NODE JS

Práctica semanal 5

adn POTENCIÁ TU FUTURO

Nación Servicios

IT Patagonia

Codeki™


Ejemplo 1

01

Enunciado

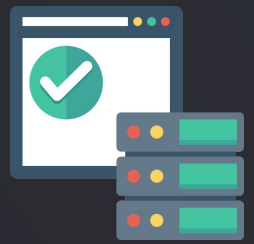
Un supermercado desea evaluar sus ventas diarias durante el mes que pasó. Para ello se necesita ingresar en correspondientes arreglos los ingresos y los costos obtenidos en cada día del mes.

Se pide:

a) Determinar la ganancia total del mes.

b) ¿Qué días del mes presentan ganancias negativas?

c) ¿Cuántos días hay con ingreso mayor al promedio?



Top-Down

```
// 1 Leer Datos
// 2 Calcular ganancia total del mes
// 3 Dias con ganancias negativas
// 4 Cantidad de dias con ingresos mayor al promedio
// 5 Mostrar datos
```

Top-Down

Nación Servicios

```
// 1.1 Leer cantidad de días N
// 1.2 Leer Costos
// 1.3 Leer Gananacia
```

Top-Down

```
// 2.1 Sumar todas las ganancias
// 2.2 Sumar todos los costos
// 2.3 Ganancias = Suma Ganancias - Suma Costos
```

Top-Down

```
// 3.1 si ganancia - costo < 0
// 3.2 contar dia de perdida
```

Top-Down

```
// 4.1 Promedio Ganancias / N
// 4.2 si ( ganancia - costo > promedio )
// 4.3 contarMayores
```

Top-Down

```
5.1 Mostrar Ganancias totales
5.2 Mostrar cantidad de dias con pérdidas
5.3 Mostrar cantidad de dias con ingresos mayores al promedio
```

Función calcularGanancia

```
/**
 *
 * @param {*} mensaje
 * @returns
 */
const leerDatos = (mensaje) => parseFloat(prompt(mensaje));
```

Función calcularGanancia

```
/**
 *
 * @param {*} costo
 * @param {*} ganancia
 * @returns
 */
const calcularGanancia = ( gananciaTotal , costo , ganancia ) => (gananciaTotal +=(ganancia - costo));
```

Función contarDiasConPerdidas

Nación Servicios

```
/**
 *
 * @param {*} cantidadPerdidads
 * @param {*} costo
 * @param {*} ganancia
 * @returns
 */
const contarDiasConPerdidas = ( cantidadPerdidads , costo , ganancia ) => ( !(ganancia - costo > 0 )? (cantidadPerdidads + 1): cantidadPerdidads );
```

Función calcularDiasMayores

Nación Servicios

```
/**
 *
 * @param {*} cantidadDiasMayores
 * @param {*} costo
 * @param {*} ganancia
 * @param {*} gananciaTotal
 * @param {*} N
 * @returns  Cantida de dias con ingresos mayores al promedio
 */
const calcularDiasMayores = ( cantidadDiasMayores , costo , ganancia , gananciaTotal , N ) =>{
        return (( ganancia - costo ) > ( gananciaTotal/N ))? (cantidadDiasMayores + 1):( cantidadDiasMayores );
}
```

Función App

```
const App = () => {
    let costos = [];
    let ganancias = [];
    let gananciaTotal = 0;
    let cantidadPerdida = 0;
    let cantidadMayoresPromedio = 0;
    let N = leerDatos('Ingrese cantidad de datos');
        for (let i = 0; i < N; i++) {
            costos[i] = leerDatos('Ingrese costo '+(i+1)+' :');
            ganancias[i] = leerDatos('Ingrese ganancia '+(i+1)+' :');
            gananciaTotal = calcularGanancia(gananciaTotal,costos[i],ganancias[i]);
        }
        for (let i = 0; i < N; i++) {
            cantidadPerdida = contarDiasConPerdidas( cantidadPerdida , costos[i] , ganancias[i] );
            cantidadMayoresPromedio = calcularDiasMayores( cantidadMayoresPromedio , costos[i] , ganancias[i] , gananciaTotal );
        }
        console.log('Ganancias Totales -> '+gananciaTotal);
        console.log('Cantidad de días con perdidas ->'+cantidadPerdida);
        console.log('Cantidad de dias con ingresos mayores al promedio ->'+cantidadMayoresPromedio);

}
```

Invocación de función principal App()

```
App();
```


Top-Down

02

Ejercicio 2

Se realiza una encuesta entre una cantidad N de familias. Se les solicita el ingreso mensual de dinero de cada familia, y el número de integrantes del grupo familiar.

02

Ejercicio 2

a) Listar los datos correspondientes a las familias con un número impar de integrantes, mayor a tres y un ingreso superior a $1200.

b) Obtener el promedio de ingreso per cápita de toda la encuesta.

c) Mostrar los datos correspondientes al grupo familiar con menor ingreso, incluyendo el orden, en el que ingresó.

Top Down

```
// 1 Leer Datos
// 2 Calcular familias con un número impar de integrantes, mayor a tres y un ingreso superior a $1200.
// 3 Calcular promedio de ingreso per cápita de toda la encuesta.
// 4 Calcular los datos correspondientes al grupo familiar con menor ingreso, incluyendo el orden, en el que ingreso.
// 5 Mostrar datos calculados
```

Top Down

Nación Servicios

```
// 1.1 Leer cantidad N de datos
// 1.2 Leer Ingresos
// 1.3 Leer Integrantes
```

Top Down

```
// 2.1 si Integrantes % 2 != 0 && Integrantes > 3 && Ingresos>1200
// 2.2 si contarImpar++
```

Top Down

Nación Servicios

```
// 3.1 sumar ingresos de todas las familias
// 3.2 sumar integrantes de todas las familias
// 3.3 promedioPercápita = sumaIngresos/sumaIntegrantes
```

Top Down

```
// 4.1 a IngresoMenor lo inicializo con un número muy grande
// 4.2 a si IngresoMenor > Ingresos
// 4.3 a IngresosMenor le asigno Ingresos
// 4.4 posicionMenor = i;
```

Top Down

```
// 5.1 Mostrar cantidad de impares
// 5.2 Mostrar promedioPercápita
// 5.3 Mostrar Ingresos Menores y posición de Ingresos Menores
```

Código JavaScript

```javascript
/**
 *
 * @param {*} mensaje
 * @returns Datos
 */
const leerDatos = ( mensaje ) => parseFloat(prompt( mensaje ));
```

Código JavaScript

```javascript
/**
 *
 * @param {*} contadorImpar
 * @param {*} integrantes
 * @param {*} ingresos
 * @returns Cantidad de familias con un número impar de integrantes, mayor a tres y un ingreso superior a $1200.
 */
const contarImpares = ( contadorImpar , integrantes , ingresos ) => ((integrantes % 2 != 0) && (integrantes > 3) && ingresos>1200)? contadorImpar + 1: contadorImpar;
```

Código JavaScript

```javascript
/**
 *
 * @param {*} dato
 * @param {*} sumaDato
 * @returns Suma de los datos
 */
const sumarDatos = ( dato , sumaDato ) => sumaDato + dato;
```

Código JavaScript

Nación Servicios

```javascript
/**
 *
 * @param {*} suma
 * @param {*} cantidad
 * @returns Promedio
 */
const calcularPromedio = ( suma , cantidad ) => suma / cantidad;
```

Código JavaScript

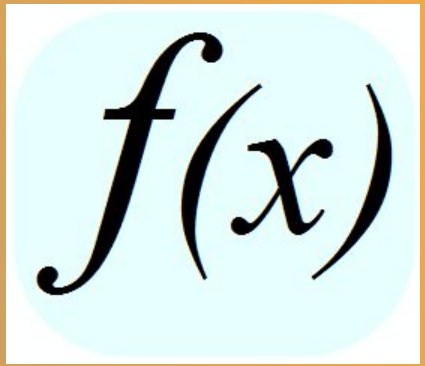
```
const App = () => {
    let ingresos = [];
    let integrantes = [];
    let sumaIntegrantes = 0;
    let sumaIngresos = 0;
    let promedio = 0;
    let contadorImpares = 0;
    let menor = 99999999;
    let posicionMenor =0;
    let N = leerDatos('Ingrese numero de familias :');
        for (let i = 0; i < N; i++) {
            ingresos = leerDatos('Ingreso familia '+(i+1)+' :');
            integrantes = leerDatos('Integrantes familia '+(i+1)+' :');
        }
        for (let i = 0; i < N; i++) {
            contadorImpares = contarImpares(contadorImpares,integrantes[i],ingresos[i]);
            sumaIntegrantes = sumarDatos(integrantes[i],sumaIntegrantes);
            sumaIngresos = sumarDatos(ingresos[i],sumaIngresos);
            [ menor , posicionMenor ] = calcularPosYDatoMenor(menor,posicionMenor,ingresos[i],i);
        }
        promedio = calcularPromedio(sumaIngresos,sumaIntegrantes);

        console.log('Cantidad de impares :'+contadorImpares);
        console.log('promedioPercápita :'+promedio);
        console.log('Ingresos Menores y posición de Ingresos Menores $'+menor+' poscision: '+posicionMenor);


}
```

Funciones

```
App();
```



¡Muchas gracias!