

Realizar utilizando PHP una aplicación web de una galería que permita a los usuarios almacenar, publicar y gestionar sus propias imágenes utilizando la P.O.O. y una base de datos MariaDB:

- En la base de datos:
  - Se creará un usuario de nombre: *gallery* con una base de datos (esquema) asociada con el mismo nombre y contraseña *loimportantenoelimagen*:
  - La base de datos (esquema) dispondrá de:

- Una tabla *USERS* con los siguientes campos:

USER	PASSWORD	AUTHOR
(PrimaryKey, Cadena (128))	(Cadena (4096))	(Unique, Cadena (256))

- Una tabla *IMAGES* con los siguientes campos:

ID	USER	NOMBRE	FECHA	PUBLICO	MINI
(PrimaryKey, Entero, AUTO_INCREMENT)	(ForeignKey:USERS(USER), Cadena (128))	(Cadena (256))	(Fecha)	(Booleano)	(Blob)

- Se creará un directorio de nombre: *img* que almacenará las imágenes que subamos al servidor.
- Se creará un directorio de nombre: *meninas* que almacenará las clases (un fichero php por clase).
- Se creará un fichero *config.php*:
  - Contendrá las constantes de la aplicación (como mínimo las de la conexión a la bbdd *gallery*).
- Se creará un fichero *index.php*:
  - Toda la página debe generarse desde PHP. No habrá código HTML fuera de los bloques de PHP. Se embebe HTML en PHP no PHP en HTML.
  - La página mostrará:
    - En la parte superior derecha:
      - ✓ Un formulario para loguearse, con los campos: usuario y contraseña. El formulario debe entregarse a la propia página *index.php*, la cual:
        - Si el usuario y la contraseña son correctos, almacenará el login y se reemplazará el formulario por un enlace a la página de gestión de las imágenes del usuario (*user.php*), el enlace se abrirá en una nueva ventana.
      - ✓ La opción Registrarse, se enlazará a la página de gestión de las imágenes del usuario (*user.php*), el enlace se abrirá en una nueva ventana. En el caso, que el usuario se haya logueado, no se incluirá esta opción.
    - En la parte central:
      - ✓ Un formulario para buscar imágenes públicas, con los campos:
        - Nombre: campo de texto para buscar todas las imágenes que incluyan la cadena en el nombre.
        - Autor: desplegable que se rellenará con los nombres de todos los autores del sistema. Para buscar todas las imágenes de ese autor.  
[Opcional: permitir seleccionar múltiples autores]
        - Rango de fechas: dos controles de tipo fecha. Para buscar todas las imágenes entre la fecha menor y la mayor (incluidas).
      - ✓ Este formulario debe entregarse a la propia página *index.php*, la cual, pintará en forma de galería (en la parte inferior) todas las miniaturas que cumplan todos los requisitos recibidos. En el caso, de no recibir ningún requisito se mostrarán las últimas 16 imágenes públicas.

- En la parte inferior:
      - ✓ Una galería, para cada imagen se pintará su miniatura y se enlazará con la imagen original, el enlace se abrirá en una nueva ventana.
      - ✓ *[Opcional] Abrir en primer plano la imagen en lugar de en una nueva ventana.*
  - Se creará un fichero *user.php*:
    - Toda la página debe generarse desde PHP. No habrá código HTML fuera de los bloques de PHP. Se embebe HTML en PHP no PHP en HTML.
    - Si el usuario no se ha logueado se mostrará el formulario para que se registre, con los campos: usuario, password, repetición password y autor. Este formulario debe entregarse a la propia página *user.php*:
      - Antes de añadir un nuevo usuario se debe verificar que:
        - ✓ usuario: cuenta con una longitud de al menos 2 caracteres y máximo 128 después de eliminar los espacios en blanco al principio y al final. Además, el usuario no debe existir previamente en la base de datos.
        - ✓ password y repetición password: que sean iguales, tengan una longitud mínima de 4 caracteres y máxima de 64, por último, que contenga al menos una letra y un número.
        - ✓ autor: mismas condiciones que para usuario.
      - Si todo es correcto, se añadirá en nuevo usuario a la tabla *USERS*, password se almacenará cifrado con un hash, a continuación, se redireccionará al usuario a la página *index.php*.
      - En caso contrario, se informará al usuario del error y se pintará de nuevo el formulario de registro.
    - Si el usuario se ha logueado se mostrará:
      - La opción de desloguearse: eliminará el login y redirigirá al usuario a la página *index.php*.
      - Un formulario para subir una nueva imagen, solicitará: el fichero a enviar.
        - ✓ El formulario debe entregarse a la propia página *user.php*.
        - ✓ Como mínimo deben aceptarse imágenes: jpg y png
        - ✓ Si la imagen enviada es correcta se almacenará en el directorio *img*.
        - ✓ Si la imagen enviada es correcta se almacenará en la bbdd:
          - Un ID de imagen que se generará automáticamente.
          - El nombre del usuario al que pertenece la imagen.
          - El nombre de la imagen (incluida extensión).
          - La fecha actual.
          - Por defecto las imágenes serán privadas (no publicas).
          - Se generará la miniatura (png) la cual se almacenará en el campo *MINI*.
      - Una galería con todas las imágenes subidas al servidor por el usuario:
        - ✓ Para cada imagen se cargará su miniatura.
        - ✓ Junto a la miniatura debe añadirse:
          - Una opción para borrar la imagen.
          - Una opción para hacer pública/privada la imagen.
  - Se creará un fichero *mini.php*:
    - Recibirá un identificador de imagen
    - Si la imagen es pública y/o pertenece al usuario logueado:
      - En caso afirmativo, se devolverá la correspondiente miniatura.
      - En cualquier otro caso, se generará y devolverá una miniatura en blanco.
- [Opcional: generar miniaturas personalizadas para cada caso incorrecto]*

- Se modelarán como mínimo los siguientes objetos:
  - Usuario: encargado de gestionar toda la lógica de los usuarios.
  - Galeria: encargado de gestionar toda la lógica de las imágenes.
  - Pagina: encargado de generar la página HTML.
  - BaseDatos: interacciona directamente con la bbdd, el único. Puede ser llamado por los anteriores.
- Podrán modelarse todas las clases que se deseen, cada clase debe situarse en su propio fichero php. Además, para importar las clases se utilizará la carga automática utilizando el espacio de nombres.