



Tarea online

1.- Descripción de la tarea

1.1.- Ejercicio 1.

1.2.- Ejercicio 2.

1.3.- Ejercicio 3.

1.4.- Ejercicio 4.

2.- Información de interés

3.- Evaluación de la tarea

Anexo. Licencia de recursos

Tarea online

Título de la tarea: Aplicaciones web híbridadas.

Unidad: 7.

Ciclo formativo y módulo: DAW, Desarrollo Web en Entorno Servidor.

Curso académico: 2020/21

¿Qué contenidos o resultados de aprendizaje trabajaremos?

En esta tarea se aborda la creación de aplicaciones web híbridadas, tanto desde el punto de vista de consumo de servicios web, como desde el punto de vista de creación de un API web sencillo.

Tarea online

1.- Descripción de la tarea

1.1.- Ejercicio 1.

1.2.- Ejercicio 2.

1.3.- Ejercicio 3.

1.4.- Ejercicio 4.

2.- Información de interés

3.- Evaluación de la tarea

Anexo. Licencia de recursos

1.1.- Ejercicio 1.

Este primer ejercicio es una toma de contacto con las aplicaciones híbridas. Fíjate en el siguiente código:

```
<?php

function get($url, $params = null)
{
    $ch = curl_init();
    $tail=is_array($params) && !empty($params)?'?'.'http_build_q
    curl_setopt($ch, CURLOPT_URL, $url.$tail);
    curl_setopt($ch, CURLOPT_FOLLOWLOCATION, 1);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    $output = curl_exec($ch);
    curl_close($ch);
    return $output;
}

function post($url,$postdata,$json=false)
{
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_POST, 1);
    curl_setopt($ch, CURLOPT_POSTFIELDS, $json?json_encode($postdata):$postdata);
    curl_setopt($ch, CURLOPT_FOLLOWLOCATION, 1);
    $json && curl_setopt($ch, CURLOPT_HTTPHEADER, array('Content-Type: application/json'));
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    $output = curl_exec($ch);
    curl_close($ch);
    return $output;
}

echo post('http://hilite.me/api',
    [ 'code'=>'<?php phpinfo();'
      , 'lexer'=>'php'
    ]
);
```

En el código anterior se usa un API pública del servicio web <http://hilite.me> que es una aplicación web que simplemente formatea código fuente y lo pone "bonito". Puedes ver los parámetros a enviar a esta API si pones en el navegador <http://hilite.me/api>.

Tu misión en este ejercicio es:

- ✓ Documentar las funciones anteriores, investigando todo lo que necesites por internet, indicando:
 - que hace cada una de las líneas del código anterior.
 - que hace, a nivel global, cada función, con un comentario tipo phpdoc.
- ✓ Adaptar el código anterior para que permita introducir el código fuente a formatear a través de un formulario, donde se pregunte por:
 - el código fuente a formatear (usando un `<textarea>`)
 - y el lenguaje (usando un desplegable, donde se pueda seleccionar: php, javascript o java).

El resultado de este ejercicio debe almacenarse en el script `ejercicio1.php`.

Ten en cuenta que `http://hilite.me/api` NO es un api REST, pero los principios son similares.

Tarea online

1.- Descripción de la tarea

1.1.- Ejercicio 1.

1.2.- Ejercicio 2.

1.3.- Ejercicio 3.

1.4.- Ejercicio 4.

2.- Información de interés

3.- Evaluación de la tarea

Anexo. Licencia de recursos

1.2.- Ejercicio 2.

En este segundo ejercicio vamos a aproximarnos a la estructura de un API básica. El código del API es el siguiente:

```
<?php

Class Response {
    public $status;
    public $result;
}

$postData=file_get_contents("php://input");

$data=json_decode($postData);

$response=new Response();
if ($data)
{
    if (isset($data->n1) && isset($data->n2) &&
        is_numeric($data->n1) && is_numeric($data->n2))
    {
        $response->result=$data->n1+$data->n2;
        $response->status='OK';
    } else
    {
        $response->status='ERR';
    }
}
else
{
    $response->status='NOJSON';
}

echo json_encode($response);
```

El código anterior implementa un API web simple. En dicha API se espera recibir los datos en forma de documento JSON, y a su vez, genera una respuesta JSON con el resultado del procesamiento. **Analiza el código con cuidado, investiga por internet todo lo que necesitas, y documenta el código.** Las operaciones que realiza el código son realmente sencillas.

Imagina ahora que el código anterior se almacena en un archivo llamado `http://localhost/dwes07/service.php`. Para usar dicha API podrías emplear el método `post` del ejercicio anterior de la siguiente forma:

```
$resultadoPeticion=post('http://localhost/DWES07_SOLUCION/service.php',  
    [  
        'n1'=>7, 'n2'=>10  
    ],true);
```

El código anterior estaría obviamente en otro script (`test.php`, por ejemplo) obtendría un resultado en formato JSON que deberías procesar con `json_decode`. Analiza también este ejemplo con cuidado, **siempre teniendo en mente que la parte que implementa el API web estará en un servicio diferente a donde estará la aplicación web que usa dicho API**.

Basandote en el código anterior, vas a crear un API accesible desde <http://localhost/dwes07/ejercicio2service.php>. Ese API recibirá un documento JSON como el siguiente:

```
{  
    "place": "EIFFEL_TOWER"  
}
```

En la petición JSON anterior va incluido un texto que identifica un lugar. Tu código debe responder con las coordenadas de dicho lugar junto a una descripción breve:

```
{  
    "status": "OK"  
    "latitude": 48.858093,  
    "longitude": 2.294694,  
    "description": "Torre Eiffel"  
}
```

No tienes que acceder a ningún API ni nada para obtener las coordenadas del lugar indicado. Solamente eliges 4 lugares a tu elección y buscas sus coordenadas en Internet.

Si en el documento JSON de petición se indica uno de esos cuatro lugares que has elegido, el API debe retornar la coordenada adecuada, si no se incluye un lugar de los 4 previstos, retorna un documento JSON con "STATUS" a "ERROR".

Recuerda: para realizar el ejercicio tienes que adaptar el código facilitado.

Tarea online

1.- Descripción de la tarea

1.1.- Ejercicio 1.

1.2.- Ejercicio 2.

1.3.- Ejercicio 3.

1.4.- Ejercicio 4.

2.- Información de interés

3.- Evaluación de la tarea

Anexo. Licencia de recursos

1.3.- Ejercicio 3.

En este ejercicio se va a utilizar el API que has implementado en el ejercicio 2 (`ejercicio2service.php`) en un script llamado `ejercicio3cliente.php` (<http://localhost/dwes07/ejercicio3cliente.php>).

Para realizar este ejercicio debe consultarse el servicio web usando los métodos vistos en el ejercicio 1, tenlo en cuenta.

Pasos para realizar este ejercicio:

- ✓ Usando un desplegable, haz un formulario donde el usuario pueda seleccionar entre las cuatro localizaciones que has elegido en el apartado anterior. Este formulario se mostrará siempre.
- ✓ Una vez que el usuario ha seleccionado una de las 4 localizaciones en el desplegable, el script `ejercicio3cliente.php` consultará al API <http://localhost/dwes07/ejercicio2service.php> las coordenadas y descripción de la localización seleccionada.
- ✓ Después de recibir los datos del API, el script `ejercicio3cliente.php` mostrará un mapa con la ubicación de la localización seleccionada (usando OpenStreetView con Leaflet). Ten en cuenta que el API puede retornar "ERROR" si la localización no existe. En tal caso no deberá mostrarse el mapa y debería indicarse que la localización no está registrada.

Tarea online

1.- Descripción de la tarea

1.1.- Ejercicio 1.

1.2.- Ejercicio 2.

1.3.- Ejercicio 3.

1.4.- Ejercicio 4.

2.- Información de interés

3.- Evaluación de la tarea

Anexo. Licencia de recursos

1.4.- Ejercicio 4.

Este último ejercicio es un ejercicio libre. En la siguiente URL tienes un colección de APIs públicas que puedes usar:

 <https://github.com/public-apis/public-apis>

Selecciona una de las APIs y crea un pequeño script para consultar dicha API. Para no complicarte mucho, selecciona una que no necesite autenticación (que ponga "No" en la columna Auth).

Aunque muchas de estas APIs son tipo REST, eso no es problemático, dado que los métodos de acceso son los mismos (puedes emplear las funciones del ejercicio 1).

Tu solución debe incluir al menos un formulario donde indicar los datos a consultar y al menos 3 ejemplos de uso.

Tarea online

1.- Descripción de la tarea

1.1.- Ejercicio 1.

1.2.- Ejercicio 2.

1.3.- Ejercicio 3.

1.4.- Ejercicio 4.

2.- Información de interés

3.- Evaluación de la tarea

Anexo. Licencia de recursos

2.- Información de interés

Recursos necesarios y recomendaciones

Recomendaciones generales

- ✓ Como ya sabes, para escribir aplicaciones en PHP necesitarás un entorno XAMPP, LAMPP o similar. No te vamos a pedir que instales un entorno u otro, o que trabajes con un sistema operativo concreto, pero sí que todos los **archivos vayan codificados en UTF-8** y que **la rutas a los archivos sean siempre relativas**.
- ✓ Como plataforma de desarrollo (para escribir código), te aconsejamos **NetBeans 12**. Es la misma herramienta que se usa en otros módulos no tendrás que cambiar de herramienta.
- ✓ No abordes la tarea hasta que hayas repasado todos los contenidos, hecho al menos uno de los intentos para el examen en línea, de forma que hayas tenido ocasión de consultar y resolver en los foros cualquier duda que te haya podido surgir.



Indicaciones de entrega

Una vez realizada la tarea, el envío se realizará a través de la plataforma. Comprime la carpeta del proyecto en un fichero .zip y nómbralo siguiendo las siguientes pautas:

Apellido1_Apellido2_Nombre_DWES_Tarea07

Tarea online

1.- Descripción de la tarea

1.1.- Ejercicio 1.

1.2.- Ejercicio 2.

1.3.- Ejercicio 3.

1.4.- Ejercicio 4.

2.- Información de interés

3.- Evaluación de la tarea

Anexo. Licencia de recursos

3.- Evaluación de la tarea

Criterios de evaluación implicados

- Se han reconocido las ventajas que proporciona la reutilización de código y el aprovechamiento de información ya existente.
- Se han identificado librerías de código y tecnologías aplicables en la creación de aplicaciones Web híbridas.
- Se ha creado una aplicación Web que recupere y procese repositorios de información ya existentes.
- Se han creado repositorios específicos a partir de información existente en Internet y en almacenes de información.
- Se han utilizado librerías de código para incorporar funcionalidades específicas a una aplicación Web.
- Se han programado servicios y aplicaciones Web utilizando como base información y código generados por terceros.
- Se han probado, depurado y documentado las aplicaciones generadas.



[Peggy Marco](#) (([Pixabay License](#)))Correcto.

¿Cómo valoramos y puntuamos tu tarea?

En esta puedes observar la puntuación máxima asignada a cada ejercicio de la tarea:

Rúbrica de la tarea		
Se ha implementado el ejercicio 1 apropiadamente y funciona correctamente.	hasta 2 puntos	
Se ha implementado el ejercicio 2 apropiadamente y	hasta	1,5

funciona correctamente.	puntos
Se ha implementado el ejercicio 3 apropiadamente y funciona correctamente.	hasta 2,5 puntos
Se ha implementado el ejercicio 4 apropiadamente y funciona correctamente.	hasta 4 puntos