

Proyecto

Integrantes: Estefany Morillo y Mateo García.

Opción 11: Numero de la Suerte

Se desea conocer el lucky number (número de la suerte) de cualquier persona. El número de la suerte se consigue reduciendo la fecha de nacimiento a un número de un solo dígito.

Ejemplo, la fecha de nacimiento de Emma es la siguiente: 16-08-1973 > 16+8+1973 = 1997 > 14+9+9+7=26 > 2+6 =8. El número de la suerte de Emma será el 8.

Datos:

Entrada	Proceso	Salida
dia= día de la fecha de nacimiento mes = día del mes de nacimiento ano = día del año de nacimiento	Mientras sum>=10 Hacer n_suerte = 0 Mientras sum>0 Hacer n_suerte = n_suerte+(sum MOD 10) sum = Trunc(sum/10) FinMientras sum = n_suerte	sum= suma de los números n_suerte= resultado del número de la suerte

Resolución de Estefany, Propuesta 1:

Código en C:

```
#include <stdio.h>
#include <conio.h>
#include <stdbool.h>
#include <stdlib.h>
#include <time.h>

int obtenerAnioActual() {
    time_t now;
    time(&now);
    struct tm *local = localtime(&now);
    return local->tm_year + 1900;
}

bool esFechaValida(int dia, int mes, int anio) {
    int anio_actual = obtenerAnioActual();

    if (anio < 1900 || anio > anio_actual)
        return false;

    if (mes < 1 || mes > 12)
        return false;
```

```

    if (dia < 1 || dia > 31)
        return false;

    switch (mes) {
        case 4: case 6: case 9: case 11:
            if (dia > 30)
                return false;
            break;
        case 2:
            if ((anio % 4 == 0 && anio % 100 != 0) || (anio % 400 == 0)) {
                if (dia > 29)
                    return false;
            } else {
                if (dia > 28)
                    return false;
            }
            break;
    }

    return true;
}

int sumarDigitos(int num) {
    int suma = 0;
    while (num > 0) {
        suma += num % 10;
        num /= 10;
    }
    return suma;
}

int obtenerFechaDeNacimiento(char *fecha_nacimiento) {
    printf("Ingrese la fecha de nacimiento (dia-mes-ano): ");
    scanf("%s", fecha_nacimiento);
    return 0;
}

int numeroSuerte(char *fecha) {
    int suma_total = 0;

    int dia, mes, anio;
    sscanf(fecha, "%d-%d-%d", &dia, &mes, &anio);

    if (!esFechaValida(dia, mes, anio)) {

```

```

        printf("La fecha ingresada no es válida. Por favor, ingrese una fecha
válida (dia-mes-ano): ");
        return -1;
    }

    while (*fecha != '\0') {
        if (*fecha >= '0' && *fecha <= '9') {
            suma_total += *fecha - '0';
        }
        fecha++;
    }

    while (suma_total > 9) {
        suma_total = sumarDigitos(suma_total);
    }

    return suma_total;
}

int main() {
    char fecha_nacimiento[11];
    int numero_lucky;
    bool fecha_valida = false;

    do {
        if (obtenerFechaDeNacimiento(fecha_nacimiento) != 0) {

            return 1;
        }

        numero_lucky = numeroSuerte(fecha_nacimiento);

        if (numero_lucky == -1) {

            fecha_valida = false;
        } else {
            fecha_valida = true;
        }
    } while (!fecha_valida);

    printf("El numero de la suerte es: %d\n", numero_lucky);

    return 0;
}

```

Pseudocódigo:

Algoritmo Numero_de_la_suerte

Definir dia, mes, ano, sum, n_suerte Como Entero;

Definir seguir Como Caracter ;

seguir = "S"

Mientras seguir = "S" Hacer

 Escribir "Ingrese el día de nacimiento:";

 Leer dia;

 Escribir "Ingrese el mes de nacimiento:";

 Leer mes;

 Escribir "Ingrese el año de nacimiento:";

 Leer ano;

 Mientras dia < 0 O mes < 0 O ano < 0 Hacer

 Escribir "No se permiten valores negativos. Ingrese nuevamente su fecha de nacimiento:"

 Escribir "Ingrese el día de nacimiento:";

 Leer dia;

 Escribir "Ingrese el mes de nacimiento:";

 Leer mes;

 Escribir "Ingrese el año de nacimiento:";

 Leer ano;

 FinMientras

 Si dia >= 1 Y mes >= 1 Y mes <= 12 Y ano >= 1000 Y ano <= 9999 Entonces

 Segun mes Hacer

 Caso 1, 3, 5, 7, 8, 10, 12:

 Si dia >= 1 Y dia <= 31 Entonces

 sum = dia + mes + ano;

 Sino

Escribir "El día es inválido para este mes.";

FinSi

Caso 4, 6, 9, 11:

Si $\text{dia} \geq 1$ Y $\text{dia} \leq 30$ Entonces

sum = dia + mes + ano;

Sino

Escribir "El día es inválido para este mes.";

FinSi

Caso 2:

Si ($\text{dia} \geq 1$ Y $\text{dia} \leq 29$ Y $\text{ano} \% 4 = 0$ Y ($\text{ano} / 100 < 0$ O $\text{ano} / 400 = 0$)) Entonces

sum = dia + mes + ano;

Sino

Si ($\text{dia} \geq 1$ Y $\text{dia} \leq 28$) Entonces

sum = dia + mes + ano;

Sino

Escribir "El día es inválido para el mes de febrero.";

FinSi

FinSi

De Otro Modo:

Escribir "El mes es inválido.";

FinSegun

Mientras sum ≥ 10 Hacer

n_suerte = 0;

Mientras sum > 0 Hacer

n_suerte = n_suerte + (sum Mod 10);

sum = Trunc(sum / 10);

FinMientras

sum = n_suerte;

FinMientras

Escribir "Su número de la suerte es:", sum;

Sino

Escribir "La fecha es inválida.";

FinSi

Escribir "Para finalizar escribe (F) y para seguir jugando escribe (S)";

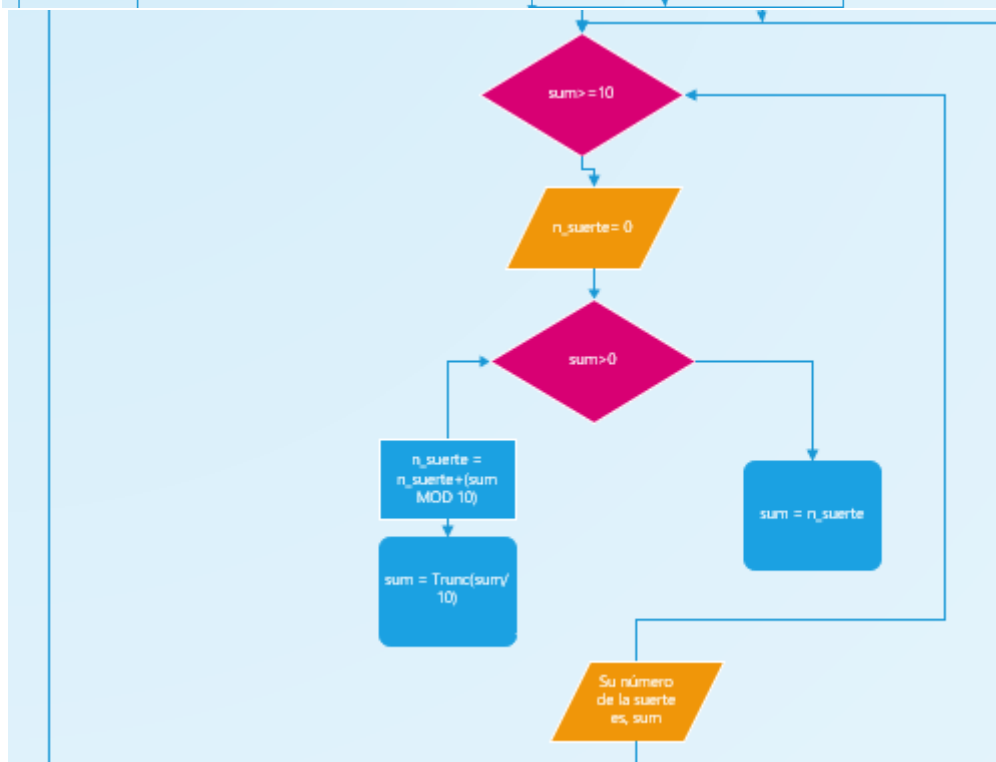
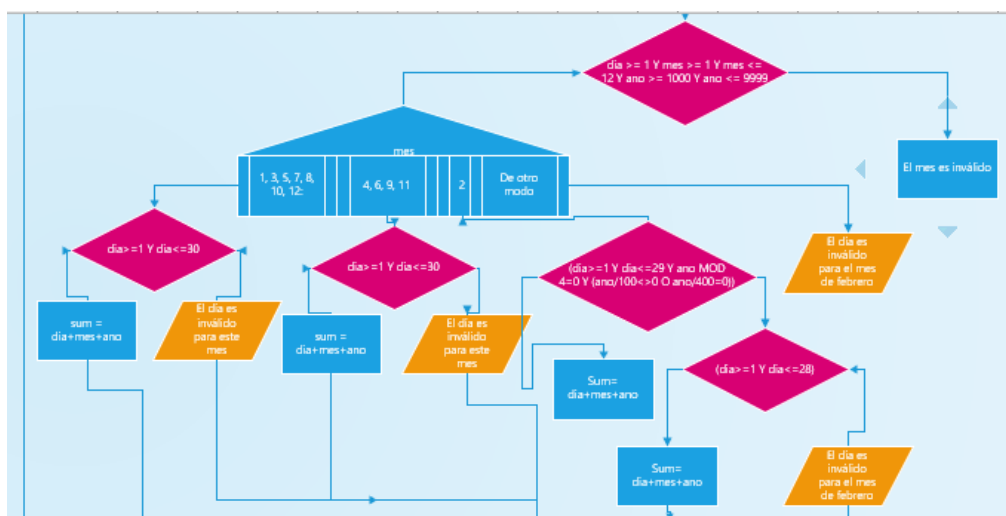
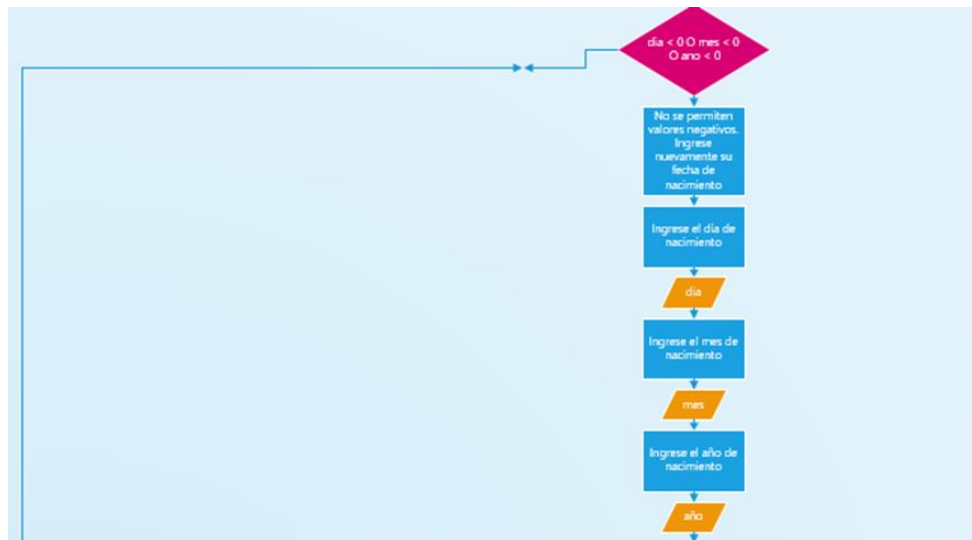
Leer seguir;

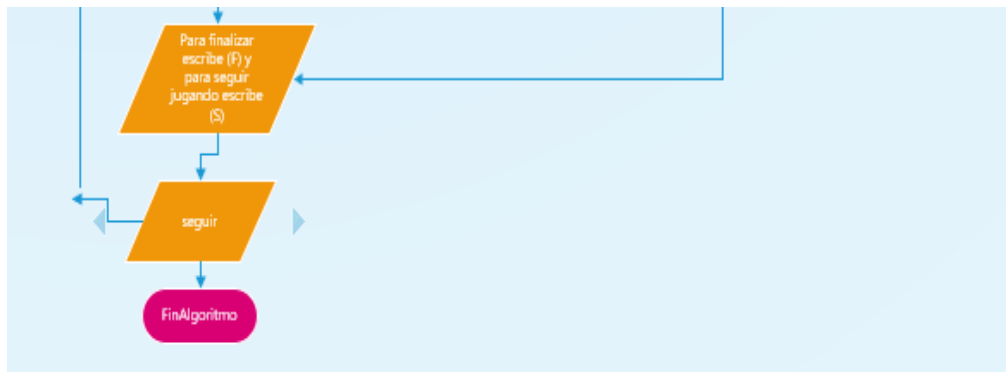
FinMientras

FinAlgoritmo

Diagrama de flujo:







Prueba de escritorio:

P a s o	d i a	m e s	a ñ o	d i a > => =1 Y m e s < =1 2 Y a n o > =1 0 0 0 Y a n o < =9 9	d i a > => Y d i a < =1 Y m e s < =3 1	d i a > => Y d i a < =1 Y m e s < =3 1	(dia> =1 Y dia<= 29 Y ano MOD 4=0 Y (ano/ 100<> 0 0 ano/4 00=0))	(di a>= 1 Y dia <=2 8)	su m> =1 0	s u m > 0	n_suerte= n_suerte+ (sum MOD 10)	sum = Trunc(sum/10)	sum = n_s uer te	Par a fin ali zar esc rib e (F) y par a seg uir jug and o esc rib e (S)	Salida
------------------	-------------	-------------	-------------	--	---	---	---	---------------------------------------	---------------------	-----------------------	---	--------------------------------	------------------------------	---	--------

				9 9											
1	1 5														
2		0 7													
3			2 0 0 6												
4				3											
5									8						
6										11					
7											3	f			
8															Su número de la suerte es:3 Para finali zar escrib e (F) y para seguir jugand o escrib e (S) > f *** Ejecuc ión Finali zada** *

Resolución de Mateo, Propuesta 2:

Entrada	Proceso	Salida
día= Fecha en días mes = Fecha en meses año = Fecha en años	suma = día + mes + año lucky_n = lucky_n + (suma % 10) suma = suma / 10	Suerte & lucky_n = Numero de la suerte suma= suma de digitos

Pseudocódigo:

Algoritmo Lucky_Number

```
//Variables
```

Definir dia, mes, año, suma, lucky_n Como Entero

Definir continuar Como Caracter

continuar ="S"

Mientras continuar = "S" o continuar = "s"

Escribir "Ingrese su fecha de nacimiento (dd-mm-aa). Ingrese 0 para salir."

Leer dia

Leer mes

Leer año

// Validar la fecha

Segun mes Hacer

1:

Si (dia>=1 y dia<=31) Y (año >= 1000 y año <= 9999) Entonces;

suma = dia + mes + año;

SiNo

Mostrar "Fecha fuera de rango";

Fin Si

2:

Si (dia>=1 y dia<=28) Y (año >= 1000 y año <= 9999) Entonces;

suma = dia + mes + año;

SiNo

Mostrar "Fecha fuera de rango";

FinSi

3:

Si (dia>=1 y dia<=31) Y (año >= 1000 y año <= 9999) Entonces;

suma = dia + mes + año;

SiNo

Mostrar "Fecha fuera de rango";

Fin Si

4:

Si (dia>=1 y dia<=30) Y (año >= 1000 y año <= 9999) Entonces;

suma = dia + mes + año;

SiNo

Mostrar "Fecha fuera de rango";

Fin Si

5:

Si (dia>=1 y dia<=31) Y (año >= 1000 y año <= 9999) Entonces;

suma = dia + mes + año;

SiNo

Mostrar "Fecha fuera de rango";

Fin Si

6:

Si (dia>=1 y dia<=30) Y (año >= 1000 y año <= 9999) Entonces;

suma = dia + mes + año;

SiNo

Mostrar "Fecha fuera de rango";

Fin Si

7:

Si (dia>=1 y dia<=31) Y (año >= 1000 y año <= 9999) Entonces;

suma = dia + mes + año;

SiNo

Mostrar "Fecha fuera de rango";

Fin Si

8:

Si (dia>=1 y dia<=31) Y (año >= 1000 y año <= 9999) Entonces;

suma = dia + mes + año;

SiNo

Mostrar "Fecha fuera de rango";

Fin Si

9:

Si (dia>=1 y dia<=30) Y (año >= 1000 y año <= 9999) Entonces;

suma = dia + mes + año;

SiNo

Mostrar "Fecha fuera de rango";

Fin Si

10:

Si (dia>=1 y dia<=31) Y (año >= 1000 y año <= 9999) Entonces;

suma = dia + mes + año;

SiNo

Mostrar "Fecha fuera de rango";

Fin Si

11:

Si (dia>=1 y dia<=30) Y (año >= 1000 y año <= 9999) Entonces;

suma = dia + mes + año;

SiNo

Mostrar "Fecha fuera de rango";

Fin Si

12:

Si (dia>=1 y dia<=31) Y (año >= 1000 y año <= 9999) Entonces;

suma = dia + mes + año;

SiNo

Mostrar "Fecha fuera de rango";

Fin Si

De Otro Modo:

Mostrar "Fecha inválida"

Fin Segun

// Calcular el número de la suerte

Mientras suma>=10

lucky_n<-0

Mientras suma>0

lucky_n=lucky_n+(suma % 10)

suma=Trunc(suma /10)

FinMientras

suma = lucky_n

FinMientras

Escribir "Su número de la suerte es: ", suma

Escribir "¿Desea calcular otro número de la suerte? (S/N)"

Leer continuar

FinMientras

FinAlgoritmo

Código en C:

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

int main() {
    // Variables
    int dia, mes, año, suma, lucky_n;
    char continuar = 'S';

    while (continuar == 'S' || continuar == 's') {
        printf("Ingrese su fecha de nacimiento (dd-mm-aa). Ingrese 0 para salir.\n");
        printf("Día: ");
        scanf("%d", &dia);
        if (dia == 0) break;
        printf("Mes: ");
        scanf("%d", &mes);
        printf("Año: ");
        scanf("%d", &año);

        // Validar la fecha
        switch (mes) {
            case 1: case 3: case 5: case 7: case 8: case 10: case 12:
                if (dia >= 1 && dia <= 31 && año >= 1000 && año <= 9999) {
                    suma = dia + mes + año;
                } else {
                    printf("Fecha fuera de rango\n");
                    continue;
                }
                break;

            case 2:
                if (dia >= 1 && dia <= 28 && año >= 1000 && año <= 9999) {
                    suma = dia + mes + año;
                } else {
                    printf("Fecha fuera de rango\n");
                    continue;
                }
                break;
        }
    }
}
```

```

    }
    break;

    case 4: case 6: case 9: case 11:
        if (dia >= 1 && dia <= 30 && año >= 1000 && año <= 9999) {
            suma = dia + mes + año;
        } else {
            printf("Fecha fuera de rango\n");
            continue;
        }
        break;

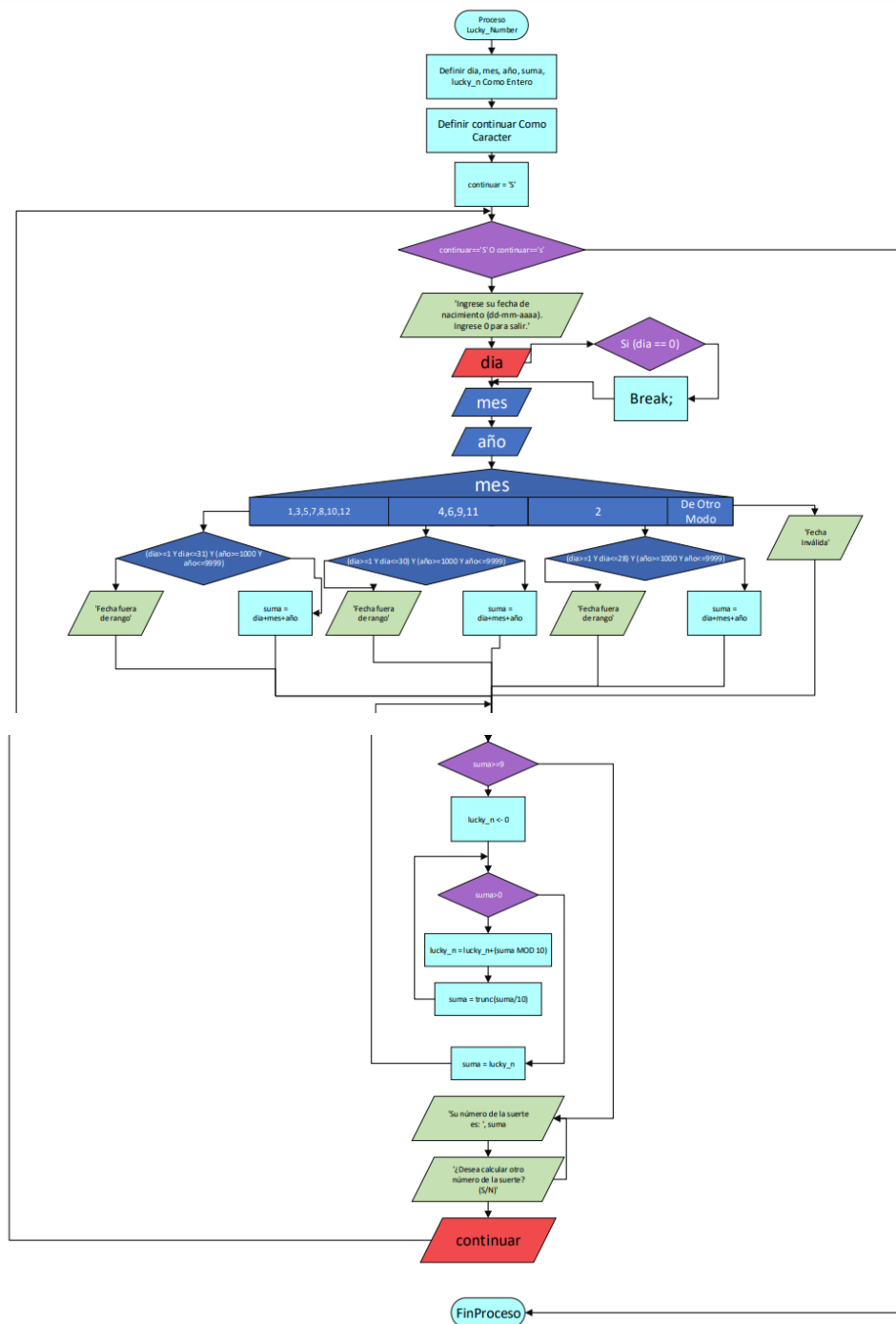
    default:
        printf("Fecha inválida\n");
        continue;
}

// Calcular el número de la suerte
while (suma >= 9) {
    lucky_n = 0;
    while (suma > 0) {
        lucky_n = lucky_n + (suma % 10);
        suma = suma / 10;
    }
    suma = lucky_n;
}

printf("Su número de la suerte es: %d\n", suma);
printf("¿Desea calcular otro número de la suerte? (S/N)\n");
scanf(" %c", &continuar);
}
getch;
return 0;
}

```

Diagrama de código



Prueba de escritorio

2		18																
3			5															
4				2004														
5					5													
6						T r u e												
7						T r u e												
8										2027								
9										True								
10											0							
11												True						
12													7					
13														202				
14															20			
15												True						
16													9					
17														20				
18															20			
19												True						

20															9				
21																2			
22																	2		
23													True						
24														11					
25															1				
26																1			
27													True						
28														1					
29															0				
30																	0		
31													False						
32											11								
33													True						
34														1					
35															1				
36																1			
37														2					
38															0				
39																	0		
40													False						

Propuesta 1: Introduce más funciones y modularidad, lo cual puede ser positivo, pero a costa de una mayor complejidad. La función “FechaValida” mejora la reutilización del código, pero la lógica general está más dispersa.

Propuesta 2: El código es más simple y directo. La lógica para la validación de la fecha está claramente encapsulada dentro del switch y las instrucciones para calcular el número de la suerte son fáciles de seguir.

3. Validación de Fecha:

Propuesta 1: Incluye validación de años bisiestos y se adapta al año actual, proporcionando una validación más robusta.

Propuesta 2: Realiza validaciones básicas de fecha, pero no considera años bisiestos.

4. Uso de Funciones:

Propuesta 1: Utiliza varias funciones, lo que mejora la estructura y la modularidad del código. Esto facilita el mantenimiento y la actualización del código.

Propuesta 2: No utiliza funciones auxiliares, lo que hace que todo el proceso esté contenido en “main”, lo cual puede ser más fácil de entender para principiantes.

5. Interacción con el Usuario:

Propuesta 1: También permite múltiples intentos hasta que se ingrese una fecha válida, pero lo hace a través de un enfoque de bucle “do-while”.

Propuesta 2: Utiliza un bucle “while” para permitir múltiples entradas del usuario, haciendo el programa más interactivo.

En conclusión, la propuesta 1 se destaca como la mejor opción debido a su estructura modular, validación robusta de fechas y capacidad de reutilización del código. Es ideal para aplicaciones que requieren precisión en el manejo de datos temporales y que pueden beneficiarse de una arquitectura más organizada y adaptable. Su enfoque en el módulo y la validación detallada lo hace particularmente adecuado para proyectos más complejos donde la fiabilidad y la mantenibilidad son cruciales.