

---

# Análise e Implementação das Técnicas de Neely para Identificação das Ondas de Elliott

***Bruno Feitosa***

---

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: \_\_\_\_\_

**Bruno Feitosa**

## **Análise e Implementação das Técnicas de Neely para Identificação das Ondas de Elliott**

Monografia apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP e ao Departamento de Estatística – DEs-UFSCar, para o Exame de Qualificação, como parte dos requisitos para obtenção do título de Mestre em Ciências de Computação e Matemática Computacional.

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientador: Prof. Dr. Vanderlei Bonato

**USP – São Carlos  
Agosto de 2019**

**Bruno Feitosa**

## **Analysis and Implementation of Neely Techniques for Elliott Wave Identification**

Monograph submitted to the Instituto de Ciências Matemáticas e de Computação - ICMC-USP and to the Departamento de Estatística – DEs-UFSCar, as part of the qualifying exam requisites of the Master Joint Graduate Program in Computer Science and Computational Mathematics.

Concentration Area: Computer Science and Computational Mathematics

Advisor: Prof. Dr. Vanderlei Bonato

**USP – São Carlos  
August 2019**

## **Resumo**

Observação de preços em mercados sempre foi do interesse daqueles que negociavam produtos, pois a capacidade de interpretar o mercado e decidir a melhor hora de comprar ou vender poderia ditar a diferença entre ganhar ou perder dinheiro. Durante a Era Meiji, no Século 19, um mercador de arroz chamado Muneshima Homma desenvolveu uma técnica de visualização e interpretação do preço do arroz, que flutuava constantemente durante um dia de negociações. Ele se tornou capaz de reconhecer os momentos em que o preço estaria mudando entre baixas e altas, reconhecendo o melhor momento para negociar o produto.

O ambiente tecnológico atual, onde a computação e a Internet facilitaram a velocidade de transmissão e acesso a informação contribuiu para fomentar o crescimento do mercado de ações. Porém, com as rápidas flutuações dos preços de ações causadas pelos inúmeros investidores negociando os ativos, o mercado de ações não é um ambiente de fácil entrada para investidores. Diversos economistas e analistas se especializaram em observar, entender e prever os movimentos dos preços no mercado de ações através da análise de gráficos e outros indicadores.

Entre estes especialistas, em 1938 se destacou Ralph Nelson Elliott e seu trabalho: O Princípio das Ondas. Sua teoria era capaz de interpretar o estado do mercado e prever movimentos futuros com base nos padrões das Ondas observadas. Porém, sua teoria foi considerada subjetiva por alguns estudiosos, o que pode ter evitado a disseminação de sua teoria. Mais tarde, Glen Neely desenvolveu um método científico capaz de implementar a teoria de Elliott, através de um algoritmo que gera resultados objetivos e repetíveis.

O objetivo deste trabalho é implementar o algoritmo de Neely para análise do preço de ações, e analisar como diferentes configurações do método alteram a análise resultante.

## Lista de Figuras

1	Gráfico do Preço do Dólar (2018-02-05) em <i>candlesticks</i> . . . . .	2
2	Preço do Dólar (2018-02-05) as 12:00 . . . . .	2
3	Preço do Dólar (2018-02-05) as 13:00 . . . . .	3
4	Gráfico do Preço do Dólar (2018-02-05) em OHLC . . . . .	4
5	Gráfico do Preço do Dólar (2018-02-05) . . . . .	4
6	Gráfico do Preço do Dólar (2018-02-05) com maior Resolução . . . . .	5
7	Gráfico do Preço do Dólar (2018-02-05) comparando resoluções . . . . .	6
8	Histórico do Índice Dow Jones 1932-1938 . . . . .	7
9	Onda Padrão 5-3 (adaptado de [1]) . . . . .	8
10	Onda Padrão 5-3 estendida (adaptado de [1]) . . . . .	9
11	Fluxograma do Método Elliott-Neely . . . . .	12
12	Diagrama de Casos de Uso . . . . .	16
13	Visão Lógica . . . . .	23
14	Visão de Processo para UC-1.1: Importar CSV . . . . .	24
15	Visão de Processo para UC-2.1: Analisar Ação . . . . .	25
16	Visão de Processo para UC-3.1: Exibir Resultado . . . . .	25
17	Visão de Processo para UC-4.1: Mudar Configuração . . . . .	26
18	Visão de Implementação . . . . .	27
19	Visão de Implantação . . . . .	29
20	Padrão de Onda Motivadora: Impulso . . . . .	36
21	Padrão de Onda Motivadora: Extensão . . . . .	36
22	Padrão de Onda Motivadora: Truncagem . . . . .	37
23	Padrão de Onda Motivadora: Diagonal Final/Inicial . . . . .	37
24	Padrão de Onda Corretiva: Zigzag . . . . .	37
25	Padrão de Onda Corretiva: Zigzag Duplo . . . . .	38
26	Padrão de Onda Corretiva: Plano . . . . .	38
27	Padrão de Onda Corretiva: Plano Estendido . . . . .	38
28	Padrão de Onda Corretiva: Plano Corrido . . . . .	39
29	Padrão de Onda Corretiva: Triângulo . . . . .	39
30	Padrão de Onda Corretiva: Combinação (exemplo 1) . . . . .	40
31	Padrão de Onda Corretiva: Combinação (exemplo 2) . . . . .	40

## Lista de Tabelas

1	Preços do Dólar (2018-02-05) em Tabela . . . . .	6
2	Notações das Ondas de Elliott de acordo com o Grau da Onda . . . . .	9
3	Padrões das Ondas de Elliott . . . . .	10
4	Representação Quantitativa dos Interesses dos <i>Stakeholders</i> . . . . .	15
5	UC-1.1 - Importar .CSV . . . . .	17
6	UC-1.2 - Importar de Terceiros . . . . .	18
7	UC-2.1 - Analisar Ação . . . . .	19
8	UC-3.1 - Exibir Resultado . . . . .	20
9	UC-4.1 - Mudar Configuração . . . . .	21
10	Requisitos Funcionais . . . . .	22
11	Decisão - Linguagem de Implementação . . . . .	30
12	Decisão - Análise de Diferentes Configurações . . . . .	31
13	Cronograma de atividades . . . . .	32
14	Progresso do algoritmo . . . . .	33

## Sumário

<b>Resumo</b>	<b>i</b>
<b>Lista de Figuras</b>	<b>ii</b>
<b>Lista de Tabelas</b>	<b>iii</b>
<b>Sumário</b>	<b>iv</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Contextualização: Mercado Financeiro e Negociação de Ações . . . . .	1
1.1.1 Representações Gráficas: OHLC ( <i>Open-High-Low-Close</i> ) . . . . .	1
1.2 Motivação: Ondas de Elliott . . . . .	7
<b>2 Extensões de Neely</b>	<b>10</b>
2.1 Limitação de Janela de Análise . . . . .	13
<b>3 Fatores de Estudo</b>	<b>13</b>
<b>4 Arquitetura</b>	<b>14</b>
4.1 Contexto do Sistema . . . . .	14
4.2 Requisitos . . . . .	14
4.2.1 Visão arquitetural . . . . .	14
4.2.2 <i>Stakeholders</i> . . . . .	14
4.2.3 Atributos de Qualidade . . . . .	15
4.2.4 Requisitos de Alto-Nível . . . . .	15
4.2.5 Casos de Uso . . . . .	16
4.2.6 Requisitos Funcionais . . . . .	21
4.3 Análise . . . . .	22
4.3.1 Suposições . . . . .	22
4.3.2 Roteiro tecnológico . . . . .	22
4.4 Arquitetura do Software . . . . .	23
4.4.1 Visão Lógica . . . . .	23
4.4.2 Visão de Processo . . . . .	24
4.4.3 Visão de Implementação . . . . .	26

4.4.4	Visão de Implantação . . . . .	29
4.4.5	Decisões de Desenvolvimento de Software . . . . .	29
<b>5</b>	<b>Cronograma e Progressão</b>	<b>32</b>
	<b>Referências</b>	<b>34</b>
	<b>Apêndices</b>	<b>36</b>
A	Padrões das Ondas de Elliott . . . . .	36



## 1. Introdução

### 1.1. Contextualização: Mercado Financeiro e Negociação de Ações

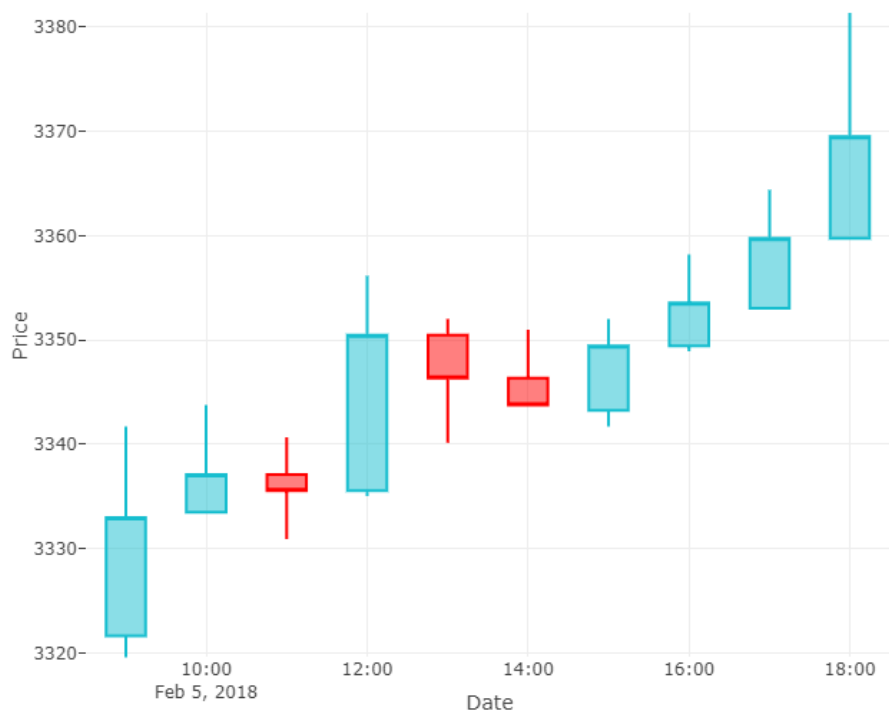
Stringham e Curott em [2] pesquisaram e descobriram que negócios com múltiplos acionistas se tornaram populares com contratos *commenda* na Itália Medieval [3], e Malmendier [4] proveu evidências que empresas acionistas remontam à Roma antiga. No entanto, Stringham e Curott identificaram que o primeiro mercado de ações do mundo nasceu em Amsterdã do século 17, onde um mercado secundário ativo de ações emergiu. As duas maiores empresas eram a Dutch East India Company e a Dutch West India Company, fundadas em 1602 e 1621.

Com a fundação do primeiro mercado de ações eletrônico, a NASDAQ, em 1971, e a popularização da computação e Internet durante as últimas décadas, investir no mercado financeiro se torna cada vez mais fácil. Porém, preços de ações no mercado financeiro flutuam de acordo com as negociações feitas no próprio mercado. A volatilidade deste tipo de negócio implica que, para se obter sucesso neste tipo de empreendimento, é necessário saber observar e entender as tendências dos preços e como elas refletem o otimismo e pessimismo dos negociadores.

#### 1.1.1. Representações Gráficas: OHLC (*Open-High-Low-Close*)

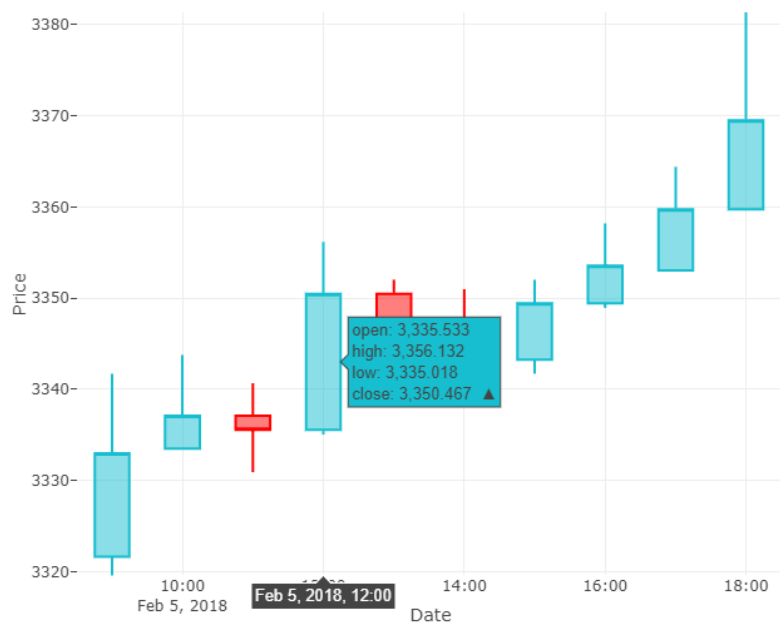
De acordo com a pesquisa de Lo e Hasanhodzic [5], o primeiro registro de preços com o objetivo de predizê-los ocorreu na Babilônia, durante o primeiro milênio a.C.. Segundo Shimizu [6] “a abertura de trocas de *commodities* na Europa Ocidental (1561) e Japão (1654) proveram o ambiente necessário para o desenvolvimento do gráfico”. Conforme Kirkpatrick e Dahlquist [7], no momento em que estas negociações aconteciam, os mercados haviam se tornado sofisticados o bastante para produzir múltiplos preços durante o decorrer do dia, e criando portanto, a necessidade de se registrar os preços de abertura, máximo, mínimo e de fechamento dos *commodities* sendo negociados.

Atualmente, a forma mais comum de representar o histórico do preço de uma ação é observando a dinâmica dos preços negociados através dos valores de abertura, máximo, mínimo, e fechamento. Os quatro valores podem ser condensados em um único marcador, sendo os dois mais comuns as “velas japonesas” (*candelstick*) e o OHLC (*Open-High-Low-Close*, literalmente Abertura-Máximo-Mínimo-Fechamento). Os valores podem ser observados durante diferentes períodos, como um dia, ou um minuto. Na Figura 1 podem ser vistos os preços da cotação do Dólar (US\$) em Reais (R\$) durante o dia 5 de Fevereiro de 2018, utilizando as marcações de *candlesticks*.



**Figura 1. Gráfico do Preço do Dólar (2018-02-05) em *candlesticks***

Cada “vela” representa um conjunto de quatro valores: abertura (*open*), máximo (*high*), mínimo (*low*) e fechamento (*close*). A extensão do “pavio” da vela indica os valores de máximo (extensão superior) e mínimo (extensão inferior), e os valores de abertura e fechamento são indicados pelo “corpo” da vela. Os preços observados as 12:00 e 13:00 podem ser vistos nas Figuras 2 e 3 respectivamente.



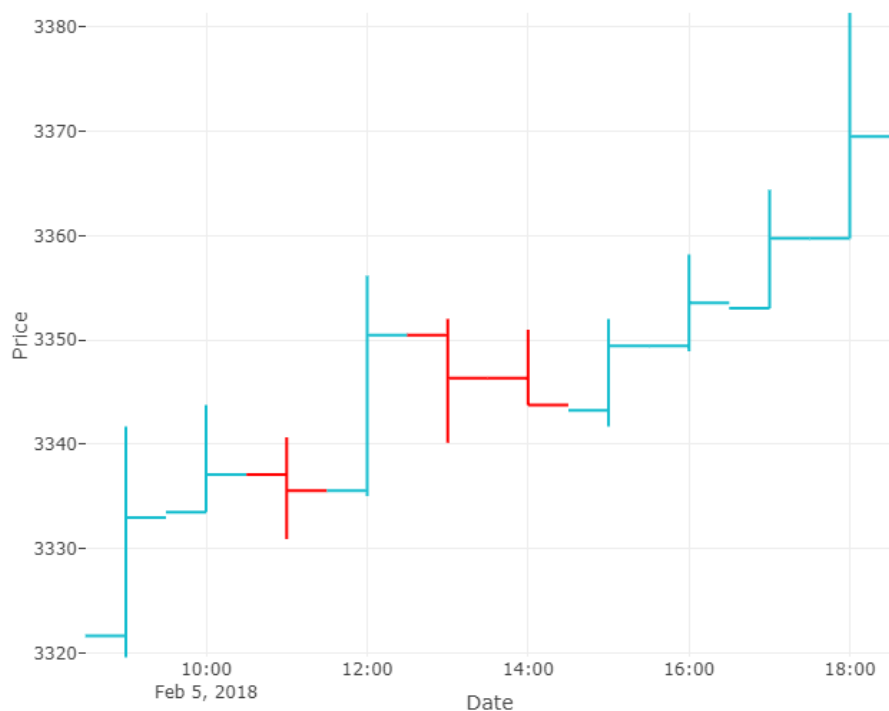
**Figura 2. Preço do Dólar (2018-02-05) as 12:00**



**Figura 3. Preço do Dólar (2018-02-05) as 13:00**

É notável observar que na Figura 2 o preço aumenta durante o período (preço de fechamento é maior que o preço de abertura), enquanto na Figura 3 ocorre o contrário (preço de fechamento é menor que o preço de abertura). Preços crescentes e decrescentes são distinguidos por cores (e.g. azul para crescente, vermelho para decrescente) ou por vela sem cor interna (crescente) ou com cor cheia (decrescente) em representações monocromáticas. Essa distinção é necessária, caso contrário não é possível observar no corpo da vela se o preço de abertura/fechamento ocorreu no topo ou na base do corpo da vela.

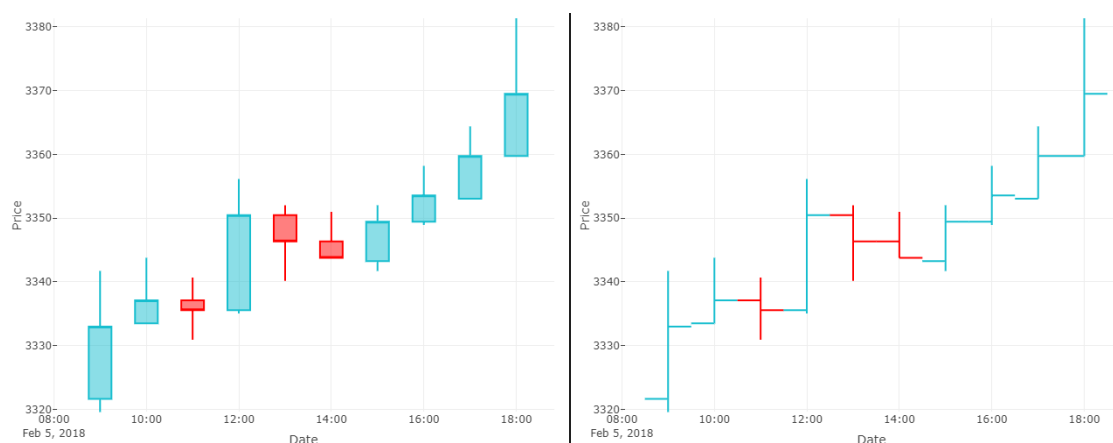
Uma outra forma de se representar os preços além do marcador de *candelsticks* são marcadores OHLC. O mesmo histórico de preço mostrado na Figura 1 pode ser visto na Figura 4 a seguir utilizando marcadores OHLC.



**Figura 4. Gráfico do Preço do Dólar (2018-02-05) em OHLC**

Na representação OHLC, o traço vertical indica os valores de máximo e mínimo, a extensão a esquerda indica o valor de abertura, e a extensão a direita indica o valor de fechamento. Em contrapartida às *candlesticks* não é necessário uma representação em cores para distinguir abertura de fechamento no indicador, porém, a visualização toma menos volume no gráfico, o que pode dificultar a leitura ao observador.

Os dois marcadores podem ser vistos lado a lado na Figura 5 para efeito de comparação.



**Figura 5. Gráfico do Preço do Dólar (2018-02-05)**

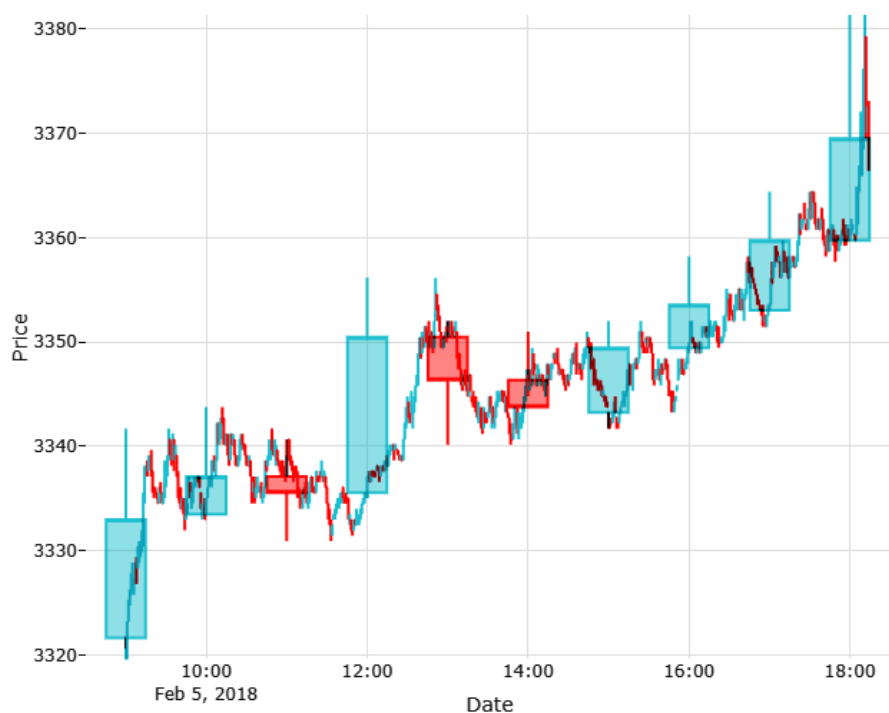
É importante ressaltar também que gráficos de preços de ações procuram representar a dinâmica do preço durante um período, já que o preço de um ativo muda constantemente. Os preços do Dólar observados na Figura 4 representam um período de 1 hora

para cada ponto no gráfico. Os valores do Dólar para o mesmo dia, porém com período de 1 minuto, podem ser observados na Figura 6 a seguir.



**Figura 6. Gráfico do Preço do Dólar (2018-02-05) com maior Resolução**

Apesar de serem observados os mesmos preços no mesmo dia, a resolução pode mudar completamente a interpretação do gráfico. Uma comparação superimposta da Figura 1 e da Figura 6 pode ser vista na Figura 7 a seguir.



**Figura 7. Gráfico do Preço do Dólar (2018-02-05) comparando resoluções**

Os preços do Dólar utilizados nos gráficos desta seção podem ser vistos na Tabela 1 a seguir. Estes preços foram exportados do programa MetaTrader 5, com uma conta associada à corretora XP Investimentos. Os gráficos desta seção foram gerados utilizando o programa em desenvolvimento.

**Tabela 1. Preços do Dólar (2018-02-05) em Tabela**

<b>Data</b>	<b>Abertura</b>	<b>Máximo</b>	<b>Mínimo</b>	<b>Fechamento</b>	<b>Volume</b>
2018/02/05 9:00	3,321628	3,341712	3,319568	3,332958	56220
2018/02/05 10:00	3,333473	3,343772	3,333473	3,337078	50250
2018/02/05 11:00	3,337078	3,340682	3,330898	3,335533	42015
2018/02/05 12:00	3,335533	3,356132	3,335018	3,350467	61545
2018/02/05 13:00	3,350467	3,352012	3,340167	3,346347	45650
2018/02/05 14:00	3,346347	3,350982	3,343772	3,343772	34520
2018/02/05 15:00	3,343257	3,352012	3,341712	3,349437	37540
2018/02/05 16:00	3,349437	3,358192	3,348922	3,353557	36450
2018/02/05 17:00	3,353042	3,364372	3,353042	3,359737	33420
2018/02/05 18:00	3,353042	3,381366	3,353042	3,369521	13830

## 1.2. Motivação: Ondas de Elliott

Como a análise dos preços no mercado de ações é uma maneira de se conhecer o estado do mercado e reconhecer momentos oportunos para investimentos, diversos especialistas financeiros se dedicaram a estudar os movimentos dos preços. Charles J. Collins, da Investment Counsel, Inc., citou no prefácio de [8] 4 especialistas nos estudos do comportamento do mercado: Arthur Cecil Pigou, Charles Henry Dow, Bernard Mannes Baruch e Ralph Nelson Elliott.

Arthur Cecil Pigou afirmou que “movimentos de preços (para cima ou para baixo) no mercado são causados por excesso de otimismo seguidos de excessos de pessimismo”. Charles Henry Dow (co-fundador da Dow Jones & Company) observou que aumentos de preços são caracterizados por três oscilações. Bernard Mannes Baruch afirmou que “eventos em si não registram flutuações no mercado de ações; o que registra as flutuações no mercado de ações são as reações humanas aos eventos”.

Ralph Nelson Elliott iniciou seu estudo no início da década de 30, observado os índices históricos da Dow Jones. Após a recessão de 29, Elliott foi capaz de prever o momento exato em que o mercado iria virar e romper o máximo de 35 e seguir um forte movimento otimista, seguido de outra forte recessão. Em 13 de Março de 1935, Elliott enviou um telegrama para Collins dizendo que todos os índices tinham chegado no mínimo final. Em 14 de Março os índices da Dow Jones chegaram no valor mínimo dos últimos 13 meses, e pelos próximos, enquanto os índices registravam somente máximo após máximo, Collins aceitou colaborar no livro de Elliott [9] sobre a Teoria das Ondas. O índice Dow Jones pode ser visto durante o período de 1932 até 1938 na Figura 8, obtido em [10].



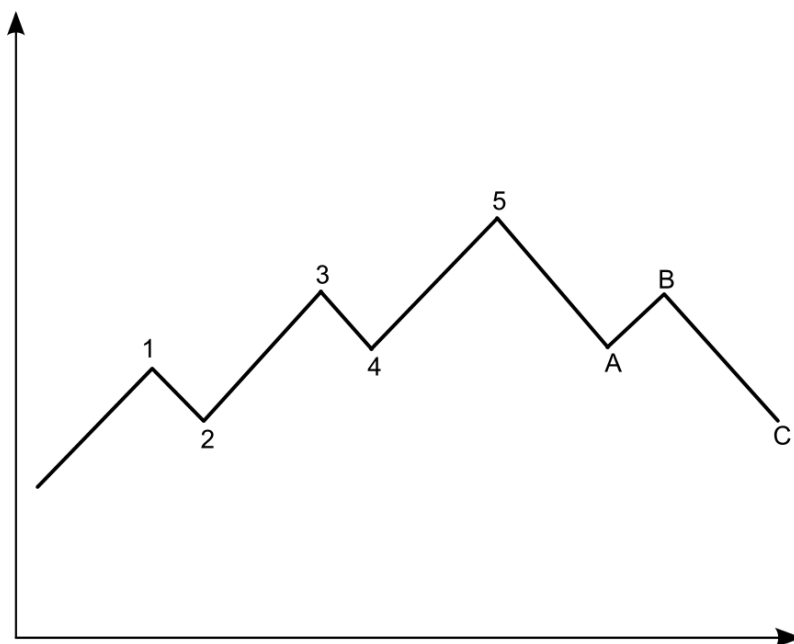
Figura 8. Histórico do Índice Dow Jones 1932-1938

A Teoria das Ondas de Elliott pode ser explicada em alguns princípios baseados

nos fenômenos que ele observou nos preços. Estes princípios são:

- Preços no Mercado de Ações seguem padrões reconhecíveis;
- Estes padrões se repetem em “forma”, mas não necessariamente em “amplitude” e “tempo”;
- Estes padrões são chamadas de **Ondas (Waves)**;
- Uma onda pode ser classificada como sendo **Motivadora** ou **Corretiva**;
  - Ondas **Motivadoras (Motive)** seguem a **mesma tendência** do mercado;
  - Ondas **Corretiva (Corrective)** seguem a **tendência contrária** a do mercado;
- As **Ondas** podem ser combinadas entre si para formarem **novas Ondas** em **resoluções de tempo** diferentes;

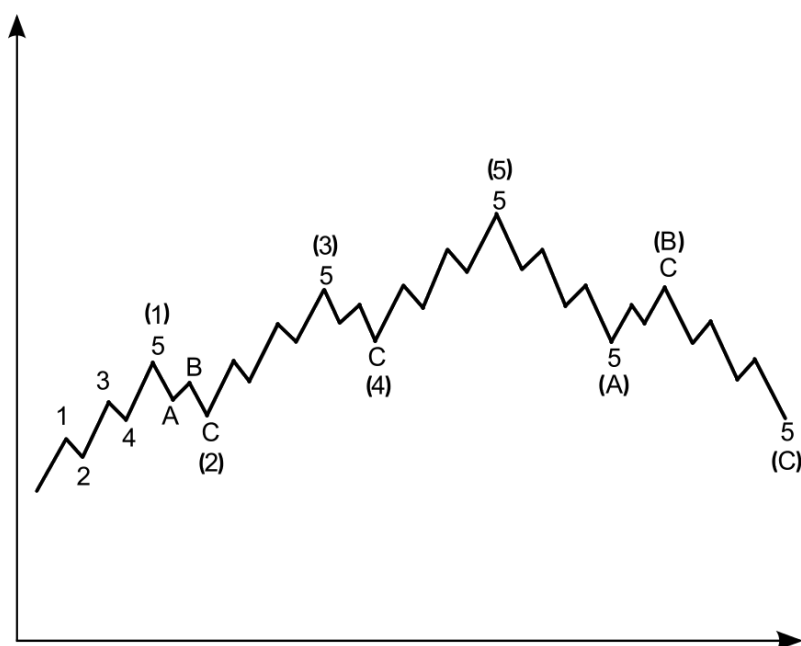
A onda idealizada observada por Elliott foi a Onda 5-3, vista na Figura 9. As Ondas de Elliott são identificadas de duas formas diferentes: motivadoras são identificadas com números, e corretivas são identificadas com letras.



**Figura 9. Onda Padrão 5-3 (adaptado de [1])**

A tendência principal do mercado é que um Impulso (Onda 1-2-3-4-5 na Figura 9) sempre é seguido de uma Retração (Onda A-B-C). Adicionalmente, como as Ondas podem ser combinadas entre si para formar novas Ondas, a Onda 5-3 também poderia ser observada como na Figura 10.





**Figura 10. Onda Padrão 5-3 estendida (adaptado de [1])**

Quando um conjunto de Ondas forma uma nova onda em uma resolução de tempo diferente, é dito que as Ondas formam uma onda de grau superior. Dessa forma, se torna necessária uma regra de enumeração para Ondas de acordo com o grau (resolução do tempo) na qual elas se encontram. As marcações utilizadas para cada grau podem ser vistas na Tabela 2 a seguir.

**Tabela 2. Notações das Ondas de Elliott de acordo com o Grau da Onda**

Grau da Onda		5's Seguindo a Tendência					3's Contra a Tendência		
1	Supermilênio	①	②	③	④	⑤	Ⓐ	Ⓑ	Ⓒ
2	Milênio	(1)	(2)	(3)	(4)	(5)	(A)	(B)	(C)
3	Submilênio	1	2	3	4	5	A	B	C
4	SuperCiclo Maior	Ⅰ	Ⅱ	Ⅲ	Ⅳ	Ⅴ	ⓐ	ⓑ	ⓒ
5	SuperCiclo	(I)	(II)	(III)	(IV)	(V)	(a)	(b)	(c)
6	Ciclo	I	II	III	IV	V	a	b	c
7	Primário	①	②	③	④	⑤	Ⓐ	Ⓑ	Ⓒ
8	Intermediário	(1)	(2)	(3)	(4)	(5)	(A)	(B)	(C)
9	Menor	1	2	3	4	5	A	B	C
10	Minuto	①	②	③	④	⑤	Ⓐ	Ⓑ	Ⓒ
11	Minuete	(i)	(ii)	(iii)	(iv)	(v)	(a)	(b)	(c)
12	SubMinuete	i	ii	iii	iv	v	a	b	c
13	Micro	①	②	③	④	⑤	Ⓐ	Ⓑ	Ⓒ
14	Submicro	(1)	(2)	(3)	(4)	(5)	(A)	(B)	(C)
15	Minúsculo	1	2	3	4	5	A	B	C

Os padrões identificados por Elliott podem ser vistos na Tabela 3, onde cada padrão pode ser visto na Figura de referência. As Figuras com cada padrão fazem parte do Apêndice A.

**Tabela 3. Padrões das Ondas de Elliott**

	<b>Padrão</b>	<b>Nome Original</b>	<b>Referência</b>
<b>Ondas Motivadoras</b>	Impulso	<i>Impulse</i>	Figura 20
	Extensão	<i>Extension</i>	Figura 21
	Truncagem	<i>Truncation</i>	Figura 22
	Diagonal Final/Inicial	<i>Ending/Leading Diagonal</i>	Figura 23
<b>Ondas Corretivas</b>	Zigzag	<i>Zigzag</i>	Figura 24
	Zigzag Duplo	<i>Double Zigzag</i>	Figura 25
	Plano	<i>Flat</i>	Figura 26
	Plano Estendido	<i>Extended Flat</i>	Figura 27
	Plano Corrido	<i>Running Flat</i>	Figura 28
	Triângulo	<i>Triangle</i>	Figura 29
	Combinação (3 Duplo e Triplo)	<i>Double/Triple 3</i>	Figuras 30 e 31

Mandelbrot [11] critica a teoria de Elliott dizendo que a predição de preços através da leitura das Ondas era um método incerto, onde o fator mais importante era julgamento subjetivo do grafista e não uma fonte objetiva e replicável como os números. Porém, Mandelbrot publicou um artigo [12] (disponível digitalmente e transcrito em [13]) sobre a característica fractal dos preços no mercado financeiro, artigo o qual foi criticado por Prechter [14] pelo fato de que Mandelbrot afirmava ter descoberto a característica fractal dos movimentos dos preços quando Ralph Nelson Elliott já havia divulgado seu trabalho *The Wave Principle* [9] em 1938 explicando os mesmos mecanismos descritos por Mandelbrot porém de forma muito mais compreensiva e fiel aos mercados reais.

Aronson [15] também critica a Teoria das Ondas por ser vaga, já que não identifica consistentemente onde uma Onda começa ou termina, e que previsões estão sujeitas a revisão subjetiva.

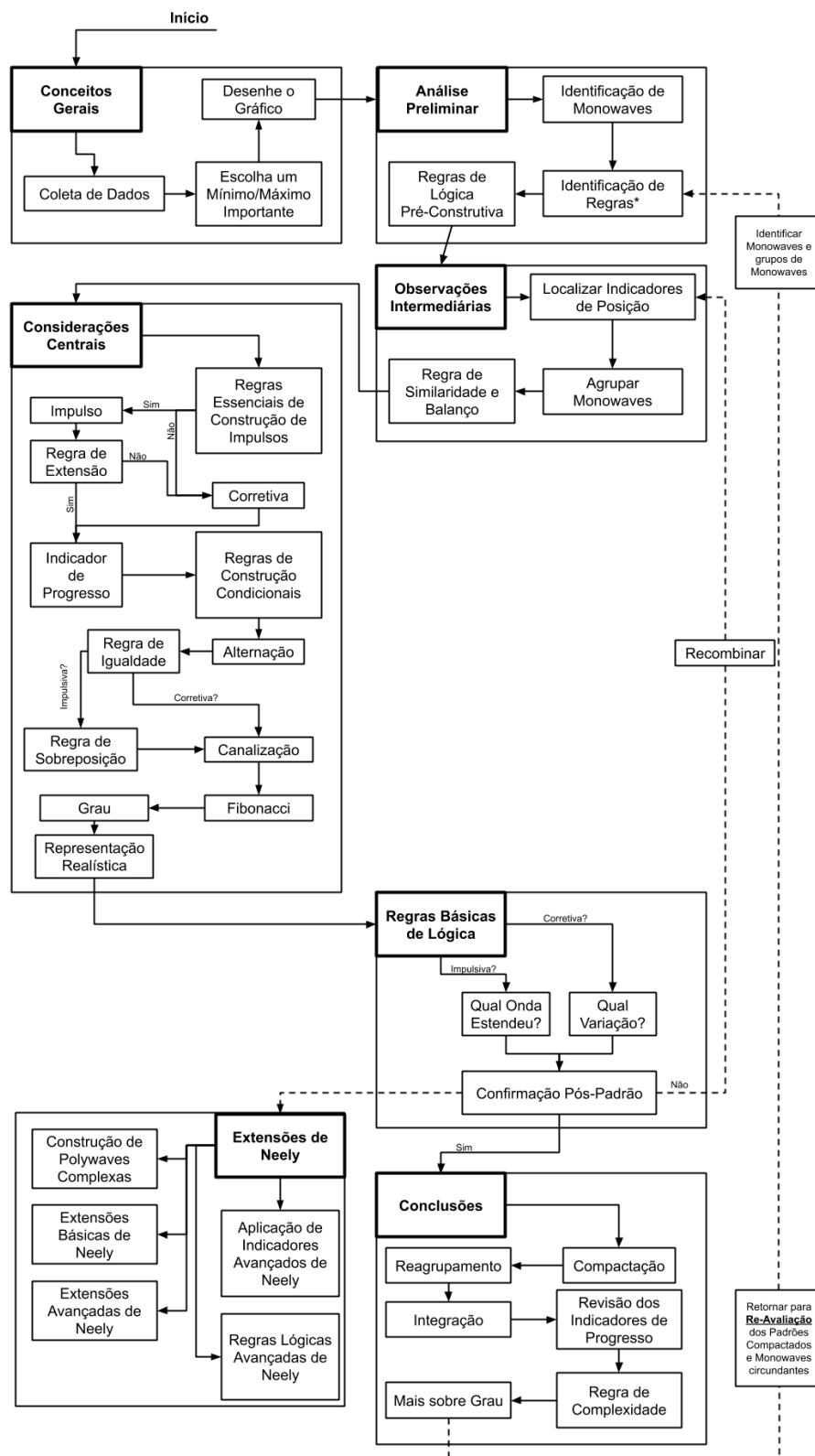
## 2. Extensões de Neely

Glenn Neely, em 1987, publicou o livro “Mastering Elliott Wave” [16], introduzindo um método científico e objetivo para análise de mercado. O **Método Neely para Análise das Ondas de Elliott**, resumido neste trabalho como **Método Elliott-Neely**, cria regras numéricas para classificação e formação das Ondas de Elliott, eliminando quaisquer subjetividade da análise das Ondas de Elliott.

Neely criou uma visualização intermediária para as Ondas chamadas **Monowaves**. As **monowaves** são a menor unidade de variação do preço de uma ação. Enquanto, por exemplo, na Teoria de Elliott uma Onda Impulso é o conjunto de movimentos numerados de 1 a 5 (Figura 20), para Neely cada movimento é uma única **monowave**.

Utilizando a visualização intermediária das **monowaves**, Neely criou um algoritmo que combina **monowaves** em conjuntos que representam as Ondas de Elliott. Além disso, o algoritmo também recombina os conjuntos identificados entre si formando Ondas de grau superior (como listados na Tabela 2), e aplica conjuntos de regras chamadas

“Extensões de Neely” que, segundo Neely, estendem as capacidades de identificação dos padrões de Elliott. Um fluxograma do algoritmo criado por Neely pode ser visto na Figura 11.



**Figura 11. Fluxograma do Método Elliott-Neely**

## 2.1. Limitação de Janela de Análise

Na etapa da Análise Preliminar do Método Elliott-Neely, é necessário aplicar a Regra de Neutralidade, que pode alterar o início e/ou fim das monowaves dependendo da relação entre os pontos do gráfico e o principal Movimento Direcional. Movimentos Direcionais são conjuntos de monowaves que movimentam o preço obrigatoriamente para cima ou para baixo. Na aplicação da Regra de Neutralidade, o primeiro Movimento Direcional dita quais monowaves devem ter seu início ou fim alterado de acordo com um conjunto de regras. Porém, Neely dita que o primeiro Movimento Direcional dita a Regra de Neutralidade para todos os pontos de análise do gráfico, e que o gráfico ideal a ser analisado tem um total de cerca de 30 pontos de análise em um papel de 8 polegadas.

Visto que a implementação deste algoritmo em computador não possui a limitação física, a Regra de Neutralidade será atualizada para atualizar o Movimento Direcional usado para julgar as monowaves a cada novo Movimento Direcional que ocorrer. Dessa forma, todas as monowaves serão julgadas pelo Movimento Direcional que ocorreu imediatamente próximo delas.

## 3. Fatores de Estudo

As Extensões de Neely não especificam algumas configurações que podem alterar a análise resultante. O preço a ser utilizado é definido por Neely como sendo o de Negociação (*Cash Market*). Neely cita que Mercados de Negociação costumam gerar uma única cotação por dia, e essa cotação deve ser usada para compor as análises. Porém, Neely também cita que alguns Mercados de Negociação operam continuamente, e que nesses casos, uma opção é escolher um período de tempo para a análise, e usar a média entre o máximo e mínimo do período.

Uma ressalva feita por Neely é de que mercados internacionais devem ser observados somente durante o período em que o mercado está aberto no país onde se está fazendo a negociação.

- Preço da Ação
  - Média Agregada (HLC)
  - Mediana (HL)
  - Preço de Fechamento (C)
- Atualização da Regra de Neutralidade
  - Atualiza fator a cada novo Movimento Direcional
  - Não atualizar fator
  - Não usar Regra de Neutralidade
- Escala de Tempo
  - Todas as escalas devem usar mesma configuração?
  - Escalas diferentes usam configurações diferentes?

Após a finalização do algoritmo, serão feitas análises de ações com diferentes configurações, e serão observadas as diferenças nos resultados obtidos para cada configuração. O objetivo destes testes será analisar o efeito e impacto de cada configuração sobre a análise resultante.

## 4. Arquitetura

Nesta seção será descrita a arquitetura do programa que implementa o algoritmo estudado.

### 4.1. Contexto do Sistema

Um programa capaz de implementar a classificação das Ondas de Elliott de forma consistente e repetível pode se tornar uma forte ferramenta na análise de variações de preços de ações no mercado financeiro. A característica programática das Extensões de Neely tornam o método de Elliott-Neely um ótimo candidato para elaboração de um sistema de análise das variações de preços de ações no mercado financeiro.

### 4.2. Requisitos

Nesta seção são delineados os requisitos para o sistema a ser implementado. Inicialmente, a visão arquitetural e as partes interessadas são apresentadas, seguidas dos requisitos de alto nível e casos de uso. Finalmente, os requisitos funcionais, não-funcionais e evolutivos são apresentados.

#### 4.2.1. Visão arquitetural

Nesta parte os principais conceitos arquiteturais serão destacados e explicados.

A entrada utilizada no sistema são os preços de **ações**, explicados na Seção 1.1.1. O sistema atualmente utiliza os dados exportados de outros softwares em um arquivo .CSV (*comma separated values*), porém há planos de configurar o programa para obter os dados atualizados em tempo real.

A **interface gráfica** é a parte da aplicação através da qual o usuário final interage com o sistema e acessa suas funcionalidades.

O **algoritmo Elliott-Neely** é a parte crítica do sistema, sendo a principal responsável pela análise dos preços de ações alimentadas no sistema. O algoritmo é discutido na Seção 2 e sua implementação programática é o objeto de estudo deste Mestrado.

O **relatório final** é o resultado da análise fornecida pelo sistema. Ele será implementado na forma gráfica, em notações de Elliott explicadas na Seção 1.2. Caso a conclusão do algoritmo seja antecipada ao cronograma, será implementada a emissão de um relatório escrito.

#### 4.2.2. Stakeholders

O **usuário final** é o pesquisador/analizador de mercado interessado em realizar uma análise das Ondas de Elliott nos preços de uma ação do mercado financeiro. Seus interesses são obter um resultado confiável dos dados analisados e conseguir utilizar o sistema sem grandes dificuldades.

O **desenvolvedor** é um pesquisador/programador interessado em atualizar e/ou modificar o sistema, podendo por exemplo, adicionar novos métodos de análise. Seus interesses são a manutenibilidade do sistema e a testabilidade do sistema e de seus elementos.

A Tabela 4 mostra a distribuição dos interesses de cada *stakeholder*, assim como o interesse total de cada atributo do sistema. Cada *stakeholder* é igualmente importante, então, cada um recebe 100 pontos para distribuir entre seus interesses.

**Tabela 4. Representação Quantitativa dos Interesses dos Stakeholders**

Stakeholders	Confiabilidade	Manutenibilidade	Usabilidade	Testabilidade
Usuário Final	80		20	
Desenvolvedor	30	40		30
Total	110	40	20	30

#### 4.2.3. Atributos de Qualidade

Os atributos chaves de qualidade são aqueles que concentram o maior interesse dos *stakeholders* (Tabela 4).

**Confiabilidade** é o atributo chave deste sistema. Por ser um sistema de análise de dados, a confiabilidade dos resultados está diretamente ligada a qualidade do sistema em si.

**Manutenibilidade** é um atributo secundário deste sistema. A medida que novos pesquisadores se interessem no sistema, a facilidade de extensão, otimização e manutenção se toram chaves em manter o interesse dos novos desenvolvedores.

**Testabilidade** permitirá que desenvolvedores tenham mais confiança a cerca do sistema e das modificações feitas nele.

**Usabilidade** facilitará a interação dos usuários com o sistema.

#### 4.2.4. Requisitos de Alto-Nível

Os seguintes requisitos representam as funcionalidades de alto-nível do sistema. Cada requerimento de alto-nível será refinado em detalhe por vários requisitos específicos e técnicos.

##### Análise de Dados

HL-1	Obrigatório	O sistema deve permitir a importação dos dados de análise.
------	-------------	--

### Análise de Dados

HL-2 Obrigatório O sistema deve fazer uma análise das Ondas de Elliott de acordo com as Extensões de Neely de forma precisa e confiável.

### Configuração da Análise

HL-3 Obrigatório O sistema deve exibir os resultados da análise de forma legível para o usuário.

### Exibição de Resultados

HL-4 Obrigatório O sistema deve permitir a mudança da configuração da análise a ser feita sobre os dados.

#### 4.2.5. Casos de Uso

O diagrama de casos de uso é ilustrado na Figura 12, demonstrando os mais importantes casos de uso e seus atores.

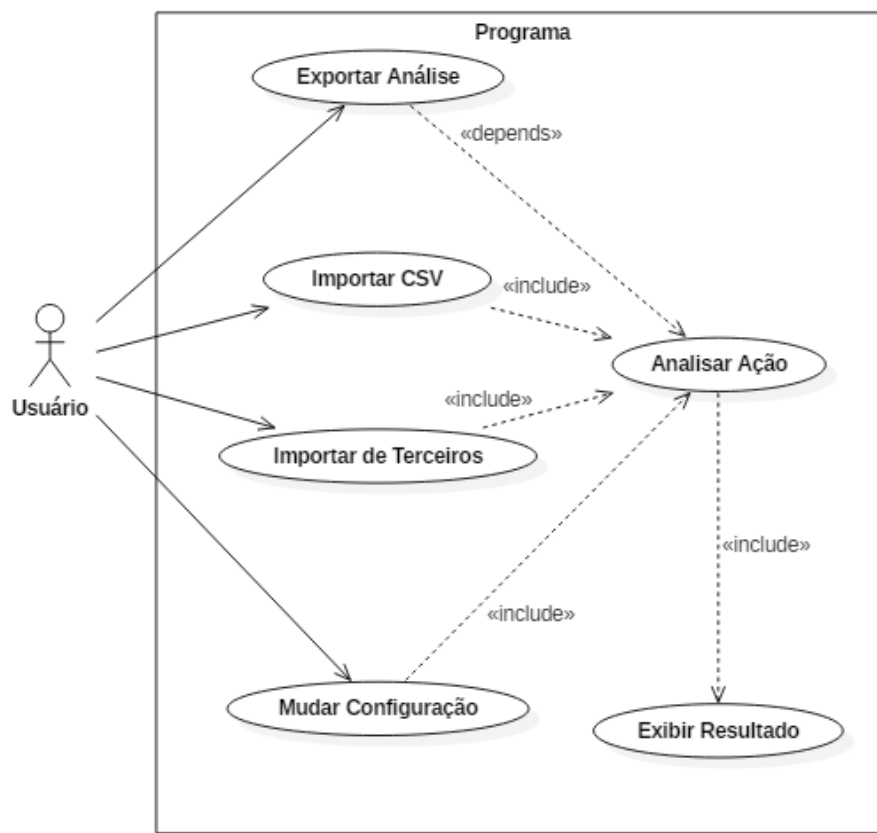


Figura 12. Diagrama de Casos de Uso

Além disso, as Tabelas a seguir mostram os casos de uso arquiteturalmente sig-



nificantes. A numeração refere-se ao requerimento de Alto-nível, por exemplo, UC-1.x refere-se a HL-1.

**Tabela 5. UC-1.1 - Importar .CSV**

### **UC-1.1 - Importar .CSV**

<b>Ator Primário</b>	Usuário Final
<b>Objetivo</b>	Importar um Arquivo .CSV com preços de ação a serem analisados
<b>Condições Iniciais</b>	O sistema está operacional, e o usuário final possui o arquivo com o histórico da Ação a ser analisada
<b>Cenário Principal de Sucesso</b>	
<ol style="list-style-type: none"><li>1. O usuário clica na opção de Inserir o Arquivo .CSV com os dados da Ação</li><li>2. O usuário localiza o arquivo .CSV a ser analisado</li><li>3. O sistema valida o arquivo .CSV</li><li>4. O sistema realiza a análise do histórico de preços da Ação (UC-2.1)</li></ol>	
<b>Extensões</b>	
<ol style="list-style-type: none"><li>3.a O sistema identifica que o arquivo .CSV não tem as colunas necessárias<ol style="list-style-type: none"><li>3.a.1 O sistema não analisa o arquivo .CSV</li><li>3.a.2 O sistema emite um aviso de que o arquivo .CSV não é válido</li><li>3.a.3 O usuário obtém um arquivo .CSV válido</li><li>3.a.4 Ir para o passo 1</li></ol></li></ol>	
<b>Sub-variações</b>	
<b>Condições Finais</b>	O histórico de preços da ação é analisado e o usuário final pode alterar as configurações de análise.
<b>Requisitos Relacionados</b>	HL-1, FR-1, FR-2

**Tabela 6. UC-1.2 - Importar de Terceiros**

## **UC-1.2 - Importar de Terceiros**

<b>Ator Primário</b>	Usuário Final
<b>Objetivo</b>	Importar os preços de ação a serem analisados via terceiros
<b>Condições Iniciais</b>	O sistema está operacional, e o usuário final possui uma fonte de dados alternativa
<b>Cenário Principal de Sucesso</b>	<ol style="list-style-type: none"> <li>1. O usuário configura o programa para estabelecer a conexão com o terceiro</li> <li>2. O sistema valida as configurações da conexão com o terceiro</li> <li>3. O sistema requisita os dados do terceiro</li> <li>4. O sistema realiza a análise do histórico de preços da Ação (UC-2.1)</li> </ol>
<b>Extensões</b>	<ol style="list-style-type: none"> <li>2.a O sistema não consegue realizar a conexão com o terceiro               <ol style="list-style-type: none"> <li>2.a.1 O sistema não realiza a análise</li> <li>2.a.2 O sistema emite um aviso de que a conexão não foi estabelecida</li> <li>2.a.3 Ir para o passo 1</li> </ol> </li> </ol>
<b>Sub-variações</b>	
<b>Condições Finais</b>	O histórico de preços da ação é analisado e o usuário final pode alterar as configurações de análise.
<b>Requisitos Relacionados</b>	HL-1, FR-6

**Tabela 7. UC-2.1 - Analisar Ação**

**UC-2.1 - Analisar Ação**

<b>Ator Primário</b>	Sistema
<b>Objetivo</b>	Analisar o Histórico de Preços de uma Ação
<b>Condições Iniciais</b>	O sistema está operacional, e o arquivo foi inserido pelo usuário e validado pelo sistema
<b>Cenário Principal de Sucesso</b>	<ol style="list-style-type: none"><li>1. O sistema realiza a análise do histórico de preços da Ação.</li><li>2. O sistema exibe os resultados para o usuário (UC-3.1)</li></ol>
<b>Extensões</b>	
<b>Sub-variações</b>	
<b>Condições Finais</b>	A interface exibe o resultado da análise e o usuário pode alterar as configurações de análise.
<b>Requisitos Relacionados</b>	HL-2, FR-3

**Tabela 8. UC-3.1 - Exibir Resultado**

**UC-3.1 - Exibir Resultado**

<b>Ator Primário</b>	Sistema
<b>Objetivo</b>	Exibir o resultado da análise feita
<b>Condições Iniciais</b>	O sistema finalizou uma análise (UC-2.1)
<b>Cenário Principal de Sucesso</b>	<ol style="list-style-type: none"><li>1. O sistema deve exibir o gráfico do histórico do preço da Ação</li><li>2. O sistema deve sobrepor a este gráfico as notações das Ondas de Elliott</li></ol>
<b>Extensões</b>	
<b>Sub-variações</b>	
<b>Condições Finais</b>	O resultado a análise é visível para o usuário final.
<b>Requisitos Relacionados</b>	HL-3, FR-4

**Tabela 9. UC-4.1 - Mudar Configuração**

### **UC-4.1 - Mudar Configuração**

<b>Ator Primário</b>	Usuário Final
<b>Objetivo</b>	Alterar as configurações de análise
<b>Condições Iniciais</b>	Um conjunto de dados já foi inicializado no sistema
<b>Cenário Principal de Sucesso</b>	
<ol style="list-style-type: none"><li>1. O usuário altera uma das configurações disponíveis na interface</li><li>2. O sistema atualiza a análise de acordo com as novas configurações (UC-2.1)</li></ol>	
<b>Extensões</b>	
<b>Sub-variações</b>	
<b>Condições Finais</b>	A análise é atualizada para a nova configuração.
<b>Requisitos Relacionados</b>	HL-4, FR-3, FR-4, FR-5

O Caso de Uso UC-1.2 descrito na Tabela 6 é uma funcionalidade planejada. O objetivo é que os dados sejam obtidos, por exemplo, diretamente através de uma interface TCP/IP, ou através de uma conexão com um programa como o MetaTrader 5. Se esta funcionalidade for implementada, a atualização dos dados e da análise poderá ser feita em tempo real, a medida que novos dados sejam disponibilizados.

#### **4.2.6. Requisitos Funcionais**

Nesta seção, os requisitos que capturam o comportamento esperado do sistema serão apresentados (Tabela 10). Este comportamento pode ser expresso como serviços, tarefas ou funções que o sistema deve executar. Estes são derivados a partir dos requisitos de Alto-nível e dos casos de uso.

**Tabela 10. Requisitos Funcionais**

FR-1	Obrigatório	O sistema deve validar se a entrada é válida.
FR-2	Obrigatório	O sistema deve informar o usuário caso a entrada seja inválida.
FR-3	Obrigatório	O sistema deve realizar a análise automaticamente.
FR-4	Obrigatório	O sistema deve exibir o resultado ao usuário automaticamente.
FR-5	Obrigatório	O sistema deve atualizar automaticamente a análise quando o usuário alterar as configurações.
FR-6	Desejável	E desejável que o sistema consiga obter os dados através de uma conexão com um terceiro.

### **4.3. Análise**

Esta seção foca em esclarecer o processo de desenvolvimento do sistema. Aqui, todas as suposições que são relevantes ao processo são mostradas, assim como o roteiro tecnológico de desenvolvimento e as decisões de alto-nível tomadas durante o projeto.

#### **4.3.1. Suposições**

- O usuário final tem conhecimento sobre o mercado de ações (Seção 1.1);
- O usuário final tem conhecimento sobre a representação dos preços de ações (Seção 1.1.1);
- O usuário final tem conhecimento sobre as Ondas de Elliott (Seção 1.2);
- O usuário final tem a sua disposição um software capaz de exportar os preços da ação que quer analisar em formato .CSV;

#### **4.3.2. Roteiro tecnológico**

A primeira versão do sistema foi desenvolvida somente para gerar a representação gráfica do histórico de preços, e implementar mudanças de configurações básicas. Esta versão teve como objetivo demonstrar que o ambiente de desenvolvimento era favorável e compatível com a aplicação a ser criada.

A segunda versão do sistema está sendo desenvolvida para implementar a análise de preços de ações no mercado financeiro conforme o método de Elliott-Neely. Esta versão será capaz de realizar uma análise das Ondas de Elliott de forma automática, com mínima ou nenhuma intervenção do usuário.

Versões futuras do sistema serão capazes de realizar previsões futuras dos preços das ações a partir, primariamente, da verificação dos padrões das Ondas de Elliott.

#### 4.4. Arquitetura do Software

A arquitetura do software é descrita nesta seção. Esta seção é composta dos principais motivadores arquiteturais, seguidos das visões arquiteturais. A representação foi criada baseada no modelo 4+1 [17], *i.e.*, consistindo na visão lógica, visão de processo, visão de desenvolvimento, visão física e visão de caso de uso. Para fortalecer o conceito original do modelo 4+1, cada visão foi projetada usando UML [18], conforme recomendado em [19].

##### 4.4.1. Visão Lógica

Esta visão mostra as abstrações chaves e mecanismos que são usados dentro do programa para realizar suas funcionalidades. A interação entre estes elementos também é descrita. Os elementos estão alocados em camadas de acordo com suas responsabilidades. A representação gráfica desta visão pode ser observada na Figura 13. Foi utilizado o padrão MVC para esta Visão [20] [21].

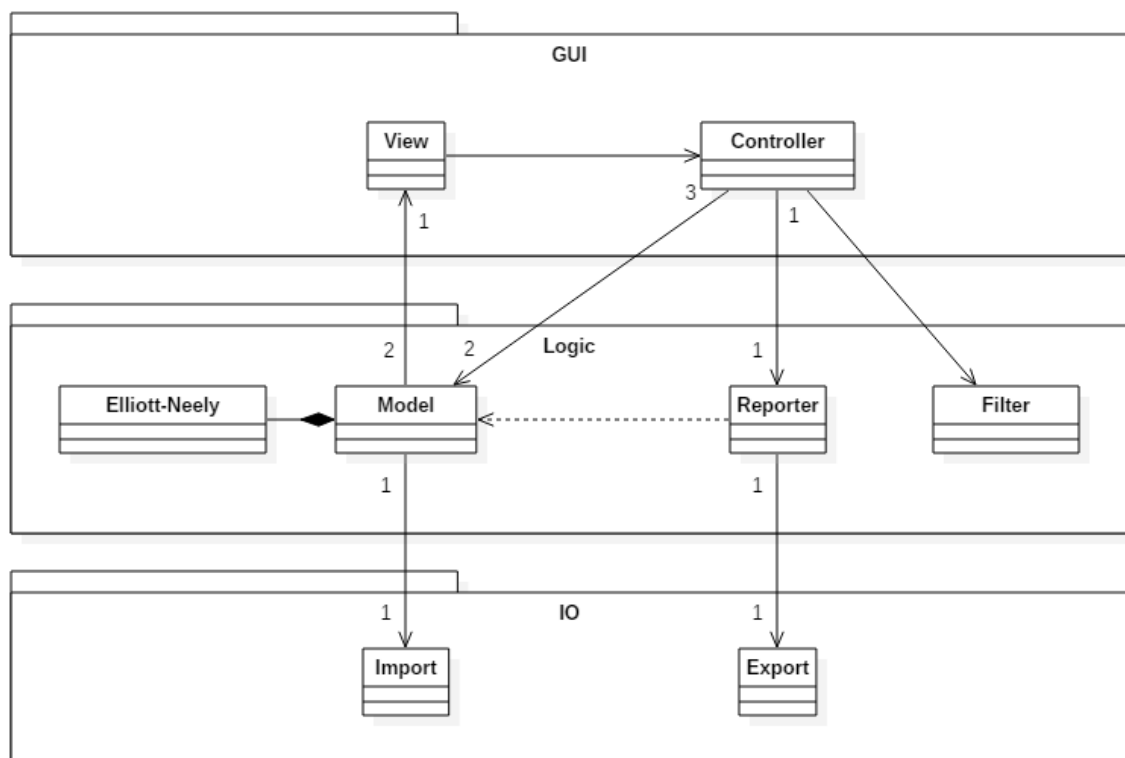


Figura 13. Visão Lógica

- **GUI:** Interface com Usuário
  - A **Visualização (View)** é a parte da camada de interface com a qual o usuário interage diretamente, acessando as funcionalidades do programa.
  - O **Controlador (Controller)** são os dispositivos lógicos e de controle que realizam as funcionalidades do programa. O programa utiliza 3 controladores: Global (para funcionalidades gerais), Data (para processamento de dados) e UI (para interfaceamento gráfico).

- **Logic:** Lógica
  - O **Modelo** (*Model*) é a parte da camada lógica onde se encontram os dados sendo processados pelo programa ou visualizados na interface. Estão sendo utilizados dois Modelos: um de dados (sob o qual o Método de **Elliott-Neely** é aplicado), e um de visualização, que é enviado para a **GUI**.
  - O **Relator** (*Reporter*) é a parte responsável por criar uma relatório para o usuário.
  - O **Filtro** (*Filter*) é responsável por filtrar quais informações serão disponibilizadas para o usuário de acordo com a configuração escolhida.
- **IO:** Entradas e Saídas
  - A **Importação** (*Import*) é responsável por receber os dados do usuário.
  - A **Exportação** (*Export*) é responsável por enviar dados para o usuário.

#### 4.4.2. Visão de Processo

Esta visão mostra a representação abstrata do processo principal no sistema. Para apresentar estas representações, os casos de uso descritos na Seção 4.2.5 foram utilizados para retratar as partes envolvidas na realização de cada caso de uso.

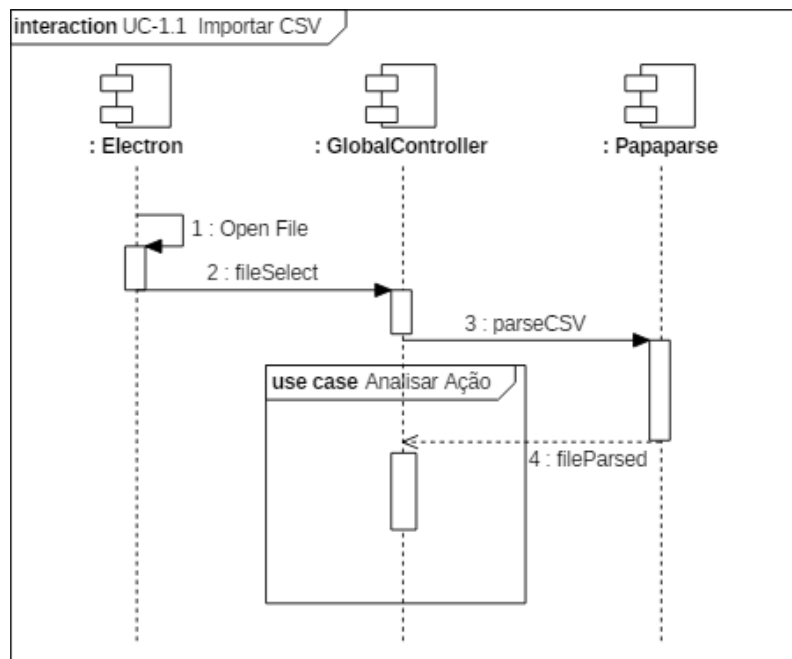


Figura 14. Visão de Processo para UC-1.1: Importar CSV



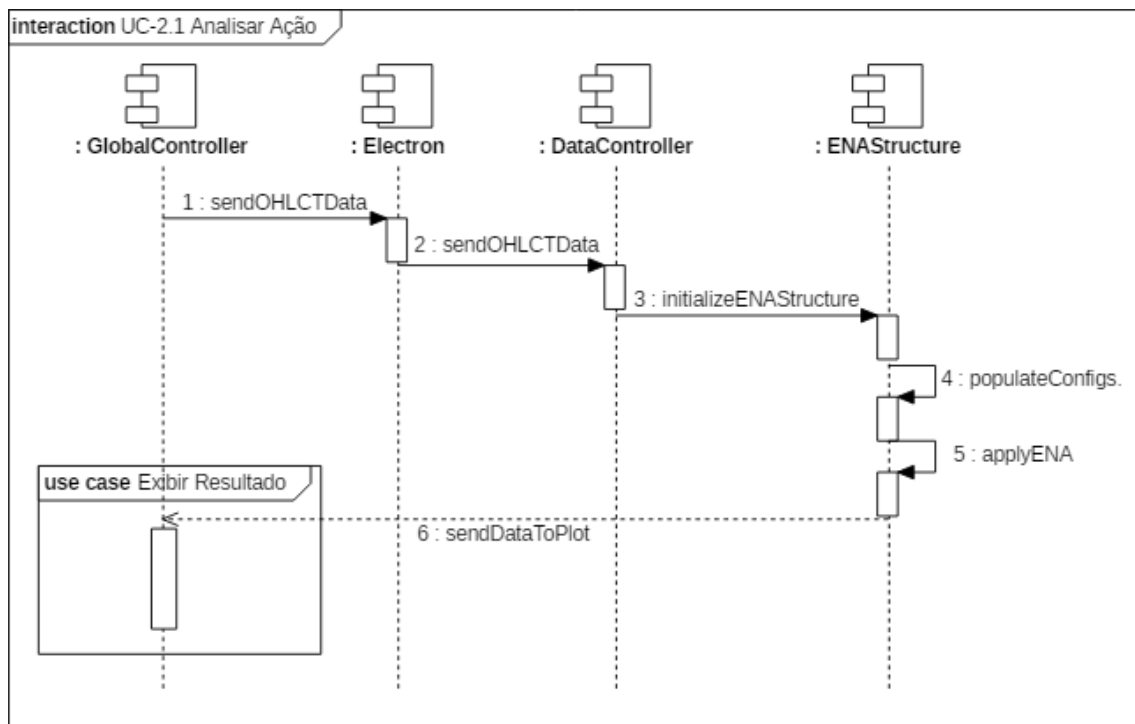


Figura 15. Visão de Processo para UC-2.1: Analisar Ação

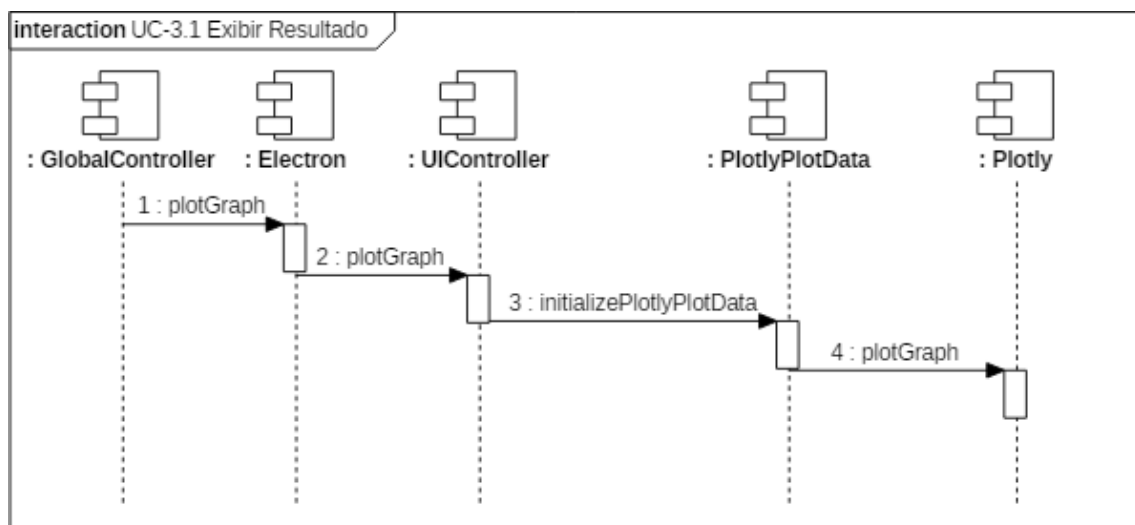
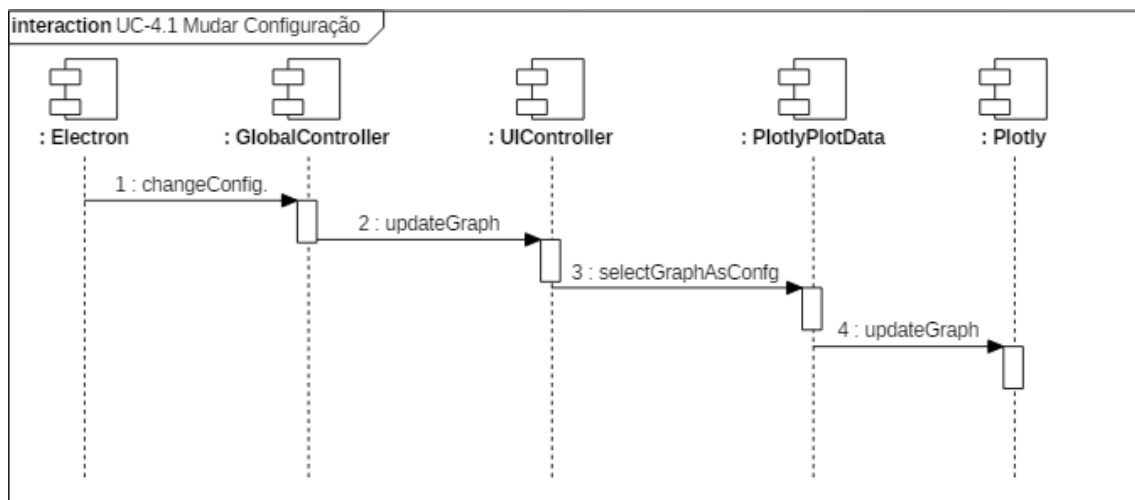


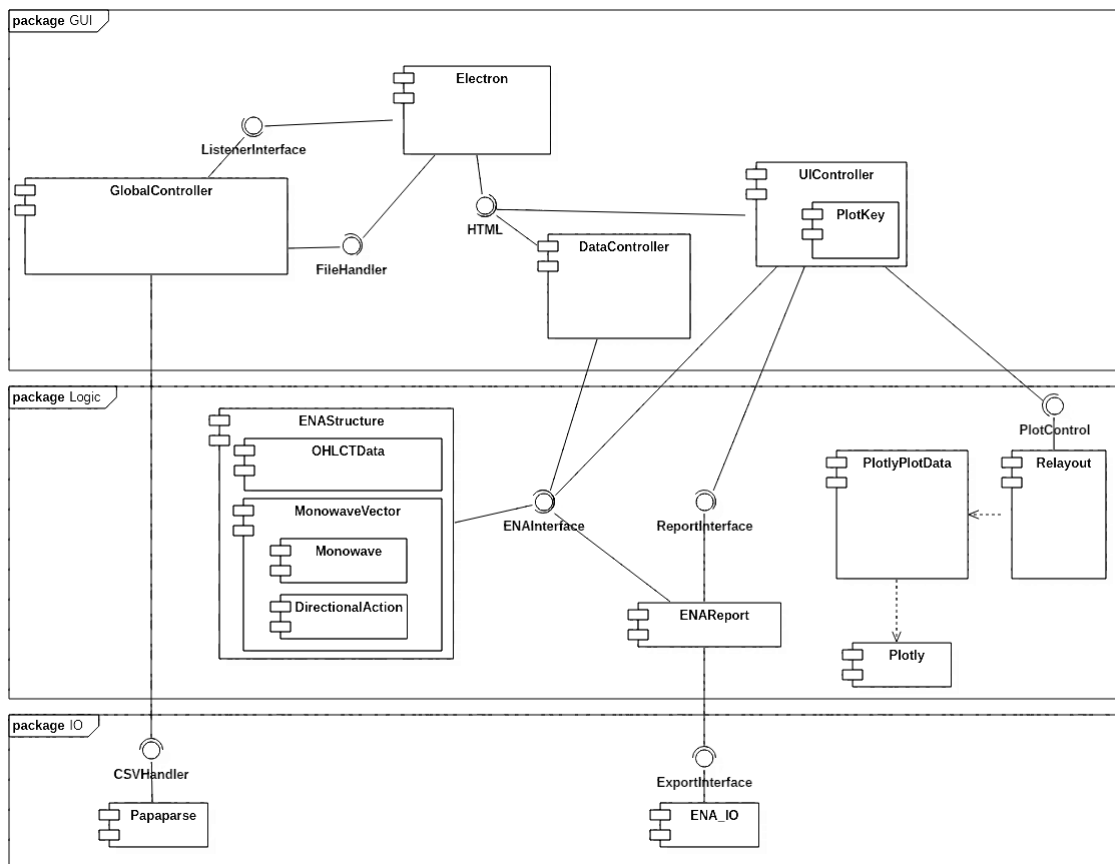
Figura 16. Visão de Processo para UC-3.1: Exibir Resultado



**Figura 17. Visão de Processo para UC-4.1: Mudar Configuração**

#### **4.4.3. Visão de Implementação**

Essa visão foca no gerenciamento de configuração e organização real do módulo de software no ambiente de desenvolvimento, englobando os principais componentes usados para montar e gerar o sistema físico. Nas seções a seguir, esses componentes são decompostos e descritos em termos de responsabilidades e interfaces. A representação gráfica desta visão pode ser observada na Figura 18.



**Figura 18. Visão de Implementação**

Foi usado um padrão de 3 Módulos para implementação do sistema. Cada módulo age como um controlador de responsabilidades específicas. O Controlador de Dados (*DataController*) gerencia os dados da aplicação: neste caso, o modelo de dados estudados (o *ENAStructure*). O Controlador de UI gerencia a parte de interface visual com o usuário: plota o gráfico (utilizando a biblioteca Plotly [22]). O Controlador Global (*GlobalController*) gerencia os outros Controladores, eventos (e.g. clique em botão, seleção de opção, entrada de arquivo), e faz a conversão do arquivo .CSV em um objeto JavaScript (usando a biblioteca PapaParse [23]).

- **Electron**
  - É o framework usado no desenvolvimento, e o programa onde a aplicação do sistema executa.
  - Ele implementa as interfaces com usuário através de HTML e CSS, funcionalidades de controle através dos observadores de evento (*ListenerInterface*), e importa os dados através da interface *FileHandler*.
- **GlobalController**
  - Controlador geral da aplicação.
  - Responsabilidades:
    - \* Implementar e gerenciar eventos (e.g. clique em botão, seleção de opção, entrada de arquivo) via **Electron** usando a interface **ListenerHTML**.

- \* Receber os dados de entrada do usuário a partir da interface **FileHandler**.
  - \* Converter o arquivo .CSV do usuário usando a biblioteca **Papaparse** via interface **CSVHandler**.
  - \* Gerenciar os outros Controladores por meio do **Electron**.
- **DataController**
  - Gerenciador dos dados da aplicação.
  - Responsabilidades:
    - \* Receber os dados entrados pelo usuário via interface **HTML**.
    - \* Salvar os dados entrados pelo usuário no modelo **ENAStructure** via interface **ENAInterface**.
- **UIController**
  - Gerenciador da visualização e interface com usuário.
  - Possui uma estrutura chamada **PlotKey** que registra as modificações feitas nas configurações de exibição quando o programa solicitar uma atualização do gráfico.
  - Responsabilidades:
    - \* Receber os dados para plotar o gráfico via interface **ENAInterface**.
    - \* Plotar o gráfico usando a biblioteca **Plotly** via interface **PlotControl**.
    - \* Enviar os dados para geração do relatório da análise para o componente **ENAREport** via interface **ReportInterface**.
- **ENAStructure**
  - Organiza os dados do usuário de acordo com as configurações possíveis.
  - Os dados do usuário são salvos na estrutura **OHLCTStructure**.
  - A análise gerada é feita e salva na estrutura **MonowaveVector**.
  - A estrutura **MonowaveVector** realiza a análise através da organização de um vetor de **Monowave(s)** e de uma estrutura de **DirectionalAction**.
  - Para cada combinação de configurações disponíveis, são gerados um par de **OHLCTData** e **MonowaveVector**
- **ENAREport**
  - Gera um relatório da análise para o usuário.
  - O relatório gerado é exportado para o usuário através do módulo **ENA\_IO** via interface **ExportInterface**.
- **Relayout**
  - Guarda o histórico de visualizações do gráfico, permitindo desfazer e refazer comandos de Zoom.
  - Permite aplicar um comando de Fit vertical para melhor utilizar o espaço horizontal do gráfico.
- **PlotlyPlotData**
  - Guarda os dados da **ENAStructure** formatados para visualização usando a biblioteca **Plotly**.

#### 4.4.4. Visão de Implantação

Esta visão foca na distribuição do software no hardware. Ela apresenta o hardware e as peças de software que o compõem. A seguir, cada parte do software e hardware serão explicadas. A representação gráfica desta visão pode ser observada na Figura 19.

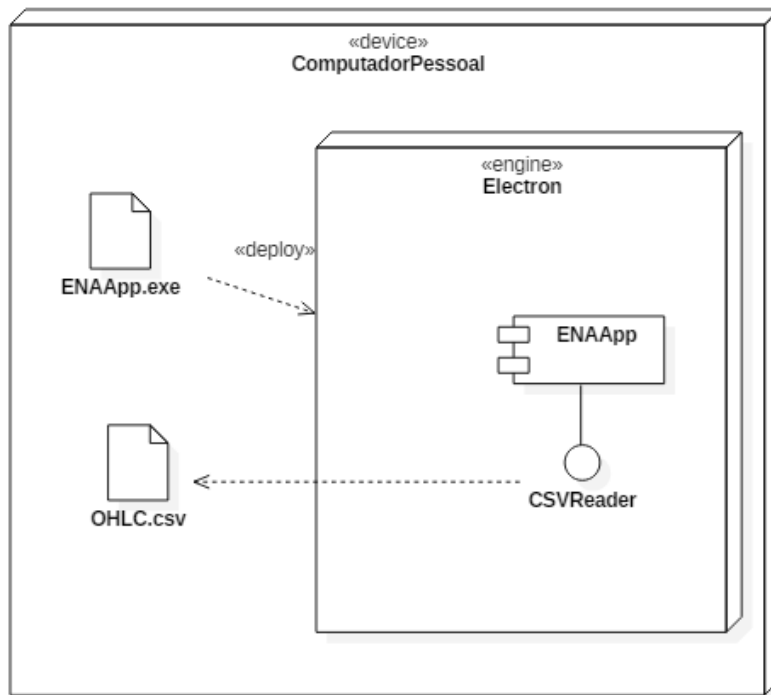


Figura 19. Visão de Implantação

O **computador pessoal** representa qualquer computador pessoal rodando um sistema operacional Windows, Linux ou MacOS.

O **ENApp.exe** é o arquivo executável gerado a partir do projeto. Ele está listado como “.exe” a título de exemplo e simplificação, pois o Electron é capaz de gerar programas para vários sistemas operacionais utilizando o mesmo código fonte.

O executável cria uma instância do **Electron**, na qual a aplicação “Elliott-Neely Analysis App” (**ENApp**) executa. A aplicação é capaz de importar arquivos .csv através da interface **CSVReader**. Na visualização, o arquivo OHLC.csv é usado como exemplo.

#### 4.4.5. Decisões de Desenvolvimento de Software

Esta seção lista as decisões relacionadas ao desenvolvimento do software e implementação. Elas podem ser vistas nas Tabelas seguintes. O modelo utilizado para estas decisões foi baseado em [24].

**Tabela 11. Decisão - Linguagem de Implementação**

<b>Nome</b>	Linguagem de Implementação
<b>Decisão</b>	1
<b>Estado</b>	Aprovada
<b>Problema</b>	Qual linguagem de programação utilizar para implementar o algoritmo
<b>Decisão</b>	Electron ( <i>framework</i> ) + JavaScript (lógica) + HTML5 & CSS (interface)
<b>Alternativas</b>	C, Java, C#, Ruby, Python
<b>Argumentos</b>	<p>JavaScript é uma linguagem interpretada, com interpretadores disponíveis em vários sistemas operacionais, muitos destes possuindo código aberto com licenças permissivas (como o Chrome V8 [25]).</p> <p>Node.js [26] é um <i>runtime</i> para JavaScript que implementa uma modificação da <i>engine</i> Chrome V8, permitindo a execução de código em JavaScript sem a necessidade de um navegador web.</p> <p>NPM [27] é um catálogo de bibliotecas e projetos em JavaScript, reunindo mais de 60 mil módulos, com bilhões de downloads semanais [28] e controle de versão para cada módulo.</p> <p>Electron [29] é um <i>framework</i> que permite a criação de programas em JavaScript + HTML5 &amp; CSS capazes de rodar em diversos sistemas operacionais sem problemas de compatibilidade.</p> <p>JavaScript possui tipificação dinâmica de variáveis, o que permite uma maior flexibilidade na criação e manuseio de dados.</p> <p>Este ecossistema permite a criação de aplicações altamente flexíveis e compatíveis, com acesso a um grande acervo de bibliotecas e módulos para serem integrados, sem problemas de licenciamento e com controle de versão individual por módulo.</p>

**Tabela 12. Decisão - Análise de Diferentes Configurações**

<b>Nome</b>	Análise de Diferentes Configurações
<b>Decisão</b>	2
<b>Estado</b>	Aprovada
<b>Problema</b>	Como fazer a análise para diferentes configurações
<b>Decisão</b>	Fazer a análise para todas as configurações possíveis e somente alterar qual está visível
<b>Alternativas</b>	<ol style="list-style-type: none"><li>1. Realizar a análise somente para a configuração atual e refazer a análise para cada mudança na configuração</li><li>2. Registrar cada análise feita e somente refazer a análise caso uma nova configuração seja escolhida</li><li>3. Fazer a análise para todas as configurações possíveis e somente alterar qual está visível</li></ol>
<b>Argumentos</b>	<p>A opção 1 implica em um menor gasto inicial de recursos, já que seria feita somente uma análise, mas causaria desperdício de recursos quando fosse solicitada a análise de uma configuração já visualizada anteriormente.</p> <p>A opção 2 levaria implicaria na necessidade de se implementar um controle adicional para verificar se a análise já foi feita antes.</p> <p>A opção 3 implica em um maior gasto inicial de recursos e facilita a mudança entre configurações.</p> <p>Visto que a implementação deste sistema visa a exploração das diferentes configurações, foi decidido pela terceira opção.</p>

## 5. Cronograma e Progressão

Nesta seção são listadas as atividades previstas para realização do trabalho proposto. Na Tabela 13 é apresentado o cronograma para estas atividades. Na Tabela 14 são listadas as atividades individuais envolvidas na elaboração e programação do algoritmo estudado.

1. Obtenção dos créditos referentes as disciplinas do Mestrado;
2. Pesquisa bibliográfica;
3. Redação da monografia para o Exame Geral de Qualificação;
4. Exame Geral de Qualificação;
5. Elaboração do algoritmo proposto e programação;
6. Geração e execução de casos de testes, e avaliação do sistema;
7. Análise dos efeitos das configurações sobre os resultados do algoritmo;
8. Redação final da Dissertação; e
9. Defesa do Mestrado.

Atividades	2018					2019										2020								
	Agosto	Setembro	Outubro	Novembro	Dezembro	Janeiro	Fevereiro	Março	Abril	Maio	Junho	Julho	Agosto	Setembro	Outubro	Novembro	Dezembro	Janeiro	Fevereiro	Março	Abril	Maio	Junho	Julho
1. Créditos																								
2. Pesquisa bibliográfica																								
3. Escrita da qualificação																								
4. Exame de qualificação																								
5. Elaboração do prog.																								
6. Testes																								
7. Análise de config.																								
8. Escrita da dissertação																								
9. Defesa do mestrado																								

**Tabela 13. Cronograma de atividades**



Tópico	Atividade	Descrição	Progresso	Estimado
Aquisição de Dados	Importação de Dados	Importação dos Dados OHLC em arquivos .CSV exportados a partir do Programa MetaTrader 5. Utilizando a biblioteca <b>Papaparse</b> .	100%	-
Pré-Tratamento de Dados	Passagem dos Dados para Classe OHLCT	Transferência dos dados importados do MetaTrader 5 para a <b>Classe OHLCT</b>	100%	-
	Escalas de Tempo	Replica dados originais em diferentes escalas de tempo. Implementados: 1 min, 1 hora, 1 dia. Escalável para fácil configuração de novas escalas.	100%	-
	Valor Típico	Cálculo do Valor Típico sobre o qual a análise será feita. Implementados: Valor Agregado (HLC), Mediana (HL). Escalável para fácil configuração de novos valores típicos.	100%	-
Método Elliott-Neely Análise Preliminar (1ª parte)	Regra de Neutralidade (1º passo)	Aplicação da primeira passagem da <b>Regra de Neutralidade</b> . Implementado através das <b>Classes Monowave</b> e <b>Monowave Vector</b> .	100%	-
	Regra de Proporção (Seleção da Janela de Análise)	Atualizada para ambiente digital e alta quantidade de dados. Afeta a segunda passagem da <b>Regra de Neutralidade</b> e a <b>Regra de Retracement</b>	100%	-
	Regra de Neutralidade (2º Passo)	Une <b>monowaves</b> e/ou altera início/fim de <b>monowaves</b> de acordo com a <b>Regra de Proporção</b> .	100%	-
	Regra de Observação	Aplicada de forma emergente através da <b>Classe Monowave Vector</b> .	100%	-
	Regra de Retracement	Classificação de <b>Monowaves</b> .	0%	3 semanas
Método Elliott-Neely Observações Intermediárias (2ª parte)	Regra de Similaridade e Balanço	Indica quais <b>Monowaves</b> pertencem ao mesmo grupo/grau para efeito de agrupamento.	0%	2 semanas
Método Elliott-Neely Considerações Centrais (3ª parte)	Construção de Polywaves	Classifica conjuntos de <b>Monowaves</b> nos padrões das <b>Ondas de Elliott</b> .	0%	3-4 semanas
	Extensões de Neely	Aplica regras de compactação que transformam grupos de <b>Monowaves</b> em <b>Monowaves</b> de grau maior.	0%	3-4 semanas
Interface com Usuário	Visualização em Gráfico	Utilizando a biblioteca <b>Plotly.js</b>	100%	-
	Configurações	Eixo Vertical (Log ou Linear), Eixo Horizontal (Tempo ou Sequencial), Tipo de Marcador (OHLC, Candelstick ou Nenhum), Resolução do Tempo (1 Minuto, 1 Hora, 1 Dia), Valor Típico (HLC ou HL), Monowave (Visualizar ou Esconder)	100%	-

**Tabela 14. Progresso do algoritmo**

## Referências

- [1] Masur. R.N. Elliott, “The Basis of the Wave Principle”, October 1940. URL <https://commons.wikimedia.org/w/index.php?curid=1599960>. Acessado em 2019-07-17.
- [2] N. A. Curott E. P. Stringham. *The Oxford Handbook of Austrian Economics*. Oxford University Press, 2005.
- [3] A. Greif. *Institutions and the Path to the Modern Economy: Lessons from Medieval Trade*. Cambridge University Press, 2006.
- [4] U. Malmendier. Law and finance “at the origin”. *Journal of Economic Literature*, 47(4):1076–1108, 2009. doi: 10.1257/jel.47.4.1076. URL <http://dx.doi.org/10.1257/jel.47.4.1076>.
- [5] J. Hasanhodzic A. W. Lo. *The Evolution of Technical Analysis: Financial Prediction from Babylonian Tablets to Bloomberg Terminals*. John Wiley & Sons, 2010.
- [6] S. Shimizu. *The Japanese Chart of Charts*. Tokyo Futures Trading Publ. Co., 1986.
- [7] J. A. Dahlquist C. D. Kirkpatrick. *Technical Analysis: The Complete Resource for Financial Market Technicians*. FT Press, 2006.
- [8] A. J. Frost R. R. Prechter. *Elliott Wave Principle: Key to Market Behavior, Tenth Edition*. New Classics Library, 2005.
- [9] C. J. Collins R. N. Elliott. *The Wave Principle*. 1938.
- [10] MacroTrends. Dow Jones 100 Year Historical Chart. URL <https://www.macrotrends.net/1319/dow-jones-100-year-historical-chart>. Acessado em 2019-07-17.
- [11] B. Mandelbrot e R. L. Hudson. *The (mis)Behavior of Markets*. Basic Books, 2004.
- [12] B. Mandelbrot. A Multifractal Walk Down Wall Street. *Scientific American*, page pg 70, 1999.
- [13] B. Mandelbrot. A Multifractal Walk Down Wall Street. URL [https://web.archive.org/web/20070207113959/http://www.elliottwave.com/education/SciAmerican/Mandelbrot\\_Article2.htm](https://web.archive.org/web/20070207113959/http://www.elliottwave.com/education/SciAmerican/Mandelbrot_Article2.htm). Acessado em 2019-07-18, Arquivado via Wayback Machine.
- [14] R. R. Prechter. Scientific Controversy: Who Discovered Financial Fractals? URL <https://web.archive.org/web/20070208084007/http://www.elliottwave.com/response/default.htm>. Acessado em 2019-07-18, Arquivado via Wayback Machine.
- [15] D. R. Aronson. *Evidence-Based Technical Analysis: Applying the Scientific Method and Statistical Inference to Trading Signals*. John Wiley and Sons, 2006.
- [16] E. Hall G. Neely. *Mastering Elliott Wave*. Windsor Books for the Elliott Wave Institute, 1990.
- [17] P. Kruchten. The 4+1 view model of architecture. *IEEE Softw.*, 12(6):42–50, November 1995. ISSN 0740-7459. doi: 10.1109/52.469759. URL <http://dx.doi.org/10.1109/52.469759>.
- [18] OGM. Unified Model Language. URL <https://www.uml.org/>. Acessado em 2019-07-16.
- [19] *Applying 4+1 View Architecture with UML 2*. First FCG Software Services, 2007. URL [http://www.sparxsystems.com/downloads/whitepapers/FCGSS\\_US\\_WP\\_Applying\\_4+1\\_w\\_UML2.pdf](http://www.sparxsystems.com/downloads/whitepapers/FCGSS_US_WP_Applying_4+1_w_UML2.pdf).

- [20] S. T. Pope G. E. Krasner. A cookbook for using the model-view controller user interface paradigm in Smalltalk-80. *Journal of Object-Oriented Programming*, 1 (3):26–49, 1988.
- [21] F. Buschmann. *Pattern-Oriented Software Architecture: A System of Patterns*. John Wiley & Sons, 1996.
- [22] Plot.ly. Plotly JS. URL <https://plot.ly/javascript/>. Acessado em 2019-07-15.
- [23] Matt Holt. Papa Parse 5. URL <https://www.papaparse.com/>. Acessado em 2019-07-15.
- [24] A. Ackerman J. Tyree. Architecture Decisions: De-mystifying Architecture. *IEEE Software*, pages 19–27, 2005.
- [25] Google. Engine Chrome V8. URL <https://v8.dev/>. Acessado em 2019-07-15.
- [26] Node.js Foundation. Node.js. URL <https://nodejs.org/>. Acessado em 2019-07-15.
- [27] npm, Inc. Node Pack Manager, . URL <https://www.npmjs.com/>. Acessado em 2019-07-15.
- [28] npm, Inc. NPM Weekly 133, 2018-02-15, . URL <https://medium.com/npm-inc/npm-weekly-133-billions-of-packages-downloaded-621a5196593>. Acessado em 2019-07-15.
- [29] Electron. Electron. URL <https://electronjs.org/>. Acessado em 2019-07-15.

Apêndices

A. Padrões das Ondas de Elliott

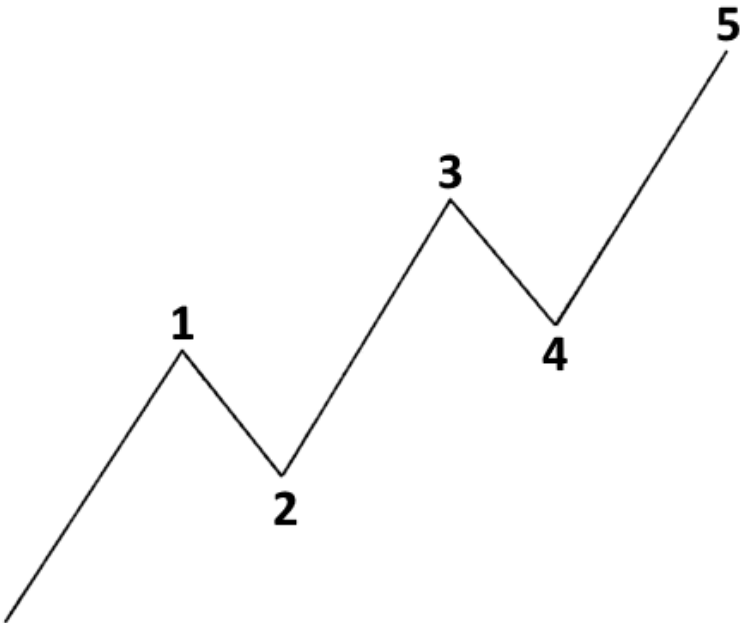


Figura 20. Padrão de Onda Motivadora: Impulso

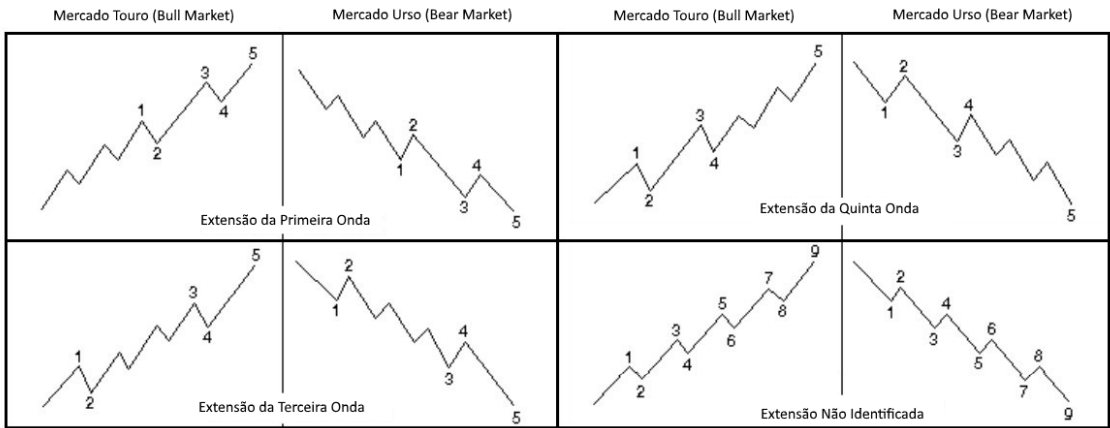


Figura 21. Padrão de Onda Motivadora: Extensão

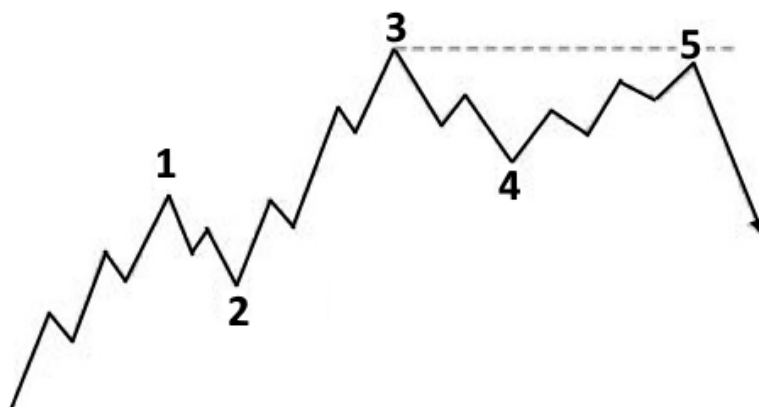


Figura 22. Padrão de Onda Motivadora: Truncagem

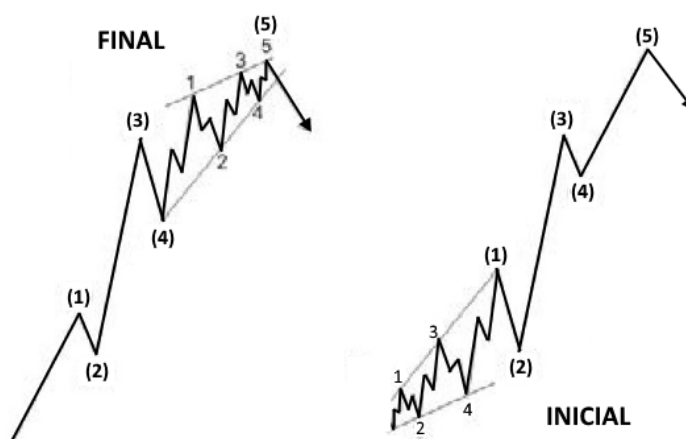


Figura 23. Padrão de Onda Motivadora: Diagonal Final/Inicial

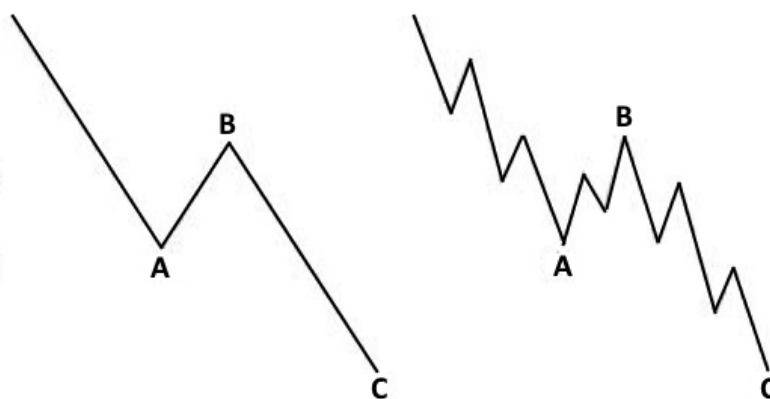
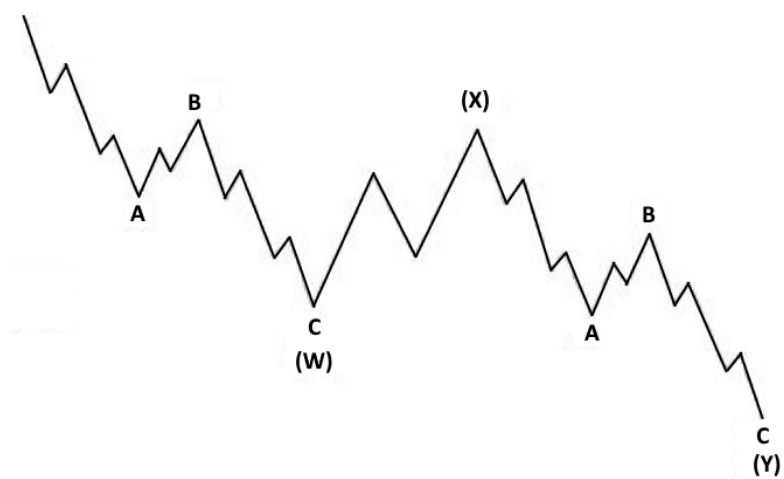
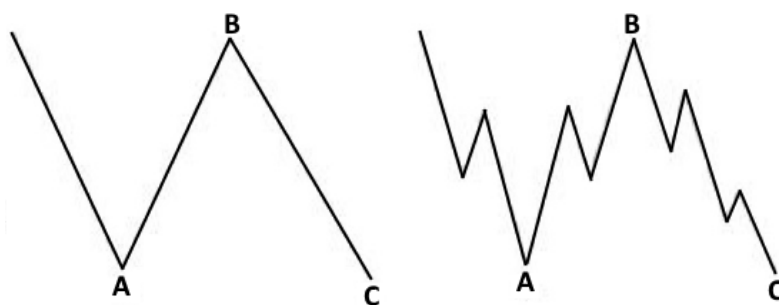


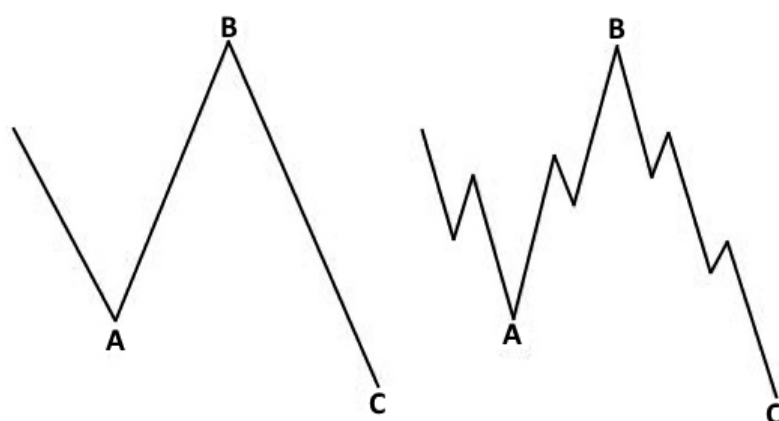
Figura 24. Padrão de Onda Corretiva: Zigzag



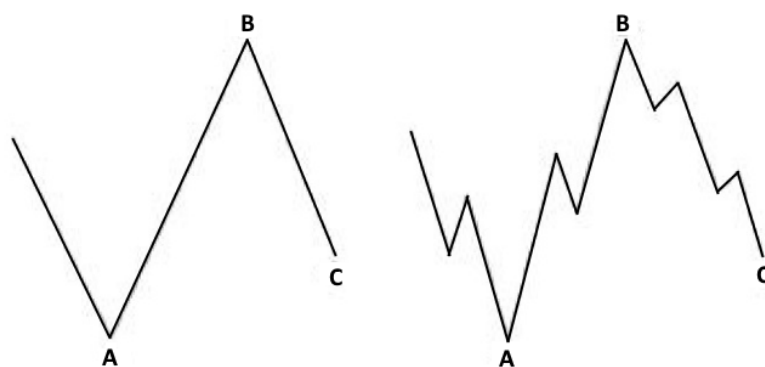
**Figura 25. Padrão de Onda Corretiva: Zigzag Duplo**



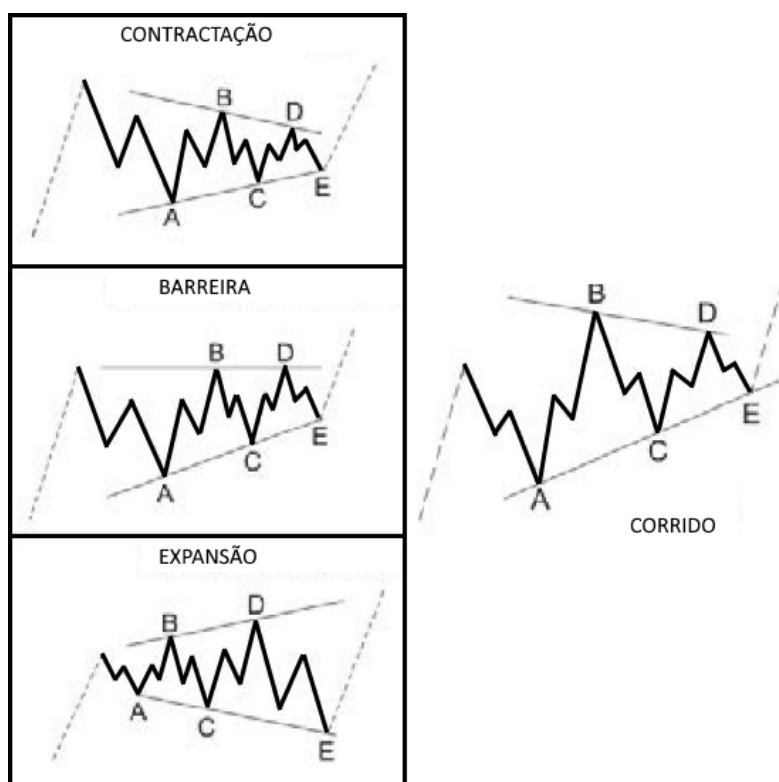
**Figura 26. Padrão de Onda Corretiva: Plano**



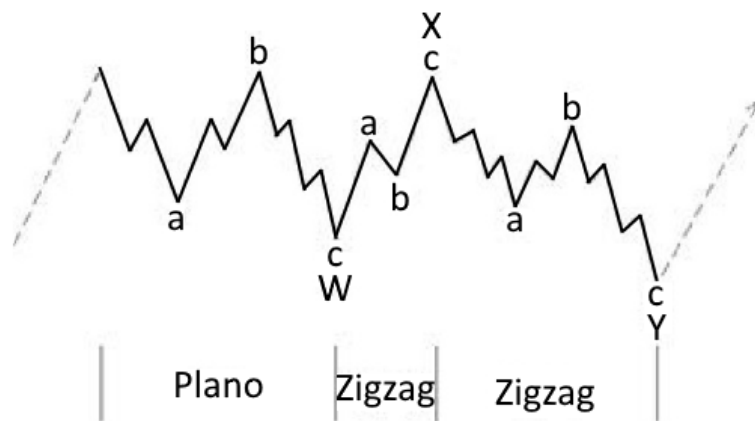
**Figura 27. Padrão de Onda Corretiva: Plano Estendido**



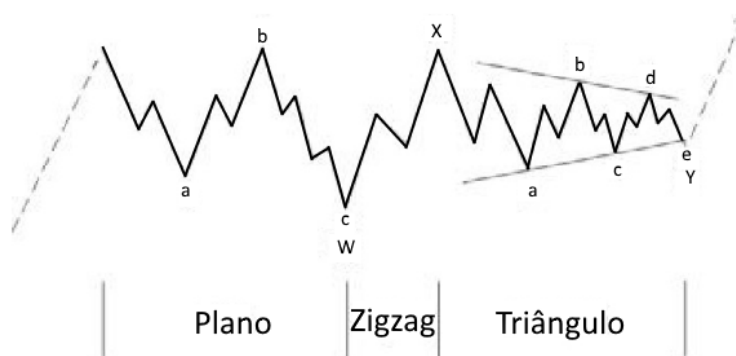
**Figura 28. Padrão de Onda Corretiva: Plano Corrido**



**Figura 29. Padrão de Onda Corretiva: Triângulo**



**Figura 30. Padrão de Onda Corretiva: Combinação (exemplo 1)**



**Figura 31. Padrão de Onda Corretiva: Combinação (exemplo 2)**