

Sistema de Gestión de Resultados Electorales en el Rotary Club de Argentina

Objetivo:

Desarrollar un sistema en C para gestionar y procesar resultados de elecciones. El sistema debe permitir cargar datos desde un archivo de texto, llamado **votos.txt**, almacenar estos datos en un archivo binario llamado **datos.dat**, y manipular y mostrar los resultados a través de una **lista enlazada**.

Descripción del Problema:

Implementa un sistema para gestionar los resultados de una elección en tres provincias, cada una con varios distritos. Cada distrito tiene candidatos agrupados en listas. El sistema debe proporcionar las siguientes funcionalidades:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define NUM_LISTAS 5 // listas de condidatos
#define MAX_CADENA 81 // maximo de caracteres en una cadena
#define MAX_PROVINCIAS 3 // maximo de provincias
#define MAX_DISTRITOS 4 // maximo de distritos

typedef struct {
    int prov;
    int distrito;
    int lista;
    int votos;
} registro;

typedef struct elemento {
    int lista; // Número del candidato
    int votos; // Total de votos para ese candidato
    struct elemento *sig;
} nodo;
```

La carga de los nombres ya se encuentra en estas funciones

```
const char* obtenerNombreProvincia(int prov) {
    static const char* provincias[MAX_PROVINCIAS] = {"Buenos Aires", "Santa Fe", "Córdoba"};
    return (prov >= 0 && prov < MAX_PROVINCIAS) ? provincias[prov] : "Desconocida";
}

const char* obtenerNombreDistrito(int prov, int dis) {
    static const char* distritos[MAX_PROVINCIAS][MAX_DISTRITOS] = {
        {"La Plata", "Mar del Plata", "Lanús", "Lomas de Zamora"},
        {"Rosario", "Rafaela", "Venado Tuerto", "Reconquista"},
        {"Río Cuarto", "Villa María", "San Francisco", "Carlos Paz"}
    };
    return (prov >= 0 && prov < MAX_PROVINCIAS && dis >= 0 && dis < MAX_DISTRITOS) ?
    distritos[prov][dis] : "Desconocido";
}
```

```

const char* obtenerNombreCandidato(int lista) {
    static const char* candidatos[MAX_LISTAS] = {"Juan Gonzalez", "Marcos Alvarez", "Ana Lopez", "Nicolas Ramirez", "Sofia Flores"};
    return (lista >= 0 && lista < MAX_LISTAS) ? candidatos[lista] : "Desconocido";
}

```

```

// Liberar memoria de la lista enlazada
while (cabeza != NULL)
{
    nodo *temp = cabeza;
    cabeza = cabeza->sig;
    free(temp);
}

return 0;
}

```

```

void manejarErrorArchivo(FILE *archivo, const char *nombre)
{
    if (archivo == NULL)
    {
        printf("Error al abrir el archivo %s\n", nombre);
        exit(1);
    }
}

```

```

int main()
{
    FILE *archivo, *binario;
    char nombre[] = "votos.txt";
    char *nombin = "datos.dat";
    nodo *cabeza = NULL;
    system("color 9f");

    // Abrir archivos
    archivo = fopen(nombre, "r");
    manejarErrorArchivo(archivo, nombre);

    binario = fopen(nombin, "wb+");
    manejarErrorArchivo(binario, nombin);
}

```

1. Cargar Datos:

- Leer los datos de resultados electorales desde un archivo de texto votos.txt. Cada línea del archivo debe seguir el formato: **provincia,distrito,lista,votos** (por ejemplo: 1,2,3,100).

- Actualizar un **archivo binario datos.dat** con los resultados leídos. Si los datos para una provincia, distrito y lista ya existen en el archivo binario, incrementar el número de votos; de lo contrario, agregar una nueva entrada.
2. **Mostrar Registros:**
 - Leer y mostrar el contenido del archivo binario datos.dat. La información debe incluir la provincia, el distrito, la lista y el número de votos.
 3. **Buscar Datos:**
 - Permitir al usuario buscar los resultados de una provincia, distrito y lista específicos. Imprimir la información correspondiente si se encuentra en el archivo binario.
 4. **Mostrar Resultado General de Votos:**
 - Calcular y mostrar el resultado total de votos para cada lista de candidatos a nivel nacional. Los resultados deben ser ordenados de mayor a menor número de votos.
 5. **Ingresar Resultados a la Lista Enlazada:**
 - Leer el archivo binario datos.dat y almacenar el total de votos por lista en una lista enlazada. Cada nodo de la lista enlazada debe contener el número de lista y el total de votos.
 6. **Mostrar Lista Enlazada:**
 - Imprimir el contenido de la lista enlazada, mostrando la lista y el total de votos correspondientes.
 7. **Salir:**
 - Finalizar la ejecución del programa.

Datos de Entrada:

- **Archivo de Texto votos.txt:**
- Contiene líneas con el formato: **provincia,distrito,lista,votos.**

Estructura de Datos:

- **Estructura registro:** Contiene información sobre la provincia, distrito, lista y votos.
- **Lista Enlazada:** Utilizada para almacenar y mostrar los resultados de votos por lista.

Funciones Requeridas:

- cargar(FILE* archivo, registro reg): Cargar datos desde el archivo binario.
- mostrar(FILE* archivo): Mostrar registros del archivo binario.
- buscar(FILE* archivo, int provincia, int distrito, int lista): Buscar y mostrar resultados específicos.
- imprimegeneral(FILE* archivo): Mostrar el resultado general de votos a nivel nacional.
- insertar(nodo** cabeza, int lista, int votos): Insertar resultados en la lista enlazada.
- listar(nodo* cabeza): Mostrar el contenido de la lista enlazada.

