



CONCEPTO DE DATOS ESTRUCTURADOS

REGISTROS SIMPLES Y REGISTROS GERARQUIZADOS

OBJETIVOS:

- ✚ Que él alumno Comprenda, seleccione y organice en forma estructurada a los datos.
- ✚ Que él alumno Comprenda el concepto de estructura de datos y describa las aplicaciones adecuadas de los registros.
- ✚ Que él alumno Represente adecuadamente y resuelva situaciones problemáticas que involucren estructuras de tipo registros.

TEMAS:

- ✚ Registros Estáticos.
- ✚ Registros Jerarquizados.
- ✚ Array de Registros.
- ✚ Registros de Array.
- ✚ Definición, lectura e impresión.
- ✚ Representación en lenguaje C.

PROBLEMAS RESUELTOS

EJEMPLO N° 01

OBJETIVO: Definir un **REGISTRO ESTÁTICO** o estructura de datos estáticos, mostrando como se realiza la carga, lectura y visualización de un registro en pantalla.

ENUNCIADO:

Dada la siguiente configuración, correspondiente a los **productos de un negocio**:

- ✚ Código (entero),
- ✚ Descripción (Carácter) y
- ✚ Precio (Real).

SE PIDE:

- 1- Defina el registro apropiado para los productos del Negocio.
- 2- Ingrese un producto y
- 3- Muestre los datos ingresados en pantalla.

CODIFICACIÓN:

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

```
struct registro
{
    int codigo;
    char descri[30];
    float precio;
};
```

Definición de una Estructura (registro) formada por 3 miembros de datos simples.

```
main ()
{
```

```
    registro reg;           //Declara una variable (reg) del tipo registro que es una estructura.
    /*----- Ingreso de Datos al Registro -----*/
    /*Los datos se almacenan directamente en el registro -----*/
    printf("Ingrese Los Datos Del Producto\n");
    printf("===== \n\n");
    printf("Codigo: ");
    scanf("%d", &reg.codigo); //Los datos son almacenado directamente en el miembro codigo.
    printf("\nDescripcion: ");
    _flushall();             //Limpia el buffers, esto es necesario cuando hay un scanf y luego un gets.
    gets(reg.descri);        //Los datos son almacenado directamente en el miembro descri.
    printf("\nPrecio: ");
    scanf("%f", &reg.precio); //Los datos son almacenado directamente en el miembro precio.
    /*----- Muestra los datos ingresado en el registro -----*/
    printf("\n\n Los Datos Ingresados Son\n");
    printf("===== \n\n");
    printf("\tCodigo: %d", reg.codigo);
```

```
printf("\n\n\t Descripción: ");
puts(reg.descri);
printf("\n\t Precio: %.2f\n\n\n", reg.precio);
system("PAUSE");
}
```

EJEMPLO N° 02

OBJETIVO: Definir un **REGISTRO JERARQUÍA** o Jerarquizados, Transferir un registro a una función utilizando Parámetros por referencia. La transferencia e registro a una función se utiliza de la misma forma en registros Estáticos.

ENUNCIADO:

Dada los siguientes datos correspondiente a los artículos deportivos de un negocio: código, descripción, precio y fecha de adquisición (compra) (que a su vez, fecha, es un registro que posee los miembros simples: día, mes y año).

SE PIDE:

- 1- Defina de manera apropiada el registro para los datos y requerimientos descriptos.
- 2- Permita la registración de una compra (Ingreso al registro).
- 3- Muestre los datos ingresados en pantalla (Muestra los datos que tiene el registro).

CODIFICACIÓN:

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

```
struct fecha
{
    int dia;
    int mes;
    int anio;
};
```

Estructura (fecha) con tres miembros simples.

```
struct registro
{
    int codigo;
    char descri[30];
    float precio;
    fecha fec;
};
```

La Estructura Jerarquizada (*Registro*) contiene en uno de sus Miembros (*fec*), que es otra estructura (*Estruct fecha*).

//Prototipos

```
void agregarArticulo(struct registro &art); //Función con parámetro por referencia (una estructura).
void mostrarArticulo(struct registro &art); //Función con parámetro por referencia (una estructura).
```

main ()

```
{
    registro reg; //Define o declara la variable (reg) como un tipo de dato estructura de la estructura registro).
    agregarArticulo(reg); //Llamada a la función para agregar los datos al registro, transfiere como como parámetro la variable reg.
    mostrarArticulo(reg); //Llamada a la función para mostrar los datos almacenados (guardados) en la variable reg.
    printf("\n\n ..... Fin del programa ..... \n");
    system("PAUSE");
}
```

/* Función recibe como parámetro por referencia el registro con los datos de un Artículo para ser visualizado en pantalla.*/

void mostrarArticulo(struct registro &art1)

```
{
    printf("\n\n LOS DATOS INGRESADOS SON\n\n");
    printf("=====\n");
    printf("Codigo: %d", art.codigo);
    printf("\nDescripcion: ");
    puts(art.descri);
    printf("Precio: %f", art.precio);
    printf("\n\n Fecha de Vto: %d - %d - %d\n\n", art.fec.dia, art.fec.mes, art.fec.anio);
    system("PAUSE");
}
```

Pantalla
Salida de datos

```
LOS DATOS INGRESADOS SON
=====
Codigo: 1254
Descripcion: Caracoles del mar negro
Precio: 1254.540039
Fecha de Vto: 12 - 2 - 2014
Presione una tecla para continuar . .
```

/* Función recibe como parámetro por referencia el registro para ser llenado con los datos de un Artículo.*/

void agregarArticulo(struct registro &art1)

```
{
    system("CLS");
    printf("Ingrese los datos del Artículo\n");
    printf("=====\n\n");
    printf("Codigo: ");
    scanf("%d", &art.codigo);
    printf("Descripcion: ");
    _flushall();
    gets(art.descri);
    printf("Precio: ");
    scanf("%f", &art.precio);
    printf("\n\n Fecha de Vto: \n");
    printf("\tDia: ");
    scanf("%d", &art.fec.dia);
    printf("\tMes: ");
    scanf("%d", &art.fec.mes);
    printf("\tAño: ");
    scanf("%d", &art.fec.anio);
    printf("\n\n Fin de la Carga.....\n");
    system("PAUSE");
}
```

Pantalla
Entrada de Datos

```
Ingrese los datos del Artículo
=====
Codigo: 1254
Descripcion: Caracoles del mar negro
Precio: 1254.54
Fecha de Vto:
Dia: 12
Mes: 02
Año: 2014
```

EJEMPLO Nº 03

OBJETIVO: Definir un **ARREGLOS DE REGISTRO** o **Arreglos estáticos**. Ingreso de datos a un vector y mostrar los datos ingresados en el vector.

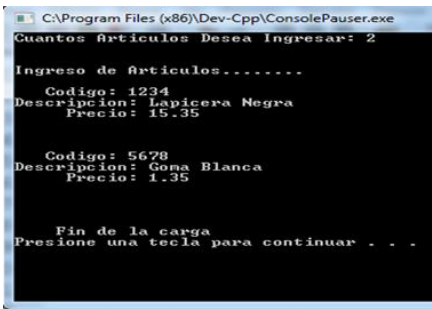
ENUNCIADO:

Un negocio dedicado a la venta de artículos variados, los cuales poseen los siguientes datos: código (Entero), descripción (carácter) y precio (float).

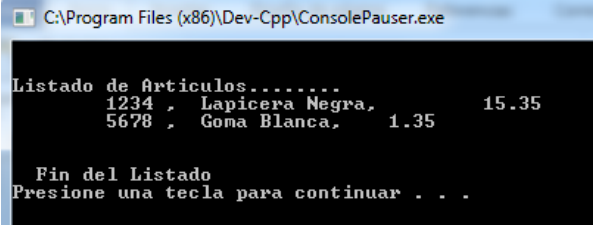
SE PIDE:

- 1- Defina la estructura del registro apropiada a los datos descriptos.
- 2- Ingresar **N** Artículos y **almacene los mismos en un vector**. Use una **función sin tipo**.
- 3- Muestre los datos ingresados en el vector por pantalla. Use una **función sin tipo**.

CODIFICACIÓN:

| Bloque Principal de la Codificación | Pantalla |
|---|---|
| <pre>#include <stdio.h> #include <stdlib.h> #include <string.h> /*----- Estructura o Registro Lógico.-----*/ struct registro { int codigo; char descri[30]; float precio; }; /*----- Prototipos a utilizar. -----*/ void cargarDatosVector(registro A[100], int N); void mostrarDatosVector(registro A[100], int N); /*----- Bloque Principal. -----*/ main() { registro Array[100]; //Define el vector de tipo registro de 100 elementos como máximo. int N; printf("Cuantos Articulos Desea Ingresar: "); scanf("%d", &N); // cargarDatosVector(Array, N); mostrarDatosVector(Array, N); }</pre> |  |

| Función Sin Tipo Para El Ingreso De Los Datos | Pantalla |
|---|----------|
| <pre> /* ----- Nombre : cargarDatosVector(...) Descripcion : Permite almacenar (Guardar) los datos en el arreglo. Tipo de Función: Sin Tipo. Retorna : ----- Parametros : Recibe dos parametros, 1- El arreglo de tipo registro. 2- Cantidad de elementos a ingresar. -----*/ void cargarDatosVector(registro A[100],int N) { printf("\n\nIngreso de Articulos.....\n"); for(int i=0; i<N; i++) { printf("\n Codigo: "); scanf("%d", &A[i].codigo); //Almacena el codigo, usa scanf(...). printf("Descripcion: "); _flushall(); gets(A[i].descri); //Almacena la descripcion, usa gets(...). printf(" Precio: "); scanf("%f", &A[i].precio); //Almacena el precio, usa scanf(...). printf("\n\n"); _flushall(); //Limpia el buffers de entrada. } printf("\n\n Fin de la carga \n"); system("PAUSE"); system("CLS"); } </pre> | |

| Función Sin Tipo Que Muestra En Pantalla Los Datos Registrados En El Vector | Pantalla |
|---|---|
| <pre> /* ----- Nombre : mostrarDatosVector(...) Descripcion : Muestra los datos que se ingresaron al vector. Tipo de Función: Sin Tipo. Retorna : ----- Parametros : Recibe dos parametros: 1- El vector de tipo registro. 2- Cantidad de elementos a ingresar. -----*/ void mostrarDatosVector(registro A[100], int N) { printf("\n\nListado de Articulos.....\n"); for(int i=0; i<N; i++) { /* Muestra los 3 datos almacenado en la misma posicion del vector */ printf("\t%d , \t%s, \t%.2f\n", A[i].codigo, A[i].descri, A[i].precio); } printf("\n\n Fin del Listado \n"); system("PAUSE"); system("CLS"); } </pre> |  |

EJEMPLO N° 04

OBJETIVO: Que el alumno aprenda a trabajar con **REGISTRO DE ARREGLO**.

ENUNCIADO:

Ingrese los datos de un alumno en una estructura como la siguiente: legajo, notas (donde notas es un vector de orden 3).

SE PIDE:

- 1- Defina la estructura apropiada a los requerimientos.
- 2- Ingrese la información de un alumno (código y sus 3 notas).
- 3- Muestre los datos ingresados con su promedio.

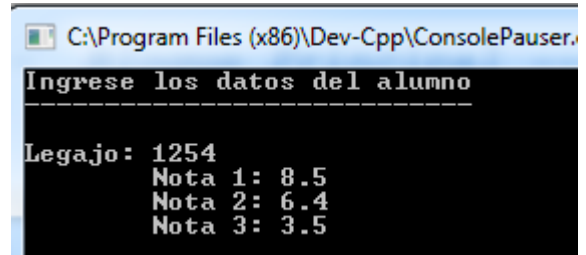
CODIFICACIÓN:

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

/*----- Define la Estructura -----*/
struct registro
{
    int legajo;
    float notas[3]; //Vector de orden 3 dentro de la estructura.
};

/*----- Bloque Principal -----*/
main ()
{
    registro reg;
    int j;
    float proNotas, sumNotas=0;

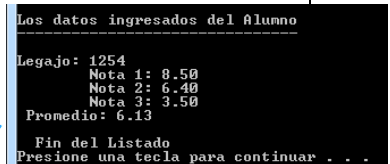
    /*----- Carga del Registro -----*/
    printf("Ingrese los datos del alumno\n");
    printf("-----\n\n");
    printf("Legajo: ");
    scanf("%d",&reg.legajo);
    for (j=0; j<3; j++) //For para cargar las notas al vector.
    {
        printf("\tNota %d: ", j+1);
        scanf("%f",&reg.notas[j]);
    }
}
```



```
/*----- Listado del Registro -----*/
printf("\n\nLos datos ingresados del Alumno\n");
printf("-----\n\n");
printf("Legajo: %d", reg.legajo);
for (j=0; j<3; j++) //For para listar las notas del vector.
{
    printf("\n\tNota %d: %.2f", j+1, reg.notas[j] );
    sumNotas = sumNotas + reg.notas[j]; // Suma las notas del alumno.
}

/*----- Calculo del promedio -----*/
proNotas =float (sumNotas/3 ); //Calcula el Promedio.
printf("\n Promedio: %.2f \n ",proNotas); //Muestra el promedio.

printf("\n Fin del Listado \n" );
system("PAUSE");
}
```



| DATOS DEL GRUPO DE ALUMNOS | | |
|----------------------------|-------------------|------------------------------|
| División: | Profesor: | Fecha de Entrega __/__/__ |
| | Auxiliar o JTP: | |
| Legajo o DNI | Apellido y Nombre | |
| | | |

Problemas Propuestos

Dados los siguientes enunciados con manejo de registro, encuentre una solución para cada uno de los problemas propuestos.

Ejemplo de: **Array Registro Jerarquizado**

- 1) ✎ Un negocio desea manejar a través de un sistema informático, los datos de sus **N** empleados para agilizar sus consultas. La información que maneja es la siguiente: Número de Legajo, Apellido y Nombre, Cargo, Sueldo, Fecha de Ingreso (**anidar**), Fecha de Nacimiento (**anidar**), ambas fechas son dd/mm/aa.

SE PIDE: Definir la estructura Adecuada para los datos. o
puedes utilizar la estructura de la derecha.

- Ingreso de datos.
- Calcular el Total de los sueldos.
- Calcular el promedio de los sueldos.
- Saber cuántos empleados ingresaron en un determinado año.
- Determinar el promedio de las edades.

La estructura es la siguiente:

| | |
|-------------------|--------------|
| Numero de Legajo | -> Entero. |
| Apellido y nombre | -> Char 40. |
| Cargo | -> Char(15). |
| Sueldo | -> Float. |
| Fecha Ingreso | -> Struct. |
| Fecha Nacimiento | -> Struct. |

La estructura Fecha:

| | |
|-----|------------|
| día | -> Entero. |
| mes | -> Entero. |
| Año | -> entero. |

Ejemplo de: **Registro Estáticos**

- 2) 🧑 Leer una lista de **N** alumnos y sus correspondientes notas de final de curso que se encuentran en un Registro. **Los datos son: Apellido/s y nombre/s del alumno y su nota.**

SE PIDE:

- Usando una función, realizar la carga de los datos.
- Usando una función con tipo, calcular el total de alumnos aprobados. En la función main mostrar los alumnos aprobados y desaprobados.
- Usando una función calcular el porcentaje que hubo de alumnos aprobados y desaprobados.

Ejemplo de: **Arreglo de Registro**


- 3) 🧑 Un hotel recibe huéspedes de distintos puntos del país, y desea almacenar los datos de sus **N huéspedes** en un vector. Cada elemento del vector deberá ser un registro con los siguientes estructura de campos:

- 🚩 Apellido y nombre del huésped.
- 🚩 Ciudad de origen(**0**:Buenos Aires, **1**:Rosario, **2**:Tucumán)
- 🚩 Cantidad de día que se va a hospedar.
- 🚩 Importe que debe abonar por los días que reservo.
- 🚩 Tipo de servicio (**0**:Pension Completa, **1**:Media Pensión, **2**:Solo Desayuno, **3**:Solo habitación)

SE PIDE:

- Cargar el vector con los datos de los huéspedes.
- Determinar el porcentaje de huésped de cada ciudad.
- Determinar el importe total obtenido, por tipo de servicio de pensión completa

Ejemplo de: **Registro de Arreglo**

- 4) . Un médico necesita tener almacenados ciertos datos de sus pacientes. Dichos datos son los siguientes:


- | | |
|--|---|
| -Apellido y nombre. | -Dirección |
| -Teléfono. | -Fecha de la última consulta |
| -Si tiene o No obra social (1=Tiene, 0=No tiene) | -Si tiene alergia o NO. (1=Tiene, 0=No tiene) |

SE PIDE:

Los datos deberán ser almacenados en un vector de registro, y se desea un programa que ofrezca un menú con las siguientes opciones:

- Introducir un registro en el arreglo. (*Alta*)
- Visualizar un registro completo dado su Apellido y nombre. (*Consulta*)
- Dado el nombre de un paciente mostrar la fecha de su última consulta.
- Dado el nombre de un paciente eliminarlo del vector. Usar un arreglo auxiliar (*Baja física*)
- Dar apellido y nombre de todos los pacientes que tengan alergia. (*Consulta*)

Ejemplo de: **Registro Jerarquizados**


- 5)  Se almacena en un Vector de registros, con fines estadísticos, la descripción de algunas personas. Cada registro deberá contener Apellido y Nombre, sexo, Fecha Nacimiento, color de ojos, Provincia de Residencia.

SE PIDE:

- Almacenar los **N** registros en el arreglo, usando una función de carga.
- Usando una función, Ordenar los datos del vector por apellido y nombre.
- Visualizar los datos ordenados, usando una función de visualización.
- Determinar la cantidad de personas de ojos color "Marrones" de sexo femenino en la provincia de Tucumán.

Ejercicios Adicionales (Ejercicio 06)

Este ejercicio solo debe ser realizado, por los alumnos que no presentaron el TP en la fecha estipulada. El ejercicio deberá ser presentado junto al resto de los ejercicios resuelto.

- 6)  En el Autódromo provincial de Termas de Río Hondo- Santiago del Estero se desarrollará una competencia automovilística internacional organizada por la **FIA World Touring Car Championship (WTCC)**. Se desea registrar la matriculación de los **N vehículos** para dicho evento y almacenar la información en un vector.

Los datos que se registrarán son:

Código: Generado de manera consecutiva según el Nro. de inscripción.

Conductor: Nombre del conductor del vehículo.

Fecha de Inscripción: La fecha en que se registró para la carrera. (DD-MM-AA)

País: País al que pertenece el vehículo.

Fecha de la Carrera: Tener en cuenta que no pueden correr más de 10 autos por fechas.

Fabricante: Pueden ser Chevrolet, Lada, Seat, Honda, Ford, Alfa Romeo, Peugeot, otros.






SE PIDE que por medio de un menú se pueda realizar lo siguiente:

- Registrar la inscripción de automóviles.
- Listar los miembros inscriptos para las carreras.
- Ordenar y listar por fabricante.
- Mostrar cuantos inscriptos hay por cada país.

Ejercicio Complementario

Este ejercicio es para que el alumno practique y amplíe su conocimiento y destreza.

En una biblioteca se registran los libros existentes, informándose:

-  **CODIGO DE LIBRO** (Numero de 1 a 100, caso contrario informar error y volver a ingresar)
-  **CANTIDAD DE EJEMPLARES.**
-  **AUTOR.**
-  **TITULO.**
-  **CANTIDAD DE VECES PRESTADO**

Luego de registradas las existencias se visualiza un menú que permite al bibliotecario elegir alguna de las siguientes opciones:

OPCION 1: PRESTAMOS DE LIBROS

El bibliotecario ingresar:

CODIGO DE LIBRO (De 1 a 100, caso contrario informar error y volver a ingresar)

El programa debe mostrar el autor y el título y sólo se presta el libro si la cantidad de ejemplares es mayor a 1. Si el usuario confirma el préstamo restar 1 a la cantidad de ejemplares y sumar uno a la cantidad de veces de prestado.

Si la cantidad de ejemplares es 1, mostrar la leyenda *"Solo queda el ejemplar de lectura en sala"* y no se registra el préstamo.

OPCION 2: DEVOLUCIÓN DE LIBROS

El bibliotecario ingresa:

CODIGO DE LIBRO (De 1 a 100, caso contrario informar error y volver a ingresar)

El programa debe mostrar el autor y el título y si el usuario confirma, registrar la devolución sumando 1 a la cantidad de ejemplares.

OPCION 3: FIN

Significa que no se sigue trabajando con el menú de opciones y que antes de finalizar debe informar por medio de un listado ordenado en forma decreciente por cantidad de veces que se prestó un libro lo siguiente:

| TITULO | AUTOR | VECES PRESTADO |
|--------|-------|----------------|
| XXXXX | XXXXX | XXX |