



## **Ingeniero en Software y tecnologías emergentes**

**Materia:** Programación Estructurada / Clave 36276

**Alumno:** Fernando Haro Calvo

**Matrícula:** 372106

**Maestro:** Pedro Núñez Yépiz

**Actividad No. 11**

**Tema - Unidad 1: FUNCIONES y METODOS DE ORDENACION Y  
BUSQUEDA ESTRUCTURAS Y LIBRERIAS (p2)**

**Ensenada Baja California a 31 de octubre del 2022**



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

### 1. INTRODUCCIÓN

Esta práctica es una combinación de los conocimientos adquiridos de la práctica 10 más nuestros conocimientos acerca de estructuras y generación de la Clave Única de Registro de Población (CURP) visto en la práctica 9 <sup>3</sup>/<sub>4</sub>. Consiste en generar manualmente y aleatoriamente información para rellenar un vector de datos de estudiantes, donde también podremos realizar métodos de búsqueda y ordenación, al igual que imprimir los registros en la consola en forma de tabla o de registro y poder generar un archivo de texto con estos mismos.

### 2. COMPETENCIA

El objetivo principal poner en práctica los conocimientos adquiridos en programación en C, especialmente abordará:

**Estructuras de dato 'struct':** Se utiliza para almacenar diferentes tipos de datos en una sola entidad.

**Algoritmos de búsqueda (secuencial y binaria):** se refiere a la capacidad de diseñar e implementar algoritmos que permitan encontrar un elemento específico en una colección de datos, como un arreglo.

**Validación de Datos:** El programa debe ser capaz de validar los datos de entrada, como el nombre, fecha de nacimiento, sexo y entidad federativa, para asegurarse de que sean correctos y cumplan con los requisitos oficiales.

**Modularidad y Organización:** La práctica debe demostrar una estructura modular y organizada del código, utilizando funciones y procedimientos para dividir el problema en tareas más pequeñas y manejables.

**Excepciones y Control de Errores:** El programa debe ser robusto, manejando excepciones y errores de manera adecuada. Debe ser capaz de informar al usuario si se ingresan datos incorrectos y nunca terminar inesperadamente.



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

### 3. FUNDAMENTOS

**Programación en C:** La práctica se basa en la programación en el lenguaje C, que es ampliamente utilizado en el desarrollo de sistemas y aplicaciones de software.

**Funciones:** Las funciones son bloques de código reutilizables que realizan tareas específicas. La práctica se enfoca en la creación y uso de funciones para organizar y modularizar el código.

**Structs:** Son una colección de variables llamadas "miembros" que pueden tener diferentes tipos de datos, como enteros, caracteres, flotantes u otros structs. Los structs se utilizan para representar una entidad o conjunto de datos que tiene múltiples atributos relacionados.

**Cadenas:** Son una secuencia de caracteres, es decir, una serie de caracteres que se organizan de manera consecutiva. Estos caracteres pueden ser letras, números, símbolos y espacios en blanco.

**Validación:** Es una parte esencial de la programación para garantizar que los datos sean correctos y seguros, y para evitar errores o comportamientos inesperados en el programa.

**Generación de archivos ".txt":** Utilizando los datos de un struct, generar un archivo de texto usando las funciones de la librería estándar de C.



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

### 4. PROCEDIMIENTO

Realiza el programa que contenga el siguiente menú

#### M E N Ú

- 1.- Agregar
  - a) manual (1)
  - b) Automático (100)
  - c) Regresar
- 2.- Eliminar Registro
- 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.

**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)



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

**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

### 5. RESULTADOS Y CONCLUSIONES

La actividad no fue tan compleja de realizar, debido a que fue principalmente una consolidación de conocimientos previos y nada realmente nuevo, se necesitaron implementar structs, incluso structs anidados para almacenar datos de estudiantes, implementamos algoritmos para generar datos aleatorios en los registros y agregarlos a nuestro vector de alumnos, implementamos métodos de ordenación y de búsqueda en el vector vistos en clase, al igual que un proceso para seleccionar automáticamente el algoritmo más óptimo para cada situación.

Logré comprender mejor los conceptos esenciales relacionados con cada tema. Mejoré el proceso para generar la CURP, ya que su eficiencia sería más importante al correr hasta 2000 veces el algoritmo, añadí funciones a mi librería que serán útiles.

### 6. ANEXOS

**Código y salidas:** HCF\_ACT11\_ANEXOS

**Archivo:** HCF\_ACT11\_932.c

**Librería:** alexandria.h



# 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