Big Data Aplicado	
Duración: hasta 11 de febrero 2026	***Organización: 2 personas máx.***

Reto: Observatorio de telemetría en tiempo real (edificio/planta)

Construir un sistema Big Data que recibe eventos en tiempo real (telemetría) a **través de Kafka**, los procesa con Spark Structured Streaming, los **almacena en HDFS** como Data Lake, y los **publica** como tablas consultables en **Hive** para **explotarlas con Spark SQL**.

Generar **alertas** en tiempo real (por umbrales, picos y/o dispositivos "offline") y preparar un conjunto de **consultas analíticas** (KPIs) sobre los datos almacenados.

Qué aprenderás en el reto:

- Qué es un **Data Lake**: estructura de datos en HDFS por capas (Bronze/Silver/Gold).
  1. **Bronze / Raw**
     - Datos tal como llegan (ej. eventos JSON de Kafka)
     - Se guardan en HDFS: /datalake/bronze/telemetry/dt=2026-01-21/...
  2. **Silver / Clean**
     - Datos ya limpios y tipados (fechas bien, números bien, sin campos basura)
     - Suelen ser Parquet, particionados
  3. **Gold / Curated**
     - Datos agregados y listos para negocio/BI (KPIs por hora/día, tablas finales)

- Qué es un **DWH** (Data Warehouse) y cómo, en este reto, se aproxima con tablas "curadas" (Gold) en Hive.
- Cómo integrar **Kafka** → **Spark Streaming** → **HDFS/Hive** y explotar con **Spark SQL**.

Entradas que se entregan por parte de profesorado:

- DIMENSIONES (las facilita el profesorado con el enunciado y el video explicativo): devices.json, locations.json, metrics.json. y un evento de ejemplo de telemetry_events.ndjson.
- EVENTOS: los enviará el profesorado al topic Kafka acordado (formato JSON por evento).
- Video explicativo del reto.

Objetivos y Competencias

Competencias técnicas e indicadores de logro (R.A.):

RA1: Gestiona soluciones a problemas propuestos, utilizando sistemas de almacenamiento y herramientas asociadas al centro de datos

Criterios de evaluación:

a) Se ha caracterizado el proceso de diseño y construcción de soluciones en sistemas de almacenamiento de datos.

b) Se han determinado los procedimientos y mecanismos para la ingestión de datos.

c) Se ha determinado el formato de datos adecuado para el almacenamiento.

d) Se han procesado los datos almacenados.

RA2. Gestiona sistemas de almacenamiento y el amplio ecosistema alrededor de ellos, facilitando el procesamiento de grandes cantidades de datos, sin fallos y de forma rápida.

Criterios de evaluación:

a) Se ha determinado la importancia de los sistemas de almacenamiento para depositar y procesar grandes cantidades de cualquier tipo de datos rápidamente.

b) Se ha comprobado el poder de procesamiento de su modelo de computación distribuida.

d) Se ha determinado que se pueden almacenar tantos datos como se desee y decidir cómo utilizarlos más tarde

RA3. Genera mecanismos de integridad de los datos, comprobando su mantenimiento en los sistemas de ficheros distribuidos y valorando la sobrecarga que conlleva en el tratamiento de los datos.

Criterios de evaluación:

a) Se ha valorado la importancia de la calidad de los datos en los sistemas de ficheros distribuidos.

c) Se ha reconocido que los sistemas de ficheros distribuidos implementan una suma de verificación para la comprobación de los contenidos de los archivos.

d) Se ha reconocido el papel del servidor en los procesos previos a la suma de verificación

RA4. Realiza el seguimiento de la monitorización de un sistema, asegurando la fiabilidad y estabilidad de los servicios que se proveen.

Criterios de evaluación:

a) Se han aplicado herramientas de monitorización eficiente de los recursos.

b) Se han recogido métricas, procesamiento y visualización de los datos.

d) Se ha comprobado que las herramientas usadas ofrecen un rendimiento elevado con rapidez.

RA5. Valida las técnicas de Big Data para transformar una gran cantidad de datos en información significativa, facilitando la toma de decisiones de negocios.

Criterios de evaluación:

a) Se han seleccionado gran cantidad de datos estructurados y no estructurados para reforzar la función de BI.

b) Se ha realizado la limpieza y transformación de datos en base a los objetivos predeterminados.

e) Se ha evaluado e interpretado la información extraída de los datos y su influencia en el triunfo de diferentes negocios.

Objetivos de aprendizaje:

h) Utilizar soluciones de Big Data para integrar sistemas de explotación de datos.

i) Analizar y evaluar soluciones Big Data para su implantación en las funcionalidades, procesos y sistemas de decisiones.

k) Determinar la solución de Inteligencia Artificial y Big Data para configurar las herramientas y lenguajes específicos.

l) Aplicar técnicas Big Data para gestionar los datos de la organización y obtener conocimiento a partir de ellos.

m) Analizar y utilizar los recursos y oportunidades de aprendizaje relacionados con la evolución científica, tecnológica y organizativa del sector y las tecnologías de la información y la comunicación, para mantener el espíritu de actualización y adaptarse a nuevas situaciones laborales y personales.

n) Desarrollar la creatividad y el espíritu de innovación para responder a los retos que se presentan en los procesos y en la organización del trabajo y de la vida personal.

p) Identificar y aplicar parámetros de calidad en los trabajos y actividades realizados en el proceso de aprendizaje, para valorar la cultura de la evaluación y de la calidad y ser capaces de supervisar y mejorar procedimientos de gestión de calidad

1. Tareas a realizar

- **Comprender el reto y preparar el Data Lake**:

  Leer el enunciado, entender qué datos llegan por Kafka y qué significan las DIM proporcionadas, definir la estructura de **carpetas en HDFS** (bronze/silver/gold) y comprobar permisos/rutas de trabajo.

  Estructura de carpetas:

```
/datalake/
        bronze/
                telemetry/
                  dt=YYYY-MM-DD/
                    hour=HH/
                      part-*.json (o parquet)
        silver/
                telemetry/
                  dt=YYYY-MM-DD/
                    part-*.parquet
                alerts/
                  dt=YYYY-MM-DD/
                    part-*.parquet
        gold/
                kpi_hourly/
                        dt=YYYY-MM-DD/
                          part-*.parquet
                kpi_daily/
                        dt=YYYY-MM-DD/
                          part-*.parquet
```

- **Crear el modelo en Hive**:

  Crear en Hive las tablas necesarias apuntando a las rutas del Data Lake, definiendo el esquema correcto, el formato (Parquet) y la partición por fecha (dt), dejando listas las tablas para que Spark pueda escribir.

  1) Bronze (opcional como tabla, útil para debug)
     bronze_telemetry_raw: `(string JSON o columnas básicas)`
  2) Silver
     silver_telemetry: (Parquet, particionada por dt)
       ts `(timestamp)`, device_id `(string)`, location `(string)`,metric `(string)`
       value `(double)`, status `(string)`, ingest_ts `(timestamp)`,
       dt `(string o date)`
       PARTITIONED

     silver_alerts: (Parquet, particionada por dt)

alert_ts, device_id, metric, severity, reason, window_start, window_end, dt

3) Gold

gold_kpi_hourly

hour_ts, metric, location, avg_value, max_value, p95_value, alert_count, dt

gold_kpi_daily

dt, metric, location, avg_value, max_value, p95_value, devices_reporting, alerts_total

- **Implementar la ingesta en streaming desde Kafka**:

  Desarrollar en Zeppelin un proceso de Spark Structured Streaming que lea el topic de telemetría, convierta el JSON a columnas tipadas y escriba de forma continua los datos en HDFS (bronze y/o silver) usando checkpointing para tolerancia a fallos.

- **Garantizar calidad mínima del streaming**:

  Aplicar deduplicación, trabajar con event-time (ts) y watermark para soportar eventos fuera de orden, controlar nulos/valores inválidos y asegurarse de que las particiones se generan correctamente y Hive "ve" los datos.

  "La deduplicación de datos es un proceso que elimina copias excesivas de datos y reduce significativamente los requisitos de capacidad de almacenamiento."

- **Generar alertas en tiempo real**:

  Crear un segundo flujo (o una rama del mismo) que, a partir de la telemetría ya tipada, detecte al menos tres situaciones (umbral fuera de rango según dim_metric, picos en ventanas temporales y dispositivos sin actividad durante X minutos) y escriba las alertas resultantes en la tabla silver_alerts (y/o en una ruta HDFS asociada).

- **Explotar la información con Spark SQL**:

  Construir un conjunto **mínimo de 10 consultas analíticas** sobre las tablas Hive (telemetría y alertas) para obtener KPIs por hora/día, por ubicación y por métrica, identificar dispositivos problemáticos y analizar tendencias, y opcionalmente materializar agregados en tablas "gold".

- **Preparar la demo y la entrega**:

  Verificar que, al recibir eventos por Kafka, se cargan datos en Hive, aparecen alertas y las consultas devuelven resultados; organizar los notebooks de Zeppelin, exportar o adjuntar el DDL de Hive y las consultas SQL, y redactar un documento breve explicando arquitectura, rutas HDFS, tablas y decisiones técnicas (ventanas, watermark, deduplicación).

2. Criterios de evaluación

- **COMPETENCIAS TÉCNICAS 80%**

  - Aspectos técnicos de la solución 100%

  Es necesario superar (nota >= 5) para dar por superado el reto.

	0	2.5	5	7.5	10
Diseño de la solución y Data Lake RA1 a, c RA2 a, d	No existe estructura clara en HDFS ni explicación; datos sueltos o inexistentes.	Hay carpeta en HDFS pero sin convención ni particiones; formato no justificado (ej. JSON sin criterio).	Existe estructura **Bronze/Silver** en HDFS **y** partición por dt definida; formato elegido (Parquet) aunque con poca justificación.	Estructura **Bronze + Silver** consistente, partición por dt aplicada y comprobable, formato Parquet razonado, rutas documentadas.	Todo lo anterior **+** estructura **Gold** definida y coherente con KPIs (rutas y particiones), y se refleja en tablas/outputs.
Ingesta desde Kafka con Structured Streaming RA1 b RA2 b	No hay streaming o no lee Kafka.	Lee Kafka pero no parsea bien o no escribe resultados de forma estable.	Lee Kafka + parsea JSON a columnas + escribe a HDFS/Hive en modo streaming (aunque con fallos menores).	Streaming robusto: lectura Kafka, schema tipado, escritura estable y checkpoint configurado y funcionando.	Todo lo anterior **+** pipeline completo alimenta también GOLD (KPIs o tablas gold generadas de forma reproducible).
Hive: tablas y particiones RA1 a, c RA5 a	No hay tablas Hive o no funcionan.	Tablas creadas pero sin particiones o no apuntan bien a HDFS.	Tablas silver_telemetry y silver_alerts creadas y consultables.	Tablas correctas **+** partición por dt funcionando (consultas por dt devuelven resultados y se ven particiones).	Todo lo anterior **+** tablas GOLD (gold_kpi_hourly y/o gold_kpi_daily) creadas y alimentadas.
Integridad y calidad de datos RA3 a RA1 d RA5 b	No hay controles; datos inconsistentes (tipos erróneos, nulos masivos) sin tratar.	Alguna limpieza superficial pero sin estrategia de integridad.	Tipos correctos en silver y limpieza básica (filtrado nulos/invalids).	Incluye **deduplicación** (por event_id o clave) **y** manejo de eventos fuera de orden con **event-time + watermark**.	Todo lo anterior **+** indicadores de calidad (conteo de descartes/errores, métricas de nulos/outliers) y/o tabla de "rejects" (datos que se desecha) y GOLD calculado con datos ya validados.
Alertas en tiempo real RA5 e RA1 d	No hay alertas.	Hay alertas pero manuales o incompletas (solo una regla sin consistencia).	Al menos **2 tipos** de alerta funcionando y persistidas en Hive.	**3 tipos** de alerta funcionando y persistidas: 1) umbral (min_ok/max_ok), 2) spike (ventana), 3) offline/silencio.	Todo lo anterior **+** severidad/razón coherente y explotación en GOLD (KPIs de alertas por hora/día o top dispositivos) materializada.
Spark SQL: consultas analíticas RA5 e RA1 d	No hay consultas o no funcionan.	Menos de **5 consultas** o triviales sin valor.	5–9 consultas funcionales con sentido (filtros, agregaciones básicas).	**≥10 consultas** variadas y útiles (KPIs, top, tendencias, joins con dimensiones).	Todo lo anterior **+** KPIs **materializados en GOLD** (tablas gold) y consultas que usan gold para BI/decisión.
Monitorización y operativa (Ambari/Hue) RA4 a, b, d	No se ha usado monitorización; no hay evidencias.	Capturas sueltas sin interpretación.	Se muestran métricas básicas/estado de servicios relevantes (Kafka, HDFS, Hive, Spark) y se comprueba que están OK.	Se monitoriza durante la demo: estado servicios + revisión de logs/errores del streaming + verificación de latencia/ingesta.	Todo lo anterior **+** se justifican decisiones para rendimiento (particiones, formato, checkpoint) y se demuestra estabilidad con GOLD actualizado.

***** Formatos: `ts = 2026-01-21T08:03:01Z`, `dt = 2026-01-21`. * El cluster tiene la fecha en **UTC** * Ejemplo de kpi: **Media por hora** de cada métrica (temperature, power, co2…) por ubicación

- **COMPETENCIAS TRANSVERSALES 20%**

  - Evaluación profesor - alumno/a 70%
  - Autoevaluación 30%

  Las competencias transversales se evaluarán mediante la herramienta SET de Tknika.

3. Recursos

Plataforma / herramientas del centro:

- Cluster Hadoop con **HDFS**
- **Kafka** (topic de telemetría proporcionado)
- **Zeppelin Notes** (desarrollo y ejecución)
- **Ambari** (estado de servicios, logs, métricas)
- **Hue** (Hive, HDFS browser, ejecución SQL)

Material entregado por el profesorado:

- DIMENSIONES: dim_device, dim_location, dim_metric (CSV/JSON o tablas Hive)
- Esquema del evento JSON y nombre del topic Kafka **(topic=telemetria).**

Web importantes:

- https://spark.apache.org/docs/latest/streaming/structured-streaming-kafka-integration.html
- https://spark.apache.org/docs/latest/sql-data-sources-hive-tables.html

IA permitida

4. Temporización

Fecha	Horas	¿Qué hacer?
26 enero – 29 enero	7	Contexto + Data Lake vs DWH + revisar DIM y formato evento + definir rutas HDFS + diseñar tablas Hive
2 febrero – 5 de febrero	7	Crear tablas Hive + montar streaming Kafka→Bronze/Silver con checkpoint + particiones dt + primeras pruebas y correcciones
9 febrero – 11 febrero	7	Implementar alertas + 10 consultas Spark SQL + Gold KPIs + demo final + entrega de DDL/notebooks/documento
Total horas	21