Trabajo Práctico N° 1

Parte 1

Comunicación y Sincronismo

Grupo: M3

Integrantes:

- • Arab, Santiago Martin. DNI: 46500280
- • Barcia, Matias Alejandro. DNI: 43975241
- • Monteros, Matias Javier. DNI: 40886497
- • Goicoechea, Tomas Lucas. DNI: 42038019

Link al repositorio del Colab

El código y el archivo del cuaderno de Colab (.ipynb) relacionado a este Trabajo Práctico se pueden encontrar en el siguiente repositorio Colab: https://colab.research.google.com/drive/1_QtGenUg0zT0kMb_DK1e6_LlhATE7LeR?usp=sharing

Conclusiones:

En conclusión, la resolución de problemas de concurrencia en Java, C++ y Python presentó distintos niveles de complejidad. La implementación en **Java** la sincronización fue más sencilla gracias a las herramientas nativas como `Semaphore`. Sin embargo, la gestión de los hilos y el control de recursos requerirá un manejo cuidadoso para evitar problemas de *interbloqueo*. Por otro lado, la implementación en **Python** resultó ser la más sencilla, gracias a su simplicidad en la gestión de hilos con `threading`. Sin embargo, la eficiencia fue limitada por el *Global Interpreter Lock* (GIL), lo que lo hace menos adecuado para aplicaciones de concurrencia intensiva. Al final en **C++**, fue la más desafiante, ya que el lenguaje requiere un control manual de la concurrencia utilizando `mutex` necesitando mayor atención a los detalles para evitar errores como *condiciones de carrera*. No obstante, C++ fue el lenguaje más eficiente en términos de ejecución y uso de recursos.