

**ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA**

**GRADO EN INGENIERÍA INFORMÁTICA**

**Curso Académico 2022/2023**

**Trabajo Fin de Grado**

**INGENIERÍA DE DATOS CON EL FRAMEWORK DE BIG DATA SPARK Y SCALA**

**Autor**: Milagros Mouriño Ursul

**Director**: Juan Manuel Serrano Hidalgo

# Resumen

El presente trabajo pretende profundizar en Apache Spark, la cual es una de las herramientas más importantes hoy en día existentes para el procesamiento y análisis de *big data*. Permite a las organizaciones el cómputo de grandes conjuntos de datos de manera eficiente para extraer información que permita realizar una toma informada de decisiones y hacer prosperar su negocio. Este motor de computación distribuida está escrito en el lenguaje de programación funcional Scala. Se promueve el desarrollo de código siguiendo las técnicas del paradigma de programación sobre el que apoya, ya que la utilización de funciones matemáticas permite una mejor división de las tareas entre los nodos del clúster respecto a la utilización de iteraciones y estructuras de datos mutables. De igual modo, permite la reutilización de código al usar las APIs de las que consta el framework como GraphFrames o GraphSQL y la mantenibilidad de este permite un mantenimiento y una evolución eficientes y rentables del código base a lo largo del tiempo.

# Palabras clave

Big Data, Apache Spark, Programación funcional

# Índice

[Resumen 2](#_Toc125455808)

[Palabras clave 2](#_Toc125455809)

[Índice 3](#_Toc125455810)

[1. Introducción 4](#_Toc125455811)

[2. Objetivos 6](#_Toc125455812)

[3. Descripción informática 7](#_Toc125455813)

[3.1. Creación del dataset de partidas 7](#_Toc125455814)

[3.1.1. Especificación 8](#_Toc125455815)

[3.1.2. Arquitectura 9](#_Toc125455816)

[3.1.3. Diseño 10](#_Toc125455817)

[3.2. Consultas con GraphFrames 11](#_Toc125455818)

[3.3. Despliegue en AWS 12](#_Toc125455819)

[4. Experimentos 13](#_Toc125455820)

[4.1. Resultados de las consultas 13](#_Toc125455821)

[4.2. Rendimientos comparados 14](#_Toc125455822)

[5. Conclusiones 15](#_Toc125455823)

[5.1. Trabajo futuro 16](#_Toc125455824)

[6. Bibliografía 17](#_Toc125455825)

[7. Apéndices 18](#_Toc125455826)

# 1. Introducción

*Big Data* se refiere al gran volumen de datos estructurados y no estructurados que generan y recopilan organizaciones e individuos para mejorar sus negocios. Estos datos pueden recopilarse de redes sociales, sensores, sistemas transaccionales, dispositivos móviles entre otras muchas fuentes. El sistema de recomendaciones de Amazon no existiría de no ser por el procesamiento y análisis de búsquedas usuales de compradores. Estos datos suelen ser demasiado grandes y complejos para ser procesados ​​y analizados utilizando las herramientas y técnicas tradicionales.

Apache Spark es un framework de código abierto creado para el procesamiento de estos datos masivos. Es escalable, ya que puede ejecutarse desde un ordenador local hasta miles de clústeres. Fue desarrollado en la Universidad de California a principios de la década de 2010 y donado a la Apache Software Foundation en 2014 permitiendo la creación de APIs más potentes y bibliotecas de alto nivel dentro del framework. Se impuso frente a Apache Hadoop, el cual también permite el procesamiento y análisis por lotes de datos mediante el sistema de archivos distribuidos de Hadoop (HDFS) y el paradigma MapReduce para el cálculo a través del nodo máster y los esclavos, pero el primero es más rápido al permitir almacenar y procesar datos en memoria, mientras que el último está basado en disco al operar con los datos que residen en HDFS. Además, Spark presenta flexibilidad al poseer APIs que permiten a los desarrolladores escribir código en los lenguajes de programación Java, Scala, Python y R. No solo permite el procesamiento por lotes de datos, sino también el de flujos de datos con la librería Spark Streaming, lo cual puede ser interesante, por ejemplo, para realizar consultas sobre tweets que emergen instantáneamente mediante la API de Twitter. Asimismo, goza de librerías complementarias como SparkSQL para la realización de consultas sobre los datos; MLib para observar patrones o tendencias en los datos mediante *machine learning* o GraphX para el procesamiento y análisis de grafos.

La programación funcional es un estilo de programación que se centra en el uso de funciones puras y evita el uso de estado mutable y efectos secundarios. En Spark, se pueden utilizar tanto enfoques de programación funcional como de programación imperativa. Sin embargo, el enfoque funcional se adapta mejor a Spark debido a que al utilizar funciones puras, es más fácil dividir y distribuir el trabajo de procesamiento de datos a través de varios nodos de un clúster, lo que permite una mayor escalabilidad y rendimiento. Además, el uso de funciones puras y la evitación del estado mutable también ayuda a evitar errores y problemas de concurrencia en el procesamiento de datos distribuidos. Por lo tanto, el enfoque funcional se considera una buena práctica al trabajar con Spark.

Amazon Web Services (AWS) es una plataforma informática en la nube ofrecida por Amazon.com. Estos servicios operan desde 12 regiones geográficas en todo el mundo. Spark se puede usar con una variedad de fuentes de datos, incluido HDFS, Amazon S3 y sistemas de archivos locales. Se puede ejecutar en una variedad de administradores de clústeres, incluidos Hadoop YARN, Apache Mesos y su propio administrador de clústeres Spark independiente. Sim embargo, AWS proporciona un servicio llamado EMR (Elastic MapReduce) que permite ejecutar Spark en un clúster de instancias EC2. Esto le permite escalar fácilmente su clúster de Spark según sea necesario y pagar solo por los recursos que utiliza.

# 2. Objetivos

(1-2 páginas)

# Descripción informática

## Creación del dataset de partidas

(15-20 páginas)

### 3.1.1. Especificación

### 3.1.2. Arquitectura

### 3.1.3. Diseño

(irá el mayor número de páginas – se explica el código)

## 3.2. Consultas con GraphFrames

(15-20 páginas)

## 3.3. Despliegue en AWS

# 4. Experimentos

## 4.1. Resultados de las consultas

(3-4 páginas)

## 4.2. Rendimientos comparados

(2 páginas)

# 5. Conclusiones

(2-3 páginas)

## 5.1. Trabajo futuro

(1 página)

# Bibliografía

(1 página)

Chambers, B., & Zaharia, M. (2018). Spark: The definitive guide: Big data processing made simple. " O'Reilly Media, Inc.".

# Apéndices