**1. Introducción al Desarrollo de Software**

**¿Qué es el Desarrollo de Software?**

El desarrollo de software es el proceso de crear, diseñar, implementar y mantener aplicaciones y sistemas de software. Implica varias etapas que aseguran que el software cumpla con los requisitos del usuario y funcione correctamente en diversos entornos. El objetivo es construir software que sea funcional, eficiente y fácil de mantener.

**Fases del Ciclo de Vida del Desarrollo de Software (SDLC)**

1. **Análisis de Requisitos**:
   * **Objetivo**: Entender y documentar qué necesita el usuario y qué debe hacer el software.
   * **Actividades**: Reuniones con los stakeholders, análisis de necesidades, documentación de requisitos.
2. **Diseño**:
   * **Objetivo**: Crear un plan detallado de cómo funcionará el software y cómo se verá.
   * **Actividades**: Diseño de la arquitectura del software, diagramas de flujo, maquetas de la interfaz de usuario.
3. **Implementación**:
   * **Objetivo**: Programar el software según el diseño y los requisitos.
   * **Actividades**: Codificación, integración de módulos, desarrollo de funciones.
4. **Pruebas**:
   * **Objetivo**: Verificar que el software funcione como se espera y corregir errores.
   * **Actividades**: Pruebas unitarias, pruebas de integración, pruebas de sistema, pruebas de aceptación.
5. **Despliegue**:
   * **Objetivo**: Poner el software en producción para que los usuarios finales lo utilicen.
   * **Actividades**: Instalación en el entorno de producción, configuración del entorno, formación de usuarios.
6. **Mantenimiento**:
   * **Objetivo**: Realizar ajustes y correcciones para mejorar el software y adaptarse a cambios.
   * **Actividades**: Corrección de errores, actualizaciones, soporte al usuario.

***Herramientas y Lenguajes de Programación Comunes***

* **Python**: Popular por su simplicidad y versatilidad, ideal para desarrollo web, análisis de datos, y automatización.
* **Java**: Usado en aplicaciones empresariales, desarrollo web, y aplicaciones móviles (Android).
* **C#**: Utilizado principalmente en el desarrollo de aplicaciones para el ecosistema de Microsoft, como aplicaciones Windows y videojuegos en Unity.
* **PHP**: Ampliamente usado para desarrollo web del lado del servidor.

**2. Definición de Requisitos**

**Proyecto Sugerido: Calculadora Básica**

**Requisitos del Proyecto**:

1. **Funciones Básicas**:
   * Suma
   * Resta
   * Multiplicación
   * División
2. **Interfaz de Usuario**:
   * Un campo de entrada para los números.
   * Botones para cada operación aritmética.
   * Un botón para calcular el resultado.
   * Un área para mostrar el resultado.
3. **Comportamiento**:
   * Debe permitir la entrada de números y operaciones.
   * Debe manejar errores como la división por cero.

***3. Diseño***

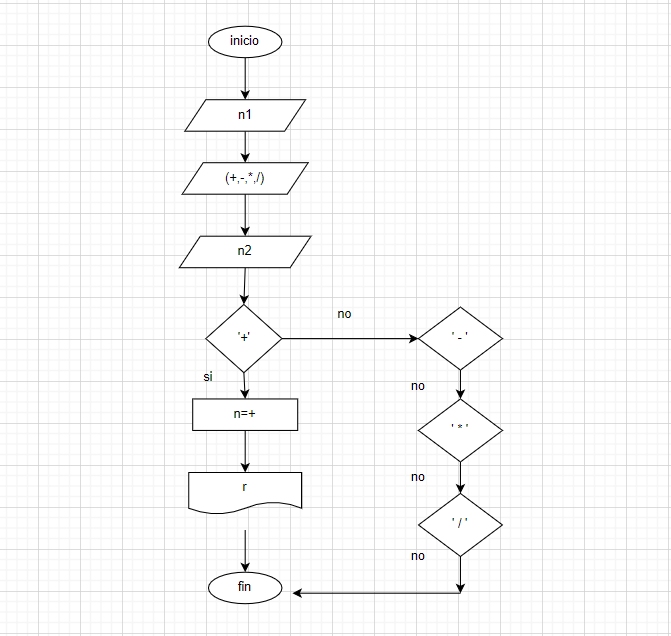
**Arquitectura del Software**

* **Interfaz de Usuario (UI)**: La UI tendrá botones y campos de entrada. Se diseñará de manera que sea intuitiva y fácil de usar.
* **Lógica de Negocio**: Implementará las operaciones aritméticas y gestionará la interacción entre la UI y los cálculos.

**Diagramas y Maquetas**

* **Diagrama de Flujo**: Representará la secuencia de operaciones desde la entrada hasta la salida.
* **Maquetas**: Bocetos de cómo se verá la calculadora, incluyendo la disposición de botones y campos.

**Ejemplo de Diagrama de Flujo**:



**Ejemplo de Maqueta**:

-------------------

| [Número 1] |

| [Operador] |

| [Número 2] |

| [= Resultado] |

-------------------

***4. Implementación***

**Código de Ejemplo en Python:**

python

# Calculadora básica en Python

def sumar(x, y):

    return x + y

def restar(x, y):

    return x - y

def multiplicar(x, y):

    return x \* y

def dividir(x, y):

    if y == 0:

        return "Error: División por cero"

    return x / y

def calculadora():

    print("Ingrese el primer número: ")

    num1 = float(input())

    print("Ingrese el operador (+, -, \*, /): ")

    operador = input()

    print("Ingrese el segundo número: ")

    num2 = float(input())

    if operador == "+":

        print(f"Resultado: {sumar(num1, num2)}")

    elif operador == "-":

        print(f"Resultado: {restar(num1, num2)}")

    elif operador == "\*":

        print(f"Resultado: {multiplicar(num1, num2)}")

    elif operador == "/":

        print(f"Resultado: {dividir(num1, num2)}")

    else:

        print("Operador no válido")

# Ejecutar la calculadora

calculadora()

***5. Pruebas***

**Tipos de Pruebas:**

1. **Pruebas Unitarias**: Verificar cada función matemática individualmente.
2. **Pruebas de Integración**: Asegurarse de que las funciones trabajen juntas correctamente.
3. **Pruebas de Usabilidad**: Probar la interfaz para asegurarse de que sea intuitiva.
4. **Pruebas de Validación**: Comprobar que el software maneje errores como la división por cero.