

1 Describa de manera teórica los siguientes conceptos: SISD, SIMD, MISD y MIMD. Indique además que lenguajes aplican a estos.

1. **SISD (Una Instrucción, Un Dato):** Este es el modelo tradicional donde un solo procesador realiza una tarea sobre un conjunto de datos. Un solo procesador ejecuta una instrucción a la vez sobre un conjunto de datos. Este es el modelo clásico de ejecución secuencial en la computación tradicional.

Los lenguajes de programación que funcionan así cuando se usan en un solo procesador son: **C, Java y Python.**

2. **SIMD (Una Instrucción, Múltiples Datos):** En este modelo, una sola instrucción se aplica a muchos datos a la vez.(unidades de procesamiento gráfico) o en procesadores vectoriales. Se usa en arquitecturas de procesamiento paralelo donde una misma operación se realiza en varios datos a la vez, como en GPUs, que son buenas para hacer muchas cosas similares al mismo tiempo.

Lenguajes como **CUDA (para programación en GPU) y OpenCL** son ejemplos de lenguajes que se aplican a SIMD.

3. **MISD (Múltiples Instrucciones, Un Dato):** Varios procesadores ejecutan diferentes instrucciones sobre el mismo conjunto de datos. Este modelo es raro y se usa en situaciones donde se necesita mucha seguridad, como en sistemas que evitan fallos. Es difícil encontrar ejemplos prácticos de lenguajes que utilicen este enfoque, ya que tiene poca aplicabilidad en la mayoría de los problemas de computación general. algunas áreas donde se puede ver este enfoque incluyen:

Lenguajes de programación para sistemas de alta disponibilidad: Algunos lenguajes utilizados en sistemas críticos, como **Ada**, pueden ser adaptados para implementar MISD en sistemas redundantes.

Lenguajes de programación de sistemas: Lenguajes como **C o C++** pueden ser utilizados en sistemas donde se implementa redundancia y procesamiento de datos para evitar fallos, aunque no están diseñados específicamente para MISD.

Lenguajes de descripción de hardware: **VHDL y Verilog**, utilizados en diseño de circuitos, pueden implementar arquitecturas que sigan el modelo MISD en sistemas redundantes.

4. **MIMD (Múltiples Instrucciones, Múltiples Datos):** Este modelo permite que muchos procesadores hagan tareas diferentes sobre datos diferentes al mismo tiempo. Este es el modelo de las arquitecturas de multiprocesador y es el más utilizado en sistemas de computación paralela.

Lenguajes como **MPI (Message Passing Interface), OpenMP, C++ con hilos (threads), Python con multiprocessing** permiten aprovechar arquitecturas MIMD.