

# MANUAL

Lista de tareas

Jhonatan Alexander Sanchez Cubillos

MOONLIGHT.DEV C.I.C

# INTRODUCCION

## CREACION DE UNA LISTA DE TAREAS

En el siguiente manual se mostrará los pasos de creación de una lista de tareas dinámica utilizando Css y JS para darle formato y dinamismo

Incluye

1. Lista de tareas (to-do-list)
2. Campo de Texto
3. Botón que permita agregar y eliminar una lista de tareas

Herramientas

1. Un archivo HTML llamado to-do-list (Nombre Opcional)
2. Un archivo Css
3. Un archivo Js (JavaScript)

Las listas de tareas estarán dentro de otra lista de tareas, esto para un mejor diseño y comprensión.

## TO-DO LIST

Una **to-do list** (o "lista de tareas" en español) es una herramienta de organización que consiste en una lista de actividades, pendientes u objetivos que una persona necesita completar. Puede ser escrita en papel o en una aplicación digital.

**Características de una to-do list:**

1. **Simple:** Enumera tareas claras y específicas.
2. **Priorizable:** Puedes ordenar las tareas por importancia o urgencia.
3. **Flexible:** Se puede modificar, tachar o agregar nuevas tareas.
4. **Motivadora:** Ayuda a visualizar el progreso al marcar lo completado.

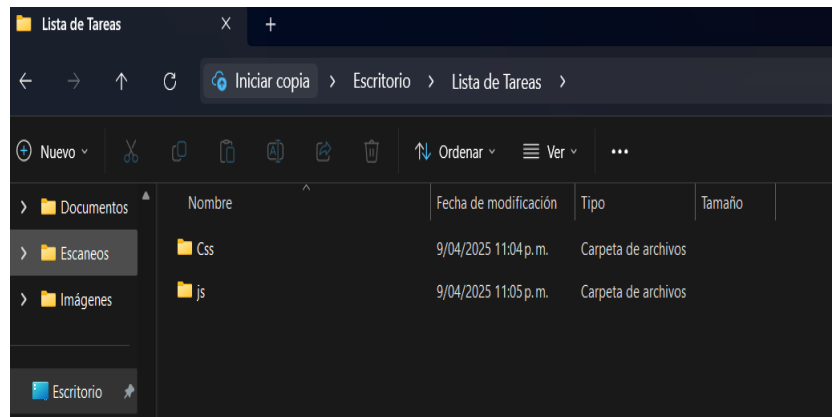
## Paso a paso

A la hora de crear una pagina web se empieza creando todo el estructurado en un archivo HTML, seguidamente se le agregan los diseños en el archivo Css para mejorar la visualización y finalmente se le agrega animaciones en JavaScript para la interacción en la página.

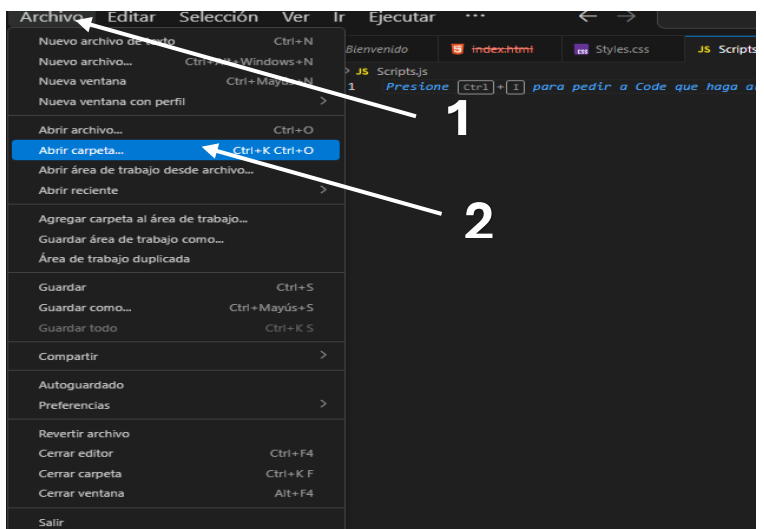
### Paso #1

#### Configuración de Visual Studio Code

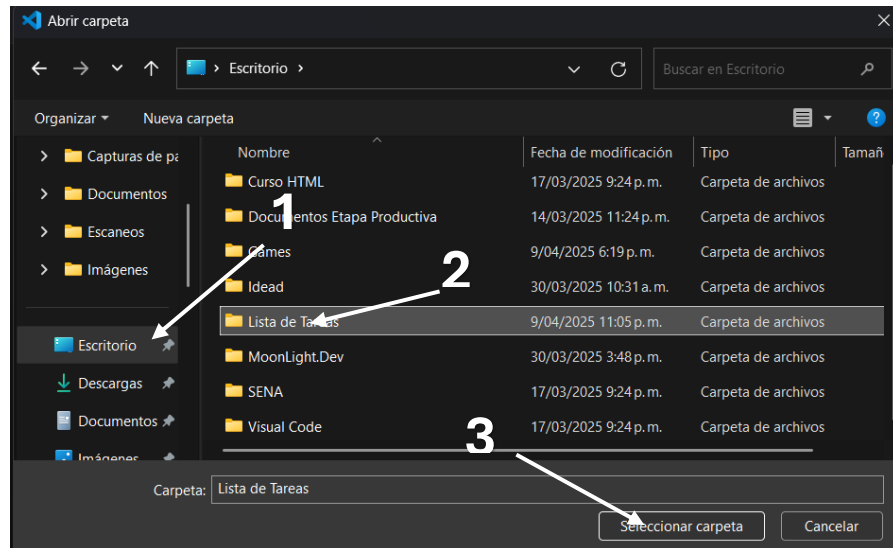
1. Para empezar, abrimos el explorador de archivos y creamos una carpeta en el escritorio llamada Lista de Tareas, seguido, en la carpeta creamos otras dos carpetas llamadas Css para el diseño y js para la interacción.



2. Abrimos Visual Code y dirigimos el cursor al apartado de archivo, buscamos el apartado de abrir carpeta y lo seleccionamos.

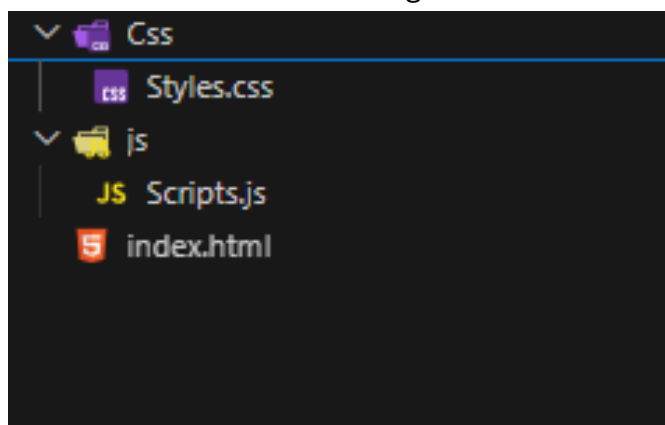


- 
3. Nos dirigimos al apartado de escritorio y seleccionamos la carpeta llamada Lista de Tareas, una vez hecho esto le presionamos Seleccionar carpeta para abrirla en el editor *puedes acceder presionando las teclas `ctrl + k + o`, o desde el escritorio arrastrando la carpeta a visual Code.*

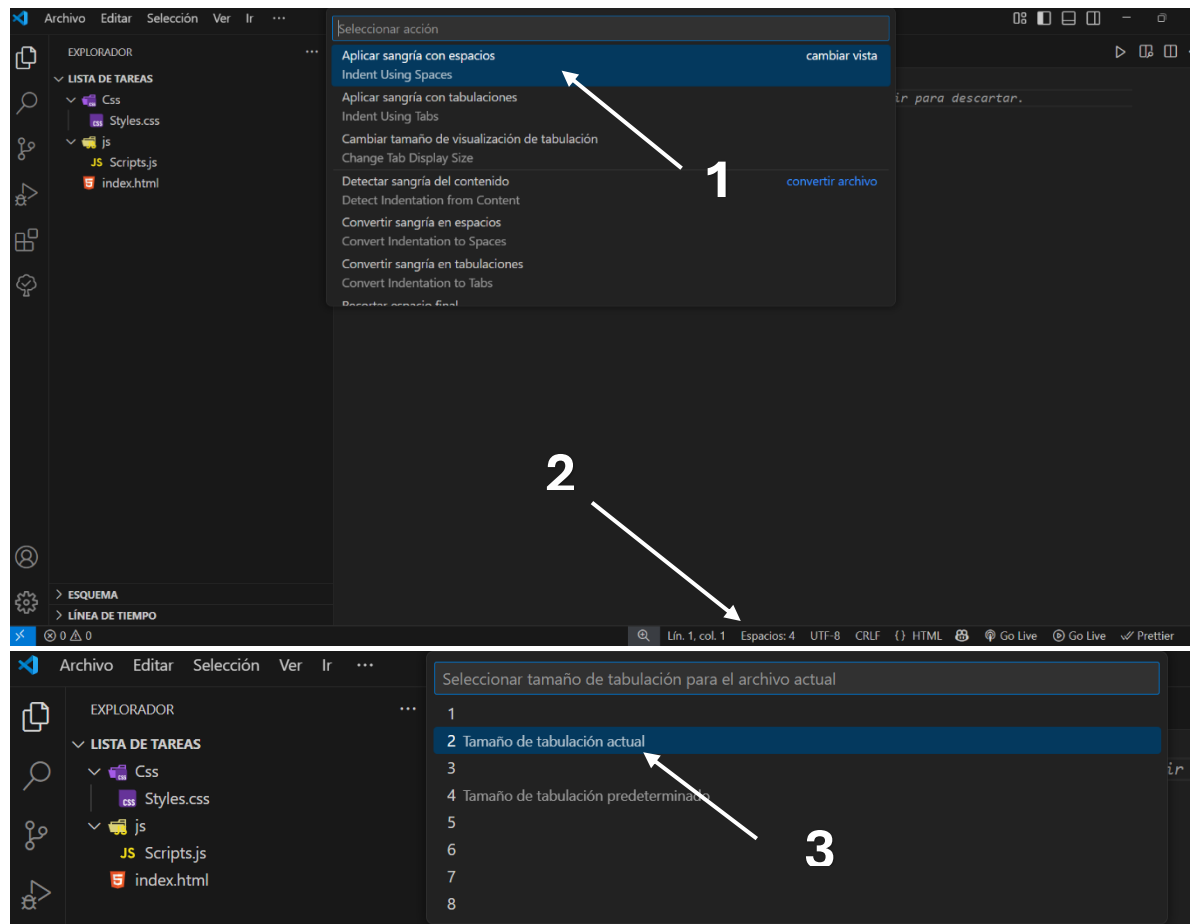


- 
4. Desde visual Code creamos un archivo HTML en la carpeta principal y otros en las carpetas Css y Js:
- El archivo HTML se nombrará **index.html** y estará en la carpeta principal
  - En la carpeta Css se creará un archivo llamado Styles.css
  - En la carpeta Js se creará un archivo llamado Scripts.js

La estructura se vería de la siguiente manera:



5. Para que los archivos se vean más estructurados nos dirigimos al apartado de **Espacios** en visual Code, seguido le presionamos **Aplicar sangría con espacios** y finalmente lo colocamos en **2**:



## Paso #2

### HTML

#### Maquetado del archivo HTML

1. Abrimos el archivo HTML y creamos lo básico de la página web, puede ser manualmente o con teclas de comando como **! + TAB** o **html + : + 5 + TAB** esto creara la estructura básica de HTML sin perder mucho tiempo, luego, cambiamos el nombre en la etiqueta **<title>** nombrándolo **To-do List**.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Lista de tareas</title>
</head>
<body>

</body>
</html>
```



---

#### Cuerpo de la página (Body)

1. Creamos un título con la etiqueta **<h1>** y colocamos dentro de la etiqueta **Lista de Tareas**.

```
<h1>Lista de tareas</h1>
```

---

2. Seguidamente creamos un cuadro de texto con la etiqueta **<input>** de tipo texto con el atributo **(type="text")** y creamos una variable con el atributo **(id="nuevaTarea")** nombrándolo "nuevaTarea", para la interacción con JavaScript *Recuerda que para que la variable sea funcional se debe colocar en mayúscula la primera letra de cada palabra* y finalmente agregamos un texto dentro del cuadro nombrándolo "Ingrese nueva tarea" con el atributo **(placeholder="Ingrese una nueva tarea")**.

```
<input type="text" id="nuevaTarea" placeholder="Ingrese una nueva tarea">
```

---

3. Agregamos un botón con la etiqueta **<button>** y le agregamos un evento con (**onclick=""**) el cual, al hacer click en el botón ocurra el evento determinado
- 

4. Dentro de las comillas agregamos una función llamada **agregarTarea** y seguido le colocamos un paréntesis **()**, dentro de la etiqueta de apertura y de cierre colocamos **Agregar tarea**.

```
<button onclick="agregarTarea()">Agregar Tarea</button>
```

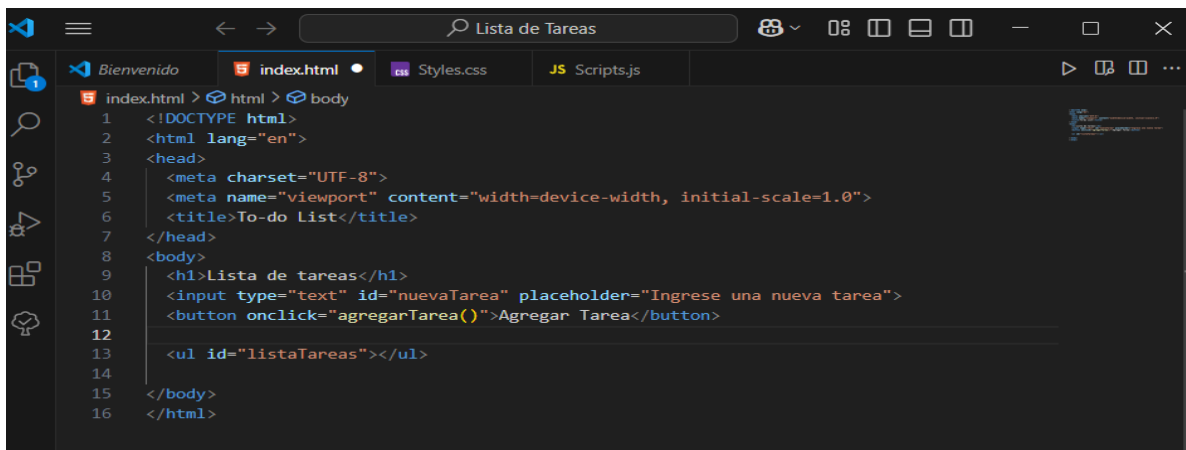
---

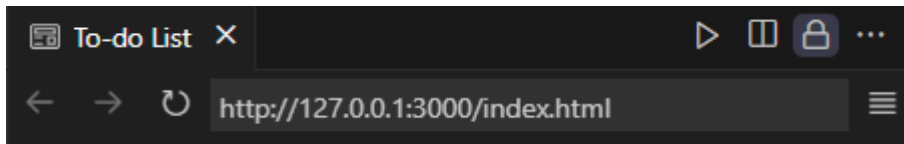
5. Agregamos una lista no ordenada con la etiqueta **<ul>** y le colocamos un **id** nombrándolo **listaTareas**

```
<ul id="listaTareas"></ul>
```

---

Y listo, terminamos el maquetado en HTML, agregamos el título, el cuadro de texto para escribir una tarea, y el botón para agregarla,





# Lista de tareas

Agregar Tarea

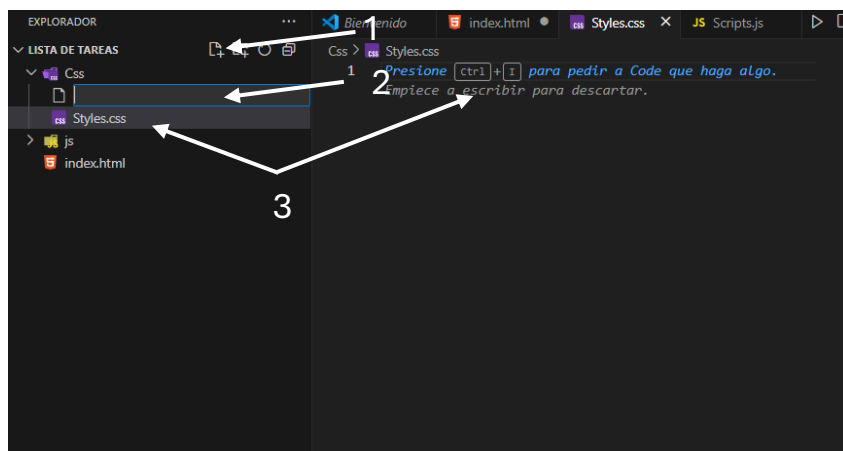
Al presionar el botón agregar tarea, no ocurrirá nada, esto debido a que no se ha creado un código en JavaScript para darle las funciones y la interacción.

## Paso #3

## Css

### Maquetado del archivo Css

1. Abrimos la carpeta Css y presionamos el botón + ubicado en la parte de arriba de los archivos, esto abrirá una ventana y ahí escribiremos **Styles.css**, esto agregará un archivo de tipo Css.



2. Nos dirigimos al archivo HTML creado y en el head debajo de la etiqueta <title> colocamos la etiqueta <link rel="stylesheet" href=""> y dentro del href



colocamos **./Css/Styles.css**. esto con el objetivo de conectar el archivo Css con el HTML

```
<link rel="stylesheet" href="./Css/Styles.css">
```

---

3. Vamos a darle formato al cuerpo del HTML **body**, y para eso escribimos en el editor **body** y seguido los corchetes **{}** para que los atributos se centren en este, el body contendrá lo siguiente:

- a. font-family: Arial, sans serif; ---  
*fuente del texto*
- b. text-align: center; --- *alineación  
del texto dentro del cuerpo*
- c. margin: 20px --- *margen del  
cuerpo*

```
body {  
    font-family: Arial, sans-serif;  
    text-align: center;  
    margin: 20px;  
}
```

- 
4. para darle mejor imagen a la lista colocamos **li** que es un elemento del **ul** o lista no ordenada, colocamos los corchetes para lo siguiente
- a. margin: 10px 0;

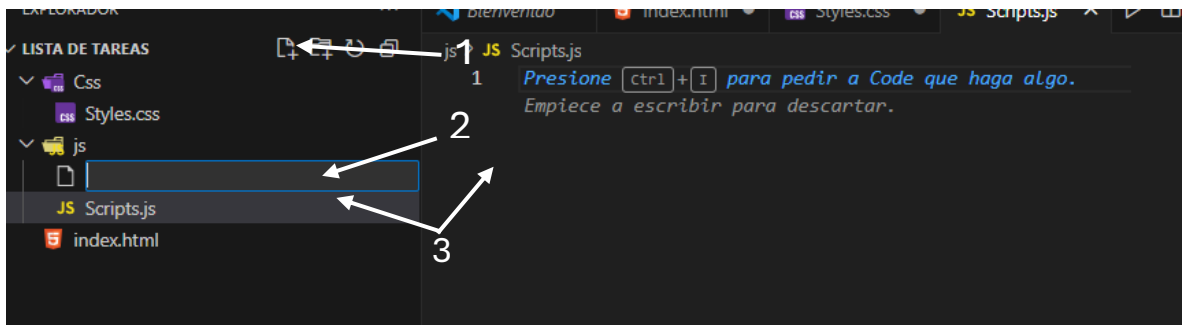
```
li {  
    margin: 10px 0  
}
```

---

## Paso #4

## JavaScript

Una vez tenemos el HTML y el Css completo procederemos a crear el código de Js así que crearemos un archivo en la carpeta Js creada con anterioridad y nombraremos el archivo **Scripts.js**



1. creamos una función con **function** y le damos la variable **agregarTarea** con los paréntesis al lado, colocamos un corchete de apertura

```
function agregarTarea() {
```

---

2. vamos a darle la interacción al input, y para eso creamos un **let** seguido colocamos una nueva variable llamada **nuevaTareaTexto** decimos que sea igual = a **document.getElementById** y colocamos la variable que se encuentra en el input el cual es **nuevaTarea**. agregamos **value** para guardar el valor del elemento del input y no el input como tal

```
let nuevaTareaTexto = document.getElementById("nuevaTarea").value; I
```

---

3. para que el código funcione tenemos que evaluar si el input está vacío o no y para eso agregamos un **if** el cual dirá que si el elemento **nuevaTareaTexto** esta estrictamente vacío **=== ""** entonces saldrá un cartel de alerta que diga que ingreses un texto en el campo.

```
if (nuevaTareaTexto === "") {
```

---

4. para que la alerta salga agregamos al código el elemento **alert** y dentro de dos paréntesis colocamos lo que queremos que salga, en este caso sería **“Por favor, Ingrese una tarea”** y colocamos **return** para que continúe con la ejecución del código.

```
alert("Por favor, Ingrese una tarea");  
return;
```

---

5. ahora para crear elementos en la lista usaremos un método que se llama **createElement**, entonces creamos un **let** con la variable **nuevaTarea** y diremos que sea igual = a **document.createElement** y colocaremos que el tipo de elemento que queremos crear es **li**

```
let nuevaTarea = document.createElement("li");
```

---

6. ahora le asignaremos el valor a el elemento en la lista, para ello llamamos a la variable **nuevaTarea** y le asignamos el valor de texto con. **textContent** y ahora lo asignamos con = para que sea igual a **nuevaTareaTexto** que es la variable que habíamos creado anteriormente, ahora, le agregamos un espacio entre comillas para agregar la función del botón eliminar tarea

```
nuevaTarea.textContent = nuevaTareaTexto + " ";
```

---

7. ahora crearemos una función para eliminar tareas y para esto creamos un **let** con el texto **botonEliminar** y diremos que sea igual = a **document.createElement** y colocaremos entre paréntesis y comillas **button**. Ahora llamamos a **botonEliminar** colocamos un **textContent** y lo llamaremos **Eliminar**

```
let botonEliminar = document.createElement("button");  
botonEliminar.textContent = "Eliminar";
```

---

8. Ahora lo que sigue es un poco más complicado y es agregar la funcionalidad que permita que al ejecutar la función **agregarTarea** se ejecute una subfunción que es la de **eliminarTarea** para esto aremos lo siguiente
- llamamos la función **eliminarTarea** le agregamos el evento. **onclick** y diremos que es igual a **function() {nuevaTarea.remove()}** el **remove** se usa para eliminar la tarea que se agregó con anterioridad

```
botonEliminar.textContent = "Eliminar";  
botonEliminar.onclick = function() { nuevaTarea.remove();}
```

---

9. seguido agregaremos el botón de eliminar tarea y esto se logra colocando el siguiente código

le ponemos a **nuevaTarea** un **appendChild** para agregar el elemento hijo, seguido entre paréntesis llamamos el elemento **botonEliminar**

```
nuevaTarea.appendChild(botonEliminar);
```

---

10. esto es un poco largo lo sé, pero ya estamos finalizando el código de JavaScript, ahora agregaremos los elementos agregados a la lista de tareas, esto se logra usando el atributo **document.getElementById** y entre paréntesis llamamos la lista, anteriormente la llamamos **listaTareas** entonces la llamaremos de esa forma, una vez hecho esto, agregamos un punto y llamamos el elemento hijo con **appendChild(nuevaTarea)**

```
document.getElementById("listaTareas").appendChild(nuevaTarea);|
```

---

11. Por último, agregaremos una función que permita reiniciar el input para así ocuparlo de nuevo con otra tarea, usamos **document.getElementById** para traer el elemento que se encuentra en el input, en este caso, sería (**"nuevaTarea"**) y colocamos que el valor **.value** sea igual a cero = ""

```
document.getElementById("nuevaTarea").value = "";
```

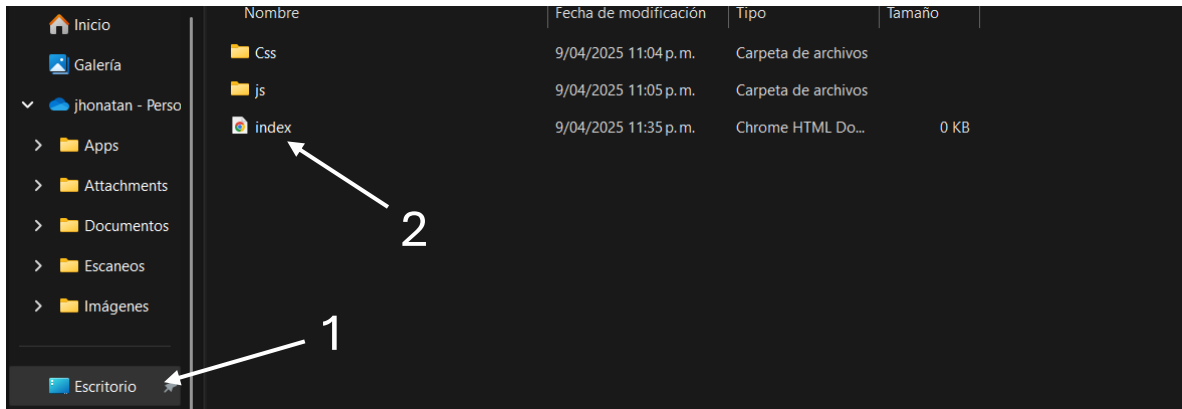
---

12. Listo ya tenemos terminado el maquetado de JavaScript ahora solo resta probar si funciona

## Paso #4

## Pruebas

Ya tenemos todo listo para la probar el código si funciona o no, nos dirigimos a la carpeta donde tenemos guardado el proyecto y le damos doble click al archivo HTML.



Y así quedaría

---

## Lista de tareas

prueba 1

Gracias por participar

Jhonatan Sanchez – Lista de Tareas