**Paso 1:** Escribimos en Google Visual Studio Code y seleccionamos donde dice “Download”.



**Paso 2:** Seleccionamos el sistema operativo que tenemos y lo descargamos.



**Paso 3:** Al darle clic nos descargará un .exe, al cual le daremos clic encima.



**Paso 4:** Lee y acepta el acuerdo de licencia. Haz clic en Next para continuar.



**Paso 5:** Puedes cambiar la ubicación de la carpeta de instalación o mantener la configuración predeterminada. Haz clic en Next para continuar.



**Paso 6:** Elige si deseas cambiar el nombre de la carpeta de accesos directos en el menú Inicio o si no deseas instalar accesos directos en absoluto. Haz clic en Next.



**Paso 7:** Selecciona las tareas adicionales, por ej. crear un icono en el escritorio o añadir opciones al menú contextual de Windows Explorer. Haz clic en Next.Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Paso 8:** Haz clic en Install para iniciar la instalación.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Paso 9:** El programa está instalado y listo para usar. Haz clic en Finish para finalizar la instalación y lanzar el programa.



**Creación de carpetas y archivos**

**Paso 1:**Creamos una carpeta raíz. En nuestro caso la llamaremos “Contador”.



**Paso 2:**Dentro de esta, crearemos tres carpetas. Uno llamado css: Donde irán los archivos que se encargaran del diseño de la pagina; el segundo llamado js: Donde irán los archivos que contengan toda la lógica e interacción del aplicativo, y el tercero llamado img que es donde meteremos imágenes

Texto

Descripción generada automáticamente

**Paso 3**:En el buscador de archivos escribimos “cmd” y se nos abrirá la consola.

Tabla

Descripción generada automáticamente con confianza media

**Paso 4:**Escribimos en la consola “code .”.

Interfaz de usuario gráfica, Texto

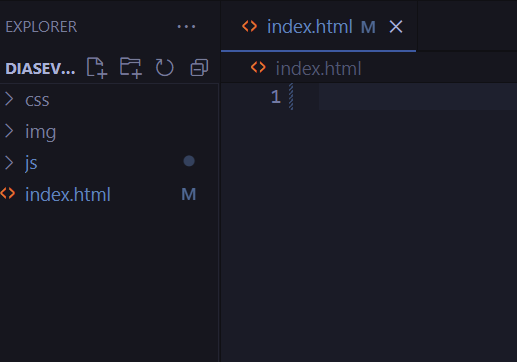
Descripción generada automáticamente

**Paso 5:** Se abre la carpeta en el visual estudio code

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Paso 6:**Creamos un nuevo archivo en la carpeta raíz llamado “index.html” y en este escribimos “html:5”



**Paso 7:**Esto habrá creado el código base de un archivo html en el cual podremos empezar a trabajaCaptura de pantalla de computadora

Descripción generada automáticamente

**Paso 8:** creamos un archivo style.css en la carpeta css.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Paso 9:** enlazamos el archivo index.html con el style.css

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

**Paso 10:** creamos un archivo en la carpeta js llamado app.js

Interfaz de usuario gráfica, Aplicación

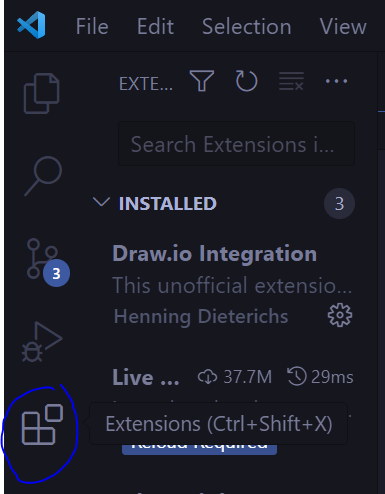
Descripción generada automáticamente

**Paso 11:** enlazamos el archivo index.html con el app.js

Texto

Descripción generada automáticamente con confianza media

**Paso 12:** en el lado derecho encontraremos un apartado de extensiones en el visual studio



**Paso 13:** en las extensiones buscamos “live server” click en la primera opción y los descargamos

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Paso 14:** en el html crearemos un form que contenga un input de tipo texto donde ingresaremos el nombre del evento, un input tipo fecha para agregar la fecha y un botón el cual guarda la información y al final un div con el id taskContainer que será el que contenga la lista de fechas que hemos agregado



Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

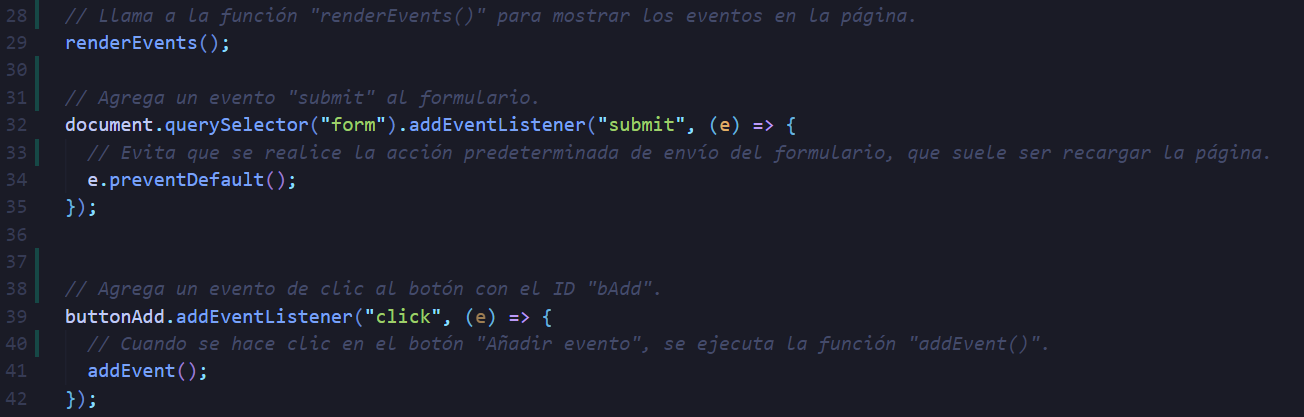
Descripción generada automáticamente

**Paso 15:** en el archivo js se inicializa una variable array (línea 1), se hace referencia a elementos con ids específicos con el querySelector (línea 5 a la 11), Se intenta cargar eventos previamente guardados desde el almacenamiento local utilizando la función load() (línea 14),Los eventos cargados se convierten de JSON a un array y se asignan a events(línea 20).

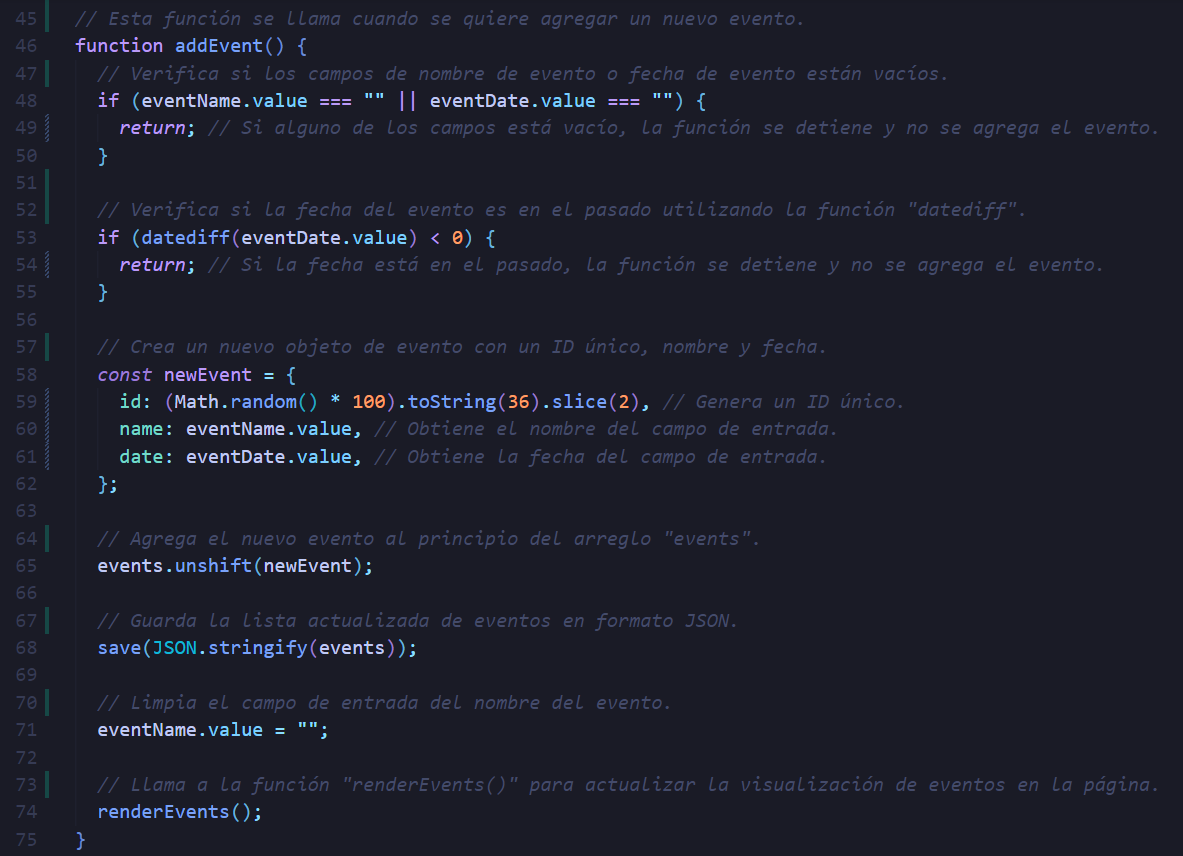
**Texto

Descripción generada automáticamente**

**Pas 16:** Se le agrega un listener al formulario para que al hacer submit la función flecha evite que se recargue la página (línea 31 al 34), En el botón de agregar al hacer click de llama a la función addEvent()(línea 39 a 41);



**Paso 17:** La función addEvent() es llamada cuando el botón "Agregar" se hace click. Esta comprueba si los campos de nombre y fecha no están vacíos (línea 48), verifica si la fecha del evento es futura(línea 53). Luego, crea un nuevo objeto (línea 58) de evento con un id aleatorio, el nombre y la fecha proporcionados. Agrega el nuevo evento al principio del array events(línea 65). Llama a la función save para almacenar el array events en el almacenamiento local como una cadena JSON (línea 68). Limpia el campo de entrada de nombre y finalmente, llama a renderEvents() para actualizar la visualización de los eventos en la página (línea 71 y 74).

****

**Paso 18:** La función renderEvents() se encarga de generar el HTML para cada evento en el array events (línea 80). Crea elementos HTML para cada evento, incluyendo el nombre, la fecha y un botón de eliminación (línea 83 a 100). Los elementos HTML del evento se concatenan y se insertan en el elemento tasksContainer (línea 105). Con el querySelectorAll tare a todos los botones con el id bDelete en un array y se recorre ese array con un ForEach (línea 108), se asigna un listener de eventos de clic a los botones de eliminación para permitir la eliminación de eventos(línea 109), obtiene el id del evento que desea eliminar para filtrar y elimiarlo, guarda los cambios y actualiza (línea 11 a 117).

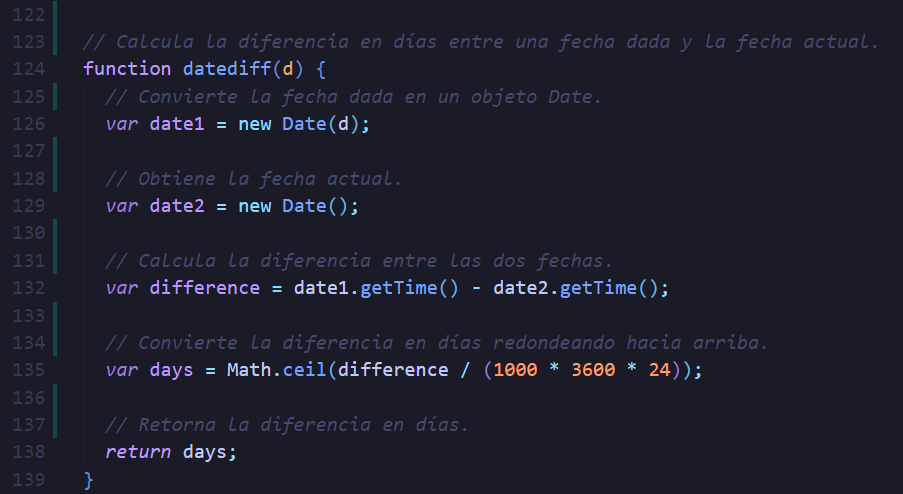
Texto

Descripción generada automáticamente

**Texto

Descripción generada automáticamente**

**Paso 19:** Esta función calcula la cantidad de días entre la fecha proporcionada y la fecha actual. Convierte las fechas a marcas de tiempo y calcula la diferencia en milisegundos, luego la convierte en días.



**Paso 20:** save guarda una representación serializada de events en el almacenamiento local bajo la clave "items". load obtiene los datos serializados del almacenamiento local.

**Captura de pantalla de un celular

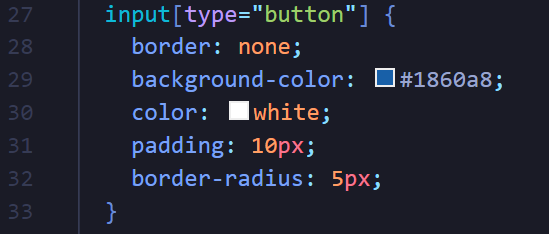
Descripción generada automáticamente**

**Paso 22:** modificamos el fondo, ancho, largo, margen, fuente de la letra y color.****

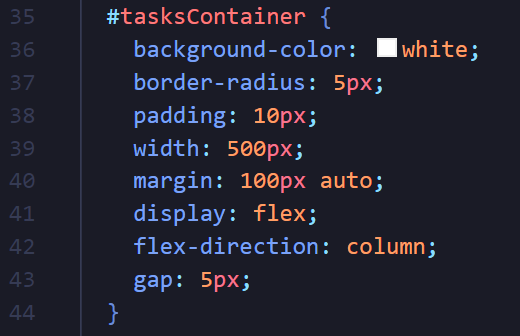
**Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente**

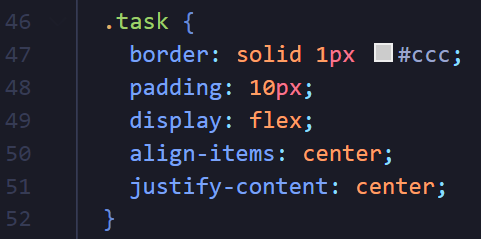
**Paso 23:** Aplica estilos específicos a los botones de tipo "button". Elimina el borde. Define un fondo azul , relleno de 10px, texto en blanco y bordes redondeados con un radio de 5px.

****

**Paso 24:** Establece un fondo blanco, bordes redondeados con un radio de 5px y un relleno de 10px para el elemento con el ID "tasksContainer".Define un ancho máximo de 500px, margen automático para centrar el contenedor horizontalmente.Utiliza un diseño de columna flexible para los elementos internos con un espacio vertical de 5px .

****

**Paso 25:** Define un borde sólido de 1px con un color gris claro alrededor de cada tarea. Establece un relleno de 10px. Utiliza un diseño de fila flexible para los elementos internos. Centra vertical y horizontalmente el contenido con



**Paso 26:** Define estilos específicos para diferentes elementos dentro de cada tarea. Ajusta el texto al centro y fija un ancho de 100px. Establece un tamaño de fuente grande y negrita. Definen anchos específicos para estos elementos. Aplica un relleno de 10px.

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

**Interfaz de usuario gráfica

Descripción generada automáticamente**