



PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA INFORMÁTICA

*Tesis Magíster en Ingeniería Informática*

# ***“High Frequency Trading automático aplicado al mercado accionario nacional”***

Enero de 2013

Marco Andrés Alarcón San Martín  
Licenciado en Ciencias de la Ingeniería PUCV  
marco.alarcon.s@gmail.com

Dr. Broderick Crawford Labrín (guía)

**Resumen:** en este documento se presenta el trabajo realizado en investigación e implementación de las prácticas conocidas como High Frequency Trading y Algorithmic Trading bajo el concepto de automatización mediante técnicas informáticas. Primero se presentan los objetivos que motivan la investigación, junto con su justificación para incluirlos. Seguidamente, se presenta qué es y cómo se compone un sistema de High Frequency Trading. Después se indican algunos de las posibles técnicas de trading que permiten cierto grado de automatización. Se finaliza con el diseño y construcción de un sistema funcional de trading automático que emplea una de las técnicas revisadas, junto con los resultados, mejoras aplicadas y análisis de algunos aspectos clave encontrados en la ejecución de las pruebas.

**Palabras claves:** Mercado Accionario, High Frequency Trading, Algorithmic Trading, Metaheurísticas, PSO.

## ÍNDICE DE CAPÍTULOS

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Introducción.....</b>  | <b>06</b> |
|          | 1.1 Discusión Bibliográfica.....  | 07        |
| <b>2</b> | <b>Análisis de Objetivos y Metodología.....</b>                               | <b>08</b> |
|          | 2.1 Objetivo General.....   | 08        |
|          | 2.2 Objetivos Específicos.....  | 08        |
|          | 2.3 Justificación de los Objetivos.....                                       | 10        |
|          | 2.4 Metodología de la Investigación.....                                      | 12        |
| <b>3</b> | <b>Trading Algorítmico y de Alta Frecuencia.....</b>                          | <b>14</b> |
|          | 3.1 Métodos estadísticos de Algorithmic Trading y High Frecuency Trading..... | 20        |
|          | 3.2 Modelos Metaheurísticos.....  | 24        |
|          | 3.2.1 Particle Swarm Optimization.....  | 26        |
|          | 3.3 Modelos basados en Aprendizaje de Máquina.....                            | 30        |
|          | 3.3.1 Redes Neuronales.....   | 31        |
|          | 3.3.2 Redes Bayesianas.....   | 35        |
| <b>4</b> | <b>Diseño y resolución del Problema.....</b>                                  | <b>38</b> |
|          | 4.1 Diseño conceptual sistema Algorithmic Trading.....                        | 39        |
|          | 4.2 Selección de mercado e instrumentos financieros.....                      | 46        |
|          | 4.3 Obtención de datos relevantes.....  | 48        |
|          | 4.4 Diseño de sistema Algorithmic Trading / High Frecuency Trading.....       | 52        |
|          | 4.4.1 Modelo adaptivo con Particle Swarm Optimization.....                    | 53        |
|          | 4.5 Implementación del Sistema.....   | 58        |
|          | 4.5.1 Implementación de Particle Swarm Optimization.....                      | 58        |
|          | 4.5.2 Implementación de Trader Automático.....                                | 62        |
|          | 4.6 Pruebas del Sistema.....  | 71        |
|          | 4.6.1 Pruebas versión inicial.....  | 72        |
|          | 4.6.2 Pruebas versión optimizada.....   | 73        |
| <b>5</b> | <b>Conclusiones.....</b>  | <b>85</b> |
| <b>6</b> | <b>Bibliografía.....</b>  | <b>88</b> |

## ÍNDICE DE TABLAS

|  |    |
|--|----|
| <b>Tabla 3.1:</b> Valores típicos para Particle Swarm Optimization.....                          | 27 |
| <b>Tabla 3.2:</b> Modelos de Redes Neuronales para Data Mining.....                              | 33 |
| <b>Tabla 4.1:</b> Ejemplo de datos de la interfaz pública intraday para acciones nacionales..... | 48 |
| <b>Tabla 4.2:</b> Muestra de los indicadores de Dólar observado.....                             | 51 |
| <b>Tabla 4.3:</b> Variables para modelo adaptivo Particle Swarm Optimization.....                | 53 |
| <b>Tabla 4.4:</b> Resultados experimento 3.....  | 75 |
| <b>Tabla 4.5:</b> Resultados experimento 4.....  | 76 |
| <b>Tabla 4.6:</b> Resultados experimento 5.....  | 78 |
| <b>Tabla 4.7:</b> Resultados experimento 6.....  | 80 |

## ÍNDICE DE FIGURAS

|   |    |
|---|----|
| <b>Figura 3.1:</b> Proceso típico de High Frequency Trading.....                              | 15 |
| <b>Figura 3.2:</b> Proceso de Trading Algorítmico.....  | 16 |
| <b>Figura 3.3:</b> Elementos básicos de un sistema High Frequency Trading.....                | 17 |
| <b>Figura 3.4:</b> Incorporación de la información en mercados ineficientes.....              | 17 |
| <b>Figura 3.5:</b> Diagrama de flujo del algoritmo básico de Particle Swarm Optimization..... | 27 |
| <b>Figura 3.6:</b> Topologías para Particle Swarm Optimization.....                           | 28 |
| <b>Figura 3.7:</b> Modelo de Neurona estándar.....  | 31 |

|   |    |
|---|----|
| <b>Figura 3.8:</b> Arquitectura de capa de un Artificial Neural System.....                   | 32 |
| <b>Figura 3.9:</b> Ejemplo de Red Bayesiana.....  | 35 |
| <b>Figura 4.1:</b> Sistemas externos con los que interactúa un Algorithmic Trader.....        | 39 |
| <b>Figura 4.2:</b> Sistemas externos y módulos internos del sistema.....                      | 41 |
| <b>Figura 4.3:</b> Ejemplo de trailing stop-loss.....   | 44 |
| <b>Figura 4.4:</b> Trabajo ETL de carga diaria de datos.....                                  | 50 |
| <b>Figura 4.5:</b> Diagrama de secuencia para Algorithmic Trading en modalidad compra.....    | 52 |
| <b>Figura 4.6:</b> Ejemplo de crossover creciente en una estrategia de 2 Moving Averages..... | 54 |
| <b>Figura 4.7:</b> Bandas de control en una estrategia de 2 Moving Averages.....              | 55 |
| <b>Figura 4.8:</b> Modelo básico de implementación de PSO.....                                | 59 |
| <b>Figura 4.9:</b> Modelo completo de implementación de PSO.....                              | 61 |
| <b>Figura 4.10:</b> Modelo básico de implementación de AT.....                                | 63 |
| <b>Figura 4.11:</b> Modelo completo de la implementación de AT.....                           | 66 |
| <b>Figura 4.12:</b> Extensión del sistema AT.....   | 70 |
| <b>Figura 4.13:</b> Variación parámetros experimento 3.....                                   | 74 |
| <b>Figura 4.14:</b> Variación parámetros experimento 4.....                                   | 77 |
| <b>Figura 4.15:</b> Variación parámetros experimento 5.....                                   | 79 |
| <b>Figura 4.16:</b> Variación parámetros experimento 6.....                                   | 81 |
| <b>Figura 4.17:</b> Aplicación modelo AT experimento 7.....                                   | 82 |
| <b>Figura 4.18:</b> Variación parámetros experimento 8.....                                   | 83 |
| <b>Figura 4.19:</b> Aplicación modelo AT experimento 8.....                                   | 84 |

## ÍNDICE DE FÓRMULAS

|   |    |
|---|----|
| <b>Fórmula 3.1:</b> Volume Weighted Average Price.....              | 20 |
| <b>Fórmula 3.2:</b> Time Weighted Average Price.....                | 20 |
| <b>Fórmula 3.3:</b> Modelo Linear Econométrico.....                 | 21 |
| <b>Fórmula 3.4:</b> Modelo de Moving Average.....                   | 21 |
| <b>Fórmula 3.5:</b> Moving Average simple.....                      | 21 |
| <b>Fórmula 3.6:</b> Moving Average simple.....                      | 21 |
| <b>Fórmula 3.7:</b> Moving Average con peso.....                    | 22 |
| <b>Fórmula 3.8:</b> Moving Average con peso.....                    | 22 |
| <b>Fórmula 3.9:</b> Moving Average exponencial.....                 | 22 |
| <b>Fórmula 3.10:</b> Coeficiente de Moving Average exponencial..... | 22 |
| <b>Fórmula 3.11:</b> Velocidad en Particle Swarm Optimization.....  | 26 |
| <b>Fórmula 3.12:</b> Posición en Particle Swarm Optimization.....   | 26 |
| <b>Fórmula 3.13:</b> Inercia exponencial para Particle Swarm.....   | 28 |
| <b>Fórmula 3.14:</b> Modelo de Red Bayesiana.....                   | 35 |
| <b>Fórmula 3.15:</b> Axiomas de Redes Bayesianas.....               | 35 |
| <b>Fórmula 3.16:</b> Inferencia de axiomas de Redes Bayesianas..... | 36 |
| <b>Fórmula 3.17:</b> Probabilidad condicional.....                  | 36 |
| <b>Fórmula 3.18:</b> Regla de cadena.....                           | 36 |
| <b>Fórmula 3.19:</b> Regla de Bayes.....                            | 36 |
| <b>Fórmula 4.1:</b> Restricciones al modelo de 4 bandas.....        | 54 |
| <b>Fórmula 4.2:</b> Función objetivo para Algorithmic Trading.....  | 55 |

**Fórmula 4.3:** Ratio de Sharpe.....56

**Fórmula 4.4:** Ratio de Sharpe para High Frecuency Trading.....56

**Fórmula 4.5:** Función objetivo extendida.....57

## 1. INTRODUCCIÓN

El comercio de valores (*stock trading*) es una práctica que lleva cientos de años, realizada en la actualidad en Bolsas de Valores alrededor de todo el mundo, en las cuales se transa diariamente una variedad enorme de activos financieros e instrumentos de deuda. Al mismo tiempo, el *stock trading* es un problema complejo de toma de decisiones, que implica múltiples variables y no siempre tiene una solución óptima, dado que las condiciones varían en el tiempo y son afectadas por factores internos y externos.

En los últimos años se ha comenzado a investigar e implementar sistemas de *stock trading* automáticos o semiautomáticos, que pueden realizar el análisis de las condiciones del mercado y tomar las decisiones necesarias para realizar los negocios requeridos. Tales sistemas han reportado desde el año 2008 ganancias significativas y constantes en Bolsas de Valores Extranjeras, tales como la New York Stock Exchange (NYSE) [Aldridge2010]. Tales sistemas se remontan al sistema “Black Box” de Morgan Stanley’s creado en el año 1985 [Pole2007], para la generación de señales de compra / venta basados en arbitraje estadístico.

El beneficio de emplear sistemas automatizados para la tarea de stock trading radica en el hecho de que las personas prestan demasiada atención a las señales (precio) e ignoran las probabilidades de transición, que es el aspecto que genera las señales. De este modo, al tener un sistema automatizado, este puede calcular las probabilidades de transición de los precios y actuar acorde [Massey2005], evitando problemas de reacción tardía o sobre-reacción a los cambios.

En la actualidad, en el mercado local el ingreso de *traders* automáticos ha sido relativamente bajo y no hay información al respecto, dado que quienes emplean tales mecanismos no pueden divulgar sus detalles por temor a la competencia, tal como ocurrió en un principio con el sistema “Black Box”. De todos modos, el mercado local se encuentra en condiciones de funcionar con *traders* automáticos de baja y alta frecuencia desde el año 2010, cuando la Bolsa de Santiago lanzó el sistema Telepregón HT, que permite el comercio de acciones de renta variable a una tasa máxima teórica de 3000 transacciones por segundo [BCS2010].

En esta Tesis lo que se busca es diseñar, implementar y probar un sistema de *trading* completamente automático que opere sobre el mercado accionario nacional, de modo que sea capaz de generar rentabilidades netas positivas a lo largo del tiempo. En particular, se desea crear un sistema que implemente trading de alta frecuencia (*High Frequency Trading*, en adelante HFT) mediante alguno de los modelos de predicción de precios expuestos en la Tesis.

## 1.1 Discusión Bibliográfica

Debido a la naturaleza del tema investigado, es posible encontrar información bibliográfica referente a los conceptos generales de Finanzas y Economía que son propios del ámbito de trabajo general del proyecto. De este modo, se puede contar con bibliografía que explique prácticas tales como el Arbitraje Estadístico [Pole2007], Volume Weighted Average Price (VWAP) y Time Weighted Average Price (TWAP) [Kendall2007] y Market Making [Aldridge2010]. Lo que no es fácil de encontrar es bibliografía referente a implementaciones en software o pseudocódigos relacionados al tema, dado que todos ellos forman parte de herramientas comerciales cuyas implementaciones son guardadas celosamente, ya que su divulgación implicaría pérdida de competitividad en un mercado altamente agresivo y competitivo. De todos modos se logran rescatar propuestas de diseños generales de sistemas HFT en textos tales como [Ketter2009], [Wang2009] y [Aldridge2010], los cuales serán la base del diseño propuesto en esta Tesis.

Sobre las técnicas de diseño y programación que se emplean en el proyecto, se recurre a bibliografía especializada en el tema. Sobre esto existen textos dedicados a cada técnica que se emplea para generar soluciones, tales como minería de datos, redes neuronales o procesamiento de eventos complejos, entre otros.

Para la estructuración de la metodología de investigación, se ha recurrido a [Hernandez2003] y [Hernandez2007], ya que ambos textos ofrecen una vista amplia de las metodologías aplicables en todos los campos de investigación, con ejemplos aplicados y estructuras bien definidas. Así, sirven como punto de partida para definir y preparar los diversos puntos que se incluyen en el proyecto, para que este se realice de manera correcta.



## 2. ANÁLISIS DE OBJETIVOS Y METODOLOGÍA

### 2.1 Objetivo General

“Definir, diseñar, implementar y probar un sistema de trading automático para el mercado accionario nacional, que sea capaz de ajustarse a condiciones cambiantes y que genere rentabilidades positivas a lo largo del tiempo”.

El tema de la Tesis de Magíster corresponde a la aplicación de herramientas informáticas avanzadas a un problema de tipo NP-Completo, donde se quiere optimizar la rentabilidad de operaciones de compra y venta de acciones. De este modo, el objetivo es lograr crear una implementación de un sistema de trading automático que sea capaz de generar rentabilidades positivas para un conjunto de datos reales del mercado accionario nacional, bajo una modalidad completamente automática, donde no exista intervención de un operador humano en la toma de decisiones y ejecución de operaciones.

### 2.2 Objetivos Específicos

1. Revisión y entendimiento de técnicas de trading actuales que puedan funcionar bajo la modalidad automática de HFT, indicando cuales técnicas informáticas se pueden aplicar.
2. Diseño de un sistema automático de trading, en modalidad HFT, indicando las restricciones sobre los datos e instrumentos financieros incluidos en el estudio.
3. Generación de algoritmos y construcción de un sistema que implemente el diseño propuesto y los algoritmos generados.
4. Documentar procesos de carga de datos a los módulos de predicción de precios.
5. Realizar pruebas individuales de los algoritmos implementados, revisando el retorno neto teórico (rentabilidad) que pueden generar aplicados sobre el último día, mes y semestre de datos reales del mercado.

6. Determinar cuál de las variantes del sistema implementado se comporta mejor, usando los retornos netos como base de comparación y aplicando otros criterios según se estime necesario.

### 2.3 Justificación de los objetivos

El objetivo general de la Tesis de Magíster tiene una relación profunda con el hecho de que no hay estudios de trading automático sobre el mercado nacional chileno, dado que es una práctica muy reciente. Del mismo modo, aun cuando exista información básica sobre los componentes que debe tener un sistema de HFT, no hay datos de rendimiento de tales sistemas que permita comparar un tipo de diseño sobre otro.

Realizar un trabajo de este tipo permite ver la aplicación práctica de un conjunto de áreas de las ciencias de la información en pos de resolver un problema complejo que no tiene una única solución. El problema en cuestión es maximizar el valor de una función objetivo, sin conocer a priori como la elección de los valores para las variables de decisión afectarán el resultado, ya que este depende de variables externas desconocidas y las consecuencias de la selección de los valores sólo se conocerán tiempo después. De este modo, el objetivo es probar que es posible modelar el problema de modo que permita cierto grado de predictibilidad.

Sobre los objetivos específicos, tenemos lo siguiente:

1. Lo primero es ver las técnicas de trading que puedan emplearse dentro del trabajo de Tesis. En base a las técnicas que tengan potencial para aplicarse de forma automática, se podrá establecer el alcance del diseño del sistema HFT que se desea construir y probar. El grado de cumplimiento de este objetivo determinará el éxito de los siguientes objetivos y permitirá definir futuras áreas de estudio en caso de querer ahondar en el tema.
2. Una vez identificadas las técnicas de trading que permitan automatización, es necesario contar con un diseño de un sistema automático que sea flexible, extensible y con facilidad para realizar pruebas de laboratorio. De este modo se requiere un diseño que sea capaz de abarcar las técnicas seleccionadas por cumplimiento del objetivo anterior. Además, debido al hecho de que incluso el mercado bursátil nacional incluye una gran cantidad de mercados primarios y secundarios, tratar de construir un sistema que opere automáticamente sobre todos ellos sería demasiado complejo y dificultaría las comparaciones. De este modo, es necesario definir sobre cuales partes del mercado se trabajará y justificar las decisiones. Del mismo modo, como cada mercado se compone de una gran cantidad de instrumentos financieros con sus propias características, es posible que se quiera trabajar sobre un subconjunto que cumpla determinadas características.
3. Una vez que se tenga el diseño general, es necesario definir los algoritmos que controlan el funcionamiento de los módulos principales. Para alcanzar este objetivo, no basta con definir algoritmos nuevos, sino que además es necesario revisar si existen algoritmos existentes, tales como metaheurísticas o minería de datos,

que permitan resolver partes del sistema. Luego se debe proceder a construir un sistema con las mismas características enunciadas como objetivo de diseño. De este modo, se debe construir un sistema que permita fácilmente realizar pruebas de laboratorio. Es deseable que además el sistema sea capaz de conectarse a interfaces existentes para comunicarse con el mercado accionario.

4. Debido a que muchas técnicas de trading requieren grandes cantidades de datos históricos, es posible que sea necesario diseñar e implementar procesos ETL o similares. En caso de realizarse tales procesos, es necesario documentar como se realiza el proceso.
5. Dado que se quiere diseñar e implementar más de una variante de técnica de trading automático o en su defecto una técnica que permita múltiples parámetros, es necesario realizar pruebas controladas de laboratorio, y de ser posible, pruebas de campo. En este caso, el principal resultado de las pruebas será el retorno neto (rentabilidad) que se obtiene al aplicar una determinada técnica de trading automático.
6. El último objetivo es la culminación de la Tesis de Magíster. Es necesario obtener una conclusión de todo el trabajo realizado, indicando cual es la mejor técnica implementada o el mejor conjunto de parámetros basándose en los resultados de las pruebas de laboratorio.

## 2.4 Metodología de la Investigación

Tomando como base las 4 categorías de investigación explicadas en [Hernandez2003], siendo estas las siguientes:

- Exploratoria: se aplica cuando se quiere examinar un tema poco estudiado o que no ha sido abordado antes. Sirve para aumentar el grado de familiaridad con fenómenos relativamente desconocidos.
- Descriptiva: se usa cuando el propósito es describir situaciones y eventos, especificando las propiedades importantes de los fenómenos estudiados. Sirve para medir con la mayor precisión posible.
- Correlacional: empleado cuando se quiere establecer el grado de relación entre dos o más conceptos o variables, en un contexto particular. Sirve para saber cómo se comporta una variable o concepto al conocer el comportamiento de otra variable o concepto relacionado.
- Explicativa: se centra el interés de la investigación en conocer por qué ocurre un fenómeno y bajo cuales condiciones se da. Es una investigación estructurada que implica las otras 3 categorías mencionadas (exploración, descripción y correlación).

De este modo, considerando la cantidad y contenido de la literatura y trabajos académicos existentes sobre el tema, la categoría principal de la investigación será de tipo “Exploratoria”, dado que sólo existen piezas dispersas de literatura sobre el tema que se quiere investigar y para el cual no se tiene una base de comparación empírica para comparar resultados.

Al ser una investigación exploratoria, la finalidad del trabajo realizado es aumentar la familiaridad con un fenómeno relativamente desconocido, lo que permite tener una base para futuras investigaciones más exhaustivas. Así, aun cuando exista literatura sobre metaheurísticas o minería de datos, no hay claridad sobre cómo se aplican dentro del contexto del trading automatizado, dado que el mismo trading automatizado no cuenta con bases sólidas en el área de la Informática, sino más bien por el área de la Economía y la Estadística.

Siguiendo los lineamientos postulados en [Hernandez2007], existen 5 tipos de proyectos, cada uno con su metodología. Estos corresponden a:

- Intervención: la fundamentación es una acción que se ejerce sobre un objeto rigurosamente determinado.
- Evaluación: se orientan hacia un conocimiento relacionado con atributos de eficacia, calidad, eficiencia o impacto.
- Desarrollo tecnológico: se orienta hacia la obtención de productos tangibles.
- Investigación: incluye la producción u obtención de algo, su evaluación, y su empleo como parte de una intervención. Tiene una intención cognoscitiva sobre cualquier otro propósito.
- Investigación-acción: coexisten en estrecho vínculo el afán cognoscitivo y el propósito de conseguir efectos objetivos y medibles.

Considerando tales tipos de proyectos, el que más se ajusta a los objetivos planteados es el de Desarrollo Tecnológico. Esto implica que se debe evaluar las propiedades del producto tecnológico que se quiere desarrollar, tales como los avances que se han realizado dentro del área de interés del producto, evaluación de sus partes (p.ej.: modelo predictivo de los algoritmos desarrollados) de modo que permita determinar cómo y con cuáles características se generará el producto final.

### 3. TRADING ALGORÍTMICO Y DE ALTA FRECUENCIA

El *trading* (comerciar) es el intercambio de propiedad de un bien, producto o servicio desde una persona o entidad para obtener algo a cambio por parte del comprador. Para efectos prácticos, se entenderá como *trading* la práctica realizada por Corredores de Bolsa o sus clientes (personas naturales, sociedades de inversiones, instituciones financieras, etc.) donde se intercambian instrumentos financieros en los mercados de valores. Esto se basa en el principio de oferta y demanda de los instrumentos transados, permitiendo que estos varíen sus precios y obteniendo una ganancia (o pérdida) generada por la diferencia entre el precio de compra original y el precio de la venta final.

Lo que se entiende por *High Frequency Trading* (HFT) es una forma de operar en los mercados de valores donde se dan una serie de condiciones especiales [Aldridge2008]:

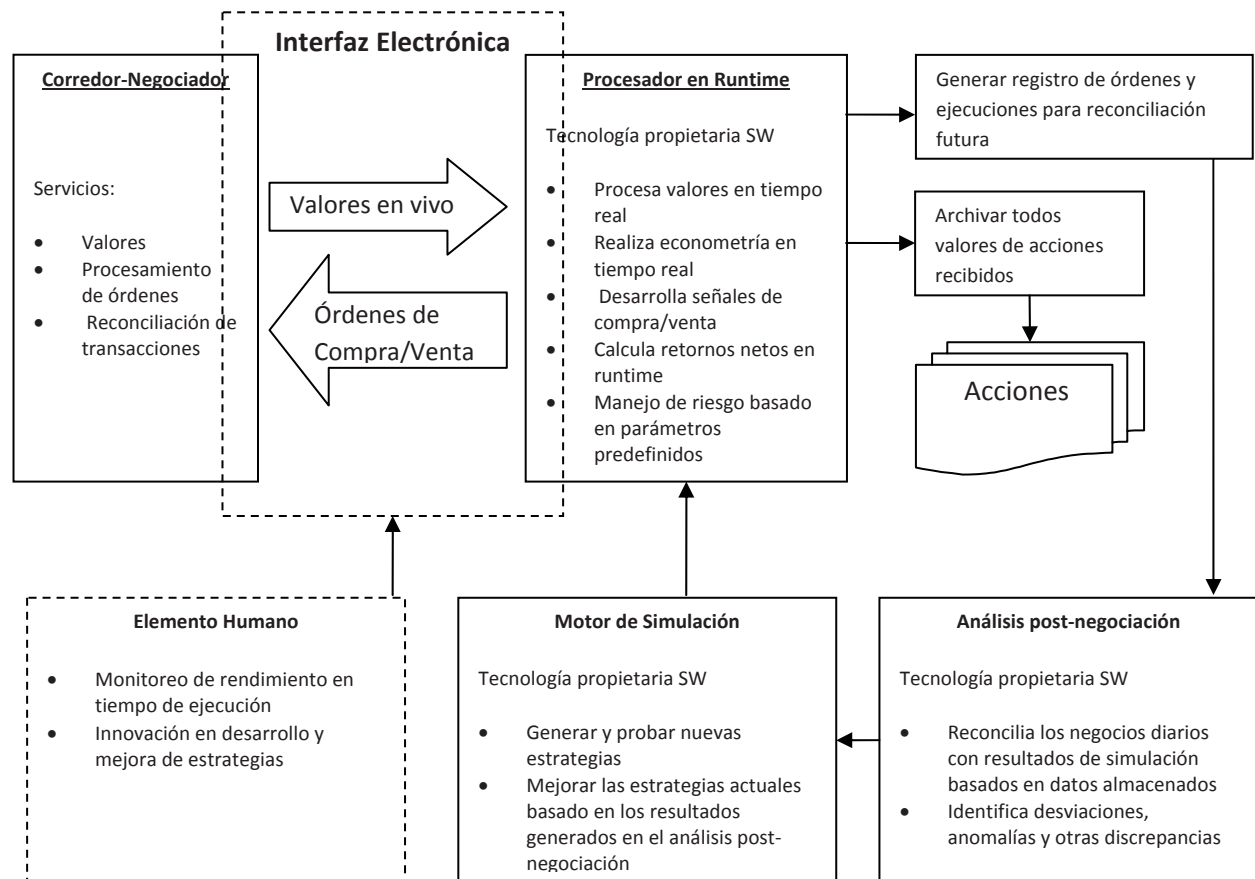
- Existe un rápido intercambio de capital
- Se realiza un alto número de operaciones
- Generalmente se obtiene una baja ganancia por operación
- No se acumulan posiciones de instrumentos financieros de un día de operación para otro, o se evita
- Se realiza mediante un sistema informático

La definición de HFT en sí misma no indica si el sistema que lo realiza es automático, semiautomático u operado por usuarios.

En cambio, lo que se entiende por *Automatic Trading* varía desde sistemas que apoyan el ingreso de órdenes de compra/venta al mercado, como sistemas capaces de realizar el ingreso de órdenes de forma automática, sin necesidad de un operador humano, pero que pueden mantener posiciones de instrumentos financieros de un día para otro. No existe un consenso único sobre la definición de *Automatic Trading* y sus alcances como sistema.

No hay una fórmula única para definir un sistema de HFT o de *Automatic Trading*. Por ejemplo, en [Wang2009] se define que un sistema de *Algorithmic Trading* (de aquí en adelante AT) corresponde a: “elementos en la toma de decisiones e inversiones financieras siendo ejecutados por un algoritmo a través de computadores y redes de comunicación de forma electrónica”. Las estrategias de inversión pueden ser predefinidas o adaptativas. Estas estrategias de inversión pueden apoyarse tanto en conocimientos de Economía, Estadística, Inteligencia Artificial, Metaheurísticas, etc., sobre las cuales no existe mucha información, dado que son lo que constituyen las ventajas competitivas de herramientas comerciales sobre el resto del mercado.

Según [Aldridge2010] un sistema HFT tiene un esquema general integrado con el mercado objetivo. De este modo, propone un proceso con funciones claramente separadas.



**Figura 3.1** Proceso típico de HFT, según Aldridge, donde se separa el sistema en múltiples subsistemas especializados

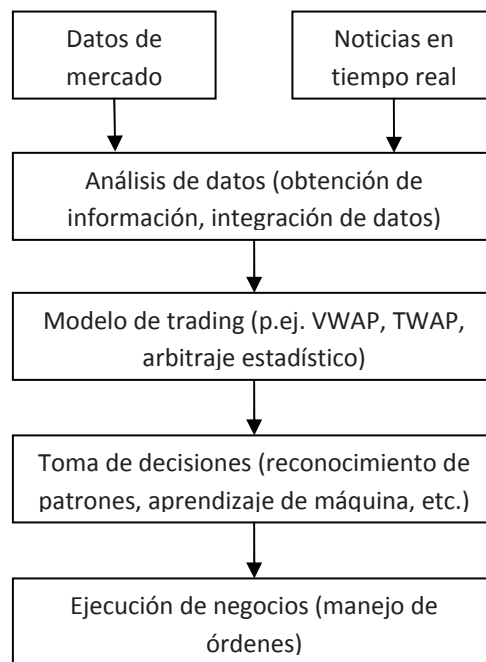
Al revisar la figura 3.1, Aldridge propone un esquema general para todos los sistemas de HFT donde el núcleo del sistema se encarga de las funciones de:

- Recibir, evaluar y archivar valor bursátiles entrantes (instrumento, cantidad, precio)
- Realizar análisis econométrico en tiempo de ejecución
- Implementar administración de portafolio en tiempo de ejecución
- Iniciar y transmitir señales de trading de compra y venta
- Escuchar por ejecuciones de órdenes y recibir confirmaciones
- Calcular Ganancias y Pérdidas (retorno neto) en tiempo de ejecución



- Manejar dinámicamente el riesgo basándose en las asignaciones actuales de portafolio y las condiciones de mercado

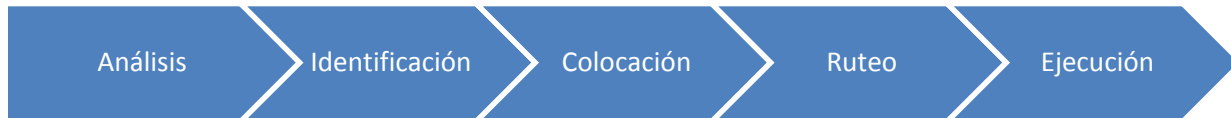
De forma similar, Wang, Dong y Deng en [Wang2009] proponen un proceso secuencial para lograr un sistema HFT, basado en 4 pasos.



**Figura 3.2** Proceso de Trading Algorítmico según Wang, Dong y Deng, donde el sistema es un flujo de tareas secuenciales

Al revisar la figura 3.2, se inicia el proceso al analizar los datos que provienen desde el mercado como desde otras fuentes externas, tales como noticias relevantes.

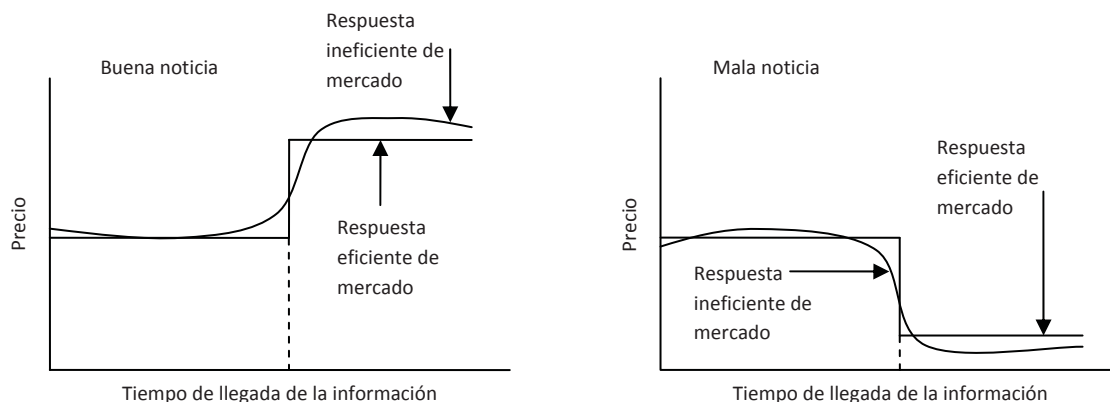
De este modo, no hay una fórmula única para producir un sistema de HFT. En lo que sí se encuentra coincidencia es que para lograr un sistema HFT efectivo, hay que tener en cuenta una serie de procesos comunes a cualquier sistema. Una versión simplificada de estos elementos básicos se puede encontrar mencionada por Kim Kendall [Kendall2007].



**Figura 3.3** Elementos básicos de un sistema HFT, según Kendall

Los puntos en que coinciden varios autores que han explorado los sistemas HFT y AT es que tales sistemas deben contar con (figura 3.3):

- Un sistema de análisis de captura de oportunidades de negocio en tiempo real.
- Identificación de ineficiencias del mercado en base a precio, volumen, liquidez, etc. para aprovecharlas en busca de un beneficio. Esto se basa en el hecho de que los mercados presentan ineficiencias en la incorporación de la información con respecto a un instrumento financiero (figura 3.4).



**Figura 3.4** Incorporación de la información en mercados ineficientes. El precio de un instrumento financiero se ajusta inmediatamente a su nuevo nivel apenas una noticia se incorpora al mercado, mientras que en mercados con respuestas ineficientes el precio se ajusta antes de conocer la noticia (filtración de información) y luego sobre-reacciona cuando dicha noticia se hace pública. Esto permite generar ganancias constantes al identificar estas condiciones.

- Colocación de órdenes de compra/venta en base a los parámetros actuales que se manejan de capital disponible (el dinero que el sistema tiene para realizar compras) y de custodia disponible (la cantidad de acciones u otro instrumento financiero que el sistema tiene para realizar ventas).

- Ruteo de órdenes generadas, ya sea mediante DMA (Direct Market Access, canales de comunicaciones dedicados que permiten generar directamente ofertas en el sistema electrónico de la Bolsa de Valores destino), protocolo FIX (Financial eXchange Protocol, protocolo común usado por la mayoría de Bolsas de Valores e instituciones financieras del mundo para intercambio de órdenes de compra/venta e información bursátil) o por un agente ruteador externo (Corredor de Bolsa u otra institución financiera).
- Monitoreo de la ejecución de órdenes enviadas, para retroalimentar el sistema o para generar anulaciones o modificaciones si las condiciones del mercado cambian.

Los componentes que presentan la mayor complejidad corresponden al análisis de oportunidades en tiempo real y la búsqueda de ineficiencias del mercado. La literatura disponible menciona métodos tales como:

- Métodos basados en reglas, tales como *Arbitraje Estadístico* [Pole2007]. En este caso, se emplea una serie de reglas basadas en el comportamiento reciente que ha tenido un instrumento financiero y se actúa acorde al resultado de aplicar dichas reglas.
- Métodos basados en modelos estadísticos y matemáticos, tales como *Volume Weighted Average Price*, *Time Weighted Average Price* y *Moving Averages* [Wang2009, Aldridge2010]. Para estos casos, se emplea un modelo matemático o estadístico que requiere una serie de parámetros que controlan su comportamiento. La selección de los parámetros de configuración se deja en manos del operador manual a cargo de transar en el mercado.
- Métodos que combinan modelos estadísticos y matemáticos junto con técnicas de optimización basadas en metaheurísticas [Pardo2008]. Se propone usar metaheurísticas para afinar automáticamente los parámetros de algoritmos conocidos, para obtener valores óptimos para las condiciones actuales de los mercados.
- Métodos basados en aprendizaje de máquina, minería de datos y procesamiento de eventos complejos [Bigus1996, Pardo2008, Aldridge2010]. Debido a la gran cantidad de información que fluye desde y hacia un mercado de valores cualquiera, su rápida asimilación por parte de un operador humano se vuelve una tarea cada día más ardua, por lo cual surgen sistemas que son capaces de detectar patrones ocultos en las variaciones de precios y las relaciones entre instrumentos financiero u otros indicadores económicos, y además pueden incorporar un componente de interpretación del “sentimiento” o “sensación” del mercado, mediante procesamiento de noticias en lenguaje natural (p.ej.: SuperX Plus de Deutsche Bank [AT2011]).

Considerando los objetivos planteados en el capítulo 2, es necesario revisar tanto los algoritmos conocidos de AT y HFT que permiten optimización, los métodos de optimización que se pueden aplicar y por último las técnicas de minería de datos que se pueden aplicar sobre un conjunto de datos bursátiles para extraer un modelo predictivo. Esto es necesario para poder generar un diseño de un sistema que sea capaz de satisfacer los requerimientos particulares de tales algoritmos y modelos de *trading*.

### 3.1 Métodos estadísticos de AT y HFT

Algunos de los algoritmos más populares de trading basados en métodos estadísticos o matemáticos [AMD2007, Wang2009] son:

- **Volume Weighted Average Price (VWAP):** se define como el ratio del volumen de transacciones valorizadas contra el volumen del instrumento sobre el horizonte de negociación. Es común evaluar el rendimiento de los traders por su habilidad de ejecutar órdenes de compra/venta a precios mejores que el precio VWAP sobre el horizonte de negociación. Su ventaja radica en su simplicidad computacional, especialmente en mercados donde obtener un nivel detallado de datos es difícil o demasiado caro. El VWAP para un instrumento  $i$  en un día  $T$  se calcula de la siguiente manera:

$$VWAP_i = \frac{\sum_t v_{it} p_{it}}{\sum_t v_{it}}, (t \in T) \quad (3.1)$$

Donde  $v_{it}$  es el volumen del instrumento  $i$  negociado en el momento  $t$  y  $p_{it}$  es el precio de mercado del instrumento  $i$  en el momento  $t$ . Uno de los usos de VWAP es minimizar los costos de las transacciones y los impactos de mercado. También se puede usar como benchmark para verificar la eficacia de otros algoritmos y estrategias de trading.

En [Humphery2011] se propone una modificación al modelo, llamado DVWAP (Dynamic VWAP) que permite incorporar las transacciones intra-day (transacciones realizadas durante el mismo día de ejecución). Esto permite aplicar el modelo a un escenario más realista de mercado, donde las noticias que van llegando afectan al precio de los instrumentos (como se mostró en la figura 3.4).

- **Time Weighted Average Price (TWAP):** es el precio promedio de un instrumento financiero sobre el tiempo específico, en el cual se trata de ejecutar una orden al precio TWAP o mejor. Se emplea para ejecutar órdenes en un tiempo específico, para mantener el precio cercano a lo que refleja el mercado en ese momento. El TWAP de un instrumento  $i$  en un período  $T$  se calcula como:

$$TWAP_i = \frac{1}{T} \sum_{t=1}^T p_{it}, (t \in T) \quad (3.2)$$

Donde  $p_{it}$  es el precio de mercado del instrumento  $i$  en el momento  $t$ . Al igual que VWAP, TWAP se puede emplear como benchmark para verificar la eficacia de otros algoritmos y modelos.

Otros tipos de algoritmos incluyen las variantes de los modelos lineales econométricos presentados por [Aldridge2010]. Estos modelos intentan predecir el comportamiento de variables aleatorias como una combinación de otras variables aleatorias, tanto contemporáneas como atrasadas, con distribuciones bien definidas. Tales modelos lineales se pueden expresar de la siguiente manera:

$$y_t = \alpha + \sum_{i=0}^{\infty} \beta_i x_{t-i} + \sum_{j=0}^{\infty} \gamma_j z_{t-j} + \dots + \varepsilon_t \quad (3.3)$$

Donde  $\{y_t\}$  es la serie temporal de una variable aleatoria sobre la cual se quiere realizar un pronóstico,  $\{x_t\}$  y  $\{z_t\}$  son factores significativos para predecir el valor de  $\{y_t\}$ ,  $\alpha$ ,  $\beta$ , y  $\gamma$  son factores por determinar, y  $\varepsilon_t$  es el error remanente.

- **Moving Averages (MA):** MA es un modelo para predecir movimientos futuros en el precio de un instrumento financiero, enfocándose en como los datos futuros reaccionarán a los cambios en los datos pasados. Para generar el modelo de  $MA(q)$ , con  $q$  retrasos, se utiliza:

$$y_t = c_0 + a_t - \theta_1 a_{t-1} - \dots - \theta_q a_{t-q} \quad (3.4)$$

Donde  $c_0$  es el intercepto,  $\theta_l$  es el coeficiente perteneciente al retraso  $l$ , y  $a_t$  es el componente inesperado del retorno en el retraso  $l$ . Existen varias formas de estimar el MA, algunas de ellas incluyen:

- MA Simple (SMA): es el promedio sin peso de los  $n$  precios previos. Esto puede ser los  $n$  días previos u otra medida de tiempo. También se puede calcular en base al SMA del período anterior, simplificando su cálculo a nivel computacional:

$$SMA_t = \frac{\sum_{i=0}^n p_i}{t} \quad (3.5)$$

$$SMA_t = SMA_{t-1} - \frac{p_{t-n}}{n} + \frac{p_t}{n} \quad (3.6)$$

- MA Acumulativo (CMA): es un promedio móvil acumulativo, donde se consideran todos los precios hasta el instante actual. Su fórmula es similar a SMA, pero partiendo desde el primer precio de mercado registrado para un instrumento. No tiene aplicación conocida en estrategias de trading.

- MA con peso (WMA): es un promedio que tiene factores de multiplicación para dar diferentes pesos a diferentes precios dentro de la misma ventana de MA (convolución de puntos de datos con una función fija de peso). En trading, se asigna un peso decreciente desde  $n$  hasta 1 a cada precio en la ventana de evaluación:

$$WMA_t = \frac{\sum_{i=0}^n (n-i)p_i}{\sum_{i=0}^n n-i} \quad (3.7)$$

$$WMA_t = \frac{\sum_{i=0}^n (n-i)p_i}{\left(\frac{n(n+1)}{2}\right)} \quad (3.8)$$

WMA provee una función de suavizado de la curva de predicción distinta de MA. En algunos casos se usan en conjunto, o cuando los precios de días anteriores no afectan el valor en mayor medida el precio actual de un instrumento.

- MA Exponencial (EMA): también llamado EWMA (MA con Peso Exponencial) es una versión parecida a WMA, donde el peso no es una función lineal sino exponencial. El peso asignado a cada precio de mercado disminuye exponencialmente, nunca llegando a cero. De esta forma, para una serie  $y$ , el EMA se calcula recursivamente como:

$$y_0 = \beta p_0 \quad (3.9)$$

$$\text{para } t > 0, y_t = \alpha p_t + (1-\alpha)y_{t-1}$$

Donde  $\alpha$  es el coeficiente de peso decreciente, un valor constante entre 0 y 1. Un valor alto del coeficiente disminuye los precios antiguos más rápidamente. Alternativamente,  $\alpha$  puede ser expresado en términos de  $n$  períodos de tiempo:

$$\alpha = \frac{2}{n+1} \quad (3.10)$$

Existen otros métodos basados en modelos lineales, pero que no se pretenden explorar dentro del trabajo de implementación. Entre estos se pueden mencionar los siguientes:

- **MA Autoregresivo (ARMA):** el modelo ARMA incluye el componente esperado de las variables atrasadas dentro del cálculo de MA.

- **Cointegración:** técnica para construcción de portafolio de inversiones y manejo de riesgo. Busca la relación entre 2 instrumentos financieros mediante un sistema de ecuaciones lineales.

Los modelos mencionados se combinan generalmente entre sí para obtener sistemas de generación de señales de compra/venta. Existen varias estrategias de combinación conocidas, algunas más exitosas que otras. Se revisarán algunas de estas técnicas al momento de implementar la parte empírica del trabajo de Tesis. Generalmente estas combinaciones dependen de cómo se configuren los parámetros temporales de las distintas variantes de MA, en conjunto con condiciones de *stop-loss* y *stop-win* (estrategias con bandas, donde *stop-loss* es la banda inferior a la que se puede llegar antes de forzar una operación de liquidación, mientras que *stop-win* es la banda superior que inicia automáticamente una liquidación de custodia).



### 3.2 Modelos Metaheurísticos

Como se expuso anteriormente, existen varios modelos y algoritmos conocidos para realizar trading. Tales algoritmos generalmente son aplicados manualmente por un operador humano para determinar cuando conviene comprar, vender o mantener la posición actual. Robert Pardo [Pardo2008] expone que es posible, para una combinación dada de estrategias, aplicar optimización para determinar un conjunto de parámetros que genere mayores ganancias.

Tal postulado no viene sin problemas asociados. El mayor problema conocido es que tales optimizaciones pueden producir sobredecuación del algoritmo a los datos empleados. Esto trae como consecuencia que, en el mejor de los casos, el algoritmo resultante no generará las ganancias esperadas, mientras que en el peor de los casos el algoritmo producirá pérdidas constantes. Una forma de entender el concepto de sobredecuación es pensar en un modelo estadístico que describe el error aleatorio o ruido en vez de describir las relaciones entre las variables.

El mecanismo propuesto por Pardo para obtener tal optimización es mediante metaheurísticas. Al respecto, tanto AT como HFT pueden entenderse como problemas complejos de optimización.

Esta clase de problemas son llamados de clase NP, donde la sigla “NP” significa “*nondeterministic polynomial time*” (tiempo polinomial no determinístico). La clase NP es la clase de problemas en la cual una solución puede ser verificada por un algoritmo de tiempo polinomial, pero dada la dificultad de los problemas, no existe un algoritmo que pueda generar soluciones en tiempo polinomial. Esto implica que los algoritmos convencionales, aplicados a esta clase de problemas, presentan tiempos de ejecución que crecen exponencialmente al crecer el tamaño del problema.

Hasta el año 1971, no existía demostración de un problema de esta clase. Ese año, Stephen Cook demostró el primer problema práctico NP-Completo [Cook1971]. El año 1972, Richard Karp tomó la idea de Cook y demostró una serie de 21 problemas de clase NP-Completo [Karp1972].

Para el caso específico de *Algorithmic Trading* y *High Frequency Trading*, ambos problemas pueden ser categorizados como clase NP [Fortnow2004], al ser problemas de negociación de instrumentos financieros en mercados con condiciones variables en el tiempo. Como ambos problemas se basan en la maximización de los retornos netos, según Chang y Johnson [Chang2008], se pueden clasificar como NP-Completo, incluso en las versiones que realizan simulaciones de mercado offline.

Una de las formas de abordar un problema de clase NP es utilizando una metaheurística, que corresponde a un algoritmo aproximado que combina métodos heurísticos básicos en un marco de trabajo más elevado, donde se explora de forma efectiva y eficiente un espacio de búsqueda de soluciones [Blum2003]. De este modo, mediante una función objetivo que guía el proceso de búsqueda, se realiza una exploración eficiente de posibles soluciones en búsqueda de una o más soluciones cercanas al óptimo.

Los algoritmos metaheurísticos tienen distintas clasificaciones. Por ejemplo, Blum [Blum2003] presenta por lo menos 5 categorizaciones distintas en base a las características que presenta cada algoritmo. Tales categorías corresponden a:

- Inspirados en la naturaleza vs no inspirados en la naturaleza: algoritmos que simulan o copian partes de procesos que se dan de forma natural.
- Basados en población vs búsqueda de punto único: algunos algoritmos manejan múltiples soluciones temporales, mientras otros sólo manejan una única solución en un momento dado.
- Función objetivo dinámica vs estática: algunos algoritmos mantienen fija la función objetivo durante toda la ejecución, mientras que otros son capaces de modificarla, para escapar de óptimos locales.
- Una estructura de vecindad vs varias: la mayoría de algoritmos maneja una única estructura de vecindad, donde la topología de aptitud del entorno no cambia. Pueden usar un set de estructuras, intercambiándolas para diversificar la búsqueda.
- Uso de memoria vs sin memoria: algunos algoritmos pueden usar el historial de búsqueda, mientras otros no recuerdan las soluciones previas.

Existe una gran variedad de algoritmos metaheurísticos, algunos con mayor afinidad para ciertos tipos de problemas que otros, como por ejemplo problemas con variables binarias, discretas o continuas. Algunos algoritmos sólo se pueden aplicar sobre un tipo de variable, o se deben hacer adecuaciones como aplicar funciones de conversión (p.ej.: función sigmooidal para convertir resultados discretos a binarios). En particular, se prestará enfoque a uno de los algoritmos existentes llamado Optimización de Enjambre de Partículas (en adelante PSO, *Particle Swarm Optimization*).

### 3.2.1 Particle Swarm Optimization

El algoritmo de PSO fue introducido por Kennedy y Eberhart en 1995 [Kennedy1995], tratando de imitar el comportamiento social de las bandadas de pájaros o escuelas de peces y sus mecanismos de comunicación para resolver problemas de optimización. En vez de basarse en un precepto de sobrevivencia del más fuerte, PSO se basa en la cooperación de los individuos.

PSO se define como un algoritmo metaheurístico que optimiza un problema mejorando iterativamente una población de soluciones candidatas, llamadas partículas, moviéndolas por el espacio de soluciones usando una fórmula basada en la posición y velocidad de la partícula. El movimiento de cada partícula es influenciado por su mejor solución local conocida y también es guiada a la mejor solución global conocida. Con esto se espera que el enjambre se mueva colectivamente hacia la mejor solución en el espacio de búsqueda.

En la versión básica de PSO, la velocidad y posición de las partículas se calcula del siguiente modo:

$$v_i^{k+1} = c_0 v_i^k + c_1 r_{1,i}^k (x_i^{k,best} - x_i^k) + c_2 r_{2,i}^k (x_{i,swarm}^{k,best} - x_i^k) \quad (3.11)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (3.12)$$

Donde:

- $x_i^k$ : posición de la i-ésima partícula en la iteración k
- $v_i^k$ : velocidad de la i-ésima partícula en la iteración k
- $c_0$ : factor de inercia, valor entre 0 y 1
- $c_1$ : factor de aceleración local (componente cognitiva del individuo)
- $c_2$ : factor de aceleración global (componente social del enjambre)
- $r_{1,i}^k, r_{2,i}^k$ : números aleatorios con distribución uniforme entre 0 y 1
- $x_i^{k,best}$ : mejor posición previa de la i-ésima partícula
- $x_{i,swarm}^{k,best}$ : mejor posición previa del vecindario de la i-ésima partícula

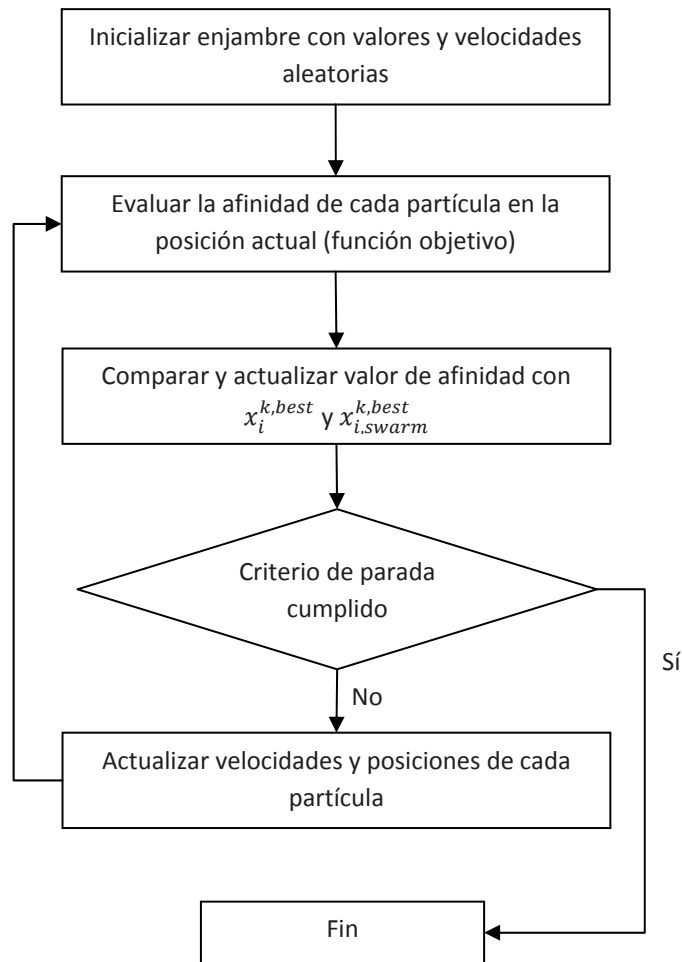


Figura 3.5 Diagrama de flujo del algoritmo básico de PSO.

En la figura 3.5 se muestra como se define el algoritmo básico de PSO. Al analizar el algoritmo y sus fórmulas asociadas, queda manifiesto que es un algoritmo perteneciente a métodos *estocástico* de búsqueda, esto significa que emplea una componente aleatoria en su cálculo.

| Parámetro        | Valor                 |
|------------------|-----------------------|
| N° de Partículas | $10 + \sqrt{\dim(X)}$ |
| $c_0$            | $1/(2\ln(2))$         |
| $c_1$            | $0,5 + \ln(2)$        |
| $c_2$            | $0,5 + \ln(2)$        |

Tabla 3.1 Valores típicos para PSO definidos en [Jabri2011]. La función  $\dim(X)$  es la dimensión del problema de optimización, esto corresponde a la cantidad de variables de optimización,

Sobre la selección de los parámetros  $c_0$ ,  $c_1$  y  $c_2$  existen diversas formulaciones. Una de las opciones es utilizar los valores sugeridos en [Jabri2011], que se muestra en la tabla 3.1.

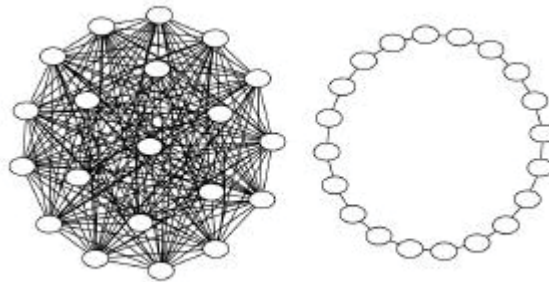
Otras formas de determinar los parámetros incluyen funciones que modifican tales parámetros durante la ejecución del algoritmo. Un ejemplo de esto se expone en [Fikret2009], donde se parte del hecho de que el valor del parámetro de inercia  $c_0$  influye en la diversificación (exploración del espacio de búsqueda) e intensificación (explotación del espacio de búsqueda). Valores altos del parámetro de inercia favorecen la diversificación, mientras que valores bajos favorecen la intensificación de soluciones locales. De este modo se define una función exponencial del parámetro de inercia definida por:

$$c_0 = c_{0,end} + (c_{0,start} - c_{0,end})e^{\frac{k\alpha}{k_{max}}} \quad (3.13)$$

Donde  $c_{0,start}$  es la inercia inicial,  $c_{0,end}$  es la inercia final,  $k$  representa la iteración actual,  $k_{max}$  es la cantidad máxima de iteraciones que se realizará y  $\alpha$  es una constante de gradiente.

Otras variantes del cálculo de  $c_0$  incluyen descenso lineal del parámetro de inercia o una función estocástica asociada a la inercia.

Sobre otras variantes del algoritmo de PSO, se puede mencionar las distintas formas que puede adoptar la topología de vecindad para determinar cuáles grupos de partículas se afectan entre sí mismas. Esto responde a una limitación del algoritmo básico, que puede quedar atrapado prematuramente en un óptimo local al considerar que todas las partículas interactúan en sí (topología *global best*, *gbest*). De esta forma se definen topologías distintas que pueden permitir evitar en cierta medida este problema.



**Figura 3.6** Topologías para PSO definidas originalmente por Kennedy y Eberhart. Se muestran las topologías *global best* y *local best*

En la figura 3.6 se pueden observar las topologías “*gbest*” y “*lbest*”. Estas corresponden a las topologías originalmente definidas por Kennedy y Eberhart que puede tomar el algoritmo.

Para *global best*, todas las partículas son vecinas de las demás, por lo que se usará la mejor partícula del enjambre en la componente social de la ecuación de velocidad. Todas las partículas serán atraídas simultáneamente por la mejor solución encontrada hasta el momento.

Para *local best*, sólo una cantidad finita de partículas puede afectar la velocidad de una partícula dada. Esto produce que el enjambre converja más lentamente hacia una mejor solución, pero permite que se busque un óptimo global a mayor distancia.

Finalmente, existen otras variantes de PSO donde se realiza un algoritmo híbrido con otros algoritmos metaheurísticos. Ejemplos de esto se mencionan en [Fikret2009], donde se exponen combinaciones de PSO con Simulated Annealing (SA, una metaheurística de trayectoria que simula el proceso de recalentado y enfriamiento controlado de un metal) o con Tabú Search (TB, un algoritmo de trayectoria con memoria donde se evita volver a soluciones encontradas recientemente).

Para el problema específico de HFT y AT, se aplica el algoritmo PSO sobre la optimización de los parámetros de una estrategia de *trading* basada en *Moving Averages* (ver sección 3.1) de 2 o más bandas. En este caso, se optimizan los parámetros temporales de los *Moving Averages* implicados en la estrategia contra una función objetivo que incluya lo siguiente:

- Obtener el mayor retorno neto (ganancias)
- Obtener el mayor beneficio por operación
- Obtener el mayor porcentaje de operaciones ganadoras, o que la estrategia presente un ratio financiero específico más alto.

De este modo la función objetivo que se aplica al algoritmo PSO mide y clasifica la calidad de la estrategia de trading que se aplica en el sistema de AT o HFT. Así, el éxito de la aplicación de PSO a un problema de HFT y AT depende en su mayoría de cómo se plantea la función objetivo, para un mismo modelo de *trading*.

### 3.3 Modelos basados en Aprendizaje de Máquina

Son los modelos más recientes y sobre los cuales existe la menor cantidad de información sobre implementaciones. Existen sólo algunas propuestas como en [Aldridge2010], donde se menciona el empleo de redes neuronales para generar un modelo predictivo basado en datos históricos de los instrumentos financieros.

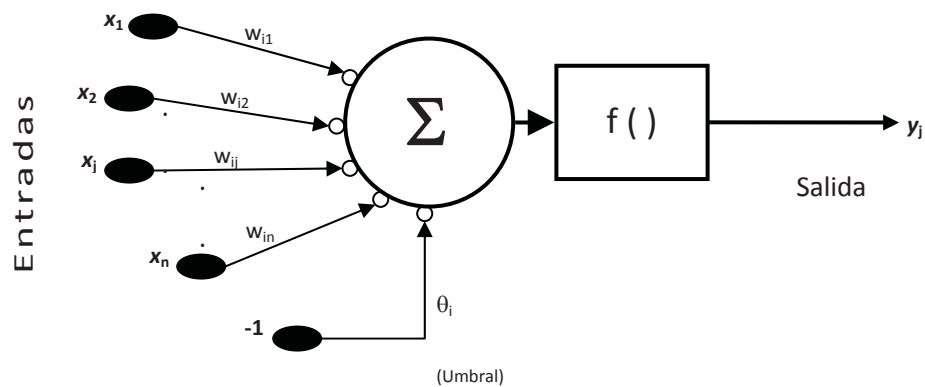
Otros trabajos apuntan al empleo de sistemas CEP (Complex Event Processing) [WST2007]. Estos sistemas permiten capturar y procesar información financiera en tiempo real y reaccionar rápidamente a los cambios en las condiciones del mercado. Generalmente se emplean en conjunto con otros sistemas que permiten generar señales de compra/venta y rutear inmediatamente las señales a sistemas expertos en emitir y controlar órdenes. Tales sistemas pueden ir integrados directamente en el sistema CEP, permitiendo altas velocidades en la toma de decisiones.

Finalmente, existen técnicas de minería de datos que permiten determinar correlaciones entre datos y generar modelos predictivos, las cuales se aplican sobre un gran conjunto de datos [Bigus1996]. Se define la minería de datos como el descubrimiento eficiente de información no obvia y valiosa a partir de una gran colección de datos. Dentro de las técnicas de minería de datos conocidas, existen en particular 2 algoritmos que permiten generar modelos de predicción de variables discretas en base a un conjunto de datos pasados que inicialmente no parecen relacionados. Tales algoritmos incluyen las Redes Neuronales y Redes Bayesianas.

### 3.3.1 Redes Neuronales

Considerando la aplicación de minería de datos mediante redes neuronales a problemas de AT y HFT debido a las características que las ANS (Artificial Neural System) presentan, hay que revisar previamente ciertos conceptos de las ANS.

En general las ANS se consideran estimadores universales de funciones. El elemento básico de la ANS es la neurona artificial.



**Figura 3.7** Modelo de Neurona estándar, con  $n$  variables de entrada y una única salida, la cual depende de una función interna sobre la cual no se conoce su fórmula.

En la figura 3.7 podemos apreciar cómo se compone una neurona  $i$  cualquiera, como se muestra en [Martín2003]. Vemos que una neurona estándar consiste en:

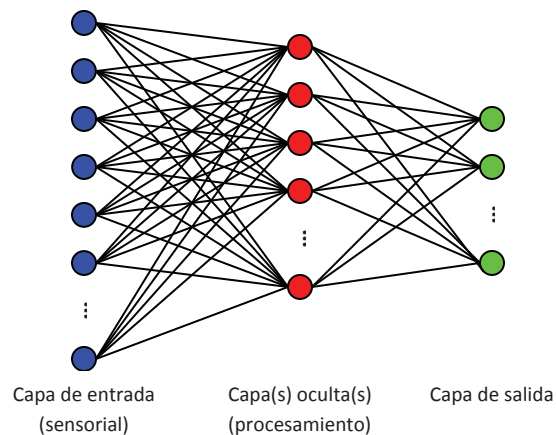
- Un conjunto de entradas  $x_j(t)$ , donde cada entrada tiene un correspondiente peso asociado  $w_{ij}$ .
- Una regla de propagación  $h_i(t)$  basada en las entradas y sus pesos, donde la sumatoria de la ponderación de las entradas por sus pesos es la más común, o sea  $h_i(t) = \sum w_{ij}x_j(t)$ .
- Una función de activación  $y_i = f_i(h_i(t))$ , que representa simultáneamente la salida de la neurona y su estado de activación.

Así, tenemos que una neurona básica (se utiliza para el ejemplo el perceptrón) toma desde la salida de otras neuronas o desde el exterior una serie de entradas, las cuales pueden ser continuas (un intervalo  $[a, b]$ ) o discretas (generalmente binarias, un conjunto  $\{0, 1\}$ ,  $\{-1, 1\}$ ). En este caso se utiliza un conjunto binario.



Las entradas se utilizan en conjunto con los pesos, en este caso sumado el valor de cada entrada multiplicado por su peso. A este valor resultante se resta un cierto umbral  $\theta$ , para usar el resultado en una función de salida. Generalmente tal función de salida corresponde a la función Escalón, cuyas salidas corresponden al mismo rango de las entradas (binarias o continuas). Otras funciones de salida aplicables son Identidad, Lineal a tramos, Sigmoidea, Gaussiana y Sinusoidal. Para más detalle sobre cada tipo de función de salida, se recomienda revisar [Martin2003].

Así, podemos concatenar neuronas de distintos modos, de modo que la activación de muchas neuronas pueda dar inicio a la activación de neuronas conectadas. De este modo, al conectar diversas neuronas se generan distintas arquitecturas de ANS, organizándose éstas generalmente por capas. De este modo, el modelo más general de arquitectura corresponde a lo que se conoce como perceptrón multicapa.



**Figura 3.8** Arquitectura de capa de un ANS, donde existe una serie de entradas, capas ocultas donde se realiza el procesamiento, y una capa de salida donde se entregan los resultados.

Se puede apreciar en la figura 3.8 que se cuenta con una serie de neuronas de entrada, que toman datos del exterior, una o más capas ocultas, que realizan el procesamiento, y una capa de salida, que entrega los resultados. La más simple arquitectura es la monocapa, donde la capa de entrada, la de procesamiento y salida son las mismas. La información por lo general fluye en un solo sentido, pero pueden darse arquitecturas con retroalimentación, hacia atrás o hacia las mismas neuronas (neuronas conectadas consigo mismas).

La potencia de las ANS radica en el hecho de que tienen 2 modos de operación: recuerdo y aprendizaje. Así, para el modo de aprendizaje, se utiliza una serie de casos de entrenamiento, para calibrar los pesos de la red, utilizando una serie de algoritmos específicos para el tipo casos de entrenamiento.

Los casos de entrenamiento pueden ser de dos tipos: conjuntos de entrada con sus salidas esperadas, o solamente conjuntos de entrada, sin sus salidas esperadas. Así, se dan 2 tipos de aprendizaje, supervisado y no supervisado. En el aprendizaje supervisado, los pesos se ajustan para que con ciertas entradas se logren resultados esperados, realizando pequeñas correcciones en cada iteración del aprendizaje. Para el aprendizaje no supervisado, los pesos se van ajustando para ir reconociendo patrones en los conjuntos de datos de entrada. Así, la red irá clasificando por similitud los casos de entrenamiento.

Una de las características del entrenamiento de ANS es que proporciona un estimador universal para funciones polinomiales.

El modo de recuerdo, para ANS de arquitectura unidireccional (la información fluye hacia las siguientes capas, sin devolverse), tenemos que ofrecen una gran estabilidad y capacidad para manejar grandes cantidades de datos. Esto se vuelve útil en escenarios de AT y HFT, donde la información puede fluir en grandes cantidades durante ciertos momentos del horario de negociación de un mercado determinado.

Sobre su aplicación en minería de datos, existen múltiples modelos basados en la función primaria por la cual se quieren aplicar [Bigus1996]. De este modo, se expande la cantidad de modelos que se pueden generar con ANS.

| Modelo   | Paradigma de entrenamiento | Topología           | Función primaria                            |
|--|----------------------------|---------------------|---|
| <b>Teoría de Resonancia Adaptiva</b>           | No supervisado             | Recurrente          | Agrupación                                  |
| <b>ARTMAP</b>                                  | Supervisado                | Recurrente          | Clasificación                               |
| <b>Back propagation</b>                        | Supervisado                | Un solo sentido     | Clasificación, modelamiento, serie temporal |
| <b>Redes de función de base radial</b>         | Supervisado                | Un solo sentido     | Clasificación                               |
| <b>Mapas de Kohonen</b>                        | No supervisado             | Un solo sentido     | Agrupación                                  |
| <b>Cuantización de vectores de aprendizaje</b> | Supervisado                | Un solo sentido     | Clasificación                               |
| <b>Back propagation recurrente</b>             | Supervisado                | Recurrente limitado | Modelamiento, serie temporal                |
| <b>Aprendizaje de diferencia temporal</b>      | Reforzamiento              | Un solo sentido     | Serie temporal                              |

**Tabla 3.2** Modelos de Redes Neuronales para Data Mining y su uso [Bigus1996]

Como se muestra en la tabla 3.2, el modelo aplicado depende del tipo de función que se requiere para el problema en cuestión. Para el caso de AT y HFT, se necesitan modelos que permitan la predicción de precios, por lo cual se deben usar modelos que permitan “Modelamiento”. La idea de aplicar ANS al problema de AT y HFT es poder determinar si el precio de un instrumento financiero tiene posibilidad de subir o bajar en el futuro inmediato e ingresar órdenes acordes al capital disponible o a la custodia que se posea de tal instrumento.

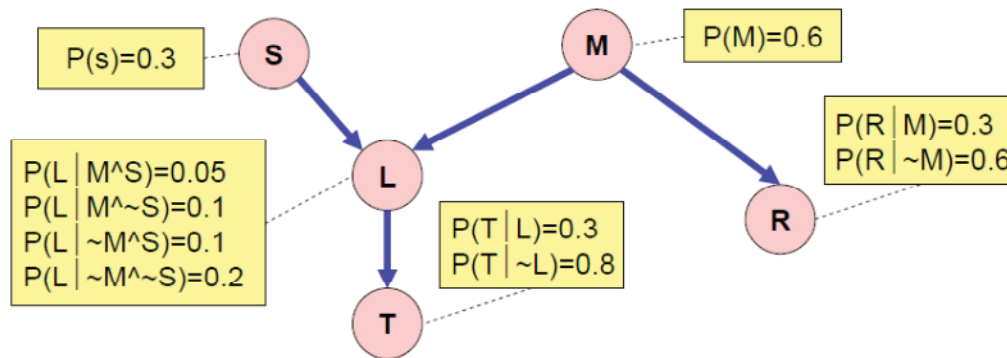
### 3.3.2 Redes Bayesianas

Las Redes Bayesianas (en adelante BN, *Bayesian Networks*) son un formalismo útil para representación y razonamiento bajo condiciones de incerteza [Cheng2001] o para representar conocimiento experto en dominios donde el conocimiento es incierto, ambiguo y/o incompleto [Shenoy2000]. Una BN es representada por:

$$BN = \langle N, A, \theta \rangle \tag{3.14}$$

Donde  $\langle N, A \rangle$  es un grafo dirigido acíclico (DAG), cada nodo  $n \in N$  representa una variable de dominio (un atributo de base de datos, el precio de un instrumento, etc.), y cada arco  $a \in A$  entre los nodos representa una dependencia probabilística entre los nodos asociados. Esto corresponde a la representación cualitativa del modelo.

A cada nodo  $n_i \in N$  se le asocia una probabilidad de distribución condicional (tabla), representadas en conjunto por  $\theta = \{\theta_i\}$ , que cuantifica cuanto depende un nodo de sus nodos padres. Esto corresponde a la representación cuantitativa del modelo.



**Figura 3.9** Ejemplo de Red Bayesiana. Se muestra el grafo dirigido acíclico, donde cada nodo tiene una tabla de probabilidad asociada.

Al observar la BN de la figura 3.9, se puede observar que para cada nodo, se emplea una notación especial para representar las probabilidades [Moore2001]. Esta notación se basa en notación de variables aleatorias discretas. De este modo, al anotar  $P(A)$ , se indica como la probabilidad de que  $A$  sea verdadero. Así, se plantean los siguientes axiomas:

$$\begin{aligned} 0 &\leq P(A) \leq 1 \\ P(\text{Verdadero}) &= 1 \\ P(\text{Falso}) &= 0 \end{aligned} \tag{3.15}$$

$$P(A \vee B) = P(A) + P(B) - P(A \wedge B)$$

Considerando lo expuesto, se puede inferir que:

$$P(\text{no } A) = P(\sim A) = 1 - P(A) \quad (3.16)$$

De esta forma, se tiene que existe el caso de que queremos saber la probabilidad de que cuando B sea verdadero A también sea verdadero. Esto se representa como  $P(A|B)$ . Empleando tal notación, la definición de probabilidad condicional se expresa como:

$$P(A|B) = \frac{P(A \wedge B)}{P(B)} \quad (3.17)$$

En consecuencia, se define la regla de cadena:

$$P(A \wedge B) = P(A|B)P(B) \quad (3.18)$$

Finalmente, con todo lo anterior se define la Regla de Bayes:

$$P(B|A) = \frac{P(A \wedge B)}{P(A)} = \frac{P(A|B)P(B)}{P(A)} \quad (3.19)$$

Para que una BN pueda aprender a partir de datos se requieren como mínimo 2 tareas: aprender la estructura de la red, que corresponde a determinar “que corresponde con que”, y aprender los parámetros, que corresponde a la fuerza de estas dependencias (las tablas de probabilidad). Existen algunos algoritmos que resuelven estas tareas, tales como SLA y TDPA.

- **Simple Learning Algorithm (SLA):** el algoritmo de aprendizaje simple que puede funcionar conociendo de antemano el orden de los nodos que componen el grafo o sin conocer el orden. Es un algoritmo que comienza con un grafo vacío y va creciendo o disminuyendo acorde a como se ingresan los datos.
- **3-Phase Dependency Analysis Algorithm (TPDA):** aun cuando el algoritmo SLA es capaz de trabajar correctamente, puede presentar problemas de ineficiencias computacionales. El algoritmo TDPA produce inicialmente un grafo que se espera sea parecido al grafo correcto. Se aplican 3 fases: “generación del

borrador”, “engrosamiento” y “adelgazamiento”. La primera fase corresponde a generar un grafo sin bucles, mientras que en las otras fases se aplica SLA.

Uno de los problemas conocidos de la aplicación de BN es la sobreadecuación (ver sección 3.2) de las salidas a los datos de entrenamiento. Una de las formas más comunes para evitar la sobreadecuación consiste en usar una gran cantidad de datos de entrenamiento. Esto se aplica tanto para BN como ANS.

Para el problema de AT y HFT, la aplicación de BN es la misma que para ANS. Se desea determinar si en el futuro inmediato el precio de un instrumento cambiará, acorde al valor y comportamiento de otros instrumentos e indicadores económicos que lo influyan o describan, para lo cual se requiere previamente determinar el grado de correlación entre los distintos indicadores y el instrumento que se quiere modelar.

#### **4. DISEÑO Y RESOLUCIÓN DEL PROBLEMA**

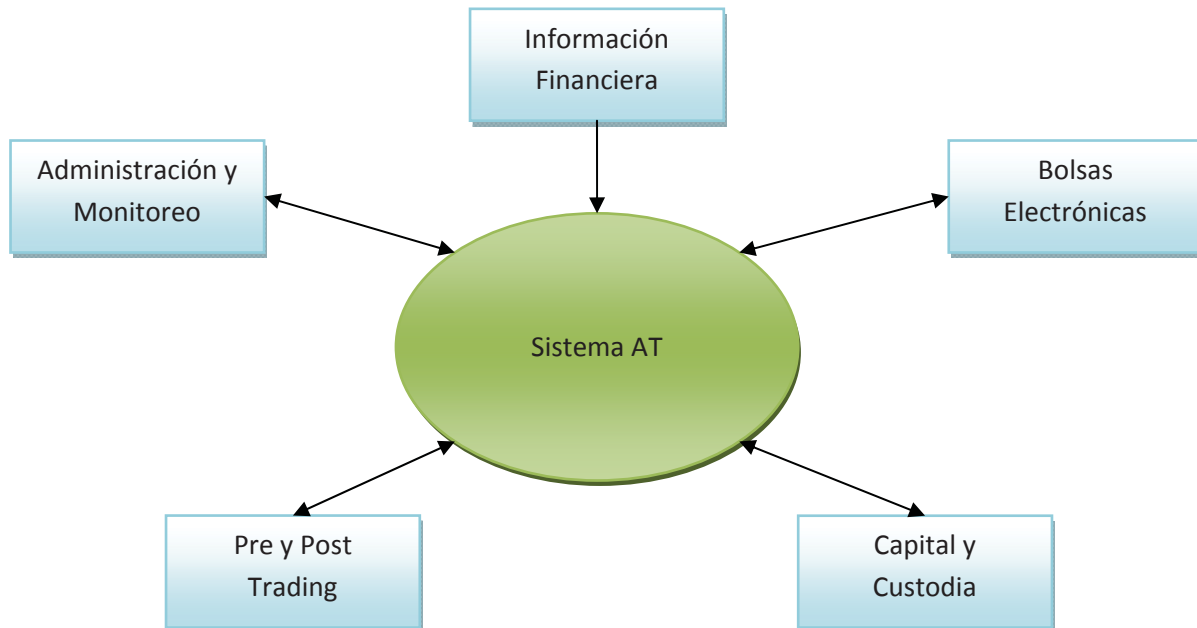
Cómo se mencionó en la sección 2, el objetivo principal es crear un sistema que pueda realizar *trading* de forma autónoma. Así, se define un diseño preliminar de un sistema que pueda aplicar durante un día completo de negociación para un mercado bursátil dado. Esto requiere como paso inicial definir y delimitar el mercado objetivo, ya que existen múltiples Bolsas de Valores en el mundo, donde cada una ofrece un abanico de mercados distintos, con sus propias regulaciones y restricciones.

Una vez definido el mercado objetivo y los instrumentos implicados, se puede proceder a diseñar un sistema que sea capaz de operar sobre éste, adaptado las regulaciones y restricciones que lo rigen. Además, esta misma definición servirá para determinar qué datos externos serán requeridos y como deberá capturarlos y tratarlos el sistema.

Para las estrategias en particular, se desea validar estrategias clásicas de *trading* basadas en *Moving Averages* en conjunto con optimización de parámetros mediante *Particle Swarm Optimization*. Independientemente de la estrategia adoptada, el sistema que se diseñe debe soportar cualquier tipo de estrategia, por lo que debe ser un sistema genérico y fácilmente extensible.

#### 4.1 Diseño conceptual sistema AT

Partiendo de la base de diseños y flujos propuestos en la sección 3, se define un sistema de *Algorithmic Trading* en base a los sistemas y factores externos con los que interactúa.



**Figura 4.1** Sistemas externos con los que interactúa un sistema AT/HFT. El sistema requiere alimentarse de información externa y al mismo tiempo interactuar con los sistemas que generan dicha información.

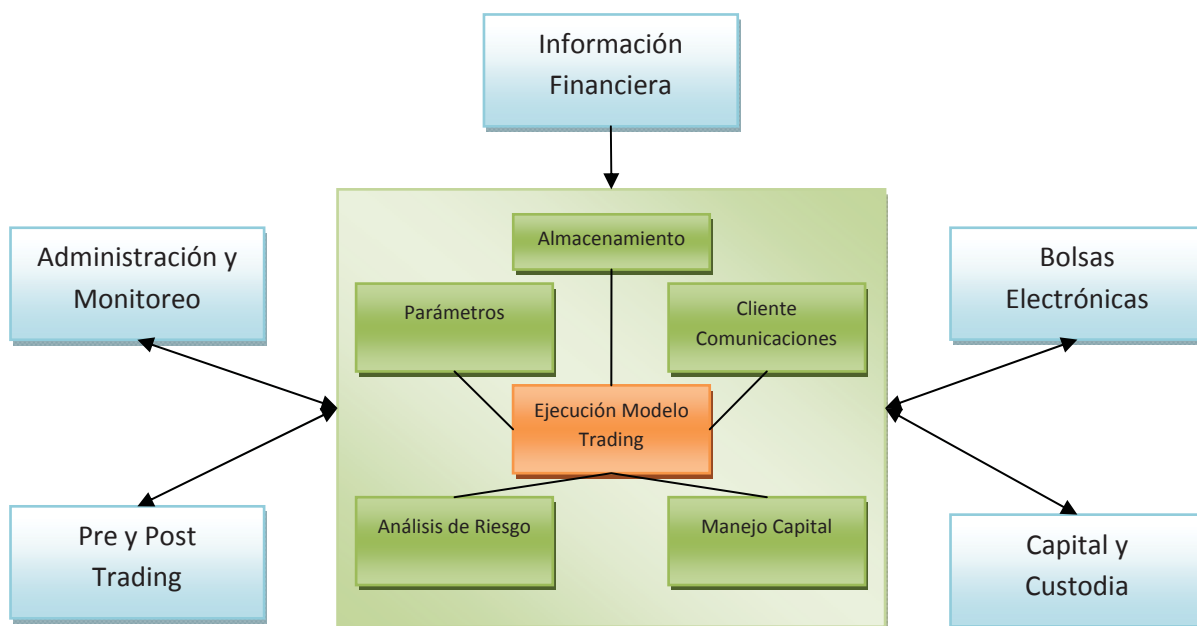
En la figura 4.1 se definen los sistemas externos con los cuales un sistema de AT/HFT debe interactuar, ya sea durante su horario de ejecución operativa como fuera de éste. El diagrama especifica los siguientes sistemas externos:

- **Administración y Monitoreo:** dependiendo de cómo se construya el sistema AT/HFT, se pueden agrupar muchos de estos sistemas bajo un único sistema de administración. Además, aun cuando el sistema AT/HFT pueda informar su estado y eventos especiales que requieran atención, es mejor consolidar estos eventos en un único sistema que permita a un operador manual tomar decisiones y acciones que se ameriten según la necesidad. Hay que recordar que la decisión experta de un *trader* humano puede, según su experiencia, ser más acertada que cualquier cálculo que un sistema automatizado pueda realizar. La comunicación con los sistemas de administración y monitoreo es bidireccional.



- **Información Financiera:** esto incluye toda fuente de datos financieros, indicadores económicos, noticias y hechos esenciales que el mercado objetivo no es capaz de entregar a través de los mecanismos de Bolsa Electrónica (ver abajo). Pueden ser distintos orígenes de datos, con formatos diversos e incluso no estructurados. Ejemplos de estos son el Banco Central, para obtener el valor del Dólar Observado, Euro, Unidad de Fomento, etc., o Bloomberg, para obtener índices internacionales como NASDAQ Composite, S&P500, FTSE Eurotop 100, etc. Estos datos alimentan algunos de los modelos existentes, como los modelos predictivos. La información fluye sólo en un sentido, desde los sistemas de información financiera hacia el sistema de AT/HFT.
- **Bolsas Electrónicas:** corresponde tanto al mismo sistema de negociación electrónica que una Bolsa de Valores provee, como sus mecanismos de comunicación. No necesariamente una Bolsa de Valores provee ambos sistemas. En algunas ocasiones se provee sólo el sistema de negociación electrónica, pero las órdenes son ingresadas manualmente (p.ej. el mercado de Renta Fija nacional en la Bolsa de Santiago, a la fecha sólo provee el sistema TeleRenta, pero no hay protocolo de ingreso de órdenes de forma electrónica). Para los protocolos de comunicación, la Bolsa de Valores que ejecuta el mercado seleccionado puede utilizar un protocolo propietario o uno estándar. El protocolo estándar de la industria es FIX (Financial Information eXchange - [www.fixprotocol.org](http://www.fixprotocol.org)), que corresponde a un protocolo de dominio público, el que puede funcionar sobre múltiples protocolos de transporte (TCP, JMS, etc.) y es utilizado por cientos de Bolsas de Valores, bancos, instituciones financieras y de inversión en el mundo. Esto incluye tanto los mecanismos de ruteo de órdenes como los mecanismos DMA (Direct Market Access, sistemas de baja latencia para el ingreso, modificación y anulación de órdenes). Independientemente del protocolo o mecanismo empleado, la comunicación siempre es bidireccional, habiendo un intercambio constante de mensajes entre el sistema de negociación o el sistema de ruteo de órdenes y el sistema de AT/HFT.
- **Capital y Custodia:** el requisito básico para realizar una orden de compra de un instrumento financiero es disponer del capital para liquidar la operación en caso de que se realice. El requisito para realizar una orden de venta es disponer de la cantidad suficiente del instrumento financiero en custodia para hacer el traspaso si se concreta la operación. Normalmente esta información reside en sistemas externos que manejan la cantidad disponible de efectivo y la cartera de inversiones que un cliente posee. De este modo, un sistema AT/HFT debe poder comunicarse de forma efectiva con un sistema de capital/custodia para saber cuáles son los límites máximos de las inversiones que puede realizar. En caso de que el sistema de capital/custodia no se alimente directamente desde las órdenes realizadas, el sistema AT/HFT deberá informar sobre las variaciones de capital y custodia producto de las órdenes que realice y se ejecuten. Generalmente este último caso no aplica, dado que el sistema de liquidación de una corredora de Bolsa se alimenta automáticamente de sus sistemas de ruteo de órdenes al mercado, por lo que la comunicación es unidireccional, desde el sistema de capital/custodia al sistema AT/HFT.

- **Pre y Post Trading:** los sistemas de Post-Trading incluyen los sistemas de conciliación y los sistemas de generación de métricas y reportes. Los sistemas de Pre-Trading incluyen los sistemas de análisis y calibración de modelos de *trading*. La conciliación corresponde a tomar las órdenes ejecutadas enviadas a los sistemas de negociación electrónica y compararlas con el estado de los sistemas de capital/custodia, para determinar si existe una descuadratura y determinar si la causa proviene desde el sistema AT/HFT o es causada por una condición exógena. La calibración y análisis de modelos de trading es una de las partes fundamentales del sistema, dado que el resultado final será dependiente de la calidad del modelo que se empleará durante el período de negociación. El sistema de calibración puede además ejecutarse durante el período de negociación, obteniendo un sistema que se autoajusta a medida que cambian las condiciones del mercado. La comunicación entre los sistemas de Pre y Post-Trading con el sistema de AT/HFT es bidireccional.



**Figura 4.2** Sistemas externos y módulos internos del sistema de AT/HFT. Se definen 5 módulos que entregan soporte al módulo principal del sistema, encargado de realizar la ejecución del modelo de *trading* durante el horario de operación.

Sobre los módulos internos que conforman al sistema propuesto, se destacan en la figura 4.2 los módulos principales que aportan la funcionalidad requerida para resolver el problema de *trading* automático. En detalle, cada módulo corresponde a:

- **Almacenamiento:** dependiendo del modelo de trading empleado, este requerirá información histórica o información bursátil actual, posiblemente en línea, previamente procesada y adaptada al modelo. De este

modo, el módulo de almacenamiento se encarga de comunicarse con las fuentes de información financiera y adaptarla a las necesidades del modelo, junto con mantener una copia de la información de mercado en tiempo real obtenida desde el sistema de negociación electrónica de la Bolsa donde se opera, ya sea mediante mecanismos DMA o mediante el protocolo de comunicaciones del Cliente de Comunicaciones (ver abajo). De este modo, el módulo de almacenamiento puede interpretarse como un repositorio de datos de uso interno y un proceso con modalidad *batch* y en línea basado en una colección de clientes de consulta ad-hoc al origen de datos financieros y al modelo de *trading*, que se encuentra continuamente en funcionamiento.

- **Cliente de Comunicaciones:** el modulo de comunicaciones se encarga de recibir las solicitudes de órdenes de compra y venta, modificaciones, anulaciones y último estado por parte del módulo central de ejecución del modelo. Para esto, necesita implementar uno de los mecanismos de comunicación con el mercado objetivo o poder comunicarse con un cliente existente que funcione con el mercado objetivo. En detalle, las solicitudes que debe poder manejar por parte del módulo central corresponden a:
  - Nueva Orden: corresponde a solicitar ingresar una orden de compra o venta que no ha sido ingresada antes. Los parámetros dependen exclusivamente del mercado objetivo (p.ej.: para el mercado de Acciones Renta Variable nacional, los parámetros serán el nemotécnico de la acción, el precio límite, la cantidad y la fecha máxima de permanencia en el mercado). Esta solicitud debe devolver un identificador único de la orden ingresada.
  - Cancelar Orden: corresponde a solicitar la anulación de una orden ingresada previamente, que posiblemente aun esté vigente en el mercado. El parámetro se que requiere es el identificador único de la orden, que se generó al ingresarla al mercado. Esta solicitud devolverá la cantidad de la orden que se logró anular.
  - Modificar Orden: corresponde a modificar alguno de los parámetros de la orden, excepto el nemotécnico (p.ej.: cambiar el precio y/o cantidad). Dado que no todos los mecanismos de comunicación soportan esta operación, generalmente se implementa como una anulación seguida de un ingreso. Devuelve el identificador de la nueva orden generada en caso de que se logra completar la operación de anulación.
  - Último estado conocido: en el caso de que el sistema AT/HFT presente una falla crítica y deba ser reiniciado, el sistema debe ser capaz de restablecer su estado previo. Para esto es posible tener que actualizar el estado de todas las órdenes que estaban vigentes antes de producirse el problema, para saber si se ejecutaron y no se deben seguir considerando. También sirve para determinar que ha ocurrido en caso de un fallo crítico por el lado del sistema de negociación electrónica (p.ej.: falla

de sistema Telepregón HT en su día de puesta en marcha el 12 Abril 2010, impidió el ingreso de órdenes por todos los canales de comunicación disponibles [E&N2010]).

Adicionalmente, el cliente de comunicaciones debe ser capaz de recibir mensajes provenientes desde el mercado objetivo, los cuales corresponden a:

- Estado de Ingreso: cuando se envía una solicitud de orden nueva, modificación o anulación, el mercado destino debe informar de vuelta el resultado de la operación. Esto puede ser un mensaje de éxito (p.ej.: la orden ingresó al mercado) o de rechazo (p.ej.: no se pudo anular la orden porque ya se había anulado anteriormente). En el caso de los rechazos es útil conocer la causa, para que el sistema pueda actuar acorde (p.ej.: si se suspende una acción en el mercado, el mensaje de rechazo indicará que el nemotécnico de esa acción se encuentra suspendido, por lo que el sistema AT/HFT debe dejar de considerar esa acción para transar por un período determinado).
- Reportes de Ejecuciones: una orden que ingresa al mercado objetivo puede llegar en algún momento a ser compatible con una oferta ingresada por otro negociador, pero esta ejecución (calce de ofertas) se realiza de forma asíncrona. De este modo, el mercado informa de vuelta que la orden se ha ejecutado parcial o totalmente, indicando la cantidad de unidades comprometidas en la transacción y el precio al que se realizó.
- **Parámetros:** el usuario puede definir ciertos parámetros requeridos por el sistema AT/HFT que son requeridos para la operación normal. De este modo, se necesita un módulo que administre estos parámetros mediante una única interfaz, para que cualquier parte del sistema pueda utilizarlos. La definición de estos parámetros puede variar según el mercado objetivo (p.ej.: el horario de negociación), el instrumento financiero que se transe (p.ej.: si es afecto o no a impuesto), el canal sobre el que se transa (p.ej.: la comisión por operación que cobra un corredor con respecto a otro), y múltiples otros factores. Se puede interpretar el módulo de parámetros como un conjunto de un sistema de almacenamiento, un mecanismo de caché de datos, una interfaz de consulta interna y una interfaz de administración que le permita a los sistemas externos interactuar con los parámetros cargados en el sistema.
- **Manejo de Capital:** como se mencionó anteriormente, uno de los requisitos para operar en cualquier mercado es contar con capital suficiente. El sistema AT/HFT necesita mantener registro de todo capital comprometido en las operaciones que realiza, junto con conocer cuánto capital y cuanta custodia de cada instrumento se encuentra disponible para transar. Para esto, el módulo de manejo de capital debe ser capaz de comunicarse con el sistema externo de capital/custodia para obtener las cantidades disponibles antes de realizar ingresar cualquier orden de compra/venta. En el caso de que el sistema externo de capital/custodia

no actualice automáticamente su estado a partir de las órdenes ingresadas, el módulo de manejo de capital debe informar sobre las variaciones de capital generadas. Se define así al módulo de manejo de capital como un cliente de consulta ad-hoc al sistema externo de manejo de capital, con una interfaz interna común para consultar los niveles de capital disponible y, opcionalmente, para informar las variaciones de capital al sistema externo.

- **Análisis de Riesgo:** el objetivo primario del manejo de riesgo es limitar las posibles pérdidas [Aldridge2010]. Las causas del riesgo son variadas y difíciles de identificar y manejar. El factor clave está en determinar cuándo se puede producir una pérdida considerable de capital debido a un movimiento de precios adverso. Pardo [Pardo2008] identifica 3 categorías: riesgos de comercio (posibilidad de pérdida financiera debido a una posición de mercado individual), riesgos de estrategia (posibilidad de pérdida financiera debido al uso de una estrategia o modelo de *trading* particular) y riesgos de portafolio (posibilidad de pérdida a nivel de portafolio de inversiones, por una combinación de múltiples estrategias o múltiples mercados).



Figura 4.3 Ejemplo de *trailing stop-loss* para nemotécnico MSFT en Nasdaq. Se utiliza un precio de stop-loss que varía en el tiempo según como se comporta el instrumento financiero (fuente: [www.tradingonlinemarkets.com/Articles/Technical\\_Analysis/trailing\\_stop\\_loss\\_order.htm](http://www.tradingonlinemarkets.com/Articles/Technical_Analysis/trailing_stop_loss_order.htm))

La forma más simple de manejo de riesgo es contar con un mecanismo de *stop-loss*, que permita cancelar una posición de compra o liquidar una custodia frente al evento de que el precio de un instrumento baje más allá de cierto umbral. Esto permite atacar los riesgos de comercio de forma efectiva y simple. El mecanismo de *stop-loss* puede ser un precio fijo, un porcentaje o una banda variable (*trailing stop-loss* en figura 4.3). En modelos autoajustables mediante optimización metaheurística corresponde a uno de los

parámetros que se quiere optimizar. Algunos sistemas de negociación electrónica proveen soporte nativo para órdenes de compra/venta con parámetros de *stop-loss* incorporados.

En el caso de que una condición de riesgo sea detectada, el módulo de manejo de riesgo debe iniciar las acciones necesarias para minimizar las pérdidas. Normalmente esto corresponde a cancelar las órdenes de compra y dejar las operaciones de compra en pausa si el precio aumenta considerablemente, y liquidar custodias mediante ventas si el precio disminuye debajo de la brecha de *stop-loss*. Existe una condición adicional que los sistemas AT/HFT manejan que es la banda de *stop-win*. Esto responde a que se logra una ganancia significativa al alcanzar un determinado precio o una determinada rentabilidad sobre la compra original, se liquide la custodia adquirida. Esto responde a que como las condiciones del mercado son inciertas y pueden variar rápidamente, es mejor quedarse con una ganancia inmediata y aprovechar los recursos obtenidos para realizar otros negocios. De este modo, el sistema de manejo de riesgo debe poder considerar la condición de *stop-win* como una segunda banda fija o móvil, que también se puede determinar mediante optimización con modelos basados en metaheurísticas.

Finalmente, el módulo principal del sistema es el módulo de ejecución del modelo de *trading*. Este módulo es el que realiza la generación de señales de compra/venta, asignación de recursos a instrumentos individuales y otras funciones más, utilizando todos los módulos internos mencionados para interactuar con los sistemas externos. El detalle de cómo se construirá el módulo principal se explica en la sección 4.4.

## 4.2 Selección de mercado e instrumentos financieros

Para la selección del mercado y los instrumentos financieros, se requiere que éstos presenten las siguientes características:

- El mercado debe tener regulaciones que lo hagan accesible. Esto incluye regulaciones de horario (un horario conocido de operación, donde el intervalo sea continuo), regulaciones legales (que sea posible operar dentro del mercado y que se conozcan las posibles grabaciones fiscales) y regulaciones operativas (que los costos por operar en un mercado sean conocidos).
- Que el mercado tenga instrumentos líquidos. La liquidez de un instrumento mide cuán fácil o difícil es convertir en dinero corriente la custodia de dicho instrumento financiero. Debe existir información suficiente para determinar la liquidez de los instrumentos o deben existir índices que determinen dicha liquidez.
- Finalmente, el mercado seleccionado debe permitir el ingreso electrónico de órdenes de compra/venta y un sistema de negociación electrónica.

El sistema propuesto se ejecuta sobre el mercado accionario nacional. Esto corresponde a todo el mercado de Instrumentos de Renta Variable en moneda nacional (Acciones Nacionales). Sobre este mercado, se tiene que presenta las siguientes características:

- Tiene un horario de operación definido. El mercado de Acciones Nacionales opera desde las 09:30 hasta las 17:00 en verano y desde las 09:30 hasta las 16:00 en invierno. Durante todo este horario es posible negociar (ingresar ofertas, modificarlas o cancelarlas). Existe un espacio de tiempo entre las 09:00 y las 09:23 (más un delta de tiempo aleatorio entre 0 y 5 minutos) llamado PreApertura, donde es posible ingresar o cancelar ofertas sin que estas sean ejecutadas (calzadas con otras ofertas).
- Para el caso de una empresa corredora de Bolsa, los costos son conocidos (derechos de bolsa) y la regulación existente es exhaustiva, principalmente en las garantías que cada corredora de Bolsa debe mantener para poder seguir operando. En el caso de un inversor particular, los costos varían según cada corredora de Bolsa, pero también son conocidos (costos fijos y comisiones variables).
- Es posible identificar cuáles son las acciones con mayor liquidez al revisar la composición del índice IPSA (Índice de Precios Selectivo de Acciones).

- Provee tanto los mecanismos de negociación electrónica con soporte para alta frecuencia como los mecanismos de comunicación electrónica para el ingreso de órdenes. Sobre esto último, existen mecanismos provisto para negociadores institucionales (corredores de Bolsa e instituciones financieras) mediante mecanismos DMA (Direct Market Access) sobre protocolo FIX 4.4, como mecanismos para mercado *retail* (para usuarios no institucionales) provisto por corredores de Bolsa, empleando clientes de ruteo de órdenes sobre FIX 4.4 con *polling* de órdenes.

Resumiendo, para el sistema de esta Tesis se utilizan acciones del mercado de Renta Variable nacional, con alta presencia y/o pertenecientes a los indicadores IPSA que no se encuentren suspendidos. Esto no implica descartar las acciones que no cumplan con tal criterio para efectos de análisis y almacenamiento, pero sí para no utilizarlas durante una ejecución normal del sistema AT/HFT a menos que cambien su condición a alta presencia.



### 4.3 Obtención de datos relevantes

Para el sistema de esta Tesis, se requieren 2 tipos de datos: datos propios de los instrumentos que cumplan los criterios de la sección 4.2 y datos de indicadores relevantes. Sobre los datos propios de los instrumentos, se pueden emplear distintos niveles de detalle de los datos. En el caso de acciones nacionales, estos pueden ser:

- **Datos públicos:** estos son datos públicos intraday e históricos, provistos por la Bolsa de Santiago directamente en su sitio Web, basados únicamente sobre los calces ocurridos. Estos datos se proveen como un archivo CSV (*Comma Separated Values*, valores separados por coma) e incluyen:
  - Fecha: para el caso intraday es la fecha y hora minuto a minuto, desde el comienzo de la negociación hasta la hora de cierre, sin incluir el período de pre-market. Para el caso histórico, es el día de negociación.
  - Open: precio de apertura, corresponde al precio de mercado del período anterior.
  - High: mayor precio de mercado de los calces realizados durante el período cubierto por la fila.
  - Low: menor precio de mercado de los calces realizados durante el período cubierto por la fila.
  - Close: precio de cierre del período cubierto por la fila.
  - Volume: volumen transado durante el período cubierto por la fila. En detalle es la cantidad de unidades transadas en el período de la fila.

| LAN             |       |       |       |       |        |      |
|-----------------|-------|-------|-------|-------|--------|------|
| Fecha           | Open  | High  | Low   | Close | Volume |      |
| 03-07-2012 9:30 | 13100 | 13100 | 13100 | 13100 | 13100  | 0    |
| 03-07-2012 9:31 | 13100 | 13100 | 13100 | 13100 | 13100  | 62   |
| 03-07-2012 9:32 | 13100 | 13000 | 13000 | 13000 | 13000  | 2700 |
| 03-07-2012 9:33 | 13000 | 13000 | 13000 | 13000 | 13000  | 839  |
| 03-07-2012 9:34 | 13000 | 13001 | 13001 | 13001 | 13001  | 550  |
| 03-07-2012 9:35 | 13001 | 13001 | 13001 | 13001 | 13001  | 0    |
| 03-07-2012 9:36 | 13001 | 13001 | 13001 | 13001 | 13001  | 76   |
| 03-07-2012 9:37 | 13001 | 13003 | 13003 | 13003 | 13003  | 1116 |
| 03-07-2012 9:38 | 13003 | 13003 | 13003 | 13003 | 13003  | 0    |
| 03-07-2012 9:39 | 13003 | 13000 | 13000 | 13000 | 13000  | 1812 |
| 03-07-2012 9:40 | 13000 | 13000 | 13000 | 13000 | 13000  | 1923 |
| 03-07-2012 9:41 | 13000 | 13000 | 13000 | 13000 | 13000  | 9908 |

Tabla 4.1 Ejemplo de datos de la interfaz pública intraday para acción nacional LAN.

Este origen de datos puede aplicarse en modelos basados en métodos estadísticos u otras que no requieran un conjunto de datos detallados. Para modelos predictivos, no presentan en nivel de detalle requerido.

- **Market Replay:** son datos privados históricos, provistos por la Bolsa de Santiago a corredores de bolsa, instituciones financieras y negociadores profesionales. Estos datos se proveen como un sistema de consulta propietario o como archivos diarios de operaciones, e incluyen información tanto de calces como de órdenes, permitiendo conocer en detalle el estado del mercado. Para el caso de las órdenes, se tiene como información relevante:
  - Fecha/Hora: fecha y hora con precisión de microsegundo en el que ingresó la orden al mercado.
  - Compra/Venta: indica si la orden es de compra o venta.
  - Movimiento: indica si se está ingresando la orden, si se está modificando o eliminando.
  - Precio: el precio límite de la orden, cero para precio de mercado.
  - Cantidad: cantidad de unidades a transar.
  - Corredor: identificador del corredor que está realizando la operación.

Para el caso de los calces, se tiene como información relevante:

- Fecha/Hora: fecha y hora con precisión de microsegundo en el que se realizó el calce.
- Precio: precio al que se realizó el calce.
- Cantidad: cantidad comprometida en la transacción.
- Corredor comprador: corredor que realizó la orden de compra que gatilló la transacción.
- Corredor vendedor: corredor que realizó la orden de venta que gatilló la transacción.

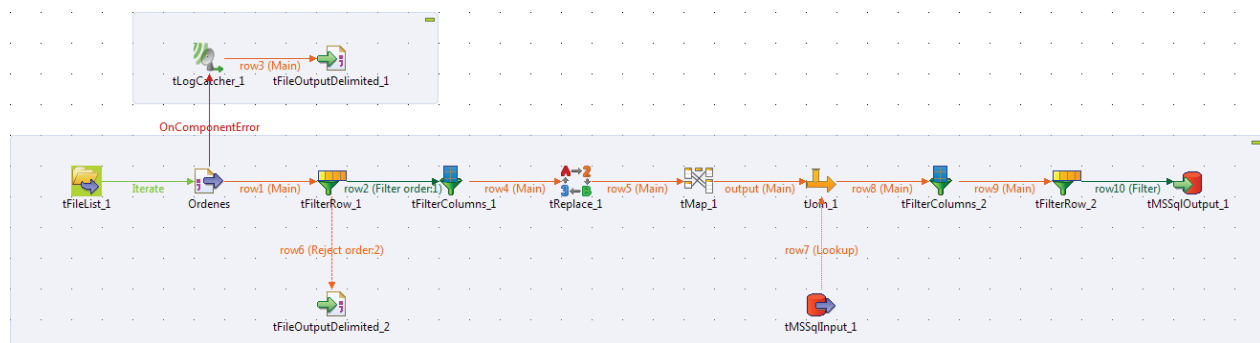
Este origen de datos resulta útil en modelos predictivos, dado el alto nivel de detalle que provee. Para modelos estadísticos, los datos deben ser tratados previamente para generar los intervalos consolidados que requieren tales modelos. Aun así, es un origen de datos superior a la información pública disponible, permitiendo afinar períodos de consulta que no son provistos por los datos públicos.

- **Bolsa Ticker:** servicios pagos de datos bursátiles sobre precios y volúmenes por instrumento, con desfase de 20 minutos. Dado que un sistema AT/HFT requiere información en tiempo real, no resulta útil como fuente de datos. Se descarta como origen de datos.
- **Market Data:** similar a Market Replay, pero funciona sobre protocolo FIX 4.4 mediante conexiones TCP multicast. Incluye toda la información de Market Replay, adicionando información tal como:

- Noticias y hechos relevantes, como texto corto.
  - Información de suspensión de un instrumento financiero dado.
  - Solicitudes on-demand de información.
- **Web Services de consulta de mercado:** servicios de datos disponibles públicamente o a través del sitio de un corredor de Bolsa. Incluyen cierta profundidad de mercado, con datos como:
    - Precio de mercado: precio del último calce realizado para un instrumento.
    - Volumen: monto acumulado transado en el día.
    - Mejor compra: mejor oferta de compra actualmente en el mercado.
    - Mejor venta: mejor oferta de venta actualmente en el mercado.

Estos servicios Web existen en diversas modalidades, tanto en formato REST, JSON o SOAP. En el caso de que no se pueda disponer de una mejor fuente de datos, pueden servir para alimentar una base de datos histórica que puede servir para modelos predictivos y estadísticos.

Para el caso particular de la implementación, se cuenta con una muestra de datos provenientes desde Market Replay, los cuales se han tratado de modo tal que permitan su procesamiento tanto por modelos predictivos como estadísticos. De este modo, se ha implementado un proceso ETL (*Extract, Transform & Load*) mediante la herramienta de código abierto Talend Data Integration.



**Figura 4.4** Trabajo ETL que carga diariamente los datos de operación bursátil a partir de Market Replay a una BD local para análisis, generación y validación de modelos AT/HFT. Se concatenan una serie de tareas que leen, filtran, modifican, combinan y almacenan los datos.

En la figura 4.4 se muestra uno de los 2 trabajos ETL que se encargan de procesar los datos de Market Replay. En particular, el trabajo mostrado corresponde al proceso de carga de órdenes de acciones de renta variable nacional. El proceso sigue la siguiente secuencia: lee una serie de archivos de texto que contienen las órdenes, itera sobre ellos, aplica una serie de filtros para descartar datos de control y comentarios, elimina las columnas innecesarias, reemplaza valores o les cambia el tipo de datos, realiza cálculos en memoria para generar nuevas

columnas, realiza Joins con otras tablas para obtener nuevas columnas con llaves foráneas, elimina columnas sobrantes y por último almacena sólo los registros que cumplan un determinado criterio. Esto forma parte integral del módulo de almacenamiento descrito en la sección 4.1, donde el medio de almacenamiento es una base de datos SQL Server Express de uso gratuito.

Adicionalmente a los datos mostrados, los modelos predictivos pueden aprovechar información adicional, externa al mercado, como variables que pueden afectar el precio de un instrumento financiero, pudiendo generar una predicción más certera que indique cuando el precio del instrumento está próximo a subir o bajar. Para esto, se pueden usar el Dólar Acuerdo, Euro, Unidad de Fomento y otros indicadores que provee el Banco Central. El mismo Banco Central entregar estos datos de forma gratuita, actualizados diariamente, a través de su sitio Web. Se puede utilizar la funcionalidad provista para descarga de archivos Excel para realizar la carga de una base de datos local que se emplee en conjunto con modelos predictivos.

| <b>Tipos de cambio (pesos por dólar)</b> |                           |                         |
|--|---------------------------|-------------------------|
| <b>Periodo</b>                           | <b>1. Dólar observado</b> | <b>2. Dólar acuerdo</b> |
| <b>02-01-2012</b>                        | 519,2                     | 675,98                  |
| <b>03-01-2012</b>                        | 518,2                     | 676,04                  |
| <b>04-01-2012</b>                        | 514,34                    | 675,56                  |
| <b>05-01-2012</b>                        | 512,23                    | 676,96                  |
| <b>06-01-2012</b>                        | 511,8                     | 677,52                  |
| <b>09-01-2012</b>                        | 509,54                    | 677,91                  |
| <b>10-01-2012</b>                        | 512,4                     | 677,74                  |
| <b>11-01-2012</b>                        | 508,76                    | 678,43                  |

**Tabla 4.2** Muestra de los indicadores de Dólar observado y Dólar acuerdo obtenibles desde el sitio Web del Banco Central (fuente: <http://si3.bcentral.cl/Siete/secure/cuadros/arboles.aspx>)

En el caso particular del Dólar observado (promedio del precio del Dólar USD alcanzado en las transacciones del día anterior entre bancos y empresas), se muestra un ejemplo de los primeros días del año 2012 para dicho indicador. Dado que procesar archivos Excel con Talend ETL es una tarea relativamente sencilla, se pueden incorporar fácilmente muchos años de información histórica al modelo predictivo del *trader* algorítmico.

#### 4.4 Diseño de sistema AT/HFT

Como se mencionó en la sección 4.1, el sistema se basa en 5 módulos anexos y un módulo central de ejecución del modelo. El módulo central es el que se encarga de mantener uno o más modelos de trading, basados en los antecedentes revisado en la sección 3, mediante una revisión diaria del comportamiento del mercado.

La forma básica de operar del módulo de ejecución del modelo consiste en una copia paralela del modelo de trading escogido por cada instrumento válido en el mercado objetivo (ver sección 4.2 para saber sobre los instrumentos). Cada copia accede a los módulos anexos de forma independiente, para solicitar información, interfaces de comunicación, etc., pero los módulos anexos manejan una única copia (*singleton*).

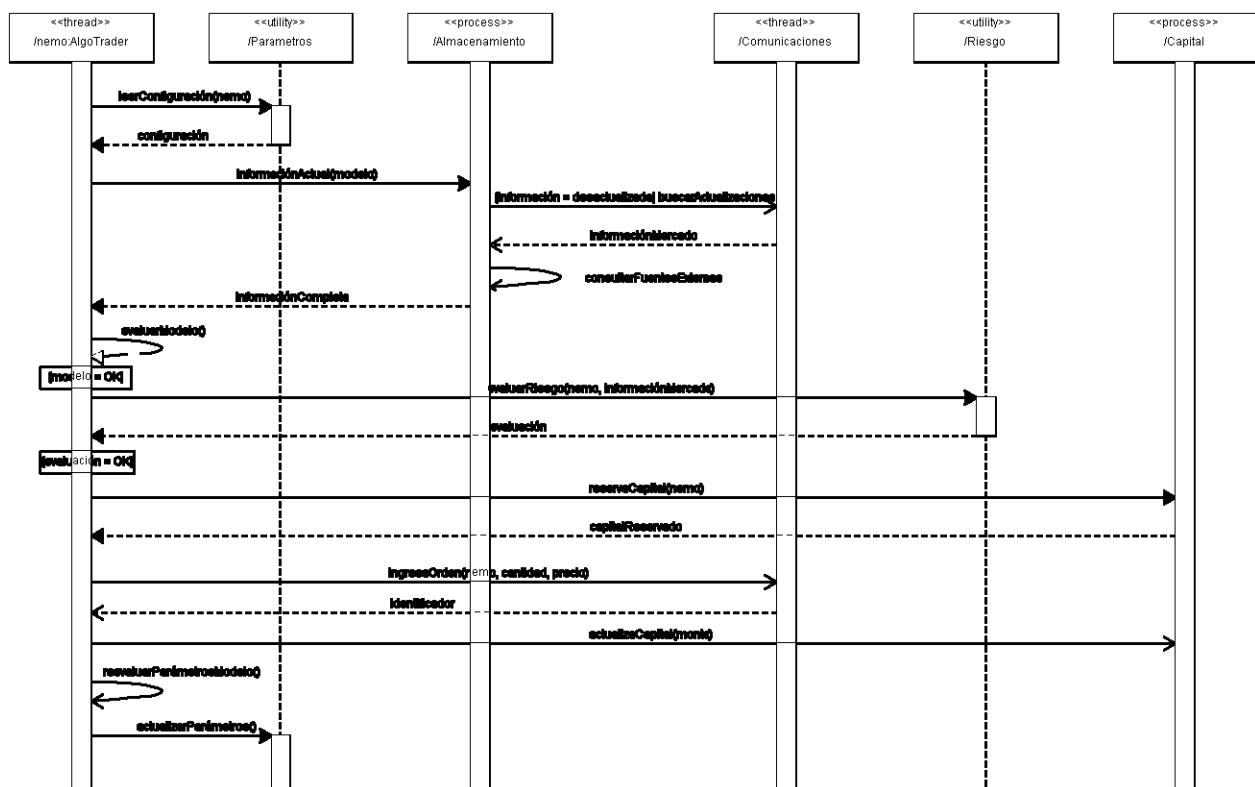


Figura 4.5 Diagrama de secuencia para AT/HFT en modalidad compra, con mecanismo de una hebra de ejecución de modelo por instrumento financiero. Las hebras de ejecución del modelo interactúan con el resto de los módulos del sistema y al final pueden generar una reevaluación del modelo para adaptarse a una condición cambiante del mercado.

Bajo el esquema propuesto, se grafica en la figura 4.5 el diagrama de secuencia UML para la funcionalidad de compra de acciones. El diagrama aplica para cualquier modelo de *trading* automático y permite ejecuciones paralelas. Se tiene una hebra por cada instrumento con posibilidad de transarse, cada una con el modelo escogido ajustado a las necesidades y características del instrumento.

La hebra solicita sus parámetros de configuración (que el operador humano pudo cambiar entre ejecuciones) para poder iniciar su ciclo. Luego solicita información de mercado actualizada para cargar al modelo. El proceso de Almacenamiento evalúa si es necesario actualizar su información, y en el caso de que la información esté desactualizada busca nueva información tanto en el mercado como en otras fuentes de datos. En cualquiera de los casos, el proceso devuelve información actualizada o reciente del mercado al ejecutor del modelo. Éste último evalúa el modelo y verifica si se produce una condición favorable para la compra. Si existe tal condición, la hebra solicita una evaluación de riesgo al módulo de Riesgo. Si el módulo de Riesgo determina que la condición de mercado y los parámetros de riesgo son correctos, autoriza la operación. Entonces se reserva una parte del capital disponible para realizar la compra, solicitando tal capital al módulo de manejo de capital y custodia. Con el capital disponible, se calculan los parámetros de la orden, para luego, a través del módulo de Comunicaciones, enviar tal orden de compra al mercado. Con la orden ingresada al mercado, se actualiza el capital disponible. El proceso se repite en ciclo durante todo el horario de negociación. Al final de cada ciclo es posible aplicar una revisión completa del modelo, para adaptarlo a las nuevas condiciones del mercado.

El proceso para las ventas es similar, pero se trabaja con la custodia de los instrumentos en vez del capital disponible.

#### 4.4.1 Modelo adaptivo con PSO

Se propone un modelo adaptivo de AT/HFT basado en alguna de las estrategias conocidas de *Moving Averages*. Para este caso, se presenta un modelo clásico de 2 *Moving Averages*, uno largo y otro corto, en conjunto con 2 bandas de manejo de riesgo mediante *stop-loss* y *stop-win*. De esta forma, existen 4 parámetros a optimizar:

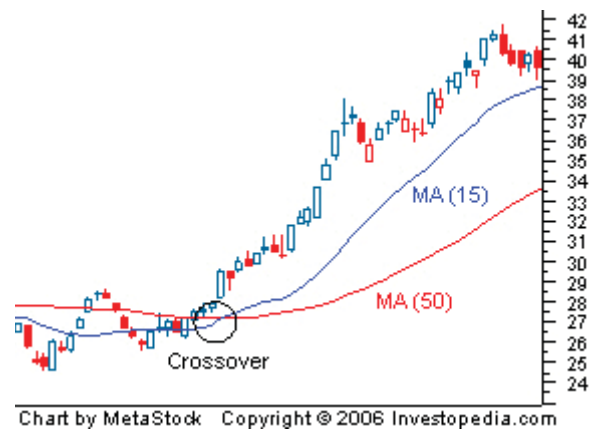
| Variable | Descripción                          | Tipo        |
|----------|--------------------------------------|-------------|
| $x_1$    | Período de tiempo para MA largo      | Entero      |
| $x_2$    | Período de tiempo para MA corto      | Entero      |
| $x_3$    | Banda de <i>stop-loss</i> porcentual | Real ]0, 1[ |
| $x_4$    | Banda de <i>stop-win</i> porcentual  | Real ]0, 1[ |

**Tabla 4.3** Variables para modelo adaptivo PSO con estrategia de 2 MA con bandas móviles

En la tabla 4.3 se muestra las variables implicadas en el modelo. Estas variables están sujetas a las siguientes restricciones:

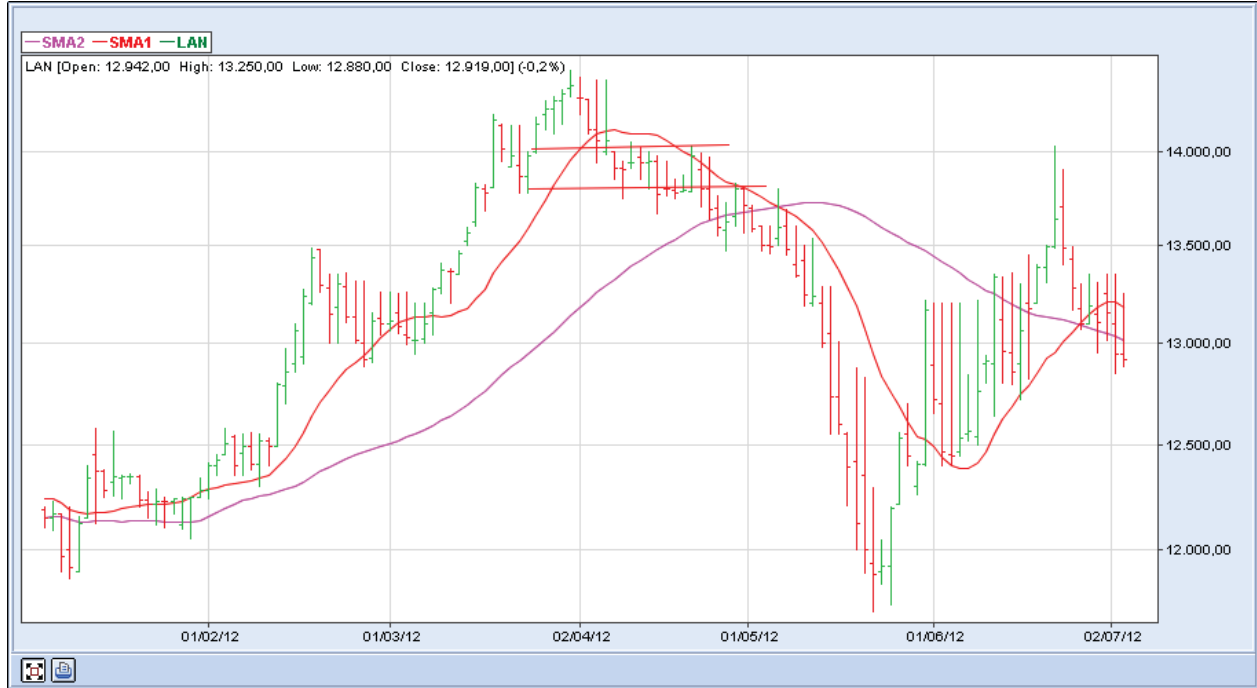
$$\begin{aligned}
 x_1 &> 1 \\
 x_2 &\geq 1 \\
 x_1 &> x_2 \\
 x_1, x_2 &\in \mathbb{N} \\
 x_3, x_4 &\in \mathbb{R}]0,1[
 \end{aligned}
 \tag{4.1}$$

El principio de una estrategia de 2 *Moving Averages* es identificar cuando existe un *crossover*, cuando la curva del MA corto interseca a la curva del MA largo.



**Figura 4.6** Ejemplo de *crossover* creciente en una estrategia de 2 MA. El MA corto interseca al MA largo, generando una condición para realiza una compra o una venta.

El cruce de las curvas se puede dar por debajo, cuando la curva del MA corto interseca a la curva del MA largo desde un valor inferior a uno superior (figura 4.6), o por arriba, cuando la curva del MA corto interseca a la curva del MA largo desde un valor superior a uno inferior. Cuando se produce un *crossover* del primer tipo (creciente) se produce una condición favorable para la compra, ya que el precio tiene una tendencia al alza. Cuando el crossover es del segundo tipo (decreciente) se genera una condición que descarta las compras y obliga a liquidar la custodia mediante ventas.



**Figura 4.7** Bandas de control en una estrategia de 2 Moving Averages. Las bandas de control permiten al mismo tiempo obtener ganancias antes de tiempo (*stop-win*) y evitar grandes pérdidas (*stop-loss*) antes de que la misma estrategia detecte una tendencia decreciente perjudicial.

Al conjunto de los 2 *Moving Averages* se aplican las bandas de control, para generar mecanismos de *stop-loss* y *stop-win* integrados al modelo (ver sección 4.1). Estas bandas cumplen el objetivo de optimizar el modelo, dado que permiten capital antes las ganancias mediante la banda superior (lo que disponibiliza antes el capital al resto de hebras concurrentes de ejecución del modelo de *trading*) y permiten reducir las pérdidas con la banda inferior. Este comportamiento se puede ver en el ejemplo de la figura 4.7.

Lo más importante para el modelo adaptivo con PSO es la función objetivo que se emplee. Como se mencionó en la sección 3.2.1, la función objetivo se realizará en primera instancia en base a optimizar el retorno neto del sistema. De este modo, la función objetivo será:

$$\text{Maximizar: } \sum_{i=0}^n cv_i pv_i (1 - v_i) - cc_i pc_i (1 + v_i) - f_i \quad (4.2)$$

Donde:

- $cv_i$ : cantidad vendida en el  $i$ -ésimo período dentro del horizonte de simulación



- $pv_i$ : precio de venta del  $i$ -ésimo período para el único instrumento transado en la simulación
- $cc_i$ : cantidad comprada en el  $i$ -ésimo período dentro del horizonte de simulación
- $pc_i$ : precio de compra del  $i$ -ésimo período para el único instrumento transado en la simulación
- $v_i$ : costos variables del  $i$ -ésimo período necesarios para transar
- $f_i$ : costos fijos del  $i$ -ésimo período necesarios para transar

Esta función objetivo no es nada más que el cálculo de los retornos netos, para un lapso de tiempo de  $n$  períodos iguales y consecutivos. Estos períodos pueden ser configurados según la granularidad de los datos de mercado que se tienen.

En el caso de que 2 simulaciones obtengan el mismo valor de la función objetivo se debe pasar al siguiente criterio de descarte, correspondiente a maximizar el beneficio por operación. Esto se puede interpretar cómo maximizar la ganancia obtenida entre una compra y su consecutiva venta. La aplicación de este criterio es en muchas ocasiones difícil de calcular, ya que la simulación tendrá que reemplazar órdenes que participaron en calces reales, lo que no siempre hace que cuadren las cantidades. En este caso sería complicado juntar la serie de operaciones de venta que corresponden a una compra previa. De este modo, es mejor pasar al tercer criterio de discriminación para un empate, que corresponde a calcular y comparar el ratio de Sharpe u otro ratio financiero válido.

El ratio de Sharpe se define como [Aldridge2010, Pardo2008]:

$$S = \frac{E[r] - r_f}{\sigma[r]}, \text{ donde}$$

$$E[r] = \frac{\sum_{i=1}^T r_i}{T}$$

$$\sigma[r] = \sqrt{\frac{\sum_{i=1}^T (r_i - E[r])^2}{T - 1}}$$
(4.3)

Finalmente, el ratio de Sharpe para procesos AT/HFT queda como:

$$S_{HFT} = \frac{E[r]}{\sigma[r]}$$
(4.4)

Donde:

- $r_i$ : retornos netos del  $i$ -ésimo período
- $r_f$ : tasa referencial libre de riesgo (constante que representa el costo de oportunidad)

- $T$ : cantidad de períodos

Teniendo calculado el ratio de Sharpe, la función objetivo queda como:

$$\text{Maximizar: } \sum_{i=0}^n cv_ipv_i(1 - v_i) - cc_i pc_i(1 + v_i) - f_i, S_{HFT} \quad (4.5)$$

Con esta función objetivo se puede generar el modelo adaptivo aplicando un algoritmo PSO que explore la mejor combinación de las variables definidas para el problema, de modo tal de tener en todo momento, de forma rápida y efectiva, un modelo adaptado al estado cambiante del mercado. En el caso de empate en las rentabilidades de las soluciones, se puede aplicar el ratio escogido para el caso.

## 4.5 Implementación del Sistema

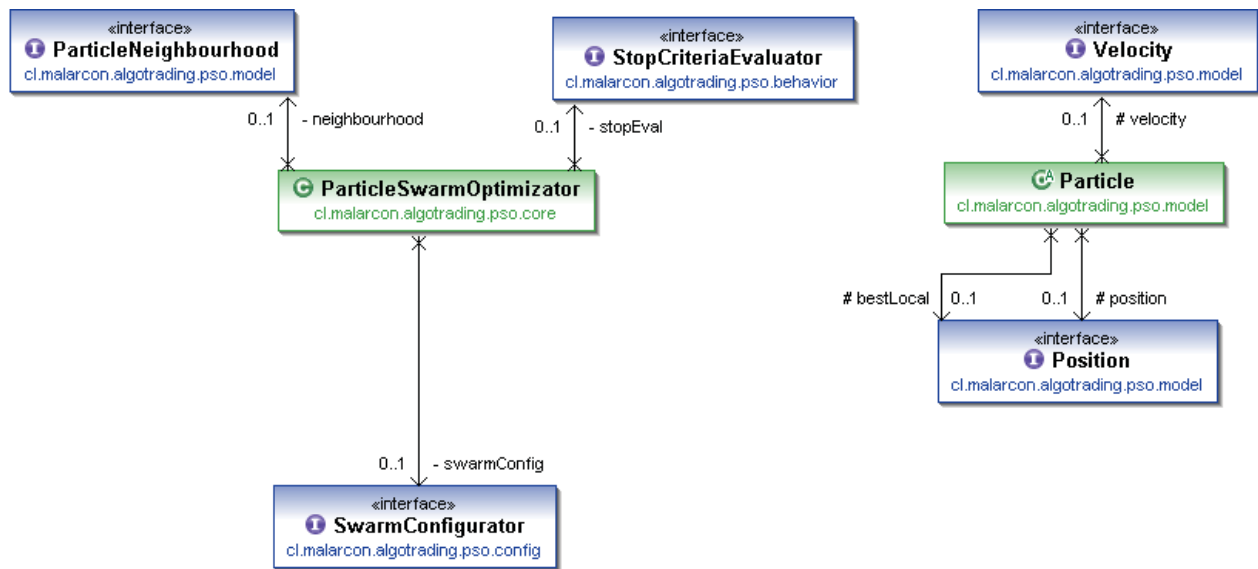
Siguiendo el diseño propuesto anteriormente, se ha procedido a construir un sistema modular que permita extensibilidad. Para implementar dicho sistema de trading automático se han empleado las siguientes herramientas:

- Motor central
  - Lenguaje de programación: Java 6 SE x64
  - Entorno de desarrollo: Eclipse 4.2.1 Juno x64
- Almacenamiento
  - Motor de Base de Datos: MS SQL Server 2008 SP1
  - Lenguaje de programación: MS Transact-SQL
  - Driver JDBC jTDS 1.5

### 4.5.1 Implementación de Particle Swam Optimization

Se han separado los módulos de implementación de PSO y del motor de trading automático. Cada implementación puede funcionar independientemente de la otra, pero requieren funcionar en conjunto para encontrar los parámetros óptimos para la estrategia de trading propuesta en la sección 4.4.1

El módulo de PSO se compone de la implementación central de la metaheurística, pero sin incluir elementos de un problema en particular.



**Figura 4.8** Modelo básico de implementación de PSO. Se tiene un optimizador central que funciona con cualquier problema que sea modelado mediante las interfaces expuestas.

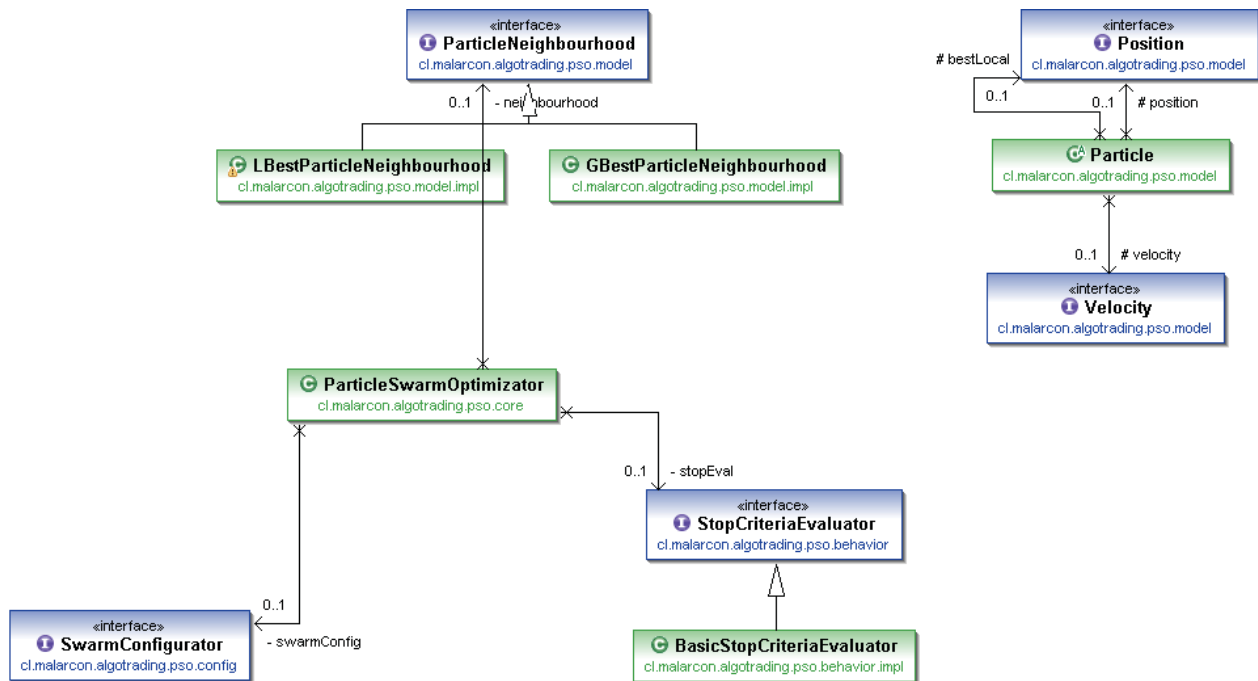
De este modo, el módulo se compone de un optimizador que requiere de 3 interfaces para su funcionamiento (figura 4.8). Las 3 interfaces se encargan de las siguientes tareas:

- *SwarmConfigurator*: las implementaciones de esta interfaz deben entregar un enjambre compuesto de partículas que extiendan a la clase abstracta *Particle*. De este modo, el configurador de enjambre debe crear la configuración inicial de las partículas para un problema en particular.
- *ParticleNeighbourhood*: consiste en la implementación de la función de vecindad, tal como se expone en la sección 3.2.1. Tiene la función de determinar cuál es la mejor partícula  $x_{i,swarm}^{k,best}$  dentro de los vecinos de una partícula dada, para que se puedan ejecutar los cálculos de velocidad y posición de las partículas del enjambre.
- *StopCriteriaEvaluator*: el optimizador requiere que se indique el mecanismo de detención del algoritmo. Las implementaciones de esta interfaz deben determinar al finalizar cada ciclo de optimización si se puede continuar con la ejecución. Para esto, se le entregan una serie de datos relevantes como el número de iteraciones realizado y el estado completo del enjambre, entre otro.

De forma conjunta, el optimizador requiere que se extienda a la implementación abstracta de la partícula, para adaptarla al problema en particular que se quiere optimizar. La clase *SwarmConfigurator* se encarga de instanciar la implementación requerida, junto con las implementaciones de las interfaces anexas. Esto corresponde a:

- *Particle*: clase abstracta de una partícula, que contiene a su posición y función de velocidad como interfaces. Almacena la mejor solución local encontrada para efectos de cálculo de velocidad. Las subclases que la extienden deben implementar un método que genere el valor de la función objetivo, junto con implementar un método que permita compararse contra otra partícula mediante el valor de su función objetivo, para determinar quien tiene un mejor valor. Para efectos prácticos, se utiliza el criterio de que si una partícula es mejor que otra si tiene un valor más alto para la función objetivo.
- *Position*: interfaz que representa la posición de una partícula, que corresponde a una de las soluciones para el problema. Las clases que la implementan deben ser capaces de calcular la distancia hacia otra posición, para poder crear distintas topologías de vecindad. Además deben aceptar una implementación de la interfaz *Velocity* para aplicarla sobre sus valores actuales, generando una nueva posición.
- *Velocity*: interfaz que representa la función de velocidad de una partícula. Las implementaciones almacenan las componentes de movimiento calculadas por su propia función de velocidad. Estas componentes de movimiento son luego aplicadas a una implementación de *Position* por parte de una partícula.

En primera instancia, se ha realizado al menos una implementación de las interfaces y clases abstractas presentadas, para poder resolver el problema de trading automático.



**Figura 4.9** Modelo completo de implementación de PSO. No incluye implementaciones propias del problema a resolver, dado que estas se implementan en el trader automático. Se cuenta con implementaciones genéricas de la función de vecindad (Local y Global Best) y del criterio de parada.

En la figura 4.9 se muestran las implementaciones de las interfaces de vecindad y criterio de parada. Las implementaciones del configurador del enjambre, partícula, función de velocidad y representación de posición se encuentran dentro del trader automático que se expone en la sección siguiente. Los detalles sobre cada implementación son:

- *GBestParticleNeighbourhood*: función de vecindad global, donde se busca la mejor partícula entre el conjunto completo del enjambre. Es la más simple y rápida, dado que sólo requiere encontrar la partícula que maximiza la función objetivo para una iteración.
- *LBestParticleNeighbourhood*: función de vecindad local, donde se buscan las 2 partículas más cercanas a una partícula dada, y se escoge la mejor de las 3 partículas. Esto es más lento porque requiere realizar cálculos de distancia entre todas las partículas para encontrar las más cercanas las unas a las otras.

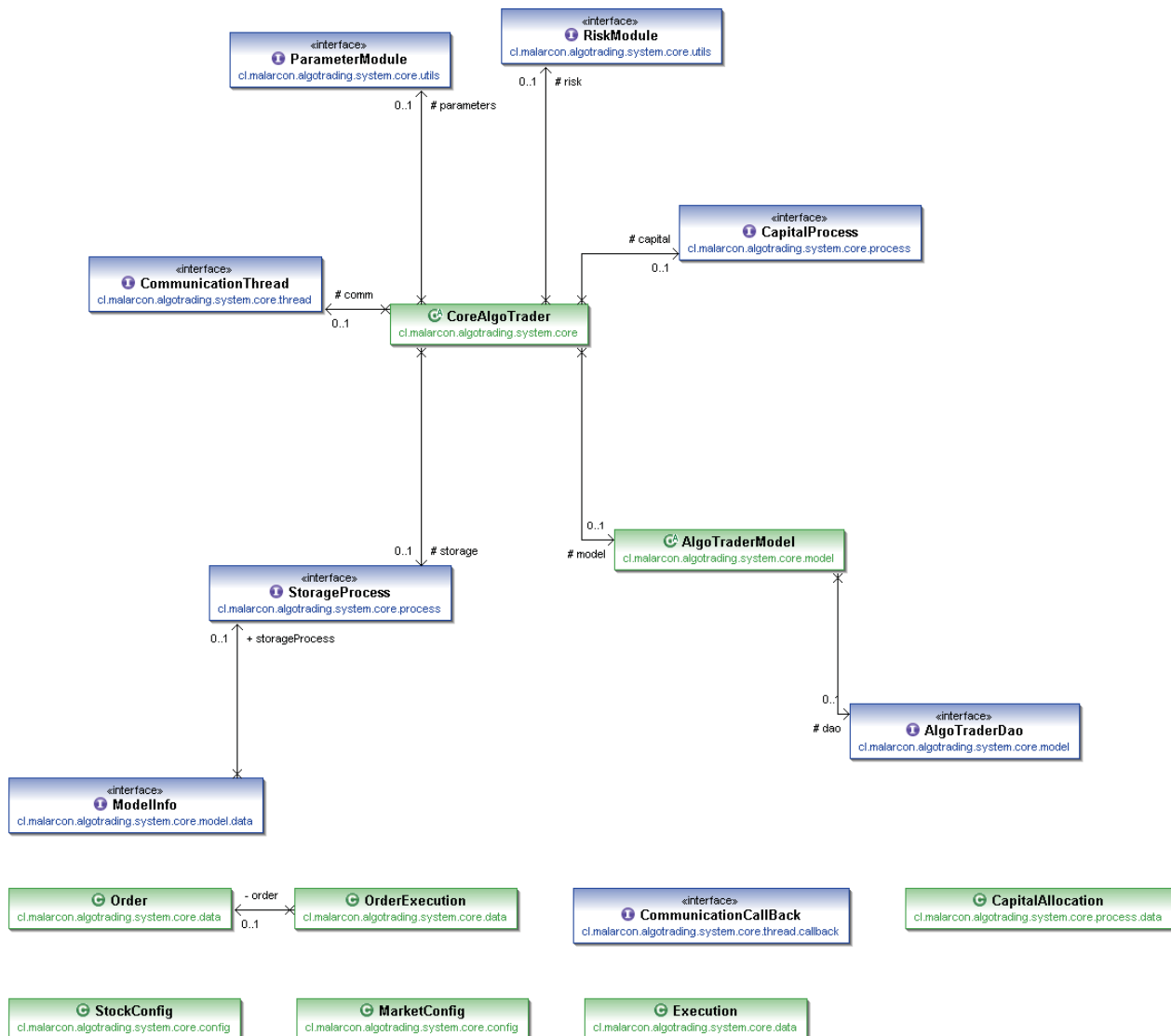
- *BasicStopCriteriaEvaluator*: criterio de detención basado en el número de iteraciones realizadas. Una vez alcanzado el número de iteraciones designado, el algoritmo PSO se detiene. Una vez detenido se obtiene la mejor solución aproximada del óptimo global.

#### 4.5.2 Implementación de Trader Automático

Para la implementación del motor de trading automático, se sigue el diseño propuesto en la sección 4.1, donde se tiene un módulo central que realiza toda la coordinación necesaria para procesar la información relativa a un instrumento financiero mediante módulos anexos especializados en tareas específicas.

El módulo central corresponde a una clase abstracta de la lógica de trading automático, generalizado a cualquier tipo de mercado bursátil (Acciones Renta Variable, Renta Fija, etc.), que mediante una serie de interfaces accede a los módulos especializados en el mercado objetivo. Este módulo central funciona a intervalos regulares (*ticks*), donde durante cada ejecución del intervalo evalúa su algoritmo interno de trading. Si se genera una señal de compra/venta, procede a utilizar a los módulos correspondientes para poder realizar la operación. El valor del tick determina si el sistema se comporta como un sistema *High Frequency Trading* o como un sistema *Algorithmic Trading*.

Para su correcto funcionamiento, la clase central del sistema debe ser extendida para generar la funcionalidad necesaria para un tipo de trader específico. Se requieren al menos 2 tipos de traders: uno de simulación del mercado y otro que se comunique con el mercado real.



**Figura 4.10** Modelo básico de implementación de AT. El módulo central de trading automático soporta cualquier tipo de mercado mediante el uso de sus interfaces genéricas.

El trader automático requiere de los mismos módulos definidos en la sección 4.1. En la figura 4.10 se detalla el modelo básico de AT y clases básicas anexas. La relación con los módulos de la sección 4.1 y el detalle de las tareas que realiza cada interfaz y clase se muestra a continuación:

- *CoreAlgoTrader*: núcleo central del sistema AT. Provee la implementación genérica del ciclo de trading, sin implementar una estrategia en particular. Al ser una clase abstracta, se debe extender para poder usarla.

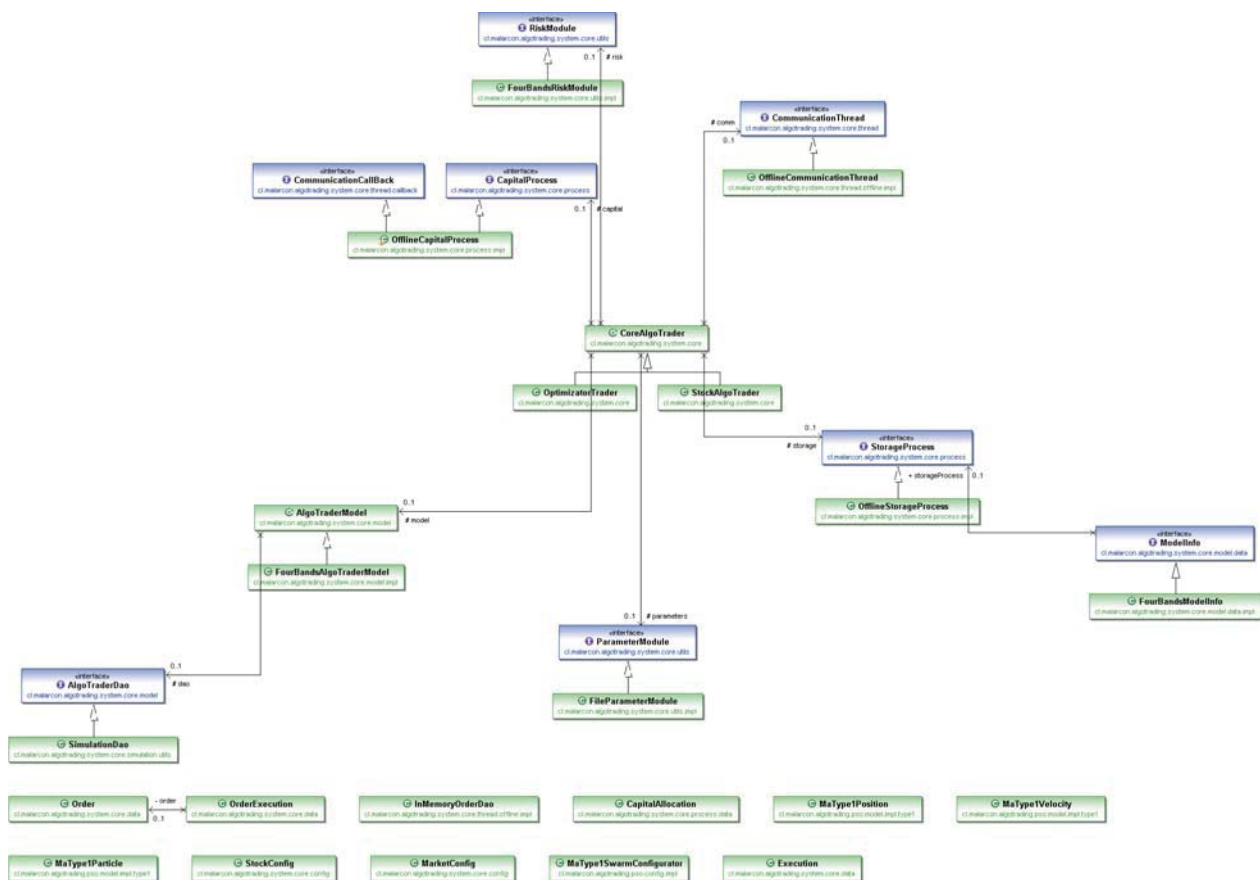


- *StorageProcess*: interfaz que representa al módulo “Almacenamiento”.
- *ParameterModule*: interfaz que representa al módulo “Parámetros”.
- *CommunicationThread*: interfaz que representa al módulo “Cliente Comunicaciones”.
- *RiskModule*: interfaz que representa al módulo “Análisis de Riesgo”.
- *CapitalProcess*: interfaz que representa al módulo “Manejo Capital”.
- *AlgoTraderModel*: clase abstracta que representa al modelo de decisión de compra/venta. Tiene un objeto de acceso a datos (DAO – Data Access Object) para obtener información adicional a la que la implementación de *CoreAlgoTrader* le entrega. Las clases que extienden al modelo de decisión deben implementar un método de evaluación del mercado, para determinar si se debe iniciar el proceso de compra/venta en ese instante.
- *AlgoTraderDao*: interfaz utilitaria para *AlgoTraderModel*. Su finalidad es dar acceso a información adicional que no es provista por el núcleo del sistema durante un ciclo normal de operación.
- *ModelInfo*: interfaz que representa a la configuración del modelo de trading. Esta configuración es mantenida por el proceso de Almacenamiento representado por *StorageProcess* y es utilizada por el modelo de trading representado por *AlgoTraderModel*.
- *CommunicationCallback*: interfaz requerida para el funcionamiento de cualquier parte del sistema que necesite enterarse de ciertos mensajes recibidos por el módulo “Cliente Comunicaciones” representado por *CommunicationThread*. Las clases que lo implementan pueden ser notificadas cuando ocurren ciertos eventos, como por ejemplo la ejecución de una orden enviada por el sistema AT.

- *CapitalAllocation*: representación de una reserva de capital, cuando el módulo central determina que se puede crear una operación bursátil y solicita al módulo de “Manejo Capital” (ver figura 4.4, sección 4.5) una cantidad de efectivo (compras) o custodia de un instrumento específico (ventas).
- *Order*: representa una orden creada por el sistema AT. Soporta mercados simples como Acciones Renta Variable. Para mercados donde la orden requiere más información que la provista, se puede extender.
- *OrderExecution*: representa la ejecución de una orden creada por el sistema AT en el mercado bursátil objetivo. Se utiliza para propagar información sobre la ejecución de la orden para que los módulos del sistema que se registren mediante la interfaz *CommunicationCallback* puedan tomar las acciones necesarias.
- *Execution*: representa una ejecución cualquiera en el mercado bursátil. Se usa en algunos modelos, como el modelo de 4 bandas propuesto, para realizar algunos cálculos internos (p.ej.: cálculo de Moving Averages).
- *StockConfig*: configuración básica específica para un instrumento bursátil. Soporta los datos mínimos que se requieren para cualquier mercado (comisiones e impuestos específicos). En caso de operar en mercados que requieran una configuración más amplia para un instrumento bursátil, se puede extender.
- *MarketConfig*: configuración básica específica para un mercado bursátil. Soporta los datos mínimos que se requieren para cualquier mercado (horarios de funcionamiento). En caso de operar en mercados que requieran una configuración más amplia, se puede extender.

El conjunto de clases e interfaces especificados sólo entregan el entorno de trabajo mínimo para generar un sistema AT que opere para un instrumento y mercado específico. Para funcionar con múltiples instrumentos del mismo mercado, se pueden instanciar múltiples copias del núcleo AT, cada una dedicada a un instrumento en particular, pero compartiendo los módulos comunes.

Para la versión inicial del sistema AT se crean las implementaciones de las interfaces para el motor de simulación requerido por el algoritmo PSO. Esta implementación simula un período extenso del mercado mediante los datos cargados a una base de datos MS SQL Server (ver sección 4.3, uso de ETL). De esta forma se puede calcular la rentabilidad máxima que genera un conjunto de parámetros del modelo de trading propuesto para el período seleccionado.



**Figura 4.11** Modelo completo de la implementación de AT. Se implementan las interfaces para crear un simulador de mercado y se genera la representación del modelo de trading de 4 bandas extendiendo el modelo de partículas de PSO.

El simulador implementa la estrategia de negociación de 4 bandas propuesto anteriormente, para lo cual requiere extender el modelo genérico de partículas del motor PSO mostrado en la sección 4.5.1. En la figura 4.11 se muestran las clases que representan dicha implementación, junto con las clases que representan a las interfaces requeridas por el sistema AT central. Con respecto al modelo anterior, se agregan las siguientes clases:

- *OptimizerTrader*: extensión del núcleo central del sistema AT. Referencia a las implementaciones de las interfaces requeridas por el sistema, junto con definir un ciclo de trading basado en la estrategia de mantener una única posición (vender todo antes de volver a comprar). Simula un mercado basándose en la implementación del módulo de comunicaciones. En cuanto al ciclo de funcionamiento detallado en la sección 4.4, este trader no realiza el ciclo de reevaluación, ya que dicha tarea la ejecuta el algoritmo PSO.
- *OfflineStorageProcess*: interfaz que representa al módulo “Almacenamiento”. No tiene lógica más allá de entregar siempre la configuración actual de la partícula para el modelo de trading de 4 bandas.
- *FileParameterModule*: interfaz que representa al módulo “Parámetros”. Obtiene todos los parámetros de forma estática mediante un archivo de configuración general.
- *OfflineCommunicationThread*: interfaz que representa al módulo “Cliente Comunicaciones”. Simula el comportamiento del mercado mediante los datos de Market Data cargados en una base de datos local.
- *FourBandsRiskModule*: interfaz que representa al módulo “Análisis de Riesgo”. No realiza análisis cuando el modelo de trading es un modelo de 4 bandas, dado que el manejo de riesgo está implícito en las bandas de *StopLoss* y *StopWin*.
- *OfflineCapitalProcess*: interfaz que representa al módulo “Manejo Capital”. Maneja en memoria el capital en base a la configuración entregada por la implementación de *ParameterModule*. Además implementa la interfaz *CommunicationCallBack* para recibir eventos de ejecución de órdenes generadas por el sistema, para actualizar las custodias y el capital de la simulación.
- *FourBandsAlgoTraderModel*: extensión de *AlgoTraderModel* que representa al modelo de decisión de compra/venta mediante el modelo de 4 bandas. Se encarga de calcular los promedios móviles para determinar la tendencia del precio del instrumento bursátil. En el caso de las ventas aplica además las bandas de control *StopLoss* y *StopWin*, para evitar pérdidas o generar ganancias, respectivamente.

- *SimulationDao*: implementación de *AlgoTraderDao*. Calcula los Moving Averages necesarios para evaluar el modelo de 4 bandas. Accede mediante procedimientos almacenados a los datos de Market Data para generar la respuesta. También se usa para determinar qué día hábil siguiente tiene un instrumento dado para una fecha particular.
- *FourBandsModelInfo*: implementación de *ModelInfo*. Almacena la configuración actual de la partícula dentro de un ciclo de PSO, junto con datos adicionales requeridos por el algoritmo de trading.

Junto con las implementaciones mencionadas, existen además algunas clases que extienden el modelo genérico del motor PSO presentado en 4.5.1. Estas clases son el nexo que permite que el algoritmo de optimización pueda ser utilizado por el sistema AT. De este modo, las clases generadas para el uso de PSO son:

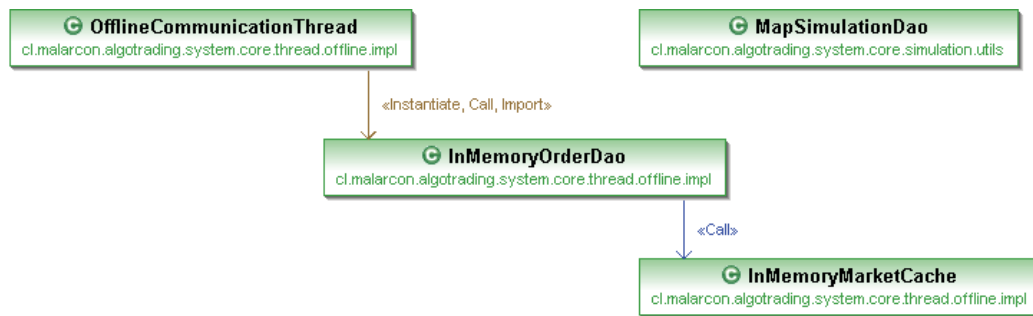
- *MaTypeISwarmConfigurator*: implementación de la interfaz *SwarmConfigurator*. Genera un enjambre de partículas *MaTypeIParticle* usando el mismo archivo de configuración que emplea *FileParameterModule*. En este caso la configuración inicial de la partícula la genera la partícula misma y no el configurador de enjambre.
- *MaTypeIParticle*: extensión de la clase abstracta *Particle*. Genera una solución inicial aleatoria para los valores de las 4 bandas, respetando las restricciones del modelo. Los valores mínimos y máximos se mantienen dentro de los límites configurados en el archivo de configuración general. Para evaluar la función objetivo empleada en PSO, genera un trader de simulación mediante la clase *SimulatorTrader*, entregándole la posición actual de la partícula como información del modelo de trading. Al finalizar la ejecución del simulador, obtiene la rentabilidad neta para usarse como medida de comparación con su mejor posición encontrada y contra el óptimo de la función de vecindad. En la implementación no se ha incluido la funcionalidad de desempate utilizando ratios financieros.
- *MaTypeIPosition*: implementación de la interfaz *Position*. Almacena el modelo de trading de 4 bandas usado por la partícula y permite el cálculo de distancia euclidiana con respecto a otra posición. También puede recibir una instancia de *MaTypeIVelocity* para calcular la siguiente posición.
- *MaTypeIVelocity*: implementación de la interfaz *Velocity*. Calcula la velocidad en 4 dimensiones dentro del modelo de trading, empleando la fórmula estándar de velocidad de PSO detallada en la sección 3.2.1.

El conjunto completo del sistema genérico PSO y el motor AT se combinan mediante un sistema simple de arranque para poder realizar las pruebas de laboratorio con los datos recopilados en Market Maker. El sistema de arranque se configura en base a un archivo de texto y un parámetro que indica cual nemotécnico se quiere ingresar al optimizador.

En base a las pruebas de laboratorio obtenidas (ver sección 4.6 más adelante), se han determinado una serie de mejoras para implementar al sistema, generando una versión optimizada para rendimiento. La finalidad de esto es que el proceso de optimización de la solución mediante PSO debe converger lo suficientemente rápido para ser ejecutado múltiples veces durante un día de operación bursátil.

Las mejoras aplicadas responden a los siguientes problemas:

- Cálculo de *Moving Averages*: la versión inicial del sistema AT invoca a la rutina de cálculo de *Moving Averages* por cada instante de operación del sistema, de forma independiente para cada partícula. Se cambia el cálculo puntual por un cálculo incremental, basado en el valor del período de tiempo anterior y el nuevo (ver sección 3.1), junto con calcular la totalidad de instantes requeridos por la ejecución de una partícula.
- Lectura de *Moving Averages*: siguiendo con el problema anterior, si para un instante dado ya se había calculado un *Moving Averages* con el mismo largo, este se recalculaba de todas formas. Esto introduce una sobrecarga al proceso de *Almacenamiento*, que debe volver a calcular el mismo valor. Para solucionarlo, se emplea un sistema de caché en memoria, que permite que los valores específicos se calculen una única vez, pero que se puedan consultar múltiples veces de manera eficiente.
- Lectura de Ejecuciones de Mercado: similar al problema anterior, esto responde a como se encuentra implementado otro de los módulos del sistema AT. En particular, el problema se encuentra en la rutina de simulación de mercado presente en *OfflineCommunicationThread*. Esta rutina se basa en la información histórica de ejecuciones de órdenes (calces), por lo cual debe cargarlos desde un sistema de almacenamiento (base de datos). En la primera implementación, cada partícula cargaba nuevamente los mismos datos desde la base de datos, por cada iteración de PSO. Se soluciona empleando una caché compartida de calces, utilizada por todas las partículas en todas sus iteraciones.



**Figura 4.12** Extensión del sistema AT para soportar optimizaciones planteadas. Se agregan 2 clases nuevas, *InMemoryCache* y *MapSimulationDao*, encargados de reducir considerablemente los tiempos de ejecución del algoritmo PSO.

En la figura 4.12 se pueden ver las clases nuevas que se debieron agregar a la segunda versión del sistema AT.

La clase *MapSimulationDao* responde a las mejoras relacionadas con el cálculo y lectura de los *Moving Averages*. Esta clase realiza el cálculo de una manera más eficiente, generando el  $MA(n)$  para  $t=0$ , para luego ir simplemente usar la fórmula iterativa de MA para todo  $t > 0$ . Esto se calcula para el período completo de la simulación de mercado y se almacena en una caché en memoria, de rápido acceso.

La clase *InMemoryMarketCache* responde a la mejora de lectura de calces bursátiles. En vez de solicitar los calces de un día de operación en particular, de forma separada por cada partícula, esta clase lee todos los calces del mercado para un instrumento y período dado y los almacena en una caché en memoria. Luego se le entrega una copia de la caché a cada partícula.

## 4.6 Pruebas del Sistema

Se han realizado 2 experimentos con el sistema en su versión inicial, bajo el siguiente entorno de ejecución local:

- Sistema Operativo: Windows 7 Ultimate x64 SP1
- Memoria RAM: 16 GB DDR3
- Procesador: Intel Core i7-3610QM 2.3 GHz, 4 núcleos, 8 hebras
- Versión Java: Oracle JDK 1.6.0 Rev37 x64
- Versión SQL Server: MS SQL Server 2008 x64 SP1
- Market Data: 2010/06 al 2012/06, inclusive

Para los experimentos se han usado los siguientes datos fijos en la configuración:

- Modalidad AT: tomar y dejar posición única
- Volumen de la orden: \$500.000
- Costos variables: 0 % (simulación de una corredora)
- Costos fijos: \$0 (simulación de una corredora)
- Impuestos: 0 % (instrumento de alta presencia bursátil, por lo cual no se cobra impuesto a las ganancias)

Para la configuración de los parámetros de PSO, se utilizan los valores indicados en [Jabri2011], que para el sistema AT con modelo de 4 bandas corresponde a:

- Tamaño del enjambre: 12 partículas
- $c_0$ : 0,72134
- $c_1$ : 1,19314
- $c_2$ : 1,19314
- Núm. Iteraciones: 30 (este valor se decide arbitrariamente)



#### 4.6.1 Pruebas versión inicial

El primer experimento es para revisar si el sistema funciona correctamente y es capaz de generar rentabilidades positivas. Como valor específico del experimento se tiene:

- Período: 4 meses (Enero-Abril 2012)
- Instrumento: LAN
- Tick: 5 minutos

Se escoge el período Enero-Abril 2012 porque en dicho período LAN tiene tanto alzas y bajas del precio de la acción. Si se escogiera un período con únicamente alzas, no se puede probar el mecanismo de manejo de riesgo que ofrece la banda de *StopLoss*. En cambio, si se usa únicamente un período con bajas, el sistema no realizará la toma de posición (compra inicial), por lo que se mantendrá inactivo hasta que aparezca un período de alza.

Los resultados del experimento 1 son:

- Duración: 3,44 horas (12.388.202 ms)
- Retornos netos:  $> 0$
- MA Corto: 30
- MA Largo: 58
- StopLoss: 4,0543
- StopWin: 11,3177

El experimento indica que el proceso consume una gran cantidad de tiempo, debido a la cantidad de iteraciones realizadas y al tamaño del enjambre. Lo positivo de los resultados es que hay ganancias al finalizar el proceso, obteniendo los parámetros para configurar un trader que opere dentro de un plazo razonablemente cercano al período de finalización de la optimización. Los retornos de todos modos son inconsistentes debido a un error en la primera versión del sistema AT, por lo cual no se presentan aquí.

Al revisar los valores de las bandas de *StopLoss* y *StopWin* se descubre un problema. Los valores escapan a los máximos definidos en el modelo, ya que se aplica la fórmula de velocidad de PSO estándar. Esto indica que se debe realizar un ajuste a la implementación de la fórmula antes de seguir con los experimentos finales. Además se descubre un error de cálculo en la implementación del sistema.

El segundo experimento es para revisar el comportamiento y el tiempo de ejecución para un período más corto. El valor de entrada corresponde a:

- Período: 2 meses (Enero-Febrero 2012)
- Instrumento: LAN
- Tick: 5 minutos

El experimento se aborta a mitad del proceso, debido a que se generan valores negativos para las posiciones de las partículas. Esto recalca el hecho de que la implementación estándar de la función de velocidad no se puede aplicar al modelo AT, por lo cual se debe adaptar.

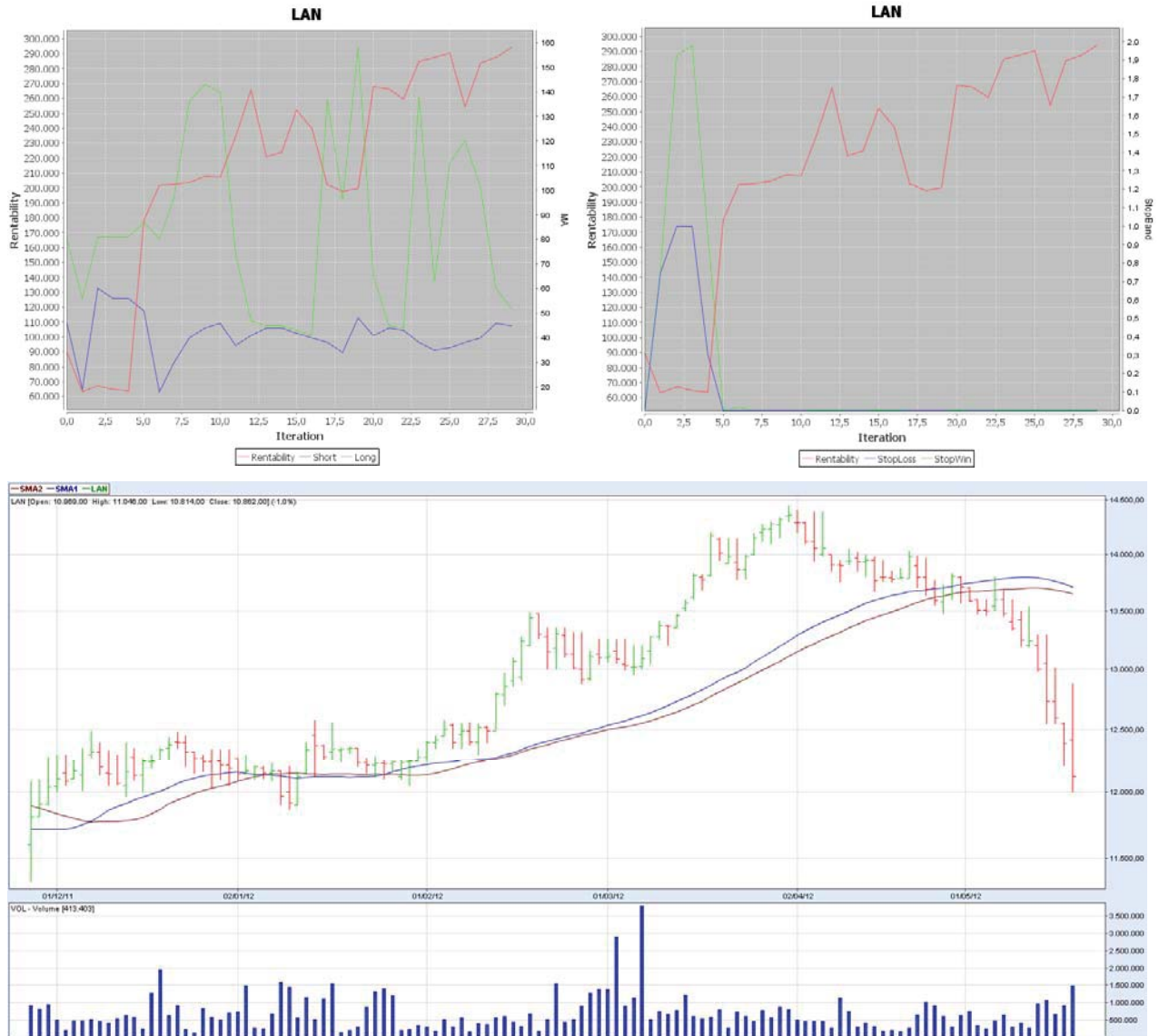
#### 4.6.2 Pruebas versión optimizada

Tal como se mencionó en la sección 4.5.2, se generó una versión optimizada del sistema AT, que además implementa una serie de correcciones a errores detectados en la versión inicial del sistema. Se repite el experimento anterior, para determinar el nivel de mejora introducido al sistema. De este modo, los valores específicos del experimento 3 son:

- Período: 4 meses (Enero-Abril 2012)
- Instrumento: LAN
- Tick: 5 minutos

Los resultados del experimento 3 son:

- Duración: 1 minuto 26 segundos (85.950 ms)
- Retornos netos: \$294.186 (Rentabilidad 69,96%)
- MA Corto: 45
- MA Largo: 52
- StopLoss: 0
- StopWin: 0, 0040370020874758985



**Figura 4.13** Variación de los parámetros y rentabilidades asociadas a la mejor partícula de la mejor iteración del experimento 3, junto con las curvas de MA obtenidas al final. Se puede observar como los valores del largo de los MA y de las bandas de StopLoss y StopWin afectan a la rentabilidad neta del sistema.

El experimento resulta exitoso, permitiendo demostrar que el sistema AT funciona correctamente dentro de un plazo de tiempo razonable, y además es capaz de encontrar los parámetros que permitan obtener una rentabilidad positiva para el período analizado. Al revisar los gráficos de la figura 4.13, se puede apreciar que el período sobre el que se aplicó la optimización de parámetros del modelo AT presenta una tendencia creciente, lo que explica el alto nivel de rentabilidad final.

Dada esta condición creciente, el algoritmo PSO determinó que, bajo las condiciones de los *Moving Averages* determinados, es más conveniente realizar una cantidad elevada de operaciones de compra/venta, reflejado por una banda *StopWin* muy cercana a cero. Además, el algoritmo determinó que es más conveniente emplear un riesgo cero, de modo tal de disminuir las pérdidas.

| Tiempo (ms)  | Retornos (CLP) | Short | Long | StopLoss    | StopWin     |
|--------------|----------------|-------|------|-------------|-------------|
| 78892        | 382304         | 42    | 48   | 0,004156852 | 0,00001     |
| 77313        | 570680         | 16    | 37   | 0,474434187 | 0,457211905 |
| 80153        | 422715         | 77    | 78   | 0,003876797 | 0,00001     |
| 102842       | 327270         | 101   | 128  | 0,008426064 | 0,00001     |
| 88634        | <b>690757</b>  | 47    | 90   | 0           | 0,00001     |
| 94216        | 351071         | 70    | 71   | 0,007386603 | 0,00001     |
| 87911        | 640351         | 69    | 74   | 0           | 0,00001     |
| 78439        | 647545         | 35    | 39   | 0           | 0,00001     |
| 79648        | 381706         | 33    | 34   | 0,003689378 | 0,00001     |
| 79306        | 404035         | 1     | 33   | 0,00257833  | 0,00001     |
| 78836        | 349468         | 47    | 48   | 0,007894228 | 0,00001     |
| 84482        | 384969         | 60    | 106  | 0,003625605 | 0,00001     |
| 77942        | 346555         | 78    | 79   | 0,007217344 | 0,00001     |
| 94406        | 373798         | 1     | 170  | 0,004226087 | 0,00001     |
| 82992        | 363428         | 1     | 32   | 0           | 0,00001     |
| 89537        | 355787         | 126   | 127  | 0,003797629 | 0,00001     |
| <b>75598</b> | 370568         | 13    | 49   | 0,003803033 | 0,00001     |
| 76913        | 651456         | 20    | 28   | 0           | 0,00001     |
| 83877        | 386283         | 32    | 37   | 0,007186894 | 0,00001     |
| 93201        | 376666         | 3     | 150  | 0,00367221  | 0,00001     |

**Tabla 4.4** Resultados de 20 ejecuciones del modelo AT del experimento 3. El mejor tiempo es de apenas 75.598 ms, mientras que el mejor retorno neto teórico es de 690.757 CLP (rentabilidad de 138%). Tales ganancias se deben a la tendencia creciente para el período del experimento, por lo que los parámetros son ajustados de forma acorde por el algoritmo PSO.

Finalmente, el experimento es ejecutado 20 veces para determinar los mejores y peores tiempos, junto con el mejor retorno neto teórico. De esta forma, el mejor tiempo de ejecución es de 75.598 ms, mientras que el peor es de 102.842 ms, ambos valores muy por debajo de los tiempos de la versión inicial. El tiempo promedio de ejecución es de 84.259,4 ms. Las rentabilidades de la mejor partícula al final de cada ejecución de PSO oscilan en un promedio de 438.870,6 CLP, destacándose una iteración en particular donde se encontró una combinación de parámetros que logra una rentabilidad del 138% sobre la primera inversión.

El experimento se repite, variando el tamaño del *tick*. Esto permite ver el comportamiento del sistema AT en una modalidad más cercana a *High Frequency Trading*. Los valores específicos del experimento 4 son:

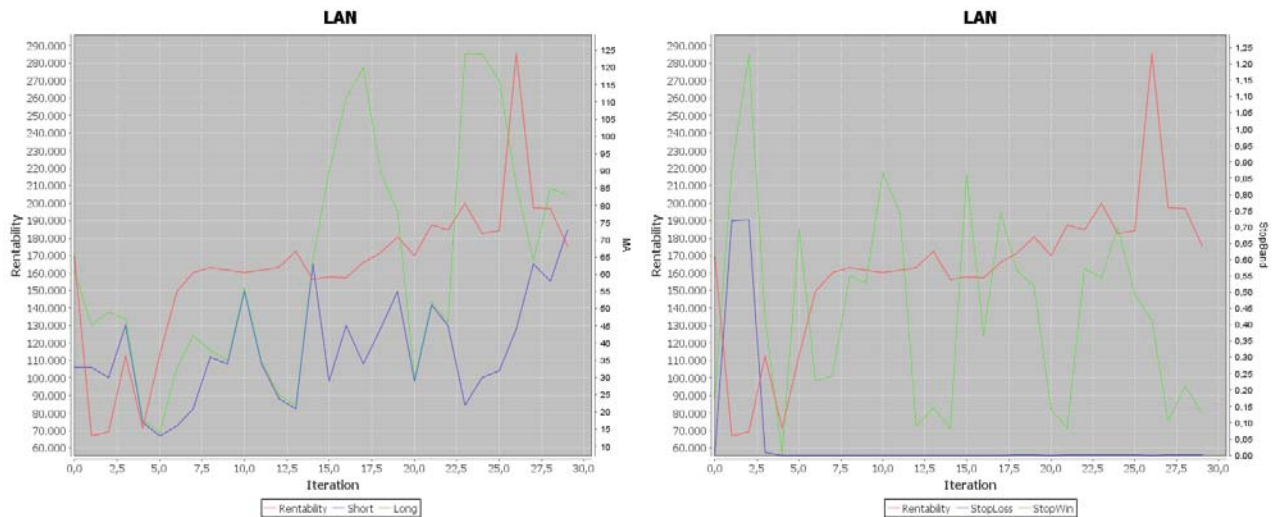
- Período: 4 meses (Enero-Abril 2012)
- Instrumento: LAN
- Tick: 1 minuto

| Tiempo        | Retornos      | Short | Long | StopLoss    | StopWin     |
|---------------|---------------|-------|------|-------------|-------------|
| 401775        | 271096        | 32    | 34   | 0           | 0,285258866 |
| 380216        | 174172        | 36    | 37   | 0           | 0,160024324 |
| 386456        | 174940        | 8     | 18   | 0           | 1,728610269 |
| 389235        | 176805        | 26    | 27   | 0           | 0,090294426 |
| 399316        | 235421        | 32    | 34   | 0           | 0,029627225 |
| 401022        | 162707        | 46    | 110  | 0           | 0,129552719 |
| 390692        | 175168        | 77    | 78   | 0           | 1,195936443 |
| 388487        | 205896        | 50    | 70   | 0,000384208 | 0,382646993 |
| 382833        | 273143        | 30    | 36   | 0           | 0,247501213 |
| 382123        | 169675        | 36    | 37   | 0           | 0,109817783 |
| 375669        | 271096        | 32    | 34   | 0           | 0,382314236 |
| 375810        | 216337        | 41    | 76   | 0,028521942 | 0,043130692 |
| 385236        | 247521        | 16    | 21   | 0           | 1,373017833 |
| 387552        | <b>285351</b> | 44    | 86   | 0,000293338 | 0,414043995 |
| 394574        | 195570        | 76    | 96   | 0,002536985 | 0,801816572 |
| 389934        | 268196        | 57    | 84   | 0           | 0,702944475 |
| 361461        | 268196        | 48    | 88   | 0           | 0,316077957 |
| <b>320494</b> | 242775        | 29    | 78   | 0,028291729 | 0,029311713 |
| 325625        | 273143        | 30    | 36   | 0           | 1,591565877 |
| 323076        | 206525        | 26    | 39   | 0,000372256 | 0,404751387 |

**Tabla 4.5** Resultados de 20 ejecuciones del modelo AT del experimento 4. El mejor tiempo aumenta a 320.494 ms, mientras que el mejor retorno neto teórico disminuye a 285.351 CLP (rentabilidad de 57%). La diferencia en el comportamiento es dada por la granularidad temporal sobre la que opera el sistema.

Al revisar los resultados del experimento en la tabla 4.5, resulta notorio el aumento de los tiempos de ejecución (a un promedio de 377.079 ms, equivalentes a 6 minutos y 17 segundos) y la disminución del retorno neto teórico. Esto es causado principalmente por un aumento en la banda de *StopWin*, que es la que permite generar ganancias en un período de aumento del precio. En muchos de los mejores resultados también se calcula una banda de *StopLoss* mayor a cero, indicando que el sistema AT aceptará cierto nivel de riesgo para generar ganancias. Este

comportamiento puede parecer desfavorable en un período de crecimiento sostenido del precio, pero puede resultar ventajoso cuando hay variación de precio en lapsos muy cortos.



**Figura 4.14** Variación de los parámetros y rentabilidades asociadas a la mejor partícula de la mejor iteración del experimento 4.

En la figura 4.14 se puede apreciar el comportamiento de las variables del modelo AT y como afectan la rentabilidad en el experimento 4. El comportamiento es similar al experimento anterior, generando valores para los MA parecidos para la mejor solución.

Se repiten los experimentos anteriores, pero cambiando el instrumento por SQM-B (Sociedad Química y Minera de Chile S.A, serie B). Se selecciona este instrumento por pertenecer al IPSA, lo cual indica que tiene alta presencia, y además por presentar un comportamiento fluctuante durante el período seleccionado para los experimentos.

De este modo, el experimento 5 tiene los siguientes parámetros específicos:

- Período: 4 meses (Enero-Abril 2012)
- Instrumento: SQM-B
- Tick: 5 minutos

| Tiempo       | Retornos      | Short | Long | StopLoss    | StopWin     |
|--------------|---------------|-------|------|-------------|-------------|
| 80778        | 248688        | 1     | 61   | 0,99999     | 2,872258817 |
| 77150        | 304209        | 57    | 58   | 0           | 0,01090185  |
| 77137        | 412966        | 130   | 131  | 0           | 0,00001     |
| 94645        | 274550        | 15    | 202  | 0,99999     | 2,278211743 |
| 76035        | 270051        | 10    | 26   | 0           | 1,423676104 |
| 73495        | 415659        | 1     | 53   | 0           | 0,00001     |
| <b>70561</b> | 76110         | 53    | 54   | 0           | 9,023721169 |
| 77738        | 347457        | 58    | 77   | 0           | 0,010769257 |
| 76080        | 65777         | 55    | 77   | 0           | 0,899867149 |
| 77107        | <b>416840</b> | 3     | 61   | 0           | 5,398129836 |
| 72527        | 415659        | 1     | 53   | 0           | 0,00001     |
| 77047        | 76081         | 46    | 52   | 0           | 0,905072667 |
| 79987        | 80145         | 46    | 53   | 0           | 0,702384999 |
| 87284        | 30268         | 174   | 175  | 0,971379989 | 1,644380018 |
| 76543        | 37323         | 58    | 78   | 0,99999     | 4,04671096  |
| 77557        | 34284         | 85    | 92   | 0,977945156 | 4,003760091 |
| 79173        | 34326         | 61    | 77   | 0,912030641 | 1,703732272 |
| 77281        | 415659        | 1     | 53   | 0           | 0,00001     |
| 75971        | 248688        | 1     | 61   | 0,513941717 | 5,096028441 |
| 78750        | 79268         | 44    | 56   | 0           | 1,473992199 |

**Tabla 4.6** Resultados de 20 ejecuciones del modelo AT del experimento 5. Los tiempos de ejecución son similares al experimento 3, pero los parámetros obtenidos son muy diferentes para los MA y la banda de *StopWin*.

Al revisar en la tabla 4.6 los resultados de 20 instancias del modelo AT optimizado mediante PSO, se puede apreciar que el algoritmo es capaz de encontrar combinaciones de parámetros capaces de generar buenos retornos netos teóricos, con una rentabilidad máxima de 83%. Para esto, las mejores rentabilidades se dan cuando el MA corto sigue de cerca la tendencia del precio de mercado, con un MA largo alejado que permite definir la tendencia a mediano plazo. Llama la atención que la mejor rentabilidad se da con una banda de *StopWin* altísima, mientras que la banda de *StopLoss* es similar al experimento 3. Para las siguientes combinaciones con retorno alto, la banda de *StopWin* se encuentra al mínimo. Esto indica que la solución encontrada puede ser demasiado arriesgada, ya que espera a que el precio suba considerablemente, generando pocas transacciones.



**Figura 4.15** Variación de los parámetros y rentabilidades asociadas a la mejor partícula de la mejor iteración del experimento 5, junto con las curvas de MA obtenidas al final. A diferencia del experimento 3, el MA corto sigue de cerca la tendencia del precio de mercado a cada instante, para verificar cuando el precio tiene una tendencia a corto plazo de subir o bajar con respecto al promedio de mediano alcance calculado por el MA largo.

Al revisar los gráficos de comportamiento de la mejor solución en la figura 4.15, se puede apreciar el ajuste cercano del MA corto al precio de mercado.

El experimento permite concluir que el sistema AT basado en PSO es capaz de descubrir los mejores parámetros para su uso inmediato para cualquier instrumento que tenga alta presencia y suficiente liquidez de mercado.



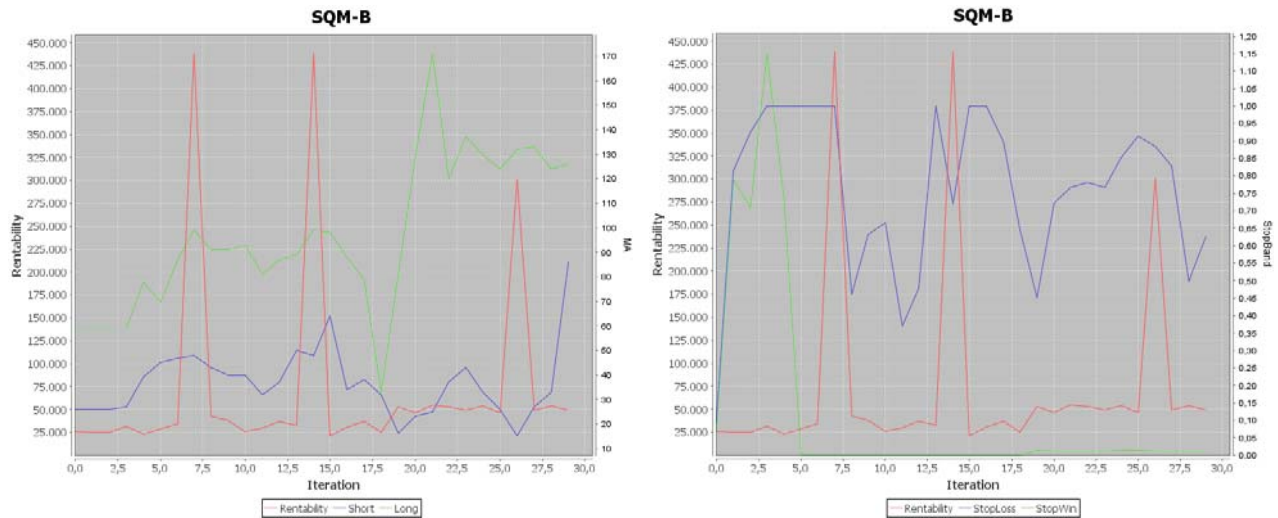
Para validar los resultados del experimento 4, se repite el mismo variando el instrumento. Nuevamente se selecciona SQM-B para efectos de comparación. Así, los datos específicos del experimento 6 son:

- Período: 4 meses (Enero-Abril 2012)
- Instrumento: SQM-B
- Tick: 1 minuto

| Tiempo        | Retornos      | Short | Long | StopLoss    | StopWin     |
|---------------|---------------|-------|------|-------------|-------------|
| 314825        | 32240         | 85    | 86   | 0,988819644 | 2,129892834 |
| 315688        | 421672        | 3     | 60   | 0,39902442  | 1,534515687 |
| 307654        | 36697         | 60    | 65   | 0,99999     | 0,00001     |
| <b>303770</b> | 110207        | 2     | 41   | 0,828671824 | 0,864597565 |
| 313627        | 32400         | 12    | 211  | 0,776228422 | 1,597603955 |
| 312537        | 41097         | 27    | 58   | 0,99999     | 1,286208793 |
| 320830        | 40540         | 63    | 67   | 0,99999     | 0,00001     |
| 314825        | 43080         | 67    | 71   | 0,99999     | 0,00001     |
| 314090        | 49998         | 13    | 14   | 0,99999     | 0,01077033  |
| 323050        | 54713         | 59    | 65   | 0,994091233 | 0,00001     |
| 310887        | 40540         | 63    | 67   | 0,694633223 | 0,00001     |
| 312032        | 45376         | 44    | 90   | 0,10559998  | 0,00001     |
| 319154        | 43080         | 67    | 71   | 0,737795553 | 0,00001     |
| 319578        | 274550        | 15    | 202  | 0,240875579 | 5,133725103 |
| 319242        | 50102         | 52    | 65   | 0,99999     | 0,018216571 |
| 316652        | 268394        | 15    | 67   | 0,99999     | 2,592752562 |
| 311474        | <b>438116</b> | 48    | 99   | 0,99999     | 0,00001     |
| 315581        | 35051         | 58    | 59   | 0,99999     | 0,00001     |
| 312363        | 54713         | 59    | 65   | 0,980335201 | 0,00001     |
| 312147        | 45388         | 55    | 69   | 0,992096381 | 0,015007295 |

**Tabla 4.7** Resultados de 20 ejecuciones del modelo AT del experimento 6. Los tiempos de ejecución son similares al experimento 4, donde la diferencia se da principalmente por un instante de menor carga del equipo de pruebas. Las rentabilidades más altas son mayores al experimento anterior.

Los resultados en este caso indican que la optimización mediante PSO encontró más ventajoso utilizar una modalidad HFT simple, obteniendo una ganancia mínima en cada operación y manteniendo las posiciones a pérdida el mayor tiempo posible. Para la segunda mejor rentabilidad obtenida del experimento, los parámetros de MA son parecidos al experimento anterior, pero al mismo tiempo disminuyendo la aversión al riesgo (aumenta el valor de *StopLoss*). Los tiempos de ejecución son similares al experimento 4.



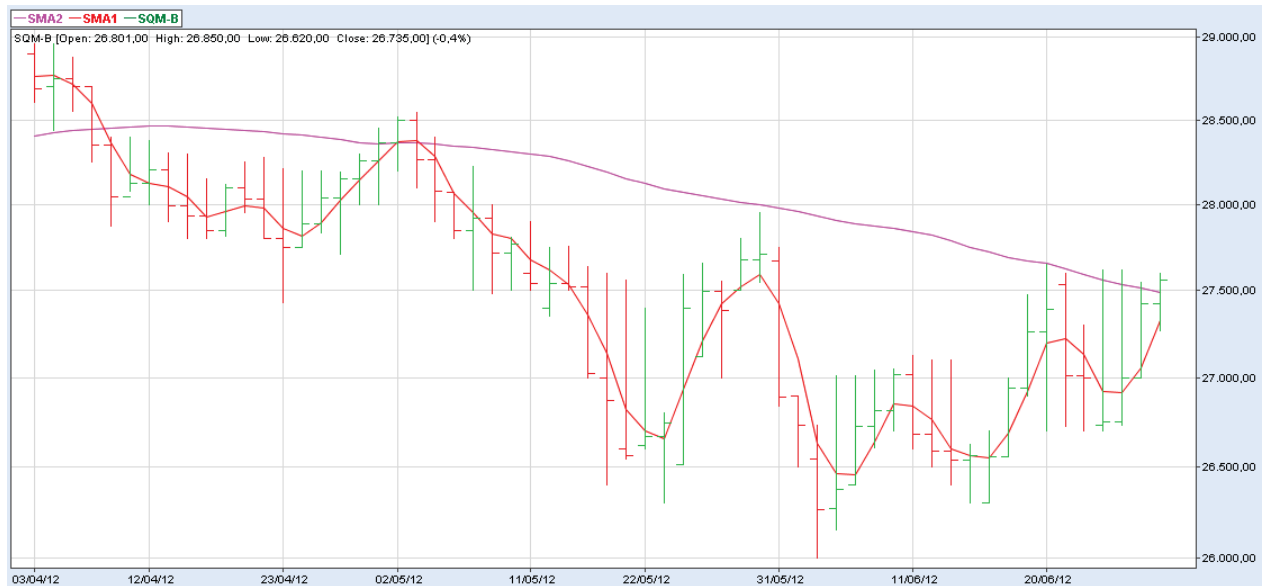
**Figura 4.16** Variación de los parámetros y rentabilidades asociadas a la mejor partícula de la mejor ejecución del experimento 6.

Al revisar la figura 4.16, se puede apreciar el comportamiento explorativo de la implementación de PSO utilizada en el sistema.

Finalmente, se prueba uno de los conjuntos de parámetros con datos de control para revisar la validez del modelo generado por el algoritmo PSO. En este caso se aplica el segundo mejor resultado del experimento 6 al mes siguiente de datos.

De este modo, el experimento 7 ya no consiste en obtener el modelo, sino en su aplicación práctica. Los datos de entrada son:

- Short MA: 3
- Long MA: 60
- StopLoss: 0,399024420367136
- StopWin: 1,53451568723673
- Instrumento: SQM-B
- Período: Mayo 2012
- Tick: 1 minuto



**Figura 4.17** Aplicación de modelo AT a datos de control del experimento 7. Se aplica el segundo mejor resultado al mes siguiente de datos disponibles, para revisar el comportamiento de los parámetros generados. En este caso el modelo logra evitar una tendencia decreciente de largo plazo.

Al aplicar el modelo a todo el mes de Mayo 2012, el sistema no realiza operación alguna, ya que determina que el precio de SQM-B pasa por un período constante de decaimiento. Este experimento es una simplificación del mecanismo de funcionamiento del sistema AT en tiempo real (como se plantea en la sección 4.4), ya que aplica el mismo conjunto de parámetros para cada *tick* de ejecución. En la práctica debiera regenerarse a intervalos regulares los parámetros de funcionamiento del modelo AT, para adaptarlo a las condiciones cambiantes del mercado.

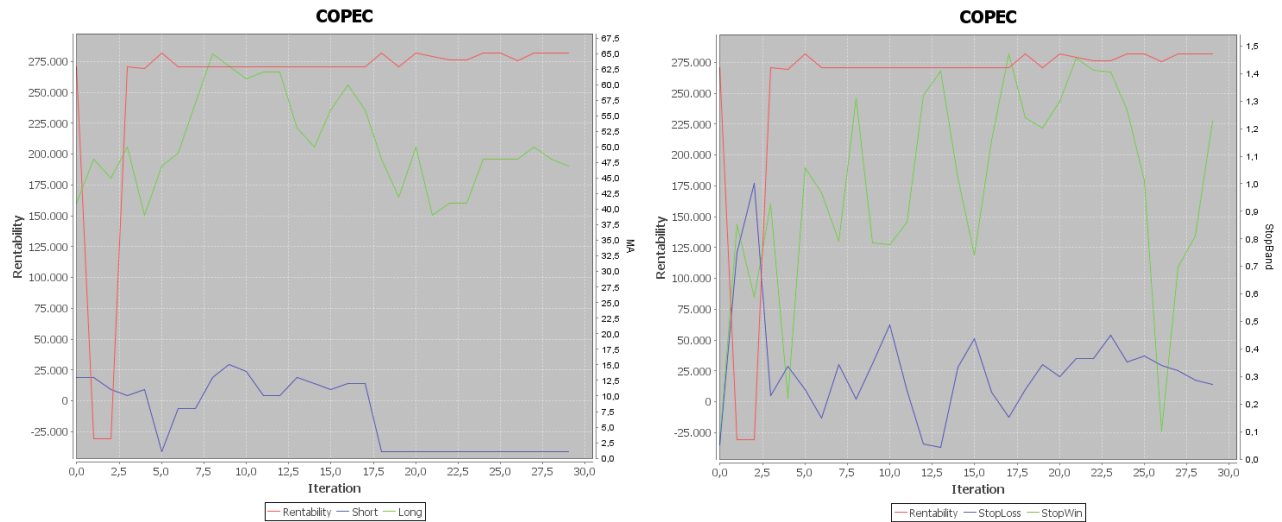
Finalmente, se realiza un experimento adicional, parecido al anterior, donde se emplea COPEC, uno de los instrumentos más transado en acciones de Renta Variable. Para tal finalidad se genera un modelo en base a los datos de un mes de operaciones con altas fluctuaciones y se aplica a un período posterior que también presenta de altas fluctuaciones de precios.

En este caso, el modelo AT del experimento 8 es generado con estos datos específicos:

- Período: 1 mes (Noviembre 2011)
- Instrumento: COPEC
- Tick: 1 minuto

El modelo resultante es:

- Short MA: 1
- Long MA: 47
- StopLoss: 0,2524628624062787
- StopWin: 1,0583116085897755



**Figura 4.18** Variación de los parámetros y rentabilidades asociadas a la mejor partícula del experimento 8.

Los parámetros generados aleatoriamente en la población inicial presentan una buena rentabilidad en la mejor partícula, la cual durante toda la ejecución del algoritmo aplica cambios menores a su configuración para obtener una combinación óptima. Esto puede indicar que el enjambre está atrapado en un óptimo local desde el principio, o que el margen de mejora es limitado.



**Figura 4.19** Aplicación de modelo AT a datos de control del experimento 8.

Al revisar la aplicación del modelo generado sobre los datos de control (figura 4.19), se puede observar que el resultado es un MA corto que sigue de cerca el precio de mercado, para aprovechar cuando el precio se dispara por sobre la tendencia a mediano plazo. Esto resulta en una rentabilidad teórica neta baja, pero positiva. El total de ganancia para el período de control es de \$4.951 CLP, aproximadamente 1% de rentabilidad. Tal como se mencionó antes, esta es una simplificación del sistema AT propuesto, dado que no se reevalúa el modelo AT a intervalos regulares para el período de control.

## 5. CONCLUSIONES

Se finaliza el Proyecto de Tesis habiendo cumplido los objetivos principales al tener las bases de los procesos de *High Frequency Trading* y *Algorithmic Trading* completamente definidas y acotadas, junto con las bases matemáticas y estadísticas del modelo de negociación que se emplea en el sistema inicial generado. También se implementó de forma iterativa un sistema de AT que pudo ser sometido a diversas pruebas. Esto permitió hacer varios ajustes a la versión inicial, logrando una mejora de rendimiento de varios órdenes de magnitud.

Sobre los objetivos cumplidos en la etapa de Seminario de Tesis, se tiene lo siguiente:

- Se estudiaron técnicas de trading que ofrecen posibilidades de funcionar bajo modalidad HFT y/o AT, escogiéndose la técnica estadística de *Moving Averages* por su simplicidad, por permitir predecir la tendencia de los precios en base al historial de un instrumento y por su aplicabilidad en técnicas de optimización.
- Se revisaron técnicas informáticas que se pueden aplicar en conjunto con las técnicas de trading, escogiéndose la aplicación de optimización de parámetros mediante metaheurísticas. Esto debido al hecho de que un problema de optimización de la rentabilidad en un mercado accionario es un problema de clase NP, para el cual la aplicación de métodos de búsqueda basados en metaheurísticas presenta muchas ventajas.

En cuanto a los objetivos planteados para el Proyecto de Tesis, se puede mencionar que:

- Se presentó un diseño tanto conceptual como en detalle sobre cómo construir un sistema AT en base a la combinación de técnicas de trading e informáticas escogidas. El diseño es lo suficientemente flexible para permitir seguir extendiendo el sistema a otras técnicas de trading y/ de búsqueda de soluciones.
- Se implementó un mecanismo de preparación y carga de los datos de mercado al sistema mediante un sistema ETL. Esto presenta la ventaja de separar el proceso de carga con respecto a la operatoria del sistema AT. De este modo el sistema sigue presentando una estructura modular, fácilmente extendible y modificable según las necesidades del mercado objetivo.
- Se construyó una versión inicial del sistema AT bajo el diseño propuesto. Esto permitió realizar las primeras pruebas de laboratorio, las cuales permitieron encontrar problemas tanto en la implementación del sistema AT como condiciones especiales para las cuales el algoritmo PSO no estaba preparado para soportar. En

particular, esta versión sirvió para determinar que en dominios continuos pero restringidos, el cálculo de velocidades de PSO debe ser acotado o modificado de algún modo.

- Finalmente, se construyó un segundo sistema AT, basado en la versión inicial, pero corrigiendo los errores detectados en la implementación del modelo AT y aplicando las limitaciones necesarias al algoritmo PSO. Gracias a las pruebas realizadas, se puede concluir que el sistema AT definido es capaz de cumplir la premisa original de la Tesis, que corresponde a la especificación de un sistema AT capaz de generar rentabilidades positivas.

De este modo, el sistema escogido corresponde a la versión mejorada. Esta versión está lejos de ser óptima, pero permite tener una base tanto teórica como práctica para investigaciones futuras, en un campo en el cual la mayor cantidad de investigación proviene desde el sector privado y no del sector académico.

En cuanto a la iteración de la implementación del sistema, el hecho de que el diseño del mismo sea completamente modular facilitó la optimización de los tiempos de ejecución de las pruebas. En su primera versión, el sistema AT generado no podría ser aplicado a la negociación en tiempo real, pero al reducir los tiempos a rangos de minutos o segundos (dependiendo de la cantidad de datos considerados), es posible aplicarlo a escenarios de tiempo real o con granularidad muy fina. Gracias a esto, se puede concluir que:

- Para que un sistema AT sea aplicable en tiempo real, se deben realizar los cálculos sobre datos temporales de forma progresiva.
- Se deben precalcular todos los valores que sea posible antes de aplicar el modelo de optimización para el instante que se quiera operar.
- El uso de un mecanismo de caché de datos en memoria o el uso de algún almacén de datos optimizados para consultas rápidas agiliza notablemente el proceso de optimización y por ende el proceso de toma de decisiones del sistema AT.

Sobre la aplicación de PSO como algoritmo de optimización, este se mostró como una solución efectiva al tipo de problema, ya que es capaz de optimizar un conjunto de variables dispares pero acotadas a un dominio específico, logrando mejorar sustancialmente la solución final. Sobre la aplicación de PSO para optimizar la rentabilidad de un sistema AT se puede concluir que:

- Se debe alterar o restringir la función de velocidad dependiendo del modelo de *trading* que se utilice. La fórmula de velocidad no funciona correctamente en dominios continuos pero muy acotados (p.e.: valores entre cero y uno).
- En conjunto la fórmula de velocidad tampoco funciona correctamente en dominios discretos positivos. En este caso, se debe aplicar algún tipo de limitación en la posición resultante para que los valores no bajen del

mínimo establecido. Para la discretización, basta con aplicar un redondeo al valor más cercano. Este tipo de redondeo fue el que se aplicó en la implementación del sistema.

Sobre posibles trabajos futuros, queda pendiente estudiar la forma de obtener el plazo óptimo de información previa a un instante dado para considerar como útil. Una de las posibles mejoras sería ver cómo afecta cambiar el *Moving Average* desde uno simple a uno exponencial. Esto permitiría favorecer la tendencia reciente de un instrumento, obviando fluctuaciones demasiado lejanas para ser de mayor importancia en el modelo.

Otra arista de investigación sería la aplicación de modelos de AT más complejos al sistema AT autoajutable, para que incluyan mecanismos de decisión con mejor manejo de riesgo o que operen sobre márgenes de ganancia más reducidos.



## 6. **BIBLIOGRAFÍA**

[Aldridge2010] I. Aldridge: “High-Frequency Trading, A Practical Guide to Algorithmic Strategies and Trading Systems”

John Wiley & Sons, Inc, Wiley trading series, 2010

[Pole2007] A. Pole: “Statistical Arbitrage, Algorithmic Trading Insights and Techniques”

John Wiley & Sons, Inc, Wiley trading series, 2007

[BCS2010] Bolsa de Comercio de Santiago: “Síntesis Mensual Mayo 2010”

<http://www.bolsadesantiago.com/Sintesis%20y%20Estadisticas/S%C3%ADntesis%20Mensual%20Mayo%202010.pdf>

[Massey2005] C. Massey, G. Wu: “Detecting regime shifts: the causes of under- and overestimation”

Management Science 51 (6), pp. 932–947, 2005

[Kendall2007] K. Kendall: “Electronic and Algorithmic Trading Technology”

Elsevier, Academic Press, 2007

[Ketter2009] W. Ketter, J. Collins, M. Gini, A. Gupta, P. Schrater: “Detecting and forecasting economic regimes in multi-agent automated exchanges”

Elsevier, Decision Support Systems 47, 2009

[Wang2009] F. Wang, K. Dong, X. Deng: “Algorithmic trading systems: design and applications”

Front. Comput. Sci. China 3, pp. 235-246, 2009

[Hernandez2003] R. Hernández, C. Fernández, P. Baptista: “Metodología de la Investigación”  
McGraw-Hill, 2003

[Hernandez2007] E. Hernández: “Cómo escribir una Tesis”  
Escuela Nacional de Salud Pública Cuba, 2007

[Bigus1996] J. Bigus: “Data Minig with Neural Networks”  
McGraw-Hill, 1996

[Pardo2008] R. Pardo: “The Evaluation and Optimization of Trading Strategies”  
John Wiley & Sons, Inc, Wiley trading series, 2008

[AT2011] Advanced Trading, I. Schmerken: “Deutsche Bank Aims New Stealth Algo At Buy-Side”  
<http://www.advancedtrading.com/algorithms/229401154>

[AMD2007] AMD WhitePaper: “Market Risk and Algorithmic Trading”  
AMD & Toomre Capital Markets LLC, 2007

[Humphery2011] M. Humphery-Jenner: “Optimal VWAP trading under noisy conditions”  
Elsevier, Journal of Banking & Finance 35, pp. 2319–2329, 2011

[Cook1971] S. Cook: “The Complexity of Theorem Proving Procedures”

Proceedings of the third annual ACM symposium on Theory of computing, pp. 151–158, 1971

[Karp1972] R. Karp: “Reducibility Among Combinatorial Problems”

In R. E. Miller and J. W. Thatcher (editors). Complexity of Computer Computations. New York: Plenum. pp. 85–103, 1972

[Fortnow2004] L. Fortnow, J. Kilian, D. Pennock, M. Wellman: “Betting Boolean-style: a framework for trading in securities based on logical formulas”

Elsevier, Decision Support Systems 39, pp. 87–104, 2005

[Chang2008] K. Chang, A. Johnson: “Online and Offline Selling in Limit Order Markets”

In Proceedings of the 4th International Workshop on Internet and Network Economics, 2008

[Blum2003] C. Blum, A. Roli: “Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison”

ACM Computing Surveys, Vol. 35, No. 3, pp. 268–308, 2003

[Kennedy1995] J. Kennedy, R. Eberhart: “Particle swarm optimization”

Proceedings of IEEE Int. Conf. on Neural Network, pp. 1942–1948, 1995

[Jabri2011] K. Jabri, D. Dumura, E. Godoya, A. Mouchetteb, B. Bèleb: “Particle swarm optimization based tuning of a modified smith predictor for mould level control in continuous casting”

Elsevier, Journal of Process Control 21, pp. 263–270, 2011

[Fikret2009] M. Fikret: “Particle Swarm Optimization and Other Metaheuristic Methods in Hybrid Flow Shop Scheduling Problem”

Singapore Polytechnic School of Electrical and Electronic Engineering, 2009

[WST2007] WallStreet & Technology, M. Rodier: “Complex Event Processing Moves Beyond Algorithmic Trading”

<http://www.wallstreetandtech.com/technology-risk-management/204203344>

[Martín03] B. Martín, A. Sanz: “Redes Neuronales y Sistemas Difusos”

Ra-Ma AlphaOmega, Segunda Edición, Mexico, 2003

[Cheng2001] J. Cheng, R. Greiner, J. Kelly, D. Bell, W. Liu: “Learning Bayesian Networks from Data: An Information-Theory Based Approach”

Department of Computing Science, University of Alberta, 2001

[Shenoy2000] C. Shenoy, P. Shenoy: “Bayesian Network Models of Portfolio Risk and Return”

The MIT Press, 2000

[Moore2001] A. Moore: “Bayes Nets for representing and reasoning about uncertainty”

School of Computer Science, Carnegie Mellon University, 2001

[E&N2010] Economía & Negocios: “Bolsa responde a las críticas por fallas de su nuevo sistema de transacciones”

<http://www.economiaynegocios.cl/noticias/noticias.asp?id=73694>