

Servidor Web

Servidor HTTP Apache

¿Qué es un servidor Web y para qué sirve?

- Un servidor web es un software que forma parte del servidor y tiene como misión principal devolver información (páginas) cuando recibe peticiones por parte de los usuarios.
- En otras palabras, es el software que permite que los usuarios que quieren ver una página web en su navegador puedan hacerlo.

Funcionamiento de un servidor web

- Para el funcionamiento correcto de un servidor web necesitamos un cliente web que realice una petición http o https a través de un navegador como Chrome, Firefox o Safari y un servidor donde esté almacenada la información.
- El proceso sería el siguiente:
 - Tras la primera consulta por parte del usuario hacia una web, se establece una conexión entre el servidor DNS y el ordenador que realiza la consulta o petición. Este servidor DNS responde con la dirección IP correcta del servidor web donde está alojado el contenido solicitado.

Funcionamiento de un servidor web

- El siguiente paso sería solicitar el contenido al servidor web mediante el protocolo HTTP/HTTPS.
- Una vez que el servidor web ha recibido la solicitud del contenido solicitado por el cliente web, deberá procesar la solicitud hasta encontrar el contenido solicitado dentro del dominio correspondiente.
- Envía el contenido solicitado al cliente web que lo solicitó.

Funcionamiento de un servidor web

- El siguiente paso sería solicitar el contenido al servidor web mediante el protocolo HTTP/HTTPS.
- Una vez que el servidor web ha recibido la solicitud del contenido solicitado por el cliente web, deberá procesar la solicitud hasta encontrar el contenido solicitado dentro del dominio correspondiente.
- Envía el contenido solicitado al cliente web que lo solicitó.

¿Cuál es la diferencia entre servidor web y hosting?

- Los servidores permiten alojar contenidos que luego son distribuidos de múltiples maneras, la más común hoy día son las páginas web.
- La diferencia entre servidor web y hosting más destacada esta en el presupuesto disponible y los conocimientos para administrar o no un servicio de esta naturaleza con lo que eso conlleva y gestionar adecuadamente los recursos disponibles.
- En base a esto la pregunta adecuada sería ¿para qué vas a utilizar el servicio de alojamiento web? y con base en eso y a tu presupuesto y conocimientos en administración de sistemas.

Tipos de servidores Web

- Servidor HTTP Apache
 - El servidor HTTP Apache es un software de código abierto y gratuito, que durante años ha sido el software más utilizado por los servidores web, rivalizando actualmente con Nginx.
 - Además destaca por ser multiplataforma, siendo compatible con sistemas operativos como Linux, Window, IOS, Unix, etc.

Tipos de servidores Web

- Servidor Nginx
 - Al igual que con Apache estaríamos ante un software de código abierto y multiplataforma compatible con Window, IOS, Linux, etc. Tiene una versión de pago además de la gratuita.
 - Este tipo de servidor web es conocido por su buen funcionamiento cuando tiene que gestionar un número alto de visitas simultáneas, ya que los usuarios no perciben retraso en la carga de la página aunque se esté produciendo ese acceso concurrente.

Tipos de servidores Web

- Servidor LiteSpeed
 - LiteSpeed nació como sustituto de Apache, para mejorar el rendimiento del servidor web en entornos de alto tráfico.
 - Tiene una versión gratuita y otra de pago al igual que Nginx.
 - LiteSpeed puede ser utilizado como reemplazo directo de Apache, ya que ofrece compatibilidad con los archivos .htaccess y las aplicaciones web diseñadas para Apache.

Tipos de servidores Web

- Servidor Microsoft IIS

- Este tipo de servidor web está desarrollado por Microsoft y, en el caso de que se necesite integrar herramientas de Microsoft, sería una alternativa recomendable por la compatibilidad que obtendríamos.
- Se ejecuta con Windows gracias a la tecnología IIS (Internet Information Services) y es compatible con páginas programadas en ASP (Active Server Pages) o .NET, a diferencia del resto de servidores web compatibles con Linux.

Tipos de servidores Web

- Sun Java System Web Server
 - Este tipo de servidor está pensado para programadores que trabajan con Java, Python o Ruby.
 - Es un servidor web creado para soportar una gran carga de trabajo con tecnologías muy específicas como Java.

XAMPP

Es un paquete de software gratuito y de código abierto que permite instalar y configurar rápidamente un entorno de desarrollo web en local. Su nombre es un acrónimo de las tecnologías incluidas en el paquete:

- X: Cross-platform (multiplataforma, porque XAMPP es compatible con Windows, macOS y Linux).
- A: Apache (servidor web).M: MySQL o MariaDB (sistema de gestión de bases de datos).

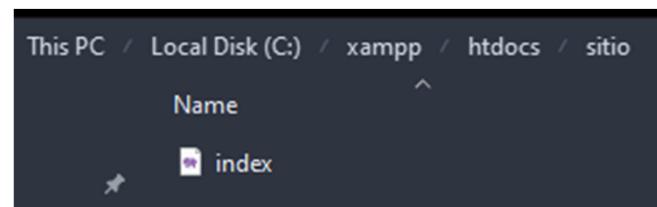
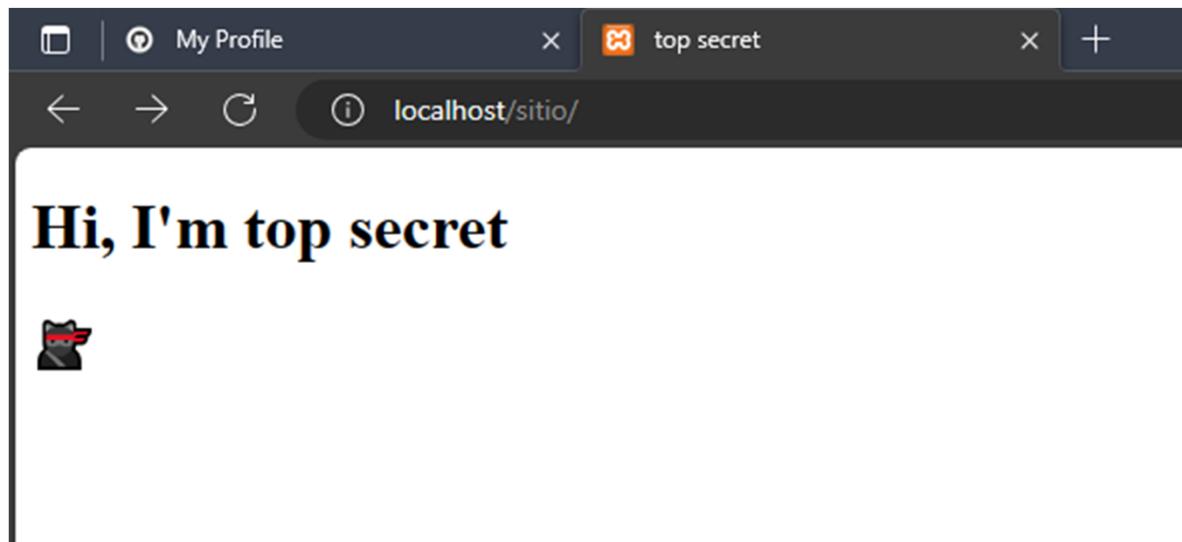
XAMPP

- P: PHP (lenguaje de programación para el desarrollo web).
- P: Perl (otro lenguaje de programación, menos utilizado hoy en día, pero también compatible).
- phpMyAdmin: Una interfaz gráfica para administrar MySQL o MariaDB, facilitando la gestión de bases de datos, tablas y registros sin necesidad de usar la línea de comandos.

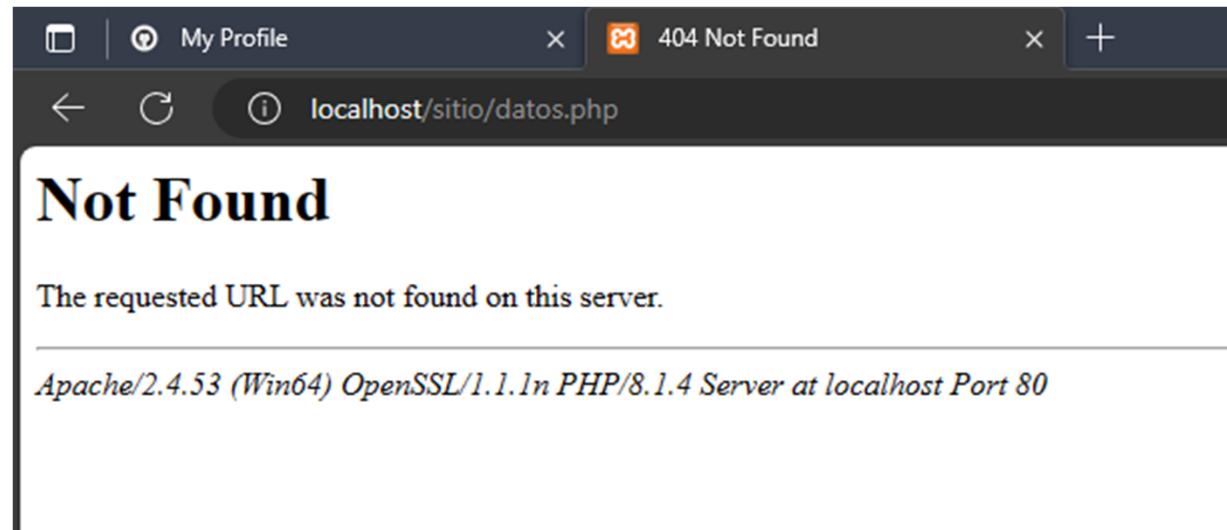
¿Para qué se usa XAMPP?

- Desarrollar sitios web dinámicos y aplicaciones web con tecnologías como PHP y MySQL.
- Probar aplicaciones web sin necesidad de una conexión a Internet.
- Aprender y practicar desarrollo web en un entorno seguro y controlado.
- Configurar entornos de desarrollo local para CMS populares como WordPress, Joomla, y Drupal.

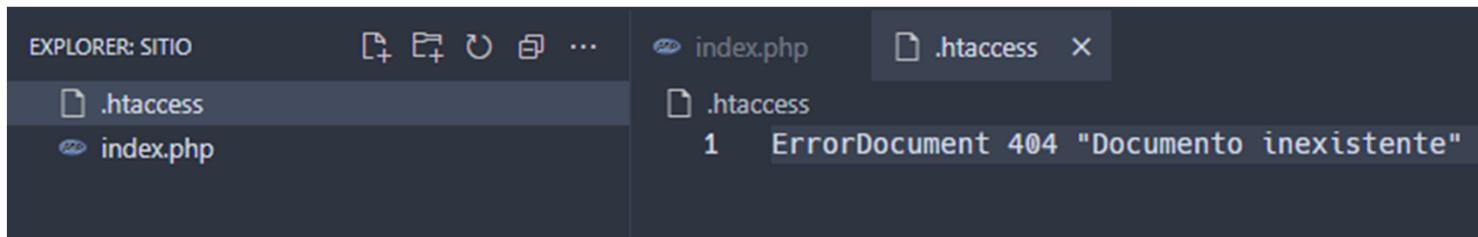
Ejemplos prácticos {404}



Ejemplos prácticos {404}

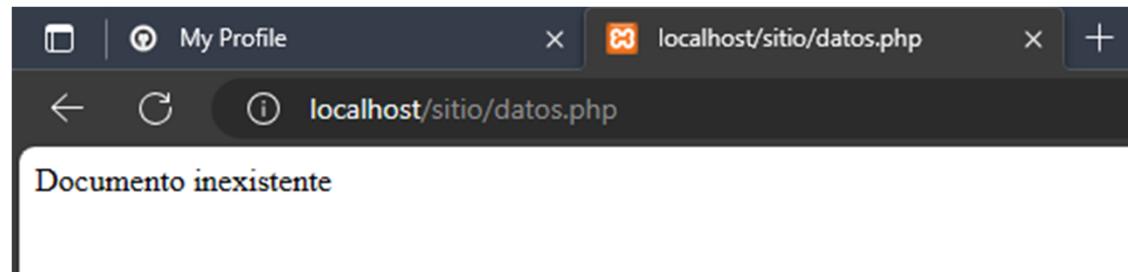


Ejemplos prácticos {404}

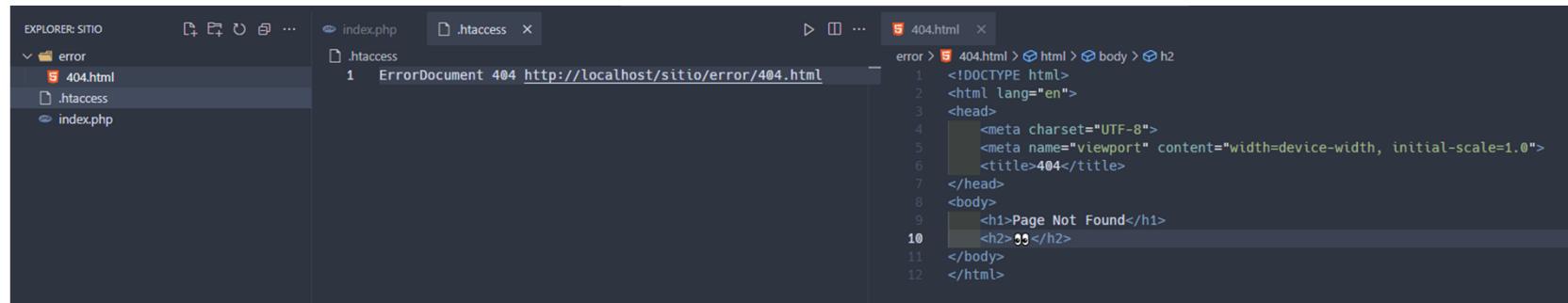


The screenshot shows a code editor interface with two tabs: "index.php" and ".htaccess". The ".htaccess" tab is active, displaying the following content:

```
1 ErrorDocument 404 "Documento inexistente"
```



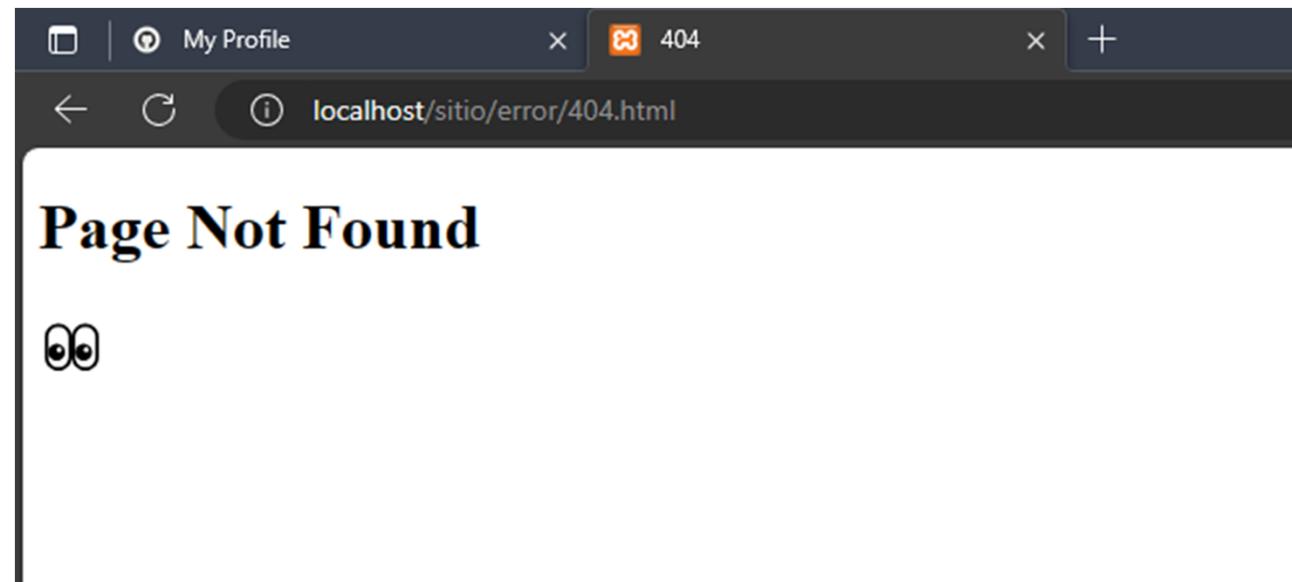
Ejemplos prácticos {404}



The screenshot shows a code editor interface with the following details:

- EXPLORER: SITIO**: Shows files: index.php, .htaccess, 404.html.
- 404.html**: Contains the following code:

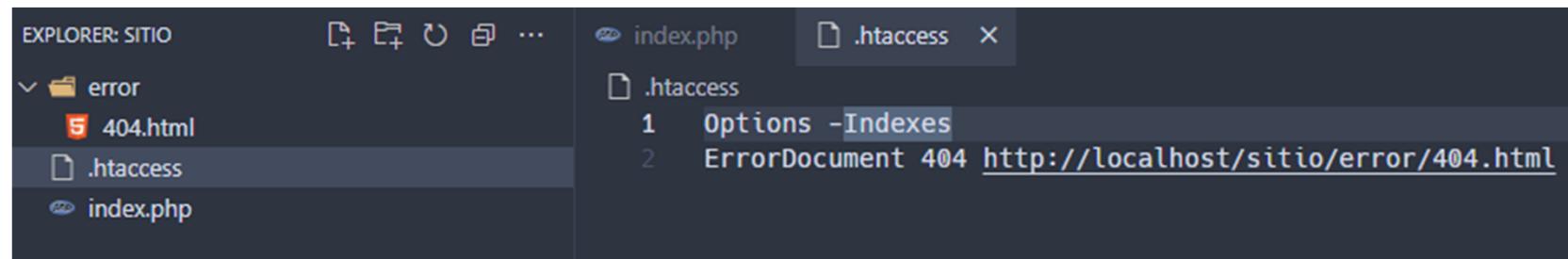
```
1  ErrorDocument 404 http://localhost/sitio/error/404.html
error > 404.html > html > body > h2
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>404</title>
7  </head>
8  <body>
9      <h1>Page Not Found</h1>
10     <h2>oo</h2>
11  </body>
12 </html>
```



Ejemplos prácticos {403}



Ejemplos prácticos {403}



The screenshot shows a code editor interface with two tabs: "index.php" and ".htaccess". The ".htaccess" tab is active, displaying the following content:

```
1 Options -Indexes
2 ErrorDocument 404 http://localhost/sitio/error/404.html
```

On the left, there is an "EXPLORER: SITIO" sidebar showing a folder structure under "error" containing "404.html" and ".htaccess", along with "index.php".



Ejemplos prácticos {403}

The image shows a code editor interface with two tabs open: '.htaccess' and '403.html'. The '.htaccess' tab contains the following configuration:

```
Options -Indexes
ErrorDocument 404 http://localhost/sitio/error/404.html
ErrorDocument 403 http://localhost/sitio/error/403.html
```

The '403.html' tab displays the source code for the error page:

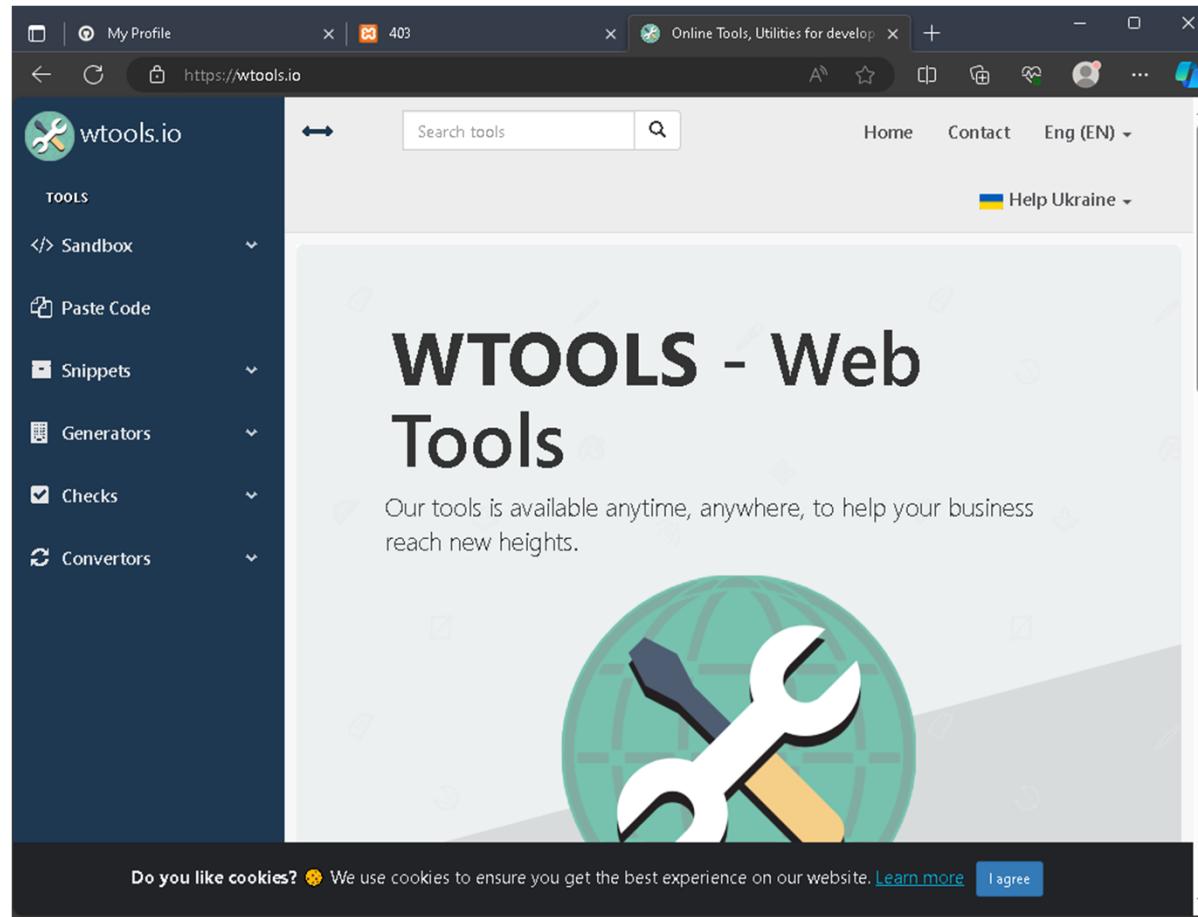
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>403</title>
</head>
<body>
    <h1>You don't have permission to access this resource</h1>
    <h2>😢</h2>
</body>
</html>
```

Below the code editor is a screenshot of a web browser window. The address bar shows 'localhost/sitio/error/403.html'. The page content is:

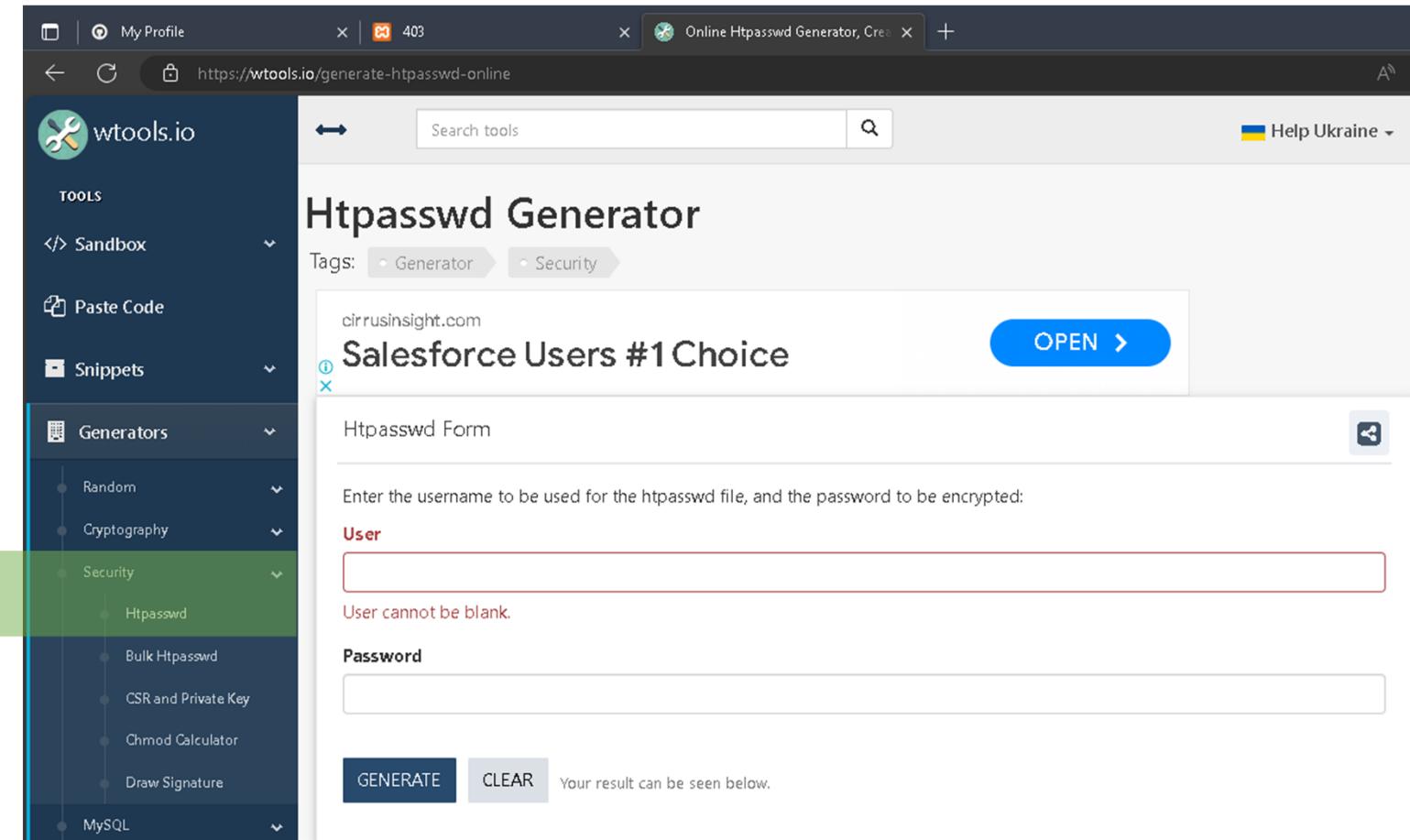
You don't have permission to access this resource

😢

Ejemplos prácticos {<https://wtools.io/>}



Ejemplos prácticos {<https://wtools.io/>}



The screenshot shows a web browser window with three tabs open:

- My Profile
- 403
- Online Htpasswd Generator, Crea

The main content area displays the "Htpasswd Generator" tool from <https://wtools.io/generate-htpasswd-online>. The sidebar on the left lists various tools under categories like "TOOLS", "Sandbox", "Paste Code", "Snippets", and "Generators". The "Generators" category is expanded, showing options such as Random, Cryptography, Security (which is currently selected), Htpasswd, Bulk Htpasswd, CSR and Private Key, Chmod Calculator, Draw Signature, and MySQL. A green bar highlights the "Security" section.

The main form for generating an Htpasswd file has two input fields: "User" and "Password". Below the "User" field is an error message: "User cannot be blank.". At the bottom of the form are "GENERATE" and "CLEAR" buttons, along with a note: "Your result can be seen below."

A promotional banner for cirusinsight.com is visible, stating "Salesforce Users #1 Choice" with a blue "OPEN >" button.

Ejemplos prácticos {<https://wtools.io/>}

Htpasswd Form

Enter the username to be used for the htpasswd file, and the password to be encrypted:

User

pedrito

Password

1234

GENERATE

CLEAR

Your result can be seen below.

Ejemplos prácticos

{<https://wtools.io/>}

Result of htpasswd generation

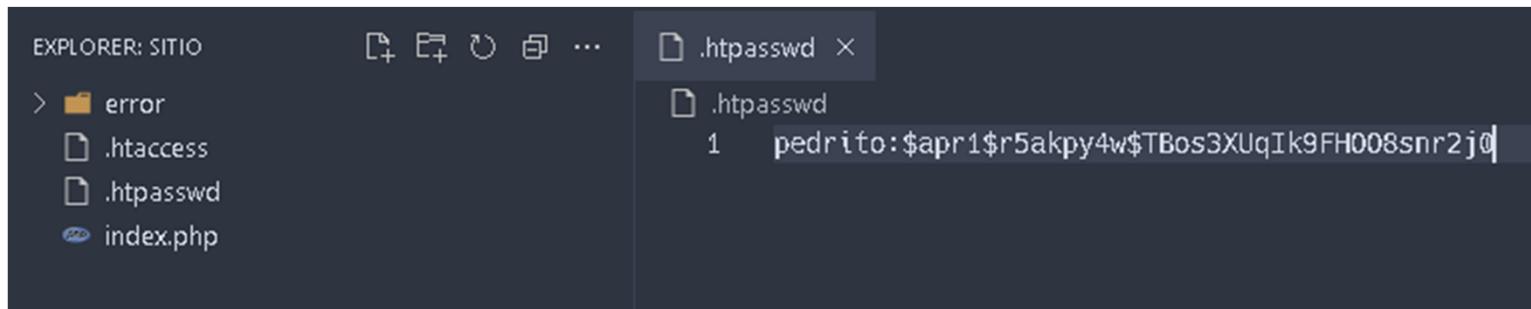
```
pedrito:$apr1$r5akpy4w$TBos3XUqIk9FHOO8snr2j0
```

Copy and paste this into your **.htpasswd** file

COPY

Ejemplos prácticos

{<https://wtools.io/>}



How to protect folders in apache?

To protect a specific folder, a `.htaccess` file is placed in the directory you want the contents of the file to affect. The rules and configuration directives in the `.htaccess` file will be enforced on whatever directory it is in and all sub-directories as well. A typical `.htaccess` file looks like the following:

```
AuthUserFile /path/to/.htpasswd
AuthType Basic
AuthName "My restricted Area"
Require valid-user
```

Ejemplos prácticos

{<https://wtools.io/>}

The image shows a code editor window and a web browser window side-by-side.

Code Editor (.htaccess File):

```
1 Options -Indexes
2 ErrorDocument 404 http://localhost/sitio/error/404.html
3 ErrorDocument 403 http://localhost/sitio/error/403.html
4
5 AuthUserFile C:/xampp/htdocs/sitio/.htpasswd
6 AuthType Basic
7 AuthName "My restricted Area"
8 Require valid-user
```

Web Browser:

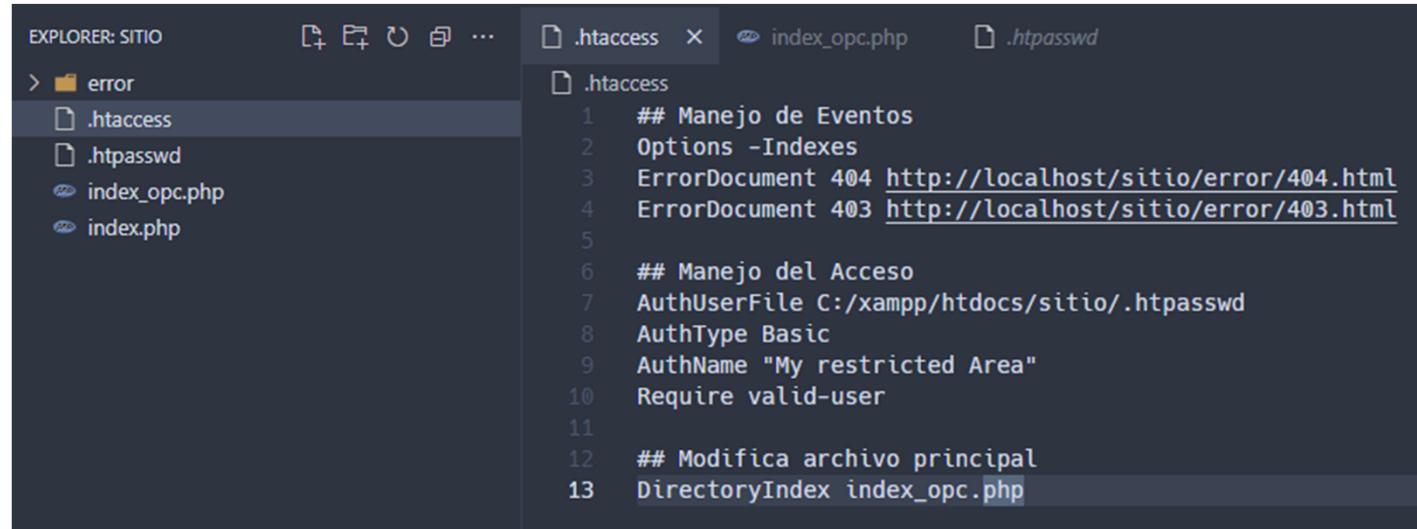
- Address bar: localhost/sitio/
- Content area: A modal dialog box titled "Sign in to access this site" with the message "Authorization required by http://localhost". It contains fields for "Username" and "Password", and buttons for "Sign in" and "Cancel".

Ejemplos prácticos {index}



The screenshot shows a code editor interface with two tabs: ".htaccess" and "index_opc.php". The "index_opc.php" tab is active, displaying the following PHP code:

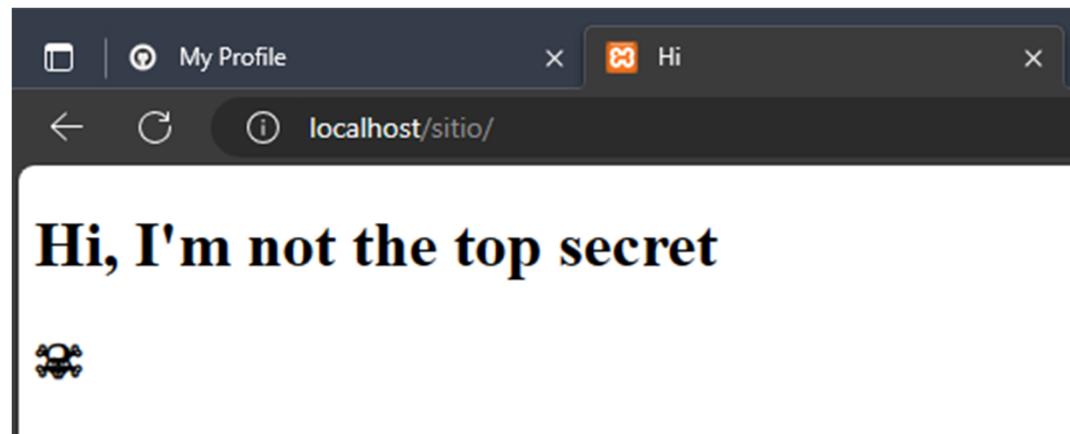
```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Hi</title>
7 </head>
8 <body>
9   <h1>Hi, I'm not the top secret</h1>
10  <h2>Hi</h2>
11 </body>
12 </html>
```



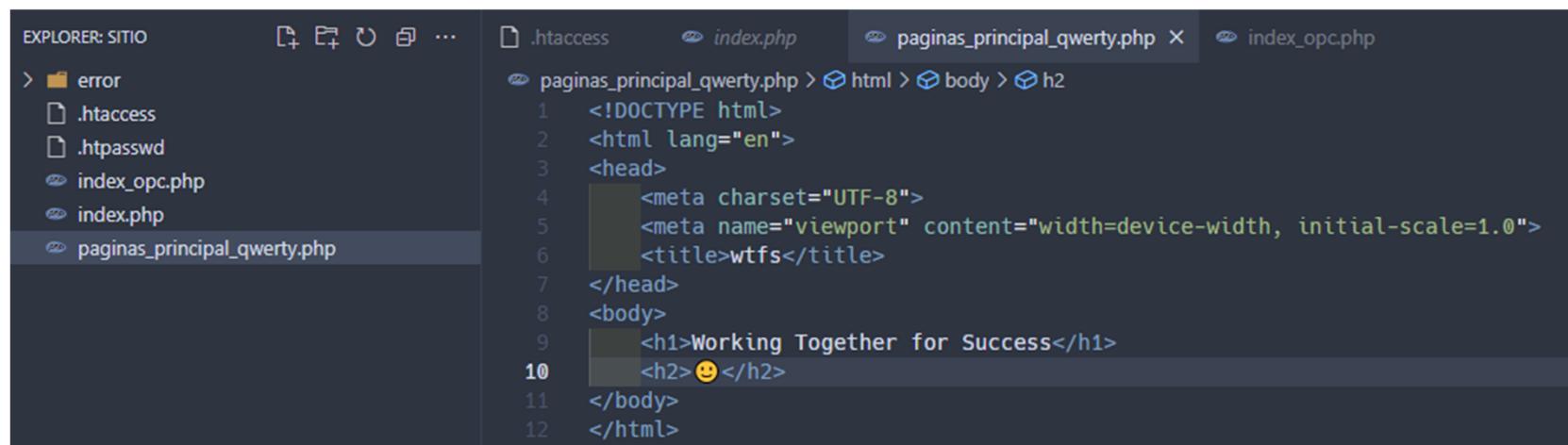
The screenshot shows a code editor interface with three tabs: ".htaccess", "index_opc.php", and ".htpasswd". The ".htaccess" tab is active, displaying the following Apache configuration code:

```
1 ## Manejo de Eventos
2 Options -Indexes
3 ErrorDocument 404 http://localhost/sitio/error/404.html
4 ErrorDocument 403 http://localhost/sitio/error/403.html
5
6 ## Manejo del Acceso
7 AuthUserFile C:/xampp/htdocs/sitio/.htpasswd
8 AuthType Basic
9 AuthName "My restricted Area"
10 Require valid-user
11
12 ## Modifica archivo principal
13 DirectoryIndex index_opc.php
```

Ejemplos prácticos {index}



Ejemplos prácticos {url's}

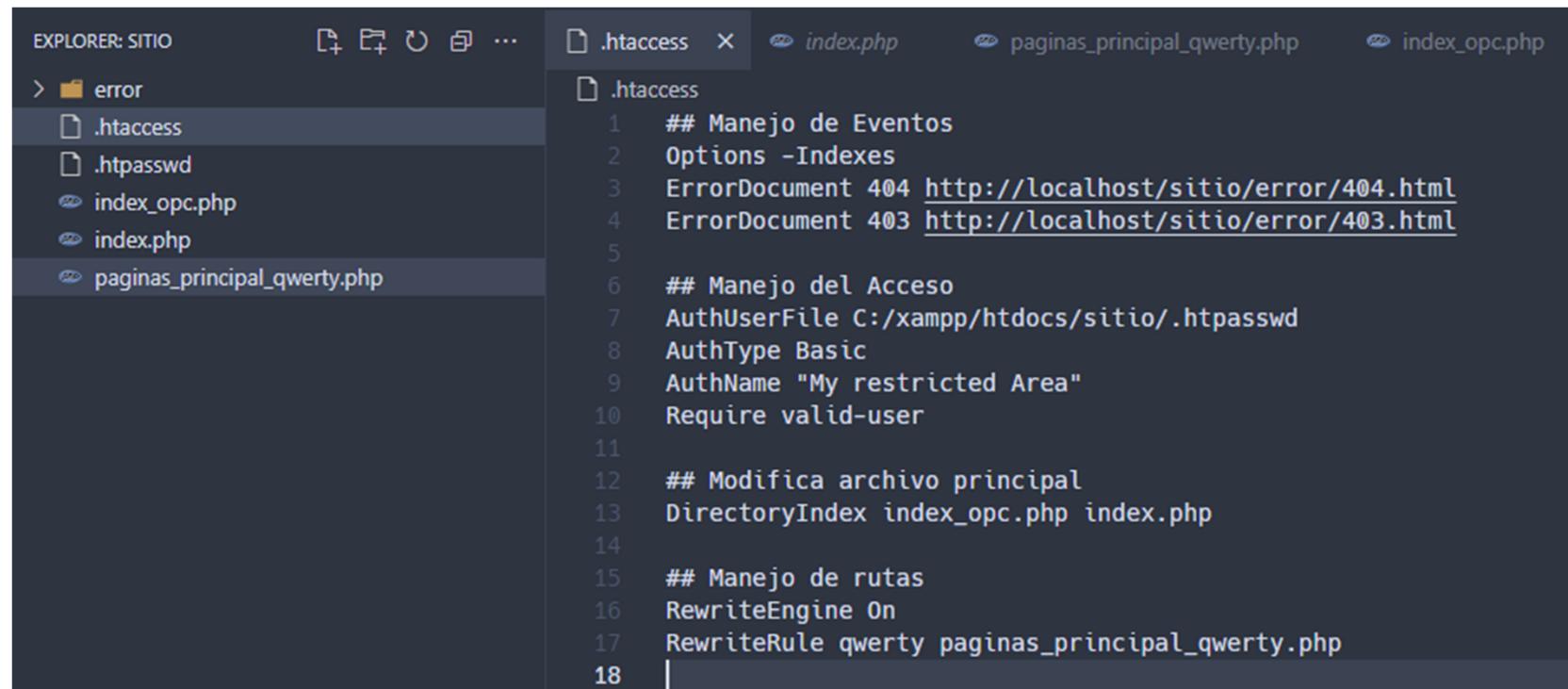


The screenshot shows a code editor interface with the following details:

- EXPLORER: SITIO** pane on the left lists files: error, .htaccess, .htpasswd, index_opc.php, index.php, and paginas_principal_qwerty.php.
- Editor Area**:
 - File tabs: .htaccess, index.php, paginas_principal_qwerty.php (active), and index_opc.php.
 - Code content for paginas_principal_qwerty.php:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>wtfs</title>
7 </head>
8 <body>
9   <h1>Working Together for Success</h1>
10  <h2>😊</h2>
11 </body>
12 </html>
```

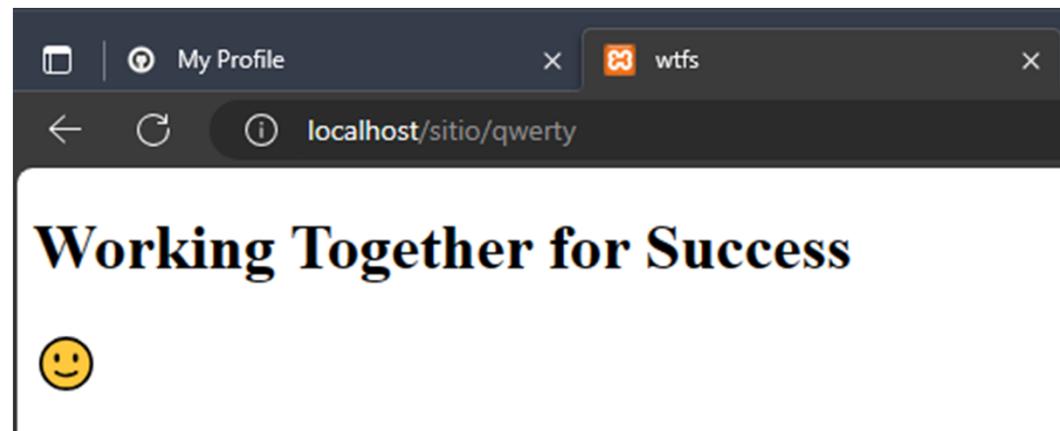
Ejemplos prácticos {url's}



The image shows a screenshot of a code editor interface. On the left, there is an 'EXPLORER: SITIO' sidebar listing several files: 'error', '.htaccess', '.htpasswd', 'index_opc.php', 'index.php', and 'paginas_principal_qwerty.php'. The '.htaccess' file is currently selected. The main panel displays the contents of the '.htaccess' file:

```
1 ## Manejo de Eventos
2 Options -Indexes
3 ErrorDocument 404 http://localhost/sitio/error/404.html
4 ErrorDocument 403 http://localhost/sitio/error/403.html
5
6 ## Manejo del Acceso
7 AuthUserFile C:/xampp/htdocs/sitio/.htpasswd
8 AuthType Basic
9 AuthName "My restricted Area"
10 Require valid-user
11
12 ## Modifica archivo principal
13 DirectoryIndex index_opc.php index.php
14
15 ## Manejo de rutas
16 RewriteEngine On
17 RewriteRule qwerty paginas_principal_qwerty.php
18 |
```

Ejemplos prácticos {url's}



Ejemplos prácticos {url's}

The screenshot shows a code editor interface with two main panes. The left pane is titled "EXPLORER: SITIO" and displays a file tree for a directory named "error". The files listed are ".htaccess", ".htpasswd", "index_opc.php", "index.php", and "paginas_principal_qwerty.php". The ".htaccess" file is selected in the tree. The right pane shows the content of the ".htaccess" file, which contains the following configuration:

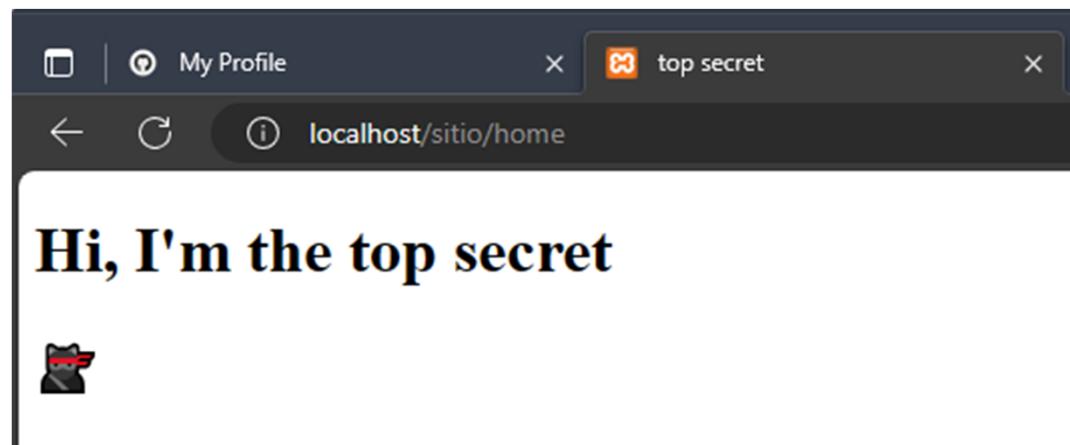
```
## Manejo de Eventos
Options -Indexes
ErrorDocument 404 http://localhost/sitio/error/404.html
ErrorDocument 403 http://localhost/sitio/error/403.html

## Manejo del Acceso
AuthUserFile C:/xampp/htdocs/sitio/.htpasswd
AuthType Basic
AuthName "My restricted Area"
Require valid-user

## Modifica archivo principal
DirectoryIndex index_opc.php index.php

## Manejo de rutas
RewriteEngine On
RewriteRule qwerty paginas_principal_qwerty.php
RewriteRule home index.php
```

Ejemplos prácticos {url's}



Configurar un alias en Apache

Si no quieres colocar tu proyecto en htdocs, puedes definir un alias para usar cualquier carpeta como raíz de tu proyecto.

- Editar el archivo de configuración de Apache:

C:\xampp\apache\conf\extra\httpd-vhosts.conf

Configurar un alias en Apache

- Agrega una configuración de alias

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    DocumentRoot "C:/ruta/a/tu/proyecto"
    ServerName mi-proyecto.local
    ErrorLog "logs/mi-proyecto-error.log"
    CustomLog "logs/mi-proyecto-access.log" common
    <Directory "C:/ruta/a/tu/proyecto">
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
```

Configurar un alias en Apache

- Reinicia Apache desde el Panel de Control de XAMPP.
- Editar el archivo **hosts** para apuntar el dominio a **localhost**

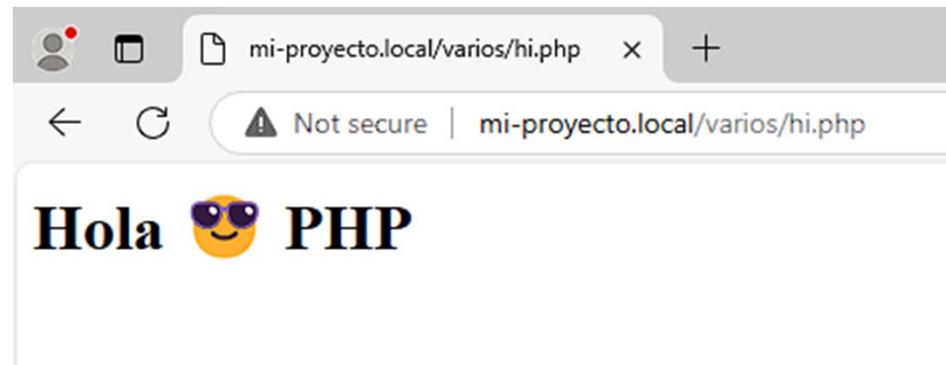
C:\Windows\System32\drivers\etc\hosts

```
# Agrega esta línea  
127.0.0.1    mi-proyecto.local
```

Configurar un alias en Apache

- Ahora puedes acceder al alias

http://mi-proyecto.local



Referencias

- ¿Qué es un servidor Web
y para qué sirve?,
<https://www.webempresa.com/hosting/que-es-servidor-web.html>