

UNIVERSIDAD DON BOSCO ESCUELA DE INGENIERIA



DESARROLLO DE APLICACIONES WEB CON SOFTWARE INTERPRETADO EN EL CLIENTE

Taller #1

Grupo de Laboratorio: G03L

Apellidos	Nombres	Carnet
Alvarenga Vásquez	Carlos Benjamín	AV221218

Fecha de presentación: viernes 09 de septiembre, 2022.

Ejercicio 1:

- Utilización de arreglos para creación de la matriz bidimensional (100%):

```
let matriz = []
for(let i = 1; i <= dimensiones; i++){
  let aux = []
  for(let j = 1; j <= elementos; j++){
    let num = Math.round( Math.random() * range )
    aux.push(num)
  }
  matriz.push(aux)
}
return matriz
```

- Función crear_matriz donde se generará la matriz bidimensional, llenándola con números aleatorios (100%):

```
// Creando funcion crear_matriz que se vale de ciclos for para crear una matriz multidimensional
function crear_matriz(dimensiones, elementos){

  let matriz = []
  for(let i = 1; i <= dimensiones; i++){
    let aux = []
    for(let j = 1; j <= elementos; j++){
      let num = Math.round( Math.random() * range )
      aux.push(num)
    }
    matriz.push(aux)
  }
  return matriz
}
```

- Función mostrar con parámetro de tipo arreglo, donde se crea el código JavaScript para mostrar los datos del arreglo en una tabla en el HTML, identificando la posición del número mayor y menor dentro de ella (100%):

```
// Creando funcion para mostrar los elementos de la matriz
function mostrar(arreglo){
    let min_max = {min : arreglo[0][0], max : arreglo[0][0]}

    let docFragment = new DocumentFragment()
    for(fila in arreglo){
        let tr = document.createElement('tr')
        for(columna in arreglo[fila]){
            let current_number = arreglo[fila][columna]
            let td = document.createElement('td')
            let text = document.createTextNode(current_number)
            td.dataset.num = current_number
            td.appendChild(text)
            tr.appendChild(td)

            // evaluacion min max
            if(current_number == 0){
                min_max.min = current_number
            }
            else{
                if(current_number < min_max.min){
                    min_max.min = current_number
                }
                else if(current_number > min_max.max){
                    min_max.max = current_number
                }
                else {
                    continue
                }
            }
        }
        docFragment.appendChild(tr)
    }
    return {docFragment, min_max}
}
```

- La acción del evento del botón se genera desde el código JavaScript (100%):

```
// Anadiendo un eventListener de tipo 'click' al elemento generate
generate.addEventListener('click', ()=>{

    // Evaluando si los datos necesarios estan presentes
    if(inDimensions.value == 0){
        inDimensions.focus()
        return
    }
    if(inElements.value == 0){
        inElements.focus()
        return
    }

    // creando una nueva matriz usando crear_matriz()
    let new_matrix = crear_matriz(inDimensions.value, inElements.value)

    // creando un fragmento de html con todo el contenido de la matriz creada anteriormente
    // y extrayendo los numeros maximos y minimos
    let {docFragment : mainFragment, min_max} = mostrar(new_matrix)

    // limpiando y anadiendo el fragmento de html a la tabla
    table.innerHTML = ''
    table.appendChild(mainFragment)

    // escribiendo los valores maximos y minimos
    minmaxHtml.innerHTML = ''
    let secFragment = mostrar_minmax(min_max.min, min_max.max)
    minmaxHtml.appendChild(secFragment)

    // Pintando el maximo y minimo en el html
    table.querySelectorAll('tr').forEach(tr=>{
        let mintd = tr.querySelector(`[data-num = "${min_max.min}"]`)
        let maxtd = tr.querySelector(`[data-num = "${min_max.max}"]`)
        if(mintd !== null){
            mintd.classList = 'min'
        }
        if(maxtd !== null){
            maxtd.classList = 'max'
        }
    })
})
```

- Realización de enlace de documento JavaScript con el HTML (100%):

```
<script src="./matriz.js" defer></script>
```

- Aspecto de página web igual a la presentada en las imágenes del enunciado (100%):



Ejercicio #1: Numero Mayor y Menor

Generar Matriz

2

10

Generar matriz

79	57	48	62	64	81	44	21	54	14
16	80	28	81	64	35	5	15	14	3

El numero menor es: 3

El numero mayor es: 81

Ejercicio 2:

- Utilización de objetos para el almacenamiento de los datos ingresados de los libros (100%):

```
class GetBookData {  
  constructor(id, title, name, last, cat, price, image){  
    this.id = id.value  
    this.title = title.value  
    this.name = name.value  
    this.last = last.value  
    this.cat = cat.value  
    this.price = price.value  
    this.image = image.files[0]  
  }  
  
  getData(){  
    return {  
      id: this.id,  
      title: this.title,  
      name: this.name,  
      last: this.last,  
      cat: this.cat,  
      price: this.price,  
      image: this.image  
    }  
  }  
}
```

- Función, que permite crear la presentación de los libros en forma de cuadrícula según lo mostrado en la figura 4 (100%):

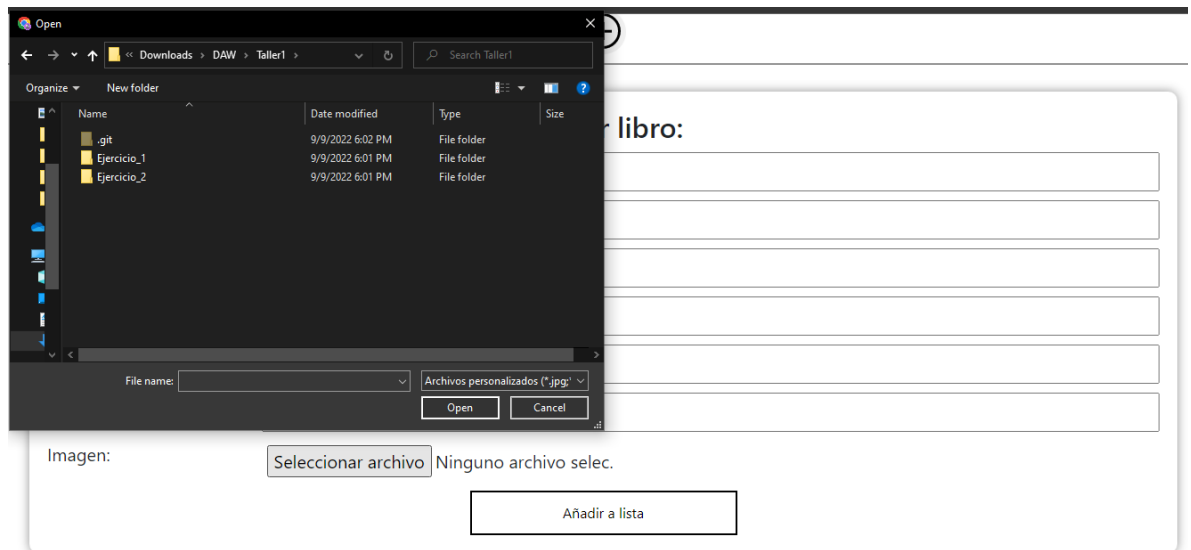
```
function crearPresentacion(data, temp){
  const template = temp.content.cloneNode(true)
  const bookElement = template.querySelector(".book-element")
  const img = bookElement.querySelector(".img")
  const id = bookElement.querySelector(".id")
  const title = bookElement.querySelector(".title")
  const auth = bookElement.querySelector(".auth")
  const cat = bookElement.querySelector(".cat")
  const price = bookElement.querySelector(".price")

  let uploadedImage = upload(data.image)
  let textId = document.createTextNode(data.id)
  let textTitle = document.createTextNode(data.title)
  let textName = document.createTextNode(data.name)
  let textLast = document.createTextNode(data.last)
  let textCat = document.createTextNode(data.cat)
  let textPrice = document.createTextNode(data.price)

  img.appendChild(uploadedImage)
  id.appendChild(textId)
  title.appendChild(textTitle)
  auth.appendChild(textName)
  auth.appendChild(textLast)
  cat.appendChild(textCat)
  price.appendChild(textPrice)

  return bookElement
}
```

- Se puede ingresar imágenes del libro desde el formulario HTML (100%):



- Los datos se ingresan desde controles HTML y se capturan los datos a través de código JavaScript (100%):

```
const list = document.querySelector(".list")
const template_element = document.querySelector("#template-element")
const addForm = document.querySelector(".add-form")
const inputs = addForm.querySelector(".inputs")
const toggler = document.querySelector(".toggle-button")

const inId = inputs.querySelector("#inId")
const inTitle = inputs.querySelector("#inTitle")
const inAuthName = inputs.querySelector("#inAuthName")
const inAuthLast = inputs.querySelector("#inAuthLast")
const inCat = inputs.querySelector("#inCat")
const inPrice = inputs.querySelector("#inPrice")
const inImage = inputs.querySelector("#inImage")
const add_button = inputs.querySelector(".add-button")
```

- Realización de enlace de documento JavaScript con el HTML (100%):

```
<script src="./libros.js" defer></script>
```