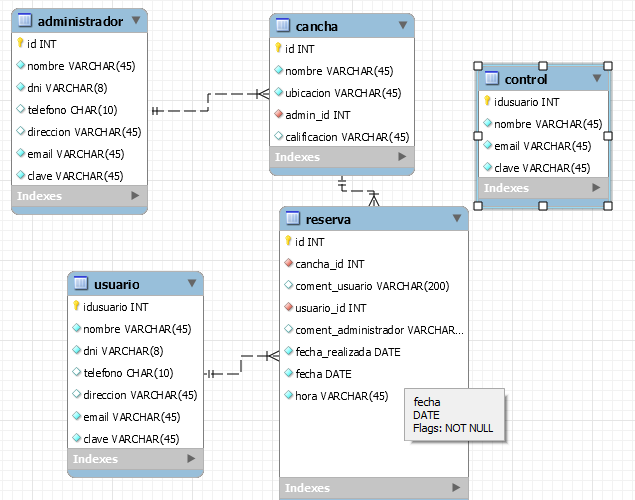
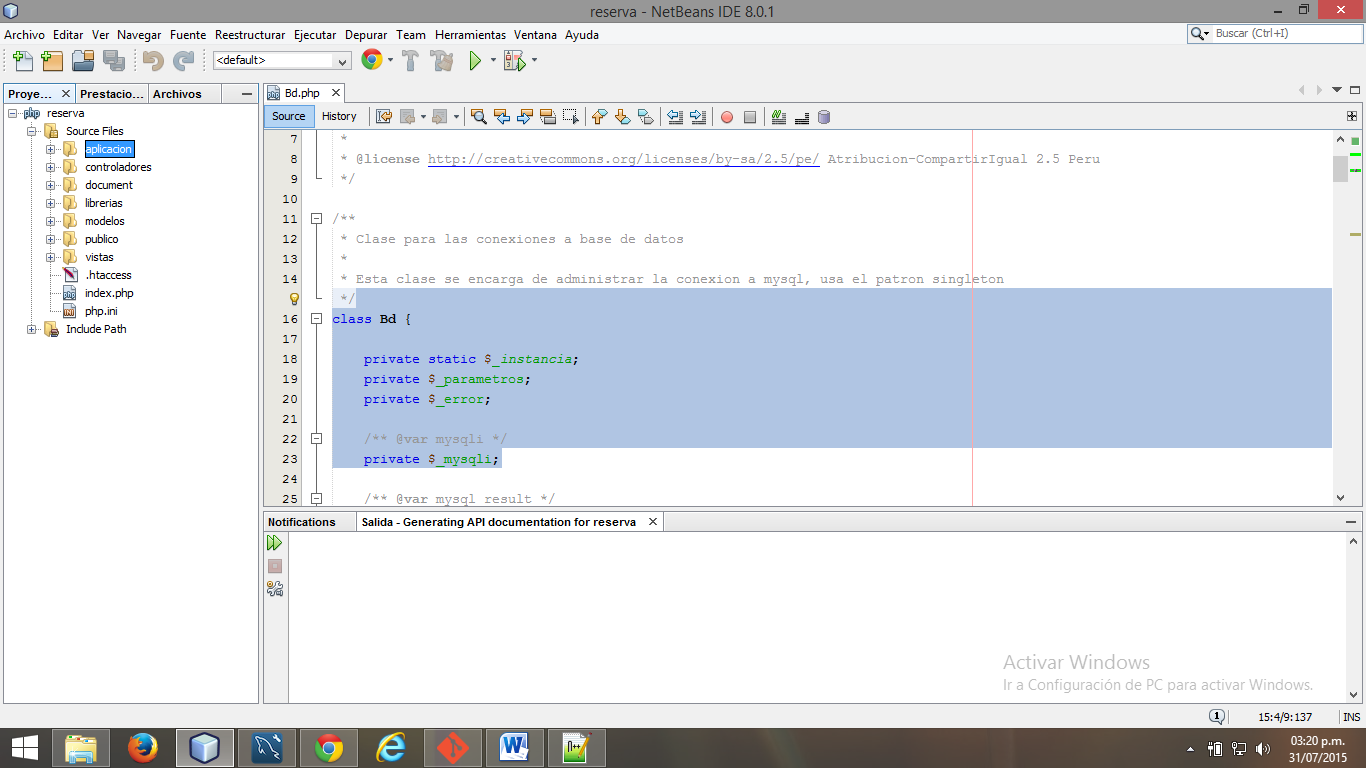
**Manual de programador**

**La base de dato**

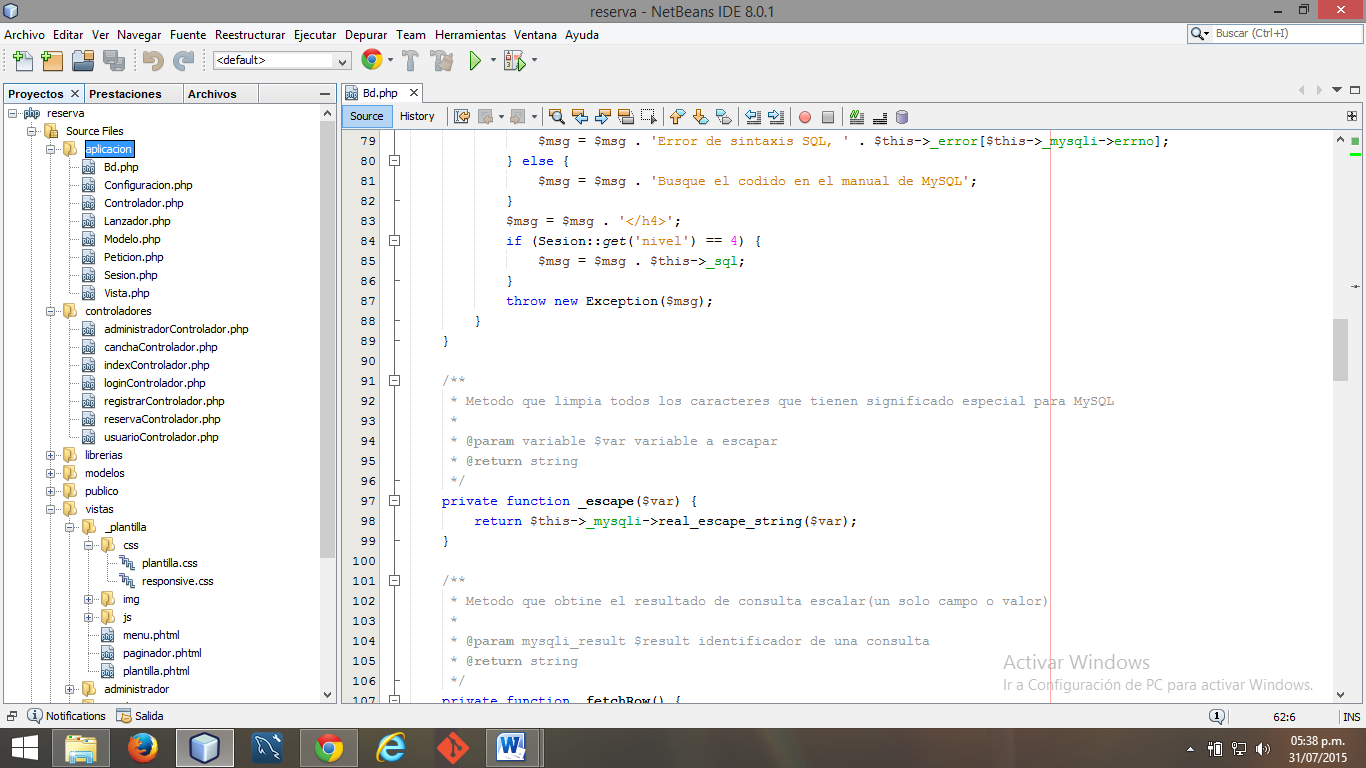
**La aplicación:**

La raíz dela aplicación es la que lleva como nombre “reserva”.



Las carpetas aplicación, librerías, publico son carpetas que contiene documento php que son de configuración y que le darán soporte a la aplicación que se desea trabajar. Estos documentos fueron copiados de <http://creativecommons.org/licenses/by-sa/2.5/pe/>.

**Carpeta aplicación**



**Funciones de Bd.php**

|  |  |
| --- | --- |
| getScalar($sql, $parametros) | Función que retorna un valor escalar de la consulta. Donde sql es la consulta y parámetro es una array. |
| getFila($sql, $parametros) | Función que retorna una fila del array asociativo del resultado de la consulta, donde sql es la consulta y parámetros un array. |
| getArray($sql, $parametros) | Función que retorna un array asociativo del resultado de la consulta, donde sql es la consulta y parámetros es una array. |

**Funciones de Controlador.php**

|  |  |
| --- | --- |
| public function redireccionar($ruta) | Método que Re direcciona a una url |
| protected function aInt($int) | Método que convierte a enteros |
| protected function getTex($clave) | Método que valida un texto enviado por post |
| protected function getAlpNum($clave, $min, $max) | Método que valida un campo alfanumérico enviado por post |
| protected function getAltTex($clave) | Método que valida un campo de texto alternativo enviado por post |
| protected function getEmail($clave) | Método que valida un campo email enviado por post |
| protected function getPas($clave) | Método que valida un campo password enviado por post |
| protected function getNum($clave, $tam) | Método que valida un valor numérico enviado por post |
| protected function getAltNum($clave) | Método que valida un valor numérico alternativo enviado por post |
| protected function getPost() | Método que retorna solo valores de la variable post |
| protected function setPost($clave, $valor) | Método que agrega un valor al final de la variable post |
| protected function getHora($clave) | Método que valida una hora en formato 24 horas |

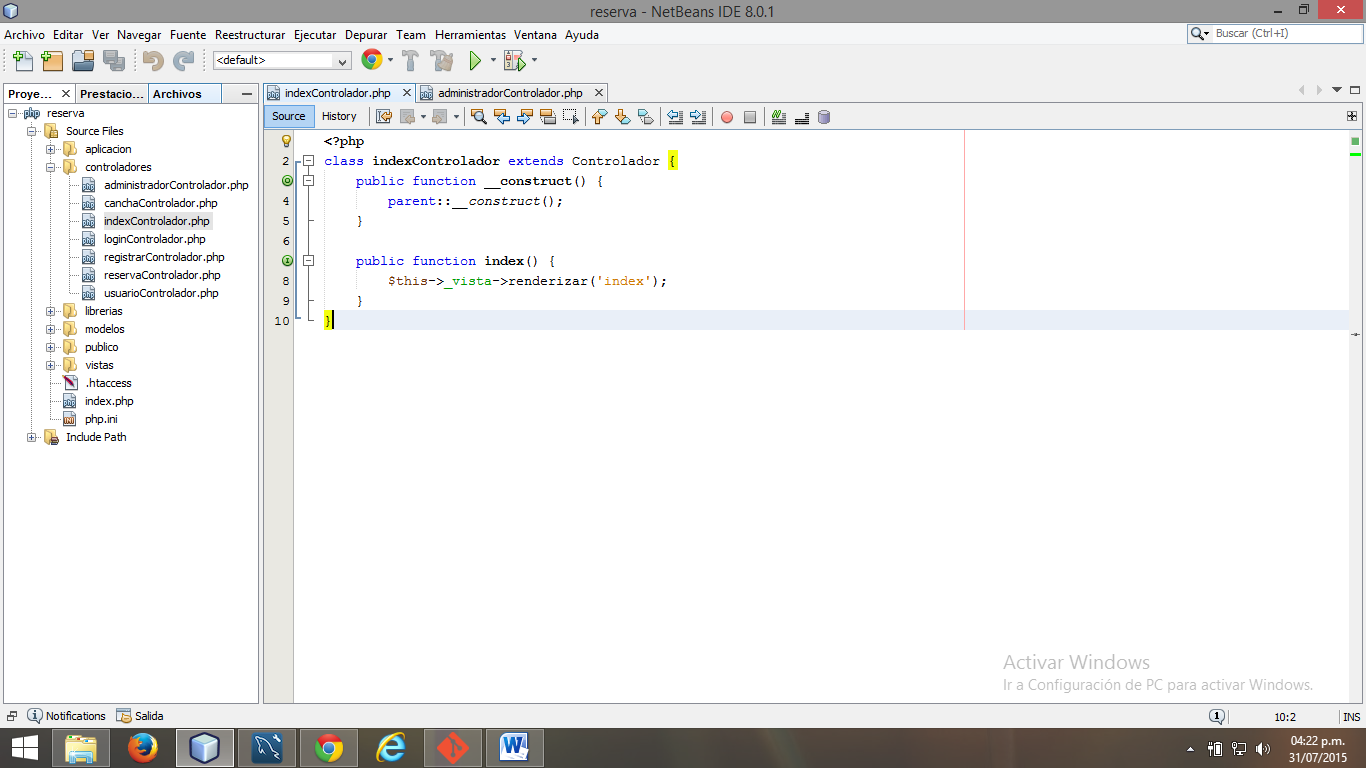
Como se está trabajando con la arquitectura MVC tenemos las carpetas controlador, modelo y vista. Tambine se cuenta con el archivo índex.php que es el archivo principal.

**índex.php**

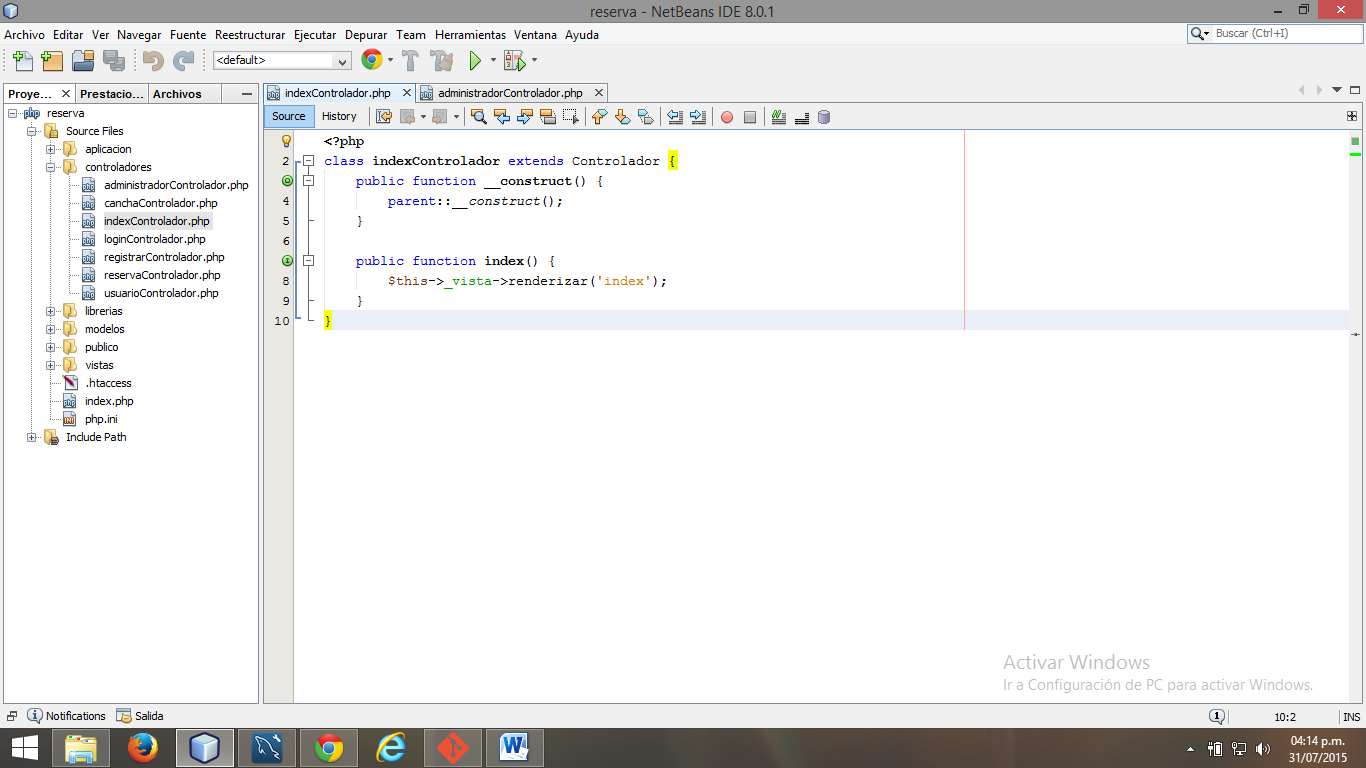
|  |  |
| --- | --- |
| define('SD', DIRECTORY\_SEPARATOR) | Define la constante que almacena el separador. |
| define('RAIZ',realpath(dirname(\_\_FILE\_\_)) . SD) | Constante que almacena la ruta raíz de la aplicación. |
| require\_once RAIZ . 'aplicacion' . SD . | Hace un requerimiento de la carpeta aplicación. |

**Carpeta controladores**

Los controladores almacenan todas las funciones y direccionamientos en este caso se ha realizado un controlador para cada entidad de la base de datos. Todos los archivos que contienen controladores son extensiones del archivo Controlador de la carpeta aplicación.



**indexControlador.php**



Se tiene la clase indexControlador que es una extensión controlador, la cual contiene una función pública index sin para metros que se encarga direccionamiento para que se muestre el archivo index de la carpeta vista.

**Funciones públicas carpeta controlador**

|  |  |
| --- | --- |
| function nuevo() | Esta función valida y redirección los datos al modelo para que sean almacenados en la bd, como también direcciona para mostrar la vista que se muestra al usuario. |
| function listar() | Función que envía los para metros que debe cumplir para listar las canchas. |
| Function calificar($id) | Función que califica valida los datos de la calificación de una cancha. |
| function listarCancha() | Función que lista todas las canchas del administrador que inicio sesión. |
| function editar($id) | Función que edita los datos correspondientes al parámetro de entra. |
| function eliminar($id) | Función que elimina los datos correspondientes al parámetro de entrada. |
| function salir() | Función que cierra la sesión. |
| function nueva($nuevo) | Función que permite inserta los datos almacenados en el parámetro de entrada. |
| function listarMio() | Función que permite listar todas las reservas del usuario que inicio sesión. |
| ver\_horas($cancha, $fecha) | Función que muestra las horas disponibles de acuerdo a la elección de la cancha y fecha. |

**Funciones públicas carpeta modelo**

|  |  |
| --- | --- |
| function insertar($datos) | Función que hace la consulta SQL para insertar datos. |
| function existeDni($dni) | Función que hace la consulta SQL, para retornar el DNI que es igual al parámetro de entrada. |
| function existeEmail($email) | Función que hace la consulta SQL, para retornar el email que es igual al parámetro de entrada. |
| function contar() | Función que hace la consulta SQL, y devuelve la cantidad de filas que cumple la condición. |
| function get($id) | Función que hace la consulta SQL, que devuelve el id que sea igual al parámetro de entrada. |
| function listar() | Función que hace la consulta SQL para listar los datos correspondientes. |
| function listarCancha() | Función que hace la consulta SQL para listar las canchas correspondientes al usuario que inicio sesión. |
| existeUbicacion ($ubicacion) | Función que hace la consulta SQL para devolver la ubicación que sea igual al parámetro de entrada. |
| function editar($datos) | Función que hace la consulta SQL para editar los datos. |
| function existeCancha($id) |  |
| function existeCalificacion($id) | Función que valida la existencia de la cancha. |