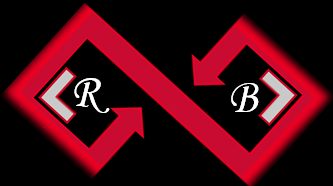
***Aplicaciones Web- Practica 3***

***Arquitectura del Proyecto***

****

***Reverse Bid***

**Autores:**

**Iván Canas**

**David Zamora**

**Daniel Reyes**

**Christian Suarez**

**David Labrador**

# Índice:

[Listado de scripts para las vistas: 3](#_Toc450065186)

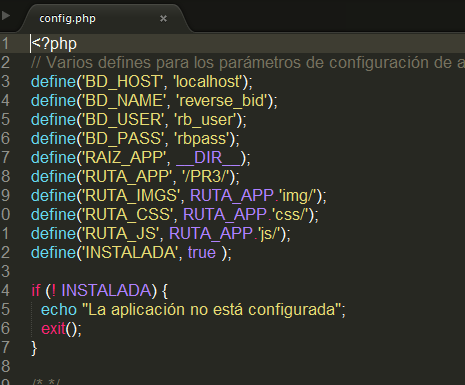
[Listado de scripts adicionales: 4](#_Toc450065187)

[Estructura de la base de datos 5](#_Toc450065188)

# Listado de scripts para las vistas:

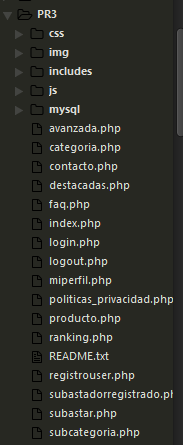
En este apartado, primero nos gustaría dar una pequeña descripción de la estructura establecida y luego, se van a detallar todos los scripts de vistas que se van a implementar para dar soporte a la interacción de la web, en cada uno se explicará brevemente para qué sirve.

* **Estructura:** Hemos tomado como base, la estructura publicada por Iván, 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.

****

* La Estructura de ficheros es la siguiente:

Ficheros de configuración, Ficheros de comunicación con BD (pensamos en 1 por tabla de momento solo Usuario), y otros scripts de gestión de funcionalidad(Formularios)

****

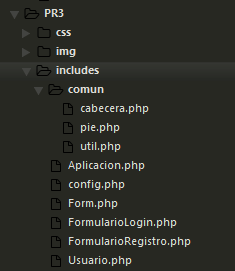
Scripts correspondientes a las vistas de la aplicación

Ficheros .sql para cargar la Base de Datos

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

# 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”.**

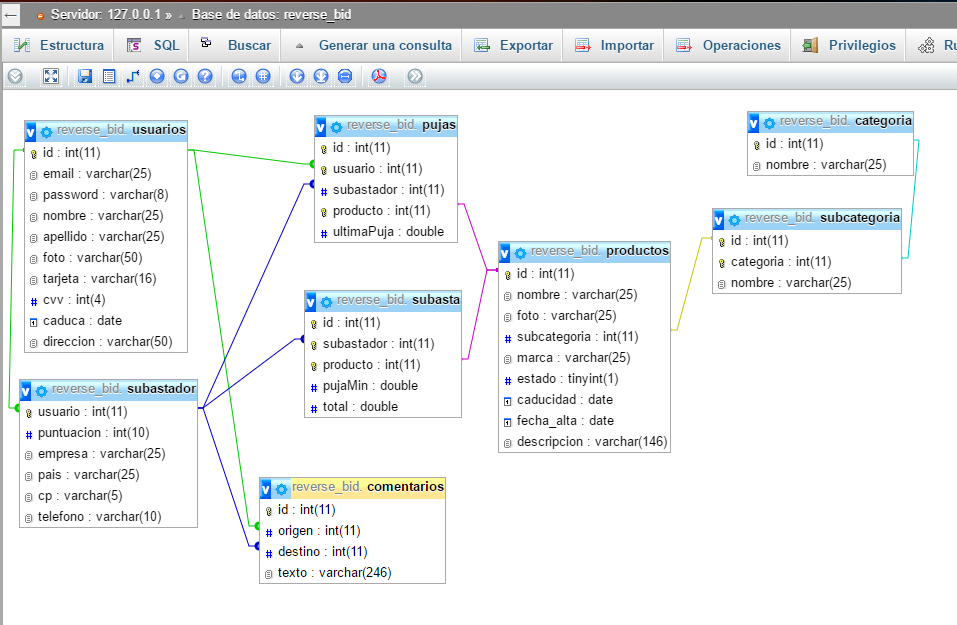
* **Scripts de Funcionalidad:** que proporcionan y procesan todas las funcionalidades principales de la aplicación, es decir vendrían a ser los **controladores** en un modelo **MVC,** que actúan recibiendo cambios (eventos) de las Vistas, procesan datos y se lo notifican al Modelo para que actué pertinentemente 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(abstractas) 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 Utiles 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 guncion 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
* **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.