**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 “comentario”.



**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”

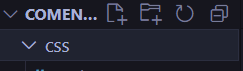
Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

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

Descripción generada automáticamente

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



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



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

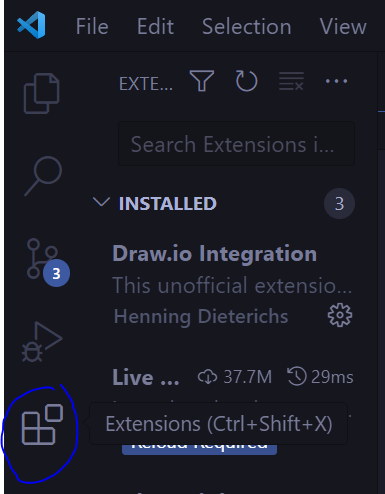
Una captura de pantalla de un celular

Descripción generada automáticamente

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



**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**

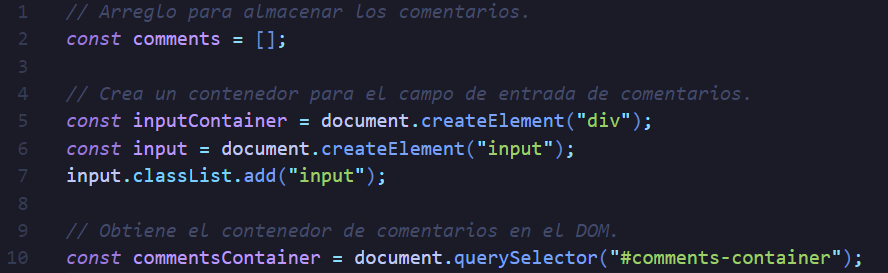
**Paso 14**: en el body del html creamos un div que será el que contenga toda la app (línea 14) y un div que será el contenedor de los comentarios(línea 16)

Texto

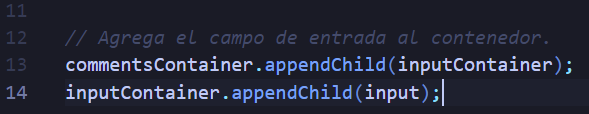
Descripción generada automáticamente

**Estructura JS**

**Paso 15**: creamos un array donde se guardarán los comentarios(linea2), creamos un div y un input (línea 5 y 6), a este ultimo se le agrega la clase input (línea 7), del DOM (estructura del documento HTML) obtenemos el contenedor e comentarios (línea 10)



**Paso 16:** con el appendChild se agrega un nuevo nodo al final de la lista de un elemento hijo de un elemento padre especificado (línea 13 y 14)(un nodo es cualquier etiqueta del cuerpo, como un párrafo, el mismo body o incluso las etiquetas de una lista).

****

**Paso 17**: se le agrega un listener al input con el evento “keydown” este se activara cuando se presione una tecla(línea 17), se crea una función flecha que lleva a la función handleEnter la cual tiene dos parámetros (línea 18).

Interfaz de usuario gráfica, Texto

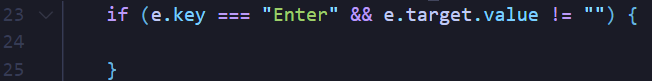
Descripción generada automáticamente

**Paso 18:** Creamos la función handleEnter que será la encargada de accionar el enter

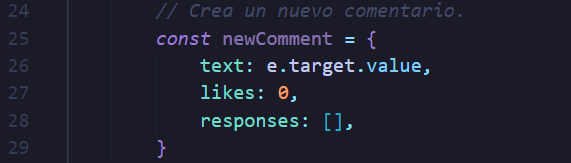
Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

**Paso 19:**dentro de la funciónhandleEnter creamos un condicional que evalua si la entrada es enter y una cadena que no este vacía.

****

**Paso 20**: si la condición es verdadera se crea un objeto con tres propiedades text, likes y responses en text el valor de la entrada, en likes el numero de likes y por ultimo en responses los comentarios en un array.



Texto

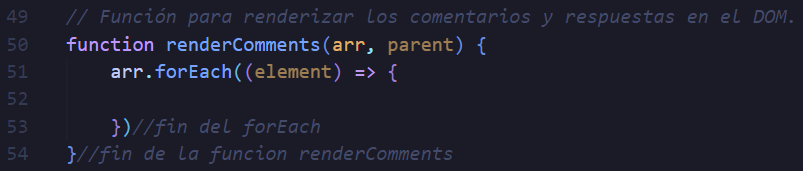
Descripción generada automáticamente**Paso 21**: se crea un condicional anidado que evalúa si el comentario actual es nulo si es nulo agrega el comentario al principio del array de lo contrario agrega una respuesta al comentario actual

**Paso 22**: limpia el campo de entrada (línea 40), borra los comentarios (línea 43 y 44) y los vuelve a renderizar (línea 45)

Texto

Descripción generada automáticamente

**Paso 23**: con esta función renderizaremos los comentarios y con el forEach recorreremos los elementos del arr.



**Paso 24:** dentro del forEach crearemos un commentContainer y le agregamos un div y la clase “comment-container”

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

**Paso 25:** dentro del forEach crearemos un responsesContainer y le agregamos un div y la clase “responses-container”

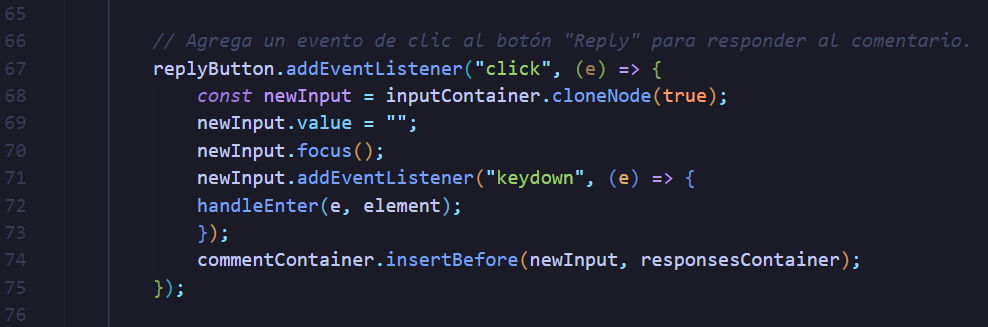
Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

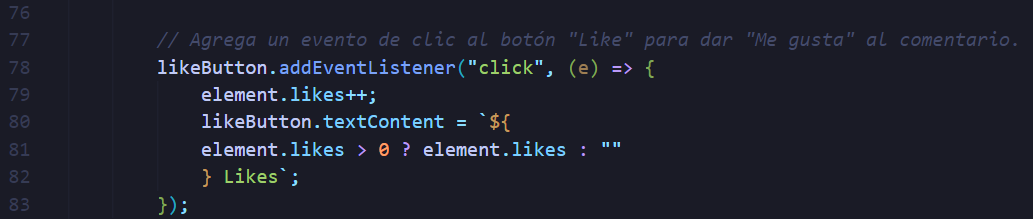
**Paso 26:** Creamos dos botones para los comentarios y para los likes (línea 60 y 61), y con el textContent que digan replay y like (línea 63 y 64)

****

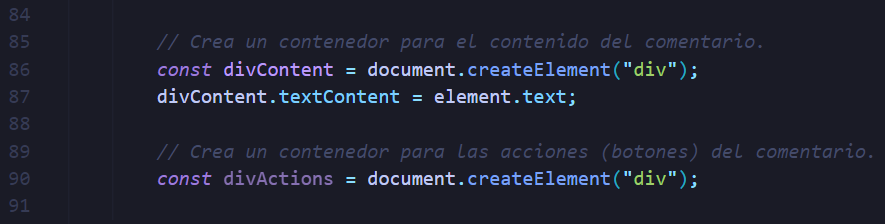
**Paso 27:** agregamos un listenera replyButtoncon el evento click (línea 67), creamos newInput y lo hacemos igual al nodo de inputContainer (línea 68), le damos el valor de una cadena vacía y le damos el foco de atención(línea 69 y 70), creamos un listener con el evento de presionar tecla y llamamos a la función handleEnter() (línea 72) fuera del primer listener decimo que commnetContainer sea insertado antes de newInput y responsesContainer(linea 74)



**Paso 28**: agregamos un listener al botón likeButton con el evento click (linea 78), en cada iteración se le sumara uno (linea 79) y con el textContent se actualizara validando con el operador ternario si los likes son mayor que cero en caso de que no el textContent permanecerá vacio (líneas 80 a la 82)



**Paso 29:** creamos dos divs uno para el contenido del comentario el cual tendrá el texto del elemento (línea 86 y 87) y el otro contendrá los botones del comentario(línea 90)

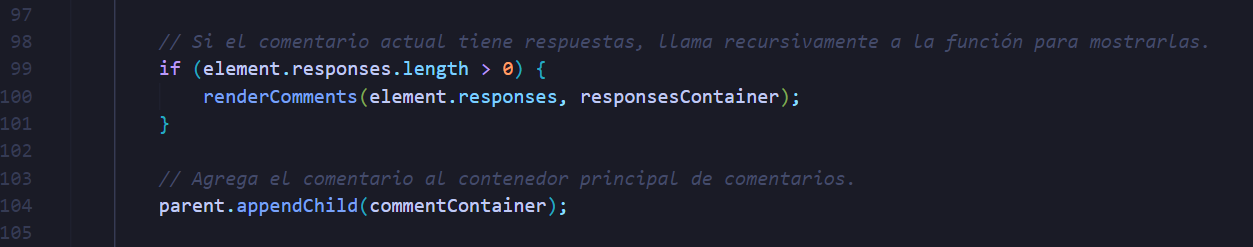


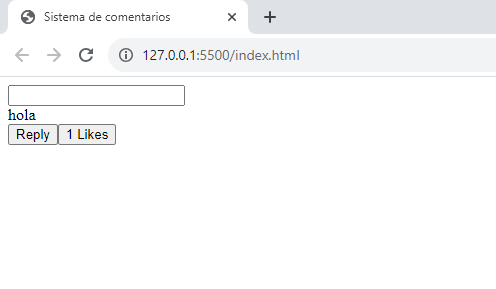
**Paso 30**: Agrega el contenido del comentario al contenedor (línea 93), Agrega los botones al contenedor (línea 94), Agrega el botón "reply" al contenedor de botones (línea 95), Agrega el botón "like" al contenedor de botones (línea 96).

Texto

Descripción generada automáticamente

**Paso 31**: se crea un condicional donde el comentario tiene respuesta se llama a la función renderComments con dos argumentos: element.responses y responsesContainer los comentarios se extraen de element.responses y se representan en el contenedor responsesContainer.



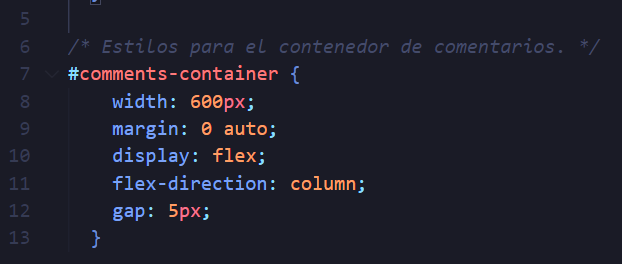


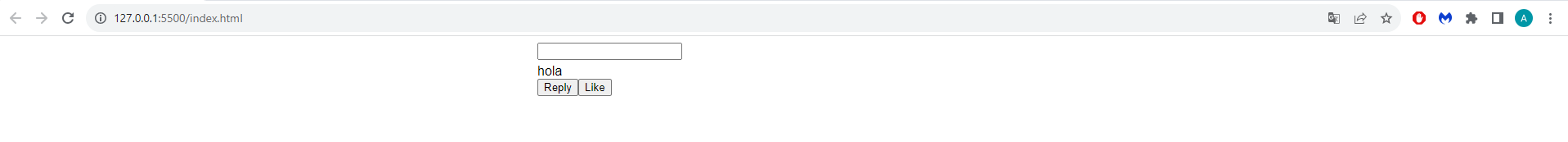
**Estructura CSS**

**Interfaz de usuario gráfica, Texto

Descripción generada automáticamentePaso 32**: en el archivo css en el body establecemos la fuentes de la letra

**Paso 33**: la priedad flex establece la capacidad que tiene un elemento para alterar sus dimensiones y llenar el espacio disponible, la propiedad gap especifica los canales entre las filas y las columnas de la cuadrícula y la propiedad flex-direction especifica cómo colocar los objetos flexibles en el contenedor con el valor column los comentarios estarán uno sobre otros





**Paso 34**: ahora en el contenedor del comentario le ajustamos el border, el radio del borde, el padding, el color de la letra y el fondo

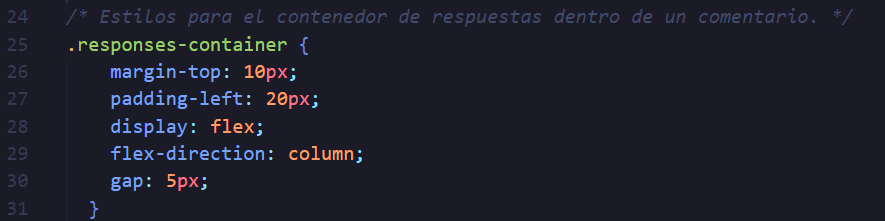
Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

Patrón de fondo

Descripción generada automáticamente

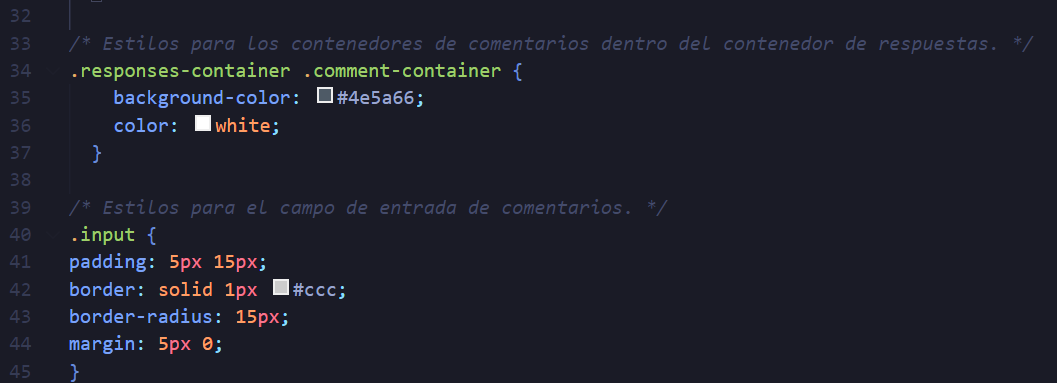
**Paso 35**: ahora en el contenedor de la respuesta le ajustamos el margen hacia arriba, el padding hacia la izquierda, establecemos que display es flex, establecemos que la dirección es columna y le damos gap



Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Paso 36**: cambiamos el color de los contenedores para que se diferencien y en el input ajustamos el padding, el borde, el radio del borde y el margen



Interfaz de usuario gráfica, Patrón de fondo

Descripción generada automáticamente con confianza media

**Paso 37**: la propiedad cursor cambia el puntero del ratón cuando esta se encuentra sobre un elemento en este caso los botones

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

**Paso 38**: la pseudo clase hover cambia el diseño de un elemento cuando interactúa con este en este caso cambia el color del fondo y de la letra del botón

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente