



SEGUNDA EVALUACION. PHP

DESARROLLO WEB EN SERVIDOR

Ejercicio 1: (6 ptos)

Deseamos diseñar una aplicación para gestionar los jugadores de un equipo de fútbol. Para dicha aplicación trabajaremos con la tabla jugadores, para ella nos crearemos la base de datos EJERCICIO1 y accederemos con el usuario root y sin contraseña.

Se deja el enlace a "tablajugadores.sql" con lo necesario para crear la base de datos y la tabla jugadores.

De cada jugador guardaremos nombre y apellidos, una posición, un número de dorsal (único)

Dependencias a instalar con Composer.

autoload.php, aunque se puede hacer también con spl_autoload_register

Se va a entrega una estructura del proyecto ya terminada, para que vaya rellenando lo que se pide, pero debes usar los nombre de carpetas y ficheros que se entregan.

public: Es la carpeta donde crearemos todas las páginas que visualizaremos desde el navegador o páginas ".php" para procesar algo. Las páginas que tendrá serán:

"crearJugador.php" : está página es el "action" del formulario para crear un jugador. Controlaremos errores para no introducir un dorsal que ya existe, que nombre y apellidos no estén vacíos.

"fcrear.php": Es el formulario para crear el jugador.

Crear Jugador

Nombre:

Apellidos:

Dorsal:

Posición:

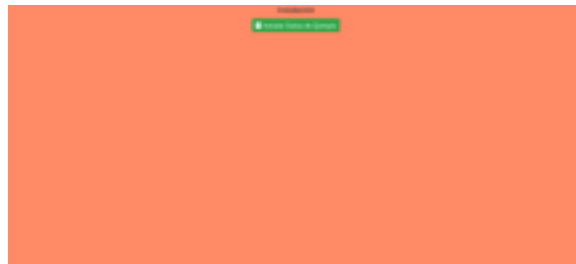
Código de barras:



2º Curso: Desarrollo Web en Servidor

"index.php": Es la página de inicio, si la base de datos Ejercicio1 no existe , llamara a "instalacion.php" para crearnos la base de datos y cargar la tabla jugadores con sus datos entregados . Si la base de datos existe y se accede bien , se saltara a "jugadores.php".

"instalacion.php": Carga la vista que aparece a continuación. Básicamente un botón para ir a "crearDatos.php" y crear la tabla con los datos de ejemplo



"jugadores.php": Muestra en una tabla los datos de los jugadores. Tiene un botón crear que llama al formulario para crear un jugador nuevo. Si un jugador NO tiene dorsal mostraremos "Sin asignar".

| Nombre Completo | Posición | Dorsal |
|----------------------------------|-------------------|-------------|
| Candelaria Betancourt, Alberto | Portero | 5 |
| Noriega Valdes, Diego | Portero | 95 |
| Padrón Reyna, Natalia | Portero | 30 |
| Pera Chaverra, María | Portero | 3 |
| Ruiz Ruiz, Esteban | Portero | Sin asignar |
| Expino De la Torre, Alex | Defensa | 34 |
| Leyva Muñoz, Angeles | Defensa | 51 |
| Nava Novales, Miguel Ángel | Defensa | 4 |
| Tovar Guardado, Francisca | Defensa | 44 |
| Ceballos Montes, Erik | Lateral Izquierdo | 13 |
| Garcido Collings, Claudia | Lateral Izquierdo | 23 |
| Téllez Correa, Nalene | Lateral Izquierdo | 21 |
| Fariñas Arrendondo, María Carmen | Centro | 14 |

3.- "sql" : Tiene el archivo ".sql", no es necesaria realmente.



"src": Contiene las Clases para gestionar la base de datos:

"Conexion.php": Para crear la conexión.

"Jugador.php": Para gestionar la tabla "jugadores". Aquí estarán los métodos para comprobar el dorsal, devolver los jugadores, insertarlos...En fin todos los que sean necesarios.

5.- "vendor": La genera Composer. Crearlo cada uno si lo considera

A parte de todo esto está el archivo: "composer.json".

Se pide:

1) (2,5 ptos) En la clase jugador.php se deben completar todos los métodos que se entregan y las líneas de código que falten.

2) (1,5 pto) En la clase conexión se debe cambiar el constructor metiendo un control de excepciones . Se intentamos acceder a una base de datos que no existe nos devuelve un getCode() con valor 1049. Si accedemos con una password incorrecta devuelve un getCode() con valor 1045. Si la base de datos no existe la crea y su tabla. Si existe guarda su conexión.

Si queremos introducir un tupla podemos, aunque en creardatos.php, incluimos mas registros.

3) (2 ptos) Añadir todo lo necesario en los distintos .php para que la aplicación funcione tal y como se describe anteriormente. En crearjugadores es importante almacenar distintos mensajes para cada posible error.

Nota: Si se cambian los nombres de los ficheros, o de las variables , carpetas, etc, SE considerara mal hecho ese apartado.



Ejercicio 2: (4 ptos)

Se va a realizar una aplicación web para una inmobiliaria. Se puede cargar el script de la base de datos Inmobiliaria, directamente sobre phpmyadmin sin necesidad de programar nada.

La aplicación debe cumplir las siguientes especificaciones:

1º (1,5 ptos)

Como la base de datos no tiene datos, en el index.php tendremos que cogerlos y cargarlos desde viviendas.xml y desde usuario.xml

2º. (0,5 ptos)

Cuando entramos en el index.php ,lo primero que debemos hacer al entrar en la aplicación es introducir nuestro usuario y contraseña . Si somos cualquier usuario de los de la tabla, iremos a tipos.php con idea de comprar una vivienda. Tipos.php es el primer de una serie de formularios.

3º Todos los datos que se presentan en los formularios, los cogerá de las tablas de la base de datos nunca serán datos estáticos.

Los siguientes pasos , pretenden preguntarnos distintos filtros hasta encontrar las casas que deseamos ver.

En concreto, el asistente irá preguntando lo siguiente en cada paso. Cada paso estará en un php.

- Paso 1: tipo de la vivienda (tipo.php)
- Paso 2: zona de la vivienda (zona.php)
- Paso 3: número de dormitorios y rango de precio de la vivienda (precio.php)
- Paso 4: características extras de la vivienda

Las figuras siguientes muestran los distintos pasos del asistente, mientras que la figura última muestra el resultado de la consulta.

Las siguientes figuras, se presentan cada una en un .php distinto.

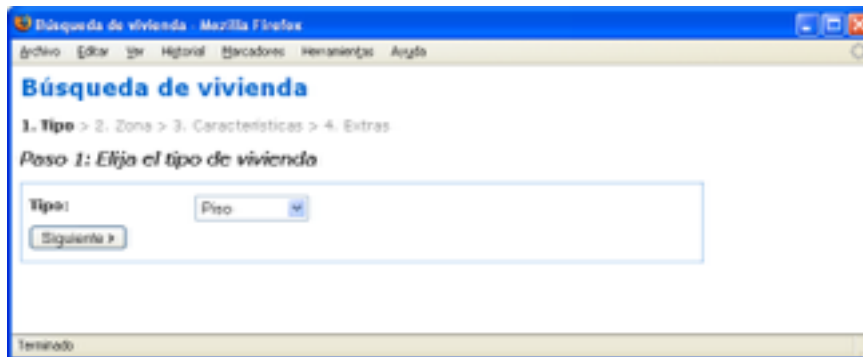


Figura 1: Tipo



Figura 2: Zona



Figura 3: Dormitorios y precio

Figura 4: Extras

Por ultimo otro formulario donde podemos especificar extras como, jardín, piscina, zonas comunes, garaje, padel, etc.



Figura 5: Resultados de la búsqueda

5º (1,5 pts)

Cada vivienda presentada en el listado de las que cumplen los filtros seleccionados, presentará un check para poder decidir si la queremos o no.

Si la queremos debe guardar la información en la tabla compras junto con la fecha.

Nota: El alumno debe subir al aula virtual un fichero con su nombre y que contendrá todas las carpetas que se necesitan, La duración del examen es de 9:20 a 13:00.