



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: Villalobos Ensaldo Luis Daniel

Matrícula: 368617

Maestro: Pedro Núñez Yépiz

Actividad No. : 11

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

Ensenada Baja California a 1 de 05 del 2024



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

NOTA: guardar el archivo de la sig forma
INICIALES (numero práctica)_PE
ejemplo: Francisco Antonio Perez Lopez
PLFA_RP08_PE.pdf

1. INTRODUCCIÓN

El archivo de texto es un tipo de archivo que almacena datos en forma de texto legible para humanos. En estos archivos, los datos se representan utilizando caracteres ASCII o Unicode, y cada caracter representa un símbolo, letra, número o espacio en blanco.

2. COMPETENCIA

Que el alumno logre el uso de estructuras anidadas para su modificacion y uso de funciones segun se necesite, al igual que la capacidad de representar datos en archivos de texto

3. FUNDAMENTOS

W (write): Abre un archivo para escritura. Si el archivo ya existe, su contenido se sobrescribe. Si el archivo no existe, se crea uno nuevo. Si el archivo existe, su contenido anterior se pierde.

R (read): Abre un archivo para lectura. El archivo debe existir; de lo contrario, la apertura fallará. Se puede leer contenido del archivo, pero no se puede modificar.

A (append): Abre un archivo para agregar datos al final del archivo. Si el archivo no existe, se crea uno nuevo. Si el archivo ya existe, los datos se agregan al final del archivo sin sobrescribir el contenido existente.

W+: Abre un archivo tanto para lectura como para escritura. Si el archivo no existe, se crea uno nuevo. Si el archivo existe, su contenido se sobrescribe. Se puede leer y escribir contenido en el archivo.

R+: Abre un archivo tanto para lectura como para escritura. El archivo debe existir; de lo contrario, la apertura fallará. Se puede leer y escribir contenido en el archivo. Si el archivo no existe, la apertura fallará.



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

4. PROCEDIMIENTO



PRÁCTICA 11

Realiza el programa que contenga el siguiente menú

MENÚ

- 1.- Cargar
- 2.- Eliminar
- 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.



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

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)

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.

5. RESULTADOS Y CONCLUSIONES

Creo que de lo mas interesante de esta actividad fue el uso del curp como funcion, ya que tuve que rehacer algunas partes del mismo, en si la practica no es tan diferente de la anterior, incluso reutilice la mayoría del codigo, solo cambiando algunas cosas como los registros al imprimir y similares, por parte de los metodos de busqueda y ordenacion ya los habia hecho anteriormente, y el archivo fue facil de generar por que es igual que si lo imprimiera en pantalla

6. ANEXOS



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
Talum Generar reg(void)
{
    Talum reg;

    Tnombre copia;

    reg.status = 1;

    char nombres[12][20] = {"JUAN", "LUIS", "DANIEL", "SAUL", "JACOB",
"ANGEL", "ALAN", "YAHIR", "GABRIEL", "ISRAEL", "WALTER"};

    char mujeres[12][20] = {"ANGELA", "SOFIA", "VIVIANA", "ANA", "MARIA",
"PAOLA", "PAULA", "SONIA", "VANESSA", "HANNA", "LOLA"};

    char appat[12][20] = {"SOTO", "VILLALOBOS", "VICTORIO", "PEREZ",
"PINEDA", "HERNANDEZ", "LEAL", "ESPINOZA", "IBregisA", "DIAZ", "DOMINGUEZ",
"MARTINEZ"};

    char apmat[12][20] = {"WAYNE", "CRUZ", "CHAVEZ", "CARMINE", "RUIZ",
"ORTIZ", "TORRES", "OROZCO", "GARCIA", "PAREDES", "ACOSTA", "CASAS"};

    int sexo, estado;

    // fecha

    reg.fecha = Fecha();

    // nombres y apellidos

    copystr(reg.nombre.appat, appat[rand() % 12]);

    copystr(reg.nombre.apmat, apmat[rand() % 12]);

    sexo = (rand() % 2) + 1;

    if (sexo == 2)

    copystr(reg.sexo, "HOMBRE");

    copystr(reg.nombre.nombre, nombres[rand() % 11]);
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
else
    copystr(reg.sexo, "MUJER");
    copystr(reg.nombre.nombre, mujeres[rand() % 11]);

    reg.matricula = (rand() % 99999) + 300000;
    estado = (rand() % 33) + 1;
    copia = reg.nombre;

    Curp(reg.curp, copia.nombre, copia.appat, copia.apmat, sexo, estado,
reg.fecha.ano, reg.fecha.mes, reg.fecha.dia);

    reg.estado[0] = reg.curp[11];
    reg.estado[1] = reg.curp[12];
    reg.estado[2] = '\\0';

    return reg;
}
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

7. REFERENCIAS



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Diseño de algoritmos y su codificación en lenguaje C

Corona, M.A. y Ancona, M.A. (2011)..

España: McGraw-Hill.

ISBN: 9786071505712

Ficheros en C++ — Fundamentos de Programación en C++. (n.d.).

https://www2.eii.uva.es/fund_inf/cpp/temas/10_ficheros/ficheros_cpp.html

Programación estructurada a fondo: implementación de algoritmos en C

:Pearson Educación.Sznajdleder, P. A. (2017)..

Buenos Aires,Argentina: Alfaomega

Algoritmos de ordenación con ejemplos en C++. (2024, March 22). SW Hosting.

<https://www.swhosting.com/es/comunidad/manual/algoritmos-de-ordenacion-con-ejemplos-en-c>

Como programar en C/C++

H.M. Deitel/ P.J. Deitel

Segunda edición

Editorial: Prentice Hall.

ISBN:9688804711

¿Como abrir un archivo txt en C++! - En-c. (n.d.).

<https://www.codigazo.com/en-c/como-abrir-archivos-txt-en-c>

Programación en C.Metodología, estructura de datos y objetos

Joyanes, L. y Zahonero, I. (2001)..

España:McGraw-Hill.



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

ISBN: 8448130138