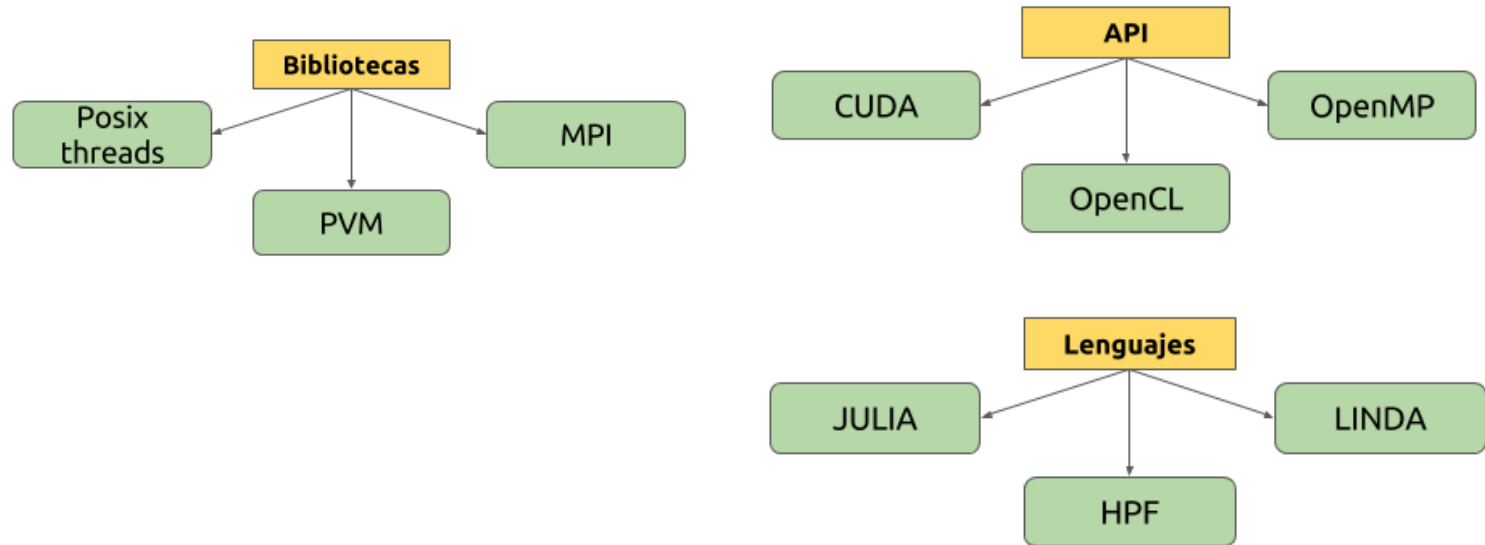


1. Explique cuál es la relación de la taxonomía de Flynn y cada una de las librerías utilizadas hasta el momento.

### Clasificación de las herramientas de programación paralela



### Lenguajes que integran el paradigma

Actualmente, existe una variedad de lenguajes de programación con un enfoque multiparadigma, los cuales permiten tener flexibilidad a la hora de querer lograr un objetivo, en cuanto a programación. Igualmente, los lenguajes suelen complementar dicho comportamiento con librerías, APIs o frameworks (ya sean comunitarios o privados), los cuales le permiten al programador el disponer de herramientas de trabajo.

El paradigma paralelo es implementado, en la mayoría de los lenguajes más conocidos actualmente, a través de dichas herramientas. En la tabla, se hace un recuento de herramientas específicas de algunos de esos lenguajes, mencionando que no son las únicas que existen.

Lenguaje	Implementación
C	<a href="#">Posix</a> , <a href="#">MPI</a> , <a href="#">OpenMp</a>
<a href="#">C++</a>	<a href="#">Rogue Wave</a> , <a href="#">Boost</a> , <a href="#">Thread</a> , <a href="#">Dlib</a> , <a href="#">OpenMP</a> , <a href="#">OpenThreads</a> , <a href="#">Parallel Patterns Library</a> , <a href="#">POCO C++ Libraries</a> , <a href="#">POSIX Threads</a> , <a href="#">Qt Qthread</a> , <a href="#">Stapl</a> , <a href="#">TBB</a> , <a href="#">IPP</a>
C#	<a href="#">Task Parallel Library</a> , <a href="#">Parallel Query PLINQ</a> ,
<a href="#">Fortran</a>	<a href="#">Co-arrays</a> , <a href="#">MPI</a> , <a href="#">MPICH</a> , <a href="#">OpenMPI</a> , <a href="#">OpenMP</a> , <a href="#">OMP</a> , <a href="#">ADAPTOR</a> , <a href="#">PGI CUDA Fortran compiler</a> , <a href="#">PIPS</a> .
Go	<a href="#">Goroutines y channels</a>
Java	Clase <a href="#">Thread</a> (interfaz <b>Runnable</b> )
Javascript	<a href="#">Parallel JS</a> (usando web workers)
<a href="#">Julia</a>	<a href="#">Tareas asíncronas</a> , <a href="#">Multihilos</a> , <a href="#">computación distribuida</a>
Matlab	<a href="#">Parallel computing toolbox</a>
Python	<a href="#">Múltiples librerías</a>
Ruby	<a href="#">Gem Parallel</a>

	<b>simple</b>	<b>múltiple</b>
<b>simple</b>	<b>SISM</b> Java,c,c++  MPI,OPENMP, Clase Thread (interfaz Runnable)	<b>SIMM</b> CUDA(Compute Unified Device Architecture)  MPI,OPENMP, Clase Thread (interfaz Runnable)
<b>múltiple</b>	<b>MISM</b> CUDA(Compute Unified Device Architecture)  MPI,OPENMP, Clase Thread (interfaz Runnable)	<b>MIMM</b> CUDA(Compute Unified Device Architecture)  MPI,OPENMP, Clase Thread (interfaz Runnable)

### **CUDA (Compute Unified Device Architecture)**

CUDA es una API de computación en paralelo, que incluye un compilador y una serie de herramientas de desarrollo creadas por NVIDIA para implementar algoritmos en sus GPU. Permite programador mediante una variación del lenguaje de programación C. Mediante wrappers se puede usar en C/C++, Java y Fortran y existen múltiples aplicaciones en Python. CUDA intenta aprovechar el gran paralelismo, y el alto ancho de banda de la memoria en las GPU en aplicaciones con un gran coste aritmético frente a realizar numerosos accesos a memoria principal, lo que podría actuar de cuello de botella.

### **OpenMP (Open multiprocessing)**

OpenMP es una API para programación multiproceso de memoria compartida en múltiples plataformas que fue publicada por la Architecture Review Board en 1997. Su modelo de ejecución se basa en memoria compartida multihilos. Se considera

el sucesor más sofisticado de Posix Threads. Dispone de directivas que apoyan al programador para convertir algoritmos secuenciales a paralelos de forma eficiente. Su especificación fue realizada para lenguajes C, C++ y Fortran. Funciona en la mayoría de las arquitecturas de procesador y sistemas operativos, incluyendo Solaris, AIX, HP - UX, Linux, Mac OS X, y las plataformas de Windows .

### **MPI (Message Passing Interface)**

MPI es una librería estándar para ser usada en programas que aprovechen arquitecturas de múltiples procesadores. Fue desarrollada por el Centro de investigación en Computación paralela en Williamsbrug, y publicada en Noviembre de 1993, año en el que nace el foro que actualmente se encarga de su mantenimiento y evolución. Sus lenguajes de especificación son C , C++ y Fortran. Existen implementaciones para Python, Ocaml, Java, .NET y PHP.

**Ventajas:** Estandarización y eficiencia en el traspaso de mensajes, portabilidad, y disponibilidad en varios lenguajes de programación.

**Desventajas:** Posibles latencias del bus de intercomunicación, la dificultad en la sincronización entre tareas.