

Desarrollo Web en Entorno Servidor

Tarea Unidad 2

Contenido

Ejercicio 1:	2
Ejercicio 2:	5

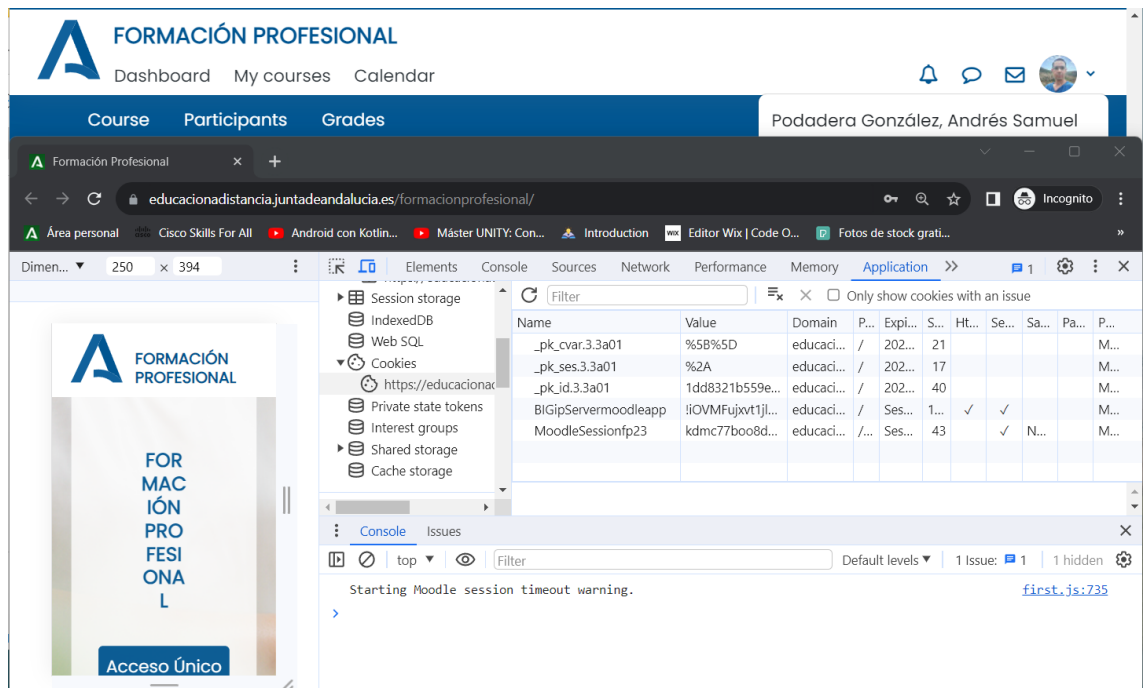
Ejercicio 1:

En esta actividad tienes que tratar de identificar la información que una determinada aplicación web almacena en el cliente web. Para ello selecciona una aplicación web cualquiera (por ejemplo, <https://educacionadistancia.juntadeandalucia.es/formacionprofesional/>) y realiza lo siguiente:

- *Describe la información que piensas que la aplicación web está almacenando en el lado del cliente y el motivo por el que se almacenaría dicha información (ej1.txt).*

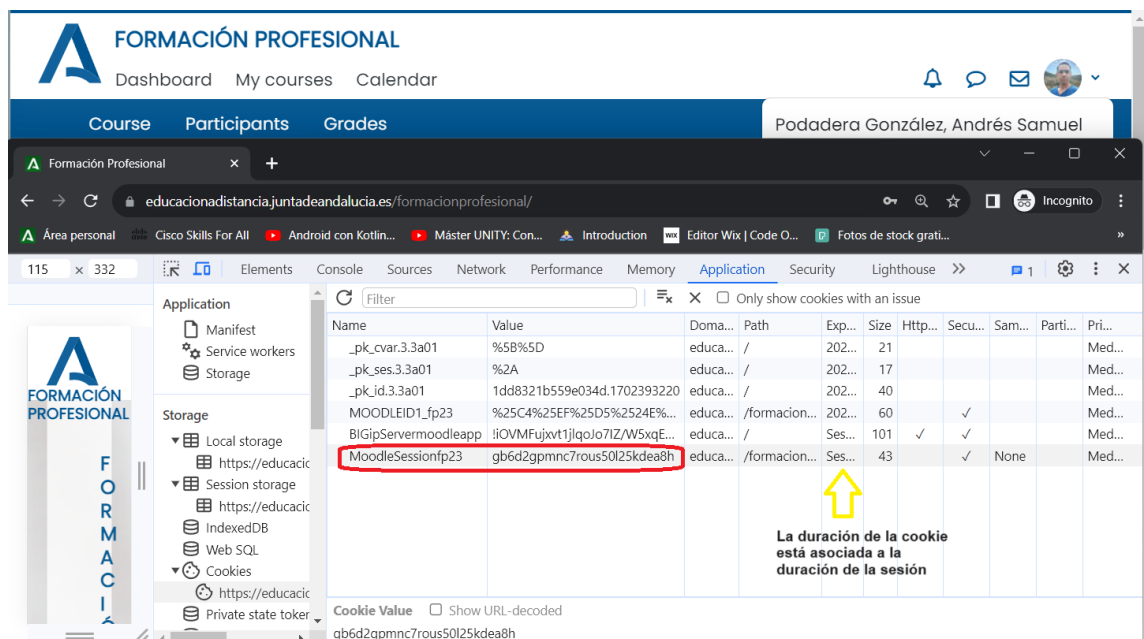
La primera vez que se accede a la página, en la mayoría de las veces, le muestran un aviso de que si quiere permitir que el sitio use cookies. Para una mayor accesibilidad, comodidad y, en definitiva, mejorar la experiencia del usuario con la web, el cliente o navegador va a guardar una especie de rastreadores(cookies) que serán capaces informar al sitio web cual ha sido la interacción con él(registro de uso, preferencias, idioma, localización, características del navegador o dispositivo de uso). Toda esta información queda almacenada para que el usuario no tenga que introducirla de nuevo cada vez que visita el sitio. Lo que por una lado tiene sus ventajas en el aspecto del rendimiento web y UX puede presentar problemas de seguridad, ya que este registro de interacción puede ser accesible por terceros, no tenemos control sobre nuestra información y puede ser usada con fines que desconocemos.

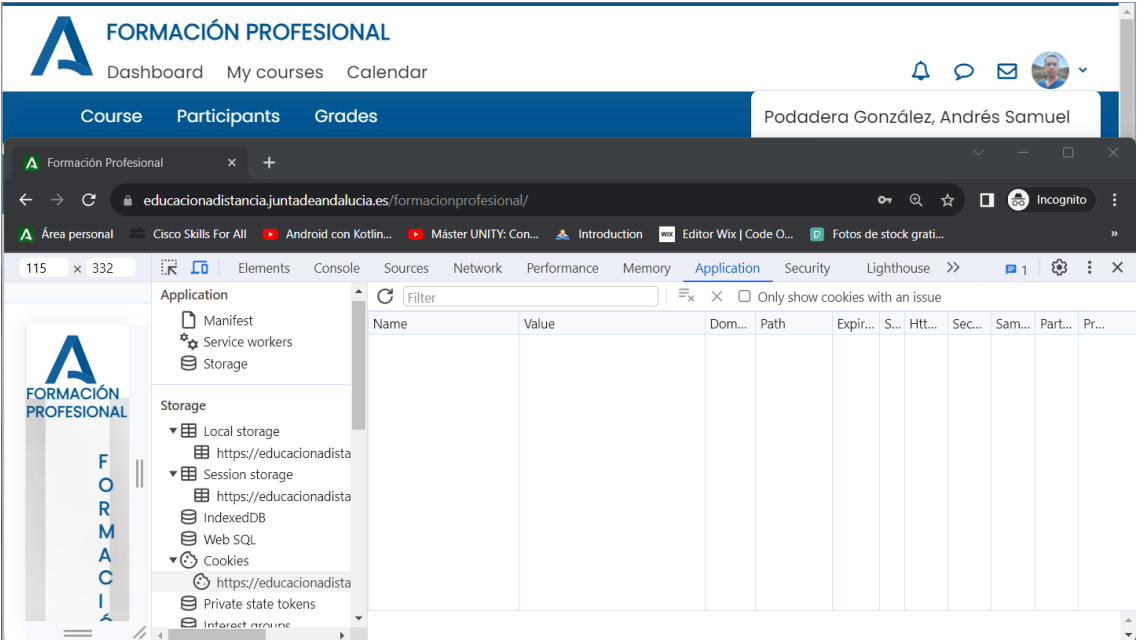
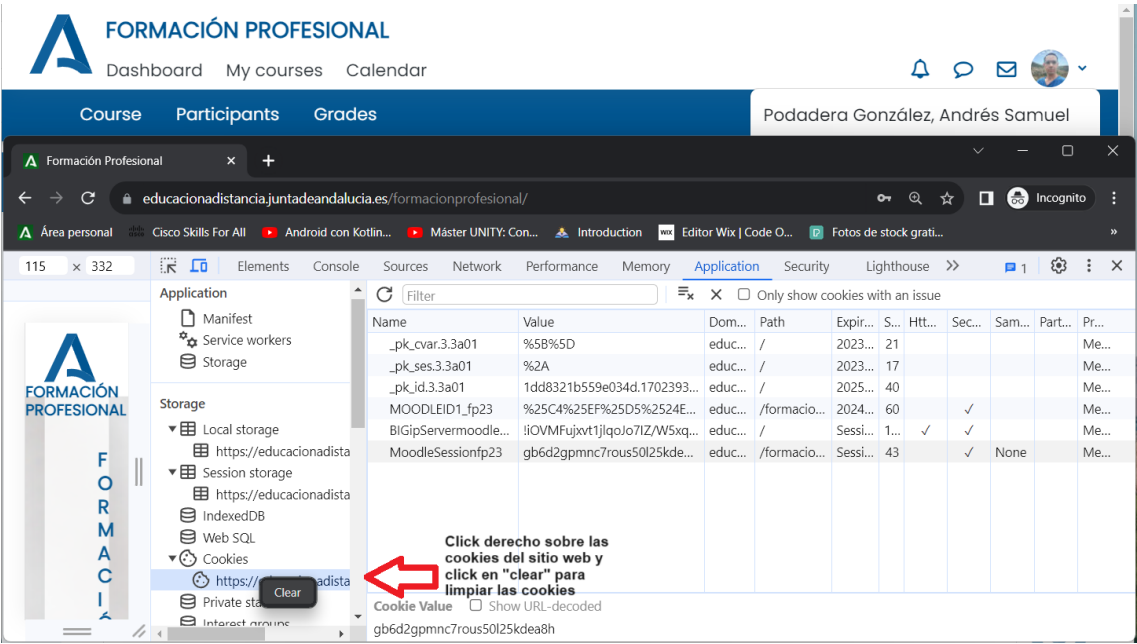
- *Abre el depurador del navegador (tecla F12 en Firefox, Chrome y Edge) y busca la sección "Almacenamiento" (Firefox) o "Aplicación" (Chrome/Edge). Localiza la información almacenada en tu equipo en forma de cookies de dicha aplicación web y añade una captura de pantalla (ej1.png).*



- Trata de identificar aquellas cookies relativas a la sesión de usuario y márcalas de alguna forma en la imagen anterior. Acto seguido, borra la cookie desde el mismo depurador.

Nota importante: por motivos de privacidad usando el **modo privado o incógnito del navegador** y debes realizar la captura de pantalla sin haber iniciado sesión en ningún servicio web.





Ejercicio 2:

En esta tarea tendrás que configurar el debugger [XDebug](#) en PHP y depurar el script siguiente para encontrar un error que tiene. Para demostrar que has realizado la depuración **se deberá adjuntar una captura de pantalla en la que aparezca el depurador de NetBeans en acción (o Visual Studio Code)**. En la captura de pantalla es imprescindible que aparezca **de fondo tu perfil en la plataforma de FP a Distancia, mostrando tu imagen en dicha plataforma**, como prueba de que la has realizado tú.

El código de la captura de pantalla deberá ser el siguiente código corregido (no hay que agregar líneas extra, solo cambiar un pequeño detalle):

```
/**
 * Código que averigua si los elementos de la siguiente cadena de texto están
 ordenados
 * en función de la letra en la primera posición y en función del número de la
 segunda posición.
 */

$datos=['a2','x3','c5','d4'];

$letra=null;
$numero=null;
$ordenados_por_letra=true;
$ordenados_por_numero=true;
foreach($datos as $key=>$val)
{
    if ($letra==null) $letra=ord($val[0]);
    elseif (ord($val[0])>=$letra) $letra=ord($val[0]);
    else $ordenados_por_letra=false;

    if ($numero==null) $numero=intval($val[1]);
    elseif (intval($val[1])>=$numero) $numero=intval($val[1]);
    else $ordenados_por_numero=false;

    if (!$ordenados_por_numero || !$ordenados_por_letra) break;
}

if ($ordenados_por_letra)
    echo "Los datos están SI ordenados por el primer símbolo alfanumérico";
else
    echo "Los datos están NO ordenados por el primer símbolo alfanumérico";
```

```
echo "<BR>".PHP_EOL;

if ($ordenados_por_numero)
    echo "Los datos están SI ordenados por el segundo dígito numérico";
else
    echo "Los datos están NO ordenados por el segundo dígito numérico";
```

El código anterior simplemente muestra si las cadenas del array están ordenadas en función del primer carácter alfanumérico: `$datos=['a2','x3','c5','d4']`; o en función del segundo carácter que contiene un dígito numérico: `$datos=['a2','x3','c5','d4']`;

Al ejecutarlo con el array anterior indica lo siguiente:

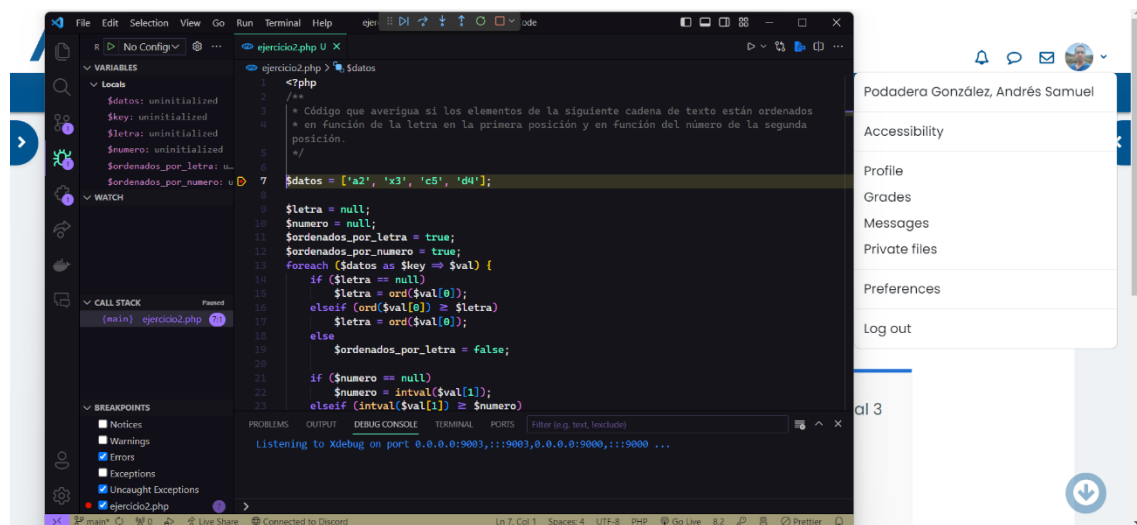
Los datos están NO ordenados por el primer símbolo alfanumérico

Los datos están SI ordenados por el segundo dígito numérico

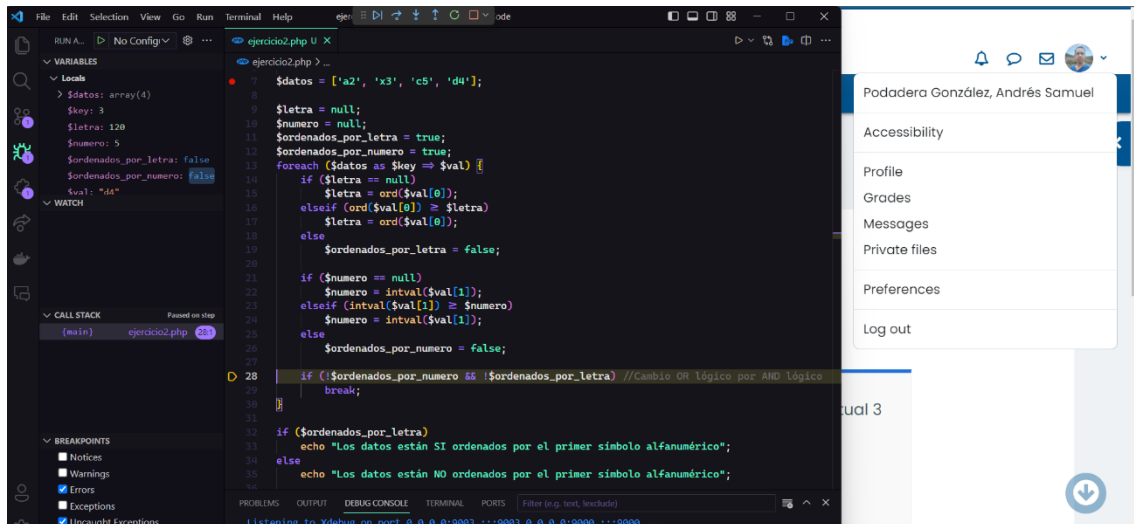
Pero debería indicar "NO" en ambos casos.

Esta captura de pantalla deberá aparecer a un lado (a tu elección) tu imagen de perfil en la plataforma de FP a Distancia. El nombre de este archivo será `ejercicio2.png`.

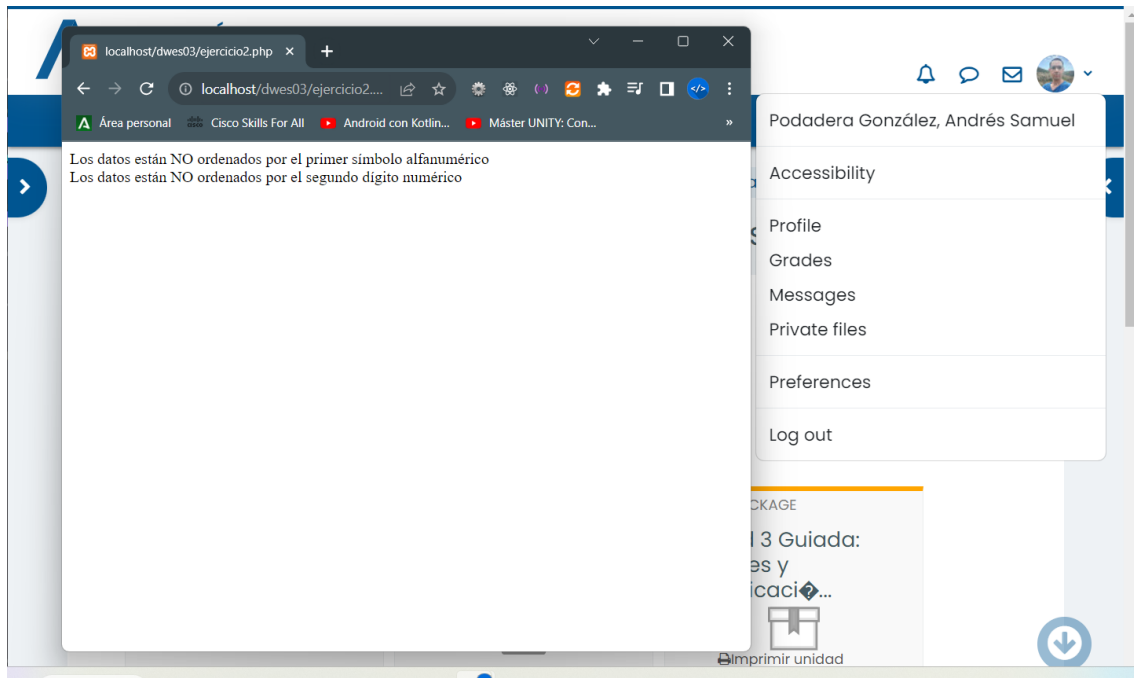
Inicio de la depuración:



Código corregido:



Salida web correcta:



Código correcto:

```
<?php
/**
 * Código que averigua si los elementos de la siguiente cadena de
 * texto están ordenados
 * en función de la letra en la primera posición y en función del
 * número de la segunda posición.
 */

$datos = ['a2', 'x3', 'c5', 'd4'];

$letra = null;
$numero = null;
$ordenados_por_letra = true;
```

```
$ordenados_por_numero = true;
foreach ($datos as $key => $val) {
    if ($letra == null)
        $letra = ord($val[0]);
    elseif (ord($val[0]) >= $letra)
        $letra = ord($val[0]);
    else
        $ordenados_por_letra = false;

    if ($numero == null)
        $numero = intval($val[1]);
    elseif (intval($val[1]) >= $numero)
        $numero = intval($val[1]);
    else
        $ordenados_por_numero = false;

    if (!$ordenados_por_numero && !$ordenados_por_letra) //Cambio OR
lógico por AND lógico
        break;
}

if ($ordenados_por_letra)
    echo "Los datos están SI ordenados por el primer símbolo
alfanumérico";
else
    echo "Los datos están NO ordenados por el primer símbolo
alfanumérico";

echo "<BR>" . PHP_EOL;

if ($ordenados_por_numero)
    echo "Los datos están SI ordenados por el segundo dígito
numérico";
else
    echo "Los datos están NO ordenados por el segundo dígito
numérico";
```

Cambiando || por && solucionamos el problema del código en el que usando OR se permitía comprobar un tipo de ordenamiento, en este caso “por número” sin verificar si se cumplía o no “por letra” lo que provocaba que la salida no fuera la correcta. Haciendo uso de && se comprueban ambos tipos de ordenamientos y se “sale” del bucle cuando no se cumple ninguna de ellas, que en este caso coincide con el último elemento del array.