A picture containing logo

Description automatically generated

|  |  |  |
| --- | --- | --- |
| A picture containing text, clipart  Description automatically generated | **I**nstituto **P**olitécnico **N**acional  Escuela Superior de Cómputo  **ESCOM** |  |

**DISEÑO DE SISTEMAS DISTRIBUIDOS**

**Tarea 1: Cálculo distribuido de PI**

Maya Martínez Alonso Rubén

**Fecha de realización**

2 de marzo del 2021

**Profesor**

Pineda Guerrero Carlos

**Grupo**

4CV1

**Descripción de la practica.**

En esta practica se hace la implementación de la serie de Gregory-Leibniz usando 5 nodos, de los cuales 1 será el servidor y los otros 4 realizaran una parte del calculo de esta serie para aproximar pi.

**Desarrollo de la practica.**

Text

Description automatically generated

Imagen 1: Compilación y ejecución del nodo 0.

Como podemos ver en la Imagen 1 el nodo 0 va a recibir todos los valores calculados en los otros nodos, estos valores los suma y los guarda en una variable, esa variable el final de la ejecución de los demás nodos será desplegada y se aproxima a pi.



Imagen 2: Ejecución del nodo 1.



Imagen 3: Ejecución del nodo 2.



Imagen 4: Ejecución del nodo 3.



Imagen 5: Ejecución del nodo 4.

Como podemos ver en las Imágenes 2, 3, 4 y 5 se realiza el calculo de una porción de la serie, dependiendo del nodo se determina que porción será y si el termino resultante es positivo o negativo. Estos valores los recibe el nodo 0 para hacer los cálculos antes mencionados. En este caso ya no es necesaria la compilación ya que se hizo previamente en la Imagen 1.

Graphical user interface, text

Description automatically generated  
Imagen 5: Vista general de la ejecución del programa.

**Conclusiones.**

Esta practica sirvió de introducción para entender como funcionan los sistemas distribuidos, además de que observamos que no cualquier algoritmo sirve para hacer este tipo de cálculos. La serie de Gregory-Leibniz es un perfecto ejemplo y por intuición nos damos cuenta que un algoritmo puede implementarse de manera distribuida si en ningún momento para obtener un resultado parcial se necesita conocer un resultado previo. En el caso de esta serie pudimos usarla de manera distribuida porque sus términos eran independientes, bastaba con saber en que numero de la secuencia iniciar y en cual terminar, y eso es fácil sabiendo el numero de nodo que es el programa actual que se esta ejecutando.