

## **ACTIVIDAD DE EVALUACIÓN (AEV3)**

### **Título**

---

#### **Control – Programación (Arrays)**

Video demo: [https://youtu.be/r\\_SBUhwF1\\_c](https://youtu.be/r_SBUhwF1_c)

### **Objetivos**

---

- Dominar de forma habitual los elementos básicos de un lenguaje de programación (variables, constantes, tipos de datos, operadores (aritméticos, lógicos y relacionales), sus precedencias, estructuras y flujos de controles (condicionales y repetitivas).
- Dominar el uso y utilización habitual de funciones.
- Dominar el uso y utilización de arrays unidimensionales y varios de sus métodos disponibles.

### **Temporalización**

---

**Previsto:** Última semana de la primera evaluación.

### **Proceso de desarrollo**

---

Crea un directorio de proyecto donde puedas almacenar el proyecto que se propone a continuación. **Debes realizar la entrega del ejercicio como resultado de la tarea.** Debes llamarlo **listaespera.js**

---

### **PROGRAMA PARA GESTIONAR UNA LISTA DE ESPERA DE MESAS EN UN RESTAURANTE**

Desarrolla un programa que te permita gestionar la lista de espera de clientes para disponer de una mesa en el restaurante Florida's Hollywood. Dicha lista deberá almacenar únicamente nombres del cliente que se apunten a la lista de espera para una mesa.

El menú de opciones de la aplicación deberá ser similar al siguiente:

```
LISTA ESPERA - Restaurante Florida's Hollywood
=====
1. Agregar nuevo cliente a la lista.
2. Siguiente cliente ocupa mesa.
3. Borrar cliente impaciente.
4. Ver turno de cliente.
5. Ver estado de la lista de espera.
6. Guardar la lista de espera.
7. Recuperar la lista de espera.
8. Salir del programa.
Introduce opcion (1..8): █
```

Todas las opciones del menú principal deben aparecer en la ejecución del programa. En caso que no se programe alguna de las funcionalidades que indican las opciones de menú, al seleccionarse dicha opción, la aplicación deberá avisar con el mensaje **“FUNCIONALIDAD NO PROGRAMADA”**.

Las funcionalidades proporcionadas por el menú de opciones serán las siguientes:

**1) Agregar nuevo cliente a la lista de espera.**

La aplicación pedirá el nombre de un cliente, que en caso de no estar ya dado de alta en la lista será agregado a la misma. En caso de haber un cliente con dicho nombre en la lista, la aplicación deberá avisar y no se dará de alta el cliente. La aplicación debe considerar el mismo un nombre tanto en mayúsculas como en minúsculas.

**2) Siguiente cliente ocupa mesa.**

Se selecciona cuando hay una mesa disponible y el siguiente cliente de la lista debe ocuparla. Esto provocará que dicho cliente desaparezca de la lista. El programa deberá avisar del nombre del cliente que debe ocupar la mesa.

**3) Borrar cliente impaciente.**

Un cliente puede decidir no esperar a su turno y que sea eliminado de la lista. Para ello deberá indicarse el nombre del cliente y en caso de existir en la lista será eliminado. En caso de no existir deberá avisar que el cliente no está en la lista.

**4) Ver turno de cliente.**

Un cliente puede consultar su turno en la lista. Para ello suministraremos el nombre del cliente y nos indicará cuál es su turno en formato numérico. Por ejemplo con un mensaje: *“Salva está en la posición 3”*, es decir, le quedarían dos clientes todavía por delante.

**5) Ver estado de la lista de espera.**

Esta parte es fundamental para poder comprobar que el resto de funcionalidades son correctas. Al seleccionar esta opción la aplicación deberá visualizar el estado de la lista. Por ejemplo, de la siguiente forma:

```
Visualizo lista  
[ 'ANDRES', 'PEDRO', 'ANA', 'JOSE', 'LUIS', 'MARIBEL', 'ANDREA' ]
```

**6) Guardar la lista de espera.**

Esta opción guardará los datos de la lista en un archivo denominado “listado.txt”. Esto permitirá recuperar la lista de espera (con la opción 7 de menú principal) en caso que salgamos de la ejecución del programa. El archivo generado deberá estar en la misma carpeta que el fichero javascript (listaespera.js) que contiene el código de la aplicación.

**7) Recuperar la lista de espera.**

Esta opción permitirá recuperar una lista de espera de un archivo denominado “listado.txt”. Dicha lista deberá haber sido guardada previamente (utilizando la opción 7 de menú principal).

**8) Salir del programa.**

Esta opción permitirá finalizar la ejecución de la aplicación. Mientras no se seleccione, la aplicación ejecutará continuamente las opciones 1 a 7 (según las seleccione el usuario) y tras ello volverá a mostrar el menú de opciones.

Piensa la mejor lógica de programación y crea las funciones que creas necesarias y te ayuden a conseguir que tu código sea lo más limpio y estructurado. Mejora tus comentarios en el código creando también comentarios de documentación en tus funciones programadas utilizando JSDoc.

Para lograr las opciones de menú 6 y 7 deberás investigar el uso del módulo/librería “fs” disponible en el entorno node.js. Dicho módulo contiene métodos y funcionalidades para manipular archivos y tratar con el sistema de archivos (p.e. fs.openSync, fs.writeSync, fs.writeFileSync, fs.readFileSync, etc...).

Como documentación adicional, en la actividad de entrega en plataforma dispones de una presentación introductoria a la entrada/salida con ficheros que te ayudará a programar las funcionalidades 6 y 7 del menú de la aplicación. Evidentemente, puedes investigar otras opciones para lograr las funcionalidades 6 y 7.

## Evaluación

Esta actividad práctica es una actividad de evaluación y de entrega obligatoria. No se aceptarán entregas de prácticas fuera del plazo estrictamente establecido. **Criterios de evaluación establecidos:**

CRITERIOS (funcionamiento correcto)	Valoración
<p><b>1) Creación del sistema de menús pedido. (1,5 pts)</b></p> <p><b>2) Agregar nuevo cliente a la lista de espera. (1 pto)</b></p> <ul style="list-style-type: none"> <li>Alta de clientes y confirmación.</li> <li>Comprobación y aviso de cliente ya existente.</li> </ul> <p><b>3) Siguiente cliente ocupa mesa. (1 pto)</b></p> <ul style="list-style-type: none"> <li>Ocupación de mesa por parte del siguiente cliente</li> <li>Visualización de nombre de cliente.</li> <li>Comprobación de existencia de cliente en la lista y aviso en caso de estar vacía.</li> </ul> <p><b>4) Borrar cliente impaciente. (1 pto)</b></p> <ul style="list-style-type: none"> <li>Borrado de cliente indicado por nombre</li> <li>Comprobación y aviso de borrado con éxito (o no).</li> </ul> <p><b>5) Ver turno de cliente. (1 pto)</b></p> <ul style="list-style-type: none"> <li>Visualización de turno/posición de cliente indicando su nombre.</li> <li>Comprobación y aviso de cliente no existente en la lista o lista vacía.</li> </ul> <p><b>6) Ver estado de la lista de espera. (1 pto)</b></p> <ul style="list-style-type: none"> <li>Visualización del estado de la lista de espera.</li> </ul> <p><b>7) Guardar la lista de espera. (0.75 pto)</b></p> <ul style="list-style-type: none"> <li>Guardado de los datos de la lista en un archivo denominado "listado.txt" (en el mismo directorio de la aplicación).</li> </ul> <p><b>8) Recuperar la lista de espera. (0.75 pto)</b></p> <ul style="list-style-type: none"> <li>Recuperación de la lista de espera de un archivo denominado "listado.txt".</li> <li>Posterior visualización de la lista cargada mediante la funcionalidad "ver estado de la lista de espera".</li> </ul> <p><b>9) Valoración general del ejercicio. (2 pts)</b></p> <ul style="list-style-type: none"> <li>Documentación de las funciones utilizadas con JSDoc.</li> <li>Formato adecuado en la escritura del código (orden, sangrado, indentaciones correctas, código limpio, etc..).</li> <li>Uso correcto de estructuras de repetición y condicionales.</li> <li>Utilización correcta y habitual de funciones para cada funcionalidad de la aplicación y aquellas secciones de código que se repitan habitualmente.</li> <li>Corrección general de la aplicación en su conjunto.</li> </ul>	