



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

**Alumno:**

**Ramírez Monjaraz Miguel Ángel**

**Profesor:**

[Pedro Nunez Yepiz](#)

**Ingeniería en computación**

**Grupo: 432**

**Materia:**

**Programación - Estructurada**

**Tema:**

**Actividad 11**

**FUNCIONES y METODOS DE ORDENACION Y BUSQUEDA  
ESTRUCTURAS Y LIBRERIAS (p2)**

**Fecha de entrega:**

**03 de Noviembre del 2023**



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

### Ingeniero en computación

**Materia:** Programación Estructurada / Clave 36276

**Alumno:** Miguel Angel Ramirez Monjaraz

**Matrícula:** 372205

**Maestro:** Pedro Núñez Yépiz

**Actividad No. :** 11 - SEMANA 12

**Tema - Unidad :** FUNCIONES y METODOS DE ORDENACION Y BUSQUEDA  
ESTRUCTURAS Y LIBRERÍAS PARTE 2

**Ensenada Baja California a 03 de Noviembre del 2023**



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

### 1. INTRODUCCIÓN

En el contexto de C++, las librerías, también conocidas como bibliotecas, se refieren a archivos que pueden ser importados o incluidos en nuestro programa. Estos archivos contienen las especificaciones de diversas funcionalidades previamente desarrolladas y utilizables que pueden ser incorporadas a nuestro programa. Ejemplos de estas funcionalidades incluyen la lectura desde el teclado o la visualización en pantalla, entre muchas otras.

La inclusión de estas librerías con definiciones de diversas funcionalidades nos permite simplificar significativamente nuestro trabajo. Imaginemos, por ejemplo, tener que crear una función desde cero cada vez que necesitemos leer desde el teclado, lo cual sería un proceso bastante complejo. Sin embargo, gracias a las librerías en C++, podemos aprovechar una amplia variedad de funciones preexistentes que simplifican nuestra labor y mejoran la modularidad de nuestro código. Es por esto que se vio la necesidad de que el alumno cree su propia librería que irá llenando al transcurso del curso, para esta práctica se utilizó funciones como búsqueda, ordenación, además se trabajo con creación de archivos txt.

### 2. COMPETENCIA

La competencia de esta práctica es utilizar los conocimientos adquiridos en las prácticas anteriores sobre, métodos de ordenación y de búsqueda, así como emplear la creación de archivos txt. Para realizar un programa eficiente.

### 3. FUNDAMENTOS



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

### Librerías

Sintaxis para declarar Librerías en C++

La declaración de librerías, tanto en C como en C ++, se debe hacer al principio de todo nuestro código, la forma de declarar librerías en C ++ es esencial para asegurar que el compilador reconozca y utilice las funcionalidades proporcionadas por esas librerías en tu programa. Esto se hace al principio del código, antes de cualquier declaración de función o línea de código. La sintaxis es la siguiente:

```
#include <nombre_de_librería>
```

```
#include "nombre_de_librería"
```

Ambas formas son válidas en C ++. Es importante notar que el nombre de la librería debe estar encerrado entre comillas angulares < > o comillas dobles " ". Puedes declarar todas las librerías que desees en tu programa, aunque es recomendable declarar solo aquellas que realmente necesitas para evitar la carga innecesaria de recursos.

Existen varias librerías estándar que son ampliamente utilizadas para realizar una amplia variedad de tareas. Algunas de las librerías más comunes y utilizadas en C son:

stdio.h: Esta librería proporciona funciones para entrada y salida estándar, como printf, scanf, fopen, fclose, entre otras. Es esencial para la lectura y escritura de datos en la consola y en archivos.

stdlib.h: Contiene funciones para la gestión de memoria dinámica, como malloc, free, y funciones para controlar la terminación de un programa, como exit.

string.h: Ofrece funciones para el manejo de cadenas de caracteres, como strlen, strcpy, strcat, y muchas otras.

math.h: Proporciona funciones matemáticas para operaciones matemáticas avanzadas, como trigonometría, exponenciación, logaritmos, etc.

time.h: Se utiliza para trabajar con fechas y horas, incluyendo funciones para medir el tiempo y realizar cálculos de tiempo.

Estas son algunas de las librerías más utilizadas en C, pero hay muchas otras disponibles para tareas específicas. La elección de la librería depende de los requisitos de tu programa y de las funciones que necesitas implementar.



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

### Metodos de ordenacion

Una gran cantidad de problemas, de muchos tipos, requieren para su solución que los datos se encuentren en orden. La utilidad de trabajar con información ordenada es inmensa, en ocasiones no sólo facilita la solución de un problema, sino que además puede llegar a ser la única solución.

Es probable que te hayas preguntado sobre la utilidad práctica de ordenar los datos. La realidad es que el ordenamiento de datos simplifica significativamente las operaciones de búsqueda. Este concepto se puede apreciar en situaciones cotidianas, como cuando buscamos un contacto en la agenda de nuestro teléfono. Por lo general, conocemos el nombre del contacto que deseamos encontrar (el valor que estamos buscando) y entendemos que la agenda organiza los contactos en orden alfabético. Como resultado, cuando buscamos a alguien cuyo nombre comienza con "Z", sabemos que estará ubicado al final de la lista de contactos. Del mismo modo, si buscamos a alguien cuyo nombre comienza con "B", sabemos que estará al principio de la lista.

Este enfoque facilita considerablemente la tarea de encontrar información en grandes conjuntos de datos, ya que evita la necesidad de revisar todos los elementos uno por uno, lo que ahorra tiempo y esfuerzo. En resumen, el ordenamiento de datos es una estrategia esencial que se aplica en numerosos contextos para mejorar la eficiencia en la búsqueda y recuperación de información. La definición más simple de búsqueda, es literalmente determinar si un elemento se encuentra en un determinado conjunto. Este concepto se puede expandir hasta ser definida como una relación posible de estados, el primer aspecto importante en la búsqueda de datos, es la forma en la que se almacenan e indexan.

Los datos que hemos tratado hasta el momento han residido en la memoria principal. Sin embargo, las grandes cantidades de datos se almacenan normalmente en un dispositivo de memoria secundaria. Estas colecciones de datos se conocen como archivos (antiguamente ficheros). Un archivo es un conjunto de datos estructurados en una colección de entidades elementales o básicas denominadas registros que son de igual tipo y constan a su vez de diferentes entidades de nivel más bajos denominadas campos. Hay dos tipos de archivos, archivos de texto y archivos binarios. Un archivo de texto es una secuencia de caracteres organizadas en líneas terminadas por un carácter de nueva línea. En estos archivos se pueden almacenar canciones, fuentes de programas, base de datos simples, etc. Los archivos de texto se caracterizan por ser planos, es decir, todas las letras tienen el mismo formato y no hay palabras subrayadas, en negrita, o letras de distinto tamaño o ancho.



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

El puntero a un archivo. El puntero a un archivo es el hilo común que unifica el sistema de E/S con buffer. Un puntero a un archivo es un puntero a una información que define varias cosas sobre él, incluyendo el nombre, el estado y la posición actual del archivo. En esencia identifica un archivo específico y utiliza la secuencia asociada para dirigir el funcionamiento de las funciones de E/S con buffer. Un puntero a un archivo es una variable de tipo puntero al tipo FILE que se define en STDIO.H. Un programa necesita utilizar punteros a archivos para leer o escribir en los mismos. Para obtener una variable de este tipo se utiliza una secuencia como está:

FILE \*F; Apertura de un archivo. La función fopen() abre una secuencia para que pueda ser utilizada y la asocia a un archivo.

Su prototipo es: FILE \*fopen(const char nombre\_archivo, cost charmodo);

Donde nombre\_archivo es un puntero a una cadena de caracteres que representan un nombre válido del archivo y puede incluir una especificación del directorio. La cadena a la que apunta modo determina cómo se abre el archivo.

Acciones permitidas para \*f;

Modo	Significado
r	Abre un archivo de texto para lectura.
w	Crea un archivo de texto para escritura.
a	Abre un archivo de texto para añadir.
rb	Abre un archivo binario para lectura.
wb	Crea un archivo binario para escritura.
ab	Abre un archivo binario para añadir.
r+	Abre un archivo de texto para lectura / escritura.
w+	Crea un archivo de texto para lectura / escritura.
a+	Añade o crea un archivo de texto para lectura / escritura.
r+b	Abre un archivo binario para lectura / escritura.
w+b	Crea un archivo binario para lectura / escritura.
a+b	Añade o crea un archivo binario para lectura / escritura.



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

La función `fopen()` devuelve un puntero al archivo. Un programa nunca debe alterar el valor de ese puntero. Si se produce un error cuando se está intentando abrir un archivo, `fopen()` devuelve un puntero nulo.

#### 4. PROCEDIMIENTO



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

### PRÁCTICA 11

Realiza el programa que contenga el siguiente menú

#### M E N Ú

- 1.- Cargar
- 2.- Eliminar
- 3.- Buscar
- 4.- Ordenar
- 5.- Imprimir
- 6.- Archivo Texto
- 0.- Salir

El programa deberá poder almacenar en un arreglo (máximo 2,000 registros) los datos para generar el CURP **la estructura debe contener 2 estructuras anidadas**, nombre y fecha nacimiento y un campo donde se escribirá automáticamente el curp basado en los datos proporcionados

#### MENÚ DESCRIPCIÓN:

- 1.- **Cargar**: Se deberá agregar 100 registros en forma automática y aleatorios (cuidar no se desborde Arreglo)
- 2.- **Eliminar**: La búsqueda se realizará por matrícula, Imprimir el registro encontrado en forma de registro y preguntar si quiere eliminar si o no. (Eliminado Lógico x campo status)
- 3.- **Buscar**: La búsqueda se realizará por matrícula, el programa deberá ser capaz de realizar la **búsqueda secuencial o Binaria** según sea el caso. Imprimir el registro encontrado en forma de registro.





# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

4.- **Ordenar**: La ordenación será por MATRICULA usar función de ordenación más adecuada según sea el caso **usar 2 métodos de ordenación** y el programa decidirá cuál es el que usará dependiendo del estado y tamaño de registros dentro del arreglo.

**Nota:** (validar si el arreglo ya está ordenado no volver ordenar por el mismo campo)

5.- **Imprimir**: El programa deberá imprimir los datos del arreglo (**solo registros activos**) en forma de tabla en pantallas de 40 registros y presionando la tecla de continuar en cada uno de los casos.

6.- **Archivo de Texto**: El programa deberá generar un archivo de texto con los datos del arreglo (**solo registros activos**) formatear salida.

**NOTA:** forma de registro es de la siguiente manera:

**MATRICULA** : 300523

**NOMBRE** : YAREMI

**NOMBRE2** : GHIZETH

**AP PATERNO** : GARCIA

**AP MATERNO** : GUERRERO

**FECHA NAC** : 03-04-2010

**EDAD** : 19

**SEXO** : MUJER

**LUGAR NAC** : BAJA CALIFORNIA SUR

**CURP** : GAGY030410MBCRRRA5

**NOTA** : Librería Propia, Usar funciones, no se permiten variables global



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

### 5. RESULTADOS Y CONCLUSIONES

En conclusión, usar archivos te permite gestionar y respaldar, datos que necesites generar, esto combinado con los tipos de datos creados con struct. Logra una unificación del código, debido a que las estructuras permiten agrupar variables relacionadas en una sola entidad, lo que facilita la organización y la manipulación de datos. En conclusión la práctica se llevo acabo bien , sin embargo las instrucciones no fueron del todo claras.

### 6. ANEXOS

[https://docs.google.com/document/d/1ASY\\_IP7duid3GXse7VXVQ-APnOL8aB-cX3K1TdDIs2U/edit](https://docs.google.com/document/d/1ASY_IP7duid3GXse7VXVQ-APnOL8aB-cX3K1TdDIs2U/edit)



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

### 7. REFERENCIAS

#### **Diseño de algoritmos y su codificación en lenguaje C**

Corona, M.A. y Ancona, M.A. (2011)..

España: McGraw-Hill.

ISBN: 9786071505712

#### **Programación estructurada a fondo: implementación de algoritmos en C**

:Pearson Educación.Sznajdleder, P. A. (2017)..

Buenos Aires,Argentina: Alfaomega

#### **Como programar en C/C++**

H.M. Deitel/ P.J. Deitel

Segunda edición

Editorial: Prentice Hall.

ISBN:9688804711

#### **Programación en C.Metodología, estructura de datos y objetos**

Joyanes, L. y Zahonero, I. (2001)..

España:McGraw-Hill.

ISBN: 8448130138

#### **Página web Poetry**

Include Poetry - Code. (2020, 4 de enero). Introducción a ordenamientos. Obtenido de <https://www.include-poetry.com/Code/C++/Metodos/Ordenamientos/>