



# 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 9**

**Librerías**

**Fecha de entrega:**

**10 de Octubre 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. :** 9 - SEMANA 9

**Tema - Unidad :** Librerías

Unidad 1

**Ensenada Baja California a 10 de Octubre 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.

### 2. COMPETENCIA

La competencia de esta práctica es familiarizarse con el uso de librerías en el lenguaje de programación C++ y comprender cómo pueden mejorar la modularidad y eficiencia de un programa.

### 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.



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

### 4. PROCEDIMIENTO

#### MENÚ

- 1.- LLENAR VECTOR
- 2.- LLENAR MATRIZ
- 3.- IMPRIMIR VECTOR
- 4.- IMPRIMIR MATRIZ
- 5.- ORDENAR VECTOR
- 6.- BUSCAR VALOR EN VECTOR
- 0.- SALIR

NOTA: El programa deberá repetirse cuantas veces lo desee el usuario, Validado el menú con la función vali\_num

#### INSTRUCCIONES

- 1.- LLENAR VECTOR .- Llenar vector con 15 números, los números generados aleatoriamente, los números entre el rango de 100 al 200 (no repetidos)
- 2.- LLENAR MATRIZ .- Llenar la matriz de 4x4 con con números generados aleatoriamente, números entre el rango de 1 al 16 (no repetidos)
- 3.- IMPRIMIR VECTOR .- Imprime el vector que se envíe, donde la función recibe como parámetro el vector,tamaño, nombre del vector.
- 4.- IMPRIMIR MATRIZ.- Imprime la matriz sin importar el tamaño de la matriz recibiendo como parámetros la matriz, la cantidad de renglones y columnas, así como nombre que se le dará a la matriz
- 5.- ORDENAR VECTOR.- Usar función que ordene el vector por el método de ordenación de la Burbuja mejorada.
- 6.- BUSCAR VALOR EN VECTOR.- Buscar un valor en el vector usando el método de búsqueda secuencial.
- 0.- SALIR



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

### 5. RESULTADOS Y CONCLUSIONES

En conclusión, combinar el empleo de librerías junto con la práctica de ordenar datos constituye un enfoque poderoso para crear aplicaciones eficaces en C. Las librerías proporcionan una base sólida de recursos predefinidos, mientras que el proceso de ordenar datos aumenta tanto la velocidad como la precisión de las operaciones de búsqueda. Esto resulta fundamental en el manejo de información y en la optimización de algoritmos. Estas estrategias son esenciales para alcanzar el éxito en el campo de la programación en C y para llevar a cabo una gestión eficiente de los datos.

Nuestros resultados en esta práctica fueron positivos, se logró observar cómo se modulaba el código al hacer uso de nuestra librería, la cual recopila las funciones que se reutilizaban en nuestro código principal.

### 6. ANEXOS

<https://docs.google.com/document/d/1zbr7uORD7NUs0Tjci6AJ-qGaW43TqPgVOlgTk0-ZLGQ/edit>



## 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/>