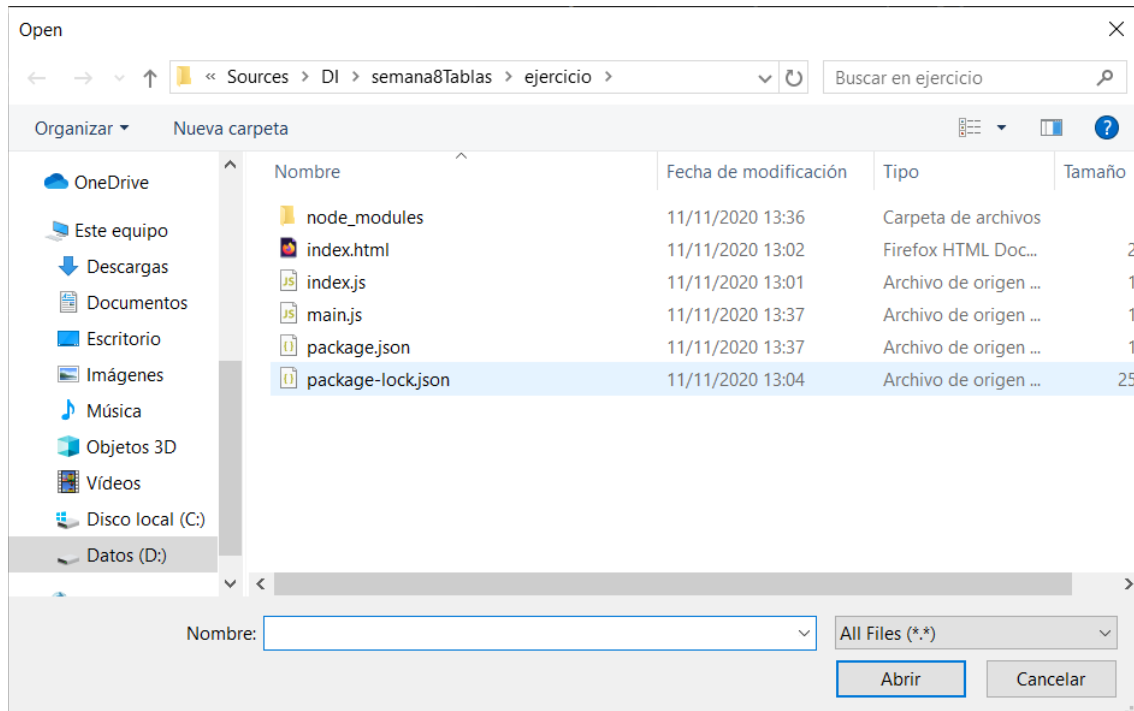


## 1. Diálogo para abrir ficheros.

Además de los diálogos de error y de información, es muy utilizado el diálogo de Abrir fichero, que le permite al usuario seleccionar ficheros.



Igual que los otros diálogos de Electron, este diálogo está incluido en el módulo dialog.

Para usarlo debemos realizar en el main.js y en index.js las opciones para utilizar remote y ya podremos llamar al diálogo de abrir archivo `showOpenDialogSync`, que forma parte de dialog:

```
fileNames = dialog.showOpenDialogSync();
```

En `fileNames` tendremos un array de strings con los paths de los archivos seleccionados por el usuario. Si el usuario cancela la acción, su valor será `undefined`.

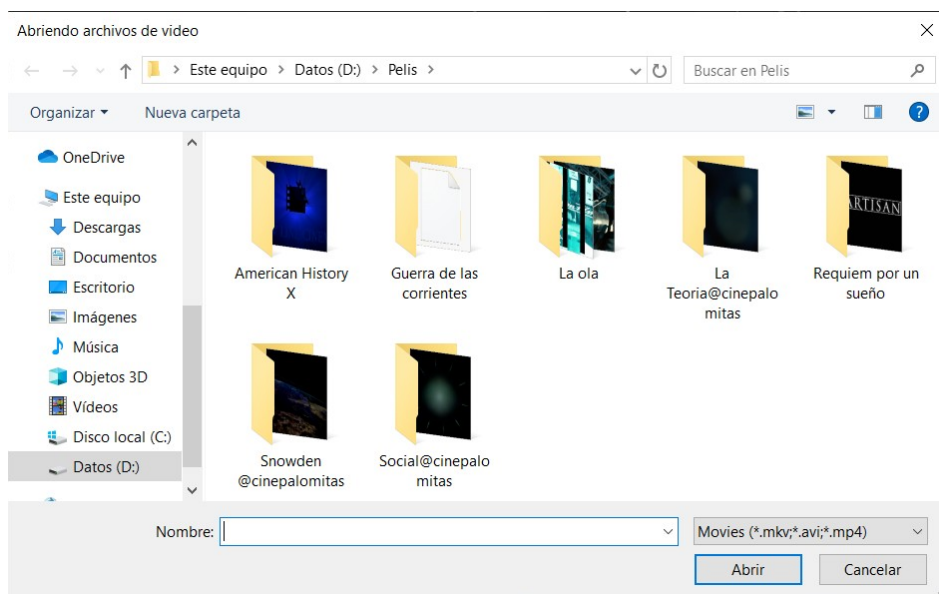
El diálogo se puede invocar sin argumentos y se abre en la misma carpeta del proyecto, o pasándole un objeto de

opciones. En este objeto se pueden indicar muchas opciones, entre otras el título de la ventana (title), el path por defecto (defaultPath) o los tipos de archivos (filters).

Por ejemplo, para abrir la carpeta D:\Pelis, aceptando solo extensiones de video y cambiando el título, usaríamos estas opciones:

```
fileNames = dialog.showOpenDialogSync({
  title: "Abriendo archivos de video",
  defaultPath: "D:\\Pelis",
  filters: [
    { name: 'Movies', extensions: ['mkv', 'avi', 'mp4'] },
    { name: 'All Files', extensions: ['*'] }
  ]
});
```

Y el resultado:



Más opciones del diálogo Open:

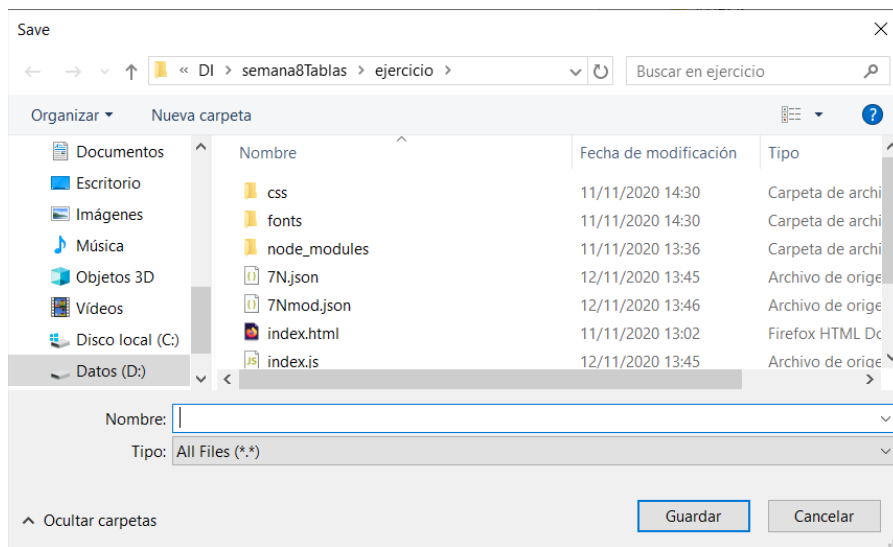
<https://www.electronjs.org/docs/api/dialog#dialogshowopendialogsyncbrowserwindow-options>

## 2. Diálogo para guardar ficheros.

También es muy utilizado el diálogo para guardar ficheros. Su uso es muy sencillo:

```
let fileSave = dialog.showSaveDialogSync();
```

Y tendremos;



En fileSave tendremos la ruta elegida por el usuario para guardar el fichero.

De la misma manera que antes, también podemos llamar a showSaveDialogSync con un objeto de opciones que funciona de forma muy similar al visto anteriormente. Más información:

<https://www.electronjs.org/docs/api/dialog#dialogshowsavedialogsyncbrowserwindow-options>

### 3. Hacer una celda editable.

El atributo global `contenteditable` es un atributo que indica si el elemento debe de ser editable por el usuario . Si es así, el explorador permitirá la edición. El atributo debe de tener alguno de los siguientes valores :

- true, que indica que el elemento debe de ser editable .

- false, que indica que el elemento no debe ser editable.

No cualquier elemento puede ser editado. Pero algunos si, por ejemplo, podemos hacer editable un label:

```
<label contenteditable="true">Example Label</label>
```

O también una celda de una tabla:

```
<td contenteditable="true">Example Celda</td>
```