

# MapReduce Pattern

Por:

**Anderson Barrientos Parra**

C.c: 1017242181

**Stiven Guerra Chaverra**

C.c: 1037672655

**Sebastián Gómez Ramírez**

C.c: 1045026756

**Hermen Esteban Theran Martínez**

C.c: 1035875072

**Robinson Coronado García**

Profesor



**Arquitectura de Software - Ingeniería de Sistemas**  
**Universidad De Antioquia**

**Medellín 2020**

## Introducción

Los patrones arquitectónicos ofrecen soluciones a problemas de arquitectura de software en ingeniería de software. Dan una descripción de los elementos y el tipo de relación que tienen junto con un conjunto de restricciones sobre cómo pueden ser usados. Es una solución general y reutilizable a un problema común en la arquitectura de software dentro de un contexto dado. Los patrones arquitectónicos son similares al patrón de diseño de software pero tienen un alcance más amplio.

## Definición

MapReduce es una técnica de procesamiento y un programa de modelo de computación distribuida basada en java. El algoritmo MapReduce contiene dos tareas importantes: Map y Reduce. Map toma un conjunto de datos y lo convierte en otro conjunto de datos, en el que los elementos se dividen en tuplas (pares clave/valor). En segundo lugar, Reduce toma la salida de un Map como entrada y combina los datos tuplas en un conjunto más pequeño de tuplas dependiendo de la lógica y las necesidades del programador. Como el nombre MapReduce implica, la reducción se realiza siempre después del mapa.

## Contexto

Las empresas tienen la necesidad urgente de analizar rápidamente enormes volúmenes de los datos que generan o a los que acceden, a escala de petabytes. Los ejemplos incluyen registros de interacciones en un sitio de red social, depósitos masivos de documentos o datos, y pares de <fuente, objetivo> enlaces web para un motor de búsqueda. Programas para el análisis de estos datos deben ser fáciles de escribir, ejecutados eficientemente y ser resistentes con respecto a fallos de hardware.

## Problema

Para muchas aplicaciones con conjuntos de datos ultra grandes, la clasificación de los datos y entonces el análisis de los datos agrupados es suficiente. El problema que resuelve el patrón de reducción del mapa es realizar eficientemente un tipo de datos distribuidos y paralelos y proporcionar un medio simple para que el programador especifique el análisis a ser hecho.

## Solución

El patrón *Map-Reduce* requiere tres partes: primero, una infraestructura especializada se encargada de asignar software a los nodos de Hardware en un entorno informático masivamente paralelo y se encarga de clasificar los datos según sea necesario. Un nodo puede ser un procesador independiente o un núcleo en un chip de múltiples núcleos. La segundo y la tercera parte son dos funciones codificadas por el programador llamadas *Map* y *Reduce*.

## Ventajas de MapReduce

- La principal **ventaja** de **MapReduce** es que es fácil de escalar procesamiento de datos en múltiples nodos.
- Permite paralelizar y distribuir el procesamiento de grandes volúmenes de información.
- Al estar ubicada en el mismo servidor esta permite un procesamiento más rápido de los datos.

## Desventajas de MapReduce

- Si no se tiene un conjunto de datos grande, el sobrecalentamiento que produce el *Map-Reduce*.
- Si no se puede dividir la información en subconjuntos de tamaño similar, las ventajas del paralelismo se pierden.
- Operaciones que requieran múltiples *Reduces* son muy complejas de orquestar.

## Ejemplo de MapReduce

<b>Escenario</b>	Contar el total del salario de los empleados basándose en su género
<b>Map(Clave, Valor)</b>	<b>Clave:</b> Género, <b>Valor:</b> Salario del empleado
<b>Reduce</b>	Agrupar por clave (género), entonces tomar la suma del valor (salario) de cada grupo.

## Conjunto de la información

Nombre del empleado	Género	Salario en millones
Anderson	Masculino	8
Susana	Femenino	7
Nicolás	Masculino	10
Jhin	Masculino	4

María	Femenino	5
Fernanda	Femenino	9

### Output de la función map

Al contar el salario por género:

**(Masculino, 8), (Femenino, 7), (Masculino, 10), (Masculino, 4), (Femenino, 5), (Femenino, 9)**

Y el Input para la función Reduce será el siguiente:

**(Masculino, <8, 10, 4>), (Femenino, <7, 5, 9>)**

### Output de la función Reduce

Finalmente, el Output de la función Reduce será el siguiente:

**(Masculino, 22), (Femenino, 21)**

Lo cual corresponde a la suma del salario de los empleados según su género.

### Conclusión

Map-Reduce es una arquitectura muy útil y eficaz sólo cuando de manejo de grandes volúmenes de datos se trata, ya que es en esas circunstancias que se aprovecha al máximo la computación paralela.

## Bibliografía

- Bass, L., Clements, P., & Kazman, R. (2012). *Software Architecture in Practice* (3rd Reviseded.) [Libro electrónico]. Addison-Wesley Professional. Tomado de: [http://jz81.github.io/course/sa/Software%20Architecture%20in%20Practice%20\(3rd\).pdf](http://jz81.github.io/course/sa/Software%20Architecture%20in%20Practice%20(3rd).pdf)
- Kat, S. (2019, 10 julio). *MapReduce Design Patterns*. dzone.com. Tomado de: <https://dzone.com/articles/mapreduce-design-patterns-1>