**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 “Teclado”.



**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

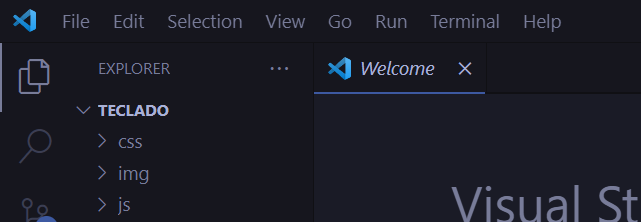
Descripción generada automáticamente con confianza media

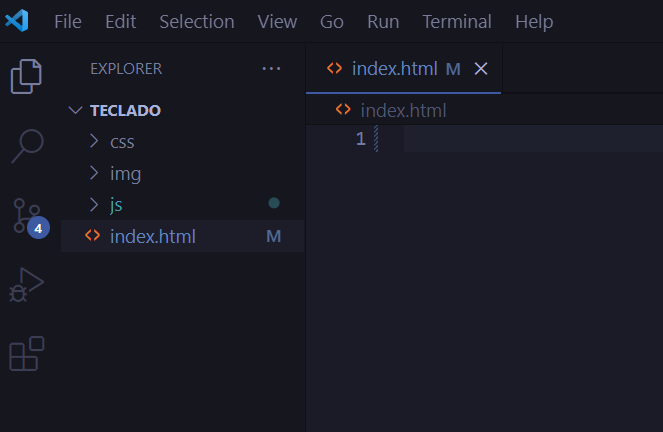
**Paso 4:**Escribimos en la consola “code .”.

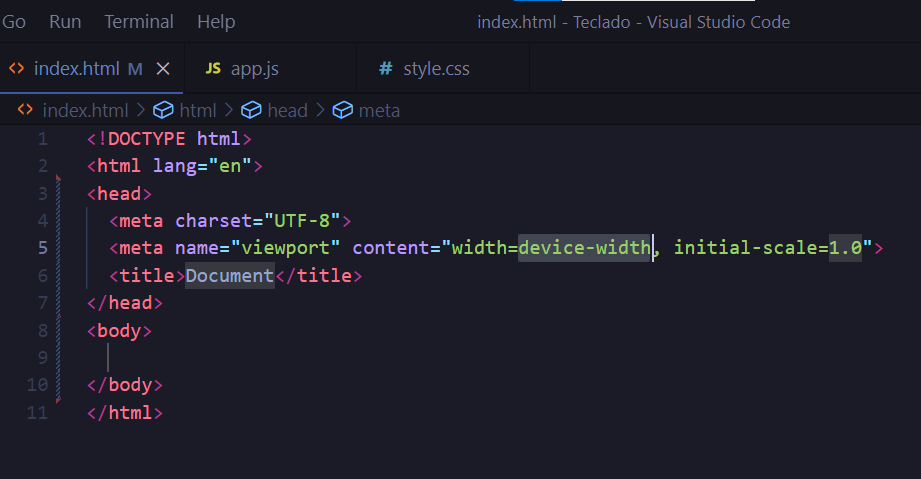
Texto

Descripción generada automáticamente

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



**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 trabaja 

**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

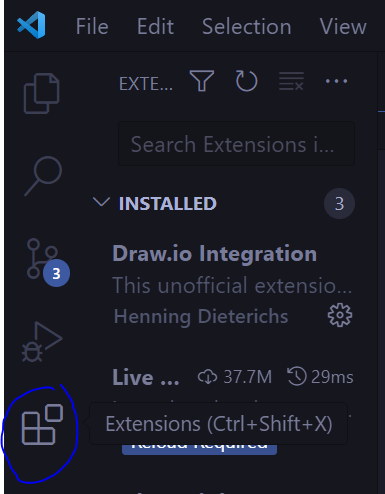
Descripción generada automáticamente

**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



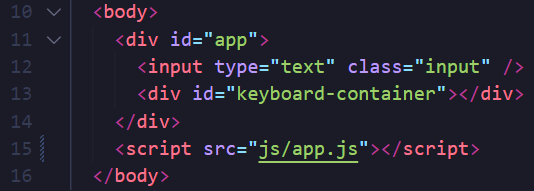
**Paso 13:** en las extensiones buscamos “live server” click en la primera opción y los descargamos

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Estructura HTML**

**Paso 14**: creamos un div con el id app el cual contendrá un input que será el que contendrá el texto y un div que contendrá el teclado.



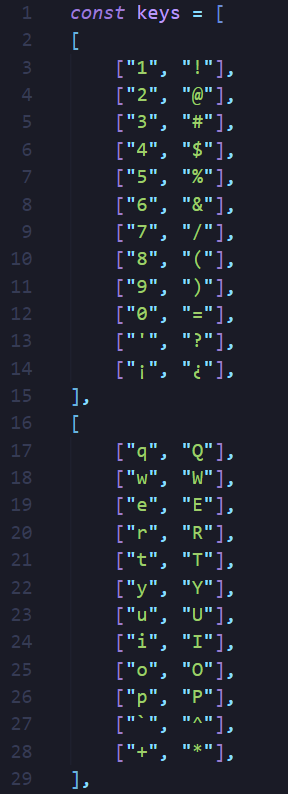
Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

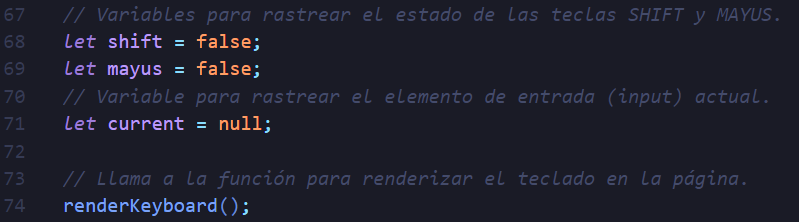
**Estructura JS**

**Paso 15**: En el archivo jsdefine el arreglo keys que representa la disposición de las teclas del teclado virtual. Cada elemento de este arreglo es un arreglo que contiene una pareja de caracteres. El primer carácter es el carácter normal, y el segundo es el carácter que se obtiene al presionar Shift. Cada subarreglo representa una fila del teclado.La última fila contiene una sola tecla ,la barra espaciadora.

**Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media**

**Paso 16:** inicializamos variables y hacemos el llamado de una funcion renderKeyboard()



**Paso 17:** creamos la funcion renderKeyboard() encargada de mostrar el teclado dentro de esta ira todo el codigo de la aplicacion.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

**Paso** **18**:Dentro de la funcion renderKeyboard inicializamos una constante para el contendor del teclado(linea 78) y una variable la cual crea un epacio vacio en el teclado (linea 81)

Texto

Descripción generada automáticamente

**Paso** **19:**ahora inicializamos la constante layers la cual recorre el array con el metodo “map” y una funcion flecha dado a que es un array tridimensional hay que usar dos map para recorre las filas y las columnas(linea 85 y 87)

Texto

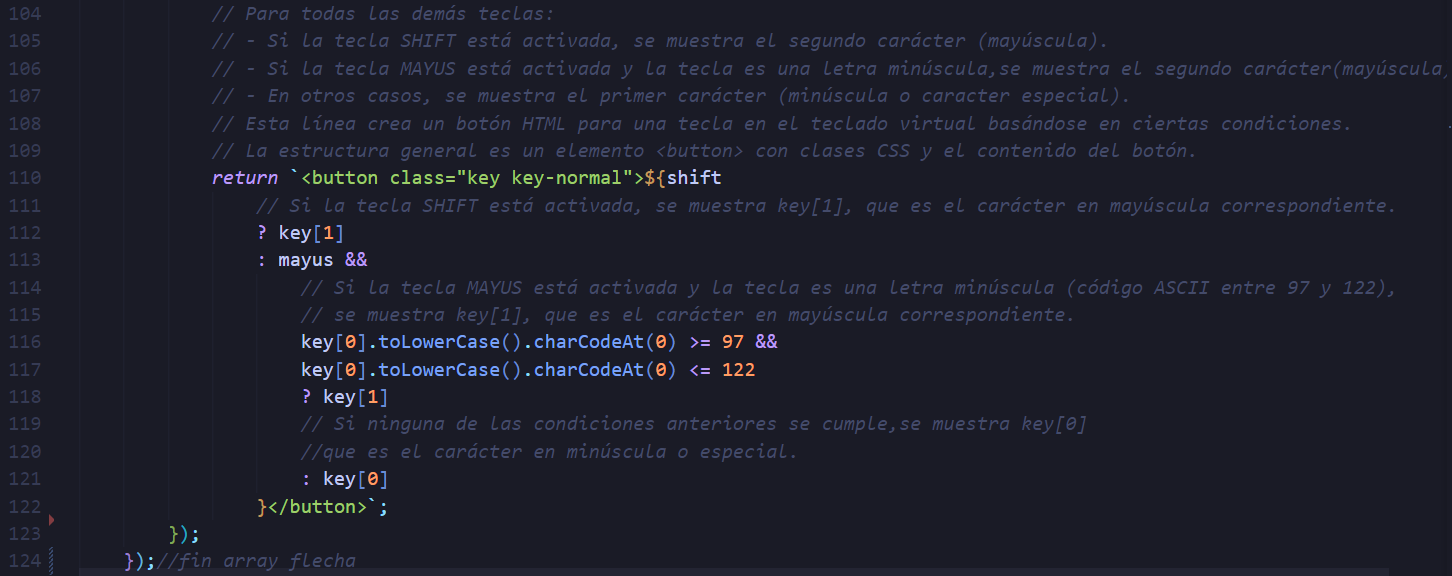
Descripción generada automáticamente

**Paso** **20:**con los condicinales validaremos si el shift, mayus y el espacio estan activados (linea 89, 94 y 100)

Texto

Descripción generada automáticamente

**Paso** **21**:se evalua con el operador ternario si el shift esta activado o si el mayus esta activado (linea 112 a la 121)



**Paso 22:** con el metodo push (linea 128) agregamos uno o mas elemento al final de un array en cambio con el metodo unshift (linea 130)agregamos uno o mas elemento al comienzo del array, en este caso agreamos a empty dos elementos el de la posicion 0 al final y el de la pocision 1 al comienzo.

Texto

Descripción generada automáticamente

**Paso 23:** la variable htmlLayers mapea el array layers y lo conecta en una sola cadena de caracteres (linea 139 a la 145)

Texto

Descripción generada automáticamente

**Paso 24:**el keyboardContainer se asegura que el contenedor este limpio borrando lo que hay dentro, se crea un forEach que itera sobre htmlLayers para asi agregar varios div que contenga la clase layer los cuales contendran las teclas del teclado.

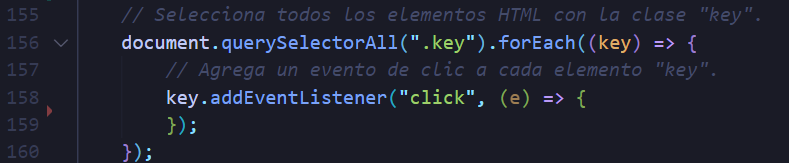
Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Paso 25:** con un querySelectorAll selecciona todos los elementos HTML con la clase "key" dado que este método trae a todos los elementos en forma de array se recorre con un forEach y se abre una función flecha, a key se le añade un event listener con el evento "click" (línea 156 y 158).



**Paso** **26**:dentro del listener se abre un condicional que evalúa el elemento actual

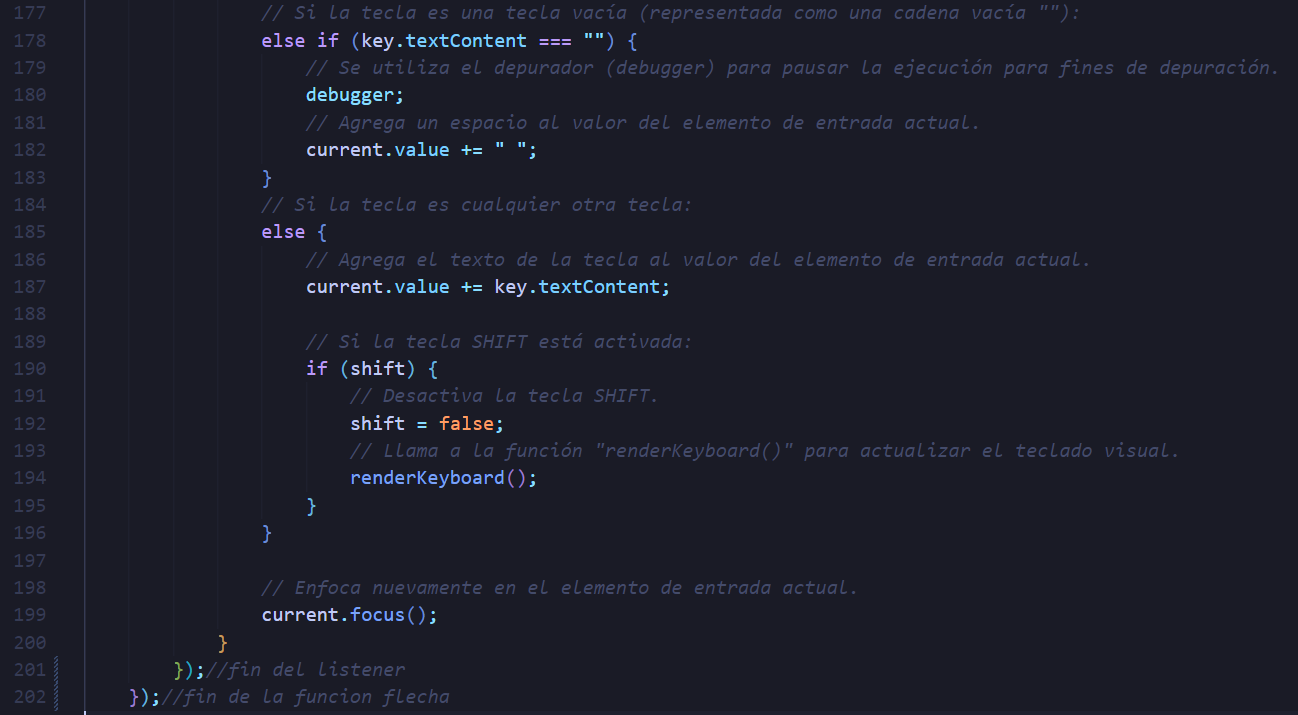
Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

**Paso** **27**:se evalúa el contenido de key si este es igual a shift o mayus se cambia el estado de las variables (línea 164 a 171)

Texto

Descripción generada automáticamente

**Paso 28:**si es esta vacio es porque representa el espacio por ende agrega un espacio a current, si no es ninguno de los anteriores se agrega el valor a current y si shift esta activado se desactiva(línea 178 a la196) se cierra la función flecha(línea 202)

**Paso 29:** se selecciona el input y se le itera con un forEach(línea 205), se le añade un evento cada que tenga el foco de atencion y current se establece en e.target lo que significa que ahora current contiene una referencia al elemento input, se cierra la función renderKeyboard() (línea 215)

Texto

Descripción generada automáticamente

Aplicación

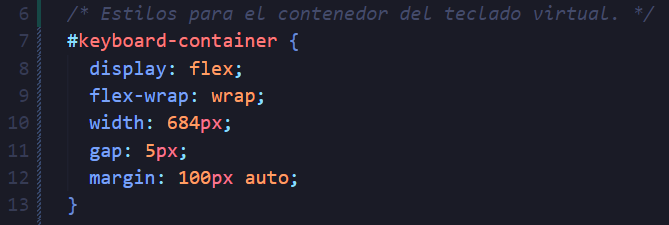
Descripción generada automáticamente con confianza baja

**Estrustura css**

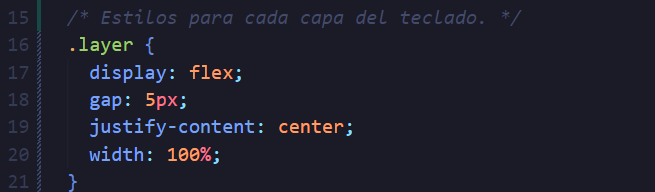
**Paso 30**: en el archivo css establece las fuentes del body

Texto

Descripción generada automáticamente

**Paso 31:** la propiedad flex establece la capacidad que tiene un elemento para alterar sus dimensiones y llenar el espacio disponible, la propiedad flex-wrap especifica si los elementos "hijos" son obligados a permanecer en una misma línea o pueden fluir en varias líneas, la propiedad gap especifica los canales entre las filas y las columnas de la cuadrícula.

**Paso 32**: la propiedad justify-content define como el buscador distribuye el contenido entre y alrededor de los objetos a lo largo del contendor.

Tabla

Descripción generada automáticamente con confianza baja

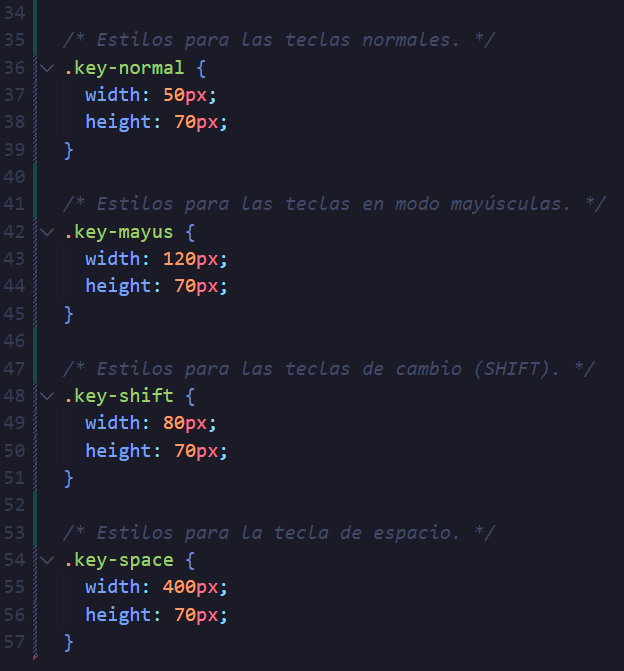
**Paso 33:** la propiedad align-items pone el valor en este caso center a todos los hijos directos como un grupo en el contenedor.



Tabla

Descripción generada automáticamente

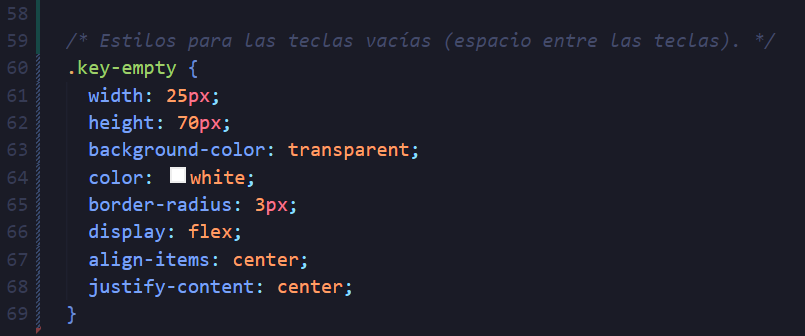
**Paso 34:** se le da estilo a las diferentes teclas



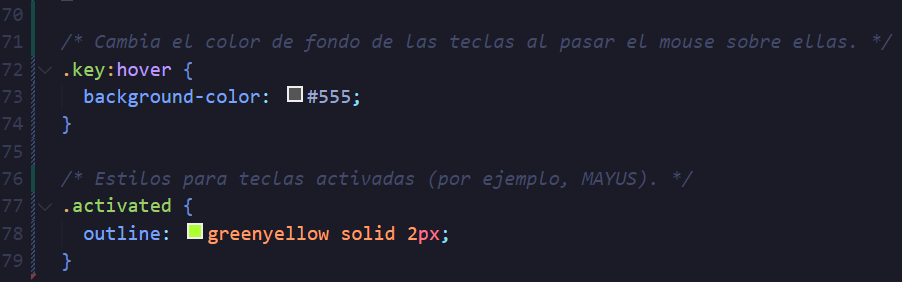
Diagrama

Descripción generada automáticamente con confianza baja

**Paso 35:**con este estilo se modifica un pequeño espacio vacio a la derecha del teclado



**Paso 36:** la pseudo clase hover se activa cuando coincide el usuario interactúa con un elemento con un dispositivo señalador, pero no necesariamente la activa, la propiedad outline marca los bordes de las teclas activadas.



Diagrama

Descripción generada automáticamente

**Paso 37:** la propiedad block genera saltos de linea antes y despues del elemento cuando este esta en un flujo normal

Texto

Descripción generada automáticamente

Diagrama

Descripción generada automáticamente