

INFORME PRÁCTICA CONTROL DE ACCESO Y CONFIGURACIÓN DE APACHE



Autor: Mario Tomé Core

Proyecto utilizado: Cyber-Tracker

Entorno: macOS + MAMP

Fecha: 2 Diciembre, 2025

ÍNDICE DE CONTENIDOS

1. INTRODUCCIÓN.....	3
2. PREPARACIÓN DEL ENTORNO.....	3
2.1. Inicio de MAMP y estado de los servidores.....	3
3. ESTRUCTURA DEL PROYECTO UTILIZADO.....	4
4. CREACIÓN DEL ARCHIVO .htpasswd.....	4
4.1. Creación del directorio seguro.....	4
4.2. Creación del usuario y contraseña.....	5
5. CREACIÓN DE PÁGINAS PERSONALIZADAS DE ERROR.....	5
6. ARCHIVOS .HTACCESS CON TODAS LAS CONFIGURACIONES.....	6
6.1. .htaccess de la carpeta private/ (solo control de acceso).....	6
6.2. .htaccess de la raíz del proyecto (solo redirecciones 301).....	7
7. PRUEBAS DE FUNCIONAMIENTO.....	8
7.1. Comprobación de autenticación.....	8
7.2. Comprobación de la redirección 301.....	10
7.3. Comprobación de las páginas de error personalizadas.....	12
8. VERIFICACIÓN MEDIANTE LOGS.....	13
9. CONCLUSIONES.....	14

1. INTRODUCCIÓN

En esta práctica he trabajado con los mecanismos de control de acceso de Apache: **autenticación** básica con fichero .htpasswd, **restricciones** por dirección IP, **redirecciones** permanentes (301) y **páginas de error** personalizadas

Aunque se planteaba inicialmente usar XAMPP, he realizado toda la práctica con **MAMP**, ya que en macOS este entorno ofrece una configuración más cómoda y totalmente compatible

He comprobado que todas las funcionalidades que exige la práctica funcionan correctamente en MAMP: .htaccess, .htpasswd, mod_rewrite, páginas de error y restricciones

Para llevar a cabo la práctica he utilizado **mi proyecto de DWES Cyber-Tracker**, desarrollado previamente siguiendo un patrón MVC

2. PREPARACIÓN DEL ENTORNO

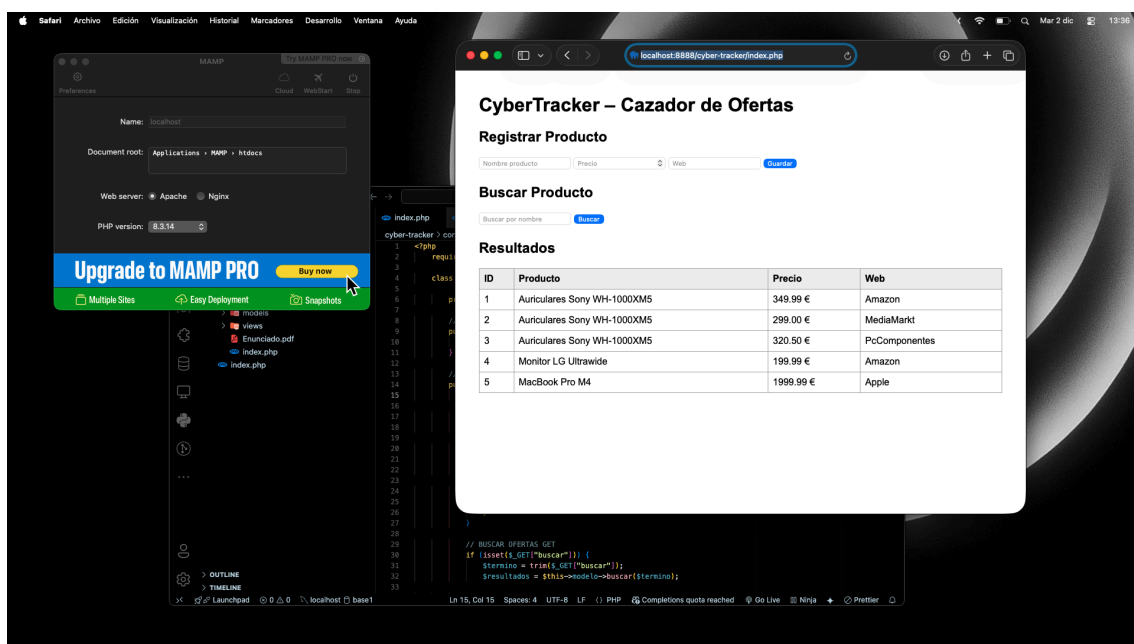
Antes de comenzar, he verificado que MAMP esté funcionando correctamente en mi portátil

2.1. Inicio de MAMP y estado de los servidores

He iniciado la aplicación MAMP y comprobado que Apache y MySQL arrancaban sin errores

La aplicación utiliza el puerto **8888** para Apache, por lo que mi proyecto queda accesible en:

`http://localhost:8888/cyber-tracker/index.php`

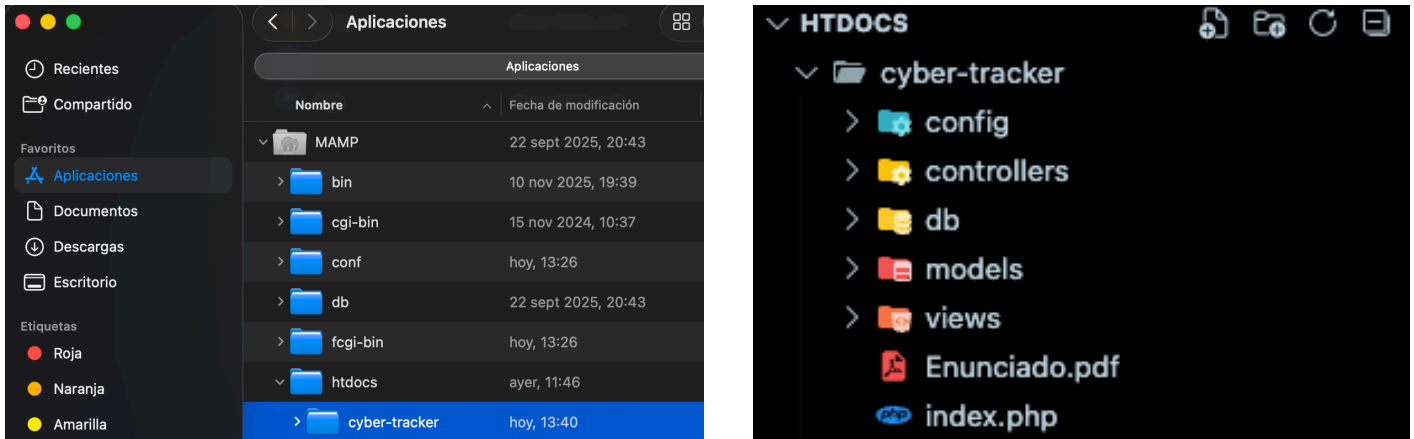


3. ESTRUCTURA DEL PROYECTO UTILIZADO

He copiado mi proyecto MVC **Cyber-Tracker** dentro de la carpeta principal de MAMP:

`/Applications/MAMP/htdocs/cyber-tracker/`

Esto me permite trabajar en un entorno local accesible fácilmente desde el navegador



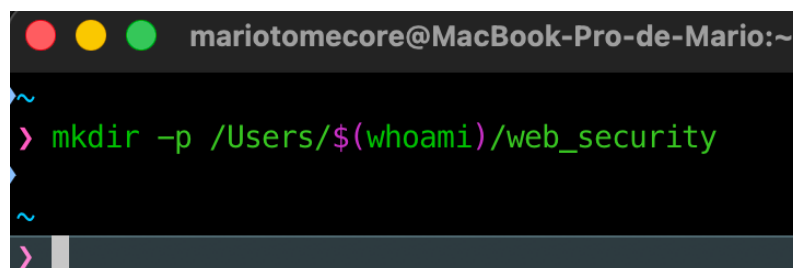
4. CREACIÓN DEL ARCHIVO .htpasswd

Para proteger un directorio mediante autenticación básica he creado un fichero .htpasswd

He decidido almacenarlo **fuera del directorio público** por motivos de seguridad

4.1. Creación del directorio seguro

En Terminal he ejecutado: `mkdir -p /Users/$(whoami)/web_security`



4.2. Creación del usuario y contraseña

He generado un usuario llamado admin con este comando:

```
htpasswd -c /Users/${whoami}/web_security/.htpasswd admin
```

El sistema me ha pedido introducir y confirmar la contraseña (contraseña: 123):

```
> htpasswd -c /Users/${whoami}/web_security/.htpasswd admin
New password:
Re-type new password:
Adding password for user admin

~ 10s
> 
```

5. CREACIÓN DE PÁGINAS PERSONALIZADAS DE ERROR

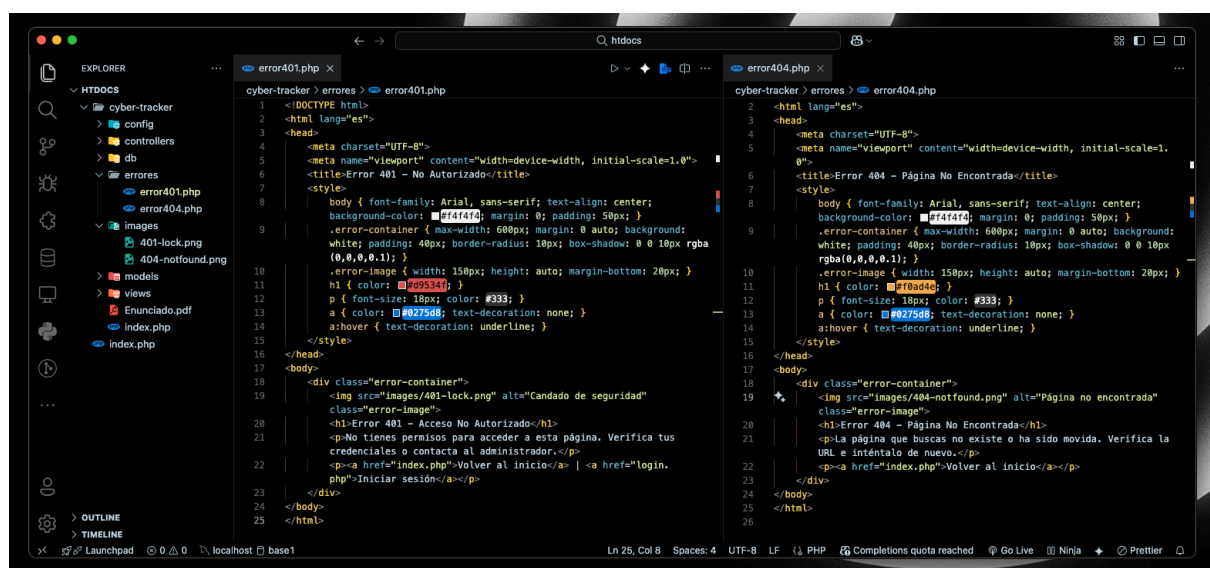
He creado un directorio dentro del proyecto para almacenar las páginas de error:

```
/Applications/MAMP/htdocs/cyber-tracker/errores/
```

Dentro he creado dos ficheros con diseño simple:

- error401.php
- error404.php

Cada uno contiene un pequeño mensaje e imagen indicando el tipo de error.



6. ARCHIVOS .HTACCESS CON TODAS LAS CONFIGURACIONES

Para esta práctica he tenido que configurar varios mecanismos de control en Apache: autenticación, páginas de error, restricciones por IP y redirecciones permanentes.

Aunque inicialmente creé un único archivo .htaccess dentro de la carpeta private, durante las pruebas descubrí que **no todas las directivas deben colocarse en el mismo fichero**, porque Apache solo aplica las reglas del .htaccess dentro del directorio donde se encuentra.

Por este motivo, he dividido la configuración en **dos archivos .htaccess**, cada uno con un propósito claro:

6.1. .htaccess de la carpeta private/ (solo control de acceso)

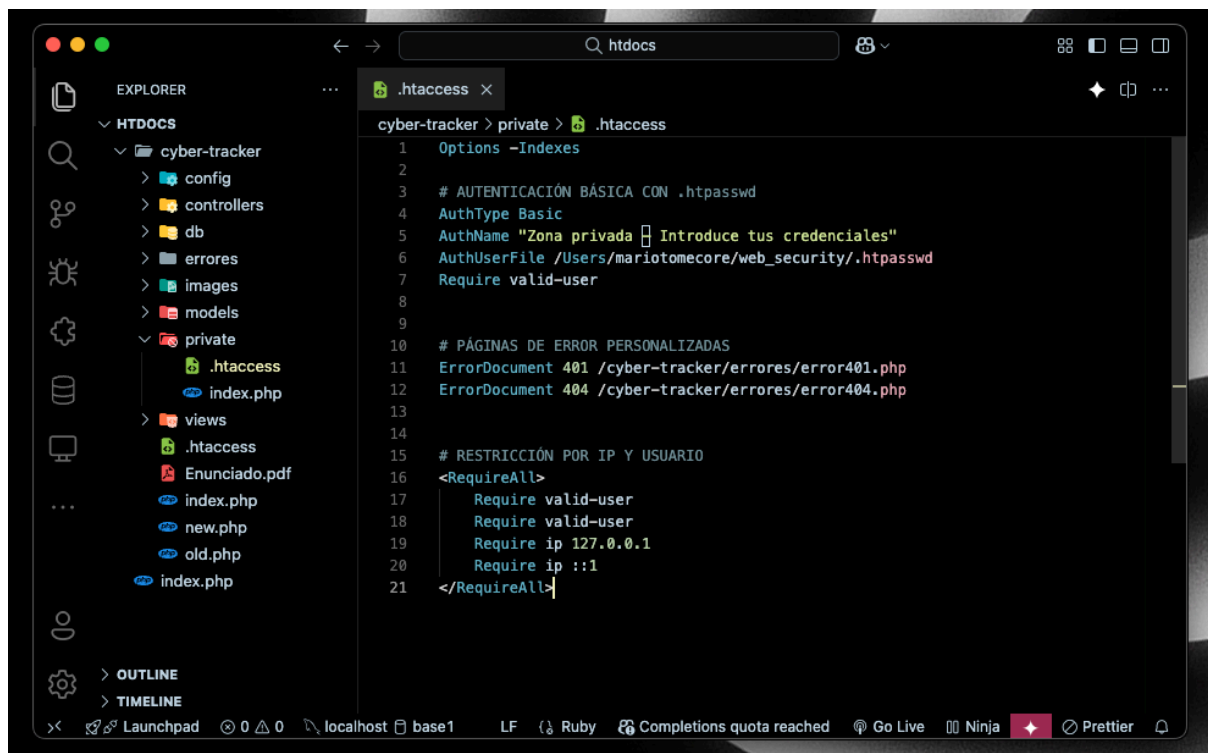
Este archivo controla **exclusivamente** la parte relacionada con la **autenticación básica**, las **páginas de error** y la **restricción por IP**, ya que son reglas que deben aplicarse únicamente cuando el usuario accede a la zona privada del proyecto.

Ubicación: `/Applications/MAMP/htdocs/cyber-tracker/private/.htaccess`

Motivo:

El control de acceso solo debe afectar a la carpeta privada.

Si lo pusiera en la raíz del proyecto, protegería todo el sitio, lo cual no es el objetivo de esta práctica.



```
cyber-tracker > private > .htaccess
1  Options -Indexes
2
3  # AUTENTICACIÓN BÁSICA CON .htpasswd
4  AuthType Basic
5  AuthName "Zona privada Introduce tus credenciales"
6  AuthUserFile /Users/mariotomecore/web_security/.htpasswd
7  Require valid-user
8
9
10 # PÁGINAS DE ERROR PERSONALIZADAS
11 ErrorDocument 401 /cyber-tracker/errores/error401.php
12 ErrorDocument 404 /cyber-tracker/errores/error404.php
13
14
15 # RESTRICCIÓN POR IP Y USUARIO
16 <RequireAll>
17     Require valid-user
18     Require valid-user
19     Require ip 127.0.0.1
20     Require ip ::1
21 </RequireAll>
```

6.2. .htaccess de la raíz del proyecto (solo redirecciones 301)

Las redirecciones permanentes **no pueden colocarse dentro de private/**, porque Apache solo lee ese archivo cuando el acceso es a esa carpeta.

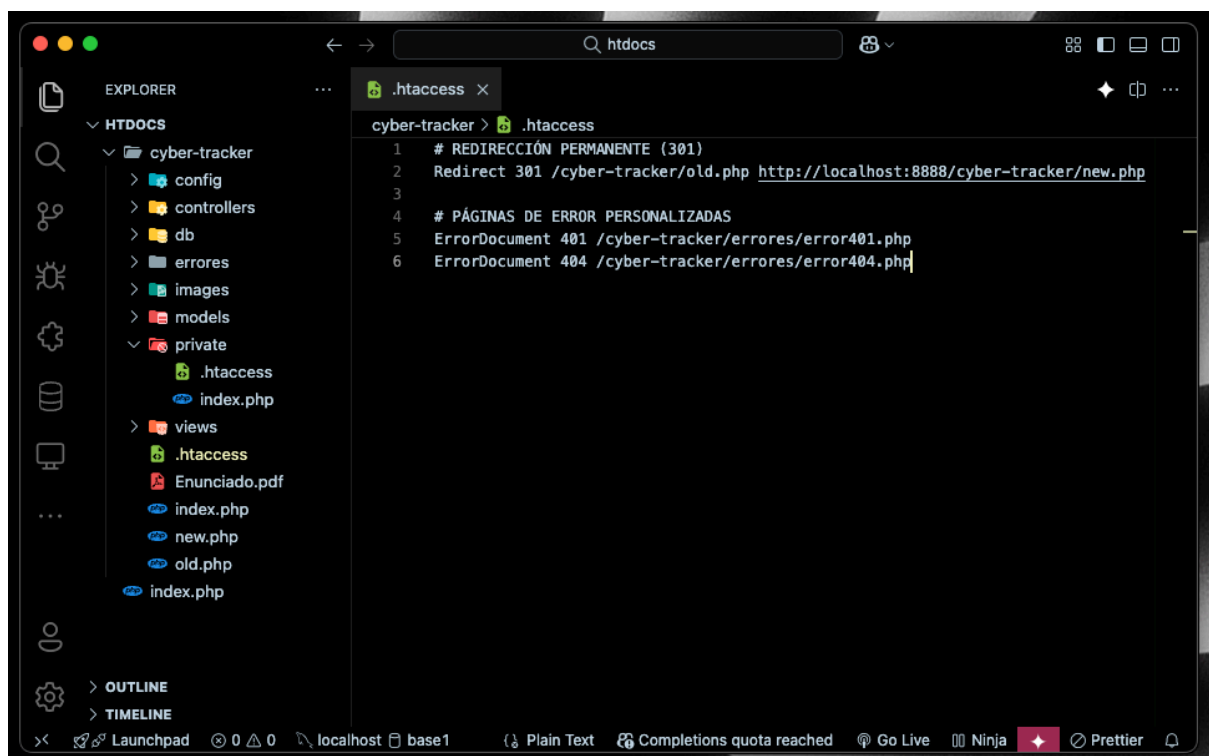
Para que la redirección funcione al acceder a: /cyber-tracker/old.php

La regla debe estar en el **.htaccess de la raíz del proyecto**, ya que afecta a rutas generales del sitio.

Ubicación: /Applications/MAMP/htdocs/cyber-tracker/.htaccess

Motivo:

Este .htaccess controla el comportamiento del **sitio completo**, y por tanto es el lugar correcto para las redirecciones 301.



7. PRUEBAS DE FUNCIONAMIENTO

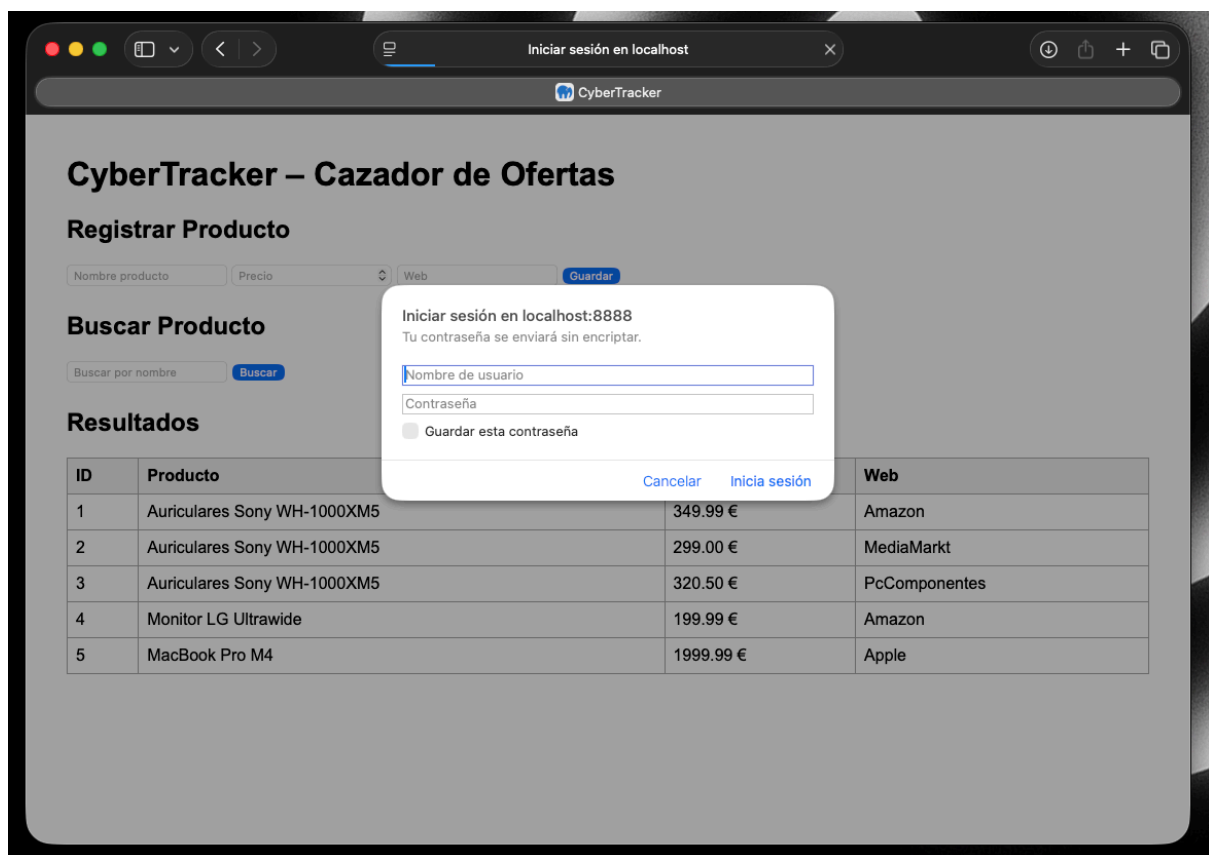
A continuación detallo cada una de las pruebas que he realizado para demostrar que la práctica funciona completamente.

7.1. Comprobación de autenticación

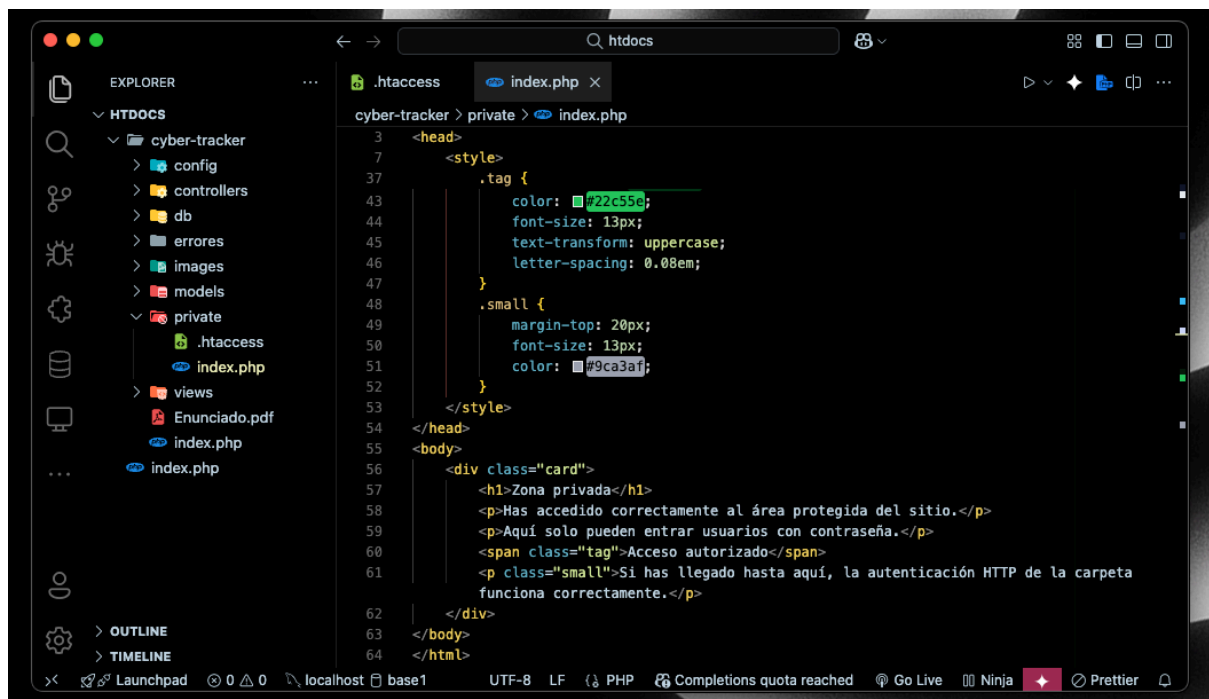
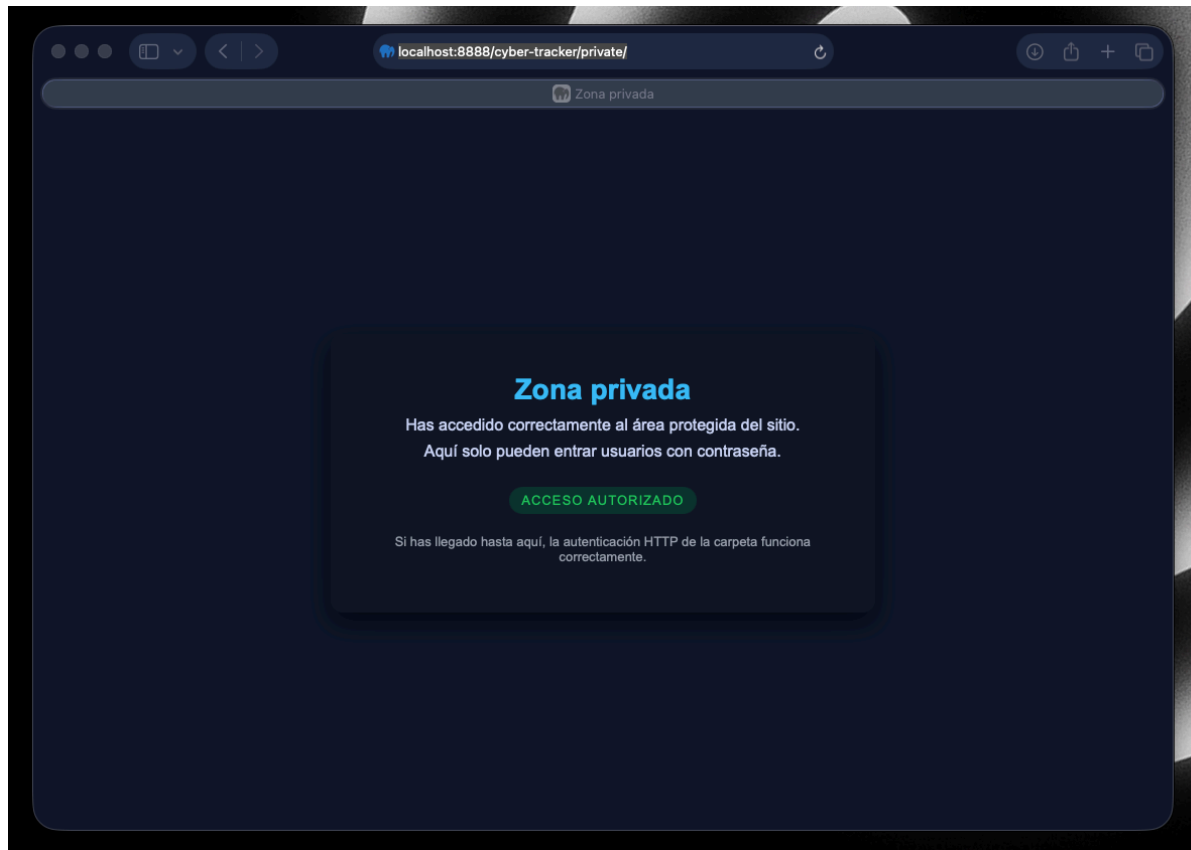
He accedido en el navegador a: `http://localhost:8888/cyber-tracker/private/`

El navegador me ha mostrado un cuadro de diálogo pidiendo usuario y contraseña.

He introducido las credenciales del fichero `.htpasswd` y el acceso ha sido concedido:



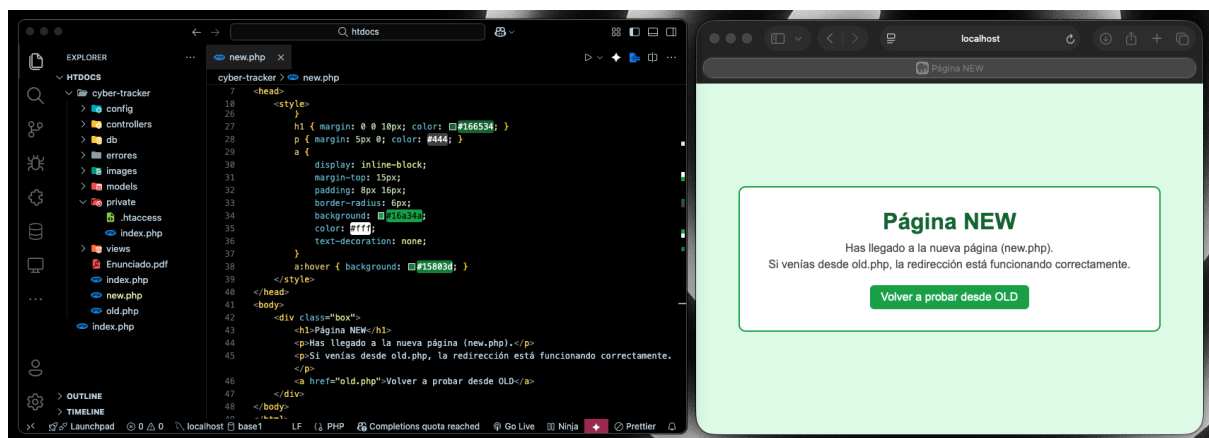
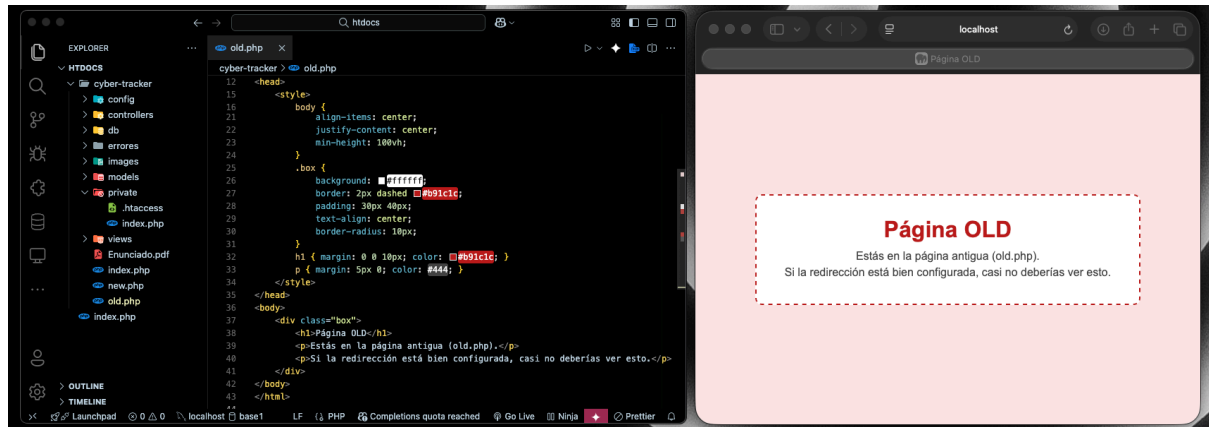
Una vez puestas las credenciales me sale el index.php que he creado para la carpeta privada: `/Applications/MAMP/htdocs/cyber-tracker/private/index.php`



7.2. Comprobación de la redirección 301

He creado los siguientes archivos (he comentado la redirección para mostrar el diseño):

- /cyber-tracker/old.php
- /cyber-tracker/new.php

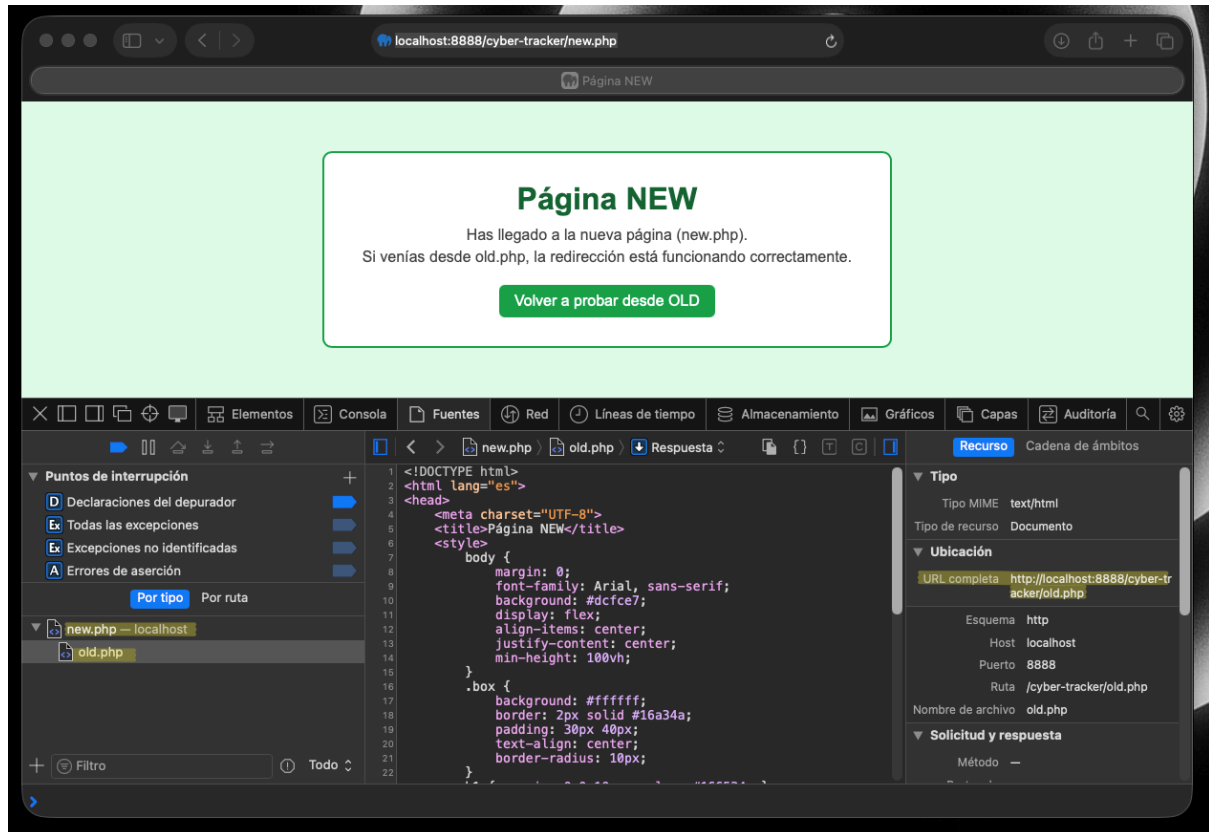


Al entrar en:

`http://localhost:8888/cyber-tracker/old.php`

se me redirige automáticamente a new.php

He verificado el código 301 usando las herramientas de desarrollo del navegador:

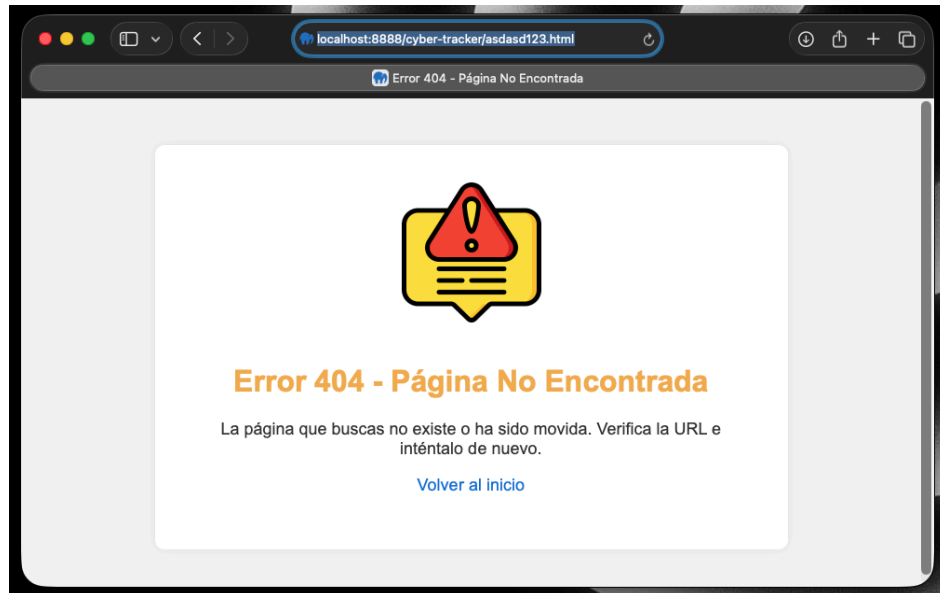


7.3. Comprobación de las páginas de error personalizadas

Error 404

He accedido a: `http://localhost:8888/cyber-tracker/asdasd123.html`

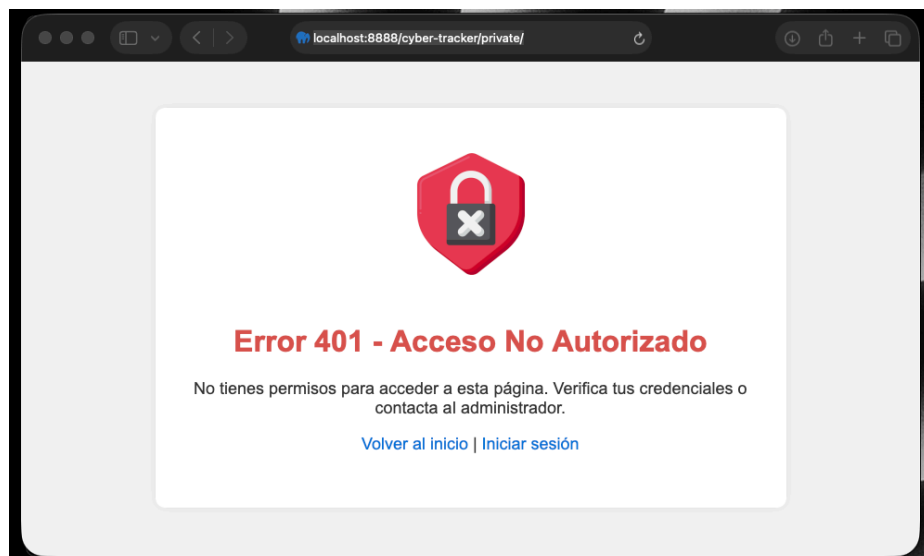
Y ha aparecido mi página error404.php:



Error 401

He accedido a: `http://localhost:8888/cyber-tracker/private/`

Y he escrito credenciales incorrectas a propósito en el cuadro de autenticación. Esto ha cargado correctamente la página error401.php:



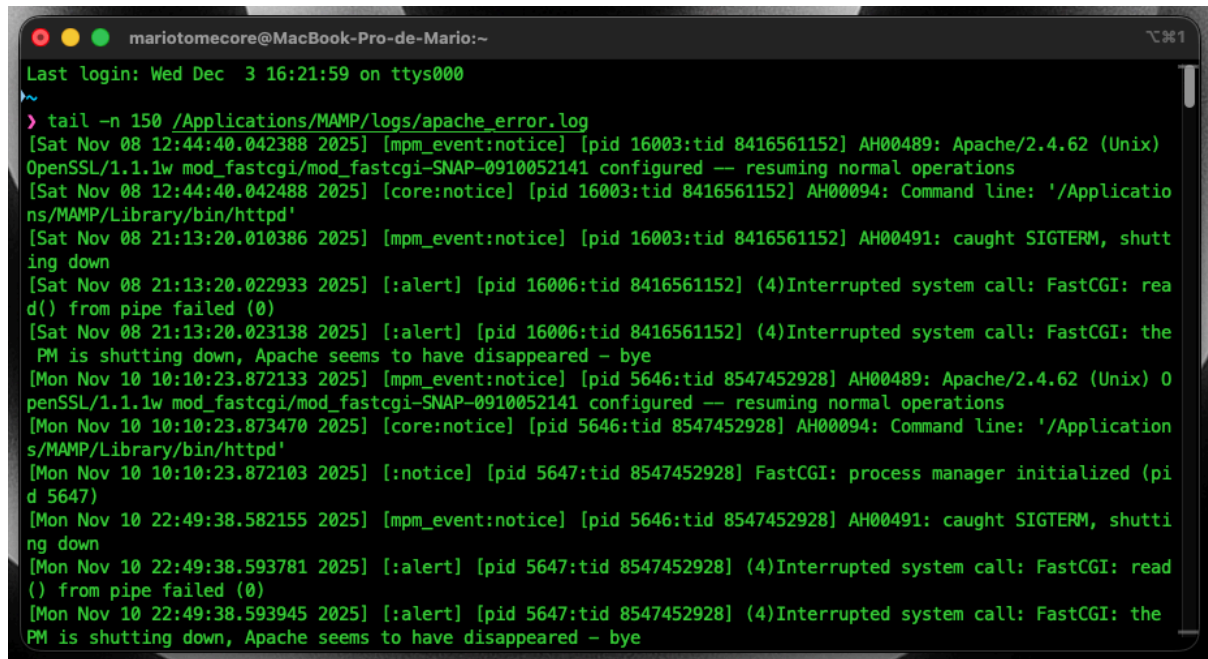
8. VERIFICACIÓN MEDIANTE LOGS

He revisado los logs de Apache para asegurar que no había errores adicionales:

```
tail -n 150 /Applications/MAMP/logs/apache_error.log
```

Los logs muestran las líneas correspondientes a:

- Autenticación correcta / fallida
- Redirecciones 301



```
mariotomecore@MacBook-Pro-de-Mario:~  
Last login: Wed Dec 3 16:21:59 on ttys000  
> tail -n 150 /Applications/MAMP/logs/apache_error.log  
[Sat Nov 08 12:44:40.042388 2025] [mpm_event:notice] [pid 16003:tid 8416561152] AH00489: Apache/2.4.62 (Unix) OpenSSL/1.1.1w mod_fastcgi/mod_fastcgi-SNAP-0910052141 configured — resuming normal operations  
[Sat Nov 08 12:44:40.042488 2025] [core:notice] [pid 16003:tid 8416561152] AH00094: Command line: '/Applications/MAMP/Library/bin/httpd'  
[Sat Nov 08 21:13:20.010386 2025] [mpm_event:notice] [pid 16003:tid 8416561152] AH00491: caught SIGTERM, shutting down  
[Sat Nov 08 21:13:20.022933 2025] [:alert] [pid 16006:tid 8416561152] (4)Interrupted system call: FastCGI: read() from pipe failed (0)  
[Sat Nov 08 21:13:20.023138 2025] [:alert] [pid 16006:tid 8416561152] (4)Interrupted system call: FastCGI: the PM is shutting down, Apache seems to have disappeared - bye  
[Mon Nov 10 10:10:23.872133 2025] [mpm_event:notice] [pid 5646:tid 8547452928] AH00489: Apache/2.4.62 (Unix) OpenSSL/1.1.1w mod_fastcgi/mod_fastcgi-SNAP-0910052141 configured — resuming normal operations  
[Mon Nov 10 10:10:23.873470 2025] [core:notice] [pid 5646:tid 8547452928] AH00094: Command line: '/Applications/MAMP/Library/bin/httpd'  
[Mon Nov 10 10:10:23.872103 2025] [:notice] [pid 5647:tid 8547452928] FastCGI: process manager initialized (pid 5647)  
[Mon Nov 10 22:49:38.582155 2025] [mpm_event:notice] [pid 5646:tid 8547452928] AH00491: caught SIGTERM, shutting down  
[Mon Nov 10 22:49:38.593781 2025] [:alert] [pid 5647:tid 8547452928] (4)Interrupted system call: FastCGI: read() from pipe failed (0)  
[Mon Nov 10 22:49:38.593945 2025] [:alert] [pid 5647:tid 8547452928] (4)Interrupted system call: FastCGI: the PM is shutting down, Apache seems to have disappeared - bye
```

9. CONCLUSIONES

He completado toda la práctica utilizando MAMP en macOS, demostrando que es totalmente compatible con los requerimientos del módulo de *Despliegue de Aplicaciones Web*

He configurado correctamente:

- Autenticación básica con .htpasswd
- Redirecciones permanentes 301
- Páginas de error personalizadas
- Uso de .htaccess
- Estructura ordenada del proyecto
- Logs de Apache como prueba adicional

Todas las pruebas han funcionado y he documentado el proceso con capturas claras para su entrega. Demostrando comprensión completa del funcionamiento de Apache, .htaccess y controles de acceso del servidor