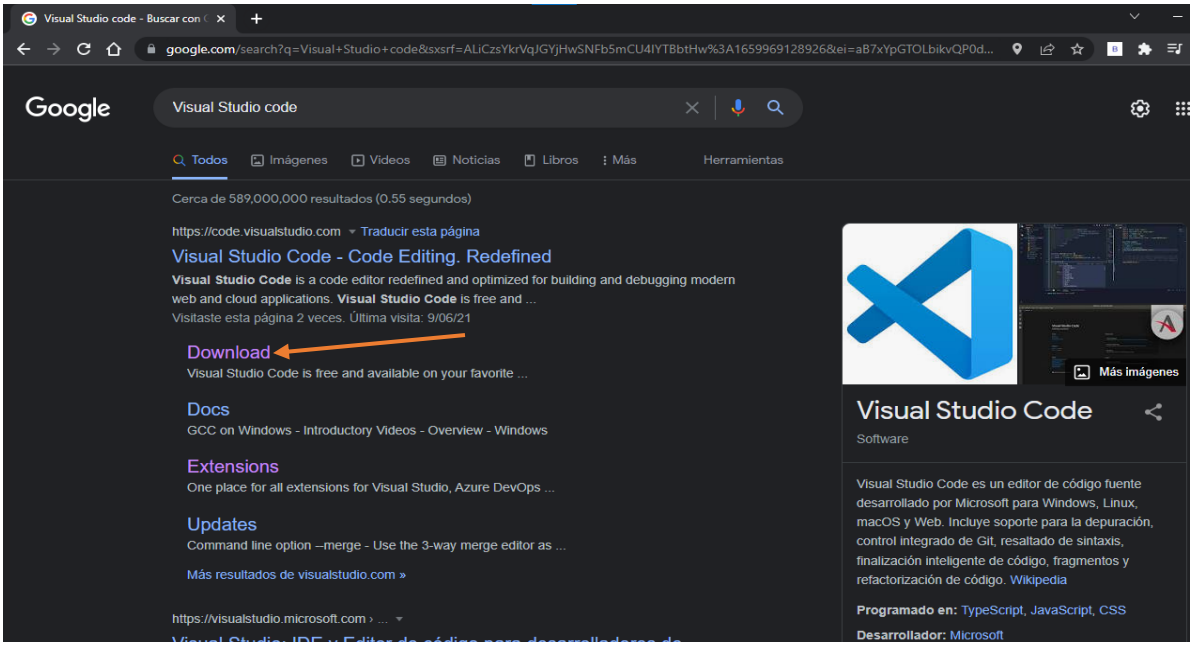
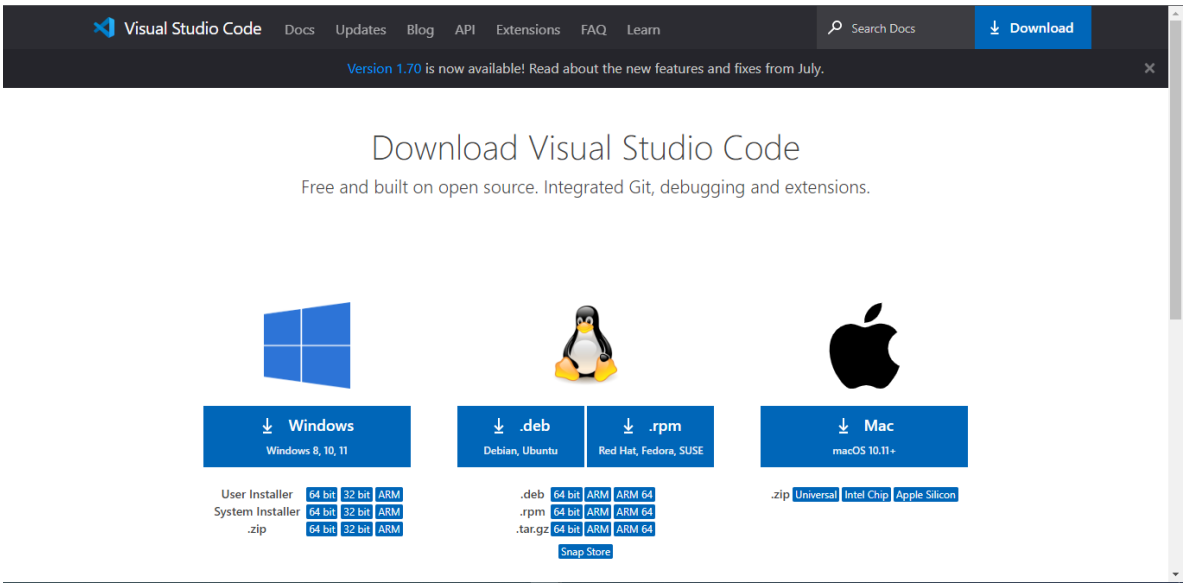
Proceso instalación Visual Studio Code

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

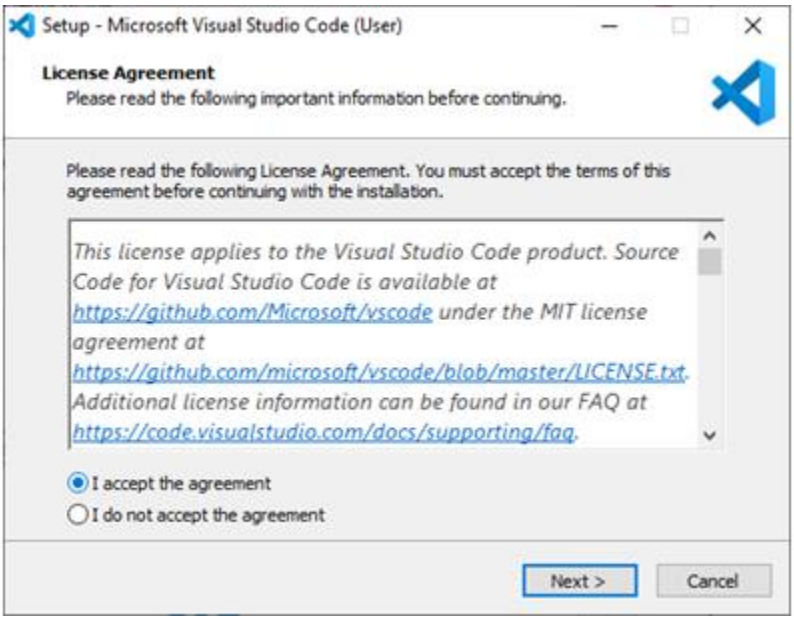


Paso 2: Seleccionamos el sistema operativo que tenemos y lo descargamos.

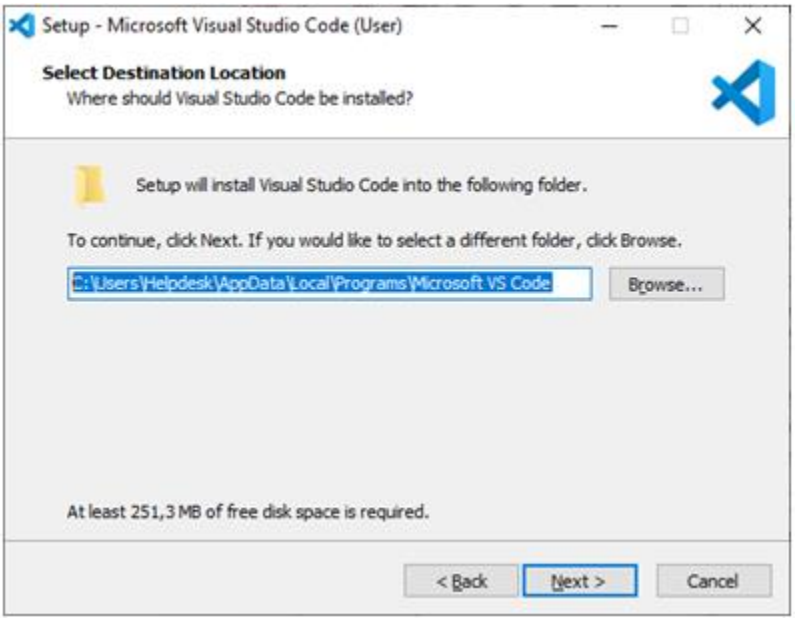


Paso 3: Al darle clic nos descargará un .exe, al cual le daremos clic encima.

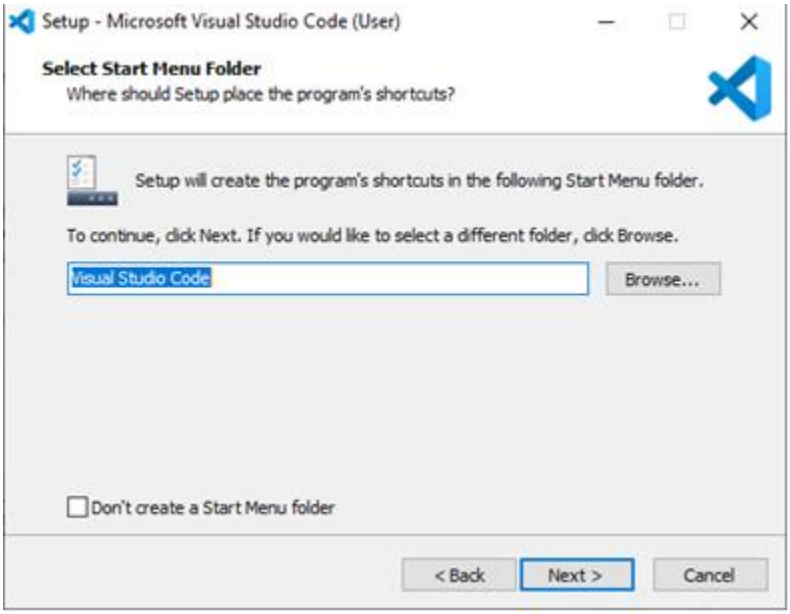
Paso 4: Lee y acepta el acuerdo de licencia. Haz clic en Next para continuar.



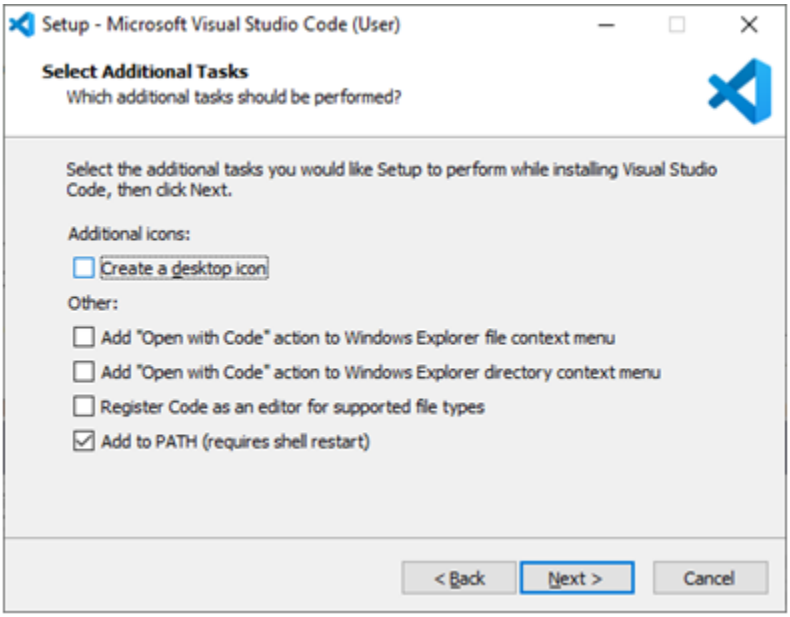
Paso 5: Puedes cambiar la ubicación de la carpeta de instalación o mantener la configuración predeterminada. Haz clic en Next para continuar.



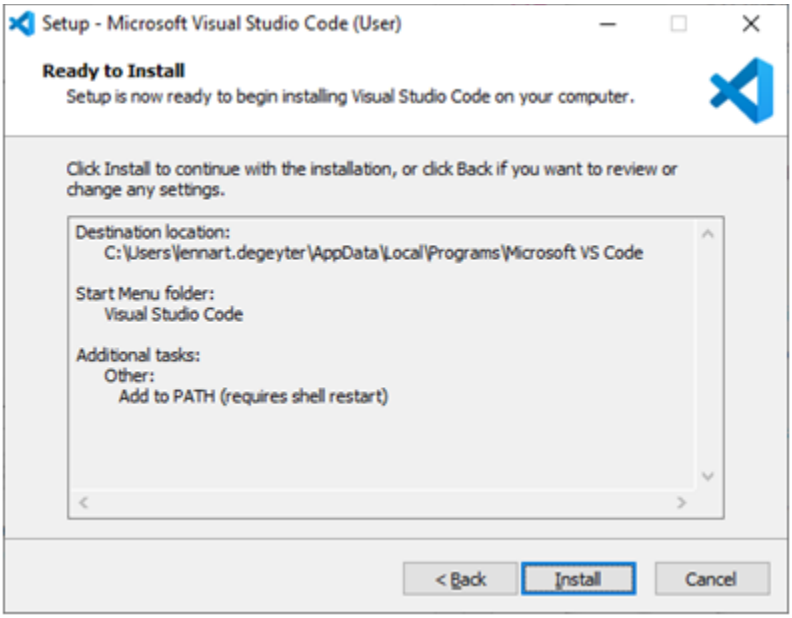
Paso 6: Elige si deseas cambiar el nombre de la carpeta de accesos directos en el menú Inicio o si no deseas instalar accesos directos en absoluto. Haz clic en Next.



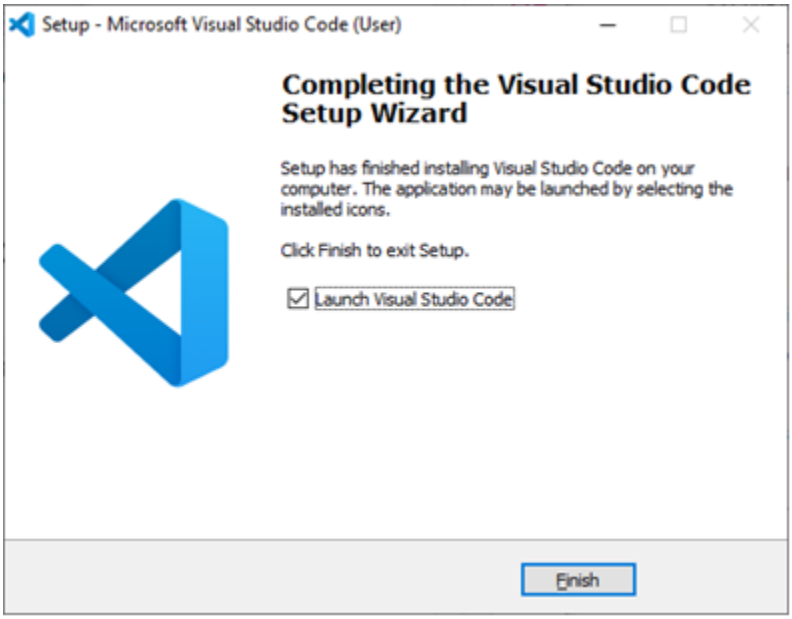
Paso 7: Selecciona las tareas adicionales, por ej. crear un icono en el escritorio o añadir opciones al menú contextual de Windows Explorer. Haz clic en Next.



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

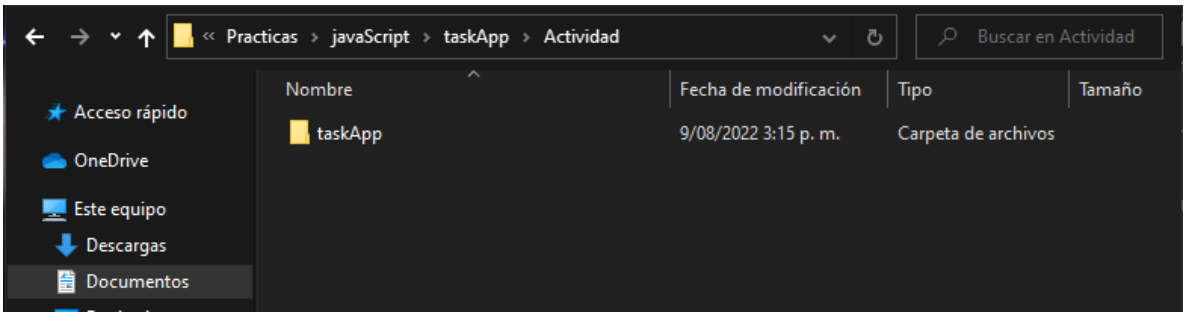


Paso 9: El programa está instalado y listo para usar. Haz clic en Finish para finalizar la instalación y lanzar el programa.

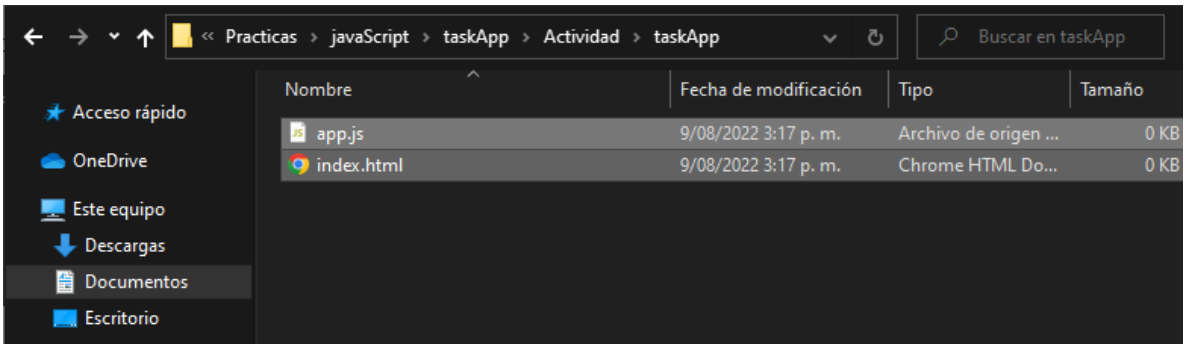


Creación de carpetas

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



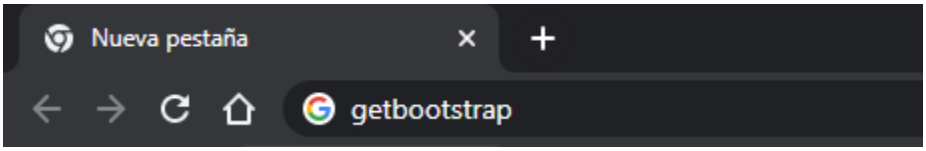
Dentro de esta, crearemos dos archivos. Uno llamado index.html y el segundo llamado app.js (Es importante tener exactamente esa extensión en el archivo).



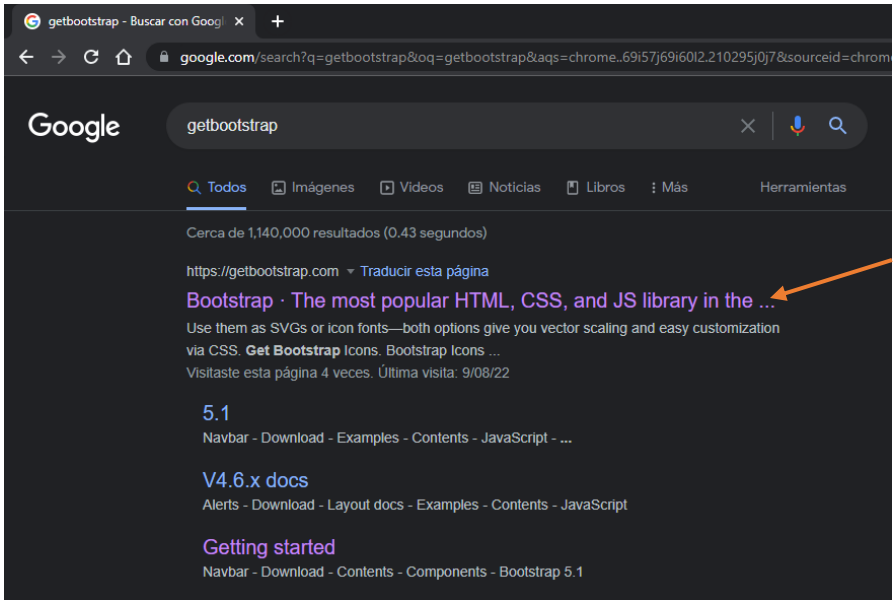
Task App – Project

En el archivo index.html empezaremos creando el html raíz.

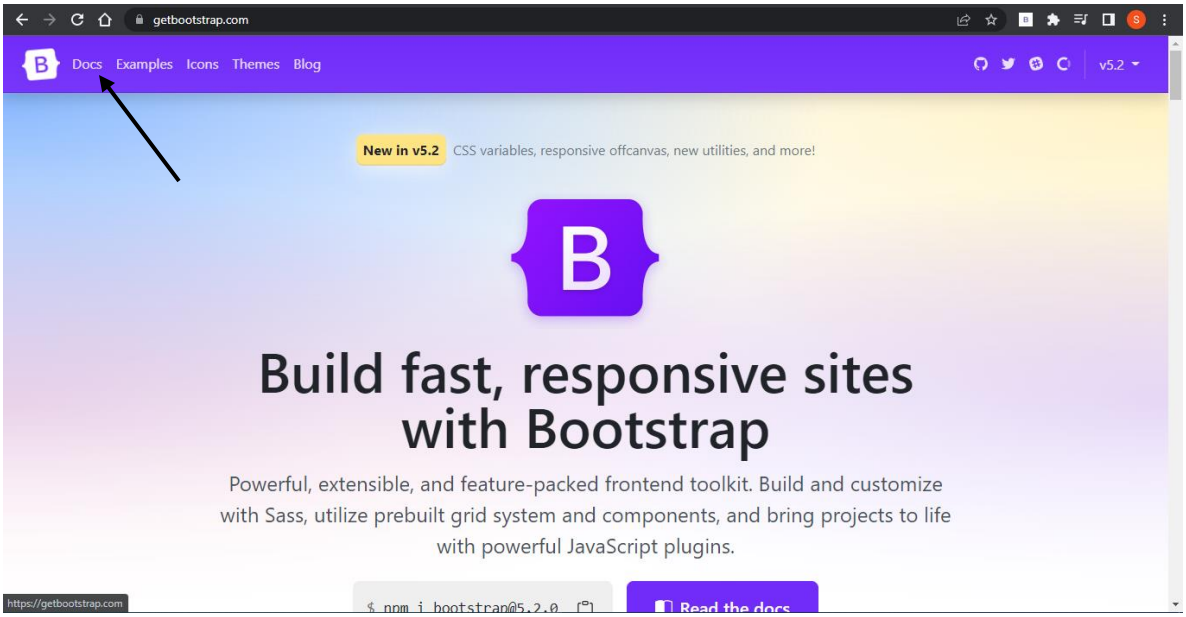
Ahora, vamos a Google para buscar “getbootstrap”.



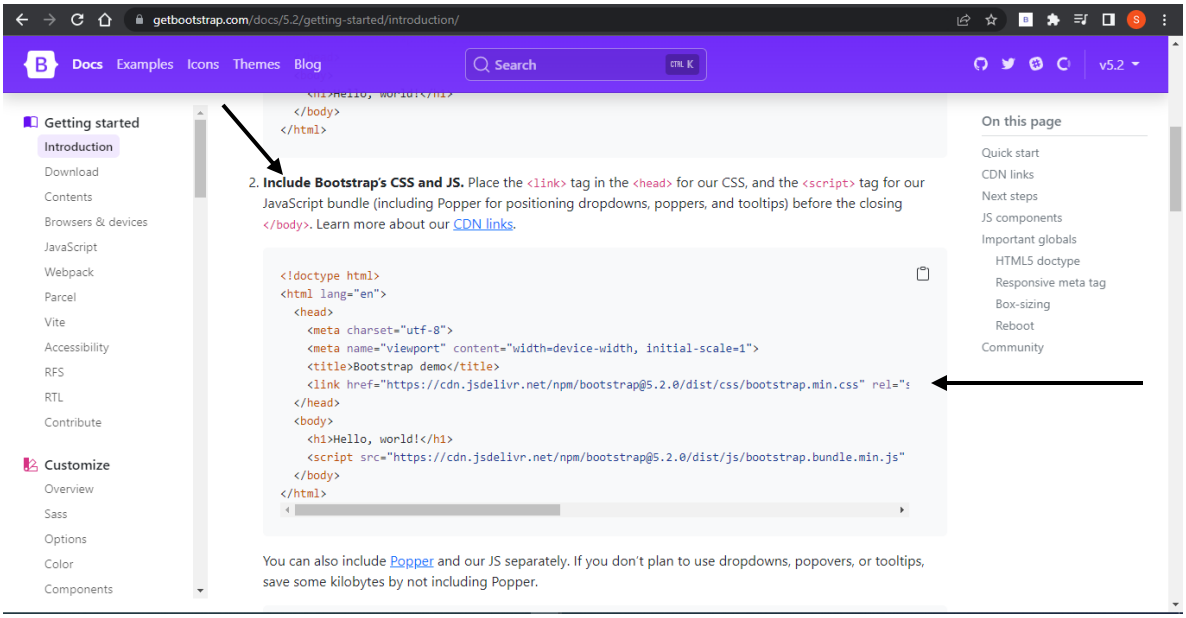
Lo buscamos, ahora le daremos al primer resultado que nos arroja Google.



Dentro de esta página, procederemos a darle donde dice “Docs” en la barra de navegación de esta.



Ahora podremos ver toda la documentación de bootstrap5, lo único que tendremos que hacer en estos momentos es bajar un poco hasta encontrar lo siguiente:



Para terminar, lo único que tendremos que hacer es copiar la línea de código que está debajo de “title”, exactamente esta:

```html
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Bootstrap demo</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/css/bootstrap.min.css" rel="s
  </head>
  <body>
    <h1>Hello, world!</h1>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/js/bootstrap.bundle.min.js"
  </body>
</html>
```

Esta línea la copiamos y pegamos exactamente en el mismo lugar, solo que ahora en nuestro html. De igual manera linkeamos el javascript en nuestro html (El link sigue hacia la derecha, por eso es importante cogerlo de la página antes dicha).

```html
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>Javascript Task App</title>
        <!-- Linkeamos bootstrap para el front-end -->
        <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/css/bootstrap.min.css" rel="stylesheet"
    </head>
    <body>

        <!-- Enlazamos el javascript -->
        <script src="app.js"></script>
    </body>
</html>
```

Creamos la barra de navegación dentro del body.

```
    <body>
        <!-- Barra de navegacion -->
        <nav class="navbar navbar-light bg-light">
            <div class="container">
                <a href="#" class="navbar-brand">Task App</a>
            </div>
        </nav>

        <!-- Enlazamos el javascript -->
        <script src="app.js"></script>
    </body>
</html>
```

Ahora crearemos un contenedor principal, donde estará todo el cuerpo (container), luego separamos la página en dos (col-md-4, col-md-8), Bootstrap separa en 12 columnas el cuerpo de la página, lo que hacemos es coger 4 para el formulario donde el usuario va a ingresar la información y los otros 8 los tomamos para mostrar la información.

```
    <body>
        <!-- Barra de navegacion -->
        <nav class="navbar navbar-light bg-light">
            <div class="container">
                <a href="#" class="navbar-brand">Task App</a>
            </div>
        </nav>

        <!-- Contenido de la pagina -->
        <div class="container">
            <div class="row pt-5">
                <!-- Para la tabla de ingreso de informacion -->
                <div class="col-md-4">

                </div>
                <!-- Para mostrar la informacion -->
                <div class="col-md-8">

                </div>
            </div>
        </div>

        <!-- Enlazamos el javascript -->
        <script src="app.js"></script>
    </body>
```

Ahora, crearemos el formulario donde el usuario va a digitar la información.

```
<!-- Para la tabla de ingreso de informacion -->
<div class="col-md-4">
    <div class="card">
        <div class="card-body">
            <!-- Formulario de ingreso de informacion -->
            <form action="" id="formTask">
                <!-- div para el titulo de la tarea -->
                <div class="form-group">
                    <input type="text" id="title" placeholder="Add a to do" class="form-control">
                </div>
                <!-- div para agregar la descipcion de la tarea -->
                <div class="form-group pt-3">
                    <textarea id="description" cols="80" rows="8"
                    class="form-control" placeholder="Add a description"></textarea>
                </div>
                <!-- div para el boton de guardar -->
                <div class="d-grid gap-2 pt-3">
                    <button type="submit" class="btn btn-primary btn-block">
                        Save
                    </button>
                </div>
            </form>
        </div>
    </div>
</div>
```

Dentro de la otra sección (col-md-8) solo pondremos un div, pero esto es porque con este es el que vamos a interactuar en js, porque ahí es donde mostraremos y borraremos la información por medio del LocalStorage.

```
<!-- Para mostrar la informacion -->
<div class="col-md-8">
    <!-- Con este div interactuamos en js -->
    <div id="tasks"></div>
</div>
```

Hemos terminado de codificar en html, ahora vamos al archivo app.js para empezar a crear la interacción. Empezamos escuchando el evento del formulario, que tendrá que ser submit y cuando eso suceda se correrá la función saveTask.

```
// Escuchamos el evento del formulario
document.getElementById('formTask').addEventListener('submit', saveTask);
```

Creamos la función saveTask, esta nos servirá para guardar nuevas tareas. Para hacer esto utilizamos la API de localStorage, donde almacenaremos todos los datos que el usuario ingrese. ¿Para qué es necesario? Es necesario ya que si sólo utilizamos javascript, al recargar la página, se pierde toda la información, en cambio, utilizando esta API, todo permanecerá alojado localmente dentro de la página.

```
// Funcion para guardar la tarea
function saveTask(e) {
    // Traemos los valores de titulo y descripcion por id
    // y lo guardamos en variables
    let title = document.getElementById('title').value;
    let description = document.getElementById('description').value;

    // Creamos una constante donde se almacenara como objeto
    const task = {
        title,
        description
    };

    // Si no hay tareas la guardamos
    if (localStorage.getItem('tasks') === null) {
        let tasks = [];
        tasks.push(task);
        localStorage.setItem('tasks', JSON.stringify(tasks));
    } else {
        // Sino, si ya hay tareas
        let tasks = JSON.parse(localStorage.getItem('tasks'));
        tasks.push(task);
        localStorage.setItem('tasks', JSON.stringify(tasks));
    }

    // Para actualizar las tareas
    getTasks();
    // Para resetear el formulario
    document.getElementById('formTask').reset();
    //Para que no se recargue la pagina
    e.preventDefault();
}
```

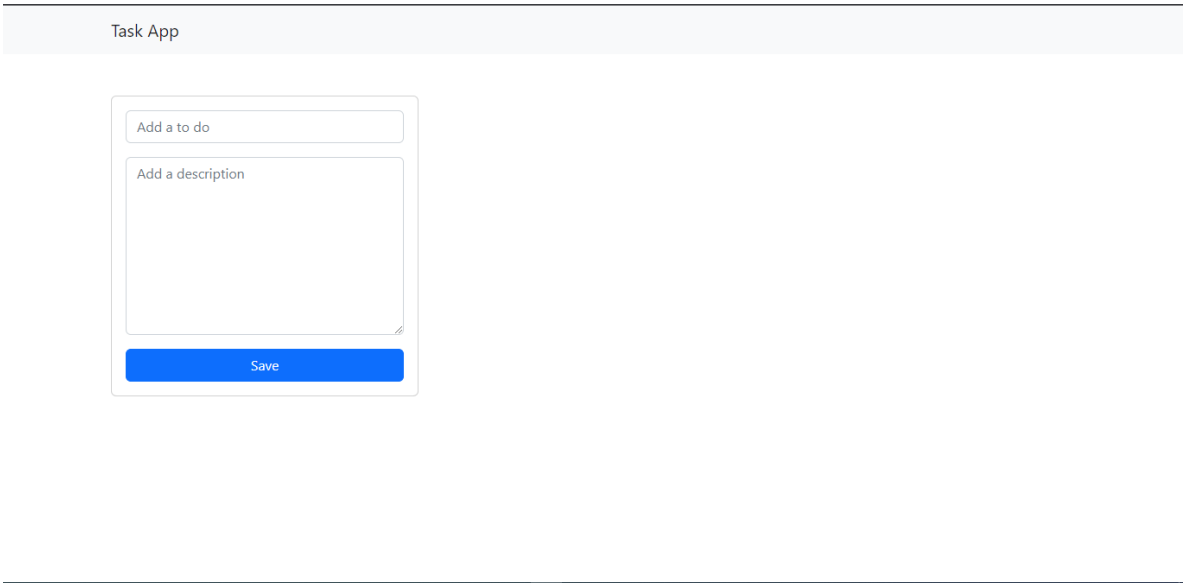
Ahora, creamos otra función llamada “getTasks”, este nos servirá para traernos toda la información que esté almacenada en el localStorage y por consiguiente mostrarla toda en la página, insertando html dentro del javascript.

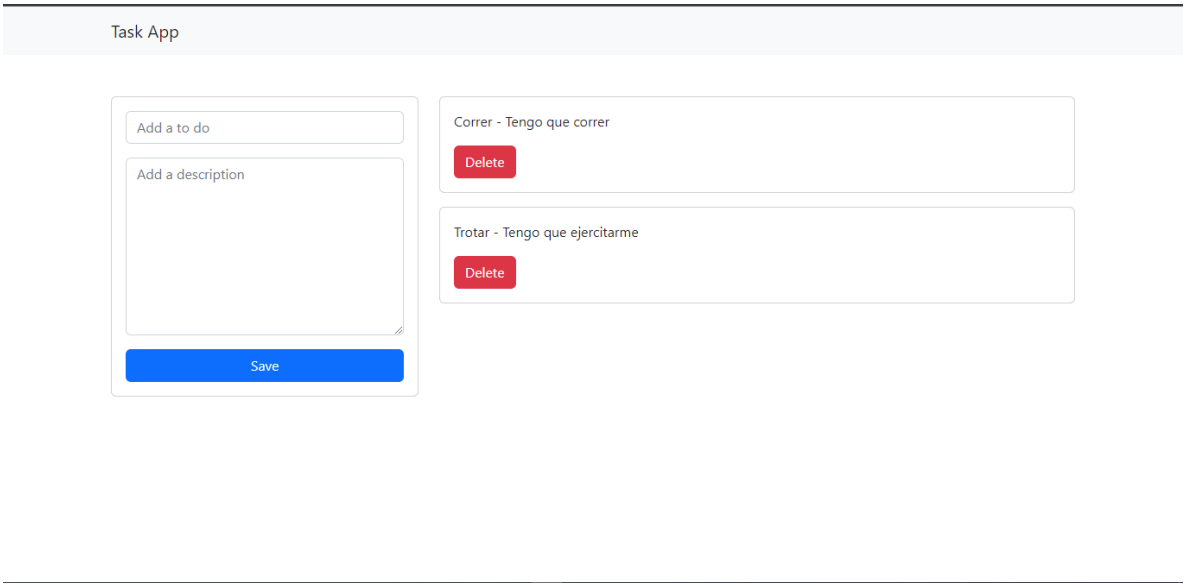
```
// Funcion para traer y mostrar las tareas
function getTasks() {
    /* Se definen dos variables, la primera para traer los
       datos almacenados en el localStorage y la segunda
       simplemente para traer el elemento tasks */
    let tasks = JSON.parse(localStorage.getItem('tasks'));
    let tasksView = document.getElementById('tasks');

    // Limpiamos el div con clase tasks
    tasksView.innerHTML = '';

    /* El for nos sirve para recorrer el localStorage
       para poder traernos toda la información */
    for (let i = 0; i < tasks.length; i++) {
        let title = tasks[i].title;
        let description = tasks[i].description;

        // Insertamos html desde js
        tasksView.innerHTML +=
        `<div class="card mb-3">
            <div class="card-body">
                <p>${title} - ${description}</p>
                <a class="btn btn-danger" onclick="deleteTask('${title}')">
                    Delete
                </a>
            </div>
        </div>`;
    }

}
```

Por último, creamos la función para eliminar datos, primero nos traemos todos los datos almacenados en el localStorage, recorremos el arreglo y por último, evaluamos si los datos coinciden, si todo esto se cumple, se debe eliminar el registro satisfactoriamente; luego volvemos a enviar todo al localStorage para que se pueda actualizar la vista. Y llamamos afuera del todo la función getTasks para poder que al iniciar se muestren los datos ya ingresados.

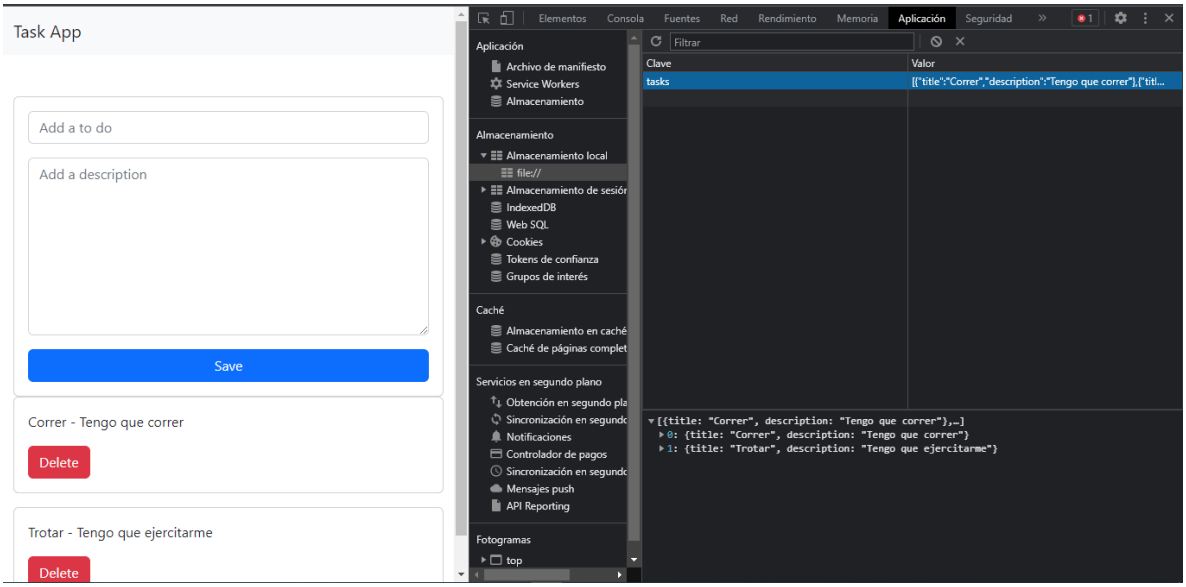
```
68  // Funcion para eliminar una tarea
69  function deleteTask(title) {
70      // Traemos toda la información almacenada
71      let tasks = JSON.parse(localStorage.getItem('tasks'));
72
73      // Recorremos el array
74      for (let i = 0; i < tasks.length; i++) {
75          /* Evaluamos si el titulo que esta guardado
76              es el mismo que se esta mostrando */
77          if (tasks[i].title == title) {
78              // Lo eliminamos
79              tasks.splice(i, 1);
80          }
81
82      }
83
84      // Enviamos la informacion pero con el registro ya borrado
85      localStorage.setItem('tasks',JSON.stringify(tasks));
86      // Llamamos la funcion para mostrar las tareas
87      getTasks();
88  }
89
90  // Corremos desde un principio la funcion para que muestre los datos
91  getTasks();
```

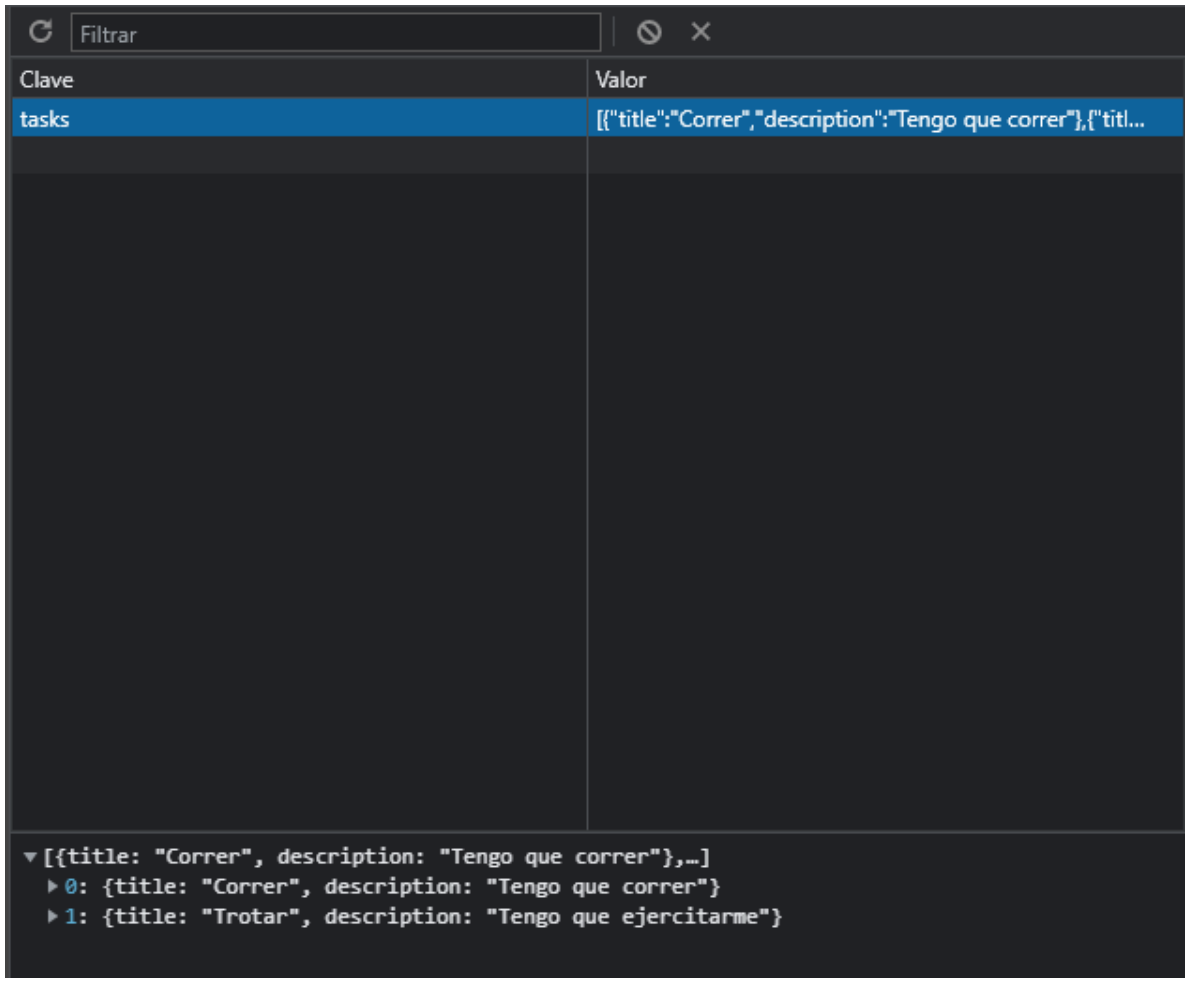
Hemos terminado el desarrollo de este aplicativo, y se debería ver así:





Aquí podemos ver el localStorage, donde están almacenados dos registros.




Filtrar
Clave
Valor
tasks
[{"title":"Correr","description":"Tengo que correr"},{"titl...
[{title: "Correr", description: "Tengo que correr"},…]
0: {title: "Correr", description: "Tengo que correr"}
1: {title: "Trotar", description: "Tengo que ejercitarme"}