

## ACTIVIDADES UD3 - SERVIDOR

Mediante las tareas que se detallan en este documento vamos a conseguir la parte de los resultados de aprendizaje cubiertos por la UD3, atendiendo a sus correspondientes criterios de evaluación:

RESULTADOS DE APRENDIZAJE	CRITERIOS DE EVALUACIÓN
<b>RA3.</b> Escribe bloques de sentencias embebidos en lenguajes de marcas, seleccionando y utilizando las estructuras de programación.	<ul style="list-style-type: none"><li>a) Se han utilizado mecanismos de decisión en la creación de bloques de sentencias.</li><li>b) Se han utilizado bucles y se ha verificado su funcionamiento.</li><li>c) Se han utilizado matrices (arrays) para almacenar y recuperar conjuntos de datos.</li><li>d) Se han creado y utilizado funciones.</li><li>g) Se han añadido comentarios al código.</li></ul>

**IMPORTANTE:** Junto con la entrega de Aules, adjunta tu propia tabla de calificaciones, con tu propia estimación de calificación. No pongas ninguna calificación en aquellas actividades que no hayas podido hacer. Todo aquello que indiques como no realizado, se entenderá que no lo has realizado.

**NOTA:** Cualquier copia (ya sea de fuentes externas, literal de los apuntes...) que no sea una respuesta original será calificada con 0.

### Actividad 1: Comentarios en el código

<b>Código:</b> UD3.1	<b>CE:</b> RA3.g	<b>IE:</b> I3, I6	<b>Puntos:</b> 10	<b>Estimación:</b> 15 min
----------------------	------------------	-------------------	-------------------	---------------------------

#### DESCRIPCIÓN

Para cada una de las actividades que siguen a ésta, inserta comentarios en el código para poder delimitar cada una de las soluciones que des. Para eso, inserta un comentario previo a cada fragmento de código que crees, especificando el código de cada tarea.

El código de cada tarea se formará con el código de la actividad, más la letra que identifica a la tarea en particular. Por ejemplo, para la tarea a) de la actividad 2, el código será: UD3.2.a

Ten en cuenta que pueden ser comentarios en PHP o en HTML.

Las respuestas que no sean identificables mediante comentario con el correspondiente código podrán no ser evaluadas.

#### ENLACES

- [Comentarios en PHP](#)
- [Comentarios en HTML](#)

### Actividad 2: Mecanismos de decisión

<b>Código:</b> UD3.2	<b>CE:</b> RA3.a	<b>IE:</b> I3, I6	<b>Puntos:</b> 10	<b>Estimación:</b> 1 h
----------------------	------------------	-------------------	-------------------	------------------------

#### DESCRIPCIÓN

Realiza las siguientes tareas:

- Modifica el enlace de la opción de menú "INICIO" para que al pinchar sobre ella, te lleve a la página de inicio. **(1 punto)**
- Modifica el enlace de la opción de menú "CONTACTO" para que al pinchar sobre ella, te lleve a la página de contacto. **(1 punto)**
- En datos.php, crea una variable donde se almacene tu nombre y apellidos, y se utilice para completar el elemento h3 de la página de Contacto. **(1 punto)**
- A la hora de representar la imagen del proyecto (en index.php y en proyecto.php), si el proyecto no tiene una imagen (atributo "imagen" vacío), que muestre una imagen por defecto. Utiliza para ello el operador ternario. **(1 punto)**
- En datos.php, crea una variable llamada \$loggedIn. Si esta variable tiene el valor booleano verdadero, entonces se mostrará una opción de menú adicional llamada "ADMINISTRACIÓN"; en caso de que tenga el valor booleano falso, no ha de existir la opción de menú adicional. **(2 puntos)**
- Añadir dos botones a la página de index.php que, al pulsar sobre ellos, ordenen los proyectos en orden ascendente o descendente por nombre de proyecto (un botón para cada caso). **(2 puntos)**
- (A realizar en combinación con la tarea [UD3.3.f](#)) En caso de filtrar por categoría, la opción de menú "CATEGORÍAS" estará activada, y no la de "INICIO" **(2 puntos)**

#### TIPS

- Si has de descargar alguna imagen, utiliza algún sitio en el que las imágenes que puedes

descargar tengan licencia libre, como pixabay.com.

- Para la creación de dos botones del apartado f), consulta los [botones](#) de bootstrap.

## ENLACES

- [Condiciones](#)

### Actividad 3: Arrays

**Código:** UD3.3

**CE:** RA3.c

**IE:** I3, I6

**Puntos:** 10

**Estimación:** 2 h

## DESCRIPCIÓN

Realiza las siguientes tareas:

- Construye un array como el mostrado en el documento de teoría, almacenado en la variable \$proyectos. Constrúyelo a partir de datos lo más reales posibles (tecnologías, descripción, etc), con imágenes diferentes (descárgalas en el directorio static/images). Ha de contener al menos 5 proyectos. **(1 punto)**
- Añade dos elementos más a los proyectos: "fecha" (con formato "DD/MM/AAAA") y categorías. El elemento "categorías" en un proyecto será a su vez un array que contendrá cadenas de texto (o palabras). Estas palabras se corresponderán con la clave del array de categorías creado en el apartado 6.2 del documento de teoría. No incluyas el nombre de la categoría en el array de proyectos, solo su clave (que ha de tener una correspondencia en el array de categorías) **(1 punto)**
- (Puntúa en la actividad 4)** Tanto en index.php como en proyecto.php, lista el nombre de las categorías del proyecto en cuestión, separadas por un espacio. No imprimas la clave, sino el nombre de la categoría. En cada pasada del bucle tendrás que recuperar el nombre de la categoría a partir de su clave, ya que el nombre no está en el array de proyectos, sino la clave de la categoría. Utiliza la función array\_key\_exists (ver TIPS).
- En index.php, completa el atributo href de cada elemento <a> con una URL que apunte a la ficha del proyecto, de la forma: localhost/proyecto?id="claveproyecto". Tras esto:
  - En proyecto.php recuperarás el valor del parámetro "id" mediante \$\_GET
  - Buscarás el proyecto concreto en el array de proyectos, iterando el array de proyectos.
  - Sustituirás los valores literales introducidos en proyecto.php por el valor correspondiente del proyecto recuperado en el paso anterior, junto con los datos adicionales: fecha y categorías.
  - Utiliza la función [nl2br](#) para imprimir la descripción del proyecto, así se respetarán los saltos de línea **(2 puntos)**
- (Puntúa en la actividad 4)** Completa la opción de menú "CATEGORÍAS". Para ello, utiliza un bucle que recorra el array de categorías y construya un dropdown, basándote en el siguiente código (sustituye el que existe en header.php):

```
<li class="nav-item" >
  <a class="nav-link dropdown-toggle" id="dropdownMenu1" data-bs-toggle="dropdown" aria-
haspopup="true">
    CATEGORÍAS
    <span class="caret"></span>
  </a>
```

```

<div class="dropdown-menu" aria-labelledby="dropdownMenu1">
  <a class="dropdown-item" href="#">Categoría1</a>
  <a class="dropdown-item" href="#">Categoría2</a>
  <a class="dropdown-item" href="#">Categoría3</a>
</div>
</li>

```

- f) Una vez construyas el dropdown, teniendo una entrada por cada una de las categorías del array de categorías, construye el href para cada una de esas entradas de la forma: localhost.php/index.php?categoria="clavecategoria". Por su parte, index.php leerá el parámetro "categoria" y filtrará los proyectos que tengan esa categoría (utiliza la función `array_filter`). No modifiques el valor del array `$proyectos`, crea uno nuevo `$proyectos_filtrado` donde almacenes los valores recuperados tras filtrar por categoría. La tarea UD3.2.g está relacionada con ésta (**3 puntos**)
- g) Crea dos ficheros [JSON](#) y almacena los proyectos en este formato (reparte los proyectos que ya tienes entre estos dos ficheros). Desde `datos.php` lee estos dos ficheros, pásalos a arrays, y forma un único array (`$proyectos`) a partir de los dos primeros (**2 puntos**)
- h) Mediante un parámetro de URL llamado "delete", borra el último de los proyectos del array de categorías, en `index.php`, y utilizando el superglobal `$_GET`. La URL será de la forma: localhost/index.php?delete=true. Puedes utilizar la función [array\\_pop](#) (**1 punto**)

#### TIPS

- En c), para recuperar los nombres de las categorías de un proyecto tendrás que:
  - Crear un bucle que recorra el array de las categorías de un proyecto. En `index.php`, será un bucle anidado.
  - Por cada clave de categoría, comprobar que existe en el array de categorías. Utiliza la función `array_key_exists`.
  - Si existe en el array de categorías, se recupera el valor de esa clave de categoría y se imprime.
- En f), para leer un fichero utiliza la función [file\\_get\\_contents](#), y para pasar de JSON a un array utiliza la función [json\\_decode](#). Para fundir dos arrays en uno, utiliza la función [array\\_merge](#). Puedes utilizar cualquier otra función, si lo ves conveniente.

#### ENLACES

- [Arrays](#)
- [array\\_filter](#)
- [How to parse a JSON File in PHP ?](#)

#### Actividad 4: Bucles

<b>Código:</b> UD3.4	<b>CE:</b> RA3.b	<b>IE:</b> I3, I6	<b>Puntos:</b> 10	<b>Estimación:</b> 30 min
----------------------	------------------	-------------------	-------------------	---------------------------

#### DESCRIPCIÓN

Realiza las siguientes tareas:

- a) Creación del bucle de la tarea UD3.3.c con "foreach" (**5 puntos**)

b) Creación del bucle de la tarea UD3.3.e con "foreach" (5 puntos)

#### ENLACES

- [Bucles](#)

### Actividad 5: Funciones

**Código:** UD3.5

**CE:** RA3.d

**IE:** I3, I6

**Puntos:** 10

**Estimación:** 45 min

#### DESCRIPCIÓN

Realiza las siguientes tareas:

- Programa una función en utiles.php que devuelva el año actual. Dicho año se ha de concatenar a la palabra "Portfolio" de la cabecera (primer enlace en header.php). Utiliza la función date (leer enlaces) de PHP. (2 puntos)
- Programa una función en utiles.php que tome el array \$proyectos y devuelva un array (que sobrescribirá a \$proyectos) con proyectos cuyo atributo "Fecha" sea de tipo fecha, no un string. Para realizar la conversión, utiliza la función strtotime (leer enlaces). Cuando representes la fecha del proyecto en HTML, utiliza la función date (leer enlaces). (4 puntos)
- Implementa una función que reordene los proyectos según la fecha, en orden ascendente o descendente, según un parámetro de entrada. Este parámetro de entrada a la función se derivará de otro parámetro que se pasará por URL a index.php, llamado "sort\_date", pasado por \$\_GET, y que tendrá el valor "1" (ordenación ascendente) o "-1" (ordenación descendente). Para la reordenación, apóyate en la función [array\\_multisort](#) (leer TIPS). Crea dos botones que disparen la reordenación ascendente o descendente. (4 puntos)

#### TIPS

- [Ordenar array por valor en subarray](#)

#### ENLACES

- [Fechas en PHP](#)

### Actividad de AMPLIACIÓN (no obligatoria)

**Código:** UD3.6

**CE:** RA3.a,  
RA3.b, RA.c

**IE:** I3, I6

**Puntos:** 10

**Estimación:** 1.5 h

#### DESCRIPCIÓN

Crea dos botones "ANTERIOR" y "SIGUIENTE" para la paginación. "ANTERIOR" no se mostrará cuando se muestre la primera página de resultados; "SIGUIENTE" no se mostrará cuando se muestre la última página de resultados. Si no existen resultados, entonces no se mostrará ninguno de los dos botones, y se mostrará un texto que diga "No se han recuperado proyectos". Ten en cuenta, además, lo siguiente:

- Crea una variable que almacene cuántos elementos por página se han de mostrar.
- El enlace "ANTERIOR" o "SIGUIENTE" serán enlaces a index.php con un parámetro

que indicará en qué posición del array de proyectos se ha de empezar, mostrando tantos como se haya almacenado en la variable del punto anterior. Puedes llamar a este parámetro "from".

- index.php leerá el parámetro "from" y determinará desde qué posición del array leer, y cuántos proyectos mostrar.
- Ten en cuenta que los proyectos también se pueden filtrar por categorías. Además del parámetro anterior en la URL, puedes encontrar otros parámetros. Cuando construyas el enlace de cada botón, añade el parámetro "from" pero también añade cualquier otro parámetro que pueda contener el superglobal \$\_GET.

**TIPS**

- Para la paginación te puedes basar en este elemento de bootstrap: [pagination](#)