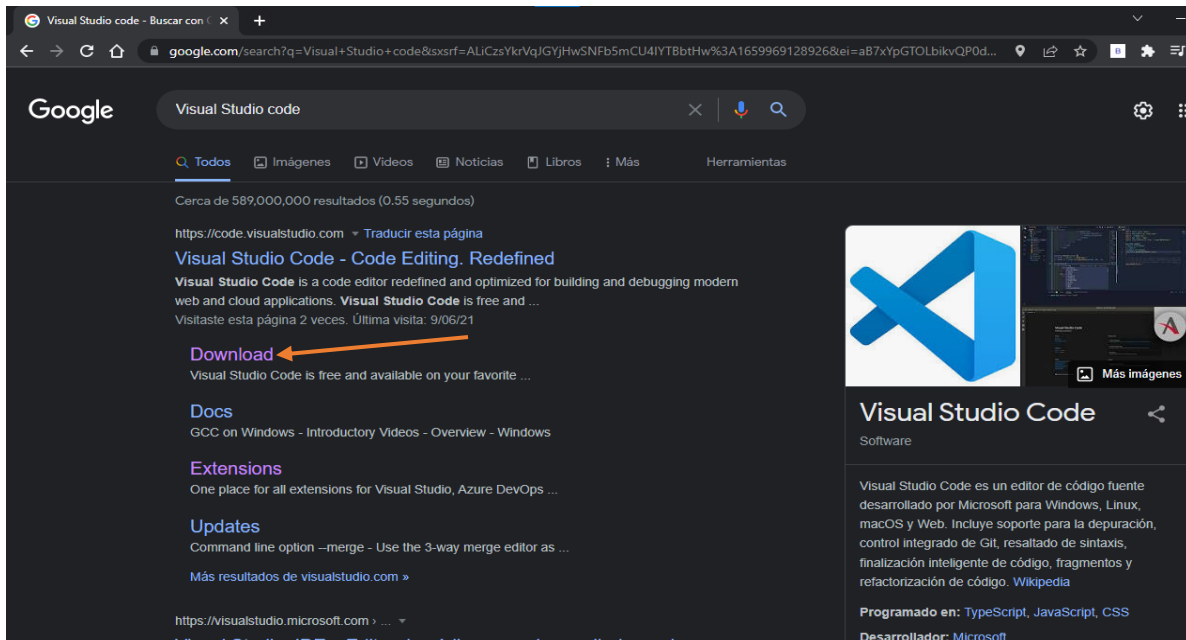
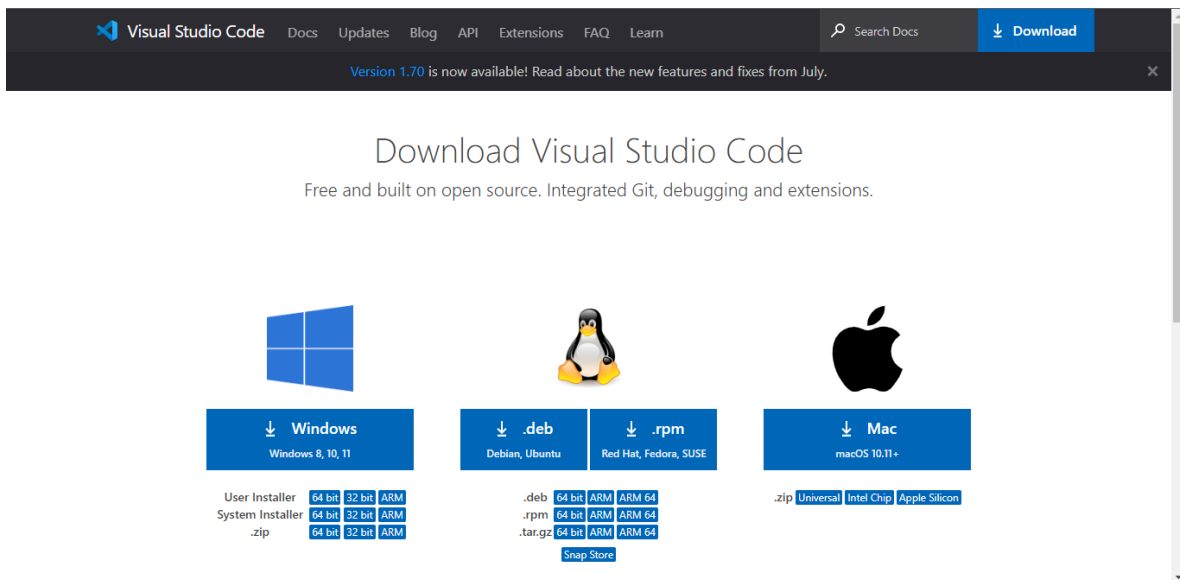


## Proceso instalación Visual Studio Code

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



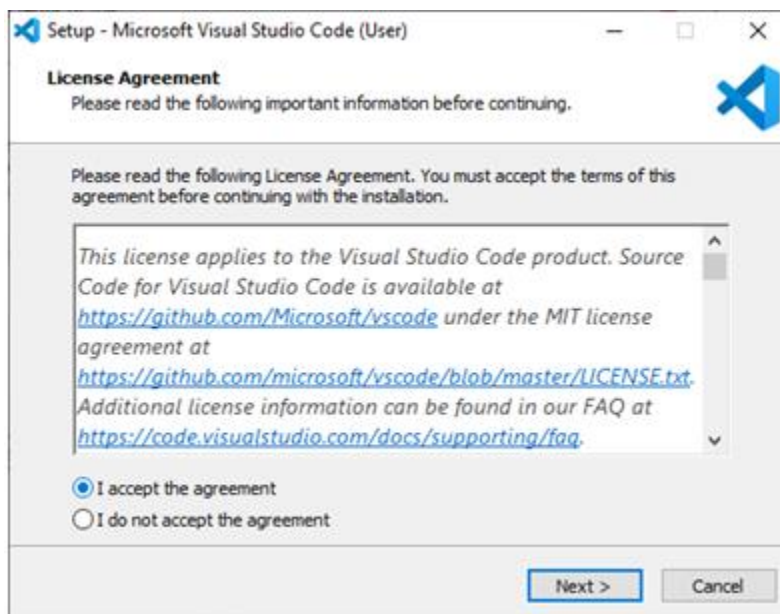
**Paso 2:** Seleccionamos el sistema operativo que tenemos y lo descargamos.



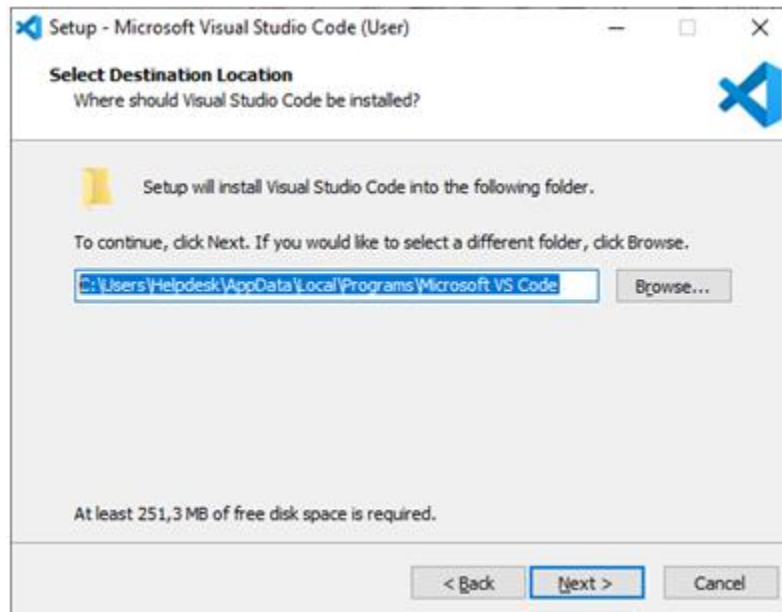
**Paso 3:** Al darle clic nos descargará un .exe, al cual le daremos clic encima.



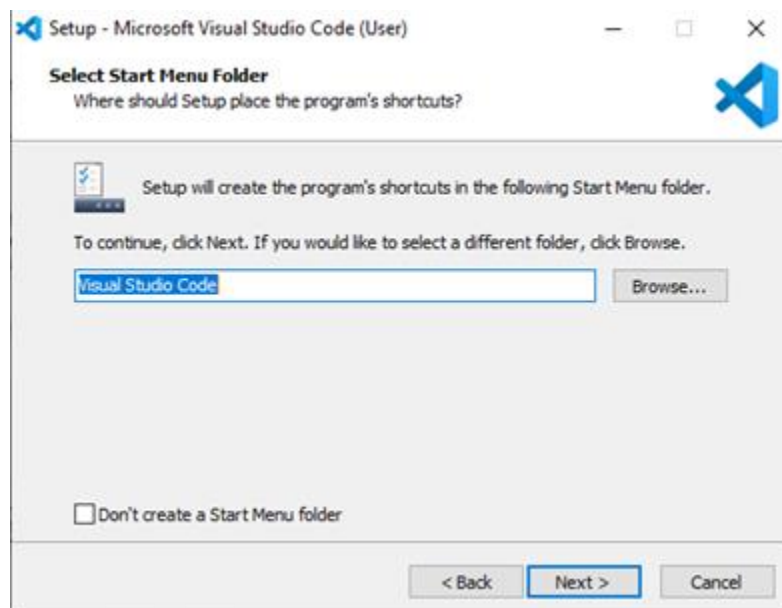
**Paso 4:** Lee y acepta el acuerdo de licencia. Haz clic en Next para continuar.



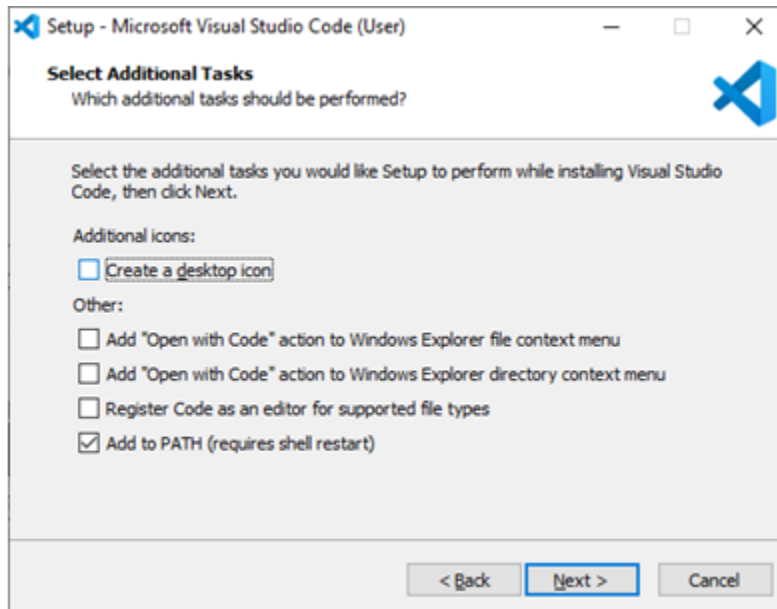
**Paso 5:** Puedes cambiar la ubicación de la carpeta de instalación o mantener la configuración predeterminada. Haz clic en Next para continuar.



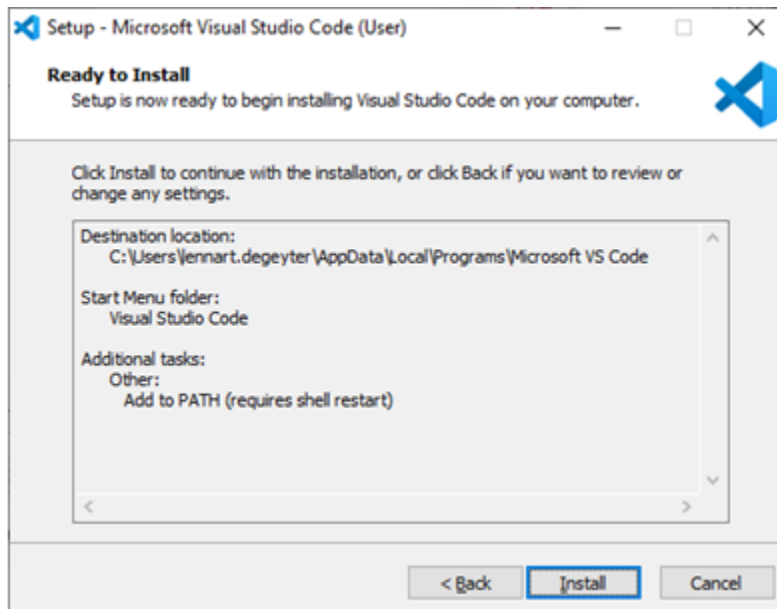
**Paso 6:** Elige si deseas cambiar el nombre de la carpeta de accesos directos en el menú Inicio o si no deseas instalar accesos directos en absoluto. Haz clic en Next.



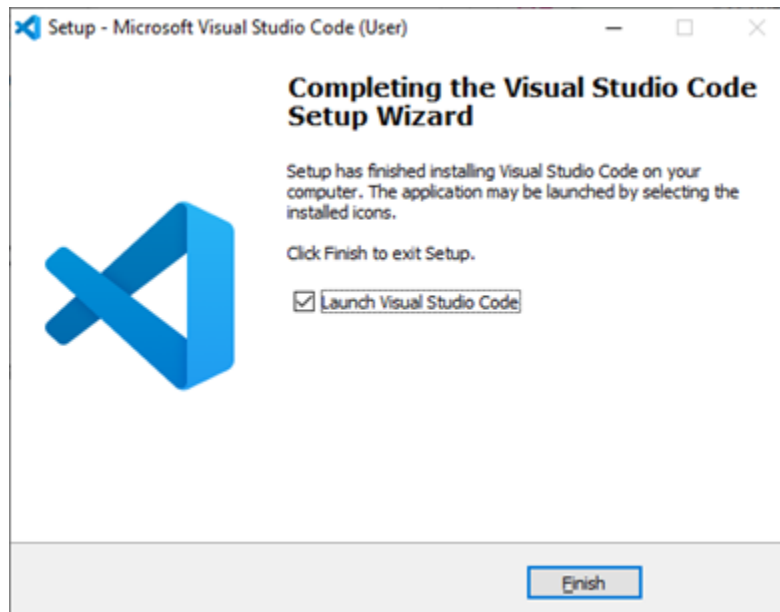
**Paso 7:** Selecciona las tareas adicionales, por ej. crear un icono en el escritorio o añadir opciones al menú contextual de Windows Explorer. Haz clic en Next.



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

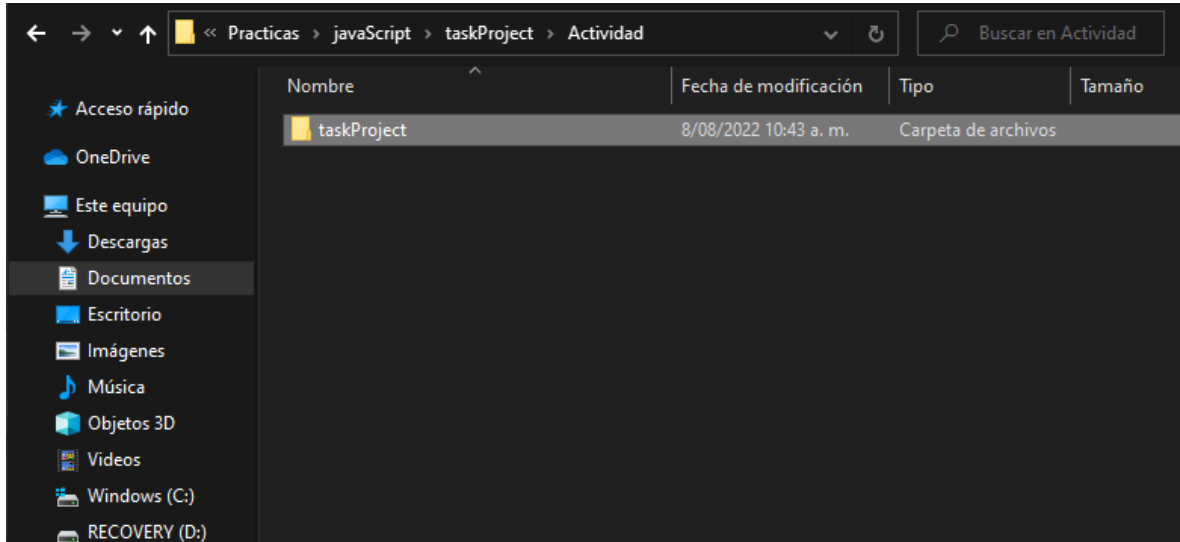


**Paso 9:** El programa está instalado y listo para usar. Haz clic en Finish para finalizar la instalación y lanzar el programa.

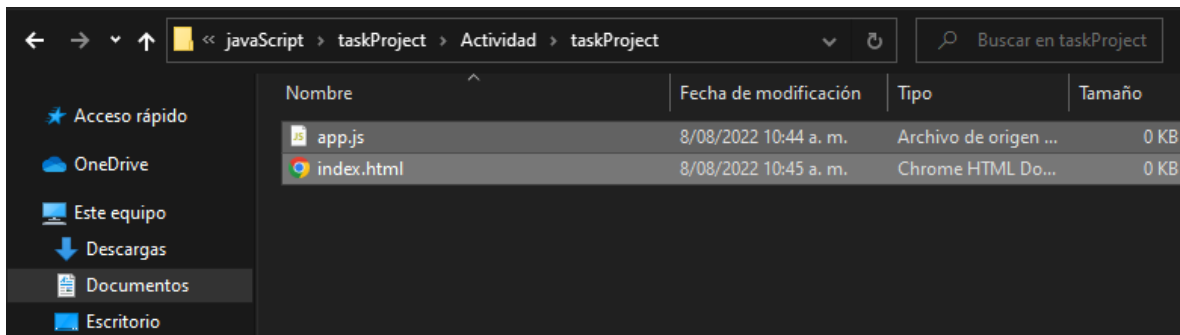


## Creación de archivos

Creamos una carpeta raíz. En nuestro caso la llamaremos “taskProject”.

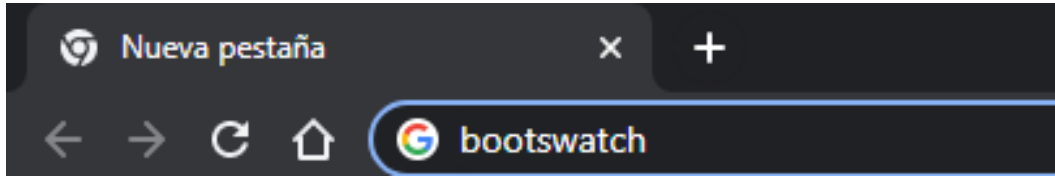


Dentro de esta, crearemos dos archivos. Uno llamado index.html y el otro llamado app.js (Es importante tener exactamente esa extensión en el archivo).

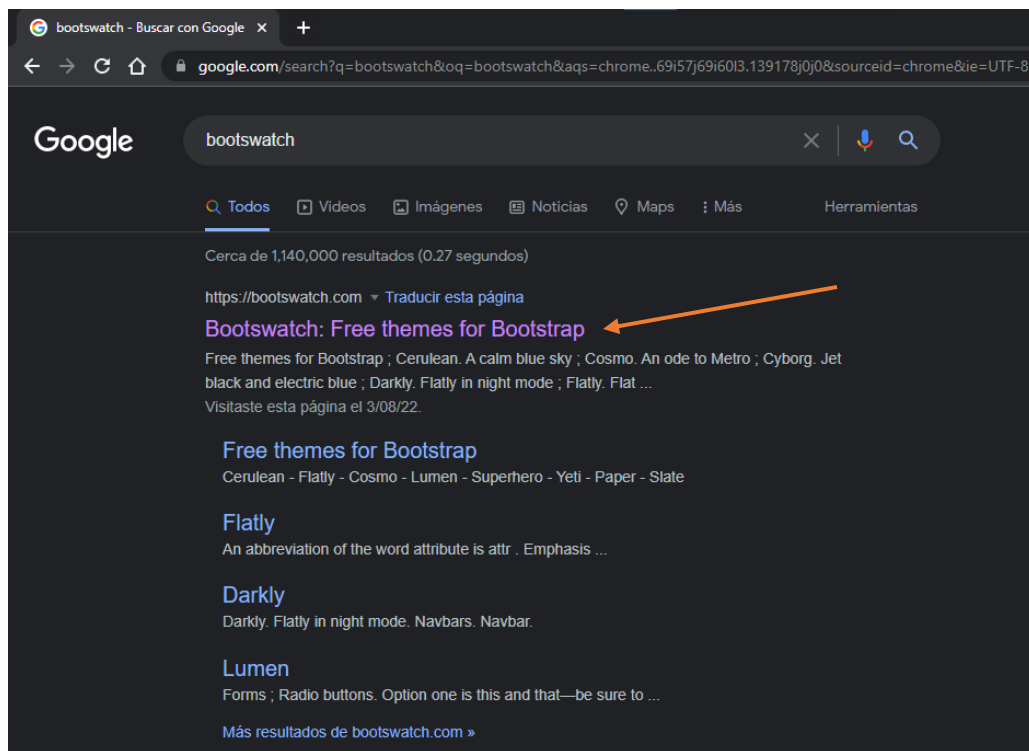


## Project – Task Product

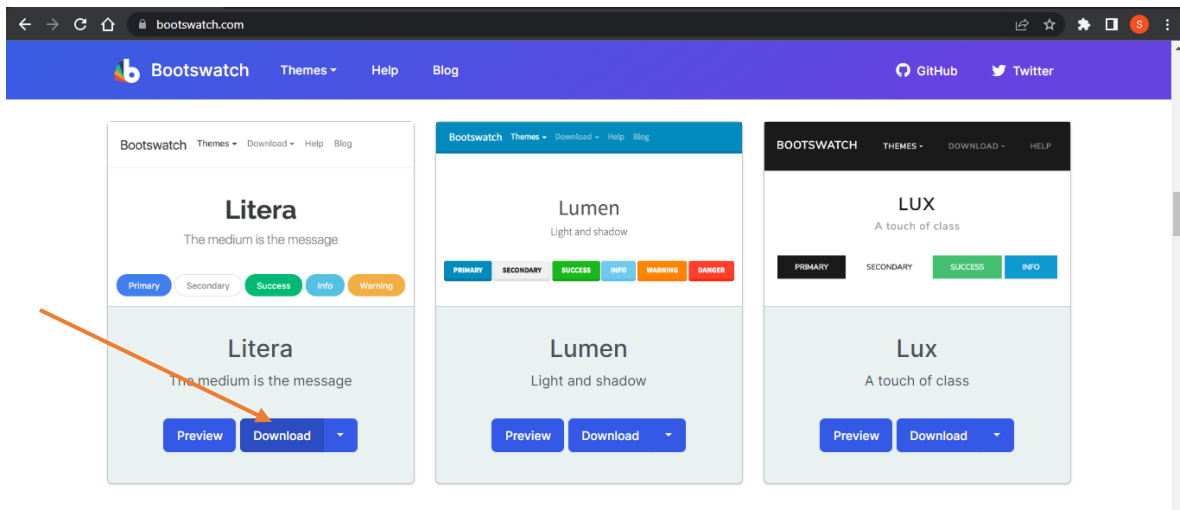
Empezamos ingresando bootswatch en el buscador de Google.



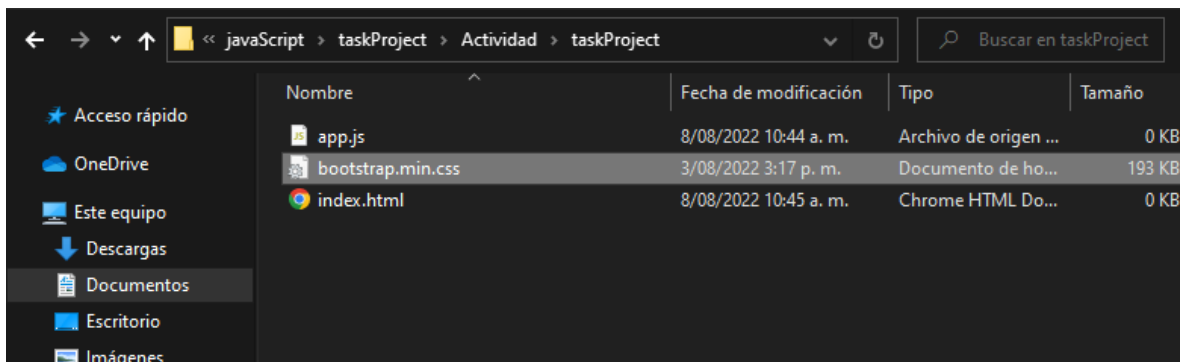
Luego de esto, se selecciona el primer resultado que nos arroja Google.



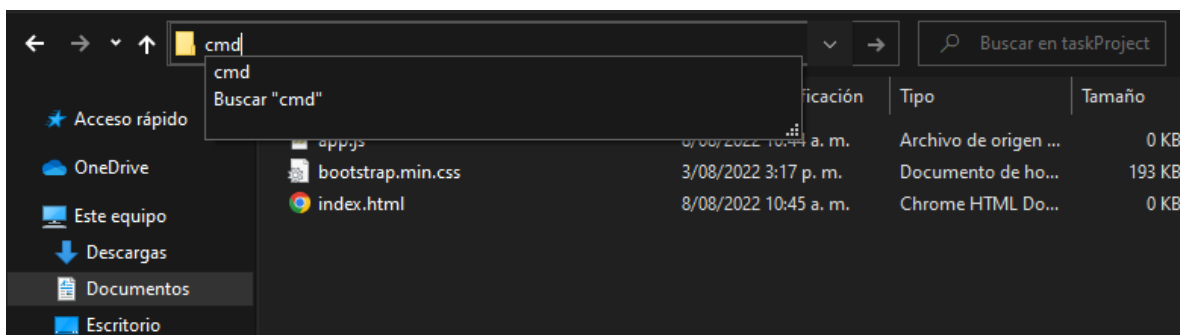
Entramos, y luego bajamos hasta que nos encontremos uno que diga “Litera” y le damos en Download.



Eso nos descargará un archivo llamado Bootstrap.min.css, este archivo lo moveremos a la carpeta anteriormente creada.



Para abrir el Visual Studio Code, haremos lo siguiente: En la barra superior, escribiremos cmd y le damos enter.



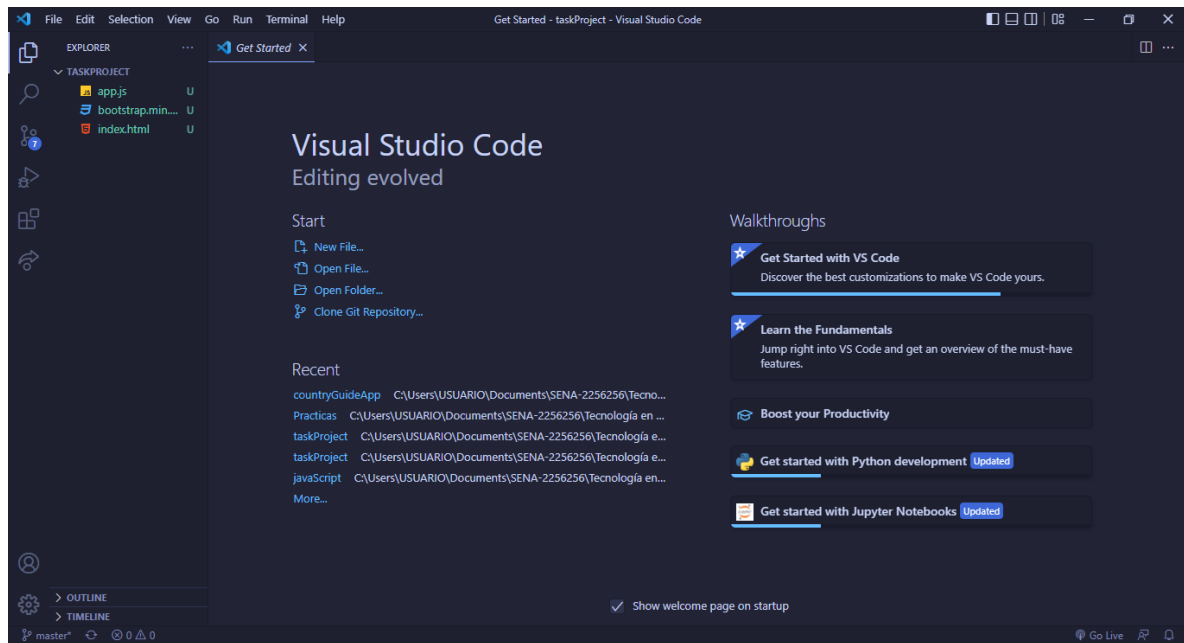


Este comando, nos abrirá una terminal, ahí dentro, solo tendremos que copiar “code .” y posteriormente darle enter, esto nos abrirá el Visual Studio Code.

```
Seleccionar C:\Windows\System32\cmd.exe
Microsoft Windows [Versión 10.0.19043.1826]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\USUARIO\Documents\SENA-2256256\Tecnología en ADSI\Practicas\javascript\taskProject\Actividad\taskProject>code .
```

Esto nos abrirá como anteriormente dije, el Visual Studio Code.



Ingresamos a index.html y codificamos lo básico: html, head y body.

```
index.html x app.js
index.html > html > body
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title>Product App</title>
8   </head>
9   <body>
10
11 </body>
12 </html>
```

Creamos la navegación.

```
11     <body>
12         <!-- Barra de navegacion -->
13         <nav class="navbar navbar-light bg-light">
14             <a href="/" class="navbar-brand">
15                 Product App
16             </a>
17         </nav>
18
```

Creamos el contenedor, donde estará el formulario.

```
19     <div class="container">
20         <!-- Esta es la aplicacion en si -->
21         <div id="App" class="row pt-5">
22             <div class="col-md-4">
23                 <div class="card">
24                     <div class="card-header">
25                         <h4>Add A Product</h4>
26                     </div>
27                     <!-- Formulario -->
28                     <form id="product-form" class="card-body">
29                         <!-- Div e input para ingresar el nombre del producto -->
30                         <div class="form-group">
31                             <input class="form-control" type="text" id="name" placeholder="Product Name">
32                         </div>
33                         <br>
34                         <!-- Div e input para ingresar el precio del producto -->
35                         <div class="form-group">
36                             <input class="form-control" type="number" step="0.01" id="price" placeholder="Product Price">
37                         </div>
38                         <br>
39                         <!-- Div e input para ingresar el año del producto -->
40                         <div class="form-group">
41                             <input class="form-control" type="number" id="year" min="1900" value="2022" placeholder="Product Year">
42                         </div>
43                         <br>
44                         <!-- Div e input para el boton de guardar -->
45                         <div class="d-grid gap-2">
46                             <input class="btn btn-primary btn-block" type="submit" value="Save">
47                         </div>
48                     </form>
49                 </div>
50             </div>
51             <!-- Contenedor de la lista de productos -->
52             <div id="product-list" class="col-md-8">
53             </div>
54         </div>
55     </div>
56
```

Y por último, importamos el JavaScript.

```
57         <!-- Se importa el javascript -->
58         <script src="app.js"></script>
59     </body>
60 </html>
```

Ahora empezaremos a codificar en JavaScript, abrimos el archivo app.js y creamos la clase Product, luego creamos un método constructor que requiera los parámetros name, price, year. Esta clase nos ayudará para definir los campos necesarios de producto.

```
1 //Clase para definir los campos de producto
2 class Product {
3     //Se crea un metodo constructor y se requieren el nombre, precio y año
4     constructor(name, price, year) {
5         //Dejamos definidas las variables
6         this.name = name;
7         this.price = price;
8         this.year = year;
9     }
10 }
```

Luego de realizar esto, creamos otra clase llama UI (User Interface), donde irá todo lo que el usuario verá. Dentro de esta, creamos un método llamado “addProduct”, este nos servirá para añadir un producto y mostrarlo en la interfaz.

```
12 //Clase User Interface (Para la vista del usuario)
13 class UI {
14     //Metodo de añadir producto, y le enviamos un parametro (product)
15     addProduct(product) {
16         //Obtenemos la lista del producto mediante el id
17         const productList = document.getElementById('product-list');
18         //Creamos un elemento (div)
19         const element = document.createElement('div');
20         //Dentro del elemento estara el nombre, precio y año
21         element.innerHTML = `
22             <div class="card text-center mb-4">
23                 <div class="card-body">
24                     <strong>Product Name</strong>: ${product.name};
25                     <strong>Product Price</strong>: ${product.price};
26                     <strong>Product Year</strong>: ${product.year};
27                     <a href="#" class="btn btn-danger" name="delete">Delete</a>
28                 </div>
29             </div>
30         `;
31
32         //Ponemos el elemento como un hijo de la lista de productos
33         productList.appendChild(element);
34         //Llamamos el metodo para resetear el formulario
35         this.resetForm();
36     }
37 }
```

Creamos también, otro método llamado “resetProduct”, este nos servirá para resetear el formulario y borrarle los datos anteriores cada vez que ingrese un nuevo producto.

```
38 //Metodo para resetear el formulario
39 resetForm() {
40     //Llamo el el formulario por medio del id
41     document.getElementById('product-form').reset();
42 }
43 }
```

Creamos otro método llamado “deleteProduct”, este como su nombre lo dice, servirá para eliminar el producto ingresado y nos mostrará un mensaje.

```
44 //Metodo para eliminar producto y le envio un elemento
45 deleteProduct(element) {
46     if (element.name == 'delete') {
47         //parentElement es para coger el elemento padre del anterior
48         element.parentElement.parentElement.parentElement.remove();
49         //Se muestra un mensaje de confirmacion
50         this.showMessage('Product Deleted Successfully', 'info');
51     }
52 }
53 }
```

Creamos el último método dentro de la clase UI, llamado “showMessage”, este servirá para mostrar mensajes en la interfaz y de igual manera definimos dentro de este, la posición donde se mostrará y el tiempo que durará.

```
54 //Metodo para crear mensajes
55 showMessage(message, cssClass) {
56     //Se crea una constante para crear posteriormente un elemento (div)
57     const div = document.createElement('div');
58     //Se le asignan clases al div creado anteriormente
59     div.className = `alert alert-${cssClass} mt-2`;
60     div.appendChild(document.createTextNode(message));
61     //Mostrando en el DOM
62     //Se crea una constante que me trae todo lo que haya dentro del container
63     const container = document.querySelector('.container');
64     //Se crea una constante que me trae todo lo que haya dentro de app
65     const app = document.querySelector('#App');
66     container.insertBefore(div, app);
67     //Se define el tiempo que se muestre el aviso
68     setTimeout(function() {
69         document.querySelector('.alert').remove();
70     }, 3000);
71 }
72 }
```

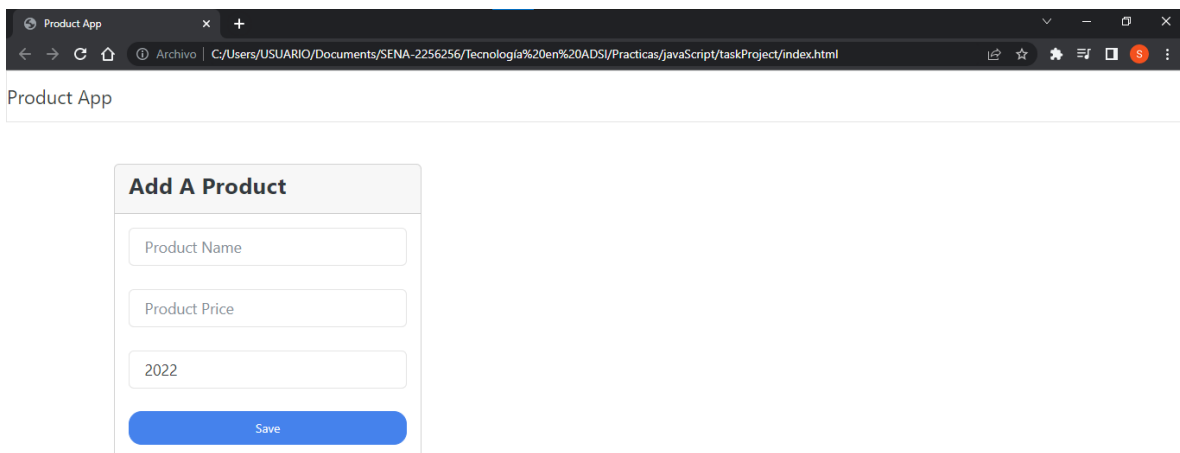
Luego empezamos a crear los DOM (Document Object Model) o los eventos del aplicativo. Traemos el elemento product-form mediante el id y escuchamos el evento realizado, el cual es submit, y luego insertamos un nuevo producto, llamando la clase Product y UI.

```
73  /*DOM EVENTS (Document Object Model)
74  Eventos del aplicativo
75  Se llama el formulario*/
76  document.getElementById('product-form')
77  //Con esta funcion se traen los valores ingresados
78  .addEventListener('submit', function(e) {
79      const name = document.getElementById('name').value;
80      const price = document.getElementById('price').value;
81      const year = document.getElementById('year').value;
82
83      //Se crea un nuevo producto y se le envian los parametros necesarios
84      const product = new Product(name,price,year);
85
86      //Se crea una nueva UI
87      const ui = new UI();
88      //Se evalua si los campos están vacios
89      if (name === '' || price === '' || year === '') {
90          //Si estan vacios, retorna un mensaje
91          return ui.showMessage('Complete Fields', 'danger');
92      }
93
94      //Se añade el producto
95      ui.addProduct(product);
96      //Mensaje si se añade el producto
97      ui.showMessage('Product Added Successfully','success');
98
99      //Esta funcion es para que no se recargue al enviar el formulario
100     e.preventDefault();
101 });
```

Para terminar, traemos el elemento product-list y escuchamos el evento, el cual debe ser click y luego se crea un nuevo UI y llamamos el método para eliminar el producto.

```
104 //Se llama a la lista de productos
105 document.getElementById('product-list')
106 //Se escucha el evento click
107 .addEventListener('click', function(e) {
108     //Se crea una nueva UI
109     const ui = new UI();
110     /*Se llama el metodo para eliminar producto
111     Y se le envian los datos del event*/
112     ui.deleteProduct(e.target);
113 });
```

De esta manera, deberíamos tener un resultado parecido al siguiente:



The screenshot shows a web browser window titled 'Product App'. The address bar displays the file path: C:/Users/USUARIO/Documents/SENA-2256256/Tecnologia%20en%20ADS/Practicas/javascript/taskProject/index.html. Below the browser window, the text 'Product App' is visible. The main content area contains a form titled 'Add A Product'. The form has three input fields: 'Product Name', 'Product Price', and a field containing the year '2022'. At the bottom of the form is a blue button labeled 'Save'.

Product Added Successfully

### Add A Product

Save

Product Name: Apple; Product Price: 1.13; Product Year: 2022;

Delete

Product Name: Orange; Product Price: 123; Product Year: 2022;

Delete

Product Deleted Successfully

### Add A Product

Save

Product Name: Orange; Product Price: 1.17; Product Year: 2022;

Delete