Universidad del Valle de Guatemala
Facultad de Ingeniería
Departamento de Ciencias de la Computación
CC3084 – Data Science
Semestre II – 2025

Laboratorio 9

Fundamentos de DataFrames de Spark y Python

Objetivo:

- Al finalizar, podrá:
  1. Iniciar una SparkSession y trabajar con PySpark DataFrames.
  2. Cargar un CSV con encabezados e inferencia de tipos.
  3. Explorar estructura: columnas, esquema y muestras.
  4. Ejecutar descriptivos y agregaciones.
  5. Aplicar filtros, transformaciones y creación de columnas.
  6. Calcular medidas estadísticas (p. ej., correlación de Pearson).
  7. Realizar consultas temporales (día con máximo precio; máximos por año).
  8. Comunicar hallazgos de forma ordenada y reproducible.

Sobre los datos:

- Archivo adjunto en Canvas: `walmart_stock.csv`
- Periodo: 2012–2017

Requisitos previos:

- Python 3.9+ (o entorno equivalente en Google Colab)
- Apache Spark 3.x con PySpark (o `pyspark` preinstalado en Colab)
- Conocimientos básicos de: tipos de datos, funciones de agregación, y uso de notebooks.

Configuración del entorno:

Opción A — Local:

```
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName("Lab Spark DF — Walmart").getOrCreate()
print("Spark version:", spark.version)
```

Opción B — Colab (sugerida si no tiene Spark local):

```
!pip -q install pyspark
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName("Lab Spark DF — Walmart").getOrCreate()
print("Spark version:", spark.version)
```

Instrucciones:

Complete en orden y deje **toda** la evidencia en el notebook

(Ver tareas detalladas en Lab9_Fund_Spark.ipynb incluido)

Buenas prácticas

- Comente bloques no triviales.
- Nombres de variables claros (`df_prices`, `max_high_year`, etc.).
- Reutilice resultados intermedios para evitar recalcular.
- Si usa Colab, fije versiones cuando sea necesario.
- Cierre la sesión de Spark al final si corres local: `spark.stop()`.

Errores comunes

- Olvidar `inferSchema=True` → todo se carga como `string`.
- Mezclar API RDD con DataFrames sin necesidad.
- Usar funciones de Python puras en `withColumn` (usa `pyspark.sql.functions`).
- Intentar graficar DataFrames de Spark directamente: primero convierte a Pandas con `.toPandas()` en subconjuntos pequeños.

Rúbrica de Evaluación:

Criterio	Puntaje Máximo
Preparación del ambiente - (10) SparkSession creada sin errores; versiones y entorno claros.	10 puntos
Carga y documentación de datos - (8) CSV cargado con `header` y `inferSchema` correctos.	15 puntos

- (7) Comentarios breves sobre las columnas y supuestos.	
Exploración básica - (4) Lista de columnas. - (3) `printSchema()` bien interpretado. - (3) `show(5)` con observaciones puntuales.	10 puntos
Descriptivos - (6) `describe()` ejecutado y leído correctamente. - (4) Al menos 2 interpretaciones numéricas.	10 puntos
Agregaciones y filtros - (5) Máx./mín. de `Volume` correctos. - (5) Conteo de días con `Close < 60` correcto.	10 puntos
Ingeniería de características - (8) Columna `Tasa_HV = High/Volume` correcta y con tipo numérico. - (2) Justificación breve del indicador.	10 puntos
Métricas estadísticas - (7) Correlación `High`–`Volume` calculada. - (3) Interpretación del valor (signo y magnitud).	10 puntos
Consultas temporales - (5) Día con `High` máximo identificado. - (5) Máximo `High` por año con agrupación y orden correctos	10 puntos
Comunicación de resultados - (6) Conclusiones finales claras y concisas (5–10 líneas). - (4) Orden, legibilidad y limpieza del notebook.	10 puntos
Estilo y calidad de código - (5) Convenciones PEP8 razonables, nombres significativos y ausencia de código muerto.	5 puntos
Puntaje Total	**100 puntos**