

**ESCUELA POLITÉCNICA  
SUPERIOR DE CÓRDOBA**  
Universidad de Córdoba



## Introducción al Big Data Analytics

*Máster Universitario en Inteligencia Computacional e*

*Internet de las Cosas*

### **Tarea del bloque 1. MapReduce**

Carlos Checa Moreno



UNIVERSIDAD DE CÓRDOBA

# ÍNDICE GENERAL

---

<b>TAREA BLOQUE 1</b>	<b>MAPREDUCE .....</b>	<b>3</b>
EJERCICIO B.....		3
<i><b>Esquema</b></i> .....		3
<i><b>Explicación</b></i> .....		4
<i><b>Pseudocódigo</b></i> .....		4
EJERCICIO D .....		4
<i><b>Esquema</b></i> .....		5
<i><b>Explicación</b></i> .....		5
<i><b>Pseudocódigo</b></i> .....		5

# ÍNDICE DE FIGURAS

---

<b>FIGURA 1:</b> ESQUEMA MAPREDUCE EJERCICIO B .....	3
<b>FIGURA 2:</b> ESQUEMA MAPREDUCE EJERCICIO D .....	5

# Tarea Bloque 1

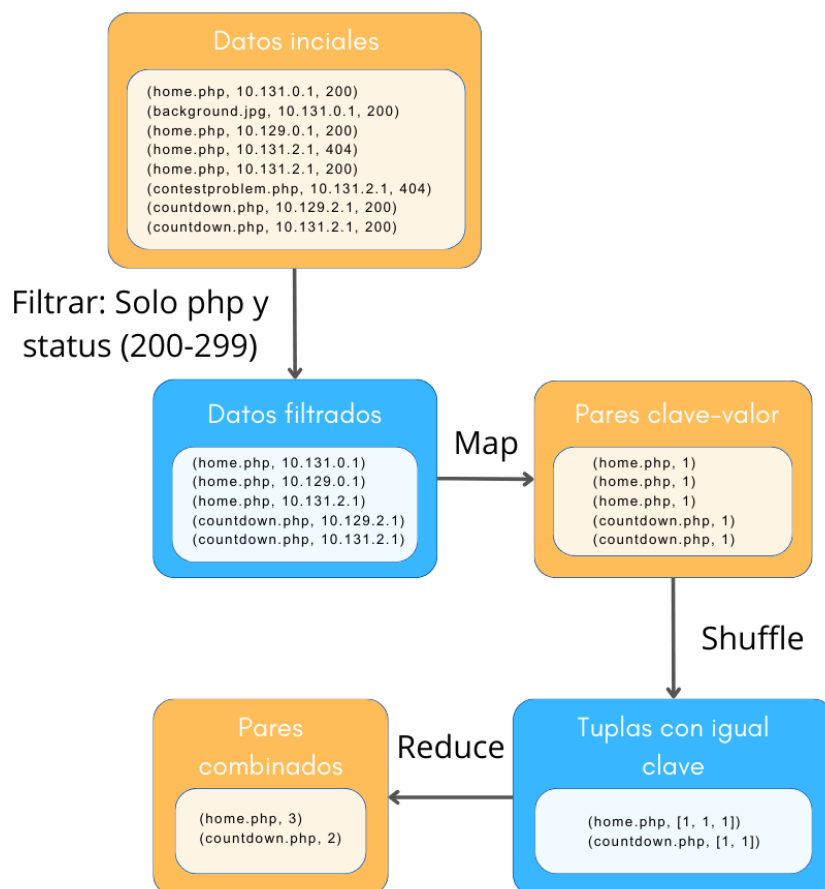
# MAPREDUCE

---

## Ejercicio B

Para aquellas solicitudes a páginas php respondidas exitosamente, queremos saber cuántos accesos únicos (distintos clientes) ha tenido cada página.

### Esquema



**Figura 1:** Esquema MapReduce ejercicio B

## Explicación

Inicialmente es necesario realizar un filtrado, se seleccionan únicamente las entradas correspondientes a páginas .php con respuestas exitosas. Posteriormente, en la fase Map, se genera un par clave-valor donde la clave es la URL de la página y el valor representa un acceso. Luego, en la etapa Shuffle, los pares con la misma clave se agrupan, y finalmente, la fase Reduce suma los valores asociados a cada clave para obtener el número total de accesos únicos por página.

## Pseudocódigo

### Función de Filtrado

```
Filtrar (URL, status, IP):  
    return (URL, IP) contiene ".php" and status >= 200 and status <= 299
```

### Función Map

```
map (key, values):  
    // keys: weblog, values: URLs  
    for each URL u in values:  
        Emit (URL, "1");
```

### Función Reduce

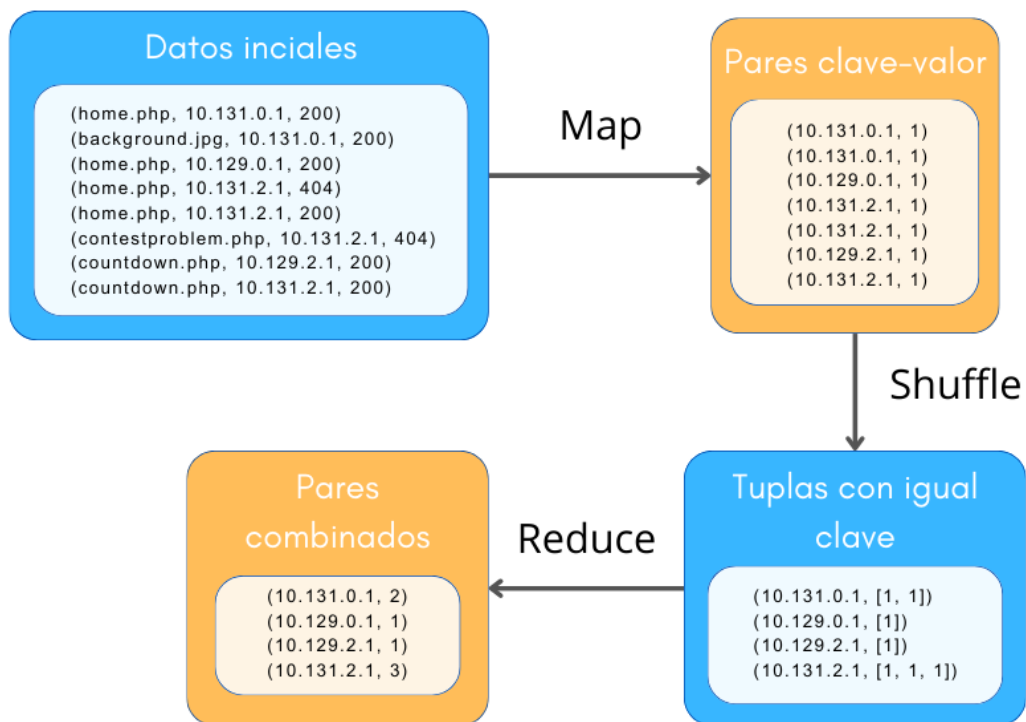
```
reduce (key, lista_de_valores):  
    // keys: URLs, valores: lista del conteo  
    suma = 0  
    for valor in lista_de_valores:  
        suma += valor  
    Emitir(key, suma)
```

## Ejercicio D

Queremos conocer la frecuencia de acceso de cada cliente a recursos de nuestro servidor.

---

## Esquema



**Figura 2:** Esquema MapReduce ejercicio D

## Explicación

En este caso, no es necesario aplicar un filtrado previo. Directamente en la fase Map, cada registro genera un par clave-valor donde la clave es la dirección IP del cliente y el valor es un contador (1). En la etapa Shuffle, se agrupan las claves idénticas (las mismas IPs), y en la fase Reduce, se suman los valores asociados para calcular la cantidad total de accesos realizados por cada cliente, representando la frecuencia de uso.

## Pseudocódigo

### Función Map

```
map (key, values):
    // keys: weblog, values: IPs
    for each IP in values:
        Emit (IP, "1");
```

### Función Reduce

```
reduce (key, lista_de_valores):
    // keys: IPs, valores: lista del conteo
```

```
suma = 0
for valor in lista_de_valores:
    suma += valor
Emitir(key, suma)
```