

# Práctica 2

---



## DataShell

### Integrantes del equipo

Castro Bouquet Ildfonso

### Profesor

César Arturo Angeles Ruiz

### Materia

Taller de Desarrollo de Aplicaciones

**Entregable:** (29 de Octubre, 11:59 horas): Especificaciones por equipo, con el nombre de los integrantes y la especificación final, también con el nombre de los que participaron.

---

## Introducción:

Se pide realizar un programa que pueda leer un archivo CSV, su contenido es incógnito por lo que hay que realizar funciones generales donde se pueda abrir un archivo con la cantidad de datos que se requiera.

Para ello se utilizara un tensor definido como un arreglo de apuntadores de tipo char. Al momento de leer los valores, validamos los datos hasta que sea el final del archivo (EOF), contaremos con una variable la cantidad de columnas introducidas, para así poder trabajar dentro de las funciones con estos parámetros.

Se usarán strings para la lectura para evitar explotaciones de parte del usuario, y así maximizar el uso de funciones generales. Se utilizarán funciones como fgets y strtok para separar los datos obtenidos. En este momento usaremos otra variable que nos guarde la cantidad de renglones a usar.

Como último punto de la práctica le daremos formato a los datos, (guardandolos en strings separados por valores y así separar columnas y renglones).

Para finalizar y visualizar los valores, se creará un archivo con los datos separados, si funciona correctamente debería de ser idéntico junto al contenido del archivo inicial de lectura.

## Objetivos:

- Leer un archivo y almacenar datos en un string.
- Separar los datos del string con strtok.
- Darle un formato a los datos obtenidos.

## Análisis:

Al iniciar el programa se creará un menú donde preguntará por el archivo a leer, se validará si este existe o no, estará en un ciclo para seguir preguntando el nombre del archivo mientras no lo encuentre, también esta la opcion de ejemplo, donde se leerá un archivo predeterminado con ciertos valores, también contará con una opción de salida.

Después de validar y abrir el archivo se llamará una función de lectura, donde leerá el archivo almacenando los valores en un apuntador, para ello la función leerá cantidad de columnas y después leerá caracter por caracter para saber de que tamaño hacer un malloc y guardar la información en el pointer.

Para que funcione se debe leer el archivo varias veces, la primera para contar caracteres, y la segunda para leer el archivo y almacenar la información con fgets.

En este caso sabemos que el archivo solo tendrá valores de tipo float, es decir un número con ".", por lo que nos ahorra un paso de validación al momento de dividirlo en tokens, para ello llamaremos una función que tome el pointer de caracteres y lo divida en tokens, de ahí la información será guardada en otro array de tipo float para almacenar toda la información, para hacer la conversión convertiremos los caracteres a float con la función atof().

Por último imprimimos todos los valores para confirmar que se guardaron correctamente.

## Funciones:

- Main: Contendrá todas las variables que usaremos por todo el código, así evitamos problemas de variables locales o de usar variables globales.
- Menú: Se mostrará un menú en la terminal donde se pedirá ingresar una opción.
  1. Introducir nombre del archivo a leer.
  2. Usar Predeterminado
  3. Salir
- AskFile: Función llamada en el menú al seleccionar la opción 1, donde leerá el nombre del archivo y validará si se puede abrir o no.
- ReadFile: Lee el archivo con cierto nombre y guarda los valores en un pointer.
- Format: leerá con strtok y validará que tipo de variable se usará para almacenar esa variable, en este caso solo float. Guardará los valores en un ptr de tipo float.

Este Programa será compilado con ayuda de makefile ya que contará con varios módulos y así será más amigable al momento de compilarlo.

## Conclusiones:

Este programa debe ser capaz de leer cualquier archivo con valores float con "n" cantidad de datos, deberá almacenarlos como valores string y después convertirlos en valores tipo float.

Para una mejor funcionalidad se usarán funciones generales, quiere decir que se pueden reutilizar en otros códigos sin hacer modificación.

