

Fundamentos de programación estructurada y estructuras cíclicas

Danny Julián Perilla Mikán

Mayo, 2024

1 Introducción

El proceso de construcción de un algoritmo de principio a fin consta de 3 fases: análisis, diseño e implementación. En la fase de análisis y diseño se usan los pseudocódigos y diagramas de flujo. La etapa de implementación se realiza en un lenguaje de programación propiamente.

2 Problemas

1. Construir un algoritmo que a partir de una fecha de nacimiento y una fecha actual, determine la edad en años actual de una persona.
2. Construir un algoritmo que permita determinar si un año dado es bisiesto o no.

3 Desarrollo

3.1 Pseudocódigo y diagramas de flujo:

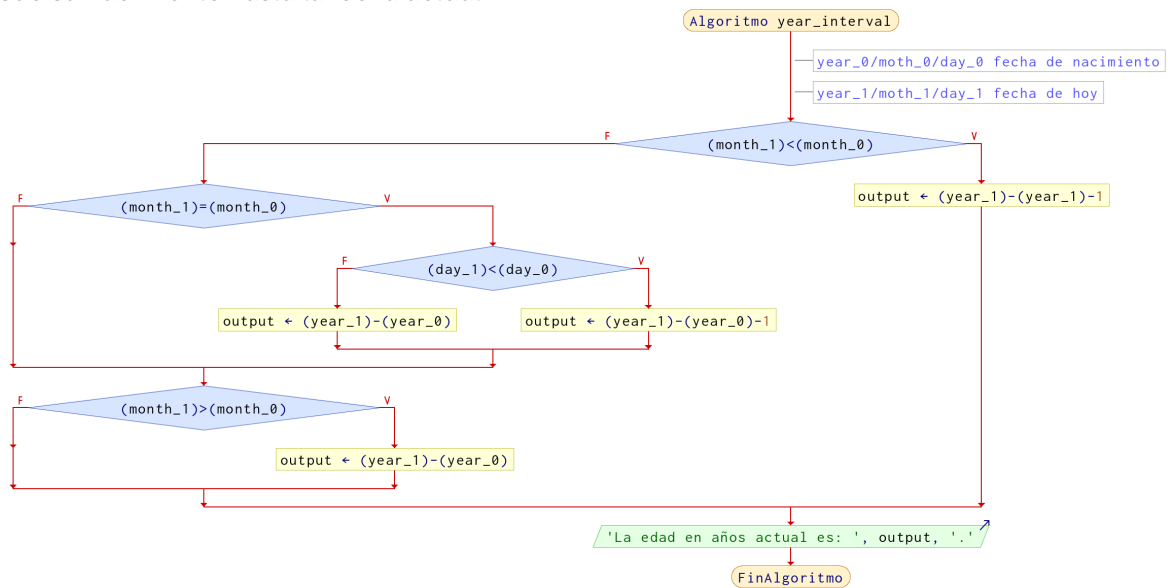
```
1 Algoritmo intervalo_de_anos
2   // year_0/month_0/day_0 fecha de nacimiento
3   // year_1/month_1/day_1 fecha actual
4   Si (month_1) < (month_0) Entonces
5     output <- (year_1) - (year_0) - 1
6   SiNo
7     Si (month_1) = (month_0) Entonces
8       Si (day_1) < (day_0) Entonces
9         output <- (year_1) - (year_0) - 1
10      SiNo
11        output <- (year_1) - (year_0)
12      FinSi
13    FinSi
14    Si (month_1) > (month_0) Entonces
15      output <- (year_1) - (year_0)
16    FinSi
17  FinSi
18  Escribir "La edad en años es: ", output, "."
19 FinAlgoritmo
```

```

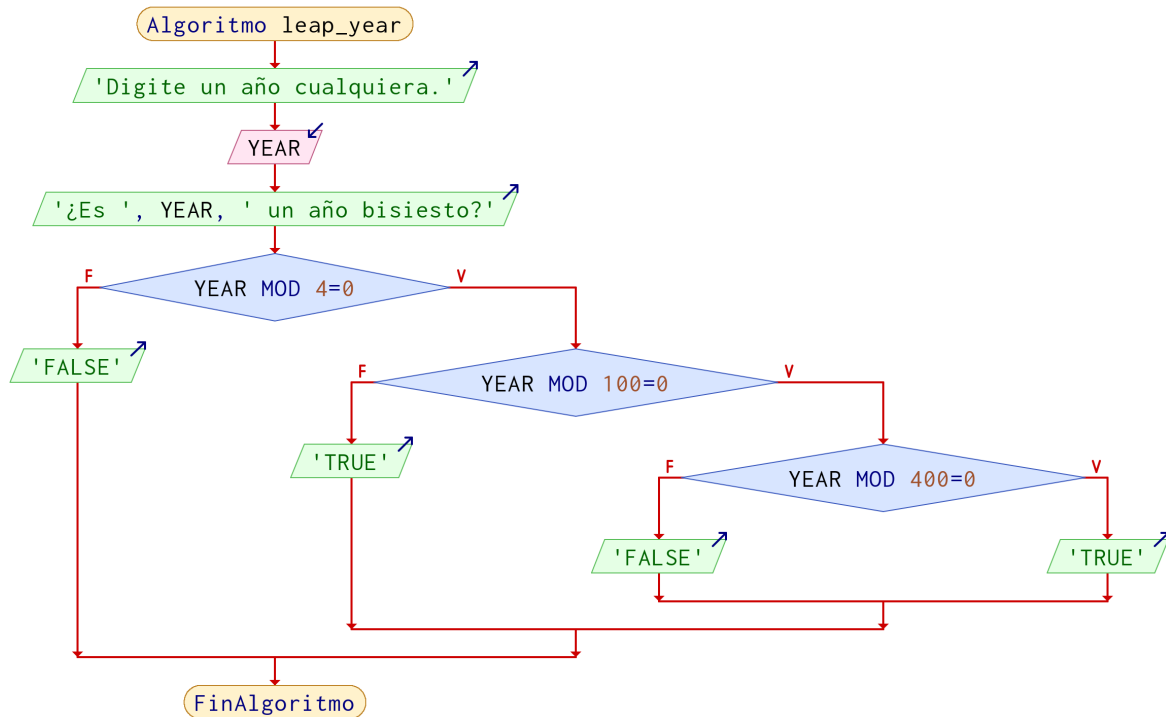
1 Algoritmo bisiestos
2   Escribir 'Digite un año cualquiera.'
3   Leer YEAR
4   Escribir 'Es ', YEAR, ' un año bisiestos?'
5   Si YEAR MOD 4=0 Entonces
6     Si YEAR MOD 100=0 Entonces
7       Si YEAR MOD 400=0 Entonces
8         Escribir 'TRUE'
9       SiNo
10        Escribir 'FALSE'
11      FinSi
12    SiNo
13      Escribir 'TRUE'
14    FinSi
15  SiNo
16    Escribir 'FALSE'
17  FinSi
18 FinAlgoritmo
  
```

3.2 Diagramas de flujo:

El primer diagrama de flujo corresponde al algoritmo que calcula el intervalo de años de una persona desde su nacimiento hasta la fecha actual.



El segundo diagrama de flujo corresponde al algoritmo para calcular si un año determinado es bisiestos o no.



3.3 Implementación en Java:

Para ver la respectiva implementación en Java entre al siguiente link: [github/dmikan](https://github.com/dmikan).

4 Conclusiones

Los diagramas de flujo nos permiten analizar desde una perspectiva lógica un algoritmo antes de implementarlo, en este caso en Java, por lo cual su uso permite un buen diseño y entendimiento del algoritmo.