

# Sistemas Operativos

## Práctica 1

### 1. Procesos y comunicación entre procesos

Se pretende realizar un programa que gestione la información contenida en el dataset CSV seleccionado por cada grupo de prácticas. El programa debe implementar un sistema de búsqueda eficiente utilizando los campos principales del dataset elegido.

Cada grupo deberá adaptar la estructura de datos y la interfaz según las características específicas de su dataset, manteniendo los siguientes principios generales:

- Identificar los campos clave del dataset (campos de búsqueda primarios y secundarios)
- Definir criterios de búsqueda relevantes para el tipo de datos
- Implementar un sistema de consulta interactivo

Al ejecutar el programa se muestra un menú adaptado al dataset seleccionado. Como ejemplo, para un dataset de tiempos de viaje, el menú sería:

Bienvenido

1. Ingresar primer criterio de búsqueda
2. Ingresar segundo criterio de búsqueda
3. Ingresar tercer criterio de búsqueda (si aplica)
4. Realizar búsqueda
5. Salir

Cada grupo deberá definir las opciones del menú según su dataset específico. Por ejemplo:

1. **Criterio 1.** Solicita el primer parámetro de búsqueda según el dataset. El usuario deberá validar el rango de valores permitidos.
2. **Realizar búsqueda.** Ejecuta la búsqueda con los criterios ingresados y presenta el resultado. En caso de no encontrar el valor, se muestra el mensaje "NA".
3. **Salir.**

### 1.1. Consideraciones técnicas

- Hacer uso de punteros y de memoria dinámica (`malloc()` - `free()`).
- Si genera un nuevo archivo indexado, se recomienda almacenar estructuras de forma binaria, no texto. El archivo con todo el conjunto de datos (dataset) deberá residir en disco, los datos **NO** deben estar en memoria. La cantidad de memoria empleada por el proceso no deberá superar 10MB.
- La búsqueda deberá ejecutarse en un tiempo menor a 2 segundos.
- Se deberán implementar dos procesos, uno para la apertura y búsqueda dentro del archivo y otro para la interfaz de usuario. La comunicación se deberá realizar mediante tuberías nombradas o memoria compartida. Los procesos deben ser **no** emparentados.
- Se debe implementar una tabla *hash* para la búsqueda dentro del archivo. Se recomienda indexar por el campo principal de búsqueda y generar estructuras de datos apropiadas (listas enlazadas, árboles, etc.) para optimizar las consultas según las características del dataset.
- Cada grupo debe documentar claramente la estructura de su dataset, los criterios de búsqueda implementados y las adaptaciones realizadas en el archivo LEEME.
- Entrega: Archivo fuente con *main* en **p1-dataProgram.c** (se pueden tener más archivos), archivo **Makefile** para compilar todo, archivo LEEME dentro de una carpeta con los nombres que aparecen en el correo para cada integrante. Ej: **apellido1a-apellido2b**. Este archivo o carpeta se entregará en clase.

### 1.2. Especificaciones por dataset

Cada grupo deberá incluir en su documentación:

- Descripción detallada de los campos del dataset seleccionado
- Justificación de los criterios de búsqueda implementados
- Rangos de valores válidos para cada campo de entrada
- Ejemplos específicos de uso del programa con su dataset

### 1.3. Calificación

Se tendrán en cuenta los siguientes aspectos para la evaluación:

- Funcionamiento del programa. 40 %
- Rapidez de la búsqueda (tabla hash, búsqueda <2 segundos). 10 %

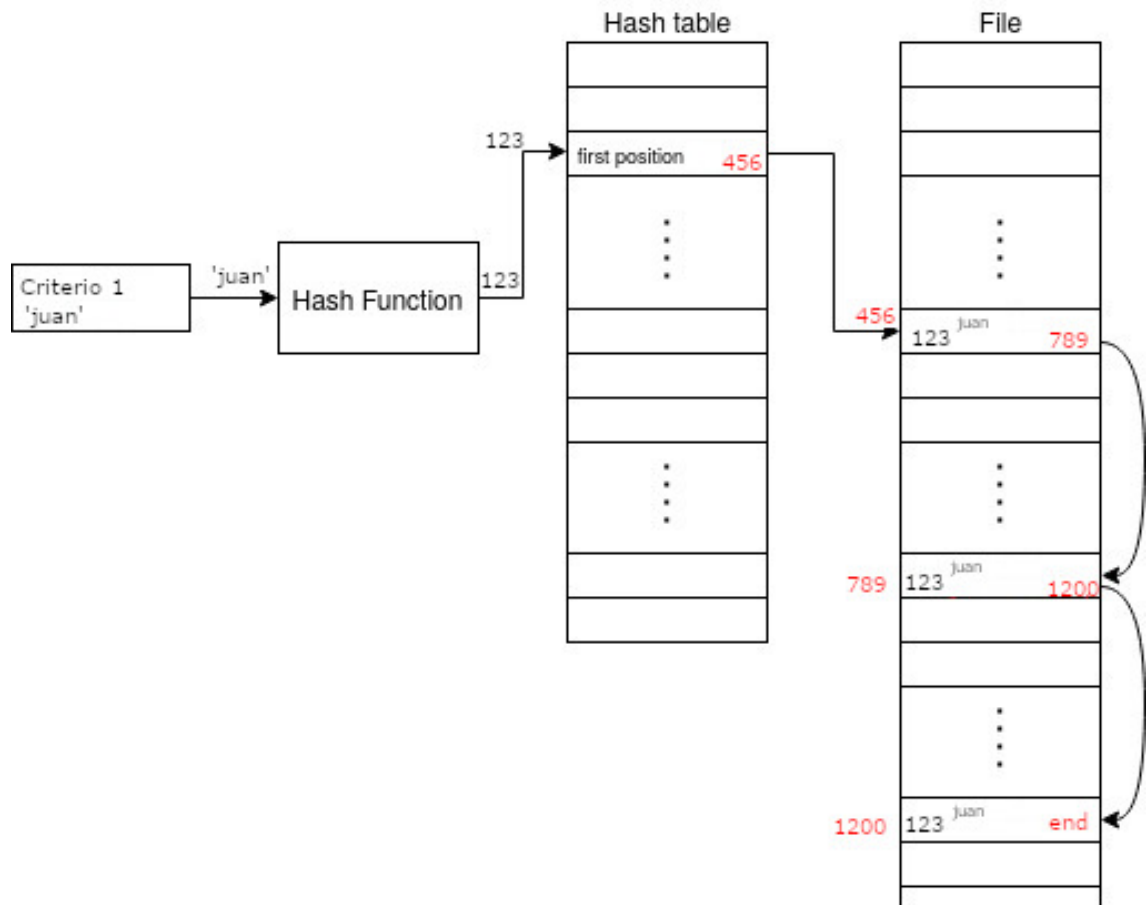


Figura 1: Estructura de indexación recomendada.



- Código limpio. 10 % (modular, tabulaciones, comentarios - básicos, declaración de constantes, etc.)
- Sustentación (se selecciona a cualquier integrante): 20 %
- Código en repositorio git (github u otro) 10 %
- Uso de herramientas como copilot, chatGPT, otras. Se demuestra en la sustentación. 10 %