**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 .”.

Texto

Descripción generada automáticamente

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

Captura de pantalla de un celular

Descripción generada automáticamente

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

Interfaz de usuario gráfica, Texto, Aplicación, Teams

Descripción generada automáticamente

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

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, Texto, Aplicación

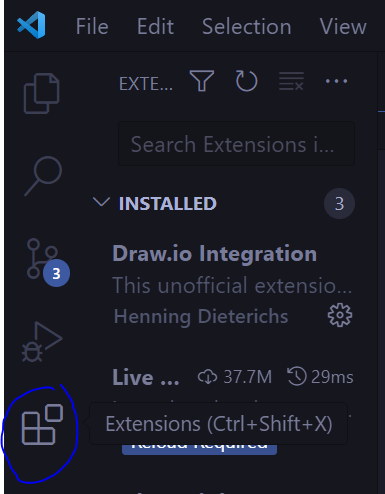
Descripción generada automáticamente

**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:** ahora en el body del index crearemos unos divs

Texto

Descripción generada automáticamente

**Paso 15:** creamos un div que contendrá el campo donde pondremos la tarea a realizar y el botón

Texto

Descripción generada automáticamente

**Paso 16:** en estos divs se mostrarán la tarea que se esta realizando y así debe ir quedando.

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

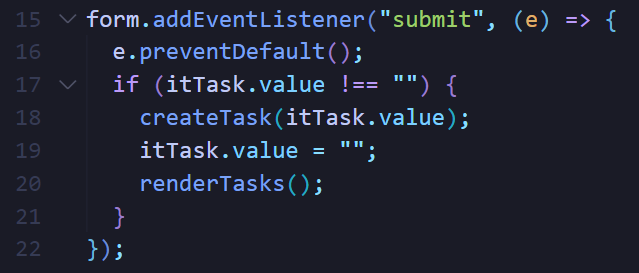
**Paso 17**: inicializamos variables

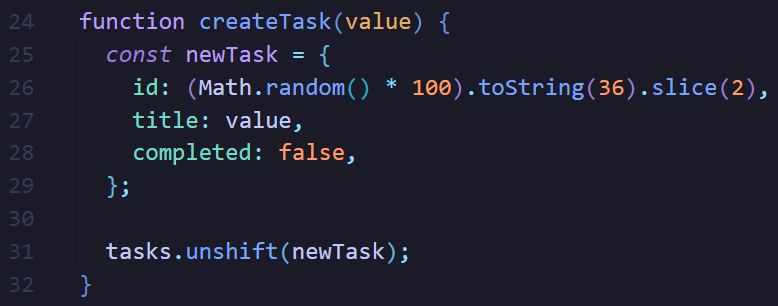


Texto

Descripción generada automáticamente

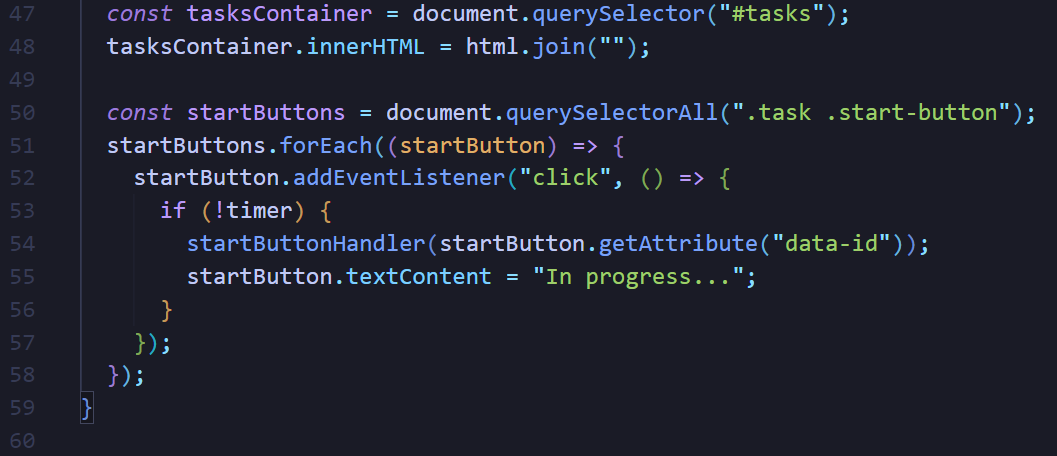
**Paso 18:** creamos un evento el cual se activa al hacer submit en el formó y evalúa si itTask tiene escrita una tarea de ser así la función createTask toma el nombre de la tarea y el valor de itTask se vuelve una cadena vacía y se ejecuta la función renderTasks que aún no se crea.

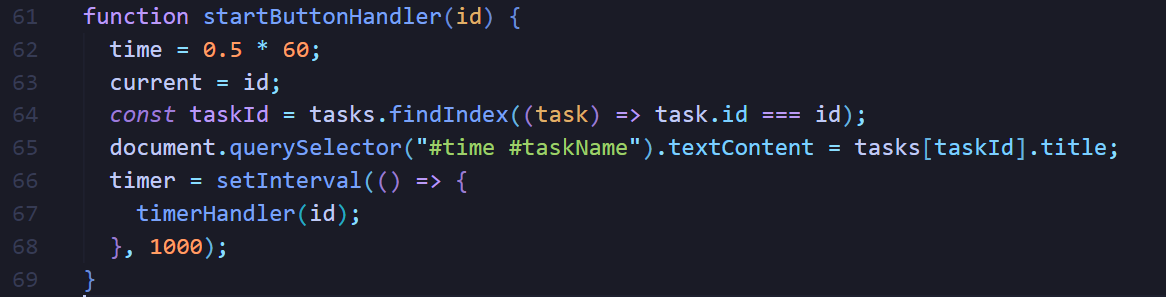


**Paso 19:** ahora creamos la función createTask la cual toma un parámetro value y crea un objeto y con el mata random que nos da un numero aleatorio creamos los id, a la variable tasks le agregamos el nuevo objeto que creamos****

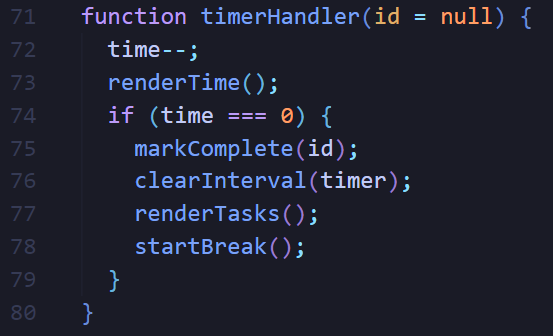
**Paso 20:** creamos la función renderTask la cual tiene como objetivo crear un fragmento de código de html al crear la tarea a realizar luego evaluamos si la tarea fue realizada para crear un span que diga done (hecho) o si no fue realizada que comience ****

**Paso 21**: creamos una variable la cual tendrá como valor un array con la clase task y start-button esto gracias al querySelectorAll leemos el array con un for each y mediante un evento usamos un función y con un text content establecemos que la tarea esta en progreso



**Paso 22:** primero se establece un tiempo inicial, con el current establecemos un id para la tarea actual, en taskid se busca el id de la tarea, actualizamos el contenido del elemento con el id dentro del elemento con el id

**Paso 23:** le restamos al tiempo y en el render time evaluamos si el contador es 0 lo que indica que la tarea fue completada lo marca como tal limpia el timer y inicia el contador del descanso

****

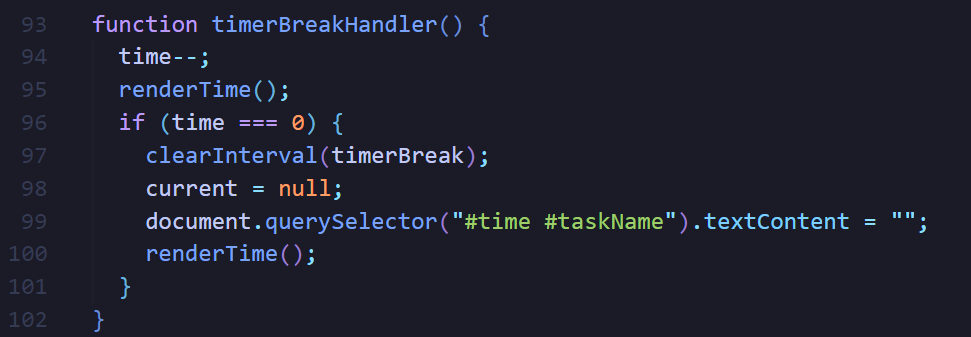
**Paso 24:**se busca el id de la tarea y se marca como completada **Texto

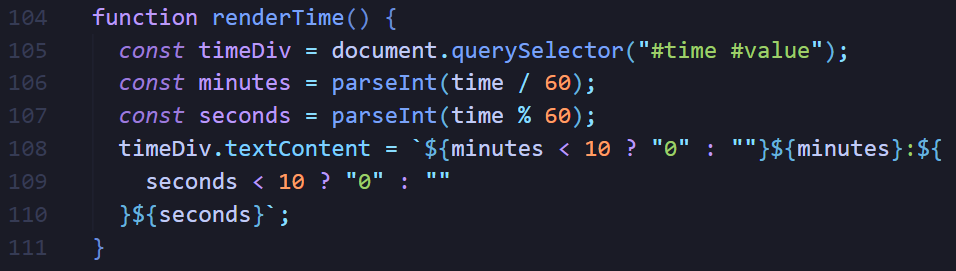
Descripción generada automáticamente**

**Paso 25:** se establece el tiempo del descanso y se rompe el time y el taskName y se establece setInterval que espera realizar funciones cada cierto tiempoTexto

Descripción generada automáticamente

**Paso 26:** le restamos al tiempo y en el render time evaluamos si el contador es 0 lo que indica que el descanso fue completada lo marca como tal limpia el timer



**Paso 27:** inicializamos variables, obtiene los minutos dividiendo entre 60 y obtiene los segundos encontrando el modulo de 60 actualiza el elemento con el tiempo formateado en formato mm:ss

**Paso 28:** se cambia el tipo de fuente y el fondo en todo el body, y se hacen cambios en el tamaño, borde, margen y padding en las clases y ids**Captura de pantalla de computadora

Descripción generada automáticamente**

**Interfaz de usuario gráfica

Descripción generada automáticamente**

**Paso 29**: se hacen cambios en el tamaño, borde, margen y padding en las clases Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

**Paso 30**: se hacen cambios en el tamaño, borde, margen y padding en las clases y ids

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

**Paso 31**: ya tendríamos nuestra app pomodoro

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente