



**COMILLAS**

**UNIVERSIDAD PONTIFICIA**

ICAI

ICADE

CIHS

# Sprint 4. Historia de Usuario 5

**Elena Conderana Medem y Sergio Cuenca Núñez**

Tecnologías de Datos Masivos  
*Big Data Technology*

# Índice

<b>ÍNDICE.....</b>	<b>2</b>
<b>INTRODUCCIÓN .....</b>	<b>3</b>
<b>1. CONTEXTO .....</b>	<b>3</b>
<b>2. DESCRIPCIÓN DEL PROBLEMA.....</b>	<b>3</b>
<b>3. OBJETIVOS .....</b>	<b>3</b>
<b>4. JUSTIFICACIÓN.....</b>	<b>3</b>
<b>METODOLOGÍA.....</b>	<b>4</b>
<b>5. DESCRIPCIÓN DEL ENTORNO DE DESARROLLO.....</b>	<b>4</b>
<b>6. DISEÑO DE LA SOLUCIÓN .....</b>	<b>4</b>
<b>7. PRUEBAS REALIZADAS .....</b>	<b>6</b>
<b>RESULTADOS.....</b>	<b>7</b>
<b>8. DESCRIPCIÓN DE LOS RESULTADOS .....</b>	<b>7</b>
<b>9. PANTALLAZOS DE LA EJECUCIÓN .....</b>	<b>7</b>
<b>10. DISCUSIÓN DE LOS RESULTADOS.....</b>	<b>7</b>
<b>CONCLUSIÓN .....</b>	<b>9</b>
<b>11. RESUMEN DEL PROCESO .....</b>	<b>9</b>
<b>12. PRINCIPALES LOGROS .....</b>	<b>9</b>

# 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 sprint consiste en la visualización de la cotización de cada criptomoneda en 2024 junto con los indicadores SMA200 y EMA50 haciendo uso de los datos almacenados en las capas Plata y Oro.

## 2. Descripción del problema

El propósito del análisis técnico de criptomonedas consiste en anticipar con mayor probabilidad cambios y tendencias en las mismas. La abundancia de métricas y de precios de cierre a lo largo de un período complican la identificación sencilla de tendencias. Por este motivo la visualización de los valores de cotización de cada criptomoneda con velas japonesas acompañadas de los indicadores SMA200 y EMA50 facilitan este proceso y resumen en un gráfico toda la información relevante.

## 3. Objetivos

Los objetivos principales de esta práctica comprenden:

- Visualización de los datos almacenados en las capas Plata y Oro.
- Visualización con `pyspark` y con `matplotlib` de la cotización de cada criptomoneda en 2024 mediante velas japonesas en gráficos independientes.
- Visualización en el gráfico de cada criptomoneda de sus indicadores SMA200 y EMA50.

## 4. Justificación

Para analizar el rendimiento histórico de las criptomonedas resulta mucho más intuitivo y sencillo visualizar la cotización a lo largo del tiempo que estudiar los valores en bruto. Además, acompañar las cotizaciones con los valores de media móvil simple y de media móvil exponencial ayudan a identificar tendencias actuales y cambios en las mismas. De esta manera estas visualizaciones aportan información de valor que permite realizar un asesoramiento informado y fundamentado.

# Metodología

En la siguiente sección se procede a describir en detalle cómo se han realizado las visualizaciones de las cotizaciones e indicadores SMA200 y EMA50 para cada criptomoneda accediendo a los datos almacenados en las capas Plata y Oro respectivamente. Se abordarán cuestiones técnicas sobre el proceso de desarrollo y sobre las tecnologías utilizadas.

## 5. Descripción del entorno de desarrollo

La plataforma empleada para el desarrollo de la solución es *Jupyter Hub*, un servidor de entornos de programación online, que permite trabajar dentro del ClusterICAI. Para establecer la sesión de Spark y poder acceder e interactuar con los datos almacenados en las capas Oro y Plata se utiliza la biblioteca ``pyspark``, la interfaz para Apache Spark en Python.

Para realizar la visualización de los datos es necesario transformar los mismos a un *dataframe* de pandas y confirmar que su tipología es numérica. Para ello se hace uso de ``pandas``, una biblioteca de software para la manipulación y análisis de datos en Python, que ofrece estructuras de datos y operaciones para la manipulación de tablas. Por último, las visualizaciones se realizan con la biblioteca ``plotly``, que permite generar gráficos interactivos y personalizados en Python.

## 6. Diseño de la solución

Para visualizar las cotizaciones e indicadores se ha implementado un bucle for que ejecute las operaciones necesarias para cada criptomoneda y genera su propio gráfico. En concreto el proceso consiste en la carga de los datos de la tabla correspondiente a 2024 de las capas Plata y Oro. De la capa Plata se seleccionan todos los parámetros, mientras que de la Oro únicamente se seleccionan los valores correspondientes a la fecha y a las métricas SMA200 y EMA50. Como último paso antes de proceder a las visualizaciones se realiza un *join* entre ambas tablas para tener un único conjunto de datos unificado.

Para las visualizaciones se hace uso del comando *Candlestick* de ``plotly`` que representa las cotizaciones con velas japonesas. Para los indicadores se utiliza el comando *Scatter* que representa las métricas con una línea. Para incluir todos los gráficos en una única figura se emplea el comando *add\_trace*. En el Código 1 se incluye la solución descrita.

```
# Import necessary libraries
import pandas as pd
import plotly.graph_objects as go

# List of cryptocurrencies
cryptos = [
    "BTCUSD", "ETHUSD", "XRPUSD", "SOLUSD",
    "DOGEUSD", "ADAUSD", "SHIBUSD", "DOTUSD",
    "AAVEUSD", "XLMUSD"
]

# Year to visualize
visualization_year = 2024
```

```
# Iterate over each cryptocurrency
for symbol in cryptos:
    # Load Silver Layer
    silver_path = f"/datos/gittba/gittba04/{symbol}_Silver/year={visualization_year}"
    df_silver = spark.read.parquet(silver_path).select("date", "open", "high", "low", "close",
"volume")

    # Load Gold Layer
    gold_path = f"/datos/gittba/gittba04/{symbol}_Gold/year={visualization_year}"
    df_gold = spark.read.parquet(gold_path).select("date", "SMA200", "EMA50")

    # Merge both dataframes on 'date'
    df_merged = df_silver.join(df_gold, "date", "inner")

    # Convert to Pandas
    df_pd = df_merged.orderBy("date").toPandas()
    df_pd["date"] = pd.to_datetime(df_pd["date"])
    df_pd.set_index("date", inplace = True)

    # Ensure all values are numeric
    df_pd[["open", "high", "low", "close", "SMA200", "EMA50"]] = df_pd[
        ["open", "high", "low", "close", "SMA200", "EMA50"]
    ].apply(pd.to_numeric, errors="coerce")

    # Create figure
    fig = go.Figure()

    # Add Candlestick trace
    fig.add_trace(go.Candlestick(
        x=df_pd.index, open=df_pd["open"], high=df_pd["high"],
        low=df_pd["low"], close=df_pd["close"],
        name = "Candlesticks"))

    # Add SMA200 trace
    fig.add_trace(go.Scatter(
        x=df_pd.index, y=df_pd["SMA200"], mode="lines",
        name = "SMA 200", line=dict(color = "blue", width=1.5)))

    # Add EMA50 trace
    fig.add_trace(go.Scatter(
        x=df_pd.index, y=df_pd["EMA50"], mode="lines",
        name = "EMA 50", line=dict(color = "orange", width=1.5)))

    # Layout adjustments
    fig.update_layout(
        title=f"{symbol} Price & Indicators - {visualization_year}",
        xaxis_title = "Date", yaxis_title = "Price (USD)",
        xaxis_rangeslider_visible = False,
        hovermode = "x unified"
    )

    # Show interactive plot
    fig.show()
```

Código 1. Cálculo de métricas y almacenamiento en tabla en capa Oro.

## 7. Pruebas realizadas

La consistencia de los datos de las capas Plata y Oro se corroboró en el sprint anterior, aunque se identificó alguna ligera divergencia en los valores de los indicadores. Para evaluar la bondad de los gráficos finales de cada criptomoneda se va a comparar los gráficos obtenidos con los gráficos ofrecidos por `TradingView` en el mismo intervalo de tiempo. No se busca una coincidencia total, pero sí un parecido muy fuerte, donde las tendencias y comportamiento de las cotizaciones e indicadores sea análogo.

# 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 han transformado los datos de cotización almacenados en la capa Plata y los valores de los indicadores en la capa Oro en información sencillamente legible y comprensible. La representación conjunta de los datos en un único gráfico permite visualizar de manera rápida y sencilla el comportamiento presentado por una criptomoneda en el último año e identificar tendencias que se han producido y predecir potenciales comportamientos a futuro.

Para cada criptomoneda se ha compuesto una gráfica con los valores de cotización representados mediante velas japonesas, y con los indicadores SMA200 y EMA50 representados mediante una línea de distinto color para facilitar su identificación.

## 9. Pantallazos de la Ejecución

La visualización resultante para cada criptomoneda se compone de la cotización de esta en 2024 y de sus indicadores de SMA200 y EMA50. La Figura 1 muestra el gráfico para la criptomoneda BTCUSDT. No se incluyen el resto de los gráficos, pues son análogos aunque diverjan en la naturaleza de sus cotizaciones.

BTCUSDT Price & Indicators - 2024



Figura 1. Representación de la cotización e indicadores SMA200 y EMA50 para BTCUSDT

## 10. Discusión de los resultados

La correcta implementación de los sprint anteriores y del actual se corrobora mediante el contraste entre la visualización obtenida para cada criptomoneda y las gráficas para el mismo intervalo temporal en `TradingView`. La Figura 2 muestra la cotización y los indicadores de BTCUSDT. Visualmente se comprueba que las tendencias

son iguales y que los valores recogidos en el eje de ordenadas se corresponden. Asimismo, los indicadores intersecan en los mismos instantes temporales y muestran el mismo comportamiento, confirmando la correcta implementación de la solución. Solo se muestra la discusión de BTCUSDT, puesto que el proceso es idéntico para todas las criptomonedas.

Como curiosidad cabe destacar que se producen dos cruces entre los indicadores. El primer cruce se conoce como “Cruce Muerte”, que se produce cuando el EMA50 corta desde arriba hacia abajo al SMA200. Recibe esta denominación, porque este cruce descendente suele marcar el fin de una tendencia alcista. El segundo cruce se denomina “Cruce Oro” y se produce cuando el cruce es inverso, indicando el inicio de una tendencia alcista, que en este caso se puede confirmar que se produce.



Figura 2. Representación de la cotización e indicadores SMA200 y EMA50 para BTCUSDT en `TradingView`



# 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 la representación de las cotizaciones de cada criptomoneda y de sus indicadores SMA200 y EMA50 durante el período de 2024 se han seguido los siguientes pasos.

1. Identificación del problema: Visualizar cotizaciones y métricas SMA200 y EMA50 durante 2024 para cada criptomoneda.
2. Diseño de la arquitectura e implementación: Carga de los datos de cotización almacenados en la capa Plata y de los valores de SMA200 y EMA50 en la capa Oro. Unificación de los datos en una única tabla y transformación a pandas confirmando la naturaleza numérica de los valores y el correcto formato de la columna *date*. Visualización conjunta de las cotizaciones mediante velas japonesas y de los indicadores con líneas de tendencia para cada criptomoneda.
3. Pruebas y análisis: Comparación de los gráficos obtenidos para cada criptomoneda con los gráficos para el período equivalente en la herramienta ``TradingView``.

## 12. Principales Logros

La solución implementada ha mostrado su eficacia para acceder a los datos procesados en *sprints* anteriores y almacenados en las capas Plata y Oro. Además, de haber conseguido de manera exitosa visualizar dichos datos para que coincidan con los gráficos que se visualizan en herramientas como ``TradingView``. 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 ``pyplot``.
- Carga de datos en capas Plata y Oro.
- Interacción con datos procesados y almacenados en ClusterICAI en distintas capas.
- Visualización mediante velas japonesas de valores de cotización.
- Inclusión de indicadores en gráfico de cotización.
- Similitud con gráficos extraídos de ``TradingView``