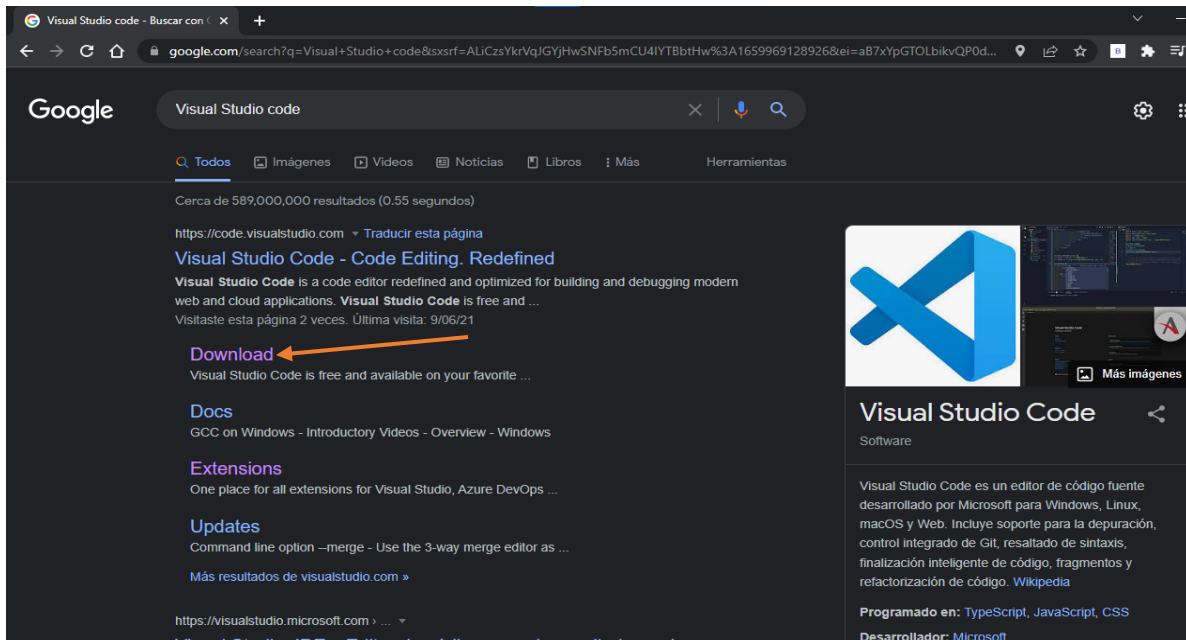
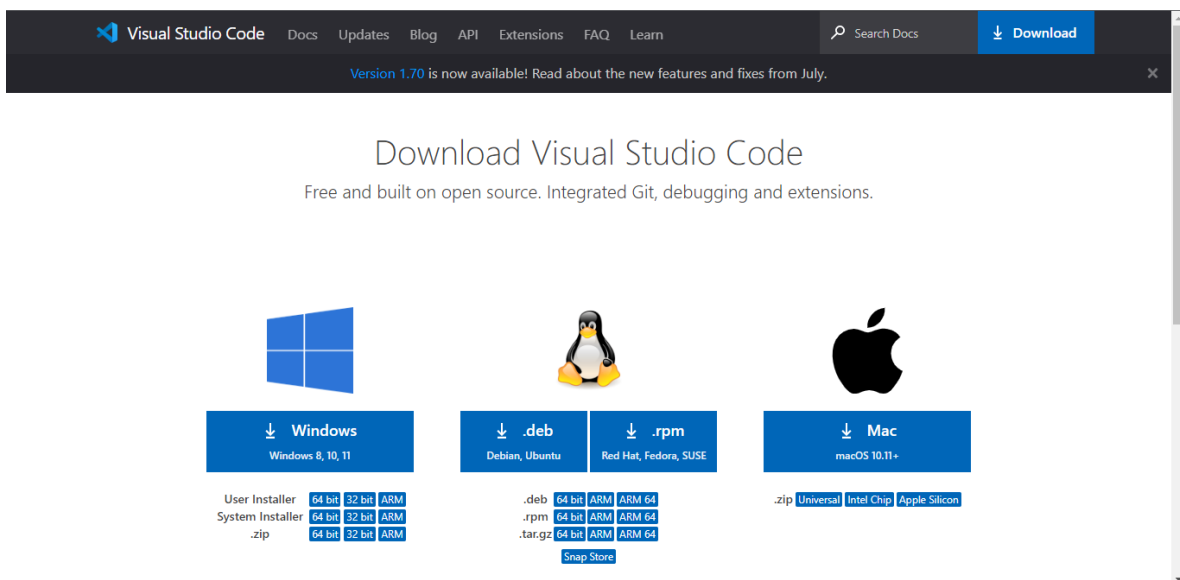


Proceso instalación Visual Studio Code

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



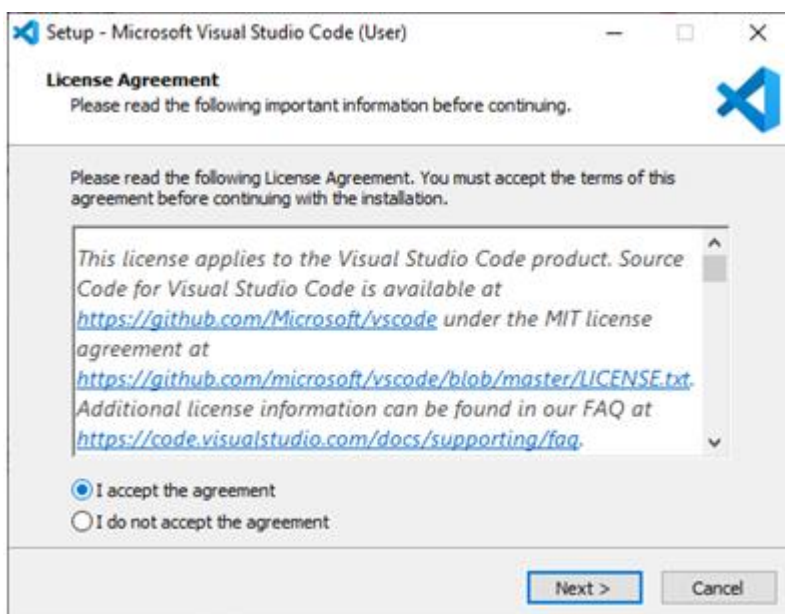
Paso 2: Seleccionamos el sistema operativo que tenemos y lo descargamos.



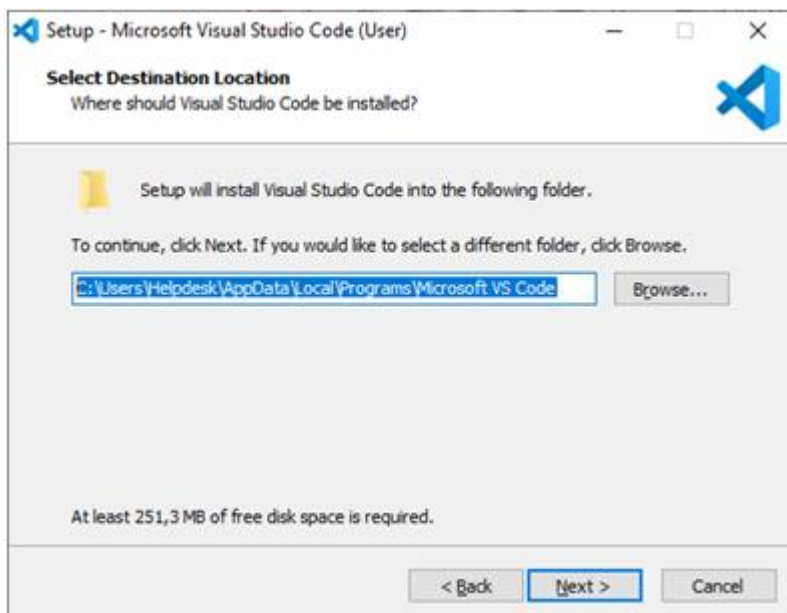
Paso 3: Al darle clic nos descargará un .exe, al cual le daremos clic encima.



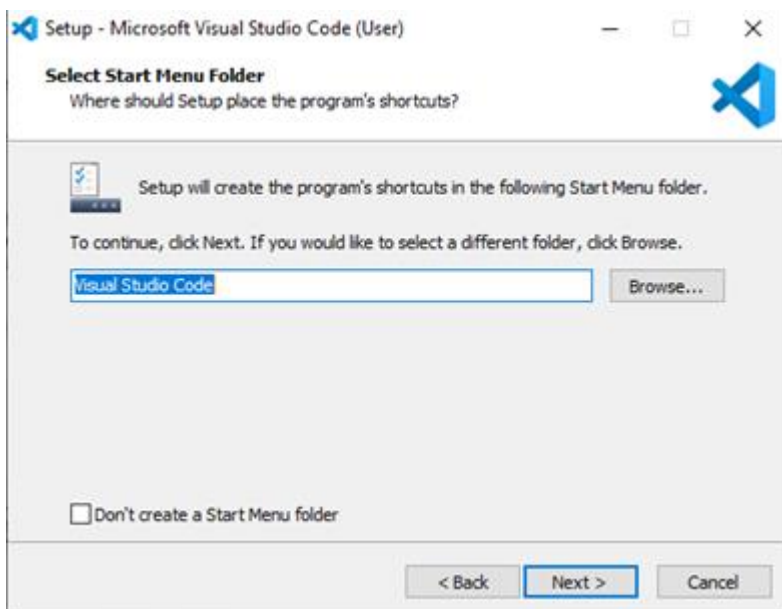
Paso 4: Lee y acepta el acuerdo de licencia. Haz clic en Next para continuar.



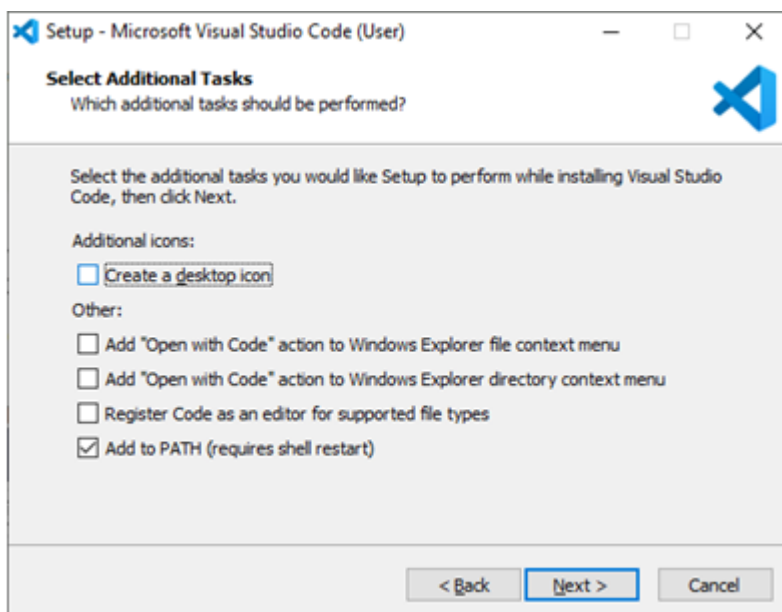
Paso 5: Puedes cambiar la ubicación de la carpeta de instalación o mantener la configuración predeterminada. Haz clic en Next para continuar.



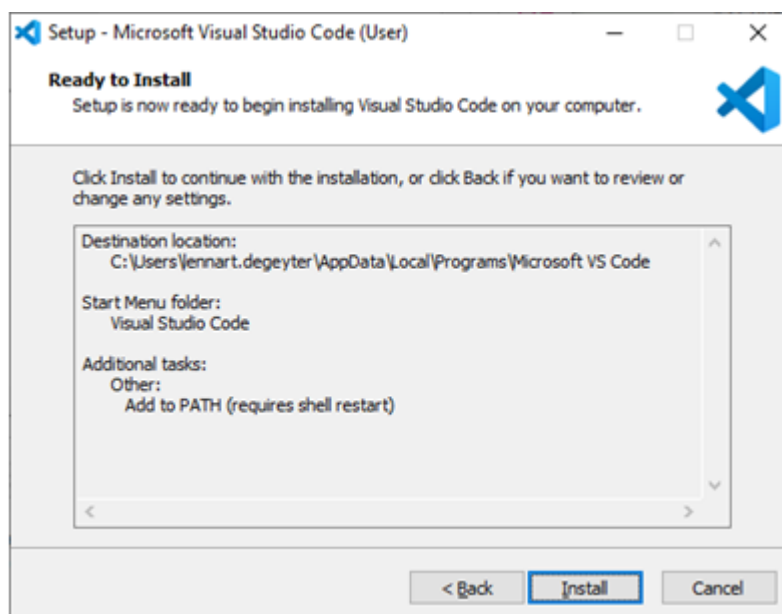
Paso 6: Elige si deseas cambiar el nombre de la carpeta de accesos directos en el menú Inicio o si no deseas instalar accesos directos en absoluto. Haz clic en Next.



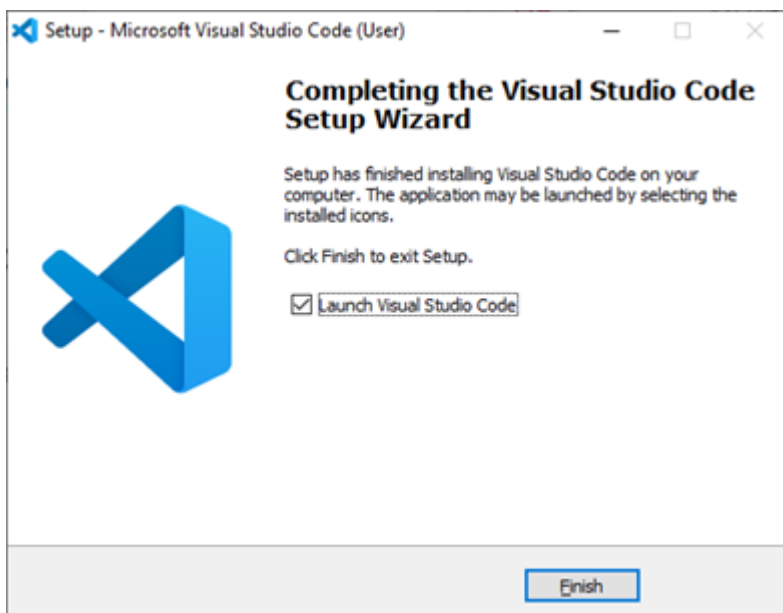
Paso 7: Selecciona las tareas adicionales, por ej. crear un icono en el escritorio o añadir opciones al menú contextual de Windows Explorer. Haz clic en Next.



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



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

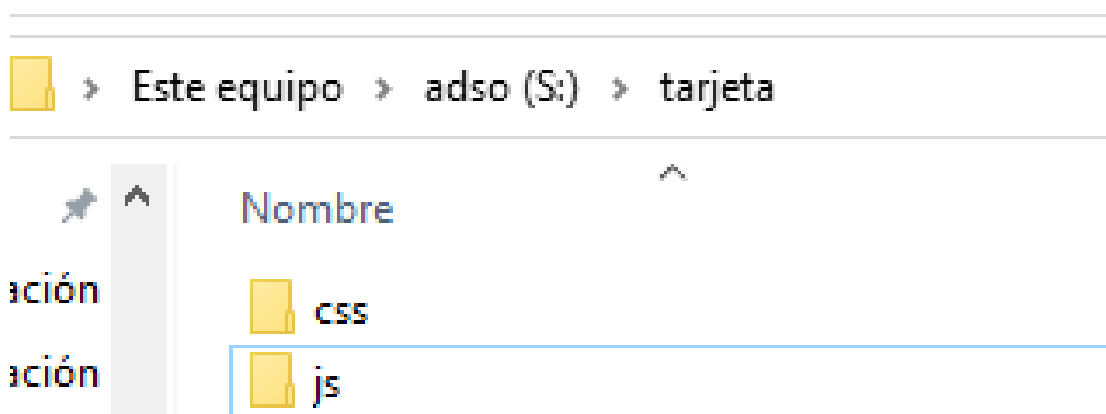


Creación de archivos y carpetas

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

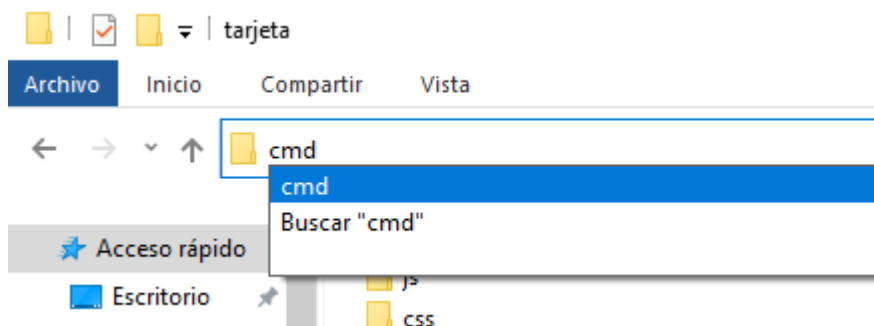


Paso 2: Dentro de esta, crearemos otras dos carpetas llamadas “css”, la cual contendrá todo lo estético de nuestra página y otra llamada “js” la cual contendrá la lógica de nuestra página.

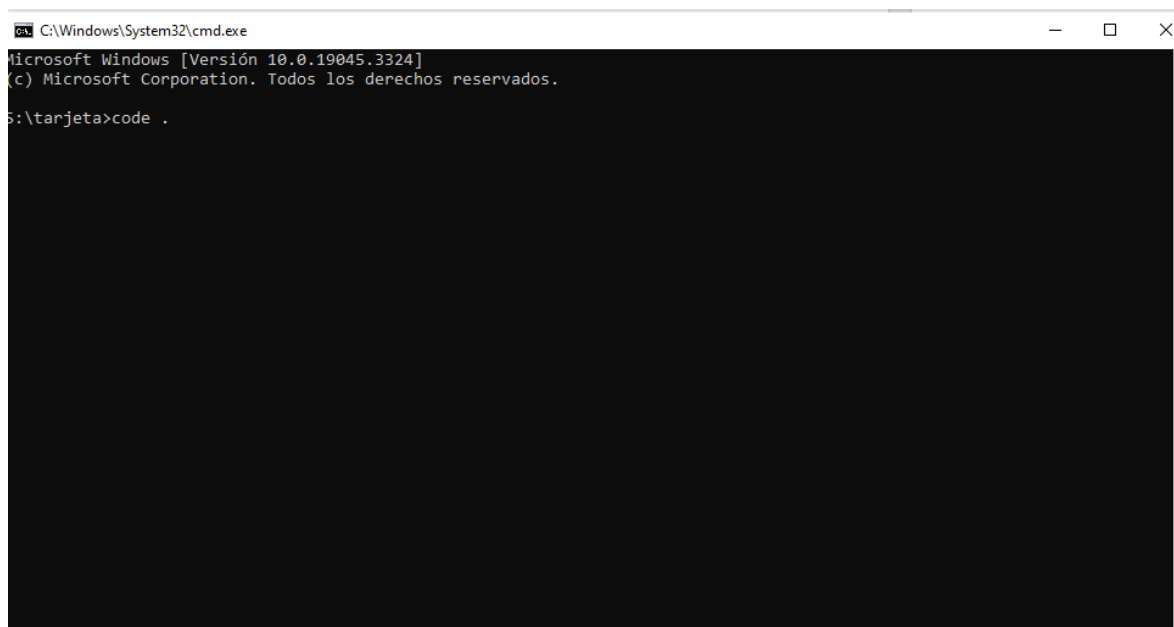


Tarjeta

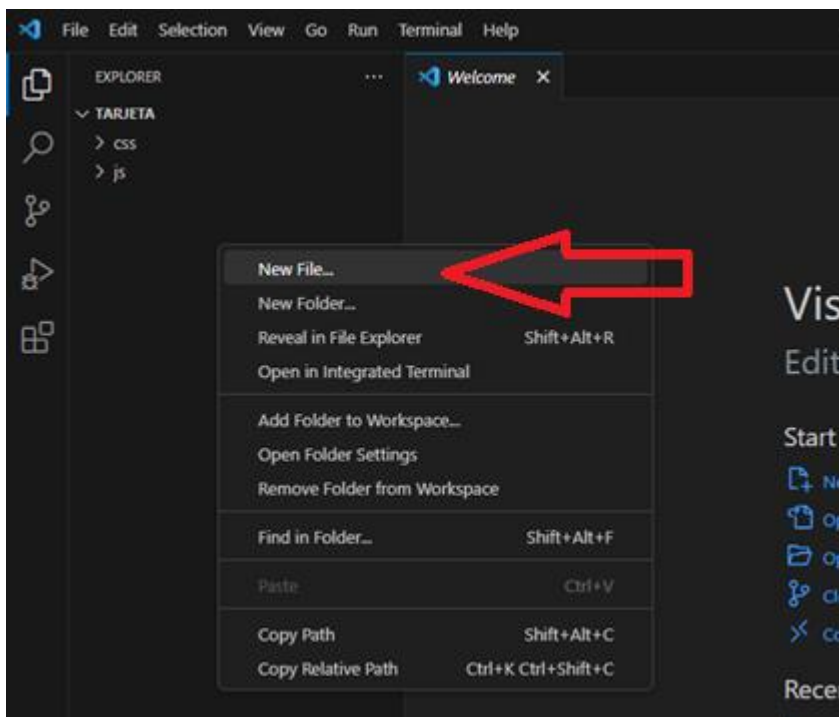
Paso 1: Para abrir el Visual Studio Code, haremos lo siguiente: Dentro de la carpeta raíz, en la barra superior, escribiremos cmd y le damos enter.



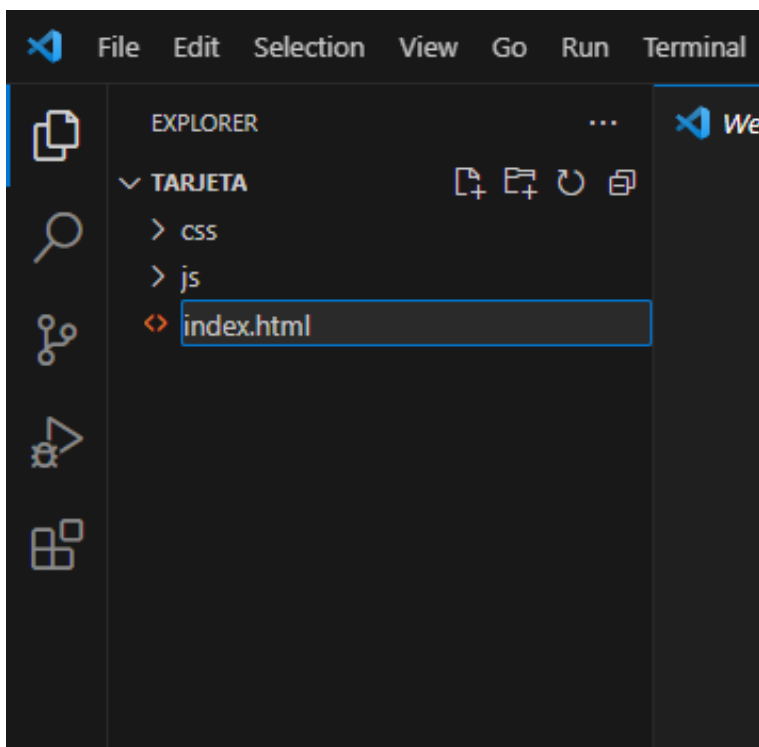
Paso 2: Eso nos abrirá una terminal, solo tendremos que escribir “code .”.



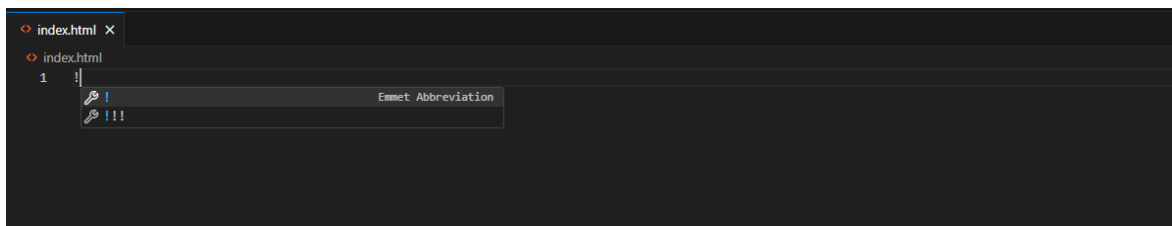
Paso 3: Esto nos abrirá el visual studio code. Ahora, ya que estamos dentro, crearemos un archivo llamado “index.html” dándole click derecho en la barra de la izquierda y luego dale “new file” o “nuevo archivo”



Paso 4: Llamaremos el archivo “index.html”



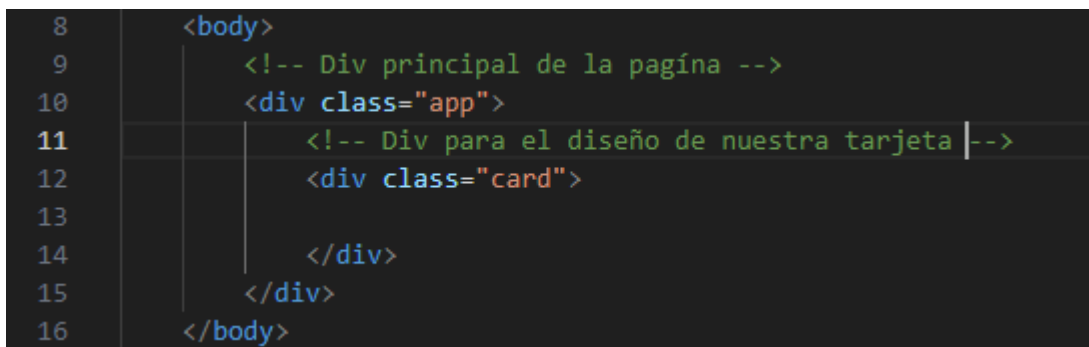
Paso 5: Una vez en el archivo recién creado, colocamos el código “!” y escogemos la primera opción.



Paso 6: Una vez colocada la estructura, cambiaremos el título (línea 6) por “Tarjeta”



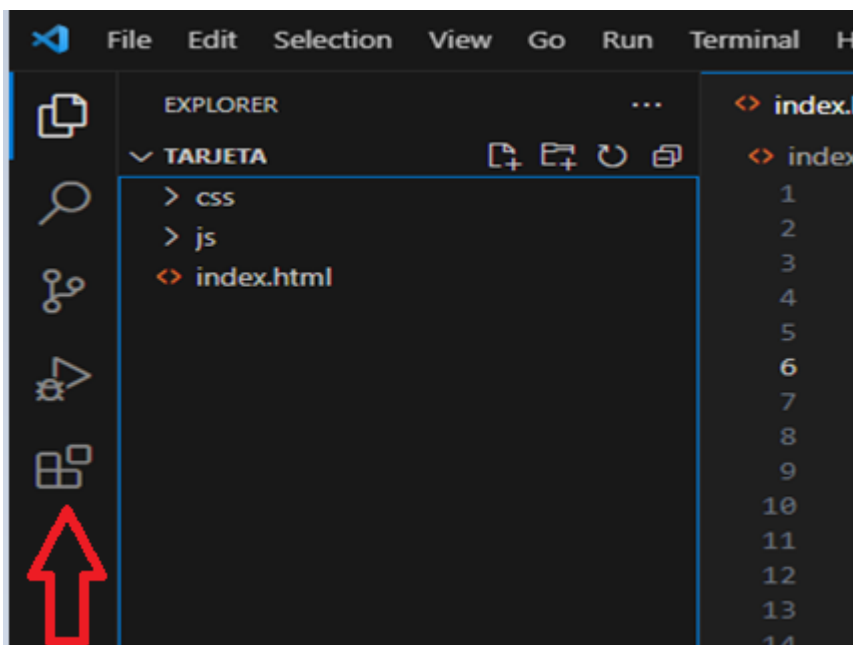
Paso 7: Ahora dentro del body crearemos un div principal para nuestra página (Línea 10) le pondremos una clase llamada “app”, y además crearemos otro div dentro de ese mismo div (línea 12) y le pondremos la clase “card” ya que será el que tendrá el diseño de nuestra tarjeta.



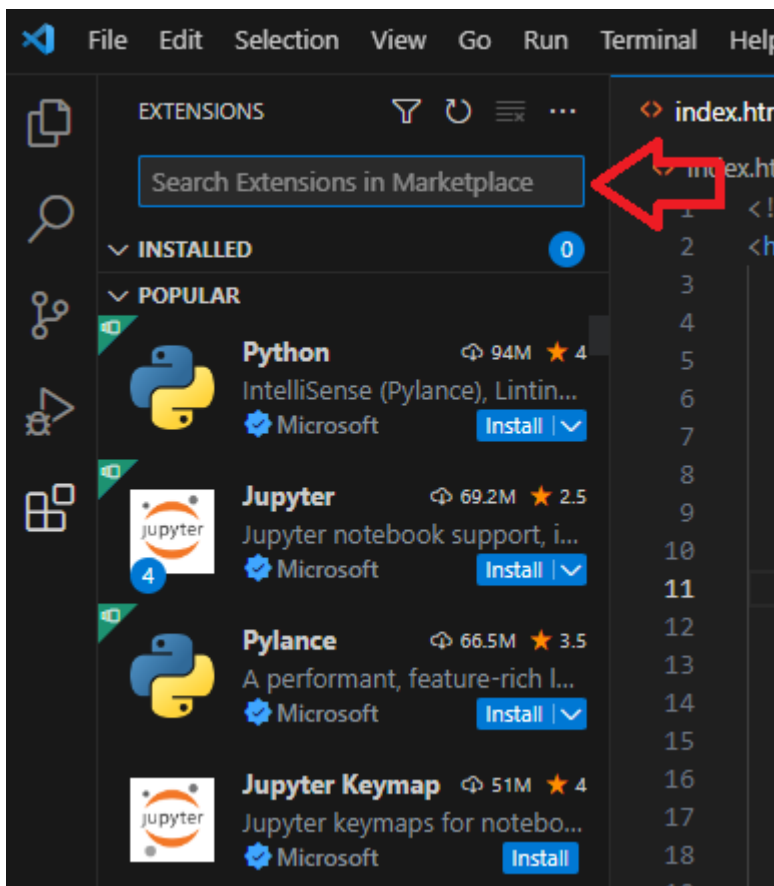
Paso 8: Dentro del div de la tarjeta (línea 12) pondremos otro div (Linea 14) el cual contendrá el nombre del dueño de la tarjeta y le pondremos como clase “holder”, también crearemos 3 inputs, los cuales contendrán los números de la tarjeta, la fecha de la tarjeta, y su CVV, respectivamente, también les pondremos sus id como “inputCard”, “inputDate” e “inputCVV” respectivamente.

```
11      <!-- Div para el diseño de nuestra tarjeta -->
12      <div class="card">
13          <!-- Div - portador de la tarjeta -->
14          <div class="holder">Santiago Flórez Echavarría</div>
15
16          <!-- Input para colocar el numero de la tarjeta -->
17          <input type="text" id="inputCard">
18          <!-- Input para colocar fecha de la expiración -->
19          <input type="text" id="inputDate">
20          <!-- Input para colocar el CVV -->
21          <input type="text" id="inputCVV">
22      </div>
```

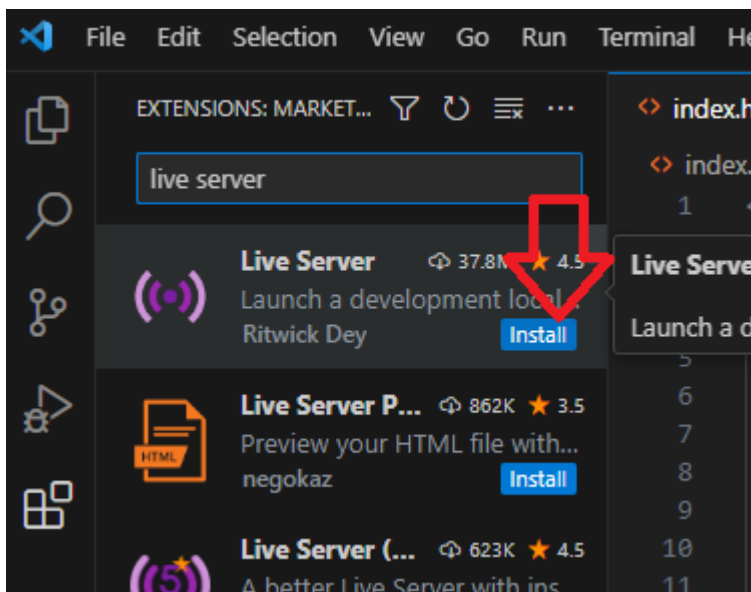
Paso 9: Ahora para poder visualizar como se esta viendo nuestro proyecto le daremos click al icono de abajo a la izquierda.



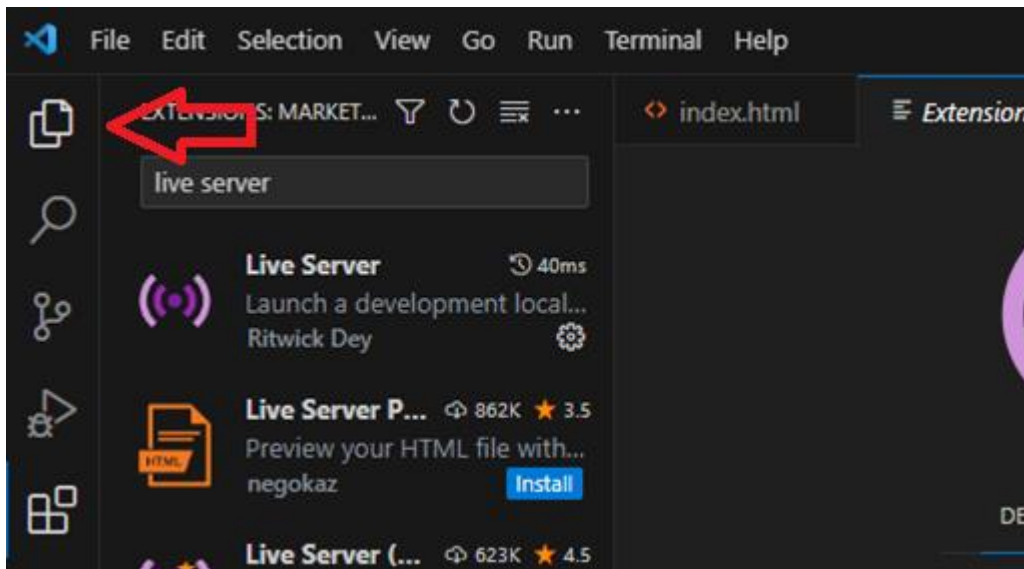
Paso 10: Nos saldrá esta pestaña y buscaremos live server, en la barra de búsqueda.



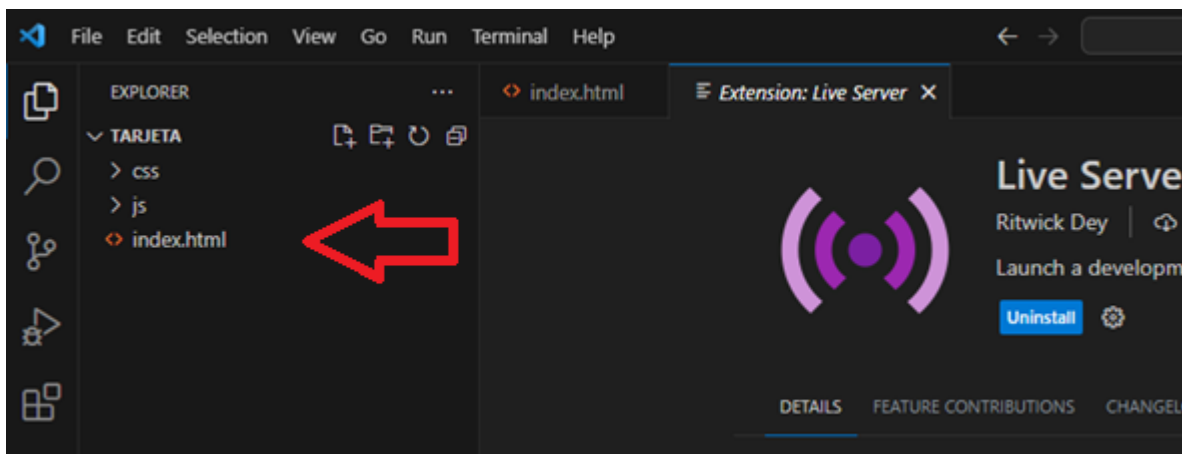
Paso 11: Ahora que encontramos lo que necesitamos, le damos click a “install”



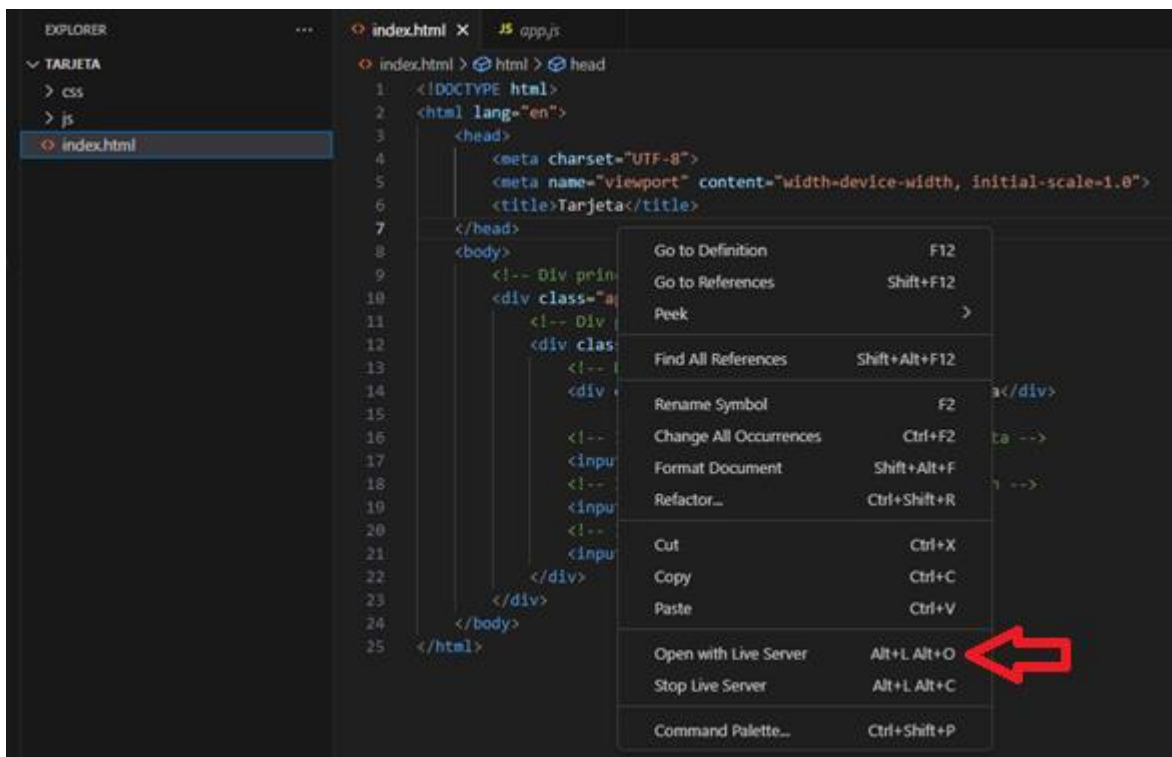
Paso 12: Ahora que tenemos instalado el live server nos debería de aparecer algo así donde podremos volver a nuestro proyecto dando click al explorador.



Paso 13: Una vez te sale el explorador volvemos al Index.html



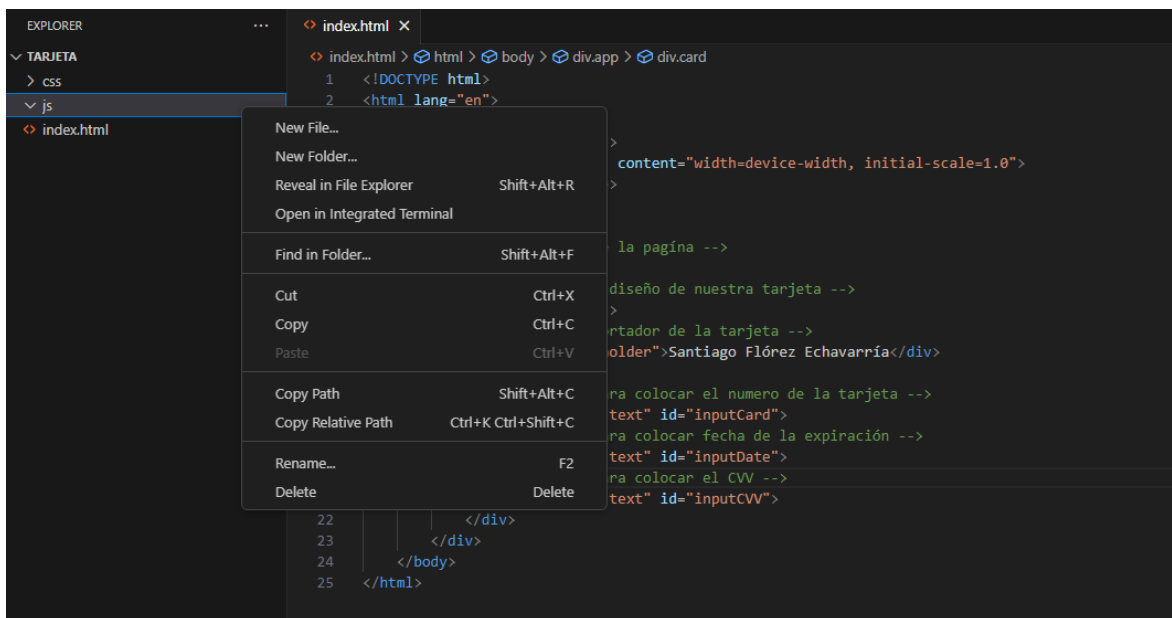
Paso 14: Una vez volvemos al index.html damos click derecho en cualquier parte del código y damos click izquierdo a “open with live server”



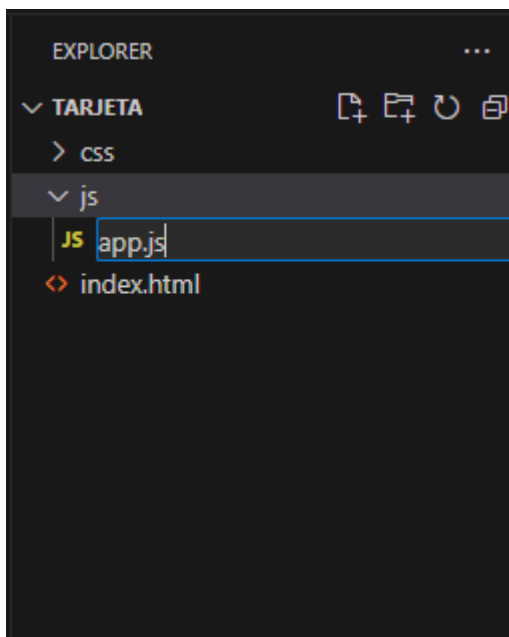
Paso 15: Esto nos abrirá una pestaña donde se verá como va actualmente nuestro proyecto.



Paso 16: Una vez que vimos que nuestro proyecto va bien, daremos click derecho a la carpeta “js” y le damos click izquierdo a “new file”



Paso 17: Creamos un archivo llamado “app.js”

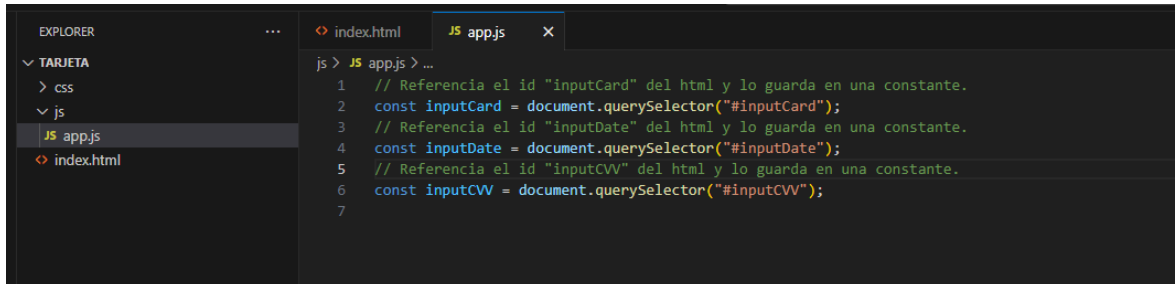


Paso 18: Una vez creado el archivo lo vinculamos en nuestro html (línea 25) justo antes de terminar el body.

```
20      <!-- Input para colocar el CVV -->
21      <input type="text" id="inputCVV">
22    </div>
23  </div>
24  <!-- JavaScript -->
25  <script src="js/app.js"></script>
26 </body>
27 </html>
```

Paso 19: Ahora vamos a referenciar los id de nuestro archivo "index.html" y lo almacenaremos en una constante

Usamos el código : "document.querySelector("#id")" ya que son id.



```
js > JS app.js > ...
1  // Referencia el id "inputCard" del html y lo guarda en una constante.
2  const inputCard = document.querySelector("#inputCard");
3  // Referencia el id "inputDate" del html y lo guarda en una constante.
4  const inputDate = document.querySelector("#inputDate");
5  // Referencia el id "inputCVV" del html y lo guarda en una constante.
6  const inputCVV = document.querySelector("#inputCVV");
7
```

Paso 20: Ahora vamos a crear una estructura que debe de seguir la tarjeta, guardando esa estructura en constantes, por ejemplo la tarjeta de creamos tendrá 16 numero y cada 4 los separa 1 guion, la fecha tendrá el mes y el día XX/XX, y el CVV tendrá solo 3 números.

```
6  const inputCVV = document.querySelector("#inputCVV");
7
8  // Creación de la estructura de la tarjeta.
9  const maskNumber = "####-####-####-####";
10 const maskDate = "##/##";
11 const maskCVV = "###";
```

Paso 21: Ahora crearemos otras 4 constantes, las cuales 3 serán arrays (línea 15,16,17) y estas se encargarán de que se cumpla el formato de la tarjeta.

```
13 // Constantes que asegura que se cumpla el formato de la tarjeta.
14 let current = "";
15 let cardNumber = [];
16 let dateNumber = [];
17 let cvvNumber = [];
18
```

Paso 22: Ahora vamos a crear una función llamada "handleInput" el cual tendrá 3 parametros, los cuales permiten a la función "handleInput" recibir y procesar información para aplicar la mascara de formato a la entrada de datos, asegurándose de que solo se permitan caracteres válidos y que se respete la longitud de la máscara.

Dentro de esta función vamos a declarar una variable con todos los numero válidos para la tarjeta (línea 22).

```
18
19 // Función que maneja la entrada de números en un campo según la máscara proporcionada.
20 function handleInput(mask, key, arr) {
21   // Lista de números válidos.
22   let numbers = ["0", "1", "2", "3", "4", "5", "6", "7", "8", "9"];
23 }
```

Paso 23: Ahora creamos dentro de la función una condicional que hará que en caso de que haya algo escrito, y se aprete el suprimir, eliminará el último elemento. (Línea 25-28)

```
20 function handleInput(mask, key, arr) {
21   // Lista de números válidos.
22   let numbers = ["0", "1", "2", "3", "4", "5", "6", "7", "8", "9"];
23
24   // Si se presiona "Backspace" y hay elementos en el arreglo, elimina el último elemento.
25   if (key === "Backspace" && arr.length > 0) {
26     arr.pop();
27     return;
28   }
29
30 }
```

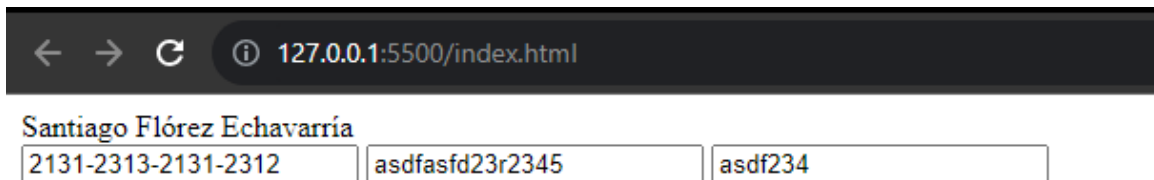

Paso 24: Debajo de esa condición crearemos una nueva, que lo que hará es que la tecla que se incluya si sea válida, y que no se exceda del array (línea 31), además dentro de esa condición crearemos una nueva que reconocerá si lo que sigue en el arreglo es un “-“ o “/” y lo añadirá antes del número(línea 33 - 34) de lo contrario solo añadirá el número (línea 37)

```
24 // Si se presiona "Backspace" y hay elementos en el arreglo, elimina el último elemento.
25 if (key === "Backspace" && arr.length > 0) {
26     arr.pop();
27     return;
28 }
29
30 // Si la tecla ingresada es un número válido y no se excede la longitud de la máscara, agrega el número al arreglo.
31 if (numbers.includes(key) && arr.length + 1 <= mask.length) {
32     // Si la posición actual de la máscara es un guión o barra, agrega ese carácter antes del número.
33     if (mask[arr.length] === "-" || mask[arr.length] === "/") {
34         arr.push(mask[arr.length], key);
35     } else {
36         // Si no, simplemente agrega el número al arreglo.
37         arr.push(key);
38     }
39 }
40 }
```

Paso 25: Ahora creamos un evento que escuchará la tecla en el campo de la tarjeta (línea 42), crearemos una condición de que en caso se aprete “tab”, no se realice ninguna acción (línea 44-46), Ahora vamos a quitar el comportamiento predeterminado de la tecla que se aprete (línea 48), además de vamos a llamar a la función “handleInput” para que maneje la entrada del campo de la tarjeta, ósea las teclas que son válidas, y lo demás que pusimos allí (línea 50) y por último vamos a colocar para que se actualice el valor del campo de entrada con los números ingresados siguiendo la máscara.

```
40 }
41 // Evento de escucha para la tecla en el campo de entrada de la tarjeta.
42 inputCard.addEventListener("keydown", (e) => {
43     // Si se presiona la tecla "Tab", no realiza ninguna acción y sale del evento.
44     if (e.key === "Tab") {
45         return;
46     }
47     // Evita el comportamiento predeterminado de la tecla
48     e.preventDefault();
49     // Llama a la función "handleInput" para manejar la entrada en el campo de la tarjeta.
50     handleInput(maskNumber, e.key, cardNumber);
51     // Actualiza el valor del campo de entrada con los números ingresados, siguiendo la máscara.
52     inputCard.value = cardNumber.join("");
53 });
54 }
```

Paso 26: Ahora vamos a mirar en nuestra página como hicimos antes (Paso 14) y veremos como el primer campo ya sigue el formato mientras que los demás no.



← → ↻ ⓘ 127.0.0.1:5500/index.html

Santiago Flórez Echavarría


2131-2313-2131-2312 asdfasfd23r2345 asdf234

Paso 27: Ahora volvemos al código para terminar los otros 2 campos.

Creamos un nuevo evento para la fecha y ponemos de nuevo la condición del tab (línea 57), volvemos a quitar la función predeterminada (línea 60), y volvemos a llamar la función “handleInput” y esta vez colocamos la información de la fecha como parámetros, (línea 61) y por último vamos a colocar para que se actualice el valor del campo de entrada con los números ingresados siguiendo la máscara. (línea 63)

```
54 |
55 | // Evento de escucha para la tecla en el campo de entrada de la fecha.
56 | inputDate.addEventListener("keydown", (e) => {
57 |     if (e.key === "Tab") {
58 |         return;
59 |     }
60 |     e.preventDefault();
61 |     handleInput(maskDate, e.key, dateNumber);
62 |     // Actualiza el valor del campo de entrada de la fecha con los números ingresados, siguiendo la máscara.
63 |     inputDate.value = dateNumber.join("");
64 | });
65 |
```

Paso 28: Ahora vamos a mirar en nuestra página como hicimos antes (Paso 14) y veremos cómo los 2 primeros campos ya siguen el formato y el último no.



Santiago Flórez Echavarría

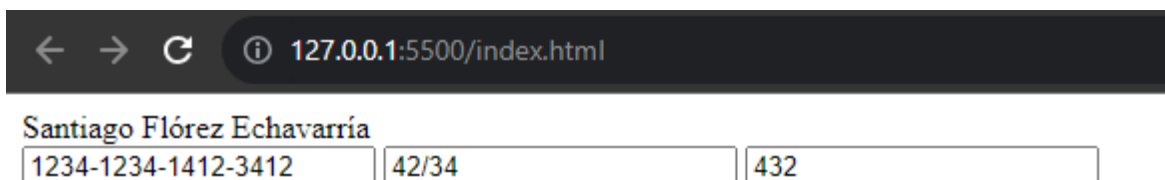
1231-2313-2131-2313 31/23 312313qweqwedas

Paso 29: Ahora volvemos al código para terminar los otros 2 campos.

Creamos un nuevo evento para la fecha y ponemos de nuevo la condición del tab (línea 68), volvemos a quitar la función predeterminada (línea 71), y volvemos a llamar la función “handleInput” y esta vez colocamos la información de la fecha como parámetros, (línea 72) y por último vamos a colocar para que se actualice el valor del campo de entrada con los números ingresados siguiendo la máscara. (línea 74)

```
66 // Evento de escucha para la tecla en el campo de entrada del CVV.
67 inputCVV.addEventListener("keydown", (e) => {
68     if (e.key === "Tab") {
69         return;
70     }
71     e.preventDefault();
72     handleInput(maskCVV, e.key, cvvNumber);
73     // Actualiza el valor del campo de entrada del CVV con los números ingresados, siguiendo la máscara.
74     inputCVV.value = cvvNumber.join("");
75 });
76
```

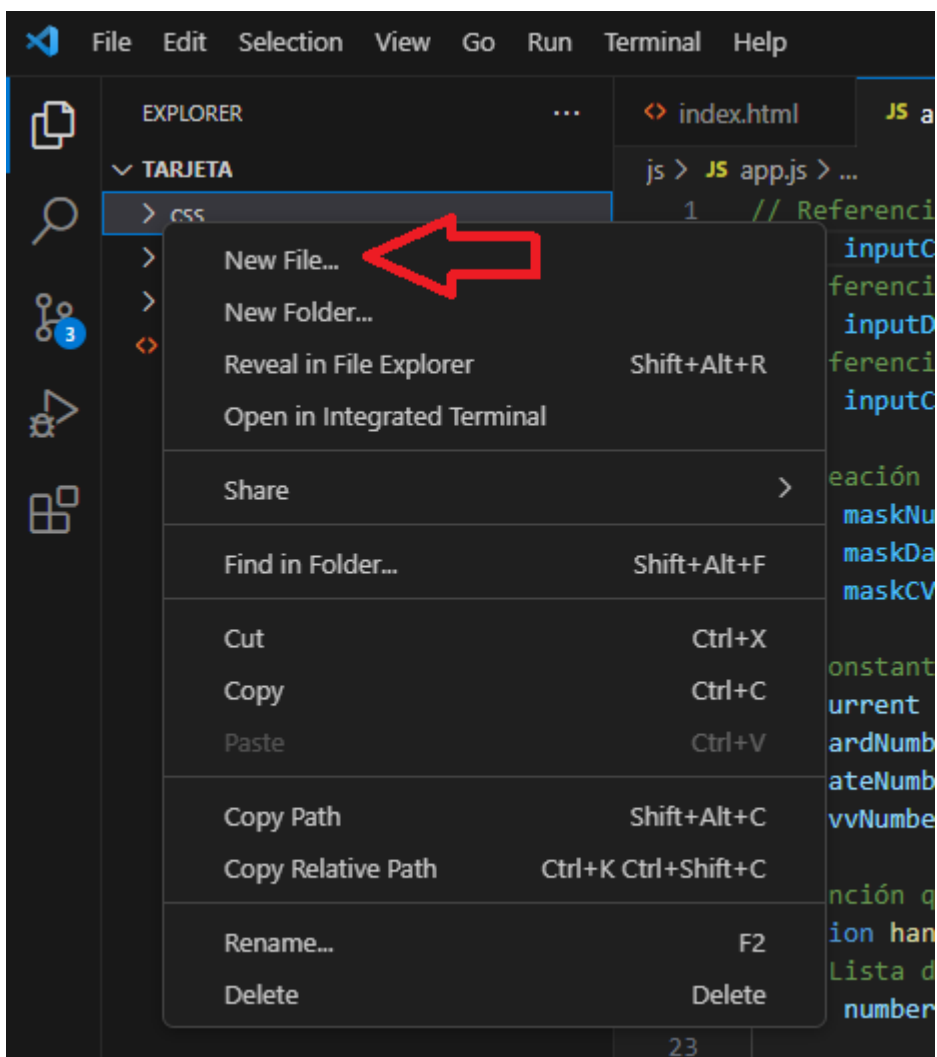
Paso 30: Ahora podremos apreciar como ha quedado todo perfectamente funcionando en nuestra página (Paso 14). Ahora podemos apreciar que los 3 campos están funcionando perfectamente.



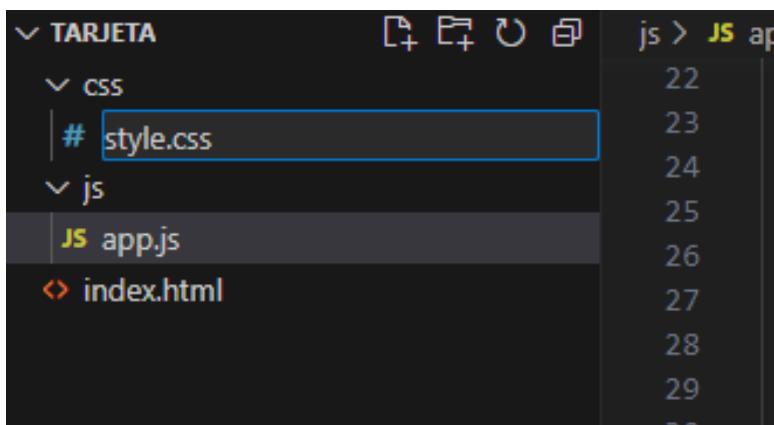
Santiago Flórez Echavarría

1234-1234-1412-3412 42/34 432

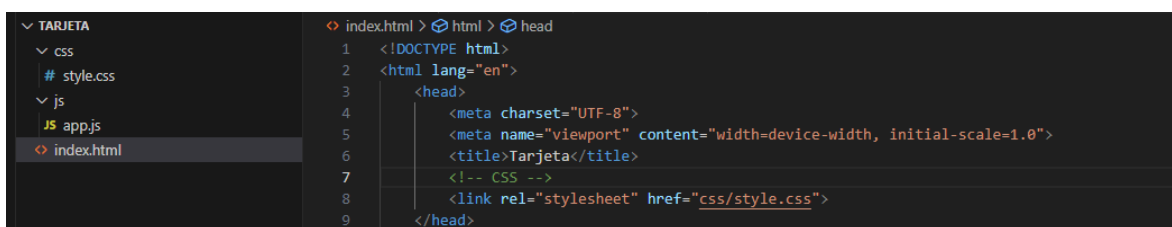
Paso 31: Ahora que la función de nuestra página va bien, vamos a añadirle el estilo, lo primero que haremos será crear un archivo en la carpeta “css” seleccionándolo desde el explorador



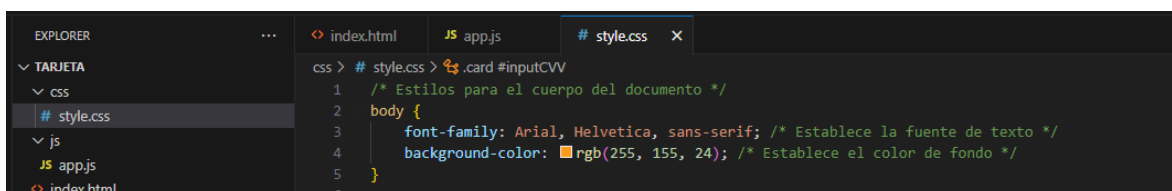
Paso 32: Le pondremos al archivo “style.css”



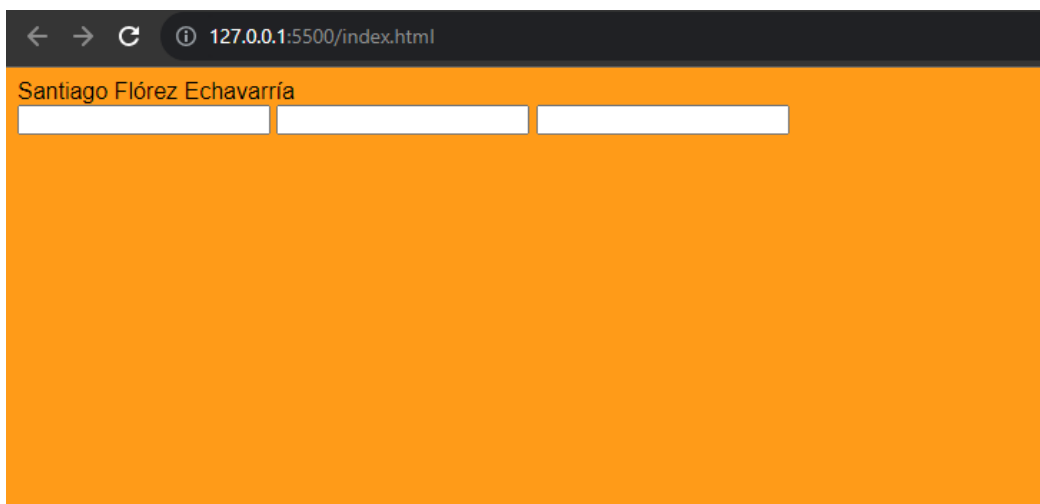
Paso 33: Ahora en el archivo index.html vamos a vincular el dentro del head “css” que acabamos de crear (Línea 8)



Paso 34: Ahora entraremos en el archivo “style.css” y vamos a crear un diseño para el body, poniéndole una fuente (línea 3) y un color al background (línea 4)



Paso 35: Ahora vemos cómo va quedando.

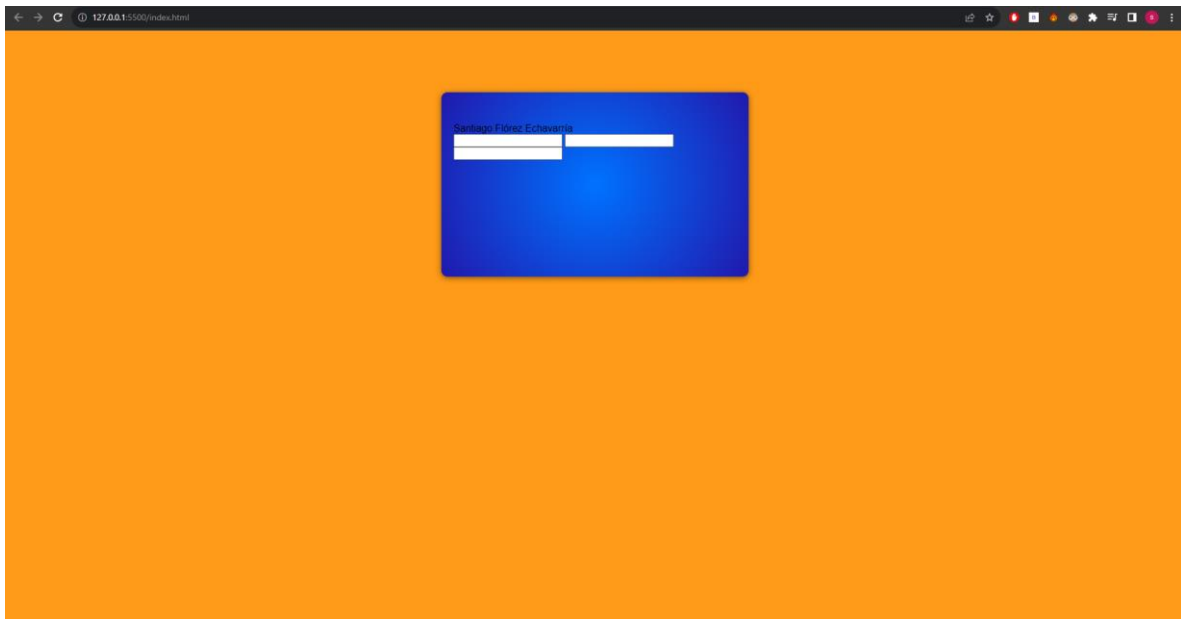


Paso 36: Ahora crearemos el estilo de la tarjeta.

Con cosas como el border-radius, background, y demás.

```
7  /* Estilos para la tarjeta */
8  .card {
9      width: 500px; /* Ancho de la tarjeta */
10     height: 300px; /* Altura de la tarjeta */
11     border-radius: 10px; /* Bordes redondeados */
12     padding: 50px 20px 0 20px; /* Relleno interno */
13     box-sizing: border-box; /* Modelo de caja */
14     background: linear-gradient(to top right,
15         rgba(0, 114, 255, 1) 1%,
16         rgba(33, 25, 173, 1) 100%); /* Fondo degradado */
17     }
18
19
20 }
```


Paso 39: Así nos queda luego de centrar y ponerle sombra.

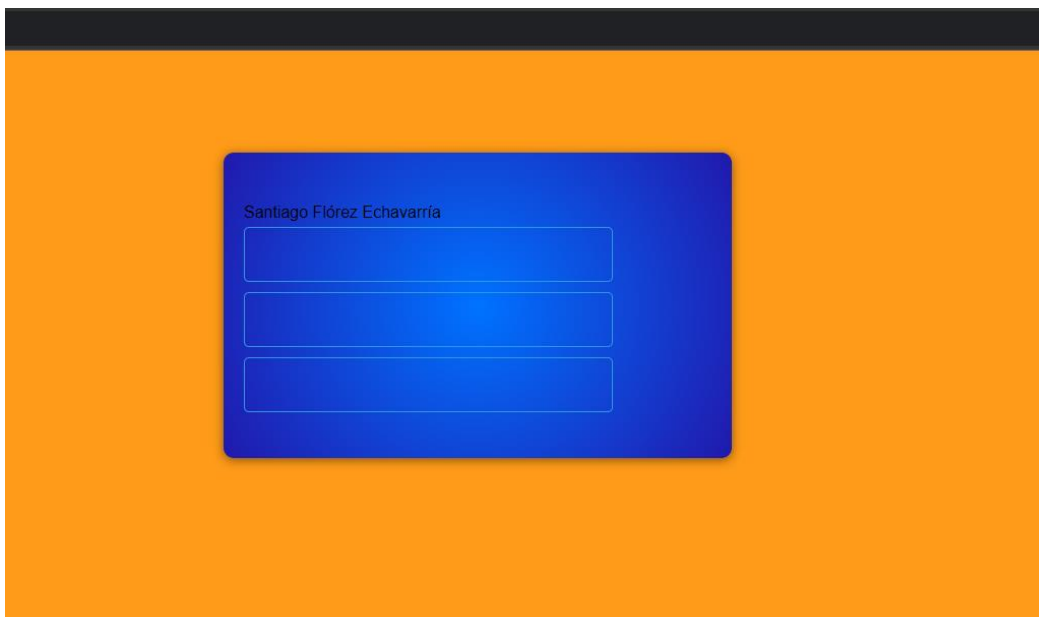


Paso 40: Ahora vamos a cambiar el diseño de los inputs dentro de la tarjeta.

Poniendo algunos colores, bordes y demás.

```
24  /* Estilos para los campos de entrada dentro de la tarjeta */
25  .card input {
26      font-size: 28px; /* Tamaño de fuente */
27      padding: 10px; /* Espacio interno */
28      border: solid 1px #45, 154, 226; /* Borde sólido con color */
29      color: white; /* Color del texto */
30      border-radius: 5px; /* Bordes redondeados */
31      background-color: transparent; /* Fondo transparente */
32      margin: 5px 0; /* Margen vertical */
33      outline: none; /* Elimina el contorno al hacer clic */
34  }
```

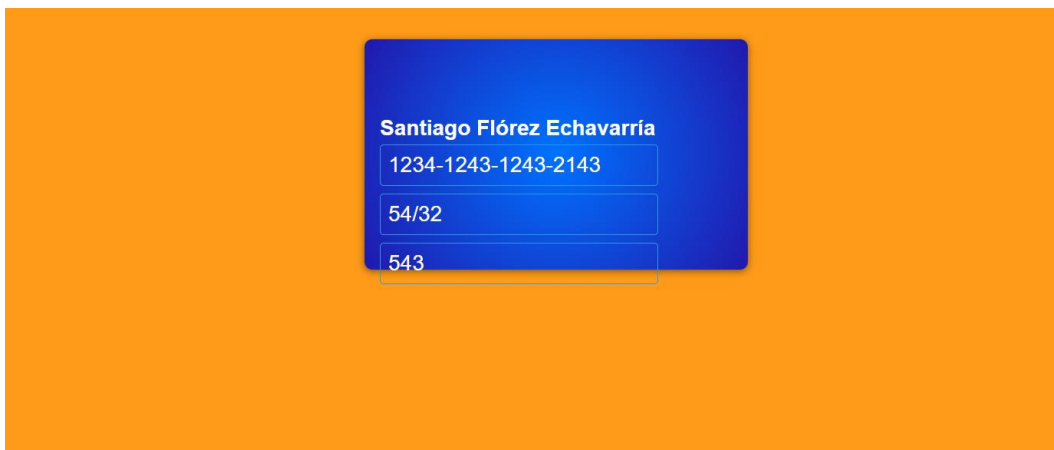

Paso 41: Ahora se nos verá algo así.



Paso 42: Ahora procederemos a cambiar el diseño del nombre del portador de la tarjeta.

```
35
36  /* Estilos para el elemento con clase "holder" */
37  .holder {
38      color: white; /* Color del texto */
39      font-weight: bold; /* Grosor de la fuente */
40      font-size: 28px; /* Tamaño de fuente */
41      padding-top: 50px; /* Espacio superior interno */
42  }
43
44
```

Paso 43: Ahora podemos ver como nos queda el nombre.



Paso 44: Ahora procedemos a cambiar el tamaño de los inputs donde va la fecha y la CVV

```
44  /* Estilos para el campo de fecha de la tarjeta */
45  .card #inputDate {
46  |    width: 70px; /* Ancho del campo */
47  |  }
48
49  /* Estilos para el campo de CVV de la tarjeta */
50  .card #inputCVV {
51  |    width: 50px; /* Ancho del campo */
52  |  }
```

Paso 45: Apreciamos el final de nuestro proyecto.

