COMPUTACIÓN I
TEMA 8.
Constructor de tipos: struct
Definición de nuevos tipos: typedef.
Arreglos de estructuras

Prof. Mireya Morales

CONTENIDO

- Constantes de enumeración
- Definición de estructuras. Uso de *struct*
- Disposición de las Estructuras en memoria.
- Ejemplo de una declaración de una estructura, usada en un programa.
- Creación de sinónimos o alias. Uso de *typedef*
- Arreglos de estructuras

Constantes de Enumeración

- El lenguaje C proporciona un tipo de dato definido por el usuario, llamado enumeración.

- La palabra reservada que se utiliza es *enum*, es un conjunto de constantes enteras, representadas por identificadores

Constantes de Enumeración

- Ejemplo:
  - enum meses {ene, feb, mar, abr, may, jun, jul, ago, sep, oct, nov, dic};
  - En este caso el identificador ene se inicializa en 0, luego feb es 1 y asi sucesivamente hasta dic=11
  - Si se desea que ene comience con 1, se hace:
  - Enum meses {ene=1, feb, mar, abr, may, jun, jul, ago, sep, nov, dic}; de esta forma dic=12
  - Enum meses mes;

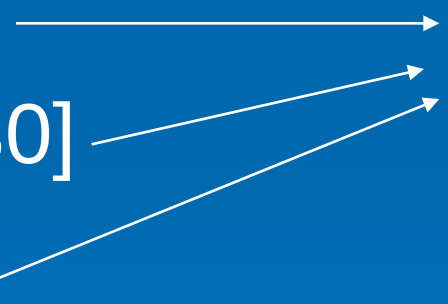
Definición de estructuras.

- Una estructura es un tipo de datos que permite empaquetar elementos bajo un mismo nombre. Estos elementos pueden ser de un mismo o de distinto tipos de datos, que se encuentran relacionados lógicamente.
- También es conocida con el nombre de registro.

Definición de estructuras. Uso de *struct*

En lenguaje C, se define:

```
struct Alumno{
   char nombre[30]
   char password[30]
   char email[50]
};
```



Definición de estructuras. Uso de *struct*

Otro ejemplo:

```
struct Alumno{
    int cedula;
    char nombre[30];
    char carrera[30];
    float promedio;
    char direccion[20];
};
```

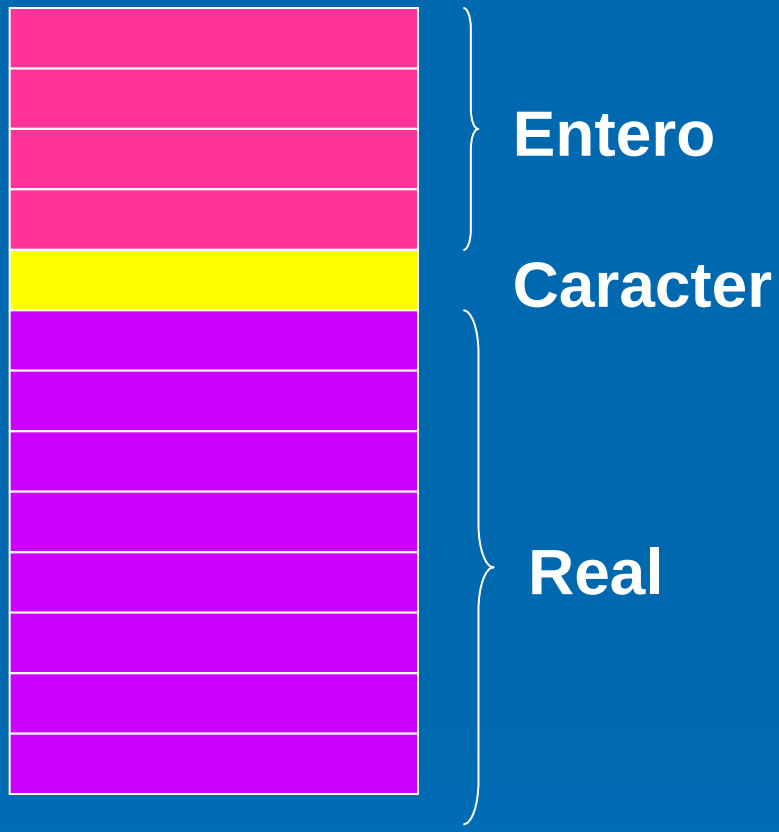
Una definición general de estructura

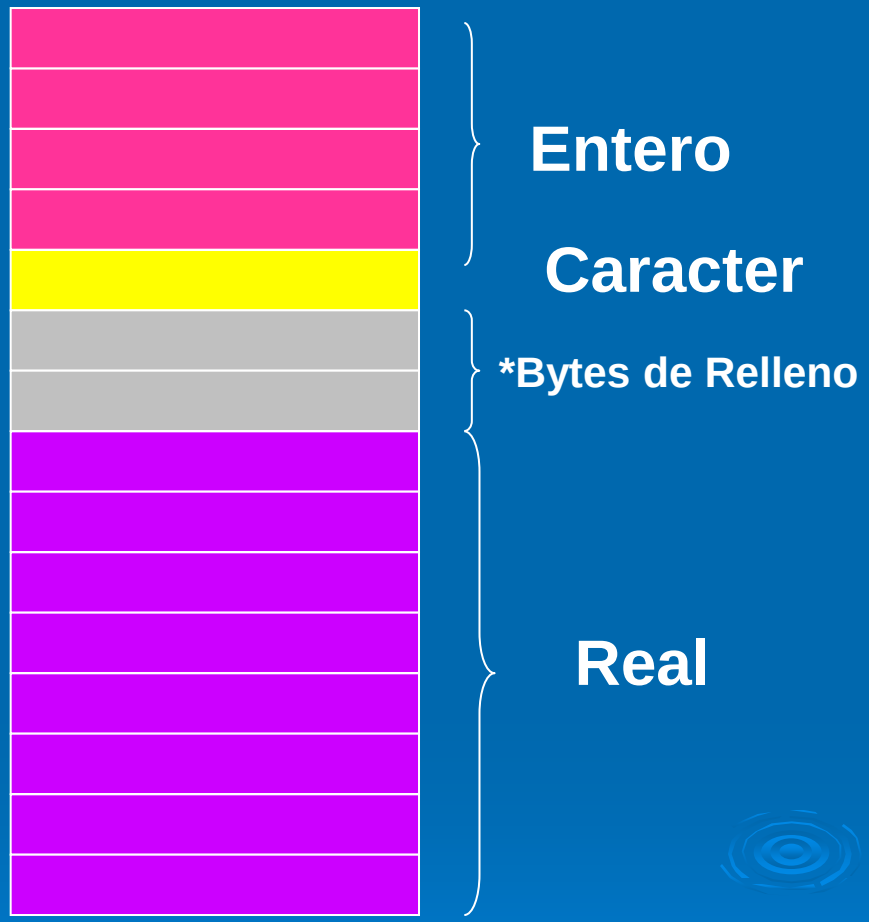
```
struct  nombreEstructura
{
   TipoDato1          miembro1;
   TipoDato2          miembro2;
   .
   .
   TipoDatoN          miembroN;
}
```

Disposición de las estructuras en memoria

```
struct Datos{
int entero;
char carácter:
double real;
}
sizeof(struct Datos) >= sizeof(int) +
   sizeof(char) + sizeof(double)
```

Disposición de las estructuras en memoria





Este tipo de rellenos es necesario cuando el computador necesita que un tipo de dato real comience en una dirección de memoria que sea múltiplo de 4.

Ejemplo de una declaración de una estructura, usada en un programa.

Ejemplo1

Creación de sinónimos o alias. Uso de *typedef*

- La instrucción *typedef* permite al usuario definir alias o sinónimos.
- El objeto de esta instrucción es utilizar nombres más apropiados y más cortos para los tipos de datos. Evita escribir la palabra *struct* en la declaración de variables.
- Ejemplo:
  - *typedef int entero;*
  - *entero c1, c2, c3:*

Creación de sinónimos o alias. Uso de *typedef*

- *Ejemplo 2*

- *typedef enum {enero, febrero, marzo, abril, mayo, junio, julio, agosto, septiembre, octubre, noviembre, diciembre} Meses;*
- *Meses Mes;*

Arreglos de estructuras

- Es frecuente el uso conjunto de estructuras y arreglos.
- Un arreglo de estructuras representa una lista de entidades, que actúa como una pequeña base de datos, formando una tabla que tiene como identificadores de columna los atributos y como identificadores de fila, el índice del arreglo.

Arreglos de estructuras

```
#define NUMERO_FECHAS 100
Struct Fecha
{
   int dia;
   int mes;
   int anyo;
};

Struct Fecha fechas[NUMERO_FECHAS];
```

Arreglos de estructuras

Struct Fecha fechas[4]