1 Introducción a la algoritmia

La aplicación apropiada de una serie de pasos detallados puede garantizar una solución correcta de un problema; es así como la programación de computadores o construcción de software se basa en esta práctica.



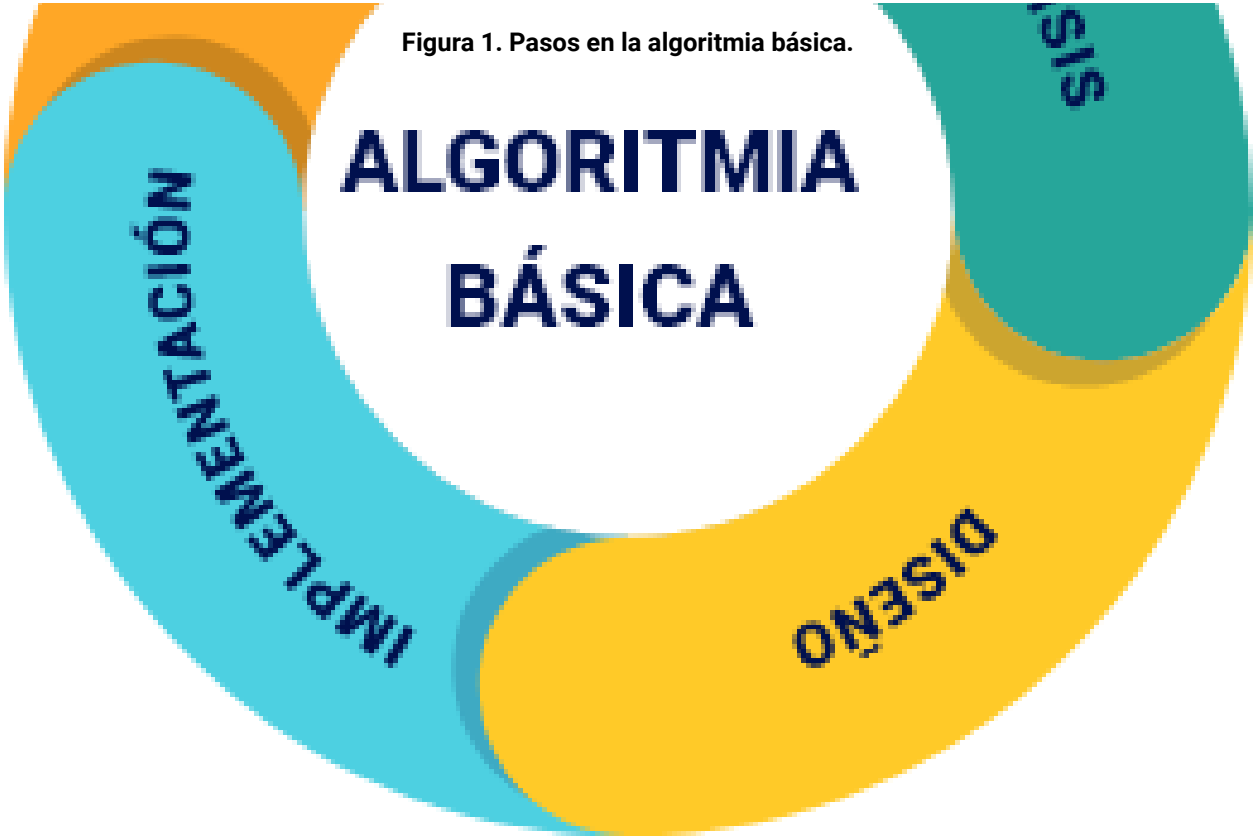


Figura 1. Pasos en la algoritmia básica.

1.1. Definición de algoritmo

Según la Real Academia Española (2020) un algoritmo está definido como un "**conjunto ordenado y finito de operaciones que permite hallar la solución de un problema"**.

Es importante comprender que computacionalmente las máquinas están en capacidad de hacer unas pocas operaciones matemáticas muy básicas y sencillas, pero también se considera que computan enormes cantidades de estas operaciones en unidades de

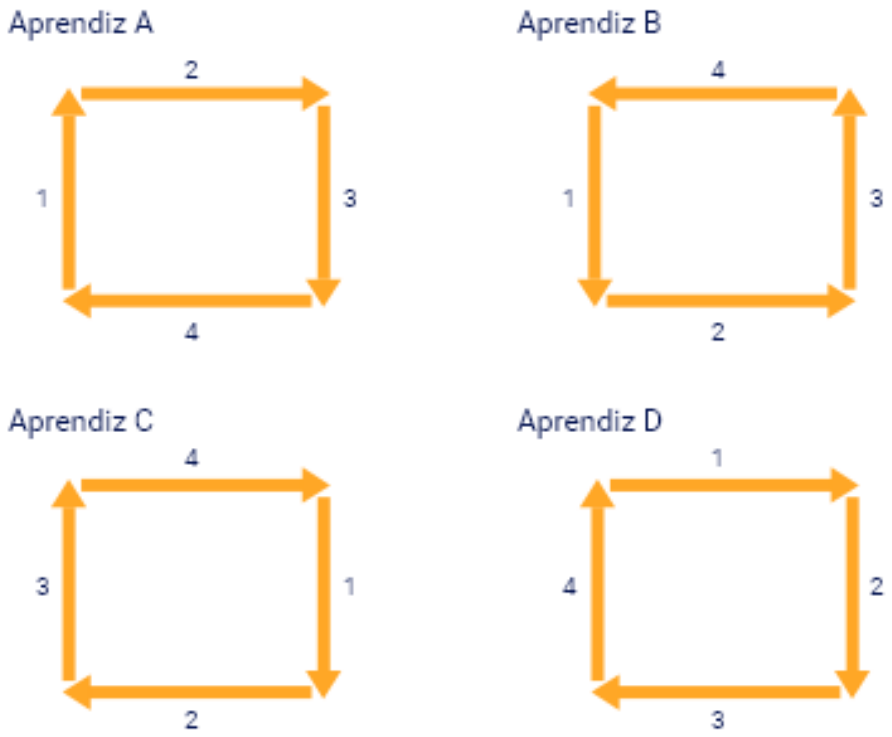
tiempo muy corto.

Entonces, la solución de un problema utilizando computadoras, fácilmente se puede convertir en una enorme cantidad de pequeñas operaciones matemáticas, tal vez abrumadoras desde el punto de vista humano. Es por eso que se requiere la especialización de conjuntos de operaciones, los cuales se agrupan en pequeños procesos que componen la solución de un problema.

Para ampliar la definición puede visualizar el siguiente video (https://www.youtube.com/watch?v=U3CGMyjzlvM)

1.2. Pensamiento algorítmico

Es así como un problema puede ser abordado de varias maneras y aportar la misma solución, pero lo que siempre tendrán en común es que para resolverlo usaron una secuencia de pasos ordenados, por ejemplo, si se pasa una hoja de papel cuadriculado a los aprendices de una clase y se les pide que dibujen un cuadrado sin levantar el lápiz, habrá quienes lo hagan de diferente manera y obtengan el mismo resultado como lo muestra la siguiente figura:



Como se observa en la figura el aprendiz A inició realizando una línea vertical hacia arriba, mientras que el aprendiz B una vertical hacia abajo, o el aprendiz D una horizontal hacia la derecha, y todos los siguientes llevaron a cabo un proceso que les permitió cumplir con la solución al problema planteado, de forma global prácticamente existe una gran cantidad de soluciones al mismo problema.

‹ Regresar

Figura 2. Pensamiento lógico y procedimental.

Continuar ›

Al imaginar que el problema es mucho más complejo que el descrito en el ejemplo anterior se concluye que existen tantas soluciones como el enfoque o lógica empleada por cada aprendiz o persona. Pero lo que sí es común en todos los casos es que se realizaron unas fases mentales secuencialmente ordenadas, las cuales son:

1. **Entender el problema enunciado por el instructor (entender el problema).**
2. **Escoger una ruta a seguir del rectángulo (hacer un plan).**
3. **Plasmar en el papel cuadriculado ese plan (ejecutar el plan).**
4. **Revisar el resultado y ver que se alcanzó el objetivo (revisar).**

Cuando el problema es muy complejo de resolver, muchas veces toca volver y analizar el problema, o trazar un nuevo plan o plasmar ese nuevo plan una y otra vez, revisando los resultados y volviendo a analizar hasta encontrar la solución del problema.

1.3. Solución de problemas y programación



‹ Regresar

Continuar ›

“Los algoritmos son una herramienta que permite describir claramente un conjunto finito de instrucciones"

> “En el ámbito de la computación los algoritmos son una herramienta que permite describir claramente un conjunto finito de instrucciones ordenadas secuencialmente y libres de ambigüedad, que debe llevar a cabo un computador para lograr un resultado previsible. Vale la pena recordar que un programa de computador consiste de una serie de instrucciones muy precisas y escritas en un lenguaje de programación que el computador entiende (Logo, Java, Pascal, etc.)” (López, 2019, p. 7).

Ejemplo 1: se propone diseñar un algoritmo para apagar una computadora.

```
     Algoritmo: apagar computadora.
INICIO
1.-Dar clic en el icono INICIO.
  2.-Dar clic en el icono que dice APAGAR.
    3.-Esperar que se apague completamente.
      4.-Desconectar los cables del enchufe.
FIN
```

Ejemplo 2: se propone diseñar un algoritmo para cargar un celular.

```
     Algoritmo: cargar un celular.
INICIO
1. Buscar el cargador.
  2. Agarrar el cargador.
    3. Agarrar el celular.
      4. Poner el enchufe en el celular.
        5. Conectarlo al tomacorriente.
          6. Esperar a que se ponga a cargar.
            7. Desconectarlo del tomacorriente.
              8. Ponerlo en un lugar seguro.
FIN
```

Con base en los ejemplos anteriormente planteados se puede concluir que un algoritmo debe ser:

Realizable:



El proceso algorítmico debe terminar después de una cantidad finita de pasos. Se dice que un algoritmo es inaplicable cuando se ejecuta con un conjunto de datos iniciales y el proceso resulta infinito o durante la ejecución se encuentra con un obstáculo insuperable sin arrojar un resultado.

Comprensible:



Debe ser claro lo que hace, de forma que quien ejecute los pasos (ser humano o máquina) sepa qué, cómo y cuándo hacerlo. Debe existir un procedimiento que determine el proceso de ejecución.

Preciso:



El orden de ejecución de las instrucciones debe estar perfectamente indicado. Cuando se ejecuta varias veces, con los mismos datos iniciales, el resultado debe ser el mismo siempre. La precisión implica determinismo. (López, 2009, p. 22)

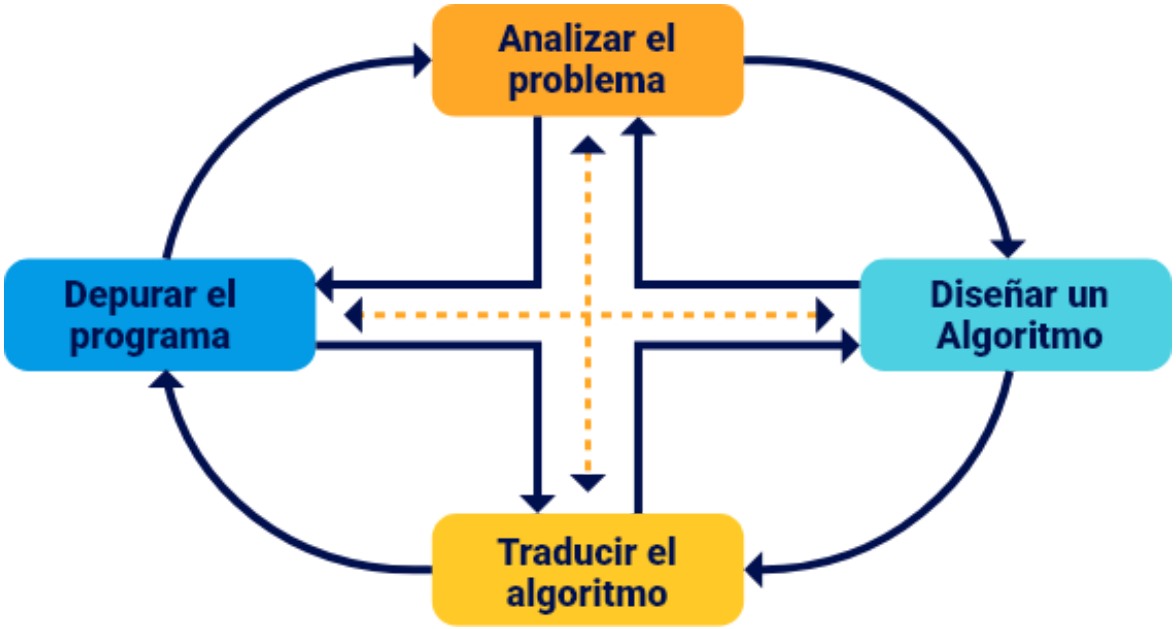


Figura 3. Fases para elaborar un programa de computador. Figura recuperada de Educación básica algoritmos y programación. Guía para docentes, de J. C. López García, 2007-2009, Fundación Gabriel Piedrahita Uribe www.eduteka.org p. 11.

La figura 3 muestra las fases y sus relaciones en el proceso de diseño de algoritmos de programas informáticos, que están basados en un algoritmo que posteriormente es escrito en un lenguaje de programación.

1.4. Análisis del problema (entenderlo)

Como los programas de computador tienen por objetivo resolver problemas muy puntuales, lo primero que se debe realizar para resolverlos es lograr una mejor comprensión posible de estos.

Para realizar esta actividad se debe:

1. **Formular claramente el problema.**
2. **Especificar los resultados que se desean obtener con la solución.**
3. **Identificar la información disponible en los datos.**

Formular el problema | Precisar elementos de entrada

Precisar resultados esperados



1.4.1. Formular el problema.

A continuación, algunas preguntas previas a la formulación del problema:

- ¿Es posible definir de forma más clara el problema?
- ¿Qué palabras no son conocidas o se desconoce su significado?

- ¿Se ha resuelto antes algún problema similar?
- ¿Qué información es importante para resolver el problema?
- ¿Qué información no es importante y se puede omitir?

Ejemplo

Don Juan necesita decidir cómo comprar un teléfono celular que cuesta $870.000 de contado o $980.000 a crédito; él tiene $630.000 pesos en efectivo.

Luego de revisar las preguntas previas se puede concluir:

Como el efectivo que tiene don Juan no le alcanza para comprar el teléfono celular de contado, entonces él tiene dos opciones: comprarlo completamente a crédito o también pagar una parte de contado (como cuota inicial) y el resto diferirlo a crédito.

Para poder resolver este problema se debe conocer el número de cuotas en el que difiere si desea pagarlo totalmente a crédito o conocer el número de cuotas y el valor total del celular si se da una cuota inicial de $630.000 pesos.

1.5. Desarrollo de la creatividad: elementos, modelos, fases y sus objetivos

Resolver algoritmos es una actividad que requiere en la mayoría de los casos creatividad por parte del programador, esta es una habilidad que se va fortaleciendo en la medida que se desempeña más y más en procesos de desarrollo de software.

Para estimular “el pensamiento creativo propuesto por Resnick (2007) el programador debe imaginar qué quiere hacer; crea un proyecto basado en sus ideas; juega y comparte sus ideas y creaciones con otros para reflexionar sobre sus experiencias y permitirse imaginar nuevas ideas y nuevos proyectos. La espiral genera un proceso indefinido de mejoramiento continuo” (López, 2009, p. 19).

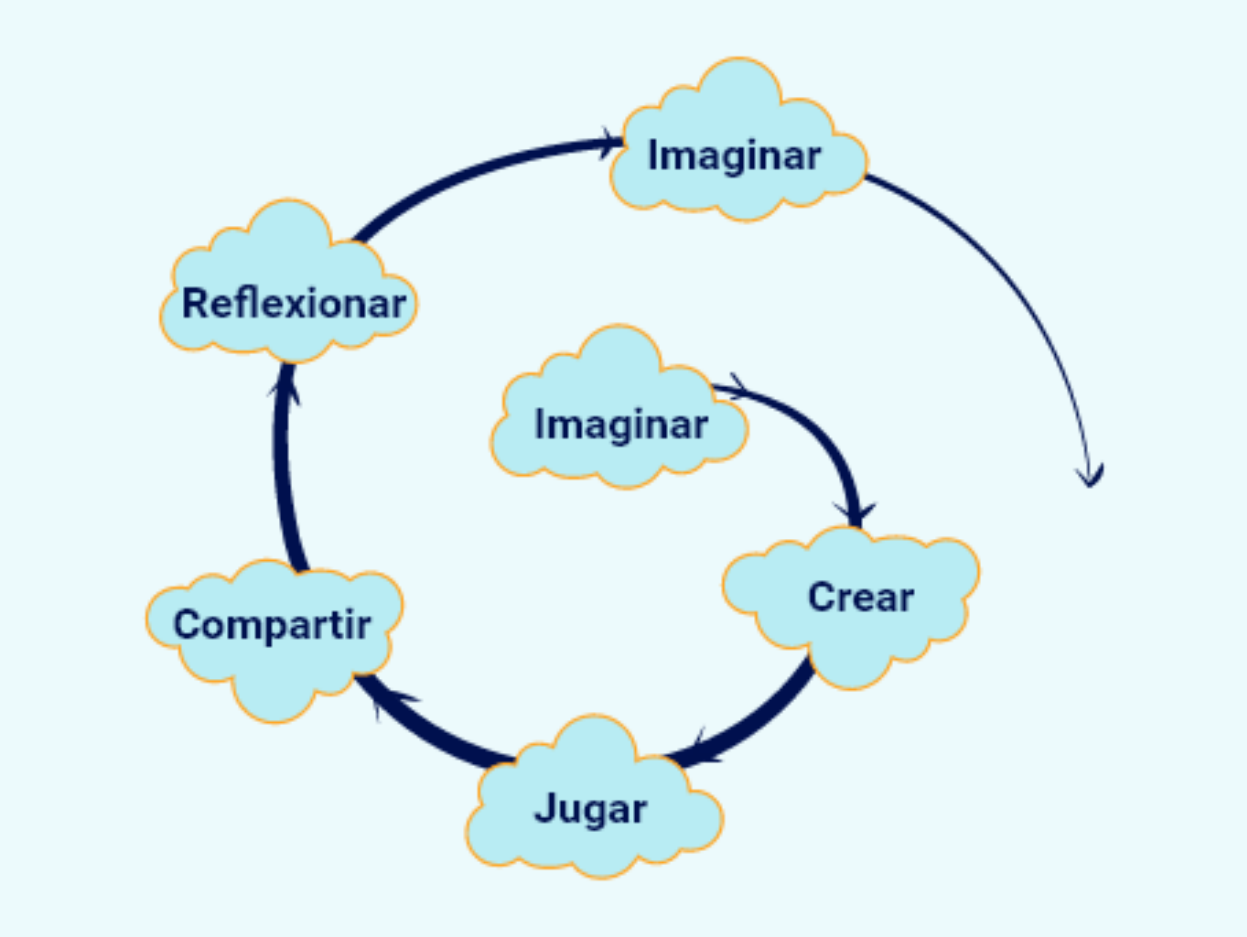


Figura 4. Espiral del Pensamiento Creativo diseñada por el Dr. Mitchel Resnick. Figura recuperada de Educación básica algoritmos y programación. Guía para docentes, de J. C. López García, 2007-2009, Fundación Gabriel Piedrahita Uribe, www.eduteka.org p. 19.

En conclusión, se puede decir que un algoritmo es la definición de una secuencia lógica de pasos que permiten solucionar un problema. Formular algoritmos es la base de los grandes adelantos en las ciencias matemáticas y la tecnología, porque se pueden aplicar a cualquier área del conocimiento.

2 Lógica matemática

La lógica matemática es la disciplina que trata de métodos de razonamiento. En un nivel elemental, la lógica proporciona reglas y técnicas para determinar si es o no válido un argumento dado. El razonamiento lógico se emplea en matemáticas para demostrar teoremas; en ciencias de la computación para verificar si son o no correctos los programas; en las ciencias física y naturales, para sacar conclusiones de experimentos; y en las ciencias sociales y en la vida cotidiana, para resolver una multitud de problemas. Ciertamente se usa en forma constante el razonamiento lógico para realizar cualquier actividad. (EcuRed, s.f.)



Para ampliar la definición puede visualizar el siguiente video (https://www.youtube.com/watch?v=9zYDaZhS7Ac)

Regresar

2.1. Componentes de la lógica matemática

Todo razonamiento lógico está compuesto por una serie de elementos que se interrelacionan entre sí, y la correcta interpretación de estos componentes y de sus relaciones corresponde al análisis deductivo. Estos componentes son las preposiciones y conectores lógicos que se describen a continuación:

a. Proposiciones

Una proposición o enunciado es una oración que puede ser falsa o verdadera, pero no ambas a la vez. La proposición es un elemento fundamental de la lógica matemática.

Las proposiciones pueden ser simples o compuestas. Las primeras están formadas por una sola proposición, por el contrario, las proposiciones compuestas están conformadas por dos o más proposiciones unidas por un conector.

A continuación, se muestra una serie de ejemplos de proposiciones válidas y no válidas, con estos se explicará por qué algunos enunciados no son proposiciones.

Para los enunciados anteriores se determina lo siguiente:

p y q son proposiciones simples válidas porque pueden tomar un valor verdadero o falso; sin embargo, los enunciados r y s son proposiciones no válidas, porque no pueden tomar un valor de falso o verdadero, uno de ellos es un saludo y el otro es una orden.

p: la tierra es plana.
q: -17 + 38 = 21
r: hola ¿cómo estás?

Regresar

S: lava el carro por favor.

b. Conectores lógicos y proposiciones compuestas

Existen las proposiciones compuestas que son las que se componen de varias proposiciones, las cuales utilizan para su conexión operadores o conectores lógicos que son:

− Operador AND (y) conjunción

Conecta dos proposiciones que se deben unir para que se pueda obtener un resultado verdadero, esta se representa por el siguiente símbolo ∧.

Ejemplo:

- Nuestra bandera es de color amarillo, azul y rojo.

+ Operador OR (o) disyunción

+ Operador NOT (no) negación

‹ Regresar

2.2. Tablas de verdad

Una estrategia de la lógica que tiene como objetivo determinar la validez de varias propuestas en alguna situación, es decir, determina las condiciones necesarias en las que un enunciado es verdadero, estas condiciones se representan con una tabla de verdad que determina el valor de verdad de todas las proposiciones que componen la situación. Así como lo explica el principio de bivalencia: **una proposición puede ser verdadera o falsa únicamente, nunca ambas**.

una proposición puede ser verdadera o falsa únicamente, nunca ambas.

Algunas consideraciones:

a. El valor de una proposición verdadera se representa por “V” o un “1”.

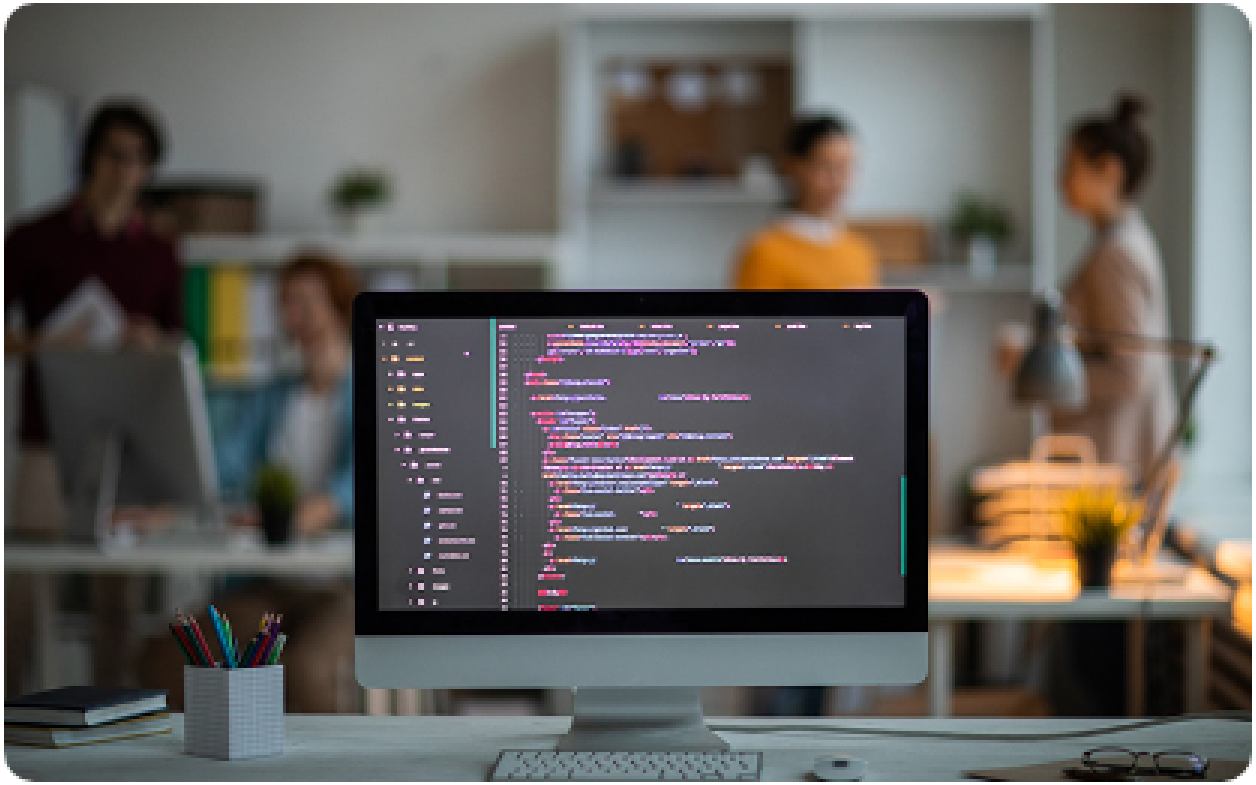
---

b. El valor de una proposición falsa se representa por “F” o un “0”.

---

c. Existe una tabla de verdad para cada uno de los conectores lógicos, lo que se tiene que evaluar para responder si una proposición compuesta es verdadera o falsa es el estatus actual de las proposiciones simples. A continuación, se muestran las tablas de verdad.

‹ Regresar



P	Q	P^Q
V	V	V
V	F	F
F	V	F
F	F	F

O puede ser

P	Q	P^Q
1	1	1
1	0	0
0	1	0
0	0	0

Tabla 1: Tabla de verdad conjunción (Y)

Regresar

P	Q	PvQ
V	V	V
V	F	V
F	V	V
F	F	F

O puede ser

P	Q	PvQ
1	1	1
1	0	1
0	1	1
0	0	0

Tabla 2: Tabla de verdad disyunción (O)

P	¬P
V	F
F	V

O puede ser

P	¬P
1	0
0	1

Tabla 3: Tabla de verdad negación (no)

Existe otro tipo de conectores que se llaman proposiciones condicionales, aquellas que están formadas por dos proposiciones simples o compuesta por p y q, la cual se representa y se lee de la siguiente manera:

p → q se lee “Si p entonces q”

Ejemplo 1:	Ejemplo 2:	Ejemplo 3:

Regresar

Si no llueve se arruina la cosecha.

Este tipo de proposición se conoce como la condición “para que”, de manera que para que no se arruine la cosecha se requiere que llueva.

p: si no llueve

q: se arruina la cosecha

Al aplicar el operador p → q su tabla de verdad quedaría de la siguiente manera:

P	Q	P→Q
V	V	V
V	F	F
F	V	V
F	F	V

O puede ser

P	Q	P→Q
1	1	1
1	0	0
0	1	1
0	0	1

Tabla 4: Tabla de verdad condicional (si... entonces)

De igual manera existe una proposición bicondicional que indica que una proposición es verdadera si y solo si las proposiciones simples o compuestas que forman la proposición son verdaderas o son falsas si y solo si ambas proposiciones son falsas, esta se representa y se lee de la siguiente manera:

p ↔ q se lee "p si solo si q"

‹ Regresar

2.3. Ejercicios de lógica proposicional

A continuación, se plantean a manera de ejemplo algunos ejercicios con su respectiva solución que permiten identificar un problema, reconocer los datos de entrada y los procesos a ser aplicados como estrategia a seguir con la utilización de la lógica proposicional en la identificación y solución de problemas.

2.3.1. Ejercicio 1.

Se necesita personal femenino que sepa Microsoft Office o Gsuite.

Para dar solución al enunciado anterior se realiza las siguientes actividades.

Pasos a seguir:

- Identificar el problema.
- Descomponer el enunciado en proposiciones atómicas.
- Evaluar los posibles casos derivados del enunciado anterior.
- Determinar los casos donde se cumpla la condición y hacer la selección de las personas de acuerdo con el requerimiento.

Regresar

Solución:

1

Identificar el problema

Se trata de buscar o contratar personal femenino que sepa utilizar Microsoft Office o Google Suite.

2

Descomponer el enunciado

Descomponer en proposiciones atómicas del enunciado (cuando están separadas del contexto del problema se llaman proposiciones atómicas).

p: es personal femenino.
q: sabe utilizar el Microsoft Office.
r: sabe utilizar la suite de Google.

3

Determinar los casos

Determinar las relaciones lógicas de acuerdo con lo planteado en el enunciado, utilizando las proposiciones atómicas definidas en el punto anterior.

$p \wedge (q \vee r)$.

4

Evaluar los posibles casos derivados del enunciado anterior.

Regresar

2.3.2. Ejercicio 2.

Juan sabe utilizar Microsoft Office y Gsuite, Martha sabe utilizar Gsuite, Natalia sabe utilizar Gsuite y Microsoft Office.

Crear la tabla de verdad y aplicar la lógica proposicional para elegir las personas que cumplan con los requisitos exigidos.

Para la construcción de la tabla de verdad se evalúa cada una de las preposiciones atómicas y, se van validando para verificar si se cumple lo que la haría verdadera o si no se cumple, lo cual haría que la proposición sea falsa.

Se analiza cada posible caso por lo que se evalúa a Juan, para la primera proposición sería un valor falso, ya que p requiere que sea personal femenino, el cual se pondrá en la columna 2, luego se evalúa si Juan sabe utilizar Microsoft Office o la Suite de Google, el valor va en la columna tres y por último, se evalúa la conjunción para determinar la solución.

Regresar

Nombre	p	(q V r)	p ^ (q V r)
Juan	F	V	F
Martha	V	V	V
Nathalia	V	V	V

De acuerdo con la anterior tabla, se concluye que Martha y Natalia son aptas para el trabajo, lo que permite seleccionarlas.

2.3.3. Ejemplo 3.

Sandra dice adelgazo o no hago dieta.

Para este enunciado además se propone una serie de alternativas que podrían ser válidas o no, dependiendo de la solución propuesta utilizando la lógica matemática. Las siguientes son las posibles respuestas de las cuales se deberá elegir la que se considera es real al problema planteado.

1. **Si adelgazo, hago dieta.**
2. **Si no adelgazo, hago dieta.**
3. **Si hago dieta, no adelgazo.**
4. **Si hago dieta adelgazo.**
5. **Hago dieta o no adelgazo.**

Regresar

Solución:

1

Problema

Identificar la respuesta correcta al problema presentado por Sandra.

2

Definir

Definir las proposiciones atómicas:

p: adelgazo.
q: hago dieta.

3

Determinar

Determinar la relación lógica.

p V ¬ q.

4

Crear

Crear la tabla de verdad.

p	(q)	(p V ¬ q)
V	V	V
V	F	V
F	V	F
F	F	V

Regresar

Regresar

Evidencia de conocimiento: GA3-220501093-AA2-EV01

Fundamentos de programación estructurada

y estructuras cíclicas

PRESENTADO POR

Miguel Ignacio Díaz Campaña

INSTRUCTOR

CARLOS ERNESTO MAYA VALLEJO

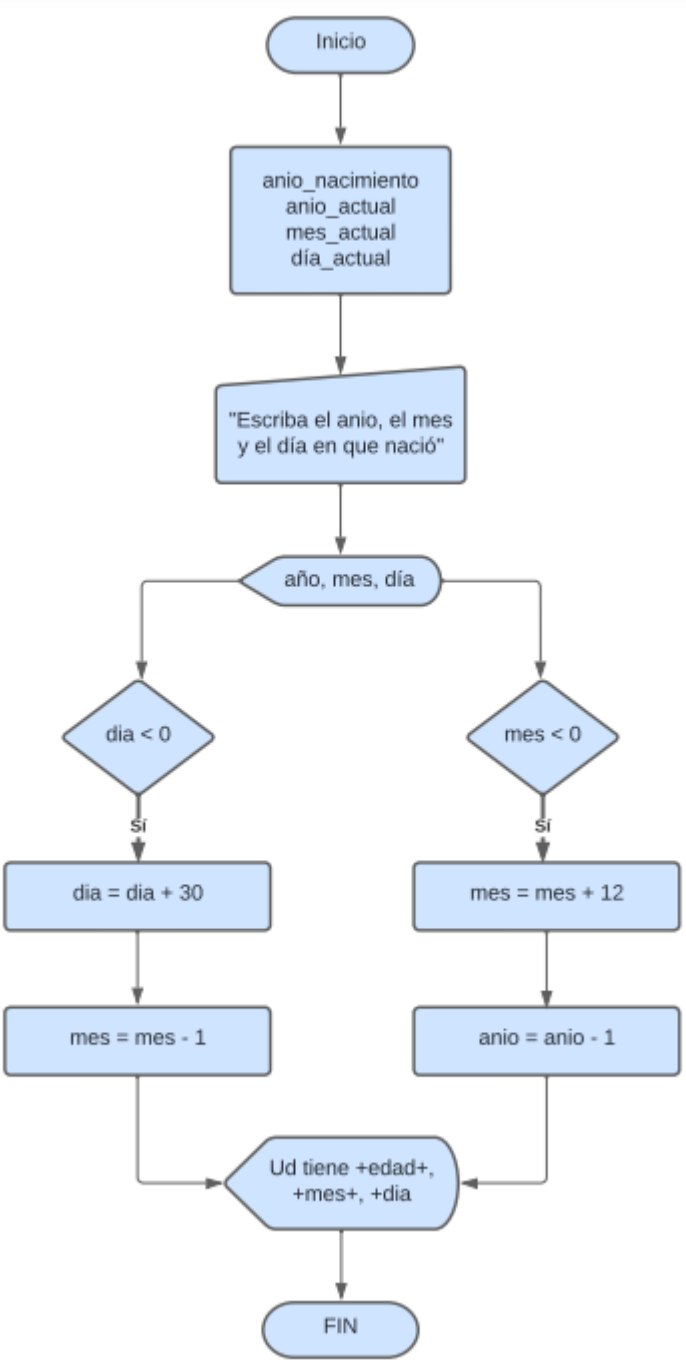
SERVICIO NACIONAL DE APRENDIZAJE – SENA

ANÁLISIS Y DESARROLLO DE SOFTWARE

Julio – 2021

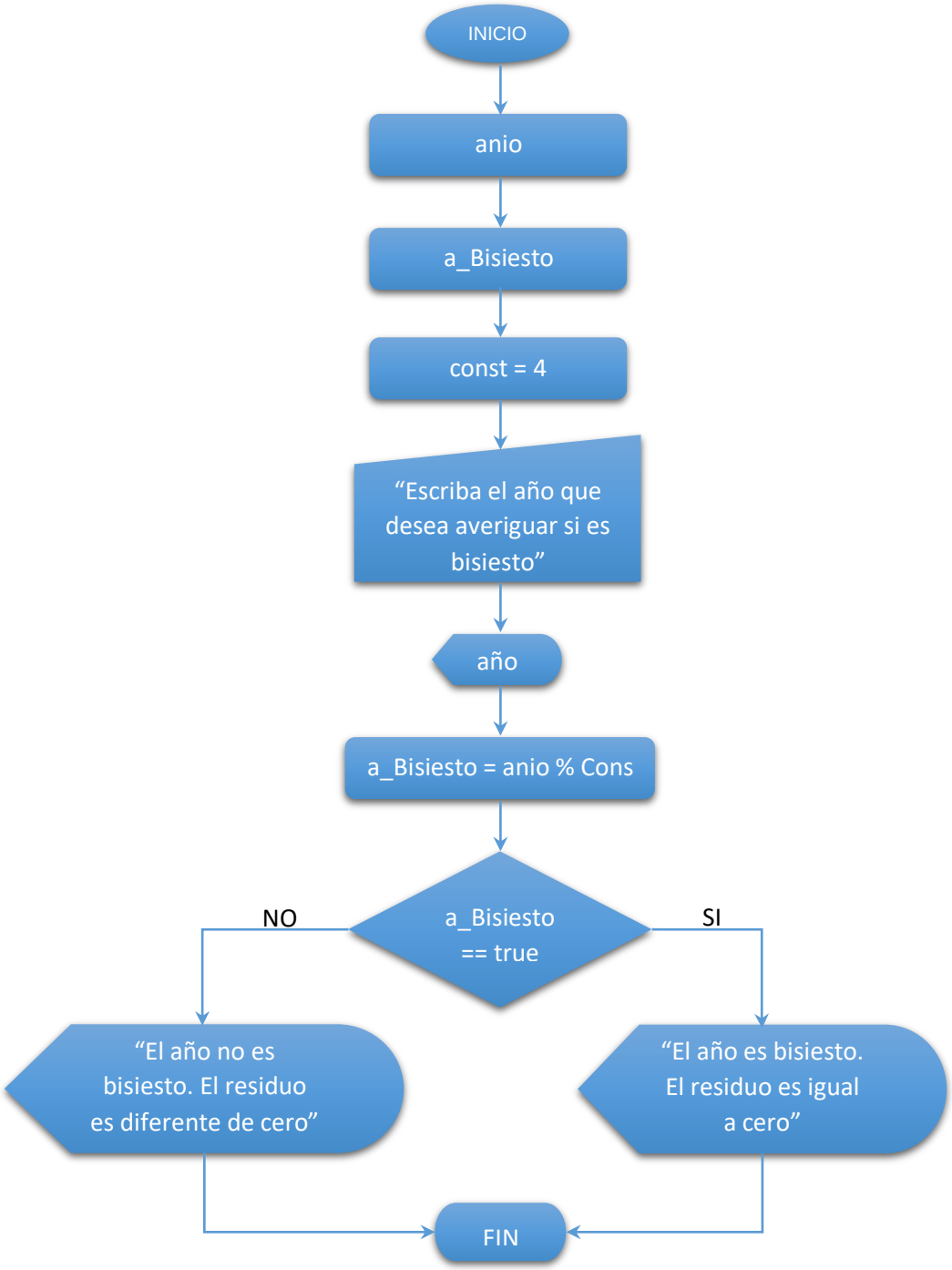
CALCULAR LA EDAD DE UNA PERSONA

```
ALGORITMO calcular la edad de una persona
VAR
  Entero anio_nacimiento
  Entero anio_actual
  Entero mes_nacimiento
  Entero mes_actual
  Entero dia_nacimiento
  Entero dia_actual
INICIO
  ESCRIBIR ("Ingrese el año, el mes y el día en que nació")
  LEER (año, mes, día);
    edad = anio_actual – anio_nacimiento
    mes = mes_nacimiento – mes_actual
    dia = dia_actual – dia_nacimiento
      SI dia < 0
        dia = dia + 30
        mes = mes – 1
      FINSI
      SI mes < 0
        mes = mes + 12
        anio = anio – 1
      FINSI
        ESCRIBIR (“Ud tiene + edad +, + mes +, + dia +”)
FIN
```


Inicio
anio_nacimiento
anio_actual
mes_actual
día_actual
"Escriba el anio, el mes y el día en que nació"
año, mes, día
dia < 0
mes < 0
sí
sí
dia = dia + 30
mes = mes + 12
mes = mes - 1
anio = anio - 1
Ud tiene +edad+, +mes+, +dia
FIN

DETERMINAR SI UN AÑO INDICADO ES O NO UN AÑO BISIESTO

```
ALGORITMO determinar si un año indicado es o no un año bisiesto
  VAR
    Entero anio;
    const = 4;
    Entero a_Bisiesto
INICIO
  ESCRIBIR ("Escriba el año que desea averiguar si es bisiesto");
  LEER (año);
    a_Bisiesto = anio % cons;
          SI a_Bisiesto == 0;
            LEER ("El año es bisiesto porque el residuo es igual a cero");
          NO a_Bisiesto != 0;
            ESCRIBIR ("El año no es bisiesto porque el residuo es diferente de cero");
          FIN SINO
FIN
```


INICIO
anio
a_Bisiesto
const = 4
“Escriba el año que desea averiguar si es bisiesto”
año
a_Bisiesto = anio % Cons
a_Bisiesto == true
NO
SI
“El año no es bisiesto. El residuo es diferente de cero”
“El año es bisiesto. El residuo es igual a cero”
FIN

SERVICIO NACIONAL
DE APRENDIZAJE

FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE:
FUNDAMENTOS DE PROGRAMACIÓN ESTRUCTURADA Y ESTRUCTURAS CICLICAS

PRESENTA:
MILENA FAIZURE TORRES HERNANDEZ

ANÁLISIS Y DESARROLLO DE SOFTWARE

FICHA:
2455255

COLOMBIA, BOGOTA D.C.,
11 DE AGOSTO DE 2022

Esta evidencia se centra en consolidar los conceptos relacionados con las metodologías de algoritmos y sus tres fases: análisis, diseño e implementación. Respecto a la implementación se aborda los pseudocódigos y diagramas de flujo.

Para su desarrollo es importante la lectura del componente formativo. Debe construir un documento donde registre el análisis, diseño e implementación de algoritmos que permitan resolver los siguientes problemas:

- Algoritmo que a partir de una fecha de nacimiento y una fecha actual determine la edad en años actual de una persona.

INICIO

VAR anio_nacimiento, mes_nacimiento, dia_nacimiento, anio_actual, mes_actual, dia_actual

ESCRIBA (“Por favor ingrese el año de nacimiento”)
LEER (anio_nacimiento)
ESCRIBA (“Por favor ingrese el mes de nacimiento”)
LEER (mes_nacimiento)
ESCRIBA (“Por favor ingrese día de nacimiento”)
LEER (dia_nacimiento)

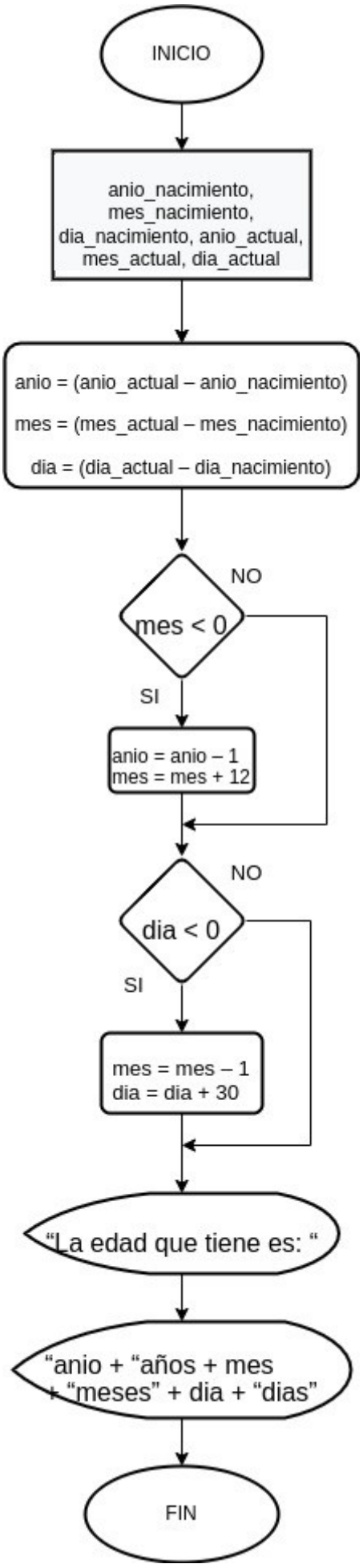
anio = (anio_actual – anio_nacimiento)
mes = (mes_actual – mes_nacimiento)
dia = (dia_actual – dia_nacimiento)

SI mes < 0 **ENTONCES**
anio = anio – 1
mes = mes + 12
FinSi

SI dia < 0 **ENTONCES**
mes = mes – 1
dia = dia + 30
FinSi

ESCRIBIR (“La edad que tiene es: “)
ESCRIBIR (“anio + “años + mes + “meses” + dia + “dias”)

FIN


INICIO
anio_nacimiento,
mes_nacimiento,
dia_nacimiento, anio_actual,
mes_actual, dia_actual
anio = (anio_actual – anio_nacimiento)
mes = (mes_actual – mes_nacimiento)
dia = (dia_actual – dia_nacimiento)
NO
mes < 0
SI
anio = anio – 1
mes = mes + 12
NO
dia < 0
SI
mes = mes – 1
dia = dia + 30
“La edad que tiene es: “
“anio + “años + mes
+ “meses” + dia + “dias”
FIN

- Algoritmo que permita determinar si un año indicado es o no bisiesto.

INICIO

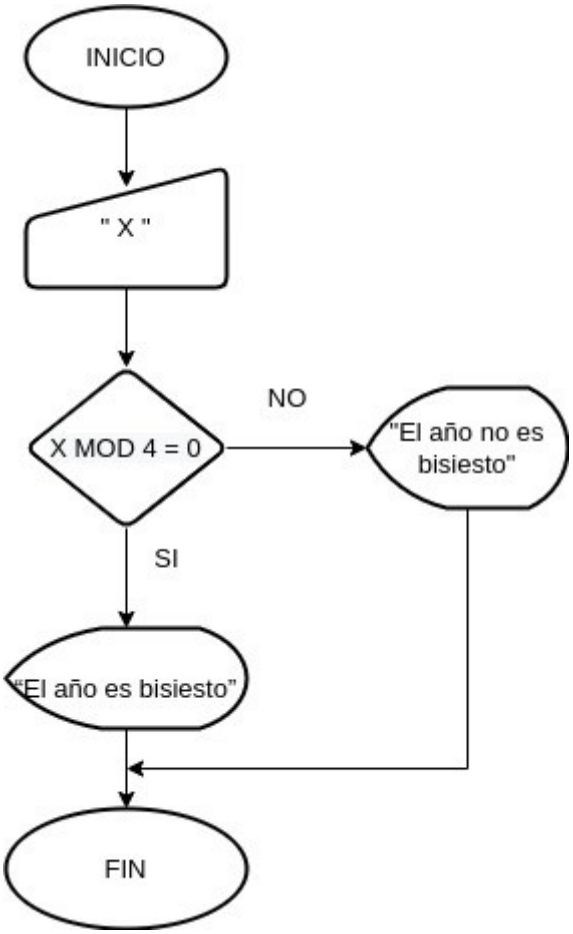
VAR

ESCRIBA (“Por favor ingrese el año”)
LEER (X**);**

SI (X MOD 4 = 0)
ESCRIBIR(“El año es bisiesto”);
SI NO ESCRIBIR(“El año no es bisiesto”);

FINSI

FIN





EVIDENCIAS DE CONOCIMIENTO:
GA3-220501093-AA2-EV01 fundamentos de programación estructurada y estructuras cíclicas

PRESENTA:
ORLANDO URAN BETANCOURT

ANÁLISIS Y DESARROLLO DE SOFTWARE

FICHA:
2455255

COLOMBIA, BOGOTA D.C.,
23 DE JULIO DEL 2022

EV01 fundamentos de programación estructurada y estructuras cíclicas

Esta evidencia se centra en consolidar los conceptos relacionados con las metodologías de algoritmos y sus tres fases: análisis, diseño e implementación. Respecto a la implementación se aborda los seudocódigos y diagramas de flujo.

Para su desarrollo es importante la lectura del componente formativo. Debe construir un documento donde registre el análisis, diseño e implementación de algoritmos que permitan resolver los siguientes problemas:

-Algoritmo que a partir de una fecha de nacimiento y una fecha actual determine la edad en años actual de una persona.

-Algoritmo que permita determinar si un año indicado es o no un año bisiesto.

SOLUCIÓN

1. Algoritmo que a partir de una fecha de nacimiento y una fecha actual determine la edad en años actual de una persona.

Análisis

Inicialmente se tendrían 6 valores enteros, de los cuales 3 de ellos almacenarán el día, mes y año de nacimiento de la persona y los 3 restantes a la fecha actual. Los valores de la fecha actual serán constantes y se pedirá al usuario que ingrese por teclado los valores de su fecha de nacimiento. Para calcular la edad en años, deberá ejecutarse la siguiente ecuación:

Result = Aactual – Auser + (Mactual – Muser)/12 + (Dactual – Duser)/365;

El resultado se almacenará en Result que es una variable de tipo real. Finalmente, el algoritmo mostrará en pantalla la edad del usuario imprimiendo el contenido de Result.

Diseño

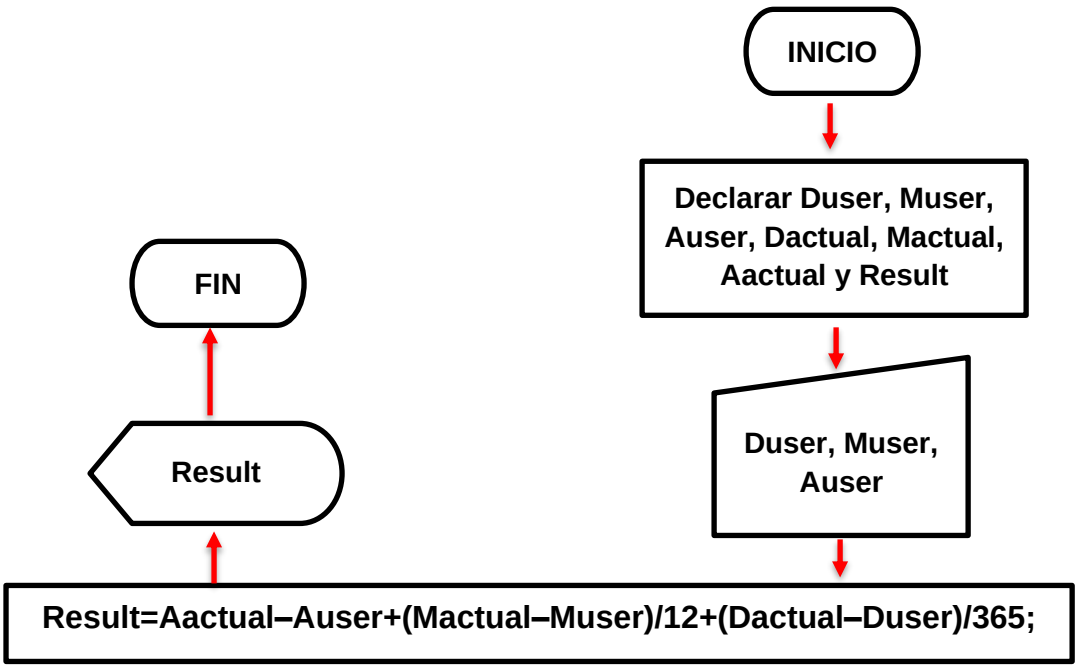


Diagrama de flujo

ALGORITMO Calcular edad en años de una persona a partir de su fecha de nacimiento

VAR

ENTERO Duser;
ENTERO Muser;
ENTERO Auser;
ENTERO Dactual = 23;
ENTERO Mactual = 7;
ENTERO Aactual = 2022;
REAL Result;

INICIO

ESCRIBIR (“Ingrese el número del día del mes de su nacimiento: “);
LEER Duser;
ESCRIBIR (“Ingrese el número de mes de su fecha de nacimiento: “);
LEER Muser;
ESCRIBIR (“Ingrese el número de su año de nacimiento: “);
LEER Auser;

Result = Aactual – Auser + (Mactual – Muser)/12 + (Dactual – Duser)/365;

ESCRIBIR (“La cantidad de años del usuario es: “);
ESCRIBIR Result;

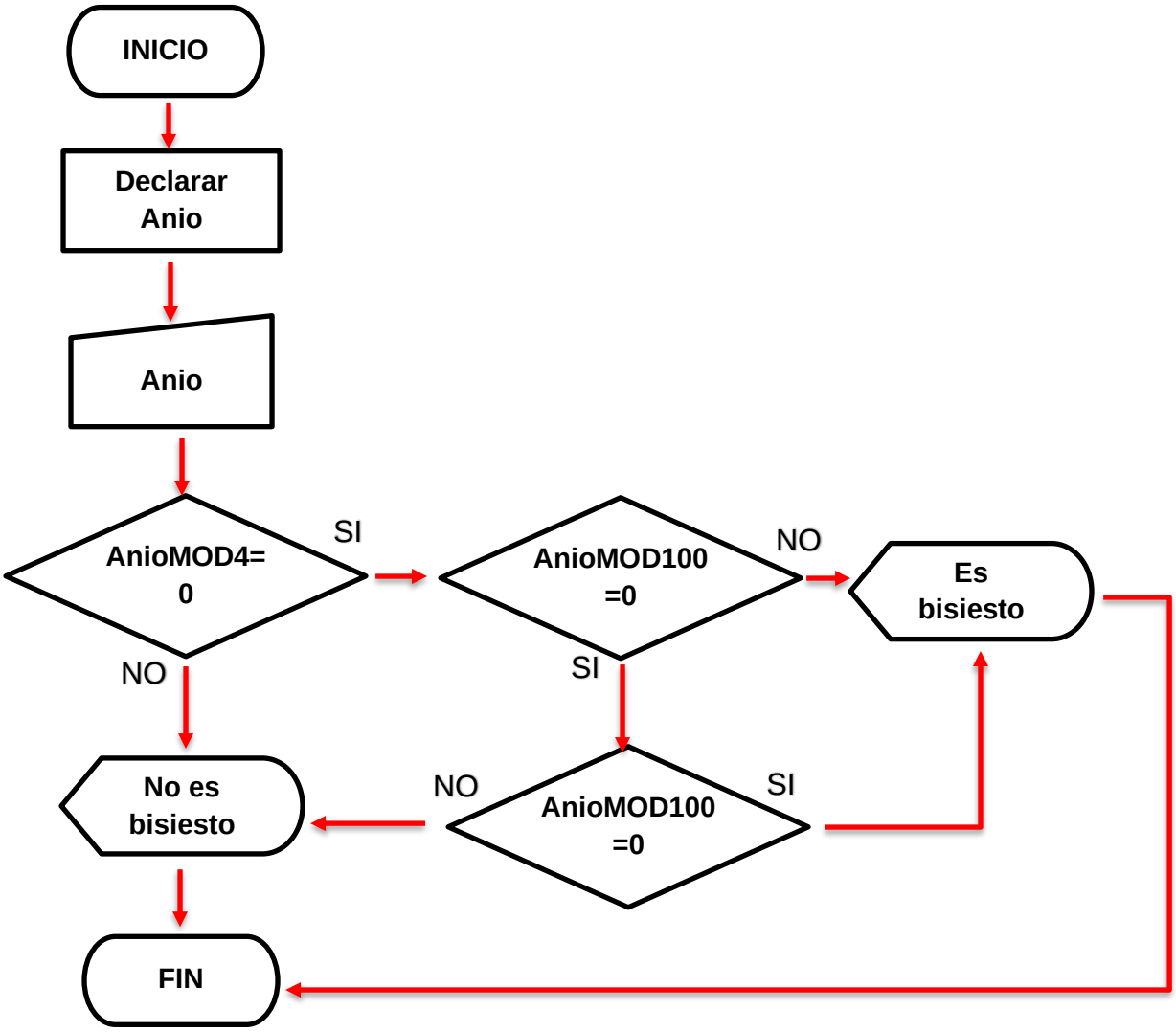
FIN

2. Algoritmo que permita determinar si un año indicado es o no un año bisiesto.

Análisis

Para saber si un año es bisiesto solo basta con realizar una división por 4, y si es exacta entonces se dice que es bisiesto, de lo contrario no es bisiesto. Pero, es necesario realizar unas correcciones debido a un error de cálculo (fuente: https://docs.microsoft.com/es-es/office/troubleshoot/excel/determine-a-leap-year).
Así, si un año es divisible exactamente por 4 debe validarse que también lo sea por 100 y si no lo es entonces es bisiesto, pero si lo es debe validarse la división exacta por 400, si el residuo es 0 entonces el año será bisiesto y si el residuo no es 0 entonces no será bisiesto.

Diseño





ALGORITMO Informar si un año es bisiesto o no.

VAR
    ENTERO Anio;

INICIO

ESCRIBA (“Ingrese un número de año por favor: “);
LEER Anio;

    SI (Anio**MOD**4 = 0)
        SI (Anio**MOD**100 = 0)
            SI (Anio**MOD**400 = 0)
                ESCRIBIR (“El año ingresado es Bisiesto”);
            SI NO
                ESCRIBIR (“El año ingresado no es Bisiesto”);
            FIN SI
        SI NO
            ESCRIBIR (“El año ingresado es Bisiesto”);
        FIN SI
    SI NO
        ESCRIBIR (“El año ingresado no es Bisiesto”);
    FIN SI

FIN

FUNDAMENTOS DE PROGRAMACIÓN ESTRUCTURADA Y ESTRUCTURAS CÍCLICAS GA3-220501093-AA2-EV01.

ANALISIS Y DESARROLLO DE SOFTWARE

REGIONAL CAUCA (2455255)

SENA

2022

FUNDAMENTOS DE PROGRAMACIÓN ESTRUCTURADA Y ESTRUCTURAS CÍCLICAS GA3-220501093-AA2-EV01.

Jency Yosmary Sanchez Garzón

ANALISIS Y DESARROLLO DE SOFTWARE

REGIONAL CAUCA (2455255)

SENA

2022

CONTENIDO

1. Algoritmo que a partir de una fecha de nacimiento y una fecha actual determine la edad en años actual de una persona. 4

Análisis: 4

Diseño: 5

Implementación del Algoritmo: 7

2. Algoritmo que permita determinar si un año indicado es o no un año bisiesto. 7

Análisis: 7

Diseño: 8

Implementación del Algoritmo: 9

CITAS BIBLIOGRAFICAS 10

1. Algoritmo que a partir de una fecha de nacimiento y una fecha actual determine la edad en años actual de una persona.

Para el presente problema se plantea el siguiente Análisis, diseño e implementación para el algoritmo solicitado:

Análisis:

- **Formular el problema:** Algoritmo que permite obtener la edad en años actual de una persona a partir de la fecha de nacimiento y la fecha actual.
- **Resultados esperados:** Se espera que el algoritmo sea capaz de calcular de acuerdo a los datos tomados la edad en años actual de la persona y mostrarlo en pantalla.

- **Datos disponibles:** Fecha de nacimiento y la fecha actual.
- **Restricciones:** Ninguna
- **Procesos necesarios:** Solicitar siempre al usuario la fecha de nacimiento indicando el día, mes y año; de igual forma el usuario debe ingresar la fecha actual indicando día, mes y año.

Diseño:

Definir_Edad_De_Una_Persona.psc

```
Algoritmo Definir_Edad_De_Una_Persona
    Definir diaNacimiento,mesNacimiento,anioNacimiento Como Entero
    Definir diaActual,mesActual,anioActual Como Entero
    Definir edad Como Entero

    Escribir "Por favor, Ingrese la fecha de Nacimiento"
    Escribir "Día de Nacimiento: "
    Leer diaNacimiento
    Escribir "Mes de Nacimiento: "
    Leer mesNacimiento
    Escribir "Año de Nacimiento: "
    Leer anioNacimiento

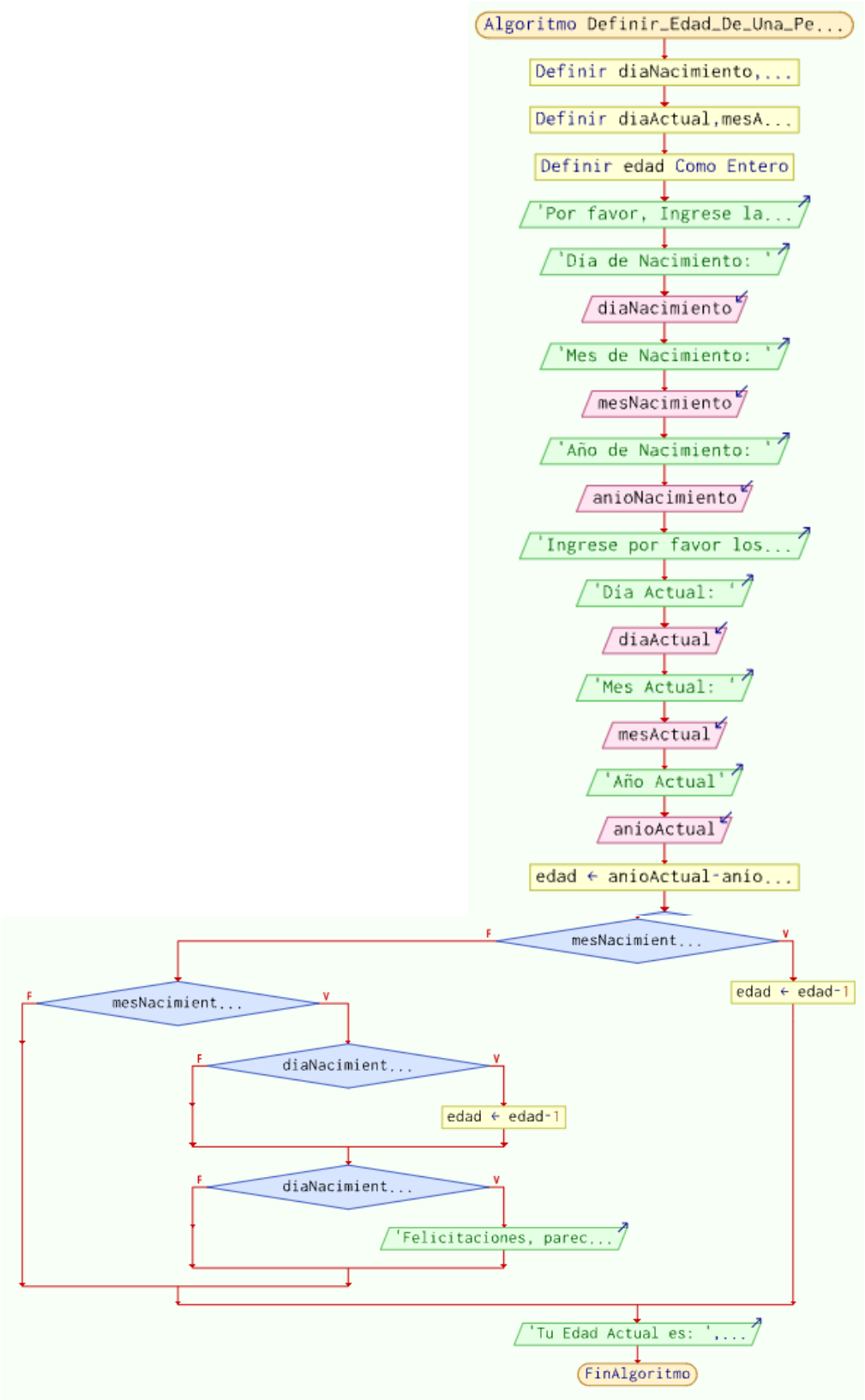
    Escribir "Ingrese por favor los datos de la Fecha Actual"
    Escribir "Día Actual: "
    Leer diaActual
    Escribir "Mes Actual: "
    Leer mesactual
    Escribir "Año Actual"
    Leer anioActual

    edad←anioActual - anioNacimiento


    Si mesNacimiento > mesActual Entonces
        edad=edad-1
    SiNo
        Si mesNacimiento = mesActual
            Si diaNacimiento > diaActual
                edad=edad-1
            FinSi
            Si diaNacimiento=diaActual
                Escribir "Felicitaciones, parece que el día de hoy es tu cumpleaños!!!"
            FinSi
        FinSi
    Fin Si

    Escribir "Tu Edad Actual es: " edad, " Años."
FinAlgoritmo

```


Algoritmo Definir_Edad_De_Una_Pe...
Definir diaNacimiento,...
Definir diaActual,mesA...
Definir edad Como Entero
'Por favor, Ingrese la...
'Dia de Nacimiento: '
diaNacimiento
'Mes de Nacimiento: '
mesNacimiento
'Año de Nacimiento: '
anioNacimiento
'Ingrese por favor los...
'Dia Actual: '
diaActual
'Mes Actual: '
mesActual
'Año Actual'
anioActual
edad ← anioActual-anio...
mesNacimient...
F
V
edad ← edad-1
mesNacimient...
diaNacimient...
edad ← edad-1
diaNacimient...
'Felicitaciones, parec...
'Tu Edad Actual es: ',...
FinAlgoritmo

Implementación del Algoritmo:

```
PSeInt - Ejecutando proceso DEFINIR_EDAD_DE_UNA_PERSONA
*** Ejecución Iniciada. ***
Por favor, Ingrese la fecha de Nacimiento
Día de Nacimiento:
> 26
Mes de Nacimiento:
> 8
Año de Nacimiento:
> 1993
Ingrese por favor los datos de la Fecha Actual
Día Actual:
> 7
Mes Actual:
> 8
Año Actual
> 2022
Tu Edad Actual es: 28 Años.
*** Ejecución Finalizada. ***
```

2. Algoritmo que permita determinar si un año indicado es o no un año bisiesto.

Para el presente problema se plantea el siguiente Análisis, diseño e implementación para el algoritmo solicitado:

Análisis:

- **Formular el problema:** Se debe determinar si el valor (año) ingresado corresponde a un año bisiesto o si no corresponde.
- **Resultados esperados:** El usuario podrá ver en pantalla el resultado de si el año ingresado es o no Bisiesto.
- **Datos disponibles:** Numero del año ingresado por el usuario.
- **Restricciones:** Ninguna

- **Procesos necesarios:** Solicitar al usuario el ingreso de un año; validar si cumple con las condiciones para ser bisiesto; de ser el caso el resultado será “El año ingresado es Bisiesto”, de lo contrario arrojara mensaje de “El año ingresado no es Bisiesto”.

- **Condiciones para determinar si el año es Bisiesto:**

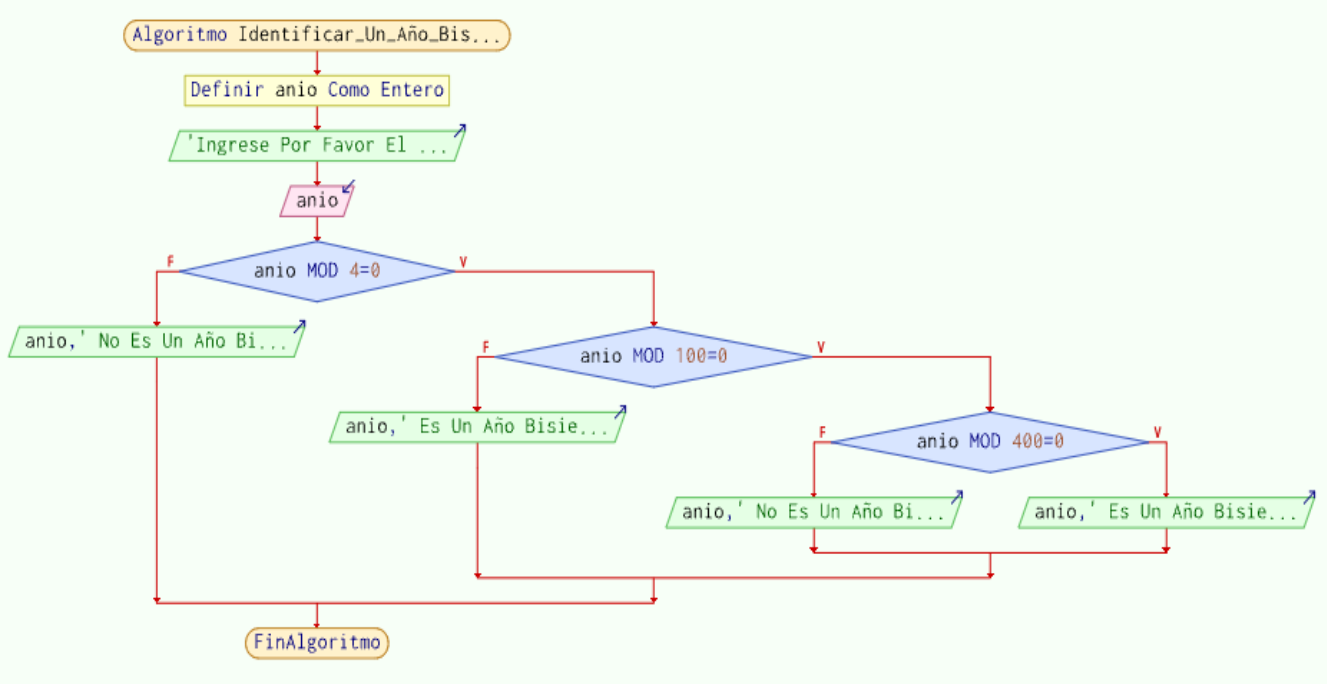
  El año debe ser **divisible entre 4**, Ejemplo 2004 – 2008.
  El año no debe ser **divisible entre 100**, Ejemplo 2200 – 2500.
  El año debe ser **divisible entre 400**, Ejemplo 2400 – 2800

  Nota: Para que el año sea Bisiesto debe cumplir las tres condiciones o la primera y la tercera condición, de lo contrario no cumpliría con las condiciones para ser bisiesto.

Diseño:

<sin_titulo> Identificar_Un_Año_Bisiesto.psc

```
1  Algoritmo Identificar_Un_Año_Bisiesto
2      Definir anio Como Entero
3      Escribir 'Ingrese Por Favor El Año A Evaluar:'
4      Leer anio
5      Si anio MOD 4=0 Entonces
6          Si anio MOD 100=0 Entonces
7              Si anio MOD 400=0 Entonces
8                  Escribir anio,' Es Un Año Bisiesto.'
9              SiNo
10                 Escribir anio,' No Es Un Año Bisiesto.'
11             FinSi
12         SiNo
13             Escribir anio,' Es Un Año Bisiesto.'
14         FinSi
15     SiNo
16         Escribir anio,' No Es Un Año Bisiesto.'
17     FinSi
18 FinAlgoritmo
19
```



Implementación del Algoritmo:

```
PSeInt - Ejecutando proceso IDENTIFICAR_UN_AÑO_BISIESTO
*** Ejecución Iniciada. ***
Ingrese Por Favor El Año A Evaluar:
> 2002
2002 No Es Un Año Bisiesto.
*** Ejecución Finalizada. ***
```

```
PSeInt - Ejecutando proceso IDENTIFICAR_UN_AÑO_BISIESTO
*** Ejecución Iniciada. ***
Ingrese Por Favor El Año A Evaluar:
> 2008
2008 Es Un Año Bisiesto.
*** Ejecución Finalizada. ***
```

CITAS BIBLIOGRAFICAS

https://sena.territorio.la/content/index.php/institucion/Titulada/institution/SENA/Tecnologia/228118/Contenido/OVA/CF13/index.html#/curso/tema1#t_1_2

https://www.youtube.com/watch?v=axrdBobi44I

https://www.youtube.com/watch?v=PS_lhfxK0Ag

https://www.youtube.com/watch?v=eaFUBPrNsyg

https://www.youtube.com/watch?v=QShH3ctjhUY

https://www.youtube.com/watch?v=sqTHWzwgjoI

https://www.youtube.com/watch?v=T5GaisQjPiE

FUNDAMENTOS DE PROGRAMACION ESTRUCTURADA Y ESTRUCTURAS CICLICAS

GA3-220501093-AA2-EV01

APRENDIZ:

JHOSSED PINEDA MOLINA

INSTRUCTOR:

CARLOS ERNESTO MAYA VALLEJO

TECNOLÓGICA ANÁLISIS Y DESARROLLO DE SOFTWARE. (2455255)

CENTRO DE COMERCIO Y SERVICIOS CAUCA – POPAYÁN

SERVICIO NACIONAL DE APRENDIZAJE-SENA

07 DE AGOSTO DE 2022

- **Algoritmo que a partir de una fecha de nacimiento y una fecha actual determine la edad en años actual de una persona**

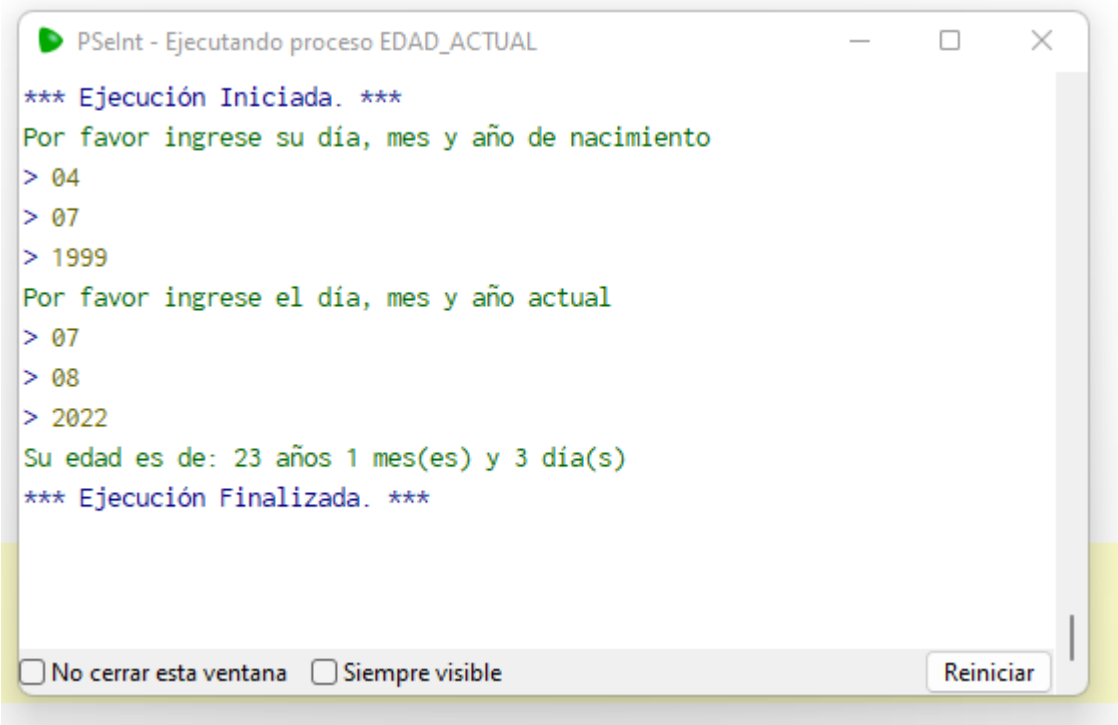
```
Algoritmo Edad_Actual
    Definir birthDay, currentDay, monthBirth, currentMonth, yearBirth, currentYear, age, day, month Como Entero
    Escribir "Por favor ingrese su día, mes y año de nacimiento"
    leer birthDay, monthBirth, yearBirth

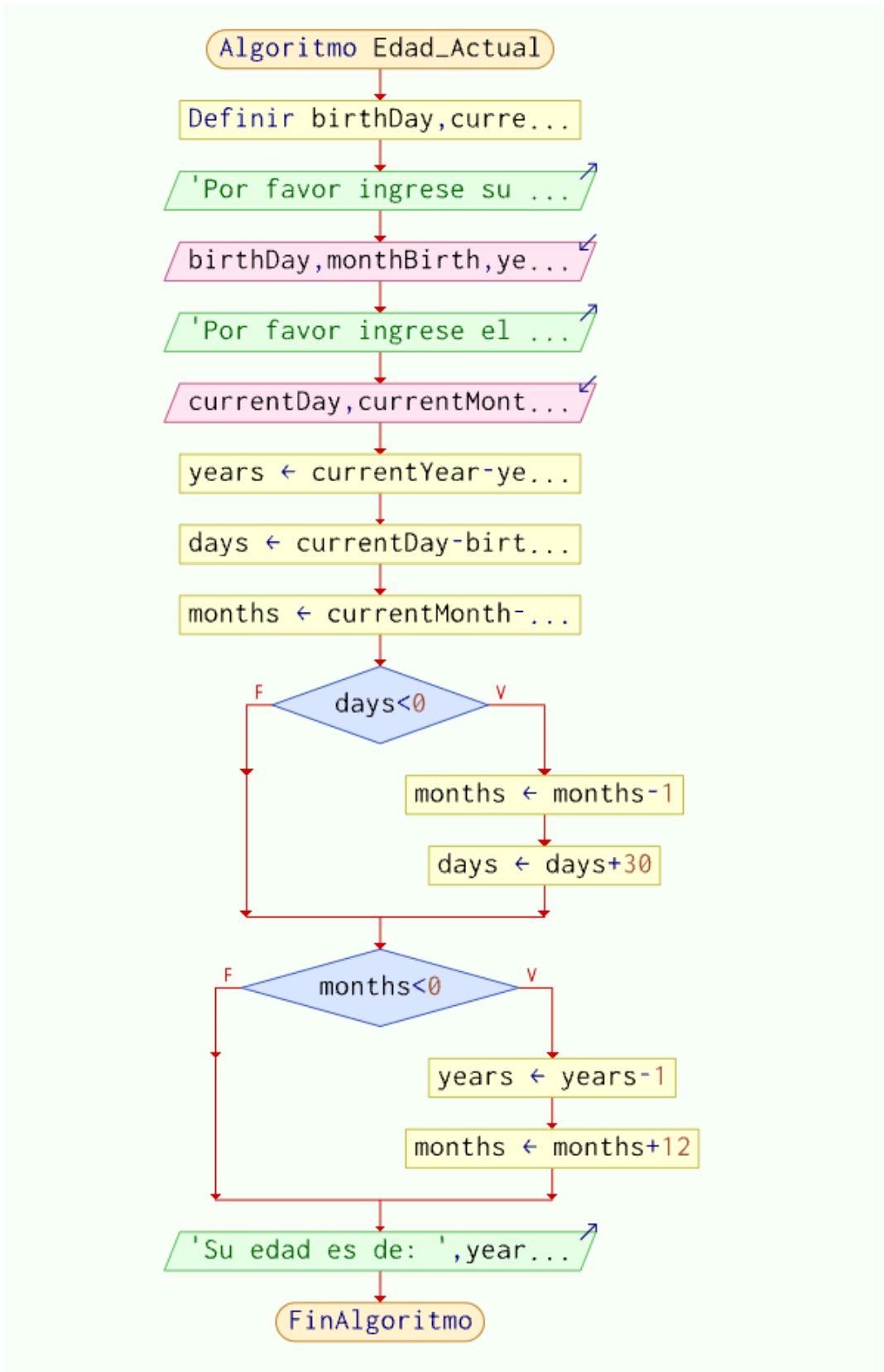
    Escribir "Por favor ingrese el día, mes y año actual"
    leer currentDay, currentMonth, currentYear

    years ← currentYear - yearBirth
    days ← currentDay - birthDay
    months ← currentMonth - monthBirth

    Si days < 0 Entonces
        months ← months - 1
        days ← days + 30
    FinSi
    Si months < 0 Entonces
        years ← years - 1
        months ← months + 12
    FinSi

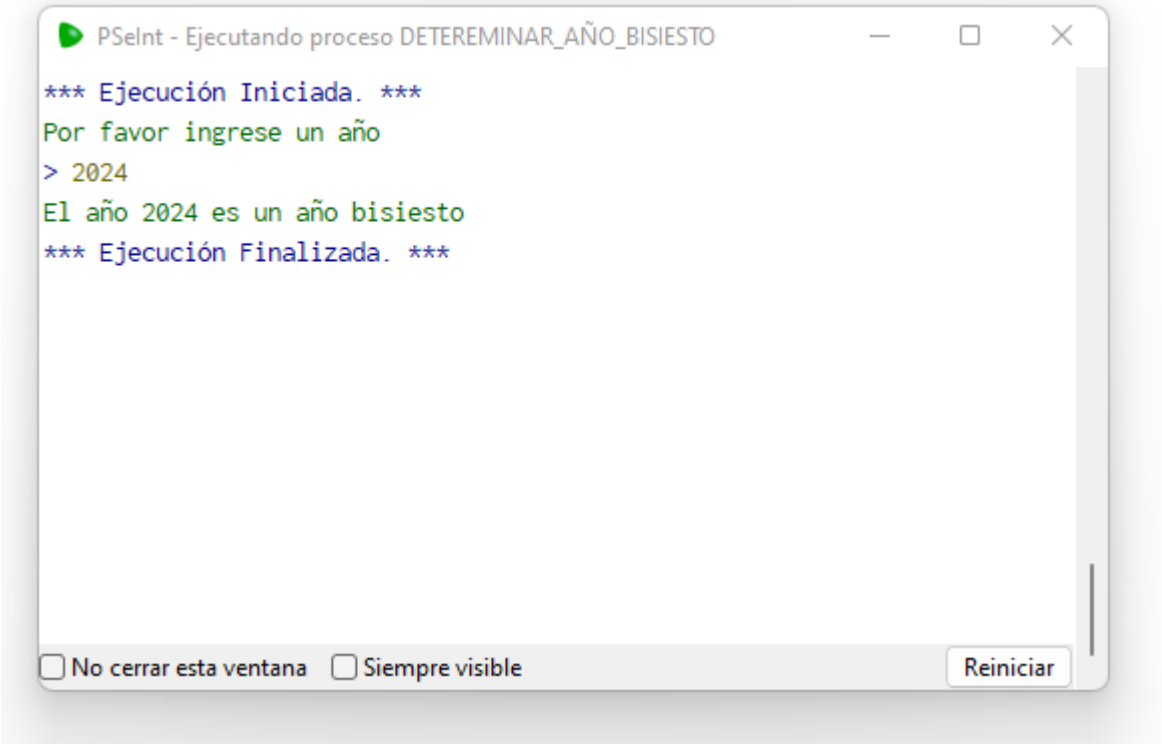
    Escribir "Su edad es de: ", years, " años ", months, " mes(es) y " days, " día(s)"
FinAlgoritmo
```

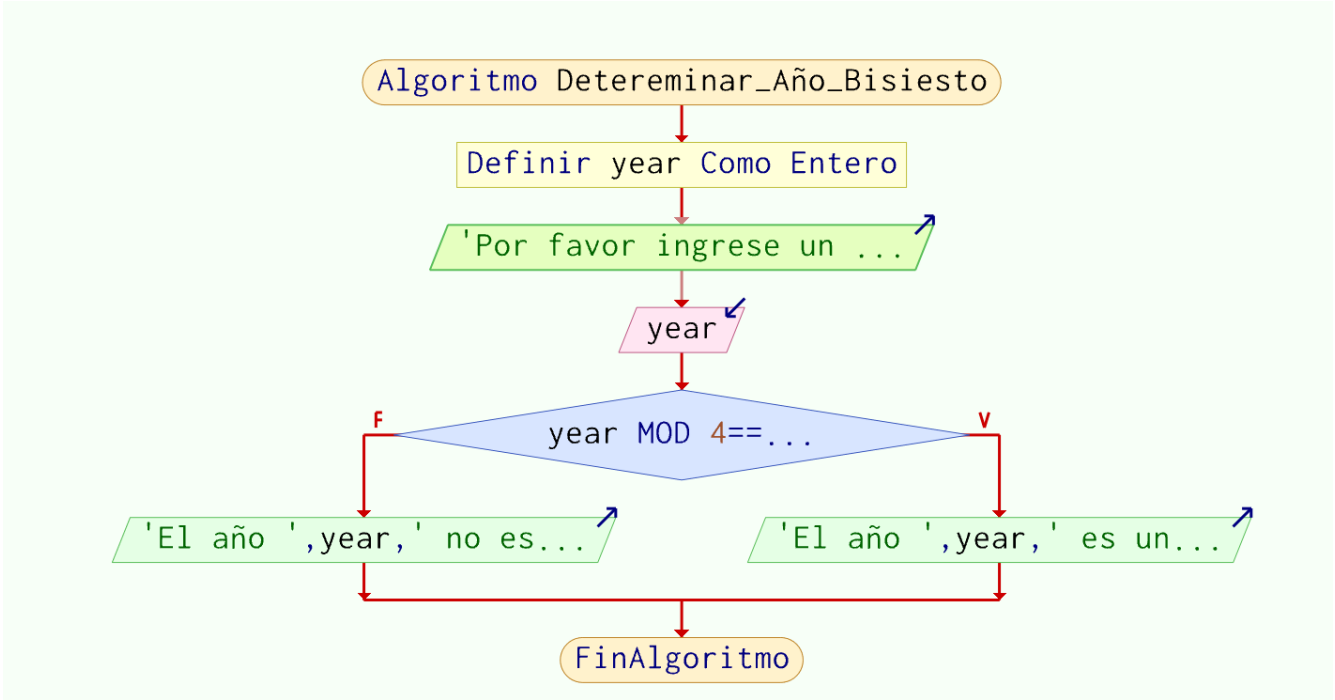



Algoritmo Edad_Actual
Definir birthDay,curre...
'Por favor ingrese su ...
birthDay,monthBirth,ye...
'Por favor ingrese el ...
currentDay,currentMont...
years ← currentYear-ye...
days ← currentDay-birt...
months ← currentMonth-...
days<0
F
V
months ← months-1
days ← days+30
months<0
F
V
years ← years-1
months ← months+12
'Su edad es de: ',year...
FinAlgoritmo

- **Algoritmo que permita determinar si un año indicado es o no un año bisiesto.**

```
1   Algoritmo Detereminar_Año_Bisiesto
2       Definir year Como Entero
3       Escribir "Por favor ingrese un año"
4       leer year
5       Si year mod 4 == 0 y year mod 400 ≠ 0  o year mod 100 == 0 Entonces
6           Escribir "El año ", year, " es un año bisiesto"
7       SiNo
8          Escribir "El año ", year, " no es un año bisiesto"
9       Fin Si
10  FinAlgoritmo
11
```




Algoritmo Detereminar_Año_Bisiesto
Definir year Como Entero
'Por favor ingrese un ...
year
F
year MOD 4==...
V
'El año ',year,' no es...
'El año ',year,' es un...
FinAlgoritmo

TECNOLOGO EN ANÁLISIS Y DESARROLLO DE SOFTWARE



EVIDENCIA DE CONOCIMIENTO: GA3-220501093-AA2-EV01 FUNDAMENTOS DE PROGRAMACIÓN ESTRUCTURADA Y ESTRUCTURAS CÍCLICAS

APRENDIZ:

JORGE LUIS CARDOZO

FICHA:

2455255

INSTRUCTOR:

CARLOS ERNESTO MAYA VALLEJO

TABLA DE CONTENIDO

INTRODUCCION 3

ALGORITMO QUE A PARTIR DE UNA FECHA DE NACIMIENTO Y UNA FECHA ACTUAL DETERMINE LA EDAD EN AÑOS ACTUAL DE UNA PERSONA. 4

CONDICIONES PARA CALCULAR LA FECHA DE UNA PERSONA A PARTIR DE LA FECHA DE NACIMIENTO Y LA FECHA ACTUAL. 5

..... 5

..... 5

ALGORITMO QUE PERMITA DETERMINAR SI UN AÑO INDICADO ES O NO UN AÑO BISIESTO. 7

CONDICIONES PARA QUE UN AÑO SEA BISIESTO. 8

CONCLUSIONES 10

CITAS BIBLIOGRAFÍAS 11

INTRODUCCION

En el presente documento estaré desarrollando diferentes algoritmos con seudocódigos y diagramas de flujo. Este análisis me ayuda a aprender de cómo implementar y programar de manera eficaz junto con la práctica.

Para mi concepto personal esto me ayuda a profundizar más de cómo va estructurado todos fundamentos del desarrollo de software.

ALGORITMO QUE A PARTIR DE UNA FECHA DE NACIMIENTO Y UNA FECHA ACTUAL DETERMINE LA EDAD EN AÑOS ACTUAL DE UNA PERSONA.

Algoritmo con Análisis y Diseño del ejercicio "Determinar la Edad en Años de una persona a partir de la fecha de Nacimiento y la fecha actual". El Pseudocódigo se realiza en el programa PSeInt..

ANÁLISIS

- ***Formular el problema:*** *algoritmo que permite obtener la edad en años de una persona a partir de la fecha de nacimiento y la fecha actual.*
- ***Datos disponibles:*** *fecha de nacimiento y fecha actual.*
- ***Restricciones:*** *ninguna.*
- ***Procesos necesarios:*** *solicitar al usuario la fecha de nacimiento indicando el día, mes y año; igualmente el usuario debe ingresar la fecha actual indicando día, mes y año.*

CONDICIONES PARA CALCULAR LA FECHA DE UNA PERSONA A PARTIR DE LA FECHA DE NACIMIENTO Y LA FECHA ACTUAL.

FECHA NAC	DIA_NACIMIENTO.	MES_NACIMIENTO	AÑO_NACIMINETO	DIA _ACTUAL	MES_ACTUAL	AÑO_ACTUAL	EDAD
8/05/1988	8	5	1988	8	8	2022	34
9/05/1988	9	5	1988	8	8	2022	34
10/08/1988	10	5	1988	8	8	2022	34
8/05/1988	8	5	1988	8	8	2022	34

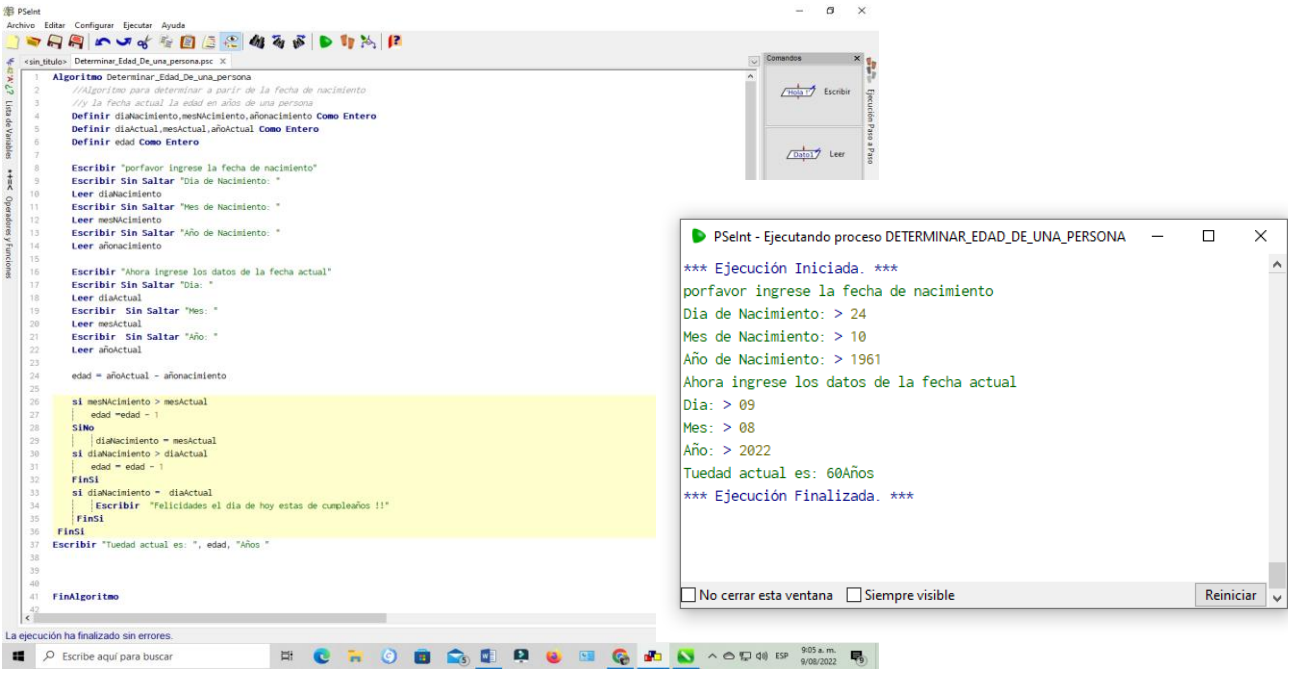
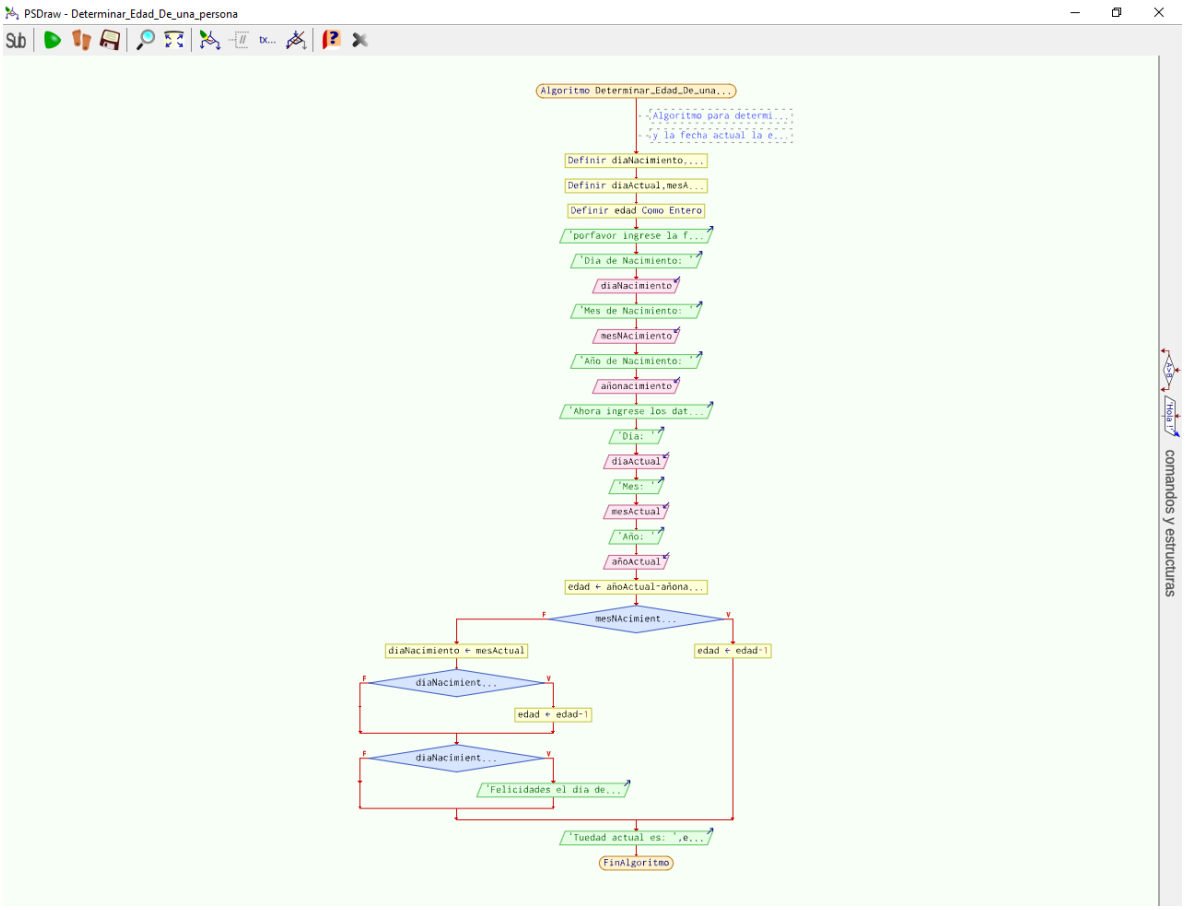


Diagrama De Flujos:



ALGORITMO QUE PERMITA DETERMINAR SI UN AÑO INDICADO ES O NO UN AÑO BISIESTO.

Algoritmo con Análisis y Diseño del ejercicio "Determinar si un año es bisiesto o no"

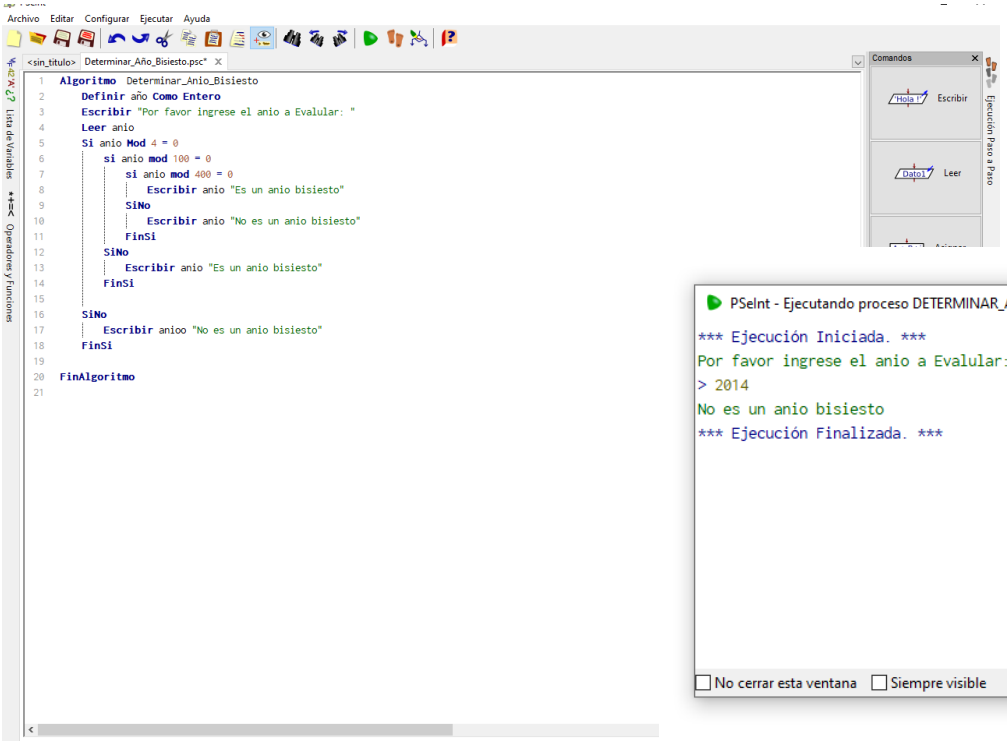
ANÁLISIS

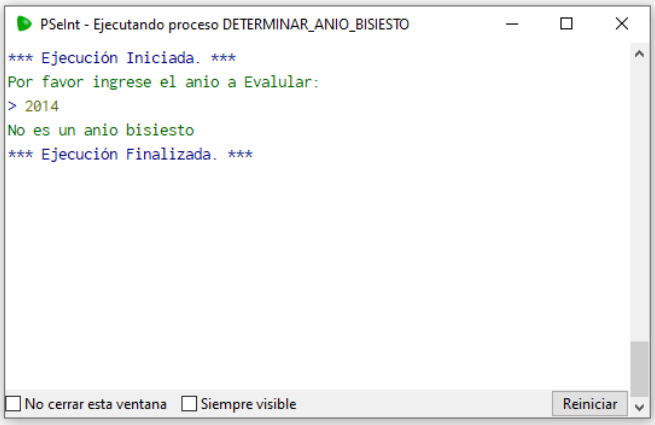
- ***Formular el problema:*** *de acuerdo a lo planteado en las condiciones determinar si el valor ingresado correspondiente a un año es o no año bisiesto.* ***Resultados esperados:*** *imprimir un mensaje que nos indique que el año ingresado es o No bisiesto.*
- ***Datos disponibles:*** *un número que representa un año ingresado por el usuario desde el teclado.*
- ***Restricciones:*** *ninguna.*
- ***Procesos necesarios:*** *solicitar al usuario ingrese un año, evaluar si es bisiesto. En caso de no cumplir las condiciones para ser bisiesto reportar "El año ingresado es bisiesto" en caso contrario reportar "El año ingresado no es bisiesto"*

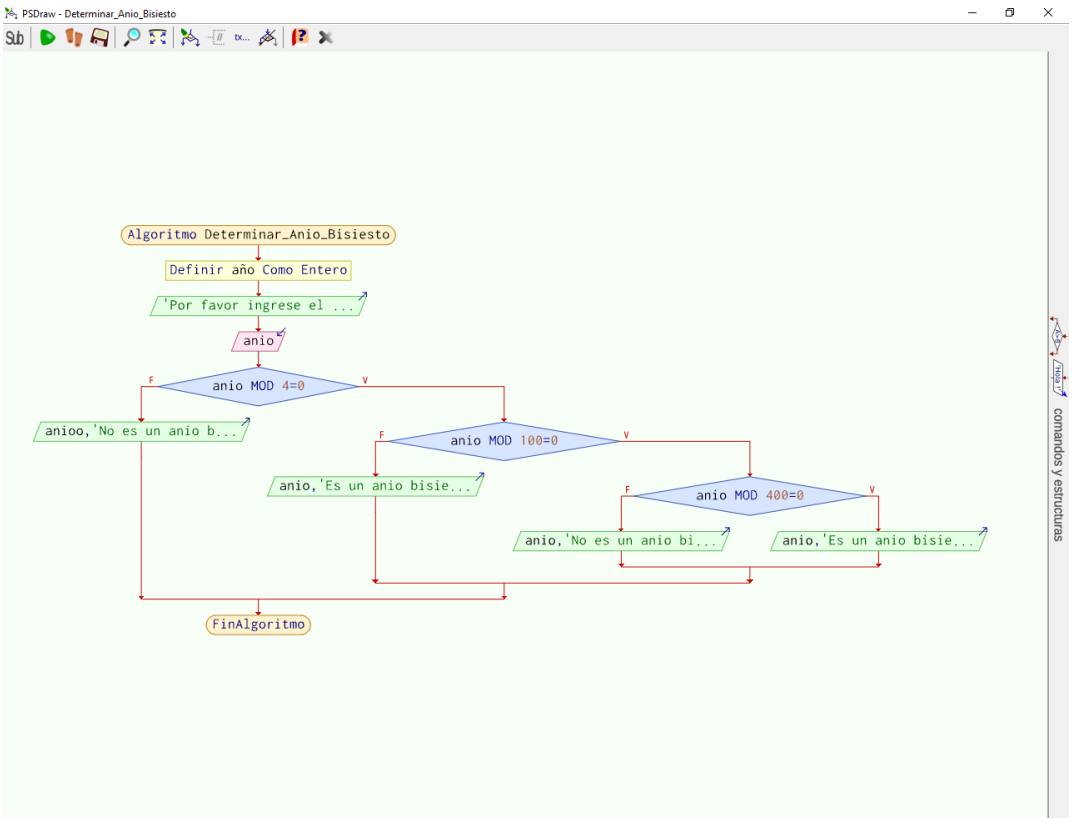
CONDICIONES PARA QUE UN AÑO SEA BISIESTO.

- ✓ *Debe ser divisible entre 4*
- ✓ *No debe ser divisible entre 100*
- ✓ *Debe ser divisible entre 400*

Año	Divisible entre 4	Divisible entre 100	Divisible entre 400	Mensaje
2014	C =503 R =0	C =20 R=4		Año no es bisiesto
1800	C =450 R =0	C =18 R=0	C =4 R=300	Año no es bisiesto
2400	C = 600 R = 0	C = 24 R=0	C =6 R= 0	Año bisiesto
2100	C =525 R = 0	C = 21 R= 0	C = 5 R= 100	Año no es bisiesto






PSDraw - Determinar_Anio_Bisiesto
Algoritmo Determinar_Anio_Bisiesto
Definir año Como Entero
'Por favor ingrese el ...'
anio
anio MOD 4=0
F
V
anioo,'No es un anio b...'
anio MOD 100=0
F
V
anio,'Es un anio bisie...'
anio MOD 400=0
F
V
anio,'No es un anio bi...'
anio,'Es un anio bisie...'
FinAlgoritmo
comandos y estructuras

CONCLUSIONES

Este trabajo me permite aprender sobre los algoritmos y de cómo puedo implementar esto para calcular la edad y el año ya sea bisiesto o no.

CITAS BIBLIOGRAFÍAS

https://sena.territorio.la/content/index.php/institucion/Titulada/institution/SENA/Tecnologia/228118/Contenido/OVA/CF13/index.html#/curso/tema1#t_1_2

https://diagramaweb.com/simbologia-de-diagrama-de-proceso-2/

SENA

PLATAFORMA DE APRENDIZAJE VIRTUAL

GA3-220501093-AA2-EV01 fundamentos de programación estructurada y estructuras cíclicas

PRESENTADO POR

MANUEL CUBILLOS

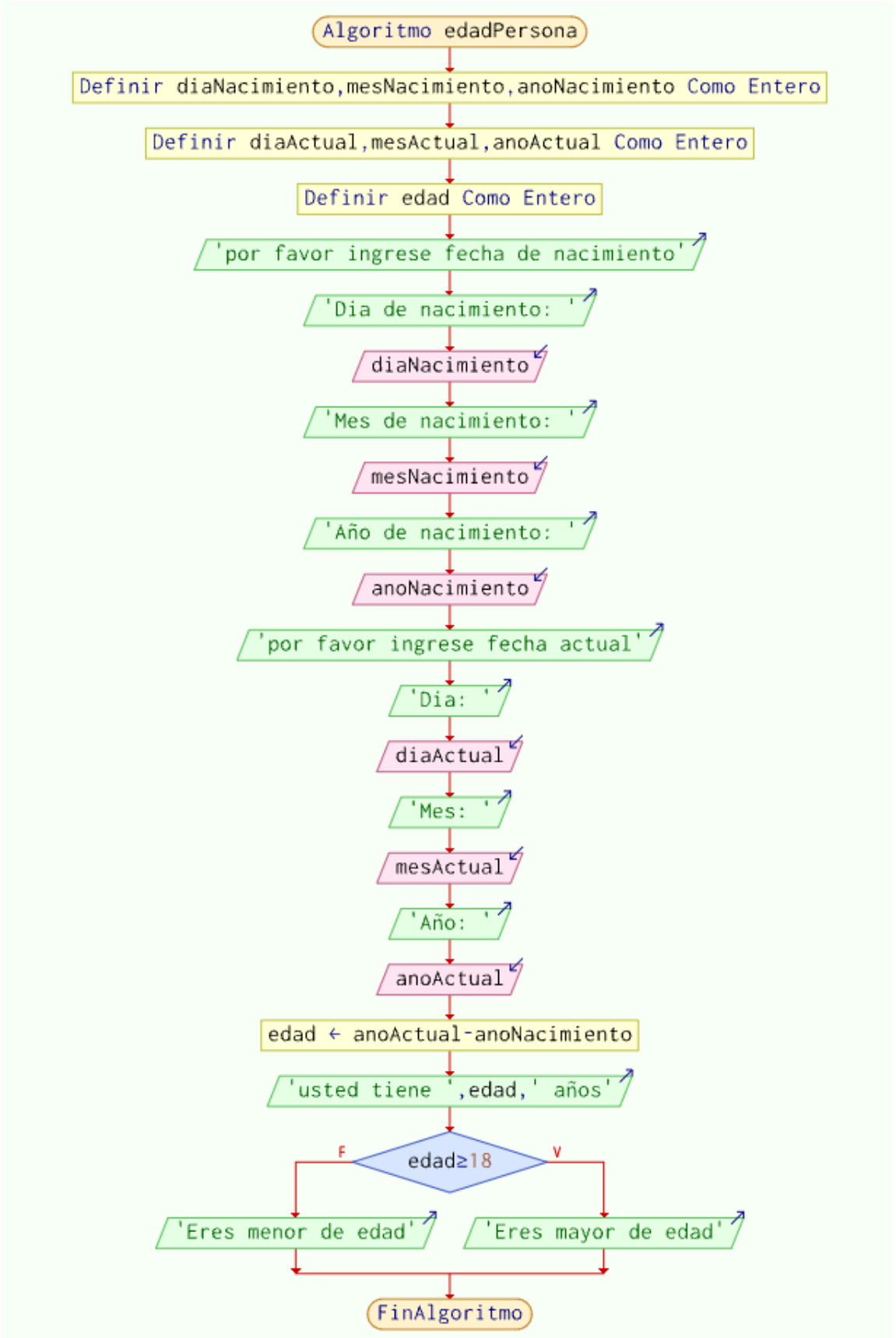
INSTRUCTOR:

CARLOS MAYA

Marzo, 2022

1. Algoritmo que a partir de una fecha de nacimiento y una fecha actual determine la edad en años actual de una persona.

```
Algoritmo edadPersona
    definir diaNacimiento, mesNacimiento,anoNacimiento Como entero
    definir diaActual, mesActual, anoActual Como Entero
    definir edad Como Entero
    Escribir "por favor ingrese fecha de nacimiento"
    Escribir "Dia de nacimiento: "
    Leer diaNacimiento
    Escribir "Mes de nacimiento: "
    Leer mesNacimiento
    Escribir "Año de nacimiento: "
    Leer anoNacimiento
    Escribir "por favor ingrese fecha actual"
    Escribir "Dia: "
    Leer diaActual
    Escribir "Mes: "
    Leer mesActual
    Escribir "Año: "
    Leer anoActual
    edad = anoActual - anoNacimiento
    Escribir "usted tiene ",edad " años"
    Si edad >= 18
        Escribir "Eres mayor de edad"
    SiNo
        Escribir  "Eres menor de edad"
    FinSi
FinAlgoritmo
```


Algoritmo edadPersona
Definir diaNacimiento,mesNacimiento,anoNacimiento Como Entero
Definir diaActual,mesActual,anoActual Como Entero
Definir edad Como Entero
'por favor ingrese fecha de nacimiento'
'Dia de nacimiento: '
diaNacimiento
'Mes de nacimiento: '
mesNacimiento
'Año de nacimiento: '
anoNacimiento
'por favor ingrese fecha actual'
'Dia: '
diaActual
'Mes: '
mesActual
'Año: '
anoActual
edad ← anoActual-anoNacimiento
'usted tiene ',edad,' años'
F
edad≥18
V
'Eres menor de edad'
'Eres mayor de edad'
FinAlgoritmo

2. Algoritmo que permita determinar si un año indicado es o no un año bisiesto.

```
Algoritmo añoBisiesto
    definir ano Como Entero
    Escribir "ingresa un año"
    Leer ano
    si ano mod 4 == 0 y ((ano mod 100 <> 0) o (ano mod 400=0)) Entonces
        Escribir ano " es un año bisiesto"
    SiNo
        Escribir ano " no es un año bisiesto"
    FinSi
FinAlgoritmo
```

