

1º Proyecto : Blog desde 0

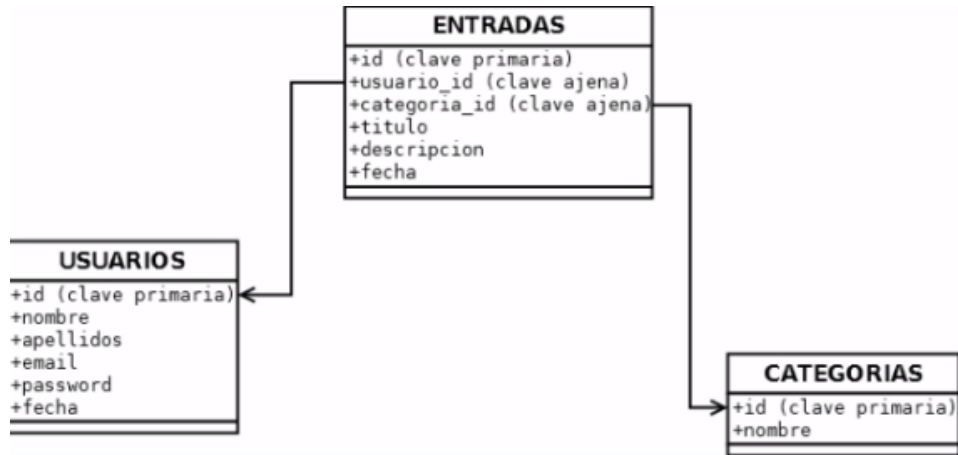
Este primer proyecto consiste en realizar un blog desde 0 de tema libre utilizando las herramientas que llevas aprendidas: HTML5, CCS3, PHP y MySQL utilizando programación estructurada y dejando orientación a objetos para la 2ª práctica. El blog dispondrá de varios perfiles de usuarios que podrán crear categorías y entradas para esa categoría. Cada usuario solamente podrá gestionar (editar y borrar) las entradas que ha creado.

Los pasos para seguir son los siguientes

1. Diseño de la Base de datos usuarios
2. Maquetación
3. De HTML a PHP
4. Registro de usuarios
5. Login de usuarios
6. Categorías y entradas
7. Actualizar datos usuario y cerrar sesión
8. Edición y borrado de entradas (**Subir nota**)
9. Buscador PHP (**Subir nota**)

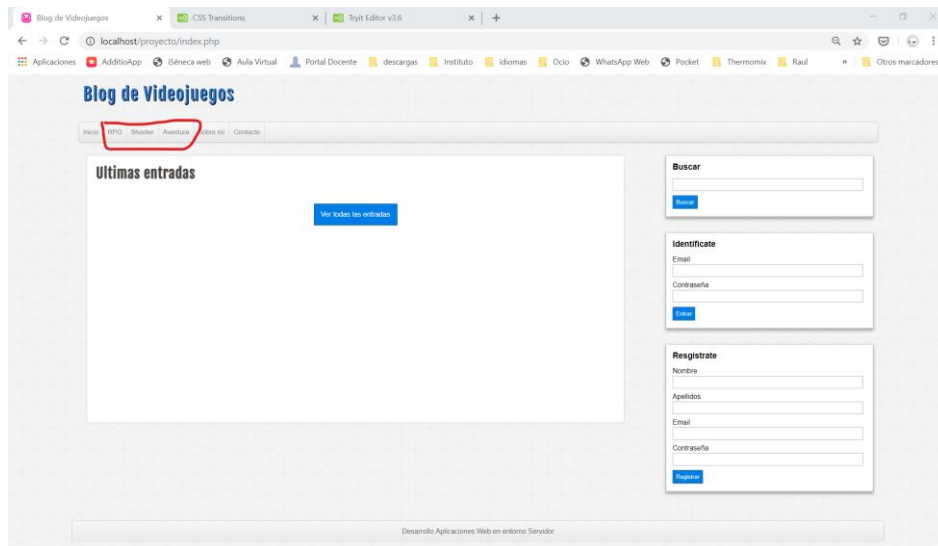
1. Diseño de la Base de datos usuarios

Las tablas y atributos que usaremos son los siguientes (El Script se proporciona en la práctica).



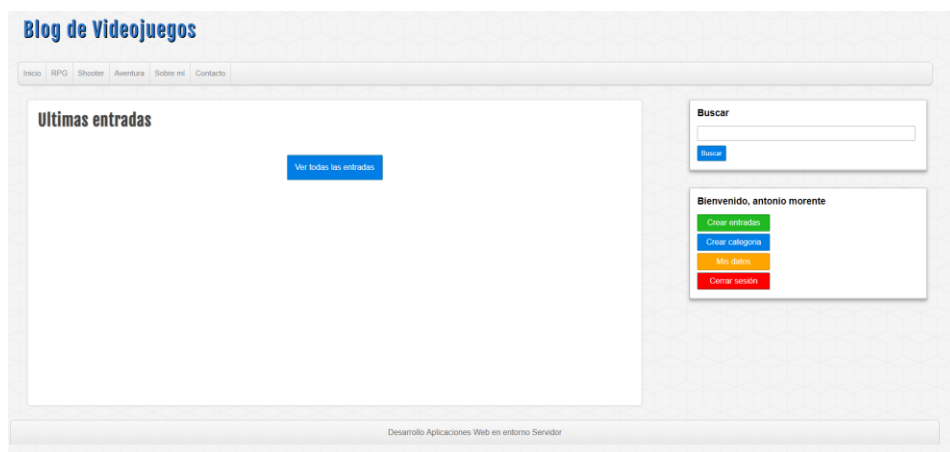
2. Maquetación

Realiza una maquetación parecida a la fotografía utilizando



- Etiquetas de html5 (nav, aside , footer , header, article) y maquetación. Con grid
- Utiliza un conjunto de carpetas organizado para el proyecto.
- Página principal (index.php): En la barra del menú tendrá 3 opciones estáticas (Inicio, sobre mí y contacto) y la opciones que están en rojo son categorías que ha creado los usuarios que aparecerán de forma dinámica
- HTML y CSS externo
 - Utiliza un CSS organizando indicando siempre en la capa que estás trabajando
 - Introduce un degradado de abajo arriba en el menú con dos tonos de grises
 - Introduce sombra (box-shadow) para los div que están dentro aside.
 - El logo blog de videojuegos es un enlace a la página principal, utiliza text-shadow y transition
 - Ultima entradas es un h1
 - Para cada entrada utiliza la etiqueta article

Cuando el usuario esta logueado



Crear entradas

3. De HTML a PHP

Se recomienda que el código que mas se repita se incluya en una carpeta que vamos a llamar includes y llamaremos a los ficheros con `require_once 'fichero.php'`. Realiza como mínimo los siguientes ficheros

- **Cabecera.php**
- **Lateral.php**: En cada uno de los formularios tendrás que comprobar los errores dentro de las variables de sesión que han podido producirse en PHP y mostrar dentro del formulario. Una vez mostrados tendrás que llamar a una función para borrarlos. Depende si existe la variable `$_Session['usuario']` se mostrará el formulario para loguearse o el formulario con la gestión de entradas y categorías.
- **Menú.php**
- **Pie.php**
- **Conexión.php** : Además de conectar con la base de datos , comprueba si existe `$_SESSION` para iniciar la sesión (`sesión_start()`).
- **Helpers.php**: Un fichero con las funciones que veas que te pueden resultar útiles. Por ejemplo (MostrarCategorias, mostrarEntradas, MostrarErrores, etc)

Recomendación:

La validación del formulario podría hacer en el lado del cliente utilizando Javascript pero es mucho más seguro realizar todas las comprobaciones dentro del PHP devolviendo el error correspondiente dentro de un array

Es aconsejable que antes de cada formulario HTML se ejecute la función **Mostrar errores(\$_SESSION['errores']):string**; que mostrará por pantalla todos los errores que haya en el formulario. Comprueba antes de ejecutar la función si existe `$_SESSION['errores']` por si quieres mostrar un mensaje de que ha ido todo correctamente. Una vez mostrados los mensajes de error tendrás que liberar la variable.

4. Registro de Usuarios

- Realiza unos formularios HTML utilizando atributos del estilo: **required, placeholder, size, maxlength,...**
- Utiliza la función `Mostrarerrores()` delante del formulario HTML y `borrarErrores()` detrás.
- Recoger los valores del formulario

```
$nombre = isset($_POST['nombre']) ? mysqli_real_escape_string($db, trim($_POST['nombre'])) : false;
```

Nota: Aquí tenéis un ejemplo para recoger el valor de nombre, la función `mysqli_real_escape_string` sirve para evitar la inyección de sql a través de formulario y con la función `trim` eliminamos todos los espacios

- Comprueba cada uno de los campos y almacena los errores en un array

```
$errores = array();
```

- Cifrar la contraseña (4 veces)

```
$password_segura = password_hash($password, PASSWORD_BCRYPT, ['cost'=>4]);
```

- Insertar el usuario si el array de errores es 0

```
$sql = "INSERT INTO usuarios VALUES(null, '$nombre', '$apellidos', '$email', '$password_segura', CURDATE());";  
$guardar = mysqli_query($db, $sql);
```

- Si no hay errores guarda en una variable de sesión (`$_SESSION['completado']`), si hay un error en el insert (`$_SESSION['errores']['general']`) y con cualquier error anterior (`$_SESSION['errores']=$errores;`)
- Redirigir a la página principal

```
header('Location: index.php');
```

5. Login de usuarios

- Utiliza la función `Mostrarerrores()` delante del formulario HTML y `borrarErrores()` detrás.
- En el fichero `php` iniciar la sesión y la conexión a la BD

Nota: Puedes utilizar una función `borrarErrores()`; dentro de helpers para comprobar cada uno de los errores que utilices para borrarlos

- Recoger los datos del formulario
- Comprobar contraseña utilizando hash

```
$verify = password_verify($password, $usuario['password']);
```

- Consulta para comprobar la credenciales del usuario

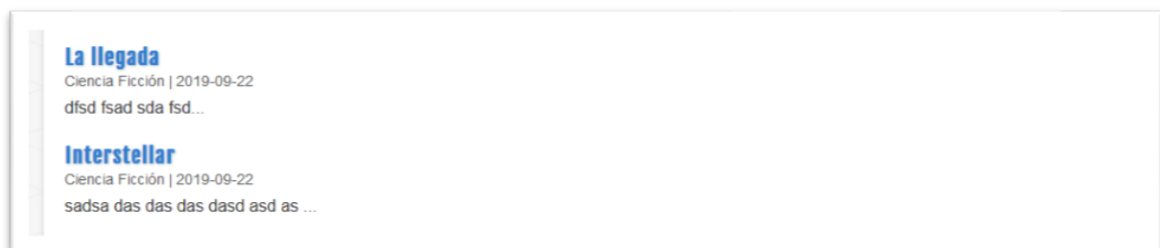
- Utilizar una sesión para guardar todos los datos del usuario si esto correcto. Si no existe el email, no se verifica el password o hay algún error en el acceso a la base de datos guardaríamos el mensaje en `S_SESSION['error_login']` para mostrarlo en el `index.php`.

```
$_SESSION['usuario'] = $usuario;
```

- Si algo falla enviar un sesión con el fallo
- Redirigir al `index.php`

6. Categorías y entradas

- Para **mostrar las categorías** en el menú entre inicio y sobre mi se puede utilizar una función externa (`conseguirCategorías($conexion)`) en helpers que te devuelva el array de categorías o falso si no hubiera ninguna. Recuerda que cada categoría tendrá un enlace de estilo `categoría.php?id=<? Categoría['id']>`
- Para mostrar todas las entradas de una categoría también podríamos utilizar una función en helpers que se llame `conseguirEntradas($conexion)`. Esa función podrías reutilizarla para mostrar todas las entradas o mostrar todas las entradas de una categoría y para la búsqueda (introduciendo diferente Where al SQL) dependiendo de los parámetros que le pases a la función o hacer varias funciones diferentes. Te recuerdo que dentro del select tendrás que hacer un INNER JOIN de entradas y categorías.



- **Crear entradas**

The screenshot shows the 'Crear entradas' form. It includes fields for 'Titulo:' and 'Descripción:', a 'Categoría' dropdown menu currently set to 'RPG', and a 'Guardar' button. To the right, there is a 'Buscar' search bar and a sidebar with a 'Bienvenido, ds sf' message and a list of buttons: 'Crear entradas' (highlighted with a red circle), 'Crear categoría', 'Mis datos', and 'Cerrar sesión'.

- Crear categorías

7. Actualizar datos usuario y cerrar sesión

- A partir del formulario de registro crea un nuevo fichero misdatos.php que actualicen el nombre, apellidos y Email
 - En el formulario t
 - Redirige a misdatos

- Crea un fichero cerrar.php que
 - Inicie la sesión
 - Compruebe si existe la variable `$_SESSION['usuario']` y si es así libera (**Unset**) la variable y destruye la sesión (**session_destroy();**)
 - Redirige a index.php

8. Edición y borrado de entradas (Subir nota)

- Añade 2 iconos pequeños al lado de cada título de entrada(editar y borrar) solamente si el usuario conectado es igual al usuario que ha creado la entrada. Cada uno de los iconos tendrá que enviar por la url el identificador de la entrada
- Haz dos ficheros borrar-entrada.php (condición where comprueba el identificador del usuario y de la entrada) y editar-entrada.php (formulario con los datos introducidos) y se reenvía a guardar-entrada.php.

9. Buscador (Subir nota)

- Crea el formulario en el aside y un nuevo fichero buscar.php
- Dentro del fichero buscar.php comprueba si no existiera la variable `$_POST['busqueda']` entonces redirige al index.php y si no crea la página con los resultados de la búsqueda
- Para generar los resultado de la búsqueda se realiza de forma parecida a los apartados anteriores creando una nueva función `buscarEntradas($conexion,$busqueda)` y que te devuelva un array con los resultados.