

Registro de Partidos de Tenis

Parte 1: Implementación de Clases

1. Clase Jugador
 - Atributos privados:
 - Nombre (string): Debe tener entre 3 y 50 caracteres.
 - Edad (número): Entre 10 y 60 años.
 - Métodos:
 - constructor(nombre, edad): Inicializa los atributos validando sus valores.
 - toString(): Devuelve una representación textual del jugador.
 - Setters y getters con validaciones que lancen errores si los valores no cumplen las restricciones.
2. Clase Partido
 - Atributos privados:
 - Jugadores (array de 2 objetos Jugador).
 - Sets (array de strings en formato X-Y, donde X y Y son números).
 - Métodos:
 - constructor(jugadores, sets): Inicializa los atributos validando los valores.
 - toString(): Devuelve una descripción del partido.
 - Setters y getters con validaciones.

Cada clase debe estar en un archivo independiente y usar la sintaxis de módulos de ES6.

Parte 2: Formulario de Registro

Diseñarás un formulario en HTML para introducir los datos de un partido de tenis.

1. Estructura del Formulario
 - Jugador 1 y Jugador 2:
 - Campos: Nombre y Edad (en la misma fila).
 - Validaciones:
 - Nombre: Solo letras y espacios, entre 3 y 50 caracteres.
 - Edad: Número entre 10 y 60 años.
 - Resultados de Sets:
 - Tres campos (Set 1, Set 2, Set 3) en formato X-Y.
 - Validación: Solo acepta el formato numérico (ejemplo: 6-3).
2. Diseño del Formulario
 - Usa el archivo proporcionado (index.html) como base.
 - Mantén el fondo temático de pista de tenis y el diseño responsivo.

Parte 3: Validación de Formulario

1. Usa HTML5 para definir reglas básicas (required, pattern, min, max).
2. Crea un archivo index.js que:
 - Valide cada campo con la función checkValidity().
 - Muestre errores personalizados si algún campo no cumple las reglas.
 - Resalte los campos con errores en rojo.
 - Limpie los errores cuando el usuario corrija los datos.
3. Interactividad
 - Si el formulario es válido, muestra un mensaje de éxito en pantalla.
 - Si no es válido, evita el envío y muestra un mensaje global indicando los errores.

Parte 4: Funcionalidad Adicional

1. Una vez enviado el formulario:

- Crea instancias de los objetos Jugador y Partido con los datos introducidos.
 - Usa `console.log` para mostrar los detalles del partido registrado.
2. Si ocurre algún error durante la creación de los objetos, captura la excepción y muestra el mensaje en consola.

Consideraciones

1. Organiza el código en módulos, colocando cada clase en un archivo independiente.
2. Usa sintaxis moderna (ES6).
3. Asegúrate de capturar y manejar errores con bloques `try-catch`.

Entrega

- Los siguientes archivos:
 1. Jugador.js y Partido.js con las clases implementadas.
 2. index.html con el formulario.
 3. index.js con la lógica de validación e interacción.