

 C mo puedo guardar la lista de reserva de mesas del restaurante Florida's Hollywood y recuperarla en la pr xima ejecuci n de la aplicaci n?



Unas pistas...

```
LISTA ESPERA - Restaurante Florida's Hollywood
=====
1. Agregar nuevo cliente a la lista.
2. Sigui nte 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): █
```

Hasta ahora...

- Iniciamos nuestros programas
- Asignamos valores a nuestras variables
- Ejecutamos instrucciones
- El programa termina
- El valor de las variables se pierde
- Y no necesitamos nada m s...



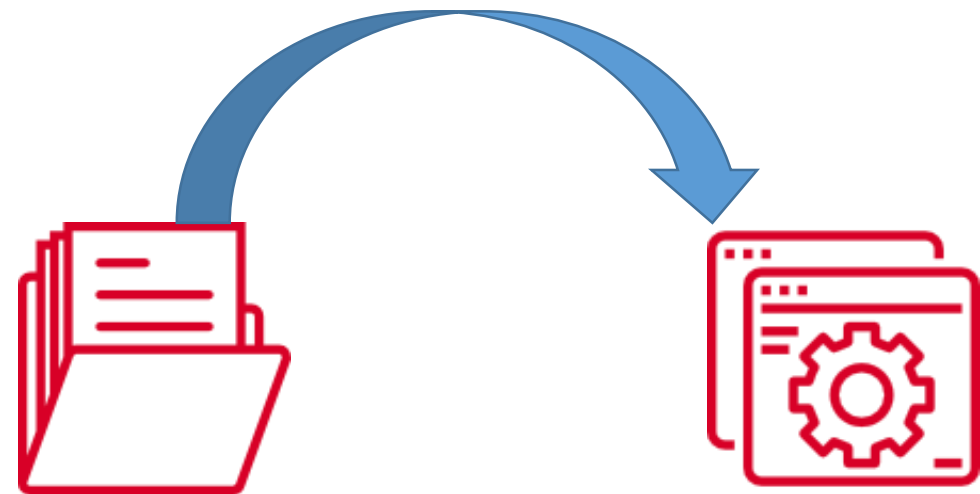
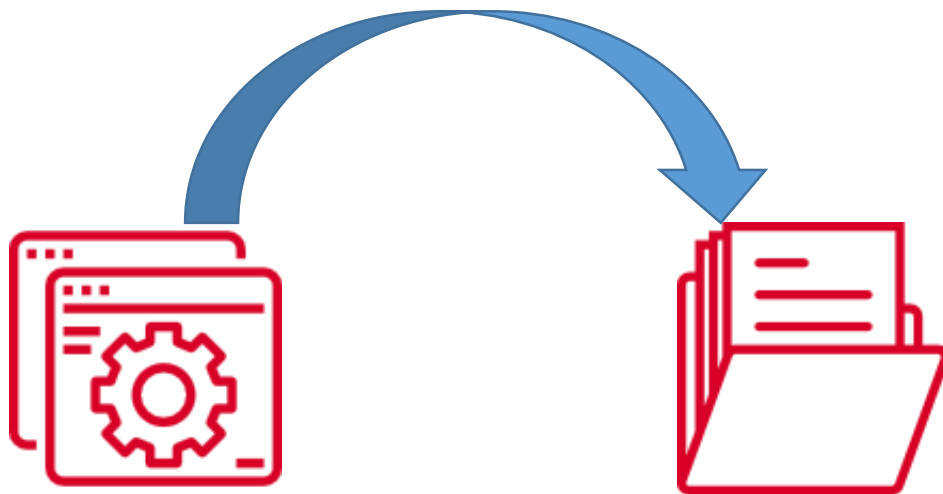
Muchas veces...

- Queremos tener preferencias de usuario que se mantengan de una ejecuci n a otra.
- Queremos guardar una partida y volver a cargarla tras lanzar el juego de nuevo.
- Queremos ir construyendo una colecci n de datos a lo largo de las ejecuciones del programa y que dichos datos se mantengan y crezcan entre dichas ejecuciones.
- Necesitamos disponer de informaci n que est  en ficheros externos a nuestro programa.

...Necesitaremos ser capaces de...

PERSISTIR datos/informaci n

RECUPERAR
datos/informaci n persistida



En este documento dispones de informaci n sobre ...

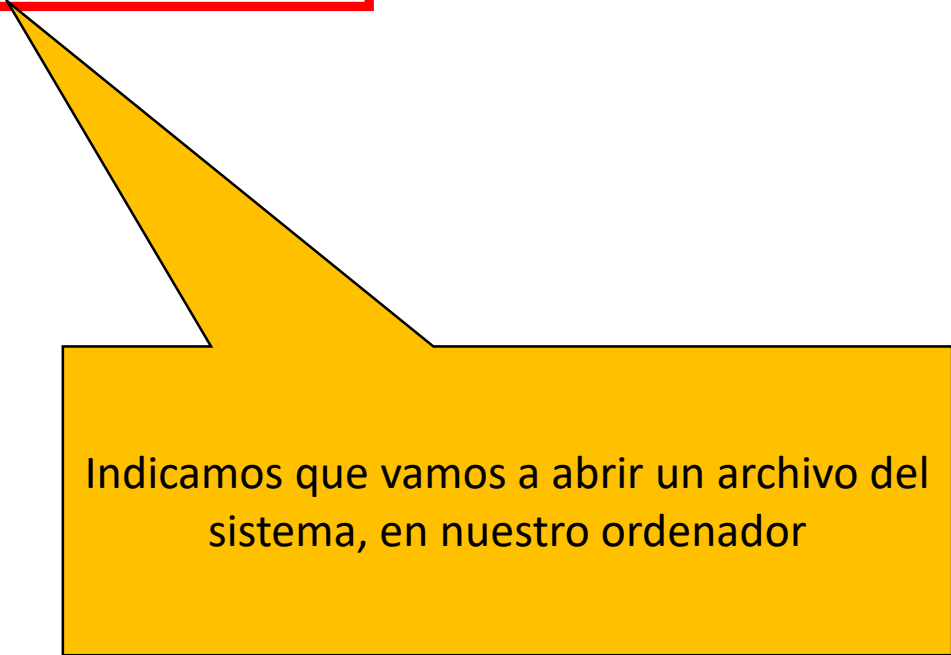
- La persistencia m s sencilla posible:
 - Lectura de un fichero de texto
 - Escritura en un fichero de texto
- Hay formas m s avanzadas, pero ya las ir is trabajando a lo largo del ciclo y de esta asignatura.
- Por ahora, esto nos sirve para intentar resolver algunas de las funcionalidades propuestas en la AEV3 de PROGRAMACI N.

Leer/Escribir en un fichero en JS

- Antes de nada, al principio requerimos importar c digo de un m dulo est ndar en NodeJS
 - `const fs = require('fs');`
- El m dulo fs tiene definido toda una serie de funciones que nos permiten abrir ficheros, escribir sobre ellos, y leer de los mismos.
 - Tiene tanto versiones s ncronas como as ncronas
 - **En la AEV3 necesit is  nicamente emplear funciones s ncronas.**

Abrir un fichero de forma s ncrona

```
let file = fs.openSync('archivo.txt', 'r');
```



Indicamos que vamos a abrir un archivo del sistema, en nuestro ordenador

Abrir un fichero de forma s ncrona

```
let file = fs.openSync('archivo.txt', 'r');
```

Ruta del archivo relativa desde donde ejecutamos el c digo, o completa. Puede ser una cadena de texto o una variable que contenga la cadena de texto

Abrir un fichero de forma s ncrona

```
let file = fs.openSync('archivo.txt', '\r');
```

Modo de apertura. Algunos modos comunes:

r → Lectura de texto

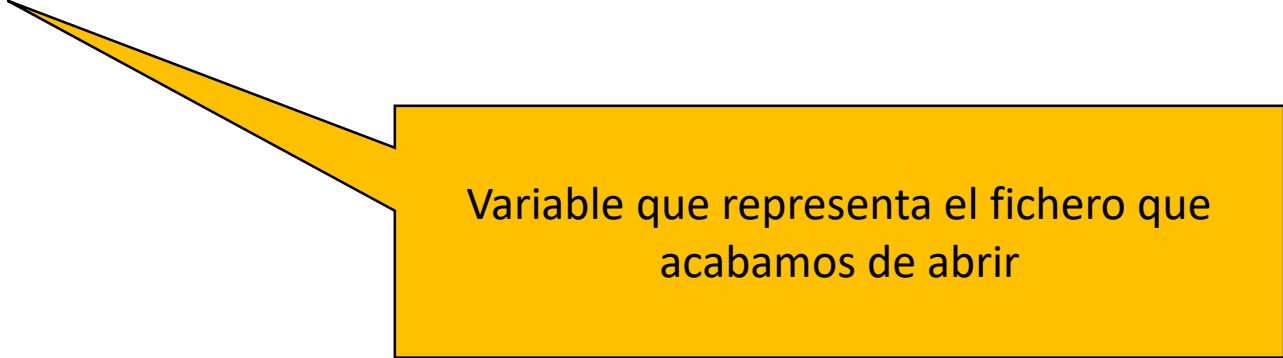
w → Escritura de texto (Crea archivo si no existe. Si existe lo trunca)

a → Escritura de texto (a adiendo al final)

M s modos: <https://bit.ly/2OaEMMV>

Abrir un fichero de forma s ncrona

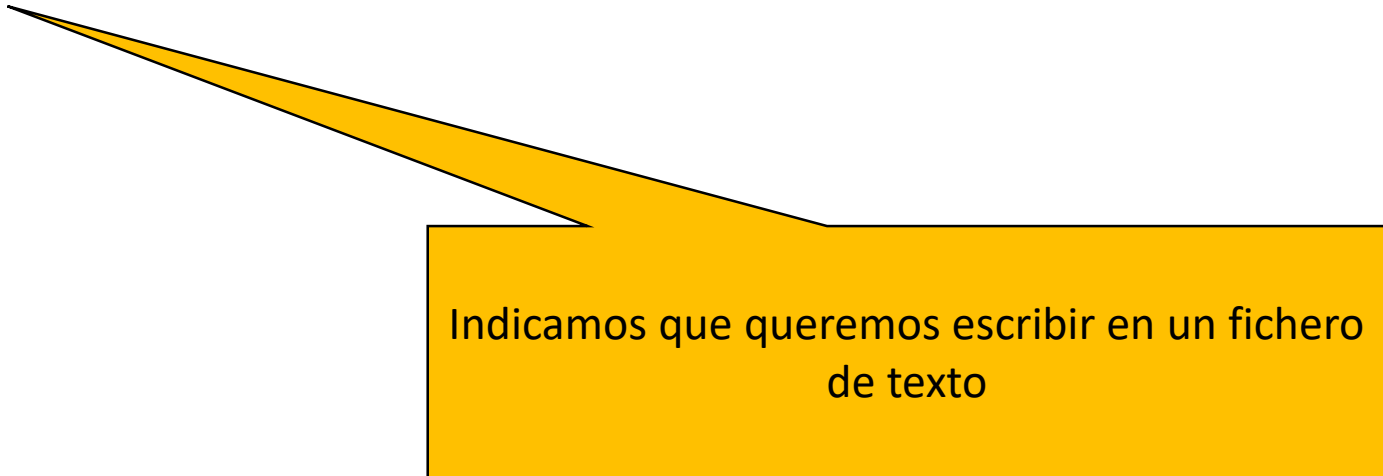
```
let file = fs.openSync('archivo.txt', 'r');
```



Variable que representa el fichero que acabamos de abrir

Escribir en un fichero una cadena

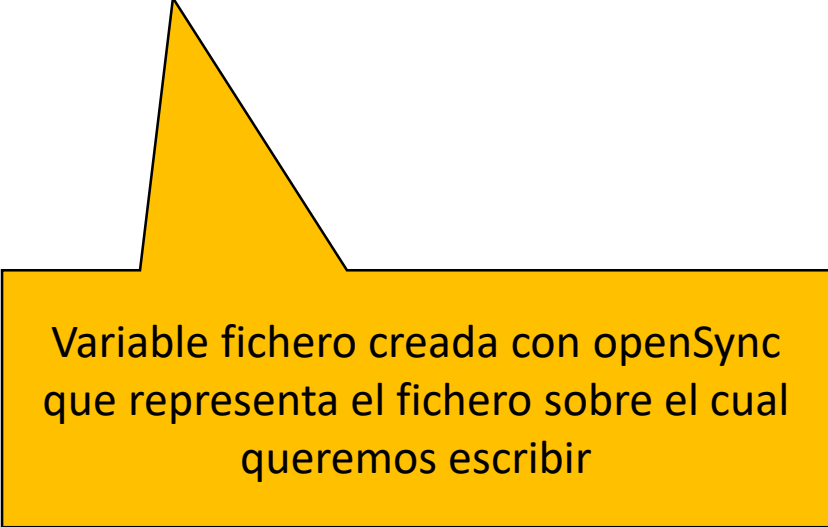
```
let file = fs.openSync('archivo.txt', 'w');  
fs.writeFileSync(file, 'Texto a escribir', undefined, 'utf-8');
```



Indicamos que queremos escribir en un fichero de texto

Escribir en un fichero una cadena

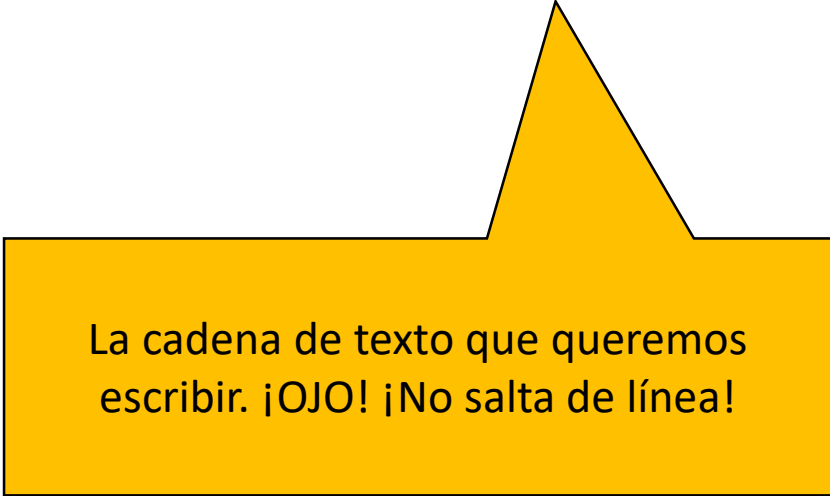
```
let file = fs.openSync('archivo.txt', 'w');  
fs.writeFileSync(file, 'Texto a escribir', undefined, 'utf-8');
```



Variable fichero creada con openSync
que representa el fichero sobre el cual
queremos escribir

Escribir en un fichero una cadena

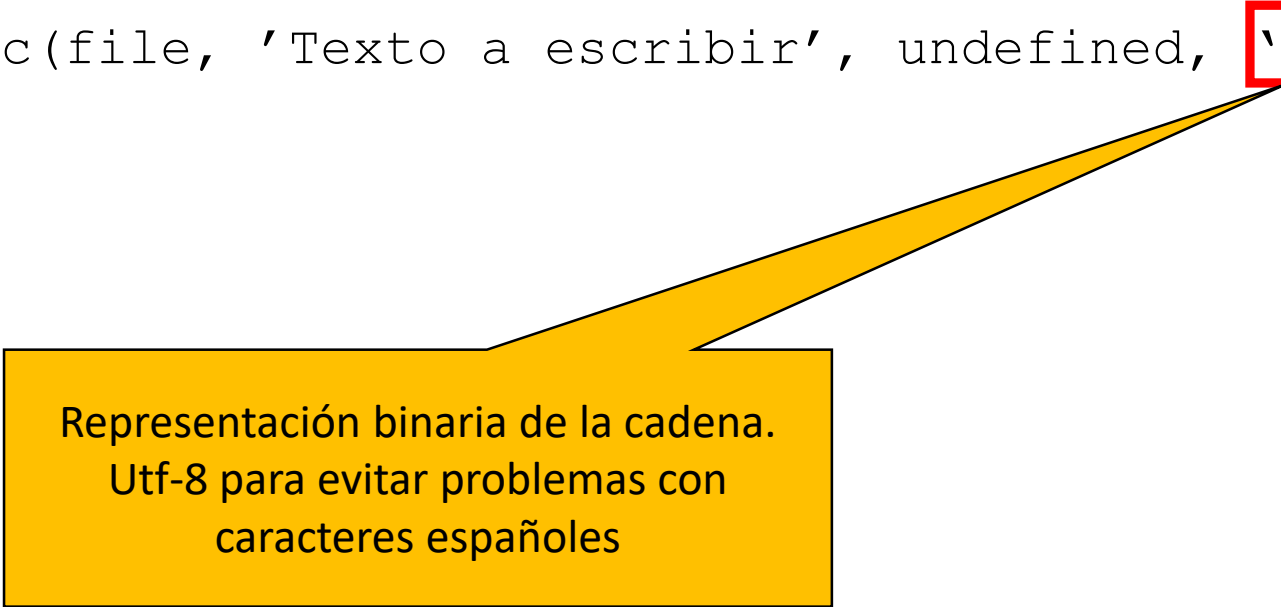
```
let file = fs.openSync('archivo.txt', 'w');  
fs.writeFileSync(file, 'Texto a escribir', undefined, 'utf-8');
```



La cadena de texto que queremos escribir. ¡OJO! ¡No salta de línea!

Escribir en un fichero una cadena

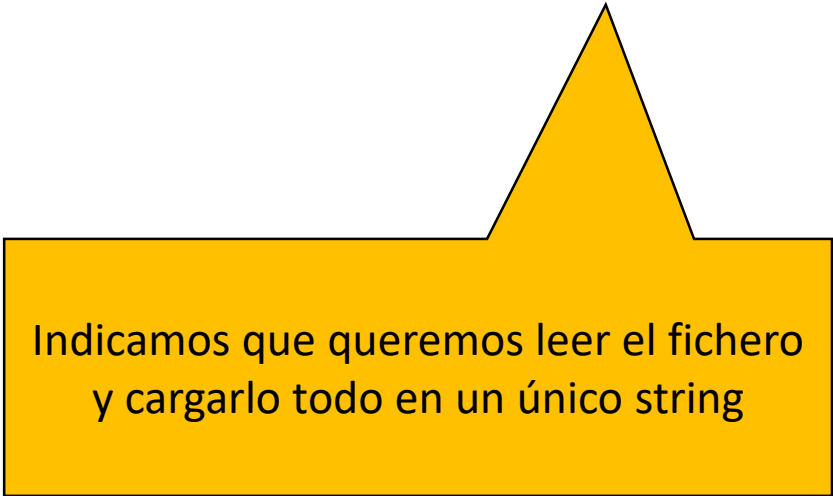
```
let file = fs.openSync('archivo.txt', 'w');  
fs.writeFileSync(file, 'Texto a escribir', undefined, 'utf-8');
```



Representaci n binaria de la cadena.
Utf-8 para evitar problemas con
caracteres espa oles

Cargar un fichero en un string


```
let file = fs.openSync('archivo.txt', 'r');  
let lines = fs.readFileSync(file, 'utf8');
```



Indicamos que queremos leer el fichero
y cargarlo todo en un  nico string

Cargar un fichero en un string

```
let file = fs.openSync('archivo.txt', 'r');  
let lines = fs.readFileSync(file, 'utf8');
```



Variable que representa un fichero
abierto en modo lectura

Cargar un fichero en un string

```
let file = fs.openSync('archivo.txt', 'r');  
let lines = fs.readFileSync(file, 'utf8');
```

Codificaci n del archivo. Suele ser utf8
en muchos sistemas modernos

Cargar un fichero en un string y leer l nea a l nea

```
let file = fs.openSync('archivo.txt', 'r');  
let lines = fs.readFileSync(file, 'utf8');  
for(let line of lines.split('\n')) {  
    //Procesar la l nea  
    ...  
}
```

Cerrar ficheros

- Una vez hemos terminado de escribir en un fichero hay que cerrarlo

`fs.closeSync(file);`

- ¿Por qu ?
- Evitamos uso de recursos innecesarios.
- Parte de las escrituras a veces solo se llevan a cabo al final, al cerrar el fichero.
- Hay un n mero l mite de ficheros que pueden abrirse en un programa.
- Algunos sistemas operativos bloquean los archivos para otros programas mientras estos no se cierren.

M s informaci n: <https://nodejs.org/api/fs.html>