Caso de Estudio: Plataforma Serverless de Datos Financieros en Azure

Desarrollado por: Julián Nevado (Data Engineer)

Visión General del Proyecto

Este proyecto está diseñado para la ingesta, limpieza y enriquecimiento de información bursátil y financiera. El objetivo principal era construir una arquitectura propia capaz de procesar grandes volúmenes de datos financieros de forma fiable, evitando reprocesamientos innecesarios y manteniendo los costes de infraestructura al mínimo.

Para lograrlo, diseñé un sistema basado en la Arquitectura Medallion (Bronze, Silver, Gold) completamente desplegado en la nube mediante Azure Functions, utilizando una base de datos PostgreSQL, también de Azure, como almacén final.

El Reto Arquitectónico

Manejar datos de diversa naturaleza presenta un desafío claro: los precios de las acciones cambian todos los días, pero los reportes financieros de las empresas (balances, cuentas de resultados y flujos de caja) solo se publican cada tres meses, y cada empresa lo hace en una fecha distinta.

Procesar todo al mismo tiempo consumiría recursos de computación de forma ineficiente y correría el riesgo de saturar la API proveedora. Para solucionar esto, separé el sistema en tres pipelines de datos independientes orquestados por cadencia y dominio.

Los Tres Pipelines de Datos

1. **Stock Data (Frecuencia Diaria):** Se encarga de descargar los precios históricos (OHLCV) de un conjunto de activos. Extrae el delta de nuevos precios, limpia valores nulos o volúmenes a cero, y finalmente calcula indicadores técnicos complejos (RSI, MACD, Medias Móviles) utilizando la librería Pandas y TA, garantizando siempre una ventana histórica suficiente (250 días) para que los cálculos matemáticos sean precisos.
2. **Fundamental Data (Frecuencia Diaria):** Descarga los metadatos y KPIs corporativos de las empresas (capitalización de mercado, sector, número de empleados). El reto aquí fue transformar grandes diccionarios anidados en un esquema relacional limpio y tipado.
3. **Financial Data (Frecuencia Semanal):** Un proceso que cuenta con 3 tablas finales y que se ejecuta los fines de semana. Descarga los reportes contables periódicos (Cash Flow, Balance Sheet, Income Statement) almacenándolos primero como

JSONB para no perder estructura original, y luego aplanándolos en tablas específicas para su análisis.

La Solución: Arquitectura Medallion

Para asegurar la calidad e integridad del dato en todo momento, implementé un diseño de tres capas:

- **Capa Bronze (Ingesta Cruda):** Los datos entran al sistema tal y como los entrega la API, almacenándose como JSONB o texto plano. Esto actúa como un histórico inmutable; si la lógica de negocio cambia en el futuro, no es necesario volver a descargar los datos desde internet.
- **Capa Silver (Limpieza y Tipado):** Aquí interviene Python y Pandas. Se leen únicamente los datos nuevos de Bronze, se filtran anomalías, se fuerzan los tipos de datos correctos y se estructuran en tablas relacionales.
- **Capa Gold (Consumo y Enriquecimiento):** La capa final orientada a negocio. Aquí es donde los datos de mercado se cruzan con los indicadores técnicos. En el caso de los reportes financieros, la promoción de Silver a Gold se realiza de forma nativa en SQL puro, moviendo los datos directamente dentro del motor de la base de datos sin sobrecargar la memoria del servidor de Python, dado que los datos en Silver ya son completos y precisos.

Decisiones Técnicas Destacadas

Durante el desarrollo, tomé varias decisiones arquitectónicas enfocadas en la resiliencia y la escalabilidad:

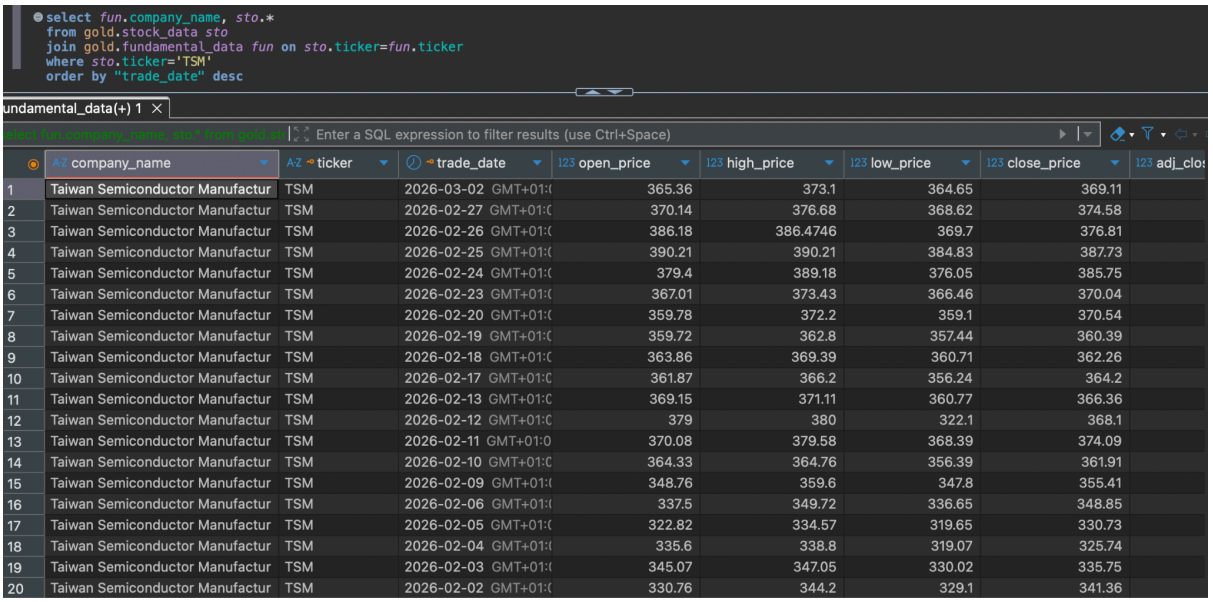
- **Orquestación Resiliente con Durable Functions:** En lugar de utilizar disparadores de tiempo estándar (que tienen límites estrictos de ejecución de 5 a 10 minutos en la nube), implementé Azure Durable Functions. Esto me permitió crear un patrón de encadenamiento donde un orquestador central gestiona el estado del flujo. Si la capa Bronze de una de las ETL falla por un problema de red, el sistema detiene la ejecución antes de corromper las capas Silver y Gold, permitiendo un manejo de errores centralizado y sin riesgo de timeouts.
- **Cargas Incrementales y Operaciones Idempotentes:** Para optimizar el rendimiento de la base de datos, diseñé el sistema basándome en lógica de cargas incrementales (Delta Loads). Python solo extrae aquellas filas cuya fecha sea posterior a la última actualización registrada. Además, las escrituras en PostgreSQL utilizan sentencias Upsert combinadas con operadores lógicos de comparación de campos para evitar actualizaciones fantasma que consumen ciclos de disco innecesarios si los datos no han mutado realmente.
- **Eficiencia Serverless:** Al empaquetar todo el código en contenedores lógicos independientes dentro de un plan de consumo de Azure, el sistema solo incurre en costes durante los escasos segundos que dura la ejecución diaria y semanal, reduciendo el gasto de infraestructura a la mínima expresión.

Tecnologías y Herramientas Utilizadas

- **Cloud & Orquestación:** Microsoft Azure, Azure Functions (Modelo V2), Azure Durable Functions.
- **Procesamiento de Datos:** Python 3, Pandas, Pandas-TA (Technical Analysis), yfinance.
- **Bases de Datos:** PostgreSQL, modelado relacional, JSONB, SQL avanzado (CTEs, Window Functions, Upserts masivos con psycopg2).
- **Ingeniería de Software:** Patrones de diseño ETL, Arquitectura Medallion y despliegue continuo (Azure CLI).

Ejemplos de extracciones

Esta extracción nos permite ver datos bursátiles, en este caso sobre el precio de apertura y cierre diario de las acciones de la empresa TSM, así como el precio más alto y el más bajo alcanzado a lo largodel día. Dada la cantidad de columnas de la tabla stock_data, no es posible visualizar todo su contenido en una única captura.



```
select fun.company_name, sto.*
from gold.stock_data sto
join gold.fundamental_data fun on sto.ticker=fun.ticker
where sto.ticker='TSM'
order by "trade_date" desc
```

undamental_data(+) 1

Enter a SQL expression to filter results (use Ctrl+Space)

	company_name	ticker	trade_date	open_price	high_price	low_price	close_price	adj_clos
1	Taiwan Semiconductor Manufactur	TSM	2026-03-02 GMT+01:(	365.36	373.1	364.65	369.11	
2	Taiwan Semiconductor Manufactur	TSM	2026-02-27 GMT+01:(	370.14	376.68	368.62	374.58	
3	Taiwan Semiconductor Manufactur	TSM	2026-02-26 GMT+01:(	386.18	386.4746	369.7	376.81	
4	Taiwan Semiconductor Manufactur	TSM	2026-02-25 GMT+01:(	390.21	390.21	384.83	387.73	
5	Taiwan Semiconductor Manufactur	TSM	2026-02-24 GMT+01:(	379.4	389.18	376.05	385.75	
6	Taiwan Semiconductor Manufactur	TSM	2026-02-23 GMT+01:(	367.01	373.43	366.46	370.04	
7	Taiwan Semiconductor Manufactur	TSM	2026-02-20 GMT+01:(	359.78	372.2	359.1	370.54	
8	Taiwan Semiconductor Manufactur	TSM	2026-02-19 GMT+01:(	359.72	362.8	357.44	360.39	
9	Taiwan Semiconductor Manufactur	TSM	2026-02-18 GMT+01:(	363.86	369.39	360.71	362.26	
10	Taiwan Semiconductor Manufactur	TSM	2026-02-17 GMT+01:(	361.87	366.2	356.24	364.2	
11	Taiwan Semiconductor Manufactur	TSM	2026-02-13 GMT+01:(	369.15	371.11	360.77	366.36	
12	Taiwan Semiconductor Manufactur	TSM	2026-02-12 GMT+01:(	379	380	322.1	368.1	
13	Taiwan Semiconductor Manufactur	TSM	2026-02-11 GMT+01:0	370.08	379.58	368.39	374.09	
14	Taiwan Semiconductor Manufactur	TSM	2026-02-10 GMT+01:(	364.33	364.76	356.39	361.91	
15	Taiwan Semiconductor Manufactur	TSM	2026-02-09 GMT+01:(	348.76	359.6	347.8	355.41	
16	Taiwan Semiconductor Manufactur	TSM	2026-02-06 GMT+01:(	337.5	349.72	336.65	348.85	
17	Taiwan Semiconductor Manufactur	TSM	2026-02-05 GMT+01:(	322.82	334.57	319.65	330.73	
18	Taiwan Semiconductor Manufactur	TSM	2026-02-04 GMT+01:(	335.6	338.8	319.07	325.74	
19	Taiwan Semiconductor Manufactur	TSM	2026-02-03 GMT+01:(	345.07	347.05	330.02	335.75	
20	Taiwan Semiconductor Manufactur	TSM	2026-02-02 GMT+01:(	330.76	344.2	329.1	341.36	

[Mi Linkedin]

Esta vez nos centramos en métricas y datos técnicos acerca de las acciones de la empresa Brookfield Corporation, consultando la misma tabla que en la extracción anterior.

```
select fun.company_name, sto.ticker, sto.trade_date, daily_return_pct, volatility, moving_avg_50, sma_200, rsi_14, macd
from gold.stock_data sto
join gold.fundamental_data fun on sto.ticker=fun.ticker
where sto.ticker='BN'
order by "trade_date" desc
```

fundamental_data(+) 1

Enter a SQL expression to filter results (use Ctrl+Space)

	company_name	ticker	trade_date	daily_return_pct	volatility	moving_avg_50	sma_200	rsi_14	macd
1	Brookfield Corporation	BN	2026-03-02 GMT+01:00	0.2737	2.5077	46.4104	44.3599	40.4043	-0.5416
2	Brookfield Corporation	BN	2026-02-27 GMT+01:00	-3.5424	2.5445	46.431	44.338	39.7468	-0.4425
3	Brookfield Corporation	BN	2026-02-26 GMT+01:00	-0.3071	2.4363	46.4604	44.3144	46.0801	-0.2974
4	Brookfield Corporation	BN	2026-02-25 GMT+01:00	1.3336	2.4392	46.4646	44.2762	46.6807	-0.2718
5	Brookfield Corporation	BN	2026-02-24 GMT+01:00	2.25	2.4279	46.4844	44.2403	43.7638	-0.2494
6	Brookfield Corporation	BN	2026-02-23 GMT+01:00	-4.3062	2.3592	46.533	44.2016	38.619	-0.1561
7	Brookfield Corporation	BN	2026-02-20 GMT+01:00	-0.4331	2.1852	46.587	44.1634	46.5234	0.0661
8	Brookfield Corporation	BN	2026-02-19 GMT+01:00	-2.6765	2.1884	46.5962	44.1156	47.434	0.1471
9	Brookfield Corporation	BN	2026-02-18 GMT+01:00	-0.6907	2.146	46.6068	44.0697	53.6228	0.2279
10	Brookfield Corporation	BN	2026-02-17 GMT+01:00	-0.0209	2.3767	46.5922	44.0127	55.3657	0.1985
11	Brookfield Corporation	BN	2026-02-13 GMT+01:00	0.2517	2.3766	46.567	43.9527	55.4164	0.1245
12	Brookfield Corporation	BN	2026-02-12 GMT+01:00	2.2523	2.3763	46.5378	43.8926	54.9569	0.0272
13	Brookfield Corporation	BN	2026-02-11 GMT+01:00	-2.3256	2.3201	46.512	43.832	50.8406	-0.0852
14	Brookfield Corporation	BN	2026-02-10 GMT+01:00	1.8349	2.276	46.5196	43.7764	55.8509	-0.1205
15	Brookfield Corporation	BN	2026-02-09 GMT+01:00	3.6718	2.2384	46.4968	43.7141	52.4819	-0.2768
16	Brookfield Corporation	BN	2026-02-06 GMT+01:00	2.0311	2.0748	46.4802	43.6498	44.9528	-0.3847
17	Brookfield Corporation	BN	2026-02-05 GMT+01:00	-3.422	2.0111	46.4782	43.5906	40.1812	-0.3478
18	Brookfield Corporation	BN	2026-02-04 GMT+01:00	2.8008	2.0762	46.4864	43.5291	46.7447	-0.2041
19	Brookfield Corporation	BN	2026-02-03 GMT+01:00	-4.2069	1.9608	46.4428	43.4653	39.43	-0.1775
20	Brookfield Corporation	BN	2026-02-02 GMT+01:00	2.2832	2.063	46.4248	43.405	49.2865	-0.0137
21	Brookfield Corporation	BN	2026-01-30 GMT+01:00	-2.1482	2.0294	46.3616	43.3365	42.163	-0.004
22	Brookfield Corporation	BN	2026-01-29 GMT+01:00	-1.1467	1.9781	46.3106	43.2725	48.2092	0.1144

Esta extracción nos muestra datos sobre las cuentas de pérdidas y ganancias de la empresa MSCI de los últimos 5 años. En este caso le hemos aplicado un filtro period_type='annual', por lo que tan solo veremos la información reportada por la empresa en sus informes anuales.

```
select *
from gold.income_statement
where ticker='MSCI' and period_type = 'annual'
order by report_date desc
```

me_statement 1

Enter a SQL expression to filter results (use Ctrl+Space)

	ticker	report_date	period_type	total_revenue	cost_of_revenue	gross_profit	operating_income	net_income	ebitda
	MSCI	2025-12-31 GMT+01	annual	3,134,459,000	550,366,000	2,584,093,000	1,713,567,000	1,202,305,000	1,922,715,000
	MSCI	2024-12-31 GMT+01	annual	2,856,128,000	514,382,000	2,341,746,000	1,528,518,000	1,109,128,000	1,747,943,000
	MSCI	2023-12-31 GMT+01	annual	2,528,920,000	446,581,000	2,082,339,000	1,384,609,000	1,148,592,000	1,714,959,000
	MSCI	2022-12-31 GMT+01	annual	2,248,598,000	404,341,000	1,844,257,000	1,207,640,000	870,573,000	1,357,908,000

Por último consultamos datos sobre el estado de flujos de efectivo, que nos permite analizar, por ejemplo, el crecimiento del flujo de caja operativo y el CAPEX de la empresa a lo largo de los últimos, entre otros datos.

Mi Linkedin

```sql
select
  ticker,
  report_date,
  operating_cash_flow,
  capital_expenditure,
  investing_cash_flow ,
  financing_cash_flow
  from gold.cashflow
  where ticker='META'
  and period_type='annual'
  order by report_date desc
```

flow 1

Enter a SQL expression to filter results (use Ctrl+Space)

ticker	report_date	operating_cash_flow	capital_expenditure	investing_cash_flow	financing_cash_flow
META	2025-12-31	115,800,000,000	-69,691,000,000	-102,003,000,000	-20,370,000,000
META	2024-12-31	91,328,000,000	-37,256,000,000	-47,150,000,000	-40,781,000,000
META	2023-12-31	71,113,000,000	-27,045,000,000	-24,495,000,000	-19,500,000,000
META	2022-12-31	50,475,000,000	-31,186,000,000	-28,970,000,000	-22,136,000,000

[Mi Linkedin]