### **Documentación del Proyecto de Registro de Puntos: Tribu Negra vs Tribu Roja**

Daniela Wlk

#### **1. Descripción del Proyecto**

Este proyecto consiste en una aplicación web para registrar los puntos obtenidos en distintas disciplinas por dos equipos, Tribu Negra y Tribu Roja, en las intertribus PROA. La aplicación permite sumar puntos para cada equipo en deportes como handball, ajedrez y resistencia, y mostrar los resultados acumulados al final.

#### **2. Funcionalidad y Estructura del Código**

El código está estructurado en tres archivos principales:

* **HTML** (index.html): estructura del contenido y los elementos visuales de la página.
* **JavaScript/TypeScript** (archivo.js): lógica de la aplicación para gestionar el puntaje y mostrar resultados.
* **CSS** (style.css): estilos para hacer que la interfaz sea visualmente accesible y agradable.

### **HTML (index.html)**

El archivo index.html define la estructura de la página con un diseño simple y accesible. Contiene elementos clave como:

* Un contenedor principal con id="app".

<div id="app">

* Dos secciones (div) para las tribus "Negra" y "Roja", que contienen:
  + Títulos (h2) para diferenciar las tribus.

<h2>Tribu Negra</h2>

<h2>Tribu Roja</h2>

* + Campos de entrada (input) para ingresar puntos por disciplina.

<input type="number" id="handball-roja" placeholder="Handball">

<input type="number" id="resistencia-roja" placeholder="Resistencia">

<input type="number" id="ajedrez-roja" placeholder="Ajedrez">

Para la tribu negra es lo mismo pero el id es “deporte-negra”

* Botones para agregar puntos a cada tribu, llamando a la función agregarPuntos.

<button onclick="agregarPuntos('Tribu Negra')">Agregar Puntos</button>

* Un botón para mostrar los resultados, llamando a la función mostrarResultados.

<button onclick="mostrarResultados()">Mostrar Resultados</button>

* Un párrafo (p) con id="resultado" para mostrar el puntaje acumulado y el equipo líder.

### **<p id="resultado"></p>**

### 

### 

### 

### **JavaScript/TypeScript (archivo.js)**

El archivo archivo.js contiene la lógica para:

1. **Registrar los puntos**: Se utiliza la función agregarPuntos, que obtiene los valores ingresados en cada disciplina (Handball, Resistencia, Ajedrez) y actualiza el puntaje total de la tribu correspondiente.
   * Nota: Inicialmente, esta función también contenía el código para mostrar los resultados. Sin embargo, esto ocasionó problemas, ya que el botón "Agregar Puntos" también mostraba los resultados. Se solucionó moviendo la función de mostrar resultados a un botón separado, asegurando que cada botón realice su acción específica.

<button onclick="agregarPuntos('Tribu Negra')">Agregar Puntos</button>

<button onclick="agregarPuntos('Tribu Roja')">Agregar Puntos</button>

1. **Mostrar los resultados**: La función mostrarResultados muestra el puntaje acumulado de ambas tribus y un mensaje indicando cuál tribu está ganando o si hay un empate.

function mostrarResultados(): void {

if (Tribunegra.punto > Triburoja.punto) {

OutputResultado.innerHTML += "¡Tribu Negra está ganando!";

} else if (Tribunegra.punto < Triburoja.punto) {

OutputResultado.innerHTML += "¡Tribu Roja está ganando!";

} else {

OutputResultado.innerHTML += "¡Es un empate!";

}

Tuve un problema cuando estaba programando en TypeScript: al principio no puse la función que sumaba los puntos. Después, cuando la agregué, le puse también que mostrara los resultados. Esto hacía que al tocar el botón de “Agregar Puntos”, además de sumar, mostrara quién iba ganando, lo cual no era lo que el profe pidio. Para ao, quité la parte de mostrar resultados de la función agregarPuntos y dejé eso solo en el botón de “Mostrar Resultados”. Ahora cada botón hace lo que tiene que hacer: los de agregar solo suman, y el de mostrar solo muestra el puntaje y quién va ganando.

function agregarPuntos(tribu: string): void {

### **CSS (style.css)**

Para mejorar la accesibilidad y experiencia de usuario:

* **Contraste**: Utiliza colores llamativos y contrastantes (por ejemplo, negro para Tribu Negra y rojo para Tribu Roja) para distinguir visualmente las secciones.

.Tribu.roja {

background-color: #ff4d4d;

color: white;

.Tribu.negra {

background-color: #333;

color: white;

* **Botones grandes**: Los botones de "Agregar Puntos" y "Mostrar Resultados" son grandes y visibles, con cambios de color en el hover para mejorar la experiencia.

button {

background-color: #333;

color: white;

border: none;

border-radius: 8px;

padding: 10px 15px;

font-size: 1rem;

cursor: pointer;

transition: background-color 0.3s ease;

margin-top: 10px;

width: 100%;

}

button {

background-color: #ff4d4d;

color: white;

border: none;

border-radius: 8px;

padding: 10px 15px;

font-size: 1rem;

cursor: pointer;

transition: background-color 0.3s ease;

margin-top: 10px;

width: 100%;

}

* **¿Para qué use void?**

En TypeScript, voidse usa para decir que una función no devuelve nada. En este código:

- agregarPuntos: suma puntos a las tribus según el puntaje ingresado y no devuelve nada.

- mostrarResultados**:** muestra el puntaje total de cada tribu en pantalla, sin devolver nada tampoco.

**¿Qué pasaría si no está?**

Sin void, TypeScript igual sabe que las funciones no devuelven nada. Pero ponerlo me ayudo a que el código sea fácil de entender.

function agregarPuntos(tribu: string): void {

function mostrarResultados(): void {