



# Proyecto final de carrera

---

## Desarrollo de un módulo de aprendizaje para programación en paralelo

Autor: José María Cámara Nebreda

Tutor: Jesús Manuel Hernández Mangas



# Antecedentes

- Problemática de la docencia en Arquitectura de Computadores
- Contenidos:
  - Teóricos: arquitecturas avanzadas
  - Prácticos: programación
- Condicionamientos presupuestarios

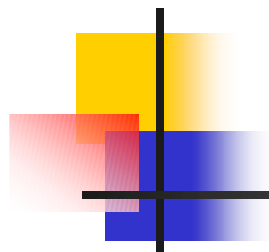


# Objetivos

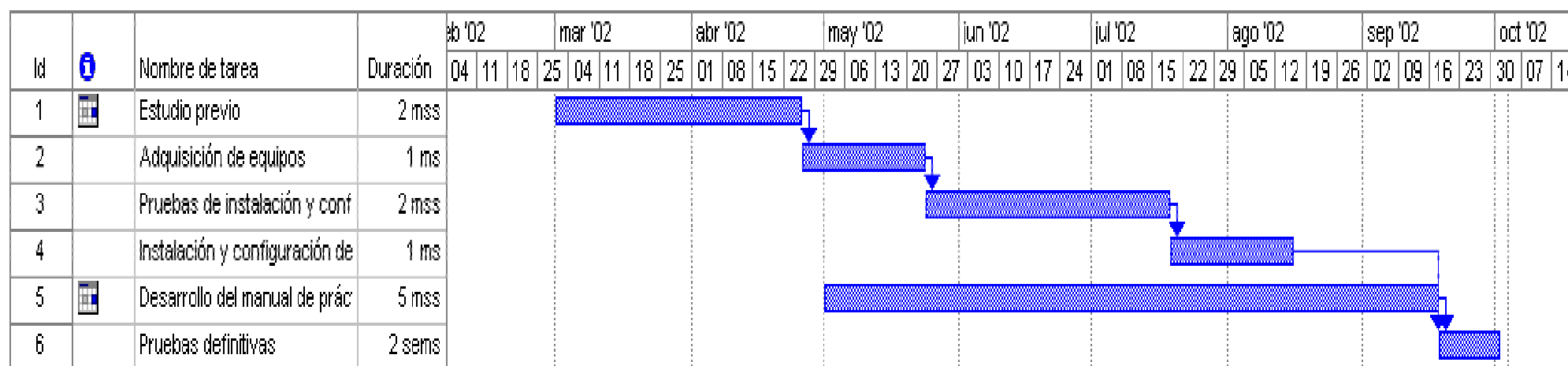
---



- Estudiar las posibles soluciones existentes
- Decidir cuál es la más apropiada para cumplir:
  - Objetivos docentes
  - Restricciones presupuestarias
- Implantar la solución adoptada
- Desarrollar un manual de prácticas para facilitar su docencia



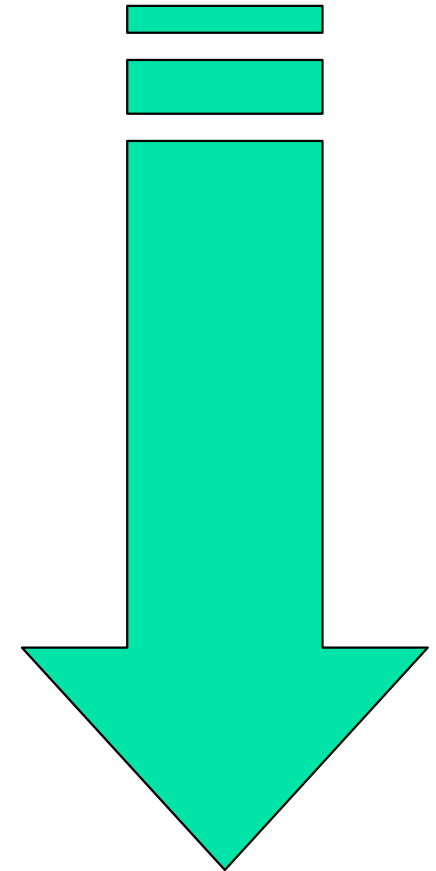
# Desarrollo del proyecto





# Estudio previo

- Programación de sistemas paralelos:
  - Paralelismo implícito
  - Paralelismo explícito
- Alternativas en paralelismo explícito:
  - Lenguajes específicos: Occam, Linda
  - Lenguajes extendidos: Fortran 90, HPF
  - Extensiones: librerías de paso de mensajes
- Librerías de paso de mensajes
  - PVM
  - MPI





# MPI vs PVM

---



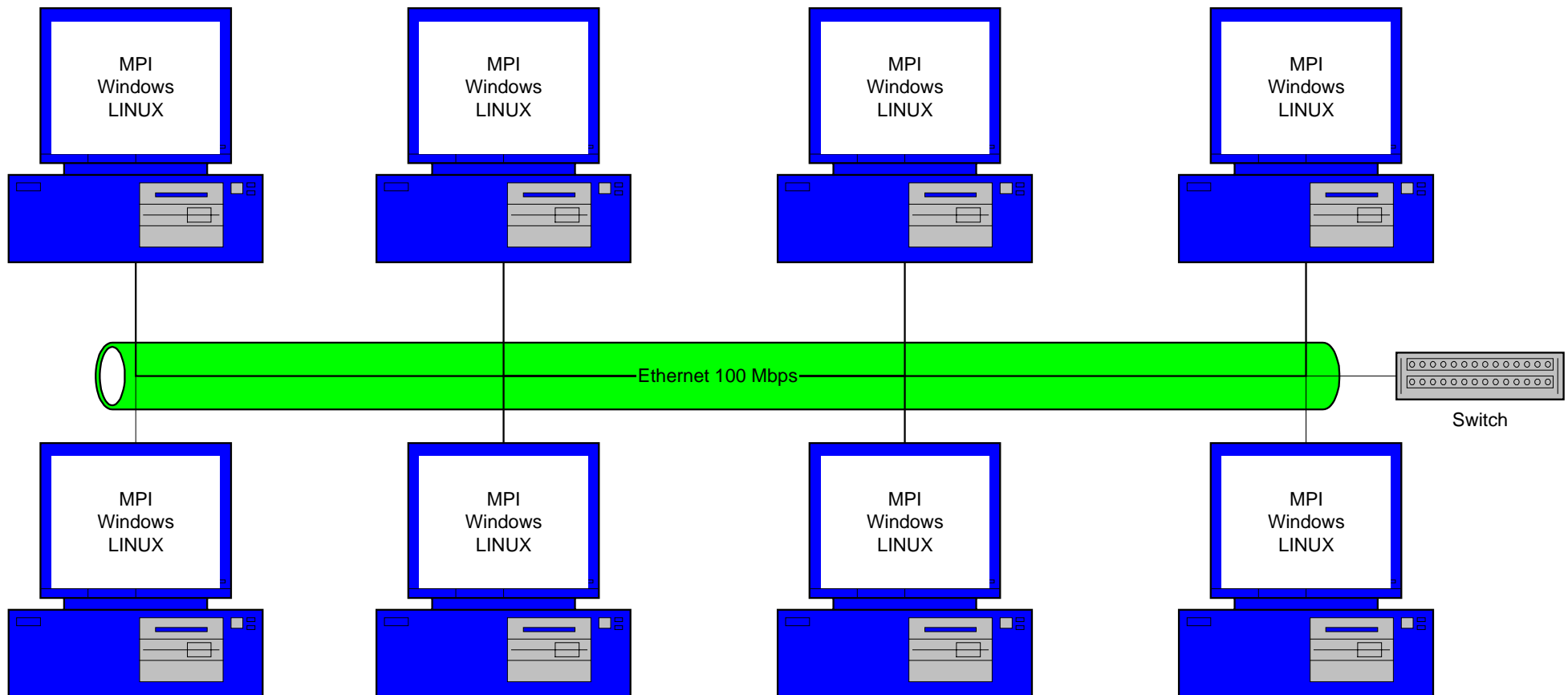
## ■ MPI

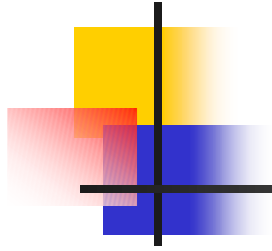
- Mayor librería de funciones
- Más dinámico
- Más extendido en grandes equipos
- Entrada-salida paralela
- Tipos de datos derivados
- Topologías virtuales
- Operaciones no bloqueantes
- Implementación para Windows

## ■ PVM

- No dependiente de implementaciones
- Tolerante a fallos
- Gestión dinámica de procesos

# Solución adoptada





# Manual de prácticas

---



- Práctica 0: Ejemplo básico
- Práctica 1: Comunicaciones punto a punto
- Práctica 2: Comunicaciones colectivas
- Práctica 3: Funciones de reparto y reducción
- Práctica 4: Topologías virtuales
- Práctica 5: Procesos de entrada/salida
- Práctica 6: Nuevos modos de envío
- Práctica 7: Tipos de datos derivados
- Práctica 8: Gestión dinámica de procesos
- Práctica 9: Ejemplo de aplicación práctica
- Práctica 10: Medida del rendimiento





# Práctica 0: Ejemplo básico

- **Objetivos:**

- Conocer la estructura de un programa paralelo
- Aprender los conceptos básicos que se manejan en MPI
- Realizar una primera prueba de programación y ejecución

- **Contenidos teóricos:**

- Estructura de un programa MPI
- Comunicadores

- **Realización práctica: Hola Mundo**



# Estructura de un programa MPI

```
■ #include <stdio.h>
■ #include <mpi.h>
■
■ int main (int argc, char *argv[])
■ {
■     int mirango, tamaño;
■
■     MPI_Init (&argc, &argv);
■     MPI_Comm_rank (MPI_COMM_WORLD, &mirango);
■     MPI_Comm_size (MPI_COMM_WORLD, &tamaño);
■
■     printf ("Proceso %d de %d: Hola Mundo\n", mirango, tamaño);
■
■     MPI_Finalize();
■     return 0;
■ }
```



# Práctica 1: Comunicaciones punto a punto

---



- **Objetivos:**
  - Estudiar modos de comunicación
  - Estudiar la estructura de los mensajes
  - Conocer las funciones de paso de mensajes básicas
  - Programar una primera aplicación de reparto de carga
- **Contenidos teóricos:**
  - Comunicación entre procesos
  - Mensajes
  - Funciones de envío y recepción
- **Realización práctica:** envío de datos entre procesos

# Modos de comunicación y mensajes



- Envíos Asíncronos
- Envíos Síncronos
- Comunicaciones bloqueantes
- Comunicaciones no bloqueantes
- Estructura de los mensajes:
  - Envoltura: fuente, destino, comunicador y etiqueta
  - Cuerpo: buffer, tipo de datos, cuenta



# Práctica 2: Comunicaciones colectivas

---

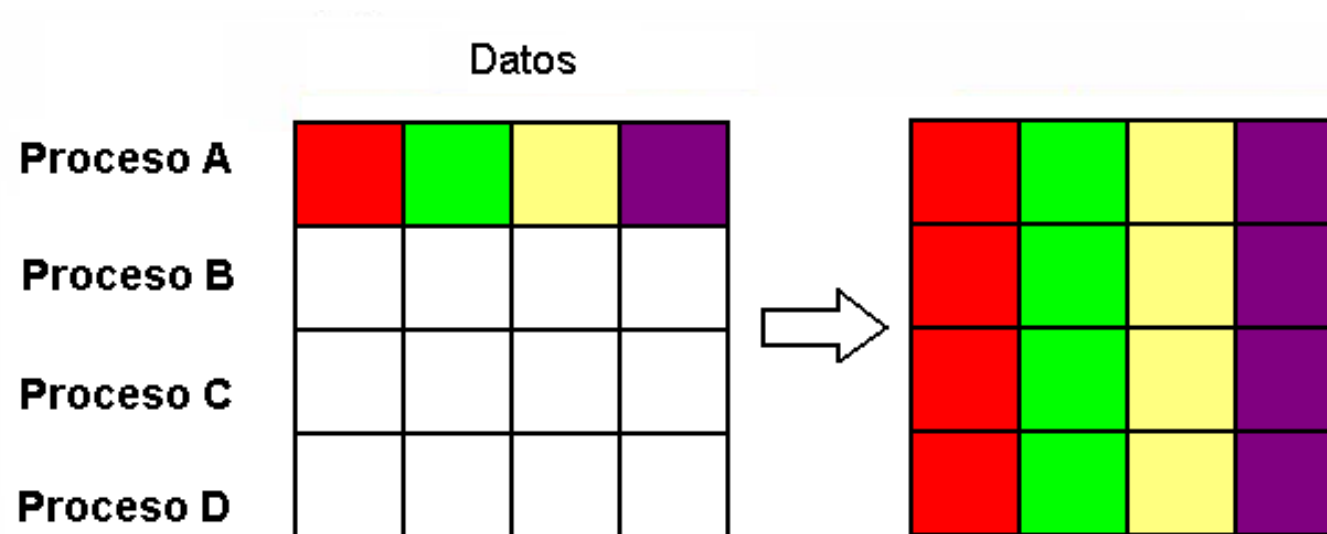


- **Objetivos:**
  - Extender las posibilidades de comunicación a las comunicaciones colectivas como método de simplificación de la comunicación
  - Estudiar posibles aplicaciones de las comunicaciones colectivas
- **Contenidos teóricos:**
  - Comunicaciones colectivas
  - Implementación en MPI
- **Realización práctica:** suma de matrices por columnas



# Comunicaciones colectivas

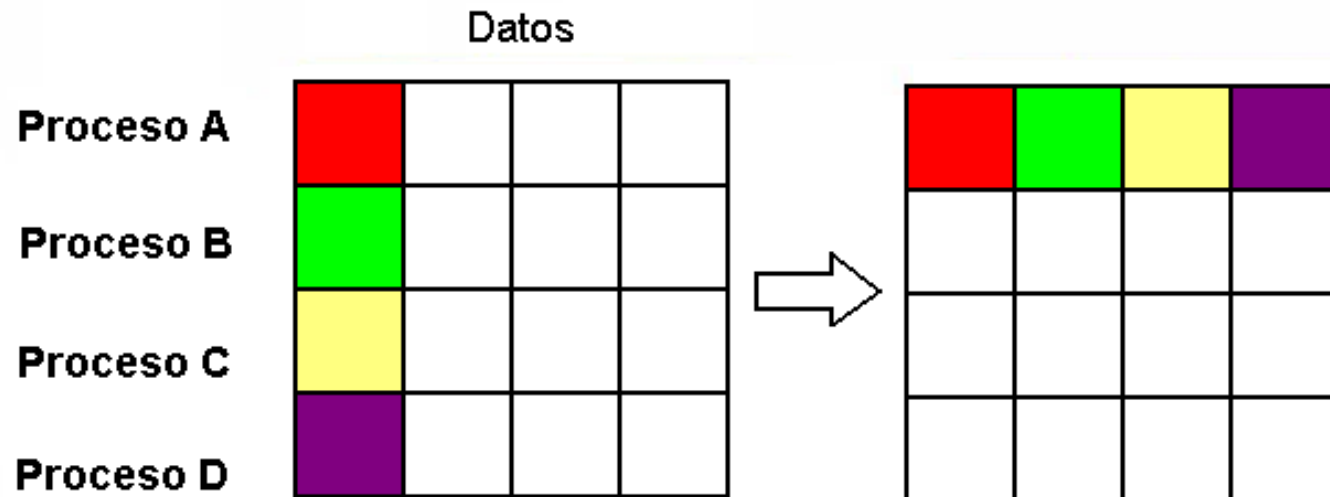
- **Broadcast (difusión):** envío de uno a muchos:





# Comunicaciones colectivas

- **Gather (recopilación):** envío de muchos a uno:



# Práctica 3: Funciones de reparto y reducción



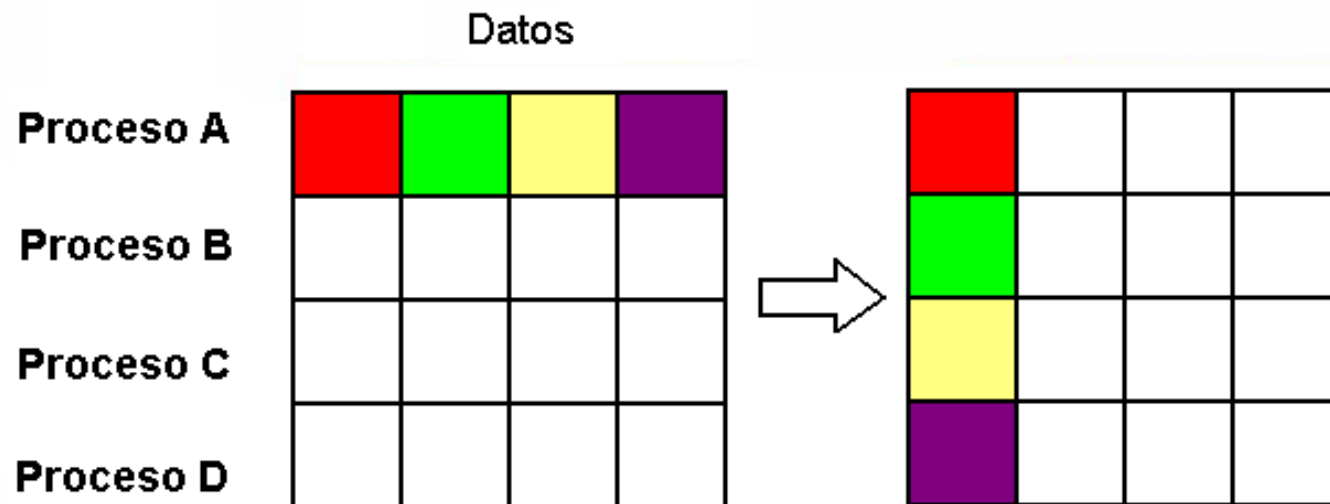
- **Objetivo:**
  - Estudiar opciones más potentes dentro de las comunicaciones colectivas
- **Contenidos teóricos:**
  - Operaciones de reparto y reducción
  - Implementación en MPI
- **Realización práctica:** desarrollo en serie de  $e^x$



# Operaciones de reparto y reducción



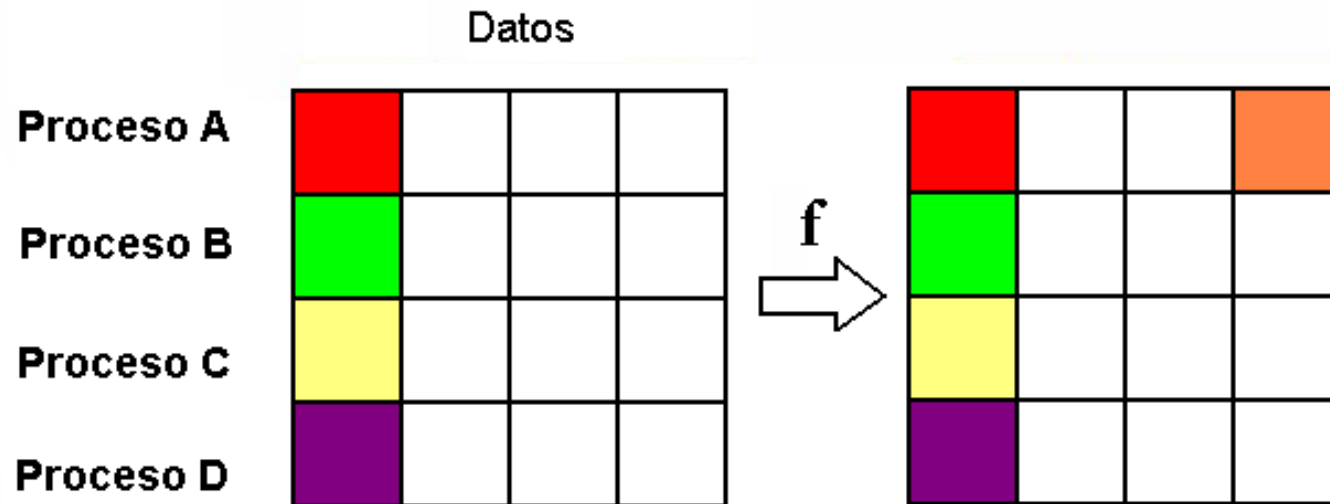
- **Reparto:** distribuir subconjuntos de datos a distintos procesos dentro de un conjunto común



# Operaciones de reparto y reducción



- **Reducción:** agrupar subconjuntos de datos procedentes de diferentes procesos a través de una función matemática





## Práctica 4: Topologías virtuales

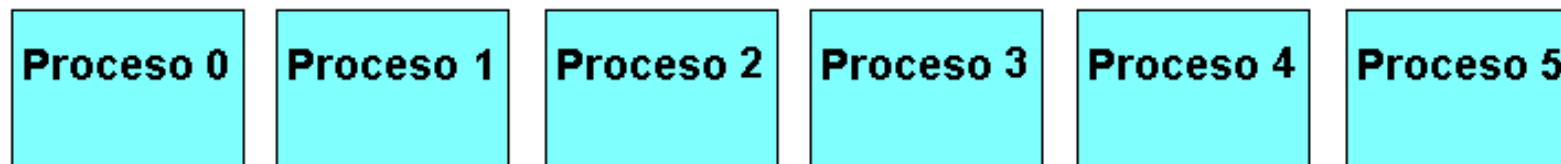
- **Objetivo:**
  - Acercamiento a las topologías virtuales como herramienta para potenciar la resolución de determinados problemas
- **Contenidos teóricos:**
  - Topología cartesiana en MPI
- **Realización práctica:** suma de matrices por elementos



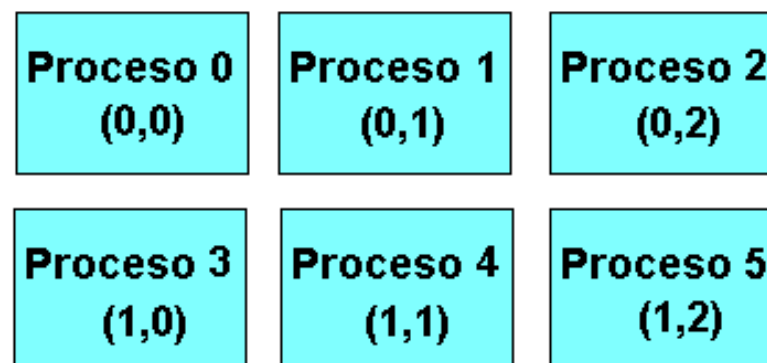
# Topología cartesiana

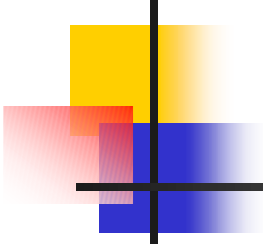
- Permite convertir el rango lineal de los procesos a coordenadas cartesianas

## Topología por defecto



## Topología cartesiana





# Práctica 5: Procesos de entrada/salida

---

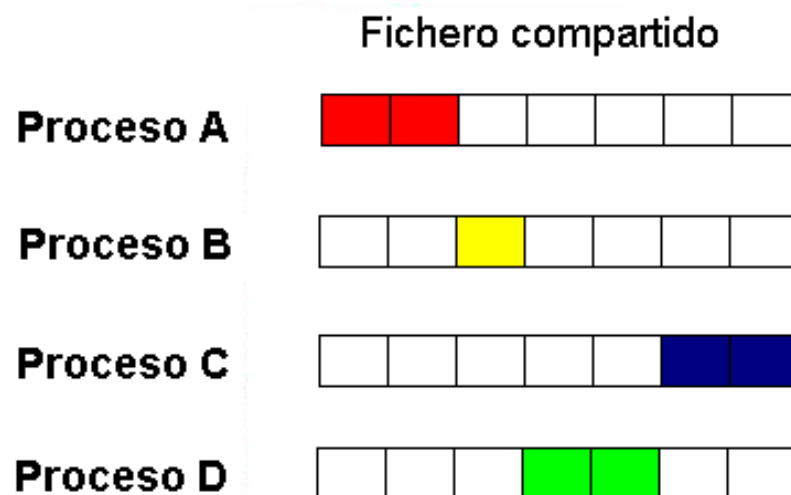


- **Objetivo:**
  - Introducción a las técnicas de entrada/salida en paralelo
- **Contenidos teóricos:**
  - Entrada/salida serie
  - Entrada/salida en paralelo
  - Entrada/salida en paralelo en MPI
- **Realización práctica:** escritura y lectura en fichero compartido



## Entrada/salida en paralelo

- Permite que varios procesos compartan un mismo fichero



# Práctica 6: Nuevos modos de envío



- **Objetivo:**
  - Ampliar el conocimiento de los modos de envío disponibles en MPI
- **Contenidos teóricos:**
  - Otros modos de envío
- **Realización práctica:** envío de datos en modo no bloqueante



# Práctica 7: Tipos de datos derivados

---



- **Objetivo:**
  - Mejorar la capacidad de intercambio de datos con el empleo de tipos de datos derivados
- **Contenidos teóricos:**
  - Tipos de datos derivados en MPI
- **Realización práctica:** suma de matrices por submatrices





# Tipos de datos derivados

**Datos en memoria**



**Tipo derivado**

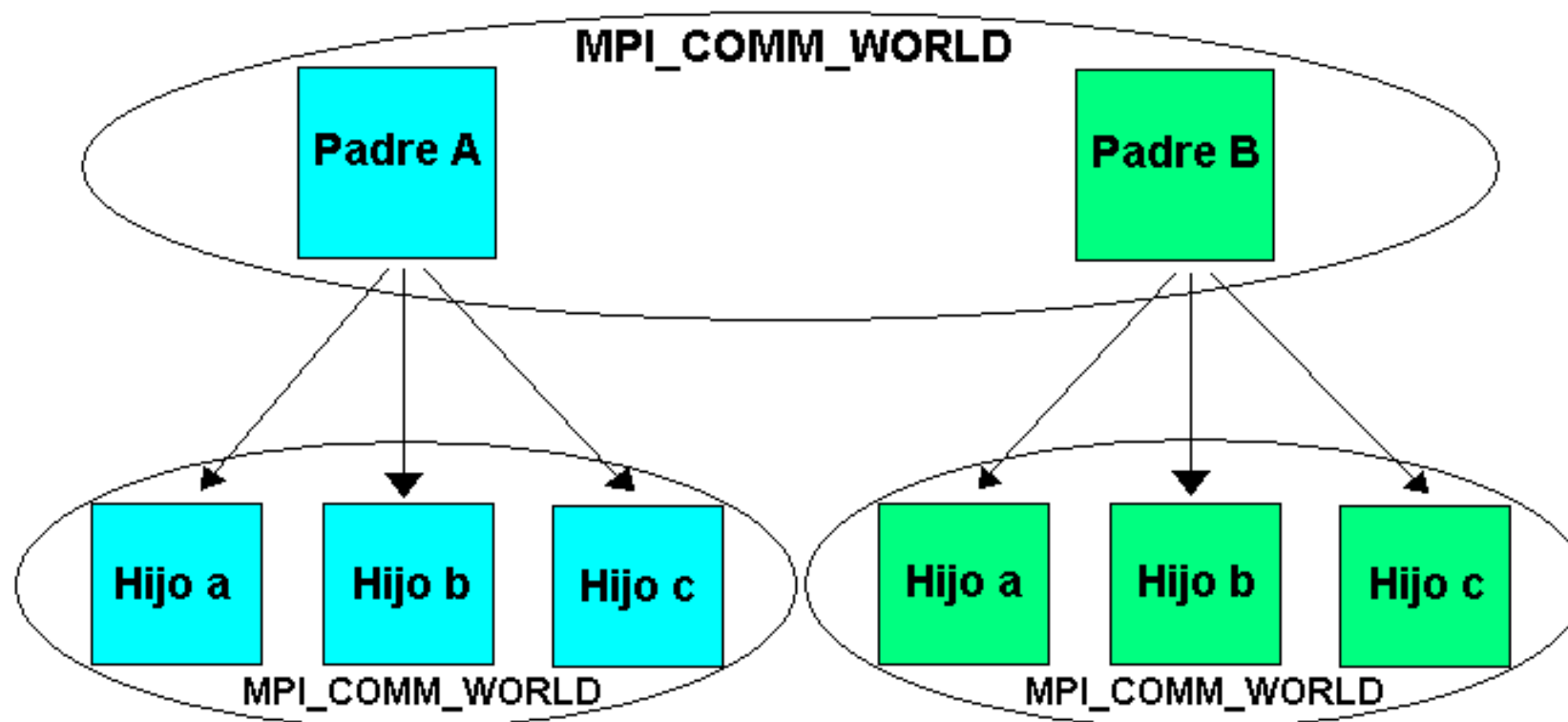


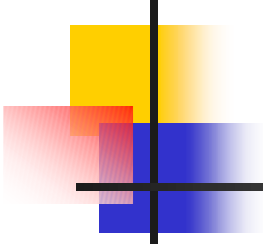
# Práctica 8: Gestión dinámica de procesos



- **Objetivo:**
  - Ensayar las técnicas de configuración dinámica del cluster como aproximación al concepto de máquina virtual
- **Contenidos teóricos:**
  - Gestión dinámica del cluster
  - Funciones de gestión de procesos en MPI-2
- **Realización práctica:** reparto de mensajes de saludo entre procesos padre e hijos

# Gestión dinámica de procesos





# Práctica 9: Ejemplo de aplicación práctica

---



- **Objetivo:**
  - Demostrar los conocimientos adquiridos desarrollando una aplicación más compleja para posteriormente evaluar el rendimiento del sistema
- **Contenidos teóricos:**
  - Sin nuevos conceptos
- **Realización práctica:** multiplicación de matrices

# Práctica 10: Medida del rendimiento

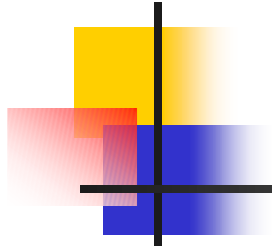


- **Objetivos:**

- Evaluar el rendimiento del sistema en diferentes situaciones
- Adquirir la capacidad de prever y extraer la potencia del sistema

- **Contenidos teóricos:**

- Grado de paralelismo
- Trabajo total realizado
- Paralelismo disponible
- Incremento de rendimiento alcanzable
- Eficiencia
- Redundancia
- Utilización
- Calidad

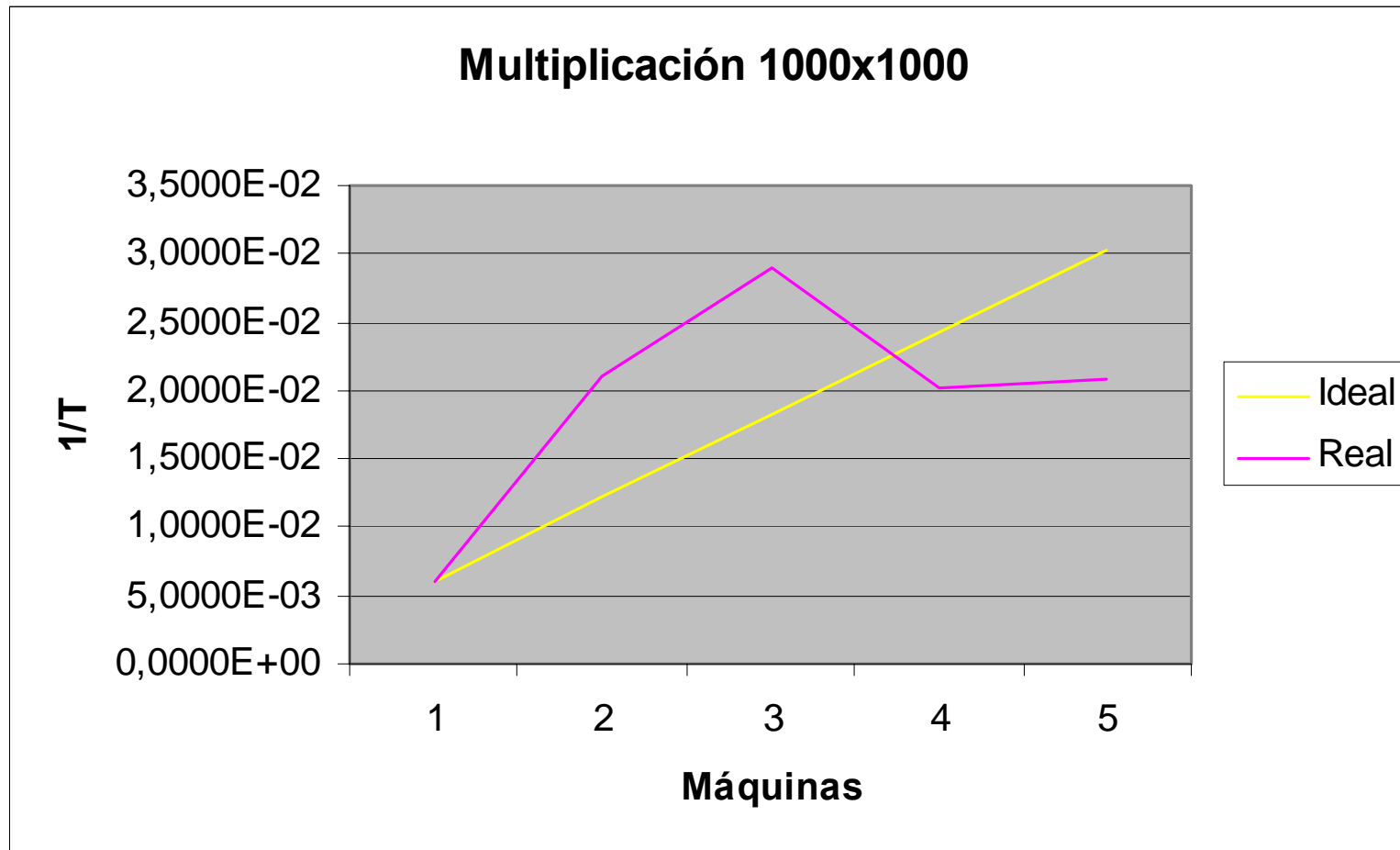


# Pruebas de rendimiento

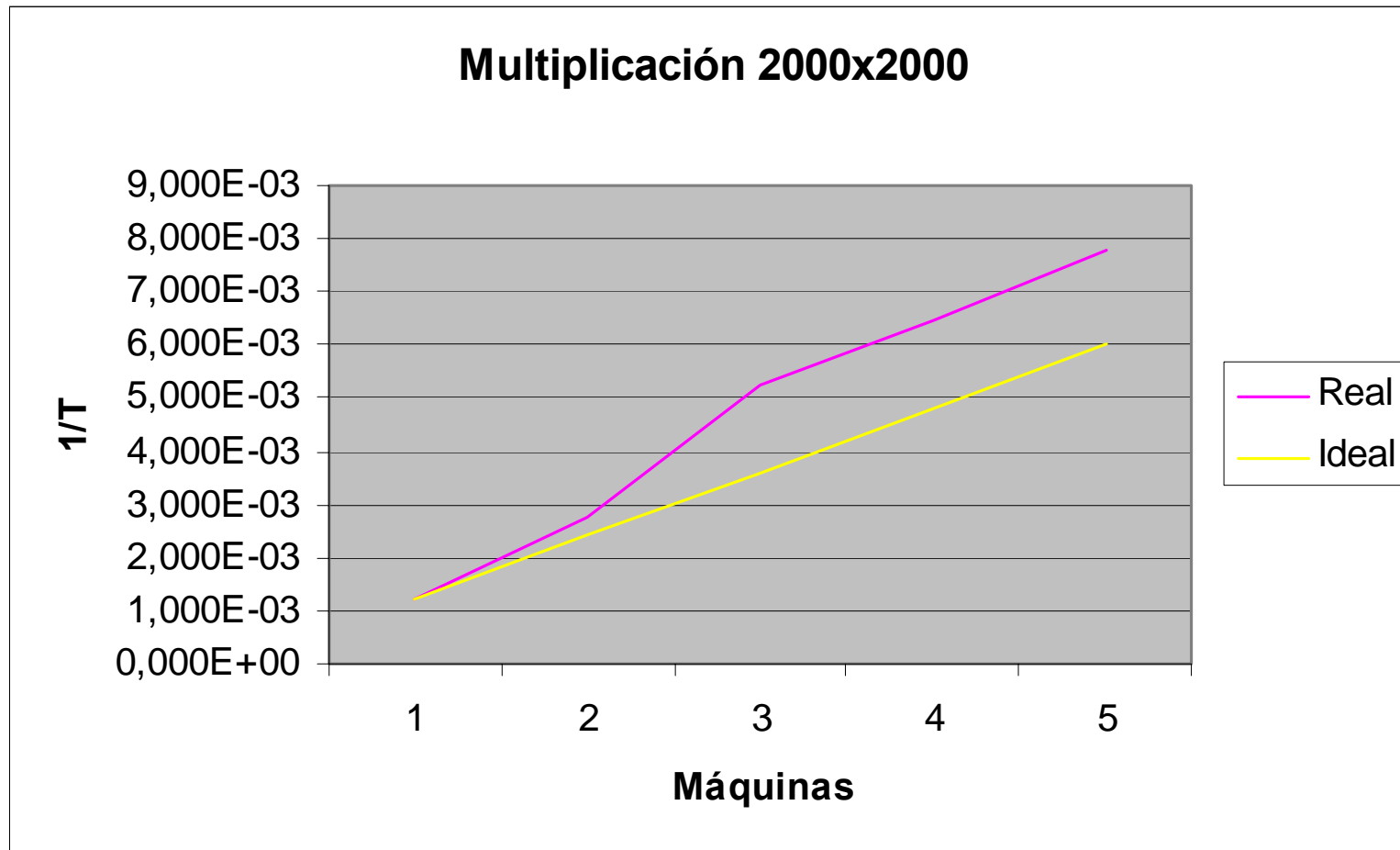


- **Programa de prueba:** multiplicación de matrices
- **Características principales:**
  - Multiplica matrices NxN
  - El tamaño se pasa como parámetro
  - Matrices de datos aleatorios de tipo "float"
  - El proceso 0 genera las matrices y envía los datos al resto; recopila los resultados
- **Pruebas realizadas:**
  - 1000x1000, 2000x2000, 3000x3000, 4000x4000

# Resultados

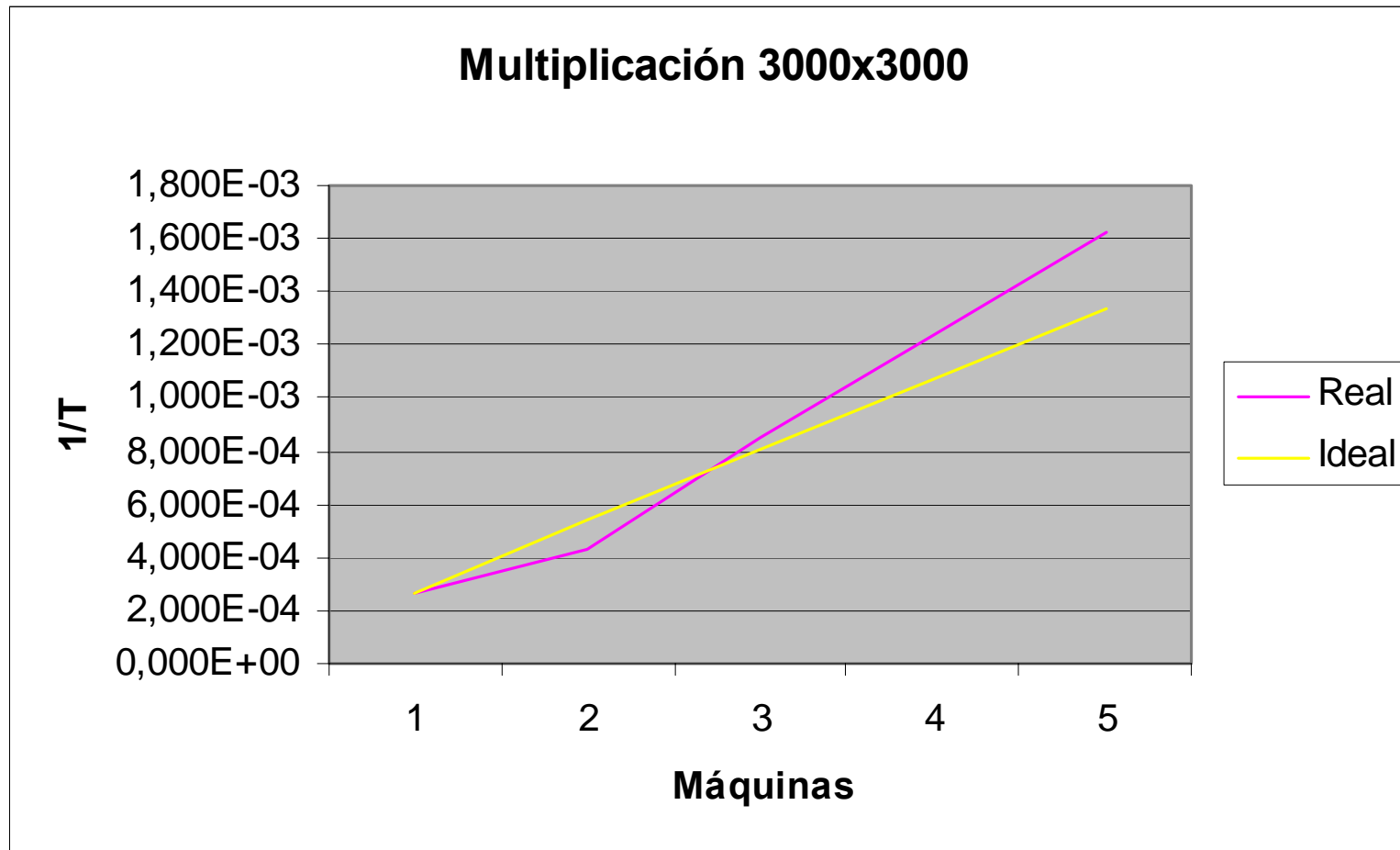


# Resultados

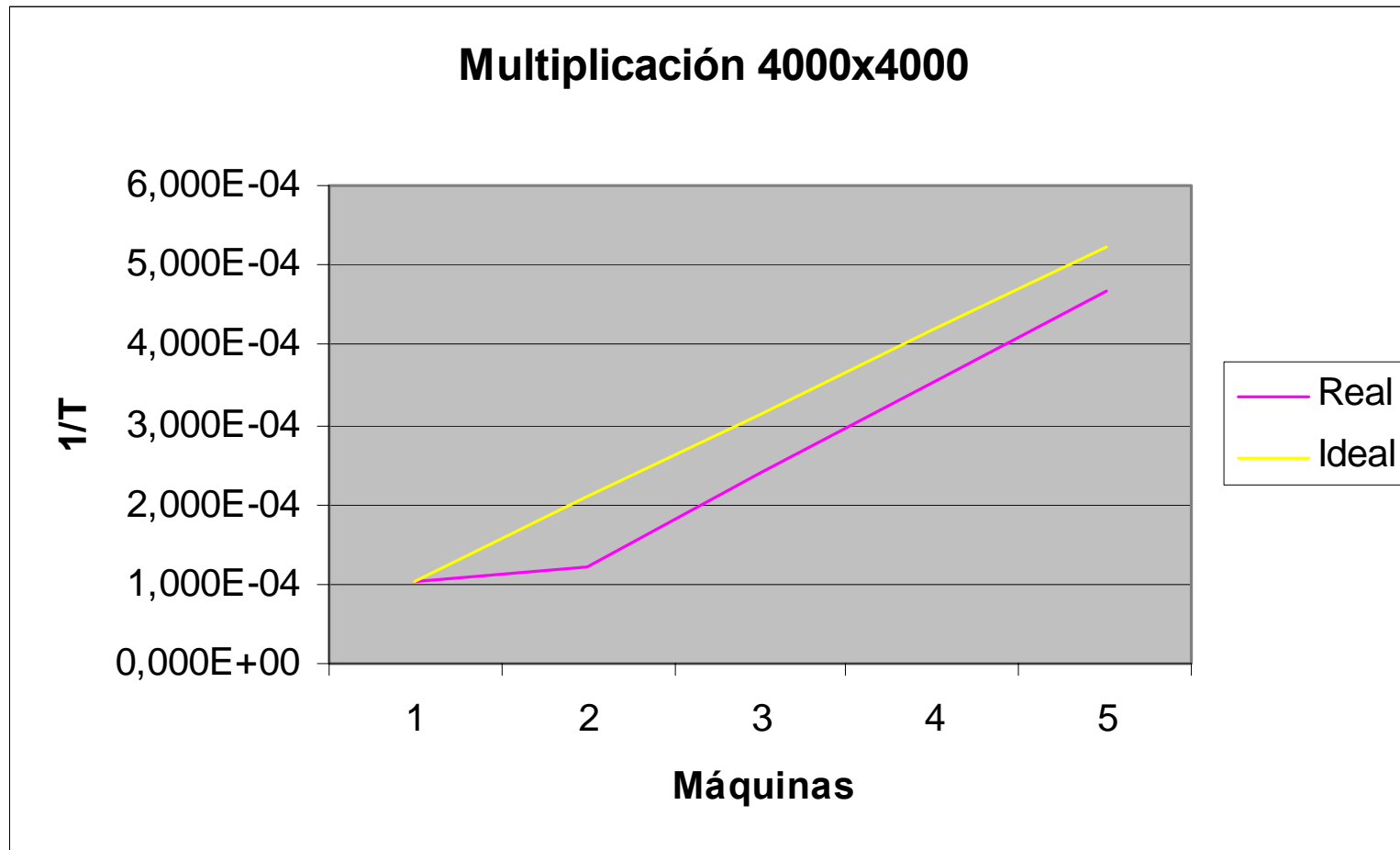




# Resultados



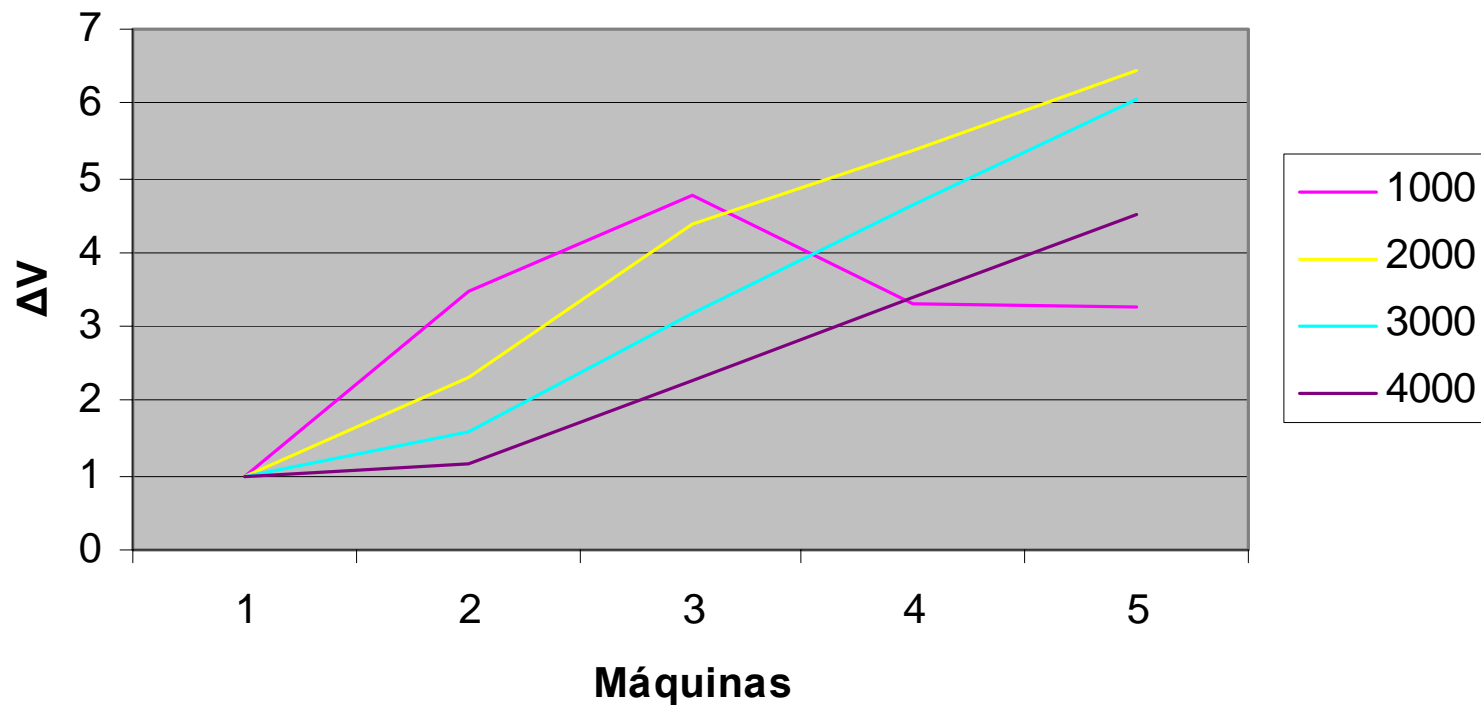
# Resultados



# Resultados



Comparativa de incremento de rendimiento





# Estudio económico



Concepto	Cantidad	Precio unitario	Precio total
Ordenador de sobremesa	6	926 €	5566 €
Sistema operativo Windows 2000 Profesional	6	169'24 €	1015'44 €
Microsoft visual estudio 2002 profesional	6	1197'45 €	7184'7 €
Switch 10/100 Mbps OvisLink	1	445'26 €	445'26 €
		Total	14211'4 €
		IVA 16%	2273'82 €



# Conclusiones

---



- Objetivos cumplidos
- La solución adoptada es buena pero no única
- La solución es “sostenible”
- El rendimiento de los clusters basados en librerías de paso de mensajes es evidente
- La fiabilidad es media-alta
- Es posible la aplicación a proyectos de investigación incluso en entorno Windows
- Exige una continua “puesta al día”



# Líneas de trabajo futuras

- Actualización constante sobre implementaciones y herramientas
- Desarrollo y prueba de aplicaciones más complejas
- Aplicación a otros lenguajes de programación: Fortran, C++, Java...
- Configuración de clusters heterogéneos y distribuidos
- Ampliación del estudio de las técnicas de programación paralela
- Compatibilidad con sistemas de reparto dinámico de carga
- Portabilidad a máquinas grandes



# Aclaraciones

# Diapositiva complementaria

## Comparativa de sistemas

