|  |
| --- |
| **IES FRANCISCO DE GOYA** |
|  |
| Proyecto fin de Ciclo Formativo  Administración de Sistemas Informáticos en Red |
| **Diego Fernández Higueras Curso 2020/2021** |





**Contenido**

1. [Objetivos del proyecto 3](#_TOC_250005)
2. [Análisis de Requisitos 3](#_TOC_250004)
3. [Tecnologías 3](#_TOC_250003)
4. [Pruebas 4](#_TOC_250002)
5. [Conclusiones 4](#_TOC_250001)
6. [Bibliografía 4](#_TOC_250000)

# Información

# 

# En esta introducción hablaré sobre qué es y de que trata el cloud computing y sus funcionalidades y qué es *Microsoft Azure*.

# Palabras clave

# Cloud Computing, Nube, Proveedores, Azure, Microsoft, Software como servicio (SaaS), Plataforma como servicio (PaaS), Infraestructura como Servicio (IaaS), Nube pública, Nube privada.

# El *Cloud Computing*, computación en la nube o la nube como se le puede llamar comúnmente es una tecnología que ha estado creciendo a pasos agigantados durante los últimos años, tanto es así que casi todo el mundo sabe lo que es la nube o la utiliza, aunque entre la gente, solo se tiene el pensamiento de que es un servidor de almacenamiento, puede ser mucho más. De hecho, la mayoría de gente la usa a diario, ya que muchas aplicaciones y servicios de lo más frecuentados la utilizan, Netflix, Spotify, Gmail, Dropbox…son algunos ejemplos de aplicaciones de uso cotidiano entre la gente corriente y que se pueden describir como servicios en la nube.

# No obstante, muchas empresas, desde grandes hasta pequeñas están empezando a apostar por esta tecnología ya que tiene muchas funciones como alojar sitios web (como va a ser mi caso), BigData, almacenar datos en bases de datos, backups, crear aplicaciones, streaming de multimedia, etc.

# Más técnicamente, el *cloud computing* consiste en suministrar recursos de informática (almacenamiento, software, servidores, bases de datos, redes, máquinas virtuales y un largo etcétera) como un servicio bajo demanda a través de internet en el que el cliente suele realizar el *pago por uso*, que trata de pagar sólo por lo que usas durante el tiempo que lo usas, nada más ni nada menos.

# Los encargados de suministrar estos servicios son llamados proveedores

# Estos proveedores pueden ser de dos tipos:

# Nube pública, como Microsoft Azure, que comenzó en 2010, es llamada nube pública porque es el proveedor el que es propietario de los recursos necesarios para mantener la nube (servidores, hardware…) por lo que todo el mundo puede acceder a ella. Tienen como principales ventajas la escalabilidad y el nulo mantenimiento puesto que de eso se encarga el proveedor. Otros proveedores son Google Cloud o Amazon Web Services (AWS) entre muchos otros.

# Nube privada, a diferencia de la pública, es un recurso que tienes que instalar en tu sistema propio para poder utilizarlo. Como ventajas principalmente destacan su mayor control y su mayor seguridad. La más conocida es Opensatck pero hay otras como OpenNebula, VMware vCloud, etc.

# Para finalizar los servicios que se ofrecen con estas nubes se pueden clasificar principalmente en tres tipos:

# Software como servicio (SaaS)

# Plataforma como servicio (PaaS)

# Infraestructura como servicio (IaaS), la más común.

# Objetivos del proyecto

Usaré una tecnología en auge como es el *Cloud Computing*, cada vez más usado por las empresas y también a nivel de usuario, ya que es una mejor opción de cara a las máquinas virtuales al estar alojadas en la nube y no en la propia máquina *host*. He decidido hacer este proyecto debido al potencial que creo que tiene esta tecnología (o conjunto de tecnologías) y puesto que he visto que en mi empresa también es un recurso recurrente como muchos otros que he utilizado en la realización de este proyecto ya que los he encontrado realmente útiles y prácticos, hasta el punto de que empresas como la mía lo utilicen habitualmente, además también lo he elegido porque en este mismo proyecto se utiliza prácticamente todo lo que hemos dado en clase estos años. La meta es crear una instancia alojada en la nube que, a su vez, aloje como si de un servidor web se tratase la página web que creamos en Base de Datos basada en PHP y conectada a una base de datos MySQL totalmente funcional y modificable en línea sin consumir apenas ningún recurso de mi ordenador. Estas son, de forma resumida, las metas a cumplir:

* + Conseguir acceso a una nube pública (*Azure* en este caso).
  + Configurar la nube y crear el entorno virtual adecuado (redes, instancias, direcciones IP, dominios…).
  + Ser capaz de conectarme a la instancia a través de SSH.
  + Configurar la instancia instalando los paquetes necesarios.
  + Atribuir una IP pública estable a la instancia y asignarle un dominio para que sea más accesible.
  + Configurar la base de datos.
  + Instalar paquetes y configurar la instancia para “convertir” la instancia en un servidor web que muestre la página web del proyecto de este curso de Base de Datos.

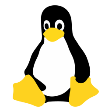
# 2.1 Análisis de Requisitos

* + - Una máquina física o virtual donde instalar y/o acceder al sistema de gestión de la nube.
    - Conexión a internet.
    - Un sistema de computación en la nube.
    - Un entorno de la nube correctamente configurado, conformado por usuarios, redes, dominios, direcciones IP, imágenes e instancias que funcionen y tengan acceso a internet.
    - Una instancia funcional y accesible desde el *host*.
    - Una base de datos MySQL.
    - Una página web en PHP que esté conectada a la base de datos.

# Tecnologías

1. Windows 10

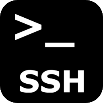
Será el sistema operativo que utilizaré en el host.

1. Linux (Ubuntu server 20.04)

Será el sistema que utilizaré para la máquina virtual dado a que es ligero, rápido, consume poco y es más conveniente para la tarea dada.

1. PuTTY

Programa utilizado para la conexión con el huésped de forma segura mediante claves y el protocolo SSH.

1. SSH

Lo utilizaré para conectarme de forma segura a través del puerto 22 a la instancia desde la máquina *host*.

1. SCP

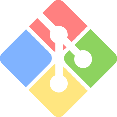
Para transferir los archivos puntualmente desde el *host* al huésped y viceversa.

1. MySQL

Sistema de base de datos utilizado para la misma que interacciona con la página web.

1. E:\Descargas\pngwing.com (2).pngGitHub

Servirá para subir los avances del proyecto, como los archivos de la página web y el sql de la base de datos y poder así transferirlos cómodamente entre el *host* y el huésped.

1. Git/GitBash

La consola de Git para usarlo desde Linux o Windows en el sistema *host.*

1. Azure

Sistema de computación en la nube para implementar la infraestructura alojada en la nube, como redes, dominios, máquinas virtuales así como sus configuraciones.

1. PHP

Lenguaje de marcas utilizado en la programación de la página web en la implementación de la base de datos en esta.

1. HTML

Lenguaje de marcas utilizado en la programación de estructuras básicas de la página web como tablas.

1. CSS

Lenguaje de diseño gráfico utilizado en la página web para ajustar el diseño, fuente, colores, etc.

# Procedimiento

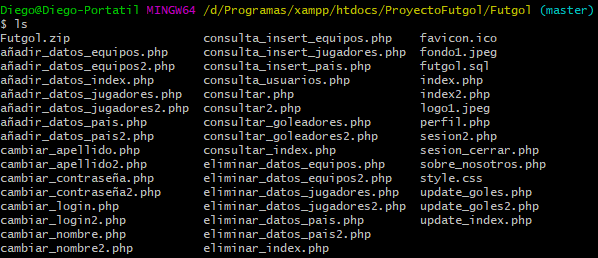
En este apartado voy a explicar paso a paso el procedimiento que he llevado a cabo para implementar y configurar todos los componentes necesarios para la realización del proyecto.

* 1. **-Comienzo: Git en el host.**

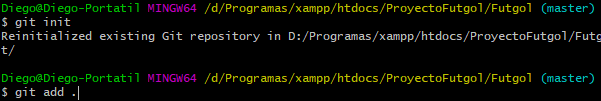
Para empezar, subiremos todo el código de la página web que ya está previamente diseñado a GitHub para así poder desplegarlos sin problemas en otros ordenadores, incluida la VM que es la que nos interesa en este caso.

Para ello, debemos tener como requisito una cuenta de GitHub e instalar el programa GitBash en caso de estar en Windows o Git en caso de Linux para poder ejecutar los comandos del propio programa.

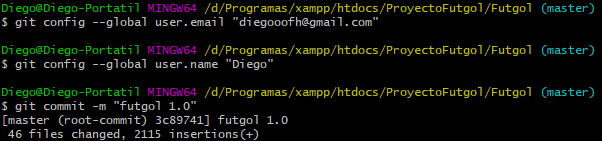
Comenzamos dirigiéndonos a la carpeta contenedora del código y comprobamos que efectivamente esta todo correcto.



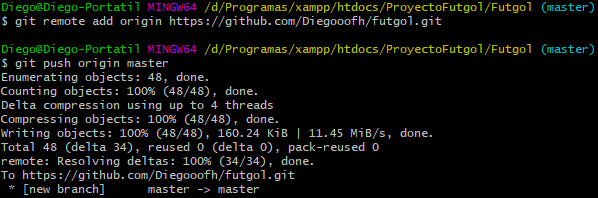
A continuación, iniciamos Git y seleccionamos los archivos.



Después, nos solicitará datos y haremos un commit para confirmar este trabajo, le daremos un nombre a la acción.



Para finalizar, nos conectamos a nuestro repositorio en la nube de GitHub y hacemos un push para subir los archivos.



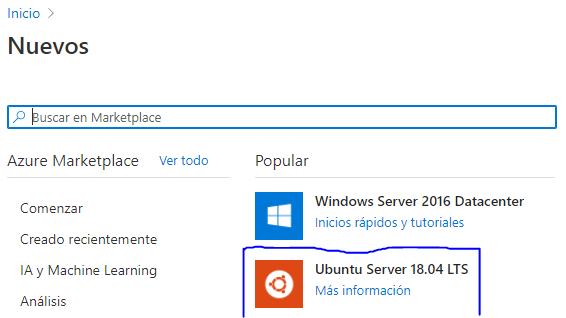
* 1. **-Azure**

Una vez hemos subido los archivos a Git, podemos comenzar con lo principal, **Azure.**

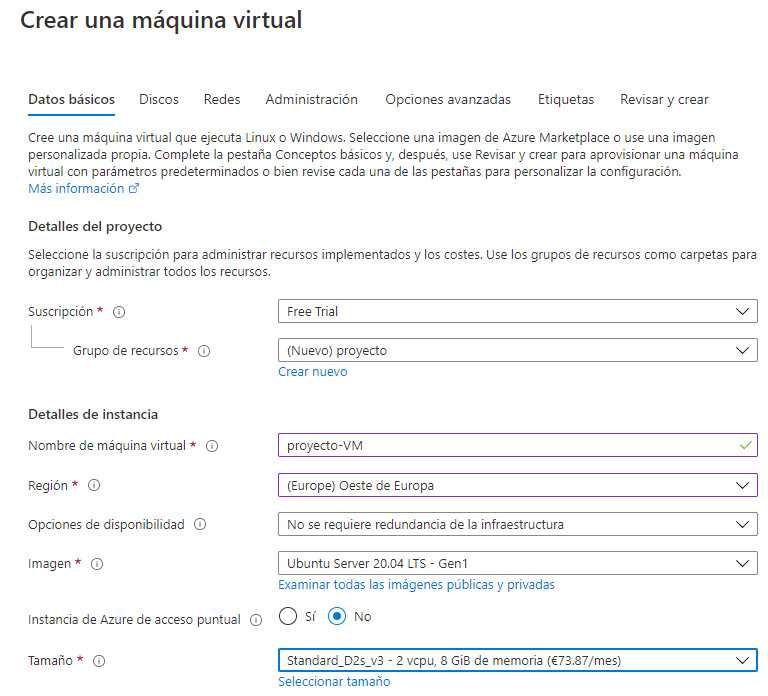
Para empezar, ya logueados en la cuenta, nos dirigimos al portal de azure y aquí en el inicio, desplegamos la barra lateral y ahí, **nuevo recurso**.



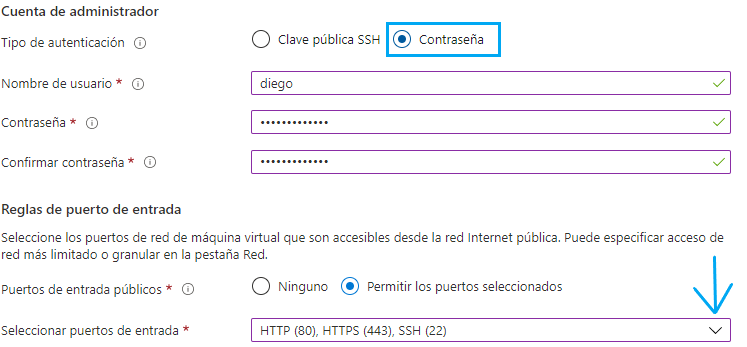
A continuación, elegiremos la imagen a partir de la cual lanzaremos nuestra instancia o máquina virtual, en este caso, elegiré Ubuntu Server ya que es un sistema potente, práctico y estoy familiarizado con él, tanto por las prácticas como durante el curso y además no ocupa ni consume muchos recursos, por lo que no necesitaremos de una máquina muy potente y fuera del presupuesto para poder con él sin ningún problema.



En este punto, comienza la configuración inicial de la instancia. En esta primera pantalla crearemos y seleccionaremos un **grupo de recursos**, daremos **nombre** a la máquina y seleccionaremos la **región** y la **potencia** más apropiadas.



Más abajo, un paso importante hacer la selección de **autenticación mediante contraseña**, ya que facilitará el proceso de conexión de a la máquina en remoto. También es importante crear un **usuario** adecuado y, en nuestro caso, abrir los **puertos** web y de ssh.



El siguiente paso, **redes**, aquí le daremos una dirección IP estática a nuestra máquina.

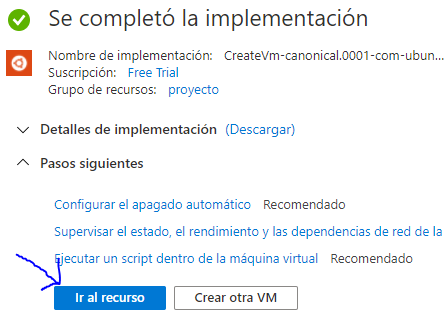


Los pasos inmediatamente siguientes, los dejaremos como viene por defecto.

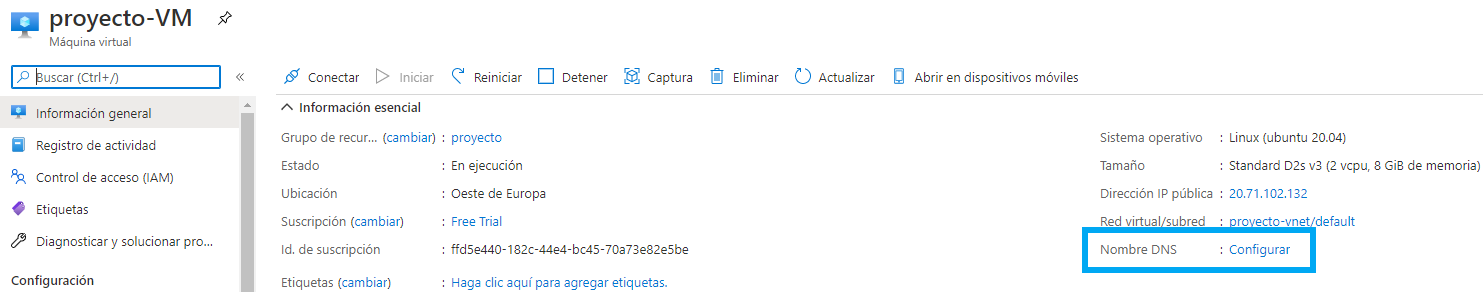
Para terminar, revisamos que todo esté correcto y nos disponemos a crear.

E:\Documentos\proyecto\crear.PNG

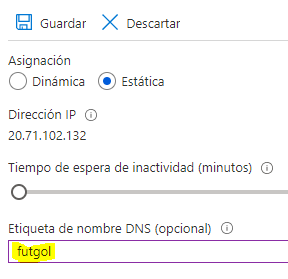
Una vez se complete la implementación, pinchamos en **ir al recurso** para ir a la máquina virtual.



Ahora vamos a asignarle un **dominio** a través de la opción que nos proporciona Azure para que la web sea mucho más accesible.



Una vez aquí, escribimos el **nombre** deseado y comprobamos que está disponible, después damos en **guardar**.

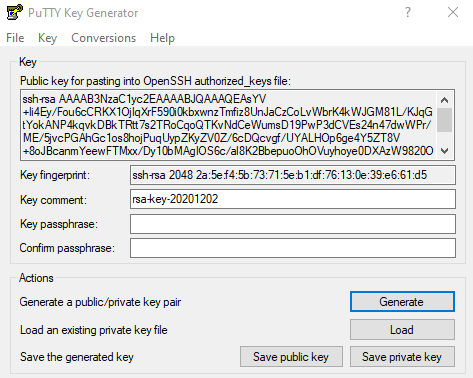
 

* 1. **-Conexión con la máquina: PuTTY y SSH.**

Ahora necesitamos conectarnos a la máquina para poder operar desde ella y desplegar la web. Comenzaremos copiando la **dirección IP** de la máquina virtual y descargando **PuTTY.**



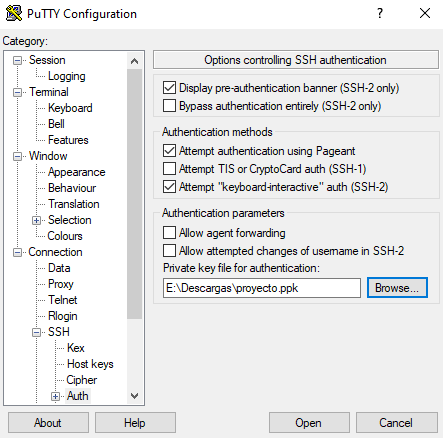
A continuación, buscaremos el programa **PuTTYgen**, un generador de claves públicas y privadas para maximizar la seguridad. Le damos a **Generate** y moveremos el ratón para generar una clave aleatoria.



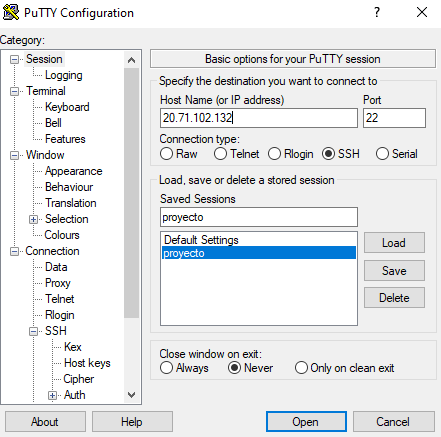
Y la guardamos. **Save private key.**



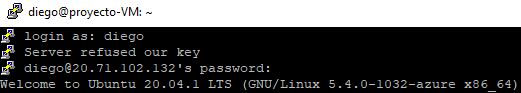
Una vez con la clave ya guardada abrimos **PuTTY** normal y nos dirigimos a **Auth** dentro del desplegable SSH. Aquí daré a **browse** y seleccionaremos nuestra clave privada.



Ahora vamos a **sesión** e introducimos la IP de la VM. Podemos guardar la sesión para mayor accesibilidad futuras veces.



Le damos a open y nos solicita el usuario con el que queremos iniciar sesión y su contraseña.



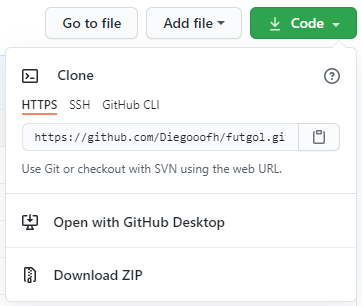
Una vez aquí, ya tenemos acceso total a la máquina a través de la consola.

* 1. **-Git en remoto.**

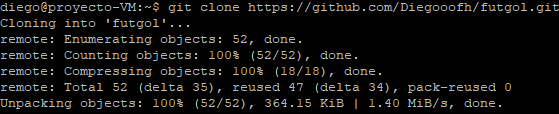
En este paso descargaremos en esta máquina todo el código y archivos necesarios para el despliegue de la web. Comenzaremos haciendo un update e instalando Git en caso de ser necesario, en nuestro caso ya viene instalado.



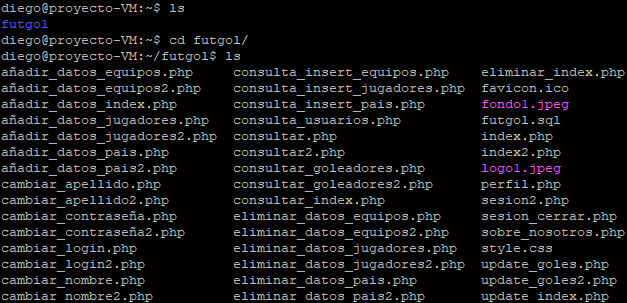
Ahora en GitHub, nos dirigimos a nuestro repositorio y copiamos la url del mismo.



Después procedemos a hacer un **git clone**.



Comprobamos que se ha copiado bien.



Y en mi caso, aunque no es necesario, moveré el script sql.

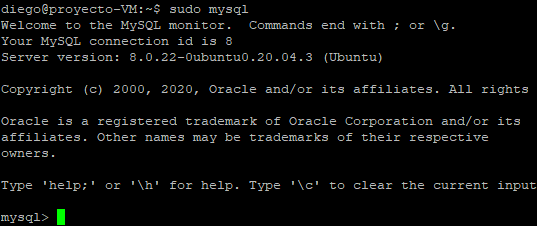


**4.5 -Base de datos: MySQL.**

Ahora instalaremos MySQL y desplegaremos la base de datos mediante el script sql.



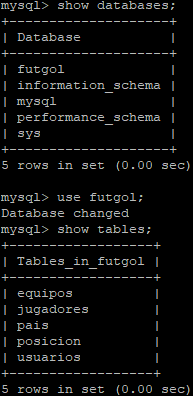
Lo abrimos.



Ahora ejecutamos el **script** para crear el usuario, la base de datos, las tablas y sus respectivos valores…



Y comprobamos que se ha creado correctamente.

**4.6 -Servidor web: Apache y PHP.**

Lo que necesitamos ahora es convertir nuestra máquina en un servidor web, para lo que necesitaremos la ayuda de algún programa como Apache o Nginx, entre otros. En mi caso utilizaré Apache ya que es con el que aprendimos en clase y ya me es familiar.



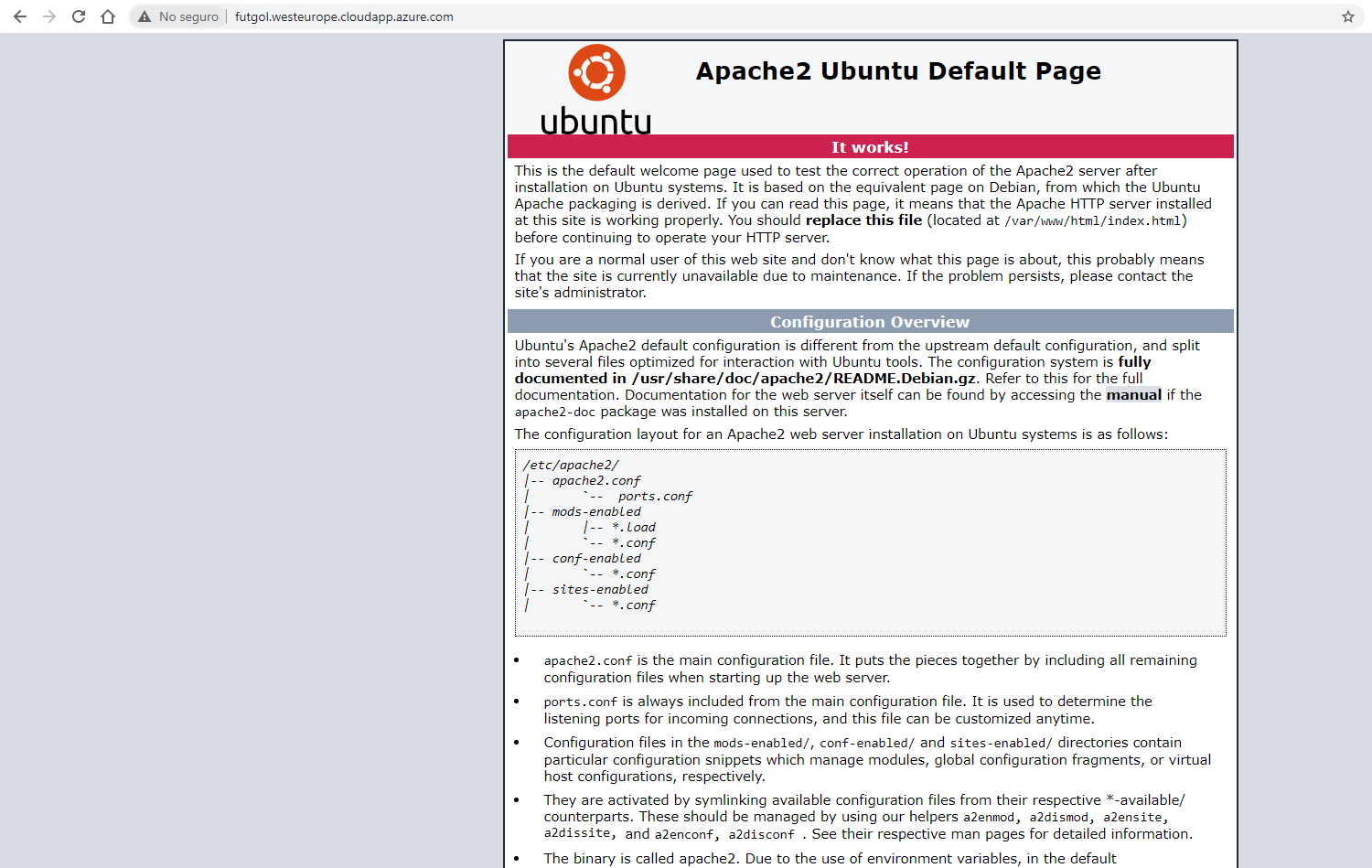
Ahora instalamos PHP







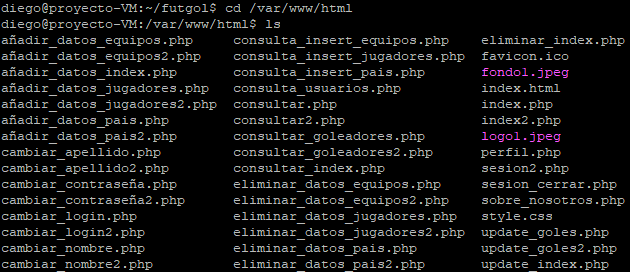
Comprobamos que se ha instalado dirigiéndonos al dominio. Debería aparecer una página por defecto de Apache como esta:



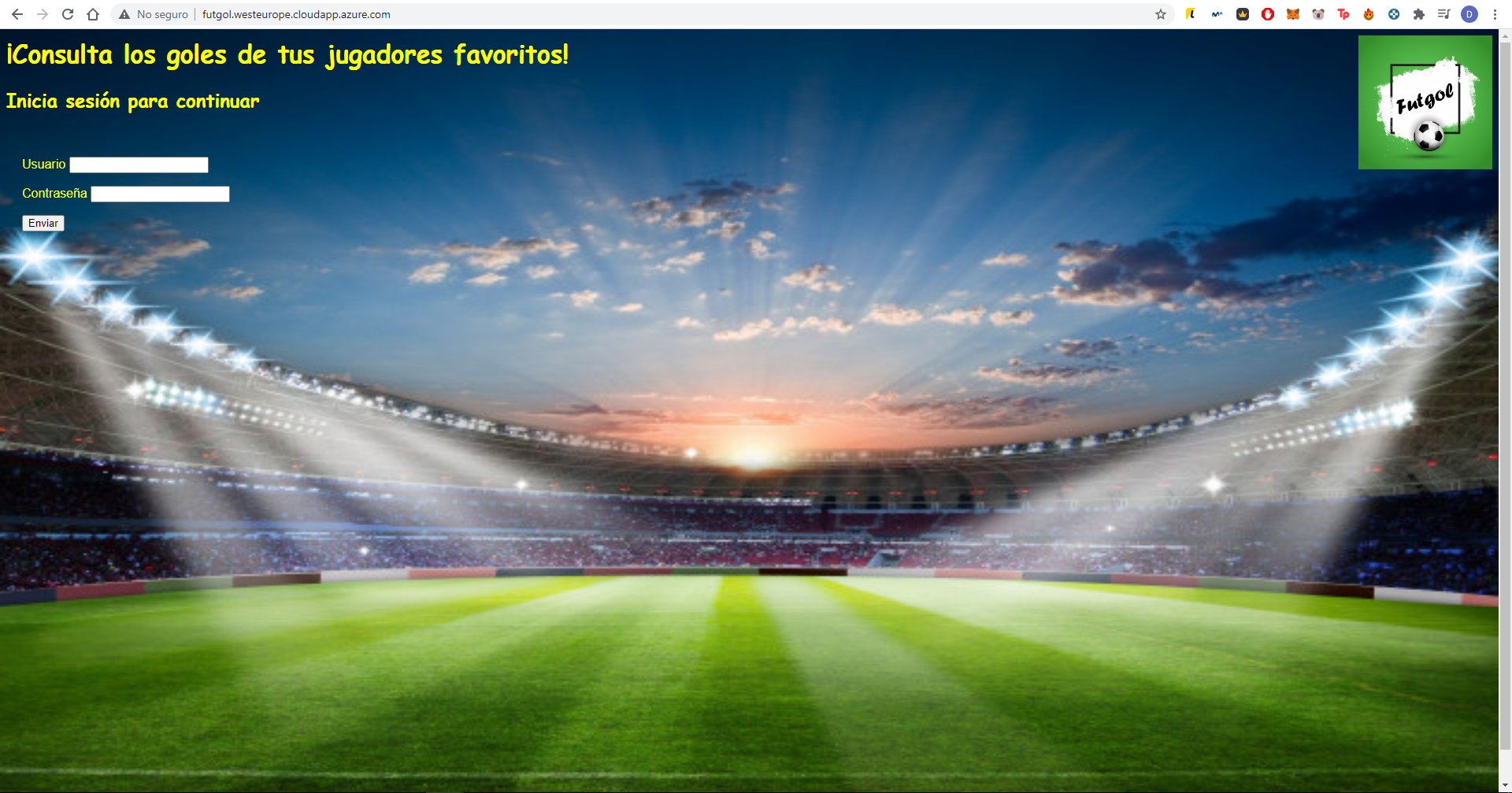
Vamos a la ruta donde deben estar los archivos de la web y borramos el index por defecto para reemplazarlo por el código de nuestra página.





Ahora al actualizar la página de nuestro dominio, debería aparecer nuestra página web.



# Conclusiones

# Un impedimento con el que me he encontrado ha sido la imposibilidad de conseguir desplegar el servidor web o el servidor MySQL a través de contenedores o *Dockers,* que en un inicio era la idea y me habría encantado ya que me parecen muy útiles, aunque en la práctica, el resultado final no se ha visto alterado ni tampoco era la situación idónea para usarlos, ya que la idea principal que tienen es de consumir pocos recursos al solo ejecutar unos pocos procesos del sistema y al estar en una máquina virtual alojada en la nube de Microsoft, no es algo importante ya que no consume nuestros recursos.

# Bibliografía

-YouTube: <https://www.youtube.com>

-OpenWebinars donde realicé un curso sobre Cloud Computing (AWS): <https://openwebinars.net/>

-GitHub: <https://github.com/>

-PuTTY: <https://www.putty.org>

-W3Schools: <https://www.w3schools.com>

-Microsoft Azure: <https://www.portal.azure.com>

-MySQL reference manual : <https://dev.mysql.com>