

---

# 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**

## **1. Resumo**

Lorem Ipsum

## **Lista de Figuras**

1	Diagrama de Casos de Uso . . . . .	11
---	------------------------------------	----

## **Lista de Tabelas**

1	Representação Quantitativa dos Interesses dos <i>Stakeholders</i> . . . . .	10
2	UC-1.1 - Análise de um Histórico uma Ação . . . . .	12
3	UC-1.2 - Mudança na Configuração de Análise . . . . .	13
4	UC-3.1 - Exibição do Resultado da Análise . . . . .	14
5	Requerimentos Funcionais . . . . .	14
6	Decisão - Linguagem de Implementação . . . . .	16
7	Cronograma de atividades . . . . .	17
8	Progresso do algoritmo . . . . .	18

## Sumário

<b>1</b>	<b>Resumo</b>	<b>4</b>
<b>2</b>	<b>Introdução</b>	<b>7</b>
2.1	Contextualização: Mercado Financeiro e Negociação de Ações . . . . .	7
2.1.1	Gráficos OHLC ( <i>Open-High-Low-Close</i> ) . . . . .	7
2.2	Motivação: Ondas de Elliott . . . . .	7
<b>3</b>	<b>Extensões de Neely</b>	<b>7</b>
3.1	Limitação de Janela de Análise . . . . .	7
<b>4</b>	<b>Fatores de Estudo</b>	<b>7</b>
<b>5</b>	<b>Arquitetura</b>	<b>9</b>
5.1	Contexto do Sistema . . . . .	9
5.2	Requerimentos . . . . .	9
5.2.1	Visão arquitetural . . . . .	9
5.2.2	<i>Stakeholders</i> . . . . .	9
5.2.3	Atributos de Qualidade . . . . .	10
5.2.4	Requerimentos de Alto-Nível . . . . .	11
5.2.5	Casos de Uso . . . . .	11
5.2.6	Requerimentos Funcionais . . . . .	14
5.3	Análise . . . . .	15
5.3.1	Suposições . . . . .	15
5.3.2	Roteiro tecnológico . . . . .	15
5.3.3	Decisões de Projeto . . . . .	15
5.4	Arquitetura do sistema? . . . . .	16
<b>6</b>	<b>Cronograma e Progressão</b>	<b>17</b>

## 2. Introdução

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

Falar de trabalhos passados. Falar de Preços de Ações.

#### 2.1.1. Gráficos OHLC (*Open-High-Low-Close*)

A visualização do preço de uma **ação** normalmente é feita durante quatro momentos de um período: preço de abertura . os preços máximo, mínimo, de fechamento e de abertura dentro de um período de tempo, o período de tempo onde estes preços foram observados, e o volume de ações negociadas durante o período para uma determinada ação durante um determinado período de tempo. Estes dados estão contidos em um arquivo .CSV (*comma separated values*). O programa Metatrader 5, assim como outros sistemas de análise de preços, é capaz de exportar os preços observados de uma dada ação organizados no formato .CSV de forma compatível ao uso neste sistema.

### 2.2. Motivação: Ondas de Elliott

Citar limitações de repetibilidade de Elliott e subjetividade.

## 3. Extensões de Neely

Lorem Ipsum Neely...

### 3.1. Limitação de Janela de Análise

Principal limitação das Extensões de Neely: número fixo de pontos na janela de análise baseadas em uma folha de papel.

## 4. 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 (manter atual, não usar, testar nova)
  - 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.



## 5. Arquitetura

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

### 5.1. 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.

### 5.2. Requerimentos

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

#### 5.2.1. 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 2.1.1. 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 3 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 2.2. Caso a conclusão do algoritmo seja antecipada ao cronograma, será implementada a emissão de um relatório escrito.

#### 5.2.2. 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 1 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 1. Representação Quantitativa dos Interesses dos Stakeholders**

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

### 5.2.3. Atributos de Qualidade

Os atributos chaves de qualidade são aqueles que concentram o maior interesse dos *stakeholders* (Tabela 1).

**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 tornam 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.

#### 5.2.4. Requerimentos de Alto-Nível

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

##### Análise de Dados

HL-1 Deve O sistema deve fazer uma análise das Ondas de Elliott de acordo com as Extensões de Neely de forma confiável.

##### Configuração da Análise

HL-2 Deve O sistema deve permitir a mudança da configuração da análise a ser feita sobre os dados.

##### Exibição de Resultados

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

#### 5.2.5. Casos de Uso

O diagrama de casos de uso é ilustrado na Figura 1, demonstrando os mais importantes casos de uso e seus atores.

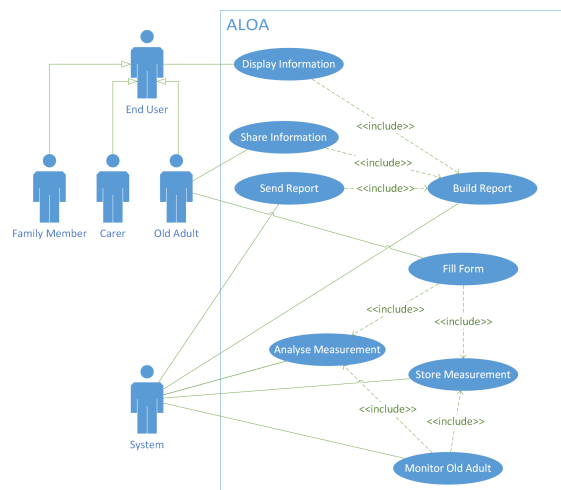


Figura 1. Diagrama de Casos de Uso

Além disso, as Tabelas a seguir mostram os casos de uso arquiteturalmente significantes. A numeração refere-se ao requerimento de Alto-nível, por exemplo, UC-1.x refere-se a HL-1.

**Tabela 2. UC-1.1 - Análise de um Histórico uma Ação**

### **UC-1.1 - Análise de um Histórico uma Ação**

<b>Ator Primário</b>	Usuário Final
<b>Objetivo</b>	Analisar o Histórico de uma Ação
<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</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>Requerimentos Relacionados</b>	HL-1, FR-1, FR-2

**Tabela 3. UC-1.2 - Mudança na Configuração de Análise**

**UC-1.2 - Mudança na Configuração de Análise**

<b>Ator Primário</b>	Usuário Final
<b>Objetivo</b>	Alterar as configurações de análise
<b>Condições Iniciais</b>	Um arquivo válido foi analisado pelo sistema (UC-1.1)
<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</li></ol>	
<b>Extensões</b>	
<b>Sub-variações</b>	
<b>Condições Finais</b>	A análise é atualizada para a nova configuração.
<b>Requerimentos Relacionados</b>	HL-1, HL-2, FR-3

**Tabela 4. UC-3.1 - Exibição do Resultado da Análise**

### **UC-3.1 - Exibição do Resultado da Análise**

<b>Ator Primário</b>	Sistema
<b>Objetivo</b>	Exibir o resultado da análise feita
<b>Condições Iniciais</b>	Um arquivo válido foi analisado pelo sistema (UC-1.1) ou o usuário alterou as configurações de 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>Requerimentos Relacionados</b>	HL-3, FR-4

#### **5.2.6. Requerimentos Funcionais**

Nesta seção, os requerimentos que capturam o comportamento esperado do sistema serão apresentados (Tabela 5). Este comportamento pode ser expresso como serviços, tarefas ou funções que o sistema deve executar. Estes são derivados a partir dos requerimentos de Alto-nível e dos casos de uso.

**Tabela 5. Requerimentos Funcionais**

FR-1	Deve	O sistema deve validar se a entrada é válida.
FR-2	Deve	O sistema deve informar o usuário caso a entrada seja inválida.
FR-3	Deve	O sistema deve atualizar automaticamente a análise quando o usuário alterar as configurações.
FR-4	Deve	O sistema deve realizar a análise automaticamente, e exibir o resultado ao usuário automaticamente.

### **5.3. 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.

#### **5.3.1. Suposições**

- O usuário final tem conhecimento sobre o mercado de ações (Seção 2.1);
- O usuário final tem conhecimento sobre a representação dos preços de ações (Seção 2.1.1);
- O usuário final tem conhecimento sobre as Ondas de Elliott (Seção 2.2);
- O usuário final tem a sua disposição um software capaz de exportar os preços da ação que quer analisar em formato .CSV;

#### **5.3.2. 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.

#### **5.3.3. Decisões de Projeto**

As decisões de alto-nível tomadas durante o projeto podem ser vistas nas Tabelas seguintes.

**Tabela 6. 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).</p> <p>Node.js é um <i>runtime</i> para JavaScript que implementa uma modificação da engine V8, permitindo a execução de código em JavaScript sem a necessidade de um navegador web.</p> <p>NPM é um catálogo de bibliotecas e projetos em JavaScript, reunindo mais de 60 mil módulos, com bilhões de downloads semanais e alto controle de versão para cada módulo.</p> <p>Electron 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, com alto controle de versão.</p>

#### 5.4. Arquitetura do sistema?

Lorem Ipsum arquitetura



## 6. Cronograma e Progressão

Nesta seção são listadas as atividades previstas para realização do trabalho proposto. Na Tabela 7 é apresentado o cronograma para estas atividades. Na Tabela 8 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. Apresentação da qualificação																								
5. Elaboração do programa																								
6. Testes																								
7. Análise de configurações																								
8. Escrita da dissertação																								
9. Defesa do mestrado																								

**Tabela 7. 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 8. Progresso do algoritmo**

## **Referências**

Electron. <https://electronjs.org/>. Acessado em 2019-07-15.

Engine Chrome V8. <https://v8.dev/>. Acessado em 2019-07-15.

Node Pack Manager, NPM. <https://www.npmjs.com/>. Acessado em 2019-07-15.

Node.js. <https://nodejs.org/>. Acessado em 2019-07-15.