



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

### Ingeniero en computación

### Ingeniero en Software y tecnologías emergentes

**Materia:** Programación Estructurada / Clave 36276

**Alumno:** BRAYAN IVAN PEREZ VENTURA

**Matrícula:** 372781

**Maestro:** Pedro Núñez Yépiz

**Actividad No. :** 12

**Tema - Unidad :** ARCHIVOS BINARIOS

Ensenada Baja California a 19 de Noviembre del 2023



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

### 1. INTRODUCCIÓN

Para la presente práctica, se crea un menú interactivo donde existirán varias opciones el cual, el usuario podrá navegar e interactuar generando archivos y cargando archivos binarios, de texto, creando archivos temporales y así mismo, cargandose y leyéndolos.

### 2. COMPETENCIA

Se evaluará la capacidad del estudiante para realizar los problemas presentados en el procedimiento, siguiendo las reglas de la programación estructurada y su correcta implementación en cada una de estas, así mismo, también la capacidad de las funciones creadas dentro de los archivos.

### 3. FUNDAMENTOS

TEORÍA TOMADA DEL MANUAL OFICIAL DE PRÁCTICAS

### 4. PROCEDIMIENTO



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

### ACTIVIDAD 13

REALICE EL SIGUIENTE PROGRAMA QUE CONTENGA UN MENÚ.

#### **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



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

### INSTRUCCIONES DEL MENU

- 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 que 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.
- 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 deba llamar datos.dll y si no existe deba 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.

## 5. RESULTADOS Y CONCLUSIONES



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

Para la presente práctica, podemos concluir que la implementación de archivos para codificar archivos binarios son muy importante durante la ejecución de un programa, siendo este una parte fundamental de cualquier código, pues, tiene la capacidad de guardar información sensible.

Así mismo, el saber cómo leerlos y generarlos es fundamental para programadores, así mismo, el conocer tipos de archivos.

```
736 void loadFileEmployee(char fileName[])
737 {
738     int i;
739     FILE *fa;
740     Twkr reg;
741     int temp = 0;
742     int tempNo = 0;
743     char firstLine[200];
744     fa = fopen(fileName, "r");
745     i = 0;
746     if (fa)
747     {
748         fgets(firstLine, sizeof(firstLine), fa);
749         printf("%-10s %-12s %-15s %-20s %-15s %-10s %-10s %-5s %-10s\n",
750             "No.",
751             "Matricula",
752             "Nombre",
753             "Ap. Paterno",
754             "Ao. Materno",
755             "Sexo",
756             "Posicion",
757             "Estado",
758             "Edad",
759             "Num. Cel");
760         while (!feof(fa))
761         {
762             temp = fscanf(fa, "%d %d %s %s %s %s %s %s %d 646%d", &tempNo,
763                 &reg.enrollment,
764                 reg.name,
765                 reg.lastName1,
766                 reg.lastName2,
767                 reg.sex,
768                 reg.JobPstion,
769                 reg.state,
770                 &reg.age,
771                 &reg.cellPhone);
772             if (temp == 10)
773             {
774                 displayListEmp(reg, i);
775                 i++;
776             }
777         }
778     }
779     else
780     {
781         printf("El archivo no existe\n");
782     }
783     fclose(fa);
784 }
785 }
```



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

### 6. ANEXOS

ANEXADO EN UN ARCHIVO PDF.

### 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