UACM

Universidad Autónoma
de la Ciudad de México

NADA HUMANO ME ES AJENO

Introducción a la ingeniería de Software.

Profesor: Raúl Jara.

Práctica 5.

Zavala Pérez Velia.

Semestre 2025-II.

Introducción.

El objetivo de esta práctica fue desarrollar un programa en lenguaje C que simule un editor de texto básico utilizando arreglos tipo buffer, funciones, apuntadores, memoria dinámica (malloc, calloc, realloc y free) y el uso de struct para representar a un grupo de personas.

El sistema permite crear un texto base, modificarlo, capturar datos de múltiples personas (nombre y cargo) y generar documentos personalizados que muestren el texto dirigido automáticamente a cada persona registrada.

Se implementó también un menú controlado mediante sentencia switch, tal como lo exige la práctica, para permitir al usuario elegir entre capturar texto, editarlo, capturar personas, mezclar texto–persona y visualizar los resultados.

Desarrollo.

El programa necesitaba cumplir con estos requisitos principales:

- Crear un texto usando un buffer temporal y luego reservar memoria dinámica con malloc.
- Permitir modificarlo usando realloc.
- Registrar personas con struct, asignando memoria con calloc y luego para cada cadena con malloc.
- Generar documentos personalizados combinando texto + persona.
- Utilizar switch para crear un menú de opciones.
- Liberar toda la memoria antes de salir con free.

Diseño de la Solución.

Se establecieron tres componentes principales:

- Editor de texto básico
- Captura texto.
- Modifica texto.
- Almacena la cadena dinámicamente.

Menú con switch

Maneja las funciones:

- Crear texto
- Modificar texto
- Capturar personas
- Mostrar documentos

- Salir

Algoritmos Usados

- Memoria dinámica:
- malloc: reservar espacio para cadenas.
- calloc: reservar arreglo del struct Persona.
- realloc: modificar el tamaño del texto cuando se edita.
- free: liberar todo al final.

Código fuente.

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

// ------------------------------------------------
//        DEFINICIÓN DEL STRUCT PERSONA
// ------------------------------------------------
typedef struct {
    char *nombre;
    char *cargo;
} Persona;

// Variables globales
char *texto = NULL;
Persona *personas = NULL;
int totalPersonas = 0;

// ------------------------------------------------
//               FUNCION CREAR TEXTO
// ------------------------------------------------
```

```
void crearTexto() {
    char buffer[500];
    printf("\nEscribe el texto base del documento:\n> ");
    fflush(stdin);
    fgets(buffer, sizeof(buffer), stdin);

    texto = (char *) malloc(strlen(buffer) + 1);
    if (texto == NULL) {
        printf("Error al asignar memoria.\n");
        return;
    }
    strcpy(texto, buffer);
    printf("Texto guardado correctamente.\n");
}

// ------------------------------------------------
//          FUNCION MODIFICAR TEXTO (REALLOC)
// ------------------------------------------------
void modificarTexto() {
    if (texto == NULL) {
        printf("\nPrimero debes crear un texto.\n");
        return;
    }

    char buffer[500];
    printf("\nIngresa el nuevo texto:\n> ");
    fflush(stdin);
    fgets(buffer, sizeof(buffer), stdin);
```

```
    texto = (char *) realloc(texto, strlen(buffer) + 1);
    if (texto == NULL) {
        printf("Error al reasignar memoria.\n");
        return;
    }

    strcpy(texto, buffer);
    printf("Texto modificado exitosamente.\n");
}

// ------------------------------------------------
//        CAPTURAR PERSONAS (USA CALLOC)
// ------------------------------------------------
void capturarPersonas() {
    printf("\n¿Cuantas personas deseas registrar? ");
    scanf("%d", &totalPersonas);
    getchar();

    personas = (Persona *) calloc(totalPersonas, sizeof(Persona));

    if (personas == NULL) {
        printf("Error al asignar memoria.\n");
        return;
    }

    for (int i = 0; i < totalPersonas; i++) {
        char bufferNombre[100];
```

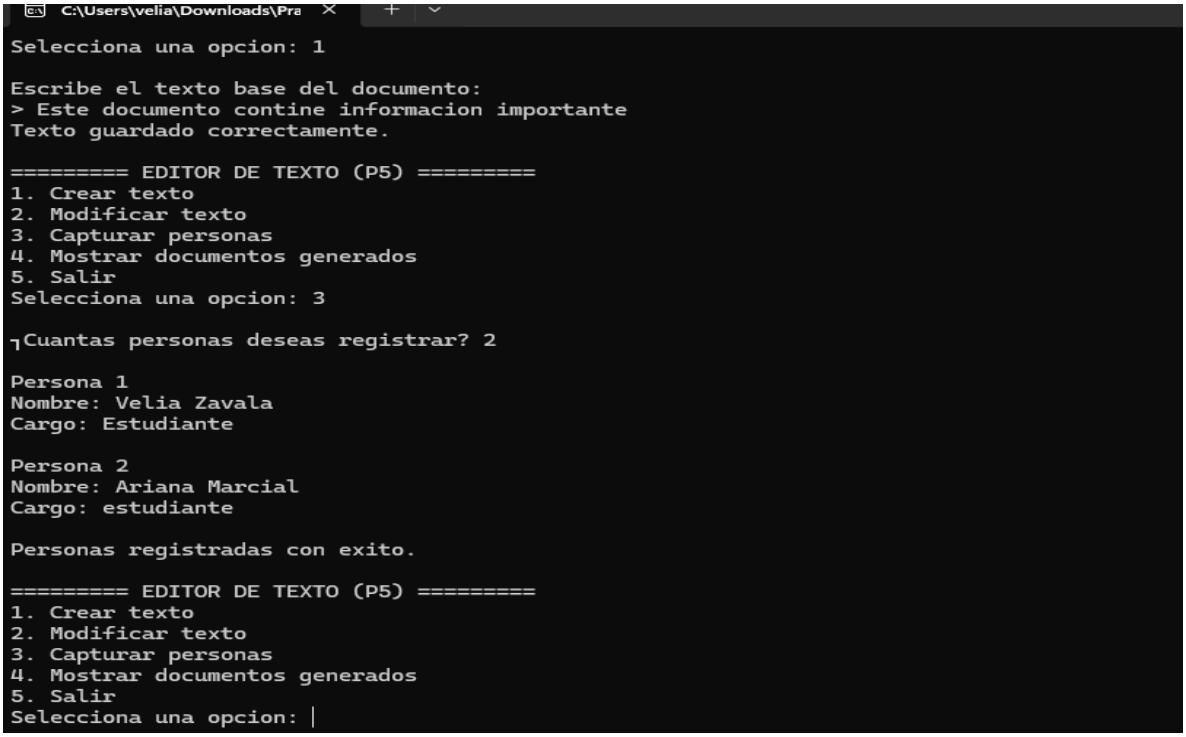
```
        char bufferCargo[100];

        printf("\nPersona %d\n", i + 1);

        printf("Nombre: ");
        fgets(bufferNombre, sizeof(bufferNombre), stdin);
        bufferNombre[strcspn(bufferNombre, "\n")] = 0;

        printf("Cargo: ");
        fgets(bufferCargo, sizeof(bufferCargo), stdin);
        bufferCargo[strcspn(bufferCargo, "\n")] = 0;

        personas[i].nombre = (char *) malloc(strlen(bufferNombre) + 1);
        personas[i].cargo  = (char *) malloc(strlen(bufferCargo) + 1);

        strcpy(personas[i].nombre, bufferNombre);
        strcpy(personas[i].cargo, bufferCargo);
    }

    printf("\nPersonas registradas con exito.\n");
}

// ------------------------------------------------
//         MEZCLAR TEXTO + PERSONA (CREAR DOC)
// ------------------------------------------------
void mostrarDocumentos() {
    if (texto == NULL || personas == NULL) {
        printf("\nDebes capturar texto y personas primero.\n");
```

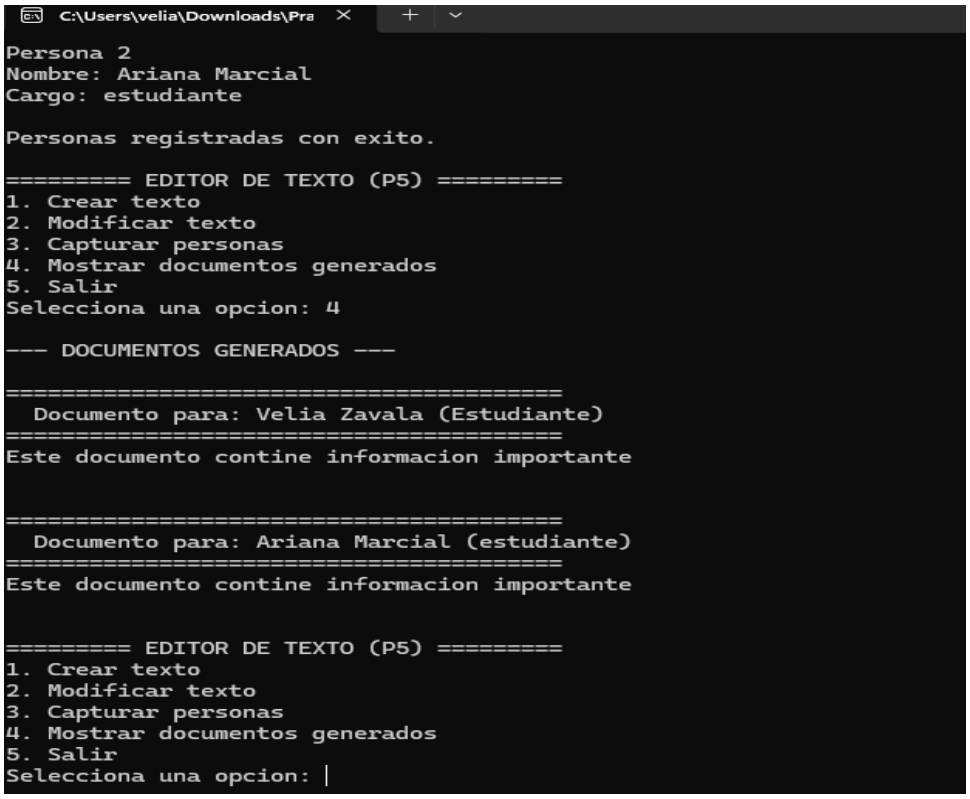
```
        return;
    }

    printf("\n--- DOCUMENTOS GENERADOS ---\n");

    for (int i = 0; i < totalPersonas; i++) {
        printf("\n========================================\n");
        printf("  Documento para: %s (%s)\n", personas[i].nombre, personas[i].cargo);
        printf("========================================\n");
        printf("%s\n", texto);
    }
}

// ------------------------------------------------
//              LIBERAR MEMORIA
// ------------------------------------------------
void liberarMemoria() {
    if (texto != NULL)
        free(texto);

    if (personas != NULL) {
        for (int i = 0; i < totalPersonas; i++) {
            free(personas[i].nombre);
            free(personas[i].cargo);
        }
        free(personas);
    }
}
```

```
// ------------------------------------------------
//              MENU PRINCIPAL (SWITCH)
// ------------------------------------------------
int main() {
    int opcion;

    do {
        printf("\n========= EDITOR DE TEXTO (P5) =========\n");
        printf("1. Crear texto\n");
        printf("2. Modificar texto\n");
        printf("3. Capturar personas\n");
        printf("4. Mostrar documentos generados\n");
        printf("5. Salir\n");
        printf("Selecciona una opcion: ");
        scanf("%d", &opcion);
        getchar();

        switch (opcion) {
            case 1:
                crearTexto();
                break;
            case 2:
                modificarTexto();
                break;
            case 3:
                capturarPersonas();
                break;
```

```
            case 4:
                mostrarDocumentos();
                break;
            case 5:
                liberarMemoria();
                printf("Saliendo...\n");
                break;
            default:
                printf("Opcion no valida.\n");
        }

    } while (opcion != 5);

    return 0;
}
```

Evidencia.





Actividad	Tiempo estimado	Tiempo real
Análisis	**1 hora**	**30 minutos**
Diseño	**40 minutos**	**30 minutos**
Codificación	**2 horas**	**90 minutos**
Pruebas	**1 hora**	**40 minutos**
Documentación	**1 hora**	**40 minutos**
Total	**5 horas 40 minutos**	**3 horas 50 minutos**

Conclusiones.

La práctica permitió reforzar conceptos clave del manejo de memoria dinámica en C, uso de buffers, cadenas, apuntadores y estructuras. El uso combinado de malloc, calloc, realloc y free permitió comprender la correcta administración de recursos.

El manejo de struct facilitó el registro y gestión de información de múltiples personas, lo que permitió generar documentos personalizados. El uso del menú con switch estructuró de manera clara la ejecución del programa.

El proyecto quedó preparado para futuras funcionalidades e integración completa con Git.

El aprendizaje obtenido es fundamental para el desarrollo de software en C y para la construcción de programas modulados y escalables.