

# UNIDAD 2: MODELO DE MAPEO Y REDUCCIÓN

## MODELO DE PROGRAMACIÓN MAPREDUCE

---

Blanca Vázquez

Febrero 2022

# ¿QUÉ ES MAPREDUCE?

- Es un modelo de programación para el procesamiento de datos distribuidos a gran escala
  - Fue inspirado en la programación funcional (LISP, 1960)
  - Se caracteriza por ser simple y elegante
  - Permite la construcción en bloques
  - Está diseñado para ser ejecutado en clústeres
- Características
  - Toma ventaja del paralelismo
  - Tolerante a fallas
  - Es extensible para diferentes aplicaciones

- Cómputo con grandes cantidades de datos
  - Astronomía, finanzas, ciencias, sitios webs....
- Cuarto paradigma de la ciencia
  - Diseño de algoritmos capaces de procesar datos en tiempo real
- No es el algoritmo, ¡son los datos!
  - Más datos, mejor precisión

# CONTANDO PALABRAS CON MAPREDUCE

## Extracto de big\_file.txt

Armstrong joined the NASA Astronaut Corps in the second group, which was selected in 1962. He made his first spaceflight as command pilot of Gemini 8 in March 1966, becoming NASA's first civilian astronaut to fly in space. During this mission with pilot David Scott, he performed the first docking of two spacecraft; the mission was aborted after Armstrong used some of his re-entry control fuel to stabilize ...



```
cat big_file.txt | tr ' ' '\n' >> out_bigfile.txt
```



Armstrong  
joined  
the  
NASA  
Astronaut  
Corps  
in  
the  
second  
group,  
which  
was  
selected  
in  
1962.  
He  
made  
his  
first  
spaceflight  
as

# CONTANDO PALABRAS CON MAPREDUCE

## Salida de out\_bigfile.txt

Armstrong  
joined  
the  
NASA  
Astronaut  
Corps  
in  
the  
second  
group,  
which  
was  
selected  
in  
1962.  
He  
made  
his  
first  
spaceflight  
as  
command  
pilot



cat out\_bigfile.txt | sort | uniq -c

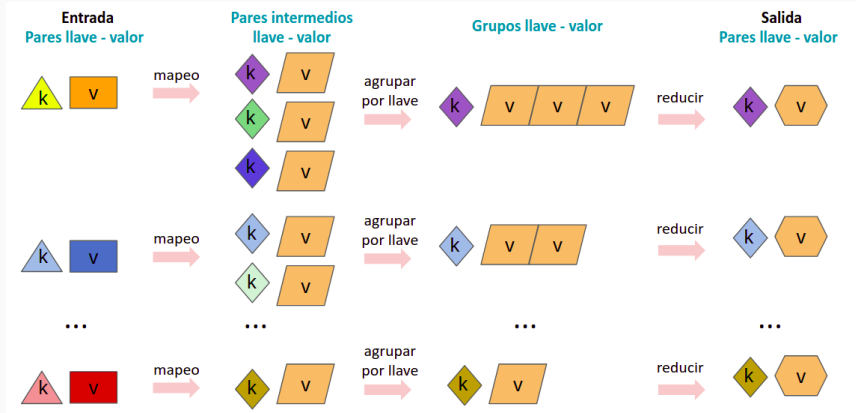


5 the  
4 in  
4 of  
3 a  
3 first  
3 to  
2 Armstrong  
2 as  
2 During  
2 he  
2 his  
2 mission  
2 pilot  
2 second  
2 spaceflight  
2 was  
1 aborted  
1 after  
1 and  
1 Apollo  
1 astronaut  
1 becoming  
1 before

# MAPREDUCE: CONTAR FRECUENCIAS DE PALABRAS EN UN DOCUMENTO

- Función de mapa
  1. Lee el documento una palabra (llave) a la vez y extrae cada ocurrencia
  2. Regresa una secuencia de pares  $(o^{(1)}, 1), \dots, (o^{(T)}, 1)$ , donde  $o^{(i)}$  es la ocurrencia de una palabra
- Agrupación por llave
  1. Agrupa las ocurrencias de cada palabra  $p_j$  (llaves con el mismo valor)
  2. Regresa una secuencia de pares de palabras con su lista de ocurrencias
- Función de reducción
  1. Realiza la función de suma (resumen, filtrado, agregación, transformación).
  2. Escribe el resultado como una secuencia de pares  $(p_1, c_1), \dots (p_2, c_2)$  (palabra y frecuencia respectivamente)

# MAPREDUCE: PROCEDIMIENTO GENERAL



# MAPREDUCE: FUNCIONES DE MAPEO Y REDUCCIÓN

- Entrada: un conjunto de pares llave - valor
- El programador especifica dos métodos
  - Función de mapeo
    - $\text{Mapeo}(k, v) \rightarrow \langle k', v' \rangle$
    - Se toma un par llave - valor y la salida es un conjunto de pares llave - valor
    - Existe un solo mapeo por cada par  $(k, v)$
  - Función de reducción
    - $\text{Reduccion}(k', \langle v' \rangle^*) \rightarrow \langle k', v'' \rangle^*$
    - Todos los valores  $v'$  con la misma llave  $k$  serán agrupados
    - Existe una sola función de reducción por cada llave única  $k'$
- Salida: un conjunto de llaves y su valor (resultado de una función)



# MAPREDUCE: EJERCICIO

**Mapeo**  
Lee una entrada  
y produce un  
conjunto de pares  
llave - valor

Hoy empecé la dieta verde:  
verde lejos la pizza,  
verde lejos los tamales,  
verde lejos las tortas,  
verde lejos el pan.

(hoy, 1)  
(empece, 1)  
(la, 1)  
(dieta, 1)  
(verde, 1)  
(verde, 1)  
(lejos, 1)  
(la, 1)  
(pizza, 1)  
(verde, 1)  
(lejos, 1)  
(los, 1)  
(tamales, 1)  
(verde, 1)  
(lejos, 1)  
(las, 1)  
(tortas, 1)  
(verde, 1)  
(lejos, 1)  
(el, 1)  
(pan, 1)

(llave, valor)

**Agrupar por  
llaves:**  
colecciona todos  
los pares con la  
misma llave

(hoy, 1)  
(empece, 1)  
(la, 1)  
(dieta, 1)  
(verde, 1)  
(verde, 1)  
(verde, 1)  
(verde, 1)  
(verde, 1)  
(lejos, 1)  
(lejos, 1)  
(lejos, 1)  
(lejos, 1)  
(lejos, 1)  
(lejos, 1)  
(pizza, 1)  
(los, 1)  
(tamales, 1)  
(las, 1)  
(tortas, 1)  
(el, 1)  
(pan, 1)

(llave, valor)

**Reducir:**  
colecciona todos  
los valores que  
pertenecen a la  
llave

(hoy, 1)  
(empece, 1)  
(la, 2)  
(dieta, 1)  
(verde, 5)  
(lejos, 4)  
(pizza, 1)  
(los, 1)  
(tamales, 1)  
(las, 1)  
(tortas, 1)  
(el, 1)  
(pan, 1)

(llave, valor)

Unicamente lecturas secuenciales

# MAPREDUCE: EJERCICIO

El programador indica  
cuántos nodos  
necesita para la tarea  
de Mapeo y cuántos  
para la tarea de  
reducción  
(5NM- 3NR)

Hoy empecé la dieta verde:  
verde lejos la pizza,  
verde lejos los tamales,  
verde lejos las tortas,  
verde lejos el pan.

**Mapeo**  
Lee una entrada  
y produce un  
conjunto de pares  
llave - valor

```
(hoy, 1)
(empece, 1)
(la, 1)
(dieta, 1)
(verde, 1)
(verde, 1)
(lejos, 1)
(la, 1)
(pizza, 1)
(verde, 1)
(lejos, 1)
(los, 1)
(tamales, 1)
(verde, 1)
(lejos, 1)
(las, 1)
(tortas, 1)
(verde, 1)
(lejos, 1)
(el, 1)
(pan, 1)
```

(llave, valor)

**Agrupar por  
llaves:**  
colecciona todos  
los pares con la  
misma llave

```
(hoy, 1)
(empece, 1)
(la, 1)
(dieta, 1)
(verde, 1)
(verde, 1)
(verde, 1)
(verde, 1)
(verde, 1)
(verde, 1)
(lejos, 1)
(lejos, 1)
(lejos, 1)
(tamales, 1)
(lejos, 1)
(pizza, 1)
(los, 1)
(tamales, 1)
(las, 1)
(tortas, 1)
(el, 1)
(pan, 1)
```

(llave, valor)

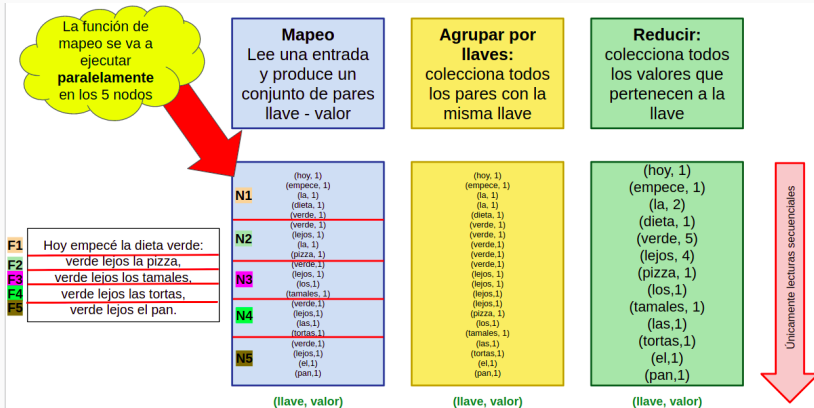
**Reducir:**  
colecciona todos  
los valores que  
pertenecen a la  
llave

```
(hoy, 1)
(empece, 1)
(la, 2)
(dieta, 1)
(verde, 5)
(lejos, 4)
(pizza, 1)
(los, 1)
(tamales, 1)
(las, 1)
(tortas, 1)
(el, 1)
(pan, 1)
```

(llave, valor)

Únicamente lecturas secuenciales

# MAPREDUCE: EJERCICIO



# MAPREDUCE: EJERCICIO

Indicamos que vamos a usar 3 nodos para reducción.

**Mapeo**  
Lee una entrada y produce un conjunto de pares llave - valor

**Agrupar por llaves:**  
colecciona todos los pares con la misma llave

**Reducir:**

F1 Hoy empecé la dieta verde:  
F2 verde lejos la pizza,  
F3 verde lejos los tamales,  
F4 verde lejos las tortas,  
F5 verde lejos el pan.

N1	(hoy, 1) (empece, 1) (la, 1) (dieta, 1) (verde, 1)	Out_m1
N2	(verde, 1) (lejos, 1) (la, 1) (pizza, 1)	Out_m2
N3	(verde, 1) (lejos, 1) (los, 1) (tamales, 1)	Out_m3
N4	(verde, 1) (lejos, 1) (las, 1) (tortas, 1)	Out_m4
N5	(verde, 1) (lejos, 1) (el, 1) (pan, 1)	Out_m5

(llave, valor)

(hoy, 1) (empece, 1) (la, 1) (dieta, 1) (verde, 1) (verde, 1) (verde, 1) (verde, 1) (verde, 1) (lejos, 1) (lejos, 1) (lejos, 1) (lejos, 1) (lejos, 1) (pizza, 1) (los, 1) (tamales, 1) (las, 1) (tortas, 1) (el, 1) (pan, 1)
--

(llave, valor)

(hoy, 1) (empece, 1) (la, 2) (dieta, 1) (verde, 5) (lejos, 4) (pizza, 1) (los, 1) (tamales, 1) (las, 1) (tortas, 1) (el, 1) (pan, 1)
--

(llave, valor)

Únicamente lecturas secuenciales

# MAPREDUCE: EJERCICIO

Indicamos que vamos a usar 3 nodos para reducción.

**Mapeo**  
Lee una entrada y produce un conjunto de pares llave - valor

F1 Hoy empecé la dieta verde:  
F2 verde lejos la pizza,  
F3 verde lejos los tamales,  
F4 verde lejos las tortas,  
F5 verde lejos el pan.

N1	(hoy, 1) (empece, 1) (la, 1) (dieta, 1) (verde, 1)	Out_m1
N2	(verde, 1) (lejos, 1) (la, 1) (pizza, 1)	Out_m2
N3	(verde, 1) (lejos, 1) (los, 1) (tamales, 1)	Out_m3
N4	(verde, 1) (lejos, 1) (las, 1) (tortas, 1)	Out_m4
N5	(verde, 1) (lejos, 1) (el, 1) (pan, 1)	Out_m5

(llave, valor)

**Agrupar por llaves:**  
colecciona todos los pares con la

**Funciones Hash**  
(enumera cada llave y determina un solo nodo)

(hoy, 1) (empece, 1) (la, 1) (la, 1) (dieta, 1) (verde, 1) (verde, 1) (verde, 1) (verde, 1) (verde, 1) (lejos, 1) (lejos, 1) (lejos, 1) (lejos, 1) (pizza, 1) (los, 1) (tamales, 1) (las, 1) (tortas, 1) (el, 1) (pan, 1)
---

(llave, valor)

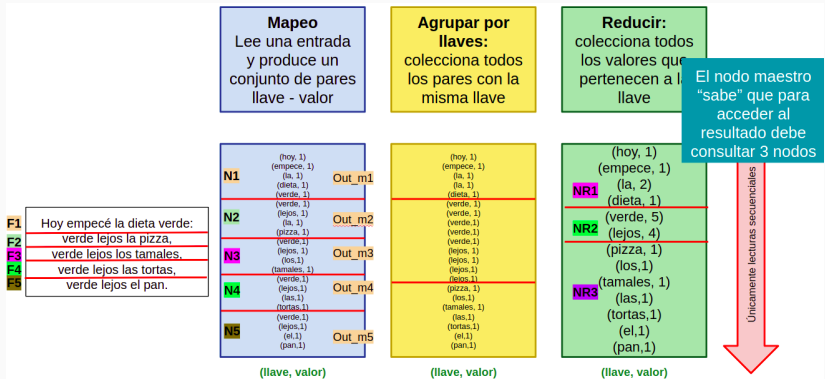
**Reducir:**  
colecciona todos los valores que pertenecen a la

(hoy, 1) (empece, 1) (la, 2) (dieta, 1) (verde, 5) (lejos, 4) (pizza, 1) (el, 1) (tamales, 1) (las, 1) (tortas, 1) (el, 1) (pan, 1)
---

(llave, valor)

Unicamente lecturas secuenciales

# MAPREDUCE: EJERCICIO



# MAPREDUCE: EJERCICIO

Las lecturas  
secuenciales son  
mucho más eficientes  
que los accesos  
aleatorios

**Mapeo**  
Lee una entrada  
y produce un  
conjunto de pares  
llave - valor

**Agrupar por  
llaves:**  
colecciona todos  
los pares con la  
misma llave

**Reducir:**  
colecciona todos  
los valores que  
pertenecen a la  
llave

F1 Hoy empecé la dieta verde:  
F2 verde lejos la pizza,  
F3 verde lejos los tamales,  
F4 verde lejos las tortas,  
F5 verde lejos el pan.

N1	(hoy, 1) (empece, 1) (la, 1) (dieta, 1) (verde, 1)	Out_m1
N2	(verde, 1) (lejos, 1) (la, 1) (pizza, 1) (verde, 1)	Out_m2
N3	(lejos, 1) (los, 1) (tamales, 1) (verde, 1)	Out_m3
N4	(lejos, 1) (las, 1) (tortas, 1) (verde, 1)	Out_m4
N5	(lejos, 1) (el, 1) (pan, 1)	Out_m5

(llave, valor)

(hoy, 1) (empece, 1) (la, 1) (dieta, 1) (verde, 1) (verde, 1) (verde, 1) (verde, 1) (lejos, 1) (lejos, 1) (lejos, 1) (lejos, 1) (pizza, 1) (los, 1) (tamales, 1) (las, 1) (tortas, 1) (el, 1) (pan, 1)
--

(llave, valor)

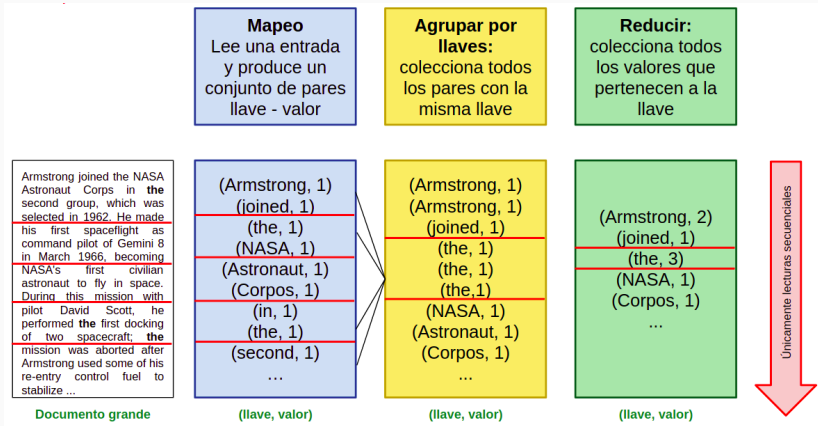
(hoy, 1) (empece, 1) NR1 (la, 2) (dieta, 1) NR2 (verde, 5) (lejos, 4) (pizza, 1) (los, 1) (las, 1) NR3 (tamales, 1) (tortas, 1) (el, 1) (pan, 1)
--

(llave, valor)

Únicamente lecturas secuenciales

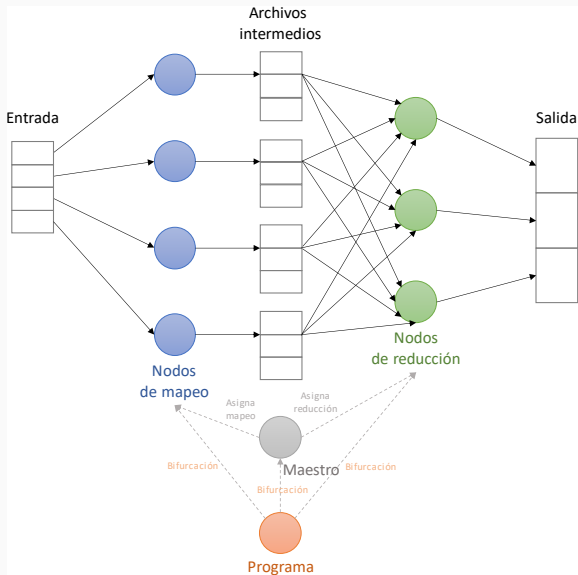
MapReduce está construido sobre lecturas de archivos secuenciales y  
nunca sobre accesos aleatorios

# MAPREDUCE: CONTEO DE PALABRAS





# MAPREDUCE: DETALLES



- Una tarea de mapeo puede producir muchos pares con la misma llave, lo cual aumenta el tamaño del archivo que se transfiere a los nodos de reducción
- Los combinadores realizan una combinación preliminar de los valores en la tarea de mapeo
  - Usualmente se usa la misma función que la de reducción
  - Solo se puede realizar si la función de reducción es asociativa y conmutativa

- Para decidir a qué nodo de reducción va una llave, se usa una función de partición por defecto:  $hash(llave) \bmod r$
- Es posible definir una función de partición distinta
  - Por ej.  $hash(autor(documento)) \bmod r$