

ESTRUCTURAS ANIDADAS

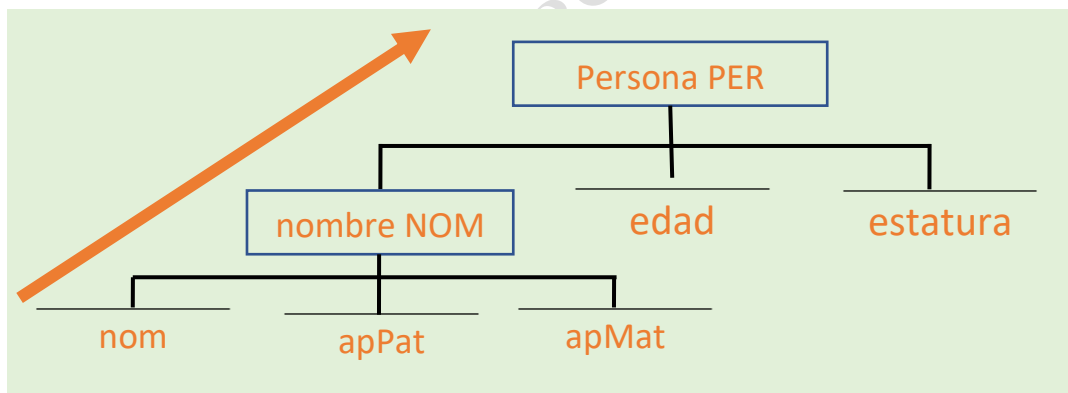
Se le llama así al hecho de que una estructura contiene elementos/ítems/campos que a su vez sean de tipo estructura.

El esquema puede llegar a repetirse, de esta manera se puede tener varios *niveles de anidación*.

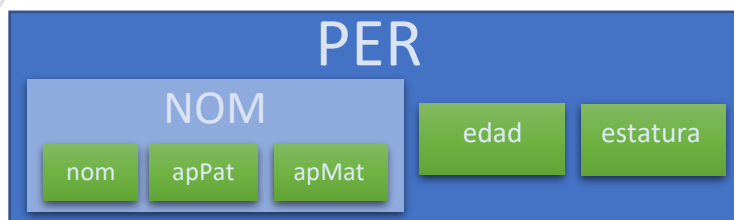
Bien, hagamos un ejemplo.

Continuando con el ejercicio realizado anteriormente, supongamos ahora que el **nombre de la persona** se manejará dividido en 3 partes: **nombre**, **ap. Paterno** y **ap. Materno**.

De manera gráfica la podemos representar así:



O también:



Ahora procedemos a declarar las estructuras necesarias, para esto lo recomendable es empezar como lo indica la flecha de abajo hacia arriba (y de izquierda a derecha, por niveles), es

decir declaramos de las estructuras más simples a las más complejas.

Entonces las estructuras quedarían así:

```
typedef struct
{
    char nom[25];
    char apPat[25];
    char apMat[25];
} NOM;
typedef struct
{
    NOM nombre; // Campo anidado
    int edad;
    float estatura;
} PER;
```

Accediendo a los datos de la estructura anidada

El acceso a los datos se realiza de manera similar a una estructura “normal” solo que ahora cada campo que sea tipo estructura se tendrá que **desglosar** nuevamente en sus ítems/campos por medio del operador “.”. Veamos el ejemplo:

```
PER p;
```

```
p.nombre.nom
```

```
p.nombre.apPat
```

```
p.nombre.apMat
```

```
p.edad
```

```
p.estatura
```

} Acceso a campos anidados

Los campos no anidados se acceden como ya se había mencionado.

Si la estructura anidada se pasa **por referencia**, entonces el acceso sería:

```
__ funX(PER *p ...)
    (*p).nombre.nom    // o    p->nombre.nom
    p->nombre.apPat
    p->nombre.apMat
    p->edad
    p->estatura
```

Bien ahora desarrollemos el programa completo.

```
/*
Objetivo: Manejo básico de una estructura anidada
Autor: M.A.C.L.
Fecha: 30/04/2021
Nota(s): La estructura representará a una persona con los
datos:
nombre(nom, apPat, apMat), edad(int), estatura(float)
*/

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

typedef struct
{
    char nom[25];
    char apPat[25];
    char apMat[25];
} NOM;

typedef struct
{
    NOM nombre;
    int edad;
    float estatura;
} PER;

void capturaPer(PER *p);
void muestraPer(PER p);

int main()
{
    PER per;

    capturaPer(&per);
    muestraPer(per);
}

void capturaPer(PER *p)
{
    printf("Nombre: ");
    gets(p->nombre.nom);
    printf("Ap. paterno: ");
    gets(p->nombre.apPat);
    printf("Ap. materno: ");
    gets(p->nombre.apMat);
    printf("Edad: ");
    scanf("%d", &p->edad);
    printf("Estatura: ");
    scanf("%f", &p->estatura);
}

void muestraPer(PER p)
{
    printf("Nombre: %s\n", p.nombre.nom);
    printf("Ap. paterno: %s\n", p.nombre.apPat);
    printf("Ap. materno: %s\n", p.nombre.apMat);
    printf("Edad: %d\n", p.edad);
    printf("Estatura: %.2f\n", p.estatura);
}
```