"Detective en la Escena del Crimen"

Este juego/ejercicio se ha desarrollado para practicar los tipos de Slots.

Requerimientos

1. Temática Thriller:
   - El escenario es una misteriosa escena de crimen.
   - Las pistas deben incluir elementos como "huellas", "un vaso roto" y "un sobre sospechoso".
     - Las pistas se pasan como props al componente ClueList.
     - Se filtran y ordenan según su nivel de importancia.
   - Resuelves el caso (resolveCase)
     - Evalúa si las pistas clave están presentes y tienen suficiente importancia.
     - Si se cumplen las condiciones, muestra un mensaje indicando que el caso está resuelto.

2. Componentes:
   - `CrimeScene.vue`:
     - Muestra una representación general de la escena del crimen.
     - Utiliza un default slot para contenido dinámico.

   - `ClueList.vue`:
     - Enumera pistas encontradas en la escena, usando scoped slots para personalizar la presentación de cada pista.

   - `DialogueBox.vue`:
     - Simula un cuadro de diálogo entre el detective y un testigo, utilizando named slots.
     - Permite modificar encabezado, contenido y pie mediante named slots.

   - Botón "resolver el caso": llama a resolveCase.

3. Funcionalidad:
   - Los estudiantes deben pasar pistas personalizadas a la escena.
   - Permitir personalizar la apariencia del cuadro de diálogo entre el detective y el testigo.
   - Mostrar cada pista utilizando scoped slots para incluir detalles adicionales (como el nivel de importancia).
   - Ordenar las pistas por nivel de importancia.
   - Crear un botón para "resolver el caso". Comprobará si las pistas elegidas por el jugador son más del 60% de las pistas “importantes” que reuelven el caso. Si es así muestra una alerta con el mensaje “¡Enhorabuena ! Has resuelto el caso”, en caso contrario “No has acertado intentalo de nuevo” y se resetea el juego para volver a jugar.

Indicaciones

1. Configura el entorno con Vue 3 + TypeScript, se recomienda usar vue@latest.
2. Crea los componentes y organiza el proyecto con una carpeta `components`.
3. Inserta contenido dinámico en `CrimeScene.vue` usando un default slot.
4. Personaliza las pistas en `ClueList.vue` utilizando scoped slots.
5. Cambia los encabezados y pies de página del diálogo en `DialogueBox.vue` con named slots.
6. Añade tu propio contenido o pistas adicionales para practicar.

Rúbrica de Corrección

Puntuación máxima: 10 puntos

1. Configuración Inicial (1 punto)

- **0.5 puntos:** Configura correctamente el entorno con Vue 3 + TypeScript (uso de `vue@latest`).
- **0.5 puntos:** Estructura el proyecto organizando los componentes en una carpeta `components`.

2. Componente `CrimeScene.vue` (1.5 puntos)

- **0.5 puntos:** Implementa un default slot que permita insertar contenido dinámico correctamente.
- **0.5 puntos:** Representa visualmente la escena del crimen con elementos temáticos.
- **0.5 puntos:** Muestra integración clara con otros componentes y contenido dinámico.

3. Componente `ClueList.vue` (2 puntos)

- **0.5 puntos:** Recibe correctamente las pistas como props, incluyendo el nivel de importancia.
- **0.5 puntos:** Ordena y filtra las pistas correctamente según su nivel de importancia.
- **1 punto:** Implementa scoped slots que personalizan la presentación de cada pista de manera efectiva.

4. Componente `DialogueBox.vue` (1.5 puntos)

- **0.5 puntos:** Implementa named slots correctamente para el encabezado, contenido y pie.
- **0.5 puntos:** Permite personalizar de manera efectiva la apariencia del cuadro de diálogo.
- **0.5 puntos:** Presenta coherencia entre el cuadro de diálogo y el resto de la temática.

5. Funcionalidad del botón "resolver el caso" (2 puntos)

- **1 punto:** Implementa correctamente `resolveCase`, verificando las pistas clave y su nivel de importancia.
- **0.5 puntos:** Muestra un mensaje (alerta o similar) que indique si el caso está resuelto correctamente.
- **0.5 puntos:** Detecta y lista las pistas principales de forma efectiva en el mensaje.

6. Uso de TypeScript (1 punto)

- **0.5 puntos:** Define correctamente las interfaces o tipos para props y datos (por ejemplo, estructura de pistas).
- **0.5 puntos:** Utiliza TypeScript en métodos y propiedades de los componentes (por ejemplo, tipado en `resolveCase`).

7. Criterios Generales (1 punto)

- **0.5 puntos:** Código limpio y legible con buena indentación y prácticas recomendadas.
- **0.5 puntos:** La aplicación funciona sin errores importantes que impidan la ejecución o el uso de funcionalidades principales.