



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

### Ingeniero en Software y tecnologías emergentes

**Materia:** Programación Estructurada / Clave 36276

**Alumno:** Diarte Salas Gilberto

**Matrícula:** 360954

**Maestro:** Pedro Núñez Yépiz

**Actividad No.** : 9

**Tema - Unidad** : librerías en c, métodos de ordenación y búsqueda

**Ensenada Baja California a 08 de Octubre del 2023**



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

### 1. INTRODUCCIÓN

Durante esta práctica el alumno comprenderá como funcionan las librerías personales, así como los métodos de ordenación de vectores y búsqueda de valores dentro del mismo.

### 2. COMPETENCIA

Se busca el desarrollo de una librería personal del alumno, haciendo comprensión de cómo realizarlas e implementarlas en otro archivo .cpp, con esto se busca que el alumno tenga a su alcance varias funciones disponibles para su posterior utilización.

Así mismo comprende los métodos de ordenación de datos dentro de un vector y la búsqueda de datos dentro del mismo.

### 3. FUNDAMENTOS

La utilización de una librería personal puede llegar a ser muy beneficiosa si es que así lo queremos, esto debido a que al crear una librería tenemos acceso a las funciones que dentro se encuentran, así como tenemos acceso también tenemos el poder de modificarlas, agregar más funciones o eliminar funciones. Todo esto implica llevar la codificación a un nivel más optimo, tanto en entendimiento por parte del alumno como de ejecución en el programa creado.



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

### 4. PROCEDIMIENTO

#### ACTIVIDAD 9

Realiza programa en C utilizando librería propia, el programa deberá tener el siguiente menú.

##### 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

Al realizar la práctica, encontramos cuan útiles pueden ser las librerías personales, así mismo encontramos la utilidad para las funciones de búsqueda y ordenamiento de valores.

```
void ordenar(int vect[], int n)
{
    // Funcion para ordenar un vector de manera ascendente
    int i, j, temp;
    for (i = 0; i < n - 1; i++)
    {
        for (j = i + 1; j < n; j++)
        {
            if (vect[j] < vect[i])
            {
                temp = vect[i];
                vect[i] = vect[j];
                vect[j] = temp;
            }
        }
    }
}

int busq_sec(int vect[], int n, int num)
{
    // Funcion para buscar un numero en un vector y devolver su posicion
    int i;
    for (i = 0; i < n; i++)
    {
        if (vect[i] == num)
        {
            return i;
        }
    }
    return -1;
}
```

### 6. ANEXOS

[https://github.com/Gilber19/Programacion\\_Estructurada](https://github.com/Gilber19/Programacion_Estructurada)

Se anexa PDF con ejecuciones de código



## 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