{ Temario: // }

INTRODUCCIÓN A BIG DATA {con Pyspark}

AMAT

AMAT

{ Temario }

1:// Introducción a Big Data y Apache Spark

1. Arquitectura de Spark: driver, executors, jobs/ stages/tasks
2. Instalación y configuración de PySpark
3. SparkSession y primeros pasos con DataFrames
4. Práctica: primer notebook/ proyecto y lectura de un dataset

2:// DataFrames y API moderna de PySpark

1. Evaluación perezosa (lazy) y linaje (lineage)
2. Transformaciones y acciones en DataFrames
3. Expresiones con Column y funciones built-in (select, withColumn, filter)
4. Manejo de nulos, tipos y casting
5. Práctica: limpieza y transformación de datos con DataFrames

3:// RDDs

1. Qué es un RDD y cuándo usarlo (vs DataFrames)
2. Transformaciones y acciones básicas (map, filter, reduce)
3. Particionado, persistencia y consideraciones de rendimiento
4. Broadcast variables y accumulators (conceptos)
5. Práctica: ejercicios básicos con RDDs y conversión (RDD,DataFrame)

4:// Lectura y escritura de datos

1. Fuentes de datos soportadas: CSV, JSON, Parquet, ORC
2. Esquemas: inferencia vs schema explícito, validación y evolución básica
3. Particionado de salida y buenas prácticas de almacenamiento
4. Introducción a tablas lakehouse: Delta Lake / Apache Iceberg (conceptos y uso básico)
5. Práctica: ingesta, estandarización de esquema y escritura a Parquet y tabla

AMAT

{ Temario }

5:// Spark SQL para análisis y transformación

1. Introducción a Spark SQL y Catalyst (visión general)
2. Registro de DataFrames como vistas/tablas y consultas SQL
3. Joins, agregaciones y funciones de ventana (window functions)
4. SQL scripting y parametrización (cuando aplique en la plataforma)
5. Práctica: resolución de casos con SQL sobre tablas/vistas

6:// Transformaciones avanzadas y UDFs (con enfoque en rendimiento)

1. Funciones avanzadas: strings, fechas, arrays y structs
2. Explode, higher-order functions (filter/transform) y manejo de semi-estructurados
3. UDF vs pandas UDF vs funciones nativas: cuándo usar cada una
4. Apache Arrow y consideraciones de performance
5. Práctica: transformación compleja con funciones nativas y (opcional) pandas UDF

7:// Machine Learning con PySpark (MLlib)

1. Introducción a MLlib y el flujo de trabajo en Spark
2. Feature engineering básico (VectorAssembler, StringIndexer, OneHotEncoder)
3. Pipelines (Transformers/ Estimators), entrenamiento y evaluación
4. Introducción ligera a tracking (MLflow o equivalente de la plataforma)
5. Práctica: modelo simple (clasificación o regresión) con pipeline

8:// IA aplicada a datos con PySpark (GenAI)

1. Casos de uso: clasificación, extracción de entidades, resúmenes, traducción
2. Embeddings y búsqueda semántica: conceptos y patrones de generación a escala
3. Estrategias para inferencia batch: particionado, rate limiting, caching y control de costos
4. Herramientas: copilots (IDE/ notebook), AI Functions/LLM endpoints, Hugging Face con
5. pandas UDF, Spark NLP (opcional)
6. Práctica: enriquecimiento de un dataset de texto con IA y evaluación básica de calidad

AMAT

{ Temario }

9:// Optimización, troubleshooting y calidad de datos

1. Planes de ejecución (explain) y Adaptive Query Execution (AQE) - visión general
2. Shuffles, particionado, cache/ persist, y estrategias de join (broadcast/sort-merge)
3. Detección de skew y técnicas comunes de mitigación
4. Uso de Spark UI para diagnóstico (jobs, stages, storage, shuffle)
5. Práctica: optimizar un pipeline y justificar los cambios

10:// Structured Streaming

1. Introducción a Structured Streaming (micro-batch/continuous - visión general)
2. Fuentes y sinks, checkpoints y tolerancia a fallos
3. Ventanas, watermarks y agregaciones en streaming