

Título:

Base de Datos

Objetivo:

Tener una base de guardado de una gran variedad de documentos y con su clasificación de estos dependiendo de hora, fecha y/o Tema

Alcance de su proyecto:

Esto en el futuro se puede utilizar para las compañías, departamentos de gobierno (Secretarías) o uso para cualquier persona para que estén programados en días y horas que avisaran al usuario cuando van a alcanzar su límite de entrega si es en caso de las compañías(cronometro o alarma), además con una implementación antihackeo. Pero esto se agregara a futuro y dependiendo de lo aprendido durante mi carrera.

Introducción:

Durante lo visto en la materia de Estructuras y Datos 1 hemos desarrollado u observado la forma de almacenamiento de datos que han utilizado compañías como Facebook, Windows y en lo que me interesa los videojuegos. En este caso me intereso mucho el hacer este proyecto.

Al principio al platicar con amigos sobre lo que sería nuestros proyectos este empezó como una pokedex (enciclopedia de Pokemones) ya que en la aplicación llama Pokemon Go no se tiene una descripción y/o evoluciones de estas criaturas a como los tiene los juegos así nombrados, en este caso seria los primeros 150.

Pero este cambio al platicar con mi madre que trabaja en la Policía Federal actualmente Guardia Nacional en donde me estaba diciendo que la manera de ordenar documentos o su base de datos es por medio de la aplicación de Excel en donde está clasificado por fecha, asunto y documentos. En esta forma de uso para un departamento de gobierno o compañía es muy obsoleta para esta ya en un caso de emergencia seria de una forma muy lenta y obsoleta a lo que nuestra tecnología nos puede permitir.

Al final de lo platicado y reflexionado he decido hacer un programa en donde se podrá manejar por medio de carpetas o apartados en donde se clasificara de lo visto o lo que desee el usuario en donde dependiendo de sus usos.

Desarrollo:

Lenguaje a utilizar: C y/o C++

Nota: En un futuro podrá cambiarse dependiendo lo que se va a ver a futuro

En este proyecto se utilizara varios temas vistos en nuestras clases como los arreglos y los apuntadores pero en lo visto en lo que se va a realizar se ocupara precisamente: datos abstractos y las pilas, esto nos servirá para la clasificación de datos o documentos y más cuando será cuando se agregue o se tenga que ver próximamente. Las pilas nos pueden ayudar en mucho ya que aplicaciones como WhatsApp y Messenger lo utilizan al mandar un mensaje reciente el chat utilizado se pone en primer lugar

- **Definición:**

- ❖ Arreglo:

Es un conjunto de datos finito y del mismo tipo. En realidad funciona como cualquier variable cualquiera, excepto que en lugar de almacenar un solo valor, guarda algunos valores. Pueden ser unidimensionales o multidimensionales. Los arreglos nos permiten hacer un conjunto de operaciones para manipular los datos guardados en ellos, estas operaciones son: ordenar, buscar, insertar, eliminar, modificar entre otras.

- ❖ Apuntador:

Es una variable que contiene una dirección de memoria, la cual corresponderá a un dato o a una variable que contiene el dato. Cada variable que se utiliza en una aplicación ocupa una o varias posiciones de memoria. Estas posiciones de memoria se accedan por medio de una dirección

- ❖ Datos abstractos:

Un tipo de dato abstracto (TDA) es un conjunto de datos u objetos creado de manera personalizada por un programador para un fin específico. Un TDA es una abstracción que permite modelar las características de un elemento en particular.

Un tipo de dato abstracto se puede manipular de forma similar a los tipos de datos que están predefinidos dentro del lenguaje de programación, encapsulando más información, según se requiera.

La implementación de un tipo de dato abstracto depende directamente del lenguaje de programación que se utilice. En lenguaje C los tipos de dato abstracto se crean mediante las estructuras (struct).

- Nota: En el desarrollo del proyecto se cambiara dependiendo del tiempo y de la creación de este. Pronto se empezara el código fuente para su desarrollo

Código:

Avance uno: Se logró poner registro y mostrar información

```
#include <stdio.h>
```

```
/*
```

```
    Programa para el almacenamiento de cualquier tipo de informacion
```

```
*/
```

```
struct datos{
```

```
    char asunto[30];
```

```
    char fecha[30];
```

```
    char info[30];
```

```
};
```

```
typedef struct datos DATO;
```

```
void ingresarDato();
```

```
void mostrarDato();
```

```
//void buscarDato();
```

```
int main(){
```

```
    short opcion = 0;
```

```
    while(1){
```

```
        printf("\n\t*** BASE DE DATOS ***\n");
```

```
        printf("1) INSERTAR REGISTRO :\n");
```

```
        printf("2) MOSTRAR REGISTROS :\n");
```

```
        //printf("3) BUSCAR REGISTRO :\n");
```

```
        //printf("4) ELIMINAR REGISTRO :\n");
```

```
        printf("3) SALIR.\n"); //5)
```

```
        scanf("%d", &opcion);
```

```
        switch(opcion){
```

```
            case 1:
```

```
                ingresarDato();
```

```
                break;
```

```
            case 2:
```

```
                mostrarDato();
```

```
                break;
```

```
            //case 3:
```

```
                //buscarDato();
```

```
                //break;
```

```
            //case 4:
```

```

        //eliminarDato();

        //break;

    case 3: //case 5:

        return 0;

    default:

        printf("Opción no válida.\n");

    }

}

return 0;

}

void ingresarDato(){

    FILE *fich;

    DATO e;

    if((fich = fopen("Registro","ab"))==NULL){

        printf("\nFichero no existe! ");

    }else{

        printf("\n Ingrese Asunto o Nombre: "); fflush(stdin);

        gets(e.asunto);

        printf(" Ingrese Fecha[Separar los dias, meses y año con /] o numero: "); fflush(stdin);

        scanf("%s",&e.fecha);

```

```
        printf(" Ingrese Informacion (poner el nombre del archivo y/o imagen  
con []): "); fflush(stdin);
```

```
        gets(e.info);
```

```
        fwrite(&e,sizeof(e),1,fich); // Escribe todo lo que tenemos en la  
estructura en el fichero . . . . .
```

```
        fclose(fich); //Cierra el fichero para que no quede abierto . . . . .
```

```
    }
```

```
}
```

```
void mostrarDato(){
```

```
    FILE *fich;
```

```
    DATO e;
```

```
    if((fich = fopen("Registro","rb")) == NULL){
```

```
        printf("\n Fichero no existe ");
```

```
    }else{
```

```
        fread(&e,sizeof(e),1,fich); // Lee los registros del Fichero Posicion por  
posicion . . . .
```

```
        while(! feof(fich)){ // Recorriendo el Fichero . . . .
```

```
            printf("\n _____");
```

```
            printf("\n Nombre/Asunto.....: %s",e.asunto);
```

```
            printf("\n Fecha/Numero.....: %s",e.fecha);
```

```
            printf("\n Informacion.....: %s",e.info);
```

```
            printf("\n _____");
```

```
            fread(&e,sizeof(e),1,fich);
```

```
        }  
    }  
    fclose(fich);  
    getch();  
}  
  
//void buscarDato();  
  
//void eliminarDato();
```

Resultados:

Conclusiones:

Referencias:

- [https://es.wikipedia.org/wiki/Memoria_din%C3%A1mica_\(programaci%C3%B3n\)](https://es.wikipedia.org/wiki/Memoria_din%C3%A1mica_(programaci%C3%B3n))
- [https://www.ecured.cu/Pila_\(Estructura_de_datos\)](https://www.ecured.cu/Pila_(Estructura_de_datos))
- <http://www.utn.edu.ec/reduca/programacion/arreglos/definiciones1.html>
- [http://www.utm.mx/~mgarcia/PE7\(Apuntadores\).pdf](http://www.utm.mx/~mgarcia/PE7(Apuntadores).pdf)
- Guía práctica de estudio 03. Manual de prácticas del Laboratorio de Estructuras de datos y algoritmos I. Tipo de dato abstracto. M.C. Edgar E. García Cano e Ing. Jorge A. Solano Gálvez. UNAM. 2017. Pags. 25 - 32