Fecha: 28 de febrero del 2026
Nombres y apellidos:
Juan Carlos Hernández de la Cruz
Marilyn Melissa Cerdas Brenes
Karen Calvo Vidaurre
Alejandro Abarca Méndez
Alejandra Torres García
Asignatura: Sistemas de Gestión de Bases de Datos
Máster: Big Data & Business Inteligente

SISTEMAS DE GESTIÓN DE BASES DE DATOS (SGBD)

Modelo de inversión multi-activo basado en aprendizaje automático para creación de portafolios diversificados de mediano riesgo a largo plazo

I. Introducción

El desarrollo del trabajo está sustentado en la construcción de un dataset robusto y consistente que permita evaluar el desempeño de una estrategia cuantitativa de inversión. Para ello, se diseña una base de datos histórica con información diaria de aproximadamente 200 activos durante los últimos 10 años, integrando acciones del S&P 500 y del FTSE 100, así como una selección de ETF´s de bonos y commodities.

Los datos son extraídos principalmente de Yahoo Finance(yfinance) y Pandas DataReader, fuentes utilizadas en investigación financiera por su disponibilidad, granularidad y trazabilidad.

El dataset central consolida variables esenciales para el análisis financiero y predictivo, incluyendo precios ajustados, volúmenes de negociación, dividendos, activos bajo gestión (AUM) y métricas macroeconómicas como la expectativa de inflación de EE. UU. (T10YIE) y el índice de precios al consumidor del Reino Unido (UKCPI).

Cada variable fue seleccionada por su relevancia en la modelización del comportamiento de los activos, ya sea como indicador de liquidez, riesgo, dinámica sectorial o sensibilidad macroeconómica.

La pertinencia de este dataset reside en tres elementos clave:

1. **Representatividad del mercado:** al incluir una cartera diversificada de activos provenientes de distintos sectores y geografías.

2. **Profundidad temporal:** con 10 años de datos diarios que permiten capturar ciclos económicos, choques de mercado y diferentes regímenes financieros.

3. **Calidad y consistencia:** garantizadas mediante procesos de limpieza, imputación (forward fill), control de liquidez mínima, filtros sectoriales y sincronización de frecuencias entre datos diarios y mensuales.

Este conjunto de datos constituye la base sobre la cual se evalúa la capacidad predictiva y la estabilidad del enfoque cuantitativo propuesto para nuestro TFM.

Fecha: 28 de febrero del 2026
Nombres y apellidos:
Juan Carlos Hernández de la Cruz
Marilyn Melissa Cerdas Brenes
Karen Calvo Vidaurre
Alejandro Abarca Méndez
Alejandra Torres García
Asignatura: Sistemas de Gestión de Bases de Datos
Máster: Big Data & Business Inteligente

1. Caracterización de cada dataset.

1.1 Objetivo del dataset en el sistema. Consolidar en una única estructura todos los datos financieros, de mercado y macroeconómicos necesarios para entrenar y evaluar un modelo de inversión. Reúne información histórica diaria de precios, volúmenes, sectores, dividendos, activos bajo gestión y variables macroeconómicas, con el fin de construir un universo estable de 200 activos que permita analizar el rendimiento de la estrategia cuantitativa.

1.2 Origen del dato. Los datos provienen de dos fuentes principales:

- **YFinance:** Mediante el API de Yahoo Finance (yfinance) se utilizará la información diaria sobre precios ajustados, volumen, dividendos, sector y AUM para acciones y ETFs.
- **FRED:** Por medio del API de FRED (Federal Reserve Economic Data) se utilizarán variables macroeconómicas, como la expectativa de inflación diaria de EE. UU. (T10YIE) y el índice de precios al consumidor del Reino Unido (UK_CPI).

Estas fuentes fueron seleccionadas por su disponibilidad histórica, actualización constante y confiabilidad para estudios cuantitativos.

1.3 Estructura del dataset. El dataset tiene estructura tabular; cada fila representa un activo en una fecha específica (frecuencia diaria) y tiene las siguientes columnas:

- Fecha
- Ticker
- Sector
- Adj_Close
- Volume
- Div_Yield
- AUM
- T10YIE
- UK_CPI

1.4 Volumen esperado del dataset. En su creación inicial, el dataset contiene 1,539,876 filas y 9 columnas, derivado de 622 activos extraídos del S&P500, FTSE100 y diversos ETFs para la construcción inicial del portafolio previo a que se haga modelado de datos y filtraciones para escoger nuestro universo final de 200 activos.

Este volumen se espera que incremente con cada ciclo de rebalanceo mensual que agrega nuevos datos a la base.

1.5 Frecuencia de actualización. El dataset se actualiza mediante procesos automáticos de ingesta en Python. La actualización inicial es de 10 años completos de datos diarios. Posteriormente, la actualización operativa es cada 30 días, que se agregan los datos diarios de los días hábiles de trading más recientes.

1.6 Requisitos de consistencia y calidad del dato. El dataset aplica mecanismos de control de calidad, según el siguiente detalle:

- Forward fill: Para cubrir días feriados y evitar huecos temporales entre mercados de EE. UU. y Reino Unido.
- Alineación de frecuencias: El UK_CPI, que es mensual, se replica durante el mes correspondiente mediante la aplicación de “lag” para evitar que haya un hueco en la columna de dicha métrica para el mes más reciente.
- Filtrado por liquidez: Volumen promedio diario mayor a 250,000.
- Filtrado por antigüedad: Activos con al menos 10 años de historial.
- Filtrado sectorial: Distribución balanceada con segmentos sectoriales clásicos (tecnología, servicios financieros, salud, consumo cíclico, industriales, servicios de comunicación, consumo defensivo, energía, bienes raíces, utilidades y materiales básicos).
- Filtro de AUM: Para las acciones del S&P500 un tamaño promedio mayor a $15 billones y para las acciones del FTSE100 un tamaño promedio mayor a $3.5 billones.

Todos estos filtros aseguran un universo final de activos estable, representativo y con datos consistentes.

1.7 Patrones de acceso previstos. El dataset se utiliza de las siguientes maneras:

- Lecturas intensivas: Durante el entrenamiento de XGBoost, que consume el archivo Parquet completo.
- Acceso incremental: Cada actualización mensual requiere leer los últimos 30 días y agregarlos a la base.
- Consultas tabulares: Para validación, reconstrucción y auditoría de los datos.

1.8 Riesgos y limitaciones.

- Dependencia de fuentes externas, debido a que las API´s de YFinance y FRED pueden sufrir caídas temporales, cambios en sus endpoints o variaciones en los datos disponibles.
- Diferencias entre mercados de EE. UU. y Reino Unido, que tienen calendarios bursátiles distintos; el *forward fill* puede introducir valores repetidos en días sin negociación.
- El volumen creciente o el almacenamiento recurrente de históricos + 30 días puede generar crecimiento progresivo de la base, afectando tiempos de lectura si no se optimiza.

2. Enfoque de modelado de datos.

Para este trabajo de investigación se ha optado por utilizar un modelo relacional que sido implementado en SQLite. Esta elección se basa en los siguientes factores:

- Flexibilidad del filtrado: Al tener que hacer diferentes filtrados (por sector, UK o EEUU).
- Preparación para siguientes etapas del trabajo: La idea de este dataset es convertirlo eventualmente en un modelo de machine learning de gradient boosting, para este tipo de modelo se requiere una estructura tabular rígida.

Además, el trabajo en su modelado de trabajo se fundamenta en una relación de uno a muchos entre la entidad de definición del activo y los registros históricos.

- Entidad Activo (Maestro): Contiene el ticker y sector, el cual se transforma en la base para aplicar las cuotas sectoriales por país.
- Entidad SerieHistórica (Transaccional): Se hace el almacenamiento de la evolución diaria de los datos, en la cual la clave primaria está compuesta por ticker + fecha, lo que ayuda a impedir que ocurran duplicidad de precios para un mismo activo en un mismo día.
- Entidad Macro (Satelital): Contiene los datos de inflación para EEUU y UK (T10YIE y CPI UK) vinculados a la tabla principal mediante la dimensión temporal.

Por último 2 factores adicionales nos ayudan a la justificación de este modelo, la trazabilidad ya que cualquier cambio se puede realizar mediante consultas sobre el modelo físico sin necesidad de tener que descargar de nuevo los datos y la interoperabilidad con los siguientes pasos a realizar en Python mediante diferentes librerías financieras que son legibles una vez completado el modelado

3. Justificación de la elección tecnológica.

En este proyecto se manejan tres datasets (Tabla 1, 2 y 3) con funciones distintas dentro del pipeline (ingesta → preparación/feature engineering → predicción/optimización). Por ello, la elección tecnológica se hace por capa, priorizando:

- Ejecución local sin dependencias externas.
- Robustez para actualizaciones periódicas (append).
- Rendimiento en lectura analítica para entrenamiento y scoring.

Dataset (Datos crudos y consolidados): SQLite (SGBD relacional embebido) + exportación a Parquet

Contexto de uso y patrones de operación

La Tabla 1 es la "tabla madre" de datos descargados, con granularidad diaria y un volumen de 1,539,876 filas y 9 columnas. Además, se espera actualizar por ciclos (rebalanceo mensual) agregando los últimos 30 días mediante un append.

Esto implica consultas típicas como:

- Filtrar por ticker y rango de fecha (series temporales).
- Comprobación de nulos/consistencia tras el merge.
- Operaciones de "append incremental" y validaciones.

Elección: SQLite

Se propone almacenar Tabla 1 (tabla_1_ingesta) generada mediante Python por consultas al API de YFinance y FRED en SQLite (BD local embebida en un único archivo .db) y desde ahí generar un Parquet para el consumo analítico/ML. Esto es coherente con tu flujo actual ("se guarda la tabla limpia como una base de datos SQL (e.g. universo_final.db)").

Justificación por criterios

1. **Adecuación a patrones de consulta/operación**
   - Excelente para consultas por claves (ticker, fecha) y para cargas incrementales (append).
   - Permite índices (por ejemplo, índice compuesto en (ticker, fecha)) para acelerar lecturas por ventana temporal.

2. **Integridad/consistencia**

- En un SGBD relacional es fácil asegurar reglas: tipos de datos, unicidad (ej. (ticker, fecha)), y evitar duplicados tras múltiples cargas mensuales.
- Mejor trazabilidad que archivos sueltos (CSV), especialmente cuando el proceso crece.

3. **Evolución del esquema**
   - Si en el futuro se agregan columnas (por ejemplo más factores macro o features base), SQLite permite ALTER TABLE y migraciones controladas.
   - En CSV la "evolución" suele volverse frágil (columnas que cambian, orden, tipos, etc.).

4. **Escalabilidad esperada**
   - Para 1M–10M filas SQLite suele rendir bien en entorno local si se indexa correctamente y las escrituras se hacen por lotes.
   - Si el proyecto escalara a decenas/cientos de millones de filas o concurrencia multiusuario, sería el punto donde PostgreSQL (o un motor analítico como DuckDB) ganaría.

5. **Mantenibilidad/operación**
   - Operación simple: un archivo .db, backup fácil, reproducible.
   - Se integra directo con Python (sqlite3, pandas.to_sql), lo cual ya estás usando.

6. **Viabilidad de ejecución en local sin dependencias externas**
   - SQLite no requiere servidor ni infraestructura. Esto cumple el requisito de ejecución local.

Alternativas comparadas (y trade-offs)

- **CSV**: Ultra simple, pero peor para integridad, duplicados y consultas.
- **Parquet-only**: Gran rendimiento analítico, pero menos natural para "append incremental" controlado y para constraints; suele requerir reescrituras/particionado.
- **PostgreSQL/MySQL**: Más robustos y escalables, pero implican servidor, configuración y dependencias externas (rompe el requisito de "local sin dependencias").
- **DuckDB**: Muy atractivo para analytics local (lee Parquet directo), pero como "repositorio maestro" con integridad y control de cargas, SQLite sigue siendo más simple de operar para tu caso.

Trade-off asumido: Se acepta que SQLite no es el motor ideal para analítica columnar pesada; por eso se complementa con exportación a Parquet para entrenamiento y transformación en pipeline.

4. Descripción del MVP implementado.

El modelo se implementa mediante 2 tablas claves en SQLite:

1. Tabla_1_Ingesta: La tabla que se genera desde Python será la primera tabla por utilizar en SQLite en la cual se almacen los datos crudos generados a partir del API para un total de 1,539,876 filas y 9 columnas. Se utiliza para tener el universo completo en la cual se hará un modelado de datos y filtraciones para elegir cual será nuestras 200 acciones que pasarán a calificar para potencialmente ser elegidas dentro de nuestro portafolio.

2. Universo_final: Esta será la tabla final generada a partir de diferentes cambios aplicados a la tabla_1_ingesta para asegurar calidad de datos, homogeneidad y garantía de activos utilizables por el modelo de machine learning. Esta tabla mantiene las 9 columnas de la tabla 1 pero disminuye la cantidad de acciones para un total de 499,467 filas. Se aplican los siguientes cambios desde la tabla_1_ingesta para llegar al universo_final:

    a. Consistencia de sectores: Se aplican cambios para asegurar que todos los activos esten clasificados bajo los mismos sectores.
        i. En esta sección de los 622 activos se encuentran 27 con sector inconsistentes que deben ser corregidos.

    b. Filtros de calidad de inversión: Se aplican los siguientes filtros para asegurar que de nuestra cantidad de acciones para potencialmente convertirse en el portafolio tengan características homogéneas que permitan un análisis válido y estandarizado. Al aplicar los siguientes filtros se encuentra que de las 503 acciones del S&P500 pasan el filtro 407, de las 99 acciones del FTSE100 pasan 87 y los ETFs se mantienen los 20.

        i. Consistencia histórica: Se filtran activos que no posean al menos 2,300 registros de esta manera capturar su comportamiento a través de diversos periodos económicos.
        ii. Liquidez: Los activos deben tener un promedio igual o mayor a 250,000 en volumen diario para eliminar el riesgo de no poder venderlos cuando sea necesario.

iii. Tamaño: Para las acciones de EEUU se debe cumplir la condición de tener un tamaño igual o mayor a $15B en promedio y para las de UK de al menos $3.5B para asegurar su continuidad en el índice.

c. Por último, al ir rellenando las cuotas por segmento para el S&P500 y FTSE100 se aplica una regla lógica de manera que si hubiese más acciones disponibles para elegir que cupos para un sector se vayan rellenando los cupos de manera que se seleccionen las acciones de mayor tamaño primero.

i. Al aplicar este filtro se encuentra que nuestro universo final pretendía ser de 200 acciones pero el FTSE100 tiene la limitación de que no tuvo suficientes acciones para llenar los cupos del sector de energia por 3 y para salud por 1, haciendo que obtengamos 196 acciones validas para nuestro modelo de machine learning de las 622 iniciales en la tabla 1.

5. Consultas/operaciones representativas.

Se han realizado diferentes consultas representativas para determinar que la transición de datos entre archivos CSV iniciales, Python y SQL:

- **Obtención de datos (Python):** Mediante Python se utilizan los tickers del archivo CSV (tickers.csv) para generar la extraccion de datos del API de YFinance y FRED a partir de la fecha de 2016-011-01 que será el archivo utilizado en SQLite para nuestro dataset (investigación_tfm.db)

- **Validación de tickers (SQLite):** Se realiza una consulta al cargar la tabla en SQLite, de manera que podamos confirmar si todos los activos solicitados mediante tickers por el API de yfinance están contenidos en el dataset o si hay faltantes y determinar si hubieron cambios o errores en el nombre o datos vacíos.

- **Validación de sectores (SQLite):** Se realiza una consulta para determinar si todos los activos tienen un sector asignado debido a que la selección final será determinada por su sector. Al identificar hay valores “unknown” crea una lista de los tickers únicos con este problema.

- **Corrección de valores (SQLite):** Al tener la lista de tickers con sector incorrecto se investiga cual debería ser el sector correcto y se realiza una operación para crear una tabla limpia con los datos adecuados en la columna de sector.

- **Filtro para selección de acciones viables (SQLite):** Se crea una tabla (universo_final) a partir de la tabla limpia contenga activos que contengan las siguientes características:

    - **ETFs:** Se selecciona un máximo de 20 ETFs para obtener diversificación en el portafolio versus solamente tener acciones, para ser considerado como válido se filtra por ETF con más de 2300 observaciones totales de manera que cumpla con aproximadamente 10 años de datos históricos y demuestre persistencia a lo largo del tiempo y comportamiento del activo en diferentes etapas económicas.

    - **FTSE100:** Para el FTSE100 nos enfocaremos en que sea una selección de acciones defensivas, es decir que no posean tanto riesgo, pero buen valor, de manera que permitan el crecimiento del portafolio a mediano y largo plazo sin crear posibles fluctuaciones grandes y balancear la cartera del S&P500 que estará tirada a un crecimiento más agresivo, pero más riesgoso. Dadas estas características se desea obtener 50 acciones del FTSE100 que cumplan con los siguientes requisitos:

        - **Sectorial (cantidad de acciones):** Servicios financieros 10, consumo defensivo 9, salud 6, energía 6, materiales básicos 6, industriales 5, consumo cíclico 3, servicios de comunicación 2, utilidades 2, bienes raíces 1. Siendo incluidas por orden de tamaño mayor a menor en caso de que hayan más de las requeridas.
        - **Tiempo:** Deberán contener al menos 2300 observaciones.

        - **Volumen:** Para evitar que compremos acciones con bajo volumen y haya riesgo de no poder venderlas en el momento adecuado se aplica un filtro que elimina las acciones con un volumen diario promedio menor a 250,000.

        - **Tamaño:** Se evita que hayan acciones con tamaño pequeño en términos del índice para evitar que haya riesgo de que ante un mal momento sean removidas de este, por lo que se aplica un filtro para eliminar las acciones que tengan un tamaño (AuM) menor a $3.5 billones.

    - **S&P500:** Para el S&P500 nos enfocaremos en que sea una selección de acciones de crecimiento, es decir que sean el principal motor para obtener rendimientos altos y dejar que las selecciones del FTSE100 y ETFs pueden

proporcionar un contrapeso en riesgo para que el portafolio tenga un buen balance. Dadas estas características se desea obtener 130 acciones del S&P500 que cumplan con los siguientes requisitos:

- **Sectorial (cantidad de acciones):** Tecnología 38, servicios financieros 17, salud 16, consumo cíclico 14, industriales 11, servicios de comunicación 10, consumo defensivo 8, energía 6, bienes raíces 4, utilidades 3, materiales básicos 3. Siendo incluidas por orden de tamaño mayor a menor en caso de que hayan más de las requeridas.
- **Tiempo:** Deberán contener al menos 2300 observaciones.
- **Volumen:** Para evitar que compremos acciones con bajo volumen y haya riesgo de no poder venderlas en el momento adecuado se aplica un filtro que elimina las acciones con un volumen diario promedio menor a 250,000.
- **Tamaño:** Se evita que hayan acciones con tamaño pequeño en términos del índice para evitar que haya riesgo de que ante un mal momento sean removidas de este, por lo que se aplica un filtro para eliminar las acciones que tengan un tamaño (AuM) menor a $15 billones.

6. Limitaciones y previsiones de evolución:

El dataset actualmente cuenta con varios limitantes:

- Los días hábiles entre las acciones del S&P500 y el FTSE100 no son exactamente iguales, esto debido a pesar de crear un filtro para eliminar acciones relativamente nuevas los días hábiles entre países son diferentes por feriados. Para equiparar esto se planea utilizar un forward fill mediante Python en siguientes procesos del proyecto en su camino a convertirse en un modelo de machine learning.

- El indicador de uk_cpi es un dato mensual y contiene un lag mensual, a diferencia del T10YIE el cual es un dato diario y disponible para todas las fechas en el momento de la consulta. Debido a esta limitación se aplica que la inflación de uk_cpi será igual para todos los días del mes y para el último mes disponible será igual al dato más reciente disponible, actualizable una vez que se haga el lanzamiento del siguiente mes.

- Al aplicar los filtros de tiempo, volumen, tamaño de mercado y sectorial, se determina que la selección de acciones del FTSE100 se disminuye en 4 para un total de 46, esto debido a que no hay suficientes acciones para cumplir con los requisitos en el sector de energía en los cuales sólo se encuentran 3 acciones versus las 6 esperadas y en el sector de salud donde sólo se encuentran 5 acciones versus las 6 esperadas.

Para siguientes pasos el dataset se espera que siga creciendo de manera mensual al añadir datos nuevos a la tabla mediante un APPEND para el universo seleccionado y como parte del proyecto de trabajo final se procederá a hacer los forward fill, rellenar el lag del uk_cpi y la creación de varios features en Python que serán utilizables por el modelo de machine learning.

II. Repositorio con el desarrollo técnico:

Repositorio con README: https://github.com/abarcamendez94/SGDB

Incluye el motor de ingesta en Python, scripts DDL/DML de transformación y documentación del modelado.