



CENTRO DE CIENCIAS BÁSICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
LENGUAJES DE COMPUTACIÓN I
1° "E"

MANUAL TÉCNICO: ICI1_2018-PROYECTO FINAL-CESARLUISJOEL

Docente: Rosalinda Avendaño López

Alumnos: Cardona Rivera José Luis
Espinoza Sánchez Joel Alejandro
Pérez Jaime Julio César

Fecha de Entrega: Aguascalientes, Ags., 10 de diciembre de 2018

Índice

Bienvenido-----	2
Análisis Algorítmico -----	2
Las Variables -----	2
Funciones-----	2
Arreglos -----	3
Variables -----	3
El Programa Fuente-----	4
Bibliografía -----	12

Bienvenido

La siguiente Guía Técnica especifica la cuestión interna del funcionamiento del programa IC11_2018-ProyectoFinal-CesarLuisJoel. Este manual tiene como propósito llevarlo a través de los esquemas de desarrollo del programa.

Análisis Algorítmico

El objetivo de este programa era realizar una base de datos de diez registros de trabajadores, con nombre, clave y sueldo aleatorios hechos por la máquina. A su vez, que el usuario tuviera la facilidad de modificar la base de datos, eliminando, añadiendo o modificando datos. Por lo que se pensó primeramente como un menú en el cual el usuario pudiera moverse por él según la acción que quería realizar.

Las entradas del usuario serían principalmente las modificaciones dentro de la base de datos, ya sea para eliminar un registro, añadir uno o modificar uno ya existente.

El proceso del programa depende mucho de las acciones a realizar, sin embargo, la propiedad que engloba a todas las características del programa es el de un manejo óptimo de información al momento de remover, añadir o modificar información. La máquina hará estas tres acciones como procesos internos para poder capturar y guardar la información requerida.

La salida principal del programa es el despliegue de la base de datos de manera visual, la cual está dada en una de las funciones del programa.

Las Variables

Funciones

Todas las funciones manejadas en el programa son de tipo “void”.

La función basededatos permite que, al ejecutarse el menú, la base de datos se despliegue en la pantalla.

La función buscarxnombre permite hacer búsquedas por nombre, opción dada en el menú.

La función buscarxclave permite hacer búsquedas por clave, opción dada en el menú.

La función nuevoregistro permite añadir un nuevo registro, a partir del menú.

La función eliminarxnombre permite eliminar registros por nombre, ejecutándose desde el menú.

La función eliminarxclave permite eliminar registros por clave, ejecutándose desde el menú.

La función modificarxnombre permite modificar registros por nombre, ejecutándose desde el menú.

La función modificarxclave permite modificar registros por clave, ejecutándose desde el menú.

La función `modificarxsueldo` permite modificar registros por sueldo, ejecutándose desde el menú.

Arreglos

El arreglo `nombre` es la base de datos que se llenará con información y los arreglos que se puntualicen a continuación.

El arreglo `nombres` es un arreglo que aleatoriamente otorgará un nombre a la base de datos.

El arreglo `apellido1` es un arreglo que aleatoriamente otorgará un apellido a la base de datos.

El arreglo `apellido2` es un arreglo que aleatoriamente otorgará otro apellido a la base de datos.

El arreglo `clave` es un arreglo de la base de datos que contendrá la información de las claves de trabajadores.

El arreglo `sueldo` es un arreglo de la base de datos con información de los sueldos de los trabajadores

Variables

`menu`: Es la variable que permite que el usuario se mueva por el programa y sus funciones.

`repetir`: Es parte del segundo menú que conforma al programa.

`flag2`: Es parte del segundo menú que conforma al programa de forma booleana.

`cl`: Es el número que se otorga de clave a un trabajador al añadirlo.

`n`: Bandera de los nombres.

`a1`: Bandera de los nombres.

`a2`: Bandera de los nombres.

`decimal`: Parte decimal del sueldo del trabajador.

`suma`: Suma entre la parte decimal y la parte entera del sueldo.

`entero`: Parte entera del sueldo de un trabajador.

`i`: Contador.

El Programa Fuente

```
#include <bits/stdc++.h>
using namespace std;//Necesario para los stings

//Base de datos
void basededatos(string nombre [], int clave[], float sueldo[]);

//Buscar por nombre
void buscarxnombre(string nombre [], int clave[], float sueldo[]);

//Buscar por clave
void buscarxclave(string nombre [], int clave[], float sueldo[]);

//Ordenar alfabeticamente
//void ordenarAlfa(string nombre [], int clave[], float sueldo[]); MEN ESTO
ESTA EN FACE BETA NO LE HAGAS CASO

//Insertar nuevo
void nuevoregistro(string nombre [], int clave[], float sueldo[]);

//Eliminar por nombre
void eliminarxnombre(string nombre [], int clave[], float sueldo[]);

//Eliminar por clave
void eliminarxclave(string nombre [], int clave[], float sueldo[]);

//Modificar por nombre
void modificarcxnombre(string nombre [], int clave[], float sueldo[]);

//Modificar por clave
void modificarcxclave(string nombre [], int clave[], float sueldo[]);

//Modificar por sueldo
void modificarcxsueldo(string nombre [], int clave[], float sueldo[]);

int main(){

    //Generales
    setlocale(LC_ALL,"spanish");//Acentos y esas cosas
    srand((unsigned int)time(NULL)); //Semilla para los random
    int menu; //Menu
    int repetir; //Para el segundo menu
    bool flag2; //Para el segundo menu

    //Nombres aleatorios
    string nombre[10]; //Base de datos
    string
    nombres[10]={"César","Joul","Jose","Jimena","Gloria","Rosy","Angel","Pedr
o","Rick","Summer"};
    string
    apellido1[10]={"Perez","Sanchez","Cardona","Padilla","Lopez","Jaime","Lir
a","Alvarez","Becerra","Parker"};
```

```

    string
    apellido2[10]={"Guerra","Avila","Vaca","Montoya","Santos","Ponce","Martin
    ez","Alonso","Cuello","Flores"};
    int n=0, a1=0, a2=0;

    //Clave de trabajador
    int clave[10]; //Base de datos
    int cl;

    //Sueldo del trabajador
    float sueldo[10]; //Base de datos
    float decimal=0, suma=0;
    int entero=0;
    //Llenado de base de datos

    for(int i=0;i<10;i++)
    {
        //Nombre

        n=1+rand()%(9); //Elige un nomre aleatorio
        a1=1+rand()%(9); //Elige un apellido aleatorio
        a2=1+rand()%(9); //Elige un apellido aleatorio
        nombre[i]=nombres[n]+"    "+apellido1[a1]+"    "+apellido2[a2];
    //Composicion de todo el nombre

        //Clave

        cl=1000+rand()%(9000); //Elige un numero aleatorio de 4 cifras
        clave[i]=cl;

        //Sueldo

        entero = ((int)rand())/3;
        decimal = 1000+((float)rand())/(9000);
        suma=entero+decimal;
        sueldo[i]=suma;
    }
    bool flag = true;
    do
    {
        system("color 0f");
        system("cls");
        printf("    \t                Menu\n");
        printf("\n");
        printf("1- \tMostrar base de datos\n");
        printf("2- \tBuscar por nombre\n");
        printf("3- \tBuscar por clave\n");
        printf("4- \tOrdenar alfabeticamente\n");
        printf("5- \tInsertar nuevo\n");
        printf("6- \tEliminar por nombre\n");
        printf("7- \tEliminar por clave\n");
        printf("8- \tModificar por nombre\n");
        printf("9- \tModificar por clave\n");
        printf("10- \tModificar por sueldo\n");
    }

```

```

printf("11- \tSalir\n");
printf("\n");
printf("¿Que quieres hacer?\n");
scanf("%d",&menu);
system("cls");
switch(menu){
    case 1:{
        basededatos(nombre,clave,sueldo);
        break;
    }
    case 2:{
        buscarxnombre(nombre,clave,sueldo);
        break;
    }
    case 3:{
        buscarxclave(nombre,clave,sueldo);
        break;
    }
    /*case 4:{
        ordenarAlfa(nombre,clave,sueldo);
        break;
    }*/
    case 5:{
        nuevoregistro(nombre,clave,sueldo);
        break;
    }
    case 6:{
        eliminarxnombre(nombre,clave,sueldo);
        break;
    }
    case 7:{
        eliminarxclave(nombre,clave,sueldo);
        break;
    }
    case 8:{
        modificarxnombre(nombre,clave,sueldo);
        break;
    }
    case 9:{
        modificarxclave(nombre,clave,sueldo);
        break;
    }
    case 10:{
        modificarxsueldo(nombre,clave,sueldo);
        break;
    }
    case 11:{
        flag = false;
        break;
    }
    default:{
        printf("Opcion invalida\n");
        break;
    }
}

```

```

    }
    if (flag){
        printf("\n");
        printf("¿Quieres regresar al menu? |1-Si| !=1-No|
");
        scanf("%d",&repetir);
        if (repetir==1){
            flag2=true;
        }
        else{
            flag2= false;
        }
        if(!flag2){
            break;
        }
    }
    else
        break;
}while(true);
}

```

```

void basededatos(string nombre [], int clave[], float sueldo[]){
    printf("Nombre del trabajador\tID\tSalario\n");
    for (int i = 0;i<10;i++){
        printf("%s\t%d\t%f\n",nombre[i].c_str(),clave[i],sueldo[i]);
        //c_str() es necesario para poder imprimir el string, lo
        concierte en char para poder imprimirlo de otra manera causa errores
    }
}

```

```

void buscarxnombre(string nombre [], int clave[], float sueldo[]){
    char buscador[50];
    bool flag = true;
    printf("Introduce el nombre que deseas buscar(Respeta mayusculas,
minisculas, espacios y signos de puntuacion):\n");
    fflush(stdin);//Limpia de buffer, añadido para evitar errores
    gets(buscador);
    string flagnombre = buscador;
    for(int i=0;i<10;i++)
    {
        if(!flagnombre.compare(nombre[i])){
            printf("Se encontro un registro \n");
            //Sintaxis para comparar strings con funcion "Compare"

            printf("%s\t%d\t%f\n",nombre[i].c_str(),clave[i],sueldo[i]);
            flag = false;
        }
    }
    printf("\n");
    if(flag){
        printf("No se encontro ningun registro");
    }
}

```



```

    }

}

void buscarxclave(string nombre [], int clave[], float sueldo[]){
    int claveb;
    bool flag = true;
    printf("Introduce la clave de trabajador que deseas buscar:\n");
    scanf("%d",&claveb);
    for(int i=0;i<10;i++)
    {
        if(claveb==clave[i]){
            printf("Se encontro un registro \n");

            printf("%s\t%d\t%f\n",nombre[i].c_str(),clave[i],sueldo[i]);
            flag = false;
        }
    }
    printf("\n");
    if(flag){
        printf("No se encontro ningun registro");
    }
}

/*
void ordenarAlfa(string nombre [], int clave[], float sueldo[]){
    std::set<std::string> sortedItems;

    for (int i = 0;i<10;i++){
        sortedItems.insert(nombre[i]);
    }
    std::for_each(sortedItems.begin(), sortedItems.end(), &print);
    printf("\n");
}
*/

void nuevoregistro(string nombre [], int clave[], float sueldo[]){
    bool flag= false;
    int posicion;
    char x[50];
    for(int i=0;i<10;i++)
    {
        if(!nombre[i].compare("")){
            flag = true;
            posicion = i;
        }
    }
    if(flag){
        printf("Dame el nombre a guardar (Nombre Apellido Apellido):
");
        fflush(stdin);
        gets(x);
        nombre[posicion]=x;
    }
}

```

```

        printf("Dame la clave a guardar (4 numeros): ");
        fflush(stdin);
        scanf("%d",&clave[posicion]);
        printf("Dame el sueldo a guardar (4 enteros y 4 decimales, SE
DEBE USAR LA COMA PARA MARCAR EL PUNTO DECIMAL): ");
        scanf("%f",&sueldo[posicion]);
        printf("\nSe guardó el nuevo registro correctamente");
    }
    else
    {
        printf("La base de datos esta llena");
    }
    printf("\n");
}

void eliminarxnombre(string nombre [], int clave[], float sueldo[]){
    char nombree[50];
    bool flag = true;
    printf("Introduce el nombre que deseas eliminar(Respeta mayusculas,
minisculas, espacios y signos de puntuacion):\n");
    fflush(stdin);
    gets(nombree);
    string nombree2 = nombree;
    for(int i=0;i<10;i++)
    {
        if(!nombree2.compare(nombre[i])){
            nombre[i] = "";
            clave[i] = 0;
            sueldo[i] = 0;
            flag = false;
            printf("El registro se elimino correctamente\n");
        }
    }
    if(flag){
        printf("No se encontro ningun registro");
    }
}

void eliminarxclave(string nombre [], int clave[], float sueldo[]){
    int clavee;
    bool flag = true;
    printf("Introduce la clave de trabajador que deseas eliminar:\n");
    scanf("%d",&clavee);
    for(int i=0;i<10;i++)
    {
        if(clavee==clave[i])
        {
            nombre[i] = "";
            clave[i] = 0;
            sueldo[i] = 0;
            flag = false;
            printf("El registro se elimino correctamente\n");
        }
    }
}

```

```

        if(flag){
            printf("No se encontro ningun registro");
        }
    }

void modificarxnombre(string nombre [], int clave[], float sueldo[]){
    char nombrem[50], x[50];
    bool flag = true;
    printf("Introduce el nombre que deseas modificar(Respeta mayusculas,
minisculas, espacios y signos de puntuacion):\n");
    fflush(stdin);
    gets(nombrem);
    string nombrem2 = nombrem;
    for(int i=0;i<10;i++)
    {
        if(!nombrem2.compare(nombre[i])){
            printf("Dame el nombre a guardar (Nombre Apellido
Apellido): ");
            fflush(stdin);
            gets(x);
            nombre[i]=x;
            printf("Dame la clave a guardar (4 numeros): ");
            fflush(stdin);
            scanf("%d",&clave[i]);
            printf("Dame el sueldo a guardar (4 enteros y 4 decimales,
SE DEBE USAR LA COMA PARA MARCAR EL PUNTO DECIMAL): ");
            scanf("%f",&sueldo[i]);
            flag = false;
            printf("\nEl registro de modifiko correctamente");
        }
    }
    printf("\n");
    if(flag){
        printf("No se encontro ningun registro");
    }
}

void modificarxclave(string nombre [], int clave[], float sueldo[]){
    int clavee;
    bool flag = true;
    char x[50];
    printf("Introduce la clave de trabajador que deseas modificar:\n");
    scanf("%d",&clavee);
    for(int i=0;i<10;i++)
    {
        if(clavee==clave[i])
        {
            printf("Dame el nombre a guardar (Nombre Apellido
Apellido): ");
            fflush(stdin);
            gets(x);
            nombre[i]=x;
            printf("Dame la clave a guardar (4 numeros): ");

```

```

        fflush(stdin);
        scanf("%d",&clave[i]);
        printf("Dame el sueldo a guardar (4 enteros y 4 decimales,
SE DEBE USAR LA COMA PARA MARCAR EL PUNTO DECIMAL): ");
        scanf("%f",&sueldo[i]);
        flag = false;
        printf("\nEl registro de modifiko correctamente");
    }
}
printf("\n");
if(flag){
    printf("No se encontro ningun registro");
}
}

void modificarxsueldo(string nombre [], int clave[], float sueldo[]){
    float sueldom;
    bool flag = true;
    char x[50];
    printf("Introduce el sueldo que deseas modificar:\n");
    scanf("%f",&sueldom);
    for(int i = 0;i<10;i++){
        if(sueldom==sueldo[i]){
            printf("Dame el nombre a guardar (Nombre Apellido
Apellido): ");
            fflush(stdin);
            gets(x);
            nombre[i]=x;
            printf("Dame la clave a guardar (4 numeros): ");
            fflush(stdin);
            scanf("%d",&clave[i]);
            printf("Dame el sueldo a guardar (4 enteros y 4 decimales,
SE DEBE USAR LA COMA PARA MARCAR EL PUNTO DECIMAL): ");
            scanf("%f",&sueldo[i]);
            flag = false;
            printf("\nEl registro de modifiko correctamente");
        }
    }
    printf("\n");
    if(flag){
        printf("No se encontro ningun registro");
    }
}
}

```

Bibliografía

- Noguera, O. F. J., & Riera, T. D. (2013). *Programación*. Retrieved from <https://ebookcentral.proquest.com>
- Jugaru, M. M. (2014). *Introducción a la programación*. Retrieved from <https://ebookcentral.proquest.com>
- Cedano, O. M. A., Rubio, G. J. A., & Vega, G. A. C. (2014). *Fundamentos de computación para ingenieros*. Retrieved from <https://ebookcentral.proquest.com>