



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

**Materia:** Programación Estructurada / Clave 36276

**Alumno:** Fernando Haro Calvo

**Matrícula:** 372106

**Maestro:** Pedro Núñez Yépiz

**Actividad No. 13**

**Tema - Unidad 1: Archivos Binarios**

**Ensenada Baja California a 12 de noviembre 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 12, ahora utilizando archivos binarios además de archivos de texto. Consiste en generar aleatoriamente información para rellenar un vector de datos de personas, 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 datos, y un archivo binario donde se almacene el vector. Además, el programa deberá mantener registro de las personas en el vector, ya sean activos o inactivos.

### 2. COMPETENCIA

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

**Archivos Binarios:** El programa deberá poder leer y escribir archivos binarios, al igual que renombrarlos usando extensiones “.dll” y “.tmp” para almacenar información de un vector usando una estructura específica.

**Archivos de Texto:** El programa deberá aplicar las estrategias de escribir, anidar y leer archivos de texto en formato “.txt” para almacenar información.

**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.

**Generación de archivos binarios:** Utilizando los datos de un struct, generar un archivo binario usando las funciones de la librería estándar de C para mantener encriptados los datos.



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

### 4. PROCEDIMIENTO

#### MENÚ

1.- AGREGAR (AUTOM 100 REGISTROS)

2.- EDITAR REGISTRO

3.- ELIMINAR REGISTRO (lógico)

4.- BUSCAR

5.- ORDENAR

6- IMPRIMIR

7.- GENERAR ARCHIVO TEXTO

8.- VER ARCHIVO TEXTO

9.- CREAR ARCH BINARIO

10.- CARGAR ARCH BINARIO

11.- MOSTRAR ELIMINADOS

0.- SALIR

UTILIZAR UN ARREGLO DE 5000 REGISTROS, SE DEBERÁ UTILIZAR ESTRUCTURAS CON LOS DATOS BÁSICOS DE UN EMPLEADO. Preguntar nombre de archivo binario o de archivo texto

**1.- Agregar:** El programa deberá ser capaz de agregar 100 registros al vector de registros (**Generar automáticamente los datos**).

**2.- Editar Registro:** El programa deberá buscar una matrícula en el vector por medio del método de búsqueda más óptimo. **Mostrar los datos en forma de registro** Preguntar qué campo quiere Editar, actualizar los datos en el vector (solo a registros activos)

**3.- Eliminar Registro:** El programa deberá buscar una matrícula en el vector por medio del método de búsqueda más óptimo. Utilizar banderas para escoger el método más adecuado., imprimir el registro y preguntar si se quiere eliminar el registro.



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

- 4.- Buscar:** El programa deberá buscar una matrícula en el vector por medio del método de búsqueda más óptimo. Utilizar banderas para escoger el método más adecuado. **Mostrar los datos en forma de registro**
- 5.- Ordenar:** El programa deberá ordenar el vector por medio del método de ordenación más óptimo. Utilizar banderas para escoger el método más adecuado se ordenará por el **campo llave (matrícula)**
- 6.- Imprimir:** El programa deberá mostrar todos los registros del vector y como están en ese momento ordenado o desordenado. **(mostrar en forma de tabla)**
- 7.- Generar Archivo Texto:** El programa deberá preguntar al usuario el nombre del archivo, solo **nombre sin extensión**, el programa generará un archivo con el nombre proporcionado por el usuario con **extensión .txt** los datos que pondrá en el archivo de texto serán idénticos a los contenidos en el Vector de registros. (ordenado o desordenado). El programa podrá generar múltiples archivos para comprobar las salidas.
- 8.- Mostrar Archivo Texto:** El programa deberá preguntar al usuario el nombre del archivo, solo **nombre sin extensión**, el programa generará un archivo con el nombre proporcionado por el usuario con **extensión .txt** mostrar el archivo de texto tal y como se encuentra.
- 9.- Crear archivo binario:** El programa deberá crear un archivo binario con los datos del vector actualizados, sustituir el archivo base, realizar respaldo del archivo anterior y guardarlo con el mismo nombre, pero **extensión .tmp** (validar msges si el archivo no se puede crear por falta de registros en el vector)
- 10.- Cargar Archivo Binario:** El programa deberá cargar al vector los registros del archivo binario **(solo podrá cargarse una sola vez el archivo, el archivo binario se deberá llamar datos.dll y si no existe deberá indicar)**
- 11.- Mostrar Borrados:** El programa deberá mostrar del archivo binario solo los registros que se eliminaron (marcados con status 0) y que fueron marcados en su momento como registros eliminados.



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

### 5. RESULTADOS Y CONCLUSIONES

La actividad me sirvió para entender el funcionamiento de los archivos binarios, y poder ver sus beneficios cuando son comparados con los archivos de texto plano, hacer lectura, escritura y otros métodos como renombrar me ayudó a sentirme más cómodo con el manejo de estos archivos y en general no fue tan complicado realizar la actividad, debido a que se podía reutilizar una gran parte del código de archivos de texto. Cada vez puedo sentir más la utilidad de los programas que se desarrollan, en este caso siendo una mini base de datos para datos de empleados, y funcionando como un sistema coherente.

En resumen, la práctica me ayudó a lograr un manejo más eficiente de archivos de texto, archivos binarios, vectores, métodos de búsqueda y de ordenación, cumpliendo con las recomendaciones establecidas.

### 6. ANEXOS

**Código y salidas:** HCF\_ACT13\_ANEXOS

**Archivos:** HCF\_ACT13\_932.c

**Librería:** alexandria.h

### 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