

# PROGRAMACIÓN

## PRÁCTICA TP N°8

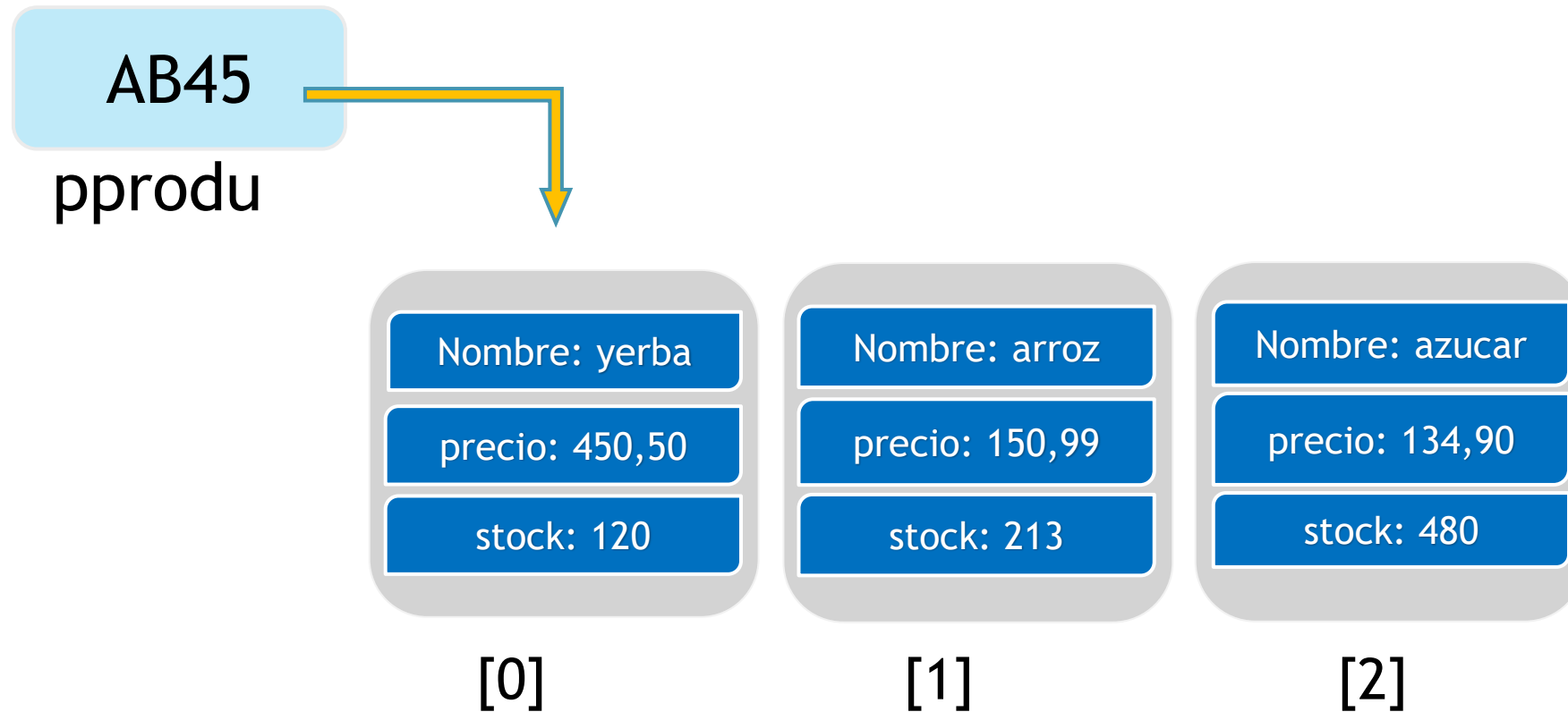
P.U. PABLO AGÜERO – LIC. GABRIELA BARRAZA – LIC. JOSEFINA LOBO



# REPASO TEÓRICO



# Arreglos de Estructuras y Punteros



Arreglo de tipo producto

```
struct producto
{
    char nombre[30];
    float precio;
    int stock;
} typedef producto;
```

# Arreglos de Estructuras y Punteros

Declaración de un arreglo de tipo estructura y un puntero .

Inicialización del puntero.



```
scanf("%d", &(*pprodu).stock );  
(*pprodu).stock = 10;
```

Acceso a cada campo de la estructura mediante el **operador flecha**.

Muevo el puntero

```
int main(){  
    producto datos[MAX], *pprodu;  
    pprodu = datos;  
  
    for (int i = 0; i < MAX; i++)  
    {  
        printf("----- PRODUCTO [%d] -----\\n", i+1);  
  
        printf("Ingrese el nombre del producto: ");  
        fflush(stdin);  
        gets( pprodu->nombre );  
  
        printf("Ingrese el precio:");  
        scanf("%f", &pprodu->precio );  
  
        printf("Ingrese el stock:");  
        scanf("%d", &pprodu->stock );  
  
        pprodu++;  
    }  
}
```

# Arreglos de Estructuras

## Con asignación dinámica de memoria

Para crear un arreglo **dinámico** de tipo estructura:

- Incluir la biblioteca `include<stdlib.h>`.
- Declarar un puntero de tipo estructura.
- Pedir la cantidad de elementos a cargar.
- Hacer reserva dinámica mediante `malloc`.
- Al finalizar liberar la memoria reservada con la función **free**.

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

struct producto
{
    char nombre[30];
    float precio;
    int stock;
} typedef producto;

int main(){
    producto *pprodu;
    int cant;

    puts("Ingrese la cantidad de clientes: ");
    scanf("%d", &cant);
    pprodu = (producto *) malloc(cant*sizeof(producto));
    //...
    free(pprodu);
```

# Arreglos de Estructuras

## Con asignación dinámica de memoria

```
pprodu = (producto*) malloc ( cant * sizeof (producto) );
```

Puntero

Cast

Cantidad de elementos que  
tendrá el arreglo dinámico

Función que permite asignar  
un bloque de memoria

Tamaño total  
del tipo struct

# Arreglos de Estructuras Con Punteros y Funciones

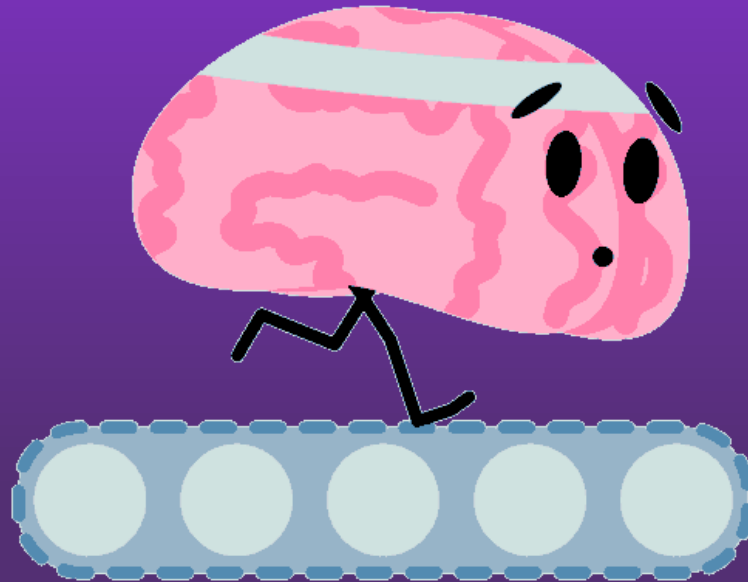
```
void cargar(producto *p,int cant);  
void mostrar(producto *p,int cant);
```

```
int main(){  
    producto *pprodu;  
    int cant;  
  
    puts("Ingrese la cantidad de productos: ");  
    scanf("%d", &cant);  
    pprodu = malloc(cant*sizeof(producto));  
  
    cargar(pprodu, cant);  
    mostrar(pprodu, cant);  
  
    free(pprodu);  
    return 0;  
}
```

```
void cargar(producto *p,int cant){  
    for (int i = 0; i < cant; i++)  
    {  
        printf("---- PRODUCTO [%d] ----\n", i+1);  
        printf("Nombre del producto: ");  
        gets( p->nombre );  
        printf("Ingrese el precio:");  
        scanf("%f", &p->precio );  
        printf("Ingrese el stock:");  
        scanf("%d", &p->stock );  
        p++;  
    }  
}
```

```
void mostrar(producto *p,int cant){  
    for (int i = 0; i < cant; i++)  
    {  
        printf("---- PRODUCTO [%d] ----\n", i+1);  
        printf("Producto: %s \n", p->nombre );  
        printf("Precio: %.2f \n", p->precio );  
        printf("Stock: %d \n", p->stock );  
        p++;  
    }  
}
```

# PRÁCTICA





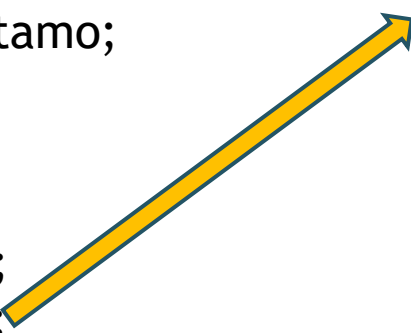
# TP8 - Punto 1

## Punto de partida:

```
typedef struct {  
    int dia;  
    int mes;  
    int anio;  
} Fecha;
```

```
typedef struct {  
    char nombre[MAX];  
    short edad;  
    Fecha fechaPrestamo;  
} Lector;
```

```
typedef struct {  
    char titulo[MAX];  
    int numLectores;  
    Lector *lectores;  
} Libro;
```



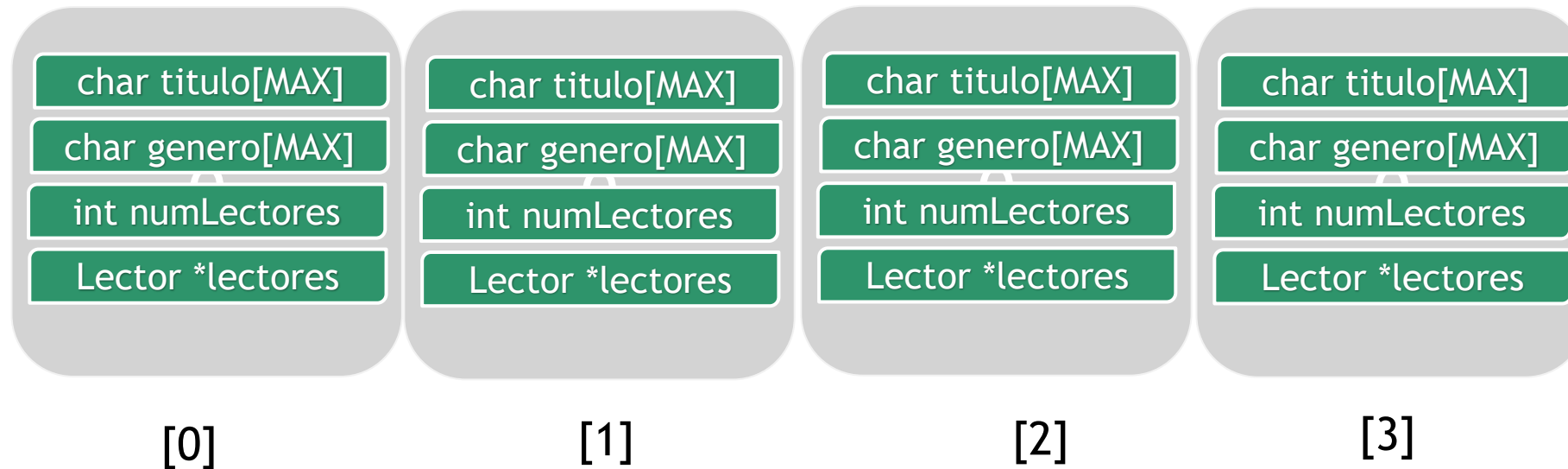
Campo que nos da la cantidad que utilizaremos para crear el arreglo dinámico de lectores

# TP8 - Punto 1

```
typedef struct {  
    char titulo[MAX];  
    char genero[MAX];  
    int numLectores;  
    Lector *lectores;  
} Libro;
```

Puntero

b) Cree un arreglo dinámico para 4 libros o más.



¿Como creo  
el arreglo?

```
int main() {
```

```
    int numLibros;  
    scanf("%d", &numLibros);  
    Libro *libros = (Libro *)malloc(numLibros * sizeof(Libro));
```

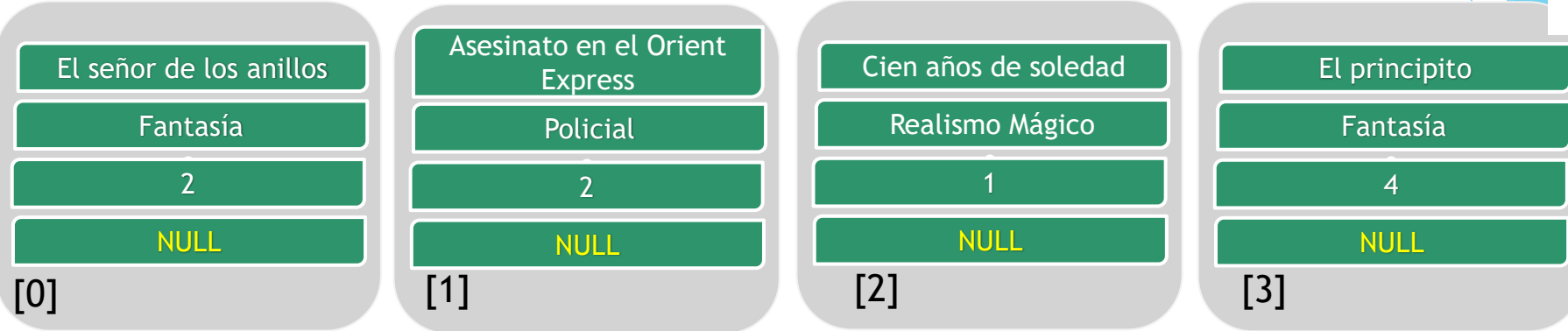
```
int main() {
```

```
    int numLibros = 4;  
    Libro *libros = (Libro *)malloc(numLibros * sizeof(Libro));
```

# TP8 - Punto 1

b) Cree un arreglo dinámico para 4 libros o más.

libros



```
typedef struct {  
    char titulo[MAX];  
    char genero[MAX];  
    int numLectores;  
    Lector *lectores;  
} Libro;
```

# TP8 - Punto 1

c) Cree arreglos dinámicos de lectores para cada libro, con diferentes cantidades de lectores.

Puntero

char titulo[MAX]  
char genero[MAX]  
int numLectores  
Lector \*lectores

[0]

char titulo[MAX]  
char genero[MAX]  
int numLectores  
Lector \*lectores

[1]

char titulo[MAX]  
char genero[MAX]  
int numLectores  
Lector \*lectores

[2]

char titulo[MAX]  
char genero[MAX]  
int numLectores  
Lector \*lectores

[3]

[0]	char nombre[MAX]; short edad; Fecha fechaPrestamo; int dia; int mes; int anio;
[1]	char nombre[MAX]; short edad; Fecha fechaPrestamo; int dia; int mes; int anio;
[2]	...

[0]	char nombre[MAX]; short edad; Fecha fechaPrestamo; int dia; int mes; int anio;
[1]	char nombre[MAX]; short edad; Fecha fechaPrestamo; int dia; int mes; int anio;
[2]	...
[3]	...

[0]	char nombre[MAX]; short edad; Fecha fechaPrestamo; int dia; int mes; int anio;
-----	---

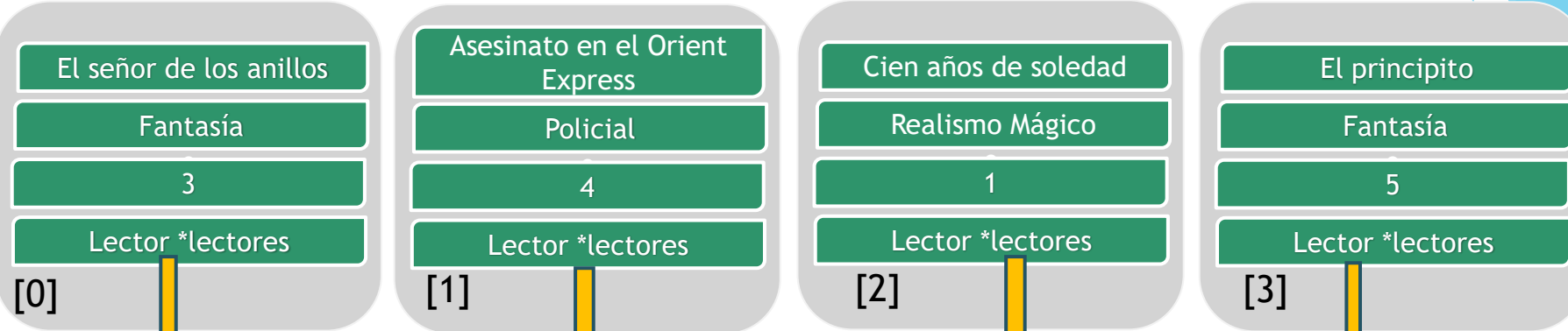
[0]	char nombre[MAX]; short edad; Fecha fechaPrestamo; int dia; int mes; int anio;
[1]	...
[2]	...
[3]	...
[4]	...

```
typedef struct {  
    int dia;  
    int mes;  
    int anio;  
} Fecha;  
  
typedef struct {  
    char nombre[MAX];  
    short edad;  
    Fecha fechaPrestamo;  
} Lector;  
  
typedef struct {  
    char titulo[MAX];  
    char genero[MAX];  
    int numLectores;  
    Lector *lectores;  
} Libro;
```

# TP8 - Punto 1

c) Cree arreglos dinámicos de lectores para cada libro, con diferentes cantidades de lectores.

libros



[0]	Carlos Pérez 25 1/6/2023
[1]	Laura Astrada 41 12/10/2023
[2]	Fátima Robles 18 9/6/2024

[0]	Silvia López 18 19/8/2024
[1]	Carlos Pérez 25 1/6/2023
[2]	Lisandro Reyes 11 9/11/2024
[3]	Fátima Robles 18 9/6/2024

[0]	Laura Astrada 41 30/6/2023
-----	----------------------------------

[0]	Silvia López 18 19/8/2024
[1]	Carlos Pérez 25 1/6/2023
[2]	Lisandro Reyes 11 9/11/2024
[3]	Fátima Robles 18 9/6/2024
[4]	Diego García 31 10/10/2023

```
typedef struct {  
    int dia;  
    int mes;  
    int anio;  
} Fecha;  
  
typedef struct {  
    char nombre[MAX];  
    short edad;  
    Fecha fechaPrestamo;  
} Lector;  
  
typedef struct {  
    char titulo[MAX];  
    char genero[MAX];  
    int numLectores;  
    Lector *lectores;  
} Libro;
```

¿Como creo el arreglo de lectores?

# TP8 - Punto 1

## Inicializar datos (con indexado de punteros)

```
int main() {

    int numLibros = 4;
    Libro *libros = (Libro *)malloc(numLibros * sizeof(Libro)); // Reserva para 4 libros

    // Inicialización del libro 1
    libros[0] = (Libro){"El señor de los anillos", "Fantasía", 3, (Lector *)malloc(3 * sizeof(Lector))};
    libros[0].lectores[0] = (Lector){"Juan Perez", 25, {12, 5, 2023}};
    libros[0].lectores[1] = (Lector){"Maria Lopez", 30, {14, 5, 2023}};
    libros[0].lectores[2] = (Lector){"Pedro Sanchez", 35, {7, 6, 2023}};

    // Inicialización del libro 2
    libros[1] = (Libro){"1984", "Distopía", 4, (Lector *)malloc(4 * sizeof(Lector))};
    libros[1].lectores[0] = (Lector){"Carlos Diaz", 21, {1, 6, 2023}};
    libros[1].lectores[1] = (Lector){"Lucia Gomez", 29, {5, 6, 2023}};
    libros[1].lectores[2] = (Lector){"Pedro Sanchez", 35, {7, 6, 2023}};
    libros[1].lectores[3] = (Lector){"Roberto Castro", 33, {20, 8, 2023}};

    // Inicialización del libro 3
    libros[2] = (Libro){"Cien años de soledad", "Realismo Mágico", 1, (Lector *)malloc(1 * sizeof(Lector))};
    libros[2].lectores[0] = (Lector){"Ana Martinez", 40, {10, 7, 2023}};

    // Inicialización del libro 4
    libros[3] = (Libro){"Don Quijote de la Mancha", "Clásico", 2, (Lector *)malloc(2 * sizeof(Lector))};
    libros[3].lectores[0] = (Lector){"Roberto Castro", 33, {20, 8, 2023}};
    libros[3].lectores[1] = (Lector){"Laura Perez", 27, {22, 8, 2023}};
```

```
typedef struct {
    int dia;
    int mes;
    int anio;
} Fecha;

typedef struct {
    char nombre[MAX];
    short edad;
    Fecha fechaPrestamo;
} Lector;

typedef struct {
    char titulo[MAX];
    char genero[MAX];
    int numLectores;
    Lector *lectores;
} Libro;
```

# TP8 - Punto 1

## Inicializar datos (con desplazamiento de punteros)

```
int numLibros = 4;
Libro *libros = (Libro *)malloc(numLibros * sizeof(Libro)); // Reserva para 4 libros

// Inicialización del libro 1
*libros = (Libro){"El señor de los anillos", "Fantasía", 2, (Lector *)malloc(2 * sizeof(Lector))};
*(libros->lectores) = (Lector){"Juan Perez", 25, {12, 5, 2023}};
*(libros->lectores + 1) = (Lector){"Maria Lopez", 30, {14, 5, 2023}};

// Inicialización del libro 2
*(libros + 1) = (Libro){"1984", "Distopía", 3, (Lector *)malloc(3 * sizeof(Lector))};
*((libros + 1)->lectores) = (Lector){"Carlos Diaz", 21, {1, 6, 2023}};
*((libros + 1)->lectores + 1) = (Lector){"Lucia Gomez", 29, {5, 6, 2023}};
*((libros + 1)->lectores + 2) = (Lector){"Pedro Sanchez", 35, {7, 6, 2023}};

// Inicialización del libro 3
*(libros + 2) = (Libro){"Cien años de soledad", "Realismo Mágico", 1, (Lector *)malloc(1 * sizeof(Lector))};
*((libros + 2)->lectores) = (Lector){"Ana Martinez", 40, {10, 7, 2023}};

// Inicialización del libro 4
*(libros + 3) = (Libro){"Don Quijote de la Mancha", "Clásico", 2, (Lector *)malloc(2 * sizeof(Lector))};
*((libros + 3)->lectores) = (Lector){"Roberto Castro", 33, {20, 8, 2023}};
*((libros + 3)->lectores + 1) = (Lector){"Laura Perez", 27, {22, 8, 2023}};
```

```
typedef struct {
    int dia;
    int mes;
    int anio;
} Fecha;

typedef struct {
    char nombre[MAX];
    short edad;
    Fecha fechaPrestamo;
} Lector;

typedef struct {
    char titulo[MAX];
    char genero[MAX];
    int numLectores;
    Lector *lectores;
} Libro;
```

# TP8 - Punto 1

## Inicializar datos (con variables auxiliares)

```
// Variables auxiliares para lectores
Lector lectorAux1 = {"Juan Perez", 25, {12, 5, 2023}};
Lector lectorAux2 = {"Maria Lopez", 30, {14, 5, 2023}};
Lector lectorAux3 = {"Carlos Diaz", 21, {1, 6, 2023}};
Lector lectorAux4 = {"Lucia Gomez", 29, {5, 6, 2023}};
Lector lectorAux5 = {"Pedro Sanchez", 35, {7, 6, 2023}};
Lector lectorAux6 = {"Ana Martinez", 40, {10, 7, 2023}};
Lector lectorAux7 = {"Roberto Castro", 33, {20, 8, 2023}};
Lector lectorAux8 = {"Laura Perez", 27, {22, 8, 2023}};

// Variables auxiliares para libros
Libro libroAux1 = {"El señor de los anillos", "Fantasía", 2, NULL};
Libro libroAux2 = {"1984", "Distopía", 3, NULL};
Libro libroAux3 = {"Cien años de soledad", "Realismo Mágico", 1, NULL};
Libro libroAux4 = {"Don Quijote de la Mancha", "Clásico", 2, NULL};

// Reserva de memoria para el arreglo de libros dinámico
int numLibros = 4;
Libro *libros = (Libro *)malloc(numLibros * sizeof(Libro));

// Asignación de libros auxiliares al arreglo dinámico
libros[0] = libroAux1;
libros[1] = libroAux2;
libros[2] = libroAux3;
libros[3] = libroAux4;
```

```
typedef struct {
    int dia;
    int mes;
    int anio;
} Fecha;

typedef struct {
    char nombre[MAX];
    short edad;
    Fecha fechaPrestamo;
} Lector;

typedef struct {
    char titulo[MAX];
    char genero[MAX];
    int numLectores;
    Lector *lectores;
} Libro;
```

```
// Reserva dinámica y
// asignación de lectores desde los auxiliares
libros[0].lectores = (Lector *)malloc(2 * sizeof(Lector));
libros[0].lectores[0] = lectorAux1;
libros[0].lectores[1] = lectorAux2;

libros[1].lectores = (Lector *)malloc(3 * sizeof(Lector));
libros[1].lectores[0] = lectorAux3;
libros[1].lectores[1] = lectorAux4;
libros[1].lectores[2] = lectorAux5;

libros[2].lectores = (Lector *)malloc(1 * sizeof(Lector));
libros[2].lectores[0] = lectorAux6;

libros[3].lectores = (Lector *)malloc(2 * sizeof(Lector));
libros[3].lectores[0] = lectorAux7;
libros[3].lectores[1] = lectorAux8;
```



# PRÁCTICA

