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

### SISD (Single Instruction, Single Data - Un solo flujo de instrucciones y un solo flujo de datos)

* **Descripción**: En esta arquitectura, una única unidad de control ejecuta una sola instrucción a la vez sobre un único conjunto de datos. Esto significa que el procesador ejecuta una sola instrucción en un momento dado y esta instrucción se aplica a un solo conjunto de datos.
* Es el modelo tradicional de procesamiento de computadoras, donde un CPU realiza operaciones secuenciales.
* **Ejemplo**: Computadoras personales antiguas y mainframes.
* **Lenguajes de programación**: C, C++, Python, Java y otros lenguajes de programación imperativos y de alto nivel se utilizan comúnmente en este modelo, ya que están diseñados para trabajar de manera secuencial.

### SIMD (Single Instruction, Multiple Data - Un solo flujo de instrucciones y múltiples flujos de datos)

* **Descripción**: En esta arquitectura, una única instrucción se aplica simultáneamente a múltiples conjuntos de datos. Es ideal para operaciones vectoriales y de matriz, como en el procesamiento de matrices o gráficos.
* Es común en aplicaciones que requieren realizar la misma operación en grandes conjuntos de datos, como el procesamiento de imágenes o en algoritmos de aprendizaje automático.
* **Ejemplo**: Procesadores vectoriales como los de Cray.
* **Lenguajes de programación**: Extensiones vectoriales en C/C++ con extensiones SIMD (como SSE, AVX) y lenguajes de programación de alto nivel que permiten paralelismo, como CUDA(para programación en GPU) o OpenCL, son ejemplos donde se implementa este modelo.

### MISD (Multiple Instruction, Single Data - Múltiples flujos de instrucciones y un solo flujo de datos. Lenguajes)

* **Descripción**: En esta arquitectura, múltiples unidades de control ejecutan diferentes instrucciones sobre el mismo conjunto de datos. Este modelo es menos común y se utiliza en situaciones muy específicas, como en algunos sistemas de redundancia para aumentar la fiabilidad de cálculos.
* Se puede encontrar en sistemas de procesamiento de señales o en aplicaciones de control crítico donde se requiere alta disponibilidad.
* **Ejemplo**: Sistemas de procesamiento de datos redundantes para aplicaciones críticas.
* **Lenguajes de programación**: No hay lenguajes específicos ampliamente utilizados para MISD debido a su rareza, sin embargo, se podrían utilizar lenguajes como Ada o C para implementar algoritmos redundantes.

### MIMD (Multiple Instruction, Multiple Data - Múltiples flujos de instrucciones y múltiples flujos de datos)

* **Descripción**: En esta arquitectura, múltiples unidades de control ejecutan diferentes instrucciones sobre diferentes conjuntos de datos simultáneamente. Es el modelo más flexible y se utiliza en sistemas multiprocesador y multinúcleo. Es la base para la mayoría de las arquitecturas de multiprocesadores y sistemas distribuidos.
* **Ejemplo**: Computadoras con múltiples núcleos y clústeres de computadoras.
* **Lenguajes de programación**: MPI (Message Passing Interface), OpenMP,
* Lenguajes como C, C++, Java y Python, junto con frameworks de paralelismo como MPI (Message Passing Interface), OpenMP y lenguajes de programación paralela como Go y Erlang, se utilizan comúnmente para implementar algoritmos en sistemas MIMD.