



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño



**Nombre del alumno:**

Luis Fernando Ochoa Angulo

**Matricula:**

372746

**Semestre:**

3ro- Computación

**Grupo:**

432

**Materia:**

Programación Estructurada

**Actividad:**

Actividad 13

**Nombre del Profesor:**

Pedro Nuñez Yepiz

**Fecha:** Jueves 23 de Noviembre del 2023



# 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:** Luis Fernando Ochoa Angulo

**Matrícula:** 372746

**Maestro:** Pedro Núñez Yépiz

**Actividad No. :** 13

**Tema - Unidad :** Archivos de texto y Archivos binarios

**Ensenada Baja California a 23 de Noviembre del 2023**



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

### 1. INTRODUCCIÓN

En el terreno de la programación en C, la gestión de datos a través de archivos se divide en dos formas principales: archivos de texto y archivos binarios. Los primeros son legibles para humanos, mientras que los segundos optimizan el espacio y preservan estructuras de datos eficientemente. Este informe aborda la manipulación de ambos en C, explorando la escritura, lectura y modificación de datos. Al entender estas dinámicas, se desentrañan las complejidades de cada formato, proporcionando una visión crucial para los desarrolladores al elegir entre archivos de texto y binarios según sus necesidades específicas. Bienvenidos a un viaje entre las dualidades de almacenamiento en C.

### 2. COMPETENCIA

El objetivo central de esta práctica es adquirir un conocimiento práctico y profundo sobre la manipulación de archivos en el lenguaje de programación C, centrándonos especialmente en dos formas clave: archivos de texto y archivos binarios.



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

### FUNDAMENTOS

#### Archivos de Texto:

**Legibilidad Humana:** Los archivos de texto contienen datos en formato legible para humanos. Cada dato se representa en caracteres, lo que facilita la inspección y edición manual.

**Interoperabilidad:** Dada su legibilidad, los archivos de texto son interoperables entre diferentes plataformas y aplicaciones. Son ideales para el intercambio de datos entre sistemas heterogéneos.

#### Archivos Binarios:

**Eficiencia de Almacenamiento:** Los archivos binarios almacenan datos de manera más compacta, utilizando representaciones binarias. Esto resulta en un uso más eficiente del espacio de almacenamiento, crucial para grandes conjuntos de datos.

**Preservación de Estructuras:** Los archivos binarios conservan la estructura de datos exacta, lo que los hace ideales para almacenar información compleja, como estructuras, matrices y registros, sin necesidad de conversiones.

#### Operaciones Comunes en Ambos Tipos de Archivos:

**Apertura y Cierre:** Ambos tipos de archivos requieren operaciones de apertura y cierre antes y después de su manipulación. Esto se realiza con funciones como `fopen()` y `fclose()`.

**Lectura y Escritura:** Para la manipulación de datos, se emplean funciones como `fread()` y `fwrite()` para archivos binarios, y `fgets()` y `fprintf()` para archivos de texto.

**Desplazamiento del Puntero de Archivo:** El puntero de archivo (`fseek()`) se utiliza para navegar a ubicaciones específicas dentro de un archivo, esencial para operaciones de lectura y escritura en posiciones particulares.

#### Consideraciones de Desempeño y Seguridad:

**Velocidad de Acceso:** Los archivos binarios son generalmente más eficientes en términos de velocidad de acceso, ya que no requieren interpretación de caracteres.

**Seguridad:** La manipulación de archivos debe llevarse a cabo con precaución para evitar vulnerabilidades, como desbordamientos de búfer. La validación adecuada y el manejo de errores son fundamentales.

#### Decisiones Estratégicas:

**Elección del Formato:** La selección entre archivos de texto y binarios depende de factores como la naturaleza de los datos, la eficiencia de almacenamiento requerida y la necesidad de legibilidad humana.

**Portabilidad y Extensibilidad:** La elección también puede depender de consideraciones de portabilidad y la posibilidad de futuras expansiones o modificaciones en la estructura de datos.



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

### 3. PROCEDIMIENTO

## 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

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

**Busqueda y Ordenacion por CAMPO LLAVE**

**nota:** usar librería propia con funciones

**nota2:** 100 % validado, Cuidar desbordamiento de vector

**nota3:** Campo llave matricula no repetido, archivos solo cargar 1 sola vez.

**nota4:** Usar el tipo Tkey para hacer mas practico el programa

 LFOA\_Act13\_PE\_432(anexo).pdf

### 4. RESULTADOS Y CONCLUSIONES

Esta práctica en archivos en C me brindó una comprensión sólida de las diferencias entre archivos de texto y binarios. Descubrí que los primeros son geniales para la legibilidad, mientras que los segundos destacan por su eficiencia.

Aprendí a escribir, leer y modificar datos utilizando funciones específicas en C, como `fread()` y `fprintf()`. Considerar la eficiencia y la seguridad es ahora una parte integral de mi enfoque al trabajar con archivos. En conclusión, esta práctica me equipó con las habilidades necesarias para tomar decisiones informadas sobre el formato de archivo adecuado para futuros proyectos en C. ¡Listo para enfrentar nuevos desafíos.



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

### 5. ANEXOS

[https://drive.google.com/file/d/17HHQB\\_\\_zvaVoHhllszJfHB7I2LuiFvj0/view?usp=sharing](https://drive.google.com/file/d/17HHQB__zvaVoHhllszJfHB7I2LuiFvj0/view?usp=sharing)

ENLACE Github:

<https://github.com/LuisFernando8a/Prog-Estructurada/tree/3cad69870ff969feaf9cf504b9ff77b398690870/Practica%2013>



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

### 6. 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

García Arreglos, M. E. (s/f). Arreglos: Vectores y Matrices. Utm.mx. Recuperado el 4 de octubre de 2023, de [https://www.utm.mx/~mgarcia/PE4\(Arreglos\).pdf](https://www.utm.mx/~mgarcia/PE4(Arreglos).pdf)

Tutorial C++. (s. f.). [http://www.omijal.org/pagina\\_c/strings.html](http://www.omijal.org/pagina_c/strings.html)

De Computadores, P., & Olave, T. A. (s/f). *Algoritmos de Búsqueda y Ordenamiento*. Utfsm.cl. Recuperado el 1 de noviembre de 2023, de <https://www.inf.utfsm.cl/~noell/IWI-131-p1/Tema8b.pdf>