



## ALGORITMOS Y ESTRUCTURAS DE DATOS – TRABAJO PRÁCTICO

**Algoritmos y Estructuras de Datos F.C** es un equipo de fútbol fundado en 1988, y desde su nacimiento participa de las ligas menores de Argentina. Bajo la coordinación de su último presidente, el club ascendió hasta la segunda división y con ello su popularidad en todo el país, lo cual genera que todos los meses sus socios sigan aumentando.

Ante este crecimiento, el sector Administrativo del club se encuentra en la búsqueda de **un sistema que les permita encargarse de manera más organizada la gestión de los socios**. Actualmente la información de los mismos es manejada desde un archivo de texto con el siguiente formato:

Campo	Tipo	
Nro. Socio (Clave)	Long	1 < Nro. Socio < 10.000.000
Apellido, Nombre	Cadena (60)	
DNI	Long	10.000 < DNI < 100.000.000
Fecha de Nacimiento	T_Fecha	
Sexo	Char	'F' (Femenino), 'M' (Masculino) u 'O' (Otro)
Fecha de afiliación	T_Fecha	
Categoría	Cadena (10)	'MENOR', 'CADETE', 'ADULTO', 'VITALICIO', 'HONORARIO', 'JUBILADO'. Validar <sup>(1)</sup>
Fecha de última cuota paga	T_Fecha	
Estado	Char	'A' Activo, 'I' Inactivo - Se genera en el alta con 'A'
Fecha de Baja	T_Fecha	Si el estado es 'A', éste es vacío

El archivo es de registros de longitud variable con los campos separados por "|" y NO se encuentra ordenado por ningún campo en particular.

### Primera aplicación: Generar el índice

Construir un programa que lea el archivo de texto "socios.txt" y genere el archivo "socios.dat" transformando los datos a binarios. Para ello, el usuario debe ingresar por teclado el path (ruta de acceso) en el que se encuentra el archivo.

Una vez finalizada la creación del archivo "socios.dat" construir un TDA índice (t\_indice) **genérico**, y utilícelo para crear un índice para el archivo de socios donde en cada elemento o entrada de este se almacena el Nro. Socio (void \*clave) y el número de registro (unsigned nro\_reg) que le corresponde a ese socio en el archivo. Una vez generado por completo el índice, debe ser guardado en un archivo con el nombre "socios.idx" antes de finalizar el programa.



**UNIVERSIDAD NACIONAL DE LA MATANZA**  
**Departamento de Ingeniería e Investigaciones Tecnológicas**  
**Algoritmos y Estructuras de Datos**

Las primitivas del TDA son:

- **void ind\_crear (t\_indice\* ind, size\_t tam\_clave, int (\*cmp)(const void\*, const void\*))**: inicializa la estructura a índice vacío.
- **int ind\_insertar (t\_indice\* ind, void \*clave, unsigned nro\_reg)**: inserta en orden el registro reg\_ind según el Nro. Socio. <sup>(1)</sup>
- **int ind\_eliminar (t\_indice\* ind, void \*clave, unsigned nro\_reg)**: recibe el Nro. Socio a eliminar en el parámetro reg\_ind y devuelve en reg\_ind el dato completo. <sup>(1)</sup>
- **int ind\_buscar (const t\_indice\* ind, void \*clave, unsigned nro\_reg)**: recibe el dni en reg\_ind, y, si existe, devuelve en reg\_ind el dato completo. <sup>(1)</sup>
- **int ind\_cargar (t\_indice\* ind, const char\* path)**: Carga el índice a partir de un archivo ordenado, cada registro del archivo debe tener la estructura definida. <sup>(1)</sup>
- **int ind\_grabar (const t\_indice\* ind, const char\* path)**: Graba un archivo con el contenido del índice <sup>(1)</sup>
- **void ind\_vaciar (t\_indice\* ind)**: deja el índice en su estado de vacío.
- **int ind\_recorrer (const t\_indice\* ind, void (\*accion)(const void \*, unsigned, void \*), void\* param)**: Recorre el índice en orden y llama a acción para cada registro del mismo. <sup>(1)</sup>

Opcionales:

- **int ind\_primer (t\_indice\* ind, void \*clave, unsigned nro\_reg)**: Deja el primer registro de índice en reg\_ind; <sup>(1)</sup>
- **int ind\_siguiete (t\_indice\*, void \*clave, unsigned nro\_reg)**: Deja el registro siguiente al último entregado en reg\_ind. <sup>(1)</sup>
- **int ind\_fin (const t\_indice \*ind)**: Devuelve *verdad* si la última operación de acceso secuencial no entregó el registro por haber llegado al fin de la secuencia y falso en caso contrario.

<sup>(1)</sup> : Devuelve 1 (uno) si la operación fue exitosa y 0 (cero) en caso contrario.

IMPORTANTE: Elegir una de las implementaciones del TDA Índice alguno de los anexos de este documento.

### Segunda aplicación: Gestión de Socios

Mostrar por pantalla un menú que permita al usuario:

- ✓ (A) Dar de alta un nuevo socio solicitando todos los campos de la estructura socios. Verificar previamente que el DNI ingresado no exista anteriormente en el club. Si existe y el estado es 'A', informarlo por pantalla, pero si el estado es 'B', cambiarlo a activo y quitar la fecha dada de baja. El campo Nro. Socio es secuencial, por lo que se tomará el mayor existente y se incrementará en uno para asignar el nuevo número.
- ✓ (M) Modificar el campo "Apellido, Nombre" a partir de un Nro. Socio a buscar. Validar el formato del campo según la estructura inicial.



**UNIVERSIDAD NACIONAL DE LA MATANZA**  
**Departamento de Ingeniería e Investigaciones Tecnológicas**  
**Algoritmos y Estructuras de Datos**

- ✓ (B) Dar de baja un socio cargando el Nro. Socio a eliminar. Si existe y el estado es 'A', cambiarlo a 'B' y solicitar por pantalla la fecha de baja, la cual se guardará en el campo Fecha de Baja. Si no existe o ya está dado de baja, también informarlo por pantalla.
- ✓ (L) Listar todos los socios dados de baja ordenado por Nro. Socio.
- ✓ (V) Visualizar todos los socios ordenados (sin mostrar los dados de baja) por Nro. Socio
- ✓ (P) Listar los 10 socios con mayor retraso en la fecha de pago de la cuota.
- ✓ (S) Salir

Una vez que el usuario seleccione Salir, No olvidar:

- Grabar el archivo de índices.
- Liberar la memoria del índice.

**IMPORTANTE PARA LA ENTREGA**

La resolución del Trabajo Práctico Integrador puede ser grupal (4 personas como máximo) pero la entrega debe realizarse desde la opción Portafolio de MIEL en **forma individual a "todos los docentes"**. La misma debe subirse zipeado en el formato:

DNI\_APELLIDO\_NOMBRE.zip (*completar su/s apellido/s y nombre/s tal cual figura en MIEL*)

Ej.: 33644949\_CUESTA\_CRISTIAN\_DANIEL.zip



### **Implementaciones de Índice**

#### **TDA Índice (Array Estático)**

Implementación del TDA *índice*, donde cada elemento o entrada del índice se guarda en las posiciones de un array definido estáticamente.

#### **TDA Índice (Array Redimensionable)**

Implementación del TDA *índice*, donde cada elemento o entrada del índice se guarda en las posiciones array creado con memoria dinámica, el cual se va a ser inicializado con un tamaño pequeño y con el uso se va a ir agrandando para guardar todos los elementos que sea necesario almacenar.

#### **TDA Índice (Lista Simplemente Enlazada)**

Implementación del TDA *índice*, donde cada elemento o entrada del índice se guarda en los nodos de una lista simplemente enlazada. El índice debe incluir el TDA Lista y utilizar sus primitivas.

#### **TDA Índice (Árbol)**

Implementación del TDA *índice*, donde cada elemento o entrada del índice se guarda en los nodos de un árbol. El índice debe incluir el TDA Árbol y utilizar sus primitivas.