

2020



Nombre: David Fajardo Barral

Fecha de presentación: 19/06/2020 a las 9:30

PÁGINA WEB CON SERVIDOR WEB PROPIO

ÍNDICE

Resumen y objetivos	3
Viabilidad y estudios de mercado	4
Análisis de requerimientos técnicos y humanos	4
Diseño de la solución adoptada	5
Presupuesto.....	5
Instalación y configuración de Ubuntu Server 20.04 LTS.....	5
Base de datos	10
Esquema de la BD según el software MySQL Workbench	11
Esquema de la BD según el software Dia	12
Código de la BD.....	12
PHPMyAdmin: importar BD y crear un nuevo usuario para la web	13
Página web	15
Esquema	15
Evitando que se pueda visualizar la estructura	16
Parte lógica	17
Conexión de la web con la BD	17
Evitar inyección SQL	18
Inicio de sesión seguro y cookie para guardarla	18
Protección anti-intrusos no autenticados para la zona privada.....	19
Zona privada: vista general	20
Gestión de usuarios	27
Web <i>responsive</i>	29
Menú desplegable para la versión móvil.....	31
Mejoras de última hora: aprendiendo JavaScript sobre la marcha	33
Salida del servidor web al exterior	34
Dificultades encontradas	37
Implementación del prototipo (si es el caso)	37
Conclusiones	37
Bibliografía, referencias y recursos utilizados	37

RESUMEN Y OBJETIVOS

La idea de este proyecto es realizar una página web que sirva para cualquier librería, de forma que enseñe el stock de libros disponibles. En la zona pública habrá un menú formado por 7 categorías para distintos tipos de libros, además de una barra de búsqueda siempre presente y un botón para dirigirse a la página de *login* y acceder a la zona privada. Además, la portada tendrá cosas vistosas como una fila de 5 libros (aleatorios cada vez que cargue la web) con efecto de zoom al dejar el ratón sobre ellos, una imagen promocionando la supuesta tarjeta de puntos...

A su vez habrá una zona privada solo accesible para los trabajadores de la empresa en cuestión, que tendrá un CRUD para mostrar todos los libros de la base de datos y botones para añadir, editar o eliminar los mismos, ordenados por páginas. Por defecto mostrando 3 en cada una, pero habrá un cuadro de texto para configurar la paginación con el número que se desee. También estará el acceso a la zona de eventos donde se pondrá el título y el link al evento previamente anunciado en Facebook. Ha decidido hacerse así en lugar de algo tipo blog porque, tras consultar fuentes del mercado laboral, la principal vía de propagación de información son las RRSS, por lo que poniendo un `<iframe>` con la noticia en RRSS, se propagarán más y de forma más eficaz.

La base de datos (en adelante "BD") incluirá una tabla para usuarios de la empresa con una columna llamada "rango", que permitirá separar a los trabajadores de la empresa en "administradores" y "empleados". De esta forma, todas las personas con una cuenta de usuario para el acceso empresarial podrán acceder a la zona privada, pero solo los administradores podrán acceder a la gestión de los usuarios, pudiendo añadir, editar o eliminarlos. Esta separación y demás detalles sobre el funcionamiento de la web se explicarán en detalle más adelante.

También se montará y configurará un equipo como servidor web para almacenar la página, teniendo así asegurada siempre su disponibilidad. En este caso concreto, por no llevar aún el proyecto a la vida real, se usará una MV en el software Oracle VirtualBox con Ubuntu Server 20.04 LTS como S.O., sin interfaz de usuario para mayor rendimiento, gestionándose remotamente con el software MobaXterm debido a su gran facilidad de transferencia de archivos, por supuesto siempre por SSH.

VIABILIDAD Y ESTUDIOS DE MERCADO

Hoy en día cualquier empresa o negocio, por pequeño que sea, debe hacer uso de Internet para promocionarse. Además, con una página web, una empresa podría llegar a mucha más gente a pesar de, si se diera el caso, estar localizada en una zona poco fructífera para su negocio. La expansión por la red es fundamental. Y con el uso, cada vez más y más creciente de los teléfonos móviles para todo tipo de operaciones al igual que los ordenadores, es muy común consultar la web de una tienda (y/o su app, pero no forma parte del contenido de este ciclo formativo) para ver su stock, consultar qué más material del interés del cliente puede haber, localización del negocio, horario...

ANÁLISIS DE REQUERIMIENTOS TÉCNICOS Y HUMANOS

El gasto menor sería el del servidor (presupuesto adjunto más adelante), pero al no ser necesarias muchas prestaciones, éste será ínfimo (incluso podría serlo más de no haberse incluido un SAI, pero se ha hecho por mera profesionalidad).

Tras haber consultado numerosas fuentes y contrastar precios de profesionales del mercado del desarrollo de sitios web, puede afirmarse que un diseñador web cobraría unos 30€ la hora (haciendo precio si se trata de una empresa pequeña, y es el caso), además se ha de tener en cuenta que no se trata de la plantilla de un CMS modificada, sino una web hecha a mano desde cero. Este trabajo ha llevado 2 meses (40 días laborables), así que... $30\text{€/hora} * 8 \text{ horas/día} * 40 \text{ días} = 9.600 \text{ €}$

Preparar el servidor es simple y en principio rápido, pero la creación de una web conlleva, entre otros, los siguientes puntos a tener en cuenta:

- Esquema del sitio y de la base de datos
- Diseño de la web (cabecera, portada, demás páginas que solo se diferenciarán en el cuerpo de la misma, zona privada con estilos mínimos para mayor rendimiento...)
- Realizar estilos CSS para hacer que la web sea *responsive* y por lo tanto cómoda de visitar tanto en PCs y televisiones como en dispositivos más pequeños como smartphones, tablets y iPads.

Una vez finalizada, testearla a fondo en distintos navegadores de PC y móviles para comprobar su correcto funcionamiento en todas las plataformas.

DISEÑO DE LA SOLUCIÓN ADOPTADA

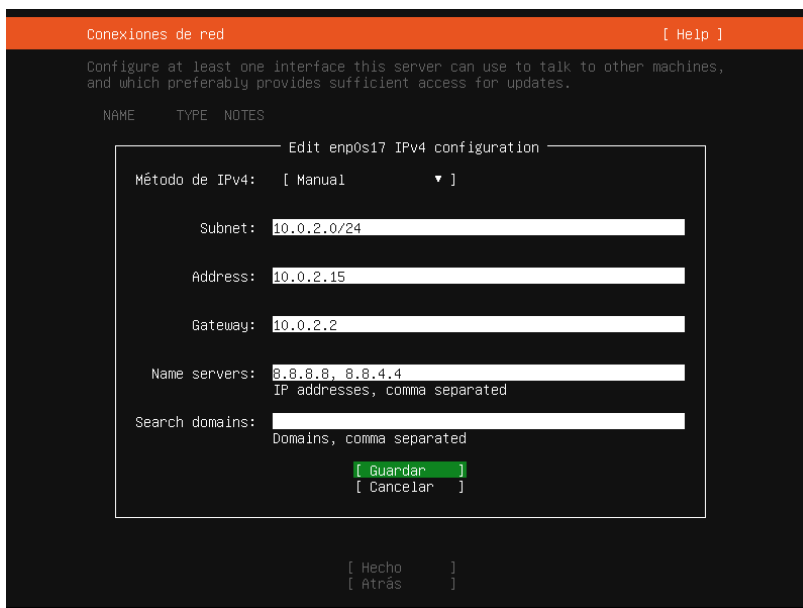
PRESUPUESTO

Equipo	Marca	Modelo	Precio
Equipo	HP	ProLiant MicroServer Gen10	361,00 €
Monitor	Samsung	S22F350FHU	85,99 €
SDD	Samsung	860 EVO Basic	89,09 €
Teclado inalámbrico	Cherry	KW 2000	17,75 €
SAI	L-Link	LL7716 Interactive	135,99 €
Coste humano	<i>30€/hora*8horas/día*40 días</i>		9.600,00 €
<u>TOTAL</u>			10.289,82 €

INSTALACIÓN Y CONFIGURACIÓN DE UBUNTU SERVER 20.04 LTS

Primero de todo elegiremos el idioma **español** tanto para el sistema como para el teclado.

A continuación, deberemos configurar la red. Por defecto se pondría en modo automático usando DHCP, así que nos toca pasarla a manual. Introduciremos IP de la red junto con una barra lateral para indicar el número de bits a 1 que tendrá la máscara, la IP del equipo, la puerta de enlace (IP del router) y los servidores DNS separados por comas.



Ahora llega el momento de elegir el particionamiento del disco. Si fuera un equipo cliente sería conveniente elegir nosotros mismos la cantidad de espacio dedicado al sistema (/), a datos (/home) y a la memoria virtual (área de intercambio, swap), pero no es el caso. Esto es un servidor y se usará única y exclusivamente para una sola tarea: actuar como

servidor web, de manera que dejaremos la primera opción de **Use an entire disk** y continuaremos-

Ahora introducimos nuestro nombre, nombre de servidor, nombre de usuario y contraseña:

Luego la instalación nos ofrece instalar el paquete **OpenSSH Server**. Marcamos la casilla y continuamos, así nos ahorraremos instalarlo a posteriori. Por supuesto, será el protocolo seguro que usaremos para acceder remotamente al servidor y transferir los archivos de nuestra web, configurar el gestor de bases de datos... y todo tipo de

Configuración de perfil [Help]

Proporcione el nombre de usuario y la contraseña que utilizará para acceder al sistema. Puede configurar el acceso SSH en la pantalla siguiente, pero aun se necesita una contraseña para sudo.

Your name: David Fajardo Barral

Your server's name: asir2dfb
The name it uses when it talks to other computers.

Pick a username: david

Choose a password: *****

Confirm your password: *****

[Hecho]

operaciones gracias al software MobaXterm. En la siguiente pantalla se ofrecen más paquetes. No marcamos ninguno y continuamos; comenzará la instalación. Terminará después de unos cuantos minutos:

Se ha completado la instalación. [Help]

```

— He finalizado la instalación.
running '/snap/bin/subiquity.subiquity-configure-apt
/snap/subiquity/1772/usr/bin/python3 true'
curtin command apt-config
curtin command in-target
running 'curtin curthooks'
curtin command curthooks
configuring apt
configuring apt
installing missing packages
configuring iscsi service
configuring raid (mdadm) service
installing kernel
setting up swap
apply networking config
writing etc/fstab
configuring multipath
updating packages on target system
configuring pollinate user-agent on target
updating in-tranfs configuration
finalizing installation
running 'curtin hook'
curtin command hook
executing late commands
final system configuration
configuring cloud-init
installing openssh-server
restoring apt configuration
downloading and installing security updates
copying logs to installed system

```

[View full log]
[Reiniciar]

```

Ubuntu 20.04 LTS asir2dfb tty1
asir2dfb login: david
Password:
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-29-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of mié 06 may 2020 11:55:27 UTC

System load: 0.39          Processes:      100
Usage of /:  17.0% of 24.48GB Users logged in: 0
Memory usage: 3%          IPv4 address for enp0s17: 10.0.2.15
Swap usage:  0%

0 updates can be installed immediately.
0 of these updates are security updates.

Last login: Wed May 6 11:54:30 UTC 2020 on tty1
david@asir2dfb:~$ _

```

Una vez instalado y reiniciado, podemos empezar a configurar el servidor propiamente dicho. La primera medida es la de siempre, ejecutar un **sudo apt update && sudo apt upgrade -y** para tener el sistema actualizado y, sobre todo, los repositorios actualizados, ya que vamos a instalar unos cuantos paquetes. Una vez hecho eso, por si de casualidad se hizo algún cambio importante en el sistema, podemos ejecutar un **reboot** antes de continuar.

Instalaremos Apache ejecutando **sudo apt install apache2 -y**. Luego un **sudo ufw status** para ver el estado del firewall de Ubuntu, que seguramente esté desactivado. Si es el caso, ejecutamos **sudo ufw enable** para activarlo. Ahora, muy importante, ejecutaremos **sudo ufw allow incoming**, porque

si no el sistema no aceptará las conexiones entrantes. Luego haremos **sudo ufw app list** para verificar que el Apache está en la lista de aplicaciones permitidas. Ahora **sudo ufw app info "Apache Full"** para ver si está abierto el puerto 80, que es el que usa Apache por defecto y no lo vamos a cambiar.

```
david@asir2dfb:~$ sudo ufw status
Status: inactive
david@asir2dfb:~$ sudo ufw enable
Firewall is active and enabled on system startup
david@asir2dfb:~$ sudo ufw default allow incoming
Default incoming policy changed to 'allow'
(be sure to update your rules accordingly)
david@asir2dfb:~$ sudo ufw app list
Available applications:
  Apache
  Apache Full
  Apache Secure
  OpenSSH
david@asir2dfb:~$ sudo ufw app info "Apache Full"
Profile: Apache Full
Title: Web Server (HTTP,HTTPS)
Description: Apache v2 is the next generation of the omnipresent Apache web
server.

Ports:
  80,443/tcp
david@asir2dfb:~$
```

Seguimos con MySQL: ejecutamos **sudo apt install mysql-server -y**. Una vez instalado, **sudo mysql_secure_installation**. Y elegimos las siguientes opciones en este orden: **n** para no usar el componente de validación de contraseñas y pulsamos **Enter**. Ahora escribimos una contraseña para la cuenta de root de MySQL y pulsamos **Enter**. La repetimos y

Enter de nuevo. Ahora pulsamos la secuencia de **y** y **Enter** 4 veces para, respectivamente: impedir el login anónimo, impedir el acceso remoto como root, eliminar las bases de datos de prueba y recargar los privilegios de las tablas.

```
david@asir2dfb:~$ sudo mysql_secure_installation
Securing the MySQL server deployment.

Connecting to MySQL using a blank password.

VALIDATE PASSWORD COMPONENT can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD component?

Press y|Y for Yes, any other key for No: n
Please set the password for root here.

New password:

Re-enter new password:
By default, a MySQL installation has an anonymous user,
allowing anyone to log into MySQL without having to have
a user account created for them. This is intended only for
testing, and to make the installation go a bit smoother.
You should remove them before moving into a production
environment.

Remove anonymous users? (Press y|Y for Yes, any other key for No) : y
Success.

Normally, root should only be allowed to connect from
'localhost'. This ensures that someone cannot guess at
the root password from the network.

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : y_
```

```
a user account created for them. This is intended only for
testing, and to make the installation go a bit smoother.
You should remove them before moving into a production
environment.

Remove anonymous users? (Press y|Y for Yes, any other key for No) : y
Success.

Normally, root should only be allowed to connect from
'localhost'. This ensures that someone cannot guess at
the root password from the network.

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : y
Success.

By default, MySQL comes with a database named 'test' that
anyone can access. This is also intended only for testing,
and should be removed before moving into a production
environment.

Remove test database and access to it? (Press y|Y for Yes, any other key for No) : y
- Dropping test database...
Success.
- Removing privileges on test database...
Success.

Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.

Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y
Success.

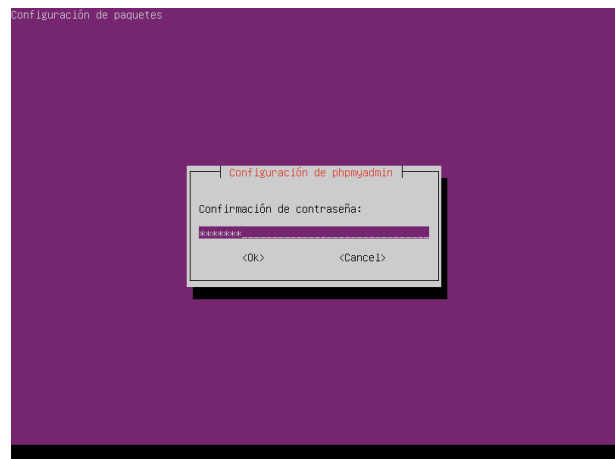
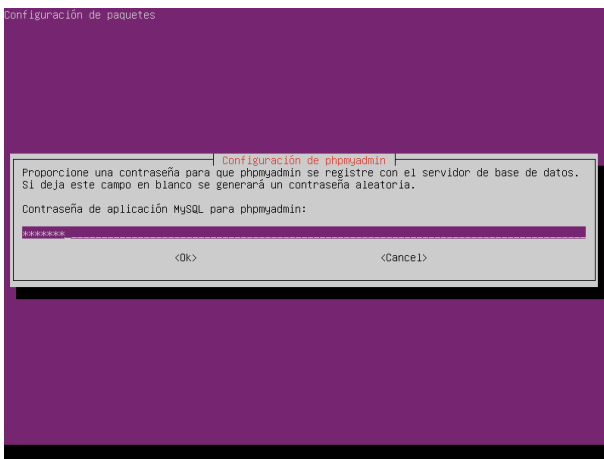
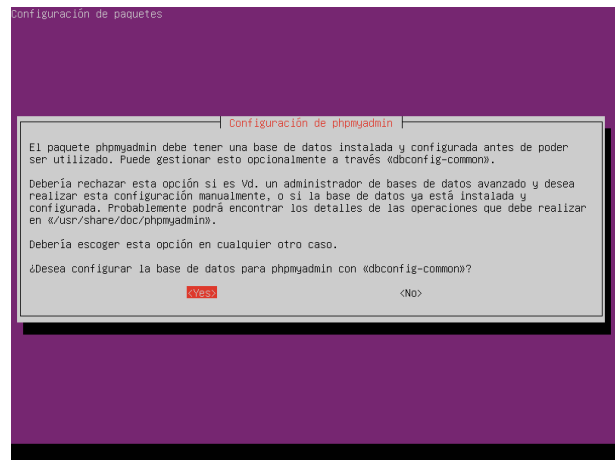
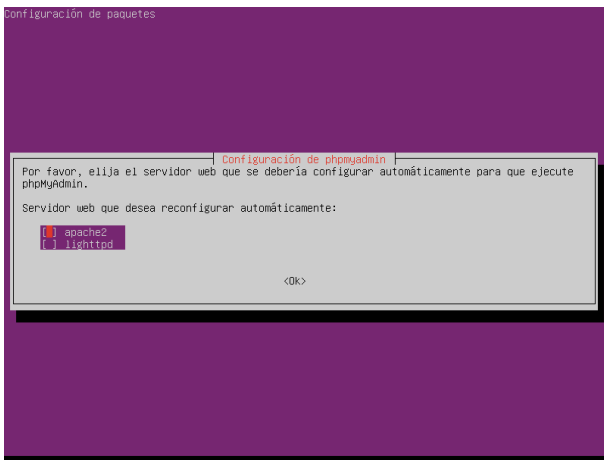
All done!
david@asir2dfb:~$
```

Ahora instalaremos PHP con **apt install php libapache2-mod-php php-mysql -y**.

Después debemos hacernos propietarios de la carpeta del Apache donde se guardan las páginas web, porque como lo instalamos con sudo, el propietario es root, así que ejecutamos **sudo chown -R \$USER:root /var/www**. Después podemos desplazarnos a /var y usar un **ls -l** para asegurarnos de que el comando se completó correctamente.

Para testear (más adelante, cuando tengamos configurado el acceso desde el exterior) que PHP está correctamente instalado, usaremos el editor de texto nano para crear un archivo que contenga simplemente la función `phpinfo()`, que nos mostrará una completa tabla de información sobre la versión de PHP instalada y demás, así que **sudo nano /var/www/html/info.php**

Para terminar el colofón de paquetes, ejecutamos **sudo apt install phpmyadmin -y** para instalar el famoso gestor de bases de datos. Durante la instalación nos preguntará qué servidor web usaremos, seleccionamos **apache2**, en la siguiente pantalla elegimos **yes** para que el programa se autoconfigure solo e introducimos una contraseña de acceso al gestor:



Ahora debemos indicarle al Apache que el PHPMyAdmin existe, porque no se enterará hasta que editemos su archivo de configuración con **sudo nano /etc/apache2/apache2.conf** y añadir al final la línea **Include /etc/phpmyadmin/apache.conf**

Ahora toca reiniciar el servicio apache2 para aplicar todos los cambios que hemos hecho: **sudo service apache2 restart**. Y además un **status** de este y del SSH para asegurarnos de que todo funciona.

Ahora deberemos ir al archivo **/etc/ssh/sshd_config** y descomentar la línea de uso del puerto 22. Ya es el que usa por defecto, pero es mejor hacerlo para evitar posibles inconurrencias.

```
GNU nano 4.0 /etc/apache2/apache2.conf Modified
# Include of directories ignores editors' and dpkg's backup files,
# see README.Debian for details.

# Include generic snippets of statements
IncludeOptional conf-enabled/*.conf

# Include the virtual host configurations:
IncludeOptional sites-enabled/*.conf

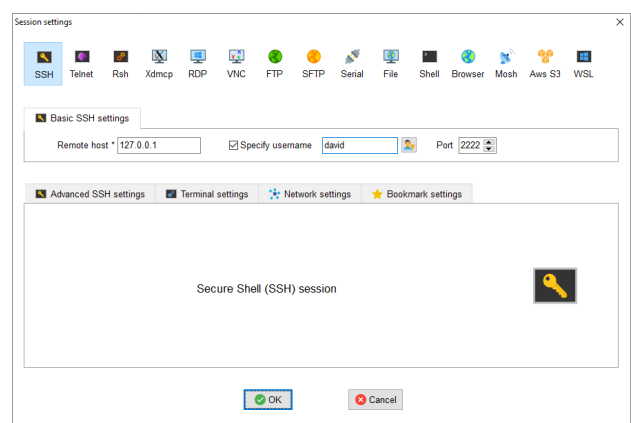
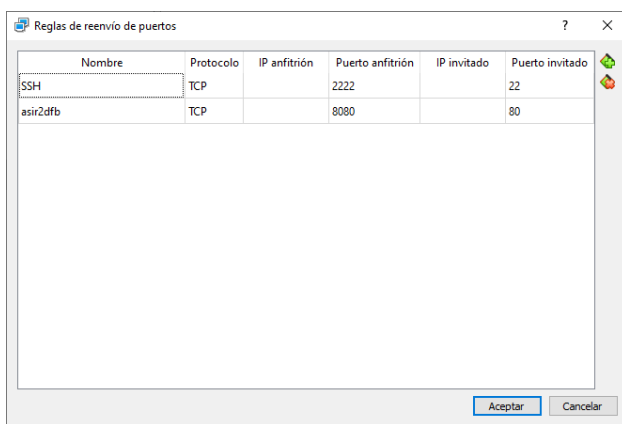
# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
Include /etc/phpmyadmin/apache.conf
```

Guardamos cambios y hacemos un **sudo service ssh restart**. Ahora iríamos a nuestro ordenador cliente (en este caso ejecutando Windows 10) y estableceremos conexión con el servidor mediante SSH con el software MobaXterm. En este caso, al estar el servidor alojado en una máquina virtual dentro del propio cliente (MV con puestos redireccionados como se indica en la imagen a continuación), introduciremos los siguientes datos de conexión:

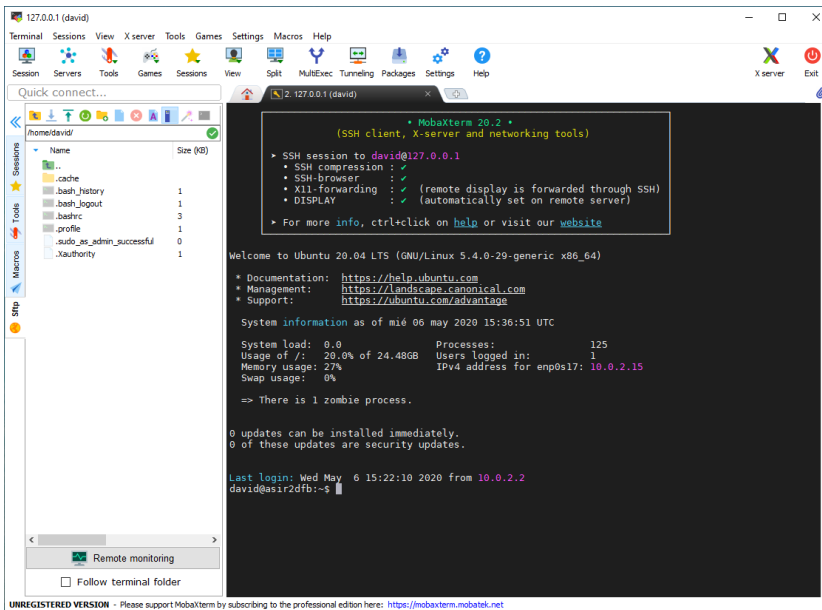
Remote host: 127.0.0.1

Username: David

Port: 2222



El resto del proceso es automático. Al cabo de un par de segundos podremos ver la pantalla de nuestro servidor:



Fijémonos en el panel lateral izquierdo que nos permite transferir archivos y carpetas entre cliente y servidor de forma cómoda e instantánea, incluso arrastrando y soltando.

BASE DE DATOS

Ahora, lo único que nos queda hacer en el propio servidor es crear un usuario con todos los permisos en mysql para poder trabajar cómodamente desde el PHPMYAdmin del servidor en el cliente, así que ejecutamos los siguientes comandos:

```
sudo mysql
```

```
create user 'david'@'localhost' identified by 'abc123.';
```

```
grant all privileges on *.* to 'david'@'localhost' with grant option;
```

```
exit
```

Aquí queda demostrado el poder controlar remotamente el servidor. Ahora todo lo que resta es usar el MySQL Workbench para crear la BD y luego el PHPMYAdmin del servidor para configurarla, y esto último lo haremos desde el navegador del equipo cliente.

Aunque más adelante, al terminar la página web, deberemos usar este programa para transferirla al directorio **`/var/www/html/`**.

```

david@asir2dfb:~$ sudo mysql
[sudo] password for david:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 189
Server version: 8.0.20-0ubuntu0.20.04.1 (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.

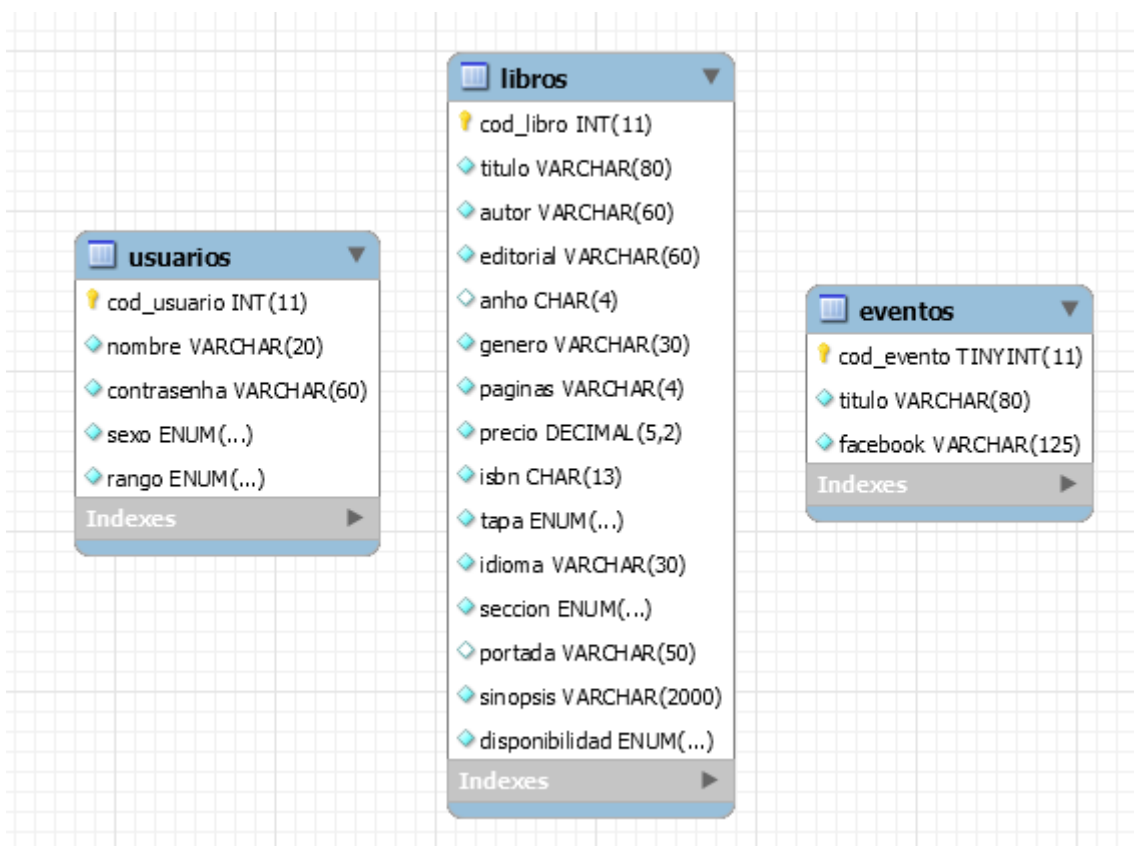
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create user 'david'@'localhost' identified by 'abc123';
Query OK, 0 rows affected (0.00 sec)

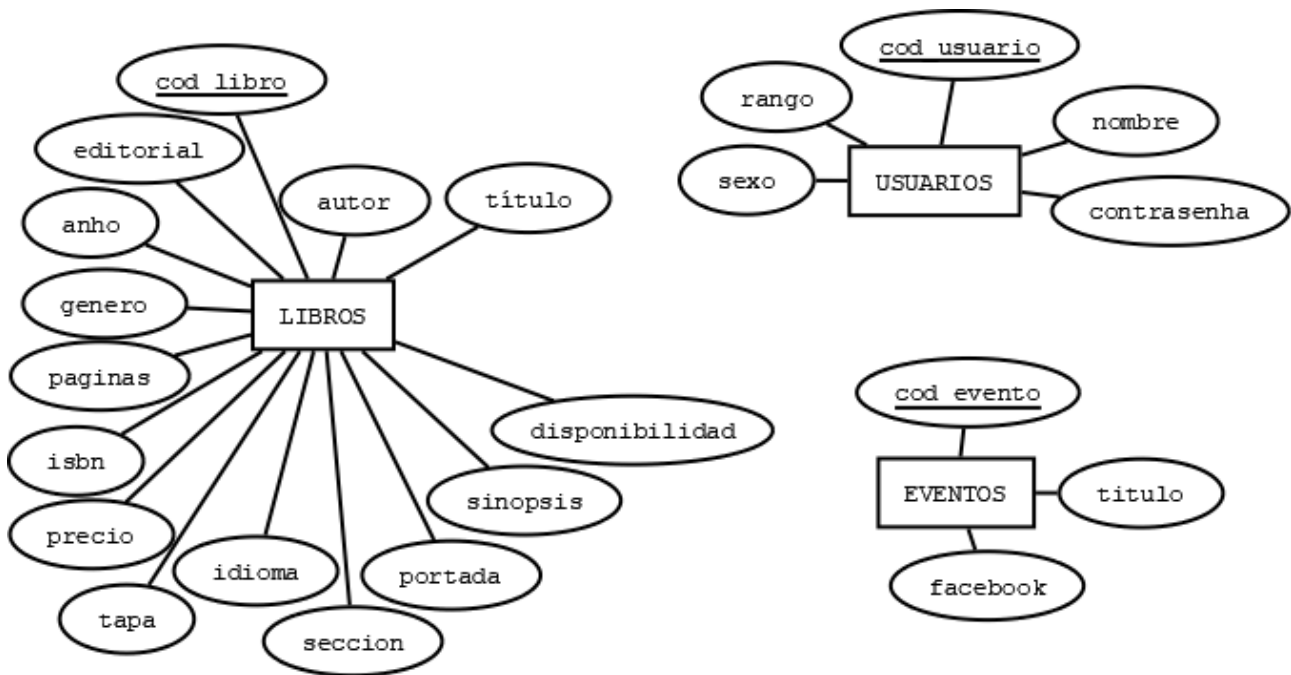
mysql> grant all privileges on *.* to 'david'@'localhost' with grant option;
Query OK, 0 rows affected (0.29 sec)

mysql> exit
Bye
david@asir2dfb:~$
  
```

ESQUEMA DE LA BD SEGÚN EL SOFTWARE MYSQL WORKBENCH



ESQUEMA DE LA BD SEGÚN EL SOFTWARE DIA



CÓDIGO DE LA BD

```

1 • DROP DATABASE IF EXISTS libreria;
2 • CREATE DATABASE IF NOT EXISTS libreria;
3 • USE libreria;
4
5 • DROP TABLE IF EXISTS usuarios;
6 • CREATE TABLE IF NOT EXISTS usuarios (
7     cod_usuario    TINYINT    UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
8     nombre         VARCHAR(20) NOT NULL UNIQUE,
9     contraseña     VARCHAR(60) NOT NULL,
10    sexo           ENUM ('muller', 'home') NOT NULL,
11    rango          ENUM ('administrador', 'empregado') NOT NULL
12 );
  
```

```

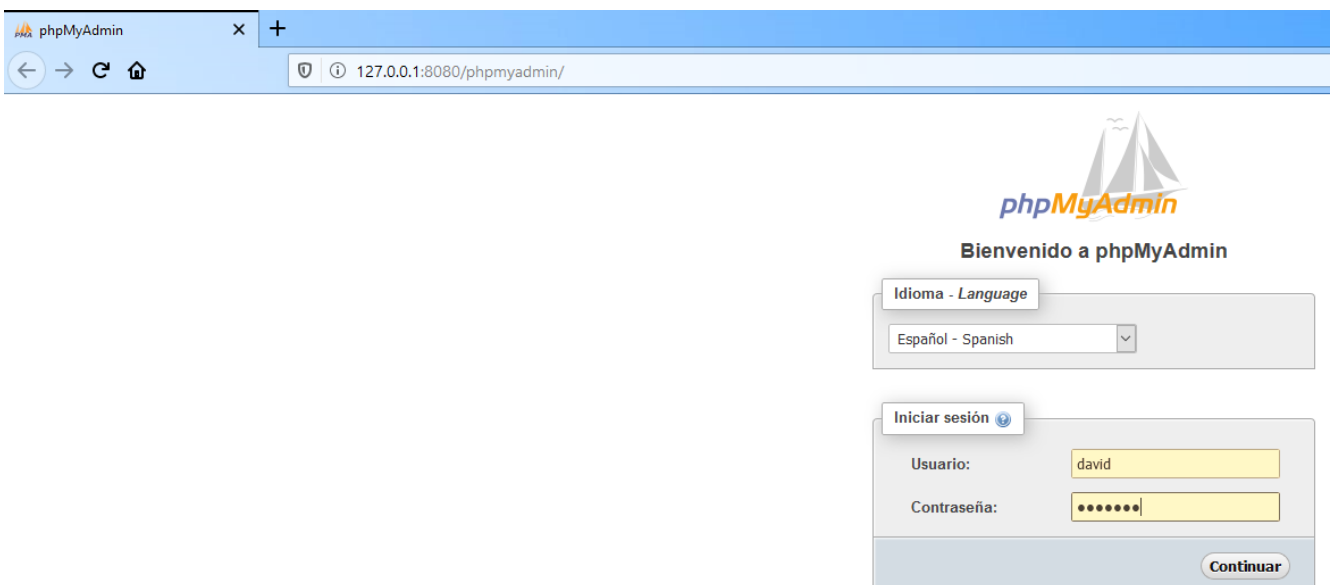
14 • DROP TABLE IF EXISTS libros;
15 • CREATE TABLE IF NOT EXISTS libros (
16     cod_libro      INT      UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
17     titulo         VARCHAR(80) NOT NULL,
18     autor          VARCHAR(60) NOT NULL,
19     editorial       VARCHAR(60) NOT NULL,
20     anho           CHAR(4),
21     genero         VARCHAR(30) NOT NULL,
22     paginas        VARCHAR(4) NOT NULL,
23     precio         DECIMAL(5,2) UNSIGNED NOT NULL,
24     isbn           CHAR(13) NOT NULL,
25     tapa           ENUM ('blanda', 'dura', 'solapas') NOT NULL,
26     idioma         VARCHAR(30) NOT NULL,
27     seccion        ENUM ('non', 'novedad', 'maisVendido', 'mangaComic', 'infantil') NOT NULL,
28     portada        VARCHAR(50),
29     sinopsis       VARCHAR(2000) NOT NULL,
30     disponibilidad ENUM ('sí', 'sí, por encargo', 'descatalogado') NOT NULL
31 );

33 • DROP TABLE IF EXISTS eventos;
34 • CREATE TABLE IF NOT EXISTS eventos (
35     cod_evento     TINYINT   UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
36     titulo         VARCHAR(80) NOT NULL,
37     facebook       VARCHAR(125) NOT NULL
38 );

```

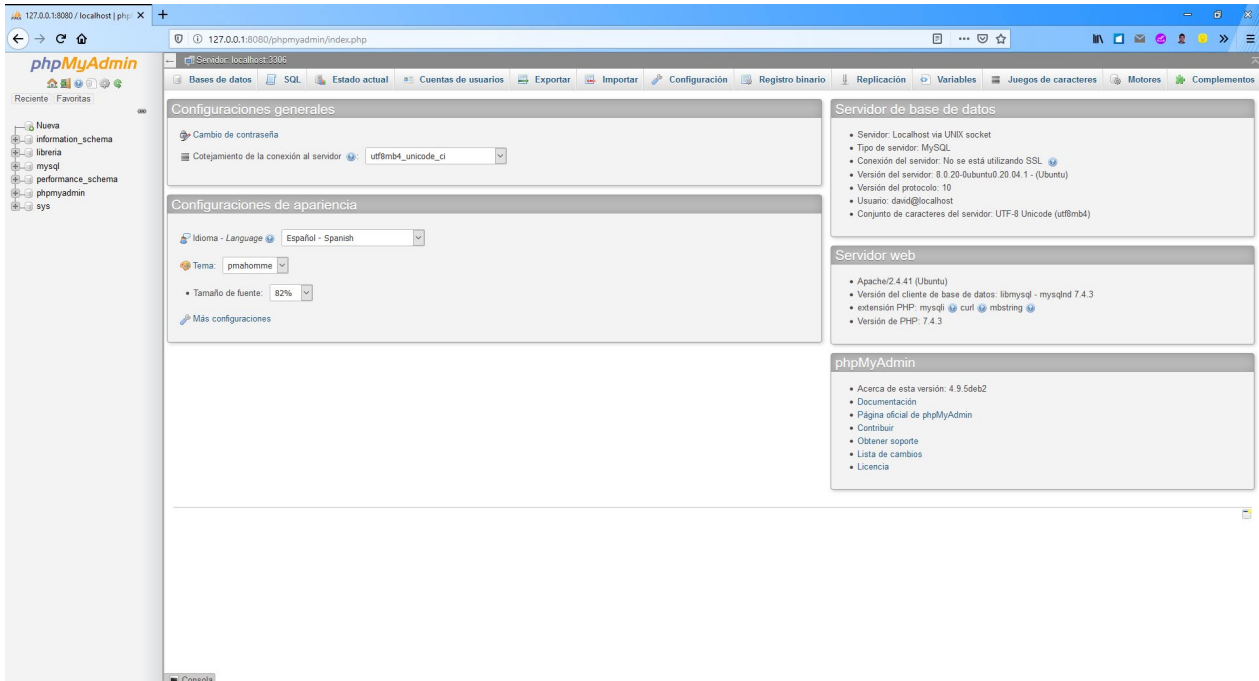
PHPMYADMIN: IMPORTAR BD Y CREAR UN NUEVO USUARIO PARA LA WEB

Nos dirigimos a nuestro ordenador personal y abrimos nuestro navegador habitual para dirigirnos a la dirección **127.0.0.1:8080/phpmyadmin/**. Introducimos los datos **david** y **abc123**. previamente configurados y entraremos en la gestión de las bases de datos:

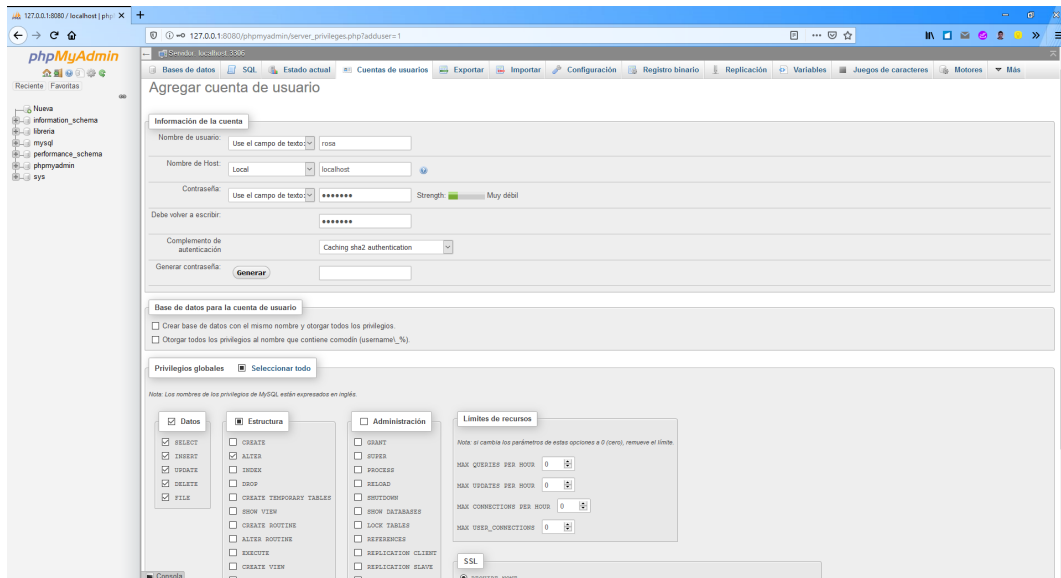


The screenshot shows a web browser window with the address bar containing '127.0.0.1:8080/phpmyadmin/'. The page displays the phpMyAdmin logo and the text 'Bienvenido a phpMyAdmin'. Below this, there is a language selection dropdown menu currently set to 'Español - Spanish'. Underneath is a login section titled 'Iniciar sesión' with two input fields: 'Usuario:' containing the text 'david' and 'Contraseña:' containing a series of dots. A 'Continuar' button is located at the bottom right of the login form.

Ahora debemos crear una nueva BD a la que llamaremos “librería”. Una vez creada, hacemos clic en “Importar” y seleccionamos nuestro archivo SQL generado por el MySQL Workbench, sin modificar ninguna de las opciones del PHPMyAdmin. Una vez hecho esto, ya tendremos nuestra base de datos en el servidor lista para empezar a usarse (véase el panel lateral izquierdo):



Ahora deberemos crear un nuevo usuario a mayores (en este ejemplo lo llamaremos “rosa”), que será el que llevará implantado el código PHP de la web, y que por lo tanto y por seguridad, no tendrá todos los permisos, ya que le bastará con los 5 permisos DML (manipulación de datos) (SELECT, INSERT, UPDATE, DELETE y FILE) y el permiso ALTER de los permisos DDL (definición de datos) (necesario al borrar el registro de cualquier tabla para resetear el contador del campo “id” con AUTO_INCREMENT). El usuario “david” solo será conocido por el creador de la web y administrador de sistemas para su correcta gestión y futuras reparaciones, ampliaciones... etc. Así que nos dirigimos al apartado “Cuentas de usuario” y así lo hacemos:



Una vez hecho esto, habremos acabado aquí también.

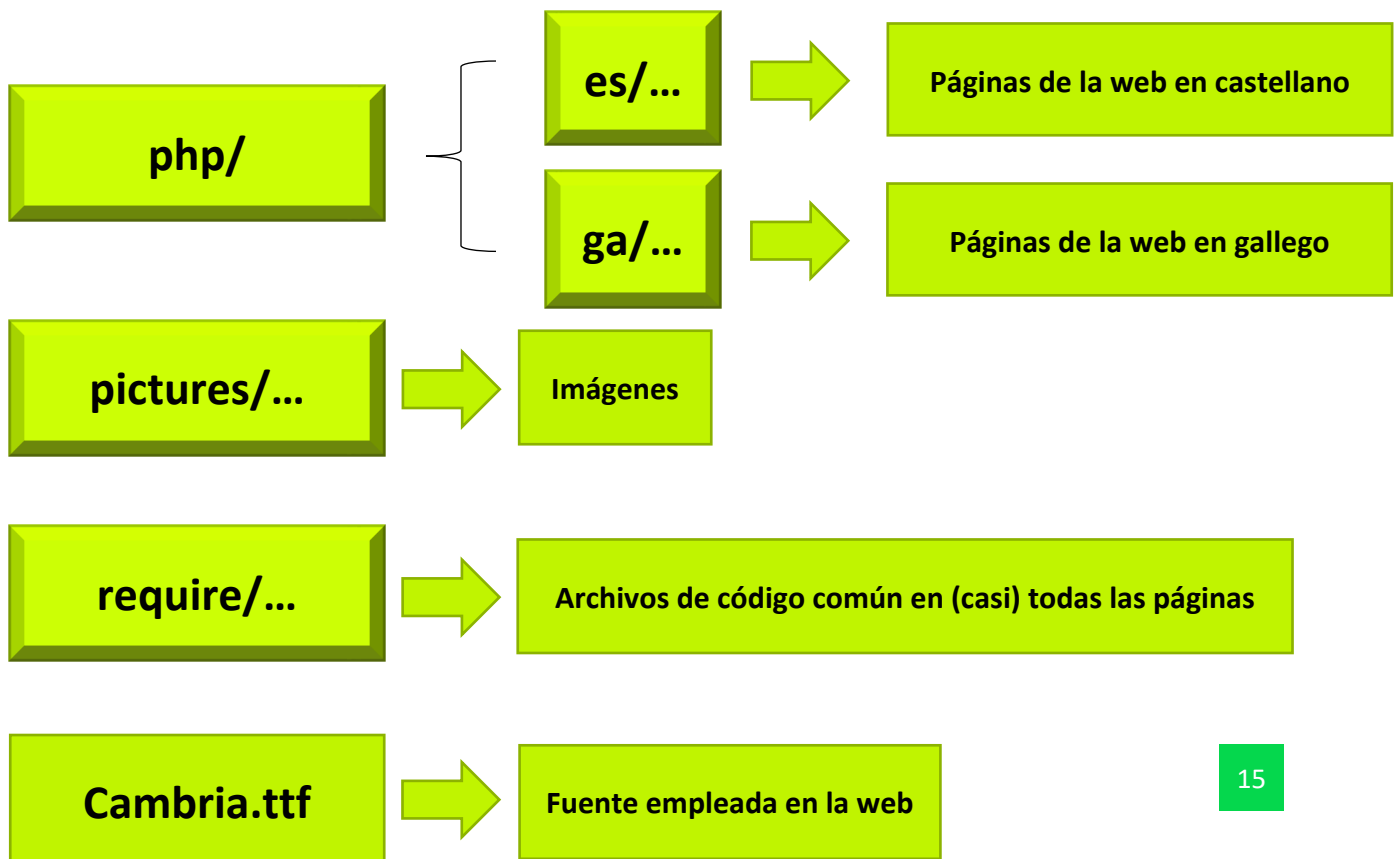
Pasamos al desarrollo web.

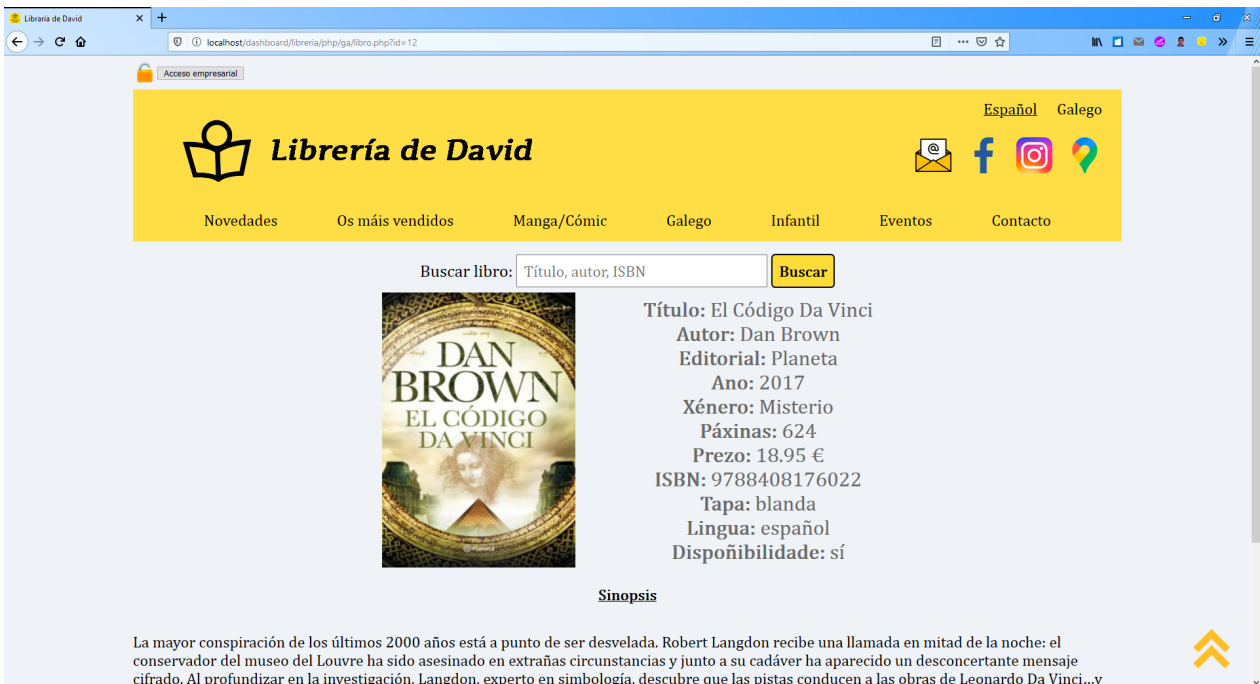
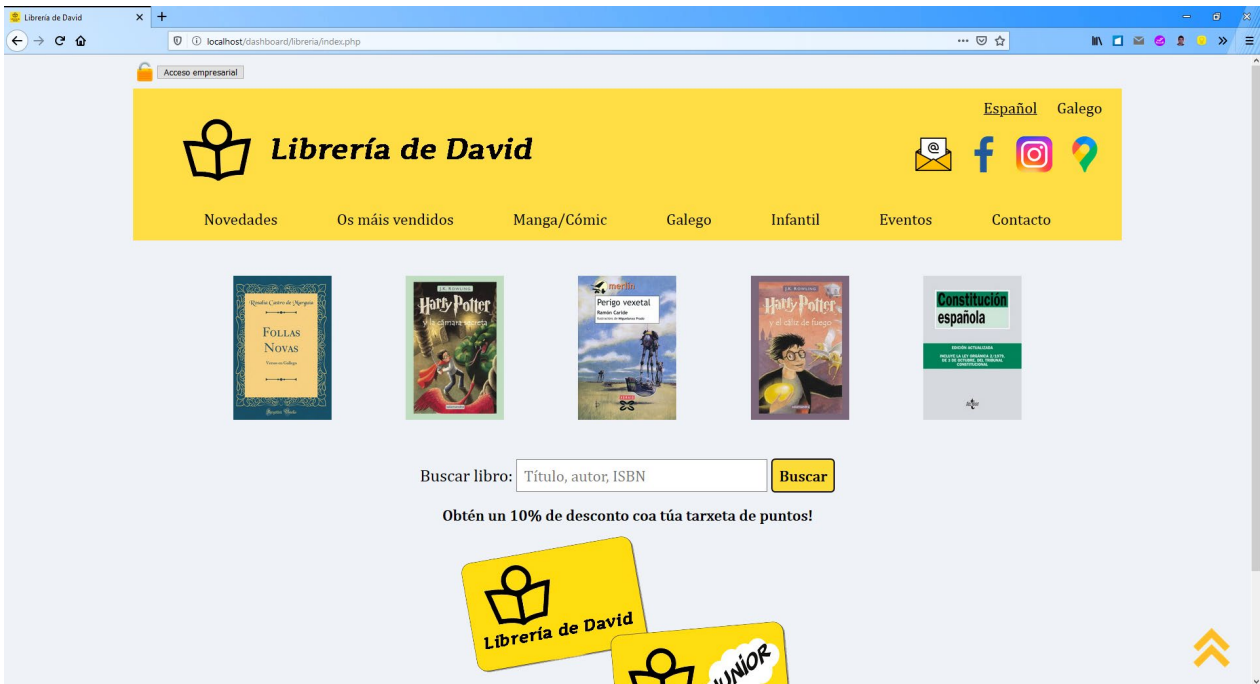
PÁGINA WEB

ESQUEMA

Se ha usado el software Brackets junto con XAMPP otorgando Apache y MySQL para realizar la página web. A continuación, se muestra un esquema de los directorios de la misma.

Los rectángulos en 3D son carpetas y los 2D, archivos.





EVITANDO QUE SE PUEDA VISUALIZAR LA ESTRUCTURA

En cada carpeta se ha colocado un archivo llamado **index.php**, pero que, a diferencia del original situado en la raíz del proyecto y el que se encuentra en **/php/es/index.php** (la portada está en gallego), solo contienen una línea de código para redireccionar al archivo **index** principal. De esta forma se protege el árbol

```
1 |<?php
2 |   header('location:../index.php');
3 | ?>
```

de directorios de posibles curiosos, ya que, si alguien intentase acceder a una carpeta, pero no a un

archivo en concreto, normalmente se vería un explorador de archivos que ya confeccionan los navegadores, pero de esta forma siempre se redireccionará a la portada en gallego de la web.

PARTE LÓGICA

```

1 <?php
2 class libreria {
3     private $conexion;
4
5     function constructor() { ... }
10
11     function buscarIndex() { ... }
14     function buscarZonaPublica_galego() { ... }
17     function buscarZonaPublica() { ... }
200
201     function libro_galego() { ... }
251     function libro() { ... }
301     function sliderPortadasIndex() { ... }
320     function sliderPortadasEsp() { ... }
339
340     function librosNovedad_galego() { ... }
397     function librosMaisVendido_galego() { ... }
454     function librosManga_galego() { ... }
511     function librosGalego_galego() { ... }
568     function librosInfantil_galego() { ... }
625
626     function librosNovedad() { ... }
683     function librosMaisVendido() { ... }
740     function librosManga() { ... }
797     function librosGalego() { ... }
854     function librosInfantil() { ... }
911
912     function email() { ... }
933
934     function mostrarEventos() { ... }
955
956     function iniciarSesion() { ... }
994
995     function saludo() { ... }

```

utilización de frameworks ni plugins ni plantillas (como toda la página web).

El archivo viene a contener una clase, que a su vez contiene 43 funciones:

La web utilizará la API *PDO* (por maximizar compatibilidad para un futuro, ya que *mysqli* solo soporta bases de datos MySQL, mientras que *PDO* soporta esa y otras tantas más).

En la carpeta **require/** de la web habrá varios archivos PHP, pero el que contendrá todas las funciones que consiguen que la parte lógica funcione será **libreria.php**, que consta de casi 2,000 líneas de código, todo hecho a mano en su totalidad, sin

```

1017     function buscarZonaPrivada() { ... }
1102
1103     function insertarLibro() { ... }
1186     function insertarUsuario() { ... }
1211     function insertarEvento() { ... }
1230
1231     function editarLibro() { ... }
1316     function editarSinopsis() { ... }
1343     function editarUsuario() { ... }
1449     function editarEvento() { ... }
1483
1484     function actualizarLibro() { ... }
1514     function actualizarSinopsis() { ... }
1533     function actualizarUsuario() { ... }
1559     function actualizarEvento() { ... }
1579
1580     function borrarLibro() { ... }
1611     function borrarUsuario() { ... }
1641     function borrarEvento() { ... }
1671
1672     function tablaPrivada() { ... }
1812     function tablaEventos() { ... }
1851     function tablaUsuarios() { ... }
1927
1928     function usuariosYeventos() { ... }
1939     function eventos() { ... }
1949
1950     function cerrarSesion() { ... }
1969
1970 ?>

```

CONEXIÓN DE LA WEB CON LA BD

Como se dijo anteriormente, todo por PDO y, además, usando un estilo orientado a objetos:

```

1 <?php
2 class libreria {
3     private $conexion;
4
5     function constructor() {
6         $this->conexion=new PDO('mysql:host=localhost;dbname=libreria;charset=utf8', 'rosa', 'abc123. ');
7
8         $this->conexion->setAttribute(PDO::ATTR_ERRMODE,PDO::ERRMODE_EXCEPTION);
9     }

```

EVITAR INYECCIÓN SQL

La inyección SQL se basa en cambiar la consulta previamente establecida por el desarrollador para ver más datos de los permitidos o incluso manipularlos. Todo consiste en cerrar la consulta escrita en el backend con unas comillas y abrir otras para añadir a mayores la que a ti te interese.

Imaginemos una barra de búsqueda de libros pública. Si el código quedase vulnerable, se podría buscar lo siguiente:

' or '1'='1

Con esto se mostrarían todos los registros de esa tabla. Ha de saberse que, si en una sentencia se introduce una igualdad absoluta (1=1, patata=patata...), eso equivale a mostrar todas las filas, ya que la segunda condición se cumpliría siempre; algo siempre es igual a sí mismo. Ya ni hablemos de si en vez de ser una barra de búsqueda (que solo muestra datos) fuera un botón de borrar usuarios. Usando el botón de borrar nuestra cuenta de usuario (habiendo previamente un campo de texto para indicar nuestro nombre), podríamos borrarlos todos.

En pro de evitar la inyección SQL y que un cracker (ciberdelincuente, que no “hacker”) pueda acceder a datos privados o incluso manipularlos, absolutamente todas las consultas irán dentro de la función *prepare()*, que las convierte en consultas preparadas, esto es, inmunes a este tipo de manipulación. He aquí el ejemplo de una barra de búsqueda:

```
// Aquí se averigua cuántos coinciden con lo que se ha buscado
$consulta = "SELECT * FROM libros WHERE titulo LIKE :busq OR autor LIKE :busq OR isbn LIKE :busq";
$resultado=$this->conexion->prepare($consulta);
$resultado->execute(array(':busq'=>"%$busqueda%"));
$libros = $resultado->rowCount();

// Aquí empiezan a mostrarse de 3 en 3
for ($empezar = 0; $empezar < $libros; $empezar+=3) {
    $consulta = "SELECT * FROM libros WHERE titulo LIKE :busq OR autor LIKE :busq OR isbn LIKE :busq LIMIT " . $empezar . ", 3";
    $resultado=$this->conexion->prepare($consulta);
    $resultado->execute(array(':busq'=>"%$busqueda%"));
}
```

INICIO DE SESIÓN SEGURO Y COOKIE PARA GUARDARLA

El acceso a la zona privada de la web se hará con un formulario simple de usuario, contraseña y un `<input>` de



```

function iniciarSesion() {
    if (isset($_POST['login'])) {
        try {
            $usuario=htmlentities(addslashes($_POST['usuario']));
            $contrasena=htmlentities(addslashes($_POST['contrasena']));

            $this->constructor();

            $consulta = "SELECT * FROM usuarios WHERE nombre = :usuario";
            $resultado=$this->conexion->prepare($consulta);
            $resultado->execute(array(':usuario'=>$usuario));

            foreach ($resultado as $usuarios) {

                // Si la contraseña es correcta, continuamos
                if (password_verify($contrasena, $usuarios['contrasena'])) {

                    // Crear sesión si está todo OK y cookie con rango de usuario
                    session_start();
                    $_SESSION['sesion_iniciada']=$_POST['usuario'];
                    setcookie('rango_usuario',$usuarios['rango'],time()+2592000);

                    // Crear cookie si se marcó la casilla de "Recordar sesión" (duración de 1 mes)
                    if (isset($_POST['recordar']))
                        setcookie('recordar_sesion',$_POST['usuario'],time()+2592000);

                    header('location: private.php');
                }
            }

            // Mostrar error y su código si pasó algo con la BD / Cerrar conexión al acabar
        } catch (Exception $e) {
            exit("Error: " . $e->getMessage());
        } finally {
            $this->conexion=null;
        }
    }
}

```

tipo *checkbox* por si se desea guardar la sesión, que lo que hará es crear una cookie para ello.

Las contraseñas de los usuarios son almacenadas en la BD usando la función de PHP *password_hash()*, por lo que, para comprobarla, es necesario usar la función *password_verify()*.

Adicionalmente, siempre se creará una cookie que

contenga el rango del usuario (para así, recordemos, dirigirle a la zona privada con o sin gestión de usuarios).

PROTECCIÓN ANTI-INTRUSOS NO AUTENTICADOS PARA LA ZONA PRIVADA

Como se acaba de explicar, al iniciar sesión se crea una ídem y una cookie de rango de usuario (e incluso una cookie de recordar la sesión si se quiere), por lo que, si se accede a una zona privada, siempre se debe comprobar antes de nada que exista al menos una de las 2 cosas, la sesión o la cookie de haberla recordado.

Código presente en todas las páginas de la zona privada:

```

session_start();
// Si no existe la sesión ni la cookie, se redirige al login
if (!isset($_SESSION['sesion_iniciada']) &&
    !isset($_COOKIE['recordar_sesion'])) {
    header('location: login.php');
}

```

Análogamente, la página principal de la zona privada (la tabla de libros) llevará una comprobación extra para saber si el usuario es un “administrador” o un “empleado” (rangos definidos en la BD), para así mostrar o no el botón “Usuarios”.

```

if (isset($_COOKIE['rango_usuario']))
    if ($_COOKIE['rango_usuario']=='administrador')
        $web->usuariosYe eventos();
    elseif ($_COOKIE['rango_usuario']=='empleado')
        $web->eventos();

```

```
// Si el usuario no es admin, se redirige al login
if (isset($_COOKIE['rango_usuario']))
    if ($_COOKIE['rango_usuario']=='empregado')
        header('location: login.php');
```

Y, por supuesto, la página de gestión de usuarios está protegida para permitir el acceso solo a

usuarios administradores.

ZONA PRIVADA: VISTA GENERAL

Título	Autor	Editorial	Ano	Xénero	Páxinas	Prezo	ISBN	Tapa	Lingua	Sección	Portada	Sinopsis	Disponibilidade	Operacións
Una España mejor	Mariano Rajoy Brey	PLAZA & JANES EDITORES	2019	Biografía	384	20.80 €	9788401022739	dura	español	novedad		Editar	sí	Borrar
O lapis do carpinteiro	Manuel Rivas, Xosé Luis de dios	Xerais	2005	Ciencia ficción	184	25.00 €	9788497823913	blanda	galego	non		Editar	sí	Borrar
Follas novas	Rosalía de Castro	Galaxia	2019	Ciencia ficción	312	12.82 €	9788482884806	blanda	galego	non		Editar	sí	Borrar

Primero, este CRUD nos saluda con el nombre de usuario correspondiente. Esto es gracias a que, por la sesión o la cookie, tenemos guardado el nombre del usuario, así que podemos crear un mensaje personalizado muy fácilmente. Y, si le ponemos un poquito más de ganas (usando el campo “sexo” de la BD creado simplemente para tal fin), podemos personalizar la primera palabra del saludo. Incluso iremos más allá, usando la función *ucfirst()* para poner la primera letra del nombre en mayúscula.

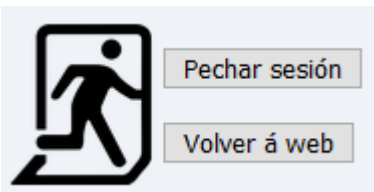
Código aquí presente:

```
function saludo() {
    $this->constructor();

    if (isset($_COOKIE['recordar_sesion'])) {
        $nombre = $_COOKIE['recordar_sesion'];
        $consulta = "SELECT nombre, sexo FROM usuarios WHERE nombre='".$nombre'";
    } elseif (isset($_SESSION['sesion_iniciada'])) {
        $nombre = $_SESSION['sesion_iniciada'];
        $consulta = "SELECT nombre, sexo FROM usuarios WHERE nombre='".$nombre'";
    }

    $resultado=$this->conexion->prepare($consulta);
    $resultado->execute();

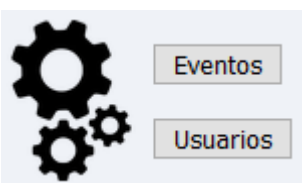
    foreach ($resultado as $nombre) {
        if ($nombre['sexo']=='Muller')
            printf("<h3>Benvenida, ".ucfirst($nombre['nombre'])."</h3>");
        elseif ($nombre['sexo']=='Home')
            printf("<h3>Benvido, ".ucfirst($nombre['nombre'])."</h3>");
    }
}
```



Después encontramos 2 botones: uno para cerrar la sesión (que destruye la sesión y la cookie, aunque una de las 2 no exista) y otro para volver a la web, que no es más que un enlace al archivo **index.php** principal. Código de la función de cerrar sesión:

```
function cerrarSesion() {
    printf("
    <div class=opciones>
    <div><img src=../../pictures/salir.png alt=salir id=salir /></div>
    <div class=botones>
    <div><a href=private.php?logout><input type=submit name=logout value=\"Pegar sesión\" /></a>
    </div>
    <div><a href=index.php><input type=submit value=\"Volver á web\" /></a></div>
    </div>
    ");

    if (isset($_GET['logout'])) {
        session_start();
        session_destroy();
        setcookie('recordar_sesion',$_POST['usuario'],time()-1);
        setcookie('rango_usuario',$usuarios['rango'],time()-1);
        header('location: index.php');
    }
}
```



Lo siguiente son otros dos botones que solo son enlaces a otras páginas, apareciendo el segundo solo cuando el usuario sea un administrador (estas secciones se verán más adelante).

Buscar libro:

Después
viene una
barra de
búsqueda
cuyo código
esencial de

```
if (isset($_GET['buscar'])) {
    $busqueda=$_GET['buscar'];

    try {
        $this->constructor();

        $consulta = "SELECT * FROM libros WHERE
titulo LIKE :busq
OR autor LIKE :busq
OR idioma LIKE :busq
OR seccion LIKE :busq
OR isbn LIKE :busq";
        $resultado=$this->conexion->prepare($consulta);
        $resultado->execute(array(':busq'=>"%$busqueda%"));
    }
}
```

funcionamiento es una simple consulta con un LIKE.

Libros a enseñar por página:
Mostrando 3 libros por página
Libros totais: 20
Páginas: [1](#) · [2](#) · [3](#) · [4](#) · [5](#) · [6](#) · [7](#)

Ahora tenemos un elemento esencial cuando los elementos totales a mostrar son muchos: la paginación. El código viene configurado para

mostrar 3 libros por página, pero el usuario final puede personalizar esto como quiera.

Y no con el típico <select> con los valores 5, 10, 20 y 50, sino con un cuadro de texto precioso de tipo *number* para escribir el número que se prefiera. Y por supuesto, dicho número quedará guardado en una cookie para que los cambios prevalezcan aunque se cierre el navegador:

```
//          Paginación          // COMIENZO

if (isset($_GET['pagina'])) {
    if ($_GET['pagina']==1)
        header('location: private.php');
    else
        $pagina = $_GET['pagina'];
} else
    $pagina = 1;

printf("<br>Libros a enseñar por página:
<form action=private.php method=get>
    <input type=number name=filas size=5 />
    <input type=submit value=\"Cambiar\" />
</form><br>
");

if (isset($_GET['filas'])) {
    setcookie('numero_paginacion', $_GET['filas'], time()-1);
    setcookie('numero_paginacion', $_GET['filas'], time()+2592000);
    $filas_pagina = $_COOKIE['numero_paginacion'];
    header('location: private.php');
}

elseif (isset($_COOKIE['numero_paginacion']))
    $filas_pagina = $_COOKIE['numero_paginacion'];
else
    $filas_pagina = 3;      // Número a cambiar para otro número de filas por defecto

$sempazar_desde = ($pagina-1)*$filas_pagina;

printf("Mostrando <b>". $filas_pagina. "</b> libros por página<br>");
```

La paginación pondrá en la barra de direcciones el número de página actual, así que colocaremos un *if()* para que, si es un 1, es decir, si estamos en la primera página, nos dirija a la propia zona privada, y si no, nos dirija a la página en la que hayamos hecho clic.

Si no se ha mandado el valor “página” por GET, por el formulario, entonces es que no se ha pulsado en ningún número de página, así que por narices la página en la que estamos tiene que ser la primera. Esta explicación es sobre el primer condicional.

Sobre el segundo: este es para el cuadro de texto que nos permite personalizar la paginación: si detecta uso de él, creará una cookie con el número indicado para aplicarse (valor de la variable **\$filas_pagina**).

A su vez y a continuación, el *elseif()* dice que si ya existe la cookie de personalizar la paginación, entonces el número de filas por página sea precisamente el guardado en la cookie.

El *else* controla que, si no existe la cookie y no se ha introducido nuevo número de paginación, se mostrarán 3 libros por página.

Ahora creamos la variable **\$empezar_desde** que hará lo siguiente: a la página en la que estamos le restamos 1, y luego lo multiplicamos por el número de filas por página (libros por página, lo que acabamos de explicar). Si mostramos 3 libros y estamos en la primera página, el resultado sería 0 $[(1-1) \times 3]$. ¿De qué nos sirve esto? Para usar con el LIMIT de la consulta que muestra los libros.

Adicionalmente se ha añadido una línea que diga el número de libros totales de la BD, lo cual se consigue haciendo una consulta muy simple y usando la función de la API PDO `rowCount()`, que cuenta las columnas que salen de la consulta. Guardamos ese número en la variable **\$total_filas** y creamos otra llamada **\$total_paginas** que será igual a la división de las filas totales entre las filas por página que mostremos; eso nos dará más adelante la fila de la paginación.

```
try {
    $this->constructor();
    $consulta = "SELECT * FROM libros";
    $resultado=$this->conexion->prepare($consulta);
    $resultado->execute(array());

    $total_filas = $resultado->rowCount();
    $total_paginas = ceil($total_filas / $filas_pagina);
    printf("Libros totais: ".$total_filas);

} catch (Exception $e) {
    exit("Error: " . $e->getMessage());
} finally {
    $this->conexion=null;
}

try {
    $this->constructor();

    $consulta = "SELECT * FROM libros ORDER BY cod_libro DESC LIMIT ".$empezar_desde.", ".$filas_pagina;
    $resultado=$this->conexion->prepare($consulta);
    $resultado->execute(array());

    // Bucle para ver los números de las páginas
    printf("<br>Páxinas: ");
    $paginas = "";
    for ($i = 1; $i <= $total_paginas; $i++) {
        $paginas .= " <a href='?pagina=".$i.">".$i."</a> ";
    }

    // Eliminar el último punto sobrante del final del bucle
    printf(trim($paginas, "."));

    //          Paginación          // FINAL
```

Usando un bucle for (un bucle determinado, ya que conocemos las veces que se tiene que repetir lo que queremos. O sea, nosotros no, pero el código sí) se ha podido hacer lo siguiente: creando una variable llamada **\$paginas** igual a una cadena de caracteres vacía, se ha establecido que con una variable **\$i** igual a 1, mientras **\$i** sea menor o igual que **\$total_paginas**, se le irá sumando 1 a **\$i**. Y reasignamos la variable **\$paginas**: ahora será igual al *string* vacío que ya tenía más una serie de enlaces a las páginas correspondientes que el bucle genere con la variable **\$i**, que valdrá primero 1, luego 2, luego 3... Según la división previamente calculada de **\$total_paginas**. Y a continuación de cada número irá un punto (·) para separarlos un poco.

Ejemplos mostrando 3, 5 y 10 libros por página. Todo el proceso es automático e instantáneo:

Libros a enseñar por página:
 Mostrando **3** libros por página
 Libros totais: 20
 Páginas: [1](#) · [2](#) · [3](#) · [4](#) · [5](#) · [6](#) · [7](#)

Libros a enseñar por página:
 Mostrando **5** libros por página
 Libros totais: 20
 Páginas: [1](#) · [2](#) · [3](#) · [4](#)

Libros a enseñar por página:
 Mostrando **10** libros por página
 Libros totais: 20
 Páginas: [1](#) · [2](#)

Pero entonces, siguiendo esta lógica, también habría un punto después del último número, el cual no nos hace ninguna falta. Otros lenguajes de programación ya entienden esta lógica y lo eliminan

```
// Bucle para ver los números de las páginas
printf("<br>Páginas: ");
$paginas = "";
for ($i = 1; $i <= $total_paginas; $i++) {
    $paginas .= " <a href='?pagina=".$i."'>".$i."</a> ";
}

// Eliminar el último punto sobrante del final del bucle
printf(trim($paginas, "."));

//          Paginación          // FINAL
```

automáticamente, pero PHP no es uno de ellos. Por suerte, es tan fácil de solucionar como no mostrar la variable **\$paginas** en el bucle for, sino solo calcular su valor. La variable se mostrará fuera del bucle y dentro de la función *trim()* a la que se le pasará un (·) como segundo parámetro para que lo elimine al final.

Sobre la tabla (CRUD) en sí: la primera fila siempre es la de añadir un libro nuevo. Todos los campos, menos el “año” y la “portada” son obligatorios (llevan un “NOT NULL” en SQL y a su vez esos <input> en PHP tienen un *required*). Y por supuesto el input de tipo *file* de las portadas lleva un *accept=image/png, image/jpeg* para evitar saturaciones del servidor por error. El botón de “insertar” usa una función que básicamente contiene un “INSERT INTO”.

Luego, cada fila contiene un libro, lee sus atributos de la BD y al final todos tienen 2 botones, uno de "editar" y otro de "borrar". Ambos pasan por GET el ID (atributo "cod_libro") de cada libro y ejecutan la función correspondiente. En el caso de la edición, la fila se transforma y los campos de las celdas pasan a ser <input> para editarlos de forma rápida y sencilla (una solución alternativa era crear una página adicional para editar el libro en cuestión, pero haciéndolo como se ha hecho, aunque más tedioso, es más profesional):

Título	Autor	Editorial	Ano	Xénero	Páxinas	Prezo	ISBN	Tapa	Lingua	Sección	Disponibilidade	Operacións
Una España mejor	Mariano Rajoy Brey	PLAZA & JANES EDITORES	2019	Biografía	384	20.80 €	9788401022739	blanda	español	non	si	<input type="button" value="Guardar"/> <input type="button" value="Cancelar"/>

A la hora de borrar un libro, antes de ejecutarse esa función, se ejecuta un sencillo código JavaScript, una función *confirm()* para mostrar un mensaje de confirmación por precaución:

SEGURO QUE DESEXAS BORRAR ESTE LIBRO?

Título	Autor	Editorial	Ano	Xénero	Páxinas	Prezo	ISBN	Tapa	Lingua	Sección	Portada	Sinopsis	Disponibilidade	Operacións
Una España mejor	Mariano Rajoy Brey	PLAZA & JANES EDITORES	2019	Biografía	384	20.80 €	9788401022739	dura	español	novedad		Editar	si	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>
O lapis do carpinteiro	Manuel Rivas, Xosé Luis de dios	Xerais	2005	Ciencia ficción	184	25.00 €	9788497823913	blanda	galego	non		Editar	si	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>
Follas novas	Rosalía de Castro	Galaxia	2019	Ciencia ficción	312	12.82 €	9788482884806	blanda	galego	non		Editar	si	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>

```
<a href=private.php?borrar=" . $tabla['cod_libro'] . ">
  <input id=borrar type=submit value=Borrar onclick=\"return confirm('SEGURO QUE
  DESEXAS BORRAR ESTE LIBRO?')\" />
</a>
```

Y, de llevarse a cabo el borrado, también se resetea el contador del AUTO_INCREMENT de esa tabla de la BD para que no empiecen a quedar huecos por doquier. De ahí que anteriormente se mencionaba que el usuario de la BD necesita el permiso ALTER:

```
function borrarLibro() {
    // Si se pulsa el botón "Borrar" de algún libro, entonces...
    if (isset($_GET['borrar'])) {
        try {
            // Se conecta a la BD
            $this->constructor();

            // Se borra el libro y se resetea el contador del cod_libro
            $consulta = "DELETE FROM libros WHERE cod_libro=".$_GET['borrar']."";
            $resultado=$this->conexion->prepare($consulta);
            $resultado->execute();
            $consulta = "ALTER TABLE libros DROP cod_libro";
            $resultado=$this->conexion->prepare($consulta);
            $resultado->execute();
            $consulta = "ALTER TABLE libros AUTO_INCREMENT=1";
            $resultado=$this->conexion->prepare($consulta);
            $resultado->execute();
            $consulta = "ALTER TABLE libros ADD cod_libro INT NOT NULL AUTO_INCREMENT PRIMARY KEY FIRST";
            $resultado=$this->conexion->prepare($consulta);
            $resultado->execute();

            // Y se redirige a la zona privada inicial
            header('location: private.php');
        } catch (Exception $e) {
            exit("Error: " . $e->getMessage());
        }
    }
}
```

Estas 3 consultas adicionales de ALTER también se realizan a la hora de borrar usuarios y eventos (las otras 2 tablas de la BD).

GESTIÓN DE USUARIOS

Como se ha dicho y enseñado, esta zona está permitida solo a usuarios administradores por la cookie “rango_usuario” que se crea y guarda al iniciar sesión. Esta página viene a ser otro CRUD:

The screenshot shows a web browser window with the URL `127.0.0.1:8080/libreria/php/ga/usuarios.php`. The page content is as follows:

Nombre (único)	Contraseña (encriptada)	Sexo	Rango	
Rosa	\$2y\$10\$7sFgx5x7NRslC3R6w4Lj0hWSH2uGj2cvUbunk7wJMYo2leFFcAku	Muller	administrador	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>
David	\$2y\$10\$8sGwDVU3USDRDVOB3gEod.HhuNIQzrv7BAcyMHBicwASWlzOWA4gq	Home	empleado	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>

Below the table, there are two links: [Volver á táboa de libros](#) and [Volver á web](#).

En la columna **Nombre (único)** se recuerda al usuario final que éstos deben ser únicos (para eso, en el código SQL ese atributo lleva el “UNIQUE”), y recordemos que las contraseñas se crean cifradas,

así que también se almacenan de la misma manera. Aparte, es una cuestión de privacidad: el administrador tiene un control total sobre todo, pero no debe conocer las contraseñas de los usuarios; si quisiera podría cambiarlas o eliminarlos.

La filosofía de la zona de eventos viene a ser la misma: otro CRUD con campos para escribir un título y el enlace de inserción del evento ya publicado en Facebook; el código que lo muestra en público (proporcionado por la propia Facebook) ya está listo para mostrarlo de forma correcta, parecido a un `<iframe>`:

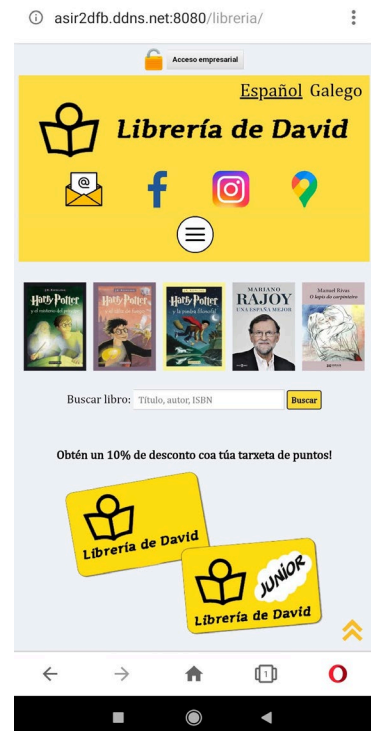
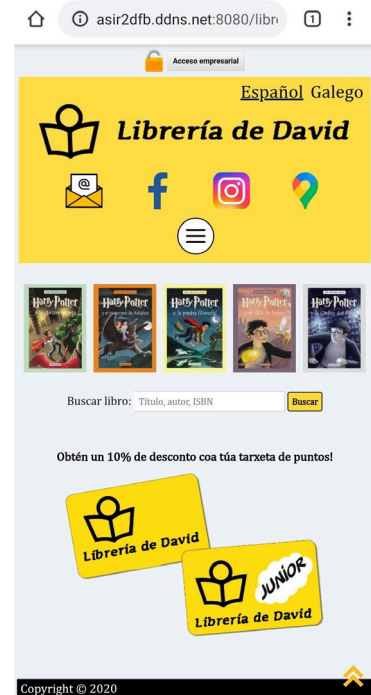
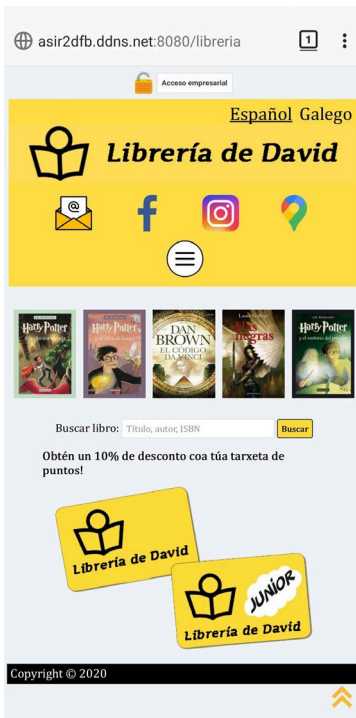
```
function mostrarEventos() {
    try {
        $this->constructor();

        $consulta = "SELECT * FROM eventos ORDER BY cod_evento DESC";
        $resultado=$this->conexion->prepare($consulta);
        $resultado->execute(array());

        foreach ($resultado as $tabla) {
            printf(
                "<h1>".$tabla['titulo']. "</h1>\n"

                <script async defer src="https://connect.facebook.net/es_ES/sdk.js#xfbml=1&version=v3.2">
                </script>
                <div class="fb-post" data-href="" . $tabla['facebook']. " "></div><br><hr><br>"
            );
        }
    } catch (Exception $e) {
        exit("Error: " . $e->getMessage());
    }
}
```





Observación:

Las líneas CSS podían ser menos de haber sabido usar Bootstrap, pero aún no ha sido el caso.

```

1  |@font-face {
2      font-family: 'Cambria';
3      src: url('Cambria.ttf') format('truetype');
4  }
5
6  /* ESTILOS PARA CUALQUIER DISPOSITIVO */
7  ▶ @media all { [...] }
40
41
42 /* ESTILOS PARA MÓVILES EN VERTICAL */
43 ▶ @media screen and (orientation: portrait) { [...] }
356
357
358 /* ESTILOS PARA MÓVILES EN HORIZONTAL */
359 @media screen and
360 (max-width : 1365px) and
361 ▶ (orientation: landscape) { [...] }
595
596
597 /* ESTILOS PARA ORDENADORES Y TELEVISIONES */
598 @media screen and
599 (min-width: 1366px) and
600 ▶ (orientation: landscape) { [...] }

```

MENÚ DESPLEGABLE PARA LA VERSIÓN MÓVIL

Al no usar JavaScript (por falta de conocimientos en un principio), la tarea de realizar un menú desplegable para móviles ha sido algo tediosa, pero no imposible. Hace falta lo siguiente: un `<input>` de tipo *checkbox* y un `<label>` que apunte al *input* con su atributo *for* (y dentro del `<label>` irá una imagen que será la que pulsaremos para desplegar/ocultar el menú). Luego, como siempre, una lista con los elementos correspondientes.

```

<div id=moviles>
  <input type=checkbox id=casilla />
  <label for=casilla class=abrir><img src=../../pictures/menu.png alt=menu/></label>
  <ul id=menu_moviles>
    <li class=elemento><a href=novedades.php>Novedades</a></li>
    <li class=elemento><a href=vendidos.php>Os más vendidos</a></li>
    <li class=elemento><a href=manga.php>Manga/Cómic</a></li>
    <li class=elemento><a href=galego.php>Galego</a></li>
    <li class=elemento><a href=infantil.php>Infantil</a></li>
    <li class=elemento><a href=eventos.php>Eventos</a></li>
    <li class=elemento><a href=contacto.php>Contacto</a></li>
  </ul>\n
</div>

```

Después, mediante **CSS** y el pseudo-selector **:checked** se realizan las siguientes acciones: se personaliza el menú como se haría siempre, con sus colores, rellenos y demás estilos. Ahora queda ocultarlo por defecto para que solo se muestre si se pulsa donde queramos, en este caso en la imagen. Le asignamos (entre otras cosas) un **position: absolute** y un **margin-left: -200%** (de esta forma permanecerá oculto, a la izquierda de toda la web) y luego a ese menú con el selector anteriormente mencionado le decimos que su **margin-left** sea 0. Hablando en plata: tenemos una casilla invisible y vinculada a una imagen, así que cuando pulsamos sobre la imagen es como si

pulsáramos la casilla, y ésta, al tener 2 estados (marcada y desmarcada), puede hacer 2 cosas diferentes, así que la usamos para mostrar/ocultar el menú.

```

188 ▾ #menu_moviles {
189     position: absolute;
190     background-color: #FFDE45;
191     font-family: Cambria;
192     font-size: 70px;
193     width: 97.5%;
194     margin-left: -200%;
195     margin-top: 51%;
196     padding: 0;
197     transition: 0.5s;
198     -webkit-transition: 0.5s;
199     -moz-transition: 0.5s;
200     -o-transition: 0.5s;
201     -ms-transition: 0.5s;
202 }
203
204 ▾ @-moz-document url-prefix() {
205     #menu_moviles {
206         position: absolute;
207         background-color: #FFDE45;
208         font-family: Cambria;
209         font-size: 70px;
210         width: 97.5%;
211         margin-left: -200%;
212         margin-top: 110%;
213         padding: 0;
214         transition: 0.5s;
215         -webkit-transition: 0.5s;
216         -moz-transition: 0.5s;
217         -o-transition: 0.5s;
218         -ms-transition: 0.5s;
219     }
220 }

```

```

222 ▾ #casilla {
223     display: none;
224 }
225
226 ▾ #casilla:checked ~ #menu_moviles {
227     margin-left: 0;
228 }

```

Nótese el pequeño *hack* (**@-moz-document url-prefix()**) que ha sido necesario hacer para que Firefox mostrase el menú en la posición correspondiente, o aparecía más arriba de lo deseado. En todos los demás navegadores, sin embargo, respetaban la cifra original de **margin-top**: para Firefox ha sido necesario especificar un 110%, pero para todos los demás bastó un 51%. De izquierda a derecha, **Firefox** y **Chrome**:



Una vez terminada la web, volveremos al software MobaXterm para copiar la carpeta al servidor en /var/www/html/.

MEJORAS DE ÚLTIMA HORA: APRENDIENDO JAVASCRIPT SOBRE LA MARCHA

Como se indicará en el apartado “Dificultades encontradas” situado al final de este proyecto, por falta de conocimientos del lenguaje JavaScript y de Bootstrap, la web ha llevado más tiempo de realización del que llevaría normalmente, pero eso no ha impedido que, al haber acabado el proyecto solo con PHP y empezado a redactar esta documentación, haya empezado a aprender algunos conceptos básicos y otros no tanto con el que poder aportar mejoras que paso a detallar ahora (siempre usando previamente la librería jQuery en su versión 3.5.1, forma mínima):

```
<script src="jquery-3.5.1.min.js"></script>
<script>
  $(document).ready(function() {
    $(".abrir").click(function() {
      $("#menu_moviles").slideToggle();
      $(".abrir").toggleClass("abrir2");
    });
  });
</script>
```

- El menú desplegable para móviles podría hacerse de forma sencilla tan solo con la función **slideToggle()** y, a mayores para mayor profesionalidad, la

función **toggleClass()** para tener un efecto de cambio de imagen que lo abre/cierra.

- Las barras de búsqueda podrían funcionar con AJAX, lo que permitiría que cargasen su contenido (la consulta SQL) en vivo y sin recargar la página. Y se eliminaría el botón de búsqueda tipo *submit*, dejando solo la barra.

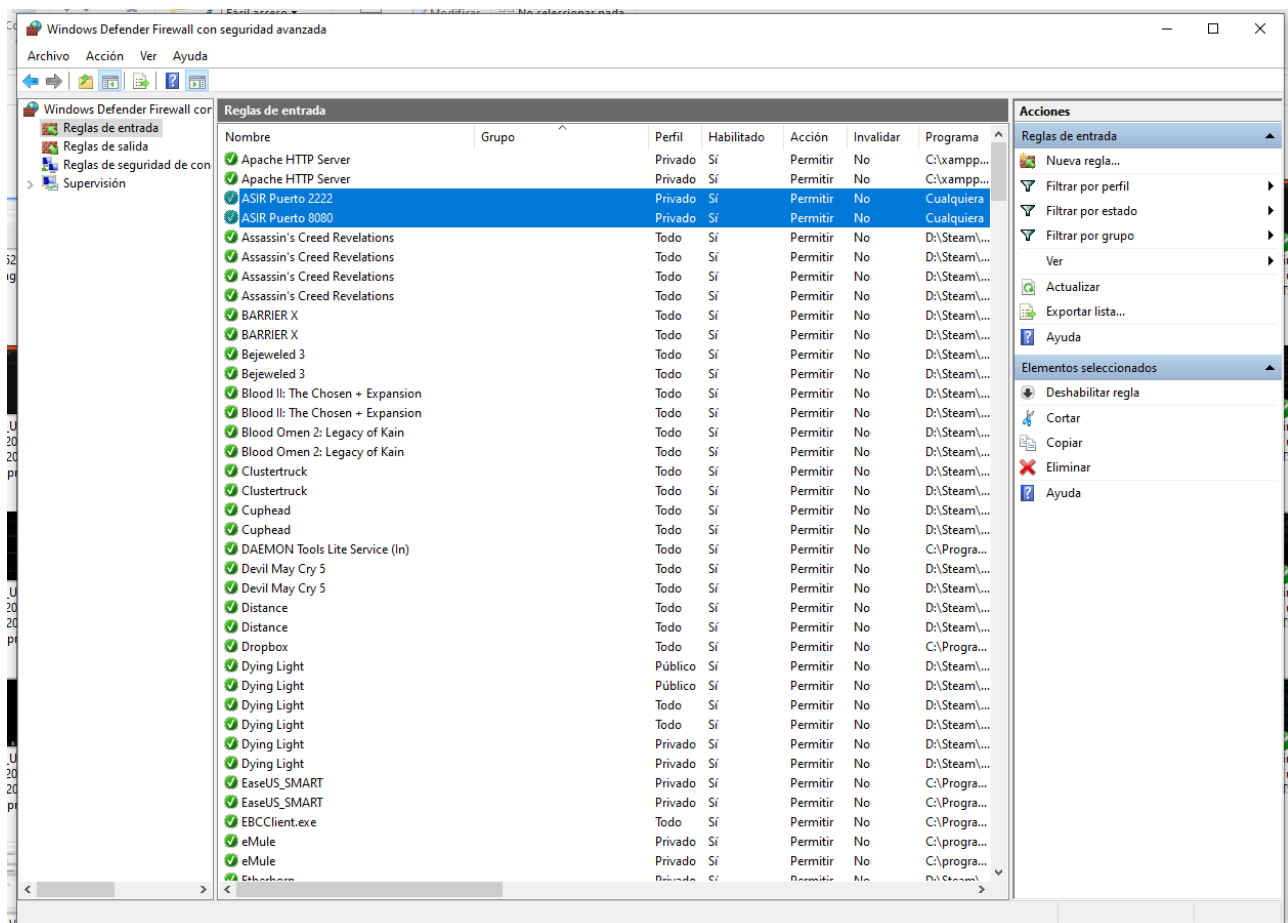
```
1 ▼ $(document).ready(function () {
2
3     var consulta;
4
5 ▼     $("#busqueda").keyup(function (e){
6
7         consulta = $("#busqueda").val();
8
9 ▼     $.ajax({
10         type: "POST",
11         url: "busqueda.php",
12         data: "LibroBuscado="+consulta,
13         dataType: "html",
14 ▼     beforeSend: function(){
15         $("#resultadoBusqueda").html("<p align='center'><img src='../pictures/ajax-loader.gif' />
16         </p>");
17 ▼     },
18     error: function(){
19         alert("error al buscar");
20 ▼     },
21     success: function(data){
22         $("#resultadoBusqueda").empty();
23         $("#resultadoBusqueda").append(data);
24     }
25     });
26 }|
```

SALIDA DEL SERVIDOR WEB AL EXTERIOR

Una vez lista y disponible la web en el servidor (en este caso ubicado en una máquina virtual en VirtualBox, lo que nos obligará a hacer algunos pasos extra), toca hacer que se pueda acceder a él desde fuera de nuestra red LAN. Para ello tendremos que configurar ciertas redirecciones (proceso explicado desde dentro hacia fuera):

1 – Redireccionar 2 puertos de la MV para que ese tráfico concreto del host real le llegue a ella: del 8080 de la real al 80 de la MV y del 2222 de la real al 22 de la MV (configuración ya aplicada y enseñada al final de la configuración del servidor, pues si no, no funcionaría el MobaXterm).

2 – En el host real (en este caso con Windows 10 como S.O.) se deben exceptuar los puertos 2222 y 8080 en el firewall del sistema en las reglas de entrada:



3 – A continuación, debemos configurar la IP de nuestro equipo real en modo estático, ya que el siguiente paso implica redirigir puertos públicos del router hacia algunos privados de nuestro equipo, y hay que especificar la IP:

Propiedades: Protocolo de Internet versión 4 (TCP/IPv4) X

General

Puede hacer que la configuración IP se asigne automáticamente si la red es compatible con esta funcionalidad. De lo contrario, deberá consultar con el administrador de red cuál es la configuración IP apropiada.

Obtener una dirección IP automáticamente

Usar la siguiente dirección IP:

Dirección IP:

Máscara de subred:

Puerta de enlace predeterminada:

Obtener la dirección del servidor DNS automáticamente

Usar las siguientes direcciones de servidor DNS:

Servidor DNS preferido:

Servidor DNS alternativo:

Validar configuración al salir

4 – Ahora debemos acceder a nuestro router y hacer 2 cosas: primero y más sencillo, redirigir el puerto 8080 exterior al 8080 interior. Y lo mismo con el 2222:

Router x +

192.168.8.1/html/virtualserver.html

Español Ayuda admin Desconexión

Inicio Estadísticas SMS Actualizar **Configuración** Más

Configuración rápida

Ajustes de Red Móvil

Ethernet

WLAN

Voz

VoIP

Seguridad

Estado Del Firewall

DHCP

Filtrado de MAC

Filtros IP LAN

→ Servidor Virtual

Aplicaciones Especiales

DMZ Configuración

SIP ALG Configuración

UPnP Configuración

Ajustes de NAT

Filtro de nombres de dominio

DDNS

Modo bridge

Sistema

Política de privacidad (C) 2006-2017 HUAWEI TECHNOLOGIES CO., LTD. Aviso de código abierto

Servidor Virtual

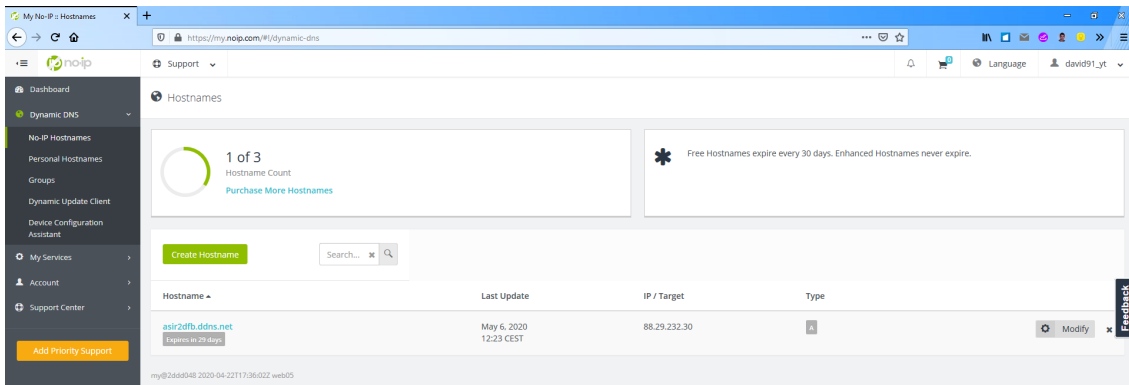
Configure un servidor virtual para permitir a los equipos externos acceder a páginas Web, FTP u otros servicios ofrecidos a través de la LAN.

- Dirección IP:** Identifica a un equipo ubicado en la LAN que proporcionará los servicios.
- Puerto LAN/WAN:** El puerto del ordenador que proporciona los servicios. Es el único puerto y el rango de valores del puerto LAN/WAN es 1-65535.
- Protocolo:** Protocolos que utilizan los servicios.
- Nota:** la configuración no surtirá efecto hasta que no haga clic en el botón "Aplicar".

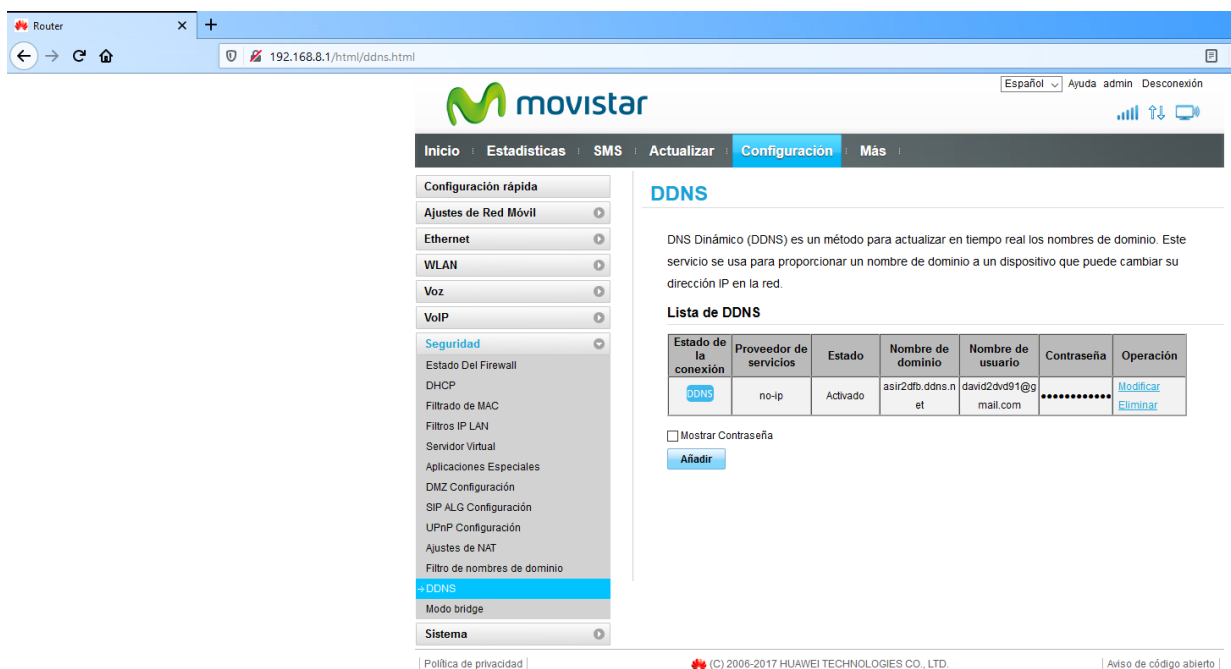
Lista de Servidores Virtuales

Nombre	Puerto WAN	Direcc. IP LAN	Puerto LAN	Protocolo	Estado	Opciones
asir2dfb	8080	192.168.8.25	8080	TCP/UDP	Activado	Modificar Eliminar
SSH	2222	192.168.8.25	2222	TCP/UDP	Activado	Modificar Eliminar

5 – Ahora tocaría vincular la IP pública de nuestro router con una web que se dedique a actuar de servidor DNS para, en este caso, proporcionar un nombre para acceder al router en vez de estar escribiendo su IP constantemente que, por otro lado, cambia cada vez que se reinicia, entre otros sucesos. En este caso se ha utilizado la web <https://www.noip.com/> y se ha creado una cuenta gratuita asignando el router al nombre **asir2dfb.ddns.net**

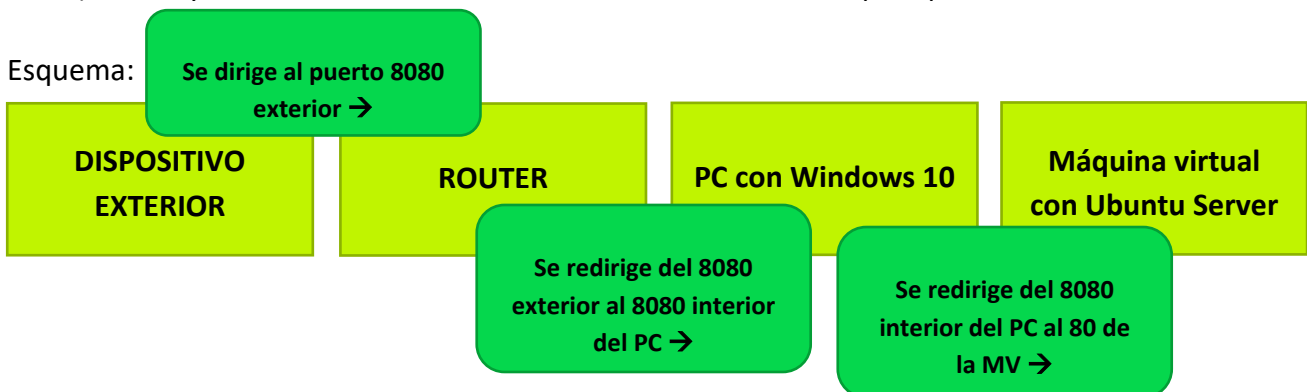


Una vez creada la cuenta, procedemos a ir al apartado de nuestro router sobre DDNS (DNS dinámico) para vincular nuestra cuenta:



Y así a partir de ahora (demostrado en las capturas de pantalla anteriores de la web en versión móvil) la web y nuestro servidor entero será accesible desde cualquier parte del mundo.

Esquema:



DIFICULTADES ENCONTRADAS

- Al no aprender conocimientos de JavaScript ni de Bootstrap, todo el código se ha tenido que realizar 100% a mano, por lo que ha llevado más tiempo del que tomaría en un caso real.
- En clase se aprendió una base de PHP, pero no todos los conocimientos plasmados en este proyecto, por lo que el proceso no ha sido de realización directa, sino de aprendizaje, práctica y error y finalmente la realización.
- Por el camino, por supuesto, han surgido errores y dificultades, pero consultando diversas páginas web se han podido resolver todas las incidencias. Éstas estarán indicadas en la bibliografía.

IMPLEMENTACIÓN DEL PROTOTIPO (SI ES EL CASO)

De momento no es el caso, puesto que habrá que tomar la BD que ya tenga una librería, fruto de su día a día, y hacer la importación correspondiente a este proyecto, pero la idea sí es llevar esto a la vida real más pronto que tarde. Aunque en ese caso no se usaría un hosting propio, sino uno de terceros, junto con un dominio de verdad y no proporcionado por un servidor DNS con sus limitaciones. Y se añadiría un certificado SSL gratuito gracias a CloudFlare.

CONCLUSIONES

- Cualquier negocio, sea grande, mediano o pequeño, debe tener una página web.
- Es costoso en varios sentidos, pero si se tienen las ganas y conocimientos necesarios, se puede hacer algo muy profesional y 100% personalizable.

BIBLIOGRAFÍA, REFERENCIAS Y RECURSOS UTILIZADOS

<https://www.campusmvp.es/>

<https://stackoverflow.com/>

<https://es.stackoverflow.com/>

<https://uniwebsidad.com/>

<https://www.youtube.com/playlist?list=PLU8oAlHdN5BkinrODGXToK9oPAlnJxmW>

<https://www.youtube.com/playlist?list=PLU8oAlHdN5BmpobVmj1lIneKIVLJ84TID>