Universidad Autónoma del Estado de México
UAEM

Unidad 4. Estructuras de control
Tema. Estructuras de Iteración (do-while)

Juan Pablo Cobá Juárez Pegueros
Programación Avanzada
Bioingeniería Médica
Facultad de Medicina

05/09/2017

Contenido

- Qué son las estructuras de control iterativas ?
- ¿cómo se ejecuta?
- ¿cómo estructura la condición de repetición?
- Componentes
- Variable de control
- Condición
- Incremento
- Diagrama de flujo y codificación en c
- Característica del ciclo do - while
- Ejercicios
- Lógica para imprimir en pantalla del 1 al 100
  Codificación
- Encontrar los factores de un número
- Encontrar los factores de un número
- ¿cuál es el factor de un número?
- Lógica para encontrar todos los factores de un número
- Codificación
- Calcular la tabla de multiplicar de un número
- Lógica para calcular la tabla de multiplicar de un número
- Codificación
- Calcular el producto de los dígitos de un número
- Lógica para encontrar el producto de los dígitos de un número paso a paso
- Codificación
- Bibliografía

Objetivo: Aplicar las estructuras de control en el lenguaje de programación C codificando instrucciones sintáctica y semánticamente correctas para controlar el flujo de ejecución en un programa computacional.

Unidad 4. Estructuras de control

Estructuras de Iteración (do-while)

¿Qué son las estructuras de control iterativas ?

- Son estructuras que permiten permiten la ejecución repetida de una secuencia de instrucciones que pueden ser:
  - Instrucciones de entrada y salida
  - Estructuras de Selección (if, if-else, switch)

¿Cómo se ejecuta?

- El número de veces que el bloque de instrucciones se ejecutará se puede especificar de manera **explícita** o a través de **una condición** que indica cuando se ejecuta de nuevo o cuando no

¿Cómo se ejecuta?

- **explícita** nosotros conocemos el valor inicial y el valor final del ciclo

- **condición** el valor que regrese la condición determina el inicio o fin del ciclo

¿Cómo estructura la condición de repetición ?

- Es una estructura de repetición condicional, es decir requiere evaluar una expresión relacional o lógico - relacional

¿Cómo estructura la condición de repetición ?

- Es una estructura de repetición condicional, es decir requiere evaluar una expresión relacional o lógico - relacional
- Ejemplo

¿Cómo estructura la condición de repetición ?

- Es una estructura de repetición condicional, es decir requiere evaluar una expresión relacional o lógico - relacional
- Ejemplo
  - X>10

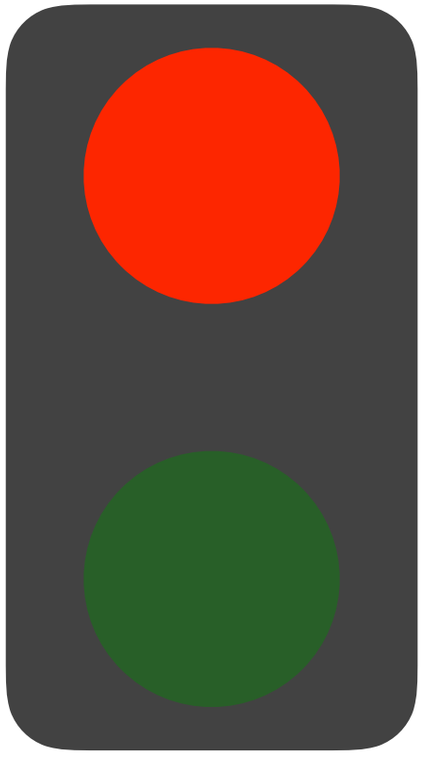
Relacional

¿Cómo estructura la condición de repetición ?

- Es una estructura de repetición condicional, es decir requiere evaluar una expresión relacional o lógico - relacional
- Ejemplo
  - X>10 — Relacional
  - z>10 && z>20 — Lógico - relacional

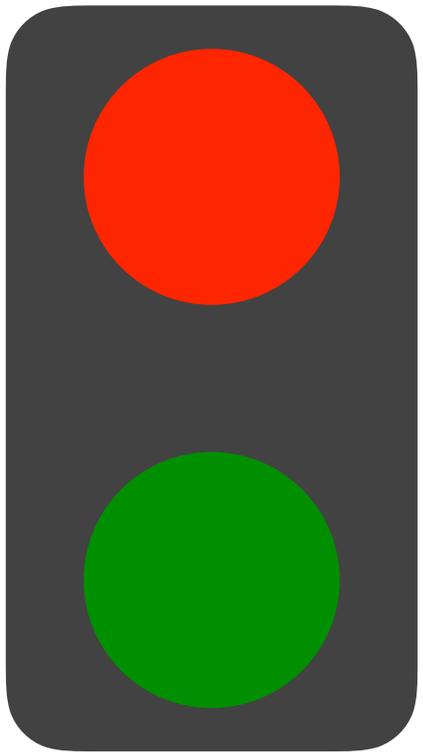
¿Cómo estructura la condición de repetición ?

- La evaluación de la expresión lógico - relacional determina:
- Termina si se evalúa Falsa



¿Cómo estructura la condición de repetición ?

- La evaluación de la expresión lógico - relacional determina:
  - Termina si se evalúa Falsa
  - Se repite si se evalúa Verdadera



- De forma general un ciclo repetitivo do - while esta conformado de los siguientes componentes:

- De forma general un ciclo repetitivo do - while esta conformado de los siguientes componentes:

  – Variable de control de ciclo

- De forma general un ciclo repetitivo do - while esta conformado de los siguientes componentes:
  - Variable de control de ciclo
  - Incremento

- De forma general un ciclo repetitivo do - while esta conformado de los siguientes componentes:
  - Variable de control de ciclo
  - Incremento
  - Condición

Variable de control

- Variable de control se utiliza para “controlar” el flujo de control del programa

Variable de control

- Variable de control se utiliza para “controlar” el flujo de control del programa

Variable de control

- Variable de control se utiliza para “controlar” el flujo de control del programa

- Puede visualizarse como un contador que puede incrementar o decrementar su valor

Variable de control

- Variable de control se utiliza para "controlar" el flujo de control del programa

- Puede visualizarse como un contador que puede incrementar o decrementar su valor
- Usualmente se utiliza i como variable de control

Condición

- Controla el momento en que se repite la estructura y el momento en que debe parar

Condición

- Controla el momento en que se repite la estructura y el momento en que debe parar
- Su función es la de un interruptor que se encuentra encendido si la condición es evaluada como verdadera y se cierra de ser falsa.

- Controla el momento en que se repite la estructura y el momento en que debe parar
- Su función es la de un interruptor que se encuentra encendido si la condición es evaluada como verdadera y se cierra de ser falsa.

- Su función es modificar el valor de la variable de control, lo que permite modificar el valor dentro de la condición en cada iteración.

Diagrama de Flujo y codificación en c

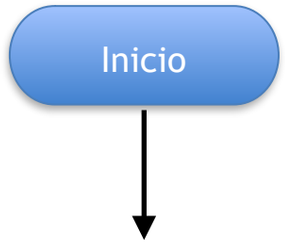


Diagrama de Flujo y codificación en c

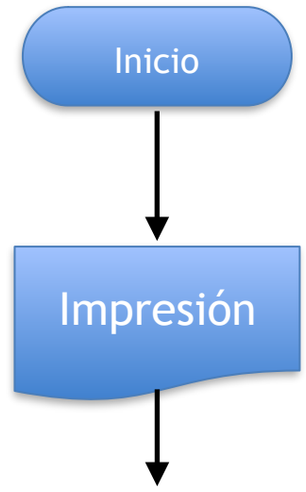


Diagrama de Flujo y codificación en c

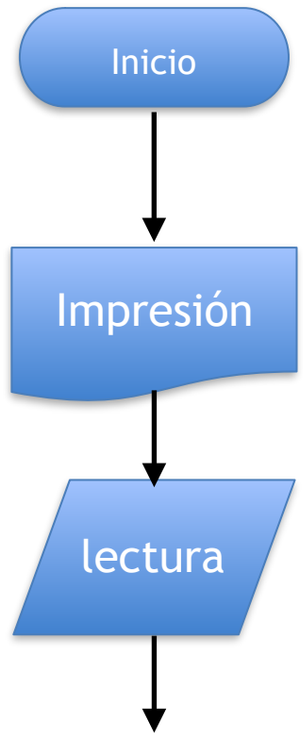


Diagrama de Flujo y codificación en c

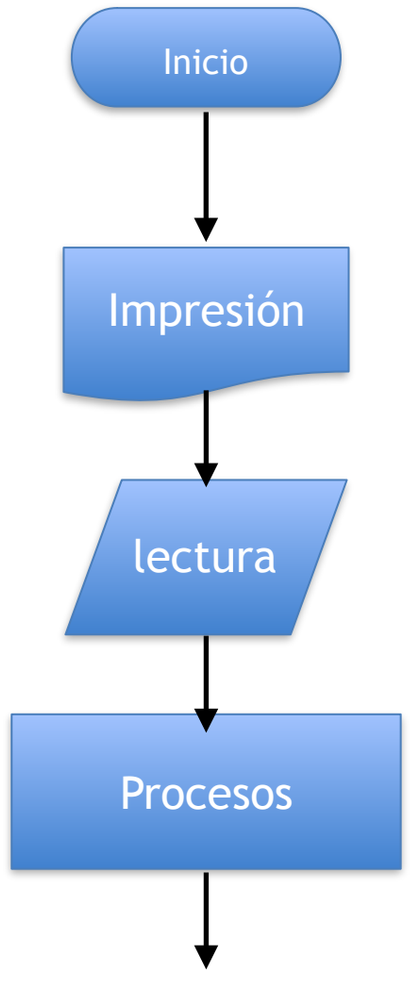


Diagrama de Flujo y codificación en c

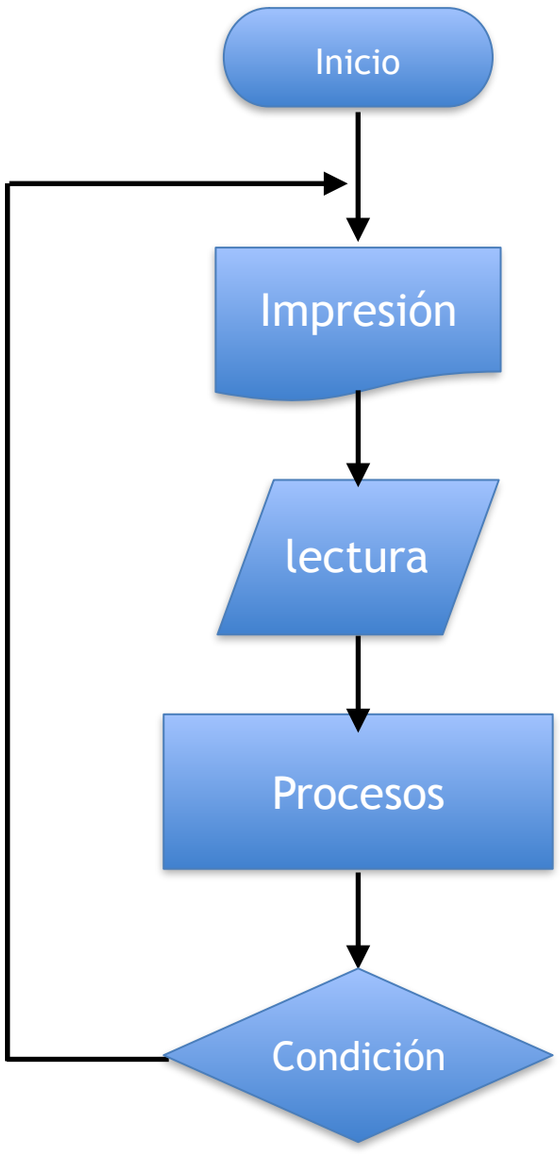


Diagrama de Flujo y codificación en c

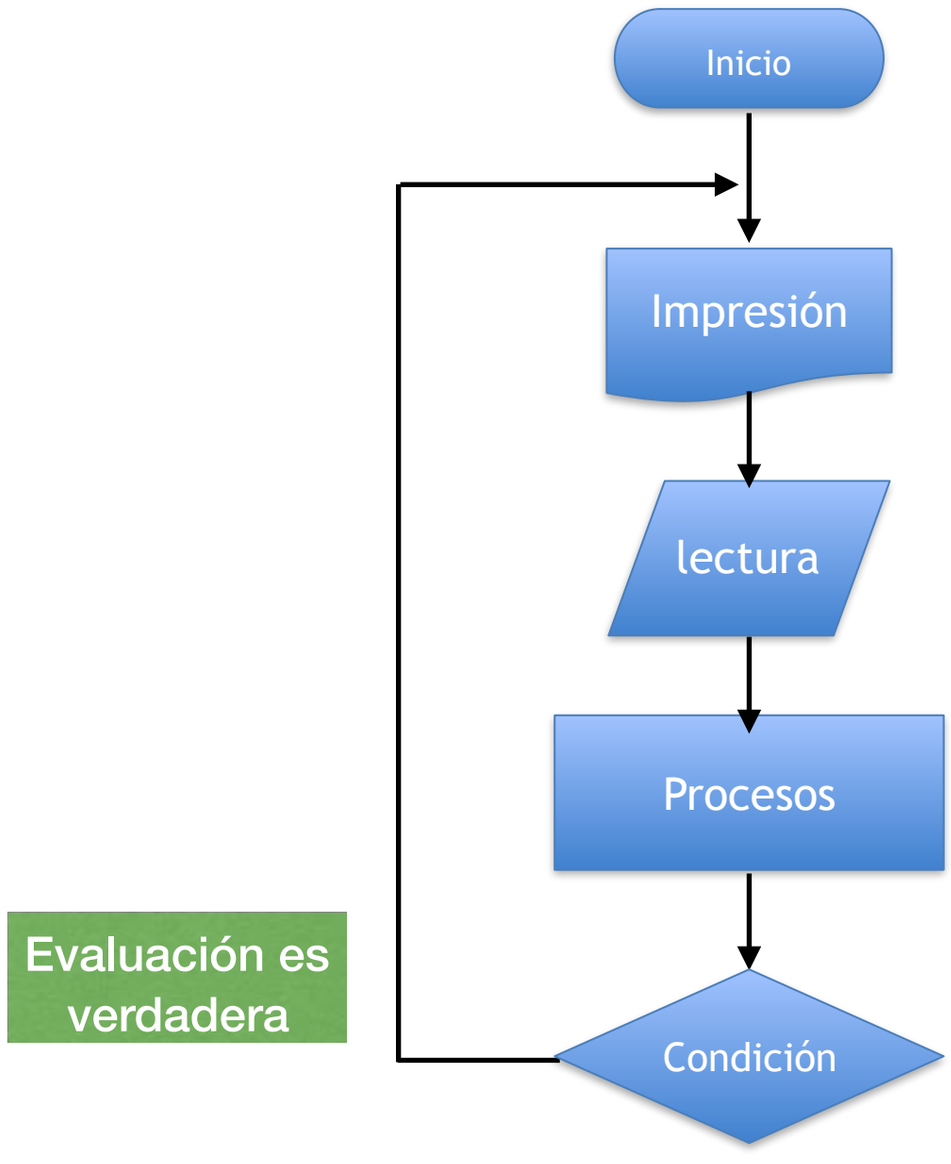


Diagrama de Flujo y codificación en c

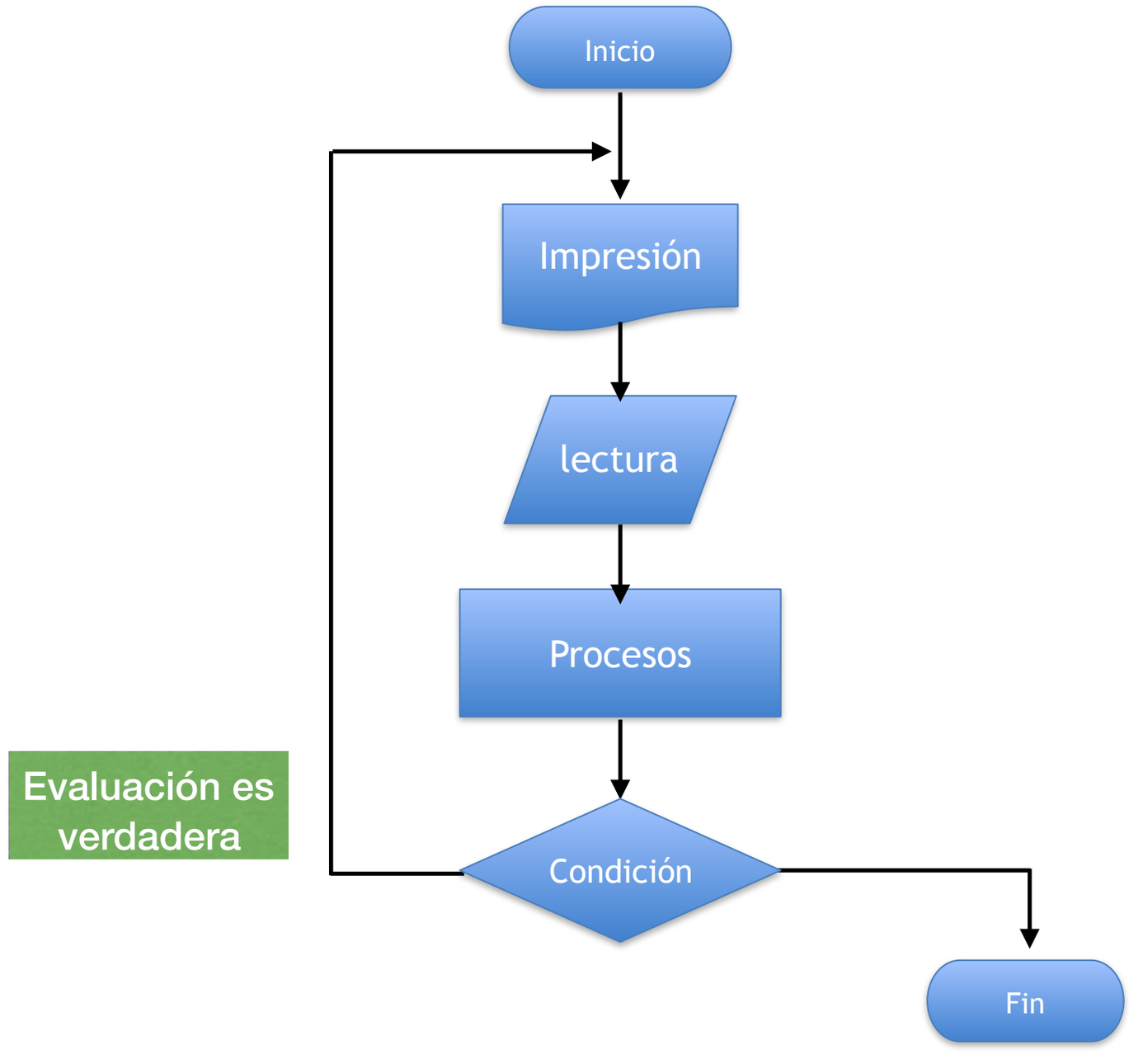


Diagrama de Flujo y codificación en c

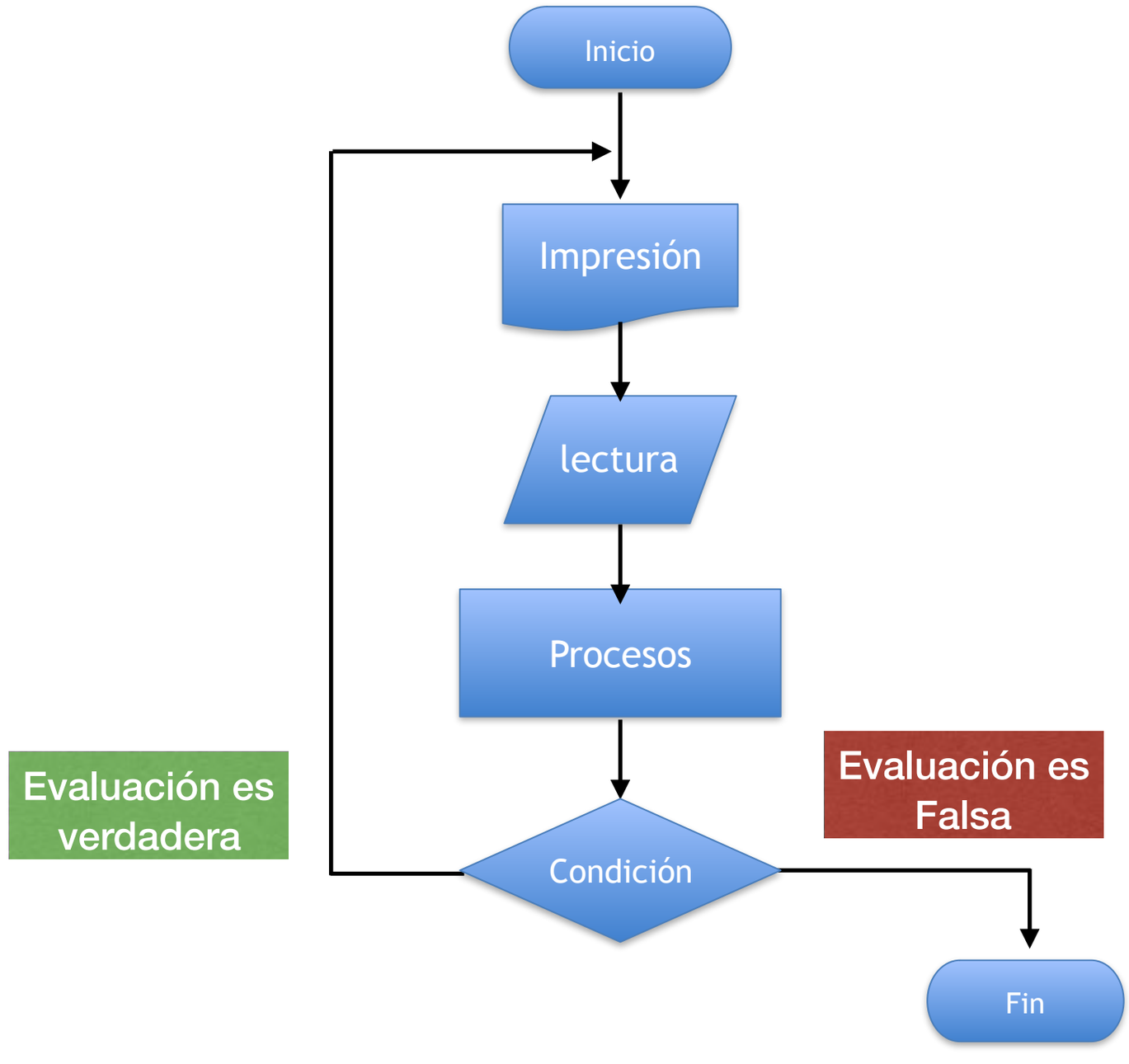


Diagrama de Flujo y codificación en c

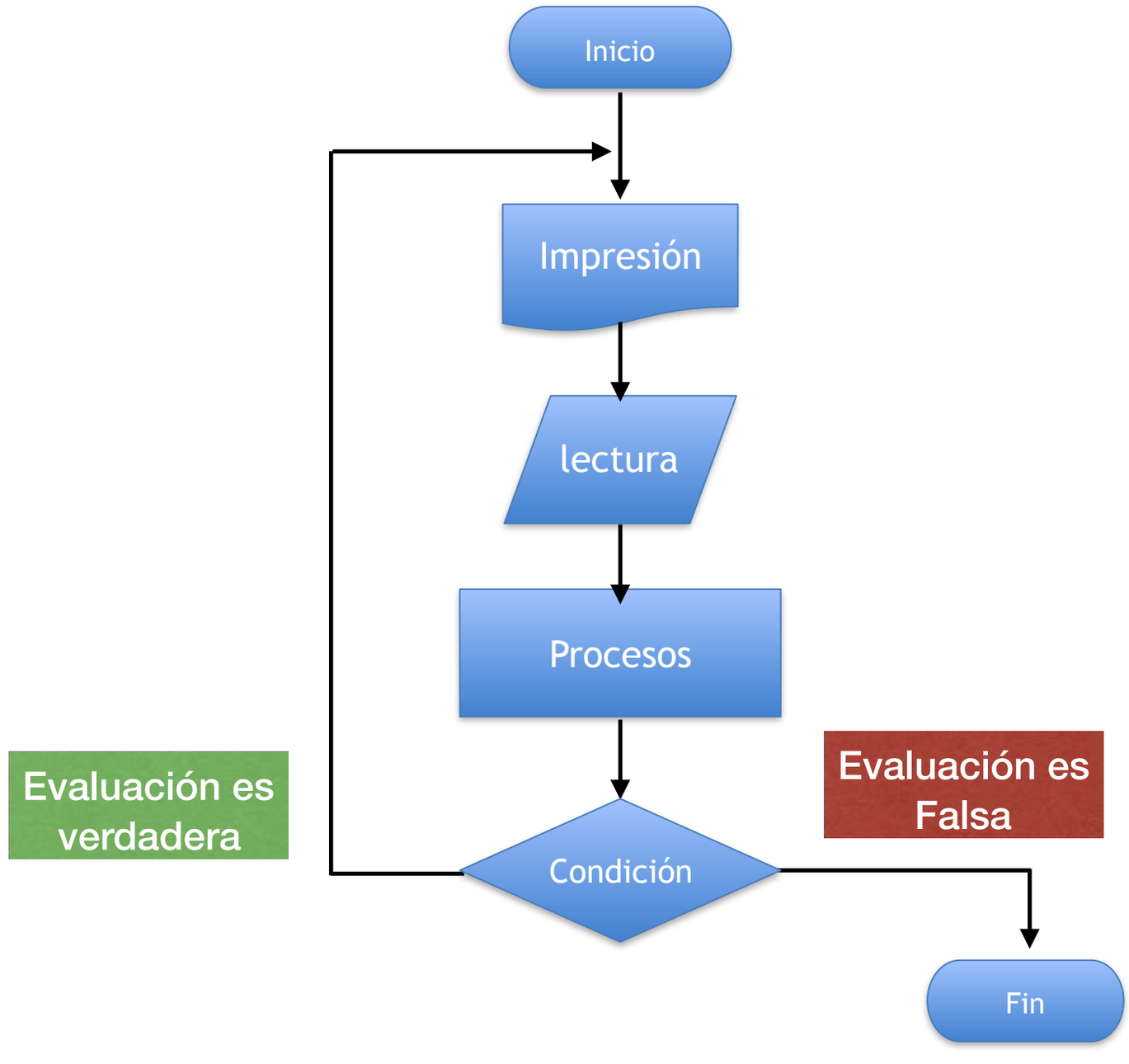


Diagrama de Flujo y codificación en c

do

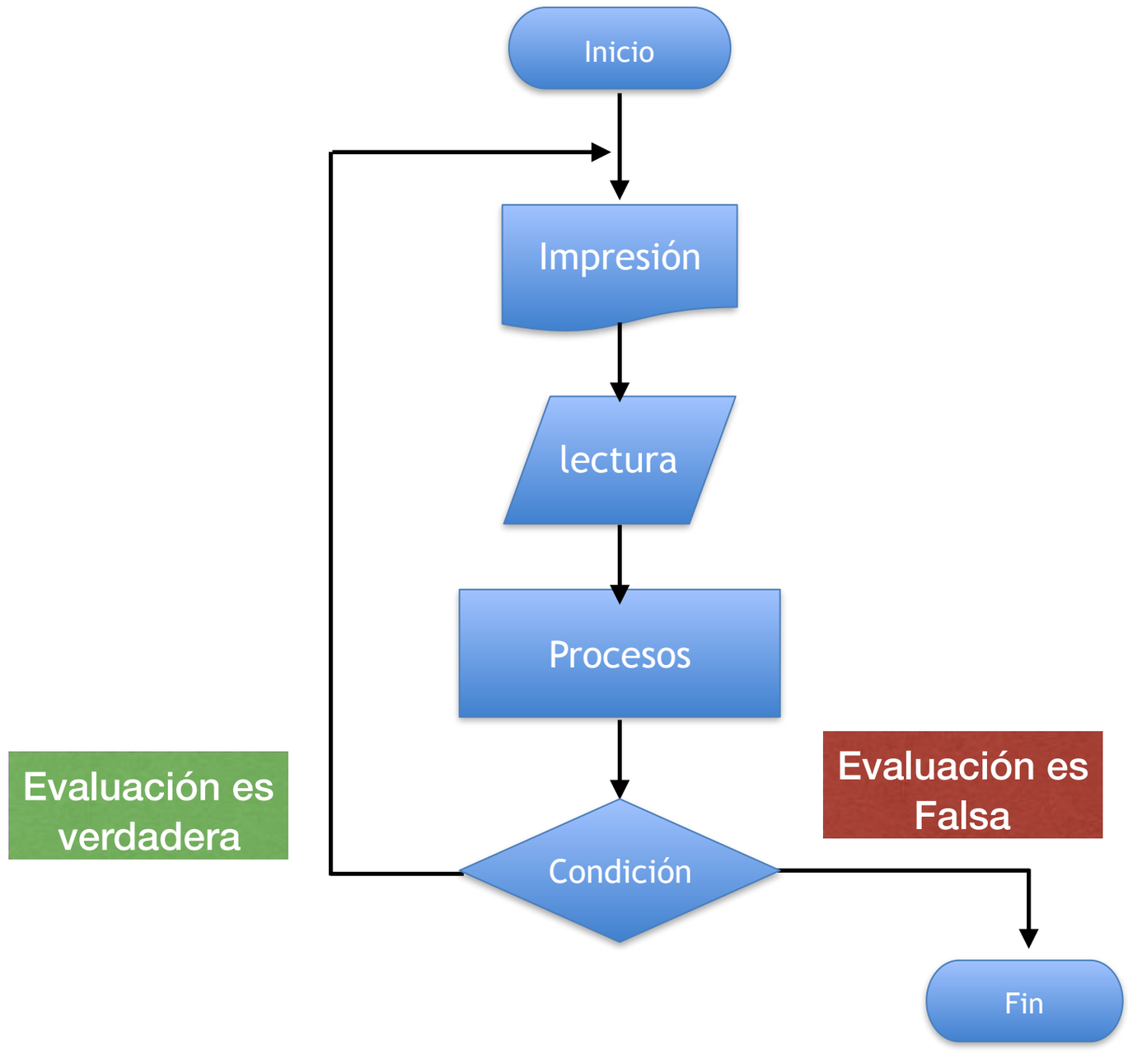


Diagrama de Flujo y codificación en c

```
do
{
```

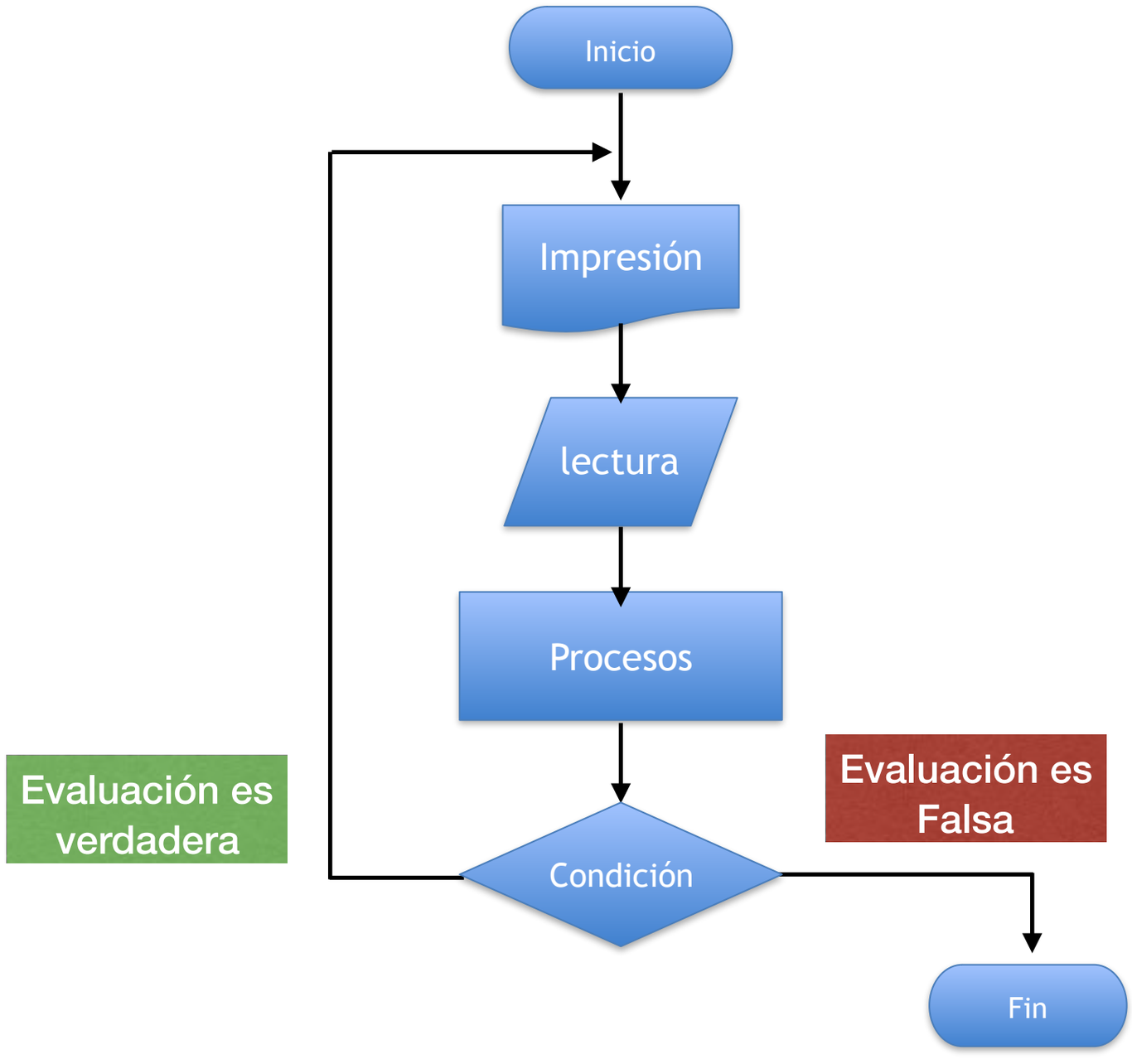


Diagrama de Flujo y codificación en c

```
do
{
    // cuero del ciclo do
    while
```

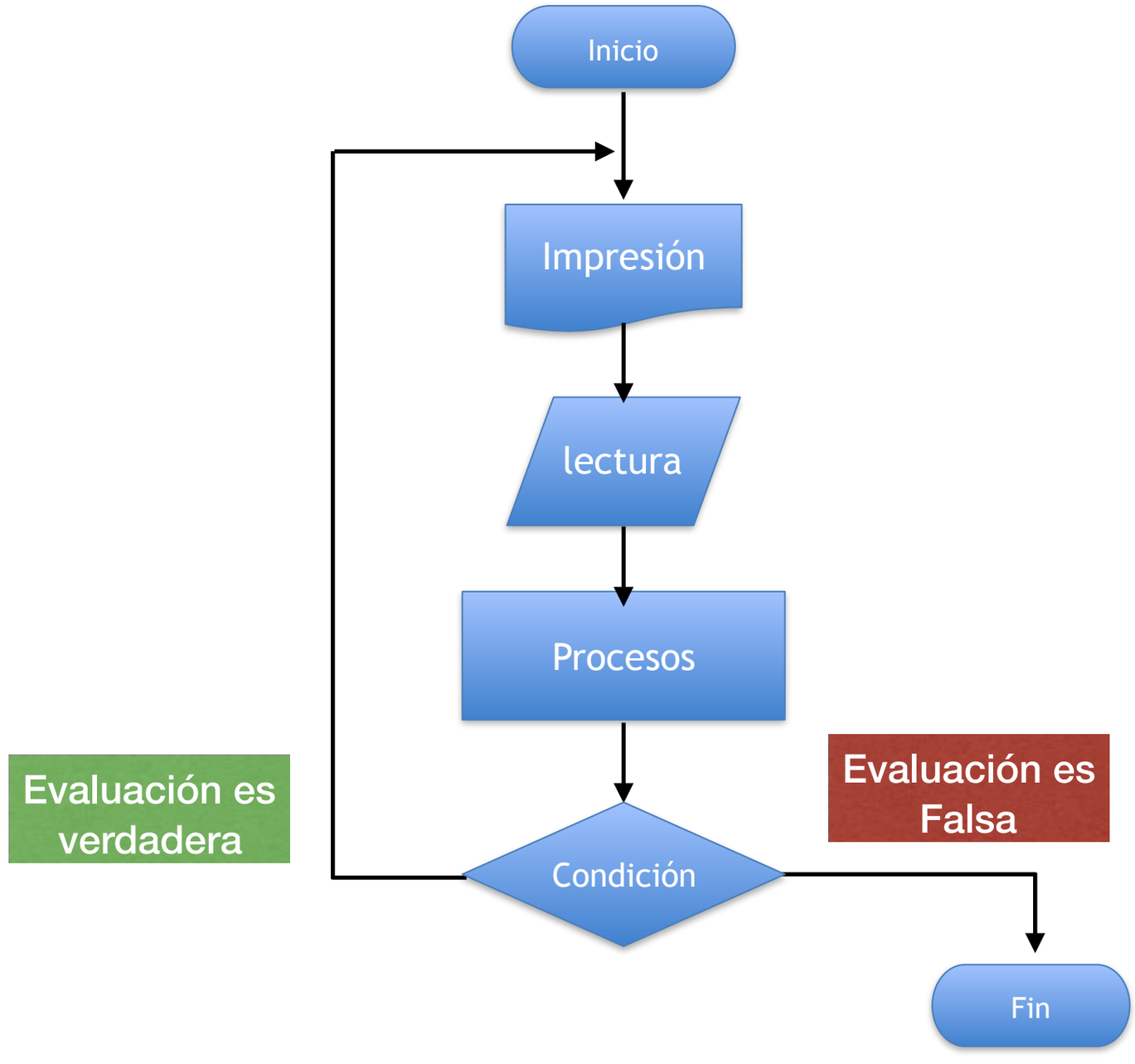
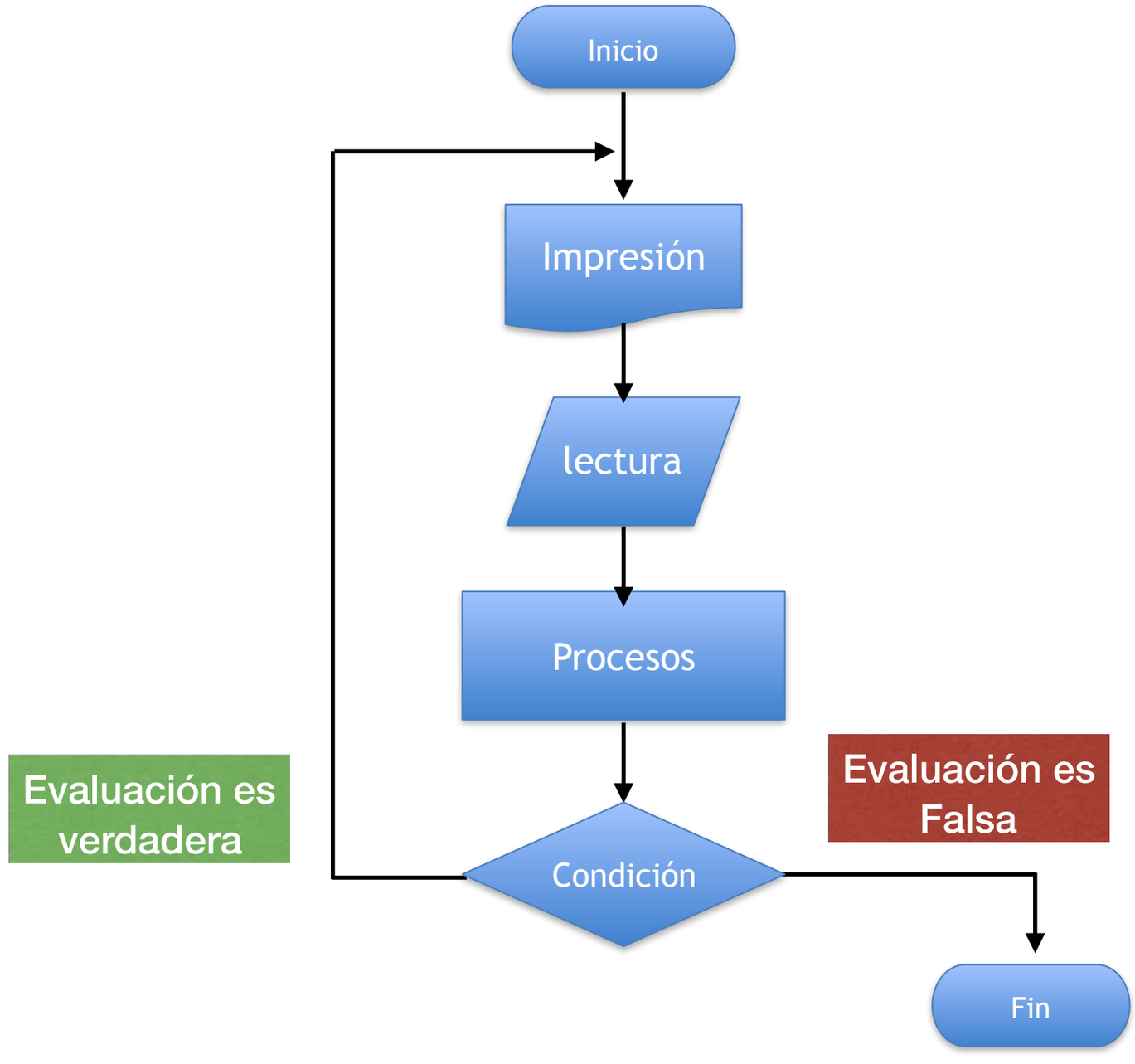


Diagrama de Flujo y codificación en c

```
do
{
    // cuero del ciclo do
    while
} while (condition);
```



Característica del ciclo do - while

- Por si estructura el ciclo do - while se ejecuta por lo menos una ven durante la ejecución del algoritmo

Universidad Autónoma del Estado de México
UAEM

Ejercicios

Imprimir en pantalla del 1 al 100

Codifique un programa en C que imprima en pantalla los 100 primeros números enteros utilice el ciclo Do - while.

Lógica para imprimir en pantalla del 1 al 100

1. Establecer i como la variable de control e iniciarla en 0
2. Establecer el inicio del ciclo “do”
3. Incrementa la variable de control i<—i+1
4. Imprime el valor de i.
5. Mientras (while) la condición (i <=99) sea verdadera

```
#include <stdio.h>
int main(int argc, char *argv[]) {
    int i=0;
    do{
    i=i+1;
    printf(" %d",i);
    }while (i<=99) ;
}
```

Encontrar los factores de un número

Codifique un programa en C que permita ingresar un número del usuario y encuentre todos los factores del número dado usando el ciclo de repetición Do - while. que permita encontrar factores de un número

¿Cuál es el factor de un número?

El factor de cualquier número es un número entero que divide exactamente el número en un número entero sin dejar ningún resto.

Por ejemplo: 2 es un factor de 6 porque 2 divide 6 exactamente dejando ningún residuo.

Lógica para encontrar todos los factores de un número

Paso a paso la lógica descriptiva para encontrar todos los factores de un número.

1. Introduzca el número del usuario. Guárdelo en la variable num
2. Establecer el inicio del ciclo “do”
3. Para cada iteración dentro de ciclo compruebe si la variable de control del ciclo i es un factor de num o no. Para comprobar el factor, verificamos la divisibilidad del número realizando la división del módulo, es decir, si (num% i == 0) i es un factor de num.

   Si i es un factor de num, entonces imprime el valor de i.
4. Incrementar i en 1
5. Ejecutar el ciclo while i sea menor o igual que num. La condición del ciclo debe ser similar a (i <= num)

```
int main(int argc, char *argv[]) {
        int i=1;
        int num;
        printf("introduce un digito");
        scanf("%d",&num);
        do {
            if ((num%i) == 0) {
                printf("%d ",i);
            }
            i=i+1;
        } while (i <= num);
```

Calcular la tabla de multiplicar de un numero

Codifique un programa en C que permita introducir un número al usuario y calcule la tabla de multiplicación del número dado usando el ciclo Do - while.

Lógica para calcular la tabla de multiplicar de un numero

1. Establecer i como la variable de control e iniciarla en 1
2. Inicializar otra variable para almacenar producto, es decir, producto = 1
3. Introduzca el número del usuario. Guárdelo en la variable num
4. Establecer el inicio del ciclo “do”
5. Multiplique num por i y guarde el resultado en la variable producto
6. Imprime el valor de la variable producto.
7. Incrementa la variable de control i<—i+1
8. Mientras (while) la condición (i <=10) sea verdadera

```
#include <stdio.h>
int main(int argc, char *argv[]) {

        int i=1;
        int num;
        int producto=0;
        printf("introduce un digito");
        scanf(“%d",&num);

        do {

                producto =num*i;
                printf("%d ",producto);
                i=i+1;
        } while (i <= 10);
}
```

Calcular el producto de los dígitos de un numero

Codifique un programa en C que permita introducir un número al usuario y calcular el producto de sus dígitos. Para encontrar el producto de los dígitos utilice un ciclo de repetición Do -While.

Lógica para encontrar el producto de los dígitos de un número paso a paso

1. Introduzca un número del usuario. Guárdelo en alguna variable dio .
2. Inicializar otra variable para almacenar producto, es decir, producto = 1.
3. Establecer el inicio del ciclo “do”
4. Encuentra el último dígito del número realizando la división de módulo en 10 ultDig = dig % 10 .
5. Multiplica el último dígito encontrado con el producto por ejemplo producto = producto * ultDig .
6. Quite el último dígito dividiendo el número por 10, es decir, num = num / 10 .
7. Repita el paso 4-6 hasta que el diga se convierta en 0 . Finalmente quedará con producto de dígitos en la variable producto .

```
#include <stdio.h>

int main(int argc, char *argv[]) {
        int i=0;
        int producto=1;
        int dig, ultDig;
        printf("introduce un digito");
        scanf("%d",&dig);
        do{
        ultDig = dig % 10;
        producto=producto*ultDig;
        dig=dig/10;
        printf(" %d",producto);
        }while (dig!=0) ;
}
```

Gracias !!!

05/09/2017

Bibliografía

- Kernighan y Ritchie 1995: El lenguaje de programación C,México, Prentice Hall.
- Cairó, Osvaldo. 2006: Fundamentos de Programación. Piensa en C, México: Pearson Prentice Hall
- Paul Deitel 2009: C:How to program, US: Prentice Hall Samuel Harbison 2002: C: A reference manual, US: Prentice Hall
- Aitken y Jones. 1994: Aprendiendo C en 21 días, México: Prentice Hall
- Gottfried, Byron. 2005: Programación en C (2a. Edición Revisada). México: Mc Graw Hill
- Joyanes, Aguilar Luis y Zahonero, Martínez I. 2005: Programación en C, Metodología, Estructura de Datos y Objetos, México: McGraw Hill.
- Joyanes, Aguilar Luis y Zahonero, Martínez I. 2002: Programación en C, Libro de problemas, México: McGraw Hill

PROGRAMACIÓN AVANZADA

D i a p o s i t i v a s

Material Didáctico sólo visión

Nombre del material:

Estructuras de Iteración Do-While.

Juan Pablo Cobá Juárez Pegueros
Facultad de Ingeniería
Bioingeniería Médica

Contenido

Presentación

La programación es el proceso de diseñar, escribir, depurar y mantener el código fuente de programas computacionales. El código fuente es escrito en un lenguaje de programación. Un lenguaje de programación es un idioma artificial diseñado para expresar computaciones que pueden ser llevadas a cabo por máquinas como las computadoras. Los programas creados pueden usarse para controlar el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana.

Por lo anterior es importante formar profesionales de la Bioingeniería Médica, con un alto sentido de responsabilidad, de ética y vocación de servicio, y con las competencias y aprendizajes necesarias.

Objetivo general de la unidad de aprendizaje

Implementar programas utilizando estructuras de datos y estructuras de control utilizando un lenguaje de alto nivel para la solución de problemas de bioingeniería.

Guion explicativo para el uso del material didáctico

Las unidades del programa son las siguientes:

UNIDAD 1. INTRODUCCIÓN AL LENGUAJE DE PROGRAMACIÓN C
UNIDAD 2. TIPOS, OPERADORES Y EXPRESIONES
UNIDAD 3. ENTRADA Y SALIDA DE DATOS
UNIDAD 4. ESTRUCTURAS DE CONTROL
UNIDAD 5. ARREGLOS UNIDIMENSIONALES Y MULTIDIMENSIONALES
UNIDAD 6. PROGRAMACIÓN MODULAR

El material se encuentra enfocado exclusivamente a la Unidad 4, específicamente en el tema 3 Estructuras de Iteración (for, while, do-while)

Este material está dirigido a toda persona interesada en el tema, pero específicamente a los alumnos de la unidad de aprendizaje Programación Avanzada.

El uso de este material es sencillo, ya que sólo contiene imágenes e ideas centrales del tema, que facilitan la concentración del alumno.

Orden de las diapositivas

1. Qué son las estructuras de control iterativas?
2. ¿cómo se ejecuta?
3. ¿cómo estructura la condición de repetición?
4. Componentes
5. Variable de control
6. Condición
7. Incremento
8. Diagrama de flujo y codificación en c
9. Característica del ciclo do - while
10. Ejercicios
11. Lógica para imprimir en pantalla del 1 al 100 Codificación
12. Encontrar los factores de un número
13. Encontrar los factores de un número
14. ¿cuál es el factor de un número?
15. Lógica para encontrar todos los factores de un número
16. Codificación
17. Calcular la tabla de multiplicar de un número
18. Lógica para calcular la tabla de multiplicar de un número
19. Codificación
20. Calcular el producto de los dígitos de un número
21. Lógica para encontrar el producto de los dígitos de un número paso a paso
22. Codificación
23. Bibliografía