Aplicação de análise de dados de bolsa

2011 / 2012

20093093 Paulo Fernandes

Licenciatura em Informática de Gestão

Aplicação de análise de dados de bolsa

2011 / 2012

20093093 Paulo Fernandes

Licenciatura em Informática de Gestão

Outubro de 2012

Orientador ULHT: Prof. **Nuno Garcia**

Agradecimentos

Primeiro gostaria de dar uma palavra especial de agradecimento ao meu orientador,

Professor Nuno Garcia, pelo apoio, disponibilidade e compreensão demonstrados ao longo do período de execução deste trabalho.

Gostaria também de deixar o meu apreço a todos os professores do curso de Licenciatura em Informática de Gestão 2009-2012 com os quais cresci em termos de conhecimento e que, sem dúvida, deixaram a sua marca tanto na minha forma de encarar o mundo da Informação bem como no presente trabalho.

A minha gratidão também para com os meus colegas de trabalho, com os quais troquei ideias, ajustei decisões, que de alguma forma desviei tempo e atenção e que me ajudaram a superar os desafios que me foram sendo apresentados.

Finalmente, dizer obrigado à minha família que mesmo com o tempo de partilha familiar reduzido a nada, sempre me apoiou, deu a maior das compreensões, e arranjou forma de superar todos os obstáculos que foram aparecendo.

Resumo

Este trabalho teve como objectivo criar uma aplicação de análise de dados de bolsa em tempo real. Adicionalmente também eram objectivos criar sinais sobre os dados, nomeadamente de verificação da qualidade dos dados.

Foi realizado um estudo sobre as aplicações existentes com características semelhantes o que tornou claro que a maior dificuldade seria a originalidade do trabalho realizado.

A aplicação desenvolvida permite aplicar métricas pré-existentes e criar novas métricas sobre os dados obtidos, realizar testes e análises dos dados de uma forma ilimitada.

Abstract

This work objective was to create a stock market analysis over online real data. Additional objectives were to create signal algorithms over these data to generate buy/sell triggers and to analyze data quality.

Existing software applications with similar functionalities were analyzed and it was clear, though this analysis, that the biggest difficulty would be originality.

The implemented software application allows to apply pre-existent signals and create news ones and them on the acquired online data. It allows extending the application with new signals which allows testing and analyzing data without limits.

1. **Palavras Chave (Tema):** Bolsa, Séries Temporais, Sinais, Indicadores
2. **Palavras Chave (Tecnologias):** C#, Windows, dotNet,

Índice

Índice de Figuras

Índice de Tabelas

Notação e Glossário

Biblioteca	Módulo autónomo que contém conjunto de classes e funcionalidades
Broker	Agente que executa as transacções entre duas entidades
Componente	Módulo autónomo que contém conjunto de classes e funcionalidades
CSV	Conjunto de dados separados por vírgula
Forex	Mercado descentralizado de troca de moeda
Produtos Derivados	São produtos financeiros complexos utilizados pelas instituições financeiras e não financeiras para reduzir os riscos associados à sua actividade
Série Temporal	Colecção de observações feitas sequencialmente ao longo do tempo
Sinal	Sequência de dados que compõe uma mensagem
Vela	Período de avaliação de um sinal de preços com indicação da abertura, máximo, mínimo e fecho

1 Introdução

1.1 Enquadramento

O estudo, análise, teste e exploração de sinais e séries temporais tem sido uma área bastante explorada pelo mundo financeiro. A grande variedade de fontes de sinais e também a multitude de aplicações e indicadores existentes são uma prova do esforço dedicado a esta vertente. As tecnologias de informação facilitam o tratamento das várias fontes de dados e existem actualmente várias aplicações que permitem capturar, testar e validar sinais.

Este trabalho de final de curso, enquadra-se no âmbito deste tipo de produtos, uma vez que tem como objectivo a obtenção *online* de séries temporais e o seu tratamento através de um conjunto de métricas. A existência de uma oferta alargada de produtos desta área, inclusive mesmo com a disponibilização das séries temporais, obrigou a uma originalidade na produção da aplicação resultante deste trabalho.

1.2 Apresentação do projecto

Este projecto consiste no desenvolvimento de uma aplicação de análise de dados de bolsa que incorpora algumas das melhores funcionalidades encontradas no conjunto das aplicações analisadas no estudo da arte e inclui alguns indicadores que não foram encontrados nas outras soluções.

Em concreto foram desenvolvidas funcionalidades de obtenção de dados *online*, um conjunto de indicadores únicos tais como a análise de Entropia de Shannon, um algoritmo de verificação de auto semelhança e alguns indicadores comuns de verificação de tendências e osciladores tais como médias móveis e o indicador RSI (*relative strengh indicator)*.

1.2.1 Planeamento de projecto

O projecto teve as seguintes fases:

1) Análise e desenho
2) Implementação
3) Produção de manual de utilizador
4) Produção de relatório final

1.2.2 Tecnologias utilizadas

Este projecto foi desenvolvido com base no dotNet Framework da Microsoft utilizando a linguagem C#, a extensão linq e a biblioteca de gráficos Microsoft Chart Controls. Foi utilizado também o componente db4o da Versant, um componente *open source* de persistência transparente para objectos doNet e Java.

1.3 Organização do relatório

Este relatório está dividido em quatro pontos, pretendendo o actual ponto introduzir as principais questões envolventes à realização do projecto.

No segundo ponto é pretendido inserir o projecto realizado no contexto mais global e descreve todos os aspectos relacionados com o trabalho, nomeadamente a área de negócio, bem como as diversas tecnologias utilizadas no desenvolvimento do projecto.

No terceiro ponto foi realizada uma descrição técnica da implementação do projecto.

Por fim são apresentadas as conclusões retiradas da realização do projecto.

Devido à sua extensão, o código fonte da aplicação criada, bem como o projecto de prototipagem e os documentos gerados por esta ferramenta e o manual de utilizador são apresentados em formato digital.

2 Contexto

Sendo o sector financeiro um dos maiores utilizadores da informática é normal que existam várias aplicações de análise de dados de bolsa em tempo real. A escolha de aplicações é muito vasta e existem inúmeras aplicações que não seria possível testar pois apenas estão disponíveis após o estabelecimento de uma relação comercial.

2.1 Estado da Arte

Foi realizada uma pesquisa sobre as várias plataformas existentes pretendendo identificar aplicações que fossem utilizadas de uma forma abrangente ou *open source.* Os critérios para a selecção das aplicações foram os seguintes: obtenção de dados em tempo real ou através de *csv*, possibilidade de criar indicadores, possibilidade de criar agentes de transacções e possibilidade de testar os agentes criados.

As aplicações analisadas foram:

1) Metatrader
2) OpenForex
3) Forex Strategy Builder
4) Jforex da Dukascopy

2.1.1. Metatrader

Esta é uma aplicação desenvolvida por uma empresa russa, Metaquotes, e que é largamente utilizada para a transacção de moeda e também de produtos derivados da bolsa. A empresa apenas desenvolve a aplicação que depois é distribuída por *brokers* que cobram comissões pelas transacções executadas.

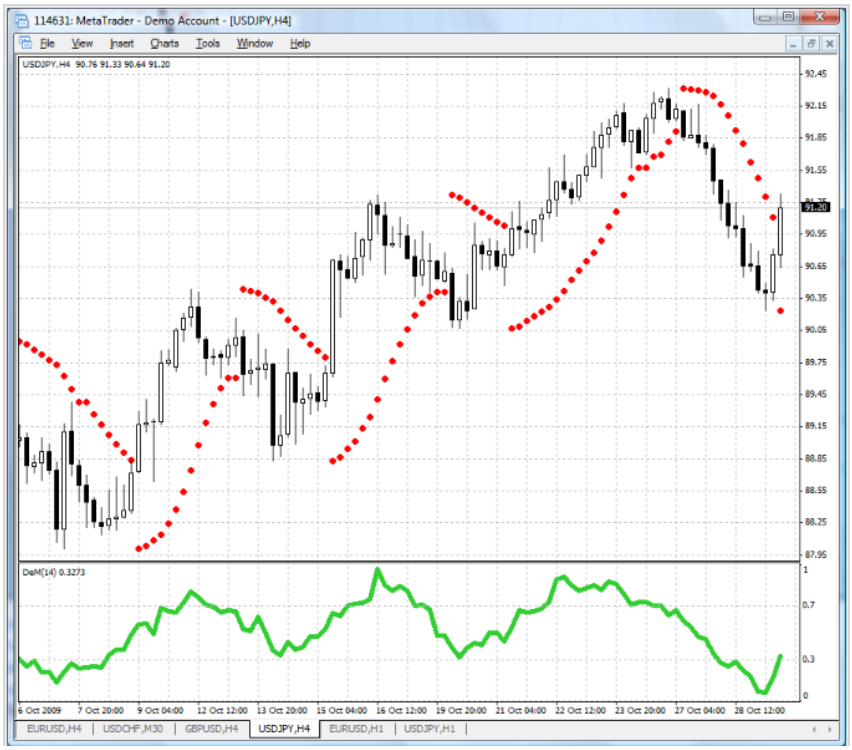

Figura 1 – Metratrader 4

Esta aplicação tem, *out-of-the-box*, um conjunto de indicadores de tendência e osciladores e permite que o utilizador desenvolva os seus próprios indicadores numa linguagem própria, mql. Esta linguagem é muito semelhante à linguagem C pelo que a criação de novos indicadores é muito simples para quem domine essa linguagem.

Para além das funcionalidades já descritas esta aplicação permite também criar agentes de estratégias automáticas de transacções, i.e. *expert advisors*. Estes agentes podem executar as transacções em tempo real sem intervenção do utilizador desde que programados para tal.

Finalmente, esta aplicação permite testar os agentes criados sobre um conjunto de instrumentos financeiros. Os testes têm a possibilidade de realizar optimizações utilizando algoritmos genéticos para optimizar os vários parâmetros dos agentes e a possibilidade de serem visuais para permitir ver a evolução das operações.

Um dos grandes problemas verificados com esta plataforma é a fiabilidade dos dados testados muitas vezes incoerentes e com falhas temporais o que conduz a testes inválidos.

2.1.2. Jforex

A aplicação Jforex é uma aplicação desenvolvida em tecnologia Java pela empresa Dukascopy que é também um *broker* para transacções de moeda.

Figura 2 - JForex

Esta é muito semelhante em funcionalidade ao Metatrader tendo algumas diferenças substanciais: os dados históricos têm uma grande fiabilidade e a linguagem de desenvolvimento de indicadores e estratégias é Java.

À parte das duas grandes diferenças citadas no parágrafo anterior, estas duas aplicações são extremamente semelhantes.

2.1.3. Openforex

Esta é uma aplicação *open source* e que funciona de uma forma modular permitindo a quem quiser desenvolver os seus módulos para acrescentar funcionalidades poder fazê-lo facilmente.

Figura 3 – Aplicação OpenForex

Esta plataforma foi desenvolvida recorrendo a quatro camadas de módulos: classes comuns, plataforma base, componentes de sistema e componentes customizados.

Embora a organização dos módulos pareça ser muito rígido em relação à distribuição das classes é ao mesmo tempo muito flexível pois permite interpretações diferenciadas tanto a nível lógico como a nível de interface de utilizador.

Como já foi verificado, esta aplicação permite desenvolver módulos customizados para qualquer das funcionalidades que apresenta e mesmo desenvolver funcionalidades novas.

Uma das funcionalidades que esta aplicação tem é poder obter dados reais *online* de *feeds* de dados como o Yahoo Finance ou da aplicação Metatrader, entre outras.

Contudo, o facto de a interface de utilizador ser tão modular torna-a um tanto complexa e que obriga a uma curva de aprendizagem acentuada.

2.1.4. Forex Strategy Builder

A aplicação Forex Strategy Builder é diferente de outras aplicações analisadas pois não tem um *feed* de dados *online*, apenas utiliza *csv* e permite a importação de dados tanto do Jforex como do Metatrader. Como vantagem tem um sistema muito fiável de testes de estratégias.

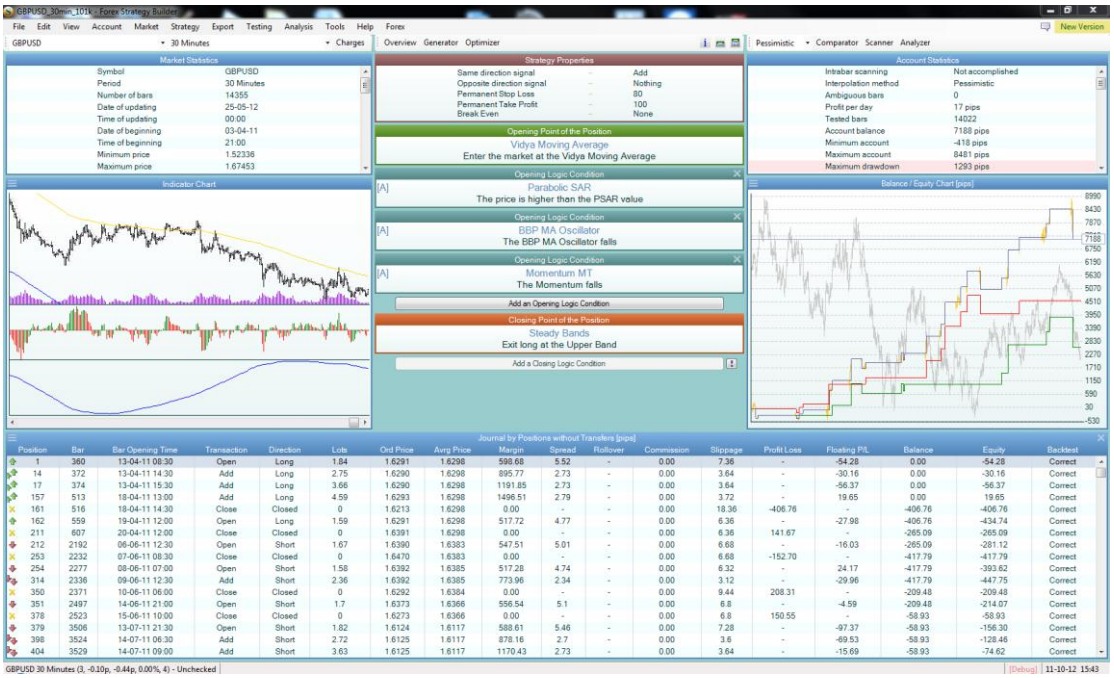

Figura 4 – Forex Strategy Builder

É uma aplicação *open source*, desenvolvida em C# sobre a plataforma dotNet. O facto de não estar construída com camadas separadas entre lógica e apresentação dificulta a criação e integração de outros módulos, tal como por exemplo, um módulo de obtenção de dados *online* em tempo real.

Mas esta aplicação tem de facto algumas funcionalidades que a distinguem das demais:

- Os resultados das estratégias são apresentados quase de imediato, em pouco mais de um segundo;
- É possível definir a estratégia de entrada e saída nos preços num determinado período, i.e., pode ser decidida uma aproximação pessimista que selecciona o preço pior, optimista que selecciona o preço mais vantajoso e uma aproximação aleatória;

Embora não exista um módulo de execução de estratégias incorporado nesta aplicação o mesmo criador desenvolveu uma segunda aplicação que importa as estratégias criadas nesta aplicação e através de um módulo de integração com a aplicação metatrader pode executar as estratégias em tempo real.

2.1.5. Resumo

As características ou funcionalidades que foram consideradas mais relevantes.

- *Open source* – embora não seja conclusivo que o software *open source* tenha mais qualidade este potencia o desenvolvimento de outras ferramentas ou mesmo a melhoria das existentes sendo assim uma característica relevante na análise de um software;
- Dados em tempo real – a obtenção de dados em tempo real é uma característica muito importante neste tipo de aplicações pois permite reagir de imediato aos sinais e permite também criar estratégias de execução em tempo real;
- Modular – a separação em módulos das várias funcionalidades permite uma evolução e reaproveitamento de bibliotecas de software para outras aplicações bem como a construção de módulos alternativos o que melhora a extensibilidade do software;

- Estratégias automáticas em tempo real – a funcionalidade de execução de estratégias em tempo real permite desenvolver módulos que reagem aos sinais aplicados aos dados em tempo real possibilitando a negociação em bolsa ou forex sem intervenção humana;
- Teste Histórico – a funcionalidade de testar dados históricos é essencial para avaliar a rentabilidade de uma estratégia no passado para poder projectar ganhos futuros;
- Criação de indicadores / sinais – a possibilidade de adicionar novos sinais a uma aplicação permite acrescentar funcionalidade à aplicação sem emitir novas versões.

Caracteristica Aplicação	Open source	Dados em tempo real	Modular	Estratégias Automáticas em tempo real	Teste Histórico	Criação de Indicadores / Sinais
Metatrader	Não	Sim	Não	Sim	Sim	Sim
JForex	Não	Sim	Não	Sim	Sim	Sim
Openforex	Sim	Sim	Sim	Sim	Sim	Sim
Forex Strategy Builder	Sim	Não	Não	Não	Sim	Sim

Tabela 1– Comparação funcionalidades e características

2.2 Análise de requisitos

2.2.1. Casos de Uso

O diagrama seguinte apresenta os principais actores e as suas interacções com a aplicação.

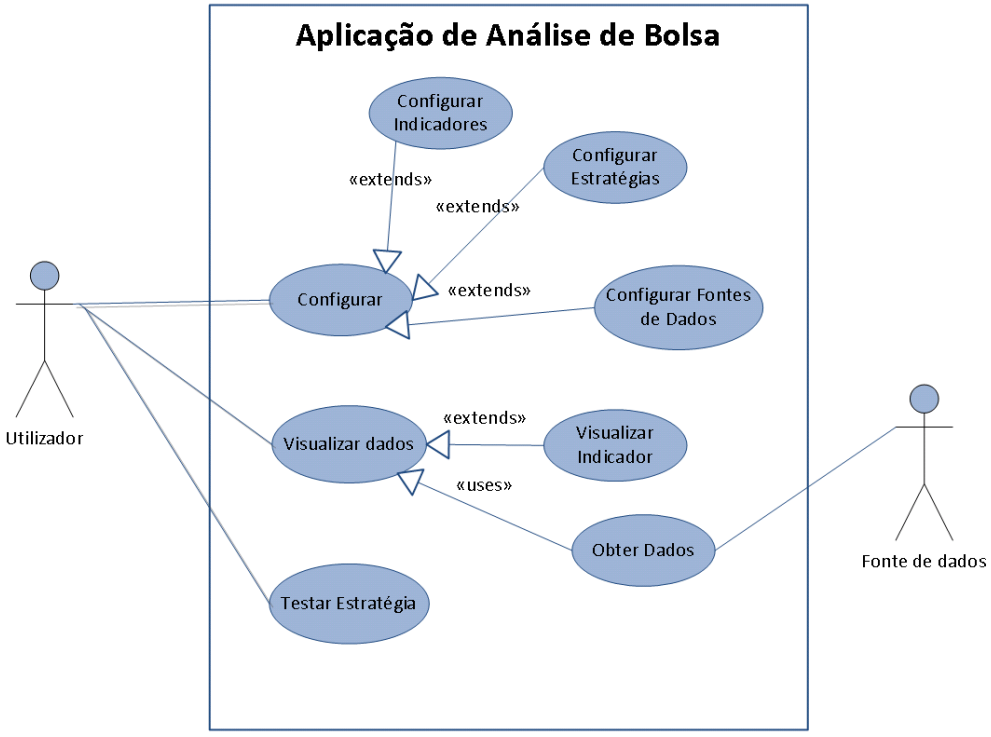

Figura 5 – Casos de uso

2.1.1.1. Actores

Actor	Descrição
Utilizador	Representa o utilizador da aplicação cujas principais interacções são a configuração da aplicação, a visualização de dados e o teste de estratégias.
Fonte de dados	Representa um sistema externo onde serão obtidos dados em tempo real sobre um dado instrumento.

Tabela 2 – Actores

2.1.1.2. Actividades

Caso de uso	Descrição
Configurar	Este caso de uso representa a configuração da aplicação por parte do utilizador e compreende três actividades de configuração: fontes de dados, indicadores e estratégias.
Configuração de Fonte de dados	Este caso de uso representa a configuração de fontes de dados que permite alterar as fontes de dados disponíveis.
Configuração de indicadores	Este caso de uso representa a configuração que permite alterar os indicadores disponíveis.
Configuração de estratégias	Este caso de uso representa a configuração que permite alterar as estratégias disponíveis.
Visualizar Dados	Este caso de uso representa a interacção principal que permite a visualização gráfica da evolução de um instrumento.
Visualizar Indicador	Este caso de uso representa a interacção que permite a visualização gráfica de indicadores aplicados aos dados de um instrumento.
Obter Dados	Este caso de uso representa a interacção com os sistemas externos fontes de dados.

Testar Estratégia	Este caso de uso representa a execução e apresentação dos resultados dessa execução de uma estratégia sobre os dados de um instrumento.

Tabela 3 – Actividades

2.1.2. Matriz de Requisitos

ID	Nome	Tipo	Prioridade	Descrição
FR1	Obtenção de dados em tempo real	Requisito Funcional	Alta	**Obtenção de dados online em tempo real. Os dados obtidos não deverão ser persistidos ou seja, deverão ser sempre obtidos a partir da fonte de dados.**
FR1.1	Dados obtidos de várias fontes	Requisito Funcional	Alta	**Os dados devem poder ser obtidos de várias fontes. Deve ser possível seleccionar a fonte de dados pretendida.**
FR1.2	Deve ser possível acrescentar fontes de dados.	Requisito Funcional	Alta	**As fontes de dados devem poder ser adicionadas por configuração. A aplicação não deve limitar as fontes de dados às distribuídas pela aplicação.**
FR1.3	Os dados devem ser obtidos em intervalos de tempo	Requisito Funcional	Alta	**Os dados devem ser obtidos em forma de séries temporais com periodicidade fixa ou variável e com indicação de máximo, mínimo, valor inicial, valor final e, se existente, quantidade. Estes valores mapeiam para os utlizados em bolsa: abertura, máximo, mínimo, fecho e quantidade.**
FR2	Criação de indicadores customizados	Requisito Funcional	Média-Alta	**A criação de indicadores customizados permite ao utilizador desenvolver os seus próprios indicadores para gerar sinais.**
FR2.1	Deve ser possível acrescentar novos indicadores	Requisito Funcional	Média-Alta	**Os indicadores podem ser adicionados por configuração. A aplicação não deve limitar os dados aos obtidos**
FR3	Criação de estratégias automatizadas	Requisito Funcional	Média-Alta	**A criação de estratégias automatizadas permite testar de uma forma automática os indicadores criados pelo utilizador para verificar a sua aplicabilidade**
FR3.1	Criação de Estratégias	Requisito Funcional	Média-Alta	**As estratégias devem ser adicionadas por configuração.**
FR4	Teste de estratégias	Requisito Funcional	Média-Alta	**O teste de estratégias permite aplicar dados históricos a uma estratégia automatizada para verificar a sua rentabilidade a longo prazo**
FR4.1	O resultado de um teste deve ser realizado em pontos	Requisito Funcional	Média	**Os testes devem apresentar os resultados utilizando pontos e não a moeda subjacente ao instrumento visto que a moeda poder não existir, como por exemplo no caso dos índices.**
FR4.2	O resultado deve apresentar estatísticas sobre o teste	Requisito Funcional	Média	**Devem ser apresentados dados sobre a quantidade de operações com sucesso, insucesso, o ponto mais negativo, o ponto mais positivo, a melhor série e a pior série.**
SR1	O sistema deve ser funcionar em sistemas Windows	Requisito de Sistema	Alta	**A aplicação deve suportar os sistemas operativos Windows XP e superiores.**
SR2	O sistema deve persistir as alterações realizadas pelo utilizador	Requisito de Sistema	Média-Alta	**As alterações de configuração bem como os últimos valores utilizados nas diversas interacções com a aplicação devem ser persistidas sem intervenção do utilizador.**
SR3	A aplicação deve ser implementada numa	Requisito de Sistema	Média	

	linguagem de alto nível			
SR4	A aplicação deve ter um manual de utilizador	Requisito de Sistema	Alta	
UR1	Visualização gráfica de dados	Requisito de Utilizador	Alta	**Permitir a visualização dos dados de forma gráfica**
UR1.1	Maximizar e minimizar	Requisito de Utilizador	Média	**Deve ser possível maximizar e minimizar o gráfico.**
UR1.2	**Alteração de cores**	**Requisito de Utilizador**	**Baixa**	**Deve ser possível definir a cor de cada indicador.**

Tabela 4 – Matriz de Requisitos

2.2 Tecnologias utilizadas

A aplicação de análise de bolsa foi desenvolvida com a ferramenta de desenvolvimento Visual Studio 2010 sobre a plataforma dotnet e na linguagem C#. Também foram utilizadas as tecnologias LINQ, uma linguagem definição de critérios de pesquisa para objectos incluída na plataforma dotnet, a plataforma Microsoft Chart Controls e também a biblioteca DB4O (*db for objects*).

2.2.1. Visual Studio 2010

O Visual Studio 2010 é uma ferramenta de desenvolvimento IDE (*Integrated Development Environment*) quer permite desenvolver em várias linguagens que executam sobre a plataforma dotNet.

2.2.2. dotNet (Microsoft .Net)

A plataforma dotNet é uma parte integrante do sistema operativo Windows que suporta a criação e execução de aplicações e que foi desenhada para cumprir os seguintes objectivos:

- Fornecer um ambiente de desenvolvimento consistente, orientado a objectos independentemente do código ser guardado e executado localmente, executado localmente e distribuído pela internet ou executado remotamente.
- Fornecer um ambiente de execução que minimize os requisitos de instalação e conflitos de versões.
- Fornecer um ambiente de execução que promova a execução código seguro, incluindo código criado por desconhecidos ou parcialmente conhecidos.
- Fornecer um ambiente de execução de código que elimine questões de performance inerentes a ambientes de *scripting* ou interpretadas.

- Tornar a experiência do programador consistente entre os vários tipos de aplicações tais como aplicações web ou aplicações Windows.
- Garantir que o código criado com o dotNet cumpre os *standards* de comunicação da indústria para permitir a integração com qualquer outro código.

O dotNet é composto por dois componentes base: o ambiente comum de execução (CLR – *common language runtime*) e a biblioteca de código do dotNet.

O ambiente comum de execução é um agente que gere código em tempo de execução, fornece serviços base tais como um gestor de memória, um gestor de *threads*, um gestor de comunicações e garante o cumprimento de regras que promovem a robustez e segurança das aplicações.

A biblioteca de código dotNet é uma colecção extensa de classes reutilizáveis e orientadas a objectos que se podem usar para criar aplicações quer estas sejam aplicações de consola até aplicações gráficas ou web.

2.2.3. LINQ

LINQ (*Language-Integrated Query*) é uma extensão do Visual Studio 2010 e da plataforma dotNet que permite reduzir a ponte entre o mundo dos objectos e o mundo dos dados.

Tradicionalmente, as pesquisas sobre os dados são expressas como simples cadeias de caracteres sem verificação de tipos.

O LINQ proporciona uma forma de construir pesquisas sobre qualquer fonte de dados, quer esta seja uma tabela de uma base de dados, um ficheiro XML ou uma colecção de objectos.

2.2.4. Microsoft Chart Controls

O Microsoft Chart Controls é uma biblioteca de componentes gráficos que permitem mostrar séries de dados de uma forma gráfica.

A utilização desta biblioteca possibilita a criação de gráficos simples, intuitivos e visualmente atraentes para análises financeiras ou estatísticas complexas.

2.2.5. db4o

db4o é uma base de dados *open source* que permite aos programadores Java e dotNet guardar e recuperar qualquer objecto aplicacional com apenas uma única linha de código, eliminando a necessidade de predefinir ou manter em separado um modelo rígido de estrutura de dados.

2.2.6. Convenção de código

Sendo a aplicação desenvolvida em C#, o código foi implementado seguindo as convenções de código para esta linguagem.

2.3 Funcionalidades

As funcionalidades pretendidas neste tipo de aplicação são:

- Obtenção de dados em tempo real

A obtenção de dados em tempos para ser analisada pelos indicadores permite desenvolver estratégias de tempo real para períodos de tempo inferiores a um dia

- Criação de indicadores customizados

A criação de indicadores customizados permite ao utilizador desenvolver os seus próprios indicadores para gerar sinais.

- Criação de estratégias automatizadas

A criação de estratégias automatizadas permite testar de uma forma automática os indicadores criados pelo utilizador para verificar a sua aplicabilidade

- Teste de estratégias

O teste de estratégias permite aplicar dados históricos a uma estratégia automatizada para verificar a sua rentabilidade a longo prazo

- Apresentação gráfica dos dados obtidos *online* e indicadores

É essencial que a aplicação apresente graficamente os dados obtidos e os valores calculados nos indicadores

2.4 Descrição técnica

2.4.1. Componentes

A aplicação desenvolvida foi dividida em três projectos (Visual Studio *projects*) separados que resultam em três componentes lógicos separados:

- PFC2012 – Componente que representa a aplicação gráfica. Este componente contém toda a lógica relacionada com a interface de utilizador.
- PFC2012Lib – Este componente contém todas as entidades lógicas da aplicação. Foram incluídos também neste componente os indicadores, fontes de dados e estratégias incluídas inicialmente na aplicação.
- PFC2012Utils – Este componente contém classes auxiliares ao funcionamento da aplicação.

2.4.2. Dependências

O diagrama seguinte apresenta as dependências ao nível das bibliotecas bem como ao nível das suas classes.

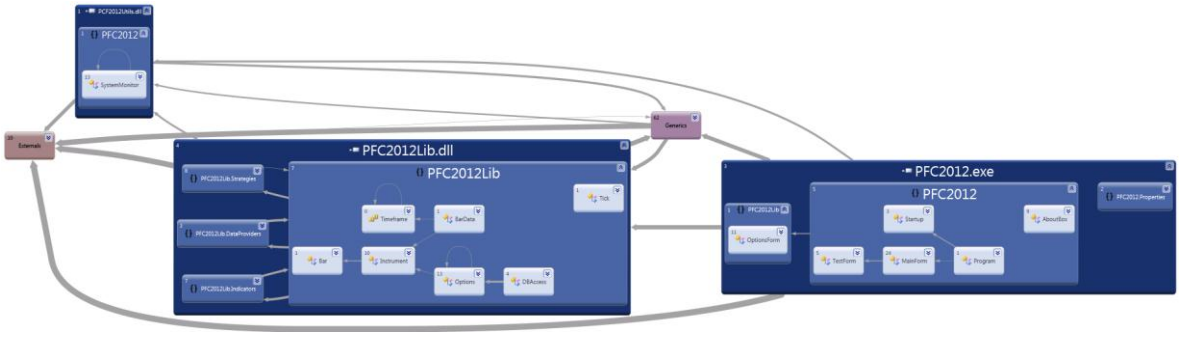

Figura 5 – Dependências

2.1.1. Interfaces

Na linguagem C# (e em linguagens baseadas no plataforma dotNet) descrevem um grupo relacionado de funcionalidades que podem pertencer a qualquer classe ou estrutura. As interfaces consistem em propriedades, métodos, eventos, índice e definem de uma forma abstracta uma funcionalidade. Quando uma classe implementa uma *interface* define o comportamento de todos os membros definidos.

A criação de indicadores e estratégias adicionais é conseguida pela implementação de *interfaces* específicos para o efeito.

Assim é possível construir sem restrições de regras um número ilimitado de indicadores e estratégias.

A implementação de diversas fontes de dados também foi realizada pela disponibilização de um *interface* que permite o desenvolvimento de fontes de dados adicionais para posterior utilização na aplicação.

Os indicadores, estratégias e fontes de dados disponibilizados pela aplicação utilizam as mesmas *interfaces* disponibilizadas para a implementação de indicadores, estratégias e fontes de dados adicionais.

2.2 Ecrãs

2.2.1. Ecrã principal

O ecrã inicial apresenta um menu que permite realizar as operações de gerir as bibliotecas de extensão (de fontes de dados, indicadores e estratégias), seleccionar a fonte de dados activa e executar uma estratégia de teste.

O elemento principal é o gráfico que apresenta os dados obtidos e o indicador seleccionado. Para tal é necessário primeiro introduzir o símbolo do instrumento desejado.

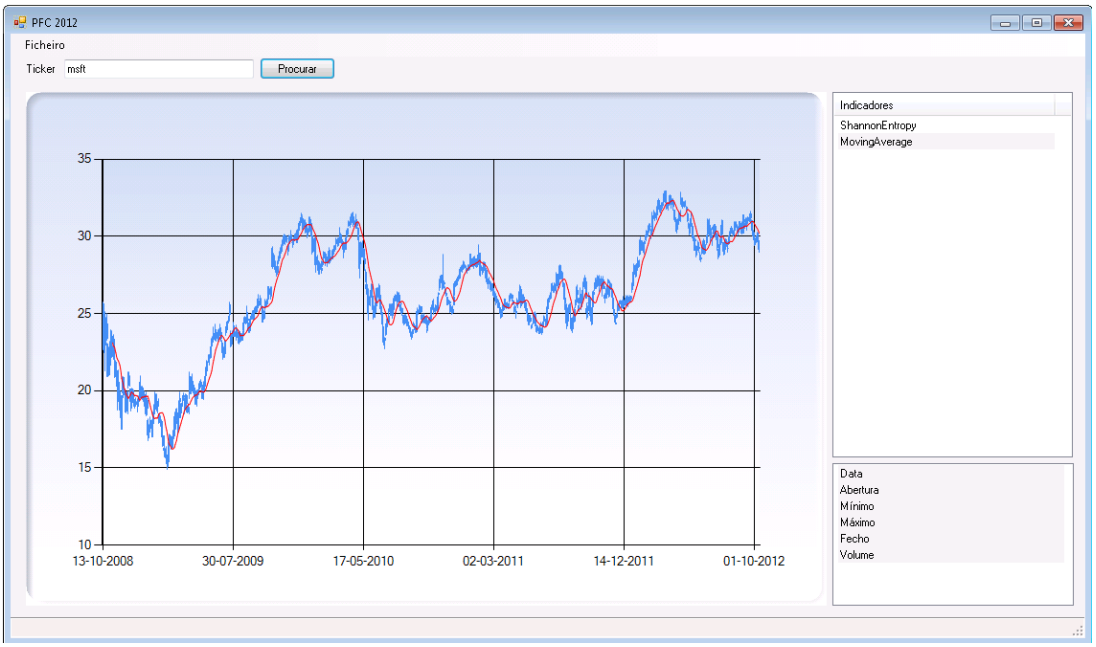

Figura 6 – Ecrã principal

2.2.2. Gestão de Bibliotecas

No ecrã de selecção de estratégias podemos adicionar e remover bibliotecas de código que contém as fontes de dados, indicadores e estratégias adicionais da aplicação. Estas bibliotecas podem ser desenvolvidas como descrito no ponto 2.2.4 – Extensibilidade.

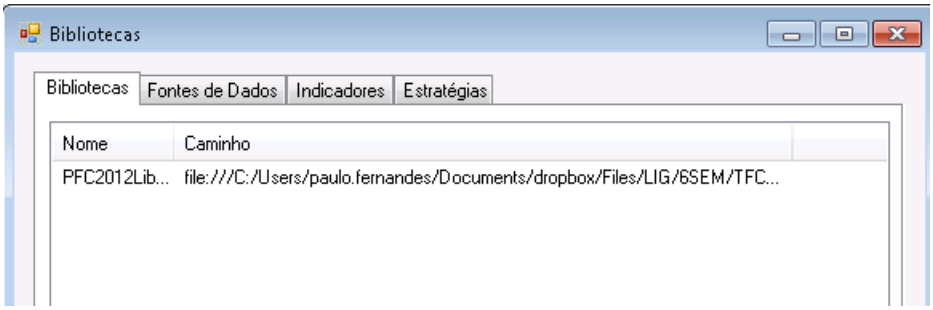

Figura 7 – Ecrã de selecção de bibliotecas

2.2.3. Teste de Estratégias

O teste de estratégias permite executar uma estratégia sobre os dados obtidos *online*. A implementação de testes realizada calcula pontos entre as várias posições assumidas ao longo da estratégia. É permitido executar compras e vendas independentemente da existência de posições assumidas previamente. Ou seja, é possível tomar posições curtas ou longas, o preço da posição vai sendo ajustado – é calculada a média – à medida que vão sendo adicionadas novas posições.

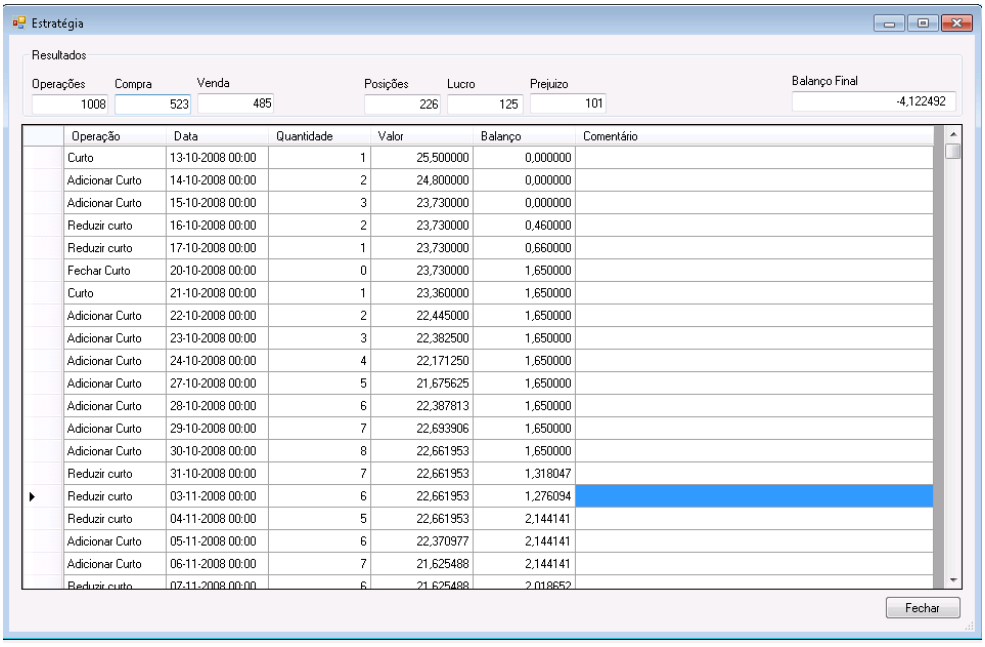

Figura 8 – Ecrã de Teste de Estratégia

2.2.4. Extensibilidade

A extensibilidade permite adicionar novas fontes de dados, indicadores e estratégias sem necessidade de uma nova versão da aplicação. Permite também aos utilizadores construir os seus próprios indicadores, as estratégias e fontes de dados.

Esta extensibilidade é realizada pela implementação de uma interface que serve de contrato numa arquitectura cliente-servidor em que o serviço implementa essa interface que o cliente utiliza quando necessitar.

Uma única biblioteca pode conter várias fontes de dados, indicadores e estratégias. No manual de utilizador são fornecidos exemplos e mais detalhes sobre a forma de implementação destas extensões.

2.2.4.1. Fontes de dados

Para implementar uma fonte de dados, é necessário implementar o interface ***IDataProvider***:

```
public interface IDataProvider
{
    string Name { get; }
    BarData GetData(Instrument instrument);
}
```

A implementação deve, na função ***GetData***, devolver os dados relativos ao instrumento passado como argumento. Os dados serão apresentados pela ordem que estiverem na colecção ***BarData***.

2.2.4.2. Indicadores

Para implementar um indicador é necessário implementar o interface IIndicator:

```
public interface IIndicator
{
    string Name { get; }
    IDictionary<string, double> ParameterValues { get; }
    void Calculate(List<Bar> bars);
    double?[] Values { get; set; }
    ScaleTypes ScaleType { get;}
}
```

A função ***Calculate*** deve ser executada para obter os resultados para a lista de velas passada por argumento. Este método calcula o resultado de cada vela e coloca os resultados na propriedade ***Values***.

2.2.4.3. Estratégias

Para implementar uma estratégia é necessário implementar o interface IStrategy:

```
public interface IStrategy
{
    string Name { get; }
    List<Indicators.IIndicator> Indicators { get; }
    List<StrategyOperation> Calculate(List<Bar> bars);
}
```

A função ***Calculate*** deve gerar um conjunto de operações de compra ou venda sobre os dados das velas fornecidas por argumento. Este conjunto de operações será posteriormente analisado pela aplicação e transformado num conjunto de posições.

Existem duas regras que devem ser observadas:

- Não é possível ter posições curtas e longas simultaneamente;
- Todas as operações são executadas com a quantidade de uma unidade.

3 Conclusões

3.1 Objectivos realizados

Este trabalho tinha como objectivo realizar uma aplicação que obtivesse *online* séries temporais e as tratasse através de um conjunto de métricas. Reconhecendo a variedade e qualidade de oferta deste tipo de aplicações, algumas disponibilizadas ou patrocinadas por instituições financeiras foi também fixado como objectivo a originalidade.

O primeiro objectivo foi realizado na plenitude e até ultrapassado, pois a aplicação desenvolvida permite não só o acesso a dados *online* e a aplicação de métricas sobre os mesmos bem como adicionar novas fontes de dados, indicadores e estratégias de teste.

O mesmo não se pode afirmar em relação ao segundo objectivo. Embora as características de permitir criar fontes de dados, indicadores e estratégias sejam inovadores existem outras aplicações que também o permitem e que têm um conjunto de funcionalidades diferenciadores.

3.2 Outros trabalhos realizados

No âmbito da realização da análise do estado da arte deste trabalho foram realizadas as seguintes tarefas:

- Desenvolvimento de indicador ShannonEntropy para a aplicação metatrader
- Desenvolvimento de várias estratégias de negociação automática para a aplicação metatrader baseadas no indicador ShannonEntropy
- Desenvolvimento e aplicação de estratégias em "*paper trading*" desenvolvidas com a aplicação Strategy Builder.
- Implementação de módulo de optimização com algoritmo genético na funcionalidade de criação automática de estratégias da aplicação Strategy Builder.
- Implementação de módulo de criação automático de estratégias contínuo para todos os instrumentos configurados da aplicação Strategy Builder.

3.3 Limitações e trabalho futuro

A aplicação desenvolvida tem algumas limitações e possibilidades de melhoria que se enumeram de seguida:

- A criação de indicadores, estratégias e fontes de dados têm que ser editados e compilados fora da aplicação. A aplicação poderia disponibilizar esta funcionalidade.
- A aplicação poderia ser disponibilizada *online* num site, o que em termos comerciais seria um factor de diferenciação.

3.4 Apreciação final

O trabalho desenvolvido permite realizar análises sobre instrumentos financeiros de uma forma simples.

A interface de utilizador simplista e focada na funcionalidade permite atingir os objectivos do trabalho proposto mas é pouco atractiva o que em termos comerciais prejudica o seu possível sucesso.

Uma futura evolução no sentido de tornar a aplicação disponível *online* pode permitir a comercialização da mesma sendo necessário garantir a qualidade das fontes de dados e os seus direitos.

A avaliação do estado da arte e o conjunto dos trabalhos adicionais realizados traduziram-se no crescimento significativo sobre as plataformas existentes.

Foi gratificante realizar melhorias em projectos já existentes, desenvolver automatismos de transacção automática para testes, aplicar esses automatismos em tempo real e analisar os maus resultados e a sua diferença em relação à fase de testes.

4 Bibliografia

Katiz, J. O., & McCormick, D. (s.d.). *The encyclopedia of trading strategies.* New York: McGraw-Hill.

Liberty, j., & Xie, D. (2007). *Programming C# 3.0, 5th Edition.* Sebastopol: O'Reilley.

Paulos, J. A. (2003). *A mathematician plays the stock market.* New York: Basic Book.

Pressman, R. S. (2002). *Engenharia de Software.* Rio de Janeiro: McGraw Hill.

Stevens, L. (2002). *Essential technical analysis.* New York: Honh Wiley & Sons.

Vliet, B. V., & Hendry, R. (2004). *Modeling Financial Markets.* New York: McGraw-Hill.

Williams, B. (s.d.). *Trading chaos.* Wiley.

Anexo 1 – Diagrama de Classes