

# **Arquitectura de Sistemas de Trading Algorítmico: Implementación de la Metodología Fusion Concepts en el Entorno Google Antigravity**

## **1. Introducción: La Convergencia entre Smart Money Concepts y la Inteligencia Artificial Agentica**

La evolución de los mercados financieros ha alcanzado un punto de inflexión donde las estrategias discrecionales de alta precisión, conocidas coloquialmente como Smart Money Concepts (SMC) y Liquidity Inducement Theorem (LIT), pueden ser sistematizadas mediante la nueva generación de entornos de desarrollo integrados. Este informe técnico detalla la arquitectura, lógica y despliegue de un sistema de trading algorítmico basado en la metodología "Trading Fusion Concepts", específicamente diseñado para operar dentro del ecosistema **Google Antigravity**.

El propósito de este documento es trascender la simple automatización de señales para construir un "Trader Artificial" capaz de interpretar la narrativa de la liquidez institucional. A diferencia de los bots tradicionales basados en indicadores rezagados (RSI, MACD), la metodología Fusion requiere una comprensión profunda de la estructura fractal del mercado, las trampas de liquidez y la psicología de masas, elementos que ahora pueden ser modelados utilizando la arquitectura de agentes de Gemini 3 Pro disponible en Antigravity.<sup>1</sup>

### **1.1 El Desafío de la Automatización de la Acción del Precio**

La metodología Fusion, tal como se describe en los documentos fundamentales "La Biblia" y "Modelos de Entrada"<sup>3</sup>, se basa en la interpretación visual de "trampas" y "bloques de órdenes". Para un trader humano, identificar un "bloque trampa" implica reconocer patrones visuales sutiles. Para un algoritmo, esto representa un desafío de visión computacional y lógica booleana compleja.

La transición de lo discrecional a lo algorítmico exige una definición matemática rigurosa de conceptos abstractos:

- ¿Qué constituye exactamente un "Cambio de Carácter" (CHOCH) válido?
- ¿Cómo se cuantifica una "Inducción" de liquidez?
- ¿Cómo se programa la "responsabilidad" y el control de la "Bestia Interna" (psicología) en un script de Python?

Este informe aborda estas interrogantes descomponiendo la estrategia en tres módulos operativos: **Fusion Fase 1** (Estructura y Continuidad), **Fusion Fase 2** (Trampas y Aceleración), y **Módulos Temporales** (Rango de Asia), integrados bajo un marco de gestión de riesgos estricto.

## 1.2 El Paradigma Google Antigravity: Desarrollo "Agent-First"

La elección de Google Antigravity como entorno de desarrollo no es trivial. A diferencia de los IDE convencionales, Antigravity introduce un flujo de trabajo "Agent-First".<sup>4</sup> Esto permite al desarrollador actuar como un arquitecto que orquesta múltiples agentes de IA especializados.

En el contexto de Fusion, esto significa que no escribiremos un código monolítico lineal. En su lugar, desplegaremos una constelación de agentes:

1. **Agente de Estructura:** Especializado en identificar rangos y fractalidad (Fase 1).
2. **Agente de Trampas:** Entrenado para detectar patrones de inducción y dobles rompimientos (Fase 2).
3. **Agente de Ejecución:** Responsable de la gestión de órdenes, cálculo de lotaje y ejecución de las "Ráfagas".
4. **Agente de Riesgo (The Guardian):** Un supervisor independiente que emula el control emocional, capaz de desconectar el sistema si se violan las reglas de pérdida diaria.<sup>3</sup>

Esta arquitectura permite una adaptabilidad y robustez que los scripts tradicionales no pueden ofrecer, utilizando la "memoria de contexto" de Antigravity para aprender de los patrones de mercado históricos y ajustar la detección de bloques de órdenes en tiempo real.<sup>2</sup>

---

## 2. Deconstrucción Matemática de la Estructura de Mercado (Fusion Fase 1)

La Fase 1 de Fusion es la base sobre la cual se construye toda la lógica operativa. Se define como una estrategia de continuidad basada en la alineación de rangos y la identificación de zonas de oferta y demanda válidas. Para el algoritmo, la "visión" del mercado se traduce en vectores de precios y matrices de tiempo.

### 2.1 Definición Algorítmica del Rango Operativo

El concepto de "Rango" es la unidad atómica de la metodología. Un rango no es simplemente la distancia entre un máximo y un mínimo; es un contenedor de acción del precio validado por eventos estructurales específicos.

#### 2.1.1 Identificación de Puntos de Giro (Swing Points)

El primer paso para el algoritmo es la digitalización de la estructura. Utilizamos la definición

de fractales de Williams o "Swing Points" para filtrar el ruido del mercado.

- **Swing High (SH):** Se define como una vela cuyo máximo es estrictamente superior a los máximos de las  $n$  velas precedentes y las  $n$  velas posteriores.
  - *Parámetro n:* Para la estrategia Fusion en M15, un valor de  $n=2$  ofrece el equilibrio óptimo entre sensibilidad y robustez.
- **Swing Low (SL):** Inverso al SH, una vela con un mínimo inferior a sus vecinas.

En Python, utilizando la biblioteca pandas, esto se implementa mediante ventanas rodantes (rolling windows) que verifican la condición de extremidad local. Este proceso transforma una serie temporal continua en una secuencia discreta de nodos estructurales.

### 2.1.2 El Algoritmo del Cambio de Carácter (CHOCH)

El documento "La Biblia" establece que el CHOCH es indispensable para marcar el inicio y fin de un rango.<sup>3</sup> La precisión en esta detección es crítica para evitar falsos positivos.

#### Lógica de Validación del CHOCH:

1. **Identificación del Último Retroceso Válido:**
  - En una tendencia bajista, es el último Swing High que generó el bajo más bajo (Lower Low) actual.
  - En una tendencia alcista, es el último Swing Low que generó el alto más alto (Higher High) actual.
2. **Condición de Ruptura (Break Condition):**
  - El precio debe cerrar *por encima* (para CHOCH alcista) o *por debajo* (para CHOCH bajista) del nivel del retroceso válido.
  - Regla Crítica de Cuerpo vs. Mecha: La metodología Fusion distingue explícitamente entre una ruptura con cuerpo y una con mecha. El algoritmo debe verificar:

$\$Cierre_{\{vela\}} > Nivel_{\{SH\}}$

Si solo el  $\$Maximo_{\{vela\}} > Nivel_{\{SH\}}$  pero el cierre es inferior, el evento se clasifica como "Toma de Liquidez" (Sweep) y no como CHOCH. Esta distinción es vital, ya que un Sweep suele indicar una reversión inmediata (trampa), mientras que un CHOCH con cuerpo indica un cambio de tendencia real.<sup>3</sup>

## 2.2 Zonificación Fractal: Premium y Discount

Una vez establecido el rango operativo mediante la detección de CHOCHs consecutivos (uno de inicio y uno de finalización, o una ruptura de continuación BOS), el algoritmo debe dividir el espacio de precios.

El sistema debe calcular dinámicamente la matriz de Fibonacci del rango actual:

- **Zona Premium (Ventas):** El área comprendida entre el 50% y el 100% de un rango

bajista.

- **Zona de Descuento (Compras):** El área comprendida entre el 0% y el 50% de un rango alcista.

Filtro de Ejecución Estricto:

El algoritmo implementará una regla de "hard-block" para cualquier señal que contradiga esta lógica.

- *Condición:* Si Tendencia == ALCISTA y Precio > 50% del Rango, la función de compra execute\_buy() retorna False inmediatamente.
- *Justificación:* Esto sistematiza la regla de "Comprar barato y vender caro" descrita en la documentación <sup>3</sup>, eliminando la impulsividad humana de perseguir el precio.

## 2.3 Alineación de Temporalidades (Timeframe Alignment)

La robustez de la Fase 1 reside en la confluencia fractal. El sistema en Antigravity debe mantener y sincronizar dos hilos de procesamiento de datos paralelos.

Variable	Temporalidad Mayor (HTF)	Temporalidad Menor (LTF)	Función
Swing Trading	H4 (4 Horas)	M15 (15 Minutos)	Captura de movimientos estructurales amplios.
Intradía	H1 (1 Hora)	M5 (5 Minutos)	Operativa diaria estándar descrita en "La Biblia".
Scalping	M15 (15 Minutos)	M1 (1 Minuto)	Entradas de alta precisión y "Ráfagas".

Lógica de Coherencia Direccional:

El algoritmo evalúa el estado de los rangos en ambas temporalidades:

1. **Alineación Perfecta (Continuidad):**

- HTF = Alcista.
- LTF = Alcista.
- *Acción:* El sistema busca agresivamente Bloques de Órdenes en LTF que coincidan con zonas de descuento de HTF. Esta es la configuración de mayor probabilidad.<sup>3</sup>

2. **Desalineación (Retroceso):**

- HTF = Alcista.
- LTF = Bajista.
- **Acción:** El sistema entra en modo "Retroceso". Busca ventas en LTF, pero con objetivos (Take Profit) limitados estrictamente al 50% del rango de HTF o al primer Bloque de Órdenes de HTF. Una vez alcanzada esa zona, el algoritmo deja de vender y comienza a buscar la alineación alcista.

Esta lógica de alineación previene que el bot opere en contra de la fuerza mayor del mercado, reduciendo drásticamente la tasa de fallos.

---

### 3. Ingeniería de la Trampa: Implementación de Fusion Fase 2 e Inducciones

Si la Fase 1 es la estructura, la Fase 2 es la astucia. Esta fase está diseñada para capitalizar los errores de otros traders (retail y SMC convencional). Para el algoritmo, esto implica dejar de buscar "patrones válidos" y empezar a buscar "patrones fallidos predecibles".

#### 3.1 Detección Algorítmica de Bloques Trampa (Trap Blocks)

El documento de Inducciones<sup>3</sup> introduce una regla mecánica específica para identificar trampas: la **Regla de los 2 BOS**.

**Pseudocódigo Lógico para Detección de Trampas:**

Python

```
def detectar_bloque_trampa(estructura_mercado):
    # Paso 1: Identificar la secuencia de rupturas
    bos_list = estructura_mercado.get_recent_bos()

    if len(bos_list) >= 2:
        ultimo_bos = bos_list[-1]
        penultimo_bos = bos_list[-2]

    # Paso 2: Localizar el Bloque de Órdenes asociado al segundo BOS
    order_block = identificar_ob_en_origen(ultimo_bos)

    # Paso 3: Etiquetar como Trampa
```

```

# La teoría dicta que el mercado induce a comprar/vender en este bloque
# pero lo violará para buscar liquidez más profunda.
order_block.tipo = 'TRAMPA'
order_block.accion = 'EVITAR_ENTRADA_DIRECTA'

return order_block
return None

```

Este bloque no se utiliza para entrar. Se utiliza como **referencia de liquidez**. El algoritmo coloca una alerta *detrás* de este bloque. Cuando el precio rompe este bloque trampa (induciendo a los traders de Fase 1 a perder), el sistema activa la búsqueda de la verdadera entrada.

### 3.2 El Bloque de Alta Probabilidad y el Filtro de Liquidez

La Fase 2 exige que operemos solo en "Bloques de Alta Probabilidad". ¿Cómo se define esto en código?

1. **Existencia de Barrido (Sweep Condition):** El bloque debe haberse formado *inmediatamente después* de tomar liquidez de un máximo o mínimo anterior. Si la vela que forma el bloque no tomó liquidez, su probabilidad disminuye.<sup>3</sup>
2. **Ausencia de Liquidez Previa (Clean Arrival):**
  - El algoritmo escanea la acción del precio *antes* del bloque.
  - Si detecta "Equal Lows" (bajos iguales con una tolerancia de +/- 1 pip) o una "Trendline Liquidity" (línea de tendencia obvia de 3 toques) justo antes del bloque, el bloque se marca como **INVÁLIDO**.
  - *Razón:* Esa liquidez actúa como un imán. El precio atravesará el bloque para capturar esa liquidez. El algoritmo debe esperar en un bloque más profundo o esperar una nueva formación tras el barrido de esa liquidez.

### 3.3 Mecánica de Ejecución de "Ráfagas" (Burst Strategy)

Una de las características más distintivas y agresivas de Fusion Fase 2 es el sistema de "Ráfagas". Esto permite al algoritmo maximizar el retorno en escenarios de alta convicción, buscando ratios de 1:10 o superiores.<sup>3</sup>

El sistema de ejecución no es unitario; es secuencial y condicional.

#### 3.3.1 Secuencia de Entrada Escalonada

1. **Entrada Primaria (The Scout):**
  - Se ejecuta en el Bloque de Alta Probabilidad tras la confirmación (CHOCH en M1).
  - Riesgo estándar (ej. 0.5%).
2. **Ráfaga 1 (Momentum Injection):**
  - *Trigger:* Una vez que la Entrada Primaria está activa, el algoritmo monitorea M1/M3. Si

el precio rompe una subestructura y deja un nuevo Imbalance (FVG), se coloca una orden límite en ese FVG.

- **Gestión:** Al activarse la Ráfaga 1, el Stop Loss de la Entrada Primaria se mueve automáticamente a Break Even (BE). Esto financia el riesgo de la segunda operación.

### 3. Ráfaga 2 (Maximization):

- **Trigger:** Repetición del proceso anterior en una etapa más avanzada del movimiento.
- **Gestión:** Los SL de Entrada Primaria y Ráfaga 1 se ajustan para asegurar ganancias (Trailing Stop técnico, no porcentual).

Gestión de Riesgo Automatizada en Ráfagas:

El Agente de Riesgo en Antigravity debe calcular el riesgo compuesto. Si la regla es arriesgar 1% total, la distribución podría ser 0.5% (Primaria), 0.25% (Ráfaga 1), 0.25% (Ráfaga 2), o utilizar las ganancias flotantes para financiar el riesgo adicional ("Free Rolling"), permitiendo un riesgo teórico de 0% en las ráfagas si la gestión de BE es precisa.

---

## 4. Estrategia Temporal: Rango de Asia y Sesión de Londres

La automatización de la estrategia del Rango de Asia es altamente eficiente debido a su naturaleza basada en reglas horarias fijas y comportamientos repetitivos de los algoritmos interbancarios.

### 4.1 Sincronización Temporal y Definición de Sesiones

El bot debe operar con una precisión de reloj atómico, manejando las conversiones de zona horaria críticas mencionadas en la documentación (UTC-5 Lima/NY vs. Hora del Servidor/Broker).

**Tabla de Horarios Operativos (UTC-5):**

Evento	Hora Inicio	Hora Fin	Acción del Algoritmo
<b>Formación Rango Asia</b>	19:00	00:00	Modo Pasivo: Registro de High/Low. No operar.
<b>Zona Muerta (Pre-London)</b>	00:00	02:00	Modo Monitorización:

			Validar estructura.
<b>London Killzone</b>	02:00	06:00	Modo Activo: Ejecución de estrategia de barrido (Sweep).
<b>New York Killzone</b>	07:00	10:00	Modo Activo: Continuación o Reversión NY.

## 4.2 Lógica de Barrido y Reversión (Sweep & Reverse)

La estrategia no busca la ruptura (breakout) del rango asiático para continuar, sino la manipulación de la misma.<sup>3</sup>

### Algoritmo de Decisión:

1. **Cálculo del Rango:** A las 00:00, el sistema fija Asia\_High y Asia\_Low.
2. **Esperar Manipulación:** El bot espera que el precio cruce uno de estos límites *durante* la London Killzone (02:00 - 06:00).
  - *Escenario A (Barrido de Bajos):* Precio < Asia\_Low. El sesgo se vuelve inmediatamente **ALCISTA**. El algoritmo asume que los stops de venta han sido capturados.
  - *Escenario B (Barrido de Altos):* Precio > Asia\_High. El sesgo se vuelve **BAJISTA**.
3. **Trigger de Entrada (Confirmación):**
  - El sistema no entra en el toque. Espera una confirmación en M3 o M5.
  - *Patrón Requerido:* Un CHOCH interno o una vela envolvente que reingrese al rango asiático.
  - *Objetivo:* El objetivo inicial siempre es el lado opuesto del rango (ej. si barre el Low, TP en Asia\_High).

## 5. Arquitectura de Implementación en Google Antigravity

La implementación en Antigravity aprovecha su capacidad para gestionar múltiples "agentes" que colaboran en un entorno unificado. Esta sección detalla la configuración técnica del proyecto.

## 5.1 Configuración del "Manager Surface"

En la interfaz de gestión de Antigravity, definiremos una topología de agentes que refleja la estructura de un fondo de trading algorítmico.

### 5.1.1 Agente de Datos (Data Ingest Agent)

- **Responsabilidad:** Conexión con APIs de proveedores de datos (ej. Binance, OANDA) y normalización de series temporales.
- **Tecnología:** Python (ccxt, pandas).
- **Tarea Específica:** Generar dataframes limpios y sincronizados para M1, M5, M15, H1 y H4. Detectar y corregir gaps de datos o velas erróneas ("bad ticks") que podrían activar falsas señales de CHOCH.

### 5.1.2 Agente de Análisis Estructural (The Analyst)

- **Responsabilidad:** Implementar la lógica de "La Biblia" y "Fusion Fase 1/2".
- **Tecnología:** Librería smartmoneyconcepts (personalizada), numpy.
- **Capacidades Gemini 3 Pro:** Utilizaremos el modelo para "ver" patrones complejos que son difíciles de codificar linealmente, como la "calidad" de una consolidación o la "limpieza" de un Trendline Liquidity. El agente puede recibir capturas del gráfico y devolver probabilidades basadas en visión artificial entrenada en los patrones del PDF.<sup>2</sup>

### 5.1.3 Agente de Riesgo y Psicología (Risk Guardian)

- **Responsabilidad:** Emular el control de la "Bestia Interna".
- **Reglas Inmutables (Hard Constraints):**
  - *Daily Loss Limit:* Si Pérdida\_Hoy >= Max\_Loss\_Perfil, deshabilitar trading por 24h.
  - *Profit Lock:* Si Ganancia\_Hoy >= Objetivo\_Diario, reducir riesgo al 10% del estándar o detener.
  - *Correlación:* Evitar abrir trades simultáneos en pares altamente correlacionados (ej. EURUSD y GBPUSD) en la misma dirección para no duplicar riesgo.

## 5.2 Flujo de Trabajo: Planificación, Implementación y Verificación

Antigravity permite un ciclo de desarrollo iterativo <sup>5</sup>:

1. **Fase de Planificación:** El usuario (Arquitecto) define la estrategia en lenguaje natural: "Implementar la regla de 2 BOS para detección de trampas en EURUSD M15".
2. **Fase de Implementación:** Los agentes generan el código Python necesario, creando clases para TrapDetector y OrderBlockManager.
3. **Fase de Verificación (Backtesting):** Antigravity ejecuta el código sobre datos históricos (últimos 2 años).
  - *Análisis de Artefactos:* El sistema genera gráficos marcando dónde detectó trampas. El usuario revisa visualmente si coinciden con la teoría.
  - *Refinamiento:* Si el bot detecta demasiados falsos positivos, el usuario ajusta la

instrucción: "Sé más estricto con la definición de BOS, requiere cierre de vela completo".

### 5.3 Stack Tecnológico Python

El núcleo del sistema se construirá sobre un stack robusto de Python, aprovechando librerías especializadas mencionadas en la investigación <sup>6</sup>:

- **pandas & numpy:** Para manipulación vectorial de datos OHLCV de alta velocidad.
- **smartmoneyconcepts (o implementación propia):** Para la detección de FVG (Fair Value Gaps), Swing Highs/Lows y Order Blocks.<sup>8</sup>
- **Jesse o Freqtrade:** Como frameworks base para el manejo del ciclo de vida del bot (backtesting, live trading, gestión de base de datos).<sup>9</sup> Estos frameworks permiten integrar la lógica personalizada de Fusion sin tener que escribir el motor de ejecución desde cero.

---

## 6. Gestión de Riesgos y Psicología Algorítmica

El documento "La Biblia" hace hincapié en que la estrategia técnica es inútil sin gestión de riesgo y control emocional.<sup>3</sup> En un sistema automatizado, el "psicotrading" se convierte en "reglas de gestión de capital codificadas".

### 6.1 Perfilamiento del Trader (Algorítmico)

El sistema debe permitir la configuración de perfiles de riesgo que evolucionan con el rendimiento de la cuenta, simulando el crecimiento del trader.<sup>3</sup>

Perfil	Riesgo por Operación	Objetivo Mensual	Drawdown Máximo Diario	Acción tras Límite
<b>Novato</b>	0.25%	3% - 5%	0.50%	Apagado Total (Kill Switch)
<b>Intermedio</b>	0.50%	5% - 7%	1.00%	Revisión de Estrategia
<b>Avanzado</b>	1.00%	> 10%	2.00%	Reducción de Riesgo a 0.25%

## 6.2 Protocolo "Kill Switch" y Anti-Tilt

Para evitar el comportamiento destructivo (la "Bestia"):

1. **Contador de Rachas:** Si el bot pierde 3 operaciones consecutivas, entra en un periodo de enfriamiento (Cooldown) de 4 horas. Esto evita que el algoritmo intente "recuperar" pérdidas en condiciones de mercado desfavorables.
2. **Filtro de Noticias:** Integración con un calendario económico. El Agente de Riesgo bloquea nuevas entradas 30 minutos antes y después de noticias de alto impacto (NFP, FOMC, CPI), protegiendo la cuenta de deslizamientos (slippage) y volatilidad errática, alineado con la recomendación de "cuidar el capital".<sup>3</sup>

## 6.3 Gestión de Retiros

Siguiendo la filosofía de Fusion Fase 2<sup>3</sup>, el sistema debe tener una lógica de "Profit Taking" macro.

- Si  $\text{Beneficio\_Mes} \geq 10\%$ , el bot notifica al usuario para considerar un retiro y reduce automáticamente el riesgo operativo para preservar el capital ganado, evitando devolverlo al mercado en los días restantes del mes.

---

## 7. Análisis de Datos y Patrones Emergentes

Al integrar toda la información de los documentos de investigación, surgen patrones de segundo y tercer orden que refinan la estrategia.

### 7.1 La Jerarquía de la Liquidez

El análisis cruzado de los documentos sugiere una jerarquía clara:

1. **Liquidez Interna:** El precio busca esto primero para alimentar movimientos cortos.
2. **Liquidez Externa (Rangos):** El precio busca esto para reversiones mayores.
3. **Inducciones (Trampas):** Son el combustible de aceleración.

**Implicación Estratégica:** El algoritmo debe priorizar entradas donde *múltiples* tipos de liquidez convergen. Una entrada de "Ráfaga" es más potente si ocurre después de barrer liquidez interna Y externa simultáneamente (ej. barrido de Asia High + mitigación de FVG interno).

### 7.2 Sinergia Fase 1 + Fase 2

No son estrategias aisladas. La Fase 2 es el filtro de calidad de la Fase 1.

- El algoritmo de Fase 1 puede detectar 10 oportunidades de compra en un día.
- El módulo de Fase 2 (Trampas) puede invalidar 8 de esas 10 por tener "liquidez previa" o

- ser "segundo BOS".
- Resultado: El bot opera menos (2 trades), pero con una tasa de acierto y ratio riesgo-beneficio dramáticamente superiores.
- 

## 8. Conclusión y Hoja de Ruta de Implementación

Este informe establece una base sólida para la construcción de un sistema de trading institucional automatizado en Google Antigravity. Hemos transformado conceptos abstractos como "inducción" y "narrativa" en lógica ejecutable.

### Pasos Inmediatos para el Despliegue:

1. **Prototipado en Antigravity:** Configurar el Agent Manager y cargar los PDFs de Fusion como base de conocimiento.
2. **Desarrollo del Agente de Estructura:** Codificar la lógica de Swing Points y CHOCH en Python y validarla visualmente con gráficos históricos.
3. **Implementación del Rango de Asia:** Al ser la estrategia más mecánica, debe ser la primera en pasar a producción (demo).
4. **Integración de Fase 2:** Añadir la capa de detección de trampas y la lógica de "Ráfagas" una vez que la estructura base sea estable.

La combinación de la metodología Fusion Concepts con la potencia de los agentes de IA de Gemini 3 Pro ofrece una oportunidad única para crear un sistema que no solo sigue reglas, sino que entiende la estructura profunda del mercado, operando con la precisión de una máquina y la sofisticación de un trader institucional experto.

### Obras citadas

1. Build with Google Antigravity, our new agentic development platform, fecha de acceso: diciembre 30, 2025,  
<https://developers.googleblog.com/build-with-google-antigravity-our-new-agentic-development-platform/>
2. The Era of Action Model with Gemini 3 Pro & Google Antigravity - Medium, fecha de acceso: diciembre 30, 2025,  
<https://medium.com/google-cloud/the-era-of-action-with-gemini-3-pro-google-antigravity-853b935c5df0>
3. Trading "La biblia" destapando mitos y leyendas.pdf
4. Getting Started with Google Antigravity, fecha de acceso: diciembre 30, 2025,  
<https://codelabs.developers.google.com/getting-started-google-antigravity>
5. Tutorial : Getting Started with Google Antigravity | by Romin Irani - Medium, fecha de acceso: diciembre 30, 2025,  
<https://medium.com/google-cloud/tutorial-getting-started-with-google-antigravity-b5cc74c103c2>
6. Algo trading research with Google's Antigravity and Gemini 3 - YouTube, fecha de

- acceso: diciembre 30, 2025, <https://www.youtube.com/watch?v=ypFG006G4WQ>
- 7. Google Antigravity Tutorial: Build a Finance Risk Dashboard - DataCamp, fecha de acceso: diciembre 30, 2025,  
<https://www.datacamp.com/tutorial/google-antigravity-tutorial>
  - 8. Smart Money Concepts (smc) BETA - smartmoneyconcepts · PyPI, fecha de acceso: diciembre 30, 2025, <https://pypi.org/project/smartmoneyconcepts/0.0.14/>
  - 9. Jesse - The Open-source Python Bot For Trading Cryptocurrencies, fecha de acceso: diciembre 30, 2025, <https://jesse.trade/>
  - 10. Algo Trading Research Using Google Antigravity, Gemini 3, and Freqtrade - YouTube, fecha de acceso: diciembre 30, 2025,  
<https://www.youtube.com/watch?v=YN25SqCFspc>