

RELOJ.JS

En mi web he querido añadir un reloj cuando el usuario inicia sesión. Para esto he creado un script JavaScript

- 1) Obtengo del documento html el elemento tiempo por su id, y lo guardo en la variable let html.

```
let html = document.getElementById("tiempo");
```

Esto está en la cabecera.

```
<span id="tiempo">00: 00 : 00</span>
```

- 2) Date ya existe en JavaScript, y permite obtener el tiempo. En Date puedo usar getHours(), getMinutes() y getSeconds() para las horas minutos y segundos. Como quiero que el formato del tiempo sea correcto, imaginemos que la hora da las 9, pues en vez de mostrarse "9", se mostrará 09. Así con los minutos y segundos también. Por último, con html.innerHTML, sobrescribo sobre el elemento tiempo la hora.

```
function actualizarReloj() {  
    const tiempo = new Date();  
    let horas = tiempo.getHours();  
    let minutos = tiempo.getMinutes();  
    let segundos = tiempo.getSeconds();  
  
    if(horas<10)  
        horas = "0"+horas;  
    if(minutos<10)  
        minutos = "0"+minutos;  
    if(segundos<10)  
        segundos = "0"+segundos;  
  
    html.innerHTML = horas+" : "+minutos+" : "+segundos;  
}
```

- 3) Hay un problema con esto, y es que para que esto funcione bien, se tiene que ejecutar cada segundo. Esto lo hago con

```
setInterval(actualizarReloj, 1000);
```

Cada 1000ms = 1s, se llama a la función actualizarReloj para que el reloj se muestre siempre bien.

No he podido usar el reloj en las colecciones.php por que me sincronizaba la ventana emergente, y no podía hacer uso de esta.

VENTANAFLOTANTE.JS

Para la ventana flotante, he trabajado con el DOMContent. He añadido un listener en el documento HTML, para que la función se ejecute una vez el HTML ha sido cargado completamente. Los listener son “escuchadores” que esperan a que ocurran algo para ejecutar algo.

```
document.addEventListener('DOMContentLoaded', function() {
```

Como en las colecciones voy a tener varias obras, por cada obra voy a añadir 2 listener, uno para cuando se pasa el ratón por encima de la imagen, y otro para cuando se quita la imagen.

Antes de nada tengo que obtener todas las imágenes y almacenarlas, lo hago con querySelectorAll sobre el document.

```
const imagenes = document.querySelectorAll('.img');
```

por cada imagen...

```
imagenes.forEach(imagen => {
```

- 1) Listener mouseover (evento definido en javascript este mouseover).

```
imagen.addEventListener('mouseover', function() {

    titulo = this.getAttribute('data-titulo');
    tema = this.getAttribute('data-tema');

    let ventanaEmergente =
this.parentNode.querySelector('.ventanaEmergente');

    if (!ventanaEmergente) {
        ventanaEmergente = document.createElement('div');
        ventanaEmergente.classList.add('ventanaEmergente');
        this.parentNode.appendChild(ventanaEmergente);
    }

    ventanaEmergente.innerHTML = `<strong>Título:</strong>
${titulo}<br>
                                <strong>Tema:</strong>
${tema}<br>`;

    ventanaEmergente.style.display = 'block';
});
```

Obtengo el título y la imagen seleccionándolas de la obra, y busco la ventana emergente para ver si existe siendo descendiente del nodo padre, que es la

imagen. Si no existe, se crea el div, que va a contener la ventana emergente. Se le añade la clase con classList, y se pone como hijo del elemento padre con parentNode.appendChild.

Con innerHTML establezco el contenido html de la ventana, y con .style.display digo que se muestre en forma de block.

2) Listener mouseout

```
imagen.addEventListener('mouseout', function() {  
    const ventanaEmergente =  
this.parentNode.querySelector('.ventanaEmergente');  
    if (ventanaEmergente) {  
        ventanaEmergente.style.display = 'none';  
    }  
});
```

Para hacer desaparecer la ventana flotante, se cambia el style.display a none, y así deja de mostrarse.

TABLAS USADAS

```
MariaDB [dbjosemiaco2003_pw2324]> describe Obra;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
titulo	varchar(255)	NO		NULL	
siglo	varchar(50)	YES		NULL	
autor	varchar(255)	YES		NULL	
tema	varchar(255)	YES		NULL	
imagen	blob	YES		NULL	
descripcion	text	YES		NULL	

```
MariaDB [dbjosemiaco2003_pw2324]> describe Comentario;
```

Field	Type	Null	Key	Default	Extra
idComentario	int(11)	NO	PRI	NULL	auto_increment
usuario	varchar(50)	NO	MUL	NULL	
titulo	varchar(255)	NO		NULL	
texto	text	NO		NULL	
valoracion	int(11)	NO		NULL	

```
MariaDB [dbjosemiaco2003_pw2324]> describe Usuario;
```

Field	Type	Null	Key	Default	Extra
dni	varchar(9)	NO	PRI	NULL	
usuario	varchar(50)	NO	UNI	NULL	
pass	varchar(255)	NO		NULL	
email	varchar(100)	NO		NULL	
nombre	varchar(50)	NO		NULL	
apellido1	varchar(50)	NO		NULL	
apellido2	varchar(50)	NO		NULL	
direccion	varchar(255)	NO		NULL	
admin	char(1)	YES		N	

GESTION DE LA PAGINACION

Al mostrar las obras, hago uso de la función count para contar cuantas obras estoy mostrando.

```
$totalObras = count($obras);
```

Como el máximo de obras es de 9 por página, establezo el máximo

```
$limiteObras = 9;
```

Obtengo el número de página en el que estoy, y si no estoy en ninguna, se pone automáticamente en la 1

```
$paginaActual = isset($_GET['page']) ? (int)$_GET['page'] : 1;
```

Calculo cuántas páginas necesitaré en función de las obras que tenga

```
//Calculo el total de páginas que van a hacer falta, redondeando hacia arriba
//Por ejemplo, si tengo 11 obras, 11/9 = 1,2. Por lo tanto, hacen falta 2 páginas
$paginasNecesarias = ceil($totalObras / $limiteObras);
```

Como las paginas que se van a mostrar en cada pagina se van a almacenar en un array, necesito calcular el índice por dónde se va quedando en el array

```
$indice = ($paginaActual - 1) * $limiteObras;
```

Por último, hago uso de la función array_slice, que me obtiene la porción del array que me interesa

```
$obrasPagina = array_slice($obras, $indice, $limiteObras);
```

Ahora, para ver si tiene que aparecer la flecha a la derecha o a la izquierda, uso lo siguiente

```
<?php
function generarUrlPaginacion($valor, $paginaActual,
$columna) {
    if ($columna === 'tema') {
        return "?tema=$valor&page=$paginaActual";
    } elseif ($columna === 'siglo') {
        return "?siglo=$valor&page=$paginaActual";
    } elseif ($columna === 'autor') {
        return "?autor=$valor&page=$paginaActual";
    } else {
        return "?page=$paginaActual";
    }
}
```

```
}  
}  
?>
```

Esto me devuelve el formato de la url al pasar de página

```
<?php if($paginaActual > 1) : ?>  
    <a href="<?php echo generarUrlPaginacion($valor,  
$paginaActual - 1, $columna); ?>">  
          
    </a>  
    <?php endif; ?>  
  
    <?php if($paginaActual < $paginasNecesarias) :?>  
        <a href="<?php echo generarUrlPaginacion($valor,  
$paginaActual + 1, $columna); ?>">  
              
        </a>  
    <?php endif; ?>
```

MOSTRAR OBRAS DINAMICAMENTE

Primero de todo, obtengo los valores que se van a mostrar con la siguiente función

```
$temas = Obra::obtenerFiltro('tema');
$siglos = Obra::obtenerFiltro('siglo');
$autores = Obra::obtenerFiltro('autor');

public static function obtenerFiltro($columnaFiltro){

    //Inicio la conexión
    $conexion = parent::conectar();
    //Consulta a realizar
    $consulta_sql = "SELECT DISTINCT $columnaFiltro FROM Obra
ORDER BY $columnaFiltro";

    try {

        //Preparar la consulta
        $st = $conexion->prepare( $consulta_sql);
        //Ejecutarla
        $st->execute();
        //Obtener los valores únicos como un array
        $filtros = $st->fetchAll(PDO::FETCH_COLUMN);
        //Desconectarme
        parent::desconectar($conexion);
        //Devolver los filtros
        return $filtros;
    }
    catch ( PDOException $e ) {
        parent::desconectar( $conexion );
        die( "No se pudo obtener el filtro: " . $e->getMessage()
);
    }

}
```

Una vez que obtengo los valores que se van a mostrar, los muestro en el aside. Al hacer click en cada una de las temáticas, construyo una nueva URL con ?loquesea=

Así también con siglo y autor.

```
foreach ($temas as $tema) {
    $url = "coleccion.php?tema=" . $tema ;
    echo "<li><a class='enlace' href='$url'>" . $tema
. "</a></li>";
}
```

¿COMO HE HECHO LA CONEXIÓN PHP?

Para la conexión he hecho uso de la dada en el guion, basada en PDO.

```
<?php
require_once('configuracion.php');
abstract class DataObject {

    protected $datos = array();
    public function __construct( $datos ) {
        foreach ( $datos as $clave => $valor )
            if ( array_key_exists( $clave, $this->datos ) ) $this->datos[$clave] = $valor;
    }

    public function devolverValor( $campo ) {
        if ( array_key_exists( $campo, $this->datos ) ) {
            return $this->datos[$campo];
        } else die( "Campo no encontrado" );
    }

    protected static function conectar() {
        try {
            $conexion = new PDO( DB_DSN, DB_USUARIO, DB_CONTRASENIA );
            $conexion->setAttribute( PDO::ATTR_PERSISTENT, true );
            $conexion->setAttribute( PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION );
        } catch ( PDOException $e ) {
            die( "Conexión fallida: " . $e->getMessage() );
        }

        return $conexion;
    }

    protected static function desconectar( $conexion ) {
        $conexion = "";
    }
}
?>
```

En mi configuración.php he metido mis datos.

```
<?php
define("DB_DSN", "mysql:host=localhost;dbname=dbjosemiaco2003_pw2324" );
define("DB_USUARIO", "pwjosemiaco2003" );
define("DB_CONTRASENIA", "23josemiaco200324" );
?>
```


COMO HE MANEJADO EL TEMA DE LAS SESIONES

Al principio de cada documento php, he añadido lo siguiente, para recuperar las variables de sesión de cada usuario.

```
<?php
    //Iniciar una sesión
    session_start();

?>
```

Cuando el administrador inicia sesión, almaceno en una variable de sesión si es admin.

```
$_SESSION['USER'] = $usuario;
    $admin = Usuario::esAdmin($usuario);
    $_SESSION['ADMIN'] = $admin;
```

El valor de admin puede ser 'Y' o null. La 'Y' la pongo yo previamente en la base de datos. Por lo tanto, en el header hago la comprobación del valor de la variable de sesión, y si es igual que 'Y' tendrá acceso a algunas cosas que los usuarios no.

```
//Obtengo si es admin o no
    $admin = $_SESSION['ADMIN']['admin'];

    //Si es administrador, además de un apartado para
hacer logout, tendrá un apartado en el que añadir las obras
    if($admin == 'Y'){
```

En las páginas que el usuario normal no puede acceder, he añadido lo siguiente

```
<?php
    //Iniciar una sesión
    session_start();

    require_once('obra.class.php');

    $admin = $_SESSION['ADMIN']['admin'];
    if ($admin != 'Y'){
        echo 'NO TIENES ACCESO SI NO ERES ADMINISTRADOR.';
        echo 'VUELVE AL INICIO';
        echo '<a href="index.php"> Inicio </a>';
        exit();
    }

?>
```

El logout lo hago destruyendo la sesión

```
<?php

//Inicio sesión
session_start();

//Elimino las variables de sesión
session_unset();

//Destruyo la sesion
session_destroy();

//Redirigo al index
header("Location: index.php");

?>
```

VALIDAR FORMULARIOS.JS

Para cada formulario he creado una función distinta. Explico solo una de ellas.

De esta forma obtengo los elementos

```
const dni = document.getElementById('dni').value;
const usuario = document.getElementById('usuario').value;
const password = document.getElementById('pass').value;
const email = document.getElementById('email').value;
const nombre = document.getElementById('nombre').value;
const apellido1 = document.getElementById('apellido1').value;
const apellido2 = document.getElementById('apellido2').value;
const direccion = document.getElementById('direccion').value;
```

Tanto para el DNI como para el email, he usado expresiones regulares.

```
const dniExpresion = /^[0-9]{8}[TRWAGMYFPDXBNJZSQVHLCKE]$/;
const emailExpresion = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
```

Ejemplo de validacion del campo email

```
// Validar longitud del DNI
if (dni.length != 9 || !dniExpresion.test(dni)) {
    mensajeError += "¡¡¡El DNI no es válido!!!\n";
}
```

Todo se va almacenando en un mensaje.

Si ese mensaje tiene contenido, se devuelve false

```
if (mensajeError != "") {
    alert(mensajeError);
    return false;
}
```

Y si no se devuelve true. En el formulario, en onsubmit, se llama a la función.