

Documentación de funciones y métodos

Desarrollador por: Kevin Urrea

28/11/2023

Datos Técnicos

Este software fue creado por motivos de aprendizaje, usando un caso real sobre la necesidad de generar reportes de una forma más rápida y agile usando otras herramientas diferentes a Excel.

Fue desarrollado con Vanilla JavaScript, con el apoyo de la librería "[read-excel-file](#)", también se usó "[Bootstrap](#)", para estilar algunas áreas de del software.

Limitante:

1. Es un software exclusivo para generar informes con un formato de archivo exclusivo, no permite la lectura de otros archivos Excel (XSLX) que tengan alguna estructura distinta, ya que esto fue hecho a la medida basándonos en el informe que genera la Base de Datos de la entidad.
2. No acepta otros archivos distintos a Excel con la extensión XSLX.
3. Debe tener habilitado el uso de JavaScript en su navegador, si se encuentra desactivado, se recomienda activarlo o investigar que puede estar ocurriendo.
4. Debe estar conectado a internet para que todo funcione correctamente.

El software esta en su etapa Beta 1.5v. por lo tanto se encuentra limitado a generar pocos informes o mostrar datos

Funciones

- **excelInput.addListener('change', async function ()):** ejecuta un evento de escucha para detectar el cambio en el input de archivos puesto en el HTML, el cual genera una función asíncrona que espera, la respuesta correcta para ejecutar los demás comandos. En esta función podemos encontrar un control de errores con un try / catch, el cual valida si el archivo que se ingreso tiene la extensión permitida (XSLX) para ser procesado el archivo sin errores, en caso que el formato del archivo no sea el correcto generara un mensaje de error al usuario. En el caso que todo este correcto este almacena la información leída en una clase llamada Excel, para mejor manejo de la información, por consiguiente, ejecuta varias funciones de las cuales se hablan más adelante.
- **function buscarIndex(busqueda):** con este método logramos obtener el índice posicional de la columna que deseamos buscar o consultar la información, enviándole por parámetro el nombre exactamente igual a como aparece en el documento Excel introducido, por ejemplo, si buscamos la columna nombres tendríamos que enviarle esto por parámetro a la función, 'NOMBRES'.
- **function validarLocalStorage():** Esta función hace una validación sobre el LocalStorage del navegador, para consultar si el objeto que se creo que cada asesor existe, en caso que no exista se crea con la función crearObjetosEmpleados().
- **function esArrayActualizado(arraySinRepetidos, objetosEmpleadosGuardados):** con esta función nos apoyamos para consultar si la información del objeto que está en el LocalStorage esta actualizada con el archivo que se acabó de subir, en caso que no lo este se vuelve a crear con una versión mas actualizada de los datos.
- **function convertirTiempoASegundos(tiempoString):** Nos permite recibir un array de distintos tiempos registrados en el servicio y convertirlos a un array donde cada tiempo pasa a segundos.
- **function tiempoPro(arrayTiempo):** Con esta función podemos conocer el promedio de tiempo que se puede tardar o durar un cliente basado en el tiempo recibido, ya sea para el promedio de espera o de atención.
- **function fechaReporte(excel):** esta función detecta la fecha del reporte, la cual se encuentra en la columna Fecha del archivo y la retorna.
- **function mostrarData(id):** con esta función se logra hacer una búsqueda en el array de objetos almacenado en el LocalStorage, basándonos en el id que es el índice posicional, luego de encontrar el objeto se empieza a pintar la información en el HTML de su respectivo asesor.
- **function mostrarAdicionales():** esta función muestra la información adicional independientemente de cada asesor, es una información general del informe, datos como la cantidad de servicios que se hicieron en ese día, actualización de RUT, Inscripción y demás datos.
- **function userSelected(id):** Con esta función se logra detectar cual fue el asesor por el cual se requiere ver la información, si el usuario elige al asesor 3, esta función nos ayuda a validar sobre que asesor se buscara la información.
- **function crearObjetosEmpleados(arraySinRepetidos):** Con esta función se crea el array de objetos con una estructura especifica para luego ser almacenada en el LocalStorage. Esta recibe por parámetro el arraySinRepetidos el cual almacena todos los nombres de los asesores previamente filtrados para evitar

duplicados, este array nos ayuda a crear un objeto por cada nombre que encuentre y usa como identificador el nombre del mismo.

- **function extraerInfo():** Esta función consulta todas las filas del documento y filtra por cada asesor para almacenar correctamente la información respectivamente por cada asesor, al finalizar el recorrido por todas las filas e ir almacenando la información en respectivas variables acumuladoras, este hace llamado a la función `actualizarInformacionEmpleado()`, para actualizar la información el objeto de asesor almacenado previamente.
- **function actualizarInformacionEmpleado(nombre, turnos, cantAtendido, pNatural, pJuridica, cantAban, cantExi, cantFall, totalTiempo, tiempoAtend, tiempoEsp):** esta función nos permite recibir por parámetro información de la cual se desea actualizar, el primer parámetro actúa como identificador único, con el cual se filtra la búsqueda entre el array de objetos, una vez encontrado se empieza actualizar la información del respectivo asesor encontrado. En esta función se hace llamado a varias funciones de las cuales nos ayuda a encontrar el promedio de tiempo.
- **function BuscarPersonal(excel):** Con esta función actuamos para filtrar los nombres del personal en la institución y evitar duplicados en el `arraySinRepetidos`, recibe como parámetro la clase Excel del cual se obtiene de manera más fácil la información del archivo.

Clases

Clase Excel:

- **Función del Constructor (constructor):** Recibe el contenido del archivo Excel como parámetro y lo almacena en la propiedad **content**.
- **Función header():** Retorna el encabezado del archivo Excel (la primera fila).
- **Función rows():** Retorna una instancia de la clase **RowsCollection**, que representa la colección de filas del archivo Excel.
- **Función columns():** Retorna una instancia de la clase **ColumnsCollection**, que representa la colección de columnas del archivo Excel.

Clase ColumnsCollection:

- **Función del Constructor (constructor):** Recibe la colección de columnas como parámetro y la almacena en la propiedad **columns**.
- **Función firstCol():** Retorna la primera columna del archivo Excel.
- **Función getCol(index):** Retorna una columna específica del archivo Excel según el índice proporcionado.
- **Función getCols():** Retorna todas las columnas del archivo Excel.
- **Función countCols():** Retorna el conteo de columnas del archivo Excel.

Clase RowsCollection:

- **Función del Constructor (constructor):** Recibe la colección de filas como parámetro y la almacena en la propiedad **rows**.
- **Función firstRow():** Retorna la primera fila del archivo Excel.
- **Función getRow(index):** Retorna una fila específica del archivo Excel según el índice proporcionado.
- **Función getRows():** Retorna todas las filas del archivo Excel.
- **Función countRows():** Retorna el conteo de filas del archivo Excel.

Clase Row:

- **Función del Constructor (constructor):** Crea una instancia de la clase **Row** y recibe una fila del archivo Excel como parámetro.

Clase Col:

- **Función del Constructor (constructor):** Crea una instancia de la clase **Col** y recibe una columna del archivo Excel como parámetro.