Planificación Curricular Detallada: Módulos de Big Data (5074 y 5075)

Enfoque Pedagógico: Modelo dual que combina una sólida fundamentación teórica (Módulo 5074) con un Aprendizaje Basado en Proyectos (ABP) aplicado a un caso de uso de alto impacto: la gestión inteligente del agua (Módulo 5075).
Stack Tecnológico Principal: PostgreSQL, MongoDB, Cassandra, Kafka, Apache Airflow, Apache Spark (PySpark), FastAPI, Power BI, Docker & Docker Compose, R & Python (Pandas, Matplotlib, Seaborn).

Módulo 1: 5074 - Sistemas de Big Data (Los Fundamentos)

Este módulo se centra en construir el andamiaje teórico y conceptual ("los ladrillos"). Los contenidos se basarán en la estructura del libro de referencia de la editorial RAMA, asegurando una cobertura exhaustiva de los principios fundamentales.

UD1: INTRODUCCIÓN AL BIG DATA. DEL DATO A LA INFORMACIÓN.

- **Conceptos Clave:** Las 5 Vs del Big Data. Ciclo de vida del dato. Ecosistema Hadoop (HDFS, MapReduce, YARN) y su evolución hacia arquitecturas modernas. Roles: Ingeniero de Datos, Científico de Datos, Analista de Datos.
- **Tecnologías Introducidas (a nivel conceptual):** Se presentará un mapa general de las herramientas del stack tecnológico y dónde encaja cada una.

UD2: ARQUITECTURAS Y PATRONES PARA BIG DATA.

- **Conceptos Clave:** Arquitectura Lambda y Kappa. Data Lake, Data Warehouse y el moderno Data Lakehouse. Patrones de ingesta (push vs. pull). Patrones de procesamiento (batch, micro-batch, streaming).
- **Tecnologías Introducidas (a nivel conceptual):** Kafka (streaming), Spark (batch/streaming), Airflow (orquestación).

UD3: ANÁLISIS EXPLORATORIO Y FUNDAMENTOS ESTADÍSTICOS (NUEVA POSICIÓN).

- **Conceptos Clave:** El proceso de Análisis Exploratorio de Datos (EDA). Estadística descriptiva (medidas de tendencia central, dispersión y posición). **Distribuciones de probabilidad (Normal, Uniforme, Poisson) como base para la generación de datos sintéticos.** Visualización para la exploración. KPIs y Business Intelligence (BI).
- **Tecnologías Introducidas (con prácticas guiadas): R (con Tidyverse)** y **Python (con Pandas, Matplotlib, Seaborn)** para realizar un EDA completo sobre un dataset. Se introducirá **Power BI** para visualizaciones interactivas.

UD4: SISTEMAS DE ALMACENAMIENTO (ANTES UD3).

- **Conceptos Clave:** Teorema CAP. Bases de datos relacionales vs. NoSQL. Familias NoSQL: Documentales, Columnares, Clave-Valor y Grafo. Modelado de datos en sistemas NoSQL.
- **Tecnologías Introducidas (con prácticas básicas): PostgreSQL** (SQL), **MongoDB** (Documental), **Cassandra** (Columnar). Se introducirá **MinIO**, un sistema de

almacenamiento de objetos compatible con S3, fundamental para simular un Data Lake.

UD5: BIG DATA POR LOTES (BATCH) (ANTES UD4).

- **Conceptos Clave:** El paradigma MapReduce en profundidad. Frameworks de procesamiento distribuido. Estructuras de datos inmutables y resilientes (RDDs, DataFrames). Optimización de jobs.
- **Tecnologías Introducidas (con prácticas guiadas): Apache Spark** a través de su API **PySpark**. Se realizarán ejercicios de ETL y análisis sobre ficheros estáticos (CSV, JSON, Parquet).

UD6: BIG DATA EN TIEMPO REAL (STREAMING) (ANTES UD5).

- **Conceptos Clave:** Sistemas de mensajería pub-sub. Semánticas de entrega (at least once, at most once, exactly once). Ventanas de tiempo (tumbling, sliding, session).
- **Tecnologías Introducidas (con prácticas guiadas): Apache Kafka** para la publicación y consumo de flujos de datos. Se puede introducir Spark Streaming o Faust (Python) para el procesamiento.

UD7: ANÁLISIS PREDICTIVO (SE MANTIENE).

- **Conceptos Clave:** Introducción al Machine Learning. El proceso KDD. Tipos de aprendizaje (supervisado, no supervisado). Regresión y clasificación. Evaluación de modelos.
- **Tecnologías Introducidas (a nivel conceptual y con ejemplos):** Se explicará cómo librerías como Scikit-learn se integran en pipelines y cómo Spark MLlib permite entrenar modelos a escala.

Módulo 2: 5075 - Big Data Aplicado (ABP: Gestión Inteligente del Agua)

Este módulo es 100% práctico. Cada UD es un hito dentro de un macroproyecto cohesionado que culmina con la creación de una plataforma completa para la gestión del agua.

UD1: CONFIGURACIÓN DE ENTORNOS PROFESIONALES (El Taller).

- **Objetivo:** Crear un entorno de desarrollo unificado, reproducible y aislado para todos los proyectos.
- **Competencias:** Administración básica de **Linux** (CLI, permisos, scripting). Virtualización. Creación de imágenes con **Docker**. Orquestación de servicios multicontenedor con **Docker Compose**. Gestión de código con **Git**.

UD2: PROYECTO 1 - SIMULADOR DE TELEMETRÍA DE CONTADORES.

- **Objetivo:** Generar un flujo de datos realista que simule la lectura de contadores de agua inteligentes.
- **Arquitectura:**
    1. Se crea una API REST con **FastAPI** que expone un endpoint /reading.
    2. Al ser llamada, la API genera una lectura simulada (ID de contador, timestamp, consumo, presión, temperatura) **basándose en las distribuciones estadísticas aprendidas en el Módulo 1**.
    3. El servicio FastAPI se ejecuta en un contenedor **Docker**.
    - **Tecnologías Aplicadas:** FastAPI, Python, Docker.

UD3: PROYECTO 2 - PIPELINE DE INGESTA Y MONITORIZACIÓN.

- **Objetivo:** Capturar los datos del simulador e inyectarlos en sistemas de almacenamiento persistentes y monitorizar el proceso.
- **Arquitectura:**
  1. Un script de Python (el "inyector") llama periódicamente a la API del Proyecto 1.
  2. Los datos maestros de los contadores (ID, ubicación, fecha de instalación) se almacenan en **PostgreSQL**.
  3. Las lecturas de telemetría (datos time-series) se inyectan tanto en **MongoDB** como en **Cassandra** para comparar su rendimiento.
  4. Todo el pipeline (API, inyector, BBDD) se define y levanta con un único fichero **Docker Compose**. Los logs de los contenedores sirven como sistema de monitorización básico.
  - **Tecnologías Aplicadas:** Python, PostgreSQL, MongoDB, Cassandra, Docker Compose.

UD4: PROYECTO 3 - ORQUESTACIÓN DE TAREAS CON AIRFLOW.

- **Objetivo:** Automatizar y calendarizar el pipeline de ingesta del Proyecto 2.
- **Arquitectura:**
  1. Se despliega **Apache Airflow** (vía Docker Compose).
  2. Se crea un DAG (Grafo Acíclico Dirigido) en Python que define las tareas: "llamar a la API", "guardar en PostgreSQL", "guardar en MongoDB".
  3. Se programará para que se ejecute cada 5 minutos, gestionando dependencias y reintentos.
  - **Tecnologías Aplicadas:** Apache Airflow, Python, Docker Compose.

UD5: PROYECTO 4A - DETECCIÓN DE FUGAS EN BATCH.

- **Objetivo:** Analizar los datos históricos almacenados para identificar patrones anómalos que sugieran fugas de agua.
- **Arquitectura:**
  1. Un job de **PySpark** se conecta a la base de datos NoSQL (MongoDB o Cassandra).
  2. Implementa un algoritmo de detección de anomalías (ej. consumo nocturno por encima de un umbral, desviaciones estándar inusuales).
  3. Los resultados (alertas de posibles fugas) se guardan en una tabla en **PostgreSQL**.
  - **Tecnologías Aplicadas:** PySpark, MongoDB/Cassandra, PostgreSQL.

UD6: PROYECTO 4B - DETECCIÓN DE FUGAS EN TIEMPO REAL.

- **Objetivo:** Modificar el sistema para detectar fugas en segundos, no en horas.
- **Arquitectura:**
  1. Se modifica el simulador (Proyecto 2) para que, en lugar de exponer una API, publique las lecturas en un topic de **Apache Kafka**.
  2. Se crea un consumidor de streaming con **PySpark (Spark Streaming)** que lee del topic de Kafka.
  3. El consumidor aplica las reglas de detección de fugas en micro-lotes y genera alertas en tiempo real.

    - **Tecnologías Aplicadas:** Kafka, PySpark (Spark Streaming), Python.

UD7: PROYECTO 5 - PREDICCIÓN DE DEMANDA.

- **Objetivo:** Utilizar el histórico de datos para entrenar un modelo de Machine Learning que prediga la demanda futura de agua.
- **Arquitectura:**
    1. Un *notebook* de **PySpark** lee todos los datos históricos de consumo.
    2. Se realiza la ingeniería de características (ej. extraer día de la semana, hora, etc.).
    3. Se entrena un modelo de regresión o series temporales (usando Spark MLlib o una librería como Prophet) para predecir el consumo para las próximas 24 horas.
    4. La predicción se guarda en una tabla en **PostgreSQL**.
    - **Tecnologías Aplicadas:** PySpark (MLlib), Jupyter Notebooks.

UD8: PROYECTO 6 - CUADRO DE MANDOS INTEGRAL.

- **Objetivo:** Crear una interfaz de visualización para que los gestores de la red de agua puedan tomar decisiones informadas.
- **Arquitectura:**
    1. Se utiliza **Power BI**.
    2. Se conecta a las distintas fuentes de datos:
        - **PostgreSQL:** Para los datos maestros de contadores, las alertas de fugas y las predicciones de demanda.
        - **MongoDB/Cassandra:** Para visualizar el histórico de consumo de un contador específico.
    3. Se crea un dashboard interactivo que unifica toda la información generada en los proyectos anteriores.
    - **Tecnologías Aplicadas:** Power BI.