



**Universidad Tres Culturas**



**PLANTEL “LONDRES”**

**Ingeniería en Sistemas Computacionales**

**Examen 2° Parcial: Estructura de  
Datos**

**Herramientas**

**Presentan:**

*García Salas Natalia*

**Grupo: “A”**

**Turno: Matutino**

**Docente: José Guadalupe Sánchez  
Hernández**

**Asignatura: Estructura de Datos**

**Fecha de entrega: 15 de julio de 2024.**

# Código

//Programa hecho por @Natalia García  
//15/07/24

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

```
struct herramienta {
    int id;
    char nombre[50];
    int cantidad;
    float costo;
};
```

```
// Prototipos de funciones
int intOpcion(void);
void crearArchivo(void);
void nuevoRegistro(void);
void actualizaRegistro(void);
void eliminarRegistro(void);
void imprimirRegistro(void);
```

```
int main() {
    int eleccion;
    system("cls");
    do {
        switch (eleccion) {
            case 1:
                crearArchivo();
                break;
            case 2:
                actualizaRegistro();
                break;
            case 3:
                nuevoRegistro();
                break;
            case 4:
                eliminarRegistro();
                break;
            case 5:
                imprimirRegistro();
                break;
            default:
                if (eleccion < 1 || eleccion > 6) {
                    printf("Opción incorrecta\n");
                }
        }
    } while (1);
}
```

```
    }
    break;
}
} while ((eleccion = intOpcion()) != 6);
return 0;
}

void crearArchivo(void) {
    FILE *ptrArchivo;
    struct herramienta vacia = {0, "", 0, 0.0};

    if ((ptrArchivo = fopen("herram.dat", "wb")) == NULL) {
        printf("No pudo abrirse el archivo herram.dat.\n");
        return;
    }

    for (int i = 0; i < 100; i++) {
        fwrite(&vacia, sizeof(struct herramienta), 1, ptrArchivo);
    }

    fclose(ptrArchivo);
    printf("Archivo inicializado con 100 registros vacíos.\n");
}

void nuevoRegistro(void) {
    FILE *ptrArchivo;
    struct herramienta nuevaHerramienta;

    if ((ptrArchivo = fopen("herram.dat", "rb+")) == NULL) {
        printf("No pudo abrirse el archivo herram.dat.\n");
        return;
    }

    printf("Ingrese ID: ");
    scanf("%d", &nuevaHerramienta.id);
    printf("Ingrese Nombre: ");
    scanf("%s", nuevaHerramienta.nombre);
    printf("Ingrese la cantidad: ");
    scanf("%d", &nuevaHerramienta.cantidad);
    printf("Ingrese el costo: ");
    scanf("%f", &nuevaHerramienta.cost);

    fseek(ptrArchivo, (nuevaHerramienta.id - 1) * sizeof(struct herramienta), SEEK_SET);
    fwrite(&nuevaHerramienta, sizeof(struct herramienta), 1, ptrArchivo);
    fclose(ptrArchivo);
}

void actualizaRegistro(void) {
```

```
FILE *ptrArchivo;
struct herramienta herramientaExistente;
int id;

if ((ptrArchivo = fopen("herram.dat", "rb+")) == NULL) {
    printf("No pudo abrirse el archivo herram.dat.\n");
    return;
}

printf("Ingrese el ID del registro a actualizar: ");
scanf("%d", &id);

fseek(ptrArchivo, (id - 1) * sizeof(struct herramienta), SEEK_SET);
fread(&herramientaExistente, sizeof(struct herramienta), 1, ptrArchivo);

if (herramientaExistente.id == 0) {
    printf("No hay información en el registro.\n");
} else {
    printf("Ingrese la nueva Herramienta: ");
    scanf("%s", herramientaExistente.nombre);
    printf("Ingrese la nueva cantidad: ");
    scanf("%d", &herramientaExistente.cantidad);
    printf("Ingrese nueva edad: ");
    scanf("%f", &herramientaExistente.costeo);

    fseek(ptrArchivo, (id - 1) * sizeof(struct herramienta), SEEK_SET);
    fwrite(&herramientaExistente, sizeof(struct herramienta), 1, ptrArchivo);
}

fclose(ptrArchivo);
}

void eliminarRegistro(void) {
    FILE *ptrArchivo;
    struct herramienta vacia = {0,"",0 ,0.0};
    int id;

    if ((ptrArchivo = fopen("herram.dat", "rb+")) == NULL) {
        printf("No pudo abrirse el archivo herram.dat.\n");
        return;
    }

    printf("Ingrese el ID del registro a eliminar: ");
    scanf("%d", &id);

    fseek(ptrArchivo, (id - 1) * sizeof(struct herramienta), SEEK_SET);
    fwrite(&vacio, sizeof(struct herramienta), 1, ptrArchivo);
}
```

```
fclose(ptrArchivo);
}

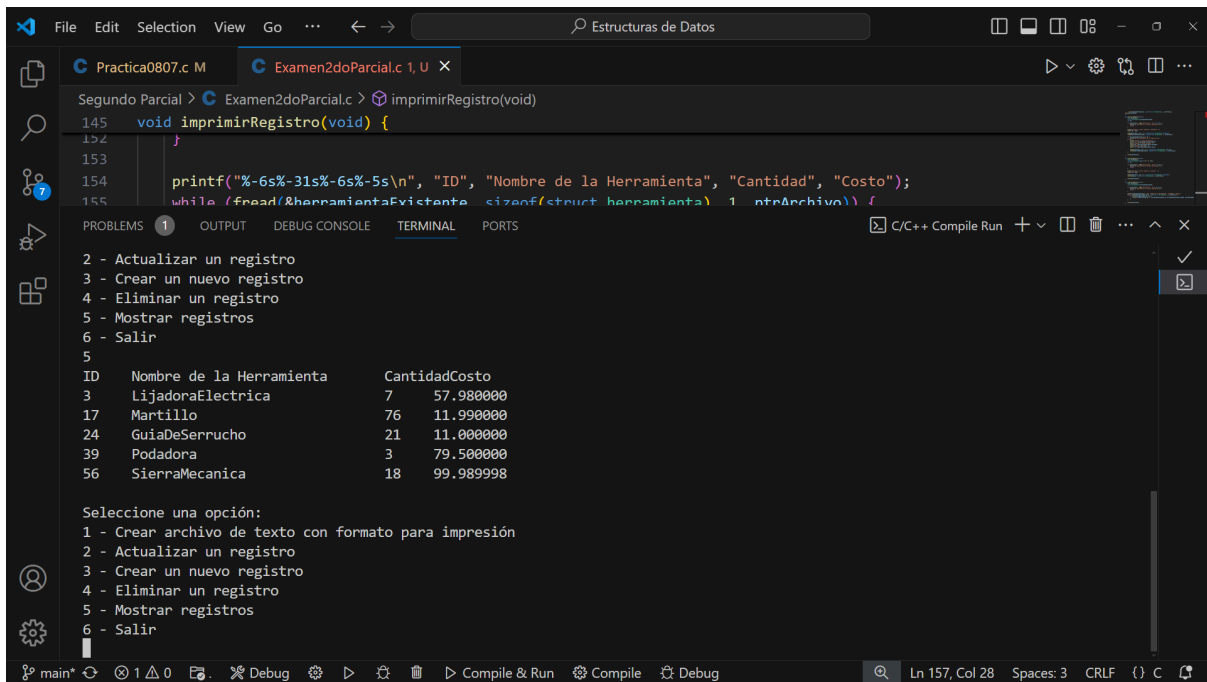
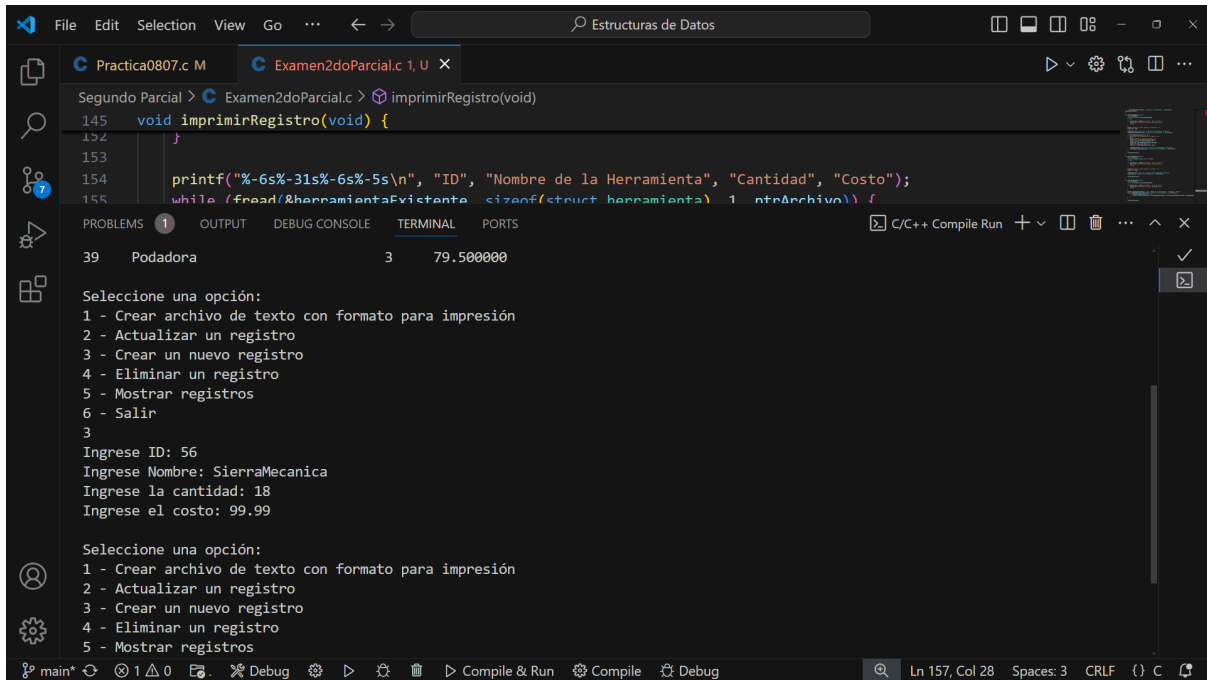
void imprimirRegistro(void) {
    FILE *ptrArchivo;
    struct herramienta herramientaExistente;

    if ((ptrArchivo = fopen("herram.dat", "rb")) == NULL) {
        printf("No pudo abrirse el archivo herram.dat.\n");
        return;
    }

    printf("%-6s%-31s%-6s%-5s\n", "ID", "Nombre de la Herramienta", "Cantidad", "Costo");
    while (fread(&herramientaExistente, sizeof(struct herramienta), 1, ptrArchivo)) {
        if (herramientaExistente.id != 0) {
            printf("%-6d%-31s%-6d%-5f\n", herramientaExistente.id,
herramientaExistente.nombre, herramientaExistente.cantidad, herramientaExistente.costo);
        }
    }

    fclose(ptrArchivo);
}

int intOpcion(void) {
    int opcion;
    printf("\nSeleccione una opción:\n");
    printf("1 - Crear archivo de texto con formato para impresión\n");
    printf("2 - Actualizar un registro\n");
    printf("3 - Crear un nuevo registro\n");
    printf("4 - Eliminar un registro\n");
    printf("5 - Mostrar registros\n");
    printf("6 - Salir\n");
    scanf("%d", &opcion);
    return opcion;
}
```



[illegible]

The screenshot shows a Visual Studio Code editor with a C++ project named 'Practica0807.c M'. The active file is 'Examen2doParcial.c', which contains a program for managing a tool inventory. The code includes a menu for creating, updating, deleting, and displaying records. The terminal output shows the program's execution, displaying a list of tools with their IDs, names, and costs.

```
Segundo Parcial > Examen2doParcial.c > imprimirRegistro(void)
145 void imprimirRegistro(void) {
152 }
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

2 - Actualizar un registro  
3 - Crear un nuevo registro  
4 - Eliminar un registro  
5 - Mostrar registros  
6 - Salir  
5

ID	Nombre de la Herramienta	Cantidad	Costo
3	LijadoraElectrica	7	57.980000
17	Martillo	76	11.990000
24	GuiaDeSerrucho	21	11.000000
56	SierraMecanica	18	99.989998
68	Destornillador	106	6.990000
77	Mazo	11	21.500000
83	LlaveInglesa	34	7.500000

Seleccione una opción:  
1 - Crear archivo de texto con formato para impresión  
2 - Actualizar un registro  
3 - Crear un nuevo registro  
4 - Eliminar un registro  
5 - Mostrar registros  
6 - Salir

main\* 0 Debug Compile & Run Compile Debug Ln 157, Col 28 Spaces: 3 CRLF {} C