



# Path Traversal – Práctica IV

## Explicación del código

# Explicación del código de la parte práctica

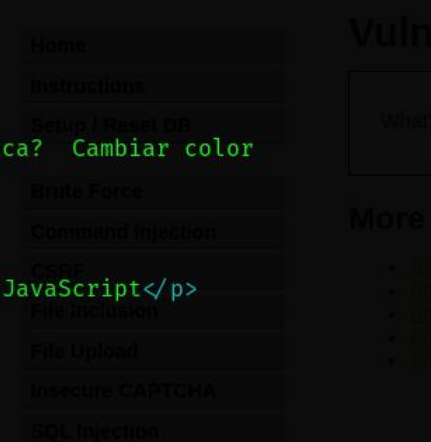
## Web estática

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Una pagina estatica</title>
  <link rel="stylesheet" href="main.css">
</head>
<body>
  <h1>Una pagina Estatica</h1>

  <div>
    <button>¿Qué es una pagina web estática? Cambiar color
      de fondo
    </button>
  </div>

  <p> Es una pagina creada con HTML, CSS y JavaScript</p>
  <script src="app.js"></script>
</body>
</html>
```



1. `<!DOCTYPE html>`: Declara que el documento es un HTML5.
2. `<html lang="en">`: Inicia el documento HTML y establece el idioma en inglés.
3. `<head>`: Contiene metadatos y enlaces a recursos externos.
  - `<meta charset="UTF-8">`: Define la codificación de caracteres como UTF-8.
  - `<meta name="viewport" content="width=device-width, initial-scale=1.0">`: Configura la escala inicial de la página para que sea responsiva en dispositivos móviles.
  - `<title>Una pagina estatica</title>`: Establece el título de la página que se muestra en la pestaña del navegador.
  - `<link rel="stylesheet" href="main.css">`: Enlaza un archivo CSS externo llamado "main.css" para dar estilo a la página.
4. `<body>`: Contiene el contenido visible de la página.
  - `<h1>Una pagina Estatica</h1>`: Muestra un encabezado principal con el texto "Una pagina Estatica".
  - `<div>`: Crea un contenedor para el botón.
    - `<button>¿Qué es una pagina web estática? Cambiar color de fondo</button>`: Un botón con texto que pregunta "¿Qué es una pagina web estática?" y sugiere que cambia el color de fondo.
  - `<p> Es una pagina creada con HTML, CSS y JavaScript</p>`: Un párrafo que explica que una página web estática está hecha con HTML, CSS y JavaScript.
  - `<script src="app.js"></script>`: Enlaza un archivo JavaScript externo llamado "app.js" para añadir funcionalidad a la página.

# Explicación del código de la parte práctica

main.css

```
body {  
    font-family: sans-serif;  
}  
  
button {  
    background: lightblue;  
    border: 1px solid blue;  
    color: blue;  
    font: inherit;  
    padding: 0.5rem 1rem;  
    cursor: pointer;  
}  
  
button:focus {  
    outline: none;  
}  
  
p {  
    background-color: aqua;  
    max-width: fit-content;  
}
```

1. `body { ... }`
  - `font-family: sans-serif;` Establece la familia de fuentes para el cuerpo de la página como sans-serif.
2. `button { ... }`
  - `background: lightblue;` Establece el color de fondo de los botones a azul claro.
  - `border: 1px solid blue;` Define un borde de 1 píxel de ancho, sólido y de color azul para los botones.
  - `color: blue;` Establece el color del texto de los botones a azul.
  - `font: inherit;` Hace que los botones hereden la fuente del elemento padre (en este caso, del `body`).
  - `padding: 0.5rem 1rem;` Añade un relleno de 0.5 rem arriba y abajo, y 1 rem a la izquierda y derecha del contenido del botón.
  - `cursor: pointer;` Cambia el cursor a una mano cuando se pasa por encima del botón, indicando que es clicable.
3. `button:focus { ... }`
  - `outline: none;` Elimina el contorno del botón cuando está enfocado (por ejemplo, después de hacer clic o al usar la tecla Tab para navegar).
4. `p { ... }`
  - `background-color: aqua;` Establece el color de fondo de los párrafos a aqua.
  - `max-width: fit-content;` Hace que el ancho máximo del párrafo se ajuste al contenido, es decir, el párrafo será tan ancho como su contenido, pero no más ancho.
- 5.

# Explicación del código de la parte práctica

app.js

```
var btn = document.querySelector('button');  
  
btn.addEventListener('click', function() {  
    document.querySelector('p').style.background = 'beige';  
})
```

1. `var btn = document.querySelector('button');`
  - Utiliza `document.querySelector` para seleccionar el primer elemento `<button>` en el documento y lo asigna a la variable `btn`.
2. `btn.addEventListener('click', function() { ... });`
  - Añade un "event listener" (escuchador de eventos) al botón `btn` para que escuche el evento 'click'.
  - Cuando el botón es clicado, se ejecuta la función anónima definida como segundo argumento de `addEventListener`.
3. `document.querySelector('p').style.background = 'beige';`
  - Dentro de la función anónima, `document.querySelector('p')` selecciona el primer elemento `<p>` (párrafo) en el documento.
  - `style.background = 'beige';` cambia el color de fondo del párrafo seleccionado a beige.

# Explicación del código de la parte práctica

## Web dinámica


index.html

Se destacan sobre todo las diferencias con respecto al fichero de igual nombre en la web estática.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Una pagina dinamica</title>
  <link rel="stylesheet" href="main.css">
</head>
<body>
  <h1>Una pagina Dinámica</h1>

  <p> Es una pagina creada con HTML, CSS y JavaScript. <br>Sin embargo, a diferencia de las estaticas,
    hay contenido generado por el servidor.</p>

  <a href="marco.php?page=login.html">Login</a>
</body>
</html>
```



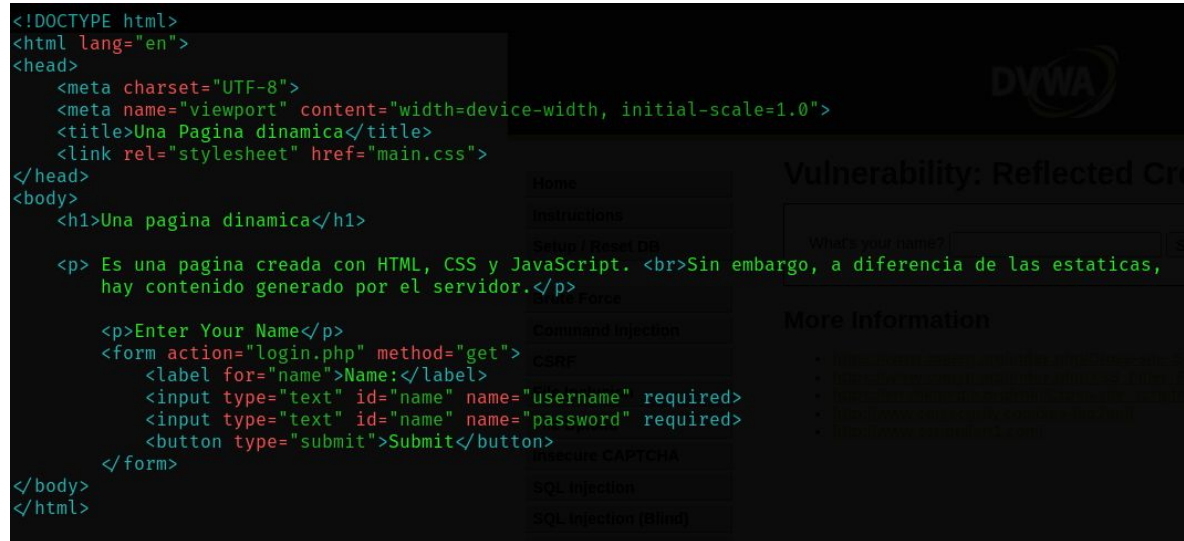
`<body>`: Contiene el contenido visible de la página.

- `<h1>Una pagina Dinámica</h1>`: Muestra un encabezado principal con el texto "Una pagina Dinámica".
- `<p> Es una pagina creada con HTML, CSS y JavaScript. <br>Sin embargo, a diferencia de las estaticas, hay contenido generado por el servidor.</p>`: Un párrafo que explica que una página web dinámica está hecha con HTML, CSS y JavaScript, y que a diferencia de las páginas estáticas, hay contenido generado por el servidor. El elemento `<br>` inserta un salto de línea dentro del párrafo.
- `<a href="marco.php?page=login.html">Login</a>`: Un enlace que apunta a una URL que aparentemente incluye un parámetro (`page=login.html`). Este enlace probablemente lleva a una página que maneja el contenido de manera dinámica a través de un script del lado del servidor, como PHP.



# Explicación del código de la parte práctica

## login.html



Se destacan sobre todo las diferencias con respecto al fichero de igual nombre en la web estática.

1. `<body>`: Contiene el contenido visible de la página.
  - `<h1>Una pagina dinamica</h1>`: Muestra un encabezado principal con el texto "Una pagina dinamica".
  - `<p> Es una pagina creada con HTML, CSS y JavaScript. <br>Sin embargo, a diferencia de las estaticas, hay contenido generado por el servidor.</p>`: Un párrafo que explica que una página web dinámica está hecha con HTML, CSS y JavaScript, y que, a diferencia de las páginas estáticas, hay contenido generado por el servidor. El elemento `<br>` inserta un salto de línea dentro del párrafo.
  - `<p>Enter Your Name</p>`: Muestra un párrafo con el texto "Enter Your Name".
  - `<form action="login.php" method="get">`: Define un formulario que envía los datos al archivo `login.php` usando el método GET.
    - i. `<label for="name">Name:</label>`: Etiqueta para el campo de entrada de texto que sigue.
    - ii. `<input type="text" id="name" name="username" required>`: Campo de entrada de texto para el nombre del usuario, con el atributo `required` que indica que este campo debe ser llenado antes de enviar el formulario.
    - iii. `<input type="text" id="name" name="password" required>`: Campo de entrada de texto para la contraseña del usuario, también requerido. Nota: Ambos campos tienen el mismo `id="name"`, lo cual es un error. Cada `id` debe ser único.
    - iv. `<button type="submit">Submit</button>`: Botón para enviar el formulario.

# Explicación del código de la parte práctica

login.php

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Welcome</title>
</head>
<body>
  <?php
    if ($_SERVER["REQUEST_METHOD"] == "GET") {
      $name = htmlspecialchars($_GET['username']);
      $password = htmlspecialchars($_GET['password']);

      $valid_name = 'admin';
      $valid_pass = 'admin';

      if ($name == $valid_name and $valid_pass == $password) {
        echo "<h1>Welcome, $name!</h1>";
      } else {
        echo "<h1>Invalid username or password</h1>";
      }
    }
  >
</body>
</html>
```

Se destacan sobre todo las diferencias con respecto al fichero de igual nombre en la web estática.

1. <?php ... ?>: Inicia y finaliza un bloque de código PHP.
  - if (\$\_SERVER["REQUEST\_METHOD"] == "GET") { ... }: Verifica si la solicitud HTTP es de tipo GET.
  - \$name = htmlspecialchars(\$\_GET['username']);: Obtiene el valor del parámetro 'username' de la URL, lo convierte a un formato seguro para evitar ataques XSS, y lo asigna a la variable \$name.
  - \$password = htmlspecialchars(\$\_GET['password']);: Hace lo mismo para el parámetro 'password', asignándolo a la variable \$password.
  - \$valid\_name = 'admin'; y \$valid\_pass = 'admin';: Define las credenciales válidas como 'admin' para el nombre de usuario y la contraseña.
  - if (\$name == \$valid\_name and \$valid\_pass == \$password) { ... }: Comprueba si las credenciales ingresadas coinciden con las válidas.
    - i. echo "<h1>Welcome, \$name!</h1>";: Si las credenciales son correctas, muestra un mensaje de bienvenida con el nombre del usuario.
  - else { ... }: Si las credenciales no son correctas, muestra un mensaje de error.
    - i. echo "<h1>Invalid username or password</h1>";: Muestra un mensaje indicando que el nombre de usuario o la contraseña son inválidos.

# Explicación del código de la parte práctica

marco.php

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Una pagina dinamica</title>
</head>
<body>
  <h1>Esta es la cabecera</h1>

  <?php
  if (isset($_GET['page'])) {
    $page = $_GET['page'];

    include($page);
  }

  ?>

  <h1>Este es el footer</h1>
</body>
</html>
```

Se destacan sobre todo las diferencias con respecto al fichero de igual nombre en la web estática.

1. `<body>`: Contiene el contenido visible de la página.
  - `<h1>Esta es la cabecera</h1>`: Muestra un encabezado con el texto "Esta es la cabecera".
  - `<?php ... ?>`: Inicia y finaliza un bloque de código PHP.
    - `if (isset($_GET['page'])) { ... }`: Verifica si el parámetro 'page' está presente en la URL.
      - `$page = $_GET['page'];`: Asigna el valor del parámetro 'page' a la variable `$page`.
      - `include($page);`: Incluye el archivo cuyo nombre es el valor de `$page`. Esto permite la inclusión dinámica de contenido.
2. `<h1>Este es el footer</h1>`: Muestra un encabezado con el texto "Este es el footer".

## Funcionamiento

- Cuando se accede a esta página con una URL que incluye un parámetro `page`, como `http://example.com/index.php?page=contenido.php`, el valor de `page` se utilizará para incluir el archivo especificado.
- Si el parámetro `page` no está presente en la URL, el bloque de código PHP no se ejecutará y no se incluirá ningún contenido dinámico.



