



**CICLO: CFGS DESARROLLO DE  
APLICACIONES WEB  
MÓDULO DE DESARROLLO WEB  
ENTORNO SERVIDOR**

# **[TAREA N° 7]**

**Andrea A. Goitia Rodríguez**

**DNI: 26066107T**

*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.Documentos que se adjuntan a este informe.**  
..... **4**

**2.Resto de epígrafes que componen el desarrollo de este informe.**  
..... **4**

## 1. Documentos que se adjuntan a este informe.

A continuación se detallan los documentos que componen la presente entrega de la tarea:

1. Informe de elaboración de la tarea.
2. Cliente.php
3. index.php
4. api.php
5. Libros.php
6. Carpeta DOCS

## 2. Resto de epígrafes que componen el desarrollo de este informe.

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

### Modificación del servicio API

Debéis recuperar de la base de datos todos los datos que devuelve la función `get_listado_autores()`.

```
require_once 'Libros.php';
$host='localhost';
$user = 'root';
$password= '';
$databse= 'libros';

function get_listado_autores(){
    $object = new Libros();
    global $host, $user, $password, $databse;
    $connect= $object->conexion($host, $user, $password, $databse);
    $lista_autores = $object->consultarAutores($connect);
    return $lista_autores;
}
```

Creamos un objeto de la clase Libros para hacer uso de sus métodos. Luego declaramos las variables usadas para la conexión con la base de datos, como globales para poder usarlas dentro de esta función. Ahora establecemos la conexión con la base de datos a través del método conexión de la clase Libros.

Hacemos la llamada a la función de la clase Libros 'consultarAutores' y el objeto que retorna la guardamos en la variables 'lista\_autores'.

Por último, devolvemos esa variable.

Método conexión:

```
function conexion($servidor, $usuario, $contrasena, $bd){
    try{
        $connection = new mysqli($servidor, $usuario, $contrasena, $bd);

        if(!$connection){
            $connection= null;
        }
        return $connection;
    } catch (mysqli_sql_exception $ex) {
        echo $ex->getMessage();
    }
}
```

## Método consultar\_autores:

```
function consultarAutores($connect, $id_autor = null){
    try{
        $info = array();
        // $connect = $this->conexion($this->host, $this->user, $this->password, $this->database);
        if($id_autor == null){
            $query = "SELECT * FROM autor";
            $res = $connect->query($query);

            while($fila = $res->fetch_object()){
                $info[] = $fila;
            }
            $res->free();
            $connect->close();
        }else{
            $query = "SELECT * FROM autor WHERE id = {$id_autor}";
            $res = $connect->query($query);
            // comprueba que el id existe. Sino existe, devuelve num_rows == 0
            if($res->num_rows == 0){
                echo "El ID que ingresaste no existe, intenta con uno diferente ";
                return null;
            }else{
                $fila = $res->fetch_object();
                $info = $fila;
            }
            $res->free();
            $connect->close();
        }
    }
}
```

Debéis recuperar de la base de datos todos los datos que devuelve la función `get_datos_autor()`.

```
function get_datos_autor($id){
    $object = new Libros();
    global $host, $user, $password, $database;
    $connect = $object->conexion($host, $user, $password, $database);
    $info_libro = $object->consultarLibros($connect, $id);

    // Buscamos los datos personales del autor
    $datos_personales = $object->consultarAutores($connect, $id);

    $info_autor = new stdClass();
    $info_autor->datos = $datos_personales;
    $info_autor->libros = $info_libro;

    return $info_autor;
}
```

Creamos un objeto de la clase Libros para hacer uso de sus métodos. Luego declaramos las variables usadas para la conexión con la base de datos, como globales para poder usarlas dentro de esta función. Ahora establecemos la conexión con la base de datos a través del método `conexion` de la clase Libros.

Hacemos la llamada a la función de la clase Libros '`consultarLibros`', en el cual introducimos como parámetro la conexión y el id, y el objeto que retorna lo guardamos en la variable '`info_libro`'.

Hacemos la llamada a la función '`consultarAutores`', la cual retorna un objeto que es introducido en la variable '`datos_personales`'.

La función `get_datos_autor` devuelve la variable '`info_autor`', la cual es un objeto que hemos creado para poder asignarle como atributos '`datos`', el cual se refiere al contenido de la variable '`datos_personales`' y el atributo '`libros`', el cual se refiere al contenido de la variable '`info_libro`'.

## Método consultarLibros:

```

10 function consultarLibros($connect, $id_autor= null){
11     try{
12         $info =array();
13         // $connect = $this->conexion($this->host,$this->user, $this->password, $this->database);
14
15         if($id_autor == null){
16             $query = "SELECT * FROM libro";
17             $res = $connect->query($query);
18
19             while($fila = $res->fetch_object()){
20                 $fila->f_publicacion = fecha($fila->f_publicacion);
21                 $info[] = $fila;
22             }
23             $res->free();
24             $connect->close();
25         }else{
26             $query = "SELECT * FROM libro WHERE (id_autor) = id_autor";
27             $res = $connect->query($query);
28
29             if($res->num_rows == 0){
30                 echo " El ID que ingresaste no existe, intenta con uno diferente ";
31                 $info = null;
32             }
33
34             while($fila = $res->fetch_object()){
35                 $fila->f_publicacion = fecha($fila->f_publicacion);
36                 $info[] = $fila;
37             }
38             $res->free();
39             $connect->close();
40         }
41     }
42     return $info;
43 }

```

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)

```

function get_listado_libros(){
    $object = new Libros();
    global $host, $user, $password, $database;
    $connect= $object->conexion($host, $user, $password, $database);

    $lista_libro = $object->consultarLibros($connect);

    return $lista_libro;
}

```

```

$posibles_URL = array("get_listado_autores", "get_datos_autor", "get_listado_libros",

```

```

switch ($_GET["action"])
{
    case "get_listado_autores":
        $valor = get_listado_autores();
        break;
    case "get_datos_autor":
        if (isset($_GET["id"]))
            $valor = get_datos_autor($_GET["id"]);
        else
            $valor = "Argumento no encontrado";
        break;
    case "get_listado_libros":
        $valor= get_listado_libros();
        break;
}

```

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).

```

function get_datos_libro($id){
    $object = new Libros();
    global $host, $user, $password, $database;
    $connect= $object->conexion($host, $user, $password, $database);

    $datos_libro = array();
    //Busca los datos de 1 libro
    $query_1 = "SELECT titulo, f_publicacion FROM libro WHERE {$id} = id";

    $res_1 = $connect->query($query_1);
    if($res_1->num_rows == 0){
        echo " El ID que ingresaste no existe, intenta con uno diferente ";
        $datos_libro = null;
    }else{
        $datos_libro = $res_1->fetch_object();
        $datos_libro->f_publicacion = fecha($datos_libro->f_publicacion);
    }
    $res_1->free();

    //Busca los datos del autor
    $datos_autor = array();
    $query_2 = "SELECT autor.nombre, autor.apellidos, autor.id FROM autor INNER JOIN libro ON autor.id = libro.id_autor WHERE libro.id={$id}";

    $res_2 = $connect->query($query_2);
    if($res_2->num_rows == 0){
        echo " El ID que ingresaste no existe, intenta con uno diferente ";
        $datos_autor = null;
    }else{
        $datos_autor = $res_2->fetch_object();
    }
    $res_2->free();
    $connect->close();
    //creamos clase
    $todos_datos = new stdClass();
    $todos_datos->libro = $datos_libro;
    $todos_datos->autor = $datos_autor;

    return $todos_datos;
}

```

Después de realizar las dos consultas, se crea un objeto llamada 'todo\_datos', el cual va a ser el objeto que retorne la función.

Y a éste objeto se le asignan dos atributos: 'libro', el cual devuelve los datos del libro que se ha consultado. Y el atributo 'autor', el cual guarda los datos del autor que se ha consultado.

Tendréis que modificar la función get\_datos\_autor() y hacer que retorne una lista de libros que ha escrito, léidos de la base de datos.

Dentro de la función, se agrega esta línea:

```
$info_libro = $object->consultarLibros($connect, $id);
```

## Modificación del 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)

```

<ul class="libros-lista">
<h2>Libros</h2>
<?php foreach ($lista_libros as $libros) { ?>
    <li>
        <a href="<?php echo "http://localhost/Tarea_7/index.php?action=get_datos_libro&id=" . $libros->id ?>">
            <?php echo $libros->id . " " . $libros->titulo ?>
        </a>
    </li>
<?php } ?>
</ul>
</div>
<?php
~

```

## Libros

- [0 El Hobbit](#)
- [1 La Comunidad del Anillo](#)
- [2 Las dos torres](#)
- [3 El retorno del Rey](#)
- [4 Un guijarro en el cielo](#)
- [5 Fundación](#)
- [6 Yo robot](#)

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`).

El link dentro de cada li:

```
<a href="<?php echo "http://localhost/Tarea_7/index.php?action=get_datos_libro&id=" . $libros->id ?>">
```

Después de hacer clic, te lleva a:

```
<?php )else if(isset($_GET["action"]) && isset($_GET["id"]) && $_GET["action"] == "get_datos_libro") {  
    //Se realiza la petición a la api que nos devuelve el JSON con la información de los autores  
    $app_info = file_get_contents('http://localhost/Tarea_7/api.php?action=get_datos_libro&id=' . $_GET["id"]);  
    //decodificar JSON, a array  
    $app_info = json_decode($app_info);  
?  
    <div class="datos-libro">  
        <h2>Datos del libro</h2>  
        <p>Título: <?php echo $app_info->libro->titulo; ?></p>  
        <p>Fecha de Publicación: <?php echo $app_info->libro->f_publicacion; ?></p>  
    </div>
```

## Datos del libro

Título: La Comunidad del Anillo

Fecha de Publicación: 29/07/1954

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 libros que escribe el autor:

```
<!-- Mostramos los libros del autor -->  
<h2>Libros escritos por este Autor</h2>  
<?php foreach($app_info->libros as $libro): ?>  
    <li>  
        <a href="<?php echo "http://localhost/Tarea_7/index.php?action=get_datos_libro&id=" . $libro->id ?>">  
        <?php echo $libro->titulo; ?>  
        </a>  
    </li>  
<?php endforeach; ?>
```

## Datos de Autor

Nombre: Isaac

Apellidos: Asimov

Fecha de nacimiento: Rusia

## Libros escritos por este Autor

- [Un guijarro en el cielo](#)
- [Fundación](#)
- [Yo robot](#)



Al hacer clic sobre un libro, me lleva a los datos del ese libro:

```
<a href="<?php echo "http://localhost/Tarea 7/index.php?action=get_datos_libro&id=" . $libro->id ?>">
```

### Datos del libro

Título: Fundación

Fecha de Publicación: 01/06/1951

### Datos de su autor

Nombre: Isaac

Apellidos: Asimov

[Volver a la lista de autores](#)

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.

Cuando pinchas sobre libro:

```
<?php }else if(isset($_GET["action"]) && isset($_GET["id"]) && $_GET["action"] == "get_datos_libro") {  
    //Se realiza la petición a la api que nos devuelve el JSON con la información de los autores  
    $app_info = file_get_contents('http://localhost/Tarea 7/api.php?action=get_datos_libro&id=' . $_GET["id"]);  
    //decodificar JSON, a array  
    $app_info = json_decode($app_info);  
?  
    <div class="datos-libro">  
        <h2>Datos del libro</h2>  
        <p>Título: <?php echo $app_info->libro->titulo; ?></p>  
        <p>Fecha de Publicación: <?php echo $app_info->libro->f_publicacion; ?></p>  
    </div>  
    <div class="autor-libro">  
        <a href="<?php echo "http://localhost/Tarea 7/index.php?action=get_datos_autor&id=" . $app_info->autor->id ?>">  
        <h2>Datos de su autor</h2>  
        <p>Nombre: <?php echo $app_info->autor->nombre; ?></p>  
        <p>Apellidos: <?php echo $app_info->autor->apellidos; ?></p>  
    </a>  
    </div>
```

### Datos del libro

Título: El retorno del Rey

Fecha de Publicación: 20/10/1955

### Datos de su autor

Nombre: J.R.R

Apellidos: Tolkien

[Volver a la lista de autores](#)

Cuando pinchas sobre un autor, estando dentro de los datos del libro:

```
<a href="<?php echo "http://localhost/Tarea 7/index.php?action=get_datos_autor&id=" . $app_info->autor->id ?>">
```

### Datos de Autor

Nombre: J.R.R

Apellidos: Tolkien

Fecha de nacimiento: Inglaterra

### Libros escritos por este Autor

- [El Hobbit](#)
- [La Comunidad del Anillo](#)
- [Las dos torres](#)
- [El retorno del Rey](#)

[Volver a la lista de autores](#)

RA7\_f) Se ha verificado el funcionamiento del servicio Web.  
Probar la api, para garantizar su correcto funcionamiento.

## Listas

### Autores

- [J.R.R Tolkien](#)
- [Isaac Asimov](#)

### Libros

- [0 El Hobbit](#)
- [1 La Comunidad del Anillo](#)
- [2 Las dos torres](#)
- [3 El retorno del Rey](#)
- [4 Un guijarro en el cielo](#)
- [5 Fundación](#)
- [6 Yo, robot](#)

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.

```
<body>
  <header class="header">

  </header>

  <main>
    <?php require 'cliente.php'; ?>

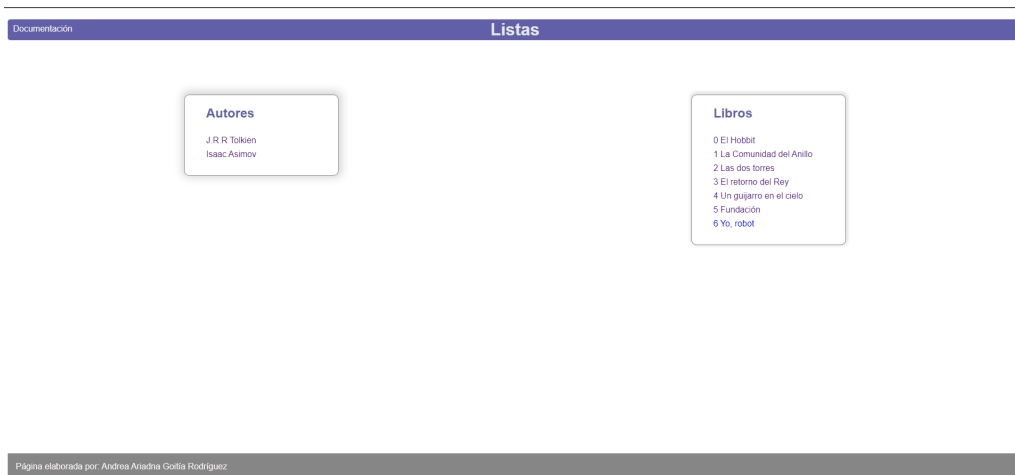
  </main>
</div>

<p>
  <a href="DOCS/index.html">Documentación</a>
</p>
<footer>
  <p>Página elaborada por: Andrea Ariadna Goitia Rodríguez</p>
</footer>
</body>
```

Algunos estilos que se crearon:

```
body{
  h1{
    text-align: center;
    font-family: sans-serif;
    background-color: #5a59a2;
    color: #e7e7e7;
    border: 1px solid transparent;
    border-radius: 5px;
  }
  h2{
    font-family: sans-serif;
    color: #5a59a2;
  }
  li{
    list-style: none;
  }
  .volver{
    position: relative;
    top: 200px;
    float: right;
    padding: 10px;
    margin: 5px;
    background-color: lightgray;
  }
  li a, a{
    font-family: sans-serif;
    text-decoration: none;
  }
  p{
    font-family: sans-serif;
  }
}
```

```
.autores-lista{
  margin-top: 80px;
  width: 13%;
  border: 1px solid gray;
  border-radius: 10px;
  padding-bottom: 30px;
  -webkit-box-shadow: 1px -2px 14px 6px rgba(171,171,171,0.47);
  -moz-box-shadow: 1px -2px 14px 6px rgba(171,171,171,0.47);
  box-shadow: 1px -2px 14px 6px rgba(171,171,171,0.47);
}
.libros-lista{
  width: 13%;
  border: 1px solid gray;
  border-radius: 10px;
  padding-bottom: 30px;
  -webkit-box-shadow: 1px -2px 4px 4px rgba(204,204,204,0.48);
  -moz-box-shadow: 1px -2px 4px 4px rgba(204,204,204,0.48);
  box-shadow: 1px -2px 4px 4px rgba(204,204,204,0.48);
}
.lista-l{
  display: flex;
  flex-direction: row;
  flex-wrap: nowrap;
  justify-content: space-around;
  align-items: baseline;
}
```



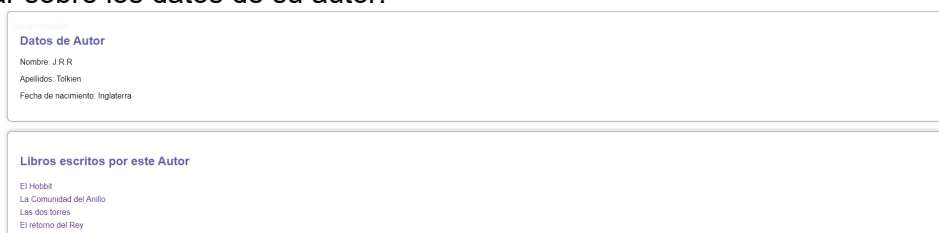
Al pinchar sobre un autor:



Al pinchar sobre algún libros escrito por ese autor:



Al pinchar sobre los datos de su autor:



Al pinchar sobre volver a la lista de autores:



Al pinchar sobre algún libro de la lista:



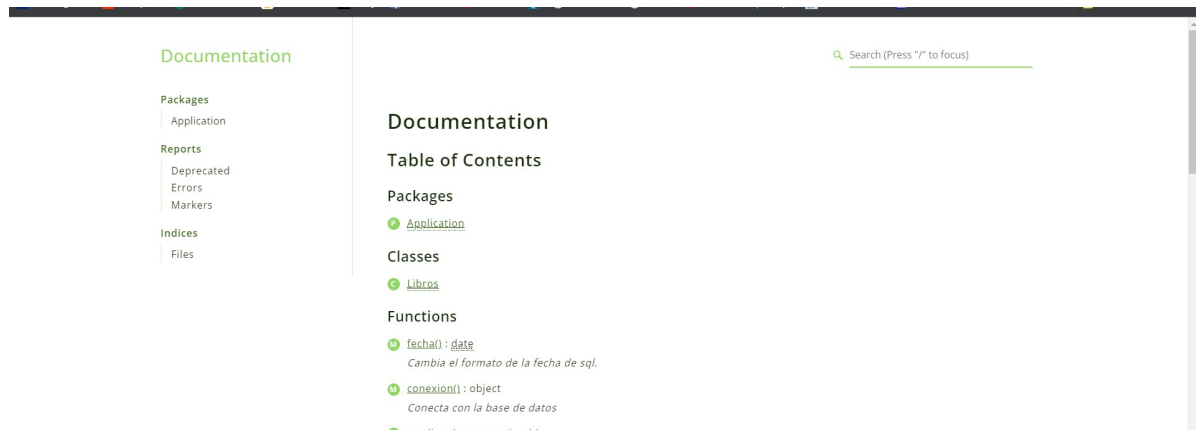
## RA7\_h) Se ha documentado el servicio Web.

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.

Documentación

Al pinchar allí:



También se encuentra en la carpeta DOCS del proyecto.

Página: <http://andreaagoitiar.byethost7.com/>