Desarrollo de un Calendario con Vue 3

Examen final - Parte 2

Índice

Objetivo

Desarrollar una aplicación en Vue 3 con TypeScript y Composition API,donde se gestione un calendario con días festivos y tareas.

Los datos se agregarán desde una sección de gestión accesible desde un navbar.

El estado del calendario se almacenará en Pinia y se persistirá en localStorage.

Se utilizará Vue Router para la navegación e i18n para las traducciones de los nombre de las secciones, no de los contenidos de la tarea que agregue el usuario.

Paso previo

1. Descarga el fichero zip que contiene la estructura del proyecto para hacer el examen.

   • El proyecto usa: Vue, Typescript, ESLint, Pinia e i18n. Se incluyen archivos JSON con datos precargados para días festivos y tareas.

   • Estructura del proyecto (los componentes están vacíos, usa Composition API cuando las escribas)

   • Librerías bootstrap dentro del carpeta src/styles (por si la quiere usar para dar estilo)

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. Despliega la aplicación con **npm run dev.**

4. Continúa leyendo los Objetivos y Componentes y funcionalidad a desarrollar.

Componentes y funcionalidad a desarrollar

1. src/views/CalendarView.vue

Vista principal del calendario

- Por defecto se muestra el mes actual (una semana por fila).
- Implementa un combobox para seleccionar el mes a mostrar.
- Completa la lógica para actualizar la rejilla de tarjetas según el mes seleccionado.
- Completa la función para cargar los datos iniciales desde el archivo JSON (fetch) y guárdalos en Pinia.
- Implementa la lógica para renderizar la rejilla de tarjetas usando CalendarDayComp.vue.
- Completa la función para guardar el mes seleccionado en Pinia y en localStorage.

2. src/components/CalendarDayComp.vue

Tarjeta de un día del calendario

- Representa un día en el calendario.
- Muestra en número del día, la festividad y la tarea.
- Recibe los datos mediante **props** y los muestra en la tarjeta

3. src/views/GestionView.vue

Vista de gestión de eventos

- Permite agregar, modificar y eliminar eventos (días festivos y tareas).
- Usa **Pinia** para gestionar el estado global.
- Contiene el **EventFormComp.vue** para manejar todos los eventos.
- Permite volver al calendario mediante **Vue Router**.

4. src/components/EventFormComp.vue

Formulario único para gestionar eventos

- Contiene un **select** para elegir entre añadir "Día festivo" o "Tarea"
- Permite ingresar un fecha y nombre
- Usa validaciones para evitar datos vacíos.
- Guarda los datos en **Pinia** y en **localStorage**.
- Emite eventos a **GestionView.vue** para actualizar la lista.

5. src/components/NavbarComp.vue

Barra de navegación

- Contiene enlaces a “`/`” (calendario) y “`/gestion`” (gestión de eventos).
- Incluye un selector de idioma con **i18n** para cambiar entre español e inglés.

6. src/composables/useCalendar.ts

Composable, Lógica del calendario

- Genera la estructura del calendario según el mes seleccionado.
- Filtra eventos por mes antes de retornarlos a **CalendarView.vue**.
- Obtiene el mes seleccionado desde **Pinia** .

7. Pinia Store (almacenamiento en Pinia)

- Gestiona el estado global de eventos y mes seleccionado.
- Usa `subscribe` para sincronizar datos con **localStorage**.

Rúbrica

Implementación de la vista principal (CalendarView.vue): El combobox para seleccionar el mes	0,5
Implementación de la vista principal (CalendarView.vue): la rejilla de tarjetas se actualiza según el mes seleccionado.	0,5
Implementación de la vista principal (CalendarView.vue): Los datos se cargan desde JSON y se guardan en Pinia.	0,5
Renderización de las tarjetas del calendario (CalendarDayComp.vue):Las tarjetas de los días muestran el número del día, festividad y tarea	1
Gestión de eventos (GestionView.vue): La vista permite agregar, modificar y eliminar eventos	0,5
Gestión de eventos (GestionView.vue): usa Pinia para gestionar el estado global.	0,5
Gestión de eventos (GestionView.vue): Incluye el formulario EventFormComp.vue para añadir eventos.	0,5
Formulario de eventos (EventFormComp.vue): El formulario contiene un select para elegir entre "Día festivo" y "Tarea",	0,5
Formulario de eventos (EventFormComp.vue): permite ingresar nombre y fecha con validaciones	0,5
Formulario de eventos (EventFormComp.vue): Los datos se guardan en Pinia y localStorage.	0,5
Funcionalidad de la barra de navegación (NavbarComp.vue):contiene los enlaces a las vistas de calendario y gestión	0,5
Funcionalidad de la barra de navegación (NavbarComp.vue): ncluye el selector de idioma con i18n.	0,5
Lógica del calendario (useCalendar.ts): filtra los eventos de acuerdo con el mes.	0,5
Lógica del calendario (useCalendar.ts): El composable genera correctamente la estructura del calendario según el mes seleccionado	0,5
Gestión del estado global (Pinia Store): gestiona el estado global de eventos y mes seleccionado	0,5
Gestión del estado global (Pinia Store): Utiliza subscribe para sincronizar con localStorage	0,5
Uso de localStorage: Los eventos y el mes seleccionado se guardan y recuperan correctamente desde localStorage.	0,5
Validaciones y manejo de errores: Se realizan validaciones para evitar datos vacíos	0,25
Validaciones y manejo de errores: se manejan correctamente los errores en el formulario.	0,25
Estilo y organización del código correcto	0,25
Comentarios para hacer un buen seguimiento del proyecto	0,25
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.