



Universidad Tres Culturas



PLANTEL “LONDRES”

Ingeniería en Sistemas Computacionales

Reporte de Prácticas
Del 15 al 19 de julio.

Presenta:

García Salas Natalia

Grupo: “A”

Turno: Matutino

Docente: José Guadalupe Sánchez
Hernández

Asignatura: Estructura de Datos

Fecha de entrega: 20 de julio de 2024.

Introducción	1
Desarrollo	1
1. CRUD de medicamentos de Farmacia	1
Diagrama de Flujo	1
Código	2
Salida de escritorio	7
Documentación por bloques de código	9
2. Examen	10
Diagrama de flujo	10
Código	12
Salida de Escritorio	16
Documentación por bloques	17

Introducción

Como estudiante de Ingeniería en Sistemas Computacionales es importante poner en práctica los conocimientos teóricos aprendidos en el aula de clases. Es por eso que se realizarán los siguientes ejercicios, para reforzar los conocimientos adquiridos en niveles anteriores de la carrera.

Desarrollo

1. CRUD de medicamentos de Farmacia

Diagrama de Flujo

Inicio -> Mostrar Menú

|

v

Seleccionar Opción

|

+--> (1) Crear Archivo

|

|

|

|

|

|

|

|

+--> (2) Actualizar Registro

|

|

|

|

|

|

|

|

```
| Actualizar Registro
| |
| v
| Menú
|
+--> (3) Crear Nuevo Registro
| |
| v
| Crear Nuevo Registro
| |
| v
| Menú
|
+--> (4) Eliminar Registro
| |
| v
| Eliminar Registro
| |
| v
| Menú
|
+--> (5) Mostrar Registros
| |
| v
| Mostrar Registros
| |
| v
| Menú
|
+--> (6) Salir
|
v
Fin
```

Código

//Programa hecho por @Natalia Garcia

//18/07/24

/*

Usted es el dueño de una Farmacia y necesita mantener un inventario que le pueda decir cuáles medicamentos tiene, cuántos tiene y el costo de cada uno.

Escriba un programa que inicialice el archivo "Medicat.dat" con 100 registros vacíos, que le permita introducir los datos relacionados con cada medicamento, que le permita listar todos los medicamentos, que le permita eliminar un registro de uno de ellos que ya no tiene, y que le permita actualizar

cualquier información en el archivo. El número de identificación de cada medicamento debe ser su número de registro comenzando con el 00 y terminando con el 99.

*/

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

```
struct medicamento {
    int id;
    char nombre[30];
    int cantidad;
    float costo;
};
```

```
int intOpcion(void);
void crearArchivo(void);
void nuevoRegistro(void);
void actualizaRegistro(void);
void eliminarRegistro(void);
void mostrarRegistro(void);
```

```
int main() {
    int eleccion;
    system("cls");
    do {
        switch (eleccion) {
            case 1:
                system("cls");
                crearArchivo();
                break;
            case 2:
                system("cls");
                actualizaRegistro();
                break;
            case 3:
                system("cls");
                nuevoRegistro();
                break;
            case 4:
                system("cls");
                eliminarRegistro();
                break;
            case 5:
                system("cls");
                mostrarRegistro();
                break;
```

```
default:
    if (eleccion < 1 || eleccion > 6) {
        printf("Opción incorrecta\n");
    }
    break;
}
} while ((eleccion = intOpcion()) != 6);
return 0;
}

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

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

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

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

void nuevoRegistro(void) {
    FILE *ptrArchivo;
    struct medicamento nuevoMedicamento;

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

    printf("Ingrese ID (0-99): ");
    scanf("%d", &nuevoMedicamento.id);
    printf("Ingrese el nombre del medicamento: ");
    scanf("%s", nuevoMedicamento.nombre);
    printf("Ingrese la cantidad: ");
    scanf("%d", &nuevoMedicamento.cantidad);
    printf("Ingrese el costo: ");
    scanf("%f", &nuevoMedicamento.costos);

    fseek(ptrArchivo, nuevoMedicamento.id * sizeof(struct medicamento), SEEK_SET);
    fwrite(&nuevoMedicamento, sizeof(struct medicamento), 1, ptrArchivo);
}
```

```
    fclose(ptrArchivo);
}

void actualizaRegistro(void) {
    FILE *ptrArchivo;
    struct medicamento medicamentoExistente;
    int id;

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

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

    fseek(ptrArchivo, id * sizeof(struct medicamento), SEEK_SET);
    fread(&medicamentoExistente, sizeof(struct medicamento), 1, ptrArchivo);

    if (medicamentoExistente.id == 0) {
        printf("No hay información en el registro.\n");
    } else {
        printf("Ingrese nuevo nombre del medicamento: ");
        scanf("%s", medicamentoExistente.nombre);
        printf("Ingrese nueva cantidad: ");
        scanf("%d", &medicamentoExistente.cantidad);
        printf("Ingrese nuevo costo: ");
        scanf("%f", &medicamentoExistente.costos);

        fseek(ptrArchivo, id * sizeof(struct medicamento), SEEK_SET);
        fwrite(&medicamentoExistente, sizeof(struct medicamento), 1, ptrArchivo);
    }

    fclose(ptrArchivo);
}

void eliminarRegistro(void) {
    FILE *ptrArchivo;
    struct medicamento medicamentoExistente;
    struct medicamento vacio = {0, "", 0, 0.0};
    int id;

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

    printf("Ingrese el ID del medicamento a eliminar: ");
```

```
scanf("%d", &id);

fseek(ptrArchivo, id * sizeof(struct medicamento), SEEK_SET);
fread(&medicamentoExistente, sizeof(struct medicamento), 1, ptrArchivo);

if (medicamentoExistente.cantidad != 0) {
    printf("No se puede eliminar el registro.\n Aún tiene productos.\n");
} else {
    fseek(ptrArchivo, id * sizeof(struct medicamento), SEEK_SET);
    fwrite(&vacio, sizeof(struct medicamento), 1, ptrArchivo);
}

fclose(ptrArchivo);
}

void mostrarRegistro(void) {
    FILE *ptrArchivo;
    struct medicamento medicamentoExistente;

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

    printf("%-6s%-30s%-10s%-10s\n", "ID", "Nombre", "Cantidad", "Costo");
    while (fread(&medicamentoExistente, sizeof(struct medicamento), 1, ptrArchivo)) {
        if (medicamentoExistente.id != 0) {
            printf("%-6d%-30s%-10d  $%-10.2f\n", medicamentoExistente.id,
medicamentoExistente.nombre,
medicamentoExistente.cantidad,
medicamentoExistente.costos);
        }
    }

    fclose(ptrArchivo);
}

int intOpcion(void) {
    int opcion = 0;
    printf("\nSeleccione una opción:\n");
    printf("1. Crear el archivo de texto\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;
}
```

The screenshot shows a Windows desktop with two main applications open: a web browser and a C++ IDE (Visual Studio).

Web Browser (Left): The browser displays a web application titled "ESTRUCTURA DE...". The page content includes a description of the pharmacy inventory system, a table of medications, and a user profile section.

Registr o	Nombre medicamento	Cantidad en stock	Precio unitario
00	Paracetamol 500 mg	100	\$25.50
01	Ibuprofeno 800 mg	200	\$35.60
02	Acetaminofen	40	\$99.00
03	Acido acetilsalicilico	50	\$102.00
...			
99	-----	22	\$22.00

The user profile section shows the name "Jose Guadalupe Sanchez Hernandez" and the date "Jueves 08:14". Below this, it says "Clase 18 julio 24 finalizó".

C++ IDE (Right): The IDE shows a C++ program named "Farmacia.c 2.0". The code defines a struct for medications and implements functions for creating, updating, and deleting records. The terminal window shows the program's output, including the list of medications and their prices.

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

struct medicamento {
    int id;
    char nombre[30];
    int cantidad;
    float costo;
};

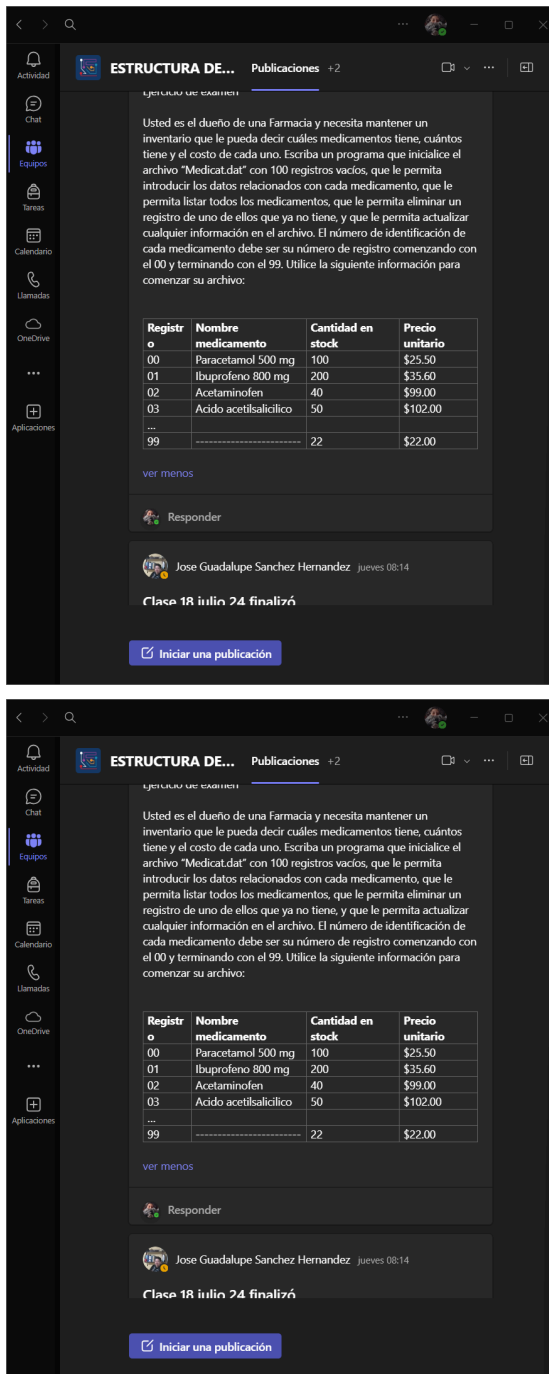
int intopcion(void);
void crearArchivo(void);
void nuevoRegistro(void);
void actualizaRegistro(void);

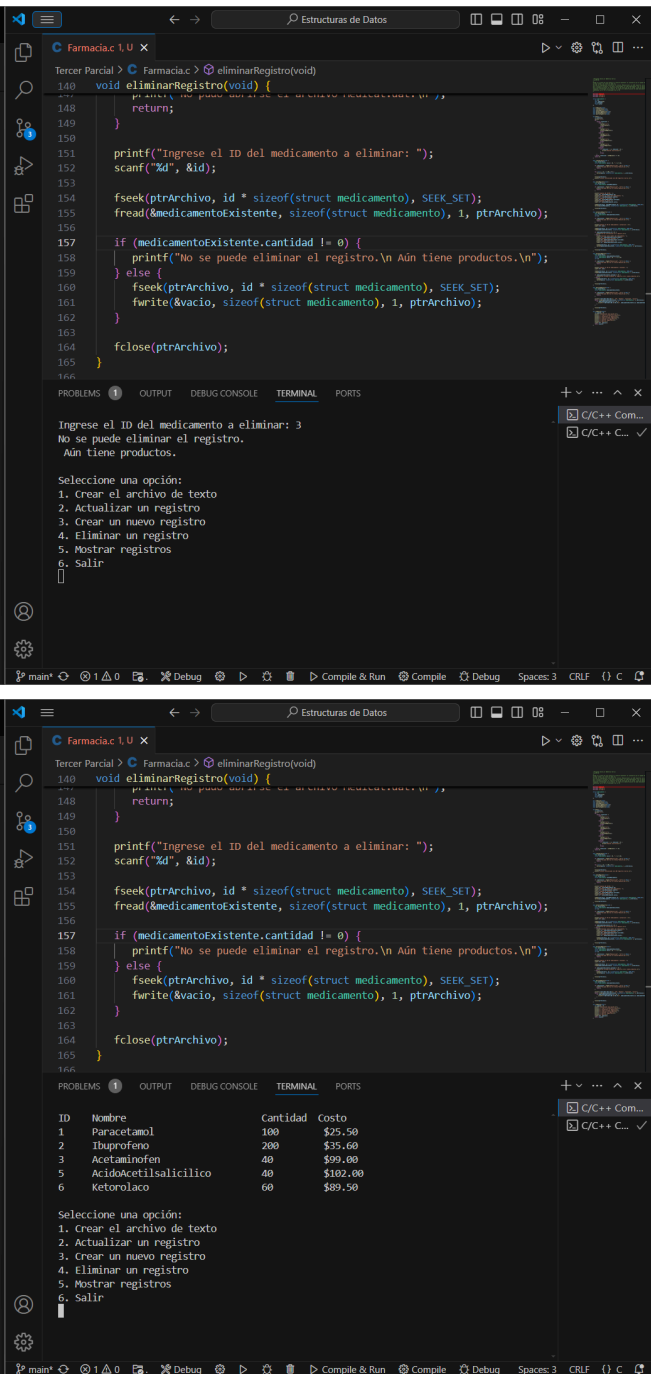
int main() {
    int opcion;
    do {
        intopcion(&opcion);
        switch (opcion) {
            case 1: crearArchivo(); break;
            case 2: nuevoRegistro(); break;
            case 3: actualizaRegistro(); break;
            case 4: break;
        }
    } while (opcion != 4);
    return 0;
}
```

The terminal output shows the program's execution, including the list of medications and their prices.

```
Ingrese ID (0-99): 5
Ingrese el nombre del medicamento: AcidoAcetilsalicilico
Ingrese la cantidad: 40
Ingrese el costo: 102

Seleccione una opción:
1. Crear el archivo de texto
2. Actualizar un registro
3. Crear un nuevo registro
4. Eliminar un registro
5. Mostrar registros
6. Salir
6
```



Documentación por bloques de código

- Bloque Verde

Declaración de variables

- Bloque Rojo

Lectura de datos

V

```

scanf("%d", &id)      |
|                      |
V                      |
fseek()               |
|                      |
V                      |
fwrite()              |
|                      |
V                      |
fclose()              |
|                      |
V                      |
actualizaRegistro()   |
|                      |
V                      |
fopen()               |
|                      |
V                      |
printf("Ingrese ID del |
registro a actualizar: ")|
|                      |
V                      |
fseek()               |
|                      |
V                      |
fread()               |
|                      |
V                      |
if id != 0            |
|                      |
V                      |
printf("Ingrese nueva  |
herramienta: ")        |
|                      |
V                      |
scanf("%s", nombre)    |
|                      |
V                      |
fwrite()               |
|                      |
V                      |
fclose()               |
|                      |
V                      |
eliminarRegistro()     |
|                      |
V                      |
fopen()                |

```

```
|
V
printf("Ingrese ID del |
registro a eliminar: ") |
|
V
fseek() |
|
V
fwrite() |
|
V
fclose() |
|
V
imprimirRegistro() |
|
V
fopen() |
|
V
printf("%-6s%-31s%-6s%-5s\n")|
|
V
while fread() |
|
V
printf("%-6d%-31s%-6d%-5f\n")|
|
V
fclose() |
|
V
Salir
```

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");
                }
                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(&vacía, 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.costos);

    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.costos);
    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) {
```



```

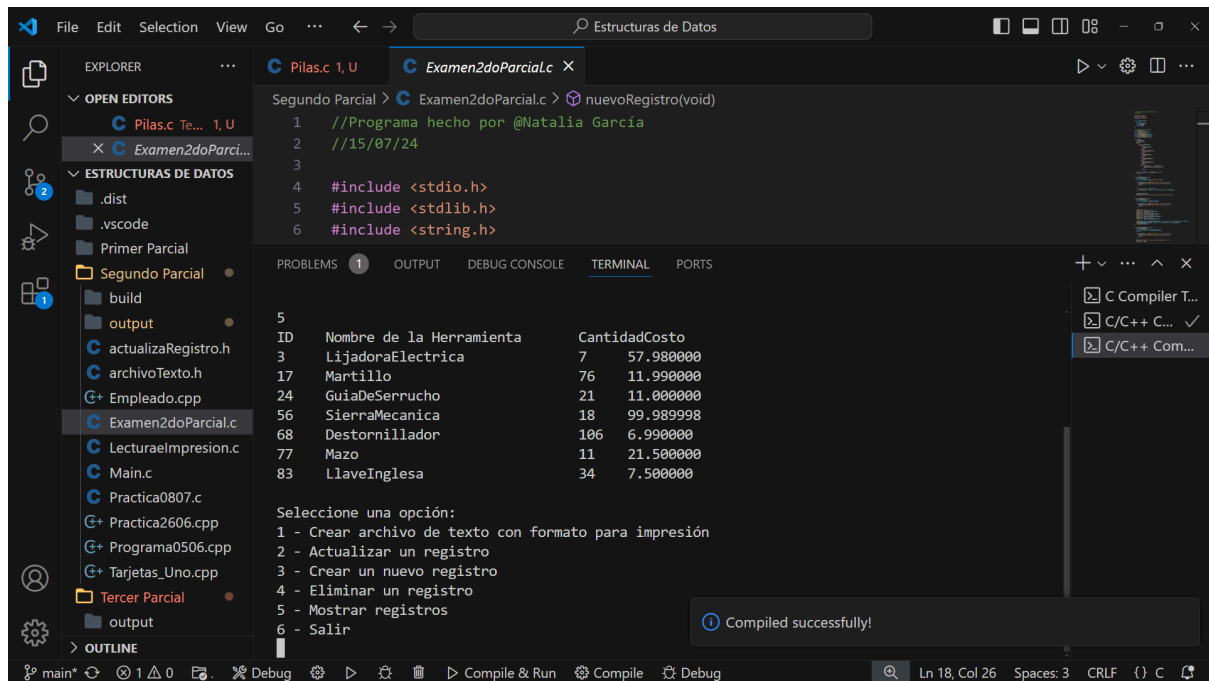
        printf("%-6d%-31s%-6d%-5f\n",    herramientaExistente.id,
herramientaExistente.nombre, herramientaExistente.cantidad, herramientaExistente.costo);
    }
}

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;
}

```

Salida de Escritorio



The screenshot shows the Visual Studio Code interface. The Explorer pane on the left displays the project structure, including folders like 'Segundo Parcial' and 'Tercer Parcial', and files like 'Examen2doParcial.c'. The main editor shows the source code of 'Examen2doParcial.c', which includes headers for `stdio.h`, `stdlib.h`, and `string.h`. The Output pane at the bottom shows the program's output, which includes a table of tools and a menu of options.

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

Compiled successfully!

Documentación por bloques

- Bloque Verde

Declaración de variables

- Bloque Rojo

Lectura de datos

- Bloque Azul

Operaciones

- Bloque Naranja

Resultado final