

**Programación 1 - ICI/INF 125**  
**Ejercicios Clases**  
**Unidad 2: Librerías**  
**Unidad 3: Procesamiento de Archivos**

**Objetivo:** El siguiente ejercicio tiene por objetivo utilizar archivos, arreglos de estructuras, y librerías (headers).

El Rally Dakar es una competencia anual de *rally raid* organizada por la ASO a partir de 1979 y disputada durante las primeras semanas de enero. Es considerado uno de los rallies más duros y el más famoso del mundo.

Las cuatro categorías participantes son **coches, camiones, motocicletas y cuatrimotos**. El terreno varía considerablemente, atravesando zonas de arena, barro, rocas y vegetación hasta carreteras secundarias. La inscripción en la prueba es abierta, lo cual la convierte en la carrera idónea para amantes de la aventura.

La historia del rally está almacenada en un archivo de texto (Dakar.txt) bajo el siguiente formato:

**Año, Inicial Nombre, Apellido, Marca Automovil, Tipo de Automovil.**

Donde cada línea del archivo está separada por un salto de línea, por ejemplo:

```
1997 S. Peterhansel Yamaha Motos
```

A continuación se describen las funcionalidades que debe tener el programa, donde funciones y declaraciones deben estar en una librería creada por usted, de acuerdo a lo siguiente:

1. Implementar un programa en C que lea cada línea del archivo y lo almacene en un arreglo de estructuras. La declaración de la estructura, que se indica a continuación, debe estar en la librería creada por usted (no puede estar en el archivo donde se encuentra la función main())

```
struct autos{  
    int anio;  
    char nombre[2];  
    char apellido[20];  
    char marca[20];  
    char tipo[11]; //coches, motos, camiones, cuatrimotos  
};
```

El tamaño del arreglo es el número de líneas del archivo Dakar.txt (son 104 líneas). Use una constante para definir el tamaño del arreglo. La declaración de la constante debe estar en la librería creada por usted.

2. Para abrir archivos y verificar que se lograron abrir de forma correcta, implementar una función de acuerdo al siguiente prototipo:

```
FILE* AbrirVerificarArchivo(char nombre[], char *modo)
```

Donde el primer parámetro es el nombre del archivo, y el segundo es el modo con el que se abre el archivo. La función debe mostrar mensaje en caso de abrir el archivo sin problemas y en caso contrario.

3. Luego debe imprimir por pantalla todos los 20 primeros elementos del arreglo, como se muestra a continuación:

```
Año: 2011 -- Nombre: V.Chagin Chagin -- Marca:Kamaz -- Tipo: Camiones
Año: 2009 -- Nombre: J.Machacek Machacek -- Marca:Yamaha -- Tipo: Cuadrimotos
Año: 2010 -- Nombre: M.Patronelli Patronelli -- Marca:Yamaha -- Tipo: Cuadrimotos
Año: 1979 -- Nombre: C.Nevu Neveu -- Marca:RangeRover -- Tipo: Coches
Año: 1980 -- Nombre: F.Kottulinsky Kottulinsky -- Marca:Volkswagen -- Tipo: Coches
Año: 1984 -- Nombre: R.Metge Metge -- Marca:Porsche -- Tipo: Coches
Año: 1985 -- Nombre: P.Zaniroli Zaniroli -- Marca:Mitsubishi -- Tipo: Coches
Año: 1993 -- Nombre: B.Saby Saby -- Marca:Mitsubishi -- Tipo: Coches
```

4. Además, el programa debe crear 4 archivos, llamados:

- coches.txt
- motos.txt
- camiones.txt
- cuadrimotos.txt

Para abrir archivos y verificar correcto funcionamiento, utilice la función definida en el punto 2.

Cada archivo debe contener los datos del archivo Dakar.txt que correspondan a cada tipo de auto. Para generar los 4 archivos anteriores, **debe obtener los datos recorriendo el arreglo de estructuras**.

Para esto, en la librería creada por usted, cree la función:

```
FILE* EscribirArchivo(FILE *file, struct autos A[], int total, char *tipoAuto)
```

Donde:

Primer parámetro indica con qué archivo se trabajará, el segundo parámetro recibe todo el arreglo de estructuras del punto 1, el tercer parámetro el largo del arreglo de estructuras del parámetro dos, y finalmente el cuarto parámetro el tipo de auto en la estructura con el que trabajará.

5. Cerrar todos los archivos que abrió o creó en su programa.