



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: Danna Guadalupe Sandez Islas

Matrícula: 373080

Maestro: Pedro Núñez Yépiz

Actividad No. : 14

Tema - Unidad : Archivos binarios (indexados)

Ensenada Baja California a 28 de Noviembre del 2022



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

1. INTRODUCCIÓN

En este ejercicio, se emplearán archivos binarios de manera más eficiente en comparación con prácticas anteriores. La optimización se logrará al utilizar un arreglo índice para acceder de manera eficiente a la información más extensa almacenada en el archivo. El proceso implica la generación aleatoria de datos de empleados para poblar un archivo de información personal, brindando la posibilidad de realizar búsquedas y ordenaciones, así como imprimir registros en la consola en formato de tabla o registro. Asimismo, se podrá generar un archivo de texto con estos datos. Además, el programa debe llevar un seguimiento de las personas en el vector, ya sean activas o inactivas.

2. COMPETENCIA

El propósito principal es aplicar los conocimientos en programación en C, centrándose en los siguientes aspectos:

1. ****Archivos Binarios:**** El programa debe leer y escribir archivos binarios, permitiendo la manipulación de información mediante extensiones ".dat" y ".bak" y almacenando datos de un vector mediante una estructura específica.
2. ****Archivos de Texto:**** Se aplicarán estrategias para escribir, anidar y leer archivos de texto en formato ".txt" con el objetivo de almacenar información de manera efectiva.
3. ****Estructuras de Datos 'struct':**** La práctica empleará estructuras de datos 'struct' para almacenar distintos tipos de datos en una entidad única.
4. ****Algoritmos de Búsqueda (secuencial y binaria):**** Se abordará la capacidad de diseñar y aplicar algoritmos que faciliten la búsqueda de elementos específicos en colecciones de datos, como arreglos.
5. ****Validación de Datos:**** El programa verificará la validez de datos de entrada, tales como nombres, fechas de nacimiento, sexo y entidad federativa, asegurándose de que cumplan con los requisitos oficiales.
6. ****Modularidad y Organización:**** La práctica demostrará una estructura modular y organizada del código, haciendo uso de funciones y procedimientos para dividir el problema en tareas más manejables.
7. ****Excepciones y Control de Errores:**** El programa será robusto al manejar excepciones y errores de forma adecuada. Informará al usuario sobre datos incorrectos y evitará terminar inesperadamente.

3. FUNDAMENTOS



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Programación en C: En esta práctica, nos enfocamos en usar el lenguaje de programación C, que es muy común en crear sistemas y programas de software.

Funciones: Las funciones son como bloques de instrucciones que pueden ser usados una y otra vez para hacer tareas específicas. Aquí, nos concentramos en crear y usar funciones para organizar y hacer más fácil el código.

Structs: Son un conjunto de datos que tienen diferentes tipos de información, como números, letras, o incluso otros conjuntos de datos. Los structs son útiles para representar cosas con varios detalles relacionados.

Cadenas: Son básicamente secuencias de letras, números o símbolos que están organizados uno tras otro. Pueden ser usadas para representar palabras o frases.

Validación: Es algo muy importante en programación, ya que ayuda a asegurarnos de que la información que estamos usando sea correcta y segura, evitando errores en el programa.

Generación de archivos ".txt": Aquí, creamos archivos de texto usando información de un struct y las funciones básicas del lenguaje C.

Generación de archivos binarios: Similar al paso anterior, pero aquí estamos creando archivos binarios para mantener los datos más seguros, utilizando las funciones estándar de C.

4. PROCEDIMIENTO



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

ACTIVIDAD 14

Archivos Binarios

(archivos indexados)

MENÚ

- 1.- AGREGAR
- 2.- ELIMINAR
- 3.- BUSCAR
- 4.- ORDENAR
- 5.- IMPRIMIR REGISTROS ARCHIVO ORIGINAL
- 6.- IMPRIMIR REGISTROS ARCHIVO ORDENADO
- 7.- GENERAR ARCHIVO TEXTO
- 8.- EMPAQUETAR
- 0.- SALIR

INSTRUCCIONES: Programa que contenga el menú anterior, el programa utiliza un vector de índices de la siguiente estructura: [llave, índice] donde *el campo llave es noemplado.*

registros.dat es el archivo con los registros a cargar en el vector de índices **archivo binario sera proporcionado,**

CARGAR ARCHIVO : El programa deberá cargar al arrancar el programa, el archivo Binario generará el vector de índices (llave, indice) **sólo con registros válidos (el tamaño del vector debera ser 25% mas grande que el la cantidad de registros que contenga el archivo binario) utiliza un archivo externo para averiguar tamaño y retorne cantidad de registros.**

1.- Agregar :

El programa deberá ser capaz de agregar un registro al arreglo de índices y al final del archivo Binario. **(agregar forma automatica no repetido el campo llave)**

2.- Eliminar :

El programa deberá buscar una **noemplado** en el vector de índices por medio del método de búsqueda más óptimo.

La **función deberá retornar, el índice** donde se encuentra la matrícula en el archivo Binario, **utilizar banderas para escoger el método más adecuado.**

Una vez obtenido el índice moverse dentro del archivo binario (usar fseek) usando el índice del vector de índices.

Leer el registro en la posición correcta, preguntar si se quiere eliminar registro.

Cambiar el status del registro si la respuesta es afirmativa, volver a posición anterior y sobrescribir el registro.

3.- Buscar :

El programa deberá buscar un **noemplado** en el vector de índices por medio del método de búsqueda más óptimo.

La **función deberá retornar, el índice** donde se encuentra la matrícula en el archivo Binario, **utilizar banderas para escoger el método más adecuado.**

Una vez obtenido el índice moverse dentro del archivo binario (usar fseek) usando el índice del vector de índices.

Leer el registro en la posición correcta, y desplegar el registro.

4.- Ordenar :

El programa deberá ordenar el vector de índices por medio del método de ordenación más óptimo. Utilizar banderas para escoger el método más adecuado por el que se ordenará por el campo llave (**noemplado**) o no ordenarse si ya está ordenado. **(utilizar 3 metodos de ordenacion diferentes segun sea el caso que se necesite Justificar los metodos en el reporte)**



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

5. RESULTADOS Y CONCLUSIONES

La actividad me permitió mejorar mi habilidad en el manejo eficiente de archivos de texto y binarios, así como en la gestión de vectores, índices, y en particular, en los métodos de búsqueda y ordenación, cumpliendo con las pautas sugeridas.

6. ANEXOS

Archivos adjuntos.



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