenes d) Los contenedores son accesibles	Sprint 2	8.0
PRUEBAS	PRU2	Como Autor Quiero Realizar pruebas iniciales con PostgreSQL Para poder Comprobar el funcionamiento	a) El contenedor con PostgreSQL se ejecuta b) Es posible acceder al contenedor c) Se pueden introducir Datos d) La BD de Guacamole se crea correctamente e) El contenedor es accesible desde el terminal f) El contenedor es accesible desde otros contenedores	Sprint 2	8.0
PRUEBAS	PRU3	Como Autor Quiero Realizar pruebas iniciales con Guacamole Para poder Comprobar el funcionamiento	a) El contenedor con Guacamole se ejecuta b) Es posible acceder al contenedor c) La interfaz de Guacamole es accesible con navegador d) Se pueden crear usuarios y conexiones e) Es posible conectarse a un contenedor a través de la interfaz	Sprint 3	13.0
PRUEBAS	PRU4	Como Autor Quiero Realizar pruebas con los contenedores laboratorio Para poder Comprobar el funcionamiento	a) La imagen del contenedor se genera sin problemas b) El contenedor es accesible c) Se levantan los servicios ssh y vnc d) Se crean los usuarios definidos	Sprint 4	13.0
PRUEBAS	PRU5	Como Autor Quiero Realizar pruebas de rendimiento Para poder Disponer de datos para evaluar el rendimiento	a) Se establece un método de recopilación de datos b) Se adquieren datos de rendimiento del entorno de laboratorio c) Se adquieren datos con distintas configuraciones de laboratorio	Sprint 4	13.0