Entrega 3 2º trimestre: Bootstrap

- Modo de entrega: Archivo comprimido zip. Tanto la carpeta como el comprimido han de llevar el nombre en formato Entrega2-ApellidoApellidoNombre.zip
- Todos los ejercicios han de estar en un proyecto de Vue, los tres primeros sobre Typescript pueden estar en el componente principal.
- Fecha de entrega: 16/02/2025

1. En typescript realiza el siguiente ejercicio:
    a. Crea una función llamada calcularAreaRectangulo que reciba dos parámetros: ancho y alto, ambos de tipo number, y retorne también un number.
    b. Define otra función imprimirArea que reciba un parámetro area de tipo number y muestre en consola "El área es: <area>".
    c. Llama a calcularAreaRectangulo con valores de tu elección y pasa el resultado a imprimirArea.
2. En typescript realiza el siguiente ejercicio:
    a. Declara una interfaz Persona con propiedades nombre: string y edad: number.
    b. Crea una variable identificador que pueda ser de tipo string o number (unión de tipos).
    c. Escribe una función obtenerSaludo que reciba un parámetro de tipo Persona y retorne un string con el saludo: "Hola <nombre>, tu edad es <edad>".
    d. Muestra en consola el resultado de obtenerSaludo usando un objeto que cumpla la interfaz Persona.
    e. 5. Asigna diferentes valores a identificador (string y number) para verificar el comportamiento del union type.
3. En typescript realiza el siguiente ejercicio:
    a. Define una interfaz Producto con propiedades id: number, nombre: string, precio: number.
    b. Crea una función crearProductoParcial que reciba un objeto de tipo Partial<Producto> y retorne un objeto Producto completo, asignando valores predeterminados a cualquier propiedad faltante.

   c. Define otra función obtenerProductosPorId que reciba un Record<number, Producto> y un id: number, y retorne el Producto correspondiente o undefined si no existe.
4. Crea un componente Timer.vue.
   a. Declara una variable tiempo (string) que muestre la hora actual (por ejemplo, HH:mm:ss).
   b. Usa onMounted() para:
      i. Iniciar un setInterval que actualice tiempo cada segundo.
   c. Usa onUnmounted() para:
      i. Limpiar el intervalo y evitar memory leaks.
   d. Renderiza la hora en el template para visualizar un reloj en tiempo real.
5. En este ejercicio usaremos una funcionalidad no vista en clase como son los slots:
   a. Crea un componente ModalContainer.vue con:
      i. Una prop visible (boolean) para controlar si el modal está abierto.
      ii. Un slot para el contenido principal del modal.
      iii. Un slot para las acciones (botones, etc.) del modal, opcional.
   b. En un componente padre, usa ModalContainer y coloca contenido en sus slots.
6. Crea un componente SimpleCalculator.vue.
   a. Declara dos variables reactivas: num1 y num2 (ambas ref<number>).
   b. Crea tres propiedades computadas:
      i. suma (retorna num1.value + num2.value)
      ii. producto (retorna num1.value * num2.value)
      iii. diferencia (retorna num1.value - num2.value)
   c. Muestra en el template un input para num1, otro para num2 (con v-model), y luego <p> o para exponer la suma, producto y diferencia.
   d. Observa cómo cambiar los valores de num1 o num2 recalcula automáticamente las propiedades computadas.
7. Crea un componente FilteredList.vue.
   a. Declara un array reactivo items (por ejemplo, [{ nombre: 'Manzana' }, { nombre: 'Melón' }, ...]).
   b. Declara una variable filtro (string) con ref('').
   c. Crea una propiedad computada itemsFiltrados que devuelva solo los ítems cuyo nombre contenga el valor de filtro.
   d. Agrega un watch que imprima en consola cuando filtro cambie.
   e. En el template, muestra un input con v-model="filtro" y renderiza la lista de itemsFiltrados.

8. Crea un componente LocalStorageExample.vue.
    a. Declara un ref<string> llamado nota, que representará algún texto que el usuario ingrese.
    b. Al montarse el componente (onMounted()), revisa si existe en localStorage una clave "miNota" y, si es así, asigna su valor a nota.
    c. Usa un watch(nota, ...) para que cada vez que nota cambie, se guarde en localStorage.
    d. Muestra un <textarea> con v-model="nota" para que el usuario escriba.