

Manual técnico

Proyecto: Editor de imágenes básico

Índice

Contenido

Componentes / Librerías Utilizados con sus respectivas versiones	2
Entorno de desarrollo	3
Archivos y Recursos Locales	3
Documentación de la interacción entre los componentes.....	4
Documentación del código del proyecto.....	5
HTML:.....	5
JavaScript:.....	6
Script.js:.....	7
Tour.js	14
Compress.js.....	17
FAQs.js	17

Componentes / Librerías Utilizados con sus respectivas versiones

Font Awesome

Versión: 6.1.1

URL: <https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.1.1/css/all.min.css>

Bootstrap

Versión: 5.3.3

URL: <https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css>

Cropper.js

Versión: 1.5.12

URL: <https://cdnjs.cloudflare.com/ajax/libs/cropperjs/1.5.12/cropper.min.css>

Hopscotch

URL (CSS): <https://unpkg.com/hopscotch/dist/css/hopscotch.min.css>

URL (JS): <https://unpkg.com/hopscotch/dist/js/hopscotch.min.js>

Compressor.js

Versión: 1.1.1

URL: <https://cdn.jsdelivr.net/npm/compressorjs@1.1.1/dist/compressor.min.js>

Browser Image Compression (este es una herramienta que utilizamos y después remplazamos pero dejamos comentado para futuros usos)

Versión: 1.0.16

URL: <https://cdn.jsdelivr.net/npm/browser-image-compression@1.0.16/dist/browser-image-compression.js>

Entorno de desarrollo

Visual Studio Code

versión: v1.58.0

GitHub web

Archivos y Recursos Locales

CSS Locales

../css/estilo.css

../css/style.css

../css/PerTour.css

JavaScript Locales

../JavaScript/Tour.js

../JavaScript/script.js

../JavaScript/compressor/compress.js

../JavaScript/FAQs.js

Otros Componentes

Imágenes:

../img/delete.png

../img/vertical-invert.png

../img/horizontal-invert.png

../img/rote izquierda.png

../img/rote derecha.png

Documentación de la interacción entre los componentes.

Esta parte de la documentación técnica se realizó mediante una plataforma llamada **Cacoo**, con el objetivo de representar de manera más fácil la estructuración del código, sus funciones, clases y como se instancian.

Adjuntamos enlace del diagrama: <https://cacoo.com/diagrams/49QmnzOJ0EohkR34/40C6F>

Documentación del código del proyecto

Para el proyecto se crearon varios documentos

HTML:

Index.html

Preguntas frecuentes.html

Explicación breve de cada una:

Archivo: index.html

Descripción: Este archivo define la estructura y comportamiento de una aplicación web para editar imágenes.

Recursos Externos Utilizados:

- Font Awesome v6.1.1
- Bootstrap v5.3.3
- Cropper.js v1.5.12
- Hopscotch v1.0.0
- Browser Image Compression v1.0.16
- Compressor.js v1.1.1
- Elementos Principales:

Tutorial Emergente:

Muestra un tutorial paso a paso para aprender a usar la página.

Botones: SI y NO para iniciar o cancelar el tutorial.

Alertas:

Alerta 1: Advierte sobre formatos de archivo aceptados.

Alerta 2: Notifica cuando la imagen es demasiado grande.

Alerta 3: Alerta sobre el uso de imágenes grandes que pueden causar errores.

Menú y Navegación:

Botones para subir, guardar, y descargar imágenes.

Opciones para guardar en galería y descargar con diferentes formatos.

Paneles de Imágenes:

Muestra la imagen original y la imagen editada.

Permite ajustar el nombre del archivo y seleccionar el formato de descarga.

Sidebar y Opciones:

Sidebar con opciones para galería, filtros y ajustes de imagen.

Botones activables para aplicar filtros y ajustes específicos.

Controles de Edición:

Controles deslizantes y botones para ajustar el brillo, contraste, opacidad, entre otros.

Botones para invertir, girar y recortar la imagen.

Scripts JavaScript:

Tour.js: Contiene el script para el tutorial paso a paso.

script.js: Otro script utilizado para funcionalidades adicionales en la página.

compress.js: Script para la compresión de imágenes.

JavaScript:

../JavaScript/Tour.js

../JavaScript/script.js

../JavaScript/compressor/compress.js

../JavaScript/FAQs.js

Script.js:

Variables Globales

Estas variables globales están referenciadas en múltiples funciones para manipular elementos del DOM y gestionar el estado de la aplicación:

- contenedor: Elemento contenedor principal.
- alert1Formatos: Contenedor de la primera alerta.
- alert2tamano: Contenedor de la segunda alerta.
- alert3tamano: Contenedor de la tercera alerta.
- botonMenu: Botón del menú.
- sidebar1: Primera barra lateral.
- sidebar2: Segunda barra lateral.
- sidebar1_Boton: Conjunto de botones de la primera barra lateral.
- opcionFiltro: Contenedor de opciones de filtros.
- opcionAjustes: Contenedor de opciones de ajustes.
- opcionGaleria: Contenedor de opciones de galería.
- btnAjuste_invertir: Botón para invertir ajuste.
- btnAjuste_girar: Botón para girar ajuste.
- btnAjuste_recortar: Botón para recortar ajuste.
- recortarContainer: Contenedor del canvas para recortar.
- containerControlPrincipal: Contenedor principal del control.
- controlPrincipal: Contenedor principal del control específico.
- controlPrincipal_ajuste: Contenedor de ajustes del control principal.
- controlSecundario: Contenedor secundario del control.
- imagen: Elemento de la imagen principal.
- imagenPrevio: Elemento de la imagen previa.
- imagenFinal: Elemento de la imagen final.
- botonImagen: Botón de selección de archivo de imagen.
- subirImagen: Botón para subir imagen.
- guardarImagen: Botón para guardar imagen.
- panelGuardarImagen: Panel de guardar imagen.
- descargarImagen: Botón para descargar imagen.
- cerrarDescarga: Botón para cerrar la descarga.
- botonAyuda: Botón de ayuda.
- sidebar3: Tercera barra lateral.
- opcionBoton: Conjunto de botones de la segunda barra lateral.
- nombreFiltro: Elemento de nombre del filtro.
- valorFiltro: Elemento de valor del filtro.

- range: Barra de rango.
- cDisable: Conjunto de elementos deshabilitados.
- cerrarAlert1: Botón para cerrar la primera alerta.
- cerrarAlert2: Botón para cerrar la segunda alerta.
- cerrarAlert3: Botón para cerrar la tercera alerta.
- selectFormatos: Selector de formatos.
- nombreArchivo: Elemento de nombre de archivo.
- atras: Botón para retroceder.
- adelante: Botón para avanzar.
- rotarOpcion: Conjunto de botones de ajuste de rotación.
- opcionesGaleriaClonado: Conjunto de opciones clonadas de galería.
- Funciones

MenuToggle(menuActivo)

- Descripción: Alterna el estado del menú y ajusta las clases CSS del contenedor y las barras laterales.
- Parámetros de entrada: menuActivo (boolean): Indica si el menú está activo o no.
- Datos de salida: Ninguno.
- Referenciada en: Evento click del botonMenu y función handleResize.

ControlToggle()

- Descripción: Alterna la visibilidad del control principal basado en el estado del filtro activo y la visibilidad de la barra lateral.
- Parámetros de entrada: Ninguno.
- Datos de salida: Ninguno.
- Referenciada en: MenuToggle.

botonMenu.addEventListener("click", ...)

- Descripción: Listener para el botón del menú que alterna el estado del menú y la visibilidad del control principal.
- Parámetros de entrada: Ninguno.
- Datos de salida: Ninguno.
- Referenciada en: Directamente en el script principal.

sidebar1_Boton.forEach(boton => { ... })

- Descripción: Listener para los botones de la primera barra lateral que alterna entre las opciones de filtros, ajustes y galería.
- Parámetros de entrada: Ninguno.
- Datos de salida: Ninguno.

- Referenciada en: Directamente en el script principal.

handleResize()

- Descripción: Listener para ajustar el template basado en el tamaño de la ventana.
- Parámetros de entrada: Ninguno.
- Datos de salida: Ninguno.
- Referenciada en: Evento DOMContentLoaded del documento.

ConfigInicio()

- Descripción: Configura el estado inicial de la aplicación, verifica cambios en la galería y ajusta clases CSS.
- Parámetros de entrada: Ninguno.
- Datos de salida: Ninguno.
- Referenciada en: No referenciada explícitamente en la primera parte del código proporcionado.

ConfigCargar()

- Descripción: Alterna el estado de carga de la imagen, mostrando u ocultando el contenedor de carga.
- Parámetros de entrada: Ninguno.
- Datos de salida: Ninguno.
- Referenciada en: No referenciada explícitamente en la primera parte del código proporcionado.

Verificar

- Descripción: No está completa en la parte proporcionada. Posiblemente verifica ciertos estados o configuraciones.
- Parámetros de entrada: No especificados.
- Datos de salida: No especificados.
- Referenciada en: No referenciada explícitamente en la primera parte del código proporcionado.

Función OpcionAjustes

- Descripción: Esta función activa opciones de ajuste como recortar, invertir y girar una imagen.
- Entradas: idAjuste (string): Identificador de la opción de ajuste seleccionada.
- Salidas: Modifica la visualización de elementos en la interfaz según la opción seleccionada.

- Referencias: Llama a la función Recortar cuando se selecciona la opción "recortar".

Función GuardarFiltro

- Descripción: Guarda el valor del rango de un filtro aplicado a una imagen.
- Entradas: Ninguna explícita (usa variables globales range, valorFiltro, opcionElegido, filtroAjusteArray, arrayPost).
- Salidas: Actualiza valorFiltro en la interfaz.
- Actualiza filtroAjusteArray y arrayPost con los datos del filtro aplicado.
- Referencias: Llama a la función AplicarFiltro().

Función AbrirPanelGuardar

- Descripción: Abre o cierra el panel para guardar una imagen.
- Entradas: Ninguna explícita.
- Salidas: Modifica la visualización del panel de guardar imagen y muestra la imagen actual.
- Referencias: Utiliza elementos como panelGuardarImagen, imagenFinal, imagenPrevio.

Función ObtenerValor

- Descripción: Obtiene un valor numérico a partir de una cadena.
- Entradas: cadena (objeto): Objeto del cual obtener el valor.
- posicion (opcional, número): Posición inicial para la búsqueda.
- Salidas: nuevoNumero (número): Valor numérico obtenido.
- Referencias: Ninguna explícita.

Función AppendGaleria

- Descripción: Añade imágenes a una galería en la interfaz.
- Entradas: Ninguna explícita.
- Salidas: Agrega elementos HTML para cada imagen en la galería.
- Aplica filtros y ajustes a las imágenes.
- Referencias: Usa variables globales como galeria, opcionGaleria.

Función VerificarCambioGaleria

- Descripción: Verifica si ha habido cambios en la imagen de la galería.
- Entradas: condicion (booleano): Condición para la verificación.
- Salidas: Muestra un mensaje de confirmación si hay cambios pendientes de guardar.
- Referencias: Utiliza variables y elementos de la interfaz como editado, filtroAjusteArray.

Función AgregarImgGaleria

- Descripción: Agrega una imagen a la galería.
- Entradas: guardarGaleria (booleano): Indica si se debe guardar la imagen en la galería.
- nombre (opcional, string): Nombre de la imagen.
- Salidas: Guarda la imagen en el objeto galeria.
- Actualiza la visualización de la galería en la interfaz.
- Referencias: Usa variables globales como galeria, llaveGaleria.

Función BorrarGaleria

- Descripción: Elimina una imagen de la galería.
- Entradas: Ninguna explícita.
- Salidas: Elimina el elemento HTML correspondiente a la imagen de la galería.
- Actualiza el objeto galeria.
- Referencias: Utiliza variables globales como galeria.

Función AgregarListenerGaleria

- Descripción: Agrega un listener a las imágenes de la galería para manejar eventos.
- Entradas: opcionClonado (elemento HTML): Elemento clonado de una imagen de la galería.
- Salidas: Asigna eventos al clic en las imágenes de la galería.
- Referencias: Llama a funciones como VerificarCambioGaleria.

Función ConfigCanva

- Descripción: Configura el canvas para una imagen con filtros y transformaciones.
- Entradas: image, canvas, ctx, transform, filtro, size: Varios parámetros relacionados con la configuración del canvas.
- Salidas: Ninguna explícita.
- Referencias: Utiliza variables globales como rotate, filtroAjusteArray.

Función Recortar

- Descripción: Inicia el proceso de recorte de una imagen.
- Entradas: trans, filter: Parámetros para configurar el canvas de recorte.
- Salidas: Actualiza la visualización del canvas de recorte.
- Referencias: Llama a funciones como ConfigCanva y CreateCropper.

Función CalculateCropValues

- Descripción: Calcula los valores de recorte para una imagen.
- Entradas: Ninguna explícita.
- Salidas: Devuelve valores calculados para el recorte de la imagen.
- Referencias: Utiliza variables globales como rotate, invertirHorizontal, invertirVertical.

Función crop

- Descripción: Aplica el recorte a la imagen activa.
- Entradas: Ninguna explícita.
- Salidas: Aplica el recorte mediante CSS a la imagen activa.
- Referencias: Utiliza funciones como CalculateCropValues y variables globales como filtroAjusteArray, arrayPrev.

Event Listeners (addEventListener)

- Descripción: Maneja eventos de clic en botones de la interfaz.
- Entradas: Ninguna explícita.
- Salidas: Asigna funciones a los eventos de clic en botones.
- Referencias: Funciones como crop, VerificarCambioGaleria.

Función CreateCropper(canvas)

- Descripción: Crea un objeto Cropper para recortar una imagen.
- Datos de Entrada: canvas (Elemento HTMLCanvasElement) - El lienzo HTML en el cual se realizará el recorte.
- Datos de Salida: Ninguno explícito.
- Referencias: Utilizada al inicializar el recorte de la imagen en el canvas.

Evento resize

- Descripción: Maneja el evento de redimensionamiento de la ventana del navegador.
- Datos de Entrada: Ninguno explícito.
- Datos de Salida: Ninguno explícito.
- Referencias: Llama a las funciones Recortar(true, true) y handleResize() después de un retraso para ajustar la imagen al tamaño de la ventana.

Función Adelante()

- Descripción: Avanza a través del historial de cambios de la imagen.
- Datos de Entrada: Ninguno explícito.
- Datos de Salida: Objeto {llave, valor1} - Clave y valor del cambio aplicado.
- Referencias: Utilizada al manejar el historial de cambios hacia adelante.

Función Retroceder()

- Descripción: Retrocede a través del historial de cambios de la imagen.
- Datos de Entrada: Ninguno explícito.
- Datos de Salida: Objeto {llave, valor1} - Clave y valor del cambio revertido.
- Referencias: Utilizada al manejar el historial de cambios hacia atrás.

Función AplicarDeshacerRehacer(arrayCambio)

- Descripción: Aplica un cambio específico a la imagen.
- Datos de Entrada: arrayCambio (Objeto) - Objeto que describe el cambio a aplicar.
- Datos de Salida: Ninguno explícito.
- Referencias: Aplica cambios como recortes, filtros y ajustes a la imagen según el objeto arrayCambio.

Función ResetAll()

- Descripción: Restablece todos los ajustes y cambios realizados en la imagen.
- Datos de Entrada: Ninguno explícito.
- Datos de Salida: Ninguno explícito.
- Referencias: Utilizada para restaurar la imagen a su estado inicial.

Evento change en range

- Descripción: Maneja el cambio de valor en un control deslizante.
- Datos de Entrada: Ninguno explícito.
- Datos de Salida: Ninguno explícito.
- Referencias: Actualiza el historial de cambios de filtros aplicados a la imagen.

Función ObtenerDatos(savedData)

- Descripción: Carga datos guardados para restaurar el estado previo de la aplicación.
- Datos de Entrada: savedData (Objeto) - Datos guardados de la aplicación.
- Datos de Salida: Ninguno explícito.
- Referencias: Utilizada al cargar datos guardados del almacenamiento local.

Función toggleAyudaMenu()

- Descripción: Muestra u oculta el menú de ayuda.
- Datos de Entrada: Ninguno explícito.
- Datos de Salida: Ninguno explícito.
- Referencias: Utilizada para interactuar con el menú de ayuda en la interfaz de usuario.

Tour.js

handleInitialLoad

- Descripción: Función para manejar la carga inicial de la página.
- Datos de entrada: Ninguno.
- Datos de salida: Ninguno.
- Referencias: Se activa cuando se carga el evento DOMContentLoaded.

Event Listeners:

- Event Listener para cerrar ventana emergente:
- Descripción: Cierra la ventana emergente al hacer clic en el botón de cerrar.
- Datos de entrada: Evento de clic (click).
- Datos de salida: Ninguno.
- Referencias: Elementos con id cerrar y vent button#cerrar.

Event Listener para iniciar el tour al hacer clic en "Sí":

- Descripción: Inicia el tour cuando se hace clic en el botón "Sí" dentro de la ventana emergente.
- Datos de entrada: Evento de clic (click).
- Datos de salida: Ninguno.
- Referencias: Elemento con id seguir.

Event Listener para mostrar la ventana emergente de tutorial:

- Descripción: Muestra la ventana emergente de tutorial al hacer clic en el enlace "Guia-Web".
- Datos de entrada: Evento de clic (click).
- Datos de salida: Ninguno.
- Referencias: Elemento con id Guia-Web.

Event Listener para cerrar el overlay:

- Descripción: Cierra el overlay al hacer clic en el botón de cancelar tour.
- Datos de entrada: Evento de clic (click).
- Datos de salida: Ninguno.

- Referencias: Elemento con id cancelar-tour.

Event Listener para verificar si se ha subido una imagen:

- Descripción: Verifica si se ha clicado el botón para subir una imagen y muestra el siguiente paso del tour.
- Datos de entrada: Evento de clic (click).
- Datos de salida: Ninguno.
- Referencias: Elemento con id subir-boton.

Event Listener para verificar si se ha clicado el botón de filtro:

- Descripción: Verifica si se ha clicado el botón de filtro y muestra el siguiente paso del tour.
- Datos de entrada: Evento de clic (click).
- Datos de salida: Ninguno.
- Referencias: Elemento con id filtro.

Event Listener para verificar si se ha clicado el botón de ajustes:

- Descripción: Verifica si se ha clicado el botón de ajustes y muestra el siguiente paso del tour.
- Datos de entrada: Evento de clic (click).
- Datos de salida: Ninguno.
- Referencias: Elemento con id ajuste.

Variables y Constantes:

tour: Objeto que define los pasos del recorrido guiado (tour).

- Descripción: Define los pasos y comportamientos del tour guiado en la página.
- Datos de entrada: Ninguno.
- Datos de salida: Ninguno.
- Referencias: Se utiliza para iniciar el recorrido con hopscotch.startTour.

Funciones Adicionales:

checkSubidaImagen

- Descripción: Verifica si se ha subido una imagen antes de continuar con el tour y muestra el siguiente paso si es así.

- Datos de entrada: Ninguno explícito.
- Datos de salida: Ninguno explícito.
- Referencias: Utilizada en varios listeners para validar la subida de imágenes.

checkfiltro

- Descripción: Verifica si se ha clicado el botón de filtro antes de continuar con el tour y muestra el siguiente paso si es así.
- Datos de entrada: Ninguno explícito.
- Datos de salida: Ninguno explícito.
- Referencias: Utilizada en varios listeners para validar el uso del filtro.

checkajuste

- Descripción: Verifica si se ha clicado el botón de ajustes antes de continuar con el tour y muestra el siguiente paso si es así.
- Datos de entrada: Ninguno explícito.
- Datos de salida: Ninguno explícito.
- Referencias: Utilizada en varios listeners para validar el uso de ajustes.

cerrarOverlay

- Descripción: Oculta el overlay al cerrar el tour o al hacer clic en el botón de cancelar.
- Datos de entrada: Ninguno.
- Datos de salida: Ninguno.
- Referencias: Utilizada en el evento onClose del objeto tour.

Compress.js

Función compressImage(file):

Esta es una función asíncrona (async function) que acepta un parámetro file, que se espera que sea un objeto File que representa el archivo de imagen a comprimir.

Promesa y constructor de Compressor: La función compressImage retorna una promesa (Promise) que envuelve la creación de un nuevo objeto Compressor.

El objeto Compressor se crea con dos argumentos principales:

file: El archivo de imagen que se va a comprimir.

Un objeto de configuración que incluye:

quality: Configurado en 1, lo que indica máxima calidad de compresión.

maxWidth y maxHeight: Ambos configurados en 1920, limitando las dimensiones máximas de la imagen resultante a 1920 píxeles en ancho y alto respectivamente.

Manejo de eventos success y error:

success: Este evento se dispara cuando la compresión es exitosa. Imprime el tamaño original y el tamaño comprimido de la imagen en kilobytes (KB) utilizando console.log, luego resuelve la promesa con el archivo de imagen comprimido (result).

error: Este evento se dispara si hay algún error durante el proceso de compresión. Imprime el error en la consola utilizando console.error, muestra una alerta al usuario indicando que no se pudo comprimir la imagen y rechaza la promesa con el error (err).

FAQs.js

Es un buscador de preguntas (FAQ) que permite filtrar y paginar preguntas almacenadas en un archivo JSON llamado question.json.

Variables Globales

- data: Almacena los datos cargados desde el archivo JSON que contiene las preguntas.
- preguntasPorPagina: Define cuántas preguntas se mostrarán por página.
- paginaActual: Almacena el número de la página actualmente mostrada.
- paginaOriginal: Variable auxiliar para almacenar la página actual antes de realizar una búsqueda, utilizada para restaurar la paginación después de una búsqueda.

Funciones

agregarPregunta(pregunta, lista, numero):

- Datos de Entrada: pregunta (objeto), lista (elemento HTML donde se agregarán las preguntas), numero (número de la pregunta).
- Datos de Salida: No hay salida explícita.
- Referenciada en: Llamada dentro de mostrarPreguntas.
- Descripción: Crea dinámicamente un elemento section que contiene el título de la pregunta, un botón para mostrar la respuesta y la respuesta en sí misma. Configura eventos para expandir o contraer la respuesta al hacer clic en el botón.

mostrarPreguntas(pagina):

- Datos de Entrada: pagina (número de página a mostrar).
- Datos de Salida: No hay salida explícita.
- Referenciada en: Llamada desde cambiarPagina y al inicio del script.
- Descripción: Muestra las preguntas correspondientes a la página actual pagina en el elemento HTML faq-list, utilizando agregarPregunta para cada pregunta mostrada. Actualiza la paginación llamando a actualizarPaginacion.

cambiarPagina(pagina):

- Datos de Entrada: pagina (número de página a cambiar).
- Datos de Salida: No hay salida explícita.
- Referenciada en: Llamada desde el evento de clic en los botones de paginación.
- Descripción: Actualiza la variable paginaActual y llama a mostrarPreguntas con la nueva página.

normalizarCadena(cadena):

- Datos de Entrada: cadena (cadena de texto a normalizar).
- Datos de Salida: Cadena de texto normalizada sin acentos y en minúsculas.
- Referenciada en: Llamada dentro de buscarPreguntas.
- Descripción: Normaliza la cadena de texto eliminando acentos y convirtiéndola a minúsculas.

buscarPreguntas(query):

- Datos de Entrada: query (cadena de búsqueda introducida por el usuario).
- Datos de Salida: No hay salida explícita.

- Referenciada en: Llamada desde el evento de entrada (input) en el campo de búsqueda.
- Descripción: Filtra las preguntas basadas en la query ingresada, utilizando expresiones regulares para una búsqueda insensible a mayúsculas y acentos. Muestra las preguntas filtradas utilizando agregarPregunta. Gestiona la paginación y oculta/muestra la paginación según corresponda.

actualizarPaginacion(pagina):

- Datos de Entrada: pagina (número de página actual).
- Datos de Salida: No hay salida explícita.
- Referenciada en: Llamada dentro de mostrarPreguntas.
- Descripción: Actualiza la visualización de los botones de paginación basados en el número total de páginas y marca la página actual con la clase active.

Carga de Datos y Eventos

- Carga de Datos: Utiliza fetch para cargar datos desde un archivo JSON (question.json) y llama a mostrarPreguntas para inicializar la página con las preguntas cargadas.
- Evento de Búsqueda: Escucha el evento input en el campo de búsqueda (search-bar). Al escribir en el campo, llama a buscarPreguntas para filtrar y mostrar las preguntas correspondientes.