



UNIVERSIDAD AUTONOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERIA, ARQUITECTURA Y DISEÑO

INGENIERIA EN SOFTWARE Y TECNOLOGIAS EMERGENTES

GRUPO: 932

MATERIA: Lenguaje C

MAESTRA: Yulith Vanessa Altamirano Flores

TITULO:
Taller 5: Clases de Almacenamiento

ALUMNA: Teresa Rivas Gómez

MATRICULA: 372565

Código:

- Documentación y Librerías utilizadas

```
// Teresa Rvas Gómez / Matricula: 372565
// Fecha de creación: Sep - 30 - 2023
// Materia: Lenguaje C / Actividad: Taller 5 "Clases de Almacenamiento"

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

- Declaración de variables

```
// Menu
int msges();
void menu();
// Ejercicios
void automatica();
void externa(int cantidad);
void estatica();
void de_registro();
void local();
// Variable global
float pi = 3.1416;
// Variable externa
extern int saldo;
```

- Menu

```
// Menu principal
int main()
{
    menu();
    return 0;
}
// Mensaje al usuario para seleccionar ejercicio en el menu
int msges()
{
    int op;
    system ("CLS");
    // Mensajes
    printf("\n TALLER 5 \n");
    printf(" MENU: Elige el tipo de variable. \n");
    printf("1. Automatica. \n");
    printf("2. Externa. \n");
    printf("3. Estatica. \n");
    printf("4. De Registro. \n");
    printf("5. Global y Local. \n");
    printf("0. Salir del programa. \n");
    scanf("%d", &op);
    return op;
}
// Seleccion de ejercicios segun sea el caso
void menu()
{
```

```
int op;  
do {  
    op = msges();  
    switch (op)  
    {  
        case 1:  
            automatica();  
            automatica();  
            automatica();  
            automatica();  
            automatica();  
            automatica();  
            automatica();  
            automatica();  
            automatica();  
            automatica();  
            break;
```

- Declaramos variable cantidad e imprimimos lo que vera el usuario, para llamar a nuestra función externa.

```
        case 2:  
            // Declarar variables  
            int cantidad;  
            // Limpiar pantalla  
            system ("CLS");  
            printf ("Saldo actualL: %d\n", saldo);  
            printf ("Añadir al saldo existente: \n");  
            scanf("%d", &cantidad);  
            // Llamar a la funcion externa  
            externa(cantidad);  
            // Pausar  
            system ("PAUSE");  
            break;  
        case 3:  
            estatica();  
            estatica();  
            estatica();  
            estatica();  
            estatica();  
            estatica();  
            estatica();  
            estatica();  
            estatica();  
            estatica();  
            break;  
        case 4:  
            de_registro();  
            break;  
        case 5:  
            local();  
            break;  
        case 0:  
            printf("Saliste del programa.\n");  
            break;  
        default:  
            printf("Opcion invalida, intentalo de nuevo y lee bien porfa.\n");  
            break;  
    }
```

```
} while(op != 0);  
}
```

- **Ejercicio 1:**

```
// *Para los ejercicios con * se invocaran 10 veces las funciones.  
// EJERCICIO 1:  
// Declara una variable automática llamada contador en una función. Incrementa  
// su valor en un bucle y muestra su valor en cada iteración. ¿Qué sucede con la  
// variable al salir de la función? *  
void automatica()  
{  
    // Declarar variables  
    int contador;  
    // Limpiar pantalla  
    system ("CLS");  
    // Bucle  
    for(contador = 0; contador <= 10; contador++)  
    {  
        // Valor en cada iteracion  
        printf("El valor de contador es: %d\n", contador);  
    }  
    // Pausar  
    system ("PAUSE");  
}
```

- **Ejercicio 3:**

```
// EJERCICIO 3:  
// Declara una variable estática llamada contador en una función y muestra su  
// valor en cada llamada a la función. ¿Qué sucede con la variable al salir de la  
// función? ¿Cómo difiere de una variable automática? *  
void estatica()  
{  
    // Declarar variables  
    static int contador = 0;  
    // Limpiar pantalla  
    system ("CLS");  
    // Incremento  
    contador++;  
    // Imprimir  
    printf("El valor de contador es: %d\n", contador);  
    // Pausar  
    system ("PAUSE");  
}
```

- **Ejercicio 4:**

```
// EJERCICIO 4:  
// Declara una variable de registro llamada temp y otra automática llamada valor  
// en una función. Compara el acceso y el tiempo de vida de estas variables. ¿Por  
// que usarías una variable de registro en lugar de una variable automática?  
void de_registro()  
{  
    // Declarar variables  
    register int temp;  
    int valor_en_funcion;  
    // Limpiar pantalla  
    system ("CLS");  
    // Asignar valores
```

```
temp = 10;
valor_en_funcion = 10;
// Imprimir
printf("Valor de temp: %d\n", temp);
printf("Valor de valor_en_funcion: %d\n", valor_en_funcion);
// Pausar
system ("PAUSE");
}
```

- **Ejercicio 5:**

```
// EJERCICIO 5:
// Declara una variable global llamada pi con un valor de 3.14159 y otra variable
// local con el mismo nombre en una función (Con diferente valor). Intenta acceder
// a ambas variables desde diferentes partes del programa. ¿Cuál es el
// resultado? Explica el concepto de ámbito y visibilidad.
void local()
{
    // Limpiar pantalla
    system ("CLS");
    // Acceder a la variable global pi
    printf("Global pi: %f\n", pi);
    // Variable local
    float pi = 3.1416;
    // Imprimir la variable local
    printf("Local pi (dentro de local()): %f\n", pi);
    // Pausar
    system ("PAUSE");
}
```

Código para la función externa:

- **Ejercicio 2:**
- **Documentación y Librerías utilizadas**

```
// EJERCICIO 2:
// Declara una variable externa llamada saldo en un archivo fuente (archivo.c) y
// accede a ella desde otro archivo fuente (otroarchivo.c). Modifica su valor en
// ambos archivos y muestra el valor final. ¿Cómo afecta la visibilidad y el tiempo
// de vida de la variable externa?

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

- **Función externa:**

```
// Variable
int saldo = 1000;
// Funcion
void externa(int cantidad)
{
    saldo += cantidad;
    printf ("Saldo: %d\n ", saldo);
    system ("PAUSE");
}
```

Respuestas a las preguntas de cada ejercicio:

1. **Declara una variable automática llamada contador en una función. Incrementa su valor en un bucle y muestra su valor en cada iteración. ¿Qué sucede con la variable al salir de la función? ***

- Deja de existir, ya que cada que se crea una cada que se vuelve a ejecutar.

2. **Declara una variable externa llamada saldo en un archivo fuente (archivo.c) y accede a ella desde otro archivo fuente (otroarchivo.c). Modifica su valor en ambos archivos y muestra el valor final. ¿Cómo afecta la visibilidad y el tiempo de vida de la variable externa?**

- Su tiempo de vida si es global y sigue ahí, si no se utiliza de manera correcta puede que afecte al programa.

3. **Declara una variable estática llamada contador en una función y muestra su valor en cada llamada a la función. ¿Qué sucede con la variable al salir de la función? ¿Cómo difiere de una variable automática? ***

- Conserva su valor para cuando se le vuelva a llamar.

- Que esta si conserva su valor y que se mantiene en todo el programa.

4. **Declara una variable de registro llamada temp y otra automática llamada valor en una función. Compara el acceso y el tiempo de vida de estas variables. ¿Por qué usarías una variable de registro en lugar de una variable automática?**

- Para acceder a la variable muy rápido.

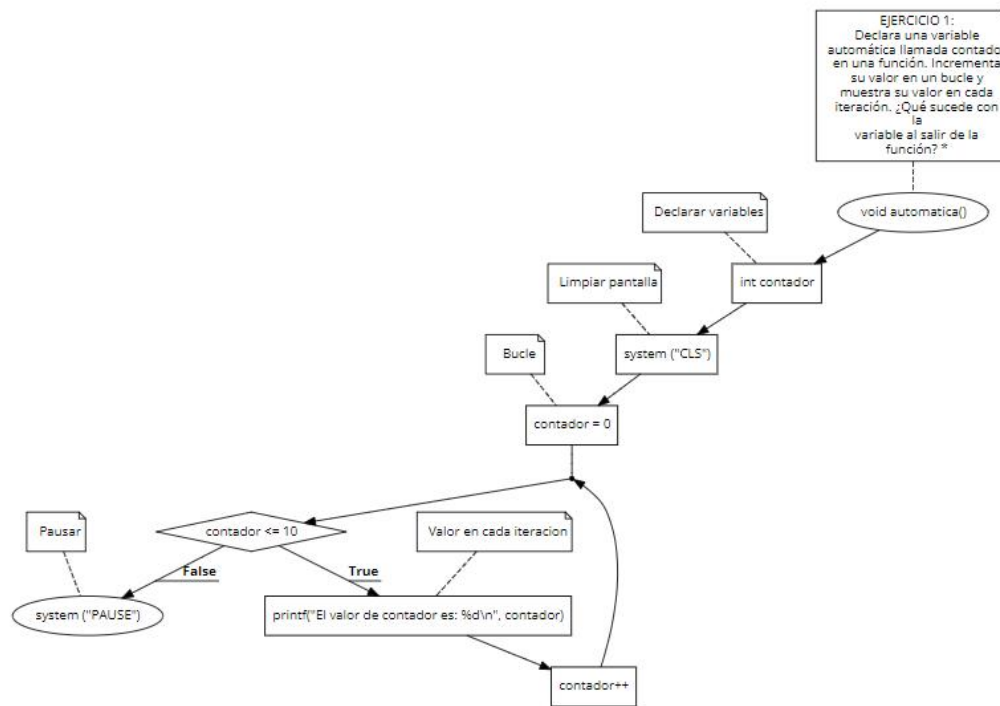
5. **Declara una variable global llamada pi con un valor de 3.14159 y otra variable local con el mismo nombre en una función (Con diferente valor). Intenta acceder a ambas variables desde diferentes partes del programa. ¿Cuál es el resultado? Explica el concepto de ámbito y visibilidad.**

- Ambito, que la variable es valida y se puede usar.

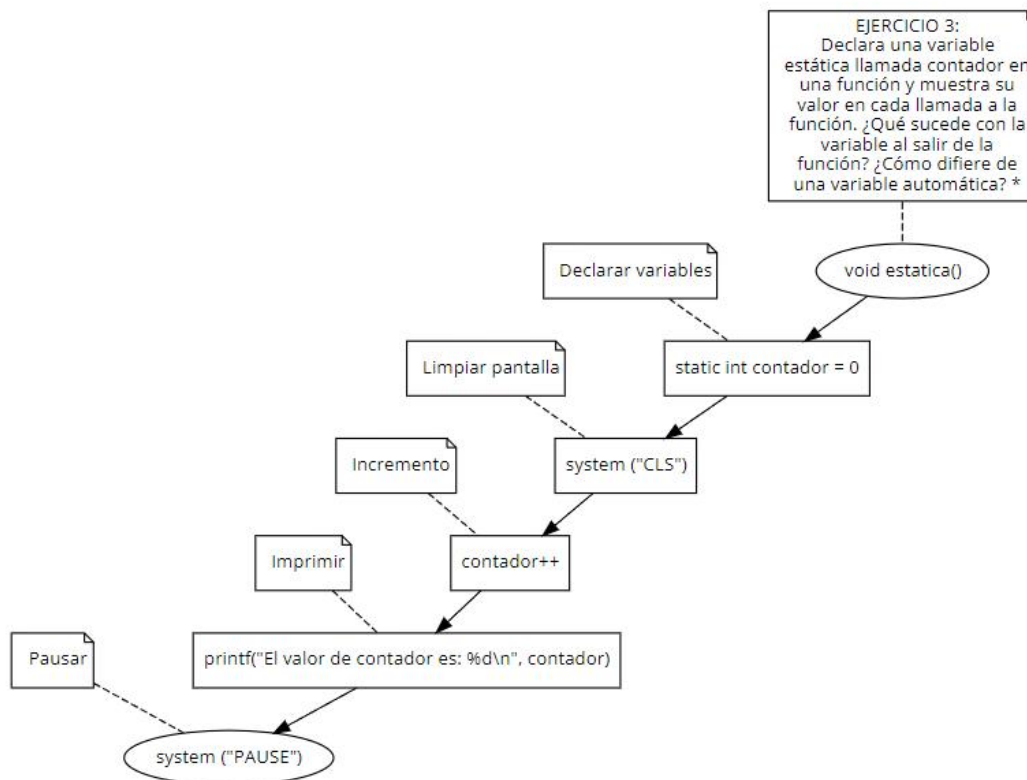
- Visibilidad, que se puede ver desde diferentes partes del programa.

- La local solo es visible dentro de su funcion al contrario de la global.

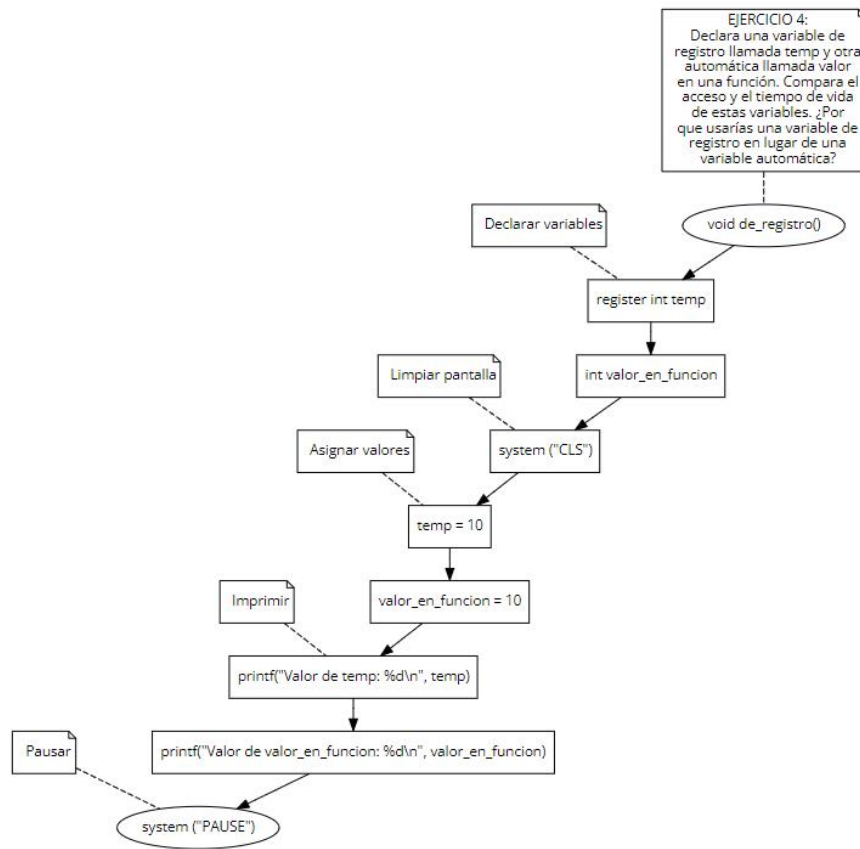
Ejercicio 1:



Ejercicio 3:



Ejercicio 4:



Ejercicio 5:

