

Actividad: Paso a Paso Javascript Frontend

Objetivo

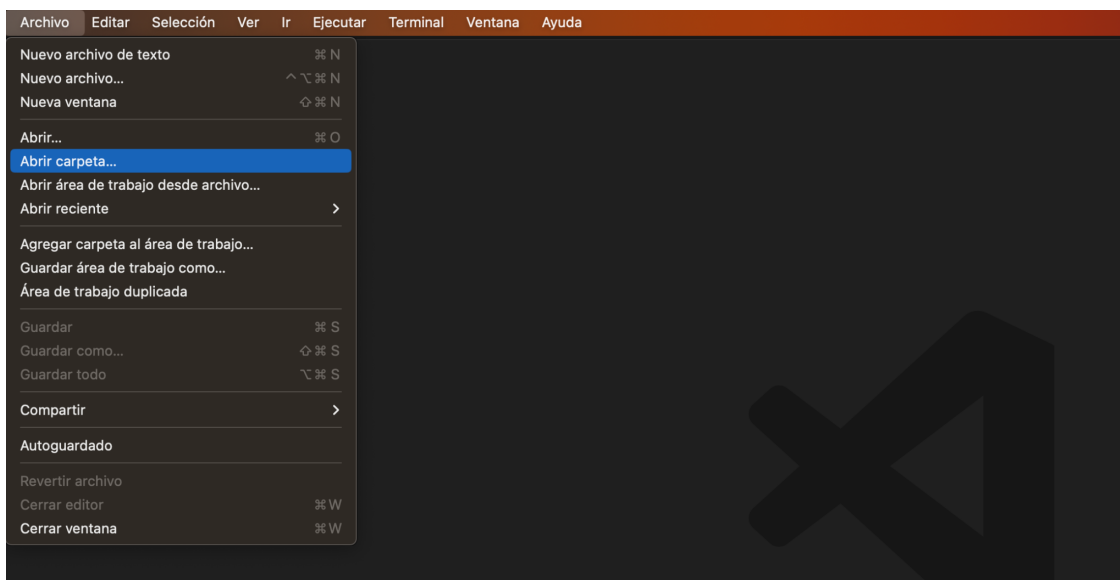
Aplicar los conocimientos adquiridos sobre programación y desarrollo frontend para crear una página simple utilizando HTML, CSS y javascript.

Consigna

Vamos a trabajar juntos en la creación de una aplicación sencilla de lista de tareas utilizando HTML, CSS y JavaScript. Nuestra aplicación permitirá a los usuarios agregar nuevas tareas, marcarlas como completadas y eliminarlas de la lista.

Desarrollo Paso a Paso

1. Iniciamos la aplicación VSCode.
2. Ir al menú Archivo -> Abrir Carpeta



3. Del paso anterior se abre una ventana de búsqueda de carpetas, aquí buscamos la carpeta Documentos, seleccionamos y luego elegimos la opción "Nueva Carpeta", indicando el nombre de la carpeta **actividad_lista_tareas**
4. Selecciona la carpeta **actividad_lista_tareas** para que se abra en VSCode.

5. Dentro de esa carpeta, creamos dos archivos: index.html, styles.css y script.js.

Agregamos Contenido HTML

6. Abrimos el archivo index.html.
7. Añadimos la estructura básica de HTML, incluyendo el <!DOCTYPE>, <html>, <head> y <body>:

```
<!DOCTYPE html>
<html>
<head>
  <title>Lista de Tareas</title>
  <link rel="stylesheet" type="text/css" href="estilos.css">
</head>
<body>
  <h1>Lista de Tareas</h1>
  <input type="text" id="nuevaTarea" placeholder="Ingrese
una nueva tarea">
  <button onclick="agregarTarea()">Agregar</button>
  <ul id="listaTareas"></ul>
  <script src="script.js"></script>
</body>
</html>
```

Estilizados con CSS

8. Abrimos styles.css.
9. Añadimos estilos básicos para las etiquetas que hemos usado.

```
body {
  font-family: Arial, sans-serif;
  margin: 0;
  padding: 0;
  background-color: #f8f8f8;
}

h1 {
```

```
    text-align: center;
    margin-top: 20px;
}

input[type="text"] {
    padding: 8px;
    width: 60%;
    border: 1px solid #ccc;
    border-radius: 5px;
    margin-right: 10px;
}

button {
    padding: 8px 15px;
    background-color: #007bff;
    border: none;
    border-radius: 5px;
    color: white;
    cursor: pointer;
}

ul {
    list-style-type: none;
    padding: 0;
}

li {
    display: flex;
    align-items: center;
    margin: 5px 0;
    padding: 8px;
    background-color: white;
    border: 1px solid #ccc;
    border-radius: 5px;
}

.completed {
    text-decoration: line-through;
    color: #888;
}
```

```
button.delete {
  margin-left: auto;
  background-color: #dc3545;
}
```

Agregamos funcionalidad con Javascript

10. Abrimos styles.css.
11. Copiamos el siguiente código para agregar funcionalidad

// Función para agregar una nueva tarea

```
function agregarTarea() {
  const nuevaTareaInput = document.getElementById("nuevaTarea");
  const listaTareas = document.getElementById("listaTareas");
  const nuevaTareaTexto = nuevaTareaInput.value.trim();

  if (nuevaTareaTexto !== "") {
    const nuevaTarea = document.createElement("li");
    nuevaTarea.textContent = nuevaTareaTexto;

    const botonEliminar = document.createElement("button");
    botonEliminar.textContent = "Eliminar";
    botonEliminar.className = "delete";
    botonEliminar.onclick = function() {
      listaTareas.removeChild(nuevaTarea);
    };

    nuevaTarea.appendChild(botonEliminar);
    listaTareas.appendChild(nuevaTarea);
    nuevaTareaInput.value = "";
  }
}
```

// Función para marcar una tarea como completada

```
function marcarCompletada(tarea) {
  tarea.classList.toggle("completed");
}
```

```
// Agregar evento de clic a las tareas para marcarlas como
completadas
document.getElementById("listaTareas").addEventListener("click",
function(event) {
    if (event.target.tagName === "LI") {
        marcarCompletada(event.target);
    }
});

// Agregar evento de clic al botón "Agregar"
document.getElementById("agregar").addEventListener("click",
agregarTarea);
```

Probamos la Página

Vista Previa:

12. Abrimos el archivo "index.html" en un navegador para ver cómo se ve nuestra página o utilizamos la vista previa de VSCode.

Desafío: Filtrar Tareas Completadas y Pendientes (script.js)

13. Agrega botones "Mostrar Completadas" y "Mostrar Pendientes".
14. Implementa funciones que muestren u oculten tareas según su estado.
15. Agrega eventos de clic a los botones para que activen las funciones correspondientes.

Solución desafío:

```
// Función para mostrar tareas completadas
function mostrarCompletadas() {
    const tareas = document.querySelectorAll("li");
    tareas.forEach(tarea => {
        if (tarea.classList.contains("completed")) {
            tarea.style.display = "flex";
        } else {
            tarea.style.display = "none";
        }
    });
}

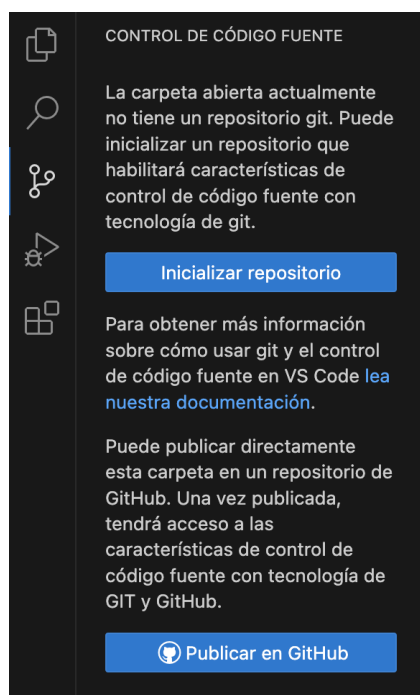
// Función para mostrar tareas pendientes
function mostrarPendientes() {
    const tareas = document.querySelectorAll("li");
    tareas.forEach(tarea => {
        if (!tarea.classList.contains("completed")) {
            tarea.style.display = "flex";
        } else {
            tarea.style.display = "none";
        }
    });
}

// Agregar evento de clic al botón "Mostrar Completadas"
document.getElementById("mostrarCompletadas").addEventListener("click", mostrarCompletadas);

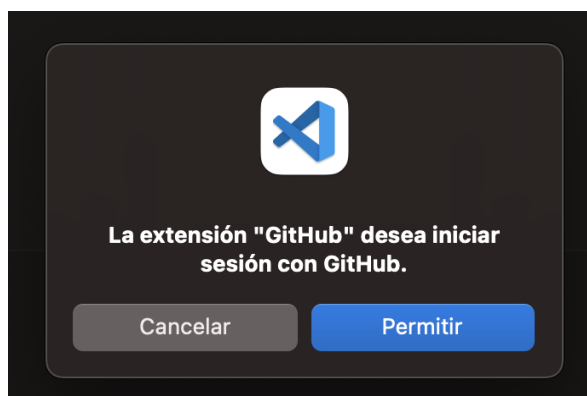
// Agregar evento de clic al botón "Mostrar Pendientes"
document.getElementById("mostrarPendientes").addEventListener("click", mostrarPendientes);
```

Entrega del Trabajo

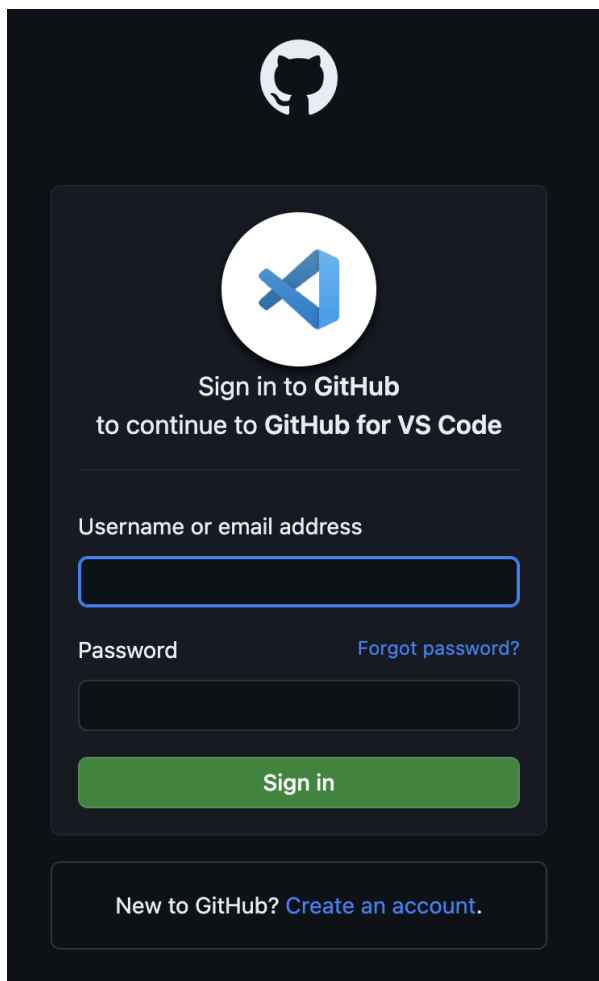
16. Ahora volvemos a VSCode y vamos a llevar a GitHub el código fuente generado usando la herramienta de git de VSCode. Hacemos click en la opción "Publicar en GitHub":



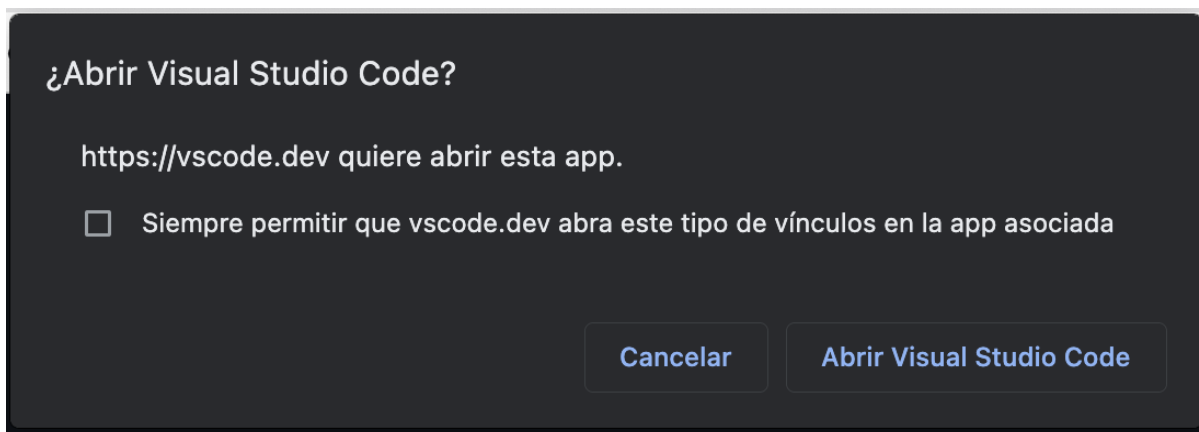
17. VSCode muestra este mensaje para que le daré permisos a iniciar sesión en GitHub a través de un navegador:



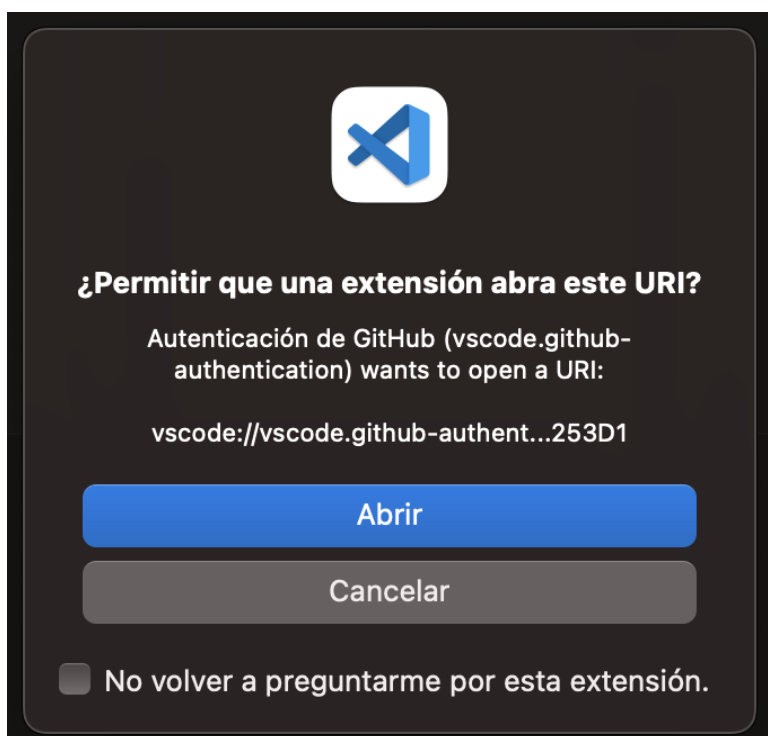
18. VSCode nos lleva al navegador web que tengamos por defecto y nos pide que ingresemos los datos de nuestra cuenta GitHub (el que no tenga cuenta se crea una nueva con su correo electrónico):



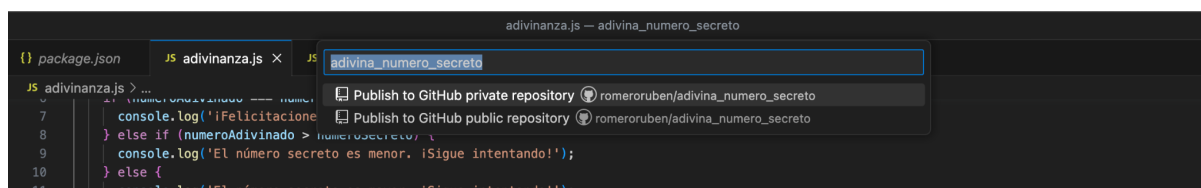
19. El navegador solicita que permitamos abrir VSCode, seleccionamos la opción “Abrir Visual Studio Code”



20. VSCode nos pide permisos para abrir una url, hacemos click en la opción “Abrir”:

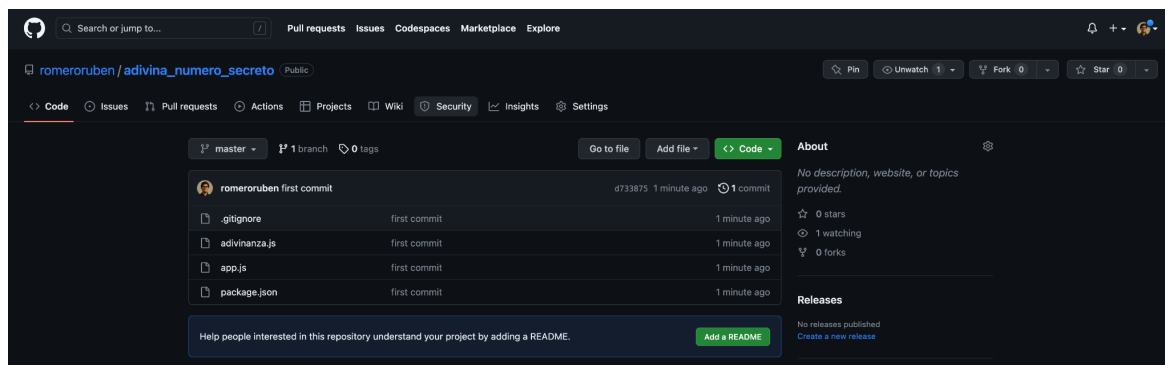


21. Ahora debemos seleccionar que tipo de repositorio queremos crear, seleccionar “Publish to GitHub **public** repository”, esto significa que vamos a publicar nuestro código en forma pública.



22. A continuación seleccionamos los archivos y carpetas que queremos “versionar” y llevar a GitHub.
23. Esperamos unos minutos hasta que en VSCode podemos ver la siguiente notificación (abajo a la derecha), hacemos click en “Abrir en GitHub”.

24. Se abre un navegador con el repositorio de GitHub disponible para que cualquiera lo pueda acceder:



25. **IMPORTANTE!** Para dar por finalizada la actividad compartir la url de GitHub en la actividad del aula virtual en <https://uve.frc.utn.edu.ar/>.