

2º Desarrollo de Aplicaciones Web
Curso 2014/2015

Desarrollo Web en Entorno Servidor

Práctica 2
Web venta de inmuebles

Salvador Camacho Soto

1.- Enlaces web.

Enlace a la página web: <http://salvacam.x10.mx/>

Enlace a la práctica: <http://salvacam.x10.mx/practica2/index.php>

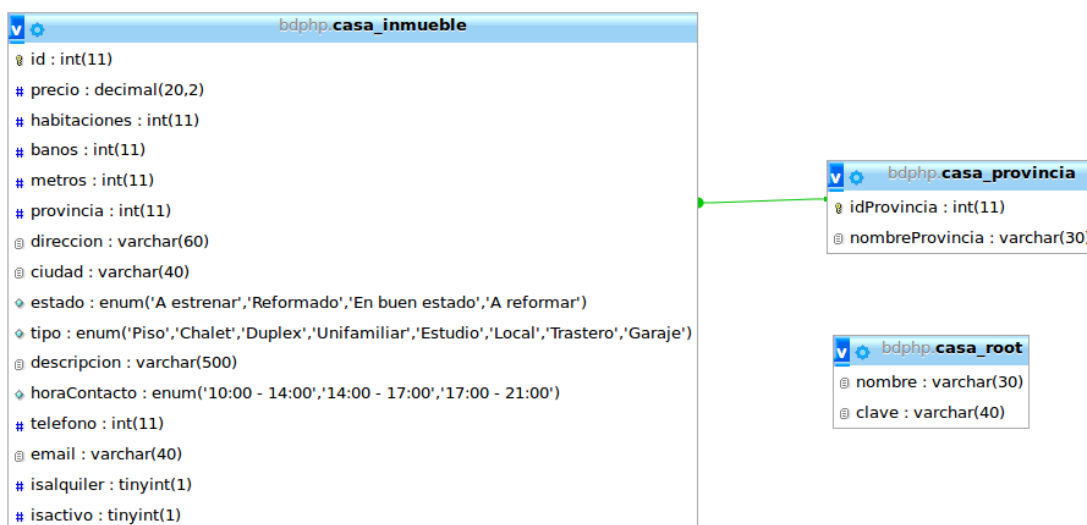
Enlace a GitHub: <https://github.com/salvacam/Practica-2>

2.- Requerimientos.

Tenemos que crear una web de venta de inmuebles, que tenga un front-end desde donde se puedan ver las ventas y un back-end desde donde se pueda insertar nuevas ventas con imágenes.

3.- Implementación.

En la base de datos usare tres tablas.



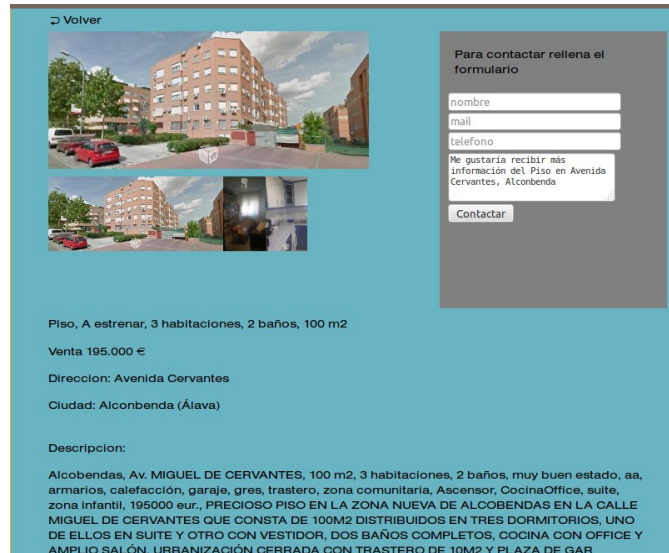
La tabla *casa_inmueble* con los datos del inmueble, la tabla *casa_provincia* con los nombres de las provincias y su id y la tabla *casa_root*, que solo lo uso para tener una tabla con el nombre y clave del root. La columna *provincia* de la tabla *casa_inmueble* es una clave foránea de la tabla *casa_provincia*.

Para el *front-end* tengo una página web, con las entradas paginadas de 5 en 5.



En la parte superior tengo el *buscador* en el que puedo usar varias condiciones, tengo un enlace para *administrar* y otro para *publicar anuncio* cualquier usuario.

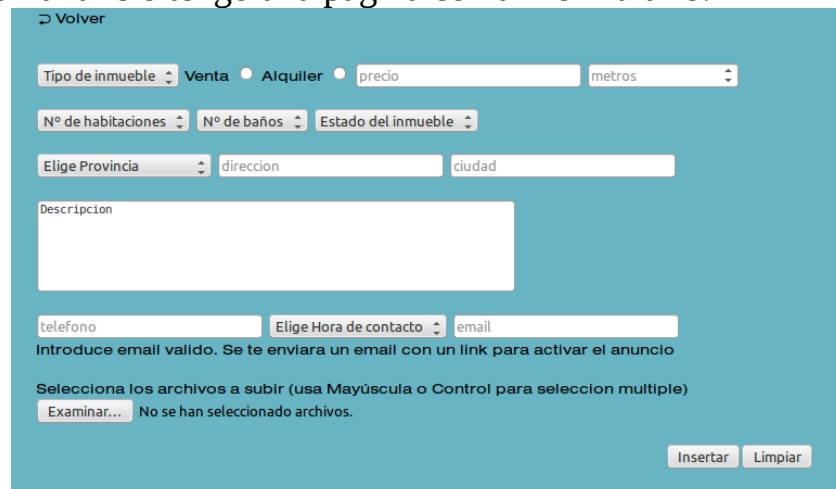
Cada entrada es un enlace para ver cada anuncio.



The screenshot shows a real estate listing for a flat. At the top left, there is a link "Volver". Below it are three images: a large photo of the building, a smaller photo of the building, and a photo of the interior. To the right of the images is a contact form titled "Para contactar rellena el formulario" with fields for "nombre", "mail", and "telefono", and a "Contactar" button. Below the images, the text reads: "Piso, A estrenar, 3 habitaciones, 2 baños, 100 m2", "Venta 195.000 €", "Direccion: Avenida Cervantes", "Ciudad: Alcobenda (Alava)", and "Descripcion: Alcobendas, Av. MIGUEL DE CERVANTES, 100 m2, 3 habitaciones, 2 baños, muy buen estado, aa, armarios, calefacción, garaje, gres, trastero, zona comunitaria, Ascensor, CocinaOffice, suite, zona infantil, 195000 eur., PRECIOSO PISO EN LA ZONA NUEVA DE ALCOBENDAS EN LA CALLE MIGUEL DE CERVANTES QUE CONSTA DE 100M2 DISTRIBUIDOS EN TRES DORMITORIOS, UNO DE ELLOS EN SUITE Y OTRO CON VESTIDOR, DOS BAÑOS COMPLETOS, COCINA CON OFFICE Y AMPLIO SALÓN. URBANIZACIÓN CERRADA CON TRASTERO DE 10M2 Y PLAZA DE GAR".

Cada anuncio muestra todas las fotos, los datos y un formulario para contactar mediante email con el vendedor mediante el email que ingresa al insertar el anuncio.

Para publicar un anuncio tengo una pagina con un formulario.



The screenshot shows a real estate listing form. At the top left, there is a link "Volver". The form has several sections: "Tipo de inmueble" with a dropdown menu showing "Venta" and "Alquiler", and a "precio" field; "Nº de habitaciones" and "Nº de baños" dropdown menus; "Estado del inmueble" dropdown menu; "Elige Provincia" dropdown menu; "direccion" and "ciudad" text fields; "Descripcion" text area; "telefono" text field; "Elige Hora de contacto" dropdown menu; and "email" text field. Below these fields is a message: "Introduce email valido. Se te enviara un email con un link para activar el anuncio". At the bottom, there is a section for file uploads: "Selecciona los archivos a subir (usa Mayúscula o Control para seleccion multiple)" and an "Examinar..." button. Below this is a message: "No se han seleccionado archivos." At the bottom right, there are "Insertar" and "Limpiar" buttons.

Una vez rellenado y validado mediante javascript y php, si se crea la entrada en la base de datos, creo una carpeta y subo las fotos (si hubiera) y a continuación se le envia un email con un enlace para que valide el anuncio, una vez validado recibe otro email con dos enlaces uno para borrar el anuncio (solo se desactiva, no lo borra) y otra para editarlo.

Root User <programacion211@gmail.com>

para mí ☐

inglés > español Traducir mensaje

Ha publicado un anuncio.

Click aquí para: [Activar anuncio](#)

Root User <programacion211@gmail.com>

para mí ☐

...

Ha activado el [anuncio](#)

No borre este correo. Lo necesitara para borrar o editar el anuncio

Click aquí para: [Borrar anuncio](#)

Click aquí para: [Editar anuncio](#)

Si borra o edita el anuncio recibe otro email avisandolo de la acción que ha realizado.

Para acceder al *back-end* tengo una página de login, en el que tengo que introducir nombre (root) y clave (toor), si no se introduce correctamente no se puede acceder a ninguna parte del *back-end*. Al acceder al *back-end* tengo una tabla con las entradas, paginadas de 10 en 10, un enlace para salir del *back-end* y un enlace para *añadir anuncio*.

➔ Salir de Adminsitrar

Id	Tipo	Estado	V / A	Precio	Hab.	Baños	Provincia	Ciudad	Direccion	Email	Tlf	Hora	Activo	Borrar	Editar
1	Piso	A estrenar	Venta	195000.00	3	2	Madrid	Alconbenda	Avenida Cervantes	salvacams@gmail.com	0		1 Desact		
2	Duplex	A estrenar	Venta	180000.00	3	2	Barcelona	Barcelona	Esquirol	salvacams@gmail.com	956235689	14:00 - 17:00	1 Desact		
3	Chalet	Reformado	Venta	140000.00	3	2	Granada	Las Gabias	Severo Ochoa	salvacams@gmail.com	0		1 Desact		
4	Piso	A estrenar	Venta	195000.00	3	2	Madrid	Alconbenda	Avenida Cervantes	salvacams@gmail.com	0		1 Desact		
5	Duplex	A estrenar	Venta	180000.00	3	2	Barcelona	Barcelona	Esquirol	salvacams@gmail.com	956235689	14:00 - 17:00	1 Desact		
6	Chalet	Reformado	Venta	140000.00	3	2	Granada	Las Gabias	Severo Ochoa	salvacams@gmail.com	0		1 Desact		
7	Piso	A estrenar	Venta	195000.00	3	2	Madrid	Alconbenda	Avenida Cervantes	salvacams@gmail.com	0		1 Desact		
8	Duplex	A estrenar	Venta	180000.00	3	2	Barcelona	Barcelona	Esquirol	salvacams@gmail.com	956235689	14:00 - 17:00	1 Desact		
9	Chalet	Reformado	Venta	140000.00	3	2	Granada	Las Gabias	Severo Ochoa	salvacams@gmail.com	0		1 Desact		
10	Piso	A estrenar	Venta	195000.00	3	2	Madrid	Alconbenda	Avenida Cervantes	salvacams@gmail.com	0		1 Desact		

< < 1 2 3 4 5 > >

Añadir anuncio

Cada entrada tiene un enlace para *borrar*, para *activar o desactivar* y otro para *editar*. Si añado un anuncio accedo a un formulario igual que se publicara un anuncio un usuario solo que ademas tiene un checkbox para crearlo activo si se quiere.

[Volver](#)

Tipo de inmueble Venta ☒ Alquiler ☐ precio metros

Nº de habitaciones Nº de baños Estado del inmueble

Elige Provincia direccion ciudad

Descripcion

telefono Elige Hora de contacto email

Selecciona los archivos a subir (usa Mayúscula o Control para seleccion multiple)
 No se han seleccionado archivos.

Activo ☐

Si edito algun anuncio tengo el mismo formulario, con los valores que tiene el anuncio y las fotos que tiene con un enlace cada una para poder eliminarlas.

[Volver](#)

Piso Venta ☒ Alquiler ☐ 195000.00 100

3 2 A estrenar

Madrid Avenida Cervantes Alcobenda

Alcobendas, Av. MIGUEL DE CERVANTES, 100 m2, 3 habitaciones, 2 baños, muy buen estado, aa, armarios, calefacción, garaje, gres, trastero, zona comunitaria, Ascensor, CocinaOffice, suite, zona infantil, 195000 eur., PRECIOSO PISO EN LA ZONA NUEVA DE ALCOBENDAS EN LA CALLE MIGUEL DE CERVANTES QUE CONSTA DE 100M2 DISTRIBUIDOS EN TRES DORMITORIOS, UNO DE ELLOS EN SUITE Y OTRO CON VESTIDOR, DOS BAÑOS

telefono Elige Hora de contacto salvacams@gmail.com

Selecciona los archivos a subir (usa Mayúscula o Control para seleccion multiple)
 No se han seleccionado archivos.

Activo ☐

 [Borrar](#)  [Borrar](#)  [Borrar](#)

4.- Código

Para cada tabla uso dos clases, una con el nombre de la tabla y otra cuyo nombre es ModeloNombre de la tabla, con lo tenga 6 clases para las tablas de la base de datos:

- Inmueble
- ModeloInmueble
- Provincia
- ModeloProvincia
- Root
- ModeloRoot

Estas clases tiene el mismo contenido que las realizadas en clases, incluso las clases *ModeloRoot* y *ModeloProvincia* tienen menos métodos, ya que solo he implementado los necesarios.

La clase **ModeloInmueble** si tiene más métodos implementados. Tengo el método *selectHTMLColumna*, para que me devuelva un select de la columna que le indique

```
function selectHTMLColumna($columna, $id, $name, $blanco = TRUE, $textoBlanco = "&nbsp;") {  
    $select = "<select name='$name' id='$id'>";  
    if ($blanco) {  
        $select .= "<option value=''>$textoBlanco</option>";  
    }  
    $lista = $this->getListColumna($columna);  
    foreach ($lista as $valor) {  
        if ($valor) {  
            $select .= "<option value='' . $valor[0] . ''>"  
                . $valor[0] . "</option>";  
        }  
    }  
    $select .= "</select>";  
    return $select;  
}
```

En este método tengo una llamada el método *getListColumna*

```
function getListColumna($columna) {  
    $list = array(); //$list = [];  
    $sql = "select distinct $columna from $this->tabla";  
    $r = $this->bd->setConsulta($sql, array());  
    if ($r) {  
        while ($list[] = $this->bd->getFila()) {  
            $list[] = $this->bd->getFila();  
        }  
    } else {  
        return null;  
    }  
    return $list;  
}
```

En el que me devuelve solo una vez elementos de la columna que existe en alguna entrada. Lo uso para que en el *buscador* solo me muestre como opciones de *Ciudad* y *Tipo* de inmueble los que existan al menos una vez.

Para que me muestre el select *Provincia* con el nombre de las provincias y no su id uso el método selectHTMLColumnExt

```
function selectHTMLColumnExt($tablaExt, $nombreColExt, $colExtComp, $columna, $id, $name, $blanco = TRUE, $textoBlanco = "&nbsp;") {
    $select = "<select name='$name' id='$id'>";
    if ($blanco) {
        $select .= "<option value=''>$textoBlanco</option>";
    }
    $lista = $this->getListColumnExt($tablaExt, $nombreColExt, $colExtComp, $columna);
    foreach ($lista as $valor) {
        if ($valor) {
            $select .= "<option value='" . $valor[0] . "'>"
                . $valor[1] . "</option>";
        }
    }
    $select .= "</select>";
    return $select;
}
```

Que a su vez llama el método getListColumnExt

```
function getListColumnExt($tablaExt, $nombreColExt, $colExtComp, $columna) {
    $list = array(); // $list = [];
    $sql = "select distinct $columna, $nombreColExt "
        . "from $this->tabla, $tablaExt "
        . "where $this->tabla.$columna=$tablaExt.$colExtComp";
    $r = $this->bd->setConsulta($sql, array());
    if ($r) {
        while ($list[] = $this->bd->getFila()) {
            $list[] = $this->bd->getFila();
        }
    } else {
        return null;
    }
    return $list;
}
```

En el archivo ***index.php*** de la raíz de la aplicación filtro los valores que le puedo pasar por post para la búsqueda, el primer parametro que le paso al select es isActivo=1, para que solo muestre los anuncios activos.

```
$parametros["isActivo"] = 1;
$ordenar = "1 DESC";
if ($tipo != "") {
    $condiciones .= " and tipo =:tipo";
    $parametros["tipo"] = $tipo;
}
```

```

}
if ($isalquiler == "0" || $isalquiler == "1") {
    $condiciones .= " and isalquiler =:isalquiler";
    $parametros["isalquiler"] = $isalquiler;
}
if ($provincia != "") {
    $condiciones .= " and provincia =:provincia";
    $parametros["provincia"] = $provincia;
}
if ($ciudad != "") {
    $condiciones .= " and ciudad =:ciudad";
    $parametros["ciudad"] = $ciudad;
}
if ($rangoPrecio != "") {
    $rango = explode(";", $rangoPrecio);
    $condiciones .= " and precio >=:rango0";
    if ($rango[1] != "1000000") {
        $condiciones .= " and precio <=:rango1";
        $parametros["rango1"] = $rango[1];
    }
    $parametros["rango0"] = $rango[0];
}
$filas = $modeloInmueble->getList($pagina, 5, $condiciones, $parametros, $ordenar);

```

Para cada inmueble pongo la primera foto que se ha subido, la busco usando *readdir*

```

$ruta = "./entradas/" . $objeto->getId();
if ($directorio = opendir($ruta)) {
    $imagenPrincipal = "";
    while ($archivo = readdir($directorio)) {
        if ($archivo != ".." && $archivo != ".") {
            $imagenPrincipal = $archivo;
            break;
        }
    }
}
}

```

El valor que obtengo se lo asigno al propiedad src de la imagenes

```
" alt="foto Principal">
```

El archivo **view.php** en la **carpeta Inmueble** muestra cada anuncio individualmente.

Le paso por get la id del anuncio, si no esta activo se redirige la pagina al inicio.

```

$id = Leer::get("id");
$bd = new BaseDatos();
$modelo = new ModeloInmueble($bd);
$modeloProvincia = new ModeloProvincia($bd);
$objeto = $modelo->get($id);
if ($objeto->getIsactivo() == 0) {
    header("Location:../index.php");
    exit();
}

```


Todas las miniaturas de las imagenes tienen la clase imagen, mediante javascript (en el archivo *practica2.js*) cuando hago click en una la pongo como la imagen mayor.

```
var lista = document.getElementsByClassName("imagen");
for (var i = 0; i < lista.length; i++) {
    lista[i].addEventListener("click", function () {
        document.getElementById("imgP").setAttribute("src", this.getAttribute("src"));
    });
}
```

En este mismo archivo solo permito que se escriban números en el input del telefono

```
var telefono = document.getElementById("telefono");
telefono.addEventListener("keypress", filtroEntero, false);

function filtroEntero(e) {
    var codigo = e.charCode || e.keyCode;
    if (codigo < 32 || codigo == 37 || codigo == 38 || codigo == 39 || codigo == 40) {
        return;
    }
    var texto = String.fromCharCode(codigo);

    var permitidos = "0123456789";
    if (permitidos.indexOf(texto) == -1) {
        if (e.preventDefault) {
            e.preventDefault();
        }
        return false;
    }
}
```

En el formulario de contacto requiero mediante HTML5 el nombre y el correo, si el formulario realiza el submit me envia a la pagina *phpCorreo.php*

```
$mensaje = Leer::post("mensaje");
$mensaje .= "<br/> Pongase en contacto por email: ".$mail;
if($telefono!=""){
    $mensaje.= "<br/> o al telefono ".$telefono;
}

$correo = Correo::enviarGmail(NULL, $destino, Configuracion::ASUNTO, $mensaje,
Configuracion::CLAVEGMAIL);
header("Location:view.php?id=$id");
```

Esta pagina envia un correo, usando una cuenta de correo de gmail con los datos que he rellenado y me devuelve a la pagina view.

Para validar mediante javascript el formulario de las páginas *viewAdd.php*, *viewAddUser.php*, *viewUpdate.php* y *viewUpdateUser.php* uso el archivo ***formulario.js***, donde uso la función *enviar* para comprobar si los valores son validos

```
function enviar(evt) {
    var error = "";
    if (tipo.value.length < 1) {
```

```

        error += "Debe seleccionar un tipo de inmueble\n";
        tipo.setAttribute("class", "error");
    } else {        tipo.removeAttribute("class");    }

    .
    .
    .
    var expMail = /^[a-z][\w.-]+@[\w][\w.-]+\.[\w.-]*[a-z][a-z]$/i;
    if (!expMail.test(email.value)) {
        error += "Email debe ser valido\n";
        email.setAttribute("class", "error");
    } else {        email.removeAttribute("class");    }

    if (telefono.value > 0) {
        var expTelefono = /^d{9}$/;
        if (!expTelefono.test(telefono.value)) {
            error += "Si el telefono se introduce debe ser valido (Ej:958123456)\n";
            telefono.setAttribute("class", "error");
        } else {        telefono.removeAttribute("class");    }
        if (horaContacto.value.length < 1) {
            error += "Si el telefono se introduce selecciona una hora de contacto\n";
            horaContacto.setAttribute("class", "error");
        }
        else {        horaContacto.removeAttribute("class");    }
    }
    if (error != "") {
        alert(error);
        evt.preventDefault();
        return false;
    }
    var pregunta = ""
    if (telefono.value <= 0) {
        pregunta += "Telefono vacio\n";
    }
    if (imagenes.value <= 0) {
        pregunta += "No ha seleccionado imagenes\n";
    }
    if (pregunta.length > 0) {
        if (!confirm(pregunta + "¿Enviar formulario?"))
            evt.preventDefault();
    }
}
}

```

Y uso la función *filtroentero* (la he mostrado en la página 9) para que solo se puede escribir números en los campos *precio* y *metros*

Para validar en php uso el método estatico *isAltainmuelle* de la clase **Validar.php**

```

static function isAltainmuelle($precio, $habitaciones, $banos, $metros, $provincia, $direccion,
$ciudad, $estado, $estados, $tipo, $tipos, $descripcion, $telefono, $horaContacto, $horas, $email,
$isactivo) {
    return self::isNumero($precio) && self::isEntero($habitaciones) &&
        self::isEntero($banos) && self::isEntero($metros) && self::isEntero($provincia) &&

```

```

        self::isLongitudMinima($direccion, "5") && self::isLongitudMinima($ciudad, "1") &&
        self::isEnum($estado, $estados) && self::isEnum($tipo, $tipos) &&
        self::isLongitudMinima($descripcion, "10") && self::isCorreo($email) &&
        self::isBoolean($isactivo) &&
        (self::isLongitudMinima($telefono, "1")?
            (self::isTelefono($telefono) && self::isEnum($horaContacto, $horas)):
            true);
    }

```

Para acceder al **back-end** accedo un formulario donde ingredo el *nombre (root)* y *clave (toor)* y en el archivo **phpLogin.php** compruebo si los valores son validos.

```

$root = $modelo->get($nombre, $clave);
if($root->getNombre() == null){
    $sesion->cerrar();
    header("Location:../index.php");
} else {
    $sesion->setRoot($root);
    header("Location:admin.php");
}

```

Si es valido uso el metodo setRoot de la clase **SesionSingleton.php**

```

function setRoot($root) {
    if ($root instanceof Root) {
        $this->set("__root", $root);
    }
}

```

y lo redirigo a al página **admin.php**, esta página y todas las del back-end comprueban que se haya registrado como usuario valido, con

```

$sesion->root("../index.php");

```

Este método *root* de la clase **SesionSingleton.php**, comprueba que se ha registrado si no lo redirige a la página que se le pasa.

```

function root($destino = "index.php") {
    if (!$this->isRoot()) {
        $this->redirigir($destino);
    }
}

```

Usa el método isRoot de la misma clase **SesionSingleton.php**

```

function isRoot() {
    return isset($_SESSION["__root"]);
}

```

Cuando borro una entrada también borro la carpeta y las imagenes de esa entrada

```

foreach (scandir($ruta) as $valor) {
    if ($valor == '.' || $valor == '..')
        continue;
    unlink($ruta . DIRECTORY_SEPARATOR . $valor);
}
rmdir($ruta);

```

5.- Recursos

Apuntes Desarrollo web en entorno servidor.

Apuntes Desarrollo web en entorno cliente.

Manual PHP: <http://php.net/manual/es/index.php>

Borrar archivos con php: <http://wordpresstutoriales.com/borrar-archivos-con-php/>

Listar archivos con php: <http://www.elcodigofuente.com/leer-archivos-directorio-carpeta-php-812/>

Formatear números en php: <http://www.desarrolloweb.com/articulos/formato-numeros-php.html>

Mostrar imagenes antes de subir:

<http://www.html5rocks.com/es/tutorials/file/dndfiles/>