



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería Mecánica Eléctrica
Proyectos Aplicados a I.E.
1er Examen Parcial
[769] Introducción A La Programación De
Computadoras
Ing. José Anibal Silva de Los Angeles

Nombre: Axel Daniel Ramirez Urbina

Carnet: 202010083

Registro académico: 202010083

Instrucciones

1. Presentación:

1. Mostrar su documento de identificación.
2. Subir un único archivo PDF con el siguiente contenido:
 - a. Código del programa.
 - b. Enlaces al repositorio externo.
 - c. Pantallazos que muestren el funcionamiento del programa.
 - d. Reporte en formato IEEE.
 - e. Nombre del archivo PDF: Su registro académico.

3. Almacenamiento del código:

- Local: Carpeta personal.
- Remoto: Repositorio privado de GitHub con el usuario @jasdalinux o jasda@ingenieria.usac.edu.gt.

4. Documentación:

- Diagrama de flujo del proceso de la solución.
- Algoritmo del programa.
- Formato del reporte: IEEE.

2. Desarrollo de los programas:

- Lenguaje: C consola.
- Almacenamiento de datos: Archivo de texto "salida.txt".
- Los programas debe procurar mitigar los errores de ejecución.
- Funcionalidades y menú:
 - Ingreso de nombre usuario.
 - Ingreso ejecución programa
 - Historial de datos.
 - Borrado de datos.
 - Salir

5. Aclaraciones:

- El reporte IEEE debe incluir el código del programa, el diagrama de flujo, el algoritmo y las capturas de pantalla.
- El archivo PDF debe contener todos los elementos mencionados en la sección "1. Presentación".
- El repositorio de GitHub debe ser privado y tener como usuario @jasdalinux o jasda@ingenieria.usac.edu.gt.

Programas

- Escribe un programa en C que solicite al usuario dos números enteros, realice la suma y muestre el resultado.
- Crea una calculadora simple que permita al usuario realizar operaciones de suma, resta, multiplicación y división.
- Diseña un programa que pida al usuario ingresar un número y determine si es positivo, negativo o cero.
- Pide al usuario que ingrese un número y muestra la tabla de multiplicar de ese número hasta 10.
- Desarrolla un programa que solicite un número y determine si es primo o no.
- Crea un programa que convierta una temperatura en grados Celsius a Fahrenheit, solicitando al usuario la temperatura en Celsius.
- Diseña un programa que calcule el monto final después de ciertos años utilizando la fórmula del interés compuesto.
- Desarrolla un programa que simule las operaciones básicas de un cajero automático, como consultar saldo, retirar dinero y depositar dinero.
- Desarrolla un programa que pida un número y determinar si es par o impar.
- Desarrolla un programa que realice la serie Fibonacci.

Crea un programa en C que presente al usuario una serie de 5 adivinanzas y registre cuántas adivinanzas acierta. Al final del juego, se solicitará al usuario su nombre y se registrará su nombre y puntuación en un archivo de puntuaciones. El programa deberá desplegar un menú para seleccionar deben ser:



Parcial 1 _suma

Axel Daniel, Ramirez Urbina, 202010083

*Escuela de Mecánica Eléctrica, Facultad de Ingeniería,
Universidad de San Carlos de Guatemala*

A. Resumen

El programa desarrollado en lenguaje C permite al usuario realizar la suma de dos números enteros, almacenar el resultado en un archivo de texto llamado "salida.txt" y acceder a funcionalidades adicionales a través de un menú interactivo.

Al inicio del programa, se solicita al usuario que ingrese su nombre, que será utilizado para personalizar los mensajes de saludo y despedida. Luego, se presenta un menú con las siguientes opciones:

- 1. Ingresar números y realizar suma:** Permite al usuario ingresar dos números enteros y realiza la suma de los mismos. El resultado se muestra en pantalla y se guarda en el archivo "salida.txt".
- 2. Ver historial de datos:** Muestra en pantalla el contenido del archivo "salida.txt", que contiene el historial de las sumas realizadas previamente.
- 3. Borrar historial de datos:** Elimina el contenido del archivo "salida.txt", limpiando así el historial de las sumas realizadas.
- 4. Salir:** Finaliza la ejecución del programa, mostrando un mensaje de despedida personalizado con el nombre del usuario.

El programa utiliza estructuras de control y funciones para organizar el flujo de ejecución y modularizar las distintas funcionalidades. Además, se implementa manejo de archivos para la persistencia de datos y se incluyen mensajes informativos para guiar al usuario a lo largo de su interacción con el programa.

B. Código

```
C suma.c > ...
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4
5  // Estructura para almacenar los datos del usuario
6  struct Usuario {
7      char nombre[50];
8  };
9
10 // Función para ingresar el nombre del usuario
11 void ingresarNombre(struct Usuario *usuario) {
12     printf("Ingrese su nombre: ");
13     scanf("%s", usuario->nombre);
14 }
15
16 // Función para realizar la suma de dos números enteros
17 void realizarSuma() {
18     int num1, num2;
19     printf("\nIngrese el primer numero entero: ");
20     scanf("%d", &num1);
21     printf("Ingrese el segundo numero entero: ");
22     scanf("%d", &num2);
23
24     int suma = num1 + num2;
25     printf("La suma de %d y %d es: %d\n", num1, num2, suma);
26
27     // Guardar la suma en el archivo de salida
28     FILE *archivo = fopen("salida.txt", "a");
29     if (archivo != NULL) {
30         fprintf(archivo, "Suma: %d\n", suma);
31         fclose(archivo);
32     } else {
33         printf("Error al guardar la suma.\n");
34     }
35 }
36
37 // Función para mostrar el historial de datos
38 void mostrarHistorial() {
39     printf("\n--- Historial de Datos ---\n");
40
41     FILE *archivo = fopen("salida.txt", "r");
42     if (archivo != NULL) {
43         char linea[100];
44         while (fgets(linea, sizeof(linea), archivo)) {
45             printf("%s", linea);
46         }
47         fclose(archivo);
48     } else {
49         printf("No hay historial de datos.\n");
50     }
51 }
52
53 // Función para borrar el historial de datos
54 void borrarHistorial() {
55     if (remove("salida.txt") == 0) {
56         printf("\nHistorial de datos borrado exitosamente.\n");
57     } else {
58         printf("\nError al borrar el historial de datos.\n");
59     }
60 }
61
62 int main() {
63     int opcion;
64     struct Usuario usuario;
65
66     printf("¡Bienvenido al programa de suma de numeros enteros!\n");
67
68     // Ingresar el nombre del usuario al inicio del programa
69     ingresarNombre(&usuario);
70
71     do {
72         printf("\nMenu:\n");
73         printf("1. Ingresar numeros y realizar suma\n");
74         printf("2. Ver historial de datos\n");
75         printf("3. Borrar historial de datos\n");
76         printf("4. Salir\n");
77         printf("Seleccione una opcion: ");
78         scanf("%d", &opcion);
79
80         switch (opcion) {
81             case 1:
82                 realizarSuma();
83                 break;
84             case 2:
85                 mostrarHistorial();
86                 break;
87             case 3:
88                 borrarHistorial();
89                 break;
90             case 4:
91                 printf("\n¡Gracias por usar el programa, %s! ¡Hasta luego!\n", usuario.nombre);
92                 break;
93             default:
94                 printf("\nOpcion no valida. Por favor, seleccione una opcion valida.\n");
95                 break;
96         } while (opcion != 4);
97     } while (opcion != 4);
98
99     return 0;
100 }
101
```

C. Resultados

```
PS C:\Users\johan\OneDrive\Escritorio\USAC\Primer semestre 2024\PROGRAMACIÓN\EXÁMENES\1> gcc suma.c
PS C:\Users\johan\OneDrive\Escritorio\USAC\Primer semestre 2024\PROGRAMACIÓN\EXÁMENES\1> ./a
¡Bienvenido al programa de suma de numeros enteros!
Ingrese su nombre: Axel
```

Prueba de la opción 1

```
Menu:
1. Ingresar numeros y realizar suma
2. Ver historial de datos
3. Borrar historial de datos
4. Salir
Seleccione una opcion: 1

Ingrese el primer numero entero: 2
Ingrese el segundo numero entero: 3
La suma de 2 y 3 es: 5
```

Prueba de la opción 2

```
Menu:
1. Ingresar numeros y realizar suma
2. Ver historial de datos
3. Borrar historial de datos
4. Salir
Seleccione una opcion: 2

--- Historial de Datos ---
Suma: 5
Suma: 5
```

Prueba de la opción 3

```
Menu:
1. Ingresar numeros y realizar suma
2. Ver historial de datos
3. Borrar historial de datos
4. Salir
Seleccione una opcion: 3

Historial de datos borrado exitosamente.
```

Prueba de la opción 4

```
Menu:
1. Ingresar numeros y realizar suma
2. Ver historial de datos
3. Borrar historial de datos
4. Salir
Seleccione una opcion: 4

¡Gracias por usar el programa, Axel! ¡Hasta luego!
```

Documento en GitHub

<https://github.com/AxelRamirez12/parcial1.git>

Parcial 1 calculadora

Axel Daniel, Ramírez Urbina, 202010083

*Escuela de Mecánica Eléctrica, Facultad de Ingeniería,
Universidad de San Carlos de Guatemala*

A. Resumen

El programa implementado en lenguaje C consiste en una calculadora simple que permite al usuario realizar operaciones aritméticas básicas, tales como suma, resta, multiplicación y división. Además, ofrece funcionalidades adicionales como el almacenamiento de los resultados de las operaciones en un archivo de texto llamado "salida.txt", acceso a un historial de datos, la posibilidad de borrar dicho historial y una opción para salir del programa.

Al inicio de la ejecución, se solicita al usuario que ingrese su nombre para personalizar los mensajes de saludo y despedida. Luego, se presenta un menú interactivo con las siguientes opciones:

1. **Ingreso ejecución programa:** Permite al usuario ingresar los dos operandos y seleccionar la operación deseada (suma, resta, multiplicación o división). El resultado de la operación se muestra en pantalla y se guarda en el archivo "salida.txt".
2. **Historial de datos:** Muestra en pantalla el contenido del archivo "salida.txt", que contiene el historial de las operaciones realizadas previamente.
3. **Borrado de datos:** Elimina el contenido del archivo "salida.txt", limpiando así el historial de operaciones.
4. **Salir:** Finaliza la ejecución del programa, mostrando un mensaje de despedida personalizado con el nombre del usuario.

El programa utiliza estructuras de control, funciones y manejo de archivos para implementar las distintas funcionalidades de manera modular y ordenada. Se incluyen mensajes informativos para guiar al usuario durante su interacción con la calculadora y se manejan posibles errores de ejecución para mejorar la experiencia del usuario.

B. Código

```
C calculadora.c > mostrarHistorial()
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4
5  // Estructura para almacenar los datos del usuario
6  struct Usuario {
7      char nombre[50];
8  };
9
10 // Función para ingresar el nombre del usuario
11 void ingresarNombre(struct Usuario *usuario) {
12     printf("Ingrese su nombre: ");
13     scanf("%s", usuario->nombre);
14 }
15
16 // Función para realizar la suma de dos números
17 float suma(float a, float b) {
18     return a + b;
19 }
20
21 // Función para realizar la resta de dos números
22 float resta(float a, float b) {
23     return a - b;
24 }
25
26 // Función para realizar la multiplicación de dos números
27 float multiplicacion(float a, float b) {
28     return a * b;
29 }
30
31 // Función para realizar la división de dos números
32 float division(float a, float b) {
33     if (b != 0) {
34         return a / b;
35     } else {
36         printf("Error: No se puede dividir por cero.\n");
37         return 0;
38     }
39 }
40
41 // Función para realizar una operación matemática
42 float realizarOperacion(int opcion, float a, float b) {
43     switch (opcion) {
44         case 1:
45             return suma(a, b);
46         case 2:
47             return resta(a, b);
48         case 3:
```



```

49         return multiplicacion(a, b);
50     case 4:
51         return division(a, b);
52     default:
53         printf("Opcion no valida.\n");
54         return 0;
55     }
56 }
57
58 // Función para mostrar el historial de datos
59 void mostrarHistorial() {
60     printf("\n--- Historial de Datos ---\n");
61
62     FILE *archivo = fopen("salida.txt", "r");
63     if (archivo != NULL) {
64         char linea[100];
65         while (fgets(linea, sizeof(linea), archivo)) {
66             printf("%s", linea);
67         }
68         fclose(archivo);
69     } else {
70         printf("No hay historial de datos.\n");
71     }
72 }
73
74 // Función para borrar el historial de datos
75 void borrarHistorial() {
76     if (remove("salida.txt") == 0) {
77         printf("\nHistorial de datos borrado exitosamente.\n");
78     } else {
79         printf("\nError al borrar el historial de datos.\n");
80     }
81 }
82
83 int main() {
84     int opcion;
85     struct Usuario usuario;
86     float num1, num2, resultado;
87
88     printf("¡Bienvenido a la calculadora!\n");
89
90     // Ingresar el nombre del usuario al inicio del programa
91     ingresarNombre(&usuario);
92
93     do {

```

C. Resultados

Menu:

1. Suma
2. Resta
3. Multiplicacion
4. Division
5. Ver historial de datos
6. Borrar historial de datos
7. Salir

Seleccione una opcion: 1

Ingrese dos numeros:

Primer numero: 1

Segundo numero: 2

El resultado de la operacion es: 3.00

Menu:

1. Suma
2. Resta
3. Multiplicacion
4. Division
5. Ver historial de datos
6. Borrar historial de datos
7. Salir

Seleccione una opcion: 2

Ingrese dos numeros:

Primer numero: 1

Segundo numero: 2

El resultado de la operacion es: -1.00

Menu:

1. Suma
2. Resta
3. Multiplicacion
4. Division
5. Ver historial de datos
6. Borrar historial de datos
7. Salir

Seleccione una opcion: 3

Ingrese dos numeros:

Primer numero: 2

Segundo numero: 2

El resultado de la operacion es: 4.00

Documento en GitHub

<https://github.com/AxelRamirez12/parcial1.git>

```

94     printf("\nMenu:\n");
95     printf("1. Suma\n");
96     printf("2. Resta\n");
97     printf("3. Multiplicacion\n");
98     printf("4. Division\n");
99     printf("5. Ver historial de datos\n");
100    printf("6. Borrar historial de datos\n");
101    printf("7. Salir\n");
102    printf("Seleccione una opcion: ");
103    scanf("%d", &opcion);
104
105    switch (opcion) {
106        case 1:
107        case 2:
108        case 3:
109        case 4:
110            printf("\nIngrese dos numeros:\n");
111            printf("Primer numero: ");
112            scanf("%f", &num1);
113            printf("Segundo numero: ");
114            scanf("%f", &num2);
115            resultado = realizarOperacion(opcion, num1, num2);
116            printf("El resultado de la operacion es: %.2f\n", resultado);
117
118            // Guardar la operación en el archivo de salida
119            FILE *archivo = fopen("salida.txt", "a");
120            if (archivo != NULL) {
121                fprintf(archivo, "%.2f %c %.2f = %.2f\n", num1, "+-*/"[opcion - 1], num2, resultado);
122                fclose(archivo);
123            } else {
124                printf("Error al guardar la operacion.\n");
125            }
126            break;
127        case 5:
128            mostrarHistorial();
129            break;
130        case 6:
131            borrarHistorial();
132            break;
133        case 7:
134            printf("\n¡Gracias por usar la calculadora, %s! ¡Hasta luego!\n", usuario.nombre);
135            break;
136        default:
137            printf("\nOpcion no valida. Por favor, seleccione una opcion valida.\n");
138            break;

```

```

139    }
140    } while (opcion != 7);
141
142    return 0;
143 }
144

```

Parcial 1 número

Axel Daniel, Ramirez Urbina, 202010083

*Escuela de Mecánica Eléctrica, Facultad de Ingeniería,
Universidad de San Carlos de Guatemala*

A. Resumen

El programa en lenguaje C diseñado permite al usuario ingresar un número y determina si es positivo, negativo o cero. Además de esta funcionalidad principal, ofrece opciones adicionales como el almacenamiento de los datos en un archivo de texto llamado "salida.txt", acceso al historial de datos, la posibilidad de borrar dicho historial y salir del programa.

El programa comienza solicitando al usuario que ingrese su nombre para personalizar los mensajes. Luego, presenta un menú interactivo con las siguientes opciones:

- Ingreso ejecución programa:** Permite al usuario ingresar un número, y el programa determina su clasificación (positivo, negativo o cero). El resultado se muestra en pantalla y se guarda en el archivo "salida.txt".
- Historial de datos:** Muestra en pantalla el contenido del archivo "salida.txt", que contiene el historial de los números ingresados y su clasificación.
- Borrado de datos:** Elimina el contenido del archivo "salida.txt", limpiando así el historial de datos.
- Salir:** Finaliza la ejecución del programa, mostrando un mensaje de despedida personalizado con el nombre del usuario.

El programa utiliza estructuras de control, funciones y manejo de archivos para implementar las funcionalidades de manera modular y ordenada. Se incluyen mensajes informativos para guiar al usuario durante su interacción con el programa, y se manejan posibles errores de ejecución para mejorar la experiencia del usuario.

B. Código

```
C numero.c > ...
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4
5  // Estructura para almacenar los datos del usuario
6  struct Usuario {
7      char nombre[50];
8  };
9
10 // Función para ingresar el nombre del usuario
11 void ingresarNombre(struct Usuario *usuario) {
12     printf("Ingrese su nombre: ");
13     scanf("%s", usuario->nombre);
14 }
15
16 // Función para determinar si un número es positivo, negativo o cero
17 void determinarNumero() {
18     float numero;
19     printf("\nIngrese un numero: ");
20     scanf("%f", &numero);
21
22     // Determinar si el número es positivo, negativo o cero
23     if (numero > 0) {
24         printf("El numero %.2f es positivo.\n", numero);
25     } else if (numero < 0) {
26         printf("El numero %.2f es negativo.\n", numero);
27     } else {
28         printf("El numero es cero.\n");
29     }
30
31     // Guardar la clasificación en el archivo de salida
32     FILE *archivo = fopen("salida.txt", "a");
33     if (archivo != NULL) {
34         fprintf(archivo, "Numero: %.2f\n", numero);
35         fclose(archivo);
36     } else {
37         printf("Error al guardar la clasificacion del numero.\n");
38     }
39 }
40
41 // Función para mostrar el historial de datos
42 void mostrarHistorial() {
43     printf("\n--- Historial de Datos ---\n");
44
45     FILE *archivo = fopen("salida.txt", "r");
46     if (archivo != NULL) {
47         char linea[100];
48         while (fgets(linea, sizeof(linea), archivo)) {
```

```

49         printf("%s", linea);
50     }
51     fclose(archivo);
52 } else {
53     printf("No hay historial de datos.\n");
54 }
55 }
56
57 // Función para borrar el historial de datos
58 void borrarHistorial() {
59     if (remove("salida.txt") == 0) {
60         printf("\nHistorial de datos borrado exitosamente.\n");
61     } else {
62         printf("\nError al borrar el historial de datos.\n");
63     }
64 }
65
66 int main() {
67     int opcion;
68     struct Usuario usuario;
69
70     printf("¡Bienvenido al programa de clasificacion de numeros!\n");
71
72     // Ingresar el nombre del usuario al inicio del programa
73     ingresarNombre(&usuario);
74
75     do {
76         printf("\nMenu:\n");
77         printf("1. Ingresar numero y determinar clasificacion\n");
78         printf("2. Ver historial de datos\n");
79         printf("3. Borrar historial de datos\n");
80         printf("4. Salir\n");
81         printf("Seleccione una opcion: ");
82         scanf("%d", &opcion);
83
84         switch (opcion) {
85             case 1:
86                 determinarNumero();
87                 break;
88             case 2:
89                 mostrarHistorial();
90                 break;
91             case 3:
92                 borrarHistorial();
93                 break;
94             case 4:
95                 printf("\nGracias por usar el programa, %s! ¡Hasta luego!\n", usuario.nombre);
96                 break;
97             default:
98                 printf("\nOpcion no valida. Por favor, seleccione una opcion valida.\n");
99                 break;
100         }
101     } while (opcion != 4);
102
103     return 0;
104 }
105

```

C. Resultados

```

Menu:
1. Ingresar numero y determinar clasificacion
2. Ver historial de datos
3. Borrar historial de datos
4. Salir
Seleccione una opcion: 1

Ingrese un numero: 2
El numero 2.00 es positivo.

Menu:
1. Ingresar numero y determinar clasificacion
2. Ver historial de datos
3. Borrar historial de datos
4. Salir
Seleccione una opcion: 2

--- Historial de Datos ---
Numero: 2.00

Menu:
1. Ingresar numero y determinar clasificacion
2. Ver historial de datos
3. Borrar historial de datos
4. Salir
Seleccione una opcion: 3

Historial de datos borrado exitosamente.

Menu:
1. Ingresar numero y determinar clasificacion
2. Ver historial de datos
3. Borrar historial de datos
4. Salir
Seleccione una opcion: 4

¡Gracias por usar el programa, Axel! ¡Hasta luego!

```

Documento en GitHub

<https://github.com/AxelRamirez12/parcial1.git>

Parcial 1 _primo

Axel Daniel, Ramirez Urbina, 202010083
Escuela de Mecánica Eléctrica, Facultad de Ingeniería,
Universidad de San Carlos de Guatemala

A. Resumen

El programa en lenguaje C diseñado tiene como objetivo determinar si un número ingresado por el usuario es primo o no. Además de esta funcionalidad principal, ofrece opciones adicionales como el almacenamiento de los resultados en un archivo de texto llamado "salida.txt", acceso al historial de datos, la posibilidad de borrar dicho historial y salir del programa.

Al inicio de la ejecución, se solicita al usuario que ingrese su nombre para personalizar los mensajes. Luego, presenta un menú interactivo con las siguientes opciones:

1. **Determinar si un número es primo:** Permite al usuario ingresar un número y determina si es primo o no. El resultado se muestra en pantalla y se guarda en el archivo "salida.txt".
2. **Historial de datos:** Muestra en pantalla el contenido del archivo "salida.txt", que contiene el historial de los números ingresados y su clasificación como primo o no primo.
3. **Borrar historial de datos:** Elimina el contenido del archivo "salida.txt", limpiando así el historial de datos.
4. **Salir:** Finaliza la ejecución del programa, mostrando un mensaje de despedida personalizado con el nombre del usuario.

El programa utiliza estructuras de control, funciones y manejo de archivos para implementar las funcionalidades de manera modular y ordenada. Además, se incluyen mensajes informativos para guiar al usuario durante su interacción con el programa, y se manejan posibles errores de ejecución para mejorar la experiencia del usuario.

B. Código

```
C primo.c > ingresarNombre(Usuario *)
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4
5  // Estructura para almacenar los datos del usuario
6  struct Usuario {
7      char nombre[50];
8  };
9
10 // Función para ingresar el nombre del usuario
11 void ingresarNombre(struct Usuario *usuario) {
12     printf("Ingrese su nombre: ");
13     scanf("%s", usuario->nombre);
14 }
15
16 // Función para determinar si un número es primo
17 int esPrimo(int numero) {
18     if (numero <= 1) {
19         return 0; // No es primo
20     }
21     for (int i = 2; i * i <= numero; i++) {
22         if (numero % i == 0) {
23             return 0; // No es primo
24         }
25     }
26     return 1; // Es primo
27 }
28
29 // Función para determinar si un número es primo y guardar el resultado en el archivo de salida
30 void determinarPrimo() {
31     int numero;
32     printf("\nIngrese un numero: ");
33     scanf("%d", &numero);
34
35     // Determinar si el número es primo
36     int primo = esPrimo(numero);
37     if (primo) {
38         printf("El numero %d es primo.\n", numero);
39     } else {
40         printf("El numero %d no es primo.\n", numero);
41     }
42
43     // Guardar la clasificación en el archivo de salida
44     FILE *archivo = fopen("salida.txt", "a");
45     if (archivo != NULL) {
46         if (primo) {
47             fprintf(archivo, "Numero %d: Primo\n", numero);
48         } else {
```



```

49         fprintf(archivo, "Numero %d: No Primo\n", numero);
50     }
51     fclose(archivo);
52 } else {
53     printf("Error al guardar la clasificacion del numero.\n");
54 }
55 }
56
57 // Función para mostrar el historial de datos
58 void mostrarHistorial() {
59     printf("\n--- Historial de Datos ---\n");
60
61     FILE *archivo = fopen("salida.txt", "r");
62     if (archivo != NULL) {
63         char linea[100];
64         while (fgets(linea, sizeof(linea), archivo)) {
65             printf("%s", linea);
66         }
67         fclose(archivo);
68     } else {
69         printf("No hay historial de datos.\n");
70     }
71 }
72
73 // Función para borrar el historial de datos
74 void borrarHistorial() {
75     if (remove("salida.txt") == 0) {
76         printf("\nHistorial de datos borrado exitosamente.\n");
77     } else {
78         printf("\nError al borrar el historial de datos.\n");
79     }
80 }
81
82 int main() {
83     int opcion;
84     struct Usuario usuario;
85
86     printf("¡Bienvenido al programa de determinacion de numeros primos!\n");
87
88     // Ingresar el nombre del usuario al inicio del programa
89     ingresarNombre(&usuario);
90
91     do {
92         printf("\nMenu:\n");
93         printf("1. Determinar si un numero es primo\n");

```

```

94         printf("2. Ver historial de datos\n");
95         printf("3. Borrar historial de datos\n");
96         printf("4. Salir\n");
97         printf("Seleccione una opcion: ");
98         scanf("%d", &opcion);
99
100        switch (opcion) {
101            case 1:
102                determinarPrimo();
103                break;
104            case 2:
105                mostrarHistorial();
106                break;
107            case 3:
108                borrarHistorial();
109                break;
110            case 4:
111                printf("\n¡Gracias por usar el programa, %s! ¡Hasta luego!\n", usuario.nombre);
112                break;
113            default:
114                printf("\nOpcion no valida. Por favor, seleccione una opcion valida.\n");
115                break;
116        }
117    } while (opcion != 4);
118
119    return 0;
120 }
121

```

C. Resultados

Menu:

1. Determinar si un numero es primo
2. Ver historial de datos
3. Borrar historial de datos
4. Salir

Seleccione una opcion: 1

Ingrese un numero: 7

El numero 7 es primo.

Menu:

1. Determinar si un numero es primo
2. Ver historial de datos
3. Borrar historial de datos
4. Salir

Seleccione una opcion: 2

--- Historial de Datos ---

Numero 7: Primo

Menu:

1. Determinar si un numero es primo
2. Ver historial de datos
3. Borrar historial de datos
4. Salir

Seleccione una opcion: 3

Historial de datos borrado exitosamente.

Menu:

1. Determinar si un numero es primo
2. Ver historial de datos
3. Borrar historial de datos
4. Salir

Seleccione una opcion: 4

¡Gracias por usar el programa, Axel! ¡Hasta luego!

Documento en GitHub

<https://github.com/AxelRamirez12/parcial1.git>