

## 1.- Pseudocódigo y Diagrama de Flujo

Escribe en pseudocódigo el algoritmo para calcular la suma de todos los números pares entre 1 y 50.

- \* Definir variables :  $n = \text{numero}$   $s = \text{suma}$
- \* Datos de entrada : numero (50)
- \* Salida : Suma

Inicio

$s = 0$

$i = 1$

$n = 0$

Mientras  $i <= 50$  hacer

$s = s + 1$

$i = i + n$

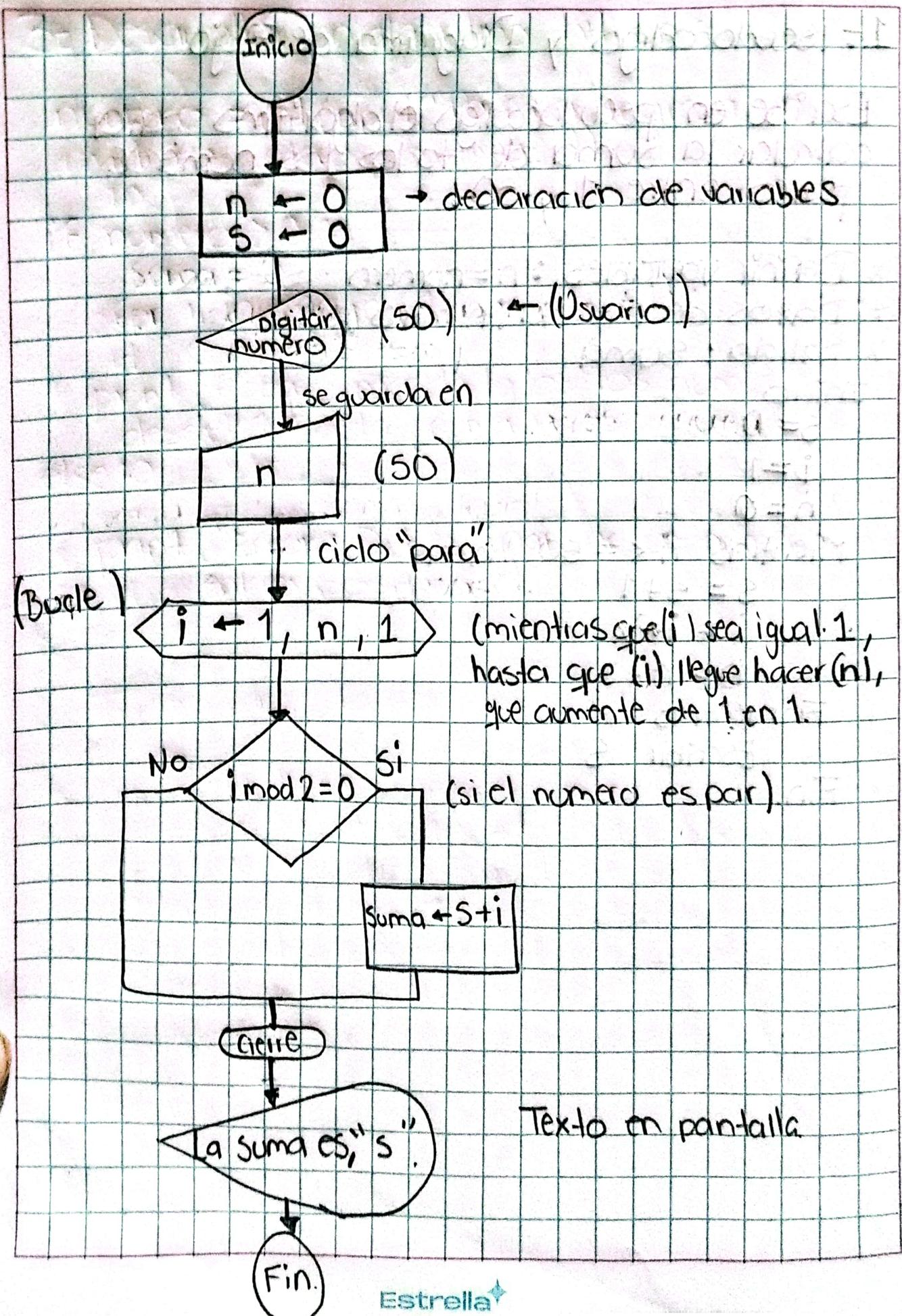
$n = n + 1$

Finmientras

Escribir  $s$

Fin.

# Diagrama de Flujo.



## 2. Introducción a Python : Programa "Hola Mundo"

Crea un programa en Python que solicite el nombre del usuario y luego imprima un mensaje de bienvenida personalizado.

```
def saludo(nombre):  
    print("¡Hola, {}, ¡Bienvenido!".format(nombre))  
    print("Muchas gracias por participar.")  
    print("Espero que la pases bien.")  
  
    return None
```

```
nombre_usuario = input("Introduzca su nombre: ")  
saludo(nombre_usuario)
```

### 3. Desarrollar un algoritmo Python:

```
numero = int(input("Por favor, ingresa un  
número entero positivo:"))

if numero > 0:  
    suma = 0  
    for i in range(1, numero + 1):  
        suma += i  
    print(f"La suma de todos los números  
desde 1 hasta {numero} es: {suma}")  
else:  
    print("Error: Debes ingresar un número  
entero positivo.")
```

#### 4.-Diseño y Desarrollo de Algoritmos y Esquemas de Base de Datos.

Diseñar el esquema de una base de datos que incluya, al menos, tres tipos de objetos. El diseño se realizará únicamente como un bosquejo en un archivo de texto, sin necesidad de implementarlo en un lenguaje de programación.

- **Propósito:** Para gestionar la información de una Universidad donde se lleve el registro de cada Alumno (asignaturas, profesores, oferta académica)
- **Incluir al menos tres tipos de objetos en el bosquejo.**

Definir las tablas principales que almacenarán los datos.

En el CREATE TABLE se puede especificar para cada atributo las restricciones a las que estará sujeto.

- Es posible definir todas las restricciones mencionadas, salvo las generales.
- Llave primaria: PRIMARY KEY
- Valor único: UNIQUE
- Valor nulo: NULL / NOT NULL
- Llave foránea: REFERENCES
- Corresponde a los valores de la llave primaria de otra tabla.

```

- Create DATABASE universidad;
- use universidad;
- drop TABLE asignaturas;

- create table asignaturas (
- numero int(10) not NULL PRIMARY key,
- matricula_estudiante VARCHAR (12) not null,
- calificacion INT (2) not NULL,
- programa VARCHAR (30),
- requisitos VARCHAR(30) not null,
- creditos INT (3) not null,
- titulo VARCHAR (30)
- );
- create table profesores(
- id_profesor INT (10) not null PRIMARY key,
- nombre VARCHAR (30) not null,
- departamento VARCHAR (30) not null,
- titulo VARCHAR (30) not null
- );
- create table oferta(
- numero_asignatura INT (10) not null PRIMARY
key,
- año INT (4) not null,
```

Cada objeto tiene sus propios atributos y su tipo de dato, los cuales nos ayudaran a relacionar la información de cada objeto en su tabla.

#### Tipos de datos

##### Numéricos

- Integer: números enteros
- Float: números reales
- Number: números enteros y reales (operaciones matemáticas)

##### Cadenas de caracteres

- Char: cadenas de longitud fija
- Varchar: cadenas de longitud variable

##### Fecha y Tiempo

- Fecha
- Date

- numero\_seccion INT (4) not NULL,
- horario DATETIME not null,
- aulas INT (4) not null,
- semestre INT (2) not null,
- profesor VARCHAR (50) not null
- );
  
- create TABLE estudiantes (
- id\_estudiantes INT (12) not null PRIMARY key,
- nombre VARCHAR (30) not null,
- programa VARCHAR (30) not null
- );
  
- use universidad;
- select \* from asignaturas;
- select \* from estudiantes;
- select \* from oferta;
- select\* from profesores;

**La obtención de datos por cada objeto(tabla)**

- **Crear vistas para facilitar la consulta de datos.** (Muestra los datos solamente requeridos)  
Forma general del bloque S-F-W

SELECT lista de atributos Atributos requeridos en el resultado  
FROM listas de tablas Lista de tablas a combinar  
WHERE condición Conjunto de condiciones simples de igualdad entre llaves primarias y foráneas.

- use universidad;
- create view vista\_calificacion\_arriba\_de\_6 as
- select \* from materias
- where calificacion is null;
- select \* from vista\_calificacion\_arriba\_de\_6;
  
- use universidad;
- create view vista\_materias\_repetidas as
- select \*from materias
- where titulo is null;
- select \* from vista\_materias\_repetidas;
  
- use universidad;
- create view calificacion\_aprobatoria as
- select \*from materias
- where calificacion is null;
- select \* from vista\_materias\_repetidas;
  
- use universidad;
- create view alumnos\_en\_contabilidad as
- select \*from alumnos
- where programa is null;
- select \* from alumnos\_en\_contabilidad;

- use universidad;
- create view alumnos\_en\_sistemas as
- select \*from alumnos
- where programa is null;
- select \* from alumnos\_en\_sistemas;
  
- use universidad;
- create view alumnos\_en\_derecho as
- select \*from alumnos
- where programa is null;
- select \* from alumnos\_en\_derecho;

**Esbozar procedimientos que realicen operaciones o cálculos con los datos.**

Alter table permite realizar a los atributos existentes los siguientes cambios:

Agregar un nuevo atributo (ADD)

Renombrar (RENAME)

Modificar (MODIFY)

Suprimir (DROP)

- use universidad;
- ALTER table asignaturas rename materias;
- use universidad;
- alter table materias add correo varchar(30);
- use universidad;
- select \* from materias;
  
- use universidad;
- alter table profesores rename maestros;
- use universidad;
- alter table maestros add telefono varchar(20);
- use universidad;
- select \* from maestros;
  
- use universidad;
- alter table estudiantes rename alumnos;
- use universidad;
- alter table alumnos add direccion varchar(50);
- use universidad;
- select \* from alumnos;
  
- use universidad;
- alter table oferta drop año;
- use universidad;
- ALTER TABLE oferta MODIFY aulas int(10);
- use universidad;
- select \* from oferta;

The image shows a Windows desktop environment with a purple taskbar at the bottom. On the taskbar, there are icons for File Explorer, Google Chrome, Autodesk Maya 2023 Update 3.1, and the Python IDLE Shell 3.13.2 application. The IDLE window is open and displays a Python script named 'NOMBRE Y SALUDO\_2.py' located at 'C:/Users/Ana Ceja/Desktop/NOMBRE Y SALUDO\_2.py (3.13.2)'. The script contains the following code:

```
def saludo(nombre):
    print("¡Hola, {}, ¡Bienvenido!".format(nombre))
    print("Muchas gracias por participar.")
    print("Espero que la pases bien.")

    return None

nombre_usuario = input("Introduzca su nombre: ")

saludo(nombre_usuario)
```

The output pane of the IDLE window shows the execution results:

```
more information.

; Y SALUDO.py =====

line 10, in <module>
    )
line 2, in saludo
    )

Y SALUDO_2.py =====
```

The status bar at the bottom right of the IDLE window indicates 'Ln: 19 Col: 25' and the time '09:51 p.m.'

A screenshot of a Windows desktop environment. In the center, there is a code editor window titled "ACT3.py - C:/Users/Ana Ceja/Desktop/ACT3.py (3.13.2)". The code is written in Python and calculates the sum of all integers from 1 to a user-specified positive integer. A terminal window is open to the right, displaying the output of the program. The desktop background is purple, and various icons for applications like Google Chrome, Microsoft Edge, and file explorers are visible on the left.

```
numero = int(input("Por favor, ingresa un número entero positivo: "))

if numero > 0:
    suma = 0
    for i in range(1, numero + 1):
        suma += i
    print(f"La suma de todos los números desde 1 hasta {numero} es: {suma}")
else:

    print("Error: Debes ingresar un número entero positivo.")
```

SC v.1942 64 bit ( formation.  
=====

The image shows a Windows desktop environment. On the left, there's a vertical taskbar with icons for Papelera de reciclaje, Google Chrome, Autodesk Maya 2023 Update 3.1, DELTA Pro, Este equipo, VLC media player, Adobe Acrobat, Word 2016, and Maya 2023. The desktop background is purple.

The main window is titled "IDLE Shell 3.13.2". It displays Python code and its execution output:

```
Python 3.13.2 (tags/v3.13.2:4f8bb39, Feb 4 2025, 15:23:48) [MSC v.1942 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>> print("Hola mundo")
Hola mundo
=====
RESTART: C:/Users/Ana Ceja/Desktop/NOMBRE Y SALUDO.py =====
Introduzca su nombre: Carlos Alberto
Traceback (most recent call last):
  File "C:/Users/Ana Ceja/Desktop/NOMBRE Y SALUDO.py", line 10, in <module>
    saludo(nombre_usuario)
  File "C:/Users/Ana Ceja/Desktop/NOMBRE Y SALUDO.py", line 2, in saludo
    print("¡Hola, {nombre}, ¡Bienvenido!".format(nombre))
KeyError: 'nombre'

===== RESTART: C:/Users/Ana Ceja/Desktop/NOMBRE Y SALUDO_2.py =====
Introduzca su nombre: CARLOS ALBERTO
¡Hola, CARLOS ALBERTO, ¡Bienvenido!
Muchas gracias por participar.
Espero que la pases bien.

>>> |
```

The status bar at the bottom right shows "Ln: 20 Col: 0" and the time "09:51 p.m.". A tooltip "Terminado en pantalla" is visible near the bottom center.

The screenshot shows a Windows desktop environment with a purple background. On the left, there is a vertical taskbar with various icons for applications like Google Chrome, Microsoft Word, and Maya 2023. Two windows are open in the center: a code editor window titled "ACT3.py - C:/Users/Ana Ceja/Desktop/ACT3.py (3.13.2)" and a Python IDLE Shell window titled "IDLE Shell 3.13.2".

The code editor window contains the following Python script:

```
numero = int(input("Por favor, ingresa un número entero positivo: "))

if numero > 0:
    suma = 0
    for i in range(1, numero + 1):
        suma += i
    print(f"La suma de todos los números desde 1 hasta {numero} es: {suma}")
else:
    print("Error: Debes ingresar un número entero positivo.")
```

The Python IDLE Shell window shows the execution of the script. It first prompts the user for input, then calculates and prints the sum of all integers from 1 to the input number. It also handles the case where a non-positive integer is entered.

```
Python 3.13.2 (tags/v3.13.2:4f8bb39, Feb  4 2025, 15:23:48) [MSC v.1942 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

=====
>>> RESTART: C:/Users/Ana Ceja/Desktop/ACT3.py =====
Por favor, ingresa un número entero positivo: 15
La suma de todos los números desde 1 hasta 15 es: 120

=====
>>> RESTART: C:/Users/Ana Ceja/Desktop/ACT3.py =====
Por favor, ingresa un número entero positivo: -3
Error: Debes ingresar un número entero positivo.
```

Actividad en clase. 1/marzo/2025

- \* Algoritmo para encontrar una palabra, y decir si es falso o verdadero.

Definir palabra  
palabra = "Caballo"

Función adivinar palabra.

def adivinar\_palabra():

Ingresar una palabra, pedir al usuario:

palabra\_usuario = input("ingresa una palabra: ")  
lower()

Validación de palabra.

if palabra\_usuario == palabra: print("Verdadera")

print("Falso.")

else:

print("Falso.")

Se llama a la función  
adivinar\_palabra()