

Estructuras de Datos y Algoritmos II

Práctico de máquina - Año 2026 - Segundo Cuatrimestre

Fecha de entrega: Viernes 25 de septiembre de 2026 hasta las 12 hs.

Se desea implementar un registro del padrón electoral de San Luis definido con la siguiente relación y los siguientes servicios de evocación:

<i>Padrón</i>	$\subseteq$	<i>DNI</i>	$\times$	<i>Circuito</i>	$\times$	<i>Nombre_Apellido</i>	$\times$	<i>Domicilio</i>	$\times$	<i>Cód..Postal</i>	$\times$	<i>Mesa</i>
S_1	:	*		?		?		?		?		?
S_2	:	?		*		?		?		?		

donde $DNI \rightarrow (Circuito, Nombre_Apellido, Domicilio, Cód..Postal, Mesa)$ y
 $Circuito \rightarrow (DNI, Nombre_Apellido, Domicilio, Cód..Postal)$

Se necesita diseñar una aplicación que permita resolver los requerimientos y para ello se cuenta con las siguientes estructuras de datos para almacenar y resolver los servicios mencionados:

- Rebalse Abierto Lineal (**RAL**) para S_1 definido como primario.
- Direccionamiento Directo (**DD**) para S_2 .

La aplicación deberá presentar un menú que muestre las siguientes opciones para administrarla:

- Consulta de electores: debe pedir el dni al usuario, realizar la evocación y con el resultado obtenido informar al usuario convenientemente.
- Consulta de circuitos: debe pedir el circuito al usuario, realizar la evocación e informar al usuario convenientemente.
- Consulta de balde contenedor: debe pedir el dni de un elector al usuario, localizar el elector retornando el balde que lo contiene y mostrar todas las ranuras del balde por pantalla presentando la información completa de los elementos presentes, y para las ranuras que no están ocupadas, diferenciar claramente las ranuras *libres* de las ranuras *nunca usadas*.
- Mostrar estructura S_1 : debe mostrar por pantalla el contenido de la **estructura completa**. Se deben mostrar las **M*r** posiciones de la estructura, presentando la información completa de los elementos presentes en ella, y para las ranuras que no están ocupadas, diferenciar claramente las ranuras *libres* de las ranuras *nunca usadas*.
- Mostrar estructura S_2 : debe mostrar por pantalla el contenido de la **estructura completa** listando para cada circuito presente los electores asociados o indicar que el circuito no tiene electores.
- Análisis de costos: en esta opción debe realizar y mostrar un análisis adecuado de lo que cuesta **realizar ingresos, eliminaciones y consultas de un elector dado**, y también el costo de **consultas de un circuito dado**. En el análisis debe considerar el peor escenario y el comportamiento esperado en cada caso. Una vez finalizada esta operación, deberá realizar un análisis de los resultados obtenidos y sacar una conclusión de los mismos; dicha conclusión deberá quedar plasmada al principio de su programa principal (donde se encuentra el main) como comentario (incluir los resultados de la comparación). Los costos para todas las operaciones se medirán tanto en accesos a disco como en celdas consultadas (mostrar valores por separado).

Para el análisis se utilizará una secuencia de operaciones detallada en el archivo de texto "*Electores.txt*" provisto por la cátedra. Esta secuencia de operaciones se deberá realizar sobre la estructura, asegurando que la misma **no contenga ningún dato inicialmente**. Una vez finalizada la secuencia de operaciones se mostrarán por pantalla los costos obtenidos.



El archivo de texto “*Electores.txt*” contiene una línea con el código de operación (1-Alta, 2-Baja, 3-Evocación asociativa y 4-Evocación no Asociativa) y a continuación los datos de la nupla necesarios para la operación en cada línea (renglón) del mismo. Un ejemplo de esa información se muestra a continuación:

```

1                               /*código de la primera operación (Alta)*/
16542177                       /*DNI*/
1030                           /*circuito*/
Carlos Ana Herrera             /*Nombre y apellido*/
calle 4 DE JUNIO 100           /*Domicilio*/
5701                           /*Código Postal*/
50                             /*Mesa de votación*/
3                               /*código de la segunda operación (Evocación asociativa)*/
23654123                       /*DNI*/
4                               /*código de la tercera operación (Evocación no asociativa)*/
1010                           /*Circuito*/
.
.
.
2                               /*código de la n-ésima operación (Baja)*/
96473442                       /*DNI*/
1040                           /*circuito*/
Maria Facundo Fernandez       /*Nombre y apellido*/
Calle Torres 6306             /*Domicilio*/
5700                           /*Código Postal*/
150                            /*Mesa de votación*/

```

Consideraciones a tener en cuenta:

- El número esperado de electores a almacenar es de 202000.
- El D.N.I. es un entero.
- El Circuito es un entero y se sabe que son 101 circuitos (desde el 1000 al 1100).
- El campo Nombre y apellido es una secuencia de hasta 80 caracteres.
- El campo domicilio es una secuencia de hasta 150 caracteres.
- El Código Postal es un entero.
- El Número de mesa es un entero.
- El factor de carga será $\rho = 0.73$ para S_1 .
- Las estructuras completas deberán guardarse en disco, para el cual se tiene un tamaño de página de 4 KB (calcular ranuras por balde, baldes y desperdicio si tuviera para S_1).
- El ingreso de datos **no debe ser sensible a mayúsculas y minúsculas**, esto significa por ejemplo que SOSA = SoSa = sosa.
- Se debe utilizar la siguiente función de hashing para S_1 :

```

int hashing (int dni, int M) {
    char x[8];
    int longitud, i;
    int contador=0;
    sprintf(x, "%d", dni);
    longitud=strlen(x);
    for (i=0; i< longitud; i++)
        contador+=((int)x[i]) * (i+1);

    return (contador % M );
}

```



- El programa deberá desarrollarse en Lenguaje C, utilizando como herramienta para tal fin **Code::Blocks** (disponible en www.codeblocks.org).

Nota Importante: La entrega del práctico se realiza por medio de la página de la materia y se debe enviar el archivo fuente del programa. El nombre del archivo deberá estar conformado de la siguiente manera: ***PnroP-GruponroG*** donde *nroP* es reemplazado por el número de práctico que se entrega y *nroG* por el número del grupo al que pertenece el programa. Por ejemplo, el nombre P1-Grupo22.c corresponde al práctico de máquina 1 enviado por el grupo 22. **Los programas cuyos nombres no respeten estas reglas de conformación no serán aceptados.** También puede ser entregado un archivo comprimido respetando la nomenclatura anterior. En cualquier caso no debe entregarse el archivo ejecutable (.exe) sólo los archivos con el código fuente.