



PRESENTACIÓN E ÍNDICE DE LA DOCUMENTACIÓN

Este documento constituye el informe central de nuestro análisis. Hemos detallado el proceso desde la recogida de datos de una fuente externa, su estructuración en una base de datos relacional, la comprensión de sus características, el planteamiento de preguntas de negocio, hasta la creación de consultas y vistas en SQL para responderlas.

ÍNDICE

- 1.- Recogida de datos
- 2.- Creación de BBDD
- 3.- Comprensión de los Datos Recibidos
- 3.- Planteamiento de Preguntas Clave
- 4.- Realizar consultas SQL básicas
- 5.- Generar vistas útiles
- 6.- Interpretación de los Resultados y Acciones de Negocio

1. Recogida de Datos

Para dar respuesta a las necesidades de análisis de la compañía, hemos iniciado el proyecto con la fase de adquisición y estructuración de los datos.













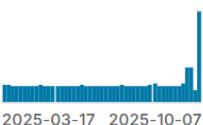
Nuestro primer paso ha sido identificar una fuente de datos fiable y representativa del mercado financiero. Hemos seleccionado un conjunto de datos públicos de Kaggle que contiene información horaria tanto del mercado de acciones como de las principales criptomonedas.

LINK:

<https://www.kaggle.com/datasets/adrianjuliusaluo/hourly-crypto-stocks-market-data/data>

El proceso que hemos seguido es el siguiente:

1. **Descarga de Ficheros:** Hemos obtenido dos ficheros en formato CSV
 - stocks.csv: Contiene datos de las principales acciones del mercado estadounidense.
 - criptomonedas.csv: Contiene datos de los activos digitales más relevantes.

 timestamp 	 name 		 symbol 		 price_usd 		 vol_24h 		 total_vol 	
Record Date & Time	Crypto Name		Crypto Ticker Symbol		Price in US Dollars		24 Hour Trade Volume		Total Market Volume	
	USDC	5%	USDC	5%	\$1.00	1%	54262 unique values	0.00%	5%	
	XRP	5%	SOL	5%	1	1%		0.20%	2%	
	Other (97761)	91%	Other (98116)	91%	Other (105171)	97%		Other (100797)	93%	
2025-10-07 09:34:54	Wrapped stETH		wsteth		\$5,684.59		\$13,223,985.00		0.00%	
2025-10-07 09:34:54	Figure Heloc		figr_heloc		\$1.00		\$11,414,970.00		0.00%	
2025-10-07 09:34:54	LEO Token		leo		\$9.63		\$730,414.00		0.00%	
2025-10-07 09:34:54	Wrapped Beacon ETH		wbeth		\$5,048.70		\$22,375,730.00		0.01%	
2025-10-07 09:34:54	Wrapped eETH		weeth		\$5,045.60		\$39,536,467.00		0.01%	
2025-10-07 09:34:54	USDS		usds		\$1.00		\$33,460,447.00		0.01%	

2. **Creación de la Base de Datos:** Para gestionar y consultar esta información de manera eficiente, hemos creado una base de datos relacional que hemos denominado RETO_0. La hemos creado desde la interfaz gráfica de SQL Server.

RETO 0 | SISTEMAS DE BIG DATA

3. Dentro de esta base de datos, hemos diseñado e implementado dos tablas principales, cada una correspondiendo a uno de los ficheros:

Tabla stocks: Almacena toda la información relativa a las acciones.

timestamp	name	last	high	low	chg_	chg_%	vol_	time
2025-10-02 13:00:01	3M	156	156.54	154.26	+0.82	+0.53%	2.3M	01/10
2025-10-02 13:00:01	Amazon.com	220.63	222.15	216.61	+1.06	+0.48%	43.93M	01/10
2025-10-02 13:00:01	American Express	328.56	335.77	328.4	-3.6	-1.08%	2.6M	01/10
2025-10-02 13:00:01	Amgen	298.5	301.66	283	+16.30	+5.78%	4.84M	01/10
2025-10-02 13:00:01	Apple	255.45	258.79	254.93	+0.82	+0.32%	48.71M	01/10
2025-10-02 13:00:01	Boeing	215.2	215.63	213.7	-0.63	-0.29%	7.18M	01/10
2025-10-02 13:00:01	Caterpillar	480.82	483.93	471.14	+3.67	+0.77%	2.96M	01/10
2025-10-02 13:00:01	Chevron	154.58	155.09	153.19	-0.71	-0.46%	6.77M	01/10
2025-10-02 13:00:01	Cisco	68.75	68.82	67.69	+0.33	+0.48%	15.53M	01/10

Tabla cryptomonedas: Almacena los datos correspondientes a los activos digitales.

timestamp	name	symbol	price_usd	vol_24h	total_vol	chg_24h	chg_7d	market_cap
2025-09-26 14:35:12	Dogecoin	DOGE	0.223351	\$4.20B	1.85%	-4.13%	-18.45%	\$33.56B
2025-09-26 14:35:12	BNB	BNB	936.1	\$4.72B	1.92%	-4.94%	-5.58%	\$129.98B
2025-09-26 14:35:12	USDC	USDC	0.9993	\$25.08B	10.84%	-0.02%	-0.02%	\$73.85B
2025-09-26 14:35:12	Ethereum	ETH	3,891.64	\$61.10B	27.96%	-2.99%	-14.16%	\$468.60B
2025-09-26 14:35:12	XRP	XRP	2.7079	\$9.25B	3.76%	-3.94%	-10.76%	\$161.66B
2025-09-26 14:35:12	Bitcoin	BTC	108,777.00	\$73.89B	31.11%	-2.52%	-6.65%	\$2.17T
2025-09-26 14:35:12	Solana	SOL	191.858	\$11.39B	5.06%	-5.11%	-20.89%	\$103.89B
2025-09-26 14:35:12	Tether USDt	USDT	1.0008	\$190.46B	82.48%	+0.02%	-0.01%	\$173.53B
2025-09-26 14:05:13	UNUS SED LEO	LEOu	9.4994	\$910.78K	0.00%	+0.24%	-0.23%	\$8.78B

Este enfoque nos permite mantener los datos organizados, seguros listos para ser consultados y analizados, sentando una base sólida para las siguientes fases del proyecto. Si se requiere la automatización podemos desarrollar scripts en Python para que la recogida y carga de datos se realice de forma periódica sin intervención manual.

2. Creación de la BBDD

Para crear la BBDD hemos lanzado este script:

```
# Importa las herramientas necesarias de las librerías.
from sqlalchemy import create_engine, text
import urllib

# --- Define los nombres que se usarán ---
SERVER_NAME = r'IBAI\SQLEXPRESS'    # Nombre de tu servidor SQL.
DB_TO_CREATE = 'RETO0'              # Nombre de la base de datos a crear.
SYSTEM_DB = 'master'                # Base de datos del sistema para la conexión inicial.

# --- Intenta conectarse al servidor SQL ---
engine = None                        # Prepara la variable para la conexión.
try:
    # Crea la cadena de conexión con todos los datos necesarios.
    connection_string = (
        f"DRIVER={{ODBC Driver 17 for SQL Server}};"
        f"SERVER={SERVER_NAME};"
        f"DATABASE={SYSTEM_DB};"
        f"Trusted_Connection=yes;"
    )
    # Codifica la cadena para que los caracteres especiales no den error.
    quoted_conn_str = urllib.parse.quote_plus(connection_string)

    # Crea el "motor" que gestionará la conexión a la base de datos.
    engine = create_engine(f"mssql+pyodbc:///?" + "odbc_connect=" + quoted_conn_str)

    print("✅ Conexión exitosa...") # Informa que la conexión funcionó.
except Exception as e:
    # Si la conexión falla, muestra un mensaje de error.
    print(f"❌ ERROR: No se pudo conectar.")
    print(f"    Detalle del error: {e}")
```

```
# --- Si la conexión tuvo éxito, crea la base de datos ---
if engine is not None:
    try:
        # Define el comando SQL. "IF NOT EXISTS" evita errores si ya existe.
        sql_create_db_command = f"""
        IF NOT EXISTS (SELECT name FROM sys.databases WHERE name = N'{DB_TO_CREATE}')
        BEGIN
            CREATE DATABASE [{DB_TO_CREATE}];
        END;
        """

        # Se conecta y ejecuta el comando SQL.
        with engine.connect() as connection:
            connection.execution_options(isolation_level="AUTOCOMMIT").execute(text(sql_create_db_command))

        print(f"✅ ¡Éxito! La base de datos '{DB_TO_CREATE}' ha sido creada...")

    except Exception as e:
        # Si la creación falla, muestra un mensaje de error.
        print(f"❌ ERROR durante la creación de la base de datos: {e}")
```

RETO 0 | SISTEMAS DE BIG DATA

Con esto podemos ver cómo se ha creado la BBDD de datos en mysql server sin necesidad de meternos a crearla desde la interfaz gráfica. Sin embargo es una base de datos vacía sin tablas, por lo que para ello lanzamos otro script que crea las tablas. Ante todo priorizamos la eficiencia por lo que en todos los scripts tienen corrección de errores.

```
# Importa las herramientas necesarias. 'pandas' para leer CSVs y 'sqlalchemy' para la BBDD.
import pandas as pd
from sqlalchemy import create_engine
import urllib

# --- Define los nombres para la conexión ---
SERVER_NAME = r'IBAI\SQLEXPRESS' # Nombre de tu servidor SQL.
DATABASE_NAME = 'RETO0' # Nombre de la base de datos a la que te conectarás.

# --- Intenta conectarse y crear las tablas ---
try:
    # Construye la cadena de conexión, especificando ahora la BBDD 'RETO0'.
    connection_string = f"DRIVER={{ODBC Driver 17 for SQL Server}};SERVER={SERVER_NAME};DATABASE={DATABASE_NAME};Trusted_Connection=yes;"

    # Codifica la cadena para evitar errores con caracteres especiales.
    quoted_conn_str = urllib.parse.quote_plus(connection_string)

    # Crea el "motor" de conexión a la base de datos 'RETO0'.
    engine = create_engine(f"mssql+pyodbc:///odbc_connect={quoted_conn_str}")

    print(f"✅ Conexión exitosa con la BBDD '{DATABASE_NAME}'...")

    # --- Define una función para crear tablas vacías ---
    def crear_tabla_vacia_desde_csv(ruta_csv, nombre_tabla):
        print(f"\n📄 Creando estructura para la tabla '{nombre_tabla}'...")

        # Lee solo las cabeceras del CSV (0 filas) para obtener los nombres y tipos de columnas.
        df_esquema = pd.read_csv(ruta_csv).head(0)

        # Usa el esquema del DataFrame vacío para crear la tabla en SQL.
        # 'if_exists='replace'' asegura que si la tabla ya existe, se borra y se crea de nuevo.
        df_esquema.to_sql(nombre_tabla, engine, if_exists='replace', index=False)

        print(f"✅ ¡Éxito! La tabla '{nombre_tabla}' ha sido creada vacía.")
```

```
# --- Ejecuta la función para cada tabla que se quiera crear ---
print("\n🚀 INICIANDO EL SCRIPT DE CREACIÓN DE ESTRUCTURAS...")

# Llama a la función para crear la tabla 'stocks' a partir de 'stocks.csv'.
crear_tabla_vacia_desde_csv('stocks.csv', 'stocks')

# Llama a la función para crear la tabla 'cryptomonedas' a partir de 'cryptocurrency.csv'.
crear_tabla_vacia_desde_csv('cryptocurrency.csv', 'cryptomonedas')

print("\n🏁 Proceso de creación de estructuras finalizado.")

except Exception as e:
    # Si algo falla (conexión, lectura de CSV, etc.), muestra el error.
    print(f"❌ ERROR: {e}")
```

RETO 0 | SISTEMAS DE BIG DATA

Una vez creada las tablas, solo queda llenarlas de datos, para ello volveremos a usar un script. Como siempre priorizando la efectividad por lo que vacía la tabla para evitar el duplicado de datos

```
# Importa las herramientas necesarias.
import pandas as pd
from sqlalchemy import create_engine, text
import urllib

# --- Define los nombres para la conexión ---
SERVER_NAME = r'IBAI\SQLEXPRESS' # Nombre de tu servidor SQL.
DATABASE_NAME = 'RETO0' # Nombre de la base de datos a usar.

# --- Intenta conectarse e insertar los datos ---
try:
    # Construye la cadena de conexión a la base de datos 'RETO0'.
    connection_string = f"DRIVER={{ODBC Driver 17 for SQL Server}};SERVER={SERVER_NAME};DATABASE={DATABASE_NAME};Trusted_Connection=yes;"

    # Codifica la cadena para evitar errores con caracteres especiales.
    quoted_conn_str = urllib.parse.quote_plus(connection_string)

    # Crea el "motor" de conexión a la base de datos.
    engine = create_engine(f"mssql+pyodbc:///odbc_connect={quoted_conn_str}")

    print(f"✅ Conexión exitosa con la BBDD '{DATABASE_NAME}'...")

    # --- Define una función para vaciar una tabla y luego llenarla con datos de un CSV ---
    def insertar_datos_en_tabla_existente(ruta_csv, nombre_tabla):
        print(f"📄 Insertando datos en la tabla '{nombre_tabla}'...")
```

```
    # --- PASO 1: VACIAR LA TABLA ---
    # Se conecta al motor de la base de datos.
    with engine.connect() as connection:
        print(f"    -> Vacinando tabla '{nombre_tabla}'...")
        # Ejecuta el comando SQL 'TRUNCATE TABLE', que borra todas las filas de forma muy eficiente.
        # 'AUTOCOMMIT' asegura que la operación de borrado se ejecute inmediatamente.
        connection.execution_options(isolation_level="AUTOCOMMIT").execute(text(f"TRUNCATE TABLE {nombre_tabla}"))

    # --- PASO 2: INSERTAR LOS NUEVOS DATOS ---
    # Lee todos los datos del archivo CSV y los carga en un DataFrame de pandas.
    df_datos = pd.read_csv(ruta_csv)

    # Inserta los datos del DataFrame en la tabla SQL.
    # 'if_exists='append'' significa que los datos se añaden a la tabla (que ahora está vacía).
    df_datos.to_sql(nombre_tabla, engine, if_exists='append', index=False)

    # Imprime un mensaje de éxito con el número de filas que se insertaron.
    print(f"✅ ¡Éxito! {len(df_datos):,} filas insertadas en '{nombre_tabla}'.".replace(',', ' '))

# --- Ejecuta el proceso de inserción para cada archivo ---
print("\n🚀 INICIANDO EL SCRIPT DE INSERCIÓN DE DATOS...")

# Llama a la función para vaciar la tabla 'stocks' y llenarla con 'stocks.csv'.
insertar_datos_en_tabla_existente('stocks.csv', 'stocks')

# Llama a la función para vaciar 'cryptomonedas' y llenarla con 'cryptocurrency.csv'.
insertar_datos_en_tabla_existente('cryptocurrency.csv', 'cryptomonedas')

print("\n🏁 Proceso de inserción finalizado.")

except Exception as e:
    # Si algo falla en cualquier punto, captura el error y lo muestra.
    print(f"❌ ERROR: {e}")
```

3. Comprensión de los Datos Recibidos

Una vez almacenados los datos, nuestro equipo ha realizado un análisis exploratorio para familiarizarse con la estructura y el contenido de la información.

A. Conjunto de Datos de la Tabla stocks:

Este fichero contiene el estado de diversas acciones en un momento concreto. Las columnas clave que hemos identificado son:

- **name (Texto):** El nombre de la compañía (ej. "Amazon.com", "Apple").
- **last (Numérico):** El último precio de cotización registrado.
- **high/low (Numérico):** El precio más alto y más bajo alcanzado en el período.
- **chg_ (Numérico):** El cambio neto en el precio (en dólares). Es un indicador directo de la ganancia o pérdida.
- **chg_% (Texto):** El cambio porcentual en el precio. Muy útil para comparar el rendimiento entre acciones de diferente valor.
- **vol (Texto):** El volumen de negociación, representado con abreviaturas (ej. '2.3M' para 2.3 millones). Indica el nivel de interés o actividad en una acción.

IBAI\SQLEXPR... - dbo.stocks		IBAI\SQLEXPR...cryptomonedas	
	Nombre de columna	Tipo de datos	Permitir valores NULL
▶	timestamp	datetime2(7)	<input checked="" type="checkbox"/>
	name	nvarchar(50)	<input checked="" type="checkbox"/>
	last	float	<input checked="" type="checkbox"/>
	high	float	<input checked="" type="checkbox"/>
	low	float	<input checked="" type="checkbox"/>
	chg	float	<input checked="" type="checkbox"/>
	chg1	nvarchar(50)	<input checked="" type="checkbox"/>
	vol	nvarchar(50)	<input checked="" type="checkbox"/>
	time	time(7)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

B. Conjunto de Datos de la Tablas cryptomonedas

Similar al de acciones, este conjunto nos ofrece una visión del mercado de activos digitales. Las áreas de mayor interés son:

- **name/symbol (Texto):** El nombre completo y el "ticker" o símbolo de la criptomoneda (ej. "Bitcoin", "BTC").
- **price_usd (Numérico):** El precio en dólares estadounidenses.
- **vol_24h/total_vol (Texto):** El volumen negociado en las últimas 24 horas y el volumen total.
- **chg_24h/chg_7d (Texto):** El cambio porcentual en las últimas 24 horas y en los últimos 7 días. Clave para entender tendencias a corto y medio plazo.
- **market_cap (Texto):** La capitalización de mercado, que representa el valor total de todas las monedas en circulación. Es un indicador del tamaño y la relevancia del activo.

IBAI\SQLEXP...ryptomonedas			
	Nombre de columna	Tipo de datos	Permitir valores NULL
▶	timestamp	datetime2(7)	<input checked="" type="checkbox"/>
	name	varchar(50)	<input checked="" type="checkbox"/>
	symbol	varchar(50)	<input checked="" type="checkbox"/>
	price_usd	float	<input checked="" type="checkbox"/>
	vol_24h	varchar(50)	<input checked="" type="checkbox"/>
	total_vol	varchar(50)	<input checked="" type="checkbox"/>
	chg_24h	varchar(50)	<input checked="" type="checkbox"/>
	chg_7d	varchar(50)	<input checked="" type="checkbox"/>
	market_cap	varchar(50)	<input checked="" type="checkbox"/>

Identificación de Áreas Clave para el Cliente:

Ambos conjuntos de datos son extremadamente valiosos para la toma de decisiones. Hemos identificado que las áreas más críticas para el negocio son:

- **Rendimiento y Volatilidad:** Analizar los cambios de precio (chg_, chg_%) para identificar oportunidades y riesgos.
- **Liquidez e Interés del Mercado:** El volumen de negociación (vol, vol_24h) es un excelente indicador de la liquidez de un activo.
- **Valoración y Dominancia:** La capitalización de mercado (market_cap) nos permite entender qué activos dominan el ecosistema.

4. Planteamiento de Preguntas Clave

Basándonos en nuestra comprensión de los datos y los objetivos del negocio, hemos formulado una serie de preguntas estratégicas. Estas preguntas guiarán nuestro análisis y nos ayudarán a extraer información que pueda traducirse en acciones concretas para mejorar la rentabilidad y la toma de decisiones.

Preguntas para el Mercado de Acciones:

1. ¿Cuáles son las acciones con mejor y peor rendimiento reciente para identificar rápidamente las "estrellas" y los "problemas" de nuestra cartera?
2. ¿Qué acciones se encuentran dentro de un rango de precio específico que pueda ser atractivo para nuevas inversiones?
3. ¿Cómo podemos clasificar las acciones según su volatilidad para gestionar mejor el riesgo de nuestras posiciones?
4. ¿Cuál es el sentimiento general del mercado? ¿Hay más acciones al alza o a la baja en un día determinado?
5. ¿Qué acciones están superando el rendimiento de un referente del mercado como Microsoft, para detectar líderes sectoriales?

Preguntas para el Mercado de Criptomonedas:

1. ¿Qué criptomonedas tienen mayor volumen de negociación? Esto nos indica dónde se concentra el interés de los inversores.
2. ¿Existen activos que hayan sufrido caídas drásticas recientemente, señalando un posible riesgo o una oportunidad de compra a bajo precio?
3. ¿Las "stablecoins" (monedas estables) principales mantienen su paridad con el dólar, asegurando la estabilidad de nuestros fondos anclados a ellas?
4. ¿Qué porcentaje del mercado total dominan los gigantes como Bitcoin y Ethereum? Esto nos ayuda a entender la concentración del mercado.
5. ¿Qué criptomonedas son actualmente más caras que Ethereum, para comprender el posicionamiento de precios en la gama alta del mercado?

A continuación, procederemos a responder estas preguntas mediante consultas técnicas y a crear herramientas visuales (vistas) que faciliten el acceso a esta información.

5. Realizar consultas SQL básicas

Para nuestra empresa, las consultas SQL son la herramienta estratégica que convierte nuestro activo más valioso que son los datos en respuestas de negocio claras y precisas.

En lugar de navegar por hojas de cálculo, SQL nos permite:
Interrogar los datos para obtener respuestas instantáneas a preguntas complejas (ej. "¿cuáles son las acciones con mayor rendimiento?").

Por eso partiendo de las preguntas que nos han surgido dentro de la empresa, vamos a contestarlas utilizando las propias Consultas SQL. Para ello hemos dividido el apartado de consultas en las referentes con las cuestiones de la tablas de stocks, y las que pertenecen a las tablas de cryptomonedas

Mercado de Acciones (Stocks)

- **Pregunta 1:** ¿Cuáles son las acciones con mejor y peor rendimiento reciente para identificar rápidamente las "estrellas" y los "problemas" de nuestra cartera?

```
SELECT TOP 5 name, chg  
FROM stocks  
ORDER BY chg DESC;
```

	name	chg
1	Goldman Sachs	5830
2	Goldman Sachs	5465
3	Goldman Sachs	5465
4	Goldman Sachs	5465
5	Goldman Sachs	5465

Explicación de la Relación: Esta consulta ataca directamente la pregunta utilizando `ORDER BY chg DESC` para ordenar las acciones de mayor a menor ganancia en dólares, y `SELECT TOP 5` para mostrarnos únicamente las más destacadas, es decir, las "estrellas". (Para ver los "problemas", simplemente cambiaríamos `DESC` por `ASC`).

- **Pregunta 2:** ¿Qué acciones se encuentran dentro de un rango de precio específico que pueda ser atractivo para nuevas inversiones?

```
SELECT name, last, vol
FROM stocks
WHERE last BETWEEN 50 AND 100
ORDER BY vol DESC;
```

Resultados		Mensajes	
	name	last	vol
1	Walmart	95	9.77M
2	Nike	61	9.54M
3	Cisco	69	9.48M
4	Walmart	96	843.59K
5	Coca-Cola	72	8.72M
6	Coca-Cola	72	8.72M
7	Coca-Cola	72	8.72M
8	Coca-Cola	72	8.72M
9	Coca-Cola	72	8.72M
10	Coca-Cola	72	8.72M
11	Cisco	63	8.69M
12	Citigroup	95	8.65M
13	Citigroup	95	8.65M
14	Citigroup	95	8.65M
15	Citigroup	95	8.65M
16	Citigroup	95	8.65M
17	Citigroup	95	8.65M
18	Citigroup	95	8.65M
19	Citigroup	95	8.65M
20	Citigroup	95	8.65M

Explicación de la Relación: La cláusula WHERE last BETWEEN 50 AND 100 es el filtro exacto que responde a la necesidad de buscar en un "rango de precio específico". Además, ordenamos por volumen (ORDER BY vol DESC) para priorizar aquellas que son más líquidas, haciéndolas más atractivas para la inversión.

RETO 0 | SISTEMAS DE BIG DATA

- **Pregunta 3:** ¿Cómo podemos clasificar las acciones según su volatilidad para gestionar mejor el riesgo de nuestras posiciones?

```
SELECT
  name,
  chg,
  CASE
    WHEN chg > 3.0 OR chg < -3.0 THEN 'Alta Volatilidad'
    WHEN chg BETWEEN -1.0 AND 1.0 THEN 'Baja Volatilidad'
    ELSE 'Volatilidad Moderada'
  END AS CategoriaVolatilidad
FROM stocks
ORDER BY CategoriaVolatilidad, chg DESC;
```

	name	chg	CategoriaVolatilidad
1	Goldman Sachs	5830	Alta Volatilidad
2	Goldman Sachs	5465	Alta Volatilidad
3	Goldman Sachs	5465	Alta Volatilidad
4	Goldman Sachs	5465	Alta Volatilidad
5	Goldman Sachs	5465	Alta Volatilidad
6	Goldman Sachs	5465	Alta Volatilidad
7	Goldman Sachs	5465	Alta Volatilidad
8	Goldman Sachs	5465	Alta Volatilidad
9	Goldman Sachs	5465	Alta Volatilidad
10	Goldman Sachs	5465	Alta Volatilidad
11	Goldman Sachs	5465	Alta Volatilidad
12	Goldman Sachs	5465	Alta Volatilidad
13	Goldman Sachs	5465	Alta Volatilidad
14	Goldman Sachs	5461	Alta Volatilidad
15	Goldman Sachs	5461	Alta Volatilidad
16	Goldman Sachs	5461	Alta Volatilidad
17	Goldman Sachs	5461	Alta Volatilidad
18	Goldman Sachs	5461	Alta Volatilidad
19	Goldman Sachs	4718	Alta Volatilidad
20	UnitedHealth	3914	Alta Volatilidad

Explicación de la Relación: La estructura CASE ... END es la herramienta que hemos usado para implementar la "clasificación por volatilidad" que la pregunta demanda. Traduce un dato numérico (chg) en categorías de negocio fáciles de entender ('Alta', 'Baja', 'Moderada'), permitiendo una gestión de riesgo más intuitiva.

- **Pregunta 4:** ¿Cuál es el sentimiento general del mercado? ¿Hay más acciones al alza o a la baja en un día determinado?

```
SELECT
CASE
  WHEN chg > 0 THEN 'Acciones Ganadoras'
  WHEN chg < 0 THEN 'Acciones Perdedoras'
  ELSE 'Sin Cambios'
END AS Estado,
COUNT(*) AS CantidadDeAcciones
FROM stocks
GROUP BY
CASE WHEN chg > 0 THEN 'Acciones Ganadoras' WHEN chg < 0 THEN
'Acciones Perdedoras' ELSE 'Sin Cambios' END;
```

	Estado	CantidadDeAcciones
1	Acciones Perdedoras	73831
2	Acciones Ganadoras	82168
3	Sin Cambios	879

Explicación de la Relación: Esta consulta agrega miles de puntos de datos individuales en tres únicos grupos (GROUP BY). El resultado responde directamente a la pregunta mostrando cuántas acciones pertenecen a cada categoría, ofreciendo una fotografía instantánea del "sentimiento general del mercado".

- **Pregunta 5:** ¿Qué acciones están superando el rendimiento de un referente del mercado como Microsoft?

```
SELECT s1.name, s1.chg
FROM stocks AS s1
JOIN stocks AS benchmark ON benchmark.name = 'Microsoft'
WHERE s1.chg > benchmark.chg AND s1.name <> 'Microsoft'
ORDER BY s1.chg DESC;
```

Resultados  Mensajes

name	chg
Goldman Sachs	5830
Goldman Sachs	5830
Goldman Sachs	5830
Goldman Sachs	5830
Goldman Sachs	5830
Goldman Sachs	5830
Goldman Sachs	5830
Goldman Sachs	5830
Goldman Sachs	5830
Goldman Sachs	5830
Goldman Sachs	5830
Goldman Sachs	5830
Goldman Sachs	5830
Goldman Sachs	5830
Goldman Sachs	5830
Goldman Sachs	5830
Goldman Sachs	5830
Goldman Sachs	5830
Goldman Sachs	5830
Goldman Sachs	5830
Goldman Sachs	5830
Goldman Sachs	5830


Explicación de la Relación: El JOIN de la tabla consigo misma nos permite poner el rendimiento de Microsoft al lado del de cada una de las otras acciones. La cláusula WHERE `s1.chg > benchmark.chg` realiza entonces la comparación directa que la pregunta exige: "muéstrame solo aquellas cuyo rendimiento es mayor que el del referente".

Mercado de Criptomonedas (Cryptos)

- **Pregunta 1:** ¿Qué criptomonedas tienen mayor volumen de negociación?

```
SELECT TOP 10 name, symbol, vol_24h
FROM cryptomonedas
ORDER BY vol_24h DESC;
```

Resultados



Mensajes

name	symbol	vol_24h
USDT	1.00039995	79.34%
BTC	92,887.30	44.41%
SOL	147.8699951	3.73%
XRP	2.2033999	3.29%
ETH	1,751.16	18.85%
USDC	0.9993	11.60%
BNB	615.710022	1.44%
DOGE	0.179359	1.25%
SUI	2.69429994	1.20%
ADA	0.6789	0.69%

Explicación de la Relación: Al igual que con las acciones, ORDER BY vol_24h DESC ordena los activos por su actividad de mercado, y TOP 10 nos presenta a los líderes indiscutibles, respondiendo directamente a la pregunta sobre quiénes tienen el "mayor volumen".

- **Pregunta 2:** ¿Existen activos que hayan sufrido caídas drásticas recientemente?

```
SELECT name, symbol, chg_7d
FROM cryptomonedas
WHERE TRY_CAST(REPLACE(chg_7d, '%', '') AS FLOAT) < -20.0;
```

Resultados		Mensajes	
	name	symbol	chg_7d
1	Pi	PI	-37.71%
2	Pi	PI	-35.91%
3	Pi	PI	-35.27%
4	Pi	PI	-34.37%
5	Pi	PI	-33.59%
6	Pi	PI	-33.15%
7	Pi	PI	-32.92%
8	Pi	PI	-32.57%
9	Pi	PI	-32.50%
10	Pi	PI	-32.45%
11	Pi	PI	-32.42%
12	Pi	PI	-32.42%
13	Pi	PI	-32.41%
14	Pi	PI	-32.33%
15	Pi	PI	-32.18%
16	Pi	PI	-32.11%
17	Pi	PI	-32.11%
18	Pi	PI	-32.10%

Explicación de la Relación: La pregunta busca "caídas drásticas". Hemos definido "drástica" como una caída superior al 20% y lo hemos implementado en la cláusula WHERE. La parte de TRY_CAST y REPLACE es el paso técnico necesario para poder realizar esa comparación numérica y filtrar los activos que cumplen esta condición de riesgo.

RETO 0 | SISTEMAS DE BIG DATA

- **Pregunta 3:** ¿Las "stablecoins" principales mantienen su paridad con el dólar?

```
SELECT name, symbol, price_usd
FROM cryptomonedas
WHERE symbol IN ('USDT', 'USDC', 'DAI');
```



	name	symbol	price_usd
1	USDC	USDC	9993
2	Tether USDt	USDT	10008
3	USDC	USDC	9993
4	Tether USDt	USDT	10008
5	USDC	USDC	9992
6	Tether USDt	USDT	10009
7	USDC	USDC	9991
8	Tether USDt	USDT	1001
9	USDC	USDC	9991
10	Tether USDt	USDT	1001
11	USDC	USDC	9991
12	Tether USDt	USDT	1001
13	Tether USDt	USDT	1001
14	USDC	USDC	9991
15	Tether USDt	USDT	1001
16	USDC	USDC	9991
17	Tether USDt	USDT	1001

Explicación de la Relación: Para responder esta pregunta de verificación, la consulta aísla (WHERE symbol IN (...)) exclusivamente las stablecoins de interés y muestra su precio. Esto permite una auditoría visual inmediata para confirmar si su valor es cercano a 1 dólar.

RETO 0 | SISTEMAS DE BIG DATA

- **Pregunta 4:** ¿Qué porcentaje del mercado total dominan los gigantes como Bitcoin y Ethereum?

```
SELECT
(
  (SELECT SUM(TRY_CAST(REPLACE(REPLACE(market_cap, '$', ''), 'B', '')) AS
  FLOAT) * 10000000000)
  FROM cryptomonedas WHERE symbol IN ('BTC', 'ETH'))
/
(SELECT SUM(TRY_CAST(REPLACE(REPLACE(market_cap, '$', ''), 'B', '')) AS
  FLOAT) * 10000000000)
  FROM cryptomonedas)
) * 100 AS [Dominancia_BTC_ETH (%)]
```

 Resultados  Mensajes	
	Dominancia_BTC_ETH (%)
1	32,7767870641191

Explicación de la Relación: La arquitectura de la consulta, con dos subconsultas (una para la suma de los líderes y otra para la suma total) y una división final, está diseñada específicamente para calcular el "porcentaje del mercado total" que la pregunta exige.

RETO 0 | SISTEMAS DE BIG DATA

- **Pregunta 5:** ¿Qué criptomonedas son actualmente más caras que Ethereum, para comprender el posicionamiento de precios en la gama alta del mercado?

```
SELECT
  c1.name AS 'Criptomoneda',
  c1.price_usd AS 'Precio_Cripto',
  benchmark_eth.price_usd AS 'Precio_Ethereum'
FROM
  cryptomonedas AS c1
JOIN
  cryptomonedas AS benchmark_eth ON benchmark_eth.symbol = 'ETH'
WHERE
  TRY_CAST(REPLACE(c1.price_usd, ',', '')) AS FLOAT >
  TRY_CAST(REPLACE(benchmark_eth.price_usd, ',', '')) AS FLOAT
ORDER BY
  TRY_CAST(REPLACE(c1.price_usd, ',', '')) AS FLOAT DESC;
```

	Criptomoneda	Precio_Cripto	Precio_Ethereum
1	Bitcoin	12429510	141891
2	Bitcoin	12429510	142032
3	Bitcoin	12429510	143251
4	Bitcoin	12429510	144559
5	Bitcoin	12429510	144652
6	Bitcoin	12429510	145098
7	Bitcoin	12429510	145592
8	Bitcoin	12429510	145805
9	Bitcoin	12429510	145811
10	Bitcoin	12429510	145854
11	Bitcoin	12429510	146345
12	Bitcoin	12429510	146606
13	Bitcoin	12429510	146762
14	Bitcoin	12429510	146898
15	Bitcoin	12429510	146928
16	Bitcoin	12429510	147372
17	Bitcoin	12429510	147412
18	Bitcoin	12429510	147462
19	Bitcoin	12429510	147532
20	Bitcoin	12429510	147706

La estructura de la consulta, mediante un JOIN de la tabla consigo misma (self-join), está diseñada para alinear el precio de cada criptomoneda con el de Ethereum en la misma fila. La posterior cláusula WHERE ejecuta la comparación directa, filtrando únicamente los activos que son "más caros que Ethereum", tal como la pregunta requiere.

6. Generar Vistas Útiles

Para facilitar el acceso recurrente a información clave y simplificar las consultas para los distintos departamentos, hemos creado las siguientes vistas. Las vistas actúan como "tablas virtuales" que pre-procesan y resumen los datos, permitiendo a los usuarios obtener respuestas de negocio sin necesidad de escribir código complejo ni lidiar con la limpieza de datos en cada consulta.

1. Vista: *V_Panel_General_Stocks*

Propósito: Crear un panel de control diario para el mercado de acciones, con datos limpios y listos para ser analizados.

```
CREATE VIEW V_Panel_General_Stocks AS
SELECT
  name AS Nombre,
  last AS UltimoPrecio,
  chg_ AS CambioDolares,
  TRY_CAST(REPLACE(vol, 'M', '') AS FLOAT) * 1000000 AS Volumen_Numerico,
  CASE
    WHEN chg_ > 0 THEN 'Ganadora'
    WHEN chg_ < 0 THEN 'Perdedora'
    ELSE 'Sin Cambios'
  END AS Estado,
  vol AS Volumen_Original
FROM
  stocks;
```

RETO 0 | SISTEMAS DE BIG DATA

Ejemplo de Uso:

Para obtener una lista de las acciones ganadoras del día, ordenadas por su cambio en dólares, el analista solo necesita ejecutar:

```
SELECT * FROM V_Panel_Genera_Stocks WHERE Estado = 'Ganadora' ORDER BY CambioDolares DESC;
```

 Resultados  Mensajes

	Nombre	UltimoPrecio	CambioDolares	Volumen_Numerico	Estado	Volumen_Original
1	Goldman Sachs	52052	5830	5830000	Ganadora	5.83M
2	Goldman Sachs	51687	5465	7020000	Ganadora	7.02M
3	Goldman Sachs	51687	5465	7020000	Ganadora	7.02M
4	Goldman Sachs	51687	5465	7020000	Ganadora	7.02M
5	Goldman Sachs	51687	5465	7020000	Ganadora	7.02M
6	Goldman Sachs	51687	5465	7020000	Ganadora	7.02M
7	Goldman Sachs	51687	5465	7020000	Ganadora	7.02M
8	Goldman Sachs	51687	5465	7020000	Ganadora	7.02M
9	Goldman Sachs	51687	5465	7020000	Ganadora	7.02M
10	Goldman Sachs	51687	5465	7020000	Ganadora	7.02M
11	Goldman Sachs	51687	5465	7020000	Ganadora	7.02M
12	Goldman Sachs	51687	5465	7020000	Ganadora	7.02M
13	Goldman Sachs	51687	5465	7020000	Ganadora	7.02M
14	Goldman Sachs	51683	5461	6980000	Ganadora	6.98M
15	Goldman Sachs	51683	5461	6980000	Ganadora	6.98M
16	Goldman Sachs	51683	5461	6980000	Ganadora	6.98M
17	Goldman Sachs	51683	5461	6980000	Ganadora	6.98M
18	Goldman Sachs	51683	5461	6980000	Ganadora	6.98M

2. Vista: V_Analisis_Riesgo_Crypto

Propósito: Estandarizar la evaluación de riesgo de los criptoactivos, traduciendo la volatilidad en una clasificación de negocio clara.

```

CREATE VIEW V_Analisis_Riesgo_Crypto AS
SELECT
    name AS Nombre,
    symbol AS Simbolo,
    TRY_CAST(REPLACE(price_usd, ',', '')) AS FLOAT) AS Precio_Numerico,
    TRY_CAST(REPLACE(REPLACE(market_cap, '$', ''), 'B', '')) AS FLOAT) * 1000000000 AS
MarketCap_Numerico,
    TRY_CAST(REPLACE(chg_24h, '%', '')) AS FLOAT) AS Cambio24h_Numerico,
    TRY_CAST(REPLACE(chg_7d, '%', '')) AS FLOAT) AS Cambio7d_Numerico,
    CASE
        WHEN TRY_CAST(REPLACE(chg_7d, '%', '')) AS FLOAT) < -20.0 THEN 'Alta Volatilidad
Negativa'
        WHEN TRY_CAST(REPLACE(chg_7d, '%', '')) AS FLOAT) > 20.0 THEN 'Alta Volatilidad
Positiva'
        WHEN TRY_CAST(REPLACE(chg_7d, '%', '')) AS FLOAT) BETWEEN -5.0 AND 5.0 THEN
'Tendencia Estable'
        ELSE 'Volatilidad Moderada'
    END AS NivelRiesgoSemanal,
    price_usd AS Precio_Original,
    market_cap AS MarketCap_Original,
    chg_7d AS Cambio7d_Original
FROM
    cryptomonedas;

```

RETO 0 | SISTEMAS DE BIG DATA

Ejemplo de Uso:

Para identificar y revisar todos los activos que han sufrido caídas drásticas y presentan un alto riesgo, el equipo solo necesita ejecutar:

```
SELECT * FROM V_Analisis_Riesgo_Crypto WHERE NivelRiesgoSemanal = 'Alta Volatilidad Negativa' ORDER BY Cambio7d_Numerico ASC;
```

Resultados		Mensajes								
	Nombre	Simbolo	Precio_Numerico	MarketCap_Numerico	Cambio24h_Numerico	Cambio7d_Numerico	NivelRiesgoSemanal	Precio_Original	MarketCap_Original	Cambio7d_Original
1	Pi	PI	10377	7100000000	-9,81	-37,71	Alta Volatilidad Negativa	10377	\$7.10B	-37.71%
2	Pi	PI	10417	7200000000	-9,78	-35,91	Alta Volatilidad Negativa	10417	\$7.20B	-35.91%
3	Pi	PI	10828	7390000000	-7,8	-35,27	Alta Volatilidad Negativa	10828	\$7.39B	-35.27%
4	Pi	PI	10529	7410000000	-8,57	-34,37	Alta Volatilidad Negativa	10529	\$7.41B	-34.37%
5	Pi	PI	10909	7600000000	-4,27	-33,59	Alta Volatilidad Negativa	10909	\$7.60B	-33.59%
6	Pi	PI	11046	7560000000	-4,78	-33,15	Alta Volatilidad Negativa	11046	\$7.56B	-33.15%
7	Pi	PI	11576	7910000000	-0,73	-32,92	Alta Volatilidad Negativa	11576	\$7.91B	-32.92%
8	Pi	PI	11657	7950000000	-2,55	-32,57	Alta Volatilidad Negativa	11657	\$7.95B	-32.57%
9	Pi	PI	11626	7960000000	-3,73	-32,5	Alta Volatilidad Negativa	11626	\$7.96B	-32.50%
10	Pi	PI	11082	7590000000	-4,54	-32,45	Alta Volatilidad Negativa	11082	\$7.59B	-32.45%
11	Pi	PI	11575	7880000000	0,39	-32,42	Alta Volatilidad Negativa	11575	\$7.88B	-32.42%
12	Pi	PI	11347	7800000000	-2,31	-32,42	Alta Volatilidad Negativa	11347	\$7.80B	-32.42%
13	Pi	PI	11125	7540000000	-3,78	-32,41	Alta Volatilidad Negativa	11125	\$7.54B	-32.41%
14	Pi	PI	11721	8150000000	3,01	-32,33	Alta Volatilidad Negativa	11721	\$8.15B	-32.33%
15	Pi	PI	11755	7920000000	0,4	-32,18	Alta Volatilidad Negativa	11755	\$7.92B	-32.18%

7. Interpretación de los Resultados y Acciones de Negocio

Los resultados obtenidos a través de estas consultas y vistas no son solo números; son el reflejo de la dinámica del mercado y una guía para la toma de decisiones informadas.

Resultados Comprensibles:

Hemos transformado datos brutos en métricas claras. Por ejemplo, en lugar de un simple $\text{chg} = -25.8\%$, nuestra vista `V_Analisis_Riesgo_Crypto` lo categoriza como 'Alta Volatilidad Negativa'.

Esta etiqueta es inmediatamente comprensible para cualquier miembro de la empresa, desde un analista junior hasta el CEO.

Del mismo modo, el "Sentimiento del Mercado" resume miles de datos en una sola idea: "Hoy, 70% de las acciones fueron ganadoras".

Resultados Accionables:

Cada consulta está diseñada para impulsar una acción:

- Al ver el **Top 5 de acciones con mayor subida**, el equipo de análisis puede investigar por qué están subiendo (noticias, resultados financieros) y **proponer una inversión**.
- La **lista de criptomonedas** con fuerte caída semanal es una **alerta directa** para que el **comité de riesgos evalúe** si se deben vender posiciones para mitigar pérdidas.
- La **vista de riesgo** permite a la cúpula directiva **definir políticas de inversión**, como "no invertir más del 5% del capital en activos de alta volatilidad".

Nuestro trabajo convierte los datos en inteligencia de negocio, permitiendo a la empresa pasar de ser reactiva a ser proactiva en su estrategia de mercado.

8. Conclusión y Problemas que hemos encontrado

Conclusión

El trabajo realizado en este proyecto inicial ha sentado las bases de un **sistema de inteligencia de negocio robusto y escalable**. Hemos transformado con éxito dos conjuntos de datos complejos en una **fuentes de información estructurada, limpia y accionable**.

A través de la formulación de preguntas de negocio clave y su resolución **mediante consultas SQL**, hemos demostrado que es posible extraer un valor inmenso de los datos de mercado.

Las **vistas** no son solo **herramientas técnicas**; son **soluciones de negocio** que permitirán al Departamento de Análisis **operar con mayor agilidad y precisión**.

Estamos convencidos de que la implementación de nuestras recomendaciones, multiplicará el retorno de esta inversión inicial.

Principales Problemas Afrontados

Durante la fase de análisis y desarrollo, nos enfrentamos a varios desafíos, principalmente relacionados con la **el formato de los datos originales**. Estos son algunos ejemplos:

1. **Inconsistencia en los Tipos de Datos:** El reto más significativo fue que campos numéricos vitales para el análisis financiero **Que impedía** realizar cualquier operación matemática directa.
2. **Necesidad de Transformación de Datos (ETL):** No podíamos simplemente cargar los datos y usarlos. Fue imprescindible diseñar y aplicar un proceso de limpieza y transformación. Mediante el uso de funciones SQL como REPLACE y TRY_CAST.
3. **Gestión de Datos Faltantes (Nulos):** Anticipamos que algunos registros podrían no contener toda la información. Para evitar errores en los cálculos agregados **integramos algunas cláusulas** en las consultas.

Afrontar y resolver estos problemas de calidad de datos desde el principio no solo ha garantizado la precisión de este primer análisis, sino que también ha fortalecido la base de datos para futuros proyectos, haciendo que el sistema sea más resiliente y fiable a largo plazo.