

PROBLEMA

1. Realizar un algoritmo que permita multiplicar (n) números introducidos por teclado.
Si N = 2 entonces solicitar número num y multiplicar entre sí.

ANALISIS

SOLUCION

Leer n

Usar ciclo for para multiplicar n veces

Leer x

Multiplica <- mult*x

1 ANALIZAR EL PROBLEMA

Especificar resultado: Multiplicar

Identificar datos a leer: n

Determinar restricciones: usar datos enteros

Definir proceso: Multiplica <- mult*x

2 DISEÑAR DIAGRAMA

ALGORITHM PRÁCTICA GENERAL DE SIS - III
PORFIRIO ELIAS QUISPE QUISPE



PRÁCTICA GENERAL DE SIS - III

PORFIRIO ELIAS QUISPE QUISPE

SOURCE IN PYTHON

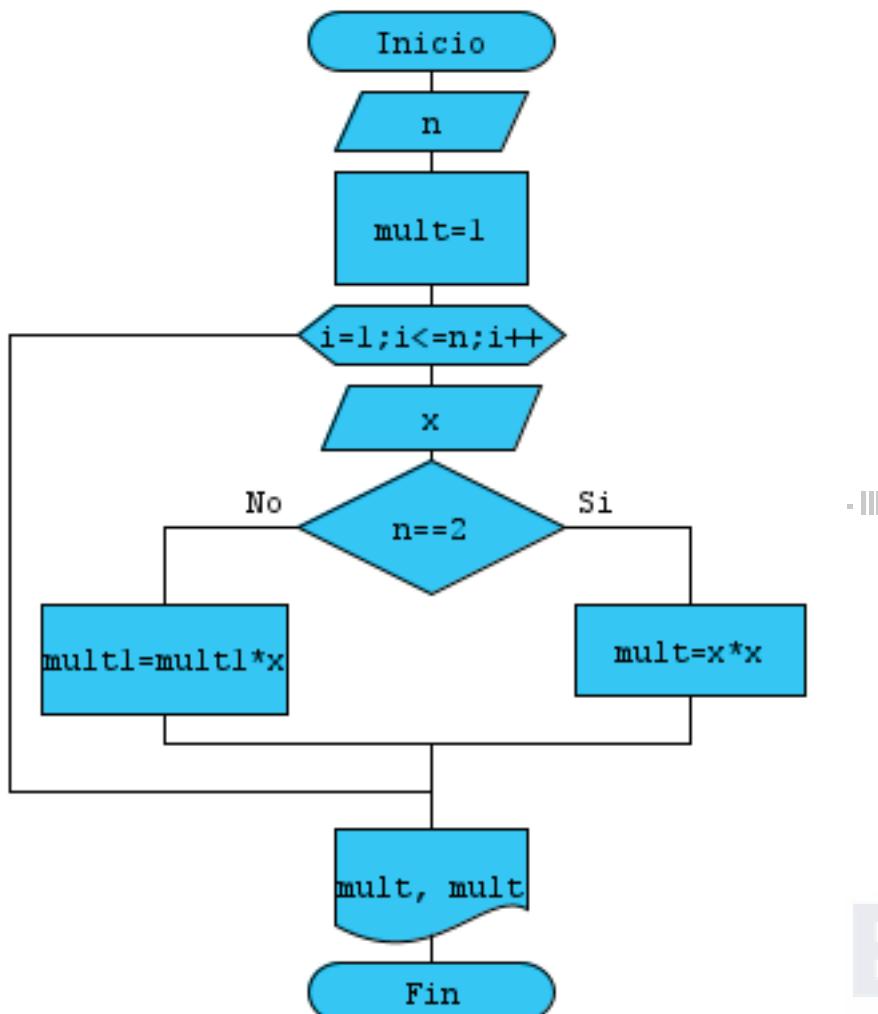
```
n=int(input("Ingrese un numero n"))
for i in range(1,n):
```

```
    x = int(input("Ingrese un valor"))
    if x==2:
        mult = x * x
    else:
```

DESK CHECK

Prueba de escritorio (PDE's)

M Inicio				
n	mult	mult1	i	x
3	1	1	1	3
		3	2	3
		9	3	3
		27	4	



PRÁCTICA GENERAL DE SIS - III

PORFIRIO ELIAS QUISPE QUISPE



ERÍA
DE SISTEMAS
UNIVERSIDAD PÚBLICA DE EL ALTO

PRÁCTICA GENERAL
PORFIRIO ELIAS QUISPE QUISPE

PROBLEMA

2. Mostrar un conteo descendente del 20 al 0 de 2 en 2

ANALISIS

SOLUCION

i=20

Usar ciclo for para generar n veces

1 ANALIZAR EL PROBLEMA

Especificar resultado: mostrar en forma descendente 20 -18 -16 - 16.....0

Identificar datos a leer: n

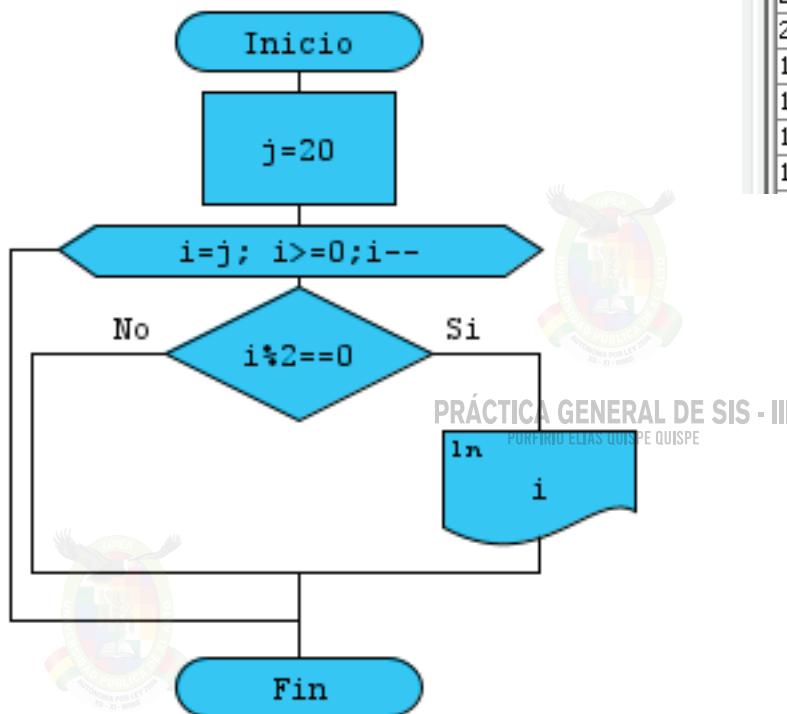
Determinar restricciones: usar datos enteros

Definir proceso: --2

2 DISEÑAR DIAGRAMA

ALGORITHM

A GENERAL DE SIS - III
PURFIRIO ELIAS QUISPE QUISPE



PRÁCTICA GENERAL DE SIS - III
PURFIRIO ELIAS QUISPE QUISPE

SOURCE IN PYTHON

```
# usaremos la funcion reversed en python
```

```
print("conteo descendente de 20 a 0")
for i in reversed(range(20)):
    print(i)
```

DESK CHECK

Prueba de escritorio (PDE's)

Inicio
i
20
20
19
18
17
16



INGENIERÍA
DE SISTEMAS
UNIVERSIDAD PÚBLICA DE EL ALTO



PROBLEMA

3. Imprimir la suma de los números ingresados por pantalla múltiplos de 3 mientras NO se introduzca como valor 0.

ANALISIS

SOLUCION

Usar ciclo while hasta que se ingrese un cero y salga del proceso

1 ANALIZAR EL PROBLEMA

Especificar resultado: suma

$$3*5=15, 3*2=6$$

Identificar datos a leer: x

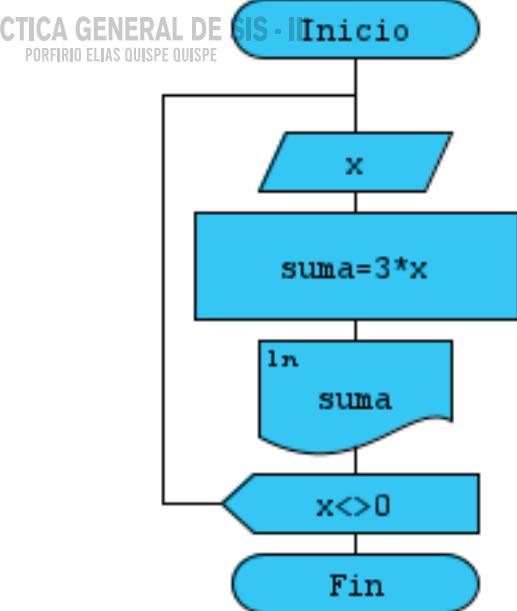
Determinar restricciones: usar datos enteros

Definir proceso: si $x==0$ salir



PRÁCTICA GENERAL DE SIS - III
PORFIRIO ELIAS QUISPE QUISPE

ALGORITHM



SOURCE IN PYTHON

```
# usaremos la funcion reversed en python
print("conteo descendente de 20 a 0")
for i in reversed(range(20)):
    print(i)
```

PRÁCTICA GENERAL DE SIS - III
PORFIRIO ELIAS QUISPE QUISPE

DESK CHECK

Prueba de escritorio (PDE's)

M	Inicio
X	suma
2	6
2	6
2	6
3	9
5	15

PRÁCTICA GENERAL DE SIS - III
PORFIRIO ELIAS QUISPE QUISPE



PRÁCTICA GENERAL DE SIS - III
PORFIRIO ELIAS QUISPE QUISPE





PROBLEMA

4. Dado una contraseña, validar hasta que esta sea la verdadera teniendo 3 intentos como mínimo. Pasados estos intentos que salga un mensaje "Terminaron sus intentos, cuenta bloqueada".

SOLUCION

contar cuantos ingreso el password al tercer intento cerrar la sesion

1 ANALIZAR EL PROBLEMA

Especificar resultado:

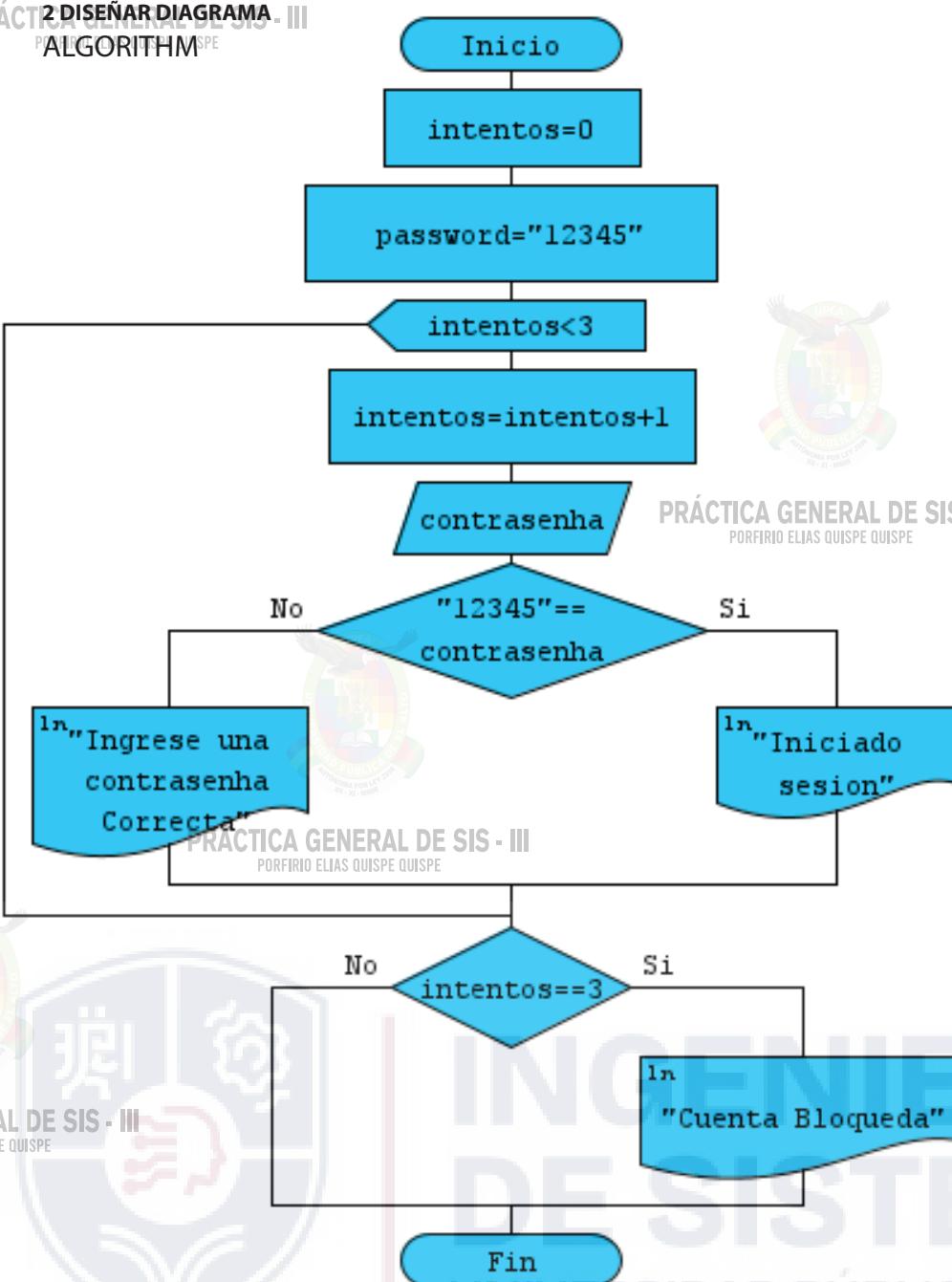
Identificar datos a leer: password

Determinar re

Definir proceso: si password <>

password correcto salir de tr

2 DISEÑAR DIAGR



SOURCE IN PYTHON

```
intentos=0
password = "contraseña"
while intentos<3:
    intentos = intentos+1
    contrashena = str(input("Ingrese su
contraseña"))
    if contrashena==password:
        print("Iniciado sesión")
    else:
        print("Ingrese una contraseña Correcta")
    if intentos==3:
        print("cuenta bloqueada")
    else:
        print("Ingrese una contraseña Correcta")

print("Tr")
```

DESK CHECK

```
Consola
contraseña = 123
Ingrese una, contraseña, Correcta
contraseña = 123
Ingrese una, contraseña, Correcta
contraseña = 12345
Iniciado ,sesion
Cuenta Bloqueda
```

S-||| Prueba de escritorio (PDE's)

M Inicio |

intentos	password	contraseña
0	12345	123
1		123
2		12345
3		



PROBLEMA

5. Dado un numero N, mostrar si el número es primo, o no.como valor 0.

ANALISIS

$$\begin{array}{ll} n \text{ Mod } 1 == 0 & n \text{ Mod } 2 == 0 \\ n \text{ Mod } 3 == 0 & n \text{ Mod } 4 == 0 \\ n \text{ Mod } 5 == 0 \end{array}$$

SOLUCION

n=5

Usar ciclo for para verificar un numero primo

1 ANALIZAR EL PROBLEMA

Especificar resultado: mostrar en forma dun mensaje si es primo o no

Identificar datos a leer: n

Determinar restricciones: usar datos enteros

Definir proceso: si $n \% i == 0$
contar 2 veces si es dos

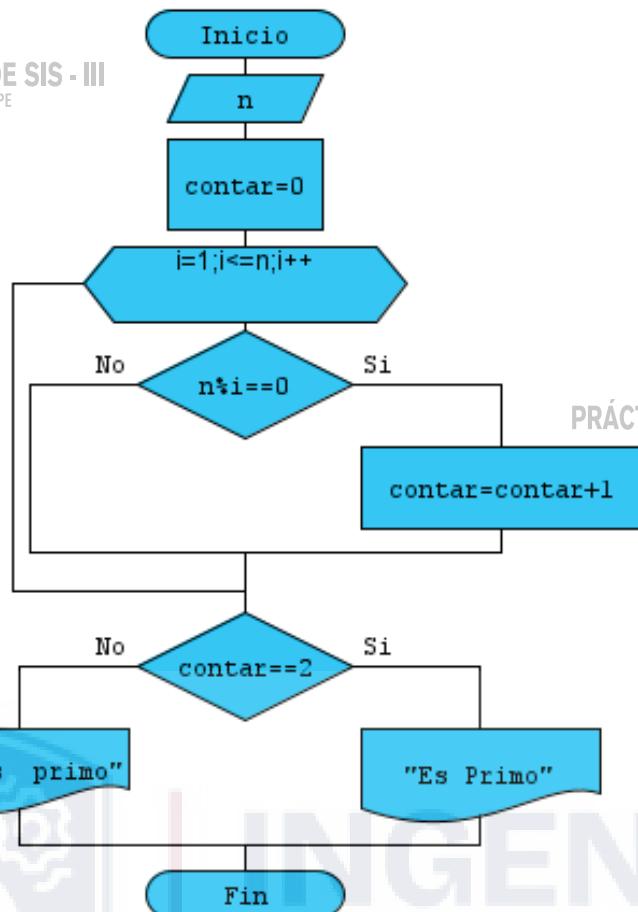
si contar== 2 es primo y por no no es primo



PRÁCTICA GENERAL DE SIS - III

PORFIRIO ELIAS QUISPE QUISPE

ALGORITHM



SOURCE IN PYTHON

```

n = int(input("Ingrese un numero n"))
contar=1
for i in range(1,n):
    if n%i==0:
        contar = contar + 1
if contar==2:
    print("es Primo")
else:
    print("No es Primo")
    
```

PRÁCTICA GENERAL DE SIS - III

PORFIRIO ELIAS QUISPE QUISPE

DESK CHECK

n = 3
Es Primo

Prueba de escritorio (PDE's)

M Inicio |

n	contar	i
3	0	1
	1	2
	2	3
		4

PRÁCTICA GENERAL DE SIS - III

PORFIRIO ELIAS QUISPE QUISPE

PRÁCTICA GENERAL DE SIS - III

PORFIRIO ELIAS QUISPE QUISPE





PROBLEMA

6. Dado un numero n, mostrar por pantalla su tabla de multiplicar.

ANALISIS

$$1 * 1 = 1$$

$$1 * 2 = 2$$

$$1 * 3 = 3$$

$$1 * 4 = 4$$

$$1 * 4 = 5$$

SOLUCION

n=1 si n=1 debe generar la tabla 1

Usar ciclo for para generar numero pares

1 ANALIZAR EL PROBLEMA

Especificar resultado: mostrar en forma de tabla de multiplicar un numero

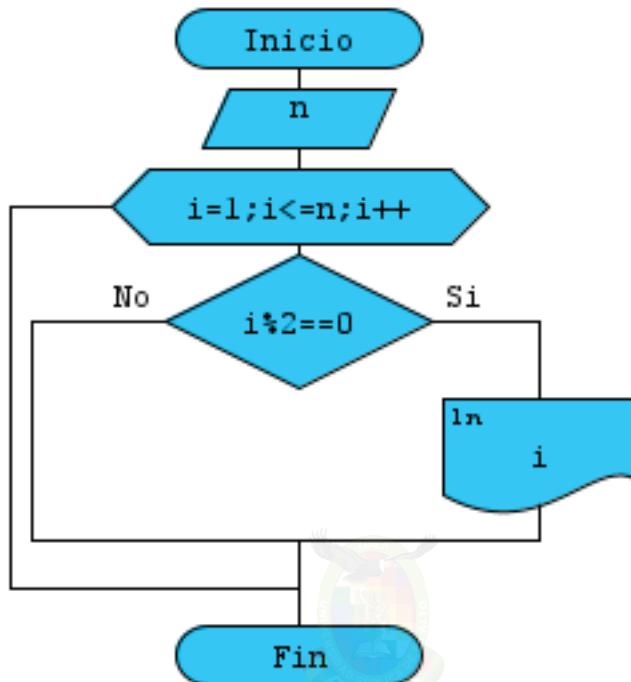
Identificar datos a leer: n

Determinar restricciones: usar datos enteros

Definir proceso: si $i \% 2 == 0$

PRÁCTICA GENERAL DE SIS - III

PORFIRIO ELIAS QUISPE QUISPE



PRÁCTICA GENERAL DE SIS - III
PORFIRIO ELIAS QUISPE QUISPE

SOURCE IN PYTHON

```

n = int(input("Ingrese un numero n"))
for i in range(1,n):
    print(i, "x", n, "=", i*n)
    
```

DESK CHECK

```

n = 7
2
4
6
    
```

Prueba de escritorio (P)

M Inicio

n	i
7	1
	2
	3
	4
	5
	6
	7
	8



PRÁCTICA GENERAL DE SIS - III

PORFIRIO ELIAS QUISPE QUISPE





PROBLEMA

7. Dado un numero n, imprimir los N números pares, ejemplo, si n = 7, => 2,4,6

ANALYSIS

i Mod 2==0

SOLUCION

n=7

Usar ciclo for para generar numero pares

1 ANALIZAR EL PROBLEMA

Especificar resultado: mostrar en forma
si $n = 7, 246$

Identificar datos a leer: n

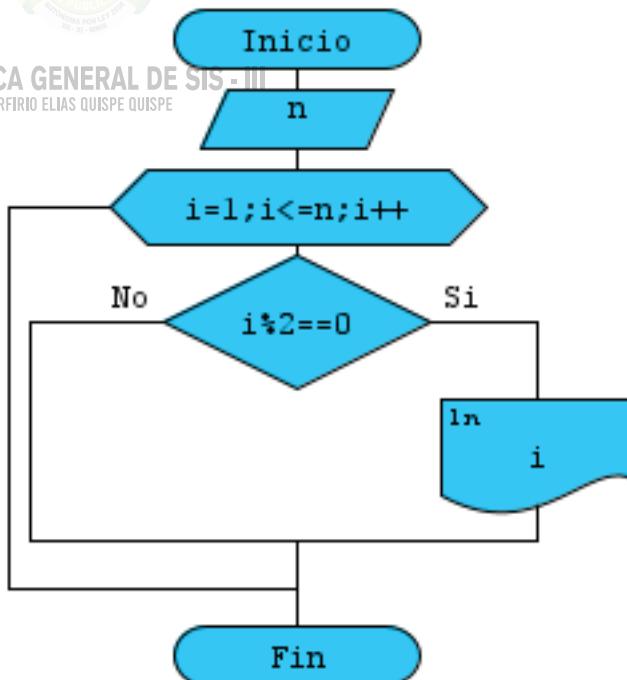
Determinar restricciones: usar datos enteros

Definir proceso: si $i \% 2 == 0$

PRÁCTICA GENERAL DE SIS - III

PORFIRIO ELIAS QUISPE QUISPE

ALGORITHM



SOURCE IN PYTHON

```
2,4,6  
n = int(input("Ingrese un numero n"))  
contar=1  
for i in range(1,n):  
    if n%i==0:  
        contar = contar + 1  
if contar==2:  
    print("es Primo")  
else:  
    print("No es Primo")
```

DESK CHECK

n	i
7	1
	2
	3
	4
	5
	6
	7
	8



PROBLEMA

8. Ingresar un numero N y solicitar N números pares e impares hay en el lote de números ing

ANALISIS

SOLUCION

n=3

Usar ciclo while para leer n numero de lotes
contar pares impares

nn = n mod 10

n = n div 10

if nn mod2==0

contar pares

caso contrario

contar impares

1 ANALIZAR EL PROBLEMA

Especificar resultado: mostrar en forma

si n= 7, 12, 4, 6, 10, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54, 57, 60, 63, 66, 69, 72, 75, 78, 81, 84, 87, 90, 93, 96, 99

Identificar datos a leer: n

Determinar restricciones: usar datos enteros

Definir proceso: si $i \% 2 == 0$

SOURCE IN PYTHON

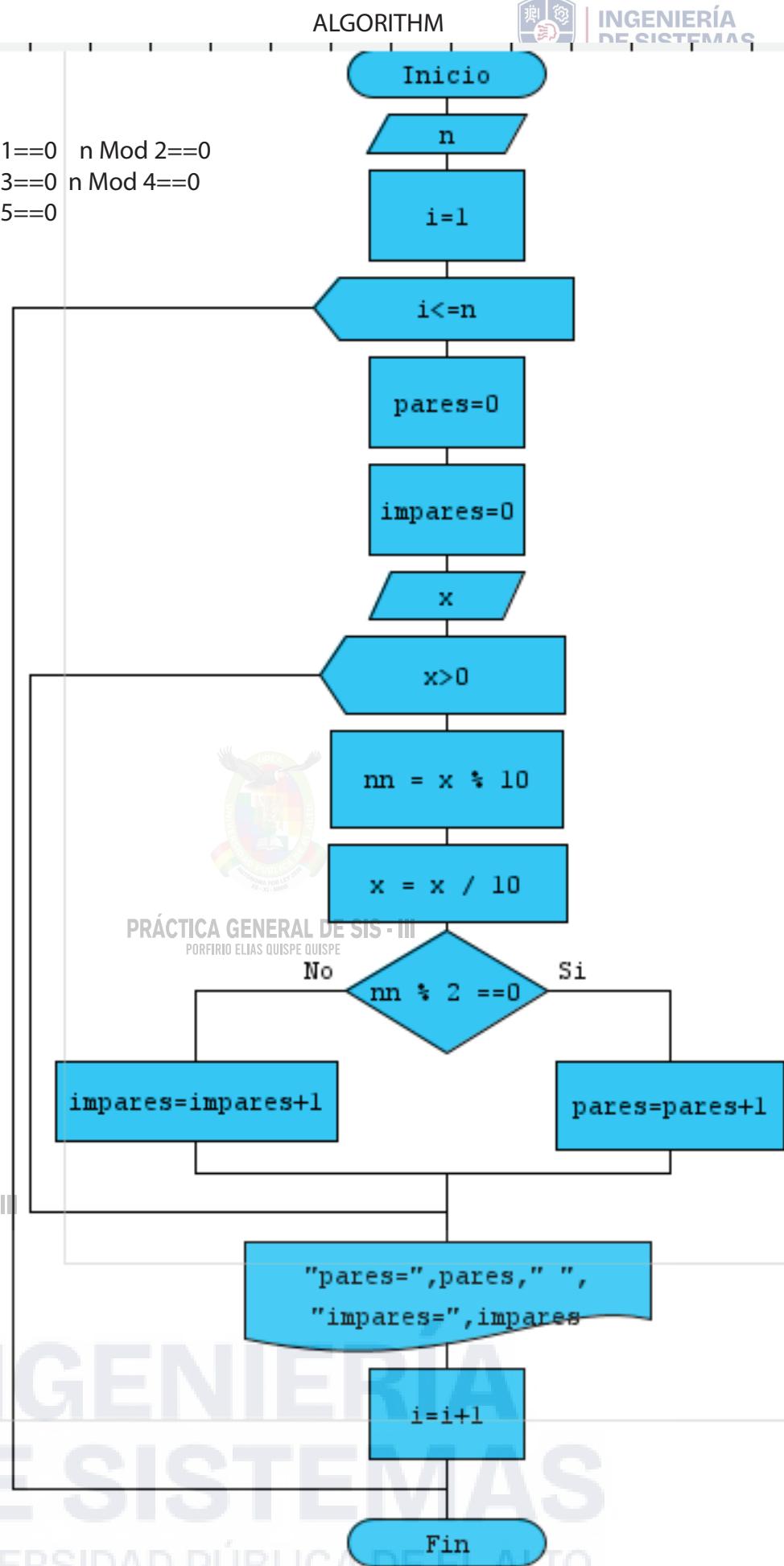
```
##como no hay do while en python
n = int(input("Ingrese un numero n"))
pares=0
impares=0
i = 1
while i<=n:
    pares = 0
    impares = 0
    x = int(input("Ingrese un numero n"))
    while x > 0:
        nn = x % 10
        x = x // 10
        if nn % 2 == 0:
            pares = pares + 1
        else:
            impares = impares + 1
    print("pares",pares,"")
    print ("impares",impares)
    i = i + 1
```

DESK CHECK

n = 2
x = 22
pares=2 impares=0 x = 22
pares=2 impares=0

Prueba de escritorio (PDE's)

Inicio	n	i	pares	impares
2	1	0	0	0
E	2	1	0	0
ELIAS QUISPE QUISPE	3	2		
		0		
		1		
		2		





PROBLEMA

9. Realizar un algoritmo que permita introducir un lote de n números y pueda mostrar el promedio de todos los números pares e impares.

ANALISIS

SOLUCION

n=3

Usar ciclo while para leer n numero de lotes

contar pares impares

nn = n mod 10

n = n div 10

if nn mod2==0

 contar pares

 suma promedio pares

caso contrario

 contar impares

 suma promedio pares

promedio /pares

promedio /impares

1 ANALIZAR EL PROBLEMA

Especificar resultado: mostrar en forma

si n= promedio de pares

Identificar datos a leer: n, nnotes

Determinar restricciones: usar datos enteros y

reales

Definir proceso: si i % 2 ==0

promedio= sumapares, sumaimpares/impares,
pares

ALGORITHMICAL DE SIS - III

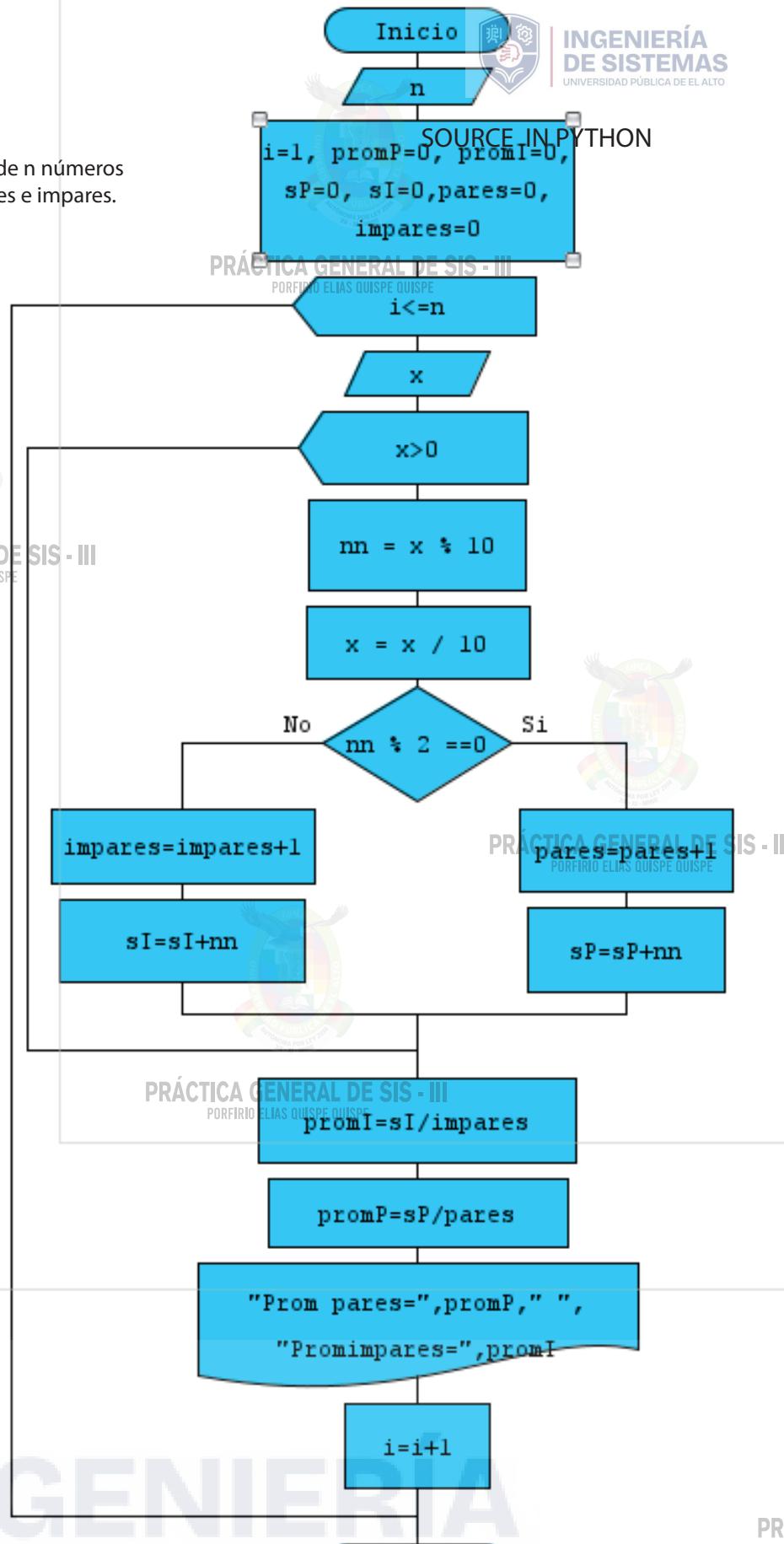
```
s = 0 i=1 pares=0 impar = 0
sP = 0 # suma de pares
sI = 0 # suma impares
proP=0 # promedio pares
promI=0# promedio impares
n = int(input("Ingrese un numero n"))
while i<=n:
    x=int(input("ingrese un numero X="))
    while x>0:
        nn = x % 10
        x = x // 10
        if nn % 2 == 0 :
            pares=pares+1
            sP=sP+nn
            print(nn)
        else:
            print("impar",nn)
            impar=impar+1
            sI=sI+nn
        promI=sI/impar
        promP=sP/pares
        print(" Promedio par",promI)
        print(" Promedio impar=",promP)
    i=i+1
```

DESK CHECK

```
= 2
= 45
om pares=4 Promimpares=5x = 74
om pares=4 Promimpares=6
```

Prueba de escritorio (PDE's)

Inicio	n	i	proP	promI	sP	sI	pares	impares	x	nn
	1	0	0	0	0	0	0	0	45	5
	2	4	5	5	4	5	1	1	4	4
	3	4	6	6	8	12	2	2	0	4



PROBLEMA

10. Leer un lote de N números enteros por teclado y determinar cuántos números son múltiplos de 5

ALGORITHM

ANALISIS

SOLUCION

n=375

Usar ciclo while

leer n numero de lotes contar multiplos de 5 y 7

nn = n mod 10

n = n div 10

if nn mod 5==0 or nn mod 7==0

 contar multiplos

mostrar multiplos

1 ANALIZAR EL PROBLEMA

Especificar resultado: mostrar en forma

si n= multiplos

Identificar datos a leer: nlotes

Determinar restricciones: usar datos enteros y

reales

PORFIRIO ELIAS QUISPE QUISPE

Definir proceso: si nn mod 5==0 or nn mod 7==0

mostrar

multiplos contados

2 DISEÑAR DIAGRAMA

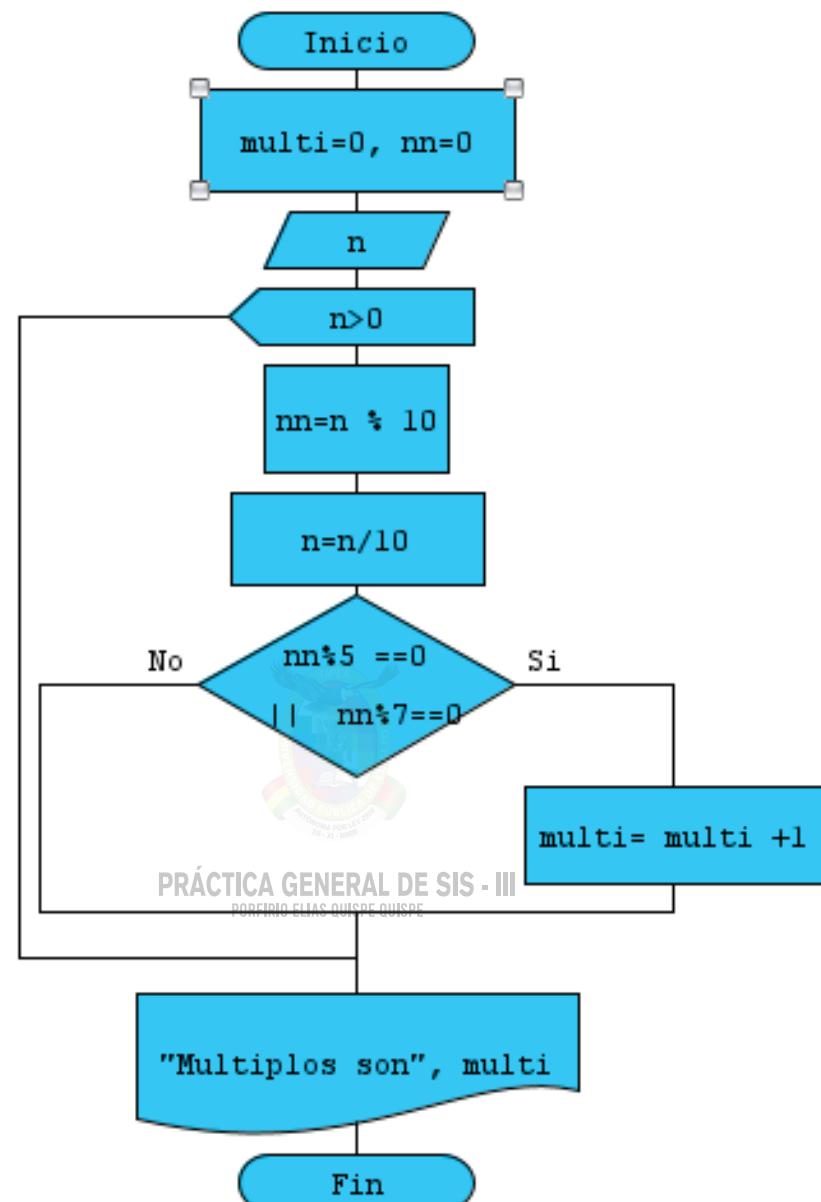
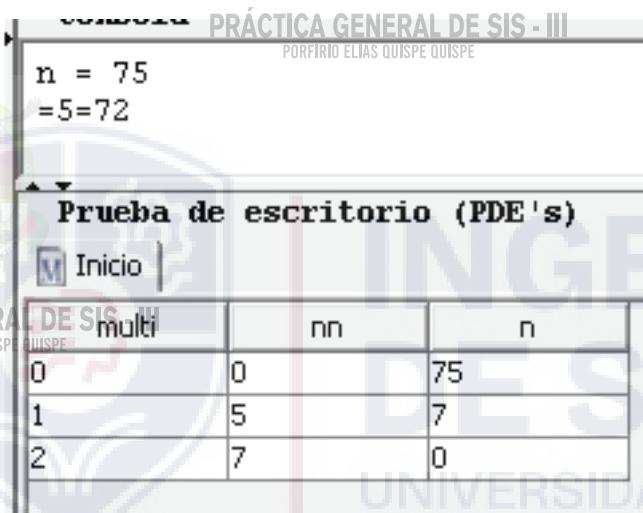
SOURCE IN PYTHON

```
nn = 0
multi = 0
n = int(input("Ingrese un numero n"))
```

```
while n>0:
    nn = n % 10
    n = n // 10
    if nn % 7 == 0 or nn % 5 == 0:
        multi=multi+1
```

```
print(" los mutiplos son",multi)
```

DESK CHECK



PROBLEMA

11. Leer números ingresados por teclado hasta que se introduzca un 0, determinar cuántos números primos se ingresaron, y realizar la sumatoria de los mismos

ANALISIS

SOLUCION

sea $n = 1325 = 3 \cdot 2 \cdot 5$ es primo la suma es 10

Usar ciclo while y for

Ler n numero de lotes sumar los primos de ese lote

```
if nn mod i==0
    contar si es multiplo el numero primo
    verificar si contador es ==2
```

sumar los primos ingresados x teclado

1 ANALIZAR EL PROBLEMA

Especificar resultado: la suma primos

Identificar datos a leer: lotes

PRÁCTICA GENERAL DE SIS - III
PORFIRIO ELIAS QUISPE QUISPE

Determinar restricciones: usar datos enteros

Definir proceso: si $n \% i == 0$ para verificar si es

primo y contar multiplos

si es primo $c=2$

```
suma=suma +n
mostrar suma
```

2 DISEÑAR DIAGRAMA

PROBLEMA EN PYTHON NO HAY DO WHILE

PRÁCTICA GENERAL DE SIS - III
PORFIRIO ELIAS QUISPE QUISPE

SOURCE IN PYTHON

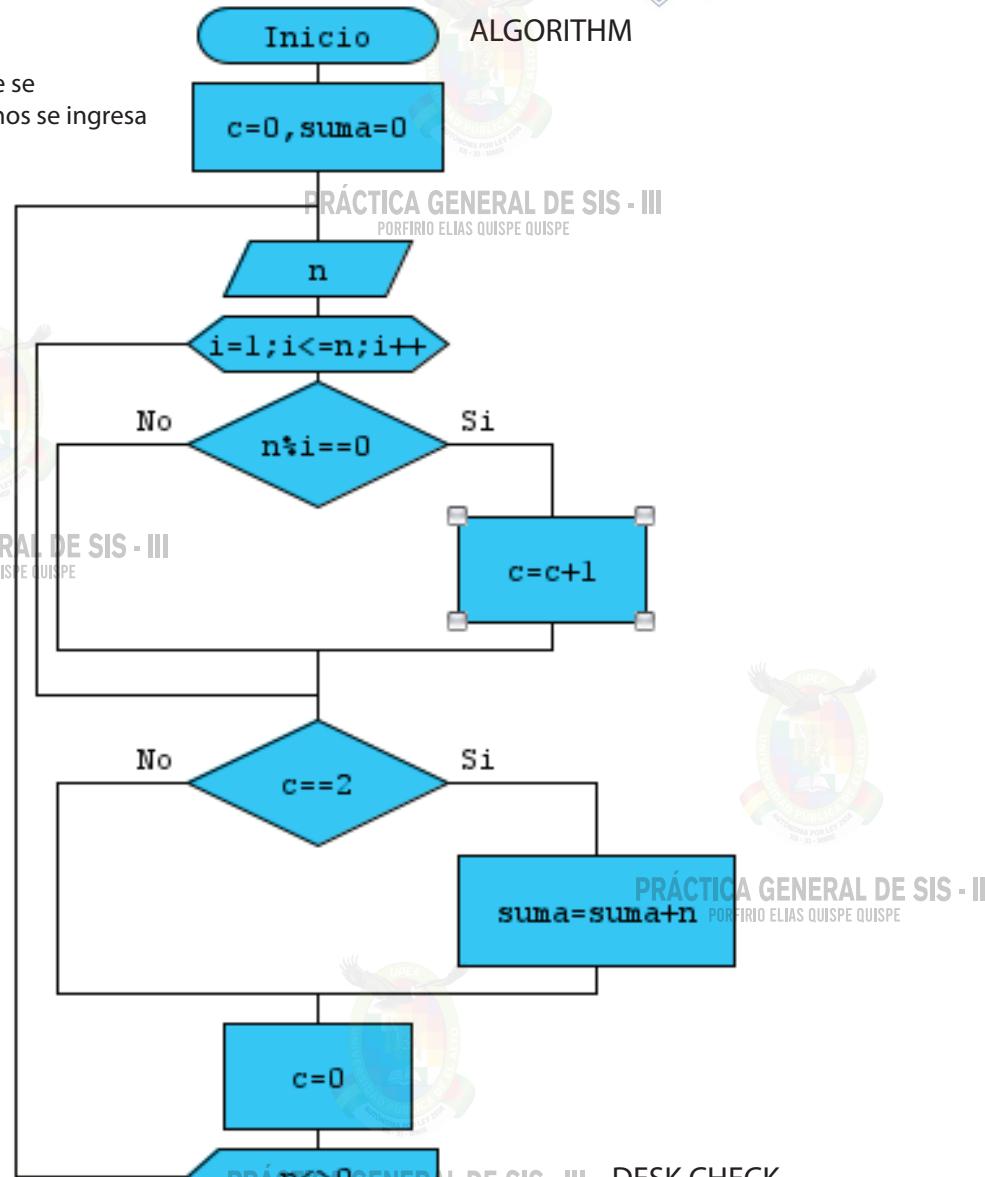
```
numeros_primos = 0
sumatoria_primos = 0

while True:
    numero = int(input("Ingrese un número (0 para terminar): "))
    if numero == 0:
        break

    es_primo = True
    if numero > 1:
        for i in range(2, int(numero**0.5) + 1):
            if numero % i == 0:
                es_primo = False
                break
        else:
            es_primo = False

    if es_primo:
        numeros_primos += 1
        sumatoria_primos += numero

print("Números primos ingresados:", numeros_primos)
print("sumatoria de números primos ingresados:",
      sumatoria_primos)
```



PRÁCTICA GENERAL DE SIS - III DESK CHECK

PRÁCTICA GENERAL DE SIS - III
PORFIRIO ELIAS QUISPE QUISPE

Consola

```
n = 3
n = 3
n = 3
n = 3
n = 0
12
```

Prueba de escritorio (PDE's)

Inicio |

c	suma	n	i
0	0	3	1
1	3	3	2
2	6	3	3
0	9	3	4
1	12	0	1
2			2
0		3	
1		4	
2		1	
0		2	
1		3	
2		4	
0		1	
0		2	
3			

PROBLEMA

12. Dado un número n, generar la siguiente serie: si $n=5$: 1,22,333,4444,55555.

SOLUCIÓN

ANÁLISIS

1 ANALIZAR EL PROBLEMA

Especificar resultado: mostrar serie

Identificar datos a leer: n

Determinar restricciones: usar datos enteros

Definir proceso: solo mostrar solo i generando n veces necesario al primer bucle

mostrar serie generada

2 DISEÑAR DIAGRAMA

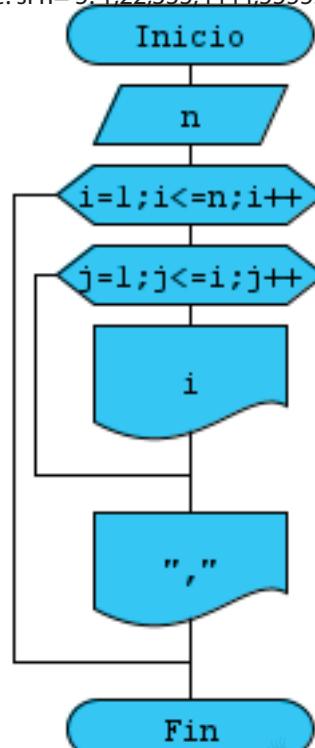
PRÁCTICA GENERAL DE SIS - III

PORFIRIO ELIAS QUISPE QUISPE

SOURCE IN PYTHON

```
n = int(input("Ingrese un numero n"))
for i in range(1, n+1):
    for j in range(1, i+1):
        print(i)
    print("")
```

ALGORITHM



DESK CHECK

Consola

```
n = 5
1,22,333,4444,55555,
```

Prueba de escritorio (PDE's)

Inicio

n	i	j
5	1	1
	2	2
	3	1
	4	2
	5	3
	6	1
		2
		3
		4
		1
		2
		3
		4
		5
		6

13. Dado un numero N, generar la serie: 0,1,0,2,3,0,4,5,6,0,...555.

SOLUCIÓN

1 ANALIZAR EL PROBLEMA

Especificar resultado: mostrar la sumatoria

Identificar datos a leer: n

Determinar restricciones: usar datos enteros

Definir proceso:

imprimir 0 cada vez que genere los numeros 1
 $2 = 1, 2$ veces $3 = 1,2,3$ veces $4 = 1,2,3,4$ veces $5 = 1,2,3,4,5$ veces

mostrar serie

2 DISEÑAR DIAGRAMA

PRÁCTICA GENERAL DE SIS - III

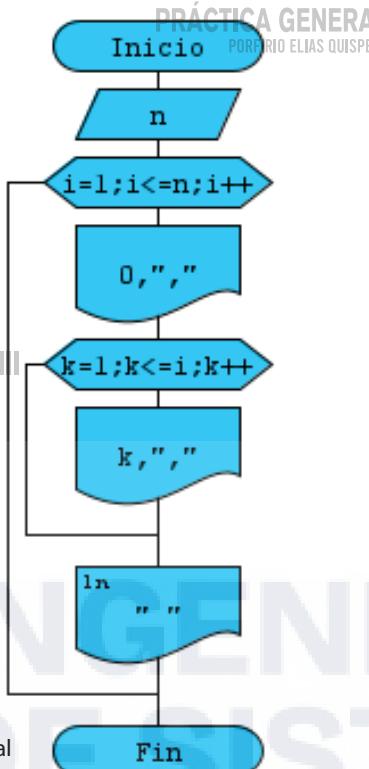
PORFIRIO ELIAS QUISPE QUISPE

SOURCE IN PYTHON

```
N = int(input("Ingrese un número N: "))
contador = 0

for i in range(N + 1):
    for j in range(i):
        print(contador, end="")
        contador += 1
    print(0, end="")

# Para asegurar que no haya una coma adicional al final
print("\b")
```



PRÁCTICA GENERAL DE SIS - III

PORFIRIO ELIAS QUISPE QUISPE

Consola

```
n = 6
0,1,
0,1,2,
0,1,2,3,
0,1,2,3,4,
0,1,2,3,4,5,
0,1,2,3,4,5,6,
```

Prueba de escritorio (PDE's)

Inicio

n	i	k
6	1	1
	2	2
	3	1
	4	2
	5	3
	6	1
	7	2
		3
		4
		1
		2
		3
		4



PROBLEMA

14. Generar la siguiente serie: 2,0,4,0,8,0,...,n. NOTA: Verificar que se ingrese un numero positivo
n=5 2,0,4,0,8

ANALISIS SOLUCION

1 ANALIZAR EL PROBLEMA

verificar que los ingresados sean mayor de cero
si($n > 0$)

para generar serie usar un ciclo for de 1 hasta n

PRÁCTICA GENERAL DE SIS - III
PORFIRIO ELIAS QUISPE QUISPE

Especificar resultado: mostrar la serie
Identificar datos a leer: n

Determinar restricciones: usar datos enteros

Definir proceso: si $i \% 2 == 0$

ser = $2 * i$

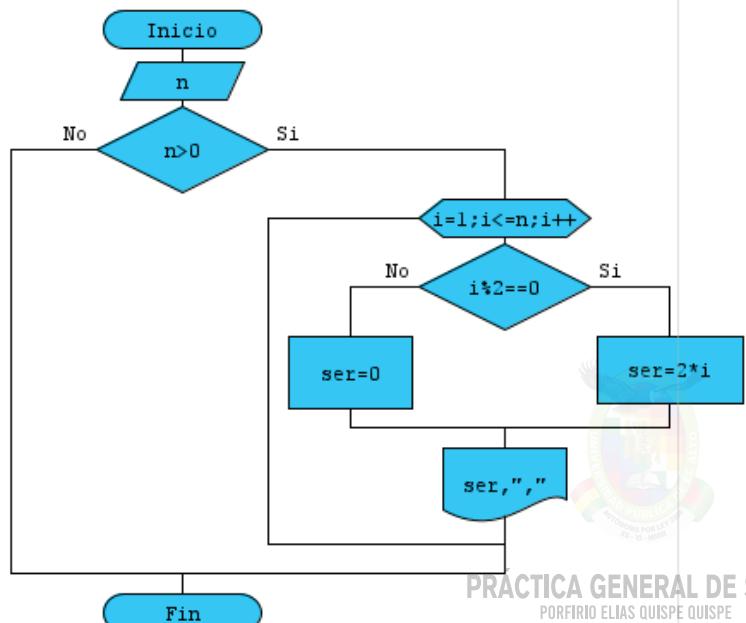
caso contrario

ser = 0

mostrar ser

2 DISEÑAR DIAGRAMA

ALGORITHM



PRÁCTICA GENERAL DE SIS - III
PORFIRIO ELIAS QUISPE QUISPE

SOURCE IN PYTHON

```

def generar_serie(n):
    if n <= 0:
        print("Por favor, ingresa un número positivo.")
        return

    serie = []
    for i in range(1, n + 1):
        if i % 2 == 0:
            serie.append(2 * i)
        else:
            serie.append(0)

    return serie
  
```

n = int(input("Ingresa el valor de n: "))
print("La serie generada es:", general_serie(n))

DESK CHECK

Consola

n = 5
0,4,0,8,0,

Prueba de escritorio (PDE's)

Inicio	n	i	ser
5	1	0	
	2	4	
	3	0	
	4	8	
	5	0	
	6		

**INGENIERÍA
DE SISTEMAS**

PRÁCTICA GENERAL DE SIS - III
PORFIRIO ELIAS QUISPE QUISPE

PROBLEMA

15. Generar la siguiente serie: 3, 6, 8, 16, 18, 36, 38, 76, 78,...

SOLUCIÓN

ANÁLISIS

1 ANALIZAR EL PROBLEMA

sigue un patrón donde cada número se duplica, y luego se le suma 2 para obtener el siguiente número.

Comenzamos con 3.

Multiplicamos por 2 y sumamos 2 para obtener 8.

Multiplicamos por 2 y sumamos 2 para obtener 18.

Continuamos con esta secuencia para generar los siguientes números.

Especificar resultado: mostrar sucesión 3, 6, 8, 16, 18

Identificar datos a leer: n

Determinar restricciones: usar datos enteros

Definir proceso: si $i \% 2 == 1$

serie = serie * 2

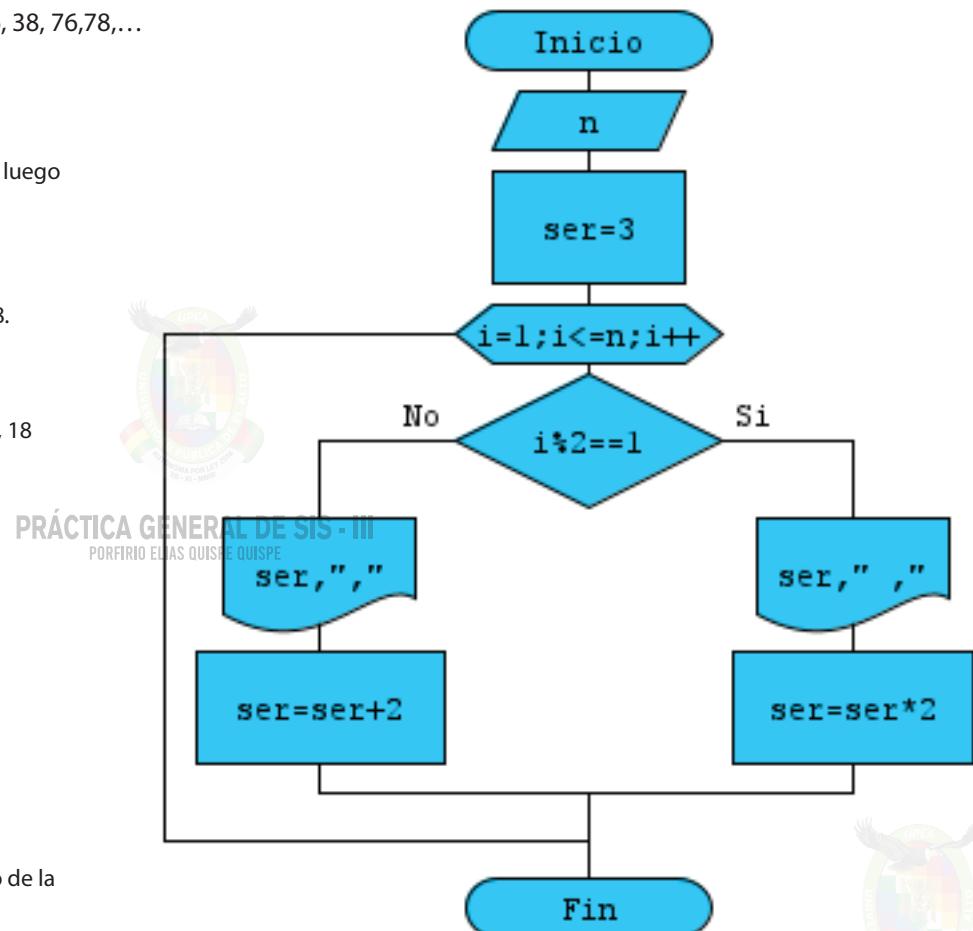
caso contrario

serie = serie + 2

mostrar serie

2 DISEÑAR DIAGRAMA

ALGORITHM



SOURCE IN PYTHON

```

def generar_serie(n):
    serie = [3] # Comenzamos con el primer número de la
    serie
    for i in range(1, n):
        if i % 2 == 1: # Si el índice es impar
            serie.append(serie[-1] * 2) # Multiplicamos el último
        número por 2
        else:
            serie.append(serie[-1] * 2 + 2) # Multiplicamos el
        último número por 2 y sumamos 2
    return serie

# Generar los primeros 10 términos de la serie
n = int(input("ingrese un numero n"))
serie_generada = generar_serie(n)
print("Serie generada:", serie_generada)
    
```

DESK CHECK

```

n = 9
3 ,6,8 ,16,18 ,36,38 ,76,78 ,
    
```

Prueba de escritorio (PDE's)

Inicio |

n	ser	i
9	3	1
	6	2
	8	3
	16	4
	18	5
	36	6
	38	7
	76	8
	78	9
	156	10





PROBLEMA

18. Generar la siguiente serie: 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0...

SOLUCIÓN

ANALISIS

1 ANALIZAR EL PROBLEMA

000
111
000
111
000..

Especificar resultado: mostrar serie

10, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1

Identificar datos a leer: n

Determinar restricciones: usar datos enteros

Definir proceso: para generar por lado derecho 000 tres veces y izquierdo 111 tres veces

usamos dos ciclos for n<=3 cada vez que genere

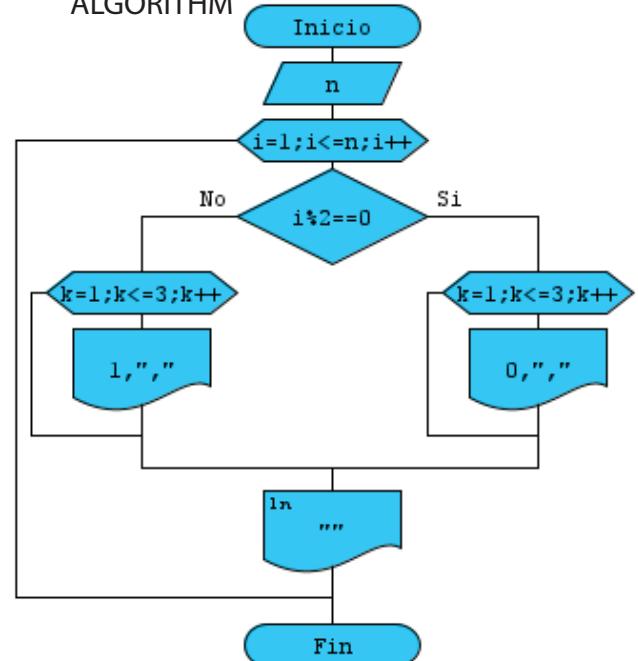
PORFIRIO ELIAS QUISPE

mostrar 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1

DESK CHECK

Consola		
n = 3		
1,1,1,		
0,0,0,		
1,1,1,		
Prueba de escritorio (PDE's)		
Inicio		
n	i	k
3	1	1
	2	2
	3	3
	4	4
		1
		2

ALGORITHM



2 DISEÑAR DIAGRAMA

SOURCE IN PYTHON

```

n = int(input("Ingrese un número N: "))
for i in range(1,n+1):
    if i%2==0:
        for i in range(1,3+1):
            print(0,"")
    else:
        for i in range(1,3+1):
            print(1,"")
    
```

PROBLEMA

19. Dado un número N, obtener la suma de los n números naturales, ejemplo: n=6 1+2+3+

ANALISIS

SOLUCIÓN

1 ANALIZAR EL PROBLEMA

Especificar resultado: mostrar la sumatoria

1+2+3+4+5+6=21

Identificar datos a leer: n,

Determinar restricciones: usar datos enteros
usar un bucle para generar la serie

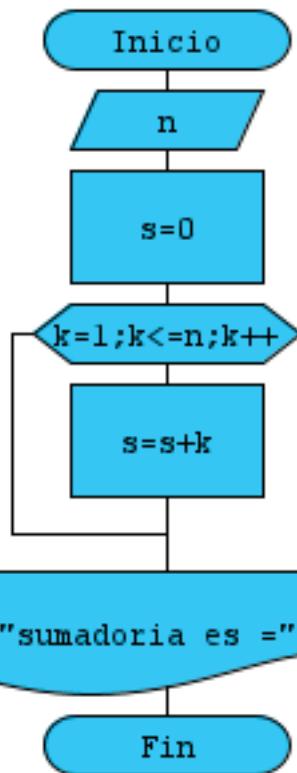
Definir proceso: s=s+k

mostrar sumatoria

SERIES



ALGORITHM



DESK CHECK

Consola		
n = 5		
sumatoria es =15		
Prueba de escritorio (PDE's)		
Inicio		
n	s	k
5	0	1
	1	2
	3	3
	6	4
	10	5
	15	6

2 DISEÑAR DIAGRAMA

SOURCE IN PYTHON

```

def suma_numeros_naturales(n):
    suma = 0
    for i in range(1, n + 1):
        suma += i
    return suma

n = 6
resultado = suma_numeros_naturales(n)
print("La suma de los primeros", n, "números naturales es:", resultado)
    
```



PROBLEMA

20. Calcular la siguiente sumatoria:

ANALISIS

SOLUCION

1 ANALIZAR EL PROBLEMA

Especificar resultado: mostrar la sumatoria en cada iteración, se suma el término

$$\text{formula} = i / i+1$$

Identificar datos a leer: n,

Determinar restricciones: usar datos enteros usar un bucle para generar la serie y realizar la sumatoria

Definir proceso: $s = i / i+1$
mostrar serie y sumatoria

2 DISEÑAR DIAGRAMA

PRÁCTICA GENERAL

PORFIRIO ELIAS QUISPE QI

SOURCE IN PYTHON

```
def suma_serie(n):
    suma = 0
    for i in range(1, n+1):
        suma += i / (i + 1)
    return suma
```

n = 6
resultado = suma_serie(n)

print("La suma de la serie para n =", n, "es:", resultado)

PROBLEMA

21. Genera la siguiente sumatoria $s = -1 + 2 - 5 + 10 - 17 + 26$

SOLUCION

ANALISIS

1 ANALIZAR EL PROBLEMA

Especificar resultado: mostrar la sumatoria y serie

Identificar datos a leer: n

Determinar restricciones: usar datos enteros

Definir proceso: si $i \% 2 == 0$ para signo

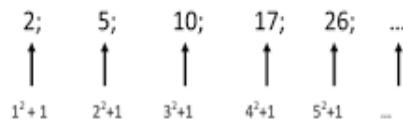
$$\text{exp} = \text{sig} * ((i*i)+1)$$

caso contrario

$$\text{exp} = (i*i)+1$$

$$\text{sumatoria} = \text{sumatoria} + \text{exp}$$

mostrar sumatoria



2 DISEÑAR DIAGRAMA

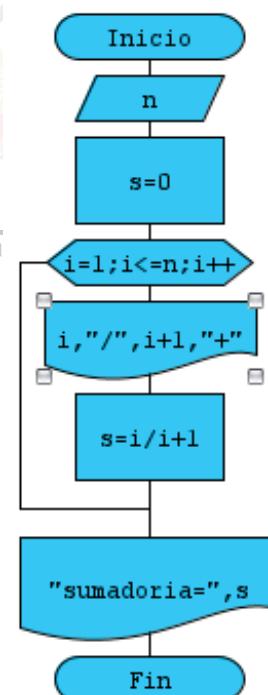
SOURCE IN PYTHON

```
n = int(input("Ingrese un número (0 para terminar): "))
exp=0
sig = 1*-1
s=0
for i in range(1, n + 1):
    if i % 2 == 0:
        exp = i**+1
        print("+"+exp)
    else:
        exp = sig*((i*i) + 1)
        print("+"+exp)
    s = s + exp
print("sumatoria=",s)
```

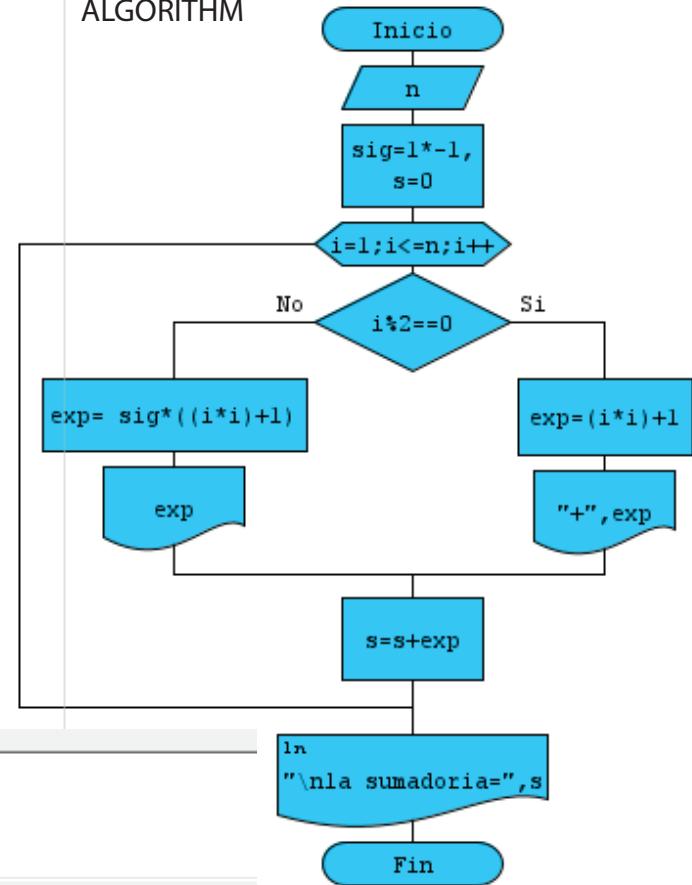
DESK CHECK

Prueba de escritorio (PDE's)		
Inicio	n	s
	5	0
		2
		2
		2
		2
		2
		2

ALGORITHM



ALGORITHM



SOURCE IN PYTHON

```
Consola
n = 3
-2+5-10
la sumatoria=-7
1n
"\nla sumatoria=",>s
Fin
```



PROBLEMA

22. Genera la siguiente sumatoria $s = 1! + 3! + 5! + 7! + \dots$

SOLUCIÓN

ANALISIS

factorial - n=5 $5 \times 4 \times 3 \times 2 \times 1 = 120$

1 ANALIZAR EL PROBLEMA

Especificar resultado: mostrar la sumatoria de factorial

Identificar datos a leer: n

Determinar restricciones: usar datos enteros

Definir proceso: generar numeros impares

impar=1

impar=impar+2

s=s+factorial

mostrar sumatoria

2 DISEÑAR DIAGRAMA

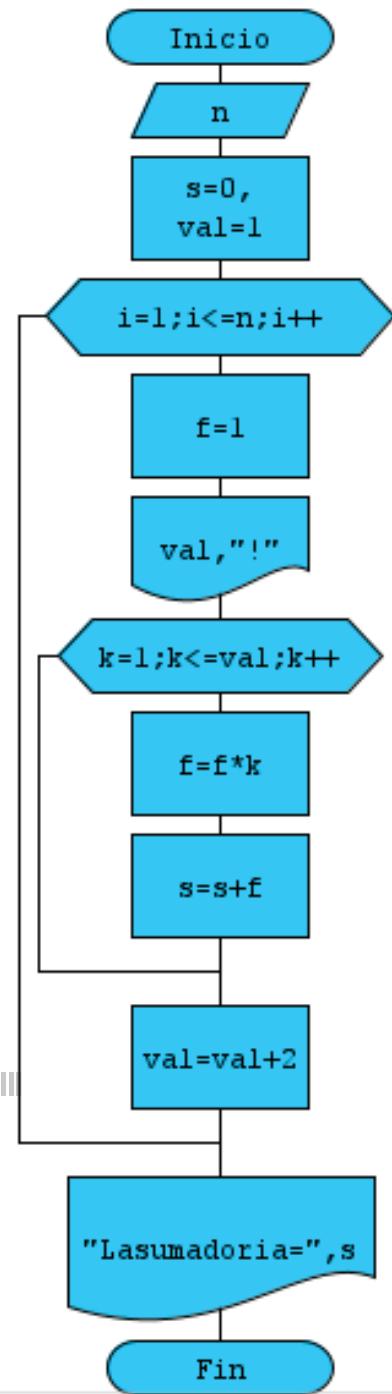
SOURCE IN PYTHON

```
nn=0
s=0
val=1
n = int(input("Ingrese un número N: "))
for i in range(1,n+1):
    f=1
    print(val,"")
    for k in range(1,val+1):
        f=f*k
        s=s+f
    val=val+2
print("la sumatoria es",s)
```

DESK CHECK

Consola						
Prueba de escritorio (PDE's)						
Inicio						
n	s	val	i	f	k	
3	0	1	1	1	1	
	1	3	2	1	2	
	2	5	3	1	1	
	4	7	4	1	2	
	10			2	3	
	11			6	4	
	13			1	1	
	19			1	2	
	43			2	3	
	163			6	4	
				24	5	
				120	6	

ALGORITHM





PROBLEMA

23. Genera la siguiente sumatoria

SOLUCIÓN

$$s = -1 - 2 + 7 - 3 - 4 + 14 - 5 - 6 + 21 - 7$$

ANÁLISIS

1 ANALIZAR EL PROBLEMA

Especificar resultado: mostrar la sumatoria

Identificar datos a leer: n,

Determinar restricciones: usar datos enteros

Definir proceso: impar+2, par+2

mul=7*i

mostrar pares impares y mutiplo 7

$$1x7=4 + 7x2=14 + 7x3=21$$

la serie conforma impares
pares y multiplo 7

imparempares PRÁCTICA C
PORFIRIO
-1,-3,-5

pares
-4,6,8
multiplo 7
 $1x7=4 + 7x2=14 + 7x3=21$

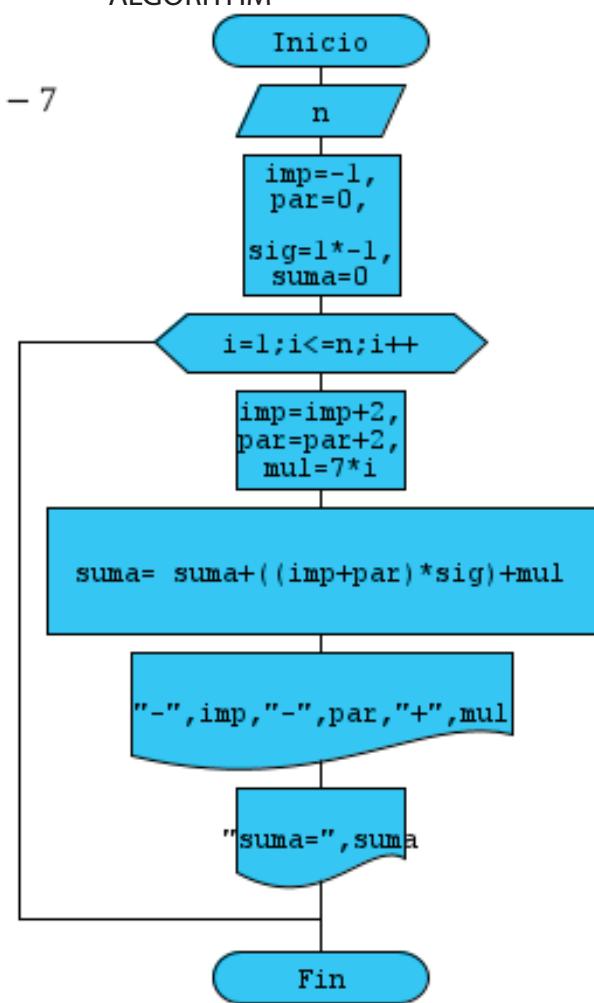
2 DISEÑAR DIAGRAMA



PRÁCTICA GENERAL DE SIS - III

PORFIRIO ELIAS QUISPE QUISPE

ALGORITHM



SOURCE IN PYTHON

A GENERAL DE SIS - III

```
ELIAS QUISPE
imp=-1
par=0
sig= 1*-1
suma=0
```

```
n = int(input("Ingrese un número N:"))

for i in range(1,n+1):
    imp=imp+2
    par=par+2
    mul=7*i
    suma=suma+((imp+par)*sig)+mul
    print("-" , imp , "-" , par , "+" , mul)
print("la suma=" , suma)
```

Consola

```
n = 4
-1-2+7suma=4-3-4+14suma=11-5-6+21suma=21-7-8+28suma=34
```

Prueba de escritorio (PDE's)

Inicio	n	imp	par	sig	suma	i	mul
	4	-1	0	-1	0	1	7
		1	2		4	2	14
		3	4		11	3	21
		5	6		21	4	28
		7	8		34	5	



PROBLEMA

24. Hallar la siguiente sumatoria

SOLUCION

ANALISIS

1 ANALIZAR EL PROBLEMA

Especificar resultado: mostrar la suma

Identificar datos a leer: n, x

Determinar restricciones: usar datos enteros

Definir proceso: for i in range(1, n + 1):

```
par=par+2
```

```
print("x^",par,"+")
```

```
for k in range(1, par + 1):
```

```
b=b*x
```

```
s=s+b
```

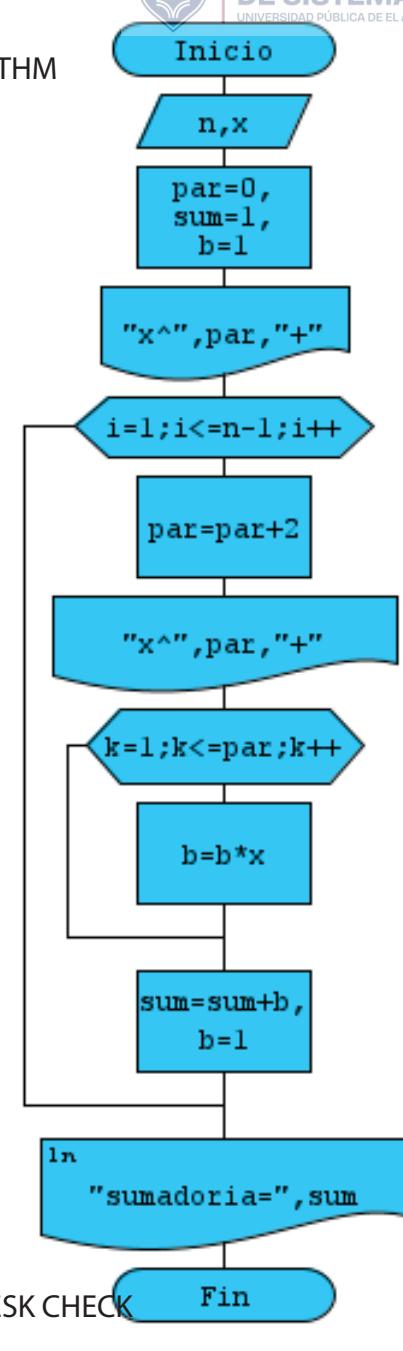
```
b=1
```

mostrar: la suma s

2 DISEÑAR DIAGRAMA

$$S = x^0 + x^2 + x^4 + x^6 + \dots + x^{j+2-2}$$

ALGORITHM



SOURCE IN PYTHON

```

n = int(input("Ingrese un número natural N: "))
x = int(input("Ingrese un número natural X: "))
mult=0
s=1
par=0
b=1
print("x^,par,")
for i in range(1, n + 1):
    par=par+2
    print("x^,par,")
    for k in range(1, par + 1):
        b=b*x
        s=s+b
        b=1
    print("la suma es=",s)
    
```



DESK CHECK

CONSOLA							
<pre>n = 3 x = 2 x^0+x^2+x^4+sumadaria=21</pre>							

Prueba de escritorio (PDE's)							
Inicio							
n	x	par	sum	b	i	k	
3	2	0	1	1	1	1	
		2	5	2	2	2	
		4	21	4	3	3	
				1		1	
				2		2	
				4		3	
				8		4	
				16		5	
				1			

**INGENIERÍA
DE SISTEMAS**
UNIVERSIDAD PÚBLICA DE EL ALTO



GENERAL DE SIS - III

RIO ELIAS QUISPE QUISPE

SIS - III



PROBLEMA

25. Calcular la siguiente sumatoria

$$S = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \frac{x^7}{7!} + \dots$$

SOLUCION

ANALISIS

1 ANALIZAR EL PROBLEMA

Especificar resultado: mostrar la sumatoria

Identificar datos a leer: n, x

Determinar restricciones: usar datos enteros y reales

Definir proceso: sor i in range(1, n + 1):

```
f=1
print("x^",imp,"/",imp,"!","+")
for i in range(1, imp+ 1):
    b=b*x
    for k in range(1, imp + 1)
        f=f*k
    s=s+b/f
    b=1
    imp=imp+2
mostrar suma de exponencia /factorial
```

2 DISEÑAR DIAGRAMA

PRÁCTICA GENERAL DE SIS - III

PORFIRIO ELIAS QUISPE QUISPE

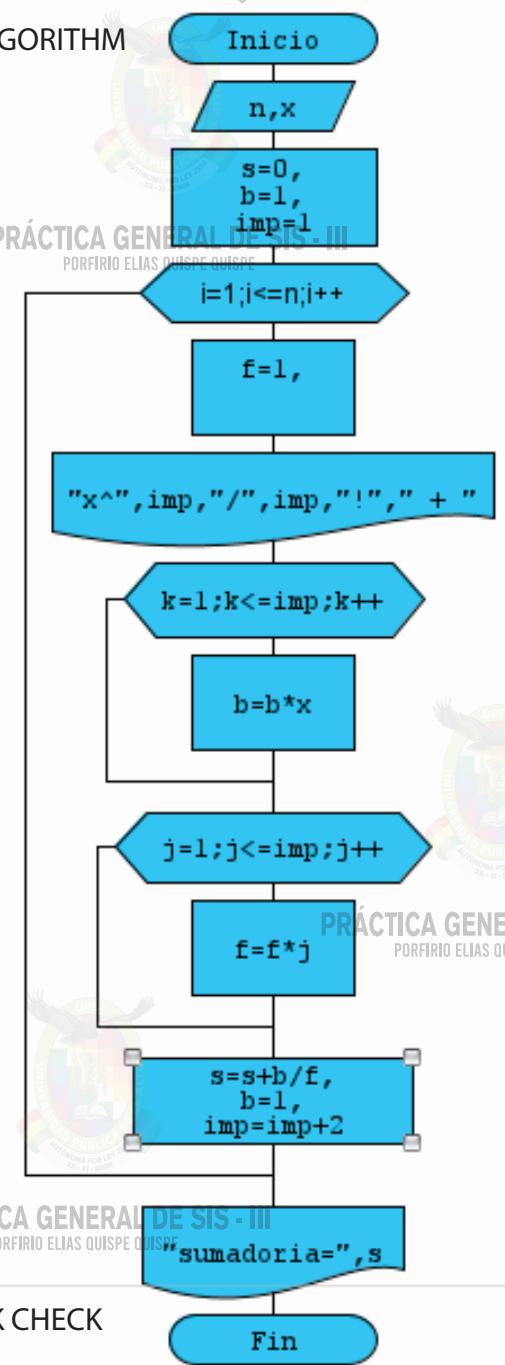
PRÁCTICA GENERAL DE SIS - III

PORFIRIO ELIAS QUISPE QUISPE

```
n = int(input("Ingrese un número entero N: "))
x = int(input("Ingrese un número entero X: "))
mult=0
s=0
imp=1
b=1
for i in range(1, n + 1):
    f=1
    print("x^",imp,"/",imp,"!","+")
    for i in range(1, imp+ 1):
        b=b*x
        for k in range(1, imp + 1):
            f=f*k
        s=s+b/f
        b=1
        imp=imp+2

print("la suma es=",s)
```

ALGORITHM



DESK CHECK

Consola

```
n = 3
x = 3
x^1/1! + x^3/3! + x^5/5! + sumadaria=9
```

Prueba de escritorio (PDE's)

Inicio |

n	x	s	b	imp	i	f	k	j
3	3	0	1	1	1	1	1	1
			3	3	2	1	2	2
			7	1	5	3	1	1
			9	3	7	4	1	2
							2	3
							27	3
								4
							1	1
							3	2
							9	3
							27	4
							81	4
							243	5
								6
							1	6
								1



INGENIERÍA DE SISTEMAS

UNIVERSIDAD PÚBLICA DE EL ALTO



DESCOMPOSICIÓN DE NÚMERO



INGENIERÍA
DE SISTEMAS
UNIVERSIDAD PÚBLICA DE EL ALTO

PROBLEMA

26. Dado un número entero, determinar cuántos dígitos son múltiplos de 3. Si $n=15398463$ La cantidad de múltiplos de 3 son: 4

ANÁLISIS SOLUCIÓN

1 ANALIZAR EL PROBLEMA

Especificar resultado: mostrar la sumatoria

Identificar datos a leer: n

Determinar restricciones: usar datos enteros

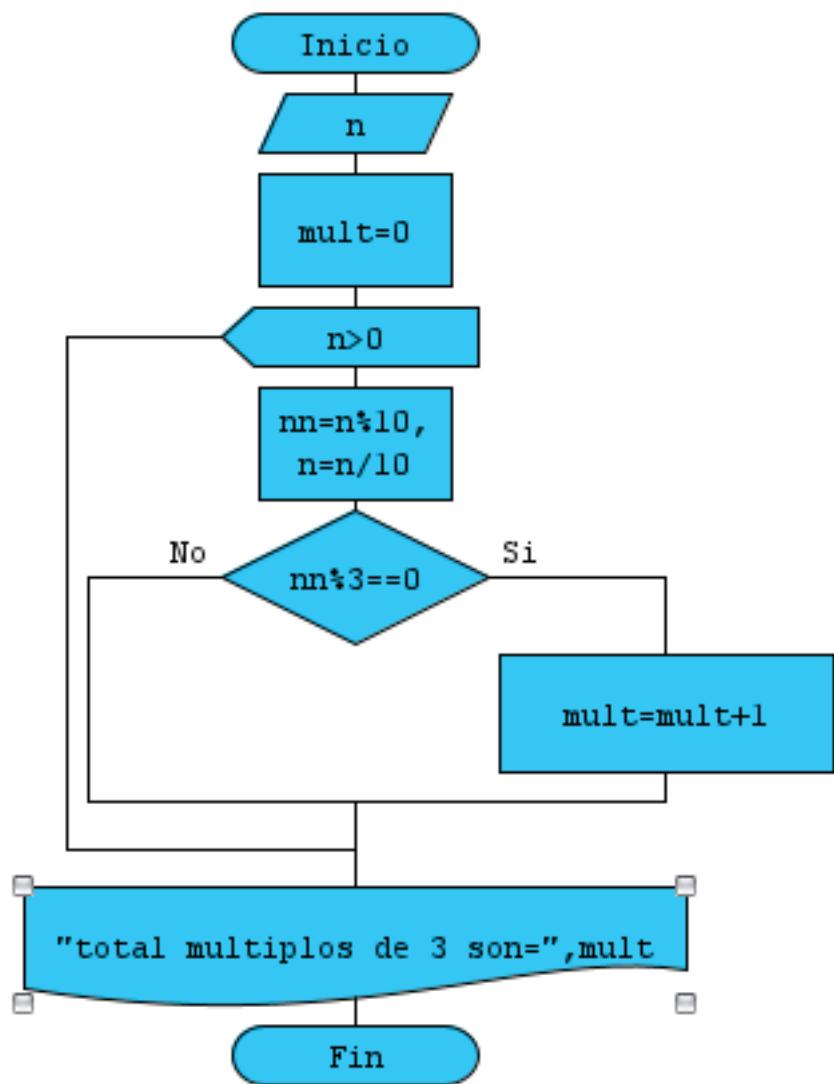
Definir proceso: while $n>0$:

```
nn=n%10
n=n//10
```

```
if nn%3==0:
    mult=mult+1
```

PRÁCTICA GENERAL DE SIS - III
PORFIRIO ELIAS QUISPE QUISPE

2 DISEÑAR DIAGRAMA



SOURCE IN PYTHON

```
mult=0
n = int(input("Ingrese un número entero N: "))

while n>0:

    nn=n%10
    n=n//10

    if nn%3==0:
        mult=mult+1

print("total multiplos de 3 es",mult)
```

PRÁCTICA GENERAL DE SIS - III
PORFIRIO ELIAS QUISPE QUISPE

DESK CHECK

```
n = 15398463
total mult de 3 son=4
```

Prueba de escritorio (PDE's)

Inicio

n	mult	nn
15398463	0	3
1539846	1	6
153984	2	4
15398	3	8
1539	4	9
153		3
15		5
1		1
0		

**INGENIERÍA
DE SISTEMAS**
UNIVERSIDAD PÚBLICA DE EL ALTO



PROBLEMA

27. Dado un número N, calcular la sumatoria de todos sus dígitos.

SOLUCIÓN

ANÁLISIS

1 ANALIZAR EL PROBLEMA

Especificar resultado: mostrar

Identificar datos a leer: n

Determinar restricciones: usar datos enteros

Definir proceso:

while n>0

 nn=n%10

 n=n//10

 s=s+nn

mostrar suma de digitos s

2 DISEÑAR DIAGRAMA

PRÁCTICA GENERAL DE SIS - III

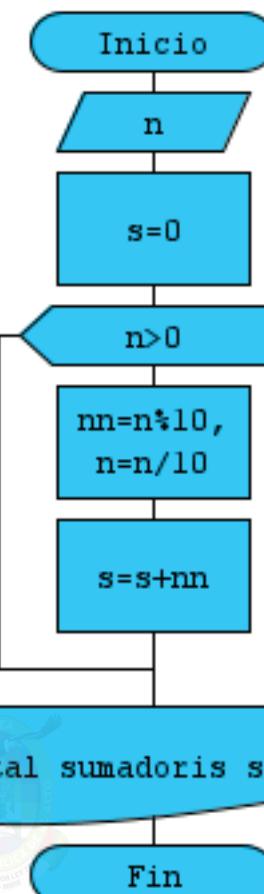
SOURCE IN PYTHON

```
s=0
n = int(input("Ingrese un número N:"))
while n>0:
    nn=n%10
    n=n//10
    s=s+nn
print(" la suma es=",s)
```

DESK CHECK

Prueba de escritorio (PDE's)		
Inicio	n	s
1234	0	4
123	4	3
12	7	2
1	9	1
0	10	

ALGORITHM



PROBLEMA

28. Dando un número n de 3 dígitos, mostrar si el número es capicúa o no.

SOLUCIÓN

ANÁLISIS

1 ANALIZAR EL PROBLEMA

Especificar resultado: mostrar la sumatoria

Identificar datos a leer: n, x

Determinar restricciones: usar datos enteros y reales

Definir proceso: while n>0

 nn=n%10

 n=n//10

 r=r*10+nn

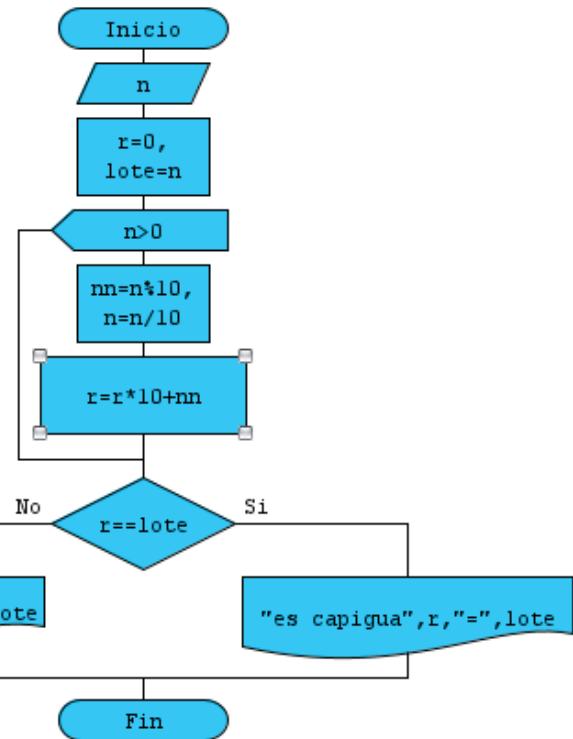
si r==lote

mostrar si es capicua o no

2 DISEÑAR DIAGRAMA

PRÁCTICA GENERAL DE SIS - III

ALGORITHM



SOURCE IN PYTHON

```
r=0
lote=0
nn=0
n = int(input("Ingrese un número N:"))
lote=n
while n>0:
    nn=n%10
    n=n//10
    r=r*10+nn
if r==lote:
    print("es capicua",r,"=",lote)
else:
    print("No es capicua",r,"=",lote)
```

Prueba de escritorio (PDE's)			
Inicio	n	r	lote
121	0	121	1
12	1		2
1	12		1
0	121		



PROBLEMA

29. Realizar un algoritmo que lea un número entero positivo hallar su dígito mayor.

SOLUCIÓN

ANÁLISIS

1 ANALIZAR EL PROBLEMA

Especificar resultado: mostrar el mayor del lote de los dígitos compuestos

Identificar datos a leer: n

Determinar restricciones: usar datos enteros

Definir proceso:

may=n%10

while n>0:

 n=n//10

 nn=n%10

 if may>nn:

 may=may

 else:

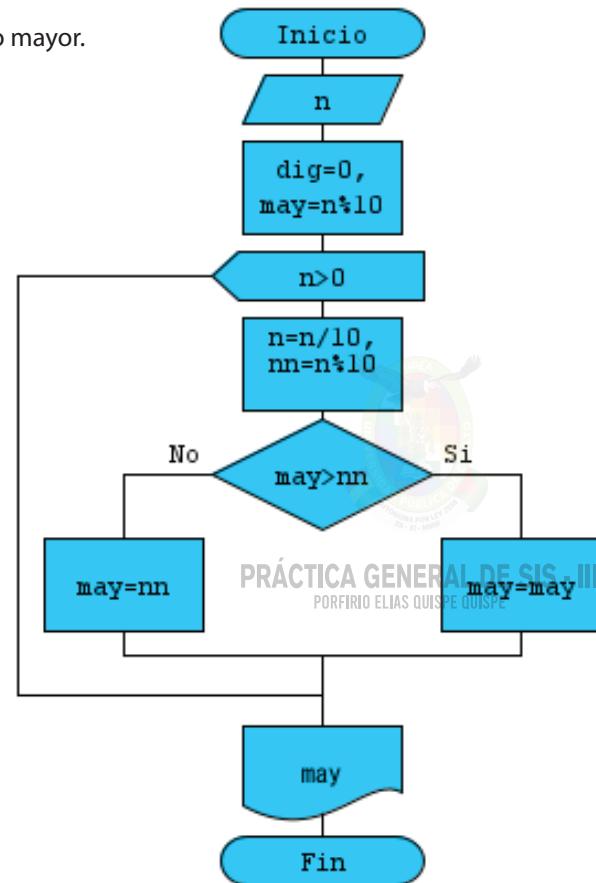
 may=nn

mostrar el mayor , may

GENERAL DE SIS - II

2 DISEÑAR DIAGRAMA

ALGORITHM



PRÁCTICA GENERAL DE SIS - III

PORFIRIO ELIAS QUISPE QUISPE

SOURCE IN PYTHON

```

may=0
dig=0
pro=0
n = int(input("Ingrese un número N: "))
may=n%10
while n>0:
    n=n//10
    nn=n%10
    if may>nn:
        may=may
    else:
        may=nn
    
```

PRÁCTICA GENERAL DE SIS - III

PORFIRIO ELIAS QUISPE QUISPE

```
print(" mayor numero es=",may)
```

DESK CHECK

n = 12587

Prueba de escritorio (PDE's)			
M	Inicio		
n	dig	may	nn
12587	0	7	8
1258		8	5
125		8	2
12		8	1
1		8	0
0		8	

PRÁCTICA GENERAL DE SIS - III
PORFIRIO ELIAS QUISPE QUISPE



PROBLEMA

30. Dado un número entero, hallar el promedio de sus dígitos.

SOLUCIÓN

ANÁLISIS

1 ANALIZAR EL PROBLEMA

Especificar resultado: mostrar promedio

Identificar datos a leer: n

Determinar restricciones: usar datos enteros

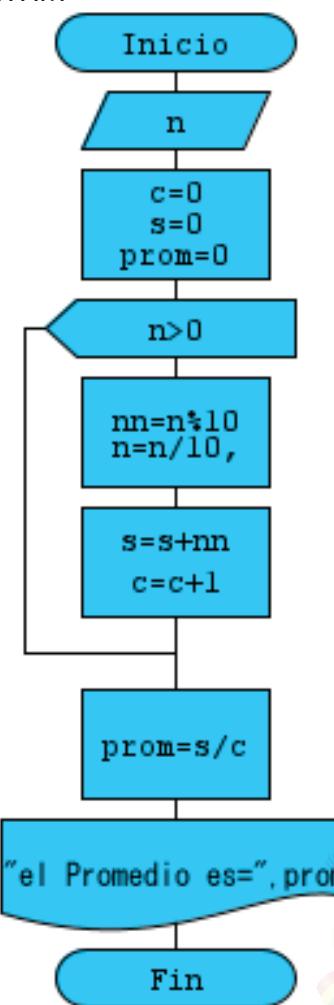
Definir proceso:

while n>0:

```
nn=n%10
n=n//10
s=s+nn
c=c+1
pro=s/c
mostrar promedio "pro"
```

sea $n = 396$ donde $396 \% 10 = 6$,
 $396 // 10 = 9$, $9 \% 10 = 3$

ALGORITHM



PRÁCTICA GENERAL DE SIS - III

PORFIRIO ELIAS QUISPE QUISPE

SOURCE IN PYTHON

```
c=0
s=0
pr=0
n = int(input("Ingrese un número N: "))
while n>0:
    nn=n%10
    n=n//10
    s=s+nn
    c=c+1
    pr=s/c
print(" promedio es=",pr)
```

PRÁCTICA GENERAL DE SIS - III
PORFIRIO ELIAS QUISPE QUISPE

DESK CHECK

Consola				
n = 12345 el Promedio es=3				
Prueba de escritorio (PDE's)				
n	c	s	prom	nn
12345	0	0	0	5
1234	1	5	3	4
123	2	9		3
12	3	12		2
1	4	14		1
0	5	15		



PROBLEMA

31. Dado un número x , obtener un nuevo número a partir de sus dígitos múltiplos de 3.

ANALISIS
SOLUCION

sea $n = 396$ donde $3/3=0$,
 $6/3=2$, $9/3=3$

1 ANALIZAR EL PROBLEMA

Especificar resultado: mostrar nuevo numero

que es divisible entre 3

Identificar datos a leer: n ,

Determinar restricciones: usar datos enteros

Definir proceso:

while $n > 0$:

$nn = n \% 10$

$n = n // 10$

 if $nn \% 3 == 0$:

$nmu = nmu + nn$

mostrar nuevo numero de los digitos, nmu

2 DISEÑAR DIAGRAMA

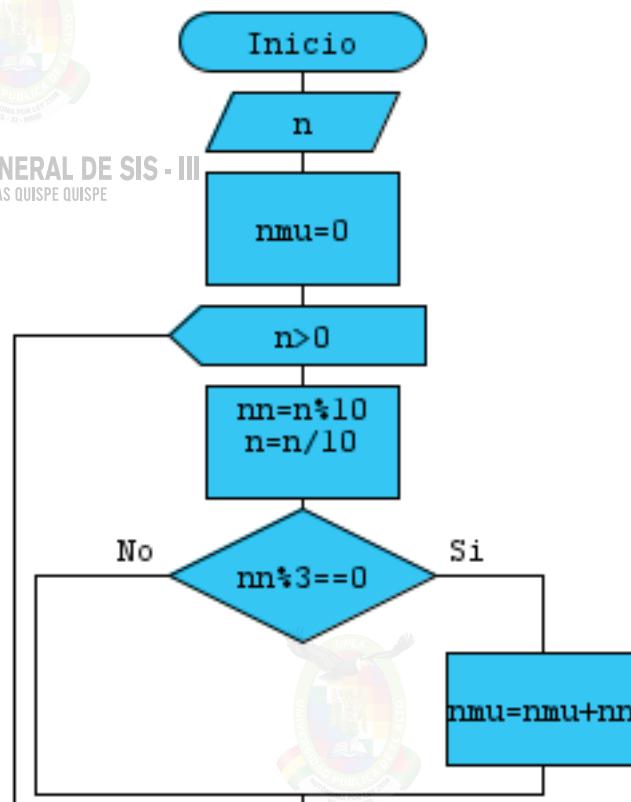
PRÁCTICA GENERAL DE SIS - III
PORFIRIO ELIAS QUISPE QUISPE

SOURCE IN PYTHON

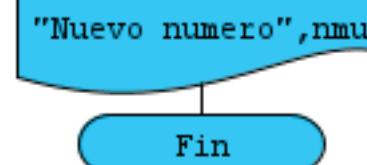
```
nn=0
nmu=0
n = int(input("Ingrese un número N: "))
while n>0:
    nn=n%10
    n=n//10
    if nn%3==0:
        nmu=nmu+nn
    print("nuevo numero es=",nmu)
```

ALGORITHM

PRÁCTICA GENERAL DE SIS - III
PORFIRIO ELIAS QUISPE QUISPE



PRÁCTICA GENERAL DE SIS - III
PORFIRIO ELIAS QUISPE QUISPE



DESK CHECK

PRÁCTICA GENERAL DE SIS - III
PORFIRIO ELIAS QUISPE QUISPE

1 Consola

```
n = 369
Nuevo numero18
```

Prueba de escritorio (PDE's)

Inicio

n	nmu	nn
369	0	9
36	9	6
3	15	3
0	18	



ING
DE
S

UNIVERSIDAD PÚBLICA DE EL ALTO



PROBLEMA

32. Realizar un algoritmo que permita ingresar un número de 4 dígitos donde la suma de sus dígitos sea igual a 15, una vez ingresado desplegar la factorial del digito menor.

SOLUCION ANALISIS

1 ANALIZAR EL PROBLEMA

Especificar resultado: mostrar el menor de un lote se lote 9123 mostrar 1 como menor y su factorial ademas k la suma del lote sea =14

Identificar datos a leer: n

Determinar restricciones: usar datos enteros

Definir proceso: si

while n>0:

 nn=n%10

 n=n/10

cdig=cdig+1

sumadig=sumadig+nn

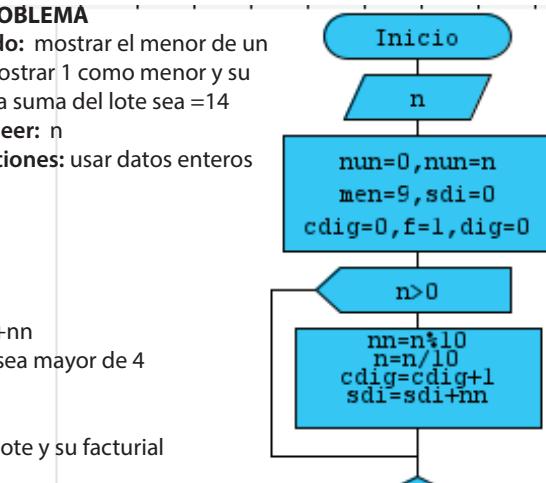
contar digitos k no sea mayor de 4

 if cdig==15:

 cero=cero+1

mostrar menor del lote y su factorial

ALGORITHM



DESK CHECK

n = 1239
factorial=1El menor =1

Prueba de escritorio (PDE's)

Inicio

n	nun	men	sdi	cdig	f	dig	nn	i
1239	0	9	0	0	1	0	9	1
123	1239	3	9	1	1	9	3	2
12	123	2	12	2		3	2	
1	12	1	14	3		3	1	
0	1		15	4		2		
	0					2		
						1		
						1		

SOURCE IN PYTHON

```

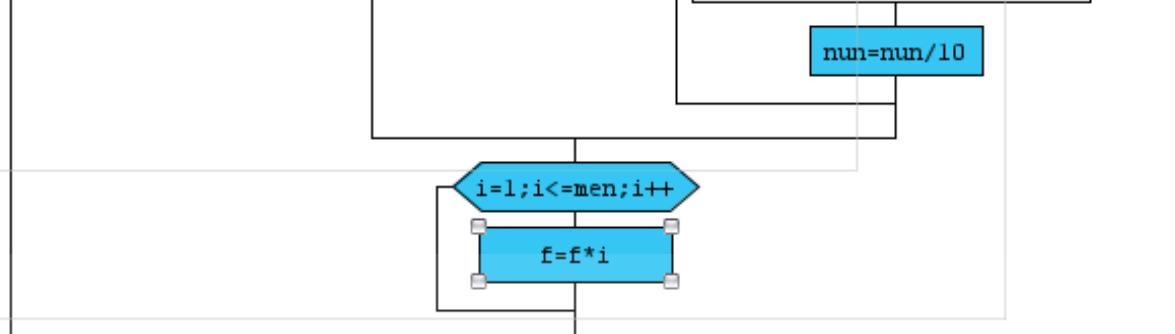
nn=0
dig=0
f=1
sdi=0
men=9
cdig=0
n = int(input("Ingresar un número N:"))
nun=n
while n>0:

    nn=n%10
    n=n/10
    cdig=cdig+1
    sdi=sdi+nn
    if cdig==4:
        if sdi==15:
            while nun>0:
                dig=nun%10
                if dig<men:
                    men=dig
                nun=nun/10
            for i in range(1,men+1):
                f=f*i
        else:
            print("Ingrese un lote de 4 digitos")

print("facturiales=",f)
print("El menor hallado es==",men)
    
```



PRACTICA GENERAL DE SIS - III
PORFIRIO ELIAS QUISPE QUISPE



"factorial=", f,
"El menor =", men

Fin

UNIVERSIDAD PÚBLICA DE EL ALTO



PROBLEMA

33. Dado un número, muestre cuantos ceros tiene el número

SOLUCIÓN

lote=100014

ANÁLISIS

1 ANALIZAR EL PROBLEMA

Especificar resultado: descomposición

Identificar datos a leer: n

Determinar restricciones: usar datos enteros

Definir proceso: while n>0:

nn=n%10

n=n/10

if nn==0:

cero=cero+1

mostrar los ceros contados, cero

2 DISEÑAR DIAGRAMA



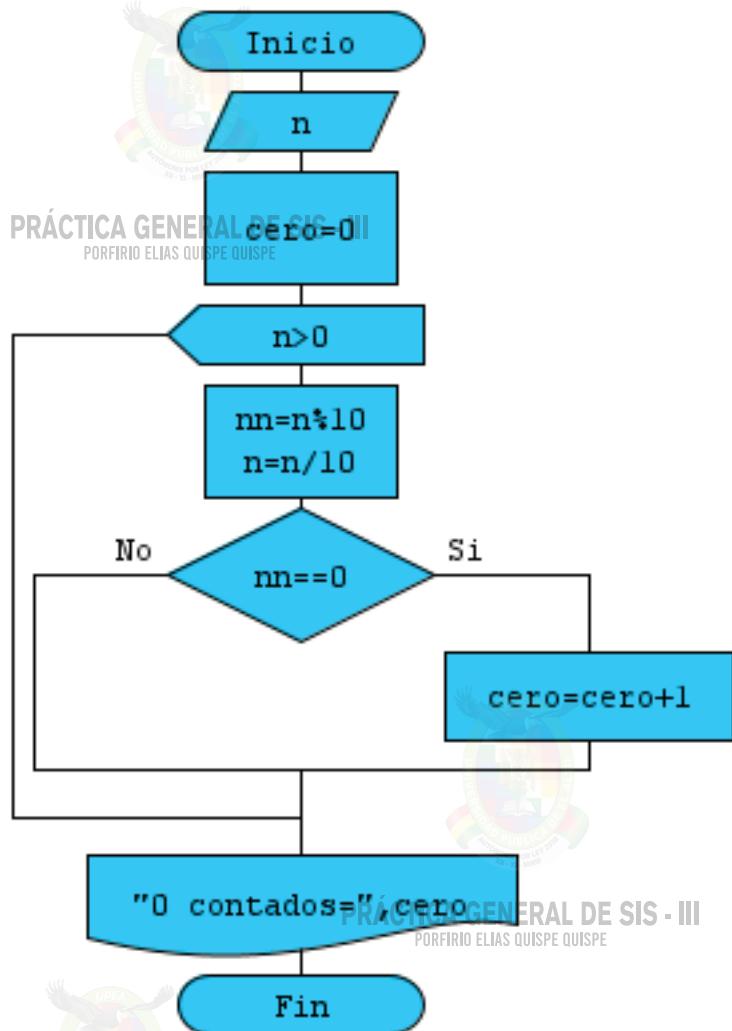
PRÁCTICA GENERAL DE SIS - III

PORFIRIO ELIAS QUISPE QUISPE

SOURCE IN PYTHON

```
nn=0
cero=0
n = int(input("Ingrese un número N: "))
while n>0:
    nn=n%10
    n=n/10
    if nn==0:
        cero=cero+1
print("0 contados=",cero)
```

ALGORITHM



DESK CHECK

Consola		
	n = 100025 0 contados=3	
Prueba de escritorio (PDE's)		
Inicio		
n	cero	nn
100025	0	5
10002	1	2
1000	2	0
100	3	0
10		0
1		1
0		

PROBLEMA

34. Ingresar n números por teclado y mostrar como resultado la cantidad de números positivos, negativos

ANALISIS

SOLUCION

1 ANALIZAR EL PROBLEMA

Especificar resultado: mostrar la sumatoria

Identificar datos a leer: n, x

Determinar restricciones: usar datos enteros y

Definir proceso:

$exp=exp*x$

negativo

$s=s+signo*exp$

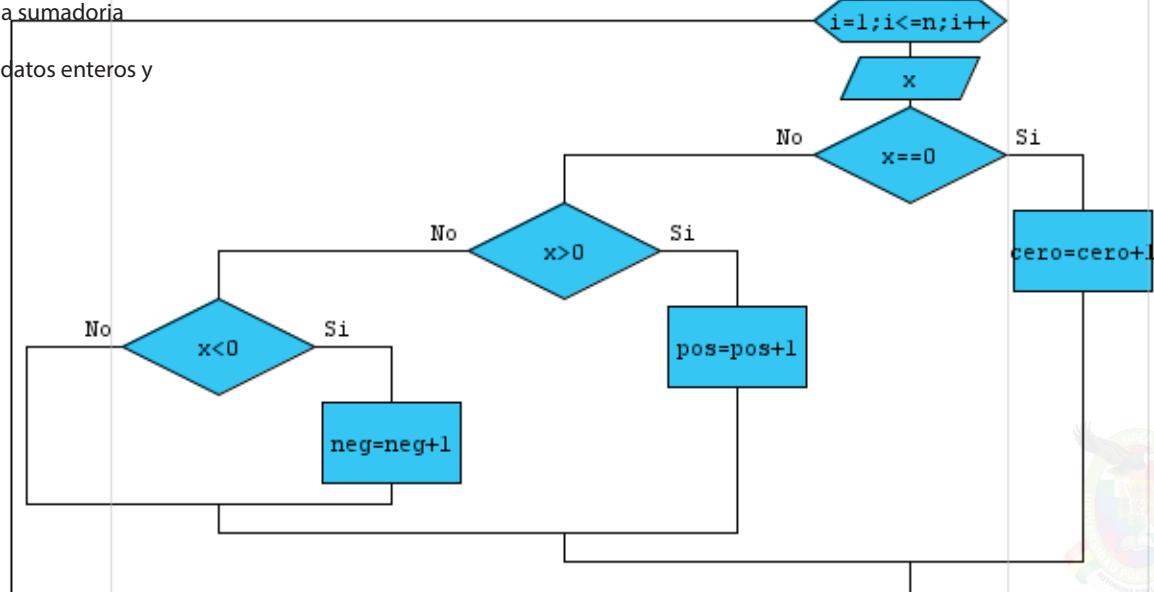
positivo

$s=s+exp$

mostrar sumatoria

2 DISEÑAR DIAGRAMA

ENER DE SIS - III
PORFIRIO ELIAS QUISPE QUISPE



SOURCE IN PYTHON

```

pos=0
neg=0
cero=0
par=0
n = int(input("Ingrese un número N: "))
for i in range(1,n+1):
    x = int(input("Ingrese un número N: "))
    if x==0:
        cero=cero+1
    else:
        if x>0:
            pos=pos+1
        else:
            if x<0:
                neg=neg+1
print("Positivo=",pos)
print("Neutros=",cero)
print("Negativos=",neg)
    
```



RÁPTICA GENERAL DE
PORFIRIO ELIAS QUISPE QUISPE

DESK CHECK

```

n = 5
x = -3
x = -1
x = 0
x = 0
x = 1
negativo=2Positivo=1Neutros=2
    
```

Prueba de escritorio (PDE's)

Inicio |

n	pos	neg	cero	i	x
5	0	0	0	1	-3
	1	1	1	2	-1
	2	2	2	3	0
				4	0
				5	1
				6	

PRÁCTICA GENERAL DE SIS - III
PORFIRIO ELIAS QUISPE QUISPE



INGENIERÍA
DE SISTEMAS

UNIVERSIDAD PÚBLICA DE EL ALTO



PROBLEMA

35. Ingresar N números por teclado e imprimir los divisores de cada número.

SOLUCION

ANALISIS

1 ANALIZAR EL PROBLEMA

Especificar resultado: mostrar serie de divisores

Identificar datos a leer: n, x

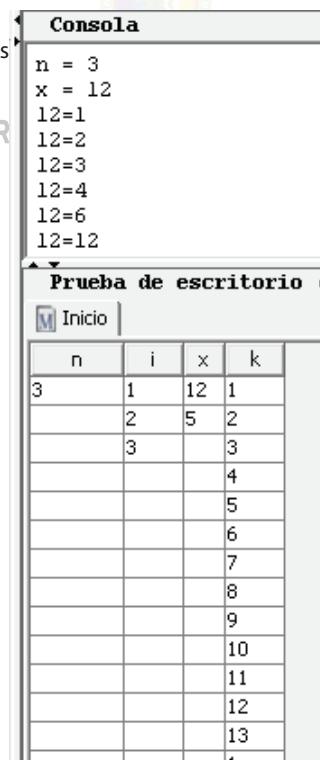
Determinar restricciones: usar datos enteros

Definir proceso: si $x \% k == 0$

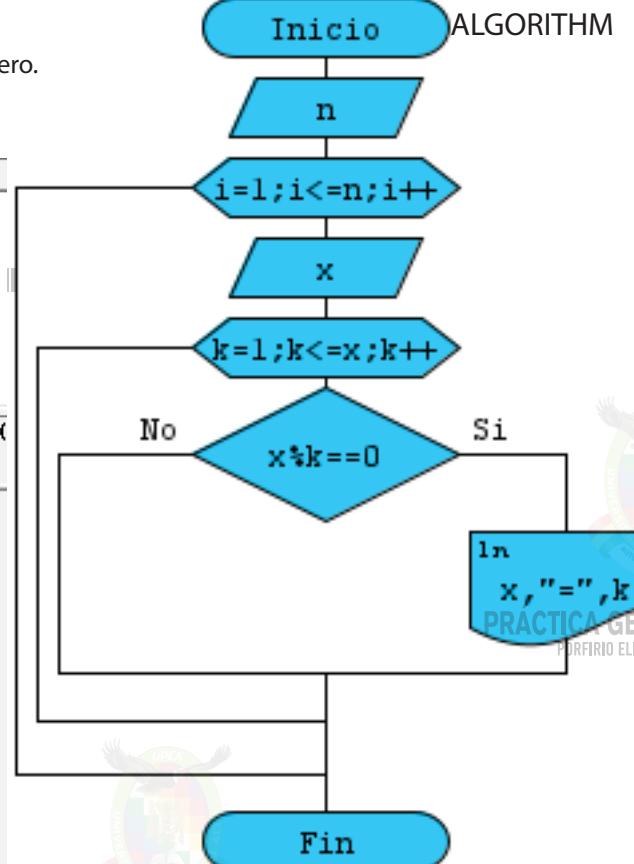
mostrar x,"es divisor de",k

2 DISEÑAR DIAGRAMA

DESK CHECK



ALGORITHM



SOURCE IN PYTHON

PRACTICA GENERAL DE SIS - III

PORFIRIO ELIAS QUISPE QUISPE

```
n = int(input("Ingrese un número N: "))
if n>0:
    for i in range(0,n+1):
        f=1
        x = int(input("Ingrese un número N: "))
        for k in range(1,x+1):
            if x%k==0
                print(x,"=",k)
```

36. Dado n, muestre la suma de los n primeros números pares.

PROBLEMA

SOLUCION

ANALISIS

1 ANALIZAR EL PROBLEMA

Especificar resultado: mostrar serie de numeros pares 2,4,6,8,10,12,14,16,18,20.....

Identificar datos a leer: n,

Determinar restricciones: usar datos enteros

Definir proceso: pa =0

pa=pa+2

mostrar pa

2 DISEÑAR DIAGRAMA



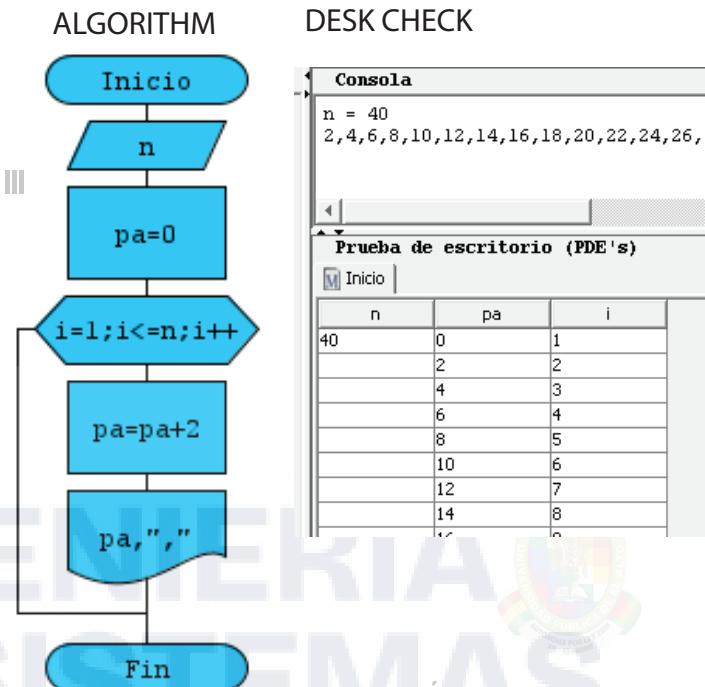
PRACTICA GENERAL DE SIS - III

PORFIRIO ELIAS QUISPE QUISPE

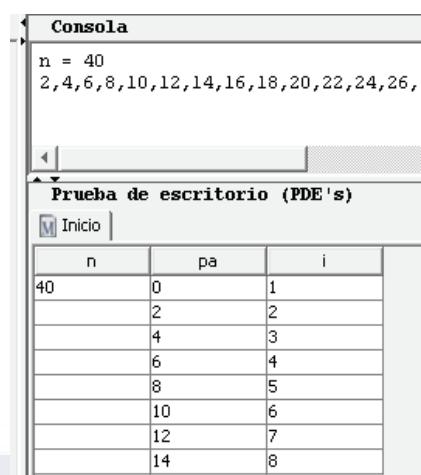
PRÁCTICA GENERAL DE SIS - III

PORFIRIO ELIAS QUISPE QUISPE

ALGORITHM



DESK CHECK



SOURCE IN PYTHON

```
n = int(input("Ingrese un número N: "))
pa=0
if n>0:
    for i in range(0,n+1):
        pa=pa+2
        print(pa,"")
```

PROBLEMA

37. Dado un numero N, desplegar los primeros números naturales descendentes. NOTA: Verificar que se ingrese un numero positivo.

ANALISIS si $n=4$ 4,3,2,1

SOLUCION

1 ANALIZAR EL PROBLEMA

Especificar resultado: mostrar la serie

Identificar datos a leer: n

Determinar restricciones: usar datos enteros

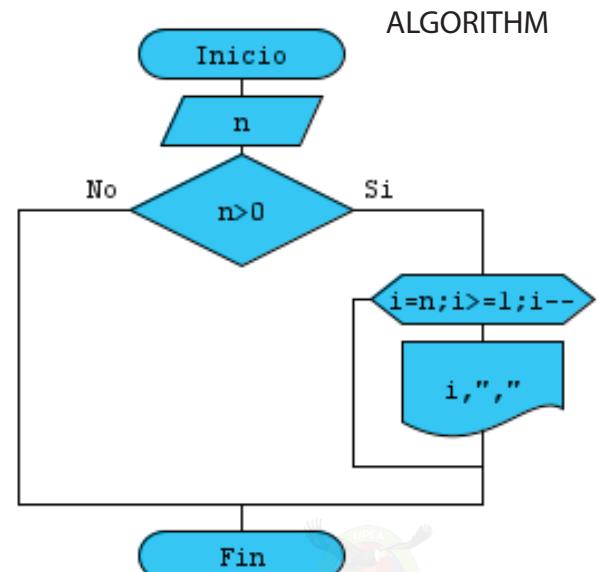
Definir proceso: n-contador

mostrar serie n-contador

PRÁCTICA GENERAL DE SIS - III
PORFIRIO ELIAS QUISPE QUISPE

DESK CHECK

CONSOLA	
n = 10	
10,9,8,7,6,5,4,3,2,1,	
Prueba de escritorio (P)	
M	Inicio
n	i
10	10
	9
	8
	7
	6
	5
	4
	3
	2
	1
	0



2 DISEÑAR DIAGRAMA

SOURCE IN PYTHON

```

n = int(input("Ingrese un número N: "))
if n>0:
    for i in range(0,n+1):
        print(n-i,"")
    
```

GERAL DE SIS - III
UISPE QUISPE

PROBLEMA

38. Dado un número N, muestra como resultado su factorial del número ingresado.

ANALISIS si $n=4$ generar par 4

si $x=3$ calularel factorial
de x 3 3x2x1= 6

1 ANALIZAR EL PROBLEMA

Especificar resultado: mostrar la sumatoria

Identificar datos a leer: n, x

Determinar restricciones: usar datos enteros

Definir proceso:

$f=f^*$ contador

mostrar factorial f

2 DISEÑAR DIAGRAMA

PRÁCTICA GENERAL DE SIS - III
PORFIRIO ELIAS QUISPE QUISPE

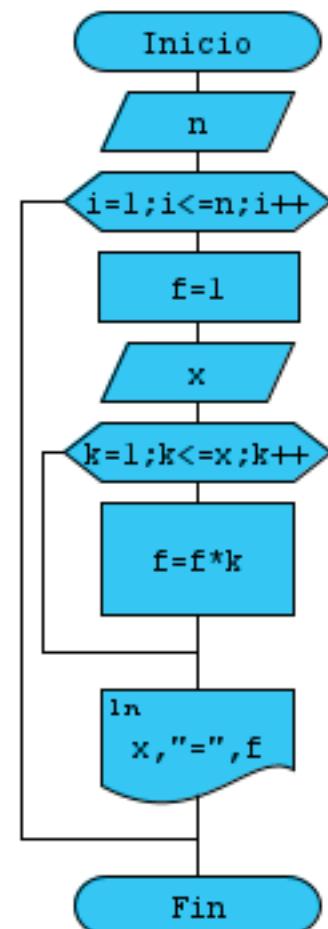
DESK CHECK

CONSOLA	
n = 3	
x = 5	
5=120	
x = 7	
7=5040	
x = 1	
1=1	

PRÁCTICA GENERAL DE SIS - III
PORFIRIO ELIAS QUISPE QUISPE

CONSOLA	
Prueba de escritorio (PD)	
M	Inicio
n	i
3	1
	2
	3
	4
	5
	6
	7
	8
	9
	10
	11
	12
	13
	14
	15
	16
	17
	18
	19
	20
	21
	22
	23
	24
	25
	26
	27
	28
	29
	30
	31
	32
	33
	34
	35
	36
	37
	38
	39
	40
	41
	42
	43
	44
	45
	46
	47
	48
	49
	50
	51
	52
	53
	54
	55
	56
	57
	58
	59
	60
	61
	62
	63
	64
	65
	66
	67
	68
	69
	70
	71
	72
	73
	74
	75
	76
	77
	78
	79
	80
	81
	82
	83
	84
	85
	86
	87
	88
	89
	90
	91
	92
	93
	94
	95
	96
	97
	98
	99
	100

ALGORITM



SOURCE IN PYTHON

```

n = int(input("Ingrese un número N: "))
if n>0:
    for i in range(0,n+1):
        f=1
        x = int(input("Ingrese un número N: "))
        for k in range(1,x+1):
            f=f*k
        print(x,"=",f)
    
```

ALGORITM



PROBLEMA

39. Pedir un lote de números hasta que ingrese un número múltiplo de cinco, de los cuales hallar la suma de

ANALISIS

SOLUCION

1 ANALIZAR EL PROBLEMA

Especificar resultado: del lote de numero descomponer

Identificar datos a leer: nlot

Determinar restricciones: usar datos enteros

Definir proceso: nn=nMod 10 par obtener el digito

n=n/10 para obtener nuevo n

si nn% 2o ==0 para vericar si es par

sumar pares obtenidos

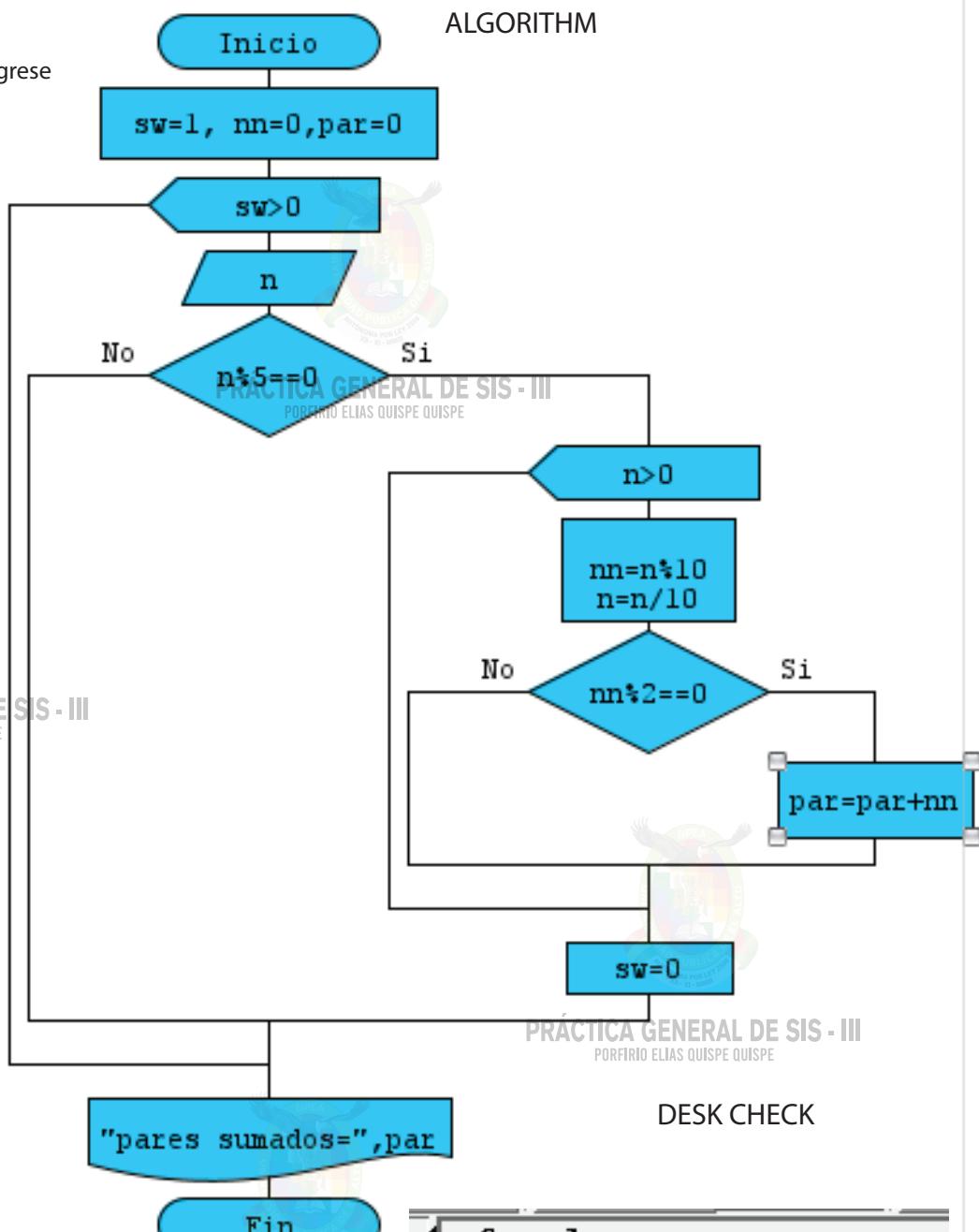
mostrar mostrar multiplo del nuevo numero compuesto

2 DISEÑAR DIAGRAMA

PRÁCTICA GENERAL DE SIS - III
PORFIRIO ELIAS QUISPE QUISPE

SOURCE IN PYTHON

```
nn=0
sw=1
nn=0
par=0
while sw>0:
    n = int(input("Ingrese un número N: "))
    if n%5==0:
        while n>0:
            nn=n%10
            n=n/10
            if nn%2==0:
                par=par+nn
            sw=0
    print("pares sumados",par)
```



PRÁCTICA GENERAL DE SIS - III
PORFIRIO ELIAS QUISPE QUISPE

Prueba de escritorio (PDE's)

Inicio	SW	NN	PAR	N
	1	0	0	450
	0	0	0	45
	5	4	4	
	4			0

INGENIERÍA
DE SISTEMAS
UNIVERSIDAD PÚBLICA DE EL ALTO



PROBLEMA

40. Dado un lote de N números, determinar cuántos múltiplos de m hay en el lote de números y desplegar el resultado. Solicitar M por pantalla.

ANALISIS SOLUCION

1 ANALIZAR EL PROBLEMA

Especificar resultado: del lote de numero descomponer para cada numero y pider su multiplo

Identificar datos a leer: n, multiplo

Determinar restricciones: usar datos enteros

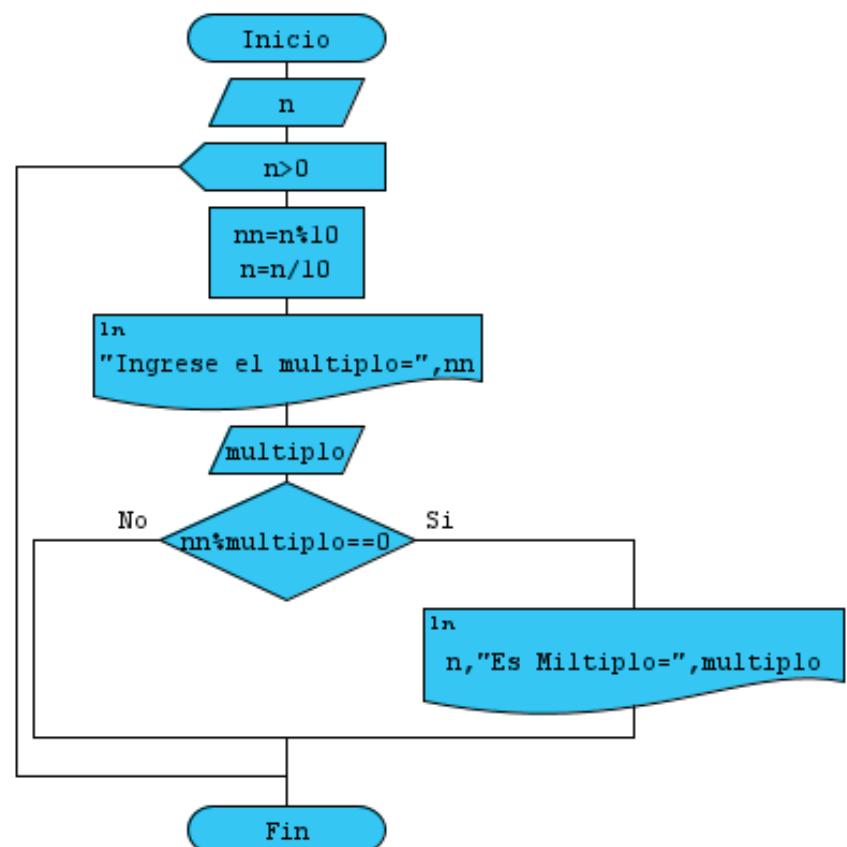
Definir proceso: nn=nMod 10 par obtener el digito

n=n/10 para obtener nuevo n

si nn% multiplo ==0

mostrar mostrar multiplo del nuevo numero compuesto

2 DISEÑAR DIAGRAMA



DESK CHECK



SOURCE IN PYTHON

```

nn=0
n = int(input("Ingrese un número N: "))
while n>0:
    nn=n%10
    n=n//10
    print("Ingrese multiplo =",nn)
    multiplo = int(input())
    if nn%multiplo==0:
        print(n,"Es multiplo de =",multiplo)
    
```

PRÁCTICA GENERAL DE SIS - III

PORFIRIO ELIAS QUISPE QUISPE



PRÁCTICA GENERAL DE SIS - III
PORFIRIO ELIAS QUISPE QUISPE

**INGENIERÍA
DE SISTEMAS**

UNIVERSIDAD PÚBLICA DE EL ALTO

Consola		
n = 1452 Ingrese el multiplo=2 multiplo = 2 1452Es Multiplo=2 Ingrese el multiplo=5 multiplo = 1 1452Es Multiplo=1 Ingrese el multiplo=4 multiplo = 2 1452Es Multiplo=2 Ingrese el multiplo=1 multiplo = 5		
Prueba de escritorio (PDE's)		
Inicio		
n	nn	multiplo
1452	2	2
145	5	1
14	4	2
1	1	5
0		



PRÁCTICA GENERAL DE SIS - III

NOMBRE: PORFIRIO ELIAS QUISPE QUISPE

PARALELO: "1A"

TURNO : MAÑANA

DOCENTE: ING. YOLANDA ESCOBAR

GESTION I-2024



**INGENIERÍA
DE SISTEMAS**
UNIVERSIDAD PÚBLICA DE EL ALTO