

# PRACTICA 1

---

## Identificación de Elementos Fundamentales en los Lenguajes de Programación

---

### 1. Nombres

- **Variables globales:**
    - Ejemplos: `heap_allocations`, `heap_deallocations`, `stack_allocations`, `stack_deallocations`.
  - **Estructuras y tipos:**
    - Ejemplos: `book_t`, `member_t`, `genre_t`, `MemoryRecord`.
  - **Funciones / Subprogramas:**
    - Ejemplos: `addBook`, `findBookById`, `displayBooksRecursive`, `displayMemoryUsage`.
- 

### 2. Marcos de Activación

- Cada vez que se llama a una función (`addBook`, `issueBook`, `returnBook`, etc.) se crea un **marco de activación** en el *stack* con sus variables locales y parámetros.
- 

### 3. Bloques de Alcance

- **Bloques delimitados por { }:**
    - Dentro de `main` y en cada función.
  - **Ámbito local:** variables declaradas dentro de funciones (`bookFound`, `memberFound`).
  - **Ámbito global:** variables definidas fuera de funciones (`bss_var`, `static_var`).
- 

### 4. Administración de Memoria

- **Heap:**
    - Uso de `malloc`, `free`, `realloc` (ejemplo: creación de libros y miembros).
    - Funciones de gestión: `incrementHeapAllocations`, `incrementHeapDeallocations`.
  - **Stack:**
    - Variables locales (`choice`, `bookID`, `memberID`).
    - Contabilizadas con `incrementStackAllocations`, `incrementStackDeallocations`.
- 

### 5. Expresiones

- Operaciones aritméticas y lógicas:
  - `memberFound->issued_count++`
  - `current->id == bookID`
  - `genre = (genre_t)genre.`

---

## 6. Comandos (Sentencias)

- **Asignaciones:**
    - `new_book->id = ...`
  - **Llamadas a funciones:**
    - `displayMemoryUsage(), findBookById(...)`.
  - **Entrada/Salida:**
    - `printf, scanf, fgets, fprintf, fscanf`.
- 

## 7. Control de Secuencia

### a) Selección

- `if, else, switch`.
  - Ejemplo: `if (!new_book) { printf("Error..."); return; }`
  - `switch (genre) { case FICTION: return "Ficcion"; ... }`

### b) Iteración

- `while, for, do...while`.
  - Ejemplo: recorrer la lista de libros con `while (current)`.
  - Menú principal con `do { ... } while(choice != 8)`.

### c) Recursión

- `displayBooksRecursive(book_t *library)` se llama a sí misma para mostrar todos los libros.
- 

## 8. Subprogramas (Funciones)

- **Gestión de libros:** `addBook, findBookById, displayBooks, freeLibrary, saveLibraryToFile, loadLibraryFromFile`.
  - **Gestión de miembros:** `addMember, issueBook, returnBook, freeMembers, saveMembersToFile, loadMembersFromFile, displayMembers, searchMember`.
  - **Memoria:** `displayMemoryUsage, incrementHeapAllocations, incrementHeapDeallocations`.
  - **Principal:** `main`.
- 

## 9. Tipos de Datos

- **Primitivos:** `int, char, float, size_t`.
  - **Compuestos:**
    - `struct book_t, struct member_t, struct MemoryRecord`.
  - **Enumerados:** `enum genre_t`.
  - **Punteros:** `book_t *library, member_t *members, void *pointer`.
- 

## REFERENCIAS

<https://github.com/adrianbalderas373488/Portafolio>