

COMPUTACIÓN DISTRIBUIDA CON SPARK

PARADIGMAS DE LA PROGRAMACIÓN 2025
LABORATORIO 3 - FRAMEWORKS



01

OBJETIVOS DE APRENDIZAJE





Comprender el concepto de frameworks

Qué es un framework y su diferencia con una biblioteca

Inversión de control y flujo de la aplicación


Instrumentalizar el uso de framework

Cuando utilizar un framework

Aplicar el framework Spark en un proyecto existente

Cómo Spark distribuye el procesamiento

Cómo lograr código extensible



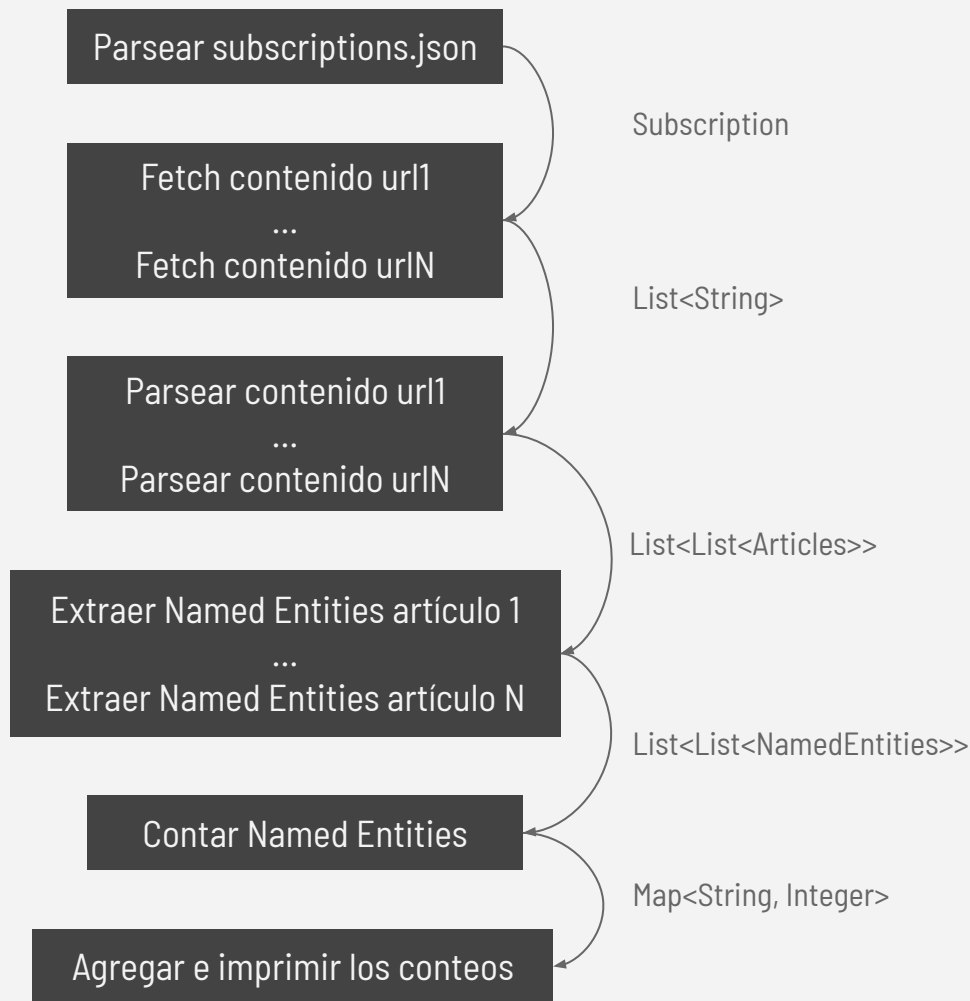
02

DESCRIPCIÓN DEL PROBLEMA



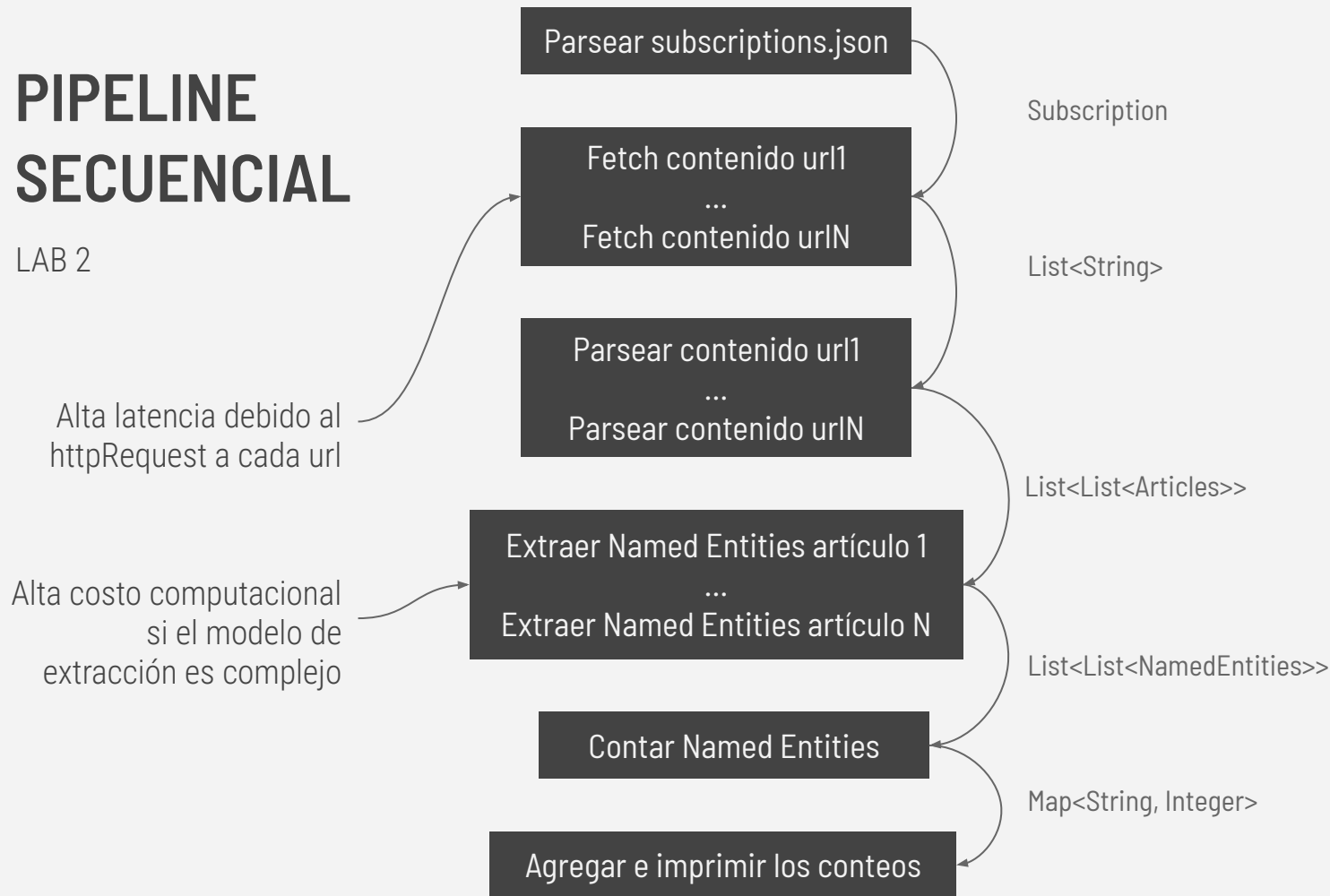
PIPELINE SECUENCIAL

LAB 2



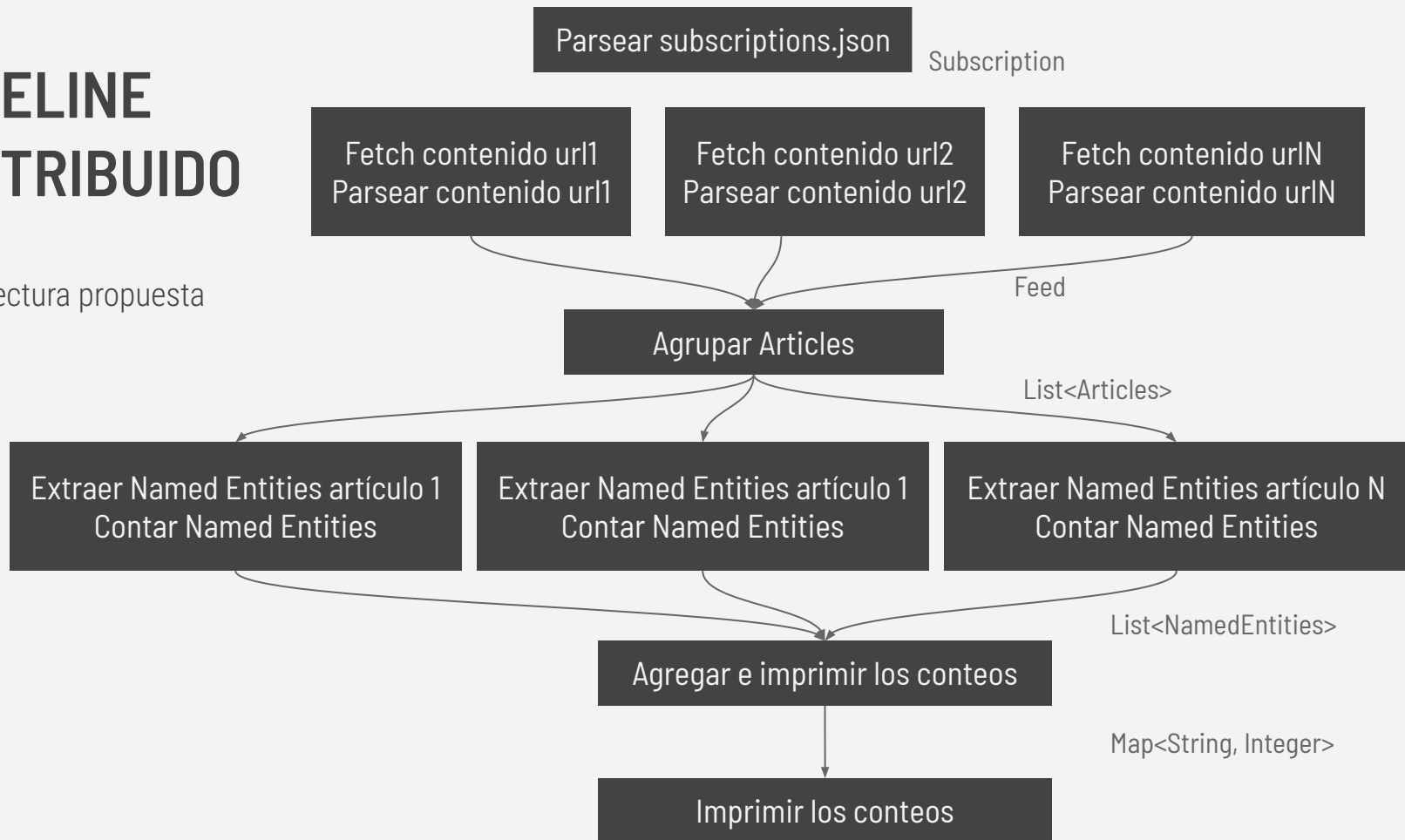
PIPELINE SECUENCIAL

LAB 2



PIPELINE DISTRIBUIDO

LAB 3
Arquitectura propuesta



¿Tiene sentido paralelizar esto usando Spark?

TIENE SENTIDO SI:

- Tenés muchas suscripciones (cientos o miles).
- Querés hacer este trabajo regularmente (ej. crawling diario).
- El procesamiento de las URL y feeds es independiente.
- Planeás escalar a un cluster o aprovechar múltiples núcleos localmente.

NO TIENE MUCHO SENTIDO SI:

- Solo hay unas pocas decenas de URLs.
- No hay presión de rendimiento o no planeás escalar.
- El overhead de iniciar Spark es mayor al beneficio.



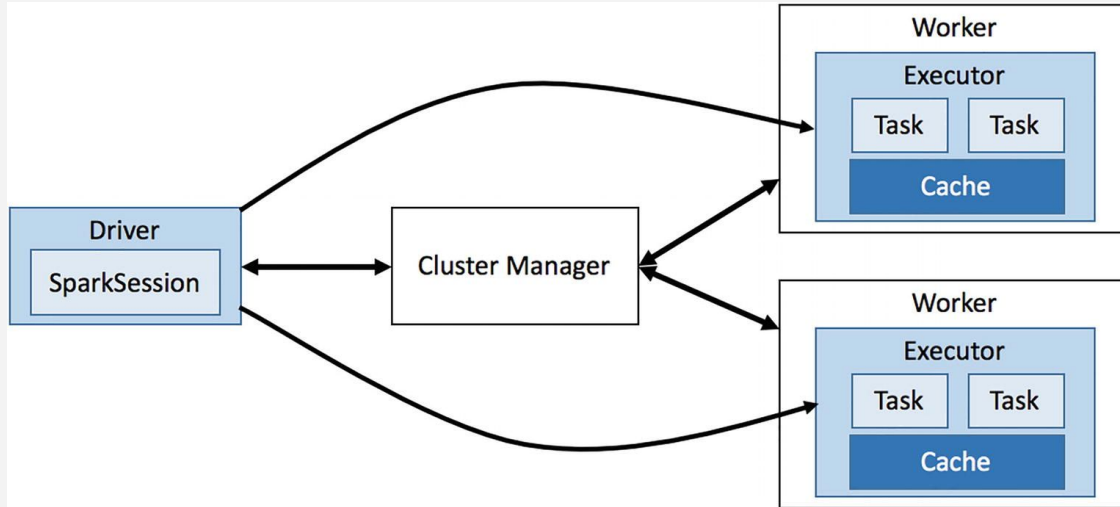
03

IMPLEMENTACIÓN





ARQUITECTURA DE SPARK



Existe una única computadora llamada Master, que coordina el trabajo.

Hay múltiples computadoras llamadas Workers (trabajadores), que realizan el procesamiento real de los datos.

EJEMPLO

```
// Crea una sesión de Spark en modo local
SparkSession spark = SparkSession
    .builder()
    .appName("WordCounter")
    .master("local[*]") // Ejecuta en modo local
    .getOrCreate();

// Lee el archivo de texto especificado como RDD de líneas
JavaRDD<String> lines = spark.read().textFile(args[0]).javaRDD();

// Divide cada línea en palabras usando espacios como separador
JavaRDD<String> words = lines.flatMap(s ->
    Arrays.asList(SPACE.split(s)).iterator());

// Asocia a cada palabra el número 1 (pares <palabra, 1>)
JavaPairRDD<String, Integer> ones = words.mapToPair(
    s -> new Tuple2<>(s, 1));
```

EJEMPLO

```
// Suma los valores (conteos) asociados a la misma palabra (reduceByKey)
JavaPairRDD<String, Integer> counts = ones.reduceByKey((i1, i2) -> i1 +
i2);

// Trae el resultado final a la máquina local como una lista de pares
List<Tuple2<String, Integer>> output = counts.collect();

// Imprime los resultados (palabra: cantidad de apariciones)
for (Tuple2<?, ?> tuple : output) {
    System.out.println(tuple._1() + ": " + tuple._2());
}

// Cierra la sesión de Spark
spark.stop();
```

ENTREGA

Todo el código a través de github.

Pueden modificar el código del lab2

Incluye preguntas en el README

Título

Configuración del entorno y ejecución

Instrucciones para el usuario sobre cómo correr las dos partes del laboratorio con spark.
Explicación del resultado que se espera luego de ejecutar cada parte.

Decisiones de diseño

Opcional. Cualquier cosa que quieran aclarar sobre la implementación del laboratorio

Conceptos importantes

1. ****Describe el flujo de la aplicación**** ¿Qué pasos sigue la aplicación desde la lectura del archivo feeds.json hasta la obtención de las entidades nombradas? ¿Cómo se reparten las tareas entre los distintos componentes del programa?

2. ****¿Por qué se decide usar Apache Spark para este proyecto?**** ¿Qué necesidad concreta del problema resuelve?

3. ****Liste las principales ventajas y desventajas que encontró al utilizar Spark.****

4. ****¿Cómo se aplica el concepto de inversión de control en este laboratorio?**** Explique cómo y dónde se delega el control del flujo de ejecución. ¿Qué componentes deja de controlar el desarrollador directamente?

5. ****¿Considera que Spark requiere que el código original tenga una integración tight vs loose coupling?****

6. ****¿El uso de Spark afectó la estructura de su código original?**** ¿Tuvieron que modificar significativamente clases, métodos o lógica de ejecución del laboratorio 2?

PREGUNTAS?

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**

Please keep this slide for attribution

