## UPDS - Universidad Privada Domingo Savio

**Fecha de entrega: 26-08-2025**

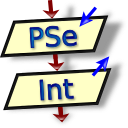
**PRACTICA 3**

**RESOLUCIÓN DE ALGORITMOS UTILIZANDO**

**TIPOS CADENAS**

**Grupo N°4**

**Integrantes**

1. Harold Jimmy Arias Pérez
2. Juan Carlos Hidalgo Sosa
3. Juan Jose Rivas Ramirez
4. Luciano Franco Vargas Rojas
5. Nestor Fabian Loras Vega (Líder)

Cochabamba-Bolivia

## FECHA DE ENTREGA:26-08-2025

**PRACTICA 3**

**CADENA DE CARACTERES**

1.- Un algoritmo que te pida tu nombre y lo muestre en pantalla separando cada letra de la siguiente con un carácter (-). Por ejemplo, si tu nombre es "Juan", debería aparecer en pantalla "J-u-a-n".

## E – ENTRADA

Nombre del usuario Tipo: cadena

## P – PROCESO

1. Leer el nombre ingresado por el usuario.
2. Recorrer cada letra del nombre.
3. Ir concatenando cada letra seguida de un guion, excepto la última.
4. Construir la cadena final con el formato deseado.

## S – SALIDA

Nombre del usuario con cada letra separada por un guion.

# Insertando imagen...**DIAGRAMA DE FLUJO**

2.-Realizar un algoritmo que ingrese una frase por teclado e indique el número de vocales en dicha frase(a,e,i,o,u).

## E – ENTRADA

Una frase escrita por el usuario (cadena de texto).

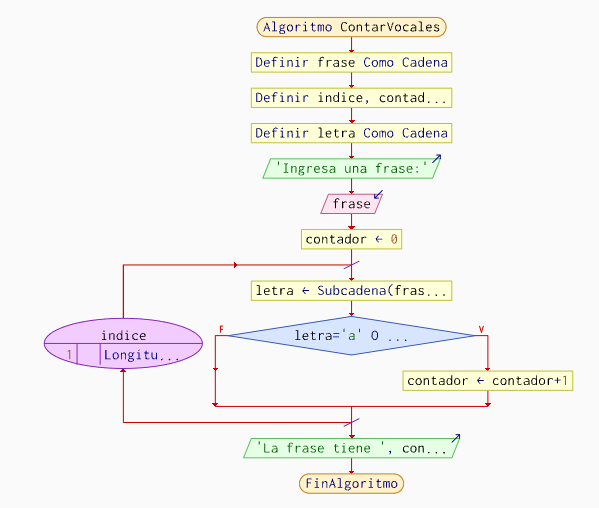
## P – PROCESO

1. Pedir una frase.
2. Inicializar un contador en 0.
3. Recorrer la frase carácter por carácter con un índice.
4. Extraer la letra actual usando Subcadena (frase, indice, indice).
5. Verificar si la letra es una vocal (a, e, i, o, u o A, E, I, O, U).
6. Si es vocal, aumentar el contador en 1.
7. Continuar hasta recorrer toda la frase.

## S – SALIDA

Mostrar la cantidad total de vocales que hay en la frase.

# **DIAGRAMA DE FLUJO**



3.- Realizar un algoritmo que verifica si una frase ingresada es palíndroma. Una frase palíndroma es aquella que se lee igual de izquierda a derecha y viceversa.

**E – ENTRADA**

Una frase ingresada

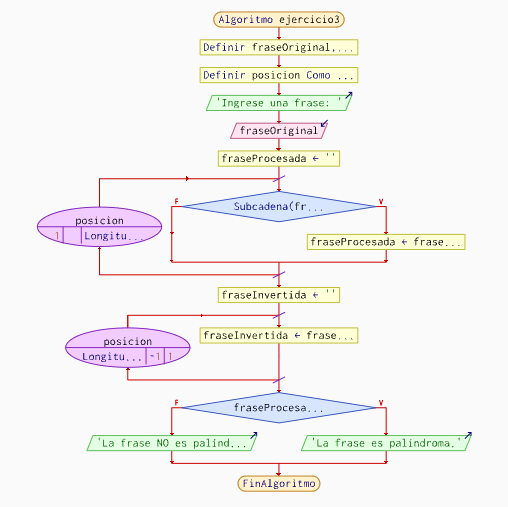
**P – PROCESO**

1. Guardar la frase original en una variable.
2. Eliminar los espacios en blanco y convertir toda la frase a mayúsculas
3. Invertir el orden de los caracteres de la frase procesada.
4. Comparar la frase procesada con su versión invertida:
   * Si son iguales → la frase es palíndroma.
   * Si son diferentes → la frase **no** es palíndroma.

**S – SALIDA**

Mostrar un mensaje indicando si la frase ingresada es palíndroma o no es palíndroma.

# **DIAGRAMA DE FLUJO**



4.- Escribir un algoritmo para contar el número de ocurrencias de cada una de las palabras 'a', 'an' y 'and' en las diferentes líneas de texto.

**E – ENTRADA**

Leer varias líneas de texto ingresadas por el usuario (hasta que introduzca una línea vacía o un fin de texto).

Palabras clave a contar: "a", "an", "and".

**P – PROCESO**

1. Inicializar contadores en cero: contA, contAn, contAnd.
2. Mientras la línea no esté vacía:
3. Convertir la línea a minúsculas para uniformidad.
4. Recorrer la línea carácter por carácter para separar las palabras (usando espacios como delimitador).
5. Cada vez que se forma una palabra:
   1. Si la palabra = "a" → incrementar contA.
   2. Si la palabra = "an" → incrementar contAn.
   3. Si la palabra = "and" → incrementar contAnd.
6. Procesar también la última palabra de la línea.
7. Pedir al usuario otra línea de texto.

**S – SALIDA**

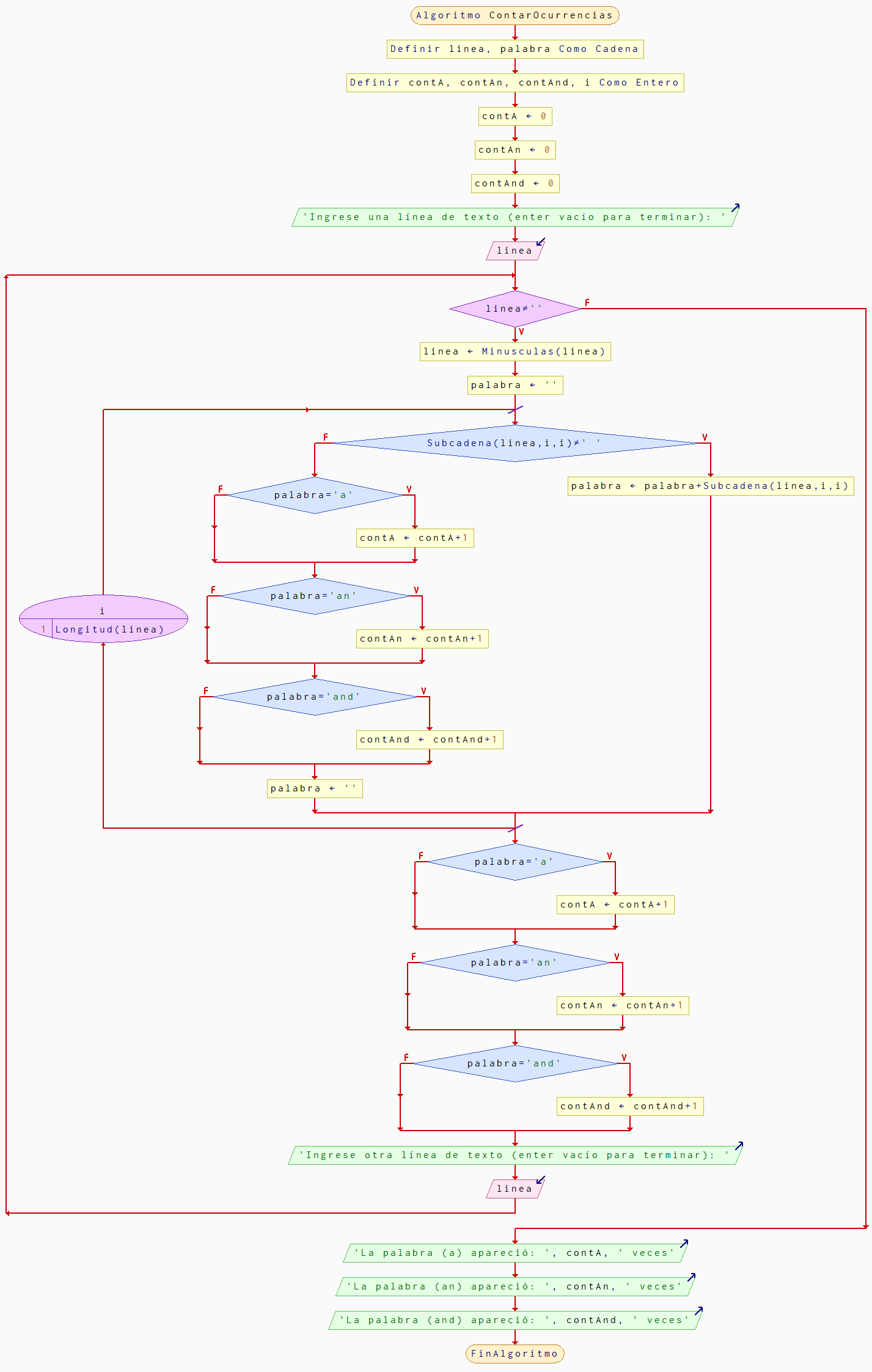
Mostrar el número de ocurrencias de cada palabra:

"a" apareció: contA veces

"an" apareció: contAn veces

"and" apareció: contAnd veces

# **DIAGRAMA DE FLUJO**



5.- Realizar un algoritmo que elimine todas las ocurrencias de cada carácter en una cadena dada partir de otra cadena dada. Las dos cadenas son:

* CADENA1: es la cadena donde deben eliminarse caracteres.
* LISTA: es la cadena que proporciona los caracteres que deben eliminarse. CADENA = 'EL EZNZZXTX'

LISTA = 'XZ'

la cadena pedida es 'EL ENT'. R Análisis

## E – ENTRADA

CAD: la cadena que contiene texto. "EL EZNZZXTX" TIPO CADENA

## LIST:la cadena con los caracteres que se deben eliminar. "XZ" TIPO CADENA

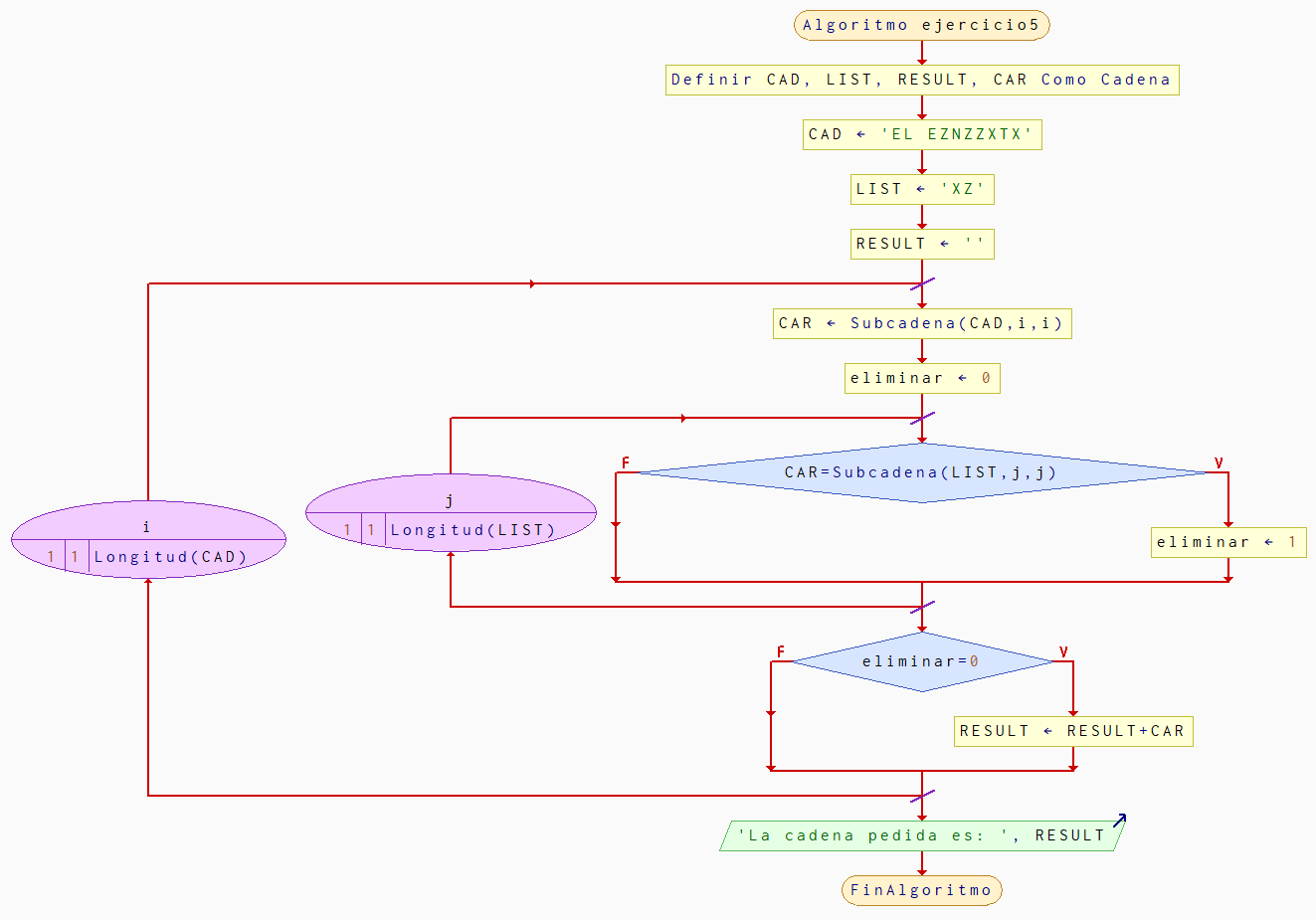
## P – PROCESO

1. Crear una variable llamada RESULT y dejarla vacía. Examinar uno por uno los caracteres de la cadena CAD. Para cada carácter tomado:
2. Comprobar si se encuentra en la cadena LIST. Si aparece en LIST, entonces no se añade.
3. Si no aparece en LIST, entonces se concatena a RESULT.
4. Cuando termine el recorrido completo de CAD, la variable RESULT contendrá la cadena final deseada.

## S – SALIDA

Escribir "La cadena pedida es: " RESULT Diagrama de Flujo

**DIAGRAMA DE FLUJO**



6.- Crear un algoritmo que te pida tu nombre y lo muestre en pantalla como un triángulo creciente. Por ejemplo, si tu nombre es "Juan", debería aparecer en pantalla:

J

Ju Jua Juan

**Análisis**

## E – ENTRADA

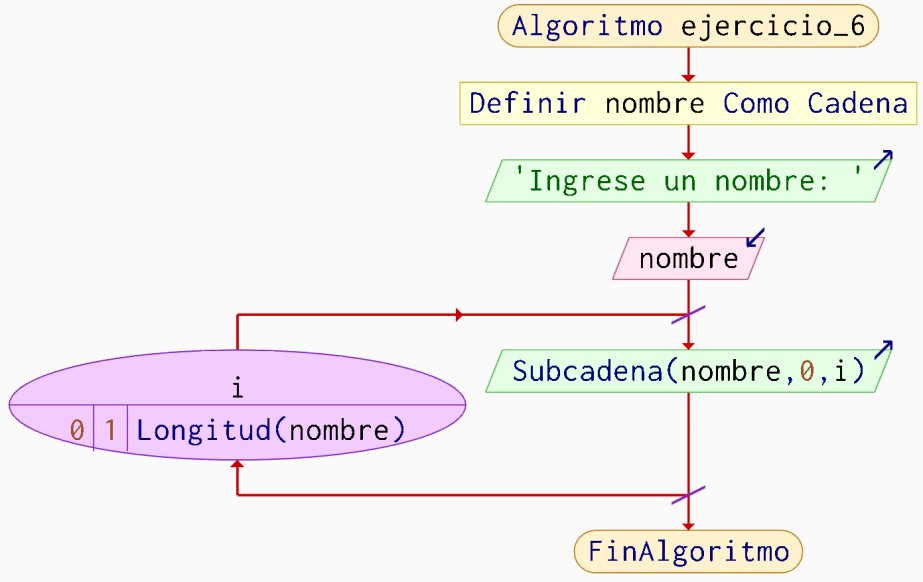
## nombre: el nombre de una persona. Tipo: cadena

## P – PROCESO

1. Iniciar un bucle Para con índice i desde 0 hasta la longitud de la cadena.
2. En cada iteración, mostrar una subcadena de nombre desde 0 hasta i

## S – SALIDA

La subcadena que se muestra en cada iteración del bucle Para. Tipo: cadena Diagrama de Flujo:

**DIAGRAMA DE FLUJO**

7.-Realizar un algoritmo cuya entrada sea una cadena S y un factor de multiplicación N, cuya función sea generar la cadena dada N veces. Por ejemplo:

S=Hey y N=3 Hey hey hey

# Análisis

## E – ENTRADA

S: una cadena. Tipo: cadena

## N: factor de multiplicación. Tipo: entero

## P – PROCESO

1. Iniciar un bucle Para con índice i desde 1 hasta N con paso 1.

## En cada iteración del Para, concatenar S\_total con S

## S – SALIDA

S\_total: la cadena total resultante después de las concatenaciones. Tipo: cadena. Diagrama de flujo:

**DIAGRAMA DE FLUJO**

