

# TRABAJO PRÁCTICO EVALUATIVO N°2

## CONSIGNAS:

### 1 – Consumo de API – Recorrer array de objetos

API: <https://www.dolarsi.com/api/api.php?type=valoresprincipales>

Deberán consumir la **API** mencionada y generar dinámicamente una **TABLA** aplicando las clases **CSS** correspondientes propuestas por **Bootstrap 5 (5.2)**. Cabe destacar, que en el diseño, tiene que figurar un **NAVBAR** y un **CONTAINER** como sectores principales. Hacer uso de filas y columnas según corresponda. La construcción de la tabla deberá estar comprendida dentro de una **función flecha**, que se invocará automáticamente.

### 2 – Arreglos

Dado el siguiente arreglo de criptomonedas hacer las siguientes operaciones:

#### Documentación de referencia:

[https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global\\_Objects/Array](https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/Array)

**A.** Crear el siguiente array tal y como se muestra a continuación. Deberá crearse de manera global. Es decir, no debe estar comprendido en ninguna función.

```
let criptos = ["Bitcoin", "Ethereum", "Tether", "BNB", "Cardano", "Dai", "Helium", "Shiba Inu", "Tron", "Cronos", "Solana"]
```

**B.** Agregar al inicio del arreglo la criptomoneda **'Stellar'**

**C.** Eliminar las criptomonedas **Shiba Inu** y **Tron** respectivamente

**D.** Agregar la criptomoneda **'Gate'** al final del arreglo.

**E.** Crear una función que imprima en un **párrafo** lo siguiente:

- a) La longitud del arreglo
- b) El último elemento del arreglo
- c) La criptomoneda **Helium** obteniendo su índice previamente con el método correspondiente

Destinar un botón exclusivamente para esta función

**F.** Crear una función que recorra el arreglo e imprimirlo en una lista. Es decir, en un **ListGroup** de **Bootstrap 5**. La función se deberá invocar a través de un botón.

### 3 – Objetos

Según el siguiente objeto, mostrar los datos solicitados.

Deberán crear una **función flecha** que al ejecutarla mediante un botón, deberá mostrar los datos que se detallan en el objeto, mostrando además sus características. Para ello, deberán imprimirlos en consola haciendo uso de los **backticks** de JavaScript (es decir, las comillas que nos permiten escribir en mas de un renglón. Con mas libertades a la hora de trabajar los datos, para un mejor ordenamiento). Todas las propiedades deberán ser almacenadas en **constantes** haciendo uso de **Destructuring Objects**

```
const framework = {  
  titulo: 'Framework utilizado en la actualidad',  
  nombre: 'Angular',  
  características: {  
    lenguaje: 'TypeScript',  
    patron: 'MVVM',  
    spa: 'si'  
  },  
}
```