

## Índice

Ejercicio 1 .....	1
Ejercicio 2 .....	2
Ejercicio 3 .....	4
Ejercicio 4 .....	4
Ejercicio 5 .....	5
Ejercicio 6 .....	7
Ejercicio 7 .....	9
Conclusiones .....	10
Acceso github .....	10

## Ejercicio 1

Se va a desarrollar un programa en siete pasos.

Cada uno de ellos supondrá la utilización de una nueva sentencia o estructura o método...en fin un cambio de desarrollo (una mejora) sobre el anterior.

Cada una de las siete versiones numeradas deben dar lugar a una nueva rama en GitHub:

tareaContinua1 a tareaContinua7.

1.- Elabora una lista de contactos

Inicialmente tendrá los valores de la tabla adjunta:

Declara e inicializa las variables donde se almacenará toda la información (nueve variables en total). Muestra en la consola información sobre el primer y último contacto en el formulario: nombre/teléfono/correo.

Nombre	Teléfono	Correo
Maxwell Wright	(0191) 719 6495	Curabitur.egestas.nunc@nonummyac.co.uk
Raja Villarreal	0866 398 2895	posuere.vulputate@sed.com
Helen Richards	0800 1111	libero@convallis.edu

```
let nombre1 = "Maxwell Wright";
let nombre2 = "Raja Villarreal";
let nombre3 = "Helen Richards";

let telefono1 = "(0191) 719 6495";
let telefono2 = "0866 398 2895";
let telefono3 = "0800 1111";

let email1 = "Curabitur.egestas.nunc@nonummyac.co.uk";
let email2 = "posuere.vulputate@sed.com";
let email3 = "libero@convallis.edu";

console.log(nombre1 + "/" + telefono1 + "/" + email1);
console.log(nombre3 + "/" + telefono3 + "/" + email3);
```

Como se puede apreciar en la imagen anterior, declaramos las variables mencionadas en el enunciado del ejercicio y las mostramos concatenadas mediante un console log en el formato indicado.

## Ejercicio 2

2.- Crea la lista de contactos en un array, de tal manera que cada uno de sus elementos será un objeto que describa a un solo usuario (tres propiedades).

Al final de la lista declarada de esta manera, agrega un nuevo contacto usando un método de array, apropiado. El nuevo contacto es: Maisie Haley / 0913 531 3030 / risus.Quisque@urna.ca.

Muestra el primer y último contacto, de nuevo en el formato: nombre / teléfono / correo electrónico. (length)

```
let array = [  
  {  
    nombre: 'Maxwell Wright',  
    telefono: '(0191) 719 6495',  
    correo: 'Curabitur.egestas.nunc@nonummyac.co.uk'  
  },  
  
  {  
    nombre: 'Raja Villareal',  
    telefono: '0866 398 2895',  
    correo: 'posuere.vulputate@sed.com'  
  },  
  
  {  
    nombre: 'Helen Richards',  
    telefono: '0800 1111',  
    correo: 'libero@convallis.edu'  
  },  
];  
  
//Añadimos una nueva posición al array de objetos  
array.push(  
  {  
    nombre: 'Maisie Haley',  
    telefono: '0913 531 3030',  
    correo: 'risus.Quisque@urna.ca'  
  }  
);
```

En este caso, declaramos un array de objetos para almacenar los datos y utilizamos el método push para añadir un nuevo contacto al array.

```
console.log(array);  
//Declaramos una variable para obtener la ultima posicion del array  
let ultimaPosicionArray = array.length-1;  
  
console.log(array[0]); //Mostramos la primera posición del array  
console.log(array[ultimaPosicionArray]); //Mostramos la última posición del array  
  
//
```

Mostramos la primera y última posición del array como se muestra en la captura anterior.

### Ejercicio 3

3.- Modifica el programa para agregar, al final de la lista, el contacto que el usuario introducirá durante la ejecución del programa. (prompt). Al final, muestra el primer y último contacto de la lista.

```
array.push({  
  nombre: prompt("Introduce el nombre del nuevo contacto, por favor"),  
  telefono: prompt("Introduce el teléfono del nuevo contacto, por favor"),  
  correo: prompt("Introduce el correo del nuevo contacto, por favor")  
});
```

Añadimos la opción de que el usuario introduzca un nuevo contacto que se añadirá al final del array.

### Ejercicio 4

4.- Modifica el programa para que el usuario pueda elegir lo que quiere hacer con la lista. Las opciones serán:

- Mostrar el primer contacto (primero)
- Mostrar el último contacto (último)
- Añadir un nuevo contacto (nuevo)

Al agregar un nuevo contacto, verifica si el usuario ha ingresado todos los datos necesarios. Si falta al menos uno de los tres valores (nombre, teléfono o correo electrónico), no se agrega el contacto.

```
//Damos al usuario la opcion de elegir que accion quiere realizar
let opcionUsuario = prompt("Las opciones son 1: mostrar el primer contacto, 2: mostrar el segundo contacto, 3: añadir un nuevo contacto; selecciona una opcion por favor");

if (opcionUsuario != null){
  switch (opcionUsuario) {
    case '1':
      console.log(array[0]); //Mostramos la primera posición del array

      break;

    case '2':
      let ultimaPosicionArray = array.length-1;
      console.log(array[ultimaPosicionArray]); //Mostramos la última posición del array
      break;

    case '3':
      let nombreContacto = prompt("Introduzca el nombre del nuevo contacto");
      let telefonoContacto = prompt("Introduzca el telefono del nuevo contacto");
      let correoContacto = prompt("Introduzca el correo del nuevo contacto");

      if (nombreContacto == null || telefonoContacto == null || correoContacto == null) {
        do {
          prompt("No has introducido todos los campos requeridos, por favor vuelvelos a introducir");

          nombreContacto = prompt("Introduzca el nombre del nuevo contacto");
          telefonoContacto = prompt("Introduzca el telefono del nuevo contacto");
          correoContacto = prompt("Introduzca el correo del nuevo contacto");

        } while (nombreContacto == null || telefonoContacto == null || correoContacto == null);

        array.push({
          nombre: nombreContacto,
          telefono: telefonoContacto,
          correo: correoContacto
        });
      }
    }
  }
}
```

Se añade un mensaje que muestre al usuario las nuevas opciones disponibles, se le da a elegir dicha opción, y mediante un switch se realiza la opción seleccionada.

Como se pide en el enunciado, si el usuario no ha rellenado alguno de los campos requeridos para añadir un nuevo contacto, en caso de haber seleccionado dicha opción, el contacto no se añade hasta que los datos sean introducidos.

## Ejercicio 5

5.- Encierra todo el programa en un bucle para que al usuario se le pregunte repetidamente qué quiere hacer. El usuario puede optar por:

- Mostrar el primer contacto (primero)
- Mostrar el último contacto (último)
- Mostrar todos los contactos (todos)
- Añadir un nuevo contacto (nuevo)
- Salir del programa (salir)

Después de ejecutar la acción seleccionada, el programa le dará la oportunidad de elegir nuevamente. El programa debe finalizar las acciones solo después de que el usuario dé un comando específico (salir).

```
81         telefono: telefonoContacto,  
82         correo: correoContacto  
83     })  
84 }  
85 break;  
86  
87 case '5':  
88     alert("Has seleccionado salir del programa");  
89     break;  
90  
91 default:  
92     alert("Opción no válida, seleccione de nuevo");  
93     break;  
94 }  
95 } while (opcionUsuario != '5');  
96  
97 alert("Programa finalizado");
```

En este caso, he añadido un bucle do-while para que el programa se ejecute hasta que se seleccione la opción de salir, que en mi caso es la número 5.

También se han añadido las nuevas opciones indicadas en el enunciado del ejercicio.

```

switch (opcionUsuario) {
  case '1':
    console.log(listaContactos[0]); //Mostramos la primera posición del array
    break;

  case '2':
    let ultimaPosicionArray = listaContactos.length-1;
    console.log(listaContactos[ultimaPosicionArray]); //Mostramos la última posición del array
    break;

  case '3':
    console.log(listaContactos);
    break;

  case '4':
    let nombreContacto = prompt("Introduzca el nombre del nuevo contacto");
    let telefonoContacto = prompt("Introduzca el telefono del nuevo contacto");
    let correoContacto = prompt("Introduzca el correo del nuevo contacto");

    if (nombreContacto == null || telefonoContacto == null || correoContacto == null) {
      do {
        alert("No has introducido todos los campos requeridos, por favor vuelvelos a introducir");

        nombreContacto = prompt("Introduzca el nombre del nuevo contacto");
        telefonoContacto = prompt("Introduzca el telefono del nuevo contacto");
        correoContacto = prompt("Introduzca el correo del nuevo contacto");

      } while (nombreContacto == null || telefonoContacto == null || correoContacto == null);

      listaContactos.push({
        nombre: nombreContacto,
        telefono: telefonoContacto,
        correo: correoContacto
      });
    } else {
      listaContactos.push({
        nombre: nombreContacto,
        telefono: telefonoContacto,
        correo: correoContacto
      });
    }
  }
}

```

## Ejercicio 6

6.- Trata de organizar el código de tu programa (¿quitando switch?) usando funciones. Define y llama a tres funciones en los lugares apropiados:

- showContact: la función debe tomar dos argumentos; el primero es la lista de contactos y el segundo es el número de índice del contacto a mostrar; dentro de la función, verifica si se pasan los argumentos correctos, es decir, si los contactos son un array (utiliza la construcción instanceof Array para esto).
- showAllContacts: la función debe tomar un argumento, la lista de contactos; dentro de la función, verifica si el argumento dado es un array.
- addNewContact: la función debe tomar cuatro argumentos, una lista de contactos y los datos del nuevo contacto, es decir: nombre, teléfono y número; antes de agregar un nuevo contacto, verifica si el argumento pasado es un array y si los datos del nuevo contacto tienen algún valor.

```

function showContact(array, indice) {
    if ((array instanceof Array) == true && indice != null) {
        console.log(array[indice]);
    }
}

function showAllContacts(array) {
    if ((array instanceof Array) == true){
        console.log(array);
    }
}

function addNewContact(array, nombreNuevo, telefonoNuevo, correoNuevo){
    if((array instanceof Array) == true && (nombreNuevo != null && nombreNuevo != " ") &&
    (telefonoNuevo != null && telefonoNuevo != " ") && (correoNuevo != null && correoNuevo != " ")){

        listaContactos.push({
            nombre: nombreNuevo,
            telefono: telefonoNuevo,
            correo: correoNuevo
        })

        console.log(listaContactos);
    } else{
        alert("Alguno de los datos no es correcto, no se introduce el nuevo contacto");
    }
}

```

En este caso sustituimos el switch que hemos utilizado en los ejercicios anteriores por las funciones indicadas en el enunciado del ejercicio. Utilizamos el instanceof Array para validar que el parámetro recibido en la función es un Array para poder ejecutar.

```

alert("Estas son las opciones disponibles:\n" + listaOpciones);
opcionUsuario = prompt ("¿Que opción eliges?");

if (opcionUsuario == '1') {
    posicionArray = prompt("¿Que posición de contacto quieres visualizar?");
    showContact(listaContactos, posicionArray);
} else if (opcionUsuario == '2'){
    showAllContacts(listaContactos);
} else if (opcionUsuario == '3'){
    let nombreContacto = prompt ("Introduzca el nombre del nuevo contacto");
    let telefonoContacto = prompt("Introduzca el telefono del nuevo contacto");
    let correoContacto = prompt("Introduzca el correo del nuevo contacto");

    addNewContact(listaContactos, nombreContacto, telefonoContacto, correoContacto);
}

```



El usuario seleccionará una de las 3 opciones disponibles y se llamará a la función correspondiente con la acción seleccionada.

## Ejercicio 7

7.- Da al usuario la opción de seleccionar una acción sort de la lista. Cuando se selecciona esta opción, el usuario debería poder especificar si desea ordenar los contactos por nombre, teléfono o correo electrónico.

```
function ordenarContactos(listaContactos , opcionOrdenar)
{
  if(opcionOrdenar == 1)
  {
    listaContactos.sort( function(a, b) {

      if (a.nombre > b.nombre) {
        return 1;
      }
      if (a.nombre < b.nombre) {
        return -1;
      }
      return 0;
    })

    for (const iterator of listaContactos) {
      console.log(iterator);
    }
  }

  if(opcionOrdenar == 2)
```

En este caso he creado una función llamada ordenarContactos para realizar la ordenación en base al campo seleccionado por el usuario. Como en los ejercicios anteriores, se le presentan al usuario las opciones disponibles, y en base a la opción seleccionada, se ordena el array.

## Conclusiones

Lo que he aprendido durante la realización de esta práctica es, en primer lugar, la utilidad de los arrays para almacenar datos con los que poder trabajar en JavaScript, como en este caso, una lista de contactos.

También hemos aprendido como manejar esa estructura de datos, pudiendo añadir nuevos datos, mostrar la posición que queramos de la misma, ordenar en base a cualquiera de los parámetros del array. También a ejecutar una estructura repetitiva en base a los parámetros que queramos hasta el momento que queramos.

## Acceso github

**[github.com/danieltorredlope/Primeras-pruebas-entorno-cliente](https://github.com/danieltorredlope/Primeras-pruebas-entorno-cliente)**