LICENCIATURA EN INFORMATICA

PROGRAMADOR UNIVERSITARIO

UNIVERSIDAD NACIONAL DE TUCUMÁN
facet
FACULTAD DE CIENCIAS EXACTAS Y TECNOLOGÍA

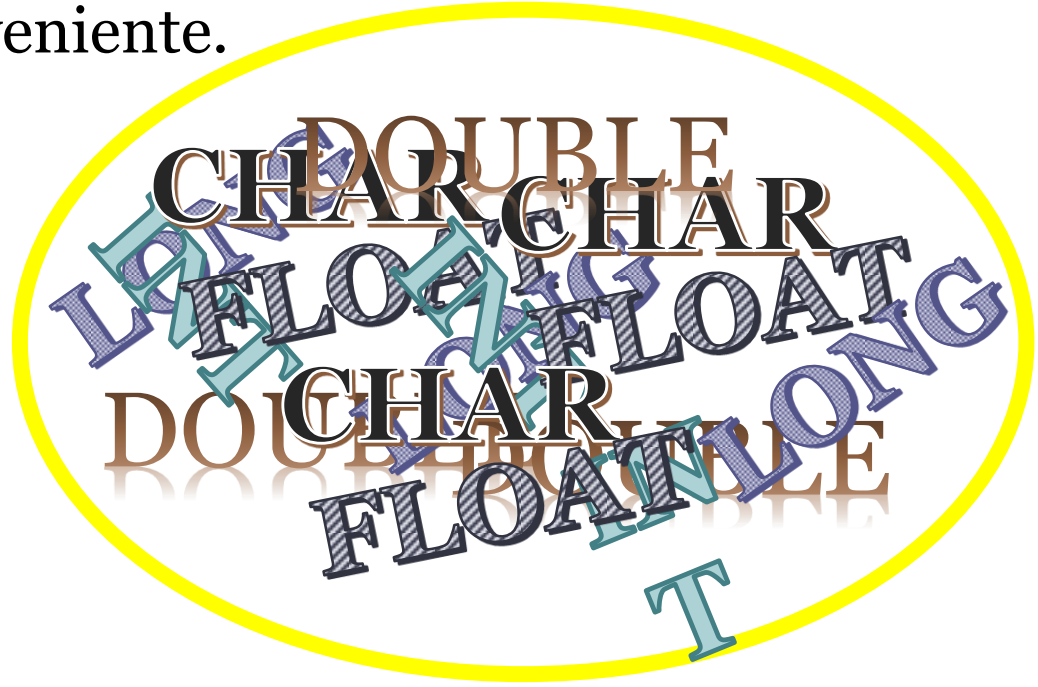
# PROGRAMACIÓN

## Unidad 7: Tipos de datos derivados: Estructuras

Lic. Mariela A. Velázquez

# ¿Qué es una Estructura?

Una estructura es una colección de uno o mas tipos de variables, agrupadas bajo un mismo nombre para un manejo conveniente.



# Ejemplo de Estructuras

Las estructuras permiten asociar información, o lo que es lo mismo, permiten que un grupo de variables relacionadas sean tratadas como una unidad en lugar de entidades separadas.

Ejemplo:

Una idea sencilla de una estructura son los datos de una persona:

- Nombre
- Apellido
- Edad
- Fecha de nacimiento
- Peso
- Estado Civil

| Datos | Tipo |
| --- | --- |
| Apellido | Char ape[MAX] |
| Nombre | Char nom[MAX] |
| Edad | int edad |
| Peso | Float peso |
| Sexo | char sexo (F o M) |

# Características generales de las estructuras

- Tienen un nombre
- Elementos de distintos tipos llamados campos o miembros
- Acceso a las componentes vía índice o puntero
- Acceso a las componentes individualmente ó como un todo (vía índice o puntero)

# ¿Como crear una Estructura?

Se habrá de usar la Palabra Reservada : **struct**

```
var1;
                tipo2
var2;
                  :
:
                tipoi
vari;
                  :
:
                tipon
```

- Nombre: optativo, se llama rótulo o etiqueta de la estructura.
- $tipo_i$ $var_i$: son los campos o miembros de la estructura. Son variables que se declaran con su tipo asociado, que podrá ser cualquier tipo válido.

# Problema:


Suponga que usted quisiera llevar registro de sus compras en mercado libre. Querría agrupar para su referencia la siguiente información de sus comprar:

- Nombre del producto
- Categoría
- Precio
- Cantidad
- Forma de pago
- Fecha de entrega

Crear una estructura que cumpla con la información detallada

# Declaración de variables tipo estructurado

Una vez definido el nuevo tipo de datos, se estará en condiciones de declarar variables asociadas a el, por ejemplo:

```
                    char
nom[MAX];
                    char
ape[MAX];
                    int
edad;
                    float
peso;
                    char
sexo;
                }
```

```
struct persona
PACIENTE;
struct persona  CLIENTE;
==================
==============

Struct persona
PACIENTE, CLIENTE;
```

La declaración de las variables "PACIENTE" y "CLIENTE" asociadas al tipo persona SI IMPLICA UNA RESERVA DE

# INICIALIZACION DE UNA VARIABLE ASOCIADA AL TIPO ESTRUCTURA

Se puede inicializar una variable asociada al tipo estructura en su declaración, ellos se hará siguiendo su declaración con una lista de inicializadores, cada uno con una expresión constante para sus miembros.

```
struct  PUNTO {
                    int puntoX;
                    int puntoY;
                  };
Int main()
{
          struct PUNTO  punto1= { 7, 5};

            …

          return 0;
}
```

# COMO ACCEDER A LOS CAMPOS O MIEMBROS DE UNA ESTRUCTURA

- El acceso a los miembros o campos:  de una variables asociada al tipo estructura se hace a través del OPERADOR PUNTO

- La forma de uso del mismo es:

**Nombre_de_ la_variable.nombre_del_campo**

# Ejemplo

```
struct  persona{
                        char nom[MAX];
                       char ape[MAX];
                       int edad;
                       float peso;
                       char sexo;
                     };
Int main()
{
            struct persona PACIENTE;
            …

            printf(“%d”, PACIENTE.edad);
            printf(“%f”, PACIENTE.peso);
            printf(“%c”, PACIENTE.sexo);

           return 0;
}
```

# OPERACIONES

Con las variables estructuradas se pueden llevar a cabo las siguientes acciones:

- copiar como una unidad
- asignar entre ellas
- acceder y trabajar con sus campos
- anidar estructuras
- usar como argumentos de funciones: completas y por partes
- tomar su dirección (&)
- ser devueltas por funciones

# COMO CARGAR DATOS EN UNA VARIABLE ESTRUCTURADA

- Una vez declarada la variable tipo estructura, veamos de que manera podemos acceder o los miembros o campos que componen la misma.
- Si considero estructura persona, declaro una variable asociada a la misma de nombre individuo, esto es: **struct persona individuo;**

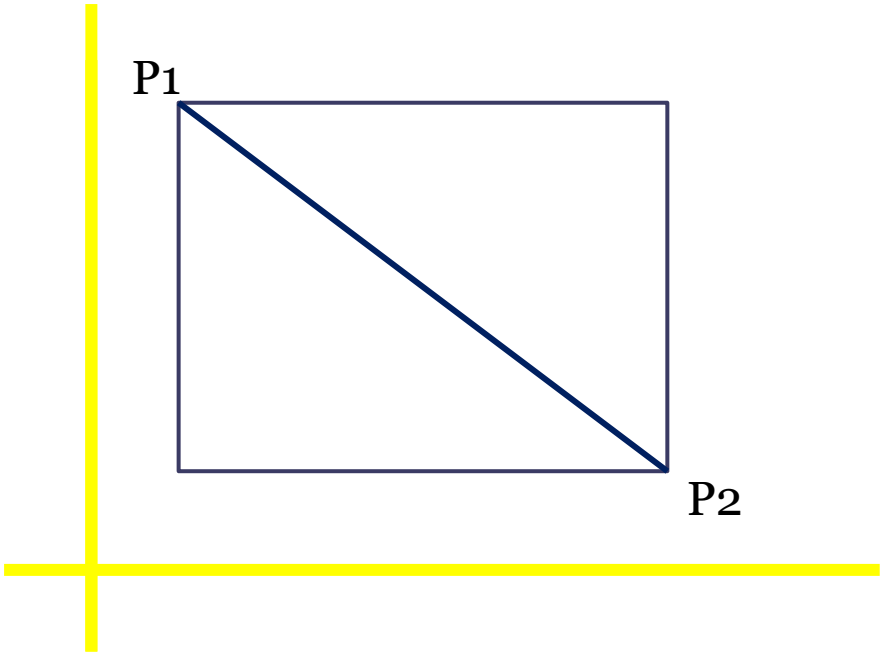
Podemos usar esta variable para conocer de que manera puedo acceder a los campos de la variable. Si lo que me propongo es por ejemplo, “cargar los datos del individuo”, deberé para ello usar el operador que me permite acceder a los campos de la variable estructurada, o sea el operador punto, de la siguiente manera: **gets(individuo.ape);**

# Ejemplo

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

struct persona {
char ape[100];
char nom[100];
};

int main(void) {

struct persona PACIENTE1;

struct persona PACIENTE3;

gets(PACIENTE1.ape);
gets(PACIENTE1.nom);

PACIENTE3 = PACIENTE1;

puts(PACIENTE1.nom);
puts(PACIENTE3.nom);


return 0;
}
```

# ESTRUCTURAS ANIDADAS

Las estructuras pueden anidarse. Una representación de un rectángulo se puede hacer como un par de puntos que denotan las esquinas diagonalmente opuestas




```
struct  PUNTO {
                    int puntoX;
                    int puntoY;
                  };

struct  RECTA {
                    struct PUNTO
P1;
                    struct PUNTO
P2;
                  };
```

# Retomemos nuestro Ejercicio:


1. ¿En que caso podríamos tener estructuras anidadas en nuestro ejercicio?
2. Cargue los datos de dos productos comprados.
3. Muestre por pantalla cada uno de los campos de la estructura.
4. Compare los precios de ambos productos, e indique cual es el mas costoso.

```
#include <stdio.h>

int main()
{
    printf("Hasta la próxima clase!!\n");

    return 0;
}
```