# 📊 Herramienta de Análisis de Mercados Financieros MCP

Una plataforma de análisis financiero impulsada por IA que combina **Model Context Protocol (MCP)**, **smolagents**, **FastAPI** y **Streamlit** para ofrecer informes de análisis de inversión de nivel profesional. El sistema utiliza Modelos de Lenguaje de Gran Escala para orquestar herramientas de estrategias de trading e interpretar datos financieros, transformando datos de mercado en bruto en información accionable para inversiones.


## 🎯 Descripción General

Esta aplicación proporciona **5 tipos distintos de análisis** a través de una interfaz web moderna:

| Tipo de Análisis | Descripción | Caso de Uso |
|---|---|---|
| 📉 **Análisis Técnico** | 4 estrategias de trading en una sola acción | Análisis profundo de patrones de precios |
| 🔍 **Escáner de Mercado** | Compara múltiples acciones simultáneamente | Encuentra las mejores oportunidades |
| 📊 **Análisis Fundamental** | Interpretación de estados financieros | Evalúa la salud de la empresa |
| 🌐 **Análisis Multi-Sector** | Comparación entre sectores | Diversificación de cartera |
| 🔄 **Análisis Combinado** | Técnico + Fundamental juntos | Tesis de inversión completa |

### ¿Qué Hace que Esto Sea Diferente?

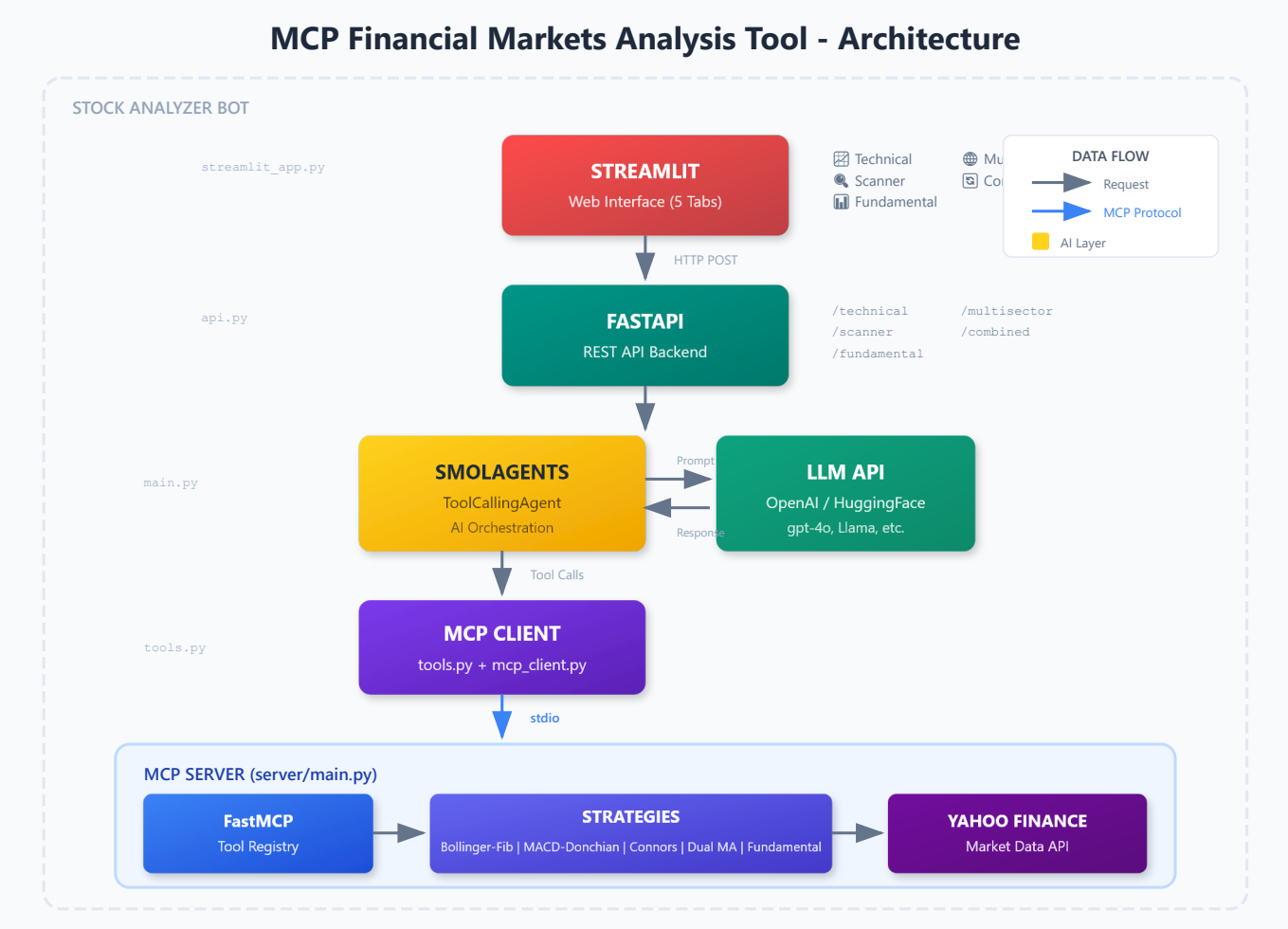
A diferencia de las herramientas de análisis tradicionales que solo muestran números, este sistema usa **IA para interpretar** los datos:

```
Herramienta Tradicional: "RSI = 28.5, MACD = -2.3, P/E = 15.2"

Esta Aplicación: "AAPL muestra condiciones de sobreventa con RSI en 28.5,
                  sugiriendo una potencial oportunidad de reversión a la media.
                  Combinado con fundamentos sólidos (P/E de 15.2 por debajo del
                  promedio del sector), esto presenta una señal de COMPRA con
                  alta convicción..."
```

# Arquitectura

## Componentes del Sistema




**Resumen del Flujo de Datos:**

1. **Streamlit** → El usuario interactúa con la interfaz web (5 pestañas de análisis)
2. **FastAPI** → La API REST recibe solicitudes, valida la entrada
3. **smolagents** → El agente de IA decide qué herramientas llamar
4. **LLM API** → OpenAI/HuggingFace procesa prompts, guía la selección de herramientas
5. **MCP Client** → Conecta smolagents con el servidor MCP vía stdio
6. **MCP Server** → Ejecuta herramientas de análisis financiero
7. **Strategies** → Calcula indicadores técnicos, ejecuta backtests
8. **Yahoo Finance** → Proporciona datos de mercado

## Estructura de Carpetas

```
mcp_financial_markets_analysis_tool/
│
├── server/                          # MCP Server (Herramientas Financieras)
│   ├── main.py                      # Punto de entrada del servidor
│   ├── strategies/                  # Implementaciones de estrategias de trading
│   │   ├── bollinger_fibonacci.py   # Bollinger + Fibonacci
│   │   ├── macd_donchian.py         # MACD + Donchian Channel
```

```
│   │   ├── connors_zscore.py        # Connors RSI + Z-Score
│   │   ├── dual_moving_average.py   # Cruce 50/200 EMA
│   │   ├── bollinger_zscore.py      # Bollinger + Z-Score
│   │   ├── fundamental_analysis.py  # Estados Financieros
│   │   ├── performance_tools.py     # Herramientas de Backtesting
│   │   └── unified_market_scanner.py# Escáner Multi-Acción
│   ├── utils/
│   │   └── yahoo_finance_tools.py   # Descarga de Datos y Cálculo de Indicadores
│   └── README.md                    # 📄 Documentación Detallada del Servidor
│
├── stock_analyzer_bot/              # Smolagents Bot (Orquestación de IA)
│   ├── __init__.py
│   ├── main.py                      # Funciones de análisis y prompts LLM
│   ├── api.py                       # Endpoints REST de FastAPI
│   ├── tools.py                     # Wrappers de herramientas Smolagents
│   ├── mcp_client.py                # Gestor de conexión MCP
│   └── README.md                    # 📄 Documentación Detallada del Bot
│
├── streamlit_app.py                 # Interfaz Web (5 Pestañas de Análisis)
├── .env                             # Variables de entorno
├── requirements.txt                 # Dependencias de Python
└── README.md                        # 📄 Este archivo
```

---

## 🤖 Entendiendo Smolagents

### ¿Qué es Smolagents?

**Smolagents** es una biblioteca Python de código abierto de Hugging Face que facilita la construcción de agentes de IA que pueden usar herramientas. Es el "cerebro" de nuestra aplicación.

> *"smolagents está diseñado para hacer extremadamente fácil construir y ejecutar agentes usando solo unas pocas líneas de código."* - HuggingFace

### Características Clave de Smolagents que Usamos

| Característica | Cómo la Usamos |
|---|---|
| **ToolCallingAgent** | Orquesta llamadas a nuestras 7 herramientas financieras |
| **LiteLLMModel** | Conecta con OpenAI (gpt-4o, gpt-4-turbo) |
| **InferenceClientModel** | Conecta con modelos de HuggingFace |
| **@tool decorator** | Envuelve nuestras herramientas MCP para uso del agente |
| **Multi-step reasoning** | El agente decide qué herramientas llamar y en qué orden |

### Cómo Funciona Smolagents en Nuestra Aplicación

```python
# 1. Definir herramientas que el agente puede usar
from smolagents import tool
```

```python
@tool
def bollinger_fibonacci_analysis(symbol: str, period: str = "1y") -> str:
    """Analiza una acción usando Bandas de Bollinger + retroceso de Fibonacci."""
    # Llama al servidor MCP, retorna datos de análisis
    return _call_mcp_tool("analyze_bollinger_fibonacci_performance", {...})

# 2. Crear un agente con el LLM y las herramientas
from smolagents import ToolCallingAgent, LiteLLMModel

model = LiteLLMModel(model_id="gpt-4o")
agent = ToolCallingAgent(
    tools=[bollinger_fibonacci_analysis, macd_donchian_analysis, ...],
    model=model,
    max_steps=25,
)

# 3. Ejecutar el agente con un prompt
report = agent.run("""
    Analiza la acción AAPL usando las 4 estrategias técnicas.
    Crea un informe markdown completo con recomendaciones.
""")
```

### Proceso de Decisión del Agente

Cuando solicitas un análisis, el agente:

```
1. LEE el prompt: "Analiza AAPL con 4 estrategias"

2. PLANIFICA: "Necesito llamar 4 herramientas: bollinger_fibonacci, macd_donchian,
               connors_zscore, dual_moving_average"

3. EJECUTA: Llama cada herramienta, recibe datos

4. SINTETIZA: Combina todos los resultados, identifica patrones

5. GENERA: Crea informe markdown profesional con recomendación
```

Por eso los resultados están **interpretados**, no solo mostrados.

---

## 🖩 Interfaz Streamlit - 5 Tipos de Análisis

### Pestaña 1: 📈 Análisis Técnico

**Propósito:** Análisis profundo de una sola acción usando 4 estrategias de trading

**Qué Hace:**

- Llama 4 herramientas de estrategia para una acción

- Cada estrategia proporciona: señal, puntaje, % de retorno, ratio de Sharpe, caída máxima
- La IA sintetiza en un informe cohesivo con recomendación

**Estrategias Utilizadas:**

1. **Bollinger-Fibonacci** - Soporte/resistencia con bandas de volatilidad
2. **MACD-Donchian** - Momentum con detección de breakouts
3. **Connors RSI + Z-Score** - Señales de reversión a la media
4. **Dual Moving Average** - Seguimiento de tendencia (Golden/Death Cross)

**Mejor Para:** "¿Debería comprar/vender/mantener esta acción específica?"

---

### Pestaña 2: 🔍 Escáner de Mercado

**Propósito:** Comparar múltiples acciones y clasificar oportunidades

**Qué Hace:**

- Ejecuta las 4 estrategias en cada acción de tu lista
- Compara el rendimiento entre acciones
- Clasifica de mejor a peor oportunidad
- Identifica selecciones de consenso

**Entrada de Ejemplo:** `AAPL, MSFT, GOOGL, META, NVDA, AMD`

**Mejor Para:** "¿Cuál acción en este grupo es la mejor oportunidad?"

---

### Pestaña 3: 📊 Análisis Fundamental

**Propósito:** Analizar la salud financiera de la empresa desde sus estados financieros

**Qué Hace:**

- Recupera estado de resultados, balance general, flujo de caja
- Calcula ratios clave: P/E, ROE, deuda-capital, márgenes
- La IA interpreta la salud financiera
- Crea tesis de inversión

**Métricas Analizadas:**

- **Rentabilidad:** Ingresos, Ingreso Neto, Márgenes
- **Crecimiento:** Crecimiento de ingresos, crecimiento de ganancias
- **Liquidez:** Ratio corriente, ratio rápido
- **Apalancamiento:** Ratios de deuda, cobertura de intereses
- **Retornos:** ROE, ROA

**Mejor Para:** "¿Esta empresa es financieramente saludable?"

---

### Pestaña 4: 🌐 Análisis Multi-Sector

**Propósito:** Comparar acciones a través de diferentes sectores del mercado

**Qué Hace:**

- Analiza múltiples sectores (Banca, Tecnología, Energía Limpia, etc.)
- Ejecuta 4 estrategias en cada acción de cada sector
- Compara el rendimiento ENTRE sectores
- Identifica las mejores oportunidades de todo el universo

**Sectores Predeterminados:**

```
Banca: JPM, BAC, WFC, C, GS, MS, USB, PNC, TFC, COF
Tecnología: AAPL, MSFT, GOOGL, META, NVDA, AMD, CRM, ORCL, ADBE, INTC
Energía Limpia: TSLA, NIO, RIVN, LCID, PLUG, SEDG, NEE, ICLN, ENPH
```

**Mejor Para:** "¿Dónde debería invertir en todo el mercado?"

⚠ **Nota:** Esto es computacionalmente intensivo (120+ llamadas de herramientas para 3 sectores × 10 acciones)

---

### Pestaña 5: 🔄 Análisis Combinado

**Propósito:** Fusionar análisis Técnico y Fundamental para una imagen completa

**Filosofía:**

- **Análisis Fundamental** = "QUÉ comprar" (calidad de la empresa)
- **Análisis Técnico** = "CUÁNDO comprar" (timing)
- **Combinado** = Vista de inversión de 360 grados

**Alineación de Señales:**

| Señal FA | Señal AT | Interpretación |
|---|---|---|
| Alcista | Alcista | ☑ Alta convicción COMPRAR |
| Alcista | Bajista | ⚠ Buena empresa, mal timing - ESPERAR |
| Bajista | Alcista | ⚠ Rebote técnico, fundamentos débiles - PRECAUCIÓN |
| Bajista | Bajista | ❌ Alta convicción EVITAR |

**Mejor Para:** "Dame la imagen completa de inversión"

---

## 🚀 Inicio Rápido

### Prerrequisitos

- Python 3.10+
- Clave de API de OpenAI (recomendado) o token de HuggingFace

- Conexión a internet (datos de Yahoo Finance)

## Instalación

```
# Clonar el repositorio
git clone <repository-url>
cd mcp_financial_markets_analysis_tool

# Crear entorno virtual
python -m venv venv
source venv/bin/activate  # Linux/Mac
# o
.\venv\Scripts\activate   # Windows

# Instalar dependencias
pip install -r requirements.txt
```

## Configuración del Entorno

Crear un archivo `.env` en la raíz del proyecto:

```
# Requerido - Clave API LLM (elegir una)
OPENAI_API_KEY=sk-tu-clave-openai-aqui
# O
HF_TOKEN=hf_tu-token-huggingface

# Opcional - Configuración del Modelo
SMOLAGENT_MODEL_ID=gpt-4o           # Modelo predeterminado
SMOLAGENT_MODEL_PROVIDER=litellm     # litellm o inference
SMOLAGENT_MAX_STEPS=25               # Pasos máximos de razonamiento

# Opcional - Valores Predeterminados
DEFAULT_ANALYSIS_PERIOD=1y
DEFAULT_SCANNER_SYMBOLS=AAPL,MSFT,GOOGL,AMZN
```

## Ejecutar la Aplicación

```
# Terminal 1: Iniciar el backend FastAPI
uvicorn stock_analyzer_bot.api:app --reload --port 8000

# Terminal 2: Iniciar el frontend Streamlit
streamlit run streamlit_app.py
```

Abrir el navegador en `http://localhost:8501`

---

# 🔧 Componentes Principales

## 1. MCP Server (`server/`)

El **Model Context Protocol Server** proporciona todas las herramientas de análisis financiero. Es un proceso independiente al que el bot se conecta vía stdio.

**Características Clave:**

- 5 estrategias de análisis técnico
- Backtesting de rendimiento con métricas
- Análisis fundamental desde estados financieros
- Escáner de mercado multi-acción

📚 **Documentación Detallada:** server/README.md

## 2. Stock Analyzer Bot (`stock_analyzer_bot/`)

La **capa de orquestación impulsada por smolagents** que usa LLMs para llamar herramientas y generar informes.

**Características Clave:**

- ToolCallingAgent con 7 herramientas envueltas
- 5 funciones de análisis para diferentes casos de uso
- Ingeniería de prompts inteligente
- Endpoints REST de FastAPI

📚 **Documentación Detallada:** stock_analyzer_bot/README.md

## 3. Frontend Streamlit (`streamlit_app.py`)

La **interfaz web** que proporciona 5 pestañas de análisis con historial de sesión.

**Características Clave:**

- 5 pestañas de tipos de análisis
- Barra lateral de configuración de modelo
- Seguimiento del historial de análisis
- Renderizado de informes Markdown

---

# 📡 Referencia de API

## Endpoints Disponibles

| Endpoint | Método | Descripción |
|---|---|---|
| `/health` | GET | Verificación de salud e info de versión |
| `/technical` | POST | Acción única, 4 estrategias |
| `/scanner` | POST | Comparación multi-acción |

| Endpoint | Método | Descripción |
|---|---|---|
| `/fundamental` | POST | Análisis de estados financieros |
| `/multisector` | POST | Análisis entre sectores |
| `/combined` | POST | Técnico + Fundamental |

### Ejemplo de Llamada API

```
curl -X POST "http://localhost:8000/technical" \
  -H "Content-Type: application/json" \
  -d '{"symbol": "AAPL", "period": "1y"}'
```

### Formato de Respuesta

```
{
  "report": "# Análisis Técnico Completo de AAPL\n...",
  "symbol": "AAPL",
  "analysis_type": "technical",
  "duration_seconds": 45.2
}
```

---

## ⚙ Configuración

### Modelos LLM Soportados

| Proveedor | Model ID | Mejor Para |
|---|---|---|
| OpenAI | `gpt-4o` | Mejor calidad, recomendado |
| OpenAI | `gpt-4o-mini` | Más rápido, económico |
| OpenAI | `gpt-4-turbo` | Buen balance |
| HuggingFace | `meta-llama/Llama-3.1-70B-Instruct` | Código abierto |

### Períodos de Análisis

Períodos válidos: `1d`, `5d`, `1mo`, `3mo`, `6mo`, `1y`, `2y`, `5y`, `10y`, `ytd`, `max`

### Parámetros de Estrategia

| Estrategia | Parámetros Clave |
|---|---|
| Bollinger-Fibonacci | window=20, num_std=2 |
| MACD-Donchian | fast=12, slow=26, signal=9 |

| Estrategia | Parámetros Clave |
|---|---|
| Connors RSI | rsi_period=3, streak=2, rank=100 |
| Dual MA | short=50, long=200, type=EMA |

## Ejemplos de Informes

### Estructura del Informe de Análisis Técnico

```
# Análisis Técnico Completo de AAPL
*Fecha de Análisis: 2024-01-15*
*Precio Actual: $185.92*

## Resumen Ejecutivo
[2-3 párrafos sintetizando todos los hallazgos de las estrategias]

## Comparación de Rendimiento de Estrategias
| Estrategia | Señal | Puntaje | Retorno | Sharpe | DD Máx |
|------------|-------|---------|---------|--------|--------|
| Bollinger-Fib | COMPRA | +45 | 12.3% | 1.2 | -8.5% |
| MACD-Donchian | MANTENER | +15 | 8.1% | 0.9 | -12.1% |
| ... | ... | ... | ... | ... | ... |

## Análisis Individual de Estrategias
[Desglose detallado de cada estrategia]

## Evaluación de Riesgo
[Análisis de volatilidad y caída]

## Recomendación Final: **COMPRAR**
[Razonamiento de apoyo]
```

## Seguridad y Descargos de Responsabilidad

### Seguridad de Claves API

- Nunca envíes archivos `.env` al control de versiones
- Usa variables de entorno para todos los datos sensibles
- Las claves API nunca se registran ni almacenan

### Descargo de Responsabilidad Financiera

⚠ **IMPORTANTE:** Este software es solo para **fines educativos e investigación**.

- Todos los resultados de análisis deben ser verificados independientemente
- El rendimiento pasado no garantiza resultados futuros
- Esto NO es asesoría financiera
- Consulta un asesor financiero licenciado antes de invertir

- Los autores no asumen responsabilidad por decisiones de inversión

### Fuentes de Datos

- Datos de mercado de Yahoo Finance (sujeto a sus términos de servicio)
- Los datos pueden tener retrasos, brechas o inexactitudes
- Siempre verifica los datos contra fuentes oficiales

---

## 🛠️ Solución de Problemas

| Problema | Solución |
|---|---|
| "MCP server not found" | Verifica que `server/main.py` existe en la raíz del proyecto |
| "Connection refused" | Inicia FastAPI: `uvicorn stock_analyzer_bot.api:app --port 8000` |
| "Authentication error" | Verifica `OPENAI_API_KEY` en `.env` |
| "Timeout" | Reduce el número de acciones o aumenta el timeout |
| "Agent stopped early" | Aumenta el parámetro `max_steps` |

---

## 📚 Documentación Adicional

| Documento | Descripción |
|---|---|
| server/README.md | Herramientas del MCP Server, estrategias, parámetros |
| stock_analyzer_bot/README.md | Integración de Smolagents, endpoints API |
| HuggingFace Smolagents Docs | Documentación oficial de smolagents |
| MCP Documentation | Especificación del Model Context Protocol |

---

## 🤝 Contribuir

1. Haz fork del repositorio
2. Crea una rama de característica (`git checkout -b feature/nueva-estrategia`)
3. Implementa tus cambios
4. Agrega pruebas si aplica
5. Envía un pull request

### Agregar Nuevas Estrategias

1. Crea módulo de estrategia en `server/strategies/`
2. Regístrala con el servidor MCP en `server/main.py`
3. Crea wrapper de herramienta en `stock_analyzer_bot/tools.py`
4. Actualiza prompts en `stock_analyzer_bot/main.py`

---

## 📄 Licencia

Este proyecto se proporciona para fines educativos. Los usuarios deben cumplir con:

- Términos de Servicio de Yahoo Finance
- Términos de Servicio de OpenAI / HuggingFace
- Regulaciones financieras locales aplicables

---

## 🙏 Agradecimientos


- Smolagents por Hugging Face
- FastMCP por el framework MCP
- yfinance por los datos de mercado
- FastAPI por la API REST
- Streamlit por la interfaz web


---