

Cloud computing

Proyecto fin de Ciclo Formativo
Administración de Sistemas Informáticos en Red

Diego Fernández Higuera
Curso 2020/2021



Índice

1 Información	3
2 Objetivos del proyecto	4
2.1 Análisis de Requisitos	4
3 Tecnologías	5
4 Procedimiento	6
4.1 -Comienzo: Git en el host	6
4.2 -Azure	7
4.3 -Conexión con la máquina: PuTTY y SSH	10
4.4 -Git en remoto	12
4.5 -Base de datos: MySQL	13
4.6 -Servidor web: Apache y PHP	14
4.7 -Comprobación de funcionamiento en conjunto	15
5 Conclusiones	17
6 Bibliografía	17

1 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.

2 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.

3 Tecnologías



1. Windows 10

Será el sistema operativo que utilizaré en el host.



2. 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.



3. PuTTY

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



4. SSH

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



5. SCP

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



6. MySQL

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



7. GitHub

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.



8. Git/GitBash

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



9. Azure

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



10. PHP

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



11. HTML

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



12. CSS

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

4 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.

4.1 -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.

```
Diego@Diego-Portatil MINGW64 /d/Programas/xampp/htdocs/ProyectoFutgol/Futgol (master)
$ ls
Futgol.zip                                consulta_insertEquipos.php               favicon.ico
añadir_datos_equipos.php                  consulta_insert_jugadores.php            fondo1.jpeg
añadir_datos_equipos2.php                 consulta_insert_pais.php                  futgol.sql
añadir_datos_index.php                    consulta_usuarios.php                     index.php
añadir_datos_jugadores.php                consultar.php                              index2.php
añadir_datos_jugadores2.php               consultar2.php                            logo1.jpeg
añadir_datos_pais.php                     consultar_goleadores.php                  perfil.php
añadir_datos_pais2.php                    consultar_goleadores2.php                 sesion2.php
cambiar_apellido.php                      consultar_index.php                       sesion_cerrar.php
cambiar_apellido2.php                     eliminar_datos_equipos.php               sobre_nosotros.php
cambiar_contraseña.php                    eliminar_datos_equipos2.php              style.css
cambiar_contraseña2.php                   eliminar_datos_jugadores.php             update_goles.php
cambiar_login.php                         eliminar_datos_jugadores2.php            update_goles2.php
cambiar_login2.php                        eliminar_datos_pais.php                  update_index.php
cambiar_nombre.php                        eliminar_datos_pais2.php
cambiar_nombre2.php                       eliminar_index.php
```

A continuación, iniciamos Git y seleccionamos los archivos.

```
Diego@Diego-Portatil MINGW64 /d/Programas/xampp/htdocs/ProyectoFutgol/Futgol (master)
$ git init
Reinitialized existing Git repository in D:/Programas/xampp/htdocs/ProyectoFutgol/Futgol/
Diego@Diego-Portatil MINGW64 /d/Programas/xampp/htdocs/ProyectoFutgol/Futgol (master)
$ git add .
```

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

```
Diego@Diego-Portatil MINGW64 /d/Programas/xampp/htdocs/ProyectoFutgol/Futgol (master)
$ git config --global user.email "diego00fh@gmail.com"

Diego@Diego-Portatil MINGW64 /d/Programas/xampp/htdocs/ProyectoFutgol/Futgol (master)
$ git config --global user.name "Diego"

Diego@Diego-Portatil MINGW64 /d/Programas/xampp/htdocs/ProyectoFutgol/Futgol (master)
$ git commit -m "futgol 1.0"
[master (root-commit) 3c89741] futgol 1.0
46 files changed, 2115 insertions(+)
```

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

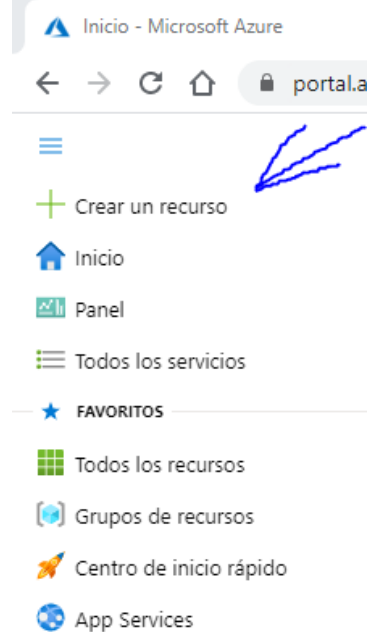
```
Diego@Diego-Portatil MINGW64 /d/Programas/xampp/htdocs/ProyectoFutgol/Futgol (master)
$ git remote add origin https://github.com/Diego00fh/futgol.git

Diego@Diego-Portatil MINGW64 /d/Programas/xampp/htdocs/ProyectoFutgol/Futgol (master)
$ git push origin master
Enumerating objects: 48, done.
Counting objects: 100% (48/48), done.
Delta compression using up to 4 threads
Compressing objects: 100% (48/48), done.
Writing objects: 100% (48/48), 160.24 KiB | 11.45 MiB/s, done.
Total 48 (delta 34), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (34/34), done.
To https://github.com/Diego00fh/futgol.git
 * [new branch]      master -> master
```


4.2 -Azure

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

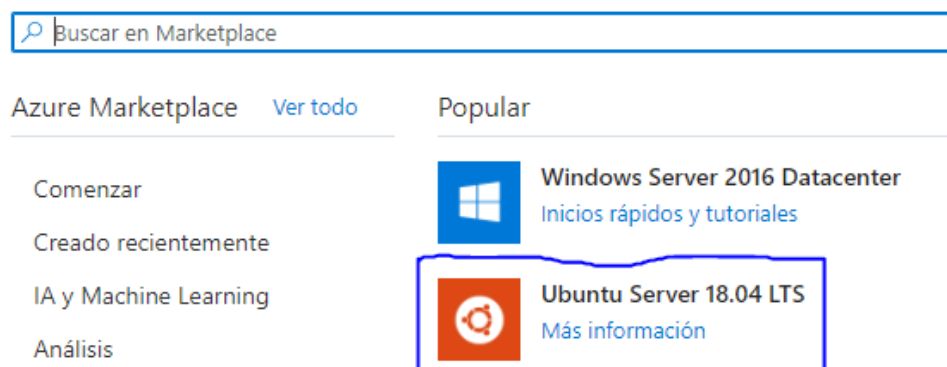
Para empezar, ya logueados en la cuenta, nos dirigimos al portal de azure y aquí en el inicio, desplegamos la barra lateral y ahí, **nuevo recurso**.



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.

[Inicio](#) >

Nuevos



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.

Crear una máquina virtual

Datos básicos Discos Redes Administración Opciones avanzadas Etiquetas Revisar y crear

Cree una máquina virtual que ejecuta Linux o Windows. Seleccione una imagen de Azure Marketplace o use una imagen personalizada propia. Complete la pestaña Conceptos básicos y, después, use Revisar y crear para aprovisionar una máquina virtual con parámetros predeterminados o bien revise cada una de las pestañas para personalizar la configuración. [Más información](#)

Detalles del proyecto

Seleccione la suscripción para administrar recursos implementados y los costes. Use los grupos de recursos como carpetas para organizar y administrar todos los recursos.

Suscripción * ⓘ

Grupo de recursos * ⓘ [Crear nuevo](#)

Detalles de instancia

Nombre de máquina virtual * ⓘ

Región * ⓘ

Opciones de disponibilidad ⓘ

Imagen * ⓘ [Examinar todas las imágenes públicas y privadas](#)

Instancia de Azure de acceso puntual ⓘ ☐ Sí ☒ No

Tamaño * ⓘ [Seleccionar tamaño](#)

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**.

Cuenta de administrador

Tipo de autenticación ⓘ ☐ Clave pública SSH ☒ Contraseña

Nombre de usuario * ⓘ

Contraseña * ⓘ

Confirmar contraseña * ⓘ

Reglas de puerto de entrada

Seleccione los puertos de red de máquina virtual que son accesibles desde la red Internet pública. Puede especificar acceso de red más limitado o granular en la pestaña Red.

Puertos de entrada públicos * ⓘ ☐ Ninguno ☒ Permitir los puertos seleccionados

Seleccionar puertos de entrada *

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

[Inicio](#) > [Nuevos](#) > [Marketplace](#) > [Ubuntu Server 20.04 LTS](#)

Crear una máquina virtual

Datos básicos Discos **Redes** Administración Opciones avanzadas Etiquetas Revisar y crear

Configure la tarjeta de interfaz de red (NIC) a fin de definir la conectividad de red para la máquina virtual. Puede controlar los puertos y la conectividad entrante y saliente con reglas de grupos de seguridad o bien aplicar una solución de equilibrio de carga ya existente. [Más información](#)

Interfaz de red

Al crear una máquina virtual, se crea una interfaz de red automáticamente.

Red virtual * ⓘ [Crear nuevo](#)

Subred * ⓘ

IP pública ⓘ [Crear nuevo](#)

Grupo de seguridad de red de NIC ⓘ ☐ Ninguno ☒ Básico ☐ Opciones avanzadas

Puertos de entrada públicos * ⓘ ☐ Ninguno ☒ Permitir los puertos seleccionados

Seleccionar puertos de entrada *

Crear dirección IP

Nombre *

SKU ⓘ ☒ Básico ☐ Estándar

Asignación ☐ Dinámica ☒ Estática

Los pasos inmediatamente siguientes, los dejaremos como viene por defecto. Para terminar, revisamos que todo esté correcto y nos disponemos a crear.

Crear

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

✓ Se completó la implementación

Nombre de implementación: CreateVm-canonical.0001-com-ubun...
 Suscripción: [Free Trial](#)
 Grupo de recursos: [proyecto](#)

▼ Detalles de implementación [\(Descargar\)](#)

^ Pasos siguientes

[Configurar el apagado automático](#) Recomendado

[Supervisar el estado, el rendimiento y las dependencias de red de la](#)

[ejecutar un script dentro de la máquina virtual](#) Recomendado

Ir al recurso

[Crear otra VM](#)

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

proyecto-VM
Máquina virtual

Buscar (Ctrl+/)

Conectar Iniciar Reiniciar Detener Captura Eliminar Actualizar Abrir en dispositivos móviles

Información general

Registro de actividad

Control de acceso (IAM)

Etiquetas

Diagnosticar y solucionar pro...

Configuración

Información esencial

Grupo de recur... (cambiar) : proyecto

Estado : En ejecución

Ubicación : Oeste de Europa

Suscripción (cambiar) : Free Trial

Id. de suscripción : ffd5e440-182c-44e4-bc45-70a73e82e5be

Etiquetas (cambiar) : Haga clic aquí para agregar etiquetas.

Sistema operativo : Linux (ubuntu 20.04)

Tamaño : Standard D2s v3 (2 vcpu, 8 GiB de memoria)

Dirección IP pública : 20.71.102.132

Red virtual/subred : proyecto-vnet/default

Nombre DNS : [Configurar](#)

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

[Guardar](#) [Descartar](#)

Asignación

☐ Dinámica ☒ Estática

Dirección IP ⓘ

20.71.102.132

Tiempo de espera de inactividad (minutos) ⓘ

Etiqueta de nombre DNS (opcional) ⓘ

futgol

.westeurope.cloudapp.azure.com

Por último, iremos a redes, agregar regla de puerto y pondremos el 3306, que es el puerto que utiliza MySQL.

Configuración



Redes

Agregar regla de puerto de entrada

Agregar regla de seguridad de entrada ✕
proyecto-VM-nsg

Básica

Origen * ⓘ
Any

Intervalos de puertos de origen * ⓘ
*

Destino * ⓘ
Any

Intervalos de puertos de destino * ⓘ
3306

Protocolo *
☒ Any ☐ TCP ☐ UDP ☐ ICMP

Acción *
☒ Permitir ☐ Denegar

Prioridad * ⓘ
350

Nombre *
Port_3306

Descripción
Permiso para el puerto de MySQL

4.3 -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**.

Dirección IP pública : 20.71.102.132

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.

PuTTY Key Generator ? ✕

File Key Conversions Help

Key

Public key for pasting into OpenSSH authorized_keys file:

```
ssh-rsa AAAAB3NzaC1yc2EAAAABJQAAQEA5YV
+ti4Eyy/Fou6cCRKX1OjIqXrF5900kxwnzTmfiz8UnJaCzCoLvWbrK4kWJGM81L/KJqG
tYokANP4kqvKDBkTRtt7s2TRoCqoQTKvNdCeWumsD19PwP3dCVes24n47dwWPr/
ME/5jvcPGAhGc1os8hojPuqUypZKyZV0Z/6cDQcvgf/UyALH0p6ge4Y5ZT8V
+8ouBcanmYeewFTMox/Dy10bMAgIOS6c/al8K2BbepuoOhOVuyhoye0DXAzW98200
```

Key fingerprint: ssh-rsa 2048 2a:5e:f4:5b:73:71:5e:b1:df:76:13:0e:39:e6:61:d5

Key comment: rsa-key-20201202

Key passphrase:

Confirm passphrase:

Actions

Generate a public/private key pair Generate

Load an existing private key file Load

Save the generated key Save public key Save private key

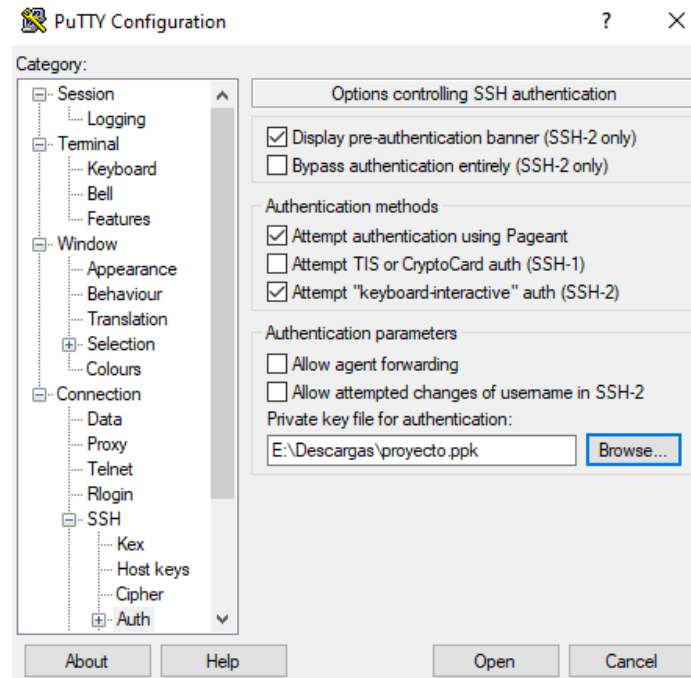
Y la guardamos. **Save private key.**

proyecto

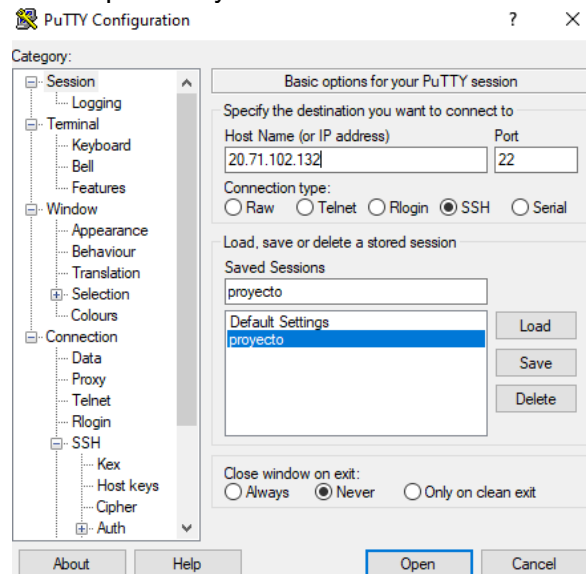
24/11/2020 17:50

PuTTY Private Key ...

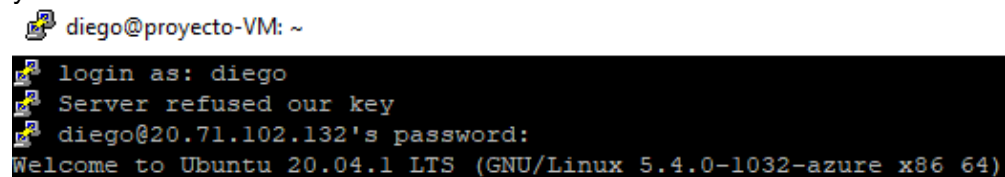
Una vez con la clave ya guardada abrimos **PuTTY** normal y nos dirigimos a **Auth** dentro del desplegable SSH. Aquí daré a **browse** y seleccionaremos nuestra clave privada.



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.



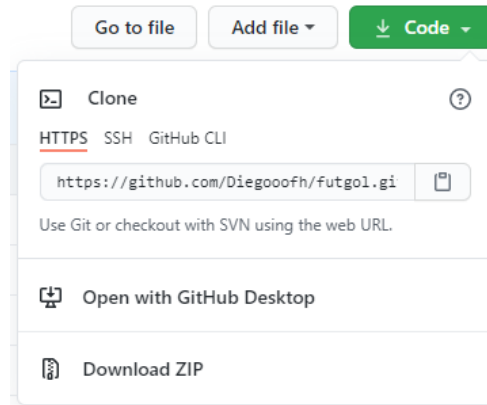
Una vez aquí, ya tenemos acceso total a la máquina a través de la consola.

4.4 -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.

```
diego@proyecto-VM:~$ git --version
git version 2.25.1
```

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



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

```
diego@proyecto-VM:~$ git clone https://github.com/Diegoooofh/futgol.git
Cloning into 'futgol'...
remote: Enumerating objects: 52, done.
remote: Counting objects: 100% (52/52), done.
remote: Compressing objects: 100% (18/18), done.
remote: Total 52 (delta 35), reused 47 (delta 34), pack-reused 0
Unpacking objects: 100% (52/52), 364.15 KiB | 1.40 MiB/s, done.
```

Comprobamos que se ha copiado bien.

```
diego@proyecto-VM:~$ ls
futgol
diego@proyecto-VM:~$ cd futgol/
diego@proyecto-VM:~/futgol$ ls
añadir_datos_equipos.php      consulta_insert_equipos.php    eliminar_index.php
añadir_datos_equipos2.php    consulta_insert_jugadores.php  favicon.ico
añadir_datos_index.php       consulta_insert_pais.php       fondol.jpeg
añadir_datos_jugadores.php   consulta_usuarios.php          futgol.sql
añadir_datos_jugadores2.php  consultar.php                  index.php
añadir_datos_pais.php        consultar2.php                 index2.php
añadir_datos_pais2.php       consultar_goleadores.php       logol.jpeg
cambiar_apellido.php         consultar_goleadores2.php      perfil.php
cambiar_apellido2.php        consultar_index.php           sesion2.php
cambiar_contraseña.php       eliminar_datos_equipos.php     sesion_cerrar.php
cambiar_contraseña2.php     eliminar_datos_equipos2.php    sobre_nosotros.php
cambiar_login.php           eliminar_datos_jugadores.php   style.css
cambiar_login2.php          eliminar_datos_jugadores2.php  update_goles.php
cambiar_nombre.php          eliminar_datos_pais.php        update_goles2.php
cambiar_nombre2.php         eliminar_datos_pais2.php      update_index.php
```

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

```
diego@proyecto-VM:~/futgol$ mv futgol.sql /home/diego/futgol.sql
diego@proyecto-VM:~/futgol$ cd
diego@proyecto-VM:~$ ls
futgol  futgol.sql
```

4.5 -Base de datos: MySQL.

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

```
diego@proyecto-VM:~$ sudo apt-get install mysql-server
```

Lo abrimos.

```
diego@proyecto-VM:~$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.22-0ubuntu0.20.04.3 (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.

mysql>
```

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

```
mysql> source futgol.sql
```

Y comprobamos que se ha creado correctamente.

```
mysql> show databases;
+-----+
| Database |
+-----+
| futgol   |
| information_schema |
| mysql    |
| performance_schema |
| sys      |
+-----+
5 rows in set (0.00 sec)

mysql> use futgol;
Database changed
mysql> show tables;
+-----+
| Tables_in_futgol |
+-----+
| equipos          |
| jugadores        |
| pais              |
| posicion          |
| usuarios          |
+-----+
5 rows in set (0.00 sec)

mysql> select * from pais;
+-----+
| nombre_pais |
+-----+
| Argentina   |
| Bélgica     |
| Brasil      |
| China       |
| Croacia     |
| Escocia     |
| Eslovenia   |
| España      |
| Francia     |
| Gales       |
| Inglaterra   |
| Japón       |
| Macedonia del Norte |
| México      |
| Montenegro   |
| Nigeria     |
| Noruega     |
| Portugal     |
| Serbia      |
| Uruguay     |
+-----+
20 rows in set (0.00 sec)
```

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.

```
diego@proyecto-VM:~$ sudo apt-get install apache2
```

Ahora instalamos PHP y algunas dependencias.

```
diego@proyecto-VM:~$ sudo add-apt-repository ppa:ondrej/php
```

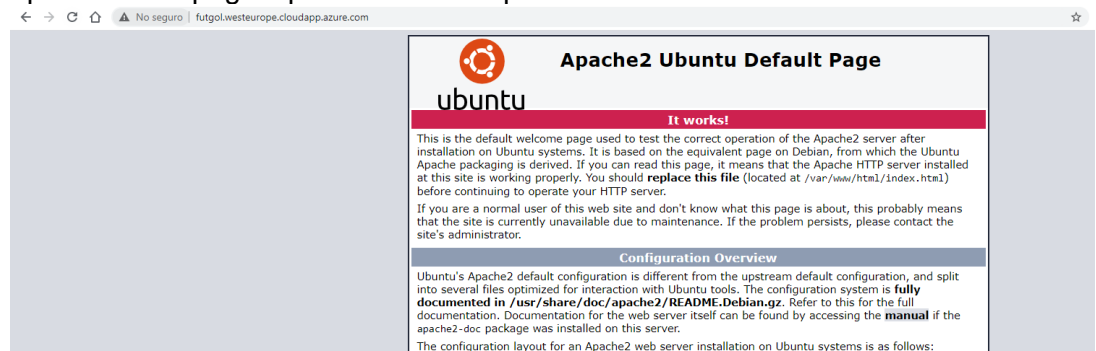
```
diego@proyecto-VM:~$ sudo apt-get install php7.2
```

```
diego@proyecto-VM:~$ sudo apt-get install php8.0-mysql
```

```
diego@proyecto-VM:~$ sudo update-alternatives --set php /usr/bin/php7.2
```

```
diego@proyecto-VM:~$ sudo systemctl restart apache2
```

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.

```
diego@proyecto-VM:/var/www/html$ sudo rm -f index.html
```

```
diego@proyecto-VM:~$ cd /var/www/html/
```

```
diego@proyecto-VM:/var/www/html$ ls
```

```
index.html
```

```
diego@proyecto-VM:/var/www/html$ cd
```

```
diego@proyecto-VM:~$ cd futgol
```

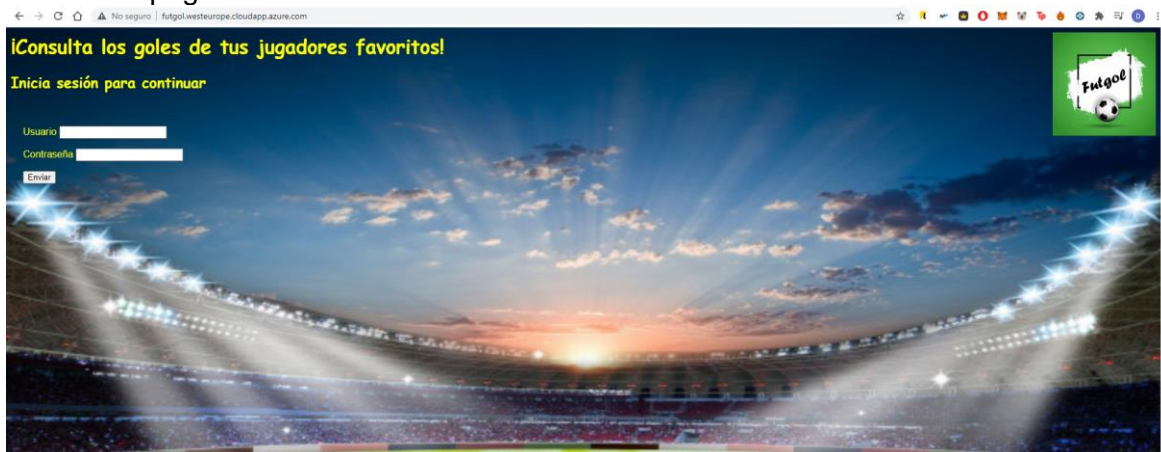
```
diego@proyecto-VM:~/futgol$ sudo mv * /var/www/html
```

```
diego@proyecto-VM:~/futgol$ cd /var/www/html
```

```
diego@proyecto-VM:/var/www/html$ ls
```

```
añadir_datos_equipos.php      consulta_insert_equipos.php      eliminar_index.php
añadir_datos_equipos2.php     consulta_insert_jugadores.php   favicon.ico
añadir_datos_index.php        consulta_insert_pais.php        fondol.jpeg
añadir_datos_jugadores.php    consulta_usuarios.php           index.html
añadir_datos_jugadores2.php   consultar.php                    index.php
añadir_datos_pais.php         consultar2.php                   index2.php
añadir_datos_pais2.php        consultar_goleadores.php        logol.jpeg
cambiar_apellido.php          consultar_goleadores2.php       perfil.php
cambiar_apellido2.php         consultar_index.php             sesion2.php
cambiar_contraseña.php        eliminar_datos_equipos.php      sesion_cerrar.php
cambiar_contraseña2.php       eliminar_datos_equipos2.php     sobre_nosotros.php
cambiar_login.php             eliminar_datos_jugadores.php    style.css
cambiar_login2.php            eliminar_datos_jugadores2.php   update_goles.php
cambiar_nombre.php            eliminar_datos_pais.php         update_goles2.php
cambiar_nombre2.php           eliminar_datos_pais2.php        update_index.php
```

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

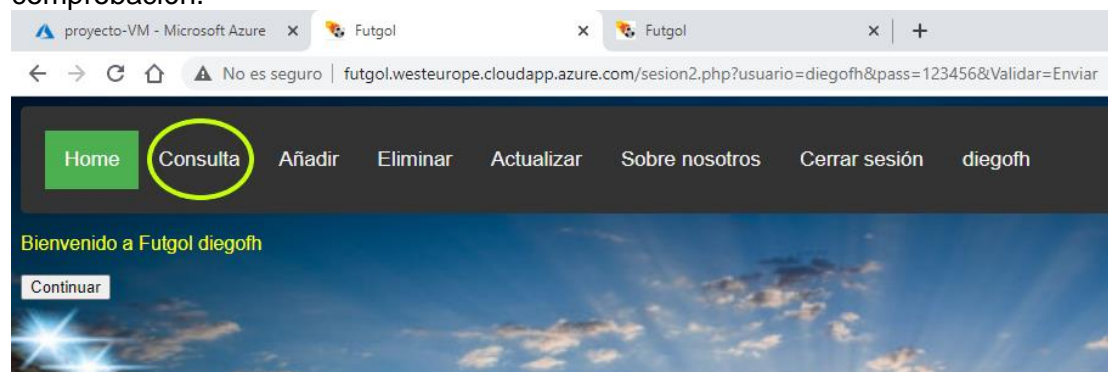


4.7 -Comprobación de funcionamiento en conjunto.

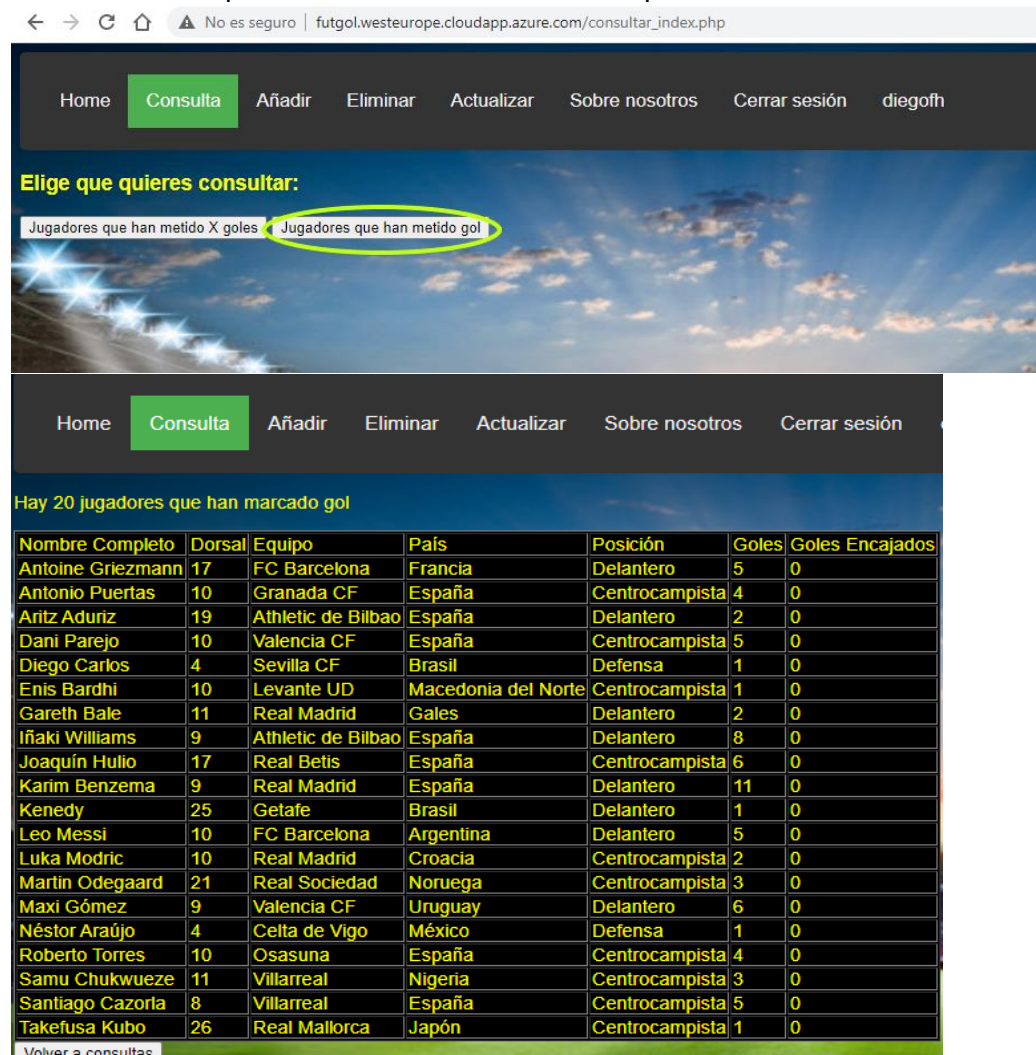
En este paso vamos a comprobar que la página web no sólo se muestra correctamente, sino que funciona también a la perfección. Para ello iniciaremos sesión para empezar.



Una vez iniciada la sesión, nos dispondremos a hacer una consulta de comprobación.



Haremos la comprobación con esta consulta simple.



Probamos también haciendo una inserción, por ejemplo.

Home Consulta **Añadir** Eliminar Actualizar So...

Introduce los siguientes datos para añadir a jugadores:

Introduce el nombre y apellido:

Introduce el dorsal:

Introduce el equipo:

Introduce el país:

Introduce la posición:

Introduce la cantidad de goles metidos:

Introduce la cantidad de goles encajados:

Y comprobamos que efectivamente se ha añadido el jugador.

Home Consulta **Añadir** Eliminar Actualizar Sobre nosotros Cerrar sesión

Nombre Completo	Dorsal	Equipo	País	Posición	Goles	Goles Encajados
Antoine Griezmann	17	FC Barcelona	Francia	Delantero	5	0
Antonio Puertas	10	Granada CF	España	Centrocampista	4	0
Aritz Aduriz	19	Athletic de Bilbao	España	Delantero	2	0
Dani Parejo	10	Valencia CF	España	Centrocampista	5	0
Diego Carlos	4	Sevilla CF	Brasil	Defensa	1	0
Diego Fernández	10	Atlético de Madrid	España	Delantero	15	0
Enis Bardhi	10	Levante UD	Macedonia del Norte	Centrocampista	1	0
Eric Cabaco	18	Levante UD	Uruguay	Defensa	0	0

Ahora comprobamos que esos mismos cambios se han efectuado localmente en la base de datos de la máquina.

```
mysql> select * from jugadores where nombre_completo='Diego Fernández';
+-----+-----+-----+-----+-----+-----+-----+
| nombre_completo | dorsal | equipo          | pais    | posicion | goles | goles_encajados |
+-----+-----+-----+-----+-----+-----+-----+
| Diego Fernández | 10     | Atlético de Madrid | España  | Delantero | 15    | 0                |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Como hemos visto, todo funciona perfectamente y está perfectamente sincronizado, además de que la página web es accesible por cualquier persona y dispositivo. Como vemos es accesible también desde cualquier teléfono, por lo que tenemos la página web al alcance de nuestro bolsillo.

vodafone ES 23:38 59 %

futgol.westeurope.cloudapp.azure.com

Home **Consulta** Añadir Eliminar Actualizar Sobre nosotros Cerrar sesión

Hay 21 jugadores que han marcado gol

Nombre Completo	Dorsal	Equipo	País	Posición	Goles	Goles Encajados
Antoine Griezmann	17	FC Barcelona	Francia	Delantero	5	0
Antonio Puertas	10	Granada CF	España	Centrocampista	4	0
Aritz Aduriz	19	Athletic de Bilbao	España	Delantero	2	0
Dani Parejo	10	Valencia CF	España	Centrocampista	5	0
Diego Carlos	4	Sevilla CF	Brasil	Defensa	1	0
Diego Fernández	10	Atlético de Madrid	España	Delantero	15	0
Enis Bardhi	10	Levante UD	Macedonia del Norte	Centrocampista	1	0
Eric Cabaco	18	Levante UD	Uruguay	Defensa	0	0
Isaki Williams	9	Athletic de Bilbao	España	Delantero	8	0
Joaquín Mullo	17	Real Betis	España	Centrocampista	6	0
Karim Benzema	9	Real Madrid	España	Delantero	11	0
Kenny	23	Colón	Brasil	Delantero	1	0
Leo Messi	10	FC Barcelona	Argentina	Delantero	3	0
Luka Modric	10	Real Madrid	Croacia	Centrocampista	2	0
Martin Ødegaard	21	Real Sociedad	Noruega	Centrocampista	3	0
Musa Sima	0	Valencia CF	Ucrania	Delantero	0	0
Néstor Araujo	4	Celta de Vigo	México	Defensa	1	0
Roberto Torres	10	Osasuna	España	Centrocampista	4	0
Samu Chukwueze	11	Villarreal	Nigeria	Centrocampista	3	0
Santiago Cuadrado	8	Villarreal	España	Centrocampista	3	0
Takafusa Kubo	26	Real Mallorca	Japón	Centrocampista	1	0

5 Conclusiones

Durante el proceso, me he encontrado con diversos problemas no muy graves que he sido capaz de solventar sin mayor problema a excepción de algún que otro problema y un principal impedimento.

Este 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 habría sido exactamente igual y no era tampoco 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.

6 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>

-VisualStudioCode: <https://code.visualstudio.com>