**HTTPS**

Autor: Reyner Badilla Arias.

En esta sección se mostrará una guía rápida de cómo crear un servidor utilizando Apache y como configurarlo para que tenga el certificado HTTPS (protocolo de Transferencia de Hiper-Texto) este es un protocolo que permite establecer una conexión segura entre el servidor y el cliente, que no puede ser interceptada por personas no autorizadas. En resumidas cuentas, es la versión segura del HTTP (Hyper Text Transfer Protocol).

Para obtener este protocolo se creará un certificado SSL (Secure Sockets Layer o capa de conexión segura) el cual es un estándar de seguridad global que permite la transferencia de datos cifrados entre un navegador y un servidor web, después de instalar correctamente el certificado en el servidor, el protocolo de aplicación (HTTP) cambiará a HTTPS, donde “S” significa seguridad.

Para realizar esta configuración, se usará el sistema operativo Ubuntu (distribución de Linux basada en Debian) en su versión 20.04.2.0.

1. Como primer paso se abrirá una terminal en la cual se realizarán todos los procedimientos y configuraciones necesarias a lo largo de esta guía, para abrir una terminal oprimimos CTR+ALT+T y se deberá mostrar una ventana como en la figura 3.1.

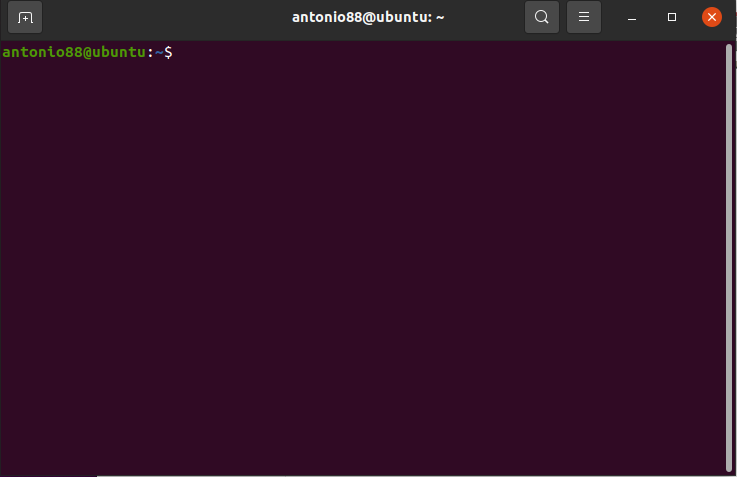


Figura 3.1. Terminal en Ubuntu (elaboración propia, 2021) []

1. Después se deberá crear una carpeta donde se guardarán todos los datos necesarios para poder ejecutar el servidor de manera correcta, para eso escribiremos el comando “**sudo mkdir pythonserver**”, seguido de esto se pedirá la contraseña, así que se procede a ingresarla.



Figura 3.2. Comando para crear carpeta (elaboración propia, 2021) []

1. Si la carpeta se creó de manera correcta, esta se deberá de guardar en “Home”, en la figura 3.3 se muestra el resultado.

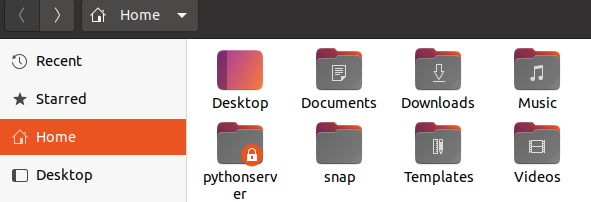


Figura 3.3. Carpeta pythonserver en directorio home (elaboración propia 2021)[]

1. Después de esto se procederá a ingresar al directorio anterior creado, ya que como se mencionó anteriormente, aquí es donde se crearán todos los archivos necesarios para crear el servidor, así que mediante el comando “***cd pythonserver***” se accederá a esta carpeta.



Figura 3.4. Accediendo al directorio pythonserver (elaboración propia, 2021) []

1. Para este paso, se procederá a hacer una prueba para corroborar que con un simple comando se podrá correr un servidor web (sin certificación https), para esto es tan simple como ejecutar el comando “***python3 -m http.server***”, se deberá de mostrar el resultado de la figura 3.5.



Figura 3.5. Ejecutando server desde el terminal (elaboración propia, 2021) [ ]

1. Eso indica que se esta ejecutando un servidor con la dirección ip local y con el puerto 8000 por defecto, si abrimos un navegador y escribimos “***http://localhost:8080***”, se deberá de mostrar algo similar a la figura 3.6 (no se mostrara una interfaz web porque no se tiene una programada, esto se realizará más adelante).

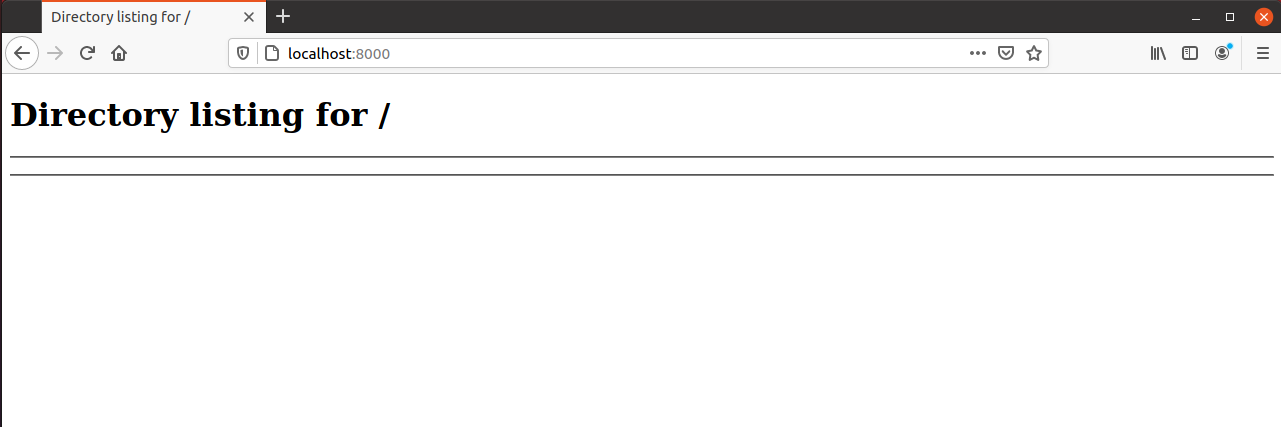


Figura 3.6. Servidor desde la web (elaboración propia,2021) [ ]

1. Para tener un mejor control sobre el servidor, se procederá a crear este mismo apoyados del lenguaje python3, para esto, se podrá ver la versión de este mismo con el comando “***python3 –version***”(python3 viene instalado por defecto en Ubuntu), luego se procederá a crear un archivo python en la carpeta pythonserver, para esto se ejecutara el comando “***sudo nano server.py”,*** una vez dentro del archivo python se deberá digitar código mostrado en la figura 3.7

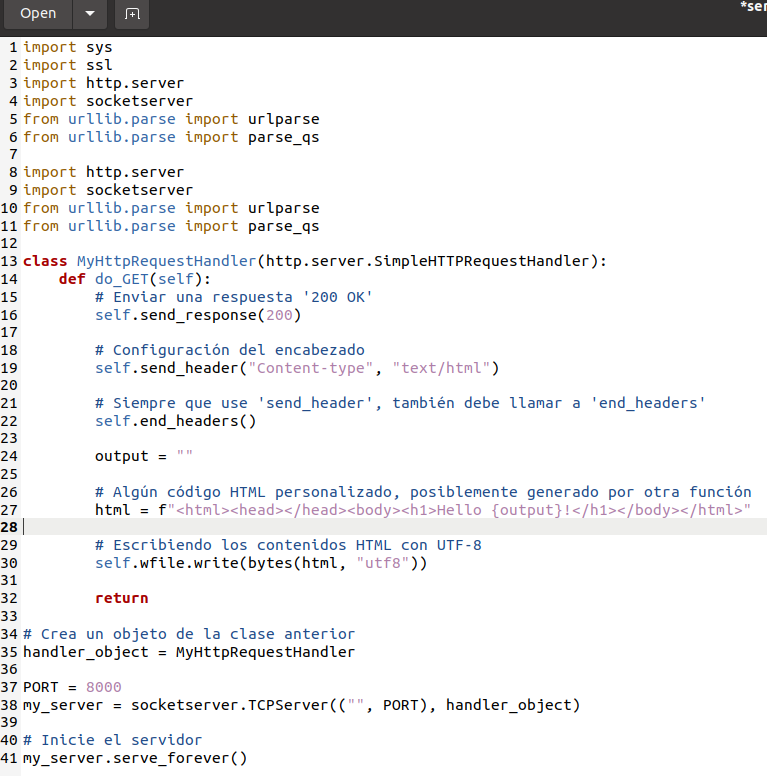


Figura 3.7. Código python a ingresar en server.py (elaboración propia,2021) [ ]

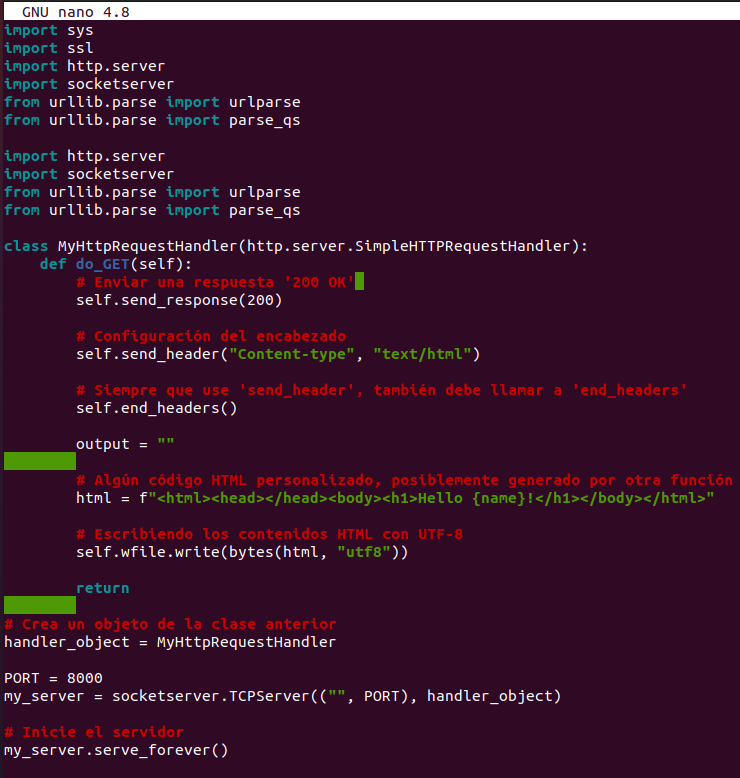


Figura 3.8. Codigo python ingresado en server.py (elaboración propia, 2021) [ ]

1. Una vez ingresado, se procede a guardar el código en el archivo, para esto presionamos CRT + O, luego ENTER y por último CTR + X para volver al terminal. Ahora procedemos a ejecutar el archivo para corroborar que todo funcione bien, esto mediante el comando “***sudo python3 server.py***”, le damos ENTER, nos pedirá la contraseña, la ingresamos y con esto el servidor comenzara a ejecutarse, para verificarlo accedemos al navegador e ingresamos “***http://localhost:8000***”, se deberá mostrar lo siguiente (aquí se usa el puerto 8000 porque se especificó desde código, pero se puede usar otro).

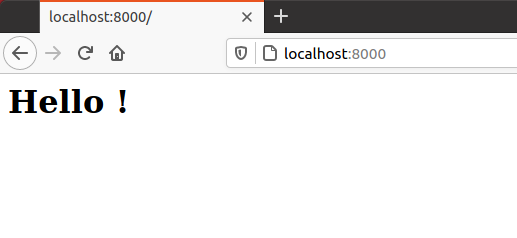
**

Figura 3.8. Servidor en la web ejecutado desde consola (elaboración propia, 2021) [ ]

1. La figura 3.8 muestra la conexión desde el navegador al servidor, sin embargo, este no cuenta con el certificado https, por esto para crearlo se deberá de ejecutar una serie de comandos, estos crearan unos archivos que darán este certificado, los cuales deberán ser agregados al código de python donde se encuentra el servidor, de manera que, a la hora de ser ejecutado, este habilite la certificación.
2. Ubicados en la línea de comandos, se procederá a ejecutar el comando “sudo openssl genrsa -out privkey.pem 2048”, el cual creará el archivo privkey.pem, esto indica que se genero una clave RSA(algoritmo de cifrado asimétrico) privada de 2048 bist.
3. Después usando la clave privada generada en el paso anterior, necesitaremos crear una solicitud de firma de certificado (CSR), por lo cual se ejecutará el comando “***sudo openssl req -new -key privkey.pem -out signreq.csr***” este comando nos generara el archivo signreq.csr, se pedirán datos como el país, provincia, ciudad, nombre de la organización y correo, estos datos se llenarán con la siguiente información (en los datos “extra” dar ENTER, no son necesarios), en la figura 3.9 se muestra el resultado

****

Figura 3.9. Servidor en la web ejecutado desde pyhton (elaboración propia, 2021) [ ]

1. Después se procederá a firmar la solicitud con una clave privada, para esto, se firmará la solicitud de certificado con la misma clave que se usó para crearla, para esto usaremos el comando “***sudo openssl req -newkey rsa:2048 -nodes -keyout privkey.pem -x509 -days 36500 -out certificate.pem***”, se pedirá información del país, provincia, ciudad, nombre de la organización y correo, para esto se repetirá lo mismo del paso 11.
2. Una vez realizado los pasos 10,11 y 12, en la carpeta del servidor se deberán de visualizar los archivos mostrados en la figura 3.10, los cuales corresponden a la clave RSA (***privkey.pem***), la solicitud de firma de certificado (***signreq.csr***) y el certificado (***certificate.pem).***

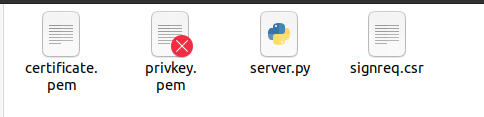


Figura 3.10. Archivos creados para el certificado https (elaboración propia, 2021) [ ]

1. Para validar el certificado en el servidor, se deberá abrir el archivo **server.py** y realizar cambios en el código mostrados en la figura 3.11 (estos cambios harán que al ejecutar el servidor se valide el certificado https).

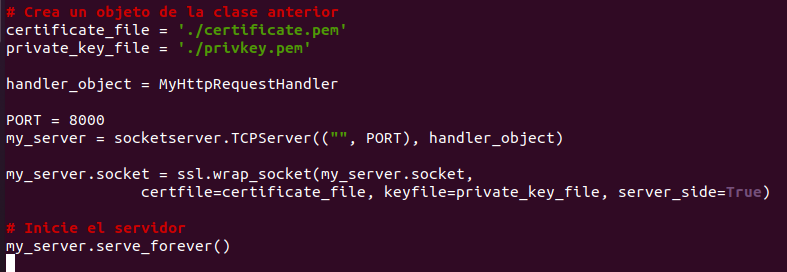


Figura 3.11. Cambios en archivo server.py (elaboración propia, 2021) [ ]

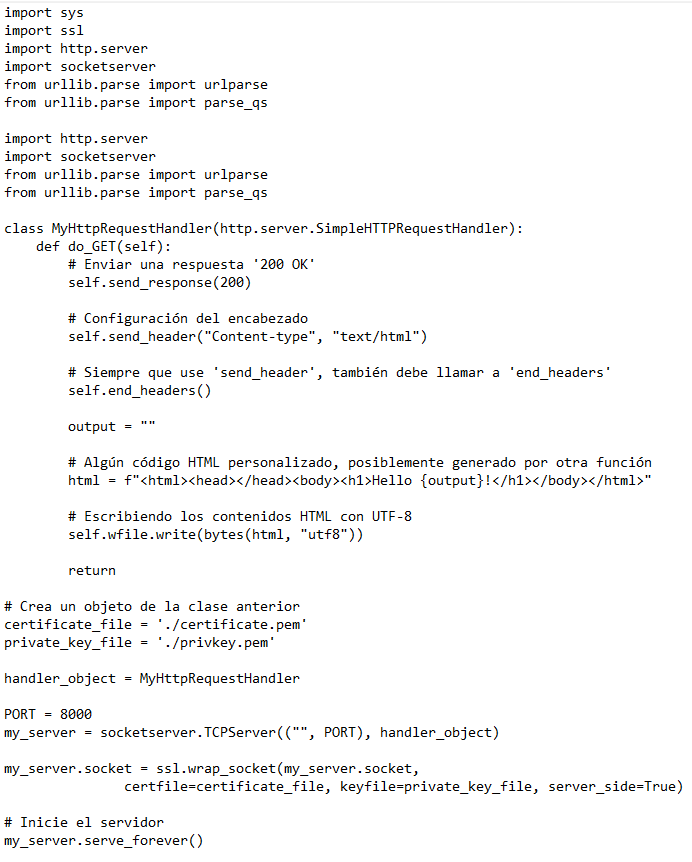
**

Figura 3.12. Cambios generales en el código de server.py (elaboración propia, 2021) [ ]

1. Una vez realizado el paso anterior, se procederá a verificar que el certificado funciona de manera correcta, para eso abriremos el navegador y se pondremos “***https://localhost:8000***”, en la figura 3.13 se muestra el resultado.

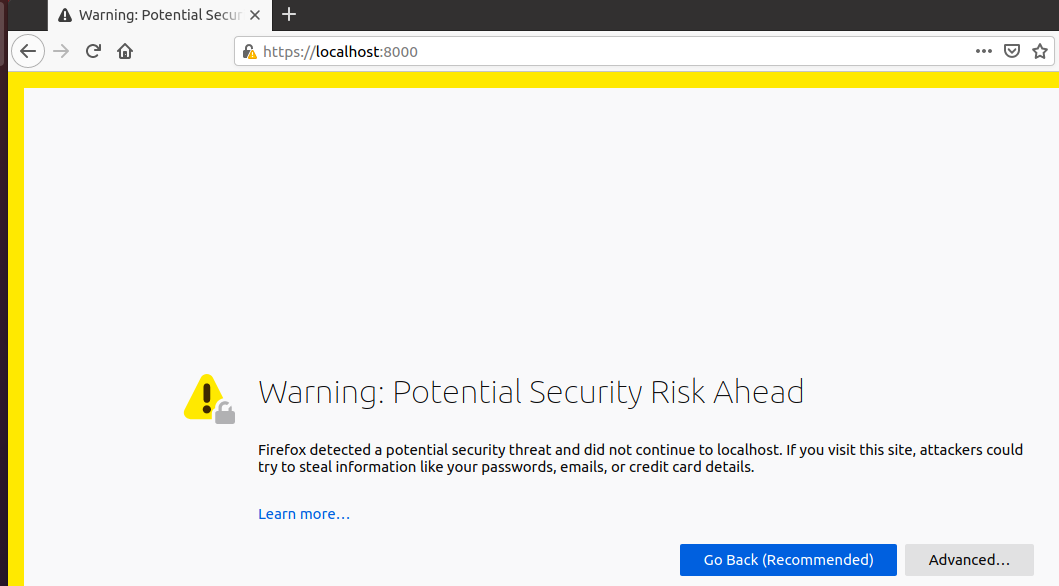


Figura 3.13. Accediendo a página con certificación https (elaboración propia, 2021) [ ]

1. Esto significa que aún no ha reconocido el certificado, por ende, se procede a presionar Advanced y luego presionar Accept the Risk and Continue.

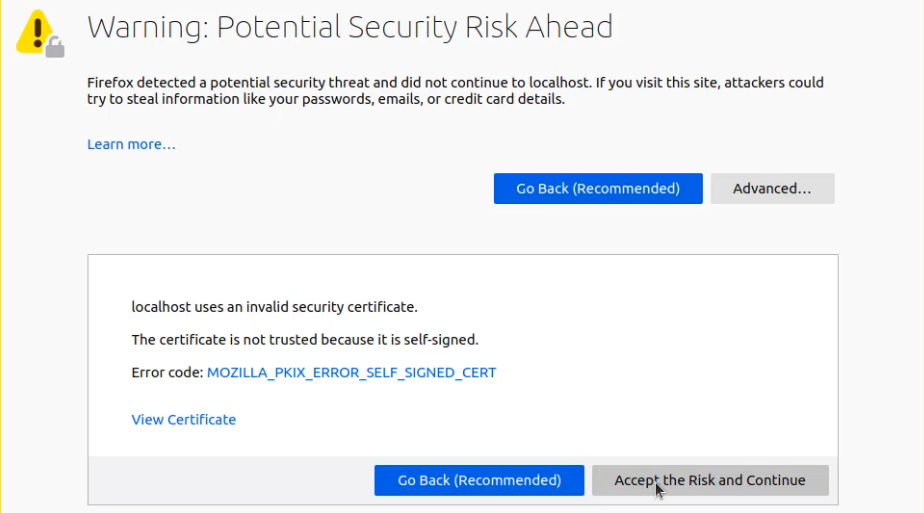


Figura 3.14. Aceptando el riesgo y continuar (elaboración propia, 2021) [ ]

1. Una vez realizado lo anterior ya se tendría en certificado digital funcionando, para verificar, se procede a dar en el candado situado a la par de https y después en la flecha situada a la par de Connection not secure (esto se debe a que el certificado es auto generado, como no hay una entidad externa que lo esté certificando, se detectara como conexión no segura, sin embargo, esto no afecta al correcto funcionamiento del certificado https).

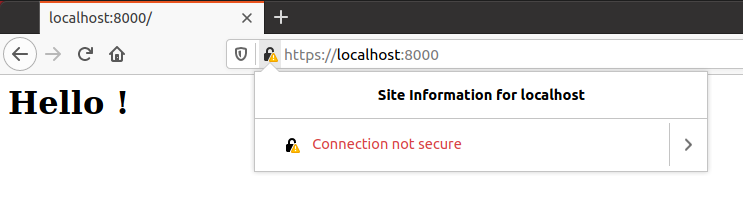
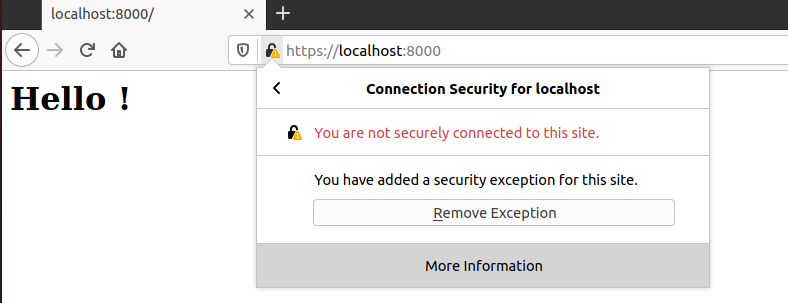


Figura 3.15. Página usando certificado https (elaboración propia, 2021) [ ]

1. Después se procederá a dar click en “more information”.

 Figura 3.16. Verificando certificado https (elaboración propia, 2021) [ ]

1. Una vez ahí se podrá ver que el certificado es totalmente funcional, en este caso se podrá ver que es verificado por UNA (nombre asignado en la configuración previamente hecha).

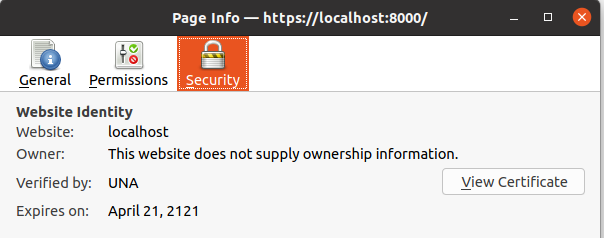


Figura 3.17. Pagina verificada por UNA (elaboración propia, 2021) [ ]

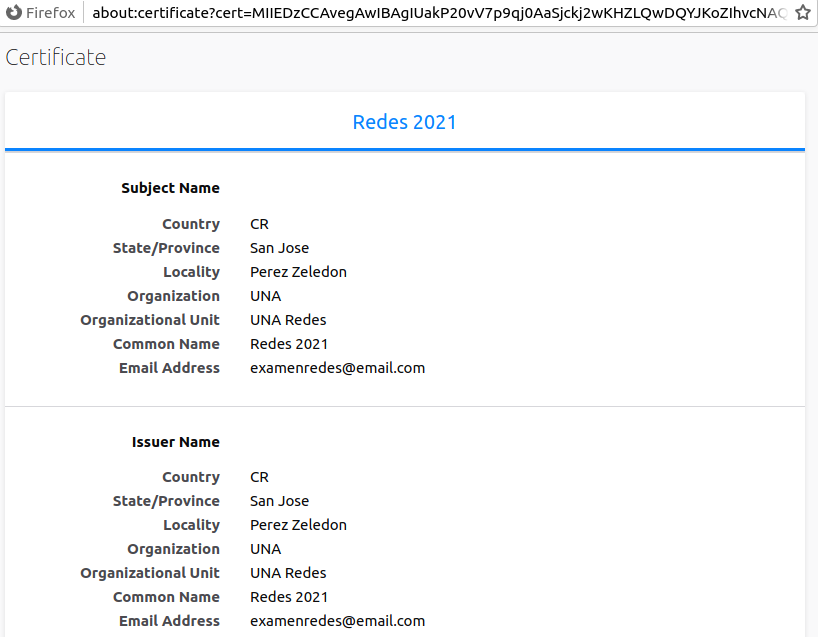


Figura 3.18. Certificado a detalle (elaboración propia, 2021) [ ]

1. Y así de sencillo se terminó de realizar la instalación de un servidor mediante python y la configuración de este mismo para la obtención de un certificado https.

Servidor HTTPS con MySQL

Para realizar estas pruebas, se tomarán los datos de una base de datos en MySQL, la cual está ubicada en la máquina virtual del Kevin Mora, mediante python, se realizará una consulta y se traerán los datos almacenados en dicha base de datos y se agregarán al servidor HTTPS elaborado por Reyner Badilla.

Para esto se realizarán los siguientes pasos:

1. Primero se verificarán los datos a mostrar en el servidor HTTPS, las cuales se adjuntan en las figuras 6 y 6

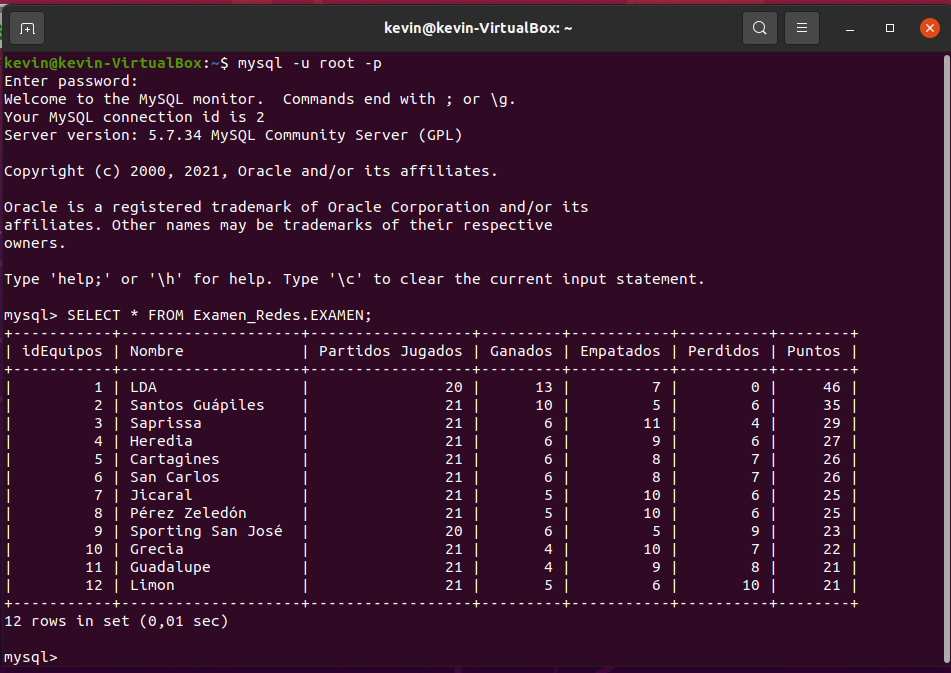


Figura 6. . Datos almacenados en la base de datos MySQL de Kevin Mora desde consola (elaboración propia, 2021) [ ]

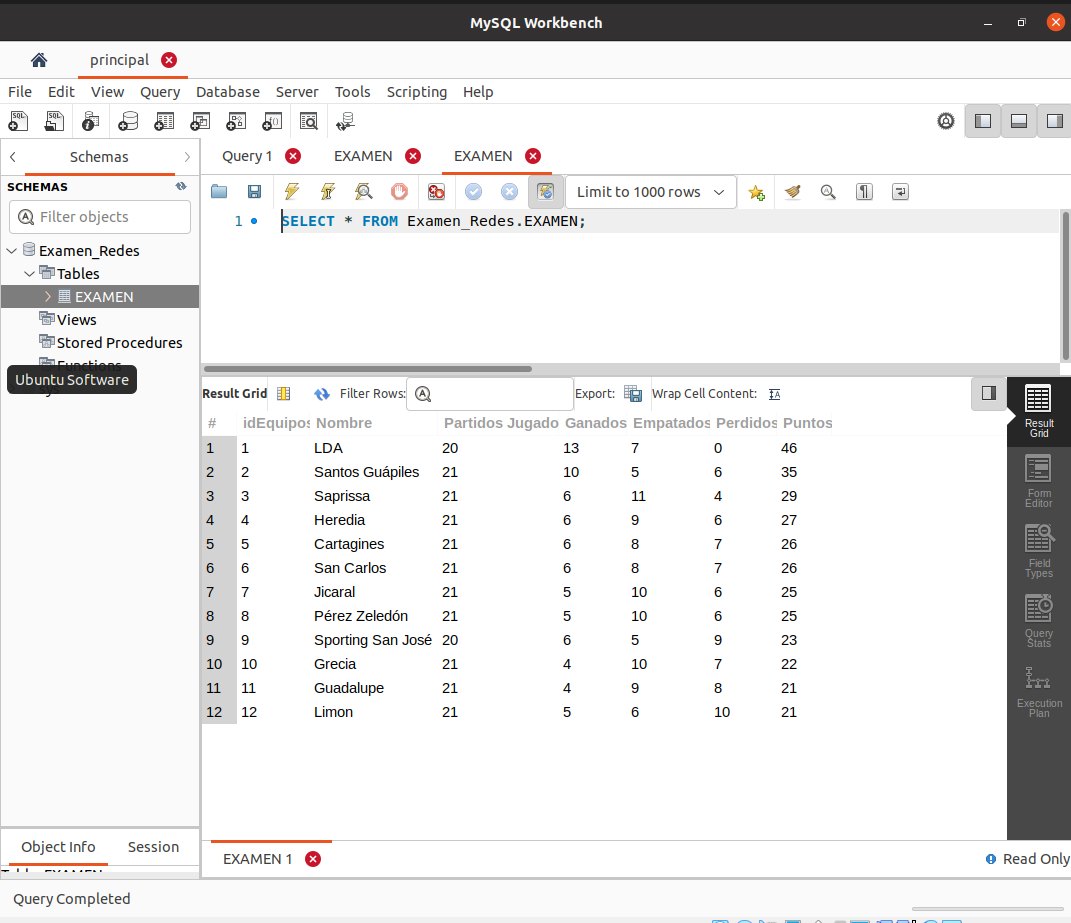


Figura 6. . Datos almacenados en la base de datos MySQL de Kevin Mora, vistos desde Workbench (elaboración propia, 2021) [ ]

1. Después se procederá a instalar el pymysql es cual nos permitirá realizar las consultas necesarias en base de datos, para instalarlo se usará el comando “***python3 -m pip install PyMySQL***”.

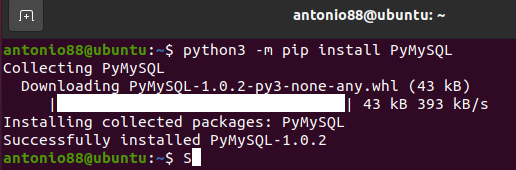


Figura 6. . Instalación de PyMySQL (elaboración propia, 2021) [ ]

1. Después de esto, se procederá a agregar el código (en python) para poder realizar la consulta a base de datos y cargar los datos necesarios, este código se muestra en la figura 6.



Figura 6. . Codigo python para conexión con base de datos (elaboración propia, 2021) [ ]

1. Ahora se procederá a incorporar el código de la figura 6 en el servidor de python, por lo tanto, con algunas modificaciones extra, el código de server.py quedará como el mostrado en la figura 6.

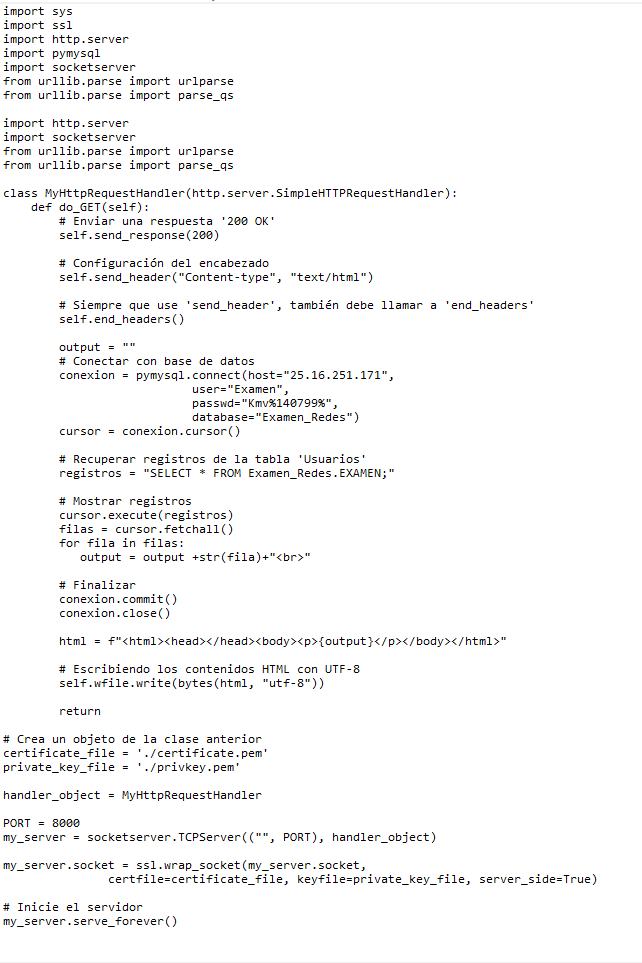


Figura 6. . Código final del servidor (elaboración propia, 2021) [ ]

1. Una vez realizados los cambios en server.py, se procederá a verificar que en dicho servidor se cargan los datos desde la base de datos en MySQL, para esto se ejecutara el server mediante el código “***sudo python3 server.py”.***



Figura 6. . Comando para ejecutar servidor (elaboración propia, 2021) [ ]

1. Una vez ejecutado el servidor, se procederá a acceder al navegador e ingresar a “***https://localhost:8000***”, el resultado es el mostrado en la figura 6.

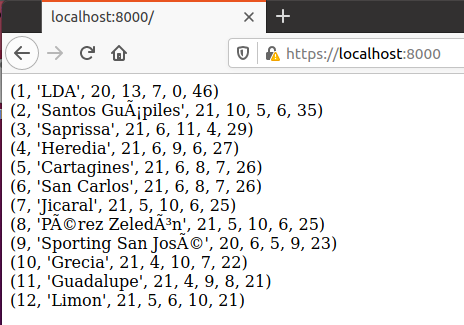


Figura 6. . Datos de MySQL en el servidor (elaboración propia, 2021) [ ]

1. Como se pudo observar, se obtuvieron los datos de la base de datos de Kevin Mora en el servidor HTTPS de Reyner Badilla, por lo tanto, se demuestra la funcionalidad de ambos servicios.