

ciclo: DWes

MÓDULO DE DESARROLLO WEB EN ENTORNO SERVIDOR

Tarea N.º 07

Alumno:

Ricardo Núñez Muñiz

49726340H

*Los documentos, elementos gráficos, vídeos, transparencias y otros recursos didácticos incluidos en este contenido pueden contener imprecisiones técnicas o errores tipográficos. Periódicamente se realizan cambios en el contenido. Fomento Ocupacional FOC SL puede realizar en cualquier momento, sin previo aviso, mejoras y/o cambios en el contenido.*

*Es responsabilidad del usuario el cumplimiento de todas las leyes de derechos de autor aplicables. Ningún elemento de este contenido (documentos, elementos gráficos, vídeos, transparencias y otros recursos didácticos asociados), ni parte de este contenido puede ser reproducida, almacenada o introducida en un sistema de recuperación, ni transmitida de ninguna forma ni por ningún medio (ya sea electrónico, mecánico, por fotocopia, grabación o de otra manera), ni con ningún propósito, sin la previa autorización por escrito de Fomento Ocupacional FOC SL.*

*Este contenido está protegido por la ley de propiedad intelectual e industrial. Pertenecen a Fomento Ocupacional FOC SL los derechos de autor y los demás derechos de propiedad intelectual e industrial sobre este contenido.*

*Sin perjuicio de los casos en que la ley aplicable prohíbe la exclusión de la responsabilidad por daños, Fomento Ocupacional FOC SL no se responsabiliza en ningún caso de daños indirectos, sean cuales fueren* *su naturaleza u origen, que se deriven o de otro modo estén relacionados con el uso de este contenido.*

*© 2020 Fomento Ocupacional FOC SL todos los derechos reservados*

Contenido

1. RA7\_d) Se han utilizado los estándares y arquitecturas más difundidos e implicados en el desarrollo de servicios web.
2. RA7\_f) Se ha verificado el funcionamiento del servicio Web.
3. RA7\_g) Se ha consumido el servicio Web.
4. RA7\_h) Se ha documentado el servicio Web.

Para realizar esta tarea es aconsejable seguir los siguientes pasos:

1. Utilizar la base de datos de la tarea 6.

2. Modificar la api para que cargue los datos de la base de datos. Utilizar la clase de acceso a datos de la tarea 6.  
3. Añadir nuevas funcionalidades a la api. En clase analizaremos la api actual y se propondrá nuevo código.  
4. Utilizar las nuevas funcionalidades en el cliente, cuidando la maquetación y la usabilidad, permitiendo probar toda la aplicación.

5. Probar la aplicación.

6. Generar la documentación, mediante phpDocumentor.

7. Publicar y probar la página y la documentación generada con phpDocumentor en un servidor web gratuito (o propietario), como Byethost o 000webhost. No olvidar incluir la url.

 Describir todos los procesos seguidos en la documentación entregada.

1. RA7\_d) Se han utilizado los estándares y arquitecturas más difundidos e implicados en el desarrollo de servicios web.

Esta tarea consiste en modificar una api ya existente. Para ello se proporcionan dos archivos api.php y cliente.php, en el siguiente repositorio de github: <https://github.com/gitfoc/tarea7>.

La api realiza únicamente dos funciones:

* retorna una lista de autores en formato JSON.
* retorna la información referente a un autor específico en formato JSON.

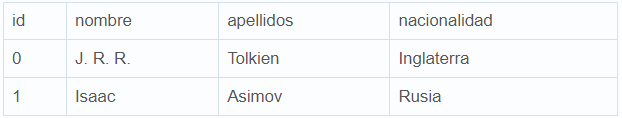
El cliente realiza dos peticiones a la api, formateando los datos recibidos:

* Solicita una lista de autores, convierte el formato JSON a una lista de objetos y muestra la lista de autores.
* Solicita la información específica de un autor, convierte el formato JSON un objeto y muestra la información del autor.

Podéis probar el ejemplo modificando las URL de las peticiones del cliente a la API y adaptándolas al directorio donde se encuentre en vuestro sistema. Para comprobar su funcionamiento sólo debéis ejecutar el cliente en vuestro navegador.

* Un ejemplo de las llamadas correctas a la api tal y como están en la entrega son las siguientes:
  + http://localhost/dwes/rest/cliente.php?action=get\_listado\_autores
  + http://localhost/dwes/rest/cliente.php?action=get\_datos\_autor&id=0
* Utilizar la base de datos Libros, de la tarea 6. Recordemos que se utilizan dos tablas.
  + Tabla Autor que contenga los siguientes campos:
    - Una columna id de tipo entero.
    - Una columna nombre de tipo cadena con una longitud máxima de 15 caracteres.
    - Una columna apellidos de tipo cadena con una longitud máxima de 25 caracteres.
    - Una columna nacionalidad de tipo alfanumérica con una longitud máxima de 10 caracteres.

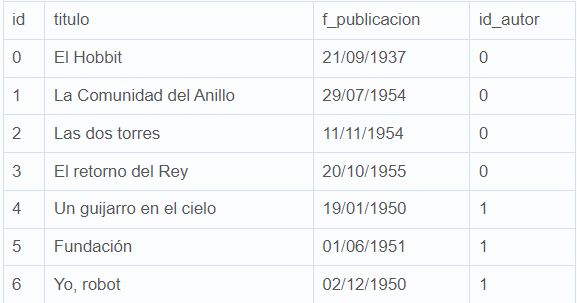
Esta tabla tiene el contenido mostrado a continuación:



La tabla **Libro** contiene los siguientes campos:

* Una columna **id**de tipo entero.
* Una columna **titulo**de tipo cadena con una longitud máxima de 50 caracteres.
* Una columna **f\_publicacion**de tipo alfanumérica con una longitud máxima de 10 caracteres.
* Una columna **id\_autor** como clave externa del id autor.

Esta tabla tiene el siguiente contenido:



* **Modificación del servicio API**

Como veréis, la API de nuestro servicio recupera los datos de un array (simulando un acceso a datos), el primer paso es recuperar esos datos de la**base de datos**. Por supuesto, **se recomienda utilizar la clase de acceso a datos ya desarrollada en la tarea 6**:

* + Debéis recuperar de la base de datos todos los datos que devuelve la función get\_listado\_autores().
  + Debéis recuperar de la base de datos todos los datos que devuelve la función get\_datos\_autor().
  + Debéis crear una nueva función get\_listado\_libros() que retorne su id y título. (Para ello debéis añadir el nombre de la función en el array *posibles\_URL*y abrir un nuevo *case* en el *switch*que recibe las peticiones)
  + Debéis crear una nueva función get\_datos\_libro() que reciba como parámetro un id y retorne título y f\_publicacion. Además, a partir de la columna id\_autor debéis recuperar el nombre y apellidos del autor y retornarlos. En definitiva, esta función devolverá (titulo, f\_publicacion, id\_autor, nombre, apellidos).
  + Tendréis que modificar la función get\_datos\_autor() y hacer que retorne una lista de libros que ha escrito, leídos de la base de datos.
* **Modificación del cliente**

Como hemos visto antes, el cliente sólo realiza llamadas y procesa las respuestas JSON que le retorna la API (añadiendo maquetación). Para aprovechar las nuevas funciones que hemos incluido a nuestra API vamos a modificar el cliente.

* + Tendréis que mostrar una lista de los libros de la base de datos. (De la misma manera que se muestra la lista de autores)
  + Cada libro deberá hacer referencia a la información específica del libro mediante un enlace. (De la misma manera que se muestra la lista de autores pero por medio de get\_datos\_libro en lugar de get\_datos\_autor).
  + Debéis modificar la visualización que se muestra del autor y añadir una lista de libros que haya escrito referenciándolos de la misma manera que el punto anterior. (La lista de libros se mostrará junto con la información del autor que ya se mostraba anteriormente). Cada libro contiene un enlace a los datos del libro.
  + Los enlaces a la información específica de un libro deben conducir a una pantalla que mostrará todos los campos de la tabla de libros (Siguiendo el formato de visualización de la información de un autor). El campo autor del libro contendrá un enlace a los datos del autor.

1. RA7\_f) Se ha verificado el funcionamiento del servicio Web.

* Probar la api, para garantizar su correcto funcionamiento.

1. RA7\_g) Se ha consumido el servicio Web.

* Desarrollar una página sencilla que llame al cliente, maquetando la salida, y que permita probar el servicio web, y navegar por todas las funcionalidades. Se recomienda mostrar siempre un enlace a la lista de autores y de libros. Se valorará el aspecto gráfico de la solución.

1. RA7\_h) Se ha documentado el servicio Web.

* Utilizando la sintaxis básica de PHPDoc, comentar las funciones creadas. Crear un comentario /\*\* \*/ en cada función que incluya un comentario sobre la función
* Utilizar @param para describir los parámetros que recibe la función y @return para comentar el valor resuelto
* Se puede verificar el correcto funcionamiento de los comentarios creados usando PHPDocumentor.
* Descargar el fichero phpdocumentor.phar de la página de PHPDocumentor
* Crear un directorio doc fuera del directorio donde se esté creando la página web
* Ejecutar c:\xampp\php\bin\php.exe phpDocumentor.phar -d c:\directorio de trabajo -t c:\directorio de documentacion creado. Estos directorios son de ejemplo..
* Tras esto se debería haber creado una página web con el contenido de la documentación.