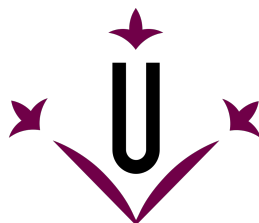


# COMPUTACIÓ DISTRIBUÏDA I APLICACIONS: PRÀCTICA 2 - MPI

Mario Fernández Rodríguez  
Àlex Codina Bracerós

Maig 2023



**Universitat de Lleida**

# Índex

<b>1</b>	<b>Estudio de eficiencia</b>	<b>3</b>
1.1	2 procesos . . . . .	3
1.2	4 procesos . . . . .	3
1.3	8 procesos . . . . .	3
1.4	Resultados Globales . . . . .	4
<b>2</b>	<b>Conclusiones</b>	<b>4</b>
2.1	Partes Opcionales . . . . .	4

# 1 Estudio de eficiencia

Para el estudio de eficiencia, primero mediremos para cada archivo lifegames el tiempo medio de 5 ejecuciones para cada X iteraciones, con esto llenaremos una tabla. Concluido la recopilación de información, daremos una media y una grafica respecto el archivo mas grande y con mas iteraciones para ver si coincide en resultado con cada test individual:

## 1.1 2 procesos

	2 PROCESOS				
ITERACIONES	10x10	100x100	1000x1000	2000x2000	4000x4000
10	0,220s	0,250s	1,460s	0,487s	2,124s
100	0,232s	0,254s	1,730s	3,359s	9,007s
1000	0,279s	0,297s	5,501s	18,456s	1m8,908s

## 1.2 4 procesos

	4 PROCESOS				
ITERACIONES	10x10	100x100	1000x1000	2000x2000	4000x4000
10	0,24	0,24	1,539s	1,653s	2,124s
100	0,223	0,252	1,956s	2,501s	5,489s
1000	0,275	0,284	3,702s	10,149s	36,326s

## 1.3 8 procesos

	8 PROCESOS				
ITERACIONES	10x10	100x100	1000x1000	2000x2000	4000x4000
10	0,245s	0,274s	1,251s	1,331s	1,311s
100	0,251s	0,251s	1,503s	1,957s	2,632s
1000	0,261s	0,280s	2,046s	4,817s	17,650s

## 1.4 Resultados Globales

	LIFEGAME X ITER1000				
Procesos	10x10	100x100	1000x1000	2000x2000	4000x4000
2	0,279s	0,297s	5,501s	18,456s	1m8,908s
4	0,275s	0,284s	3,702s	10,149s	36,326s
--	--	--	--	--	--
16	0,261s	0,280s	2,046s	4,817s	17,650s

## 2 Conclusiones

Podemos ver que en ejecuciones con bajas iteraciones y altos procesos hay un tiempo muy similar, ya que este corresponde al tiempo que tarda MPI en crear los procesos y a ponerlos a trabajar. En cambio, en ejecuciones con muchas iteraciones y un tamaño de tabla mayor, la reducción de tiempo es brutal en cuantos mas procesadores dispone la ejecución.

### 2.1 Partes Opcionales

No hemos utilizado tipos derivados ya que no intercambiamos las boards como tal, si no arrays de unsigned chars los cuales están en diferentes posiciones de memoria dadas por el malloc, dado que no tenemos matrices si no punteros de punteros y estos no se pueden enviar como matrices ya que esatias enviando posiciones de memoria.