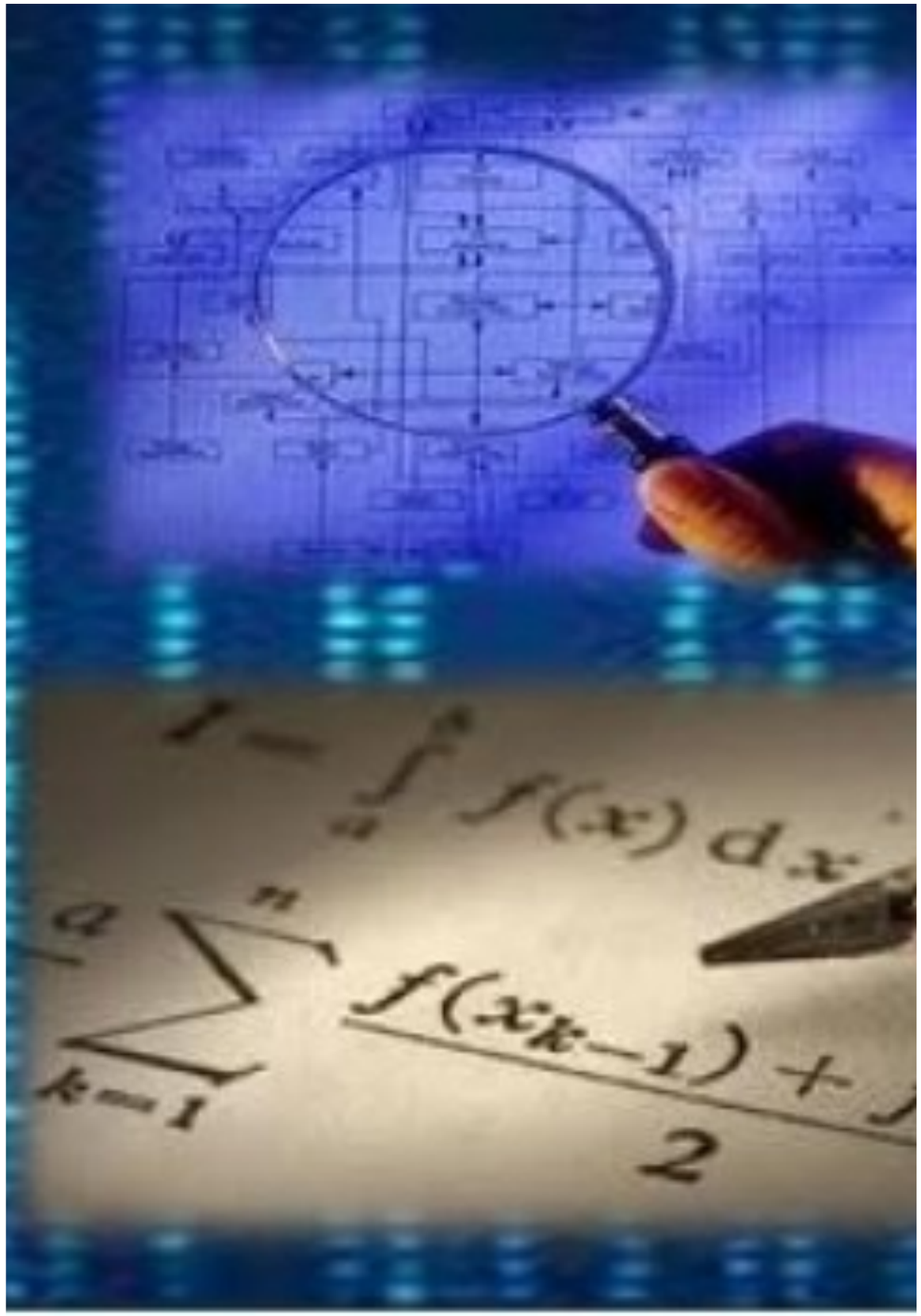
Programación básica

Alma González

Septiembre 2021



Algoritmos

- Método lógico para resolver un problema
- En programación se refiere a la secuencia de instrucciones a realizar por nuestro programa.
- Incluye cálculos, razonamiento y secuencia lógica a seguir.
- Debe indicar claramente la información de entrada, operaciones o instrucciones, y la información de salida.
- Podemos usar las variables como las definiremos en el código.
- Es una serie de pasos a seguir, como una receta de cocina. Puede ser tan detallado como queramos, lo importante es que tenga la información necesaria y suficiente para poder escribir el código.

Ejemplo conversión de temperaturas:
Celsius a Kelvin

Paso 1: Obtener la temperatura en Celsius
Paso 2: Hacer la operación para pasar de Celsius a Kelvin: TK=TC+273.15
Paso 3: Imprimir temperatura en Kelvin en pantalla

Formas de representar un algoritmo

- **Lenguaje natural: en palabras, sin símbolos o ecuaciones (o muy pocas)**
- **Pseudocódigo: indicando las instrucciones especificas.**

Ejemplo conversión de temperatura Celsius a Kelvin

Inicio:
- Declara las variables a usar
- Imprimir información de ayuda al usuario
- Leer temperatura en Celsius

Instrucciones/Operaciones:

Operaciones:
- Calcular temperatura en Kelvin sumando 273.15 a la temperatura en Celsius

Salida:
- Imprimir temperatura en Kelvin en la pantalla

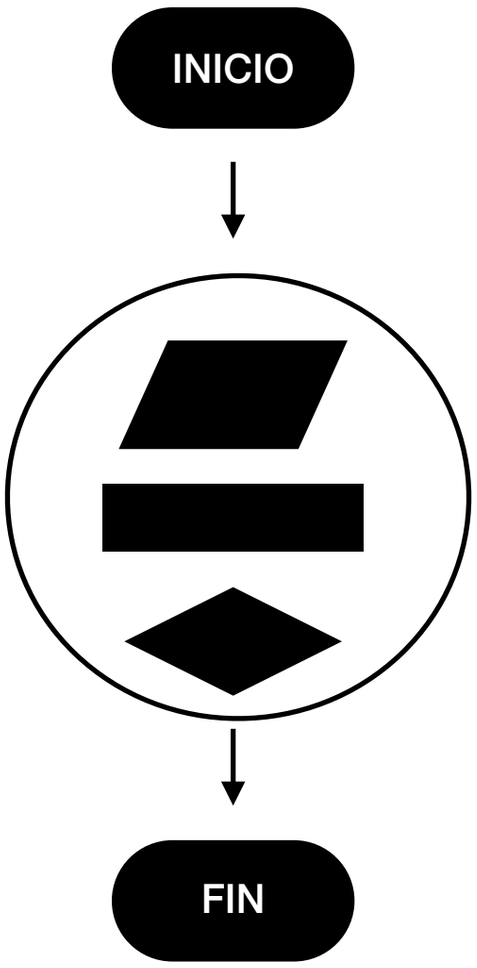
Inicio:
- declara float TC, TK
- imprimir “Dame la temperatura a convertir, en Celcius “
- Leer TC

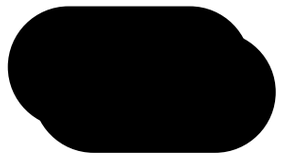
Operaciones:
- Calcular TK=TC+273.15

Salida :
- Imprimir “ La temperatura en Kelvin es: ”, TK

En ocasiones es útil primero identificar la salida, y las operaciones a realizar, para poder definir las variables que se necesitan y la secuencia lógica a seguir.

Diagrama de Flujo

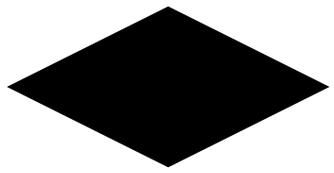




Indica inicio y fin
de funciones.

Indica instrucciones del
código, e.g printf, scanf,
declaración de variables,
etc.

Indica operaciones.



Indica opciones o
condicionales.

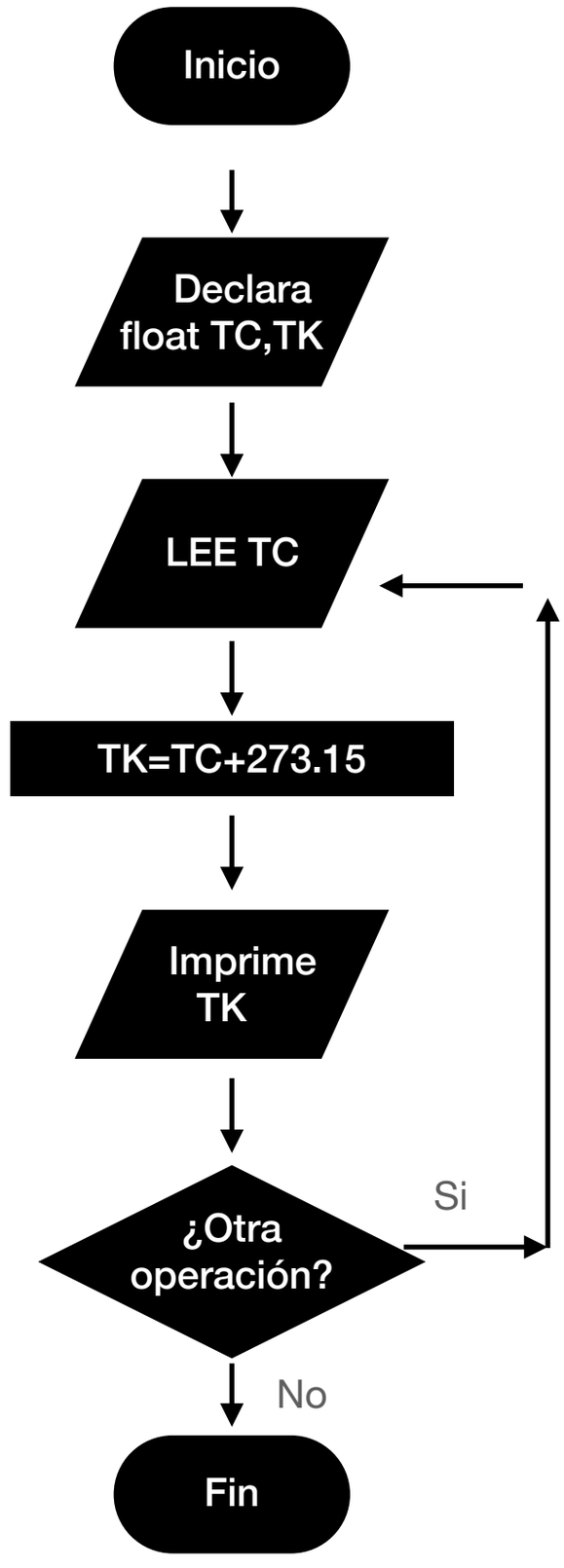
*Existen diferentes
notaciones, esta es una
de las más simples.

Ejemplo conversión de temperatura Celcius a Kelvin

Inicio:
- declara float TC, TK
- imprimir “Dame la temperatura a convertir, en Celcius “
- Leer TC

Operaciones:
- Calcular TK=TC+273.15

Salida :
- Imprimir “ La temperatura en Kelvin es: ”, TK

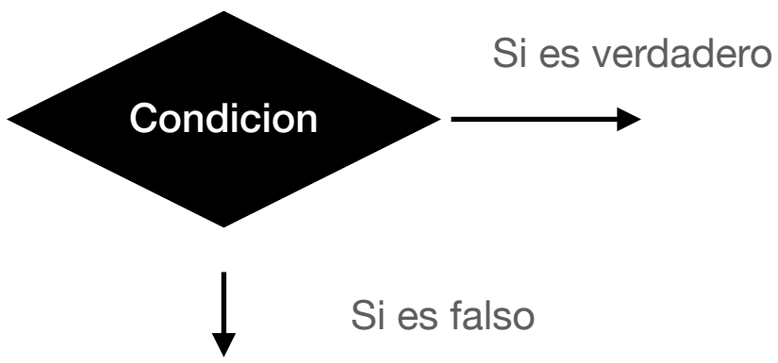


Condicional
IF…ELSE

- La instrucción if nos permite realizar una serie de instrucciones después de verificar que se cumpla una condición dada, si ésta no se cumple, entonces se realiza lo que se indique después de la instrucción else. La sintaxis es:

```
if(condicion){

Secuencia de instrucciones

} else{

Otra secuencia de instrucciones

}
```

Representación en diagrama de flujo



Operadores relaciónales

Operador relacional	Significado	Ejemplo
==	**Igual a**	**5==3 es 0 (falso)**
>	**mayor a**	**5>3 es 1 (verdadero)**
<	Menor a	**5<3 es 0 (falso)**
!=	No igual a	**5!=3 es 1 (verdadero)**
>=	Mayor o igual a	**5>=3 es 1 (verdadero)**
<=	Menor o igual a aa	**5<=3 es 0 (falso)**

Las condiciones se definen usando estos operadores.

Nota: dos numero float (double) nunca son iguales exactamente.

Operador lógico	Significado	Ejemplo
&&	“Y”	((c==5) && (d>5)) será 1 (verdadero) si ambas condiciones se cumplen
||	“O”	((c==5) ll (d>5)) será 1(verdadero) si una de las condiciones se cumple
!	“NO”	Suponiendo c = 5, la experesion !(c==5) será igual 0 (falsa)

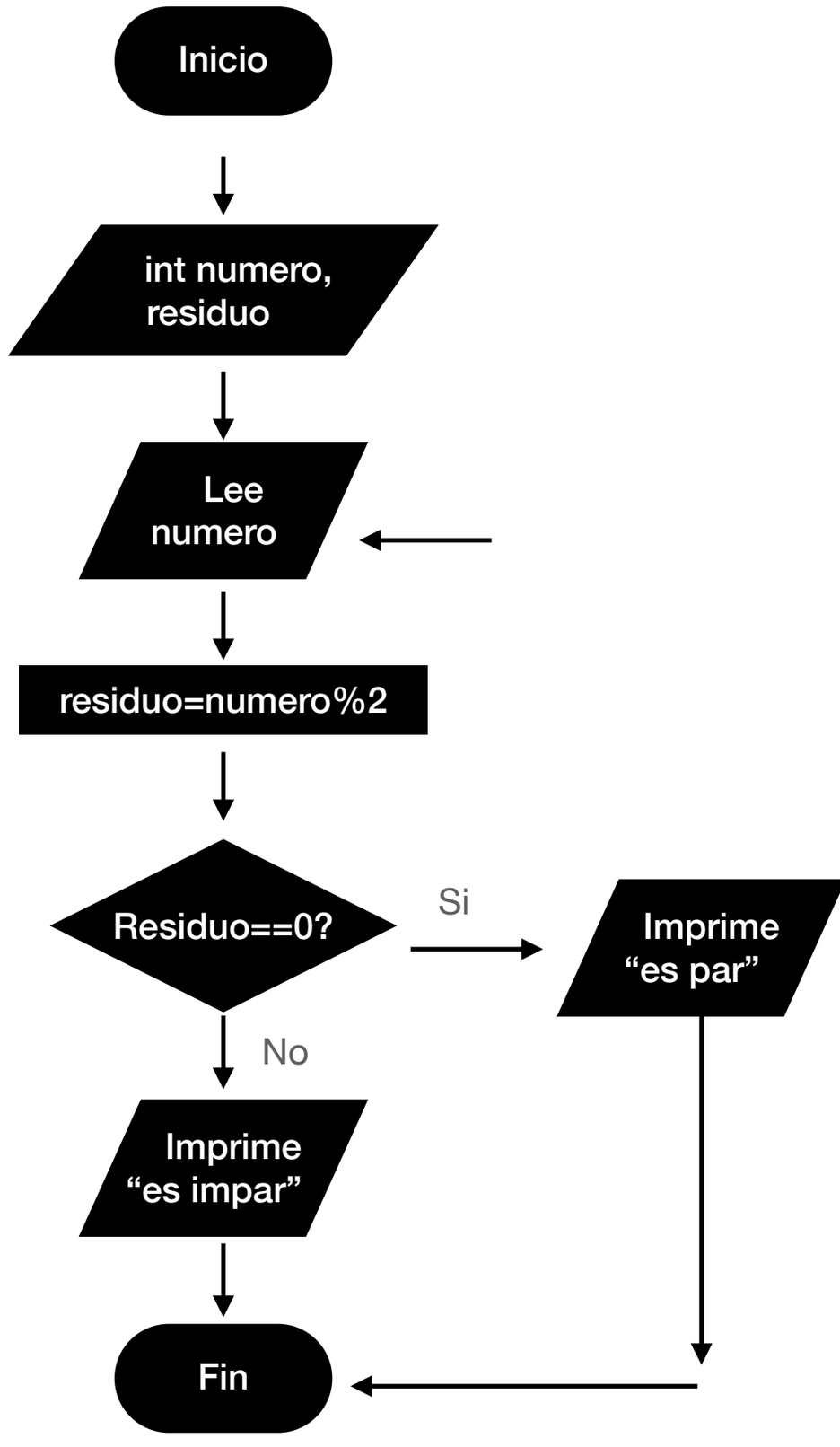
Ejemplo. Dado un numero imprimir si este es par o impar

Algoritmo

Inicio:
- declara: int numero, residuo
- imprimir “¿Qué numero quieres saber si es par o impar? “
- Leer numero

Operaciones:
- calcula el residuo de numero dividido ente 2: residuo=numero%2.

Fin:
- Si residuo es 0: imprime “numero es par”
- Si residuo no es 0: imprime “numero es impar”



Programa par_impar.c

```c
#include <stdio.h>

int main(){
    int numero,residuo;
    printf("Qué numero quieres saber si es par o impar\n");
    scanf("%i",&numero);
    residuo=numero%2;
    if(residuo==0){
        printf("%i es par\n",numero);
        }else{
            printf("%i es impar\n",numero);
            }
    return(0);
}
```

Escritura del programa

```
almagonzalez@Almas-Mac-Pro-2 Sep1 % gcc -Wall par_impar.c -o par_impar.o
```

Compilación

```
almagonzalez@Almas-Mac-Pro-2 Sep1 % ./par_impar.o
Qué numero quieres saber si es par o impar
653
653 es impar
almagonzalez@Almas-Mac-Pro-2 Sep1 %
```

Ejecución

Ejemplo. Dado un numero imprimir si este es par o impar

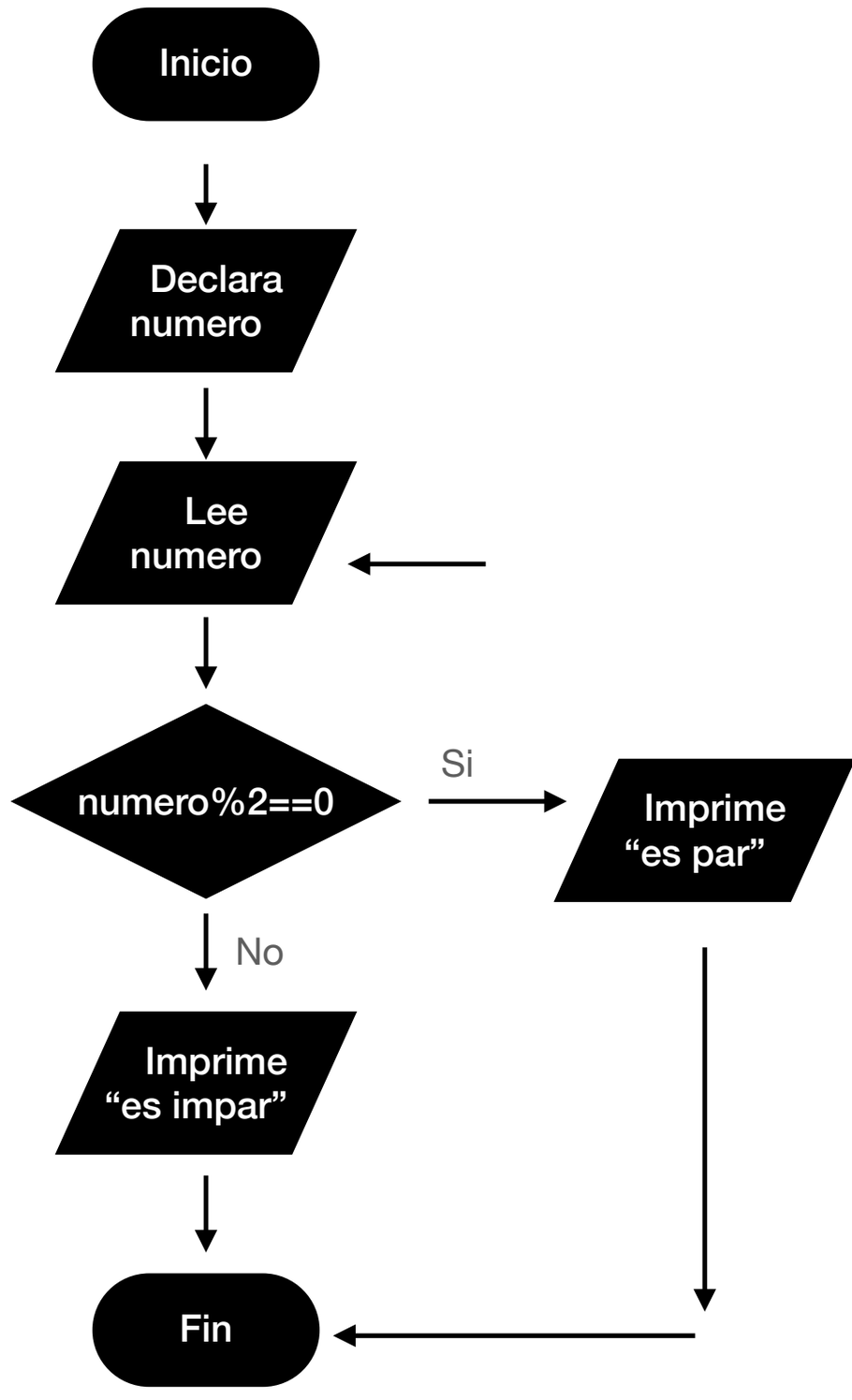
Una variante

Inicio:
- declara numero
- imprimir “¿Qué numero quieres saber si es par o impar? “
- Leer numero

Fin:
- Si residuo de numero entre 2 es 0: imprime “numero es par”
-Si residuo de numero entre 2 no es 0: imprime “numero es impar”

```c
#include <stdio.h>

int main(){
    int numero;
    printf("Qué numero quieres saber si es par o impar\n");
    scanf("%i",&numero);
    if(numero%2==0){
        printf("%i es par\n",numero);
        }else{
            printf("%i es impar\n",numero);
            }
    return(0);
}
```

```
almagonzalez@Almas-Mac-Pro-2 Sep1 % gcc -Wall par_impar_2.c -o par_impar_2.o
almagonzalez@Almas-Mac-Pro-2 Sep1 % ./par_impar_2.o
Qué numero quieres saber si es par o impar
653
653 es impar
```



El resultado es el mismo de antes, pero declaramos una variable menos y la operación modulo la hicimos como parte de la canción IF…ELSE

Diferentes formas de uso.

- En ocasiones solo queremos hacer una operación sencilla cuando se cumple una condición, en cuyo caso no es necesario poner la instrucción “else”. Ej. Si “a” es par le sumamos 1:

```c
#include <stdio.h>

int main(){
    int a;
    printf("introduce un numero\n");
    scanf("%i",&a);
    if(a%2==0) a=a+1;
    printf("el numero  mas uno es: %i\n",a);
}
```

Diferentes formas de uso.

- En ocasiones queremos verificar varias opciones, por lo que podemos poner condicionales dentro de otro condicional o bien condicionales dentro de la instrucción “else”.

```
if (condicion1) {
    if(condicion2){
    // Instrucciones a realizarse si la condicion1 y la condicion2 es verdadera
    }else{
    // Instrucciones a realizarse si la condicion2 no es verdadera.
    }
    }else if (condicion3) {
    // Instrucciones a realizarse si la condicion3 es verdadera}.
    }else{
    }
```

```
if (condicion1)
{
   // Instrucciones a realizarse si la condicion1 es verdadera
}
else if(condicion2)
{
   // Instrucciones a realizarse si la condicion2 es verdadera
}
else if (condicion3)
{
   // Instrucciones a realizarse si la condicion3 es verdadera}
.
.
else
{
   // Instrucciones a realizarse si todas las opciones anteriores son falsas
}
```

Ejemplo

Checar si un numero es igual, mayor o menor que otro.

```c
#include <stdio.h>
int main(){
    int numero1, numero2;
    printf("Introduzca dos numero enteros: ");
    scanf("%i %i", &numero1, &numero2);
    if(numero1==numero2){
        printf("Resultado: %d = %d\n",numero1,numero2);
        }else if(numero1> numero2){
        printf("Resultado: %d > %d\n", numero1, numero2);
        }else{
            printf("Resultado: %d < %d\n",numero1, numero2);
            }
    return(0);
    }
```

```
(char [20])"Resultado: %d < %d\n"
```

Ejemplo

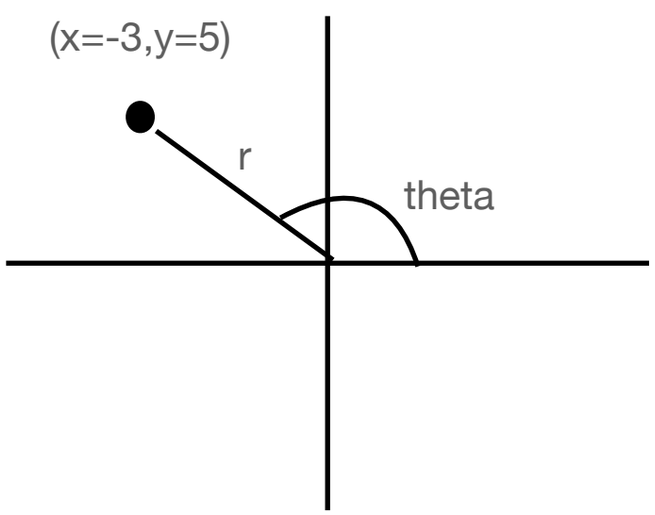
Verificar si una letra es vocal o consonante.

```c
#include <stdio.h>

int main()
{
    char c;
    int es_vocalminuscula, es_vocalmayuscula;
    printf("Introduce una letra: ");
    scanf("%c", &c);
    es_vocalminuscula = (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u');
    es_vocalmayuscula = (c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U');
    if (es_vocalmayuscula || es_vocalminuscula){
        printf(" %c es una vocal\n", c);
        } else{
        printf("%c es consonante\n", c);
        }
    return(0);
}
```

Ejercicio.

- 1) Realiza todos los ejemplos vistos en clase, y añade comentarios de que se está haciendo en cada linea. Realiza el pseudo-código y diagrama de flujo que corresponde a los ejemplos.

- 2) Escribe un programa que permita hacer la conversión de coordenadas de cartesianas a polares (en 2 dimensiones), asegúrate de tomar en cuenta en que cuadrante se encuentran las coordenadas iniciales. Antes de escribir el código, escribe el pseudo-código y el diagrama de flujo, incluye una captura de pantalla. Es importante que tu código se llame **cambio_coordenadas.c**



El programa deberá darte los valores de las coordenadas polares r,theta que corresponden a x,y