

CONTINUOUS HOSTING

CFGS Administración de sistemas informáticos y redes

Alberto Gutierrez Jiménez & David Delgado Santamaria

M14 – Proyecto Final / Institut Puig Castellar





Esta obra está sujeta a una licencia de:
[Reconeixement-NoComercial-CompartirIgual 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-sa/3.0/es/)

Resumen del proyecto:

La idea del proyecto es la de realizar un servidor que sea capaz de contener contenedores en los que se instalen servidores de ciertos videojuegos, que a su vez estos se puedan gestionar y configurar mediante un entorno web desde el que también se podrán crear y destruir estos contenedores.

El objetivo es el de simplificar la acción de crear servidores multijugador de videojuegos, ya que esto muchas veces resulta una tarea difícil. Por ello se intenta facilitar la tarea para aquellos jugadores que quieran jugar online con sus amigos y no tengan los conocimientos previos para hacerlo.

Una vez los usuarios creen el servidor multijugador podrán realizar la acción de jugar online mediante una dirección IP y puerto que podrán ver en la web junto al servidor creado.

La organización se realiza mediante un diagrama de Gantt y el servicio Asana, la cual nos permite aplicar la metodología SCRUM.

El objetivo final sería llegar a conseguir un prototipo de servidor funcional el cual funcione de manera local para ver la eficiencia de este tipo de sistemas de hosting de servidores de videojuegos y plantearse si la idea se puede llegar a trasladar a un entorno online y público.

Palabras Clave:

Contenedores, Gestión Contenedores, Servidores, Desarrollo Web, PHP.

Abstract:

The idea of the project is to create a server that is capable of containing containers in which certain video game servers are installed, which in turn can be managed and configured through a web environment from which they can also be created and destroyed.

The objective is to simplify the action of creating multiplayer video game servers, since this is often a difficult task. Therefore, we try to facilitate the task for those players who want to play online with their friends and do not have the previous knowledge to do so.

Once the users create the multiplayer server they will be able to perform the action of playing online using an IP address and port that they will be able to see on the web together with the created server.

The organization is carried out using a Gantt diagram and the Asana service, which allows us to apply the SCRUM methodology.

The final objective would be to achieve a functional server prototype that works locally to see the efficiency of this type of video game server hosting systems and to consider whether the idea can be transferred to an online and public environment.

Keywords:

Containers, Container Management, Servers, Web Development, PHP

Índice:

<i>Resumen del proyecto:</i>	2
<i>Palabras Clave:</i>	2
<i>Abstract:</i>	2
<i>Keywords:</i>	3
INTRODUCCIÓN	6
CONTEXTO	6
JUSTIFICACIÓN	7
OBJETIVOS	7
<i>Objetivos Generales</i>	7
<i>Objetivos Específicos</i>	7
ESTRATEGIA Y PLANIFICACIÓN DEL PROYECTO	7
<i>Metodología de trabajo</i>	8
ESTUDIO ECONÓMICO Y DE PRESUPUESTO	9
DESCRIPCIÓN DEL PROYECTO	10
ANÁLISIS DE REQUISITOS	10
<i>Requisitos funcionales</i>	10
<i>Requisitos no funcionales</i>	10
TECNOLOGÍAS	11
<i>Diferentes tecnologías valoradas</i>	11
<i>Tecnologías elegidas</i>	12
<i>Estructura del proyecto</i>	12
DESCRIPCIÓN DE LOS COMPONENTES	13
<i>Página web</i>	13
<i>Servidor</i>	13
<i>Contenedores</i>	13
DEFINICIÓN DE FUNCIONALIDADES	13
<i>Página web</i>	13
<i>Servidor</i>	14
<i>Contenedores</i>	14
CONTENIDOS	15
ENTORNOS DE VIRTUALIZACIÓN	15
<i>Algunas de las ventajas de LXC</i>	17
AUTOMATIZACIÓN DE TAREAS	18
<i>Bash script</i>	18
<i>Ansible</i>	18

DESARROLLO	20
ANÁLISIS DE NECESIDADES.....	20
<i>Gestión de tareas</i>	<i>20</i>
<i>Servidor Multijugador.....</i>	<i>21</i>
PÁGINA WEB	25
<i>Página de Inicio</i>	<i>26</i>
<i>Configuración de los Servidores</i>	<i>27</i>
<i>Control de Servidores.....</i>	<i>29</i>
<i>Contacto</i>	<i>30</i>
SERVIDOR	31
<i>Contenedores</i>	<i>32</i>
<i>Modificación del archivo de configuración</i>	<i>32</i>
RED	33
TRABAJO FUTURO	35
MARKETING.....	36
CONCLUSIONES.....	37
PROBLEMAS Y ERRORES	38
PROBLEMAS	38
ERRORES	38
GLOSARIO	44
WEBGRAFÍA	45
ANEXOS	47
PRESUPUESTO	48
WIREFRAME	49
PRUEBA DE INSTALACIÓN Y UTILIZACIÓN DE DOCKER	50

Introducción

La idea que hay detrás de Continuos Hosting es la de facilitar una característica de los videojuegos más importantes en estos últimos tiempos que no es otra que la del juego multijugador, muchos usuarios se compran un videojuego para poder jugar con amigos o porque les gusta el hecho de jugar en un entorno online y compartir experiencias con desconocidos, el problema que hay en esto es que no siempre las empresas que han desarrollado estos videojuegos soportan un servidor público para que los jugadores puedan disfrutar de este modo multijugador. Por ello el objetivo es el de implementar de una manera sencilla que estos usuarios que quieren jugar en este modo puedan hacerlo ofreciéndoles facilidades.

Para ello lo que se plantea desarrollar es una herramienta web en la que se puedan crear, configurar, modificar y eliminar servidores, mediante una interfaz sencilla el objetivo es que los usuarios puedan disfrutar de un espacio en un servidor el cual hará de host de un servidor de videojuegos y ofrecerá una dirección IP y un puerto para poder conectarse a este y empezar a jugar.

Su desarrollo sería mediante lenguaje HTML 5, CSS y JavaScript creada desde cero, con diseño propio la cual esté en el mismo servidor que soportara contenedores en los que estarán estos servidores funcionando.

La comunicación entre la web y los contenedores se realizará mediante PHP el cual utilizaremos para crear los contenedores y una vez creados poder modificar el estado y la configuración de los servidores.

Los servidores serán LXC ya que son sencillos de utilizar y cumple la función que se desea, la idea es la de crear contenedores con el servidor configurado, entonces una vez que mediante la web se da la opción de que se quiere crear un servidor lo que se realizaría sería una clonación de ese servidor funcional. Se iniciará el contenedor y se pasaría a editar los archivos de configuración con los datos proporcionados desde la web, una vez hecho se iniciaría el servidor del videojuego, el servidor estará a la espera para mandar una señal a la web cuando esté listo para ser utilizado, mostrando la dirección IP y el puerto que tendrán que introducir para empezar a jugar.

Contexto

Actualmente existen algunos servicios gratuitos los cuales no ofrecen los suficientes recursos para obtener un rendimiento óptimo para hostear tus servidores, debido a la gran cantidad de personas que lo utilizan para hostear sus propios servidores. Un ejemplo claro sería la plataforma aternos.org, plataforma que nos permite crear / configurar / personalizar servidores de Minecraft de una forma totalmente gratuita.

Tras ver la gran cuota de mercado que recoge dicho servicio de hosting, recreamos la misma idea pero aplicando las tecnologías que se han considerado más óptimas.

Justificación

El motivo principal por el que se llevó a cabo el desarrollo de este proyecto es porque ambos integrantes son jugadores de videojuegos y se han encontrado con problemas los cuales un proyecto como este puede solucionar, por ello crear una herramienta que realmente se pueda aprovechar y solucionar todos estos problemas que durante sesiones de juego se hayan podido encontrar es una idea atrayente.

También comprobar cuán difícil es el realizar una infraestructura como esta y poder tenerla sin necesidad de desembolsar una cantidad de dinero mensual o utilizar servicios gratuitos que no ofrecen un servicio del todo óptimo.

Objetivos

Para hacer posible este proyecto se plantearon una serie de objetivos para hacer posible su realización.

Esta serie de objetivo se puede dividir en dos categorías.

Objetivos Generales

- Utilizar metodologías ágiles durante el proyecto (Usar diagramas de Gantt, Utilizar Ansana, etc.)
- Montar un sitio Web
- Aprender a usar contenedores.
- Ejecutarlos y configurarlos desde el sitio web.

Extra:

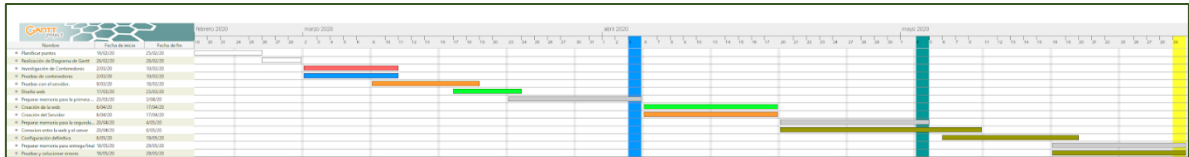
- Montar otros servidores de diferentes juegos en contenedores.

Objetivos Específicos

- Añadir y configurar Minecraft server en un contenedor.
- Hacer que funcione el servidor desde el contenedor.
- Ofrecer soporte al cliente.

Estrategia y planificación del proyecto

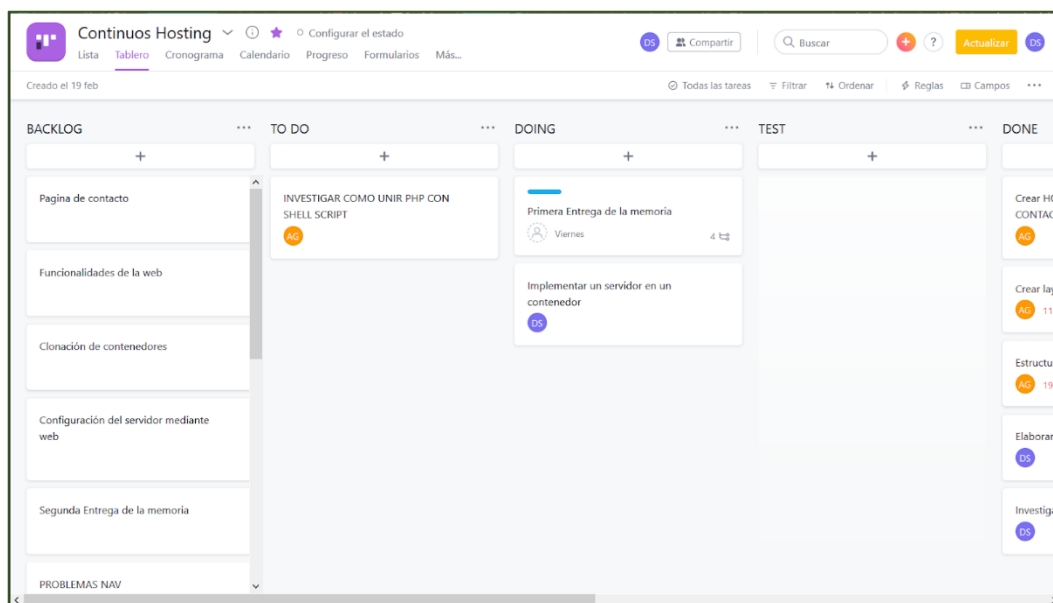
La planificación será mediante un diagrama de Gantt el cual hará de guía para el desarrollo del proyecto, intentando no sobrepasar los plazos estipulados y poder llevar a cabo de una forma ordenada y correcta la organización del proyecto.



En cuanto a la estrategia se seguirá una estrategia de desenvolvimiento desde cero pero investigando tecnologías ya utilizadas para poder desarrollar de una forma más eficiente el objetivo deseado.

Metodología de trabajo

La metodología utilizada será Scrum, mediante un servicio llamado Asana el cual ofrece entre muchas otras funciones la de tener una pizarra virtual y poder organizarnos de una forma más eficiente.



Esto unido a la planificación con diagrama de Gantt, ayuda a poder mantener un nivel de trabajo adecuado con fechas y sabiendo que puntos nos ha costado más, menos y qué problemas se han encontrado de cara introducirlo en futuras memorias.

Extra: Metodología Cuarentena

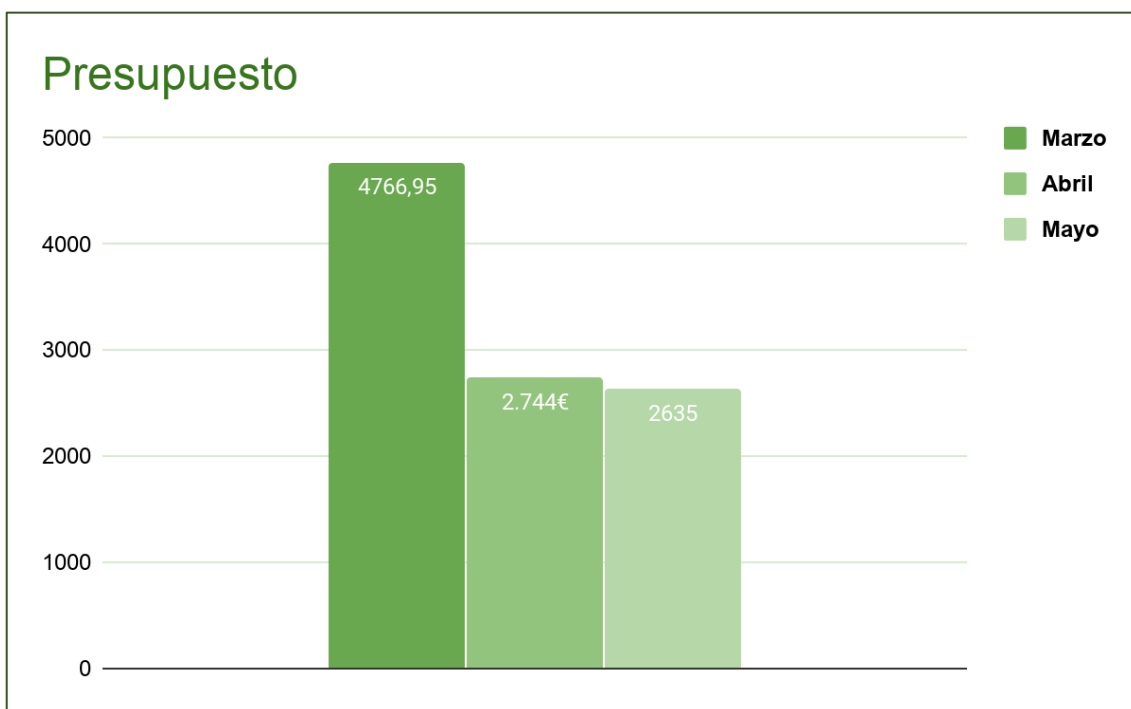
Tras el periodo confinamiento decretado por el gobierno durante el desarrollo de este proyecto se optó por realizar reuniones mediante la plataforma de videoconferencia Google Meet.



Estudio económico y de presupuesto

Para tener una mejor visión del proyecto, se ha realizado un presupuesto orientativo para hacerse una idea de los costes que tendrá la elaboración del proyecto.

	2020	Marzo	Abril	Mayo
GASTOS	10145,89	4766,95	2.744€	2635
Departamento RHH				
Nominas personal empresa		2.600€	2.600€	2.600€
Departamento Informatica				
Adquisicion Bienes		1.991,95€	108,94	0
Adquisicion Servicios		175€	35€	35€



Descripción del proyecto

Análisis de requisitos

- Tener más conocimiento sobre la funcionalidad de los contenedores.
- Adquirir conocimientos de conexión entre una web y su servidor para modificar archivos de configuración.
- Tener una web con una estructura y diseño correcto.
- Lograr el funcionamiento de juego multijugador.
- Conseguir un prototipo de una herramienta funcional de hosting.

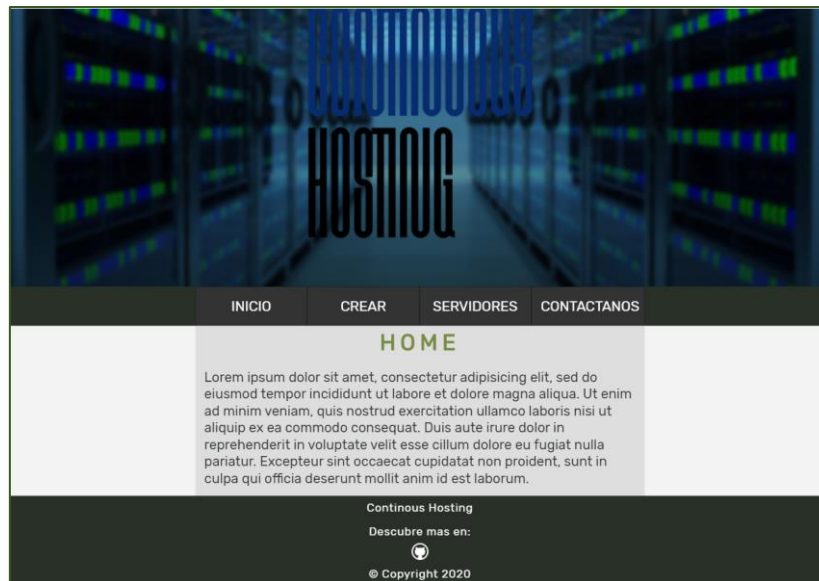
Requisitos funcionales

- Hosting de página web.
- Servidor optimizado para servidores de videojuegos.
- Conectar la web con el servidor de manera que pueda editar archivos de configuración.

Requisitos no funcionales

- La web sea completamente funcional.
- La web se intuitiva para el usuario.
- Web Responsive.





Tecnologías

Diferentes tecnologías valoradas

Contenedores:

- DOCKER
- SYSTEMD-NSPAWN
- Kubernetes
- LXC

En cuestión de contenedores, se han estado comparando estas herramientas y aunque todas y cada una de ellas tiene sus puntos fuertes y débiles al final se eligió LXC (Linux Containers), esta elección se llevó a cabo debido a que LXC es una herramienta diseñada específicamente para Linux por lo que funciona y se adapta muy bien al sistema operativo.

En comparación a Docker por ejemplo cambiar la configuración de archivos para poder modificarlos es más sencilla, ya que Docker se utiliza para ejecutar procesos, haciendo que los contenedores gasten lo menor de recursos posibles. En el caso de Continuous Hosting, los usuarios podrán modificar la configuración de los servidores de juegos por ello se consideró que Docker no podría ser del todo adecuado.

Systemd-nspawn utiliza un sistema muy similar al de LXC, ya que está inspirada en esta herramienta, incorpora nuevas cosas útiles pero para este proyecto se ha preferido utilizar algo más sencillo para desarrollar un hosting a nivel local, por ello LXC fue el contenedor elegido.

Otra herramienta investigada fue Kubernetes, con esta herramienta se pueden controlar muchos contenedores simultáneamente lo que puede ser muy útil para grandes sistemas.

Tecnologías elegidas

Xampp - Se ha seleccionado XAMPP para hostear la página web durante el desarrollo, es una herramienta importante ya que va a ser la interfaz para poder gestionar los servidores que se gestionan en el servidor.

Sublime Text - Es un software que ofrece una gran ayuda a la hora de realizar páginas web entre otras cosas, es la herramienta que se ha decidido utilizar para desarrollar la página web.

Virtual Box - Gracias a este software podemos almacenar más de un sistema operativo en la misma máquina, es la herramienta perfecta para realizar este proyecto pues ofrece el entorno de pruebas perfecto.

LXC - Es la herramienta que permite gestionar contenedores, parte esencial para la creación de los servidores multijugador en este proyecto.

Ansible – Herramienta que permite la gestión de sistemas remotos mediante ssh, incluyendo muchos módulos.

Apache – Servidor web de código abierto para todas las plataformas.

MySQL – Servicio que ofrece la creación sencilla base de datos.

Estructura del proyecto

La estructura que se ha utilizado para este proyecto ha consistido en separar en dos grandes bloques la carga de trabajo, por un lado estaría la página web que es la cara bonita del proyecto, la interfaz que los usuarios utilizaran para poder interactuar con el servidor, además de proporcionar ayuda y mostrar el estado de los servidores.

La segunda parte es el propio servidor que va a almacenar los contenedores y la propia página web.

El motivo por el que hemos seguido esta estructura es porque son los dos grandes núcleos del proyecto, por eso se ha decidido avanzar lo máximo posible hasta que se pudieran unir y comunicarse entre ellos.

Descripción de los componentes

Para la creación de todo el entorno de trabajo se ha utilizado la herramienta Virtual Box, en ella se disponía de un servidor Ubuntu Server 19.10, dicho server sería el encargado de gestionar tanto la página web como los contenedores mediante los servicios y programas utilizados.

Página web

Este componente es el que hará de intermediario entre el usuario y el servidor, realizando consultas según cada caso, para ello utilizara la función de PHP “shell_exec” para ejecutar ciertos archivos almacenados en el propio servidor que lo está hosteando, una vez que ejecuta estos comando esperara una respuesta del propio servidor, para realizar una consulta a la base de datos, esto también lo hace al acceder a ciertas páginas en las que se muestra un listado de todos los servidores.

Servidor

El servidor así como la página web se podría tratar como el intermediario entre la página web y los contenedores, utiliza los servicios de MySql para tener disponer de una base de datos donde guardar información y apache2 con PHP para poder mostrar la página que, el servidor internamente tendrá scripts que efectuaran las diferentes opciones.

El servidor es el que se encarga con todo esto de gestionar los diferentes contenedores.

Contenedores

Este componente es el que se encarga de iniciar los servidores de videojuegos, como se ha comentados antes están gestionados por el servidor, los clientes se conectaran a él mediante la dirección IP y el puerto indicado en cada caso.

Definición de funcionalidades

Las funcionalidades que ofrecen están ligadas a los componentes pues estos son los que activan las diferentes funcionalidad cuando son necesarias.

Página web

La página web tiene la funcionalidad de indicarle al servidor que acción tiene que realizar, pero incluye una funcionalidad que es la de ponerse en contacto con el servicio técnico.

Contacto

Mediante una librería de PHP permite a los usuarios que mediante un formulario puedan ponerse en contacto vía mail con el servicio técnico.

Estados de los contenedores

La web que realiza una consulta a la base de datos es capaz de mostrar una tabla con el estado de las diferentes maquinas, así como saber la dirección ip de cada una para poder conectarse al servidor desde el cliente.

Servidor

El servidor dispone de varios scripts para realizar diferentes funcionalidades con los contenedores.

Creación

Para esto el script que realiza se trata de una clonación de un contenedor prediseñado, ejecutando el comando preciso y posteriormente devolviendo la IP que se le asignara a ese servidor, se procederá al inicio del mismo para que el servidor se ejecute y proceda a la creación de todos los archivos necesarios para su funcionamiento, lo último que realizara será una llamada con ansible para modificar el archivo configuración con su configuración inicial.

Modificación

Cuando la llamada sea de configuración se realizara una llamada ansible que procederá al cambio del archivo de configuración mediante un módulo, cambiando los parámetros por los deseados.

Eliminación

Para la eliminación se realizara un simple script que incluirá el comando preciso para la eliminación del Contenedor con el nombre indicado.

Pausar o iniciar

A la hora de parar o iniciar un servidor se utilizaran dos scripts muy similares que realizaran esta función.

Contenedores

Los contenedores tendrán la funcionalidad de almacenar un servidor de Minecraft operativo.

Contenidos

Tras un análisis del objetivo del proyecto se decidieron varios puntos a tener en cuenta, uno de ellos era el público objetivo para el que iba a estar orientada esta herramienta, que era para los usuarios jugadores de Minecraft y que no tenían la capacidad de realizar un servidor online multijugador.

Este punto es el porqué de la realización del mismo, pues los desarrolladores han tenido que pasar por ello, por eso se orientó a una implantación sencilla en la que no hubiera que gastar mucho y poder realizar un prototipo con el material que se disponía.

Para esto la solución que se llevó a cabo fue la virtualización de servidores virtuales mediante la tecnología Linux containers, utilizando en todo momento software libre para en una futura implementación obtener un ahorro consistente, además de por el alto grado de seguridad que presenta.

Entre varias cosas las que más podríamos destacar serían, el ahorro, como se acaba de comentar el hecho de utilizar software libre baja los costes de licencias y para este proyecto que es una cosa humilde para que los usuarios puedan jugar de una forma cómoda, que se reduzcan los costes podría asegurar el mantenimiento del mismo, asimismo utilizando anuncios para el propio sustento y poco más, otro punto sería el de la seguridad, en el caso de que decidieran atacar mediante un servidor a la máquina en cuestión la virtualización mediante contenedores lograría no perjudicar directamente al propio servidor que va a estar hosteando todo esto, no solo eso, la clonación de un contenedor es muy sencilla por lo que se pueden realizar backups de una forma muy cómoda y rápida (de hecho la creación de los diferentes contenedores se realiza con un contenido con un servidor previamente configurado), otro punto sería la flexibilidad de este sistema, pues si en un futuro se implementaran más funcionalidades este sistema permite crear máquinas virtuales con las características necesarias para cada caso, así como ampliar recursos de un contenedor según las necesidades.

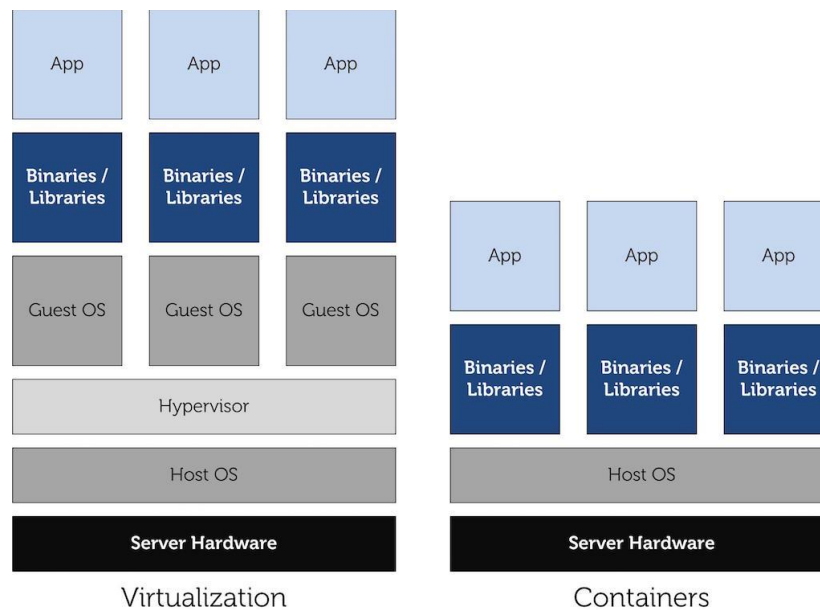
Sumando a todo esto la virtualización permite realizar una portabilidad o migración si se quisiera implementar el sistema en un servidor con más capacidad.

Entornos de virtualización

Para este proyecto lo que se ha utilizado para su implementación ha sido el software Virtual Box que propiamente es un software que permite la virtualización de un sistema operativo, el cual se tendrá que instalar mediante una imagen ISO, la utilización de esta herramienta ha sido por la facilidad de tener un sistema operativo Ubuntu Server en la misma máquina en la que se tenga un entorno gráfico donde se puedan realizar las diferentes pruebas pertinentes, pues la instalación de un entorno gráfico requeriría la utilización de recursos innecesarios y que se pueden aprovechar para el buen funcionamiento del mismo.

Por este motivo la utilización de esta herramienta que permite desarrollar el proyecto de una forma óptima para posteriormente poder lanzarlo ha sido perfecto.

El otro entorno de virtualización que se ha utilizado es como anteriormente se ha comentado LXC, este método de virtualización permite ejecutar diferentes sistemas Linux o aplicaciones dentro de diferentes entornos virtuales denominados “contenedores”, aislados entre ellos en una sola maquina host que los controla, el funcionamiento de mismo es una modificación del kernel de Linux, lo que proporciona a grandes rasgos un entorno lo más parecido posible a una distribución estándar de Linux.



*Imagen donde se comparan las dos virtualizaciones de sistemas

La clave de LXC es que al crear un nuevo contenedor no crea otras instancias del kernel del sistema operativo si no que utiliza el del propio sistema que lo está hosteando, ya que esta lo comparte con cada instancia nueva.

Esto provoca que el sistema sea capaz de virtualizar un volumen mucho mayor de contenedores que al volumen que sería capaz de alcanzar mediante máquinas virtuales por los recursos que necesitan, pero este no es el único punto importante para la elección de contenedores para la realización del proyecto pues al ejecutar se sobre el mismo kernel del sistema que lo hostea, iniciar, reiniciar, parar y apagar un contenedor es un proceso muy veloz y sencillo, este punto es uno de los más explotados para este proyecto, pues la gestión de los contenedores LXC se realiza mediante comandos, esto permite la utilización de scripts para hacer que la gestión de los mismos pueda ser de una forma más automática.

Algunas de las ventajas de LXC

- Uno de los puntos más importantes es que LXC se considera estable desde la versión 1.0 que fue publicada el 20 de febrero del 2014, esto hace que sea una herramienta muy fiable y por lo tanto se pueda confiar en que va a tener muy pocos errores que a día de hoy se puedan haber solucionado.
- La gestión de los contenedores mediante comandos facilita mucho su uso.
- Es posible realizar movimientos en las cargas de trabajo dado que no se está atado a ningún componente hardware.
- Se pueden realizar pruebas de los contenedores mediante la clonación o restablecimiento de los mismos.
- El funcionamiento de la propia herramienta es mediante el sistema de archivos de Linux, por lo que es muy sencillo acceder a los archivos.
- Es sencillo realizar copias de seguridad.
- Da la posibilidad de ejecutar diferentes versiones de aplicaciones, experimentar o probar nuevo software antes de actualizar de manera definitiva un servidor.
- LXC funciona aprovechando dos características del núcleo de Linux que son cgroups y los namespaces.

Poniendo hincapié en esta última ventaja ¿cómo ayuda esto al buen rendimiento de los contenedores?

- Cgroups es una característica del kernel de Linux que mide y es capaz de limitar el uso de recursos para un grupo de procesos, es decir que permite controlar los distintos recursos disponibles de los contenedores.
- Namespaces es una característica del kernel que proporciona y garantiza el aislamiento de procesos, con esto se puede aislar el identificador asignado a una serie de recursos del resto del sistema.

El otro sistema de virtualización utilizado es Virtual Box, esta herramienta es un software para arquitecturas x86/amd64, desarrollado actualmente por Oracle. Este software es el que ha permitido poder tener un servidor de Ubuntu disponible para realizar el proyecto de una forma cómoda y fácil, al disponer de una interfaz gráfica que permita editar la configuración de los diferentes sistemas virtualizados.

Automatización de tareas

Uno de los puntos más importantes de un proyecto así y en general de un buen administrador de sistemas es el de la automatización de tareas y es que no tendría sentido que cuando un usuario necesitara crear, iniciar o apagar un servidor se tenga que realizar manualmente con una persona pendiente las 24 horas para la realización de las diferentes tareas según el caso.

Es por esto que se ha realizado una automatización de las diferentes tareas mediante archivos Bash script y la herramienta ansible se ha logrado automatizar los diferentes procesos para no necesitar la mano humana a la hora de realizar una de las posibles tareas que se le mande mediante la página web.

Bash script

Bash un lenguaje de comandos y Shell de Unix lanzado por primera vez en 1989, que se ha utilizado ampliamente como el Shell de inicio de sesión predeterminada para la mayoría de distribuciones de Linux, es decir que Ubuntu server utiliza Bash para la interacción del administrador con el sistema, una de las características de Bash es la opción de realizar archivos denominados scripts, estos archivos mediante el texto que se introduzca permite la programación de tareas, como ejecutar comandos realizar alguna tarea en concreto...

Dado que es el lenguaje por defecto que utiliza Linux es una tecnología muy potente a la hora de realizar ciertas tareas, por ello (aunque su sintaxis puede causar malas pasadas a la hora de programarlo), se procedió a la utilización de este lenguaje que aun y no siendo tan potente como podrían haber sido otros lenguajes es el nativo y era lo suficiente mente útil para cubrir las necesidades que se deseaba.

Ansible

Ansible es una plataforma de software libre para configurar y administrar ordenadores, combina instalación multi-nodo, ejecuciones de tareas ad hoc y administración de configuraciones, su funcionamiento es mediante Python.

Su funcionamiento es el de gestionar nodos a través de SSH, con la utilización de diferente módulos que se pueden utilizar para realizar múltiples tareas es la herramienta perfecta para la automatización del sistema, pues este se puede conectar con los contenedores mediante la IP y de esa forma configurarlos .

Principales características de Ansible

A Ansible se le agrupa como una herramienta de gestión de configuración, pero lo cierto es que no se le puede encasillar solamente en eso, pues con todos los módulos de los que dispone puede ser utilizado en otro tipo de escenarios.

Aprovisionamiento

Con Ansible se pueden aprovisionar hosts virtualizado por lo que se pueden realizar descargas de nuevo contenido.

Gestión de la configuración

Sus distintos módulos permiten modificar archivos de configuración de todo tipo, permitiendo una gestión automática de la mayoría de aplicaciones o servicios de un sistema (esto viene bien para la configuración del servidor multijugador).

Despliegue de aplicaciones

Se puede definir una aplicación con Ansible que permite llevar un control de todo el ciclo de vida de una aplicación.

Seguridad y Cumplimiento

Ansible permite definir la seguridad de las sistemas de forma sencilla. Utilizando los diferentes Playbook se pueden definir reglas de firewall, gestión de usuarios y grupos y políticas de seguridad personalizada en los sistemas que se estén gestionando poseyendo un gran número de módulos que ayudan en su labor.

Orquestación

Ansible es capaz de orquestar despliegues de OpenStack por ejemplo.

Playbook de ansible

Ansible se puede utilizar de varias formas, ejecutando directamente el comando con las cosas que quieres realizar o mediante playbooks, esta segunda opción beneficia mucho la automatización de ya que la función de los playbooks es la de un archivo de extensión .yml en el que se podrá introducir de una forma muy cómoda las diferentes cosas que queremos que se realicen, es decir que se pueden realizar varias tareas una detrás de otra.

Desarrollo

El objetivo principal de este proyecto es el de realizar una herramienta a nivel local que permita crear servidores personalizados de videojuegos, todo esto gestionado por una página web diseñada también por el grupo desde cero programando con PHP aprovechando la experiencia adquirida en antiguos estudios por los dos participantes.

Análisis de necesidades

El primer paso antes de empezar a desarrollar el proyecto ha sido un análisis de las necesidades básicas del proyecto, se ha buscado información de las diferentes tecnologías que se podrían utilizar (priorizando en todo momento tecnologías/herramientas con conocimiento ya adquirido, para de esa forma ahorrar tiempo de aprendizaje), fue entonces cuando acabamos seleccionando las diferentes herramientas, Virtual Box para la creación el entorno servidor, LXC para la creación de contenedores, scripts de Bash y asible para la gestión de los contenedores y por otra parte, Sublime text y Visual Studio para la creación y edición de la página web tanto los HTML, CSS y PHP, el servicio apache2 con PHP para la visualización de la página y MySql para la creación de una base de datos.

Gestión de tareas

Desde un primer momento se ha dividido el proyecto en dos partes, por una parte el desarrollo de la página web, la creación de la misma, la obtención de recursos gráficos para la misma y la estructuración de formularios para poder configurar los archivos de configuración de los servidores. También se ha investigado formas de implementar algunas funciones gráficas para que la experiencia de los usuarios sea más agradable e intuitiva.

Y por otro lado la configuración del servidor creando contenedores y comprobando el comportamiento del mismo ante los servidores que consumen sus recursos para funcionar. Para ello se han realizado pruebas con varios contenedores para comprobar su funcionamiento.

Para la comunicación entre los desarrolladores se han utilizado dos programas fundamentales uno de ellos es Discord, para dudas o consultas rápidas, se ha utilizado este programa que incorpora un chat en vivo así como la posibilidad de realizar llamadas. Por otro lado cuando se trata de una reunión más extensa se ha optado por utilizar Google meet.

Servidor Multijugador

En la actualidad uno de los mayores medios de entretenimiento son los videojuegos, este mundo ya lleva tiempo desarrollándose, y uno de sus mayores atractivos actualmente es el modo multijugador de ciertos videojuegos, esto permite poder conectarte con gente de todo el mundo y jugar con ellos, pero la cosa va más allá de jugar con gente de todo el mundo pues lo más atractivo de los modos multijugador en línea es jugar con amigos y conocidos, esto a veces puede ser complicado ya que no todos los juegos tienen la opción de emparejarte con todos tus amigos e incluso algunos necesitas pagar mensualmente para poder tener un servidor al que conectarse y poder jugar ya que por ejemplo con los últimos sucesos de confinamiento un método para poder divertirte con ellos es jugar a algún videojuego con tus amigos para pasar un buen rato junto a ellos. Para establecer esta conexión es necesaria a parte de la conexión a internet, un servidor al que los jugadores se conectan y que gestione los paquetes de los mismos para que se pueda establecer una conexión entre ellos, de la misma forma hay servidores que no solo establecen la conexión entre los usuarios si no que todo el mundo al que los jugadores van a acceder va a estar soportado por el servidor (por ejemplo en el caso de Minecraft).

En un inicio se planteó este proyecto como un servidor que fuera capaz de hostear diversos servidores multijugador, pero con los últimos acontecimientos se decidió pasar esa opción a segundo plano y centrarnos en realizar el proyecto con solo un servidor, el servidor elegido ha sido del videojuego Minecraft, esto se debe a que justamente para este juego es necesario un pago mensual para poder tener un servidor privado con el que poder jugar con tus amigos, por lo que se ha querido dar apoyo a las personas que no pueden permitirse esa suscripción, que tengan la opción de jugar.

Para que un servidor de Minecraft funcione lo primero que necesitamos tener son los paquetes jdk de java pues Minecraft así como sus servidores en un inicio se desarrollaron con este lenguaje de programación. Para ello en el sistema operativo Ubuntu 19 qué es el que estamos utilizando tendremos que introducir el comando:

```
apt-get install default-jdk
```

De esta forma se instalara el jdk más de la versión más reciente.

Lo siguiente que tendremos que hacer es descargar el propio servidor de Minecraft, para esto tendremos que meternos en su página web <https://www.minecraft.net/es-es/download/server/> en esta obtendremos un link con el cual nos descargamos el archivo del servidor, para ello tendremos que descargarlo mediante el comando wget.

```
wget -O server.jar  
https://launcher.mojang.com/v1/objects/bb2b6b1aefcd70dfd1892149ac3a215f6c636b07/server.jar
```

Una vez realizada la descarga podremos ver que se ha descargado un archivo con el nombre que le hayamos llamado, en este caso server.jar

```
usuario@continuous hosting:~/ServerMinecraft$ ls  
server.jar
```

El siguiente paso es asegurarse que tiene permisos de ejecución, para ello se ejecutara el comando:

```
chmod +x server.jar
```

Una vez que se le den estos permisos se tendrá que crear un archivo llamado eula.txt para aceptar los términos y condición del programa, en este archivo se introducirá el texto "eula=true".

```
usuario@continuous hosting:~/ServerMinecraft$ cat eula.txt  
eula=true
```

El ultimo paso es el de iniciar el servidor, para ello ejecutaremos el comando:

```
Java -Xmx1024M -Xms1024M -jar server.jar nogui
```

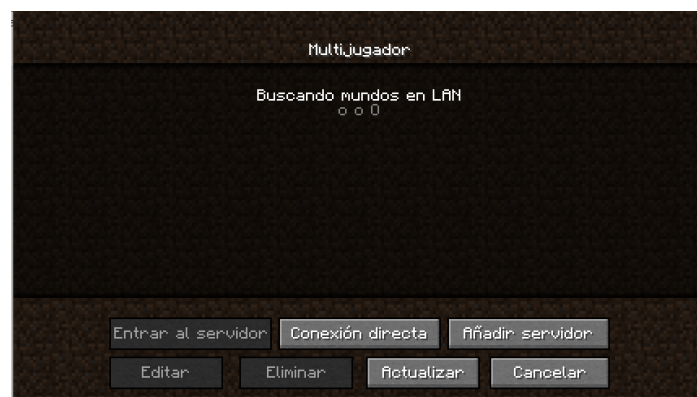
Ejecutando este comando lo que se hace es llamar a java ya que el servidor y el videojuego esta hecho con este lenguaje de programación, lo siguiente que se le indica es la memoria que queremos que recoja, en este caso 1GB, indicamos el archivo y para acabar le decimos que no se ejecute con entorno grafico con el parámetro nogui, se vera que el servidor se empieza a iniciar, si lo paramos en la carpeta donde se encuentra habran aparecido varios archivos, algunos para que el servidor función que no se deberán tocar en un principio y un archivo de configuración llamado server.properties.

```
GNU nano 3.2 server.properties  
#Minecraft server properties  
#Wed Mar 04 15:38:47 UTC 2020  
broadcast-rcon-to-ops=true  
view-distance=10  
max-build-height=256  
server-ip=192.168.0.47  
level-seed=  
rcon.port=25575  
gamemode=survival  
server-port=25565  
allow-nether=true
```

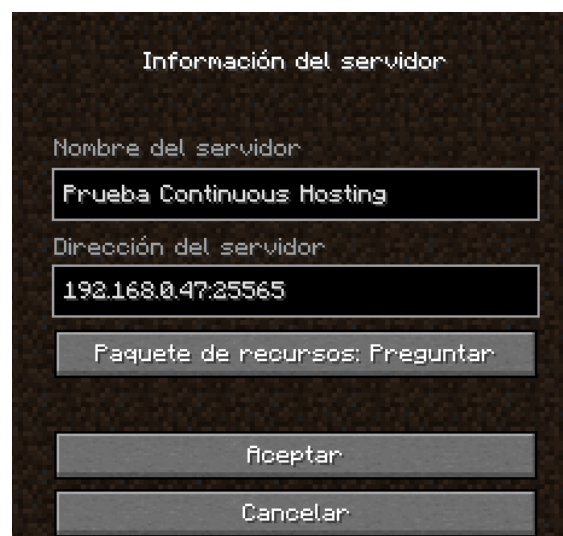
En este archivo se tendrá que modificar la línea server-ip en la que se tendrá que introducir la IP de la máquina que este hosteando el servidor, y el puerto que podremos elegir el que se desee mientras no sea uno de los utilizados.

Ahora que ya está el archivo de configuración bien configurado se tendrá que guardar y volver a ejecutar el servidor con el comando escrito anteriormente.

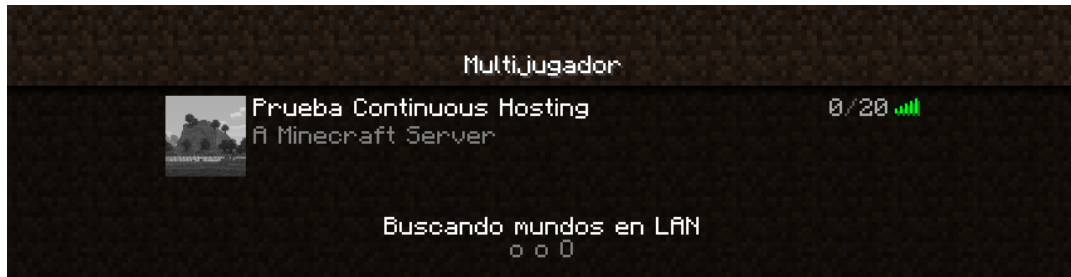
Para acabar podremos acceder al servidor desde el videojuego, para ello en el menú de inicio se pulsara el botón “Multijugador” y una vez dentro se pulsara el botón “Añadir Servidor”



Al pulsar el botón se tendrá que introducir un nombre para identificar el servidor y la dirección IP más el puerto.



Una vez que se hayan introducido los parámetros se tendrá que pulsar el botón “Aceptar”, el juego volverá a la pantalla anterior en la que ahora se podrá ver que hay un servidor con el nombre que se ha indicado.



Tras una breve carga se podrá ver que el servidor obtiene cobertura y que hay 0 personas de 20 posibles, para acceder se puede pulsar doble clic o pulsar el botón “Entrar al Servidor”.

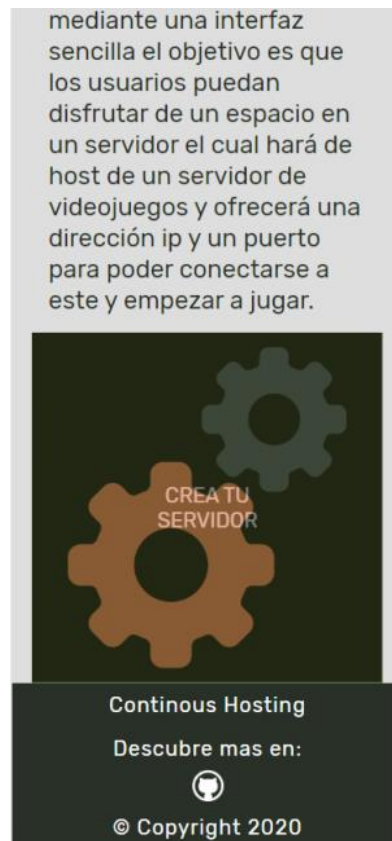


Cuando accede una persona al servidor se puede ver en el servidor un mensaje como que ha accedido un usuario.

```
[19:36:09] [Server thread/INFO]: DobleDD[/192.168.0.13:63472] logged in with entity id 164 at (297.85594652437715, 64.0, 372.2198497293105)
[19:36:09] [Server thread/INFO]: DobleDD joined the game
```

Página Web

La página web es uno de los puntos fuertes del proyecto ya que es la parte visual del mismo, no solo eso sino que va a ser el intermediario entre el usuario y el servidor, pues en esta se creará, modificará, eliminará y visualizará el estado de los contenedores que hostean estos servidores de videojuegos. Para hacer que la web sea más accesible para todos los usuarios se ha trabajado para que sea completamente responsive, de esta forma se podrá visualizar de una forma correcta tanto si se ve en un ordenador como en un teléfono móvil.



Por ello se ha optado por un diseño web sencillo e intuitivo, distribuida en cuatro páginas para que todo usuario que quiera crear un servidor lo pueda hacer de una manera sencilla y rápida, para ello se han creado 4 páginas con las que los usuarios podrán informarse, crear y gestionar servidores y una página de contactos con las que podrán informarse del equipo que ha desarrollado el equipo, así como un formulario para que los usuarios puedan notificar posibles errores.

La página web ha sido desarrollada desde cero utilizando la herramienta Sublime text, con los lenguajes HTML y CSS para la construcción, maquetar y diseño de la web, JavaScript para mejorar la distribución de la página web, entre ellos partes de los formularios y PHP para la interacción de la página web con el servidor así como con una base de datos MySQL.



Es el lenguaje utilizado para la creación de la página, la quinta revisión importante del lenguaje básico de la World Wide Web.



Es un lenguaje de diseño gráfico utilizado para definir y crear la presentación de los documentos HTML.



Lenguaje de programación interpretado, está implementado como parte de los navegadores web y permite mejor la interfaz de usuario así como hacer las webs más dinámicas.



Lenguaje de programación de propósito general, permite a las webs tener un contenido dinámico.


Página de Inicio

Será la página que se les muestre a los usuarios que accedan, en ellas obtendrán información sobre cuál es el servicio que se ofrece, el objetivo de esta página es el de atraer a los usuarios e informarles.

Que Somos

La idea que hay detrás de Continuos Hosting es la de facilitar una característica de los videojuegos más importantes en estos últimos tiempos que no es otra que la del juego multijugador, muchos usuarios se compran un videojuego para poder jugar con amigos o porque les gusta el hecho de jugar en un entorno online y compartir experiencias con desconocidos, el problema que hay en esto es que no siempre las empresas que han desarrollado estos videojuegos soportan un servidor público para que los jugadores puedan disfrutar de este modo multijugador. Por ello el objetivo es el de implementar de una manera sencilla que estos usuarios que quieran jugar en este modo puedan hacerlo ofreciéndoles facilidades.

Para ello lo que se plantea desarrollar es una herramienta web en la que se puedan crear, configurar, modificar y eliminar servidores, mediante una interfaz sencilla el objetivo es que los usuarios puedan disfrutar de un espacio en un servidor el cual hará de host de un servidor de videojuegos y ofrecerá una dirección IP y un puerto para poder conectarse a este y empezar a jugar.



* Fragmento de la página de inicio

Configuración de los Servidores

Para la creación de servidores y la monitorización del estado del mismo lo que se realiza es comunicar esta página con una base de datos MySQL, para ello cuando el servidor realiza una petición al servidor para ejecutar un script espera una respuesta que utiliza para realizar la consulta SQL pertinente a la base de datos.

En esta página los usuarios podrán:

Crear un servidor

La creación es muy simple, se tendrá que introducir el nombre de cómo quieren que se llame el servidor, una vez hecho se tendrá que pulsar el botón de crear, cuando se pulse el botón, la página web enviará una señal al servidor para que clone el servidor por defecto ya creado.

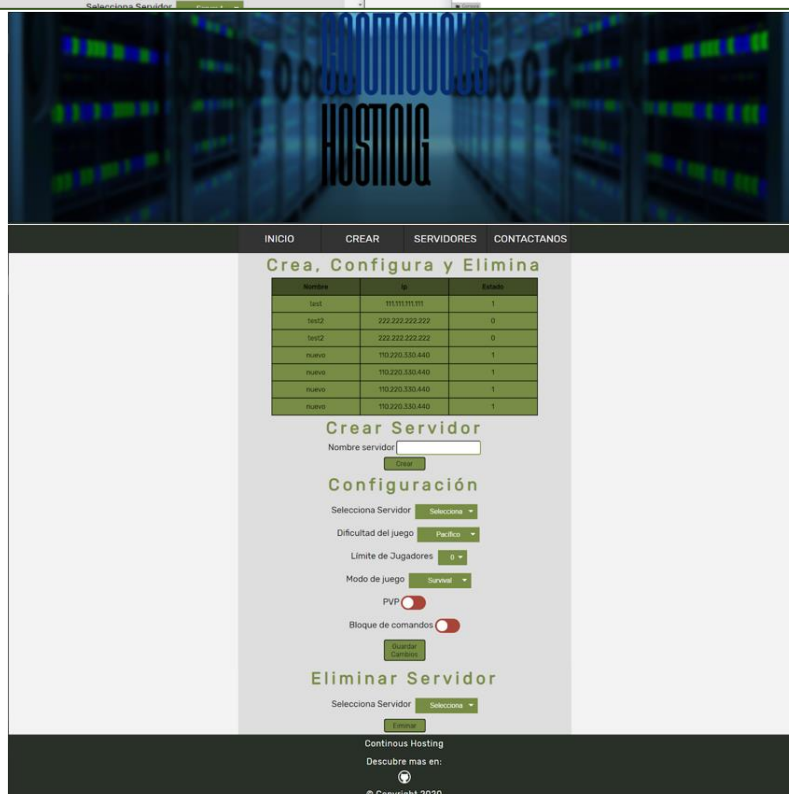
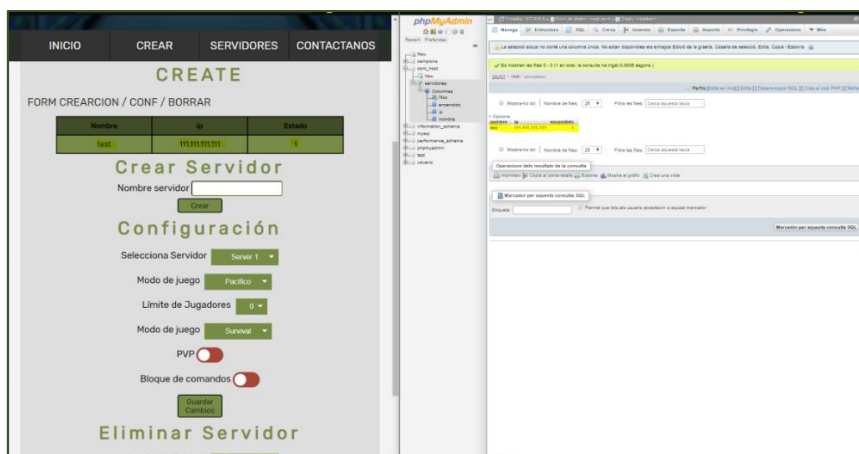
Configurar un servidor

Mediante el formulario mostrado se podrá modificar la configuración del servidor, para empezar se tendrá que seleccionar que servidor se quiere modificar, una vez hecho se podrán modificar los parámetros del formulario al gusto del usuario, cuando esté todo configurado se tendrá que pulsar el botón de Guardar Cambios, una vez pulsado la página web mandará una señal al servidor que mediante scripts modificara los archivos para configurarlos como se ha indicado.

Eliminar Servidor

Cuando un servidor ya no se esté utilizando se podrá eliminar, para ello se tendrá que seleccionar el servidor que se desea eliminar y pulsar el botón de eliminar, la web mandará una señal al servidor que eliminará el contenedor por completo.

Todo esto está respaldado por una base de datos MySQL que se actualizará con cada nuevo servidor recibiendo el nombre introducido por el usuario, y recibirá la IP y el estado de la misma del servidor para poder mostrar en la siguiente página la información de cada uno.



Control de Servidores

En esta página se mostrara el estado de todos los servidores creados, así como el nombre y la IP que tendrán que introducir en el juego para poder conectarse a él y jugar.

En esta web se recibirá de la base de datos la tabla con los servidores.



INICIO CREAR SERVIDORES CONTACTANOS

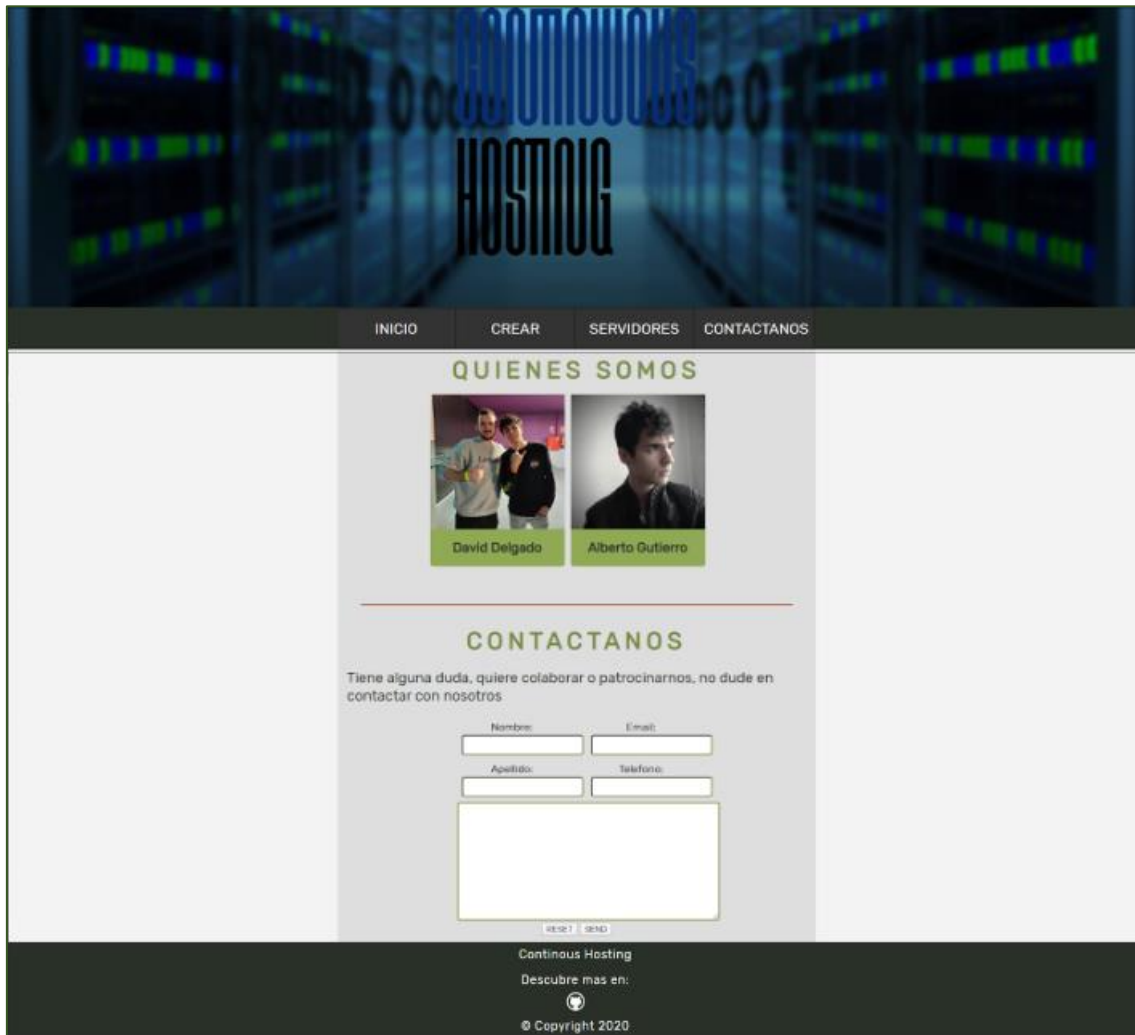
LISTA DE SERVIDORES ACTIVOS

Nombre	ip	Estado
test	111.111.111.111	1
test2	222.222.222.222	0
test2	222.222.222.222	0
nuevo	110.220.330.440	1
nuevo	110.220.330.440	1
nuevo	110.220.330.440	1
nuevo	110.220.330.440	1

Continuous Hosting
Descubre mas en:
© Copyright 2020

Contacto

En esta página se podrá encontrar un pequeño apartado en el que se mostrará la información de los desarrolladores del proyecto, además de un formulario para poder informar de posibles errores. Cuando los usuarios introduzcan los datos y manden la información la web mandará un correo para informar inmediatamente del error.



Para realizar el contacto desde la página web se utiliza una librería llamada PHP Mailer, esta, es de las librerías más comunes que se utilizan para enviar correos sin disponer de un servidor de correo propio. Gracias a esta librería, el correo se enviara directamente si tener que utilizar la “función” mailto que puedes utilizar en los formularios de HTML para enviar correos de forma no automática.



Servidor

El desarrollo del servidor para este proyecto se ha llevado a cabo mediante la herramienta Virtual Box la cual permite crear máquinas virtuales en las que se le pueden instalar los sistemas operativos de los cuales se disponen imágenes ISO para ello. El sistema operativo elegido ha sido Ubuntu 19, este sistema es una distribución de Linux y es un sistema operativo de software libre y código abierto, esto permite que se pueda configurar y adaptar de una forma más amplia las necesidades.

Este sistema ofrece una herramienta para gestionar contenedores llamada LXC (Linux Containers) que se basa en la funcionalidad cgroups de Linux, implementada desde la versión 2.6.29. Gracias a esta herramienta podemos tener más de un sistema en nuestro servidor introduciendo en cada uno de ellos el servidor de juegos deseado para cada ocasión.

La idea inicial para gestionar cada contenedor es utilizar scripts combinados con Ansible.

La página web tendrá que acceder a los scripts creados en el servidor para que se ejecuten comandos y se modifiquen archivos, de esa forma se autogestionaría de forma automática cuando se desee crear un nuevo servidor o modificar la configuración del mismo.

Con Ansible se utilizarán principalmente los módulos lineinfile con el que se podrá realizar una búsqueda en un archivo y sustituirlo por la línea que se le indique, el módulo shell que permitirá ejecutar un comando cualquiera y el módulo template que permite tener una plantilla de un archivo añadiendo algunas variables que se sustituirán por parámetros que se reciban de la web, este último módulo es muy útil para la modificación del archivo de configuración del servidor, ya que es una de las funcionalidades que tiene el proyecto.



Contenedores

Como se ha ido comentando la metodología para utilizar los contenedores es la de preparar un contenedor que esté preparado para su ejecución, en el caso de que se necesitará un servidor nuevo se procedería a una copia del servidor preparado, se le cambiaría el nombre y se editaría el archivo de configuración.

Para realizar la clonación de un contenedor es tan simple como realizar el comando:

```
lxc copy MineDef Server1
```

Donde “MineDef” será el nombre del contenedor y “Server1” será el nombre del nuevo servidor, el hecho de realizar este tipo de gestión es porque es un método sencillo rápido y eficaz, con solo un comando se realizan todos los pasos que son necesarios para descargarse e iniciar un servidor de Minecraft, para hacer esto de una forma más automatizada lo que se realizara será una llamada un script, en este script se ejecutara la línea anteriormente dicha, además de proporcionar una dirección IP que se le asignara a ese contenedor y se modificara en el archivo de configuración del servidor, de esa forma cuando el usuario inicie el servidor se ejecutara el comando de iniciar el servidor que estará añadido en el archivo rc.local, este archivo permite ejecutar un script al indicarse el sistema por lo que cada vez que se inicie automáticamente se iniciara el servidor de Minecraft. Por los recursos disponibles se ha pensado este proyecto para un máximo de 10 contenedores, para de esa forma no sobrecargar el sistema.

Modificación del archivo de configuración

Uno de los puntos importantes que tiene el videojuego de Minecraft es que se pueden editar algunos puntos de la configuración para que las partidas tengan una dificultad más elevada o un modo en el que quedan disponibles todas las herramientas que ofrece el juego para poder crear construcciones al gusto, en cuando a los servidores se refiere esto no se queda atrás pues en el archivo de configuración se pueden gestionar estas cosas, así como sí se desea que los jugadores se puedan dañar entre ellos, el límite de jugadores...

Para realizar el cambio de esta configuración se dispondrá de un formulario en la página web, esta mandara los parámetros a un script que ejecutara un comando de ansible con el módulo template.

Este módulo permite crear un archivo j2 (Jinja2 templating language), en este se introducirán en los puntos que se quieran editar los parámetros las etiquetas {{variable1}}, {{variable2}}, Estas etiquetas en el playbook serán asignadas por los valores que se han recibido de la página web una vez que se han filtrado por el script para introducir el valor necesario en cada caso, haciendo así que se sustituyan cada una con su correspondiente valor, después se realiza una copia

del archivo con los cambios realizaos al contenedor en cuestión y se reiniciará para que se guarden los cambios en el servidor y se vuelva a iniciar.

```
GNU nano 4.3 server.j2
#Minecraft server properties
#{{fecha}}
broadcast-rcon-to-ops=true
view-distance=10
max-build-height=256
server-ip={{ variable1 }}
level-speed=
rcon.port={{ variable2 }}
gamemode=survival
server-port={{ variable3 }}
```

*Archivo Jinja2 que hará de template.

```
vars:
  date: '{{ template_run_date }}'
  variable1: '$1'
  variable2: '$2'
  variable3: '$3'
tasks:
  - name: Configuración inicial Servidor
    template:
      src: /home/usuario/server.j2
      dest: /home/user/ServerMinecraft/server.properties
```

*Playbook con el módulo template

Inicio, para y eliminación de los contenedores

Para iniciar, parar y eliminar un contenedor se realizara mediante un script que contendrá el comando necesario en cada caso:

Para encender→**lxc start Server1**

Para apagar→**lxc stop Server1**

Para eliminar→**lxc delete Server1**

Red

La red es un punto importante pues sin ella el servidor no será visible y por lo tanto ningún usuario podrá acceder a él para conectarse, para este proyecto que es un prototipo de lo que podría ser en un futuro un servidor de hosting funcional únicamente los usuarios de una misma red podrán acceder al servidores puesto que para que fuese funcional al completo se hubiera necesitado una máquina que hiciera de servidor, así como un pagar un dominio en el que el servidor sería visible, si hubiese sido así los usuario se podrían conectar utilizando el dominio junto al puerto destinado en cada servidor.

En este caso al no precisar de esos recursos se ha decido limitarlo a la red local, y como se ha comentado anteriormente a un rango de 10 IP (sobre todo por no sobrecargar al sistema), para ello se ha dispuesto de un servidor dhcp con un rango de la red que va el 50 al 60, para ello se ha utilizado el servidor ics-dhcp-server.

```
subnet 192.168.0.0 netmask 255.255.255.0{
    range 192.168.0.50 192.168.0.60;
    option routers 192.168.0.1;
    option broadcast-address 192.168.0.255;
    option domain-name-servers 8.8.8.8, 8.8.4.4;
}
```

El objetivo es que la tarjeta red que ofrece señal a los diferentes servidores les ofrezca una de estas direcciones IP.

(Esta parte de la implementación lamentablemente no ha sido satisfactoria)

Otra opción a barajar sería trabajar con un fichero que ofreciera las ip de forma dinámica para de esa forma que todos los servidores tuvieran su propia ip, para ello en la creación de un contenedor se utilizaría el comando `ip add "dirección ip"`.

Trabajo Futuro

Para el futuro lo primero que se realizaría sería conseguir que el proyecto primeramente, fuera 100% funcional, para esto se procedería a la investigación profunda sobre como poder modificar la configuración de LXC o como conseguir un puente entre los contenedores y una tarjeta de red dispuesta a ello, ahora que no hay otras entregas de por medio y sin límite de tiempo de entrega creemos que con algunos meses de búsqueda e intentos se podría acabar teniendo una solución, o investigar en más profundidad de la que ya se ha realizado otro tipo de tecnología de virtualización de contenedores que incluya esta función.

El siguiente paso sería el de la optimización de los archivos para lograr la automatización de las tareas, todo y que los scripts que se han realizado cumplen bastante bien su papel se tendría que interiorizar más las consultas a la base de datos para la gestión. Así mismo la utilización de Ansible para la gestión general también se aprovecharía para monitorizar los contenedores, mediante un módulo de ansible para lxc.

(https://docs.ansible.com/ansible/latest/modules/lxd_profile_module.html)

Por ello una implementación total de ansible haría que fuera todo mucho más óptimo, de la misma manera se podría plantear el cambio de los scripts de Bash a Python, ya que ansible utiliza este lenguaje y podría aumentar la eficiencia.

Una vez que estuviera todo operativo es cuando se procedería a retocar ciertas cosas de web, para empezar implementar perfiles de usuario para que los servidores sean visibles únicamente por los usuarios que los han creado, esta tarea no se planteó para el inicio del proyecto por la complejidad que suponía barajándolo entre las opciones de implementación si se conseguían todos objetivos, asimismo también se tendría que trabajar mucho en mejorar el aspecto de la seguridad.

Para finalizar una vez completadas todas estas tareas, se procedería al lanzamiento a internet del servicio, obteniendo un dominio con el que la gente pueda acceder desde sus navegadores, pudiendo así crear servidores.

Para acabar y poder obtener beneficios se implantarían algunos anuncios no intrusivos, a modo de banners para que la monetización recibida se utilizara para mantener el dominio, y la mejora de los equipos que hostearian todo continuous hosting.

Marketing

Para promocionar el servicio que se ofrece el principal activo sería mediante las redes sociales, pues al no ser un servicio de suscripción no se obtendrían los suficientes beneficios para pagar spots publicitarios o posicionamiento web, el objetivo principal es ofrecer un servicio gratuito para los jugadores que no puedan sostener un servidor de Minecraft o quieran hacerlo de una forma más cómoda.

Para ello se procedería a crearse perfiles en las diferentes redes sociales para informar al mayor público posible sobre lo que Continuous Hosting ofrece.

Conclusiones

En cuando al proyecto se refiere ha sido reconfortante realizar un proyecto del que no se precisaba mucha información y del que mediante los conocimientos adquiridos durante el curso así como buscando herramientas que pudieran facilitarnos el proceso, nos ha motivado mucho a intentar realizar un proyecto lo más fiel posible a lo que buscamos en un principio, todo y aunque se nos han quedado cosas en el tintero por falta de conocimientos y tiempo. Aunque el resultado final no ha sido el deseado ya que lo que se buscaba era algo funcional, estamos contentos con el trabajo que se ha realizado. Creemos que con un poco más de esfuerzo y tomándolo con más calma en los próximos meses se podría encontrar una solución para poder disponer del servidor completamente funcional y poder así pensar si hacerlo público para poder aprovecharlo nosotros mismo a la hora de jugar.

Nos ha gustado mucho conocer métodos de programación web, ya que ambos estamos realizando el 3x2 y venimos de DAM, poder adquirir más experiencia sobre la programación en otro entorno ha hecho que le hayamos dedicado mucho mimo sobre todo a la parte web, haciéndola responsive para dispositivos móviles, conectándola con una base de datos y como hacer que interactúe con el propio servidor que lo está hosteando.

Aunque la realización de este proyecto ha sido tortuosa, el caso de excepcionalidad que se ha pasado, ha hecho que solo se hayan podido realizar aproximadamente unas 10 horas presenciales destinadas al proyecto, por ello quizá nos han faltado ciertas guías que nos podrían a ver venido bien a la hora del desarrollo del mismo.

De la misma manera el no poder disponer de los conocimientos de los docentes ha provocado que algunos errores no se hayan podido solucionar de la forma más optima.

Somos conscientes de que con un poco más de organización en cuanto a tiempo hubiera logrado un resultado mejor pero los fallos y problemas sucedidos han consumido muchísimo tiempo, la obcecación con intentar realizar el proyecto funcional ha provocado centrarse demasiado en ese punto, gastando demasiado tiempo que se podría haber aprovechado para obtener un resultado mejor.

Problemas y Errores

Problemas

- Uno de los problemas que más ha entorpecido, ha sido la dificultad de desarrollar de forma asíncrona, al no poder mantener contacto, a la hora de realizar ciertas pruebas se tenían que traspasar ciertos archivos mediante la red, lo que ha provocado una ralentización considerable a la hora de compartir los avances del desarrollo, pues se dependía de la velocidad de internet de subida y posteriormente de bajada para por ejemplo archivos ova de gran capacidad.
- Otro problema que se ha encontrado ha sido la realización de tareas de demás asignaturas durante el periodo excepcional, pues algunas han sido algo complicadas y más sin la presencia del profesor para resolver dudas de una forma tan directa como de forma presencial, esto ha causado que no se haya podido dedicar tanto tiempo.

Errores

El principal error, con el que se ha estado más tiempo y que no se ha logrado solucionar de ninguna de las maneras ha sido el de conseguir que los contenedores obtuvieran una dirección ip de modo adaptador puente, por la configuración normal se configura en modo nat por lo que no es visible para las maquinas externas, para intentar conseguir esto se han realizado varias cosas que han acabado en absoluto fracaso.

LXC permite tener varios perfiles de red a la vez, de forma determinada viene una que se basa en la tecnología puente nat, lo que hace que no sea visible para los equipos fuera de su red.

Esto es algo que entorpece el desarrollo del proyecto, por lo que se ha tenido que intentar modificar la configuración determinada para que si fuera visible desde otros dispositivos.

Lo primero que se tiene que realizar es la instalación del Networ-Manager, gracias a esta herramienta se creara un puente para que posteriormente pueda ser la que de la red a los contenedores lxc, para empezar tendremos que crear el puente, para ello ejecutaremos el siguiente comando:

```
usuario@continuous:~$ sudo nmcli con add ifname br0 type bridge con-name br0
Connection 'br0' (b60ccdf8-6b10-42a7-879e-b361337f7656) successfully added.
```

También se procederá a crear otro tipo de red esclava para prevenir.

```
usuario@continuous:~$ sudo nmcli con add type bridge-slave ifname eno1 master br0
Connection 'bridge-slave-eno1' (6f5137bc-95b3-46bc-bd7d-973faee3c51d) successfully added.
```

```

usuario@continuous:~$ nmcli connection show
NAME                UUID                                TYPE      DEVICE
br0                  b60ccdf8-6b10-42a7-879e-b361337f7656  bridge    --
bridge-slave-en01    6f5137bc-95b3-46bc-bd7d-973faee3c51d  ethernet  --

```

Tras haber hecho todo esto se ha encontrado que la maquina estaba utilizando el servicio de Networkd, por lo que había un conflicto, por lo tanto bajo consulta se decicio proceder a realizar un puente con el servicio Networkd, desinstalando también el Network-Manager ya que no es necesario.

```

usuario@continuous:~$ sudo apt --purge remove network-manager
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
  libbluetooth3 libdumbnet1 libjansson4 libmbim-glib4 libmbim-proxy libmm-glib0 libndp0 libnm0
  libqmi-glib5 libqmi-proxy libteamdctl0 modemmanager ppp ptp-linux usb-modeswitch
usb-modeswitch-data
Utilice «sudo apt autoremove» para eliminarlos.
Los siguientes paquetes se ELIMINARÁN:
  network-manager* network-manager-ppp*
0 actualizados, 0 nuevos se instalarán, 2 para eliminar y 0 no actualizados.
Se liberarán 7.713 kB después de esta operación.
¿Desea continuar? [S/n] _

```

El siguiente paso es crear el archivo `uplink.network`, que será un archivo para configurar el adaptador de red.

```

GNU nano 4.3 /etc/systemd/network/uplink.network
[Match]
Name=enp0s3

[Network]
Bridge=br0

```

El siguiente paso es configurar el para informar a systemd sobre en nuevo dispositivo de red puente asi como especificar la configuración de IP que queremos que tenga para ello tendremos que crear un archivo.

```

GNU nano 4.3 /etc/systemd/network/br0.netdev
[NetDev]
Name=br0
Kind=bridge

```

Lo que se hace con este archivo es decirle a systemd que se desea un dispositivo llamado `br0` que funcione como un puente de ethernet. Lo siguiente será crear un archivo para especificar la configuración de IP para la interfaz `br0`.


```
GNU nano 4.3 /etc/systemd/network/br0.network
[Match]
Name=br0

[Network]
DNS=8.8.8.8
DHCP=ipv4
Gateway=192.168.0.1
```

Para que esta tarjeta que hemos creado tenga dirección ip lo que tendremos que hacer es también configurar un servidor de dhcp, para ello instalaremos ics-dhcp-server y le pondremos la siguiente configuración para que haga una asignación automática.

```
subnet 192.168.0.0 netmask 255.255.255.0{
    range 192.168.0.50 192.168.0.60;
    option routers 192.168.0.1;
    option broadcast-address 192.168.0.255;
    option domain-name-servers 8.8.8.8, 8.8.4.4;
}
```

El segundo paso que se ha tenido que realizar ha sido la creación de un nuevo perfil de red, para ello se ha tenido que ejecutar el comando:

Lxc profile create <<nombre_adaptador>>

Una vez ejecutado con el comando lxc profile list podremos ver los perfiles que hay y cuandos contenedores los utilizan.

```
usuario@continuous:~$ lxc profile list
+-----+-----+
| NAME | USED BY |
+-----+-----+
| adaptadorpuente | 0 |
+-----+-----+
| default | 1 |
+-----+-----+
```

El siguiente paso es el de editar el perfil que se ha acabado de crear, para ello tendremos que utilizar el comando lxc profile edit <<nombre_adaptador>>, lo que nos abrirá un archivo con el editor de texto vi, en este caso se ha cambiado la configuración a la siguiente:

```
config: {}
description: Perfil de Adaptador Puente para las redes LXD
devices:
  eth0:
    name: eth0
    nictype: bridged
    parent: bridge0
    type: nic
name: adaptadorpuente
used_by: []
```

Tras iniciar un contenedor utilizando como conector de red la tarjeta que hemos creado, el resultado ha sido negativo pues el contenedor no recibía ninguna dirección IP.

```
usuario@continuous:~$ lxc profile assign MineDef default,adaptadorpuente_
```

Otro de los intentos ha sido el de modificar la configuración que ya viene por defecto, intentando así que la tarjeta de red que crea por defecto no utilizara una red tipo nat, para esto se puede ver la configuración utilizando el comando.

```
usuario@continuous:~$ lxc network show lxdbr0
config:
  ipv4.address: 10.223.126.1/24
  ipv4.nat: "true"
  ipv6.address: fd42:467f:3cce:a995::1/64
  ipv6.nat: "true"
description: ""
name: lxdbr0
type: bridge
used_by:
- /1.0/instances/MineDef
- /1.0/instances/MineTest2
managed: true
status: Created
locations:
- none
```

En un blog se encontró con una posible solución, lo que planteaban era añadir el parámetro `ipv4.routes`, esto se utilizaría para iniciar la dirección de IP pública para que LXC hiciera de puente.

Para introducirlo se tenía que introducir el comando LXC network ser `lxdbr0` `ipv4.routes` "dirección ip", de esa forma se añadía a la configuración.

```
usuario@continuous:~$ sudo lxc network set lxdbr0 ipv4.routes 192.168.0.1
usuario@continuous:~$ lxc network show lxdbr0
config:
  ipv4.address: 10.223.126.1/24
  ipv4.nat: "true"
  ipv4.routes: 192.168.0.1
  ipv6.address: fd42:467f:3cce:a995::1/64
  ipv6.nat: "true"
description: ""
name: lxdbr0
type: bridge
used_by:
- /1.0/instances/MineDef
- /1.0/instances/MineTest2
managed: true
status: Created
locations:
- none
```

Al iniciar una nueva máquina el problema continuaba estando, ya que no se conseguía que el contenedor se viera desde fuera.

Tras esta opción a los desarrolladores se les ocurrió una idea y es que como se puede ver en la configuración de la tarjeta hay un parámetro que indica “ipv4.nat” que se encuentra en estado “true”, es decir que estaría habilitado, se procedió al cambio del estado de ese parámetro, ya que todo apuntaba a que era el responsable que la red se viera en modo nat, para realizar el cambio simplemente había que realizar el comando anterior pero esta vez con el parámetro deseado.

```
usuario@continuous:~$ sudo lxc network set lxdbr0 ipv4.nat false
usuario@continuous:~$ lxc network show lxdbr0
config:
  ipv4.address: 10.223.126.1/24
  ipv4.nat: "false"
  ipv4.routes: 192.168.0.1
  ipv6.address: fd42:467f:3cce:a995::1/64
  ipv6.nat: "true"
description: ""
name: lxdbr0
type: bridge
used_by:
- /1.0/instances/MineDef
- /1.0/instances/MineTest2
managed: true
status: Created
locations:
- none
```

Como podemos ver el estado del parámetro ha cambiado por lo que la red ya no debería ser de tipo nat (al menos la de ipv4), el resultado no fue el esperado pues al cambiar este parámetro no se veía por máquinas exteriores solamente, si no que ni siquiera obtenía una dirección IP a la que conectarse.

Tras estos intentos modificando varias cosas se tomó la opción de continuar con la configuración de red más adelante ya que había consumido mucho tiempo en intentar resolver este problema que resultó ser un muro para el buen desarrollo del mismo.

Glosario

Contenedores → Sistema virtualizado utilizando el mismo kernel que la maquina real.

LXC → Herramienta utilizada para la gestión de contenedores.

Xampp → Programa que incluye un conjunto de servicios.

Ubuntu → Distribución del sistema operativo Linux.

Minecraft → Videojuego muy popular de supervivencia en mundo abierto.

Servidor Multijugador → Programa que permite la conexión entre usuarios en un mismo juego.

Videojuego → Juego electrónico en el que una o más personas interactúan por medio de un controlador.

Máquina Real → El equipo que se utiliza.

Máquina Virtual → Sistema operativo que funciona mediante una aplicación de una maquina real

Modo NAT → Modo de conexión donde una máquina virtual comparte su dirección de red con la de la maquina real.

Modo Adaptador Puente → Modo de conexión que ofrece una IP perteneciente a la red en la que se encuentra la maquina real.

Hostear/Hosting → Anglicismo que indica la opción de lanzar un servicio de alojamiento web.

Wireframe→ Esquema o plano de las pantallas de la página web.

Webgrafía

Ansible

<https://www.ansible.com/>

https://docs.ansible.com/ansible/2.9/modules/template_module.html

<https://www.mydailytutorials.com/ansible-template-module-examples/>

<https://www.toptechskills.com/ansible-tutorials-courses/ansible-template-module-tutorial-examples/>

<https://www.tecmint.com/create-ansible-templates-to-create-configuration-for-nodes/>

<https://blog.knoldus.com/ansible-playbook-using-templates/>

<https://dev.to/livioribeiro/using-lxd-and-ansible-to-simulate-infrastructure-2g8l>

https://docs.ansible.com/ansible/latest/modules/lxd_container_module.html

https://docs.ansible.com/ansible/latest/modules/lxd_profile_module.html

<https://docs.ansible.com/ansible/latest/plugins/connection/lxd.html>

<https://elpuig.xeill.net/Members/q2dg/aso-mp06/uf2/teoria-i-exercicis-ansible-i/view>

<https://elpuig.xeill.net/Members/q2dg/aso-mp06/uf2/exercicis-ansible-ii/view>

<https://elpuig.xeill.net/Members/q2dg/aso-mp06/uf2/teoria-i-exercicis-ansible-iii/view>

LXC

<https://linuxcontainers.org/>

<https://blog.simos.info/how-to-get-lxd-containers-get-ip-from-the-lan-with-routed-network/>

<https://bayton.org/docs/linux/lxd/lxd-zfs-and-bridged-networking-on-ubuntu-16-04-lts/#configuring-lxd-zfs>

<https://ubuntu.com/blog/network-management-with-lxd-2-3>

<https://ubuntu.com/blog/lxd-networking-lxdbr0-explained>

<https://lxd.readthedocs.io/en/latest/networks/>

<https://discuss.circleci.com/t/container-is-not-defined-on-lxc-attach/14684/2>

<https://blog.jorgeivanmeza.com/2015/10/los-primeros-10-minutos-con-lxc-en-ubuntu/>

<https://superadmin.es/blog/devops/ip-estatica-en-contenedores-lxd/>

<http://docs.iescdl.es/~josem/lxc.html>

<https://elpuig.xeill.net/Members/vcarcelar/articulos/mysql-en-un-contedor-lxd>

<https://manuelfrancoblog.wordpress.com/2018/01/12/lxc-lvm-containers-y-clonado/>

<https://elpuig.xeill.net/Members/vcarceler/articulos/contenedores-con-lxd-lxc>

<https://www.redhat.com/es/topics/containers/whats-a-linux-container>

<https://manuelfrancoblog.wordpress.com/2018/03/26/que-son-los-contenedores-linux-containers-o-lxc/>

<https://marcoalfonso.net/2016/09/30/contenedor-lxc-con-ip-publica/>

<https://nolte83.wordpress.com/2011/07/29/lxc-linux-containers/>

<https://www.sysadminsdecuba.com/2019/08/instalacion-y-configuracion-de-lxc-en-debian>

<https://gastack.mx/unix/50201/how-to-configure-external-ip-addresses-for-lxc-quests>

<https://github.com/lxc/lxd/blob/master/doc/configuration.md>

<https://github.com/lxc/lxd/blob/master/doc/networks.md>

<https://wiki.debian.org/LXC/SimpleBridge>

<https://blog.jorgeivanmeza.com/2015/10/los-primeros-10-minutos-con-lxc-en-ubuntu/>

Creación de la Pagina Web

https://www.w3schools.com/css/css_navbar.asp

https://www.w3schools.com/howto/tryit.asp?filename=tryhow_css_image_overlay_fade

https://www.w3schools.com/howto/howto_css_image_overlay_slide.asp

https://www.w3schools.com/howto/tryit.asp?filename=tryhow_custom_select

Creación de un logo

<https://es.freelogodesign.org/>

Paleta de colores 1a opción

<https://coolors.co/132a13-31572c-4f772d-90a955-ecf39e>

2a opción

<https://coolors.co/141914-283027-778c44-414d26-a84539>

Funcionamiento Sheel_exec:

<https://www.php.net/manual/es/function.shell-exec.php>

Documentacion MySQL

<https://dev.mysql.com/doc/>

Introducción para hacer un diseño Responsive

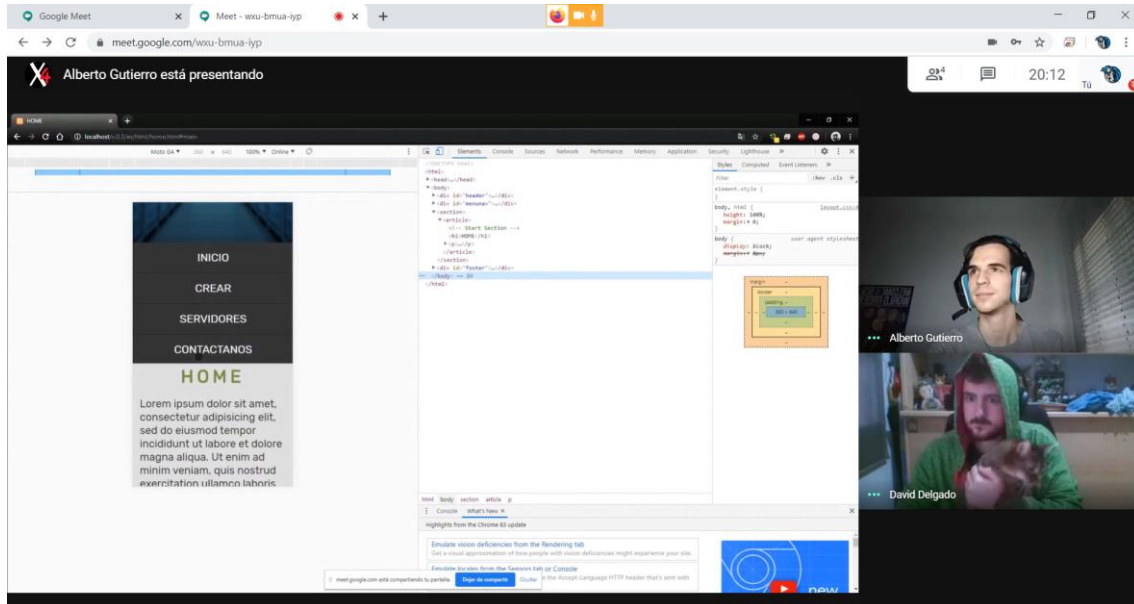
https://www.w3schools.com/html/html_responsive.asp

Introducción a la semántica

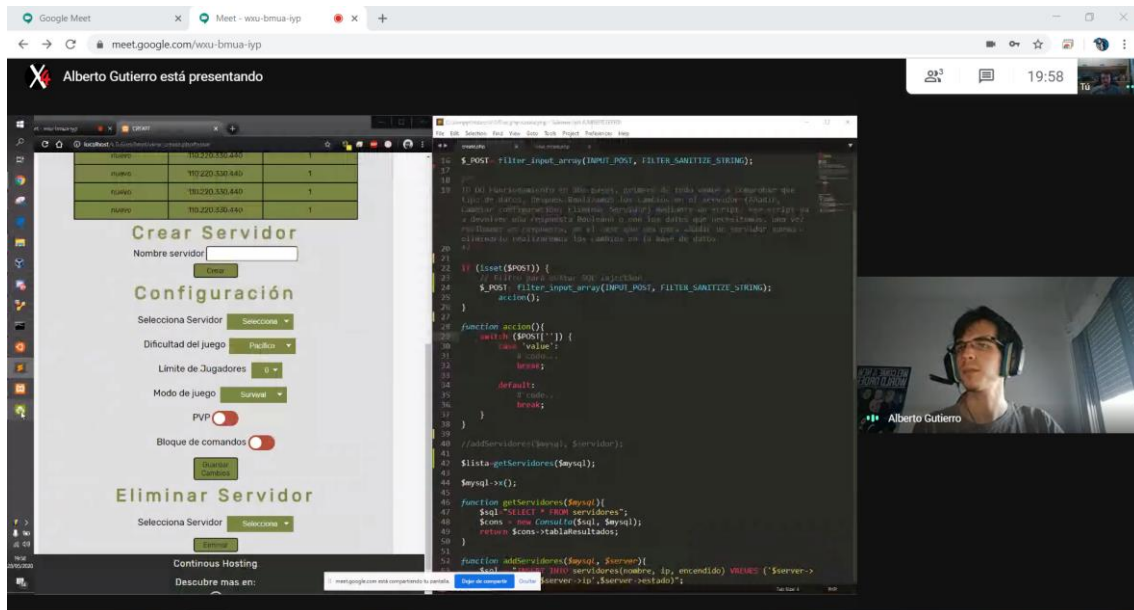
https://www.w3schools.com/html/html5_semantic_elements.asp

Anexos

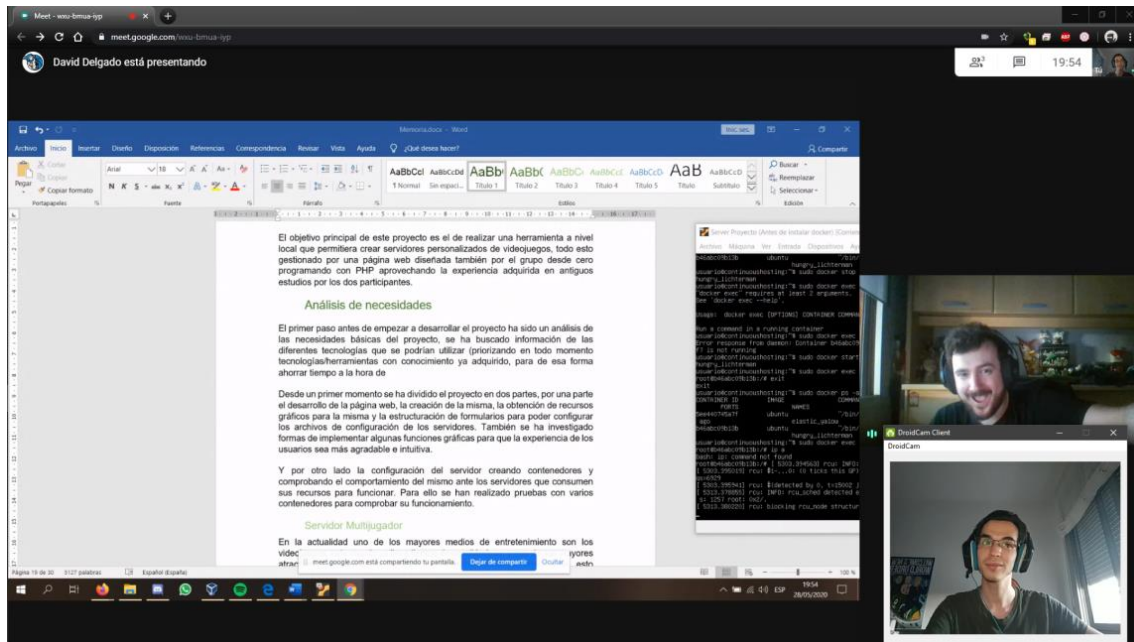
Durante algunas reuniones de los integrantes del grupo de han realizado algunas capturas que empezaron como broma pero al final se decidió implementarlas como recuerdo de la situación de excepcionalidad y por los esfuerzos requeridos.



*Imagen de la prueba de página responsive



*Imagen de la realización de la página de gestión de servidores



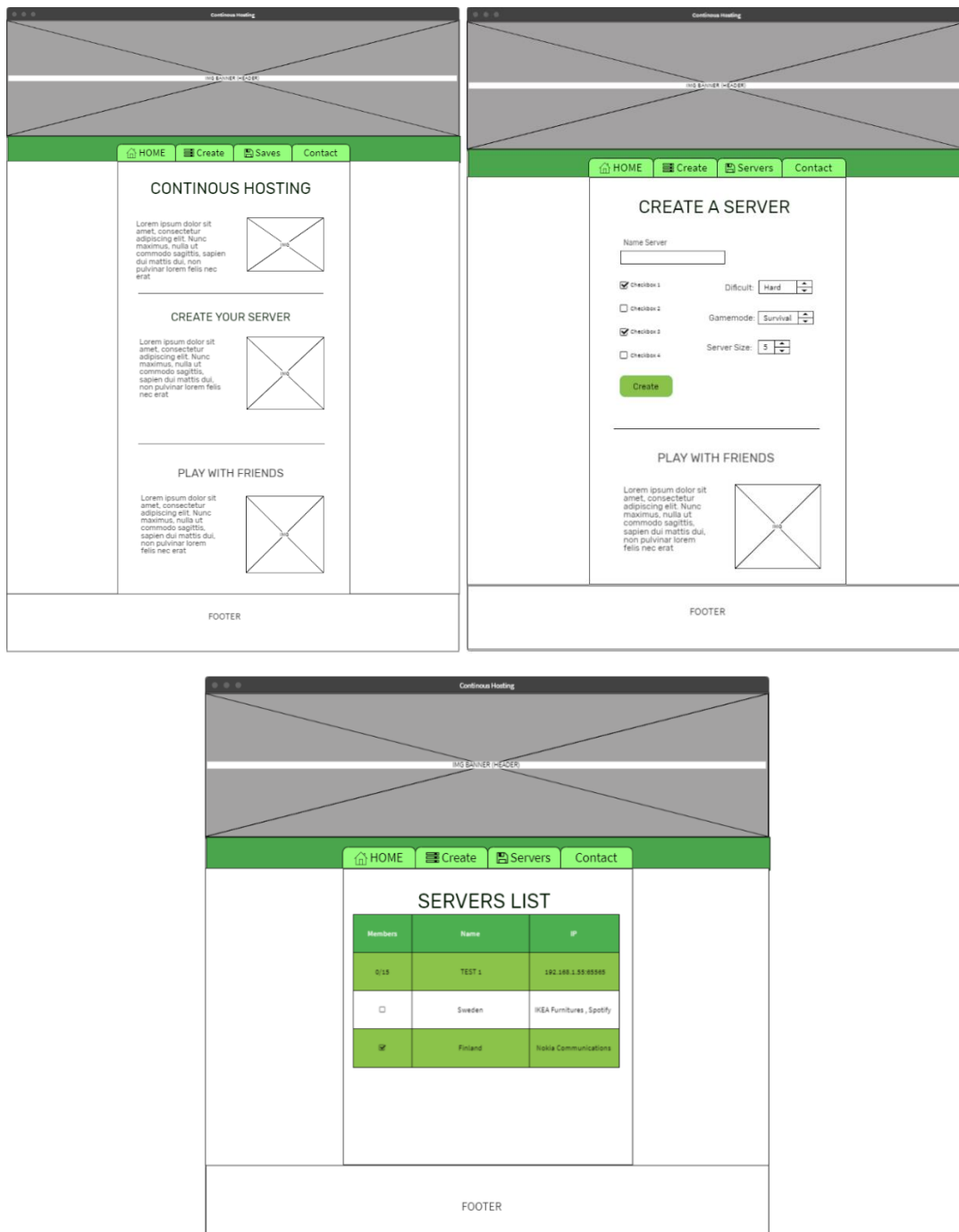
*Imagen de los últimos días para la realización de la memoria

Presupuesto

Hoja de cálculo utilizada para la realización del presupuesto.

[https://docs.google.com/spreadsheets/d/1ql2AJiJHGGQ4dzi83MsTf3-
QQFnSdr6_Z0eEU18aTdA/edit#gid=0](https://docs.google.com/spreadsheets/d/1ql2AJiJHGGQ4dzi83MsTf3-QQFnSdr6_Z0eEU18aTdA/edit#gid=0)

Wireframe



* Primera idea del diseño de la web

Prueba de Instalación y utilización de Docker

Primero se tiene que actualizar el sistema con

```
sudo apt update
```

Una vez actualizado tendremos que instalar algunos paquetes que permitan descargas web, estas serán

```
sudo apt install apt-transport-https ca-certificates curl software-properties-common
```

```
usuario@continuuoshosting:~$ sudo apt install apt-transport-https ca-certificates curl software-prop
erties-common
[sudo] password for usuario:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
ca-certificates ya está en su versión más reciente (20190110).
fijado ca-certificates como instalado manualmente.
software-properties-common ya está en su versión más reciente (0.97.11).
fijado software-properties-common como instalado manualmente.
curl ya está en su versión más reciente (7.64.0-2ubuntu1.2).
fijado curl como instalado manualmente.
Se instalarán los siguientes paquetes NUEVOS:
  apt-transport-https
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 73 no actualizados.
Se necesita descargar 1.692 B de archivos.
Se utilizarán 156 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
Des:1 http://es.archive.ubuntu.com/ubuntu disco-updates/universe amd64 apt-transport-https all 1.8.3
 [1.692 B]
Descargados 1.692 B en 0s (11,1 kB/s)
Seleccionando el paquete apt-transport-https previamente no seleccionado.
(Leyendo la base de datos ... 69669 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../apt-transport-https_1.8.3_all.deb ...
Desempaquetando apt-transport-https (1.8.3) ...
Configurando apt-transport-https (1.8.3) ...
```

Lo siguiente será instalar una clave GPG para el repositorio oficial de Docker

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

```
usuario@continuuoshosting:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key
add -
OK
```

Y añadimos el repositorio de Docker a las fuentes de APT:

```
sudo add-apt-repository "deb [arch=amd64]
https://download.docker.com/linux/ubuntu bionic stable"
```

```
usuario@continuuoshosting:~$ sudo add-apt-repository "deb [arch=adm64] https://download.docker.com/1
inux/ubuntu bionic stable"
Obj:1 http://es.archive.ubuntu.com/ubuntu disco InRelease
Obj:2 http://es.archive.ubuntu.com/ubuntu disco-updates InRelease
Des:3 https://download.docker.com/linux/ubuntu bionic InRelease [64,4 kB]
Obj:4 http://es.archive.ubuntu.com/ubuntu disco-backports InRelease
Obj:5 http://es.archive.ubuntu.com/ubuntu disco-security InRelease
Descargados 64,4 kB en 2s (35,4 kB/s)
Leyendo lista de paquetes... Hecho
```

Ahora tendremos que Volver a actualizar la base de datos de paquetes para que lea los que le acabamos de añadir, una vez hecho podemos ver que los repositorios Docker.

```
sudo apt update
```

```
usuario@continuous hosting:~$ apt-cache policy docker-ce
docker-ce:
  Instalados: (ninguno)
  Candidato: (ninguno)
  Tabla de versión:
```

Y procederemos a la instalación.

```
usuario@continuous hosting:~$ sudo apt install docker-ce
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  aufs-tools cgroupfs-mount containerd.io docker-ce-cli libltdl7 pigz
Se instalarán los siguientes paquetes NUEVOS:
  aufs-tools cgroupfs-mount containerd.io docker-ce docker-ce-cli libltdl7 pigz
0 actualizados, 7 nuevos se instalarán, 0 para eliminar y 73 no actualizados.
Se necesita descargar 85,8 MB de archivos.
Se utilizarán 385 MB de espacio de disco adicional después de esta operación.
```

Una vez que este instalado podremos ver su estado con `systemctl`, en el caso de que este apagado lo iniciaremos.

```
usuario@continuous hosting:~$ sudo systemctl status docker
• docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2020-03-11 15:43:00 UTC; 3min 22s ago
     Docs: https://docs.docker.com
    Main PID: 5428 (dockerd)
      Tasks: 11
     Memory: 39.4M
    CGroup: /system.slice/docker.service
            └─5428 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

mar 11 15:42:59 continuous hosting dockerd[5428]: time="2020-03-11T15:42:59.311230930Z" level=warning
mar 11 15:42:59 continuous hosting dockerd[5428]: time="2020-03-11T15:42:59.311252430Z" level=warning
mar 11 15:42:59 continuous hosting dockerd[5428]: time="2020-03-11T15:42:59.311274748Z" level=warning
mar 11 15:42:59 continuous hosting dockerd[5428]: time="2020-03-11T15:42:59.311699946Z" level=info ms
mar 11 15:42:59 continuous hosting dockerd[5428]: time="2020-03-11T15:42:59.654139731Z" level=info ms
mar 11 15:42:59 continuous hosting dockerd[5428]: time="2020-03-11T15:42:59.818465321Z" level=info ms
mar 11 15:42:59 continuous hosting dockerd[5428]: time="2020-03-11T15:42:59.911688820Z" level=info ms
mar 11 15:42:59 continuous hosting dockerd[5428]: time="2020-03-11T15:42:59.912352877Z" level=info ms
mar 11 15:43:00 continuous hosting systemd[1]: Started Docker Application Container Engine.
mar 11 15:43:00 continuous hosting dockerd[5428]: time="2020-03-11T15:43:00.035449096Z" level=info ms
```

Una vez que tengamos el programa instalado y hayamos comprobado que esta encendido tendremos seleccionar que sistemas queremos tener para que se ejecuten en los contenedores, para ello podremos ejecutar `Docker search "sistema"` de esta forma podremos saber todas las opciones que tiene Docker de ese sistema.

```

usuario@continuuoshosting:~$ sudo docker search ubuntu
[sudo] password for usuario:
NAME                DESCRIPTION                STARS     OFFICIAL    AUTOMATED
ubuntu              Ubuntu is a Debian-based Linux operating sys... 18605     [OK]
dorowu/ubuntu-desktop-lxde-vnc Docker image to provide HTML5 VNC interface ... 404
rastasheep/ubuntu-sshd Dockerized SSH service, built on top of offi... 243
consol/ubuntu-xfce-vnc Ubuntu container with "headless" VNC session... 211
ubuntu-upstart      Upstart is an event-based replacement for th... 106       [OK]
ansible/ubuntu14.04-ansible Ubuntu 14.04 LTS with ansible 98
neurodebian          NeuroDebian provides neuroscience research s... 67        [OK]
landinternet/ubuntu-16-nginx-php-phpmyadmin-mysql-5 ubuntu-16-nginx-php-phpmyadmin-mysql-5 50
ubuntu-debootstrap  debootstrap --variant=minbase --components=m... 43        [OK]
nuagebec/ubuntu     Simple always updated Ubuntu docker images w... 24
i386/ubuntu         Ubuntu is a Debian-based Linux operating sys... 19
landinternet/ubuntu-16-apache-php-5.6 ubuntu-16-apache-php-5.6 14
landinternet/ubuntu-16-apache-php-7.0 ubuntu-16-apache-php-7.0 13
eclipse/ubuntu_jdk8 Ubuntu, JDK8, Maven 3, git, curl, nmap, mc, ... 12
landinternet/ubuntu-16-nginx-php-phpmyadmin-mariadb-10 ubuntu-16-nginx-php-phpmyadmin-mariadb-10 11
landinternet/ubuntu-16-nginx-php-5.6 ubuntu-16-nginx-php-5.6 8
landinternet/ubuntu-16-nginx-php-5.6-wordpress-4 ubuntu-16-nginx-php-5.6-wordpress-4 7
darksheer/ubuntu   Base Ubuntu Image -- Updated hourly 5
landinternet/ubuntu-16-nginx-php-7.0 ubuntu-16-nginx-php-7.0 4
pivotaldata/ubuntu A quick freshening-up of the base Ubuntu doc... 4
pivotaldata/ubuntu16.04-build Ubuntu 16.04 image for GPDB compilation 2
landinternet/ubuntu-16-php-7.1 ubuntu-16-php-7.1 1
landinternet/ubuntu-16-sshd ubuntu-16-sshd 1
pivotaldata/ubuntu-gpdb-dev Ubuntu images for GPDB development 1
smarterentry/ubuntu ubuntu with smarterentry 1

```

En este caso vamos a instalar el sistema Ubuntu.

```

usuario@continuuoshosting:~$ docker pull ubuntu
Using default tag: latest
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.
sock: Post http://%2Fvar%2Frun%2Fdocker.sock/v1.40/images/create?fromImage=ubuntu&tag=latest: dial u
nix /var/run/docker.sock: connect: permission denied
usuario@continuuoshosting:~$ sudo docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
423ae2b273f4: Pull complete
de83a2304fa1: Pull complete
f9a83bce3af0: Pull complete
b6b53be908de: Pull complete
Digest: sha256:04d48df82c938587820d7b6006f5071dbbfffceb7ca01d2814f81857c631d44df
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest

```

Podemos ver con el comando Docker imágenes los repositorios que hemos instalado y que Docker podrá iniciar contenedores con ellos.

```

usuario@continuuoshosting:~$ sudo docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
ubuntu              latest          72300a873c2c   2 weeks ago    64.2MB

```

La prueba resulto bastante dudosa pues por varios motivos no era lo suficientemente intuitivo como LXC por ello se decidió no utilizar Docker.