



**Universidad Tres Culturas**



**PLANTEL “LONDRES”**

**Ingeniería en Sistemas Computacionales**

**Reporte de Prácticas**  
**Del 8 al 12 de julio.**

**Presenta:**

*García Salas Natalia*

**Grupo: “A”**

**Turno: Matutino**

**Docente: José Guadalupe Sánchez  
Hernández**

**Asignatura: Estructura de Datos**

**Fecha de entrega: 13 de julio de 2024.**

<b>Introducción</b>	<b>1</b>
<b>Desarrollo</b>	<b>1</b>
1. CRUD de datos de personas en un archivo	1
Diagrama de Flujo	1
Código	3
Salida de escritorio	7
Documentación por bloques de código	10

## 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 datos de personas en un archivo

#### Diagrama de Flujo

Inicio

|

v

Mostrar menú

|

v

Leer opción del usuario

|

v

```

-----
|   Switch de opciones   |
|-----|
| 1. Crear archivo      |
| |                     |
| v                     |
| Llamar a              |
| crearArchivo()        |
| |                     |
| v                     |
| Volver al menú        |
|                       |

```

```
| 2. Actualizar registro |
| | |
| v |
| Llamar a |
| actualizaRegistro() |
| | |
| v |
| Volver al menú |
| |
| 3. Crear nuevo registro |
| | |
| v |
| Llamar a |
| nuevoRegistro() |
| | |
| v |
| Volver al menú |
| |
| 4. Eliminar registro |
| | |
| v |
| Llamar a |
| eliminarRegistro() |
| | |
| v |
| Volver al menú |
| |
| 5. Mostrar registros |
| | |
| v |
| Llamar a |
| imprimirRegistro() |
| | |
| v |
| Volver al menú |
| |
| 6. Salir |
| | |
| v |
| Terminar programa |
| |
| Otra opción |
| | |
| v |
| Mostrar mensaje de |
| error |
| | |
| v |
```

| Volver al menú |

-----  
|  
v  
Fin

## Código

//Programa hecho por @Natalia García

//05/07/24

/\*

Desarrollar el siguiente programa.

Escriba instrucciones que realicen cada una de las siguientes tareas. Suponga que se definió la estructura

```
struct persona {
```

```
    //Se le puede agregar un id
```

```
    char apellido[15];
```

```
    char nombre[15];
```

```
    char edad[4];
```

```
};
```

deben crear el archivo para escritura, y realizar las siguientes tareas.

a) Inicialice el archivo "nomedad.dat" de manera que existan 100 registros con apellido = "sin-asignar", nombre= "" y edad= "0".

b) Introduzca 10 apellidos, nombres y edades, y escríbalos en el archivo.

c) Actualice un registro; si no existe información en el registro, indique al usuario que "No hay información".

d) Elimine un registro que tenga información, por medio de la reinicialización de dicho registro en particular.

Cada inciso debe estar contenido en una función diferente.

\*/

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

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

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

```
struct persona {
```

```
    int id;
```

```
    char apellido[15];
```

```
    char nombre[15];
```

```
    char edad[4];
```

```
};
```

```
// 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 persona vacia = {0, "sin-asignar", "", "0"};

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

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

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

void nuevoRegistro(void) {
    FILE *ptrArchivo;
    struct persona nuevaPersona;

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

    //for (int i = 0; i < 10; i++) {
        printf("Ingreso ID: ");
        scanf("%d", &nuevaPersona.id);
        printf("Ingreso apellido: ");
        scanf("%s", nuevaPersona.apellido);
        printf("Ingreso nombre: ");
        scanf("%s", nuevaPersona.nombre);
        printf("Ingreso edad: ");
        scanf("%s", nuevaPersona.edad);

        fseek(ptrArchivo, (nuevaPersona.id - 1) * sizeof(struct persona), SEEK_SET);
        fwrite(&nuevaPersona, sizeof(struct persona), 1, ptrArchivo);
    //}

    fclose(ptrArchivo);
}

void actualizaRegistro(void) {
    FILE *ptrArchivo;
    struct persona personaExistente;
    int id;

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

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

    fseek(ptrArchivo, (id - 1) * sizeof(struct persona), SEEK_SET);
    fread(&personaExistente, sizeof(struct persona), 1, ptrArchivo);

    if (personaExistente.id == 0) {
```

```
printf("No hay información en el registro.\n");
} else {
    printf("Ingrese nuevo apellido: ");
    scanf("%s", personaExistente.apellido);
    printf("Ingrese nuevo nombre: ");
    scanf("%s", personaExistente.nombre);
    printf("Ingrese nueva edad: ");
    scanf("%s", personaExistente.edad);

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

fclose(ptrArchivo);
}

void eliminarRegistro(void) {
    FILE *ptrArchivo;
    struct persona vacia = {0, "sin-asignar", "", "0"};
    int id;

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

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

    fseek(ptrArchivo, (id - 1) * sizeof(struct persona), SEEK_SET);
    fwrite(&vacía, sizeof(struct persona), 1, ptrArchivo);

    fclose(ptrArchivo);
}

// Mostrar registros
void imprimirRegistro(void) {
    FILE *ptrArchivo;
    struct persona personaExistente;

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

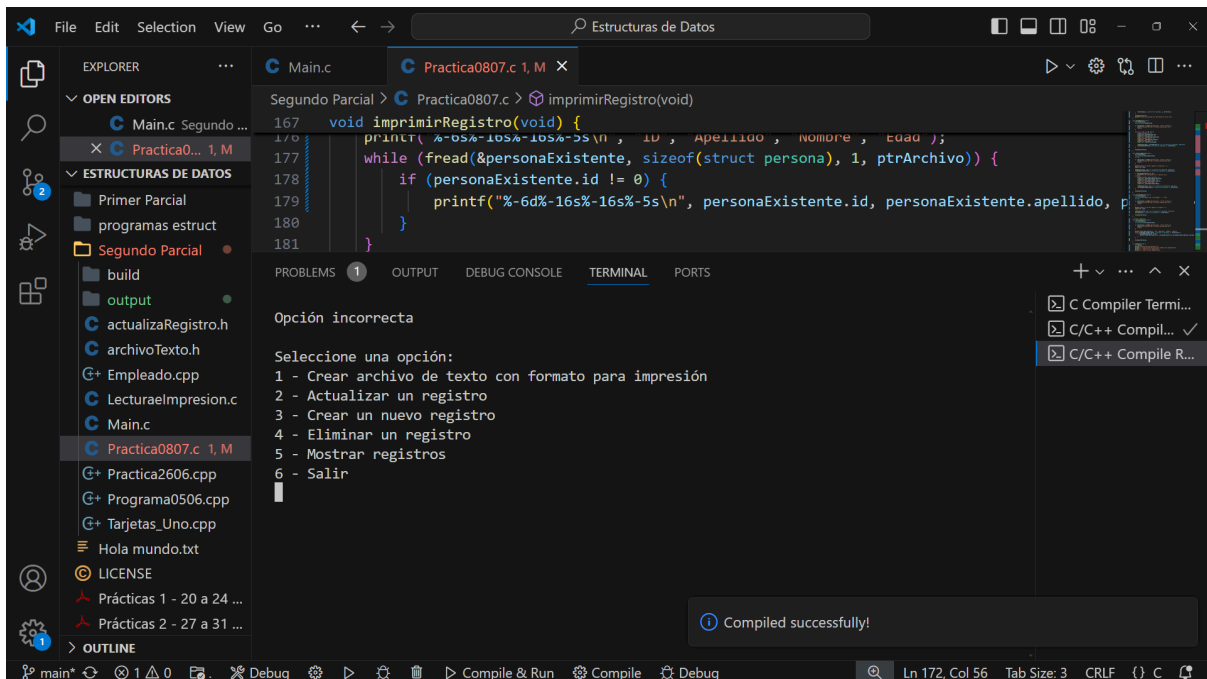
    printf("%-6s%-16s%-16s%-5s\n", "ID", "Apellido", "Nombre", "Edad");
    while (fread(&personaExistente, sizeof(struct persona), 1, ptrArchivo)) {
        if (personaExistente.id != 0) {
```

```
printf("%-6d%-16s%-16s%-5s\n", personaExistente.id, personaExistente.apellido,
personaExistente.nombre, personaExistente.edad);
}
}

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;
}
```

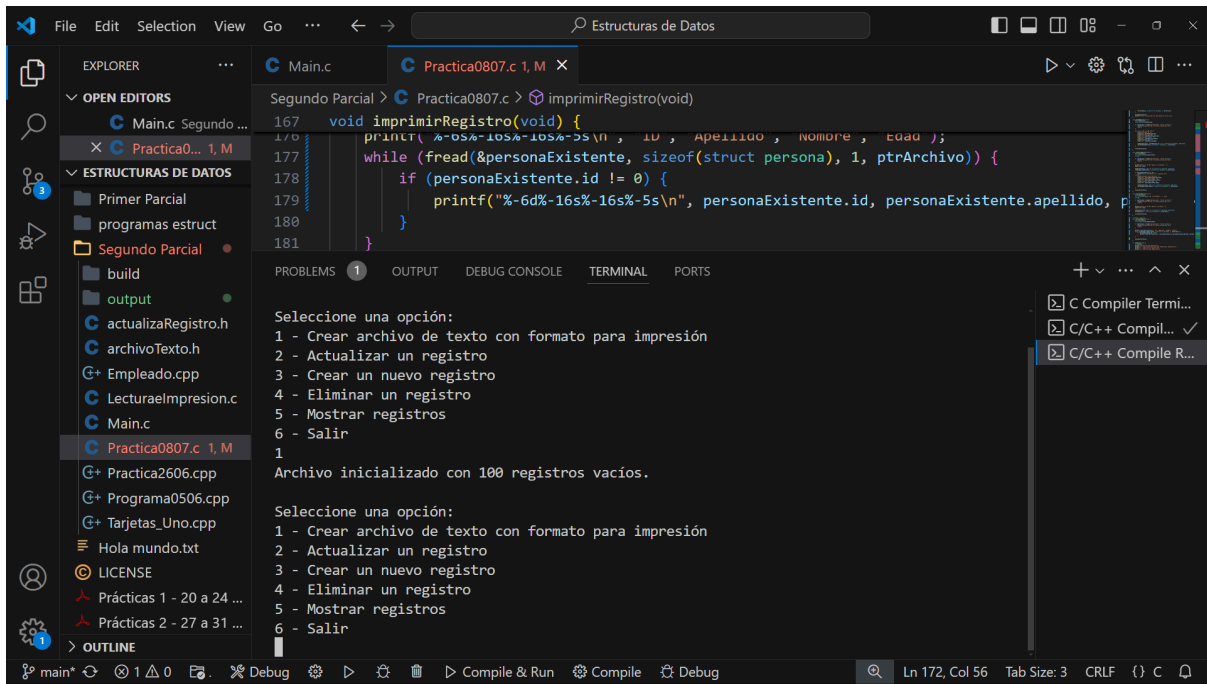
## Salida de escritorio



The screenshot shows the Visual Studio Code interface with the following components:

- EXPLORER:** Shows the project structure with folders like 'Segundo Parcial' and files like 'Practica0807.c 1, M'.
- EDITOR:** Displays the source code for 'Practica0807.c'. The function 'imprimirRegistro(void)' is visible, which uses 'fread' to read records from a file and 'printf' to display them in a formatted table.
- TERMINAL:** Shows the program's output. It prompts the user to 'Seleccione una opción:' and lists the menu options: 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. The output also shows 'Opción incorrecta' and 'Compiled successfully!'.
- PROBLEMS:** Shows a single error message: 'Opción incorrecta'.





File Edit Selection View Go ... Estructuras de Datos

EXPLORER

- OPEN EDITORS
  - Main.c Segundo ...
  - Practica0... 1, M
- ESTRUCTURAS DE DATOS
  - Primer Parcial
  - programas estruct
  - Segundo Parcial
    - build
    - output
    - actualizaRegistro.h
    - archivoTexto.h
    - Empleado.cpp
    - LecturaImpresion.c
    - Main.c
    - Practica0807.c 1, M
    - Practica2606.cpp
    - Programa0506.cpp
    - Tarjetas\_Uno.cpp
    - Hola mundo.txt
  - LICENSE
  - Prácticas 1 - 20 a 24 ...
  - Prácticas 2 - 27 a 31 ...
- OUTLINE

Practica0807.c 1, M

```

Segundo Parcial > Practica0807.c > imprimirRegistro(void)
167 void imprimirRegistro(void) {
168     printf("%-6s%-10s%-10s%-5s\n", ID, Apellido, Nombre, Edad);
169     while (fread(&personaExistente, sizeof(struct persona), 1, ptrArchivo)) {
170         if (personaExistente.id != 0) {
171             printf("%-6d%-16s%-16s%-5s\n", personaExistente.id, personaExistente.apellido, p
172         }
173     }
174 }
175
176
177
178
179
180
181

```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

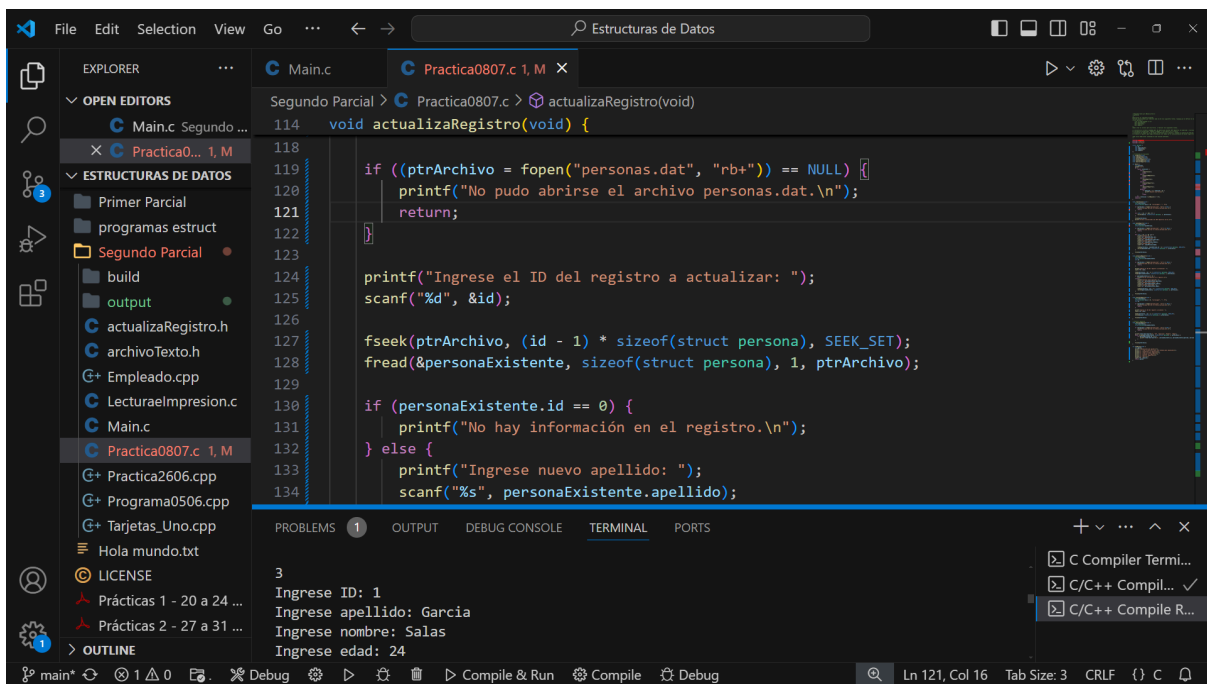
Seleccione una opción:  
 1 - Crear archivo de texto con formato para impresión  
 2 - Actualizar un registro  
 3 - Crear un nuevo registro  
 4 - Eliminar un registro  
 5 - Mostrar registros  
 6 - Salir

1

Archivo inicializado con 100 registros vacíos.

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

Ln 172, Col 56 Tab Size: 3 CRLF {} C



File Edit Selection View Go ... Estructuras de Datos

EXPLORER

- OPEN EDITORS
  - Main.c Segundo ...
  - Practica0... 1, M
- ESTRUCTURAS DE DATOS
  - Primer Parcial
  - programas estruct
  - Segundo Parcial
    - build
    - output
    - actualizaRegistro.h
    - archivoTexto.h
    - Empleado.cpp
    - LecturaImpresion.c
    - Main.c
    - Practica0807.c 1, M
    - Practica2606.cpp
    - Programa0506.cpp
    - Tarjetas\_Uno.cpp
    - Hola mundo.txt
  - LICENSE
  - Prácticas 1 - 20 a 24 ...
  - Prácticas 2 - 27 a 31 ...
- OUTLINE

Practica0807.c 1, M

```

Segundo Parcial > Practica0807.c > actualizaRegistro(void)
114 void actualizaRegistro(void) {
115
116
117     if ((ptrArchivo = fopen("personas.dat", "rb+")) == NULL) {
118         printf("No pudo abrirse el archivo personas.dat.\n");
119         return;
120     }
121
122
123
124     printf("Ingrese el ID del registro a actualizar: ");
125     scanf("%d", &id);
126
127     fseek(ptrArchivo, (id - 1) * sizeof(struct persona), SEEK_SET);
128     fread(&personaExistente, sizeof(struct persona), 1, ptrArchivo);
129
130     if (personaExistente.id == 0) {
131         printf("No hay información en el registro.\n");
132     } else {
133         printf("Ingrese nuevo apellido: ");
134         scanf("%s", personaExistente.apellido);
135     }
136 }
137
138
139
140
141
142
143

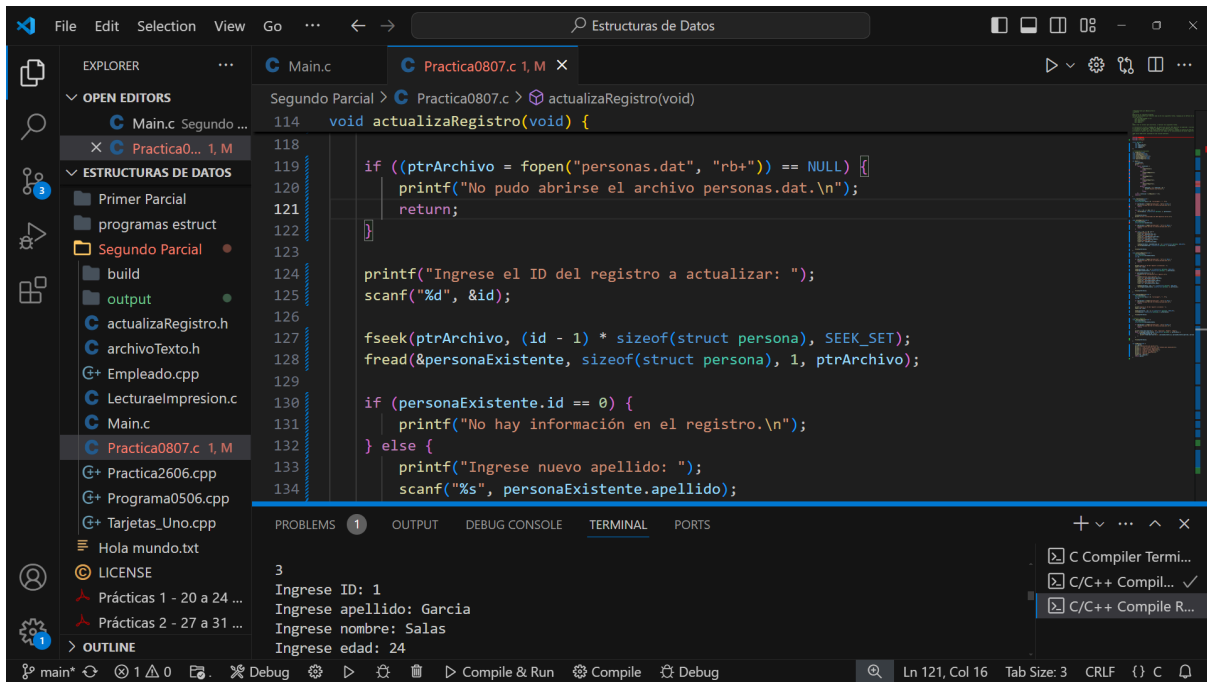
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

3

Ingreso ID: 1  
 Ingreso apellido: Garcia  
 Ingreso nombre: Salas  
 Ingreso edad: 24

Ln 121, Col 16 Tab Size: 3 CRLF {} C



File Edit Selection View Go ... Estructuras de Datos

EXPLORER

- OPEN EDITORS
  - Main.c Segundo ...
  - Practica0... 1, M
- ESTRUCTURAS DE DATOS
  - Primer Parcial
  - programas estruct
  - Segundo Parcial
    - build
    - output
    - actualizaRegistro.h
    - archivoTexto.h
    - Empleado.cpp
    - LecturaImpresion.c
    - Main.c
    - Practica0807.c 1, M
    - Practica2606.cpp
    - Programa0506.cpp
    - Tarjetas\_Uno.cpp
  - Hola mundo.txt
- LICENSE
- Prácticas 1 - 20 a 24 ...
- Prácticas 2 - 27 a 31 ...
- OUTLINE

Main.c Segundo ...

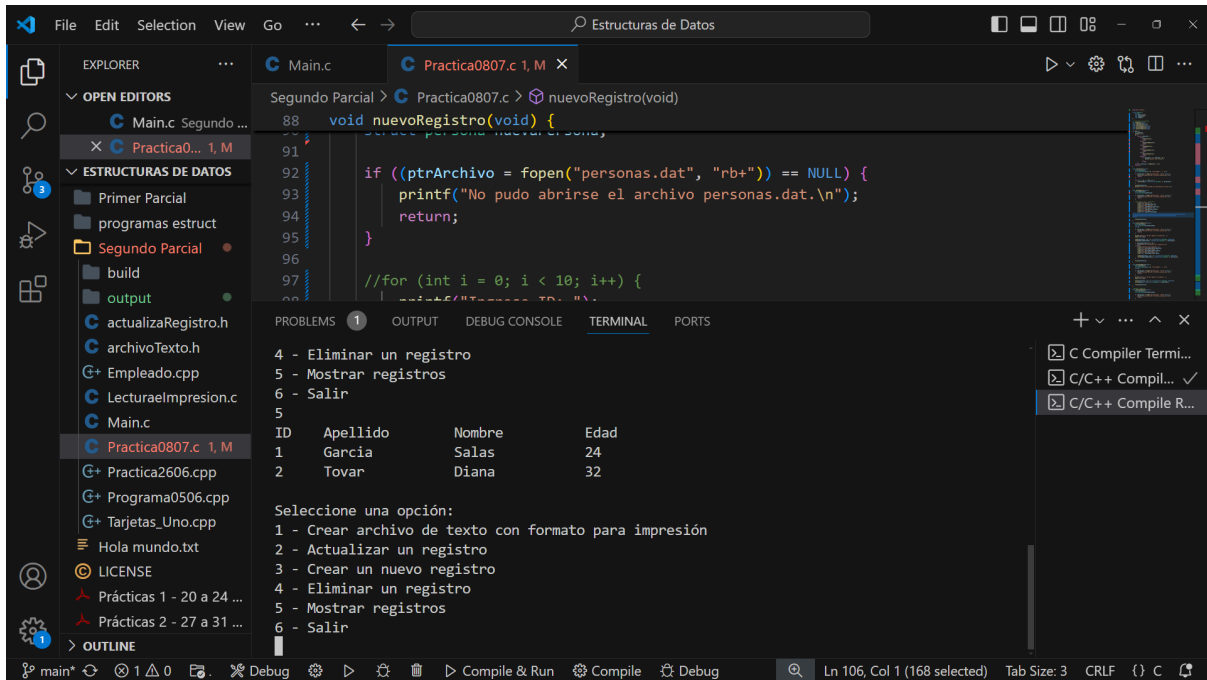
```

114 void actualizaRegistro(void) {
118
119     if ((ptrArchivo = fopen("personas.dat", "rb+")) == NULL) {
120         printf("No pudo abrirse el archivo personas.dat.\n");
121         return;
122     }
123
124     printf("Ingrese el ID del registro a actualizar: ");
125     scanf("%d", &id);
126
127     fseek(ptrArchivo, (id - 1) * sizeof(struct persona), SEEK_SET);
128     fread(&personaExistente, sizeof(struct persona), 1, ptrArchivo);
129
130     if (personaExistente.id == 0) {
131         printf("No hay información en el registro.\n");
132     } else {
133         printf("Ingrese nuevo apellido: ");
134         scanf("%s", personaExistente.apellido);
    
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

3  
Ingreso ID: 1  
Ingreso apellido: Garcia  
Ingreso nombre: Salas  
Ingreso edad: 24

Ln 121, Col 16 Tab Size: 3 CRLF {} C



File Edit Selection View Go ... Estructuras de Datos

EXPLORER

- OPEN EDITORS
  - Main.c Segundo ...
  - Practica0... 1, M
- ESTRUCTURAS DE DATOS
  - Primer Parcial
  - programas estruct
  - Segundo Parcial
    - build
    - output
    - actualizaRegistro.h
    - archivoTexto.h
    - Empleado.cpp
    - LecturaImpresion.c
    - Main.c
    - Practica0807.c 1, M
    - Practica2606.cpp
    - Programa0506.cpp
    - Tarjetas\_Uno.cpp
  - Hola mundo.txt
- LICENSE
- Prácticas 1 - 20 a 24 ...
- Prácticas 2 - 27 a 31 ...
- OUTLINE

Main.c Segundo ...

```

88 void nuevoRegistro(void) {
89     struct persona nuevoRegistro;
90
91     if ((ptrArchivo = fopen("personas.dat", "rb+")) == NULL) {
92         printf("No pudo abrirse el archivo personas.dat.\n");
93         return;
94     }
95
96     //for (int i = 0; i < 10; i++) {
97         printf("Ingrese nuevo ID: ");
98         scanf("%d", &nuevoRegistro.id);
99     }
    
```

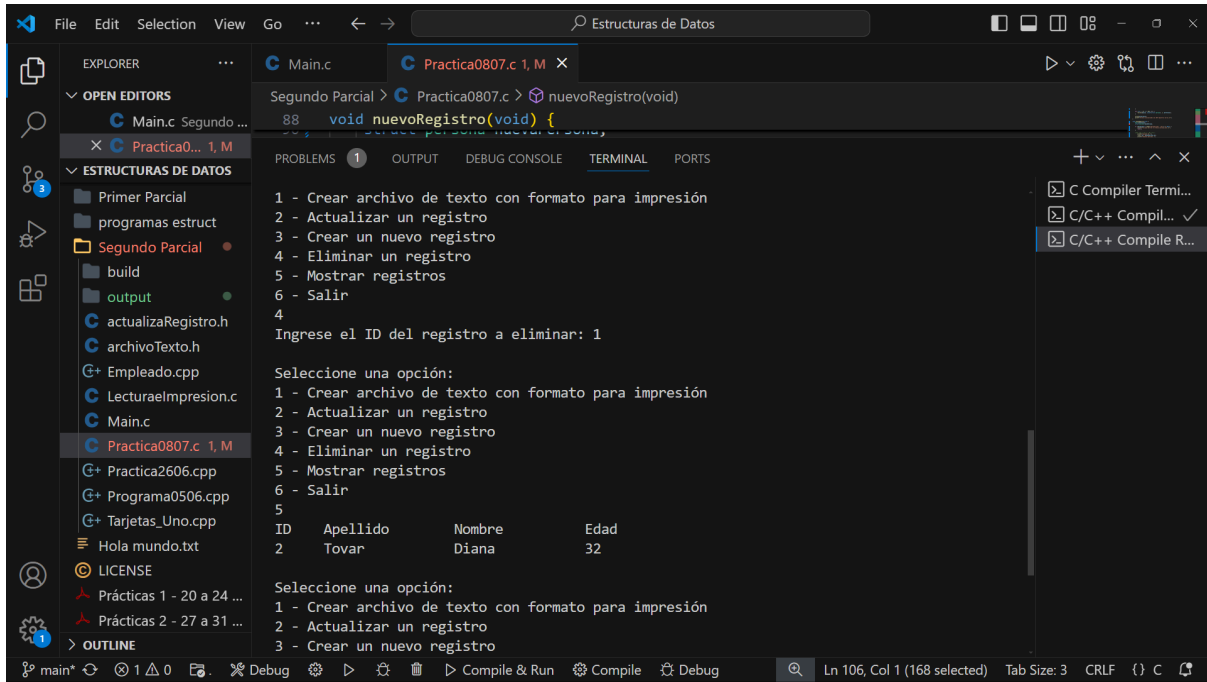
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

4 - Eliminar un registro  
5 - Mostrar registros  
6 - Salir

ID	Apellido	Nombre	Edad
1	Garcia	Salas	24
2	Tovar	Diana	32

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

Ln 106, Col 1 (168 selected) Tab Size: 3 CRLF {} C



File Edit Selection View Go ... Estructuras de Datos

EXPLORER

OPEN EDITORS

Practica0807.c 1, M

ESTRUCTURAS DE DATOS

Primer Parcial

programas estruct

Segundo Parcial

build

output

actualizaRegistro.h

archivoTexto.h

Empleado.cpp

LecturaImpresion.c

Main.c

Practica0807.c 1, M

Practica2606.cpp

Programa0506.cpp

Tarjetas\_Uno.cpp

Hola mundo.txt

LICENCE

Prácticas 1 - 20 a 24 ...

Prácticas 2 - 27 a 31 ...

OUTLINE

main\*

Debug

Compile & Run

Compile

Debug

Ln 106, Col 1 (168 selected) Tab Size: 3 CRLF {} C

Segundo Parcial > Practica0807.c > nuevoRegistro(void)

88 void nuevoRegistro(void) {

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

4

Ingrese el ID del registro a eliminar: 1

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

5

ID	Apellido	Nombre	Edad
2	Tovar	Diana	32

Seleccione una opción:

1 - Crear archivo de texto con formato para impresión

2 - Actualizar un registro

3 - Crear un nuevo registro

## Documentación por bloques de código

### ● Bloque Verde

Declaración de variables

### ● Bloque Rojo

Lectura de datos

### ● Bloque Azul

Operaciones

### ● Bloque Naranja

Resultado final