

TRABAJO PRÁCTICO NRO. 04 – JAVASCRIPT

- Objetos a partir de clases
- Objetos literales
- Rest Operator
- Spread Operator
- Conversión a formato JSON
- Destructuring

Especificaciones:

- Utilizar manejo del DOM
- Crear documento HTML y CSS a gusto del alumno
- Definir el archivo controlador.js que capture los elementos con los que interactuará el usuario y agregar los comportamientos de los objetos/elementos HTML utilizando “addEventListener”.
- Si el programa, necesita una ó mas funciones debe (expresarlas como arrow function). Y si son funciones específicas de cálculos, funciones específicas de la resolución del problema debe aislarlas en un archivo separado denominado “modelo.js”. allí deberá tener su conjunto de funciones y exportarlas para que el controlador.js pueda consumirlas.
- Desde el archivo “controlador.js” deberá importar esas funciones que están en el archivo “modelo.js”.

Ejercicio Nro. 18: (spread operator ... y métodos de vectores)

Un profesor de programación tiene 4 curso o cohortes, en donde en cada cohorte existen alumnos diferentes que cursaron su módulo de “fundamentos de programación con JavaScript”, el profesor tiene en un vector diferente los alumnos y sus notas finales del cursado. “Se aprueba con notas > 4”. El listado de alumnos es el siguiente:

```
const cohorte01 = [  
  { dni: "12345678", nombre: "Juan", apellido: "Pérez", nota_final: 8 },  
  { dni: "23456789", nombre: "Ana", apellido: "Gómez", nota_final: 6 },  
  { dni: "34567890", nombre: "Carlos", apellido: "López", nota_final: 5 },  
  { dni: "45678901", nombre: "María", apellido: "Martínez", nota_final: 7 }  
];
```

```
const cohorte02 = [  
  { dni: "56789012", nombre: "Sofía", apellido: "Ramírez", nota_final: 9 },  
  { dni: "67890123", nombre: "Luis", apellido: "Fernández", nota_final: 4 },  
  { dni: "78901234", nombre: "Elena", apellido: "Torres", nota_final: 5 }  
];
```

```
const cohorte03 = [  
  { dni: "89012345", nombre: "Pedro", apellido: "Álvarez", nota_final: 10 },  
  { dni: "90123456", nombre: "Clara", apellido: "Méndez", nota_final: 3 },  
  { dni: "12345679", nombre: "Jorge", apellido: "Salinas", nota_final: 6 },  
  { dni: "23456780", nombre: "Andrea", apellido: "Cruz", nota_final: 2 },  
  { dni: "34567891", nombre: "Sergio", apellido: "Paredes", nota_final: 8 }  
];
```

```
const cohorte04 = [  
  { dni: "45678902", nombre: "Lucía", apellido: "Ortiz", nota_final: 7 },  
  { dni: "56789013", nombre: "Miguel", apellido: "Vega", nota_final: 4 },  
  { dni: "67890124", nombre: "Raquel", apellido: "Silva", nota_final: 9 },  
  { dni: "78901235", nombre: "Hugo", apellido: "Moreno", nota_final: 5 },  
  { dni: "89012346", nombre: "Natalia", apellido: "Quinteros", nota_final: 6 },  
  { dni: "90123457", nombre: "Diego", apellido: "Arce", nota_final: 3 }  
];
```

- 1) Realice una función, que reciba como parámetro un vector de alumnos (puede ser de cualquier cohorte) los recorra con un **forEach** y muestre por consola, el apellido y el nombre.

Invoque esa misma función, pasándole como parámetro el vector de la cohorte01, luego el vector de la cohorte02 y así sucesivamente hasta la cuarta cohorte.

- 2) El profesor desea unificar todos los alumnos en un solo vector, dado que estructuralmente son iguales, utilice **spread operator** para “unificar el contenido de las cuatro cohortes” en un solo vector. Muestre por consola el vector resultante.
- 3) Sobre los datos unificados, cree dos vectores nuevos, uno puede llamarse alumnosAprobados y el otro alumnosDesaprobados.

Para crear el vector de alumnos aprobados, aplique el método **filter** sobre el Vector que tiene todos los alumnos cuando la nota > 5;

Para crear el vector de alumnos desaprobados, aplique el método **filter** sobre el Vector que tiene todos los alumnos cuando la nota <= 5;

- 4) Sobre el vector Total de alumnos (cuatro cohortes), aplique el método “**sort**” = ordenar y ordene los datos de todos esos alumnos por nota de menor a mayor.
- 5) Sobre el vector Total de alumnos, realice una búsqueda de los siguientes alumnos aplicando el método “**find**” que tienen los vectores incorporados y busque algún alumno por su “dni”. El que ud. Desea.
- 6) Sobre el vector Total de alumnos, deseamos “contabilizar” la cantidad de alumnos aprobados, es decir la cantidad de alumnos que tienen nota > 5; para ello utilicen el método reduce que tienen los vectores incorporados para contabilizar. Realice esto de dos formas
 - a. Usando una arrow function que a dentro utilice un condicional if/else para saber si la nota > 5 y por lo tanto contabilizar los aprobados;
 - b. Usando una arrow function que a dentro utilice un operador ternario para saber si la nota > 5 y por lo tanto contabilizar los aprobados;

Ambas formas deben llegar al mismo resultado, únicamente que se implementaron de dos formas diferentes.