

Aplicaciones Web- Practica 3

Arquitectura del Proyecto



Reverse Bid

Autores:

Iván Canas
David Zamora
Daniel Reyes
Christian Suarez
David Labrador

Índice:

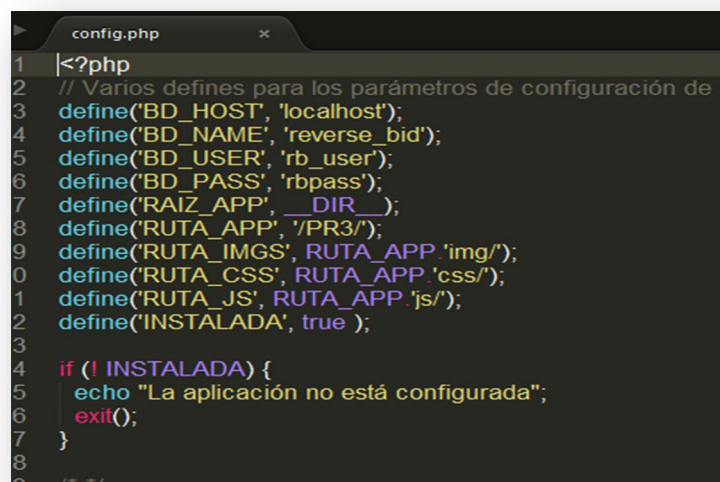
<i>Listado de scripts para las vistas:</i>	<u>3</u>
<i>Listado de scripts adicionales:</i>	<u>8</u>
<i>Estructura de la base de datos</i>	<u>10</u>

Listado de scripts para las vistas:

En este apartado, primero nos gustaría dar una pequeña descripción de la estructura establecida y luego, se van a detallar todos los scripts de vistas que se van a implementar para dar soporte a la interacción de la web, en cada uno se explicará brevemente para qué sirve.

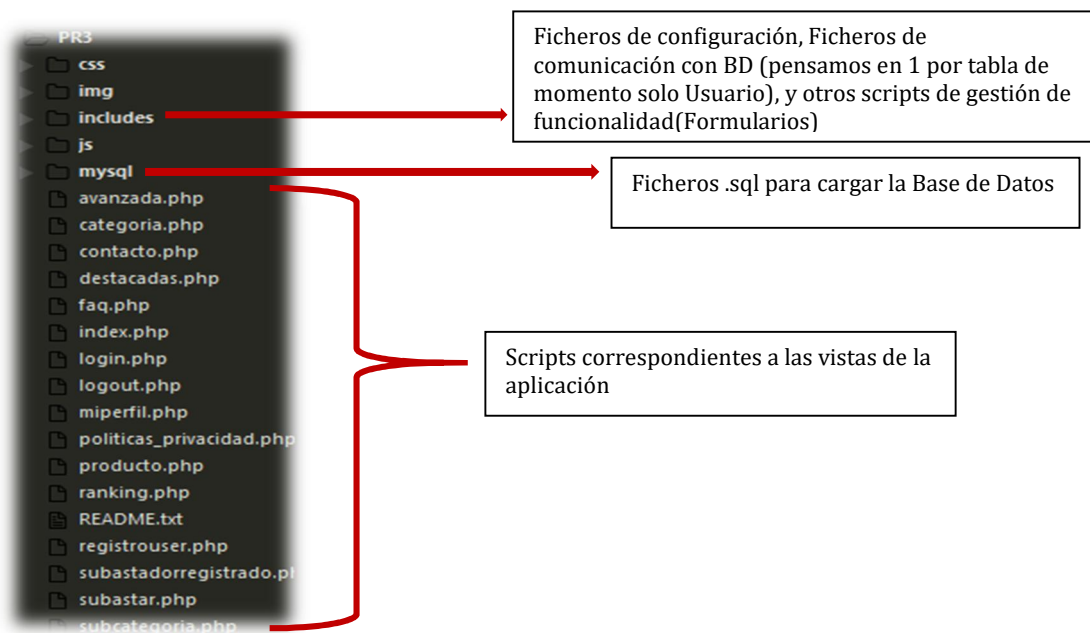
- **Estructura:** Hemos tomado como base, la estructura publicada por Iván (profesor sustituto de Laboratorio), a partir de la cual desarrollamos el **login y la funcionalidad pedida**, en nuestro caso **Registrar Usuario (cliente)** en el botón “**Regístrate**”, todo esto conlleva las siguientes implicaciones a tener en cuenta:

- Existe un fichero “config.php” que establece datos primordiales, y que se deben tener en cuenta a la hora de ejecutar la aplicación.



```
1 <?php
2 // Varios defines para los parámetros de configuración de a
3 define('BD_HOST', 'localhost');
4 define('BD_NAME', 'reverse_bid');
5 define('BD_USER', 'rb_user');
6 define('BD_PASS', 'rbpass');
7 define('RAIZ_APP', __DIR__);
8 define('RUTA_APP', '/PR3/');
9 define('RUTA_IMGS', RUTA_APP.'img/');
10 define('RUTA_CSS', RUTA_APP.'css/');
11 define('RUTA_JS', RUTA_APP.'js/');
12 define('INSTALADA', true );
13
14 if (! INSTALADA) {
15     echo "La aplicación no está configurada";
16     exit();
17 }
18
19 // ...
```

- La Estructura de ficheros es la siguiente:



- Estamos pensando también, en la posibilidad de pasar a usar un framework, para el final, con lo cual esta estructura no sería definitiva.

- **Scripts de Vistas:** en cuanto a vistas no hay mayores novedades a las ya definidas en anteriores entregas, aumentamos una de despedida (**logout.php**), para mostrar un mensaje cuando se cierra sesión; a continuación, una breve descripción de ellos:

- **Index.php:** Contiene la vista principal de nuestra aplicación. Nada más cargarse esta página, carga el config.php que contiene la configuración de acceso a la base de datos y asigna a la variable \$app una instancia mediante el patrón Singleton de Aplicación.php que se encuentra en la carpeta includes.
De esta manera, config.php le da todos los datos a la variable app para controlar la base de datos.

Para cargar los archivos css, se solicita a la variable \$app mediante su método “resuelve” (la resolución de rutas para la carga de estos archivos).

Para la carga de la cabecera y el pie de página, se solicitará a la variable \$app mediante el método doInclude() la inclusión de estos dos archivos php, el método doInclude se encarga de la resolución de rutas, y de incluir finalmente el archivo.

Para la carga del login (tanto en index.php como en login.php) se creará una variable llamada \$html a la que se le asigna el método mostrarSaludo() perteneciente a Util.php. Si no existe ningún usuario logueado, se verá el cuadro para loguearse. De lo contrario, se mostrará un saludo y un botón de cerrar sesión.

(se enlaza desde → Cabecera → Título)
(se enlaza desde → barra de Menú → Logo)
(se enlaza desde → barra de Menú → Home)

- **Login.php:** Contiene la vista de login de la aplicación, se inserta código php en la etiqueta < contenedor id= “login”>. Este código comprueba con la variable \$html si existe un usuario con sesión abierta, mostrando un saludo con botón de logOut o bien un formulario para login en caso de que no se haya iniciado sesión.
A través de esta vista, y de la clase \$formLogin se gestionarán los datos introducidos al formulario de login.

(se enlaza desde → barra de menú “Mi Cuenta”)
(se enlaza desde → después de realizar un registro de usuario correctamente)

- **RegistroUser.php (Funcionalidad escogida):** Contiene la vista para registrar un usuario. En el <contenedor> de registro, se creará una instancia del **FormularioRegistro.php** que contendrá todo lo necesario para crear el formulario de registro en HTML, las funciones para parsear correctamente los datos y comprobar que sean correctos, así como insertar en caso de que todos los datos sean correctos, una nueva entrada en la base de datos con el usuario recogido por el formulario.
Tras el correcto proceso de entrada y comprobación de datos, se **re direccionará**, a la vista login.php para que el nuevo usuario se pueda realizar el iniciar sesión.

(se enlaza desde → botón “Regístrate” en Página Principal- en cuadro de login)
(se enlaza desde → botón “Regístrate” en página login)

- **Subastar.php:** Esta vista nos muestra un formulario para dar de alta un producto en la base de datos (una subasta). Se seleccionará una categoría, un nombre, una cantidad de puja mínima, una imagen cuya ruta se almacenará en la base de datos, se indicará si el producto es usado o nuevo y se mostrará un mensaje describiendo el artículo.

(se enlaza desde → barra Menú → Subastas → Subastar)

- **subastadorregistrado.php:** muestra un formulario para dar de alta a un usuario como subastador, que a diferencia de un usuario (cliente), debe aportar más datos imprescindibles a la hora de vender algo.

(se enlaza desde → barra de Menú → Subastas → Regístrate Subastador)

- **destacadas.php:** muestra productos y breve descripción de los mismos, que destacan en función de la puntuación de los subastadores, por ejemplo.

(se enlaza desde → barra de Menú → Subastas → Subastas Destacadas)

- **categoria.php:** los productos se encuentran clasificados, cada uno pertenece a una subcategoría (p.e portátiles) que a su vez pertenece a una categoría (p.e Tecnología). Sobre ésta última (**categoría**), en esta vista, se muestra la información pertinente y enlaces a las subcategorías que contiene.

(se enlaza desde → barra de Menú → Productos → Cualquier Categoría)

- **subcategoria.php:** en esta vista, se muestra una fotografía de un producto representativo, un filtro a la izquierda, para clasificar los productos que se muestran a la derecha en forma de vista rápida (foto, descripción breve y enlace).

(se enlaza desde → barra de Menú → Productos → Cualquier Subcategoría)

(se enlaza desde → barra de Menú → Productos → Categoría → Subcategorías)

- **Producto.php:** Nos muestra información de un producto en subasta, una imagen del producto en cuestión, una descripción, el ganador si esta ha finalizado y el tiempo restante. También nos permite, si estamos registrados, realizar una puja indicando el importe.

En la parte inferior, podemos ver información del subastador, los dos últimos comentarios con la posibilidad de ver más comentarios, y escribir un comentario acerca del producto.

En la parte derecha tendremos un pequeño chat, con el comunicarnos con el subastador si está conectado.

(se enlaza desde → barra de Menú → Búsqueda rápida)

(se enlaza desde → barra de Menú → Búsqueda Avanzada → Búsqueda)

(se enlaza desde → barra de Menú → Subastas → Subastas Destacadas → producto)

(se enlaza desde → barra de Menú → Mi Cuenta → Mi Perfil)

(se enlaza desde → barra de Menú → Productos → Subcategoría → link producto)

- **Miperfil.php:** La vista nos mostrará datos del usuario registrado en nuestra base de datos, su nombre, su apellido, cuantas subastas tiene activas en este momento, y qué pujas, están activas.

Muestra en el contenedor “mispujas”, productos en los que se ha realizado una puja, una imagen de este producto, y datos del mismo, como el estado de la subasta, la puja ganadora, la última puja del usuario registrado, el tiempo que queda, y si esta ha sido ganada.

En otro contenedor “missubastas” nos muestra características de productos que el usuario ha puesto a subastar, así como el tiempo restante, el estado de la misma, la cantidad que va ganando, y el total acumulado en Euros que se va a llevar cuando finalice la misma el usuario registrado.

Esta vista sólo es **accesible** (control ya funcional), si el usuario ha iniciado sesión previamente de lo contrario nos lleva a login.

(se enlaza desde → barra de Menú → Mi Cuenta → Mi Perfil)

(se enlaza desde → barra de Menú → Mi Cuenta → Mis Subastas)

(se enlaza desde → barra de Menú → Mi Cuenta → Mis Pujas)

(se enlaza desde → botón Login, si se inicia sesión con usuario válido)

- **Avanzada.php:** Muestra un formulario para realizar peticiones de búsqueda especializada a la base de datos, pudiendo seleccionar entre categorías y subcategorías.

(se enlaza desde → barra de Menú → Búsqueda Avanzada)

- **Contacto.php:** Muestra un formulario con todos los campos que un usuario nos facilite para ponernos en contacto con él más adelante, así como información de contacto y botones de accesibilidad a redes sociales.

(se enlaza desde → barra de Menú → Ayuda → Contacto)

(se enlaza desde → footer → Contacto)

(se enlaza desde → footer → Asistencia Técnica)

(se enlaza desde → footer → Ventas y reembolsos)

- **faq.php:** Muestra preguntas y respuestas frecuentes.

(se enlaza desde → barra de Menú → Ayuda → FAQ)

- **políticas_privacidad.php:** Que mostrará las políticas de privacidad y condiciones de uso de ReverseBid.

(se enlaza desde → footer → Condiciones de Uso)

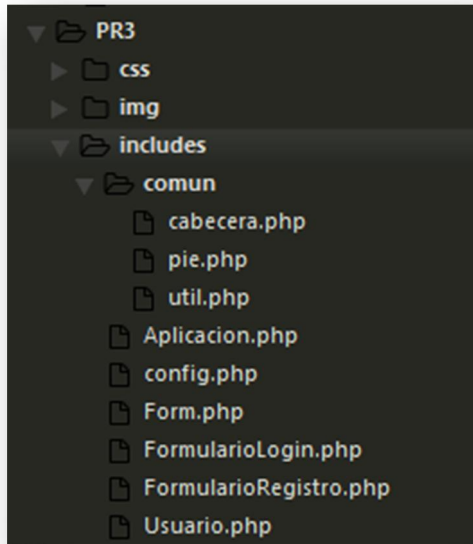
(se enlaza desde → footer → Políticas de Privacidad y Cookies)

- **ranking.php:** muestra las valoraciones de un subastador y los comentarios que ha recibido

(se enlaza desde → producto.php → Sección comentarios → Leer Más)

Listado de scripts adicionales:

Aquí, se van a detallar todos los scripts adicionales (que no se corresponden con las vistas de la aplicación) que incluirá el proyecto.



Todos ellos, se encuentran en el directorio, **PR3/includes**, y se clasifican en tres grandes grupos:

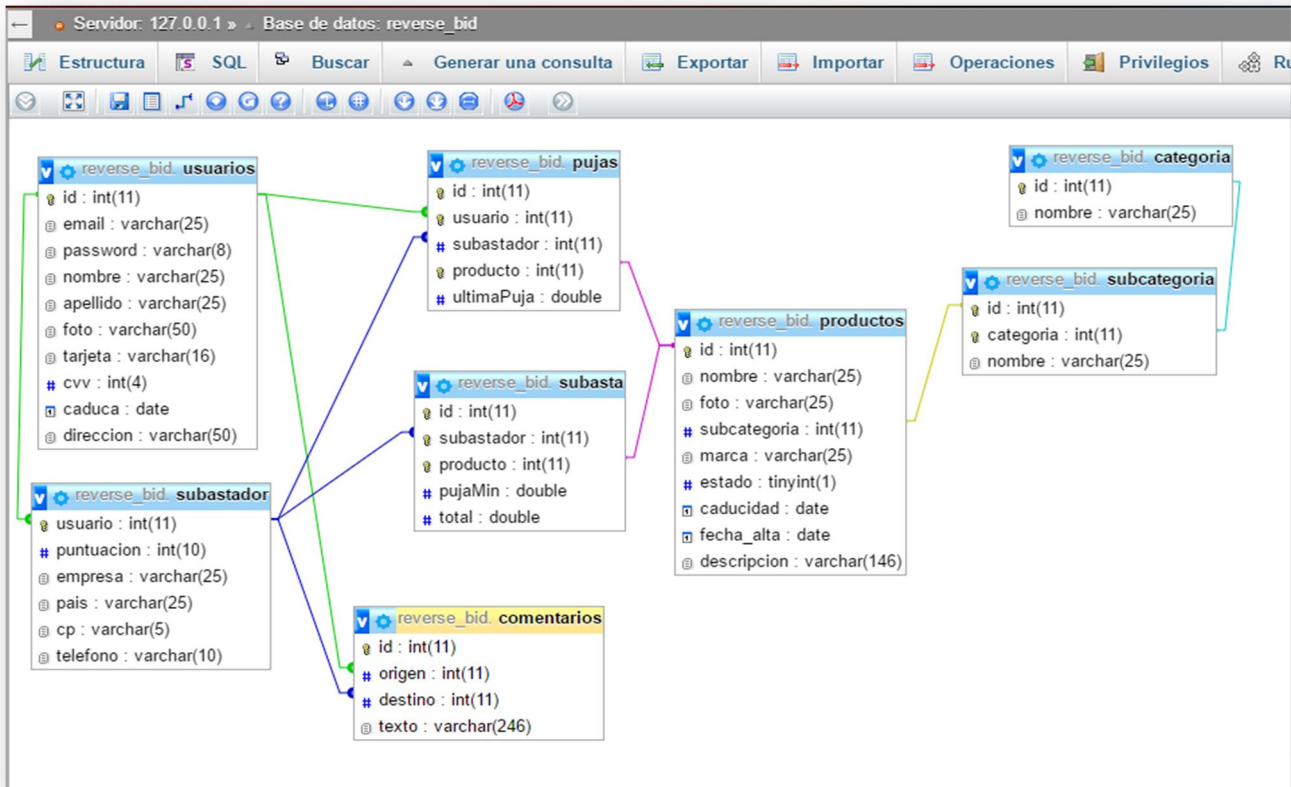
- **Scripts De Acceso a BD:** que proporcionan clases, al estilo java (**Modelo**), que tienen funciones de acceso a la BD y almacenan la información en objetos cuyos atributos corresponden con los campos almacenados en una tabla. De momento contamos con:
 - **Usuario.php:** que accede a la tabla “Usuarios” y de momento cuenta con funciones de consulta (BuscarUsuario) y de Inserción (insertarUsuario), que nos sirven para las funcionalidades implementadas, “login” y “Registro de Nuevo Usuario”.
- **Scripts de Funcionalidad:** que proporcionan y procesan todas las funcionalidades principales de la aplicación, es decir vendrían a ser los **controladores** en un modelo **MVC**, que actúan recibiendo cambios (eventos) de las Vistas, procesan datos y se lo notifican al Modelo para que actúe pertinentemente con la BD. Actualmente contamos con:
 - **Aplicación.php:** que instancia nuestra aplicación usando el patrón **singleton**; inicializa conexión a BD, ruta raíz, y directorio de instalación; lleva el control sobre la gestión de sesiones y resolución de rutas de directorio.
 - **config.php:** que define los datos necesarios para la inicialización de la aplicación (datos de acceso a BD, rutas) e inicializa la aplicación llamando al método “init(..)” de “Aplicación.php”.
 - **Form.php:** una de las clases más interesantes que nos encontramos en el esqueleto proporcionado, actúa como Abstracta; proporcionando funciones internas que generan cualquier formulario que establezcamos en las clases hijas, resguardan datos enviados por cualquier formulario en las variables de Sesión, procesa dichos datos usando las funciones(abstratas) que definamos en las clases hijas de manera que automatiza el proceso. Es decir, si queremos generar un formulario y su respuesta, solo debemos definir una clase(controlador) que extienda a Form, determinar sus campos, determinar los controles sobre dichos campos, determinar las funciones de acceso a base de datos necesarias para generar los resultados y pasar dichos resultados al padre (Form).

- **FormularioLogin.php:** extiende la clase Form, e implementa sus funciones abstractas para generar los campos de formulario de login, procesar los campos de dicho formulario obteniendo los resultados (consulta y comparación de usuario de la BD llamando al **Modelo Usuario**) y devolviéndolos.
- **FormularioRegistro.php:** extiende la clase Form, e implementa sus funciones abstractas para generar los campos de formulario de registro, procesar los campos de dicho formulario obteniendo los resultados (Inserción correcta de usuario en la BD llamando al **Modelo Usuario**) y devolviéndolos.
- **Scripts Útiles y Comunes a las Vistas:** ubicados en el directorio “PR3/icludes/comun”, son scripts que se incluirán en varios sitios en las vistas y que por lo general proporcionan código html necesario, bien porque se incluye y es común o bien porque es necesario diferenciar casos en que se genere un html u otro. Actualmente contamos con:
 - **cabecera.php y pie.php:** que como su nombre indica proporcionan el html, de cabecera y pie común a todas las vistas.
 - **util.php:** que actualmente proporciona funciones necesarias para diferenciar el código html que se muestra en index.php y login.php en función de si el usuario ha iniciado sesión o no (“**mostrarSaludo(..)**”); y diferenciar el permiso de acceso o nó a miperfil.php en función de si el usuario ha iniciado sesión o nó (“**mostrarPerfil(...)**”).

Estructura de la base de datos

En este apartado se va a detallar la estructura de la Base de Datos de la aplicación

- Hemos diseñado una primera versión de la base de datos que usaremos, en este contexto incluimos un diagrama relacional, auto explicativo, indicando las tablas, campos y relaciones:



➤ Notas Explicativas:

- Hemos añadido un **Identificador (Id)** a cada tabla como clave, para simplificar uso de claves compuestas
- Existe un **Usuario Registrado** en General **Tabla Usuarios**, a la única que de momento accede el script "**Usuario.php**" que consulta y modifica la tabla en función de si queremos iniciar sesión o registrar usuario.
- **Un Subastador** es un Usuario Registrado con más características
- **Un Comentario** relaciona un Subastador con un Usuario Normal (cliente)
- **Una Puja** relaciona un Cliente, un Subastador y un Producto

- **Una Subasta** relaciona un Subastador y un Producto
- **Un Producto** pertenece a una Subcategoría y ésta pertenece a una Categoría
- En el Directorio “**PR3/MySql**” incluimos los ficheros .sql para creación de Base de datos y su Estructura, creación del usuario usado en “config.php”, y algunos datos de prueba en tres ficheros distintos.