

# Edición de archivos RTF

## Especificaciones Técnicas

Para la creación de esta función se utilizaron las siguientes tecnologías:

❖ Visual Studio Code:

Para la instalación de este editor de código es necesario ir a la página de [Visual Studio Code](#), empezaremos con la descarga de este mismo, después de esto lo podremos usar de forma correcta para la edición del programa.

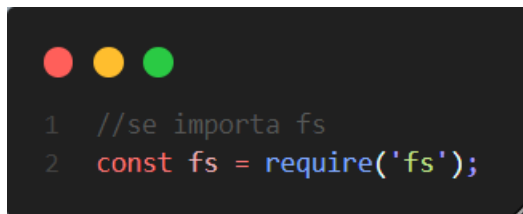
❖ NodeJS:

Para la instalación de este lenguaje, es necesario ir a la página de [NodeJS](#), de esta se descargara un paquete de instalación y podremos proseguir con los pasos necesarios para su uso.

## Carpeta principal

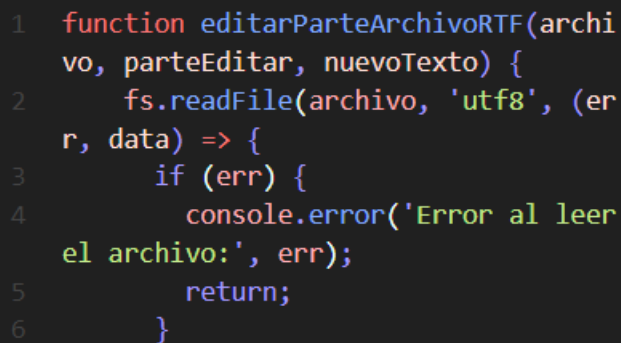
Dentro de esta carpeta meteremos lo que es nuestro [archivo RTF](#), el cual usaremos para poder ver como se va editando mediante el texto que nosotros querramos.

Funcion.js

A screenshot of a code editor window with a dark background and three colored window control buttons (red, yellow, green) at the top left. The code is as follows:

```
1 //se importa fs
2 const fs = require('fs');
```

Se importa fs, este ya viene incluido cuando se descarga Node.js

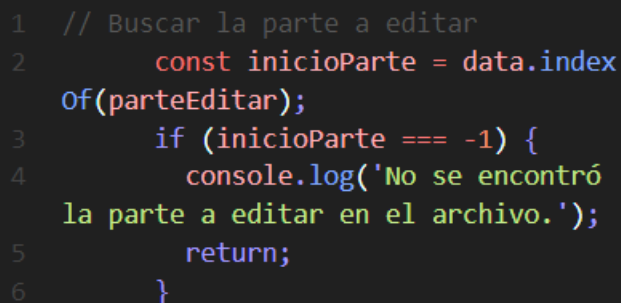


```

1  function editarParteArchivoRTF(archi
2      vo, parteEditar, nuevoTexto) {
3      fs.readFile(archivo, 'utf8', (er
4          r, data) => {
5          if (err) {
6              console.error('Error al leer
              el archivo:', err);
              return;
          }
      }
  
```

Se crea la función `editarParteArchivoRTF`, en este le ponemos las variables llamadas: “archivo”, “parteEditar” y “nuevotexto”.

Dentro de la función mandamos a llamar a `fs.readFile`, este nos ayudara a leer la variable “archivo”, seguido tenemos que ponerle la codificación en la que se encuentra nuestro archivo en este caso es “utf8”, después buscamos con la data si se encuentra un error. Después usamos un ciclo `if` por si se encuentra algún error este lanzara por consola el error.



```

1  // Buscar la parte a editar
2      const inicioParte = data.index
3      Of(parteEditar);
4      if (inicioParte === -1) {
5          console.log('No se encontró
6          la parte a editar en el archivo.');
```

Se crea una constante llamada `inicioParte`, en el cual buscaremos por parte del `data.indexOf` nuestra variable “parteEditar”. Creamos un ciclo `if` en la condición ponemos que si “inicioParte” nose encuentra ( o da -1) lanzamos en la consola que esta parte que buscamos efectivamente no se encuentra.

```
1 // Calcular el final de la parte a e
  ditar
2     const finParte = inicioParte +
    parteEditar.length;
```

Creamos una constante llamada finParte, en esta calcularemos la longitud que tendrá finalmente la parte que se editara.

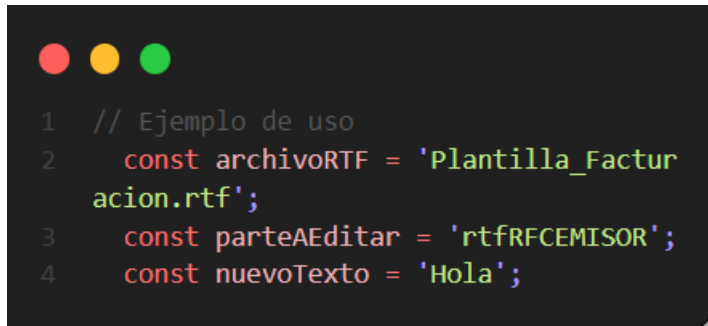
```
1 // Reemplazar la parte con el nuevo
  texto
2     const nuevoContenido = data.sl
    ice(0, inicioParte) + nuevoTexto + d
    ata.slice(finParte);
```

Creamos un constante llamado nuevoContenido, dentro de este mandamos a llamar data.slice a partir de la posición 0 de la constante "inicioParte" después le añadimos nuestra variable "nuevoTexto", llamamos otro data.slice con la condición finParte.

```
1 fs.writeFile(archivo, nuevoContenido,
    'utf8', (err) => {
2     if (err) {
3         console.error('Error al escr
        ibir en el archivo:', err);
4         return;
5     }
6     console.log('Archivo actualiza
        do correctamente.');
```

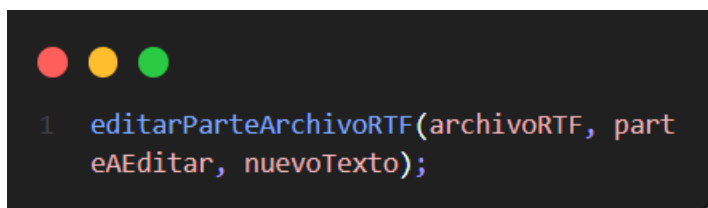
Mandamos a llamar fs.writeFile, con condiciones a nuestras variables "archivo", "nuevoContenido", la codificación de el archivo, y una data de error por si llega a haber un error. Generamos un ciclo if si llega a haber un error este lanzara por consola que no se pudo escribir dentro del archivo. Si el error no se encuentra, entonces se lanzara

por consola que el archivo fue correctamente actualizado. Cuando lance que el archivo fue correctamente editado puedes abrir dentro de tu carpeta el mismo archivo rtf y podrás darte cuenta que la variable “parteEditar” que elegiste fue correctamente reemplazado por la variable “nuevoTexto”.



```
1 // Ejemplo de uso
2 const archivoRTF = 'Plantilla_Factur
  acion.rtf';
3 const parteAEditar = 'rtfRFCEMISOR';
4 const nuevoTexto = 'Hola';
```

Estos son los ejemplos de uso que estaremos usando para la edición de nuestro archivo RTF.



```
1 editarParteArchivoRTF(archivoRTF, part
  eAEditar, nuevoTexto);
```

Mandamos a llamar la función para que nuestras constantes tomen su valor dentro de las variables de la función para que así nuestro Archivo pueda ser efectivamente modificado.

## Objetivos sin cumplir

- ❖ Añadir a serverless.
- ❖ Conectar con la otra lambda para que la variable “nuevoTexto” sea basado por una consulta de la base de datos mariadb.