DESARROLLO DE APLICACIONES WEB

5-3-2024

ROBERTO RODRÍGUEZ JIMÉNEZ

roberto.rodjim.1@educa.jcyl.es

Desarrollo Web en Entorno Servidor

Servicios web

Tarea 06

# 

Contenido

[Tarea online DWES03 2](#_Toc159151763)

[1.- Descripción de la tarea 2](#_Toc159151764)

[Enunciado 2](#_Toc159151765)

[**Caso práctico** 2](#_Toc159151766)

[2.- Información de interés 5](#_Toc159151767)

[Recursos necesarios 5](#_Toc159151768)

[Consejos y recomendaciones 5](#_Toc159151769)

[3.- Evaluación de la tarea 5](#_Toc159151770)

[Criterios de evaluación implicados 5](#_Toc159151771)

[RESOLUCIÓN 7](#_Toc159151772)

[Dependencias obligatorias para instalar con Composer. 7](#_Toc159151773)

[Estructura del proyecto 8](#_Toc159151774)

[Código 9](#_Toc159151775)

[SQL 9](#_Toc159151776)

# Tarea online DWES03

**Título de la tarea**: Creando y consumiendo servicios web.  
**Unidad 6**: Servicios web.  
**Ciclo formativo y módulo**: Desarrollo de Aplicaciones Web. Módulo DWES.  
**Curso académico**: 2023-24

## 1.- Descripción de la tarea

### Enunciado

### **Caso práctico** Carlos ha estado pensando cómo hacer que se puedan consultar los precios y otros datos de una tienda online, para que estos puedan estar disponibles en páginas de ofertas de una manera sencilla. Además, piensa que puede ser una buena idea y que es la forma usual de trabajar de muchos portales web.

### Carlos se pone a pesar sobre ello y decide que con los conocimientos adquiridos podrá llevar a cabo parte de la tarea. Animado se propone a consultar a Juan para que le eche una mano.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteVamos a seguir utilizando para esta tarea la base de datos correspondiente a la tienda web.

En primer lugar, deberás crear una base de datos de nombre "tarea6", similar a la que usamos en los ejercicios anteriores. Para acceder puedes reutilizar el usuario "gestor" y su contraseña "secreto".

A continuación, utilizarás "PHP SOAP" para crear un servicio web con cuatro funciones que expongan información de la base de datos de la tienda online. En el primer servidor no deberás utilizaremos ningún archivo WSDL. Después generaremos uno y lo usaremos.

Échale un vistazo a la estructura de la práctica propuesta para que te hagas una primera idea de los archivos que se piden.

Para la base de datos se deja en un archivo comprimido los archivos para crear la base de datos "tarea6" y sus tablas, y el archivo para guardar datos de ejemplos para las tablas: [Archivos](https://fpdistancia.educa.jcyl.es/pluginfile.php/240423/mod_resource/content/1/DWES06_v1/ArchivosUnidad/Moodle/DWES06_Tarea/DWES06_TAR_R02_sql.zip) (zip - 8392 B)

Tendremos las carpetas:

* "public": donde crearemos los ficheros para llamar a los servicios que crearemos.
* "servidorSoap": donde generaremos nuestros servicios.
* "src": Guardaremos todas nuestras clases, la clase "Operaciones" con todas las funciones que nos ofrecerá, la clase "Conexion", y las clases "Producto, Familia y Stock", con los métodos para acceder a la bases de datos y gestionar las tablas, como vimos en unidades anteriores.
* "vendor": La generará automáticamente Composer.

Para esta práctica se pedirá "autoload" de las clases, "php2wsdl" y "wsdl2phpgenerator, todo instalado con "Composer"

* Crearemos el servidor SOAP "servicio.php" en la carpeta "servidorSoap". Este servicio no tendrá asociado ningún archivo WSDL y ofrecerá las funciones siguientes (Todas implementadas en "src/Operaciones.php"):
  + **getPVP**. Esta función recibirá como parámetro el código de un producto, y devolverá el PVP correspondiente al mismo.
  + **getStock**. Esta función recibirá dos parámetros: el código de un producto y el código de una tienda. Devolverá el stock existente en dicha tienda del producto.
  + **getFamilias**. No recibe parámetros y devuelve un array con los códigos de todas las familias existentes.
  + **getProductosFamilia**. Recibe como parámetro el código de una familia y devuelve un array con los códigos de todos los productos de esa familia.
* Para comprobar la correcta ejecución del servicio, programa también un cliente con nombre "cliente.php" en la carpeta "public" que realice una llamada a cada una de las funciones programadas y muestre el resultado obtenido.
* Con la extensión "php2wsdl" crea el archivo "public/generarWsdl.php" para generar el documento WSDL "servicio.wsdl" que guardaremos en "servidorSoap". Utilizando este documento nos crearemos un nuevo servidor SOAP, "servicioW.php" que lo utilice.
* Para comprobar la correcta ejecución de este nuevo servicio, programa también un cliente con nombre"clienteW.php" en la carpeta "public" que realice una llamada a cada una de las funciones programadas y muestre el resultado obtenido.
* Partiendo de este nuevo servicio y de su descripción crearemos el fichero "public/generarClases", que utilizará la herramienta "wsdl2php" para obtener una clase PHP. Esta clase la guardaremos en la carpeta "Clases1" dentro de "src".
* Crea un nuevo cliente llamado"clienteW2.php"que se base en esta clase, para probar el nuevo servicio, mostrando los resultados obtenidos de forma similar a como hiciste en los casos anteriores.

## 2.- Información de interés

### Recursos necesarios

**Recursos necesarios para realizar la Tarea:**

1. Ordenador con entorno AMP instalado y correctamente configurado.
2. Extensión de depuración Xdebug para PHP instalada.
3. IDE como VSC instalado y funcionando.
4. Extensiones de VSC como PHP Debug instaladas.

### Consejos y recomendaciones

* Además del manual online de PHP, se recomienda dar libre acceso a Internet para la búsqueda de información.

## 

## 3.- Evaluación de la tarea

### Criterios de evaluación implicados

* Se han reconocido las características propias y el ámbito de aplicación de los servicios Web.
* Se han reconocido las ventajas de utilizar servicios Web para proporcionar acceso a funcionalidades incorporadas a la lógica de negocio de una aplicación.
* Se han identificado las tecnologías y los protocolos implicados en la publicación y utilización de servicios Web.
* Se ha programado un servicio Web.
* Se ha creado el documento de descripción del servicio Web.
* Se ha verificado el funcionamiento del servicio Web.
* Se ha consumido el servicio Web.

|  |  |
| --- | --- |
| **Rúbrica de la tarea (10 Puntos)** | |
| Crear la estructura del proyecto e instalar todas las dependencias necesarias con Composer | 1.5 Puntos |
| Programar correctamente las clases y funciones para manipular las base de datos | 1.25 Puntos |
| Programar y comentar adecuadamente la clase "Operaciones.php", para posibilitar la utilización de las funciones pedidas y la generación del documento WSDL | 1.5 Puntos |
| Crear de forma correcta "servicio.php" y "cliente.php" para testar su funcionamiento | 1.25 Puntos |
| Generar el archivo "servicio.wsdl" correctamente usando "php2wsdl" | 1.25 Puntos |
| Crear de forma correcta "servicioW.php" para que use el archivo WSDL generado y  "clienteW.php" para testar su funcionamiento | 1.25 Puntos |
| Generar correctamente la clase en la carpeta pedida a partir del archivo WSDL con la extensión "wsdl2php" y comprobar el funcionamiento de dicha clase con "clienteW2.php" | 1.5 Puntos |
| Uso correcto de "namespaces" y comentarios adecuados en el código | 0.5 Puntos |

## RESOLUCIÓN

Interfaz de usuario gráfica, Texto

Descripción generada automáticamenteGenerar la estructura básica antes de instalar Composer

Iniciar Composer

{

    "name": "roberto/tarea6",

    "description": "DWES06 Tarea",

    "type": "project",

    "license": "GPL",

    "autoload": {

        "psr-4": {

            "Clases\\": "src/"

        }

    },

    "authors": [

        {

            "name": "Roberto",

            "email": "roberto.rodjim.1@educa.jcyl.es"

        }

    ],

    "minimum-stability": "dev",

    "require": {

        "php2wsdl/php2wsdl": "dev-master",

        "wsdl2phpgenerator/wsdl2phpgenerator": "^3.5@dev"

    },

    "config": {

        "platform": {

            "php":"7.2.33"

        },

        "optimize-autoloader": true

    }

}

Librerías:

composer require wsdl2phpgenerator/wsdl2phpgenerator

Composer require php2wsdl/php2wsdl

"require": {

        "php2wsdl/php2wsdl": "dev-master",

        "wsdl2phpgenerator/wsdl2phpgenerator": "^3.5@dev"

    },

Implementamos la clase Operaciones, aunque no le damos funcionalidad.

Generamos el archivo wsdl en el script generarWsdl.php.