



Trabajo Practico #1

Vue.js

El presente documento tiene como finalidad servir como guía para la confección del primer Trabajo Practico. El trabajo practico se centra en la utilización de Vue.js usando un archivo de texto html, y declarado usando el CDN.

El desarrollo, conclusión y entrega del mismo sera en un repositorio git creado por el alumno en el punto del practico donde se indique.

En cada punto de esta guía se indicara si corresponde realizar un commit y push al repositorio git con el comentario apropiado que refleje el motivo de los cambios.

Requerimientos

Para la realización del practico se necesitan:

- A. Una cuenta Git en Gitlab o Github.
- B. Tener instalado el cliente Git en nuestro sistema operativo.
- C. Un editor de texto.

Instrucciones

- Completa todos los ejercicios dentro de este mismo archivo HTML.
- Utiliza Vue.js a través del CDN.
- No necesitas instalar nada localmente, solo abre el archivo en un navegador.
- Asegúrate de que todo el código Vue.js esté dentro de las etiquetas `<script>` en la sección `<body>` de tu HTML.

Paso #1 - Creación repositorio

Crear un repositorio vacío en una de las plataformas propuestas.

Clonar el repositorio localmente y crear dentro una carpeta con el nombre "practico01".

Hacer un commit y un push de los cambios. Como mensaje del commit utilizar la descripción del paso actual, en este caso "Creación repositorio". Esto aplica, con el comentario adecuado, a todos los commits que se hagan en el resto del practico.

Paso #2 - Información de Perfil Simple (Interpolación)

Crea una aplicación Vue que muestre la información de un usuario.

Requisitos:

Declara las siguientes variables en el data de tu instancia Vue:

```
nombre: "Ana García"
edad: 30
ciudad: "Tupungato"
```

Muestra estas variables en el HTML utilizando interpolación (`{{ }}`).

Añade un párrafo que diga "Hola, mi nombre es [nombre], tengo [edad] años y vivo en [ciudad]."

Resultado Esperado:

```
Nombre: Ana García
Edad: 30
Ciudad: Madrid
Hola, mi nombre es Ana García, tengo 30 años y vivo en Tupungato.
```

Se debe realizar un commit y un push de los cambios al repositorio.

Paso #3 - Contador Interactivo (Interpolación y Eventos)

Crea un contador que pueda incrementarse y decrementarse.

Requisitos:

Declara una variable contador inicializada en 0 en el data.

Muestra el valor actual de contador en el HTML.

Crea un botón "Incrementar" que al hacer click aumente el contador en 1.

Crea un botón "Decrementar" que al hacer click disminuya el contador en 1.

Resultado Esperado:

Un texto que diga "Contador: 0" y dos botones. Al hacer click, el número debe actualizarse.

Hacer un commit y un push de los cambios realizados.

Paso #4 - Lista de Tareas con Conteo y Estado (Propiedades Computadas, Interpolación, Eventos)

Crea una lista de tareas simple donde puedas añadir tareas y ver cuántas tareas tienes y cuántas están pendientes.

Requisitos:

- Declara una variable nuevaTarea vacía ('') en el data.
- Declara un array tareas en el data. Cada tarea debe ser un objeto con id, texto y completada (booleano). Ejemplo: `[{ id: 1, texto: 'Aprender Vue', completada: false }]`.
- Muestra la lista de tareas en el HTML. Para cada tarea, muestra su texto y un checkbox que refleje su estado completada.
- Crea un input de texto y un botón "Añadir Tarea". Al hacer click, la tarea de nuevaTarea debe añadirse al array tareas con un ID único y completada: false. Luego, limpia el input nuevaTarea.
- Utiliza una **propiedad computada** llamada tareasPendientes que devuelva el número de tareas donde completada es false.
- Utiliza una **propiedad computada** llamada totalTareas que devuelva el número total de tareas.
- Muestra en el HTML el *"Total de Tareas:"* y *"Tareas Pendientes:"*.
- Al marcar o desmarcar el checkbox de una tarea, el estado completada de esa tarea en el array tareas debe actualizarse.

Resultado Esperado:

Un input y un botón para añadir tareas. Una lista de tareas (con su texto y checkbox). Un contador de "Total de Tareas" y "Tareas Pendientes" que se actualizan automáticamente.

Hacer un commit y un push de los cambios realizados.