**Etapas de un sistema de Algo Trading (visión narrativa)**

**Etapa 0: Definición del objetivo**

El primer paso consiste en dejar claramente establecido el propósito del sistema de trading algorítmico. En este caso, la idea es construir desde cero un sistema que, de manera autónoma, sea capaz de analizar datos del mercado, probar estrategias, ejecutar operaciones en Binance y monitorear todo el proceso.

**Objetivo:** definir conceptualmente qué queremos lograr y cuáles son los resultados esperados (ej. automatizar la toma de decisiones de trading, reducir la intervención manual y asegurar un control del riesgo en todo momento).

**Etapa 1: Adquisición y almacenamiento de datos**

En primer lugar, es necesario contar con datos confiables, ya que estos serán la base de todo el sistema. Para ello, se descargan diariamente los precios históricos y recientes de los activos desde la API de Binance, tanto en intervalos diarios como de cuatro horas. Seguidamente, esos datos se almacenan en un Data Lake en formato Parquet, organizado por símbolo, intervalo y fecha, de modo que quede disponible para análisis futuros de manera eficiente. Como motor de consulta inicial se utiliza DuckDB por su rapidez y simplicidad, aunque más adelante se puede complementar con una base de datos como Postgres o Timescale.

**Objetivo:** asegurar la existencia de un repositorio central de datos estructurado, optimizado y preparado para ser utilizado en análisis, modelos y backtesting.

**Etapa 2: Procesamiento y análisis**

Luego de tener los datos almacenados, se pasa a la fase de procesamiento, donde se limpian y validan los registros (ejemplo: remover duplicados, verificar consistencia entre open-high-low-close). Seguidamente, se generan indicadores técnicos como RSI, MACD o medias móviles, que servirán como variables para el modelado. En este punto también se puede incorporar información externa, como resúmenes de noticias o datos macroeconómicos. Finalmente, se habilitan consultas rápidas mediante DuckDB para exploración y análisis preliminar.

**Objetivo:** transformar los datos brutos en información lista para alimentar modelos, estrategias y procesos de toma de decisión.

**Etapa 3: Modelos y backtesting**

Una vez preparado el entorno de análisis, se desarrollan y prueban distintas estrategias de trading. Primero se diseñan modelos, ya sean basados en machine learning, deep learning o reglas técnicas (ejemplo: cruces de medias móviles, modelos LSTM o Random Forest). Luego, se realizan pruebas retrospectivas (backtesting) sobre los datos históricos para validar su efectividad y medir métricas clave como el ratio de Sharpe, el drawdown máximo o la tasa de acierto. Finalmente, se identifican las estrategias más prometedoras para pasar a un entorno de prueba en vivo (paper trading).

**Objetivo:** evaluar la viabilidad de las estrategias de trading de manera controlada antes de aplicarlas en un entorno real con capital.

**Etapa 4: Ejecución automática en vivo**

Con estrategias validadas, se da paso a la fase de ejecución. Primero, el sistema convierte las señales generadas por los modelos en órdenes concretas (BUY, SELL, EXIT) que se envían automáticamente a Binance a través de su API. Seguidamente, se aplican reglas de gestión de riesgo, como definir el tamaño de la posición, colocar stop-loss y take-profit, y activar un mecanismo de kill switch si el drawdown excede un límite predefinido. Finalmente, todo el proceso queda registrado en logs y bases de datos para posteriores auditorías y revisiones.

**Objetivo:** automatizar la ejecución de operaciones de acuerdo con las estrategias, garantizando al mismo tiempo seguridad y control sobre el riesgo.

**Etapa 5: Monitoreo y orquestación**

Luego de activar el trading en vivo, se establece un sistema de monitoreo constante que permita verificar que todos los procesos estén funcionando correctamente. Primero, un orquestador como Prefect o Airflow se encarga de ejecutar los flujos diarios de ingesta de datos y análisis. Seguidamente, n8n coordina las integraciones con alertas (ejemplo: Telegram, Discord) y permite manejar señales externas como las provenientes de TradingView. Finalmente, se habilitan endpoints de salud (/health) en la API para que el sistema pueda confirmar en todo momento su correcto funcionamiento.

**Objetivo:** asegurar que todos los componentes (ingesta, modelos, trading en vivo) estén activos, funcionando sin errores y con mecanismos de alerta en caso de anomalías.

**Etapa 6: Integración de inteligencia externa**

Una vez en funcionamiento el ciclo básico de datos-modelos-trading, se pueden añadir fuentes de inteligencia complementaria. Primero, se pueden usar APIs como OpenAI para generar resúmenes de noticias o interpretar eventos macroeconómicos. Seguidamente, agentes en n8n o frameworks de orquestación inteligente pueden combinar esas señales externas con las técnicas, ajustando o recomendando parámetros de trading. Finalmente, se busca que el sistema sea capaz de reaccionar no solo a patrones históricos sino también a cambios contextuales del mercado.

**Objetivo:** enriquecer las decisiones del bot de trading con información externa y flexible, que complemente los modelos técnicos tradicionales.

**Etapa 7: Observabilidad y reporting**

Finalmente, todo el sistema se complementa con un módulo de observabilidad. Primero, se definen métricas clave como PnL, drawdown, número de operaciones y tasa de acierto. Seguidamente, se implementan reportes automáticos diarios con resúmenes de lo que ocurrió (ejemplo: operaciones ejecutadas, resultados del día, alertas). Finalmente, se integran dashboards o paneles de control (Grafana, Metabase) que permiten visualizar en tiempo real el estado del sistema.

**Objetivo:** brindar transparencia, trazabilidad y claridad sobre el desempeño y la salud del sistema en su conjunto.