COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI ICADE CIHS

# Sprint 2. Historia de Usuario 3

**Elena Conderana Medem y Sergio Cuenca Núñez**

Tecnologías de Datos Masivos
*Big Data Technology*

# Índice

# Introducción

## 1. Contexto

El objetivo final del proyecto busca diseñar e implementar una arquitectura Big Data completa, que permita procesar y analizar datos procedentes de `TradingView` para poder asesorar a clientes en el trading de criptomonedas. Este segundo sprint consiste en almacenar los datos obtenidos en el sprint 1 en el Clúster de ICAI utilizando HDFS y en crear tablas externas en Hive, que permitan realizar consultas sobre los datos y analizarlos.

## 2. Descripción del problema

El asesoramiento a clientes en el trading de criptomonedas requiere estudiar su comportamiento histórico. Al trabajar con una gran magnitud de datos, es necesario facilitar y agilizar las consultas sobre ellos y su posterior análisis. Por ello el almacenamiento de los archivos en el Clúster ICAI mediante una jerarquía de carpetas en HDFS y la creación de tablas externas en Hive cuyos datos se encuentran en dichas carpetas habilitan el procesamiento en *batch* de los mismos.

## 3. Objetivos

Los objetivos principales de esta práctica comprenden:

- Creación de jerarquía de carpetas en HDFS en el Clúster ICAI.
- Carga de archivos CSV a su carpeta correspondiente.
- Crear tablas externas en Hive para acceder a los archivos almacenados en las carpetas.

## 4. Justificación

Para poder proporcionar un asesoramiento fundamentado es necesario analizar el comportamiento histórico de las criptomonedas, que permitan extraer conclusiones razonadas. Dada la abundancia de datos es necesario utilizar herramientas como HDFS y Hive que permiten organizar la información y realizar consultas interactivas de manera eficiente sobre conjuntos de datos de gran calibre, a diferencia de las bases de datos relacionales tradicionales.

# Metodología

En la siguiente sección se procede a describir en detalle cómo se ha creado la jerarquía de carpetas en HDFS, se han cargado los archivos pertinentes y se han creado las tablas correspondientes en Hive. Se abordarán cuestiones técnicas sobre el proceso de desarrollo y sobre las tecnologías utilizadas.

## 5. Descripción del entorno de desarrollo

El almacenamiento de los datos se realiza en el Clúster ICAI, para tener una ubicación centralizada que contenga todos los datos y las herramientas necesarias para su explotación. La interacción con el clúster se realiza a través de la página de Hue (`https://gethue.com/`), un asistente SQL de código abierto para bases de datos y *data warehouse,* que proporciona una interfaz gráfica.

El almacenamiento de los archivos en el Clúster ICAI se va a realizar en una jerarquía de carpetas en el `Hadoop Data File System`, más conocido como HDFS. El sistema de almacenamiento distribuido de información más extendido en Big Data. Así mismo, la creación de tablas externas que proporciona acceso al contenido de los archivos mediante consultas de SQL se realiza desde el editor Hive. Esta herramienta de *Data Warehousing* y ETL funciona sobre HDFS y facilita el análisis tipo SQL de conjuntos de datos de grandes magnitudes y el encapsulamiento de datos.

## 6. Diseño de la solución

Tras cargar los ficheros CSV adquiridos en el sprint 1 en forma de fichero ZIP en el directorio `/datos/gittba/gittba04` dentro del Clúster ICAI se procede a la creación de las tablas desde el editor Hive. Como primer paso se define la estructura de tabla con el comando `CREATE EXTERNAL TABLE IF NOT EXISTS`, donde se especifica el nombre de la tabla, el nombre de las columnas y el tipo de dato que deben contener. Este comando está complementado por las cláusulas `PARTITIONED BY`, que divide la tabla en subconjuntos por año, `ROW FORMAT DELIMITED FIELDS TERMINATED BY`, que indica el separador usado en los archivos csv, `STORED AS`, que explicita el tipo de archivo y `LOCATION`, que concreta la ubicación de los archivos pertinentes dentro del clúster.

Tras definir la estructura base de la table se utiliza el comando `ALTER TABLE` acompañado de la cláusula `SET TBLPROPERTIES` para que se descarte la cabecera del fichero. El comando se repite otras 5 veces junto con las cláusulas `ADD PARTITION` para especificar el año del subconjunto y `LOCATION` para seleccionar los datos pertenecientes a ese año en concreto.

Puesto que todas las tablas presentan la misma estructura, se van a reiterar los comandos anteriores, excepto `CREATE EXTERNAL TABLE` sin alteración alguna para cada criptomoneda. La creación de las tablas se va a simplificar mediante la cláusula `LIKE` que permite clonar la estructura y propiedades definidas para la primera tabla. El Código 1 ejemplifica cada uno de los comandos explicados.

```
CREATE EXTERNAL TABLE IF NOT EXISTS gittba04.btcusdt (
    date TIMESTAMP,
```

```
    open DOUBLE,
    high DOUBLE,
    low DOUBLE,
    close DOUBLE,
    volume DOUBLE
)
PARTITIONED BY (year INT)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE
LOCATION 'hdfs:/datos/gittba/gittba04/BTCUSDT';

ALTER TABLE gittba04.btcusdt SET TBLPROPERTIES
("skip.header.line.count"="1");

ALTER TABLE gittba04.btcusdt ADD PARTITION (year = 2021) LOCATION
'hdfs:/datos/gittba/gittba04/BTCUSDT/2021';
ALTER TABLE gittba04.btcusdt ADD PARTITION (year = 2022) LOCATION
'hdfs:/datos/gittba/gittba04/BTCUSDT/2022';
ALTER TABLE gittba04.btcusdt ADD PARTITION (year = 2023) LOCATION
'hdfs:/datos/gittba/gittba04/BTCUSDT/2023';
ALTER TABLE gittba04.btcusdt ADD PARTITION (year = 2024) LOCATION
'hdfs:/datos/gittba/gittba04/BTCUSDT/2024';
ALTER TABLE gittba04.btcusdt ADD PARTITION (year = 2025) LOCATION
'hdfs:/datos/gittba/gittba04/BTCUSDT/2025';
```

Código 1. Funcionamiento de la Solución

# 7. Pruebas realizadas

El correcto funcionamiento se ha comprobado mediante la ejecución de diversas consultas sobre las tablas, que constatan que los archivos se han cargado correctamente y que las tablas son capaces de acceder a los datos.

# Resultados

A continuación, se procede a estudiar los resultados obtenidos durante la ejecución de la práctica.

## 8. Descripción de los resultados

Con la metodología anterior se ha cargado la jerarquía de carpetas descrita en el sprint anterior a HDFS utilizando un archivo ZIP. Además, se ha creado una tabla externa por cada criptomoneda para acceder a sus datos. Cada tabla se ha particionado por sus años constituyentes (2025, 2024, 2023, 2022 y 2021). Esta partición permite mejorar el rendimiento y acelerar los tiempos de consulta, pues únicamente se consultan las tablas implicadas en la consulta y no el conjunto completo de datos. El editor de Hive permite realizar consultas en SQL a las tablas y responde con el extracto de datos que se adecúa a las premisas de la consulta.

## 9. Pantallazos de la Ejecución

En este apartado se muestran dos consultas realizadas desde Hive que ejemplifican el funcionamiento del sistema. En la Figura 1 se muestran las particiones de la tabla `btcusdt`, que se corresponden con los años esperados. La Figura 2 muestra las 10 primeras entradas de los valores de la criptomoneda `dogeusdt` en 2023.

| | partition |
|---|---|
| 1 | year=2021 |
| 2 | year=2022 |
| 3 | year=2023 |
| 4 | year=2024 |
| 5 | year=2025 |

Figura 1. Resultado de la consulta `SHOW PARTITIONS gittba04.btcusdt;`

| | dogeusdt.date | dogeusdt.open | dogeusdt.high | dogeusdt.low |
|---|---|---|---|---|
| 1 | 2023-01-01 01:00:00.0 | 0.07024 | 0.07087 | 0.06906 |
| 2 | 2023-01-02 01:00:00.0 | 0.07023 | 0.07374 | 0.06906 |
| 3 | 2023-01-03 01:00:00.0 | 0.07138 | 0.07231 | 0.06943 |
| 4 | 2023-01-04 01:00:00.0 | 0.07049 | 0.07362 | 0.07019 |
| 5 | 2023-01-05 01:00:00.0 | 0.0731 | 0.07529 | 0.07035 |
| 6 | 2023-01-06 01:00:00.0 | 0.07151 | 0.07281 | 0.0695 |
| 7 | 2023-01-07 01:00:00.0 | 0.07242 | 0.07305 | 0.07167 |
| 8 | 2023-01-08 01:00:00.0 | 0.07219 | 0.07399 | 0.07121 |
| 9 | 2023-01-09 01:00:00.0 | 0.07362 | 0.07963 | 0.07293 |
| 10 | 2023-01-10 01:00:00.0 | 0.07565 | 0.0779 | 0.07372 |

Figura 2. Resultado de la consulta `SELECT * FROM gittba04.dogeusdt WHERE year = 2023 LIMIT 10;`

## 10. Discusión de los resultados

Como se ha demostrado en el apartado anterior la creación de tablas y almacenamiento de los archivos se ha realizado correctamente en la jerarquía especificada en el primer sprint. De esta manera se ha conseguido exitosamente crear un entorno de acceso de los datos históricos de las criptomonedas en Hadoop. Además, la

partición en años de los datos de cada tabla permita el acceso mediante esta variable, facilitando y eficientando el acceso a los datos.

# Conclusión

Por último, se van a resumir los principales hallazgos y aprendizajes obtenidos durante la práctica, y la relevancia de la solución implementada.

## 11. Resumen del Proceso

Para el almacenamiento de los archivos en HDFS y creación de tablas externas en Hive para proporcionar acceso a los datos se han seguido los siguientes pasos.

1. Identificación del problema: Almacenamiento y acceso a datos en arquitectura Hadoop.
2. Diseño de la arquitectura e implementación: Carga de jerarquía de carpetas y archivos CSV a Clúster ICAI en HDFS mediante archivo ZIP a través de la interfaz web `HUE`. Definición de la estructura de tablas externas y creación de estas en Hive para acceder al histórico de datos almacenado en las carpetas.
3. Pruebas y análisis: Ejecución de distintas consultas SQL desde el editor Hive para comprobar el correcto almacenamiento y acceso al mismo.

## 12. Principales Logros

La solución implementada ha mostrado su eficacia para crear un sistema de tablas siguiendo una estructura específica y para acceder a los datos correspondientes almacenados en una jerarquía de carpetas en HDFS. Para concluir se van a recorrer los logros más importantes alcanzados en base a los resultados que se han analizado en la sección anterior.

- Familiarización con la interfaz Hue.
- Familiarización con `Hadoop`.
- Almacenamiento de datos en Clúster ICAI en jerarquía de carpetas en HDFS.
- Creación de tablas externas en Hive.
- Acceso mediante consultas SQL mediante editor Hive a datos históricos de las criptomonedas.