Ejercicios de Aplicación

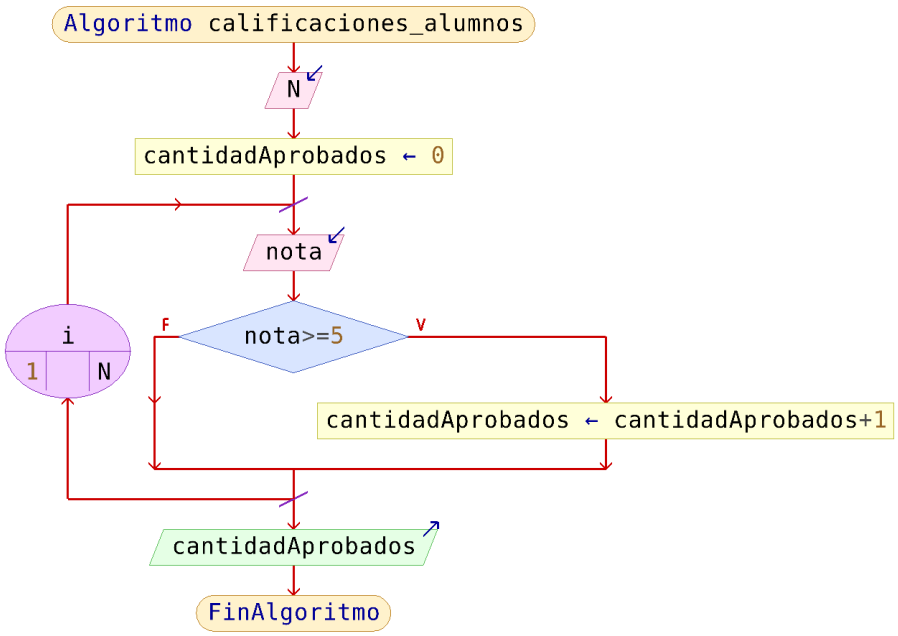
Programación Lógica

UNIVERSIDAD SIGLO 21 | MIEMBRO DE LA RED ILUMNO

1. Ejercicios de aplicación

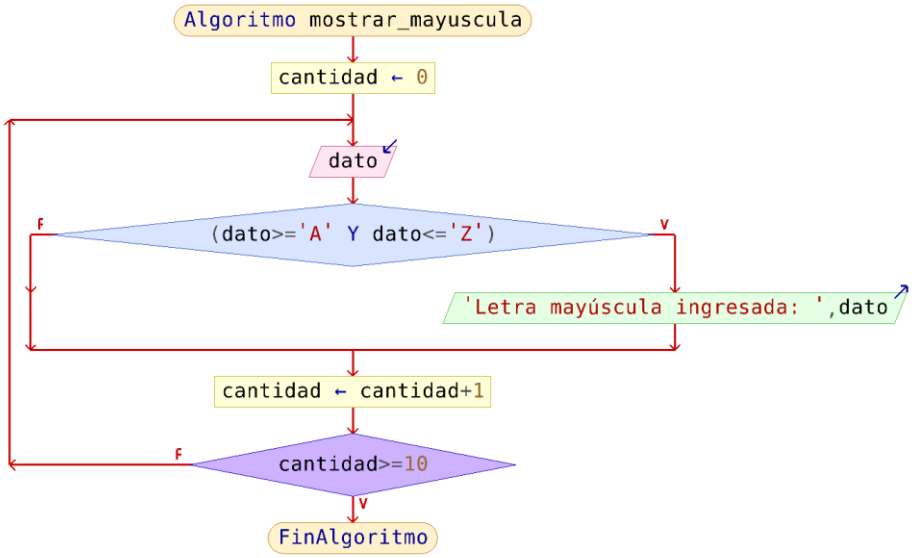
1) Implementar un algoritmo que permita ingresar *N* calificaciones de alumnos y mostrar la cantidad de alumnos aprobados.

```
algoritmo calificaciones_alumnos
var
    entero: N, nota, cantidadAprobados, i
inicio
    leer(N)
    cantidadAprobados ← 0
    desde i ← 1 hasta N hacer
        leer(nota)
        si nota >= 5 entonces
                cantidadAprobados ← cantidadAprobados + 1
        fin-si
    fin-desde
    mostrar(cantidadAprobados)
fin
```



2) Implementar un algoritmo que permita ingresar 10 caracteres por teclado y que muestre por pantalla únicamente las letras mayúsculas.

```
algoritmo mostrar_mayuscula
var
    entero: cantidad
    caracter: dato
inicio
    cantidad ← 0
    repetir
        leer(dato)
        si (dato >= 'A' y dato <= 'Z') entonces
                mostrar("Letra mayúscula ingresada: ", dato)
        fin-si
        cantidad ← cantidad + 1
    hasta-que cantidad >= 10
fin
```



3) Implementar un algoritmo que permita ingresar 10 caracteres por teclado. Si el carácter ingresado es una letra minúscula, entonces mostrarla por pantalla; si el carácter ingresado es una letra mayúscula, entonces convertirla a minúscula y mostrarla por pantalla.

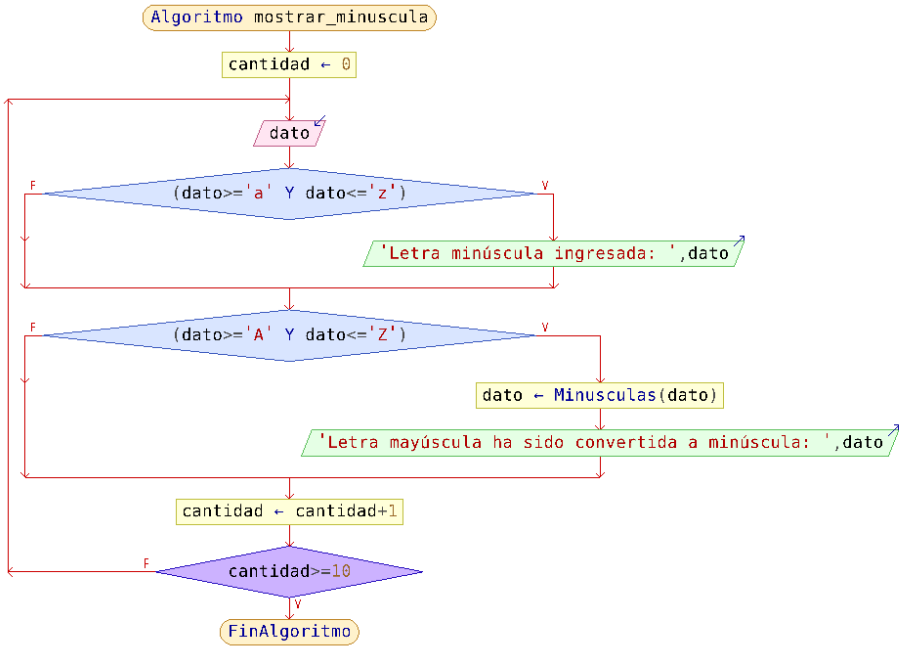
```
algoritmo mostrar_minuscula
var
    entero: cantidad
    caracter: dato
inicio
```

```
    cantidad ← 0
    repetir
        leer(dato)
        si (dato >= 'a' y dato <= 'z') entonces
                mostrar("Letra minúscula ingresada: ", dato)
        fin-si

        si (dato >= 'A' y dato <= 'Z') entonces
                dato ← dato + 32
                mostrar("Letra mayúscula ha sido convertida a minúscula: ",
dato)
        fin-si

        cantidad ← cantidad + 1
    hasta-que cantidad >= 10
fin
```

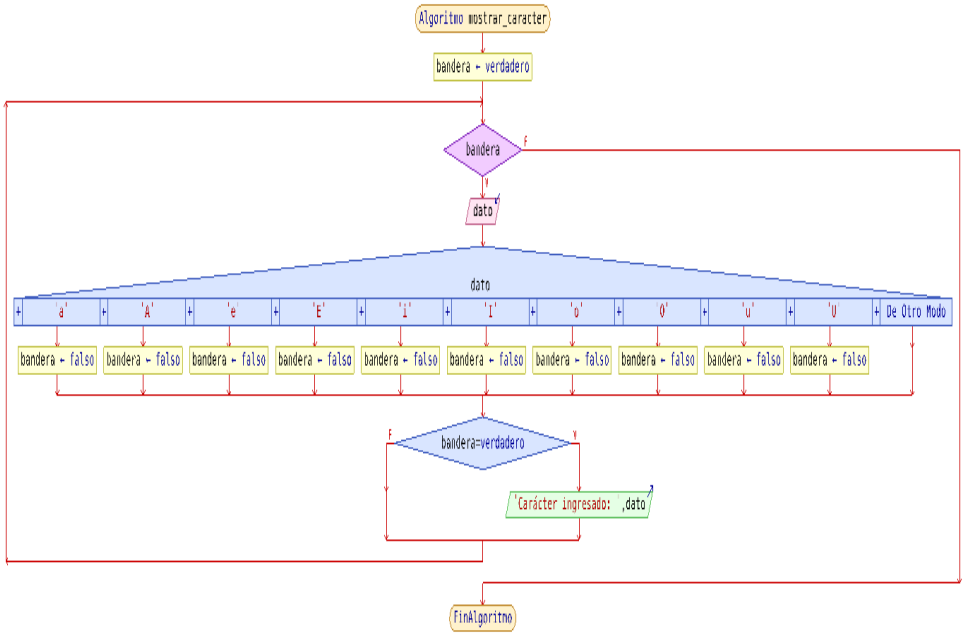
Nota: en algunos lenguajes se puede utilizar una función que permita la conversión de mayúscula a minúscula. La siguiente representación del algoritmo en diagrama de flujo del programa PSeInt utiliza la función *Minusculas*.



4) Implementar un algoritmo que permita ingresar caracteres por teclado y mostrarlos por pantalla. El programa debe finalizar cuando el usuario ingresa la primera letra vocal.

```
algoritmo mostrar_caracter
var
    logico: bandera
    caracter: dato
inicio
        bandera ← verdadero
        mientras bandera hacer
                leer(dato)
                segun-sea dato hacer
                        'a': bandera ← falso
                        'A': bandera ← falso
                        'e': bandera ← falso
                        'E': bandera ← falso
                        'i': bandera ← falso
                        'I': bandera ← falso
                        'o': bandera ← falso
                        'O': bandera ← falso
                        'u': bandera ← falso
                        'U': bandera ← falso
                fin-segun

                si bandera = verdadero entonces
                        mostrar("Carácter ingresado: ", dato)
                fin-si
        fin-mientras
fin
```



5) Implementar un algoritmo que permita ingresar tres números por teclado que representan las dimensiones de los lados de un triángulo, y que muestre por pantalla si el triángulo es equilátero, isósceles o escaleno. El algoritmo debe finalizar cuando el usuario ingresa la palabra *salir*.

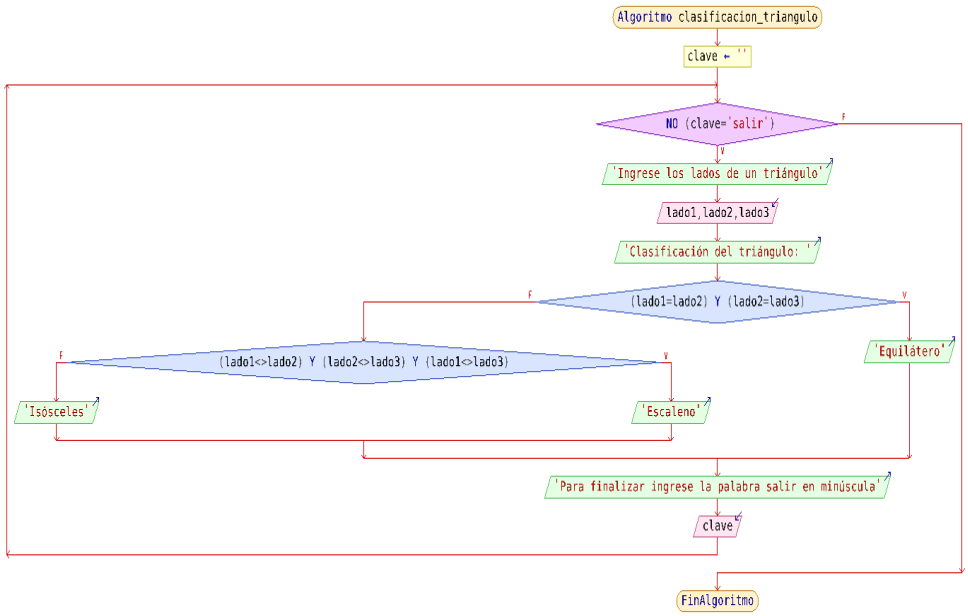
```
algoritmo clasificacion_triangulo
var
    entero: lado1, lado2, lado3
    cadena: clave
inicio
    clave ← “”
    mientras no (clave = “salir”) hacer
        mostrar("Ingrese los lados de un triángulo")
        leer(lado1, lado2, lado3)

        mostrar("Clasificación del triángulo: ")
        si (lado1 = lado2) y (lado2 = lado3) entonces
            mostrar("Equilátero")
        si-no
            si (lado1 <> lado2) y (lado2 <> lado3) y (lado1 <> lado3)
entonces
                mostrar("Escaleno")
            si-no
                mostrar("Isósceles")
```

```
            fin-si
        fin-si

        mostrar("Para finalizar ingrese la palabra salir en minúscula")
        leer(clave)
    fin-mientras

fin
```



6) Implementar un algoritmo que permita ingresar un número entero positivo por teclado y que muestre por pantalla su factorial.

Fórmula del factorial de *N*: *N! = N * (N-1) * (N-2) * … * 2 * 1*

```
algoritmo factorial_repetir
var
    entero: num, factorial, contador
    cadena: clave
inicio
    mostrar("Ingrese un número entero positivo para calcular su factorial")
    leer(num)
    si num = 0 o num = 1 entonces
```

```
        factorial ← 1
    si-no
        factorial ← num
        contador ← num -1
        repetir
            factorial ← factorial * contador
            contador ← contador - 1
        hasta-que contador <= 1
    fin-si
    mostrar("El factorial de ", num, " es ", factorial)
fin
```