Objetivo: Identificar elementos de entrada y resultados esperados de problemas planteados mediante diagramas de flujo

GA3-220501093-AA2-EV01

Isidro J Gallardo Navarro

Ficha:3070299

2025

Tecnología en Análisis y Desarrollo de Software.

ADSO

Lista de chequeo a cubrir:

●

1. Introducción

Este documento presenta el desarrollo de algoritmos fundamentales utilizando programación estructurada, con énfasis en la identificación precisa de datos de entrada, procesos de transformación y resultados esperados. La metodología se centra en la representación visual

mediante diagramas de flujo y la implementación en pseudocódigo estructurado.
El algoritmo principal abordado corresponde al cálculo de edad actual de una persona a partir de su fecha de nacimiento y la fecha actual, demostrando el uso de estructuras de control y operaciones aritméticas con fechas.

2. Algoritmo Principal: Cálculo de Edad Actual

2.1 Análisis del Problema

Descripción: Desarrollar un algoritmo que determine la edad en años de una persona, dados su fecha de nacimiento y la fecha actual.

Datos de Entrada:

Fecha de nacimiento (día, mes, año)

Fecha actual (día, mes, año)

Datos de Salida:

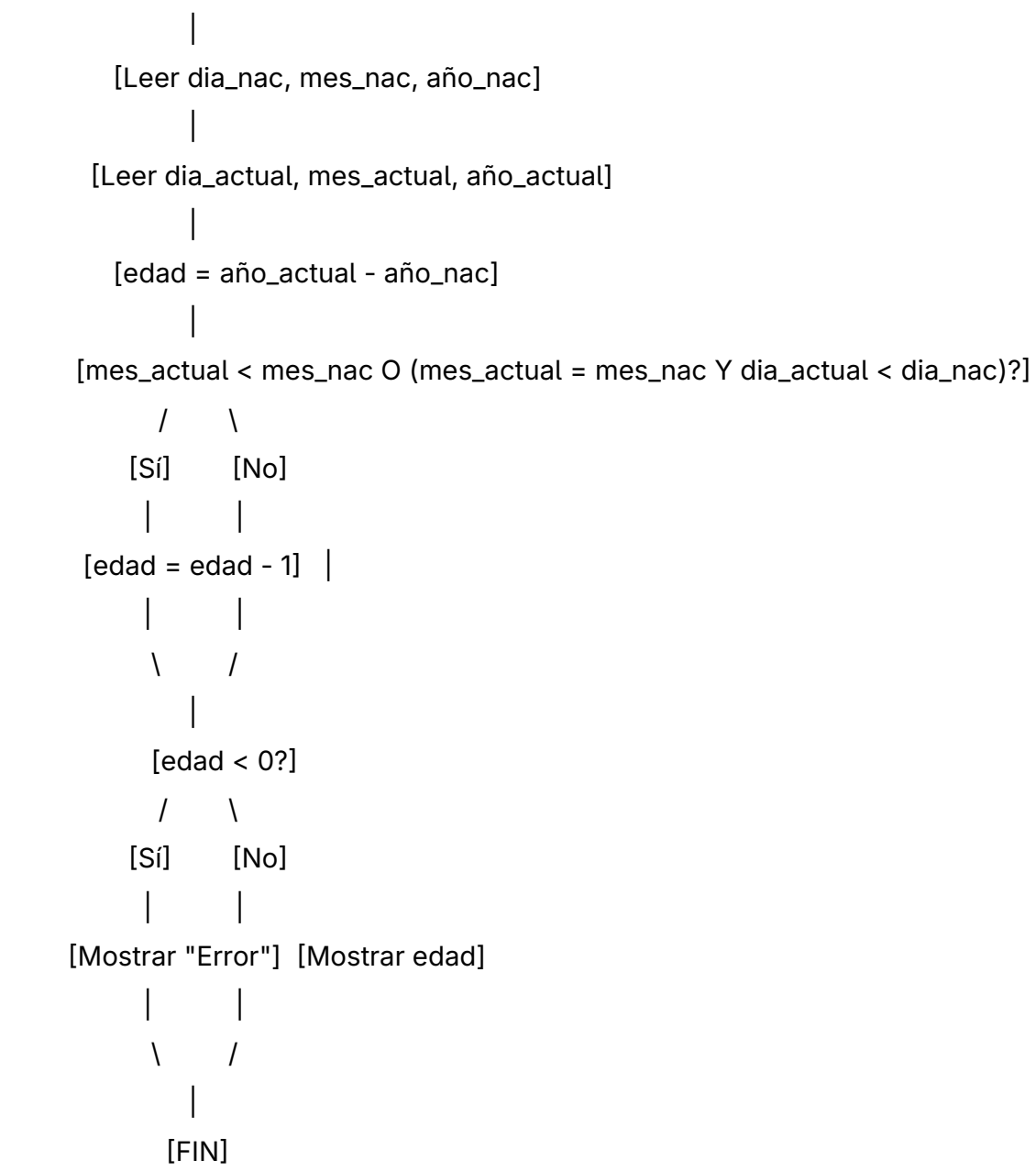
Edad actual de la persona en años

Proceso: Calcular la diferencia entre las fechas considerando si ya cumplió años en el año actual.

2.2 Pseudocódigo Estructurado

```
Algoritmo CalcularEdadActual
    // Declaración de variables
    Definir dia_nac, mes_nac, año_nac Como Entero
    Definir dia_actual, mes_actual, año_actual Como Entero
    Definir edad Como Entero

    // Entrada de datos
    Escribir "═══ CÁLCULO DE EDAD ACTUAL ═══"
    Escribir "Ingrese su fecha de nacimiento:"
    Escribir "Día: "
    Leer dia_nac
    Escribir "Mes: "
    Leer mes_nac
    Escribir "Año: "
    Leer año_nac

    Escribir "Ingrese la fecha actual:"
    Escribir "Día: "
    Leer dia_actual
    Escribir "Mes: "
    Leer mes_actual
    Escribir "Año: "
    Leer año_actual

    // Proceso de cálculo
    edad ← año_actual - año_nac

    // Verificar si ya cumplió años este año
    Si (mes_actual < mes_nac) O (mes_actual = mes_nac Y dia_actual < dia_nac) Entonces
        edad ← edad - 1
    FinSi

    // Validación de resultado
    Si edad < 0 Entonces
        Escribir "Error: La fecha de nacimiento es posterior a la fecha actual"
    Sino
        Escribir "Su edad actual es: ", edad, " años"
    FinSi

FinAlgoritmo
```

2.3 Diagrama de Flujo



2.4 Prueba Funcional

Caso de Prueba 1:

Entrada: Nacimiento: 15/03/1995, Actual: 10/08/2024

Proceso: 2024 - 1995 = 29, ya cumplió años (marzo < agosto)

Salida: 29 años ✓

Caso de Prueba 2:

Entrada: Nacimiento: 20/12/1990, Actual: 10/08/2024Proceso: 2024 - 1990 = 34, no ha cumplido años (agosto < diciembre)
Salida: 33 años ✓

Caso de Prueba 3:

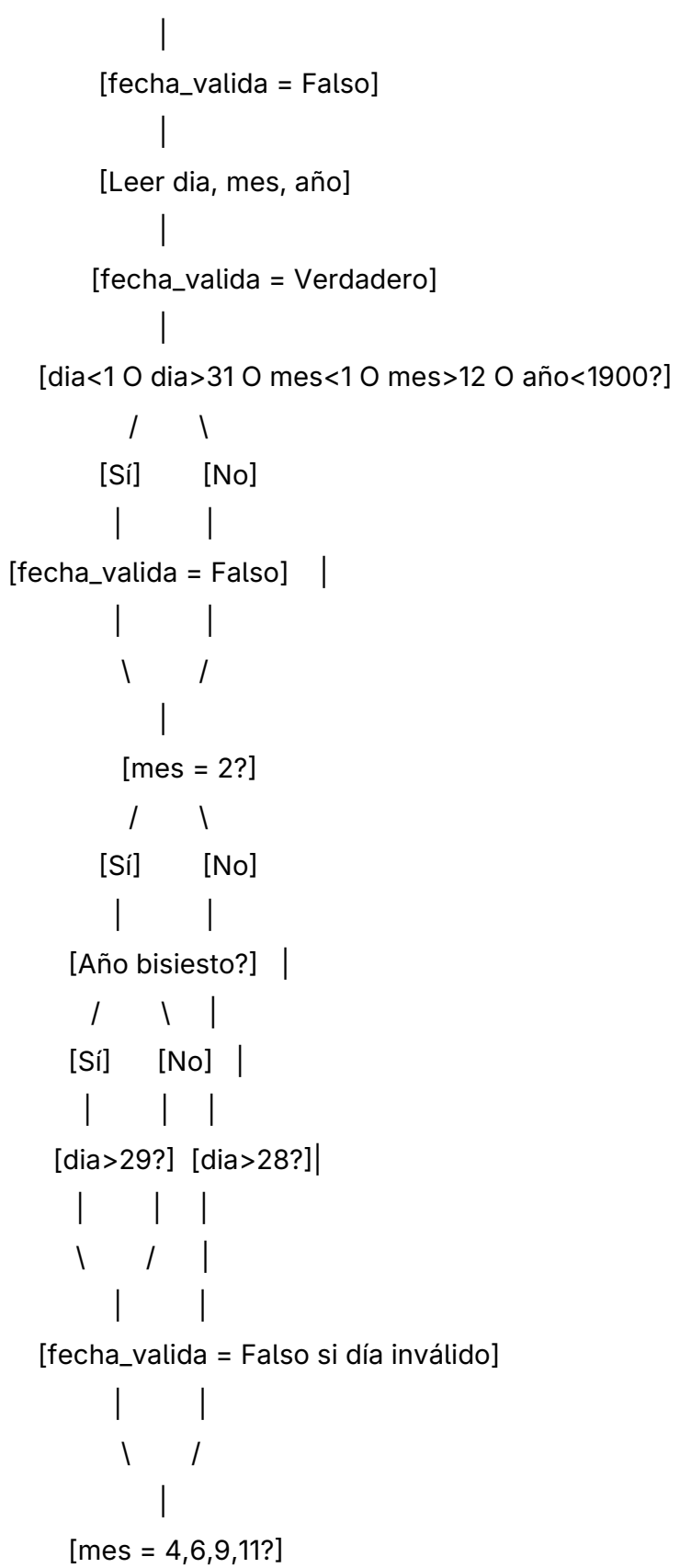
Entrada: Nacimiento: 10/08/2000, Actual: 10/08/2024
Proceso: 2024 - 2000 = 24, cumple años hoy
Salida: 24 años ✓

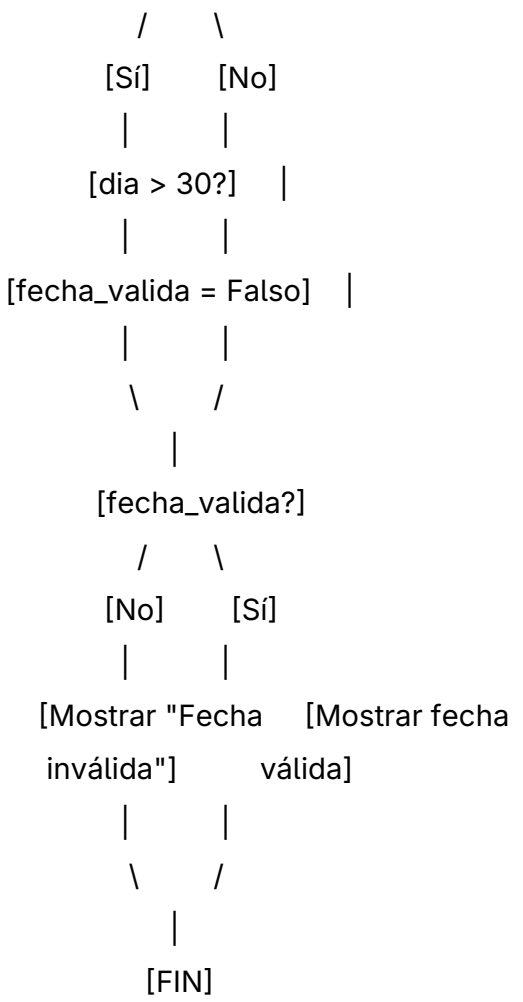
3. Algoritmos Complementarios con Estructuras Cíclicas

3.1 Algoritmo de Validación de Fechas

```
Algoritmo ValidarFecha
    Definir dia, mes, año Como Entero
    Definir fecha_valida Como Logico

    Repetir
        Escribir "Ingrese día (1-31): "
        Leer dia
        Escribir "Ingrese mes (1-12): "
        Leer mes
        Escribir "Ingrese año: "
        Leer año

        fecha_valida ← Verdadero

        // Validar rango básico
        Si dia < 1 O dia > 31 O mes < 1 O mes > 12 O año < 1900 Entonces
            fecha_valida ← Falso
        FinSi

        // Validar días específicos por mes
        Segun mes Hacer
            2: // Febrero
                Si (año % 4 = 0 Y año % 100 <> 0) O (año % 400 = 0) Entonces
                    Si dia > 29 Entonces fecha_valida ← Falso FinSi
                Sino
                    Si dia > 28 Entonces fecha_valida ← Falso FinSi
                FinSi
            4,6,9,11: // Meses de 30 días
                Si dia > 30 Entonces fecha_valida ← Falso FinSi
        FinSegun

        Si NO fecha_valida Entonces
            Escribir "Fecha inválida. Intente nuevamente."
        FinSi

    Hasta Que fecha_valida

    Escribir "Fecha válida: ", dia, "/", mes, "/", año
FinAlgoritmo
```

3.2 Diagrama de Flujo - Validación de Fechas





3.3 Algoritmo de Cálculo de Diferencia de Fechas

```
Algoritmo DiferenciaFechas
    Definir dia1, mes1, año1 Como Entero
    Definir dia2, mes2, año2 Como Entero
    Definir diferencia_dias Como Entero
    Definir dias_mes Como Entero
    Definir i Como Entero

    // Entrada de fechas
    Escribir "Primera fecha:"
    Leer dia1, mes1, año1
    Escribir "Segunda fecha:"
    Leer dia2, mes2, año2

    // Convertir fechas a días desde una referencia
    diferencia_dias ← 0

    // Calcular días transcurridos desde año base (1900)
    Para i ← 1900 Hasta año2-1 Con Paso 1 Hacer
        Si (i % 4 = 0 Y i % 100 <> 0) O (i % 400 = 0) Entonces
            diferencia_dias ← diferencia_dias + 366
        Sino
            diferencia_dias ← diferencia_dias + 365
        FinSi
    FinPara

    // Sumar días de los meses del año actual
    Para i ← 1 Hasta mes2-1 Con Paso 1 Hacer
        Segun i Hacer
            1,3,5,7,8,10,12: dias_mes ← 31
            4,6,9,11: dias_mes ← 30
            2:
                Si (año2 % 4 = 0 Y año2 % 100 <> 0) O (año2 % 400 = 0) Entonces
                    dias_mes ← 29
                Sino
                    dias_mes ← 28
                FinSi
        FinSegun
        diferencia_dias ← diferencia_dias + dias_mes
    FinPara

    // Sumar días del mes actual
    diferencia_dias ← diferencia_dias + dia2

    // Restar días de la primera fecha (proceso similar)
    // [Código similar para fecha1 ... ]

    Escribir "Diferencia en días: ", diferencia_dias
FinAlgoritmo
```

4. Análisis de Estructuras de Control

4.1 Estructuras Secuenciales

Los algoritmos presentados utilizan estructuras secuenciales para:

Declaración de variables
Entrada de datos
Cálculos aritméticos básicos
Salida de resultados

4.2 Estructuras Condicionales
Estructura Si-Entonces-Sino:

Validación de fechas
Verificación de cumpleaños
Manejo de errores

Estructura Según-Hacer:

Validación de días por mes
Cálculo de días en meses específicos

4.3 Estructuras Cíclicas
Estructura Repetir-Hasta Que:

Validación de entrada de datos
Repetición hasta obtener datos válidos

Estructura Para:

Iteración sobre rangos de años
Cálculo acumulativo de días

4.4 Ventajas de la Programación Estructurada

Legibilidad: El código es fácil de leer y entender
Mantenibilidad: Facilita las modificaciones y correcciones

Reutilización: Los módulos pueden ser reutilizados

Depuración: Simplifica la localización de errores

Documentación: El flujo lógico es evidente

Conclusiones

La implementación de algoritmos para el cálculo de edad demuestra la aplicación práctica de los fundamentos de programación estructurada. El uso de diagramas de flujo facilita la comprensión del problema y la identificación de casos especiales.

Las estructuras cíclicas permiten implementar validaciones robustas, mientras que las estructuras condicionales manejan la lógica específica del dominio del problema. La metodología estructurada garantiza código mantenible, legible y verificable.

Herramientas recomendadas: PSeInt para desarrollo de pseudocódigo y generación automática de diagramas de flujo, facilitando la transición hacia lenguajes de programación específicos.