Desarrollo de un Tablero Kanban con Vue 3

Examen Final – Parte 1

Índice

Objetivo

Construir una aplicación Kanban en Vue 3 con Typescript, usando Composition API, para gestionar tareas, utilizando únicamente ref, reactive, computed, props, emit, localStorage.

El tablero debe estar estructurado en tres columnas visibles en pantalla que representen los diferentes estados de las tareas. Se muestra esquema de cómo sería la vista:

Tablero Kanban

[Formulario para agregar nueva tarea]

| PENDIENTE | EN PROGRESO | COMPLETADO |

| [ ] Tarea 1 | [ ] Tarea 3 | [x] Tarea 5 |

| [ ] Tarea 2 | [ ] Tarea 4 | [x] Tarea 6 |

Las tareas almacenadas en localStorage deben cargarse automáticamente al iniciar la aplicación. (En código ya hay datos precargados en KanbanBoard.vue)

Cada columna debe contener una lista de tarjetas (tareas).

Las tareas deben poder moverse entre las columnas usando botones que tiene la tarjeta (tarea).

Se debe mostrar un formulario para agregar nuevas tareas.

Paso previo

1. Descarga el fichero zip que contiene la estructura del proyecto para hacer el examen.
    - El proyecto usa: Vue, Typescript, ESLint, Bootstrap
    - Estructura del proyecto (los componentes están vacíos, usa Composition API cuando las escribas)
    - Librerías bootstrap dentro del carpeta src/styles
2. Descomprime el fichero. Abre VSC, sitúate en la carpeta del proyecto y ejecuta en una Terminal → **npm install**
    - Si tienes problemas con la versión de Node ve al “Anexo: problemas con Node”
3. Levanta la aplicación con: **npm run dev**.
4. Continúa leyendo los Objetivos y Componentes y funcionalidad a desarrollar.

Elemento de ayuda:

Ya exite un localStore con datos precargados en KanbanBoard.vue.

Componentes y funcionalidad a desarrollar

1. src/components/TaskForm.vue

Formulario para agregar tareas

- Usa **ref** para manejar los valores de los campos del formulario.
- Emite (**emit**) un evento con los datos de la nueva tarea cuando se presiona el botón de agregar.
- Limpia los campos del formulario después de agregar una tarea.

El formulario debe mostrar:

- Campo de entrada para escribir el nombre de la tarea
- Campo de entrada para la descripción de la tarea
- Botón “Agregar” para agregar la nueva tarea al tablero.

El “nombre de la tarea” y “descripción” no puede estar vacíos. Advertir al usuario con ventana emergente.

A la tarea se le debe dar un ID único. Puedes generarlo usando Date.now().

Las nuevas tareas se agregarán al final de la columna Pendiente.

2. src/components/TaskItem.vue

Tarjeta de tarea individual

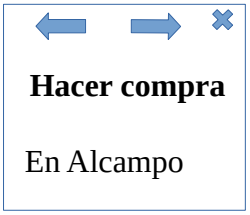
Los datos de la tarea son:

- un ID único
- nombre de la tarea
- descripción de la tarea
- el estado (Pendiente, En progreso y Completado)

La tarjeta:

- Recibe la información de una tarea mediante **props**.
- Permite cambiar su estado (Pendiente → En progreso → Completado).
- La tarea movida se pondrá al final de la columna destino.
- La tarea se puede eliminar.
- Emite (**emit**) eventos cuando la tarea cambia de estado o se elimina.
- Eliminar una tarea del estado y **localStorage.**

La tarjeta (tarea) debe mostrar:



- Texto con el título de la tarea
- Texto con la descripción de la tarea
- Botón flecha hacia izquierda y derecha para mover de estado la tarea.
  - Si una tarea está en **Pendiente**, el botón "←" debe estar deshabilitado.
  - Si una tarea está en **Completado**, el botón "→" debe estar deshabilitado.
- **Botón de eliminar tarea**

3. src/components/TaskList.vue

Lista de tareas por estado

- Cada lista de tareas se almacena por separado en **localStorage** y se cargan al iniciar la aplicación.
- Recibe mediante props la lista de tareas correspondiente al estado ("Pendiente", "En progreso" o "Completado").
- Usa **computed** para filtrar las tareas en base al estado recibido.
- Renderiza una lista de ***TaskItem.vue*** para mostrar cada tarea.
- **Escucha eventos de *TaskItem.vue* y emite eventos a *KanbanBoard.vue* cuando una tarea cambia de estado o se elimina.**

4. src/components/KanbanBoard.vue

Componente principal

Es el tablero donde se muestran el tablero Kanban con las tres columnas "Pendiente", "En progreso" y "Completado" donde las tareas se distribuyen en ellas según su estado.

Este componente:

- Almacena las tareas en un estado reactive.
- Carga las listas desde **localStorage** al iniciar la aplicación.
- Renderiza tres listas de tareas separadas según su estado utilizando el componente ***TaskList.vue.***
- Recibe eventos de ***TaskList.vue*** para actualizar el estado de las tareas en ***localStorage***.
- Contiene el componente ***TaskForm.vue*** para agregar nuevas tareas.
- Guarda en **localStorage** cualquier cambio en las listas de tareas.

Rúbrica

Se visualizan correctamente las tres columnas (Pendiente, En progreso, Completado).	1
Las tareas se almacenan y recuperan correctamente al recargar la página.	1
Las tareas pueden moverse entre columnas usando los botones.	1
TaskList.vue recibe correctamente las tareas según su estado. (props)	1
TaskItem.vue emite eventos correctamente a TaskList.vue y KanbanBoard.vue.	1
computed se usa para filtrar tareas por estado en TaskList.vue,	1
ref se usa correctamente en TaskForm.vue,	1
reactive se usa en KanbanBoard.vue para almacenar las tareas.	1
Se muestra una advertencia si se intenta agregar una tarea vacía.	0,5
Cada tarea tiene un ID generado con Date.now().	0,5
El código está organizado	0,5
Se incluyen comentarios explicativos en cada componente y función clave.	0,5
Total	10

Anexo: problemas con Node

El proyecto está hecho con la versión v18.20.4.

1. Verifica la versión que tienes. Desde terminal ejecuta: **node - - v**
2. Si es diferente comprueba qué versiones tienes instaladas. Desde terminal ejecuta: **nvm ls**
3. Si tienes instalada la v18.20.4. vamos a usarla. Desde terminal ejecuta: **nvm use 18.20.4**
4. Si no la tienes instalada vas a hacerlo. Desde terminal ejecuta: **nvm install v18.20.4**
5. Y ahora vamos a usarla. Desde terminal ejecuta: **nvm use 18.20.4**
6. Verifica que ya estas usando la versión v18.20.4. Desde terminal ejecuta: **node - - v**

Ahora ya puedes proseguir con la instalación del proyecto.