DigitalHouse>
Coding School

Certified Tech Developer
The Ultimate Degree

Programación Imperativa

Práctica integradora

Objetivos

Para este ejercicio utilizaremos todos los elementos de las clases anteriores y de la actual. Los objetivos son:

- Reconocer **patrones**, elementos con características generales que interactúan, o no, con otros. Poder crear grupos y relaciones.
- **Abstraer** en conceptos y objetos computables los elementos del problema.
- **Descomponer** en conceptos y objetos de menor complejidad, dividir el problema en partes más pequeñas.
- Modelar el problema de la consigna bajando a papel o planilla de cálculos la representación en números, textos o gráficos de estos elementos.
- Expresar el modelo en formato de código, **algoritmizando** la solución.

¡No tengas miedo! Son formas precisas de describir algo que el cerebro ya hace en situaciones cotidianas, solo que lo transformamos para que lo pueda hacer un procesador.

Descripción del problema

Hace falta determinar el ganador de un concurso de stand up que consta de 3 presentaciones por participante.
El público, máximo 100 personas, votó quién considera que estuvo mejor al terminar cada etapa. Por ejemplo, sube Alicia, se presenta y baja; sube Bob, se presenta y baja. Terminada la presentación, el público vota indicando quién cree que ganó esa vuelta. Luego, continúa la próxima tanda igual que la primera. Y finalmente una tercera.

Si quisieras seguir esta final, ¿cómo representarías en una tabla tipo planilla de cálculos este problema?

Participante	votos etapa 1	ganador v1	vot etapa 2	G v2	v etapa 3	G v3	ganador

Como estos concursos se dan online, ocurren miles cada día. Nos llega a nuestro servidor la información por cada concurso en formato de arrays, uno por cada participante, por lo tanto, los recibimos así:
El array de Alicia es: a = [23, 82, 46]
El array de Bob es: b = [45, 8, 32]

Vayan planteando, entonces, en código mientras piensan cómo realizar la comparación de cada etapa.

```
const a = [23, 45, 32];
const b = [12, 67, 7];

function encontrarGanador(a, b) {
    //su solución aquí
}
```

La tarea consiste en enfrentar estas votaciones de ambos comparando a[0] con b[0], a[1] con b[1] y a[2] con b[2].

> Si a[i] > b[i], *entonces, Alicia recibe 1 punto.*
> Si a[i] < b[i], *entonces, Bob recibe 1 punto.*
> Si a[i] = b[i], *ninguna persona recibe un punto.*

Los puntos de comparación son los puntos totales que ganó una persona. ¡Ojo! No los votos, sino los puntos ganados en cada etapa.

Ejemplo

a = [1, 2, 3]

b = [3, 2, 1]

Para los elementos en el índice 0, Bob recibe un punto porque b[0] > a[0].
Para la etapa siguiente a[1] y b[1] son iguales, no se obtienen puntos.
Finalmente, para los elementos del índice 2 (tercera etapa), a[2] > b[2] por lo que Alicia recibe un punto.

```
const a = [23, 45, 32];
const b = [12, 67, 7];

function encontrarGanador(a, b) {
 let puntosPrimerParticipante = 0;
 let puntosSegundoParticipante = 0;

 // primera etapa
 if (a[0] > b[0]) {
   puntosPrimerParticipante = puntosPrimerParticipante + 1;
 } else if (a[0] < b[0]) {
   puntosSegundoParticipante = puntosSegundoParticipante + 1;
 }

 // segunda etapa
 if (a[1] > b[1]) {
   puntosPrimerParticipante += 1;
 } else if (a[1] < b[1]) {
   puntosSegundoParticipante += 1;
 }

 // tercera etapa
```

```
 if (a[2] > b[2]) {
   puntosPrimerParticipante++;
 } else if (a[2] < b[2]) {
   puntosSegundoParticipante++;
 }

 // comparación final
 if (puntosPrimerParticipante > puntosSegundoParticipante) {
   return "primer";
 } else {
   return "segundo";
 }
}

console.log("El ganador es: " + encontrarGanador(a, b) + " participante");
```

Pero hay mucho código repetido ahí, ¿qué podemos usar para simplificar?

Consigna:

Realizar una adaptación del código mostrado arriba usando la estructura FOR que contemple evitar la redundancia de estructuras IF.