Hoja04_BBDD_06

1.- Crear un formulario que permita el alta de productos. Se debe configurar un archivo .env donde se encuentren definidas las variables DB_DSN, DB_USERNAME y DB_PASSWORD para realizar la conexión a una base de datos que debes crear denominada dwes06_productos.

Recuerda que para poder subir archivos el formulario debe tener el atributo

```
enctype="multipart/form-data"
```

Crear una clase ConexionBD, que utilice las variables definidas y que cumpla con el patrón singleton. Solo tendrá los métodos de conexión a la BD.

La tabla de productos, con los siguientes campos:

- Id, que es clave primaria y generada por el sistema
- Nombre, es un campo alfanumérico de 100 caracteres y no puede ser nulo
- Precio, es un campo con decimales y no puede ser nulo
- Descripción, es un campo de tipo text y no nulo
- Imagen, es un campo alfanumérico de 255 caracteres que almacena el nombre de la imagen

Creamos un fichero helper.php, donde tendremos unas funciones genéricas de validación:

- Función validar requerido, tiene un parámetro de entrada de tipo string y devolverá un booleano, es decir, que si existe valor devuelve un true.
- Función validar numérico, tiene un parámetro de entrada de tipo string y devolverá un booleano
- Función validar subida fichero, tiene como parámetro de entrada un array, comprueba si ha sido subido el fichero de la imagen o no por lo que devuelve un booleano. Como ayuda consultar la documentación `https://www.php.net/manual/es/features.file-upload.post-method.php`
- Función validar formato Imagen, tiene como parámetro de entrada un tipo string. Los formatos válidos solo son jpeg y png
- Función limpiar texto tendrá un parámetro de entrada y devolverá un string, La cadena de salida solo puede contener las siguientes etiquetas de html:
  `<a><strong><em><h1><h2><h3><h4><h5><h6><ul><ol><li><blockquote><br><div><span><table><thead><tbody><tr><th><td>`
- Función limpiar entrada, desinfecta todos los campos de entrada y llama a la función limpiar texto, puesto que el filtro `FILTER_SANITIZE_STRING is DEPRECATED on PHP 8.1.`
- Función redireccionar, tiene como parámetro de entrada un tipo string. Realiza una petición http get del path recibido como parámetro.

Creamos un fichero procesa.php, donde se realizan las validaciones de los datos del formulario:

- Que todos los datos son obligatorios
- Que el precio es un dato numérico
- Que la subida de la imagen ha sido correcta
- Que el formato de la imagen es correcto, solo permitido jpeg y png
- Limpiar la entrada de datos
- Las imágenes se guardarán en un subdirectorio productos, el nombre del archivo imagen será un identificador único seguido del nombre del archivo de la imagen a subir al servidor. En el campo imagen guardamos el nombre de la imagen generado.

  Ejemplo: el archivo a subir se denomina airpods-pro.jpg en el directorio /productos y en el campo imagen de la bd figura 6544c7349fef5-airpods-pro.jpg

- Los mensajes de error que se pueden dar son:
    1. Por favor, rellena todos los datos
    2. No se puede procesar el archivo
    3. El archivo no tiene una extensión válida
    4. Por favor, introduce un precio válido
    5. No se ha podido guardar el producto en base de datos

- Si todo es correcto, mostramos el mensaje el producto ha sido dado de alta correctamente

Para llevar esta aplicación también tendremos en cuenta el patrón repositorio que implica definir:

- Una interfaz repositorio del producto que tendrá el método crear como parámetro de entrada el producto y como salida un booleano.
- Una clase PDOCrearProducto, que implementa la interfaz repositorio producto creada y desarrolla el método crear producto.
- La tabla productos se creará cuando no exista por programación.

Y también vamos a implementar principio SOLID open/close, que significa nos va a permitir con cualquier proveedor de datos (pdo, mysli,msql,…) con solo cambiar la clase desarrollada en nuestro caso PDOCrearProducto funciona nuestra aplicación.

- Una clase crearProducto, donde en el constructor pasamos la interfaz y la función crear producto que llamará a la interfaz.

El principio SOLID "O" en SOLID se refiere al Principio Abierto/Cerrado. Este principio establece que las entidades de software (clases, módulos, funciones, etc.) deben estar abiertas para la extensión, pero cerradas para la modificación. En otras palabras, cuando necesites realizar cambios en el comportamiento de un sistema, no debes modificar el código existente, en lugar de eso, debes extenderlo a través de nuevas clases o módulos.

Ejemplo de ejecución

Y finalmente

Y en bd, se ha evitado que se ejecute el javascript y simplemente lo guarda como texto plano

id	nombre	precio	descripcion	imagen
1	alert('producto1')	70099	Cámara teleobjetivo de 10 MP; cámara frontal de 40...	65451a5973f16-galaxy-s21-ultra.jpg

Y la imagen en el servidor está en el directorio **productos** dentro de tu proyecto.