

GeneTrader: Guía Técnica Completa

¿Qué es GeneTrader?

GeneTrader es una **plataforma educativa** que demuestra cómo los **Algoritmos Genéticos (AG)** pueden "evolucionar" estrategias de trading automáticamente, imitando el proceso de selección natural de Darwin.

Objetivo Principal

En lugar de que un humano programe reglas de trading manualmente como:

"Compra cuando el precio cruza la media móvil de 20 días"

El algoritmo genético **descubre estas reglas automáticamente** probando miles de combinaciones y seleccionando las más rentables.

Arquitectura del Sistema

1. Módulo de Datos de Mercado (`market-store.ts`)

```
Candle = { time, open, high, low, close, volume }
```

- Carga datos históricos de precios en formato OHLCV
- Soporta CSV personalizado o datos de ejemplo (SPY, BTC, AAPL)
- Cada "vela" representa un período (día) con precios de apertura, máximo, mínimo y cierre

2. ADN de las Estrategias (`genetics.ts`)

Cada estrategia se representa como un **Cromosoma** (array de reglas):

```
Cromosoma = [
    Regla 1: "SI RSI(14) < 30 ENTONCES COMPRAR (peso: 0.8)",
    Regla 2: "SI SMA(20) > SMA(50) ENTONCES COMPRAR (peso: 0.6)",
    Regla 3: "SI precio > SMA(200) ENTONCES VENDER (peso: 0.5)",
    ...
]
```

Cada Regla contiene:

Campo	Descripción	Ejemplo
indicator1	Indicador a comparar	RSI, SMA, EMA, precio
operator	Operador de comparación	>, <, crosses_above
indicator2 O threshold	Valor a comparar	SMA(50) o número 30
action	Qué hacer si se cumple	BUY, SELL, CLOSE
weight	Importancia de la regla	0.0 - 1.0

3. Indicadores Técnicos (indicators.ts)

Calcula señales del mercado:

Indicador	Qué mide	Uso típico
SMA (Media Móvil Simple)	Promedio de precios en N períodos	Tendencia general
EMA (Media Móvil Exponencial)	Como SMA pero da más peso a datos recientes	Reacciona más rápido
RSI (Índice de Fuerza Relativa)	Momento del precio (0-100)	Sobrecompra (>70) / Sobreventa (<30)

Ejemplo de cálculo SMA:

```
SMA(20) = (Precio[hoy] + Precio[ayer] + ... + Precio[hace 19 días]) / 20
```

4. Motor de Backtesting (backtester.ts)

Simula cómo habría funcionado una estrategia en el pasado:

Para cada vela en los datos históricos:

1. Evaluar todas las reglas del cromosoma
2. Votar: cada regla que se cumple vota por su acción (BUY/SELL)
3. Ejecutar la acción con más votos (si supera umbral mínimo)
4. Registrar trades y calcular P&L

Métricas calculadas:

Métrica	Descripción	Valor ideal
Total Return	Ganancia/pérdida total (%)	> 0%
Sharpe Ratio	Retorno ajustado al riesgo	> 1.0
Max Drawdown	Mayor caída desde un pico	< 20%
Win Rate	% de trades ganadores	> 50%
Profit Factor	Ganancias / Pérdidas	> 1.5

5. Algoritmo Genético (genetic-algorithm.ts)

El corazón del sistema. Imita la evolución biológica:

GENERACIÓN 0:

Crear 50 estrategias aleatorias (población inicial)

PARA cada generación:

1. EVALUAR: Backtestear cada estrategia → obtener fitness
2. SELECCIONAR: Las mejores sobreviven (torneo)

3. CRUZAR: Combinar reglas de 2 padres exitosos → hijo
4. MUTAR: Cambiar aleatoriamente algunos parámetros
5. REPETIR

Operadores Genéticos:

Operador	Qué hace	Analogía biológica
Selección por Torneo	Elige las mejores estrategias para reproducirse	Supervivencia del más apto
Cruce (Crossover)	Combina reglas de 2 estrategias padres	Reproducción sexual
Mutación	Modifica aleatoriamente parámetros	Mutación genética aleatoria
Elitismo	Preserva las mejores estrategias sin cambios	Proteger a los campeones

Ejemplo de Cruce:

Padre A: [Regla1, Regla2, Regla3, Regla4]
 Padre B: [Regla5, Regla6, Regla7, Regla8]
 ↓ punto de cruce
 Hijo: [Regla1, Regla2, Regla7, Regla8]

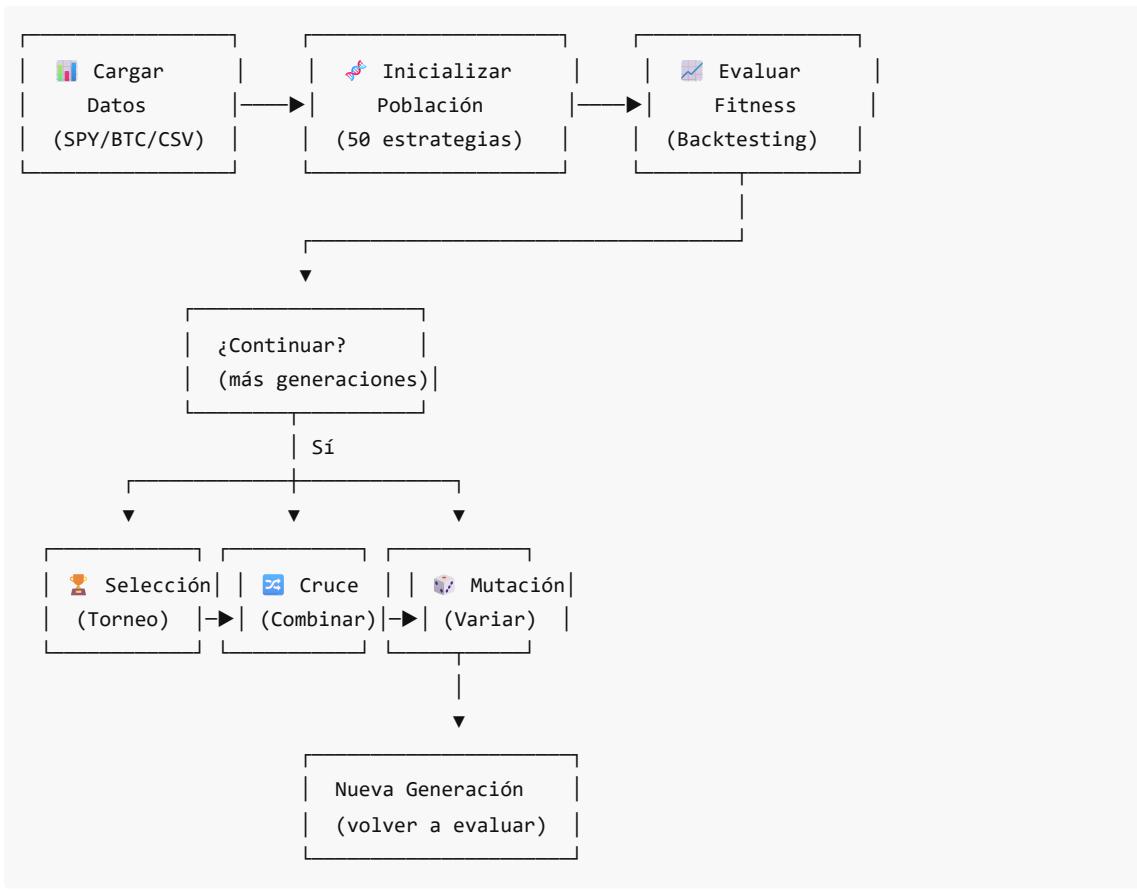
Ejemplo de Mutación:

Antes: "SI RSI(14) < 30 ENTONCES COMPRAR"
 Después: "SI RSI(21) < 25 ENTONCES COMPRAR"
 (periodo y umbral mutaron)

6. Visualización (Componentes React)

Componente	Archivo	Función
PriceChart	chart/PriceChart.tsx	Gráfico de velas con señales C/V
DataLoader	chart/DataLoader.tsx	Carga de datos CSV o ejemplo
EvolutionChart	dashboard/EvolutionChart.tsx	Progreso del fitness por generación
MetricsPanel	dashboard/MetricsPanel.tsx	KPIs de la mejor estrategia
Leaderboard	dashboard/Leaderboard.tsx	Top 10 estrategias
DNAInspector	strategy/DNAInspector.tsx	Reglas en lenguaje humano
GAControls	controls/GAControls.tsx	Sliders de parámetros
SimulationControls	controls/SimulationControls.tsx	Play/Pause/Reset
GlossaryModal	glossary/GlossaryModal.tsx	Definiciones educativas

Flujo Completo de Ejecución



📊 Parámetros Configurables

Parámetro	Rango Recomendado	Efecto
Tamaño de Población	50-100	Más = mayor diversidad genética, pero más lento
Tasa de Mutación	3-8%	Más = más exploración, menos estabilidad
Tasa de Cruce	70-85%	Más = más mezcla de características
Tasa de Elitismo	5-15%	Más = preserva más campeones
Función Fitness	Sharpe recomendado	Qué métrica optimizar

Funciones de Fitness disponibles:

1. **Net Profit**: Maximiza ganancia total (puede ignorar riesgo)
2. **Sharpe Ratio**: Equilibra retorno y riesgo (recomendado)
3. **Min Drawdown**: Minimiza pérdidas máximas
4. **Win Rate**: Maximiza % de trades ganadores
5. **Balanced**: Combina múltiples métricas

📁 Estructura de Archivos

```

src/
├── app/
│   ├── page.tsx          # Página principal
│   ├── layout.tsx        # Layout con metadata
│   └── globals.css       # Estilos del tema oscuro
├── components/
│   ├── chart/
│   │   ├── PriceChart.tsx
│   │   └── DataLoader.tsx
│   ├── controls/
│   │   ├── GAControls.tsx
│   │   └── SimulationControls.tsx
│   ├── dashboard/
│   │   ├── MetricsPanel.tsx
│   │   ├── EvolutionChart.tsx
│   │   └── Leaderboard.tsx
│   ├── strategy/
│   │   └── DNAInspector.tsx
│   └── glossary/
│       └── GlossaryModal.tsx
└── lib/
    ├── indicators.ts      # SMA, EMA, RSI
    ├── backtester.ts       # Motor de simulación
    ├── genetic-algorithm.ts # Operadores genéticos
    ├── glossary-data.ts   # Términos educativos
    └── utils.ts            # Funciones auxiliares
├── store/
    ├── market-store.ts     # Estado de datos de mercado
    └── ga-store.ts         # Estado del algoritmo genético
└── types/
    ├── trading.ts          # Tipos de trading
    └── genetics.ts         # Tipos del AG
public/
└── data/
    ├── spy_data.csv        # S&P 500 ETF
    ├── btc_data.csv         # Bitcoin
    └── aapl_data.csv        # Apple Inc.

```

Valor Educativo

GeneTrader enseña conceptos de:

Algoritmos Genéticos

- Representación cromosómica
- Operadores de selección, cruce y mutación
- Convergencia evolutiva
- Balance exploración vs explotación

Trading Algorítmico

- Indicadores técnicos (SMA, EMA, RSI)

- Backtesting histórico
- Métricas de rendimiento (Sharpe, Drawdown)
- Gestión de riesgo

Desarrollo Web Moderno

- Next.js 15 con App Router
- React 19 con hooks
- TypeScript estricto
- Zustand para estado global
- Visualización con Lightweight Charts y Recharts

🚀 Comandos

```
# Desarrollo  
npm run dev  
  
# Producción  
npm run build  
npm run start  
  
# Abrir en navegador  
http://localhost:3000
```

⚠ Disclaimer

GeneTrader es una herramienta **exclusivamente educativa**. Los resultados de backtesting histórico **no garantizan rendimientos futuros**. No es asesoría financiera. Siempre realice su propia investigación antes de invertir.

📚 Referencias

- [Lightweight Charts Documentation](#)
- [Genetic Algorithms - Wikipedia](#)
- [Technical Analysis - Investopedia](#)