



Tarea online

1.- Descripción de la tarea

1.1.- Ejercicio 1: creando la estructura de proyecto.

1.2.- Ejercicio 2: implementando las funciones.

1.3.- Ejercicio 3: implementando el servidor.

1.4.- Ejercicio 4: la interfaz.

1.5.- Ejercicio 5: registrar una medición.

2.- Información de interés

3.- Evaluación de la tarea

Anexo. Licencia de recursos

Tarea online

Título de la tarea: Aplicaciones web dinámicas: PHP y Javascript.

Unidad: 6.

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

Curso académico: 2020/21

¿Qué contenidos o resultados de aprendizaje trabajaremos?

En esta tarea vas a programar una aplicación web que haga uso de librería de servidor para facilitar la utilización de las tecnologías AJAX desde PHP.

Tarea online

1.- Descripción de la tarea

1.1.- Ejercicio 1: creando la estructura de proyecto.

1.2.- Ejercicio 2: implementando las funciones.

1.3.- Ejercicio 3: implementando el servidor.

1.4.- Ejercicio 4: la interfaz.

1.5.- Ejercicio 5: registrar una medición.

2.- Información de interés

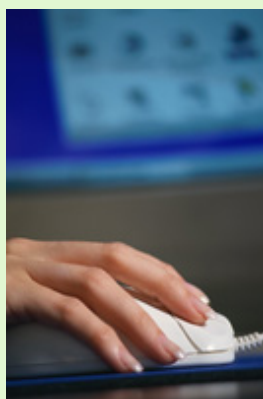
3.- Evaluación de la tarea

Anexo. Licencia de recursos

1.- Descripción de la tarea



Caso práctico



BK Programación quiere dotar a sus aplicaciones web de una mejor interactividad y velocidad de respuesta. Para eso van a combinar el lenguaje del lado de cliente (Javascript) con el del lado servidor (PHP) mediante el uso de AJAX.

En esta tarea vamos a trabajar con XAJAX. Sabemos que es un framework que no está muy actualizado, pero para el propósito de la unidad es suficiente. El objetivo es realizar una comunicación asíncrona entre el cliente y el servidor vía AJAX de forma cómoda.

Para realizar esto, lo más aconsejable, es basarse en el ejemplo que hay en los contenidos, en el apartado 3.4 e ir modificándolo conforme se indica en la tarea. Si vas siguiendo los pasos, seguro que te resultará fácil y productivo.

La tarea está dividida en dos partes:

- ✓ Realizar una pequeña aplicación con XAJAX que requiere poca complicación (para calentar motores).
- ✓ Realizar otra pequeña aplicación con XAJAX, con un pelín más de complicación (pero poco más).

Los archivos de la tarea serán accesibles a través de la URI <http://localhost/dwes06>.

Se valorará en todos los casos la corrección ortográfica y gramatical de los mensajes para comunicarnos con el usuario, así como la presentación clara de cualquier información que se muestre al usuario.

Tarea online

1.- Descripción de la tarea

1.1.- Ejercicio 1: creando la estructura de proyecto.

1.2.- Ejercicio 2: implementando las funciones.

1.3.- Ejercicio 3: implementando el servidor.

1.4.- Ejercicio 4: la interfaz.

1.5.- Ejercicio 5: registrar una medición.

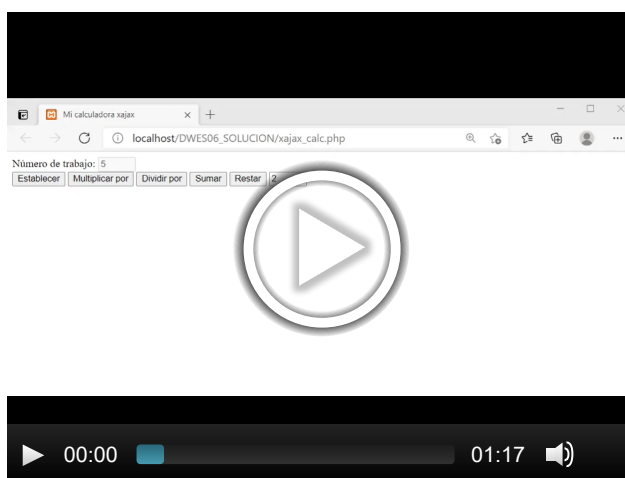
2.- Información de interés

3.- Evaluación de la tarea

Anexo. Licencia de recursos

1.1.- Ejercicio 1: creando la estructura de proyecto.

Partimos de que ya tienes contigo el ejemplo de la unidad, el del apartado 3.4. La idea de este primer ejercicio es hacer una pequeña calculadora:



En este ejemplo vamos a trabajar con 4 archivos diferentes:

- ✓ **xajax_calc.base.php**: este script será el script base, con código común que necesitan otros scripts. Sus responsabilidades serán:
 - Iniciar las sesiones (para almacenar el valor de trabajo)
 - Importante: el valor de trabajo se almacena en la sesión, solo en la sesión.
 - Incluir el archivo principal de xajax (**xajax.inc.php**)
 - crear una nueva instancia de xajax.
 - registrar las diferentes funciones.
- ✓ **xajax_calc.functions.php**: en este script se crearán las funciones registradas para cada operación ajax.
- ✓ **xajax_calc.server.php**: este script será el script base para la parte del servidor, sus funciones serán:
 - incluir **xajax_calc.base.php**,
 - incluir las funciones (del archivo **xajax_calc.functions.php**),
 - procesar las peticiones.
- ✓ **xajax_calc.php**: este es el script principal (**http://localhost/dwes06/xajax_calc.php**), donde aparece la interfaz de usuario y donde se realizan las llamadas ajax vía JavaScript. Su cometido es:
 - incluir **xajax_calc.base.php**,
 - imprimir el javascript de xajax,
 - tener la interfaz HTML y realizar la invocación a los métodos JavaScript.

En este primer paso, vamos a crear `xajax_calc.base.php`, pero para crearlo tienes que tener en cuenta una cosa: el script que muestra la interfaz web (`xajax_calc.php`) será diferente al script que procesa las peticiones AJAX (`xajax_calc.server.php`). Para indicarlo, debemos crear la instancia de `xajax` de una forma especial, pasándole como parámetro el script que hará de servidor:

```
$xajax=new xajax('xajax_calc.server.php');
```

Aparte de esto, deberás registrar 5 funciones AJAX:

- ✓ Una función para establecer el valor de trabajo.
- ✓ Una función para sumar un número al valor de trabajo.
- ✓ Una función para restar un número al valor de trabajo.
- ✓ Una función para multiplicar el valor de trabajo por un número.
- ✓ Y otra función para dividir el valor de trabajo por otro número.

Esto significa poner sentencias del estilo a:

```
$xajax->register(XAJAX_FUNCTION,"operacion");
```

Recuerda que en el ejemplo anterior, `"operacion"` hace referencia a una función php (que tendrás que implementar en `xajax_calc.functions.php`).

Tarea online

1.- Descripción de la tarea

1.1.- Ejercicio 1:
creando la
estructura de
proyecto.

1.2.- Ejercicio 2:
implementando las
funciones.

1.3.- Ejercicio 3:
implementando el
servidor.

1.4.- Ejercicio 4: la
interfaz.

1.5.- Ejercicio 5:
registrar una
medición.

2.- Información de interés

3.- Evaluación de la tarea

Anexo. Licencia de recursos

1.2.- Ejercicio 2: implementando las funciones.

Si en `xajax_calc.base.php` registraste las funciones, en el script `xajax_calc.functions.php` toca crear dichas funciones. Tendrás que definir funciones, una por cada una de las operaciones registradas en `xajax_calc.base.php`.

A la hora de enfrentarte a la realización de cada una de las funciones ten en cuenta que si la operación registrada era:

```
$xajax->register(XAJAX_FUNCTION,"operacion");
```

Deberás crear una función con el nombre "operacion":

```
function operacion (...)  
{  
    $objResponse = new xajaxResponse();  
    ...  
    return $objResponse;  
}
```

Además, debes tener en cuenta que:

- ✓ Todas las funciones que debes crear deben **tener un único parámetro**.
- ✓ Dentro de cada función debe comprobarse si el valor recibido por parámetro es un número o no.
- ✓ Ten en cuenta que el valor de trabajo **se almacena en la sesión**. Esto significa que el valor de trabajo **no se recibe como parámetro**.
- ✓ Cada una de estas operaciones debe retornar una instancia de `xajaxResponse`.

Tarea online

1.- Descripción de la tarea

1.1.- Ejercicio 1: creando la estructura de proyecto.

1.2.- Ejercicio 2: implementando las funciones.

1.3.- Ejercicio 3: implementando el servidor.

1.4.- Ejercicio 4: la interfaz.

1.5.- Ejercicio 5: registrar una medicion.

2.- Información de interés

3.- Evaluación de la tarea

Anexo. Licencia de recursos

1.3.- Ejercicio 3: implementando el servidor.

Este es posiblemente el paso más fácil, dado que solo hay que completar el script `xajax_calc.server.php`. Para ello:

- ✓ incluye `xajax_calc.base.php`,
- ✓ incluye las funciones (del archivo `xajax_calc.functions.php`),
- ✓ procesa las peticiones a través del método `processRequest` (fíjate que es un método sin argumentos).

En este apartado, no hay que hacer mucho más.

Tarea online

1.- Descripción de la tarea

1.1.- Ejercicio 1:
creando la estructura de proyecto.

1.2.- Ejercicio 2:
implementando las funciones.

1.3.- Ejercicio 3:
implementando el servidor.

1.4.- Ejercicio 4:
la interfaz.

1.5.- Ejercicio 5:
registrar una medición.

2.- Información de interés

3.- Evaluación de la tarea

Anexo. Licencia de recursos

1.4.- Ejercicio 4: la interfaz.

Ahora ha llegado el momento de generar la interfaz, a través del script `xajax_calc.php`. En este script deberás:

- ✓ Incluir `xajax_calc.base.php`.
- ✓ Crear un HTML, similar al expuesto en el vídeo del ejercicio 1, donde aparezca:
 - Un `input` para proporcionar un número.
 - Un `input` deshabilitado para mostrar la salida.
 - 5 botones para realizar las acciones de:
 - Establecer valor de trabajo.
 - Multiplicar por.
 - Dividir por.
 - Sumar.
 - Restar.
- ✓ En la sección de la cabecera del HTML (`<HEAD>...</HEAD>`) tendrás que imprimir el javascript usando el método `printJavascript` de `xajax`.

A la hora de realizar esta interfaz, ten en cuenta que:

- ✓ El valor de trabajo se almacena en la sesión, por lo que al cargar `xajax_calc.php` desde el navegador (`http://localhost/DWES06/xajax_calc.php`) deberá aparecer en el `input` correspondiente el valor de trabajo que hubiera o hubiese en la sesión de ejecuciones anteriores. Si no hubiese valor de trabajo previo entonces se considera que el valor de trabajo es 0.

Una vez realizado esto, solo te queda ejecutar los diferentes métodos `xajax` desde javascript (`xajax_establecer`, `xajax_multiplicarpor`, etc.) cuando se pulse el botón correspondiente.

A la hora de usar estos métodos, ten en cuenta que:

- ✓ A nivel de Javascript, solo hay que invocar los métodos Javascript correspondientes al pulsar cada botón. Además, cada método (`xajax_establecer`, `xajax_multiplicarpor`, etc.) solo tendrá un único parámetro (el dato introducido en el `input` correspondiente). Realmente, no hay que hacer acciones adicionales.
- ✓ La ejecución de los métodos Javascript generados por `xajax` (`xajax_establecer`, `xajax_multiplicarpor`, etc.) deben desencadenar una petición AJAX a `xajax_calc.server.php`. Si has hecho bien los pasos anteriores, debería funcionar sin tener que hacer nada más.
- ✓ Recuerda que en los métodos de `xajax_calc.functions.php` debe retornar una instancia de `xajaxResponse`. A través de `xajaxResponse` se indican las acciones a realizar en la interfaz HTML con métodos como `assign`.

Tarea online

1.- Descripción de la tarea

1.1.- Ejercicio 1: creando la estructura de proyecto.

1.2.- Ejercicio 2: implementando las funciones.

1.3.- Ejercicio 3: implementando el servidor.

1.4.- Ejercicio 4: la interfaz.

1.5.- Ejercicio 5: registrar una medicion.

2.- Información de interés

3.- Evaluación de la tarea

Anexo. Licencia de recursos

1.5.- Ejercicio 5: registrar una medicion.

Siguiendo ahora un proceso análogo a los descritos en los 4 ejercicios anteriores, se pide que se realice una interfaz para registrar mediciones, en la misma tabla proporcionada en la tarea 5:

```
CREATE TABLE medicion (  
  fecha DATE NOT NULL,  
  tramo VARCHAR(10) NOT NULL,  
  estacion VARCHAR(45) NOT NULL,  
  recuento INT NOT NULL,  
  PRIMARY KEY (fecha, tramo, estacion));
```

Para realizar esto, deberás hacer 4 scripts diferentes, con funciones equivalentes a los de los ejercicios anteriores:

- ✓ `xajax_medicion.base.php`
- ✓ `xajax_medicion.functions.php`
- ✓ `xajax_medicion.server.php`
- ✓ `xajax_medicion.php` (http://localhost/dwes06/xajax_medicion.php)

En `xajax_medicion.php` habrá un formulario para solicitar:

- ✓ Fecha (día/mes/año).
- ✓ Estación.
- ✓ Tramo (`tramo1`, `tramo2`, `tramo3` o `tramo4`).
- ✓ Recuento.

`xajax_medicion.php` enviará los datos vía AJAX usando `xajax` al servidor. Al realizar la operación anterior, pueden ocurrir tres cosas diferentes:

- ✓ Los datos son correctos y se insertan en la base de datos. En este caso, se limpia el formulario y se muestra un texto indicando que la operación se llevó a cabo con éxito.
- ✓ Los datos no son correctos y no se insertan en la base de datos. En este caso, no se limpia el formulario, pero sí se muestra un texto indicando que la operación no se llevó a cabo porque había datos incorrectos.
- ✓ Los datos son correctos, pero no se pudo insertar en la base de datos. En este caso, tampoco se limpia el formulario y se muestra un texto indicando que la operación no se llevó a cabo.

Los datos se consideran válidos cuando:

- ✓ La fecha es correcta (es una fecha posible) y no es del futuro.
- ✓ El tramo es "tramo1", "tramo2", "tramo3" o "tramo4".
- ✓ La estación es un texto no vacío.
- ✓ El recuento es un número mayor o igual a 0.

Tarea online

1.- Descripción de la tarea

1.1.- Ejercicio 1:
creando la estructura
de proyecto.

1.2.- Ejercicio 2:
implementando las
funciones.

1.3.- Ejercicio 3:
implementando el
servidor.

1.4.- Ejercicio 4: la
interfaz.

1.5.- Ejercicio 5:
registrar una medicion.

2.- Información de interés

3.- Evaluación de la tarea

Anexo. Licencia de
recursos

2.- Información de interés

Recursos necesarios y recomendaciones

Recomendaciones específicas de esta tarea

- ✓ Es recomendable que configures el depurador de PHP.
- ✓ No dudes en ver el código (y leer la documentación):
 - ➔ Clase XAJAX
https://github.com/Xajax/Xajax/blob/master/xajax_core/xajax.inc.php
 - ➔ Clase xajaxResponse
https://github.com/Xajax/Xajax/blob/master/xajax_core/xajaxResponse.inc.php
- ✓ En el repositorio de github de xajax encontrarás ejemplos que te serán muy útiles:
 - ➔ <https://github.com/Xajax/Xajax/tree/master/examples>
- ✓ Es muchas instalaciones de php los errores está configurados para no mostrarse al usuario. Mientras estas desarrollando una aplicación es imprescindible que los errores de php se muestre, de forma que podamos corregir cualquier aplicación antes de que llegue al usuario final. En el siguiente enlace se explica como:

 [¿Cómo mostrar errores.php?](#)

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 si que todos los **archivos vayan codificados en UTF-8** y que **las 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 y no tendrás que cambiar de herramienta.
- ✓ No abordes la tarea hasta que hayas repasado todos los contenidos, y hecho a 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_Tarea06

Tarea online

1.- Descripción de la tarea

1.1.- Ejercicio 1:
creando la estructura de proyecto.

1.2.- Ejercicio 2:
implementando las funciones.

1.3.- Ejercicio 3:
implementando el servidor.

1.4.- Ejercicio 4:
la interfaz.

1.5.- Ejercicio 5:
registrar una medición.

2.- Información de interés

3.- Evaluación de la tarea

Anexo. Licencia de recursos

3.- Evaluación de la tarea

Criterios de evaluación implicados

- ✓ Se han identificado las diferencias entre la ejecución de código en el servidor y en el cliente Web.
- ✓ Se han reconocido las ventajas de unir ambas tecnologías en el proceso de desarrollo de programas.
- ✓ Se han identificado las librerías y las tecnologías relacionadas con la generación por parte del servidor de páginas Web con guiones embebidos.
- ✓ Se han utilizado estas tecnologías para generar páginas Web que incluyan interacción con el usuario en forma de advertencias y peticiones de confirmación.
- ✓ Se han utilizado estas tecnologías, para generar páginas Web que incluyan verificación de formularios.
- ✓ Se han utilizado estas tecnologías para generar páginas Web que incluyan modificación dinámica de su contenido y su estructura.
- ✓ Se han aplicado estas tecnologías en la programación de aplicaciones Web.



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

¿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 1 puntos
Se ha implementado el ejercicio 2 apropiadamente y funciona correctamente.	hasta 1 puntos

Se ha implementado el ejercicio 3 apropiadamente y funciona correctamente.	hasta 1 puntos
Se ha implementado el ejercicio 4 apropiadamente y funciona correctamente.	hasta 3 puntos
Se ha implementado el ejercicio 5 apropiadamente y funciona correctamente.	hasta 4 puntos