

Instituto Universitario de Tecnología



GUIA TEORICA N° 4 PROGRAMACION I APUNTES: REGISTROS.

No temas, porque yo estoy contigo; no
Desmayes, porque yo soy tu DIOS que
Te esfuerzo; siempre te ayudaré, siempre
Te sustentaré con la diestra de mi justicia»
Isaías 41:10

Profesor : DELY M. GIL A.

VALENCIA, Marzo,2010

REGISTROS (Estructuras)

Un *registro* es una estructura que consiste de un número fijo de componentes llamados *miembros*, *lógicamente relacionados*. Los *miembros* pueden ser de diferentes tipos de datos y deben tener un *identificador de miembro*.

DECLARACIÓN

```
struct [identificador]{  
    tipo de dato miembro1  identificador del miembro1;  
    tipo de dato miembro2  identificador del miembro2;  
    .  
    .  
}[variables estructuradas] ;
```

DECLARACIÓN

Si no se definen las variables estructuras se pueden definir posteriormente con el siguiente formato:

**struct identificador variables estructuradas;
identificador variables estructuradas;**

```
struct registro{  
    long int ci;  
    char nombre[20];  
    float promedio;  
};
```

```
struct registro alumno;  
//registro alumno;
```

```
struct registro{  
    long int ci;  
    char nombre[20];  
    float promedio;  
}alumno;
```

REFERENCIANDO MIEMBROS DE UNA ESTRUCTURA

Para referenciar los miembros de una estructura es necesario hacer uso de el operador punto (.) ó flecha (->) cuando es un apuntador

Ejemplo:

```
scanf("%s", alumno.ci);           //Lectura
alumno->promedio = 15;           //Asignación
printf("%s\n", alumno.nombre);   //escritura
```

REGISTROS ANIDADOS

Un registro con uno o más campos que son del tipo registro se llaman Registro Jerárquicos o Registros Anidados.

```
struct fecha {
    unsigned int dia;
    unsigned int mes;
    unsigned int agno;
};
struct registro{
    long int ci;
    char nombre[20];
    struct fecha fecha_nac;
    float promedio;
};
struct registro alumno;
```

OPERACIONES

Asignación:

```
alumno.ci           =10717654';
strcpy(alumno.nombre, "Juan Gil");
alumno.fecha_nac.dia   =27;
alumno.fecha_nac.mes   =12;
alumno.fecha_nac.agno  =1967;
alumno.promedio       =18 ;
```

Lectura:

```
printf("\nCédula: ");
scanf("%ld",&alumno.ci);
printf("\nNombre: ");
fflush(stdin);
gets(alumno.nombre);
printf("\nFecha de Nacimiento: ");
printf("\nDia: ");
scanf("%i",&alumno[i].fecha_nac.dia);
printf("\nMes: ");
scanf("%i",&alumno[i].fecha_nac.mes);
printf("\nAño: ");
scanf("%i",&alumno[i].fecha_nac.agno);
printf("\nInsertar las notas: ");
```

Escritura:

```
.....
write('Nombre: ');
writeln(Trabajador.Nombre);
write('Edad:');
Write(Trabajador.Edad)
write('Domicilio: ');
```

REFERENCIANDO MIEMBROS DE UNA ESTRUCTURA

Para referenciar los miembros de una estructura es necesario hacer uso de el operador punto (.) ó flecha (->) cuando es un apuntador

Ejemplo:

```
scanf("%s", alumno.ci);           //Lectura
alumno->promedio = 15;           //Asignación
printf("%s\n", alumno.nombre);   //escritura
```

REGISTROS ANIDADOS

Un registro con uno o más campos que son del tipo registro se llaman Registro Jerárquicos o Registros Anidados.

```
struct fecha {
    int dia;
    int mes;
    int agno;
};
struct registro{
    long int ci;
    char nombre[20];
    struct fecha fecha_nac;
    float promedio;
};
struct registro alumno;
```

ARREGLOS DE REGISTROS

Los registros se agrupan en conjuntos conocidos como arreglos de registros.

```
#define N 4
struct fecha {
    int dia;
    int mes;
    int agno;
};
struct registro{
    long int ci;
    char nombre[20];
    struct fecha fecha_nac;
    float promedio;
};

struct registro alumno[N];
```

CAMPOS DE REGISTROS TIPO ARREGLOS

El tipo del cualquier miembro de un registro puede ser la estructura definida por el Usuario, en este caso un VECTOR

```
struct registro{
    long int ci;
    char nombre[20];
    struct fecha fecha_nac;
    float notas[3];
    float promedio;
};
```

```
//archivos de cabeceras
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
//constantes
#define N 2
#define T 2
//TDA
struct fecha {
    int dia;
    int mes;
    int agno;
};
struct registro{
    long int ci;
    char nombre[20];
    struct fecha fecha_ing;
    float notas[T];
    float promedio;
};
//Variables globales

registro alumno[N];

//prototipos
void llenar(registro []);
void mostrar(registro [],float []);
void calcularpromedio(registro alumno[N]);
void promedioParcial(registro [],float []);
```

```
int main(){
    float B[T];
    system("CLS");
    llenar(alumno);
    calcularpromedio(alumno);
    promedioParcial(alumno,B);
    mostrar(alumno,B);
    printf("\n\n\n\n\n\t\t");
    system("PAUSE");
    return 0;
}

void llenar(registro a[N]){
    for(int i=0;i<N;i++){
        printf("\nInsertar Cédula: ");
        scanf("%ld",&a[i].ci);
        printf("\nInsertar Nombre: ");    fflush(stdin);
        gets(a[i].nombre);
        printf("\nInsertar Fecha de Nacimiento: ");
        printf("\nDia: ");    scanf("%i",&a[i].fecha_ing.dia);
        printf("\nMes: ");    scanf("%i",&a[i].fecha_ing.mes);
        printf("\nAño: ");    scanf("%i",&a[i].fecha_ing.agno);
        for(int j=0;j<T;j++){
            printf("\nInsertar las notas: ");
            scanf("%f",&a[i].notas[j]);
        }
    }
}
```

```
void mostrar(registro a[N],float b[]){
    system("CLS");

    printf("\n\tCedula\tNombre\tP1\tP2\tPromedio\tFecha\n");
    for (int i=0;i<N;i++){
        printf("\t%i\t%s",a[i].ci,a[i].nombre);
        for(int j=0;j<T;j++){
            printf("\t%1.1f",a[i].notas[j]);
        }
        printf("\t%1.1f",a[i].promedio);

        printf("\t\t%i/%i/%i",a[i].fecha_ing.dia,a[i].fecha_ing.m
es,a[i].fecha_ing.agno);
        printf("\n\n");
    }
    printf("\t");
    for(int j=0;j<T;j++)
        printf("\t%1.1f",b[j]);
}

void calcularpromedio(registro a[N]){
    float ac;
    for (int i=0;i<N;i++){
        ac=0;
        for(int j=0;j<T;j++)
            ac += a[i].notas[j];
        a[i].promedio =ac/N;
    }
}
```

```
void promedioParcial(registro a[N],float b[]){
    float ac;
    for (int j=0;j<T;j++){
        ac=0;
        for(int i=0;i<N;i++)
            ac += a[i].notas[j];
        b[j] =ac/T;
    }
}
```

Tarea

1. Valide las entradas ; notas,fecha y nombre
2. Mostrar el nombre del alumno que tiene mayor promedio.
3. Mostrar el nombre del alumno que tuvo el parcial con menor nota.
4. Indique la cantidad de aprobados y reprobados según la definitiva y su porcentaje respectivo.
5. Calcule el promedio del curso.

```
//archivos de cabeceras
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
//constantes
#define N 2
#define TRUE 0
#define FALSE 1
//TDA
struct registro{
    long int ci;
    char nombre[20];
    char sexo;
    unsigned int edad;
};
//Variables globale

registro personas[N];

//prototipos
void incluir();
void eliminar();
void modificar();
void consultar();
void menu();

unsigned full=FALSE;
int ultimo = -1;
char opc;
```

```
int main(){

    system("CLS");
    menu();
    printf("\n\n\n\n\n\t\t");
    system("PAUSE");
    return 0;
}
void menu(){
    do{
        system("CLS");
        printf("\nMenu");           printf("\n1.Insertar");
        printf("\n2.Consultar");     printf("\n3.Modificar");
        printf("\n4.Eliminar");      printf("\n5.Salir");
        opc= getchar();
        switch(opc){
            case '1' : (full=TRUE)? printf("No se puede incluir"):
                            incluir();
                            break;
            case '2' : consultar();           break;
            case '3' : modificar();          break;
            case '4' : eliminar();           break;
        }
    }while (opc !='5');
}
```

```
void incluir(){
    registro temp;
    system("CLS");
    printf("Ingreso Cedula: ");
    scanf("%ld",&temp.ci);
    printf("Ingreso Nombre: ");
    fflush(stdin);
    scanf("%s",&temp.nombre);
    printf("Ingreso sexo: ");
    fflush(stdin);
    temp.sexo=getchar();
    fflush(stdin);
    printf("Ingreso edad: ");
    scanf("%i",&temp.edad);
    ultimo++;
    personas[ultimo]=temp;
    printf("Persona %s",personas[ultimo].nombre);
    system("Pause");
    if (ultimo==N)
        full = TRUE;
}
```

```
void consultar(){
    long int cedula, enc=FALSE;
    int i=0;
    int posi=-1;

    system("CLS");
    printf("Ingreso Cedula: ");
    scanf("%ld",&cedula);
    while((i<=ultimo)&&(enc==FALSE)){
        if (cedula==personas[i].ci){
            enc = TRUE;
            posi = i;
        }else
            i++;
    }
    if (enc==TRUE){
        printf("Nombre %s",personas[posi].nombre);
        printf("Sexo %c",personas[posi].sexo);
        printf("Edad %i",personas[posi].edad);
    }else
        printf("La persona no existe");
    system("PAUSE");
}
```

**/*DESARROLLE EL SUBPROGRAMA BUSQUEDA
DEFINA LAS CABECERAS DE LOS PROTOTIPOS CON
PASE DE PARÀMETROS*/**

```
void eliminar(){
    long int cedula, enc=FALSE;
    int i=0, posi=-1;

    system("CLS");
    printf("Ingrese Cedula: ");
    scanf("%ld",&cedula);
    while((i<=ultimo)&&(enc==FALSE)){
        if (cedula==personas[i].ci){
            enc = TRUE;
            posi = i;
        }else
            i++;
    }
    if (enc==TRUE){
        for(int i=posi;i<=ultimo-1;i++)
            personas[i] = personas[i+1];
        ultimo--;
    }else
        printf("La persona no existe");

    system("PAUSE");
}

void modificar(){
    long int cedula, enc=FALSE;
    int i=0, posi=-1;
    char op;

    system("CLS");
    printf("Ingrese Cedula: ");    scanf("%ld",&cedula);
```

```
while((i<=ultimo)&&(enc==FALSE)){
    if (cedula==personas[i].ci){
        enc = TRUE;        posi = i;
    }else
        i++;
    }
    if (enc==TRUE){
        do{
            printf("\n1.Nombre %s",personas[posi].nombre);
            printf("\n2.Sexo %c",personas[posi].sexo);
            printf("\n3.Edad %i",personas[posi].edad);
            printf("\n4.Salir");
            printf("\nIndique el campo a modificar o 4 para
salir");
            fflush(stdin);op= getchar();printf("valor de la op
%c",op);
            switch(op){
                case '1' : printf("\nNombre modificado");
                    fflush(stdin);
                    scanf("%s",&personas[posi].nombre);
                    break;
                case '2' : printf("\nSexo modificado");
                    fflush(stdin);
                    scanf("%c",&personas[posi].sexo);;
                    break;
                case '3' : printf("\nEdad modificado");
                    scanf("%d",&personas[posi].edad);;
                    break;
            }
        }while(op!='4');
    }else
        printf("La persona no existe");system("PAUSE");
}
```