



Soluciones

EJERCICIOS DEL TEMA 4: CGI Y PHP

EJERCICIO 1

- ✗ Escribir una aplicación CGI en C++ que compruebe los datos del formulario y muestre un mensaje de bienvenida.

```
<form action="/cgi-bin/check.cgi" method="post">  
    Usuario: <input type="text" name="usuario"/><br/>  
    Contraseña: <input type="text" name="pass"/><br/>  
    <input type="submit"/>  
</form>;
```

EJERCICIO 1 – SOLUCIÓN (1/3)

```
#include <iostream>
using namespace std;

string getValue(string param, string query) {
    ...
}

int main() {
    // Recogemos los datos del formulario
    string query;
    cin >> query;        // El método POST deja los datos
                          // en la entrada estándar
    // Parámetros deseados: "usuario" y "pass"
    string usuario, pass;
    usuario = getValue("usuario", query);
    pass = getValue("pass", query);
    ...
}
```

EJERCICIO 1 – SOLUCIÓN (2/3)

```
int main() {  
    ...  
    // Comprobamos validez y generamos respuesta  
    string respuesta;  
    if (usuario == "user" && pass == "passwd") {  
        // Datos correctos  
        respuesta = "Bienvenido " + usuario;  
    } else {  
        //Datos incorrectos  
        respuesta = "Nombre de usuario o  
                    contraseña no válidos";  
    }  
    ...  
}
```

EJERCICIO 1 – SOLUCIÓN (3/3)

```
int main() {  
    ...  
    // Generamos el código HTML a devolver  
    // 1. Cabecera simple  
    cout << "Content-Type: text/html" << endl << endl;  
  
    // 2. Página XHTML  
    cout << "<html><head>" << endl;  
    cout << "<title>Ejercicio 1</title>" << endl;  
    cout << "</head>\n<body>" << endl;  
    cout << "<p>" << respuesta << "</p>" << endl;  
    cout << "</body></html>";  
  
    return 0;  
}
```

EJERCICIO 2

- ✗ Detectar los errores en el siguiente trozo de código PHP.

```
<?php
    $x = -1;
    $y = 9;
    $suma = x + y;
    print("El valor de x es <i>$x<i>")
    <br />
    print("Valor de y es <i>$y</i><br/>");
    print('Suma: <b><i>$suma</i></b><br/>');
?>;
```


EJERCICIO 2 – SOLUCIÓN

✖ Cambios en rojo

```
<?php
    $x = -1;
    $y = 9;
    $suma = $x + $y;
    print("El valor de x es <i>$x</i>");
    <br />
    print("Valor de y es <i>$y</i><br/>");
    print("Suma: <b><i>$suma</i></b><br/>");
?>
```

EJERCICIO 3

- ✗ Completar el código para conseguir la funcionalidad deseada.

Se desea controlar el acceso a la página PHP que estamos programando **sin utilizar las variables `PHP_AUTH_USER` y `PHP_AUTH_PW` por lo que diseñamos nuestro propio formulario de acceso. Se debe comprobar que los datos aportados por el usuario coinciden con los datos de acceso válidos (usuario→“user”, contraseña→“passwd”). Se permitirán, como máximo, 3 intentos por parte del usuario.**

EJERCICIO 3 – SOLUCIÓN (1/2)

```
<html> <head> <title>Ejercicio de PHP</title> </head>
<body>
<?php
    session_start();
    $contador = ++$_SESSION['contador'];
    if (
        (!isset($_POST['user'])) ||
        (
            (htmlspecialchars($_POST['user'])!="user" ||
             htmlspecialchars($_POST['pass'])!="passwd"))
            &&
            ($contador < 4)
        )
    ) {
        //Mostrar el formulario
    }
?>
...
```

EJERCICIO 3 – SOLUCIÓN (2/2)

```
<FORM METHOD=POST ACTION="<?php echo
$_SERVER['PHP_SELF']; ?>">
Usuario: <INPUT TYPE="text" NAME="user" /><br />
Contraseña: <INPUT TYPE="text" NAME="pass" /><br />
<INPUT TYPE="submit">
</FORM>
<?php
    } else if (htmlspecialchars($_POST['user'])=="user"
        && htmlspecialchars($_POST['pass'])=="passwd") {
?> //Todo correcto, mensaje de bienvenida
<p>Bienvenido <?php echo
htmlspecialchars($_POST['user']); ?></p>
<?php } else { ?> // El contador ha llegado a 4
    <font color=red> Ha superado el número de
intentos permitidos </font>
<?php } ?>
</body></html>
```