

14. Sabiendo que el día 1 de enero de 1.583 fue sábado y que desde entonces no se han producido modificaciones en el calendario, elaborar un algoritmo para calcular el día de la semana en que cayó o caerá una fecha cualquiera posterior al 1 de enero de 1.583. Nota: Un año se considera bisiesto cuando es divisible por 400 o bien es divisible por 4 pero no por 100.

```
modulo es_bisiesto
parametros_entrada: año: ENTERO
parametros_salida: bisiesto: LÓGICO
inicio
  si (año mod 400 = 0) OR ((año mod 4 = 0) AND (año mod 100 <>0)) entonces
    bisiesto ← VERDADERO
  si_no
    bisiesto ← FALSO
  fin_si
fin
```

```
modulo dias_año
parametros_entrada: A: ENTERO
parametros_salida: num: ENTERO
variables: B: LÓGICO
inicio
  es_bisiesto (A; B)
  si B entonces
    num ← 366
  si_no
    num ← 365
  fin_si
fin
```

Ingeniería Técnica de Informática de Gestión

Programación I: GRUPO 1 y GRUPO 3

Ejercicio 14

```
modulo dias_mes
parametros_entrada: M, A : ENTERO
parametros_salida: num: ENTERO
variables: B: LÓGICO
inicio
  según_el_caso M
    caso 1: num ← 31
    caso 2: inicio
      es_bisiesto (A; B)
      si B entonces
        num ← 29
      si_no
        num ← 28
      fin_si
    fin
    caso 3: num ← 31
    caso 4: num ← 30
    caso 5: num ← 31
    caso 6: num ← 30
    caso 7: num ← 31
    caso 8: num ← 31
    caso 9: num ← 30
    caso 10: num ← 31
    caso 11: num ← 30
    caso 12: num ← 31
  fin_según_el_caso
fin
```

Ingeniería Técnica de Informática de Gestión

Programación I: GRUPO 1 y GRUPO 3

Ejercicio 14

```

modulo dias_entre_fechas
parametros_entrada: DI, MI, AI, DF, MF, AF: ENTERO
parametros_salida: num: ENTERO
variables: dias, mes, año: ENTERO
inicio
  num ← 0

  si (AI+1) <= (AF-1) entonces
    año ← (AI+1)
    mientras año <= (AF-1) hacer
      dias_año (año; dias)
      num ← num + dias
      año ← año + 1
    fin_mientras
  fin_si

  si (AI <> AF) entonces
    si (MI < 12) entonces
      mes ← (MI+1)
      mientras mes <= 12 hacer
        dias_mes (mes, AI; dias)
        num ← num + dias
        mes ← mes + 1
      fin_mientras
    fin_si
    si (MF > 1) entonces
      mes ← 1
      mientras mes <= (MF-1) hacer
        dias_mes (mes, AF; dias)
        num ← num + dias
        mes ← mes + 1
      fin_mientras
    fin_si
  si_no (*AI = AF*)
    si (MI+1) <= (MF-1) entonces
      mes ← (MI+1)
      mientras mes <= (MF-1) hacer
        dias_mes (mes, AI; dias)
        num ← num + dias
        mes ← mes + 1
      fin_mientras
    fin_si
  fin_si

  si (AI = AF) AND (MI = MF) entonces
    num ← num + (DF-DI)
  si_no
    dias_mes (MI, AI; dias)
    num ← num + (dias-DI+1)
    num ← num + (DF-1)
  fin_si
fin

```

Ingeniería Técnica de Informática de Gestión

Programación I: GRUPO 1 y GRUPO 3

Ejercicio 14

modulo validar_fecha

parametros_entrada: DI, MI, AI, DF, MF, AF: ENTERO

parametros_salida: valido: LÓGICO

variables: dias1, dias2: ENTERO

inicio

si (MI<1) OR (MF<1) OR (MI>12) OR (MF>12) OR (DI<1) OR (DF<1) OR (DI>31) OR (DF>31) **entonces**
valido ← FALSO

si_no

si (AI>AF) **entonces**
valido ← FALSO

si_no

si (AI<AF) **entonces**
dias_mes (MI, AI; dias1)
dias_mes (MF, AF; dias2)
si (DI ≤ dias1) AND (DF ≤ dias2) **entonces**
valido ← VERDADERO

si_no
valido ← FALSO

fin_si

si_no (*AI=AF *)
si (MI>MF) **entonces**
valido ← FALSO

si_no

si (MI<MF) **entonces**
dias_mes (MI, AI; dias1)
dias_mes (MF, AF; dias2)
si (DI ≤ dias1) AND (DF ≤ dias2) **entonces**
valido ← VERDADERO

si_no
Valido ← FALSO

fin_si

si_no (*MI=MF y se sigue cumpliendo que AI=AF*)
si (DI>DF) **entonces**
Valido ← FALSO

si_no

dias_mes (MI, AI; dias1)
si (DI ≤ dias1) AND (DF ≤ dias1) **entonces**
valido ← VERDADERO

si_no
valido ← FALSO

fin_si

fin_si

fin_si

fin_si

fin_si

fin_si

fin_si

fin

Ingeniería Técnica de Informática de Gestión

Programación I: GRUPO 1 y GRUPO 3

Ejercicio 14

algoritmo ejercicio14

variables

A, D, M, N: ENTERO

correcto: LOGICO

inicio

repetir

LEER (D, M, A)

validar_fecha (1, 1, 1583, D, M, A; correcto)

hasta_que correcto

dias_entre_fechas (1, 1, 1583, D, M, A; N)

$N \leftarrow N \bmod 7$

según_el_caso N

caso 0: ESCRIBIR ("Sabado")

caso 1: ESCRIBIR ("Domingo")

caso 2: ESCRIBIR ("Lunes")

caso 3: ESCRIBIR ("Martes")

caso 4: ESCRIBIR ("Miercoles")

caso 5: ESCRIBIR ("Jueves")

caso 6: ESCRIBIR ("Viernes")

fin_según_el_caso

fin