

The background is a dark navy blue. It is decorated with several large, overlapping, semi-transparent geometric shapes in various colors: bright green, cyan, magenta, orange, and light blue. These shapes are arranged in a way that creates a sense of depth and movement, with some appearing to be layered on top of others.

PROGRAMACIÓN PARALELA

Antonio Jose Suarez Fortich
Samael De Jesus Salcedo Amortegui
Diego Fernando Chacon Arango



1. INTRODUCCIÓN

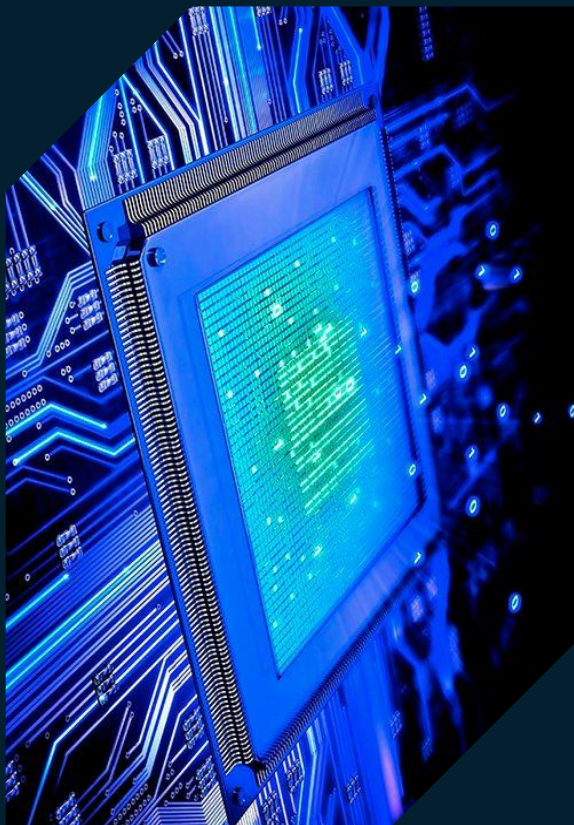
Programación paralela

(No concurrente, ni secuencial)

Pensar en paralelo

- › Colas de supermercado
- › Construcciones de edificios
- › Preparaciones culinarias





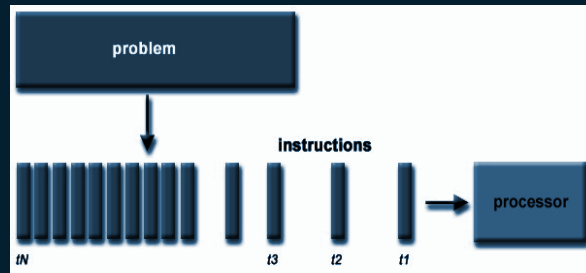
Procesos computacionales

Desde el desarrollo hasta el
funcionamiento

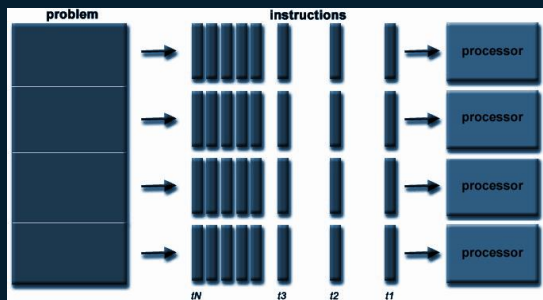
The slide features a dark navy blue background. In the top-left and bottom-left corners, there are overlapping, semi-transparent geometric shapes in shades of cyan, orange, and magenta. Similarly, in the top-right and bottom-right corners, there are overlapping, semi-transparent geometric shapes in shades of cyan, magenta, and orange. The central text is white and bold.

2. FILOSOFÍA DEL PARADIGMA

SERIALIZAR

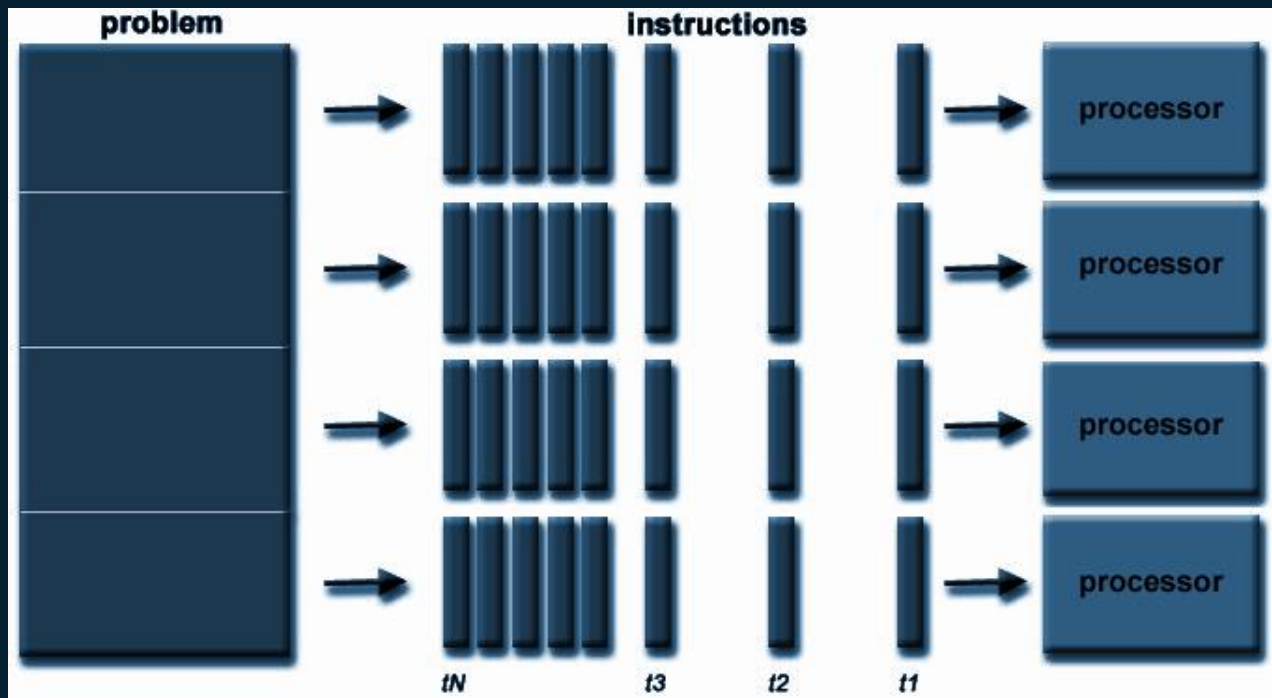


VS

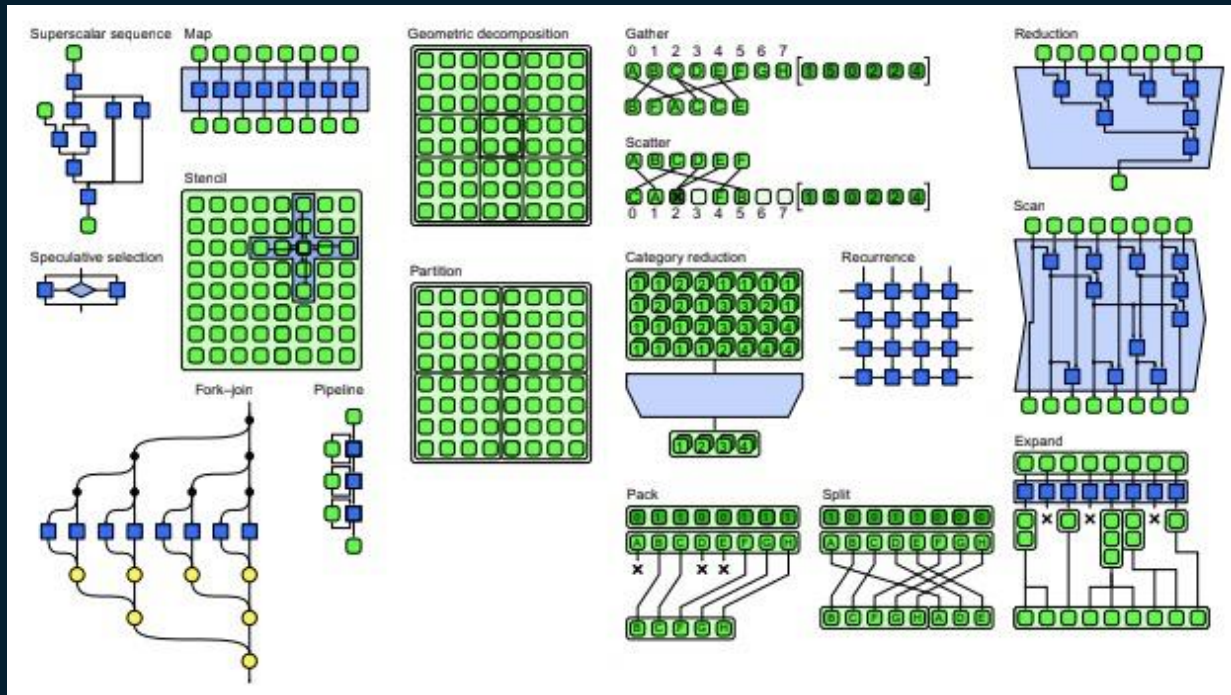


PARALELIZAR

Paralelizar



Paralelizar



The slide features a dark blue background with abstract, overlapping geometric shapes in the corners. On the left, there are shapes in shades of green, blue, orange, and pink. On the right, there are shapes in shades of green, blue, pink, and orange. The central text is white and bold.

3. CONCEPTOS CLAVE

Datos



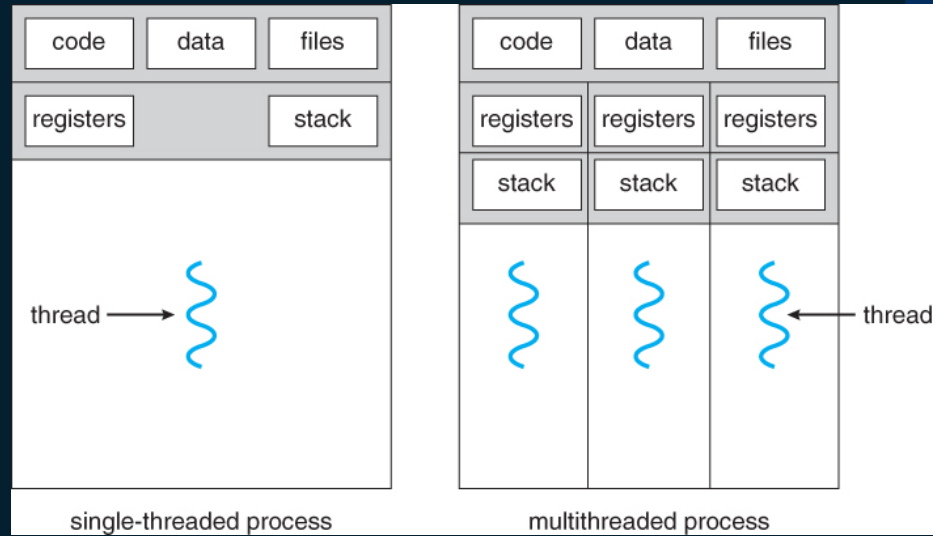
Tareas

Dependencia de Datos

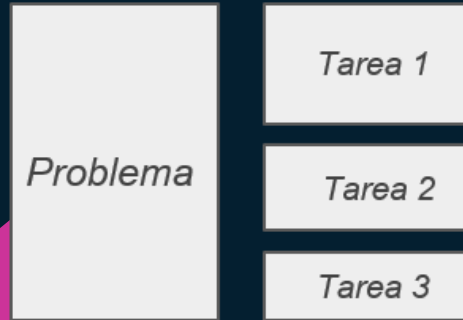


Dependencia de Control

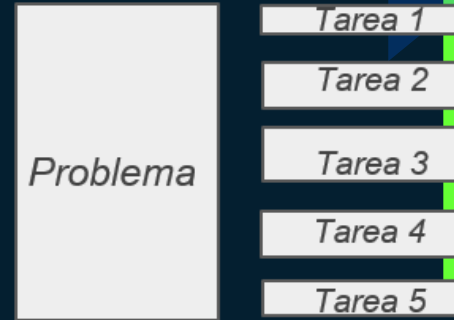
Hilos



Granularidad



Gruesa

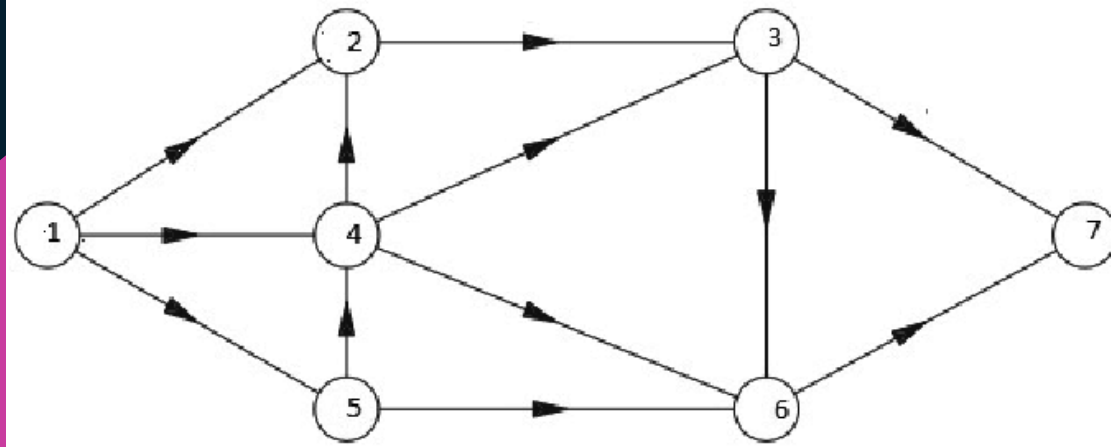


Fina

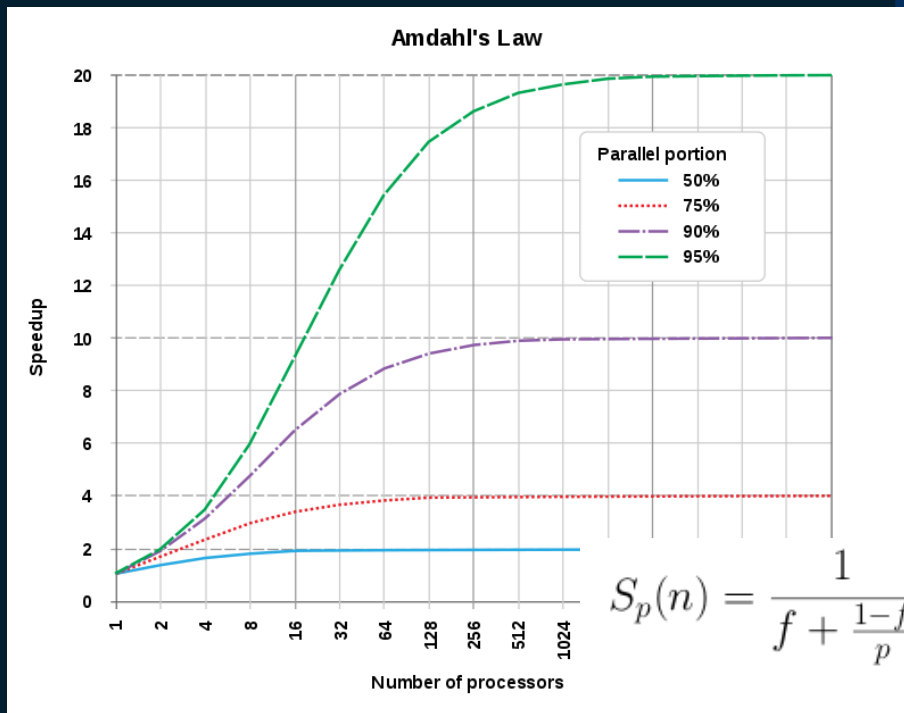
Sincronización y Scheduling



Grafo computacional y sección crítica



Ley de Ahmdal



Condiciones de carrera





4. VENTAJAS Y DESVENTAJAS

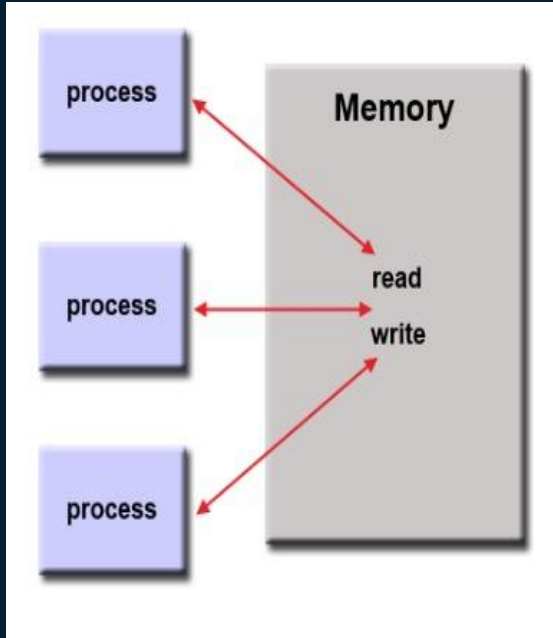
Ventajas

- Las tendencias de hardware apuntan al apoyo del paralelismo
- Con el paralelismo podemos resolver mayor número de problemas, que de otra manera no se podrían realizar en una sola CPU
- Podemos acelerar la ejecución de los programas en general
 - Obtención acelerada de resultados

Desventajas

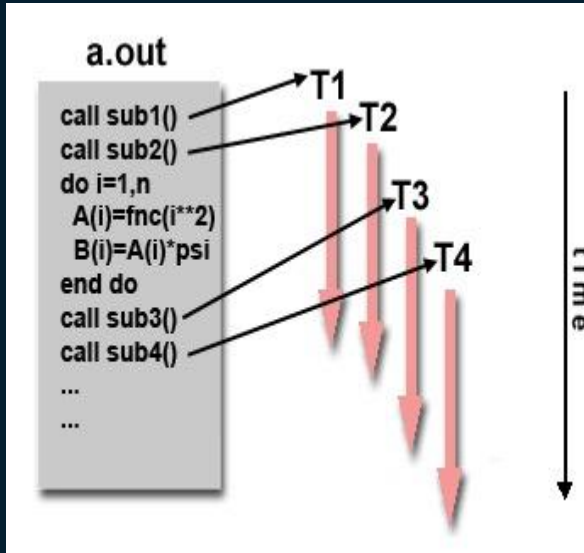
- Mayor cantidades de unidades de procesamiento -> Mayor consumo de energía
- Necesidad de procesos de sincronización
 - Mayor dificultad a la hora de programar

5. MODELOS DE PROGRAMACIÓN PARALELA



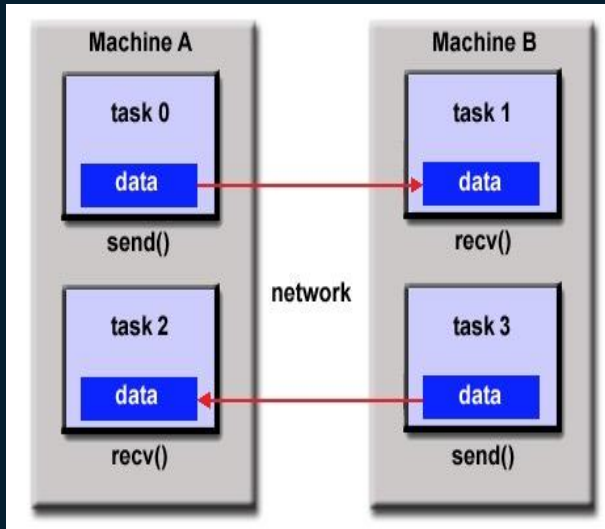
Memoria compartida

- > Quizás el modelo más sencillo.
- > Todos los procesos tienen el mismo acceso a la memoria, no hay "dueños" de los datos.
- > Se hace difícil el manejo de datos localmente



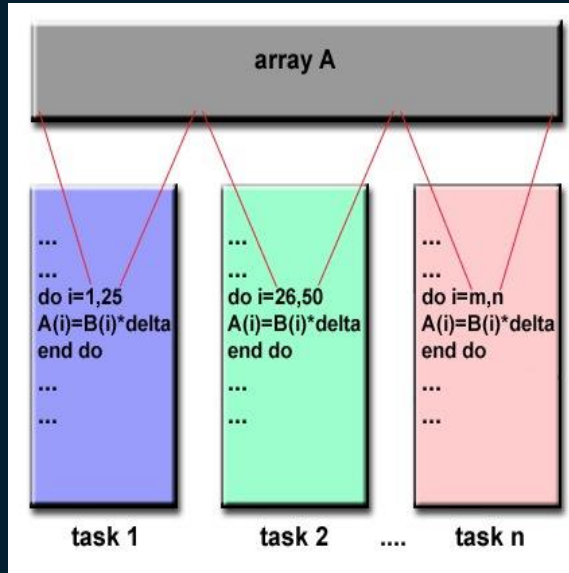
Modelos con hilos

- > Es un tipo de memoria compartida
- > Tarea compleja dividida en varias menos complejas.



Memoria distribuida

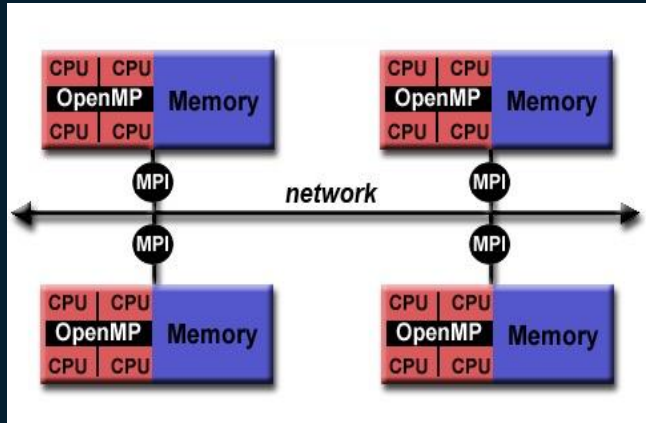
- > Los procesos se pueden ubicar en una o varias máquinas.
- > Los procesos intercambian datos
- > `send()` y `receive()`



Datos paralelos

- > Estructura de datos común.
- > Los procesos se ejecutan en diferentes particiones de la estructura.

Modelos híbridos



- > Combinación de los modelos anteriores
- > Ejemplo: Memoria distribuida combinada con hilos.

6. PATRONES DE CONTROL PARALELOS

Patrones de control

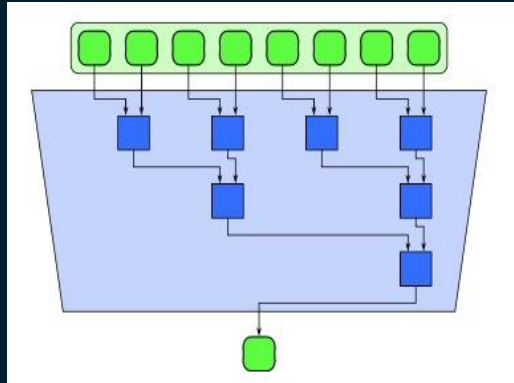
Seriales

- > Secuencial
- > Iteración
- > Selección
- > Recursión

Paralelos

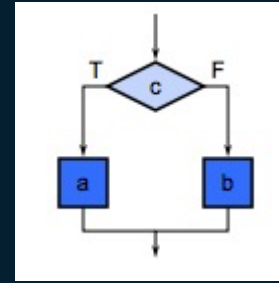
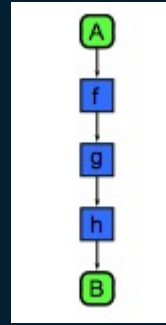
- > Fork-join
- > Map
- > Stencil
- > Scan
- > Recurrencia
- > Reducción

SERIALIZAR



VS

PARALELIZAR



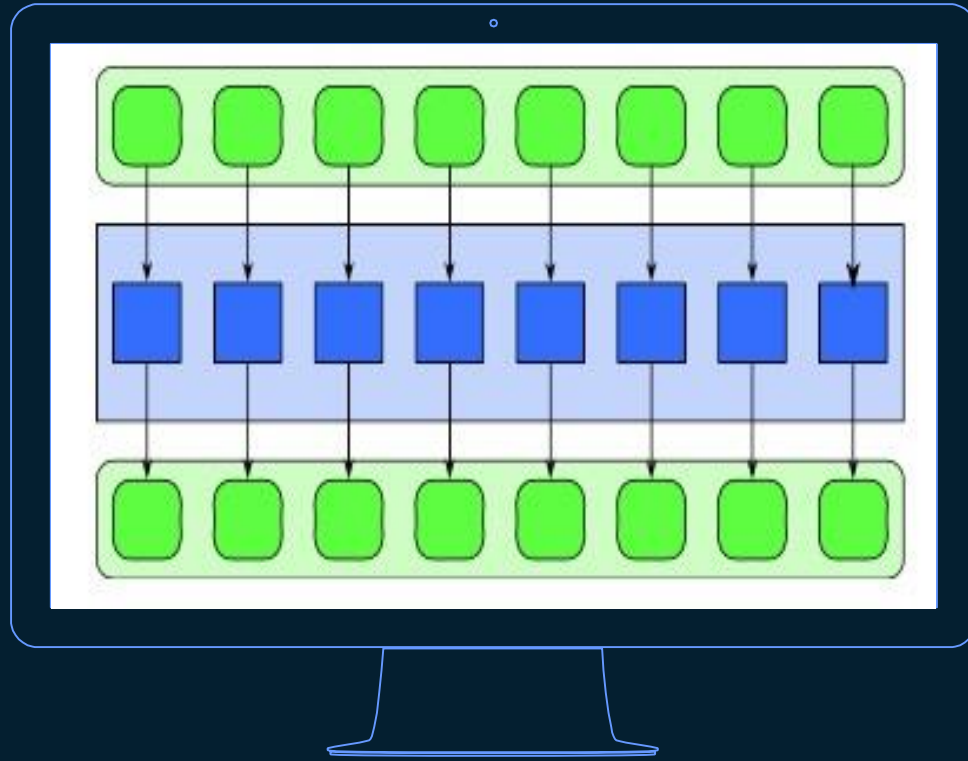
Patrones de control

Recurrencia

Ciclos donde los cuerpos de la función son independientes y por lo tanto se pueden paralelizar

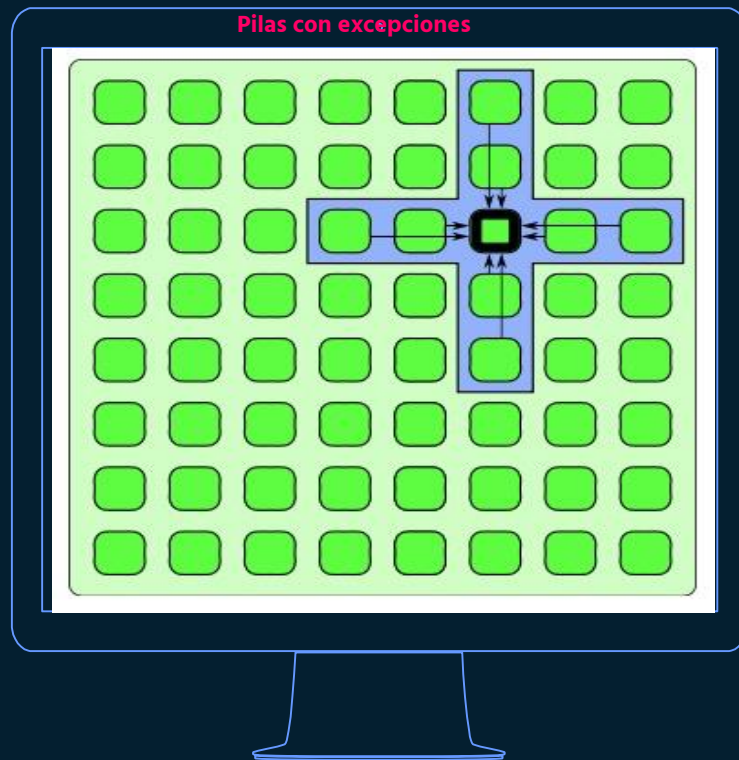
MAP

Utilizado cuando se le hace una operación independiente a cada elemento de una estructura de control



STENCIL

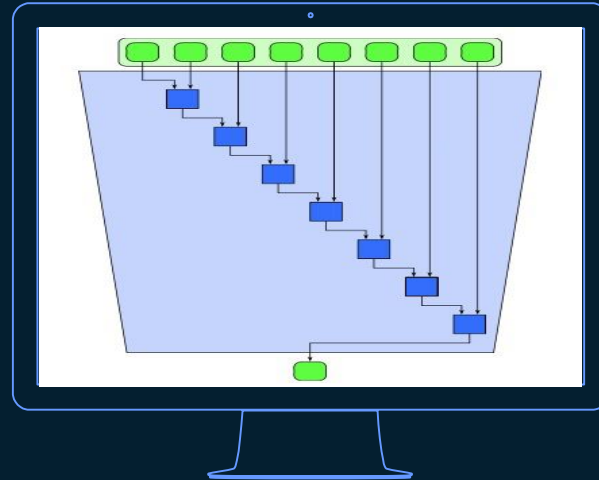
Generalización de map.
Acceso a algunos
vecinos.



REDUCCIÓN

Función combinatoria asociativa

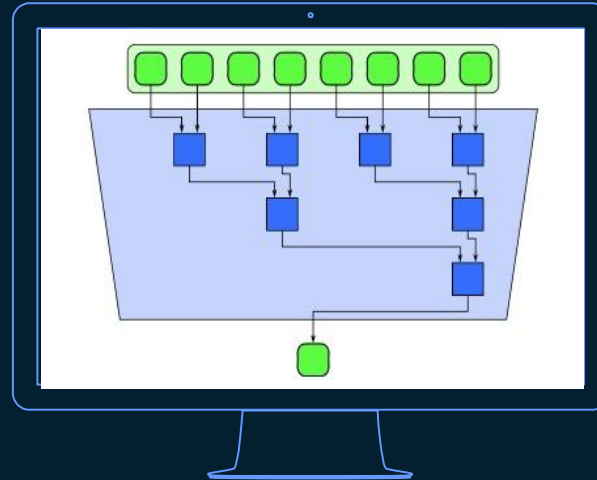
```
double my_add_reduce(  
    const double a[], //input array  
    size_t n          // number of elements  
) {  
    double r = 0.0; //initialize with the identity for addition  
    for (int i = 0; i < n; ++i)  
        r += a[i]; // each iteration depends on the previous one  
    return r;  
}
```



REDUCCIÓN

Función combinatoria asociativa

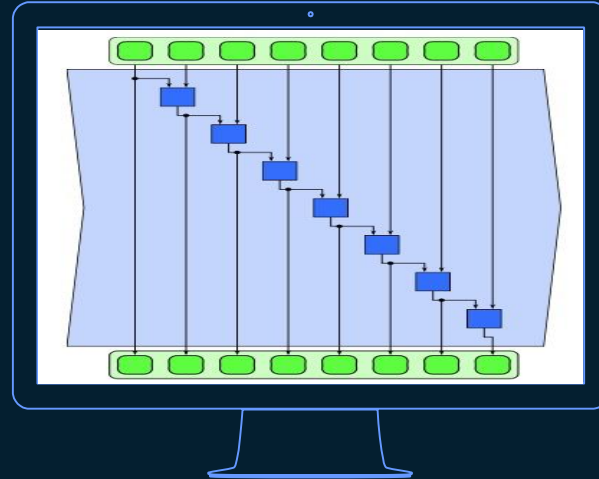
```
double my_add_reduce(  
    const double a[], //input array  
    size_t n          // number of elements  
) {  
    double r = 0.0; //initialize with the identity for addition  
    for (int i = 0; i < n; ++i)  
        r += a[i]; // each iteration depends on the previous one  
    return r;  
}
```



SCAN

Dobles o ***folds*** que permiten reorganizar las operaciones

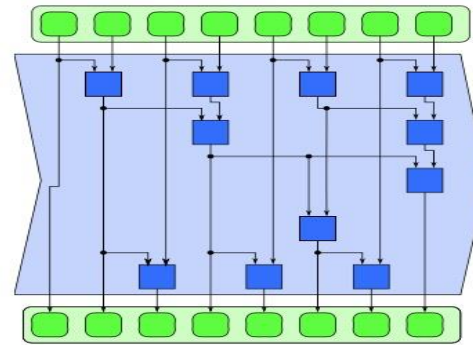
```
void my_add_iscan(  
    const float a[], // input array  
    float b[],       // output array  
    size_t n         // number of elements  
) {  
    if (n>0) b[0] = a[0]; // equivalent to assuming b[i-1] is zero  
    for (int i = 1; i < n; ++i)  
        b[i] = b[i-1] + a[i]; // each iteration depends on the previous one  
}
```




SCANNER

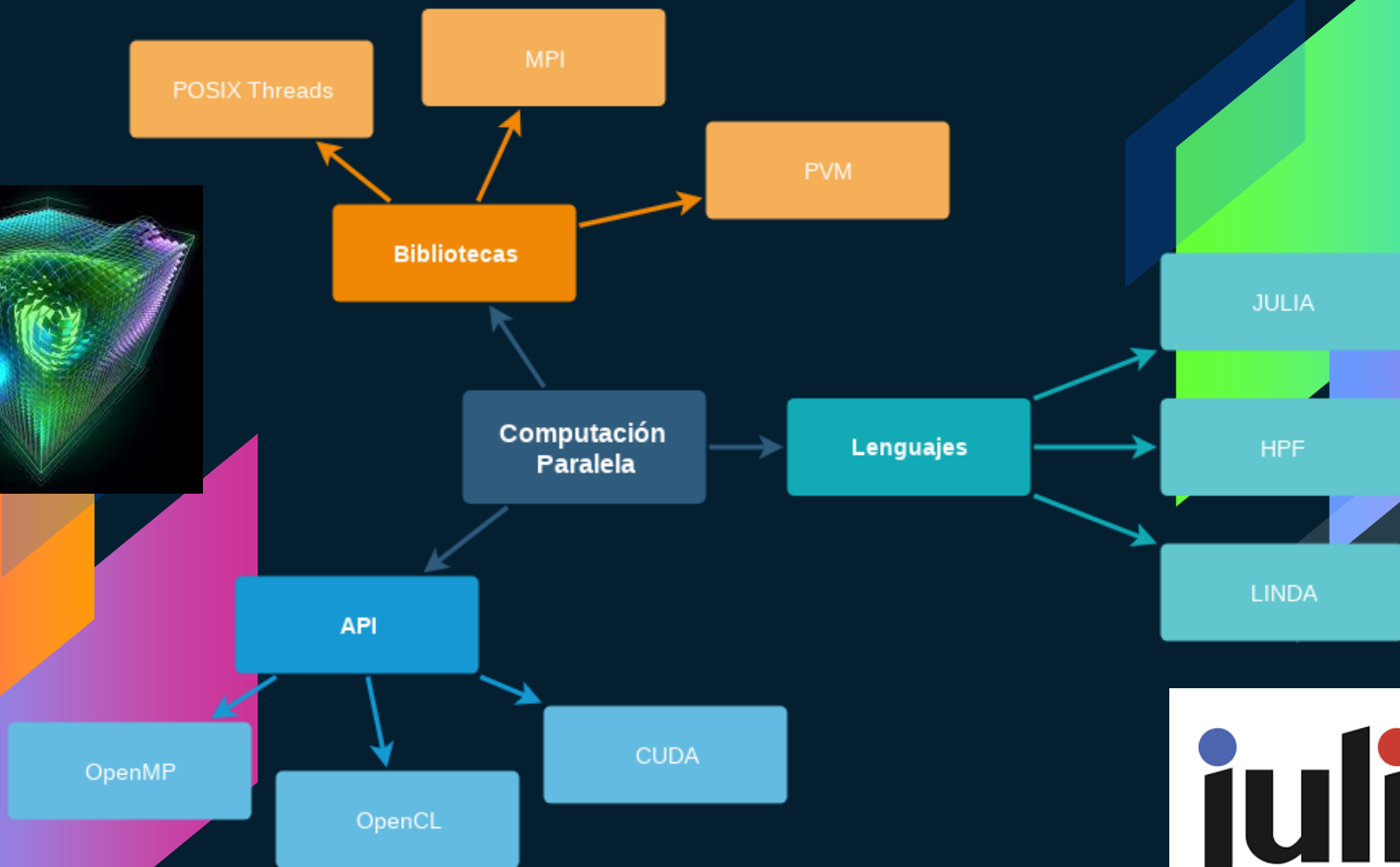
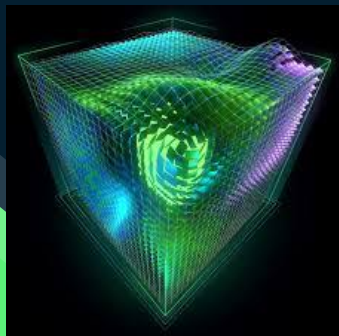
Doblesces o ***folds*** que permiten reorganizar las operaciones

```
void my_add_iscan(  
    const float a[], // input array  
    float b[],       // output array  
    size_t n         // number of elements  
) {  
    if (n>0) b[0] = a[0]; // equivalent to assuming b[i-1] is zero  
    for (int i = 1; i < n; ++i)  
        b[i] = b[i-1] + a[i]; // each iteration depends on the previous one  
}
```





7. LENGUAJES DE PROGRAMACIÓN



OPENMP

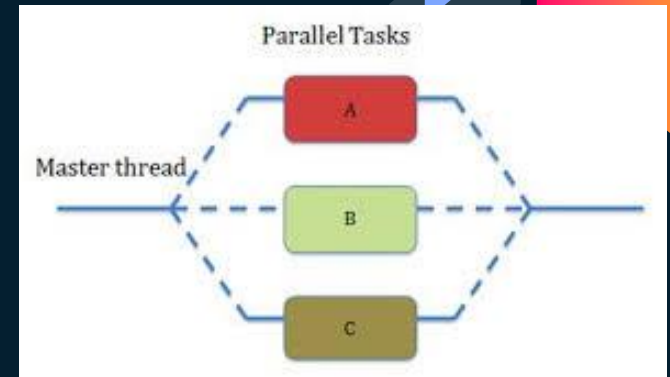


**Api utilizada en programación
multiproceso - multihilo de
memoria compartida, el cual se
fundamenta en el modelo
Fork/Join.**

Trabaja en C/C++

Modelo FORK/JOIN

Una Tarea **pesada** puede ser dividida en tareas independientes **sencillas** (**Fork**), las cuales se combinan cuando tienen la solución a la tarea inicial (**Join**)



WEB CL

Web **Computing Language** es una tecnología que permite la integración de **OpenCL** en código JavaScript para poder usar programación paralela heterogénea.



OpenCL

Consta de una **interfaz de programación de aplicaciones** y de un **lenguaje de programación**.
Juntos permiten crear aplicaciones con paralelismo a nivel de datos y de tareas



8. APLICACIONES

Predicción y Cambio Climático

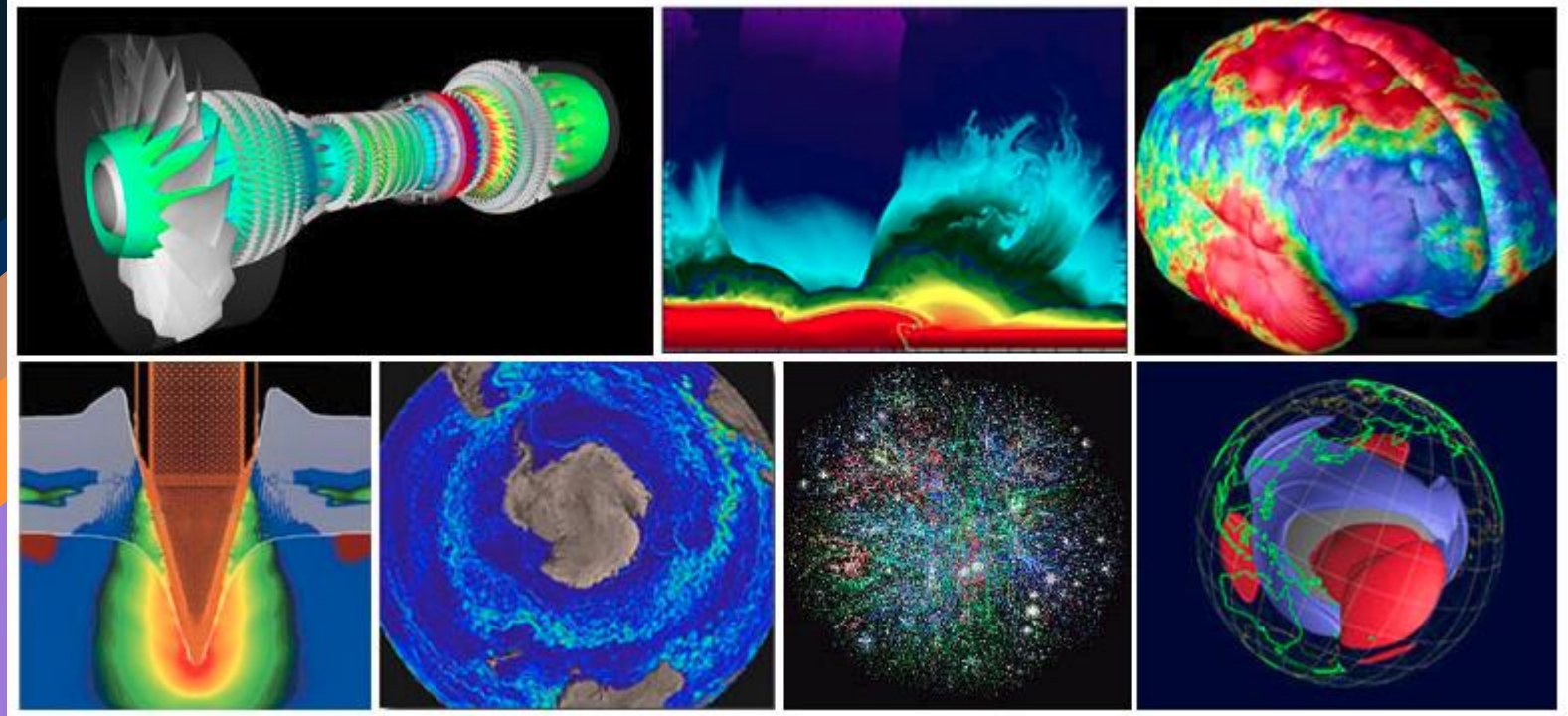
La información utilizada para predecir el clima ha sido recolectada durante más de 500 años. En 1904 se propuso un **modelo de ecuaciones diferenciales** para la predicción meteorológica



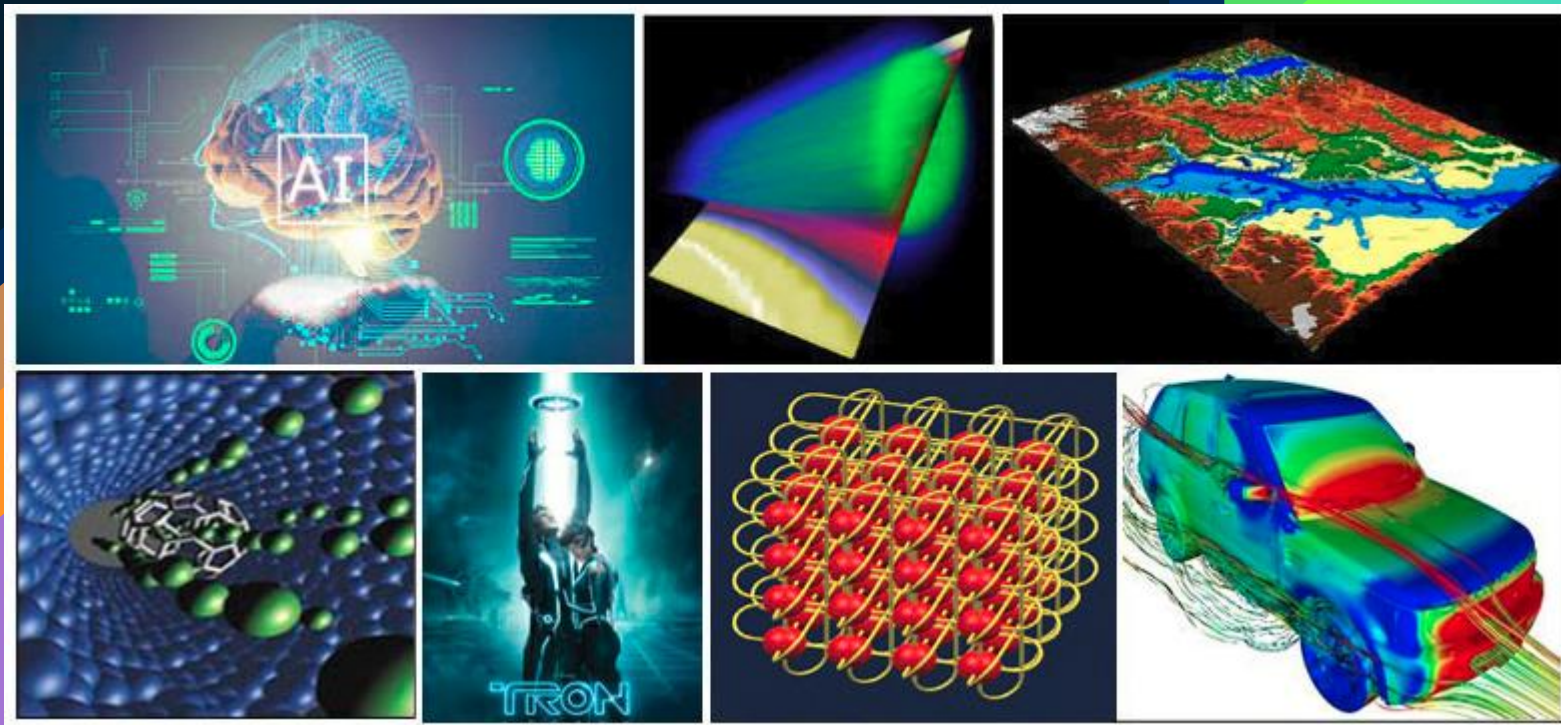
National Oceanic and
Atmospheric Administration

U.S. Department of Commerce

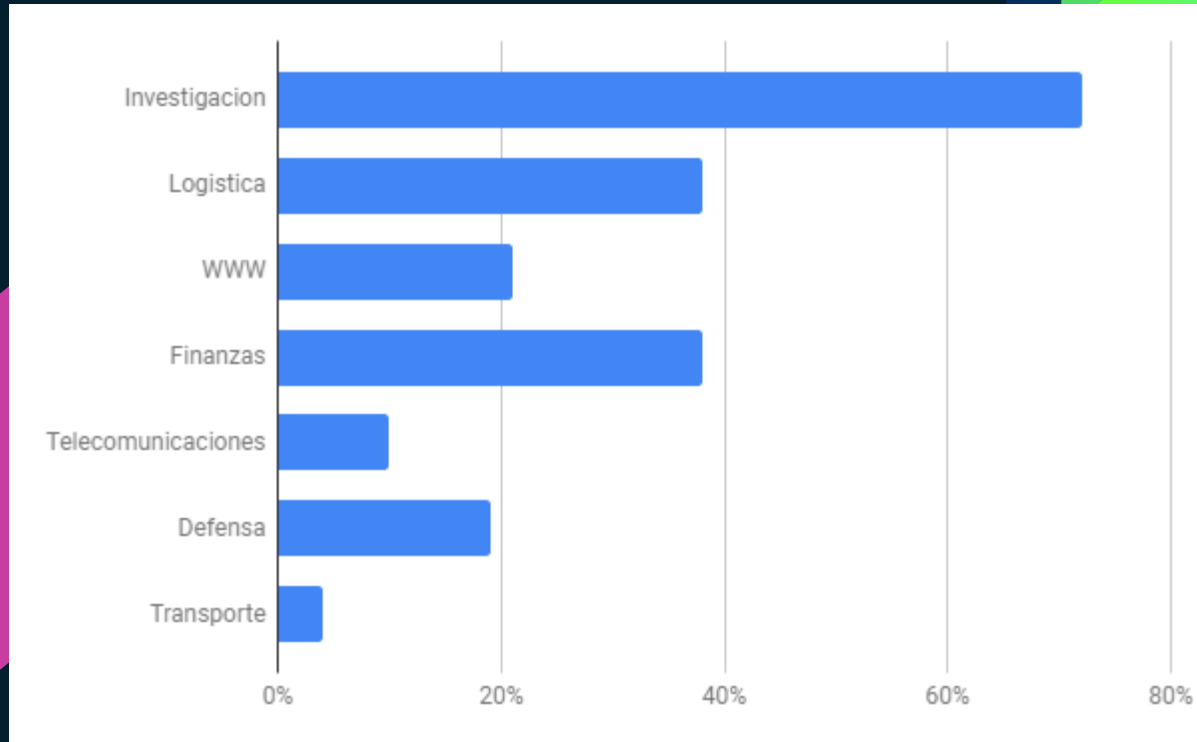
Ciencia e Ingeniería



Industria y Comercio



TOP ÁREAS DE APLICACIÓN

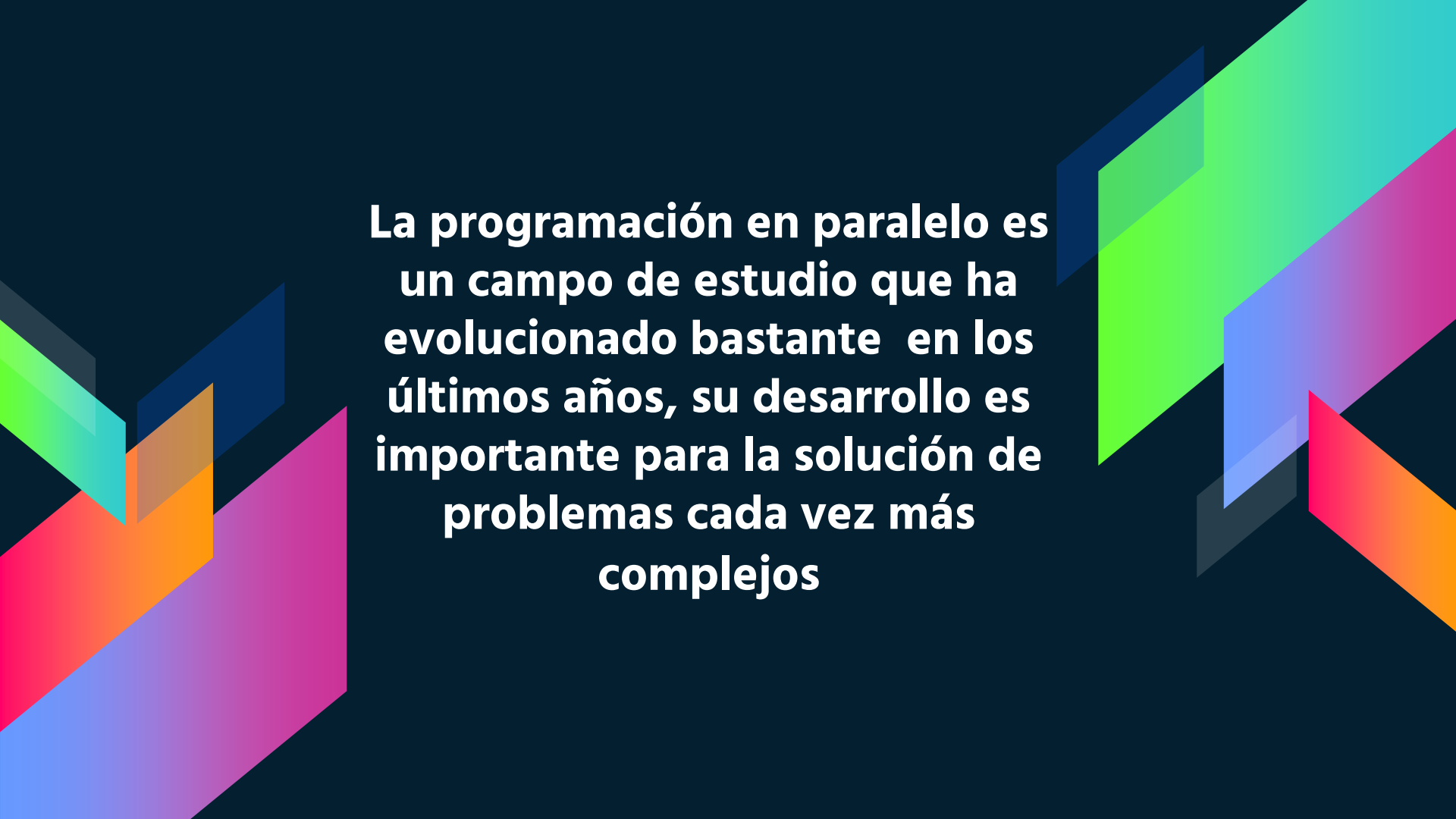


9. EJEMPLOS

Serie de Fibonacci



9. Conclusiones

The background is a dark navy blue. It features several large, overlapping, semi-transparent geometric shapes in various colors: bright green, cyan, magenta, orange, and light blue. These shapes are arranged in a way that creates a sense of depth and movement, with some appearing to be layered on top of others. The overall aesthetic is modern and tech-oriented.

**La programación en paralelo es
un campo de estudio que ha
evolucionado bastante en los
últimos años, su desarrollo es
importante para la solución de
problemas cada vez más
complejos**

BIBLIOGRAFÍA

- Presentaciones profesor Oscar Agudelo (2018)
- https://computing.llnl.gov/tutorials/parallel_comp
- McCOLL M., ROBISON A., REINDERS J., Structured Parallel Programming Patterns for Efficient Computation (2012)