

PUC

ISSN 0103-9741

Monografia em Ciência da Computação n° 04/2022

Uma Metodologia para Desenvolvimento de Sistemas Inteligentes de Apoio a Decisão

Guilherme Nascimento Pate Santos

Carlos José Pereira de Lucena

Departamento de Informática

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

RUA MARQUÊS DE SÃO VICENTE, 225 - CEP 22453-900

RIO DE JANEIRO - BRASIL

Uma Metodologia para Desenvolvimento de Sistemas Inteligentes de Apoio a Decisão

Guilherme Nascimento Pate Santos¹, Carlos José Pereira de Lucena¹

¹Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio)
{gnascimento, lucena}@inf.puc-rio.br

Abstract: When importing products, the process can be performed in different ways and with different actors. In our main scenario, the actors are the importing, exporting and trading companies, which will intermediate the negotiation. The scenarios have in common the international contract that correlates different currencies; the exchange adds the risk of this variation to the operation. Thus, it is common to carry out a procedure known as currency hedge to mitigate this risk. This scenario motivated the development of the methodology aiming to speed up the process of building an intelligent decision support system. In this sense, a challenge in creating the methodology was to build methods that could unite, for example, concepts of agility and architecture in order to bring greater synergy and fluidity to development. The methodology uses meetings of a maximum of 15 minutes and small packages, to support the agile concept, despite this, the agility resulting from this methodology is not only guaranteed by these factors reused from agile methods, but also by artifacts made available to its users. The four artifacts available are the 3-layer model, the conceptual framework for negotiation, the agile guide for developing the methodology, and the agentrank classification model based on pagerank, which can be reused. In this way, when using these artifacts, users will come to perceive them as factors that drive the development and reasoning of how an intelligent decision support system should work. The methodology brings advantages in terms of speed and suitability of the architecture to the objectives of the intelligent decision support system. In this sense, it was perfectly adherent to the negotiation ecosystem concept and to scenarios with exchange of messages between the parties, which corroborates the use of software agents. To ensure adequacy, the methodology provides a control phase and, in addition, provides and advocates continuous control, so that, while the system is in use, it can be adapted. In the control phase, it is expected, an iterative execution will take place in order to ensure that intersections between social and technical factors, such as trends and technical analysis and/or rankings, so that together they can present increasingly efficient results. The main contribution of the article is the initial version of a methodology for the development of intelligent decision support systems, called MoIDSS. The methodology shows that multi-agent systems can represent the uncertainties of the real world and, based on this information; recommend the execution of the contract with the company with the lowest risk factor. Another contribution is the modification of the pagerank algorithm to the agentrank model, allowing the algorithm to run in negotiation cycles, thus performing a classification of agents and without generating confusion about the roles of each agent. Finally, throughout the iterative cycle of negotiation, the methodology advocates the enrichment of data in a continuous way, which supports the decision-making process.

Keywords: Intelligent Methodology for Decision Support, Multi-Agent Decision Support Systems, pagerank, Data Enrichment.

Resumo. Na importação de produtos, o processo pode ser executado de diversas formas e com atores distintos. No cenário em questão os atores são as empresas importadora, exportadora e a *trading*, esta quem intermediará a negociação. Os cenários têm em comum o contrato internacional que correlaciona moedas distintas, o câmbio acrescenta o risco desta variação à operação. Assim, é comum que se realize um procedimento conhecido como hedge cambial para mitigar este risco. Este cenário motivou o desenvolvimento da metodologia visando tornar mais célere o processo de construção de um sistema inteligente de apoio a decisão. Neste sentido, um desafio na criação da metodologia, foi construir métodos que pudessem unir, por exemplo, conceitos de agilidade e arquitetura de modo a trazer uma maior sinergia e fluidez ao desenvolvimento. A metodologia utiliza reuniões de no máximo 15 minutos e pacotes pequenos, para apoiar no conceito ágil, apesar disto, a agilidade proveniente desta metodologia não é apenas garantida por estes fatores reutilizados de métodos ágeis, mas também por artefatos disponibilizados a seus usuários. Os quatro artefatos disponibilizados são, o modelo de 3 camadas, o framework conceitual de negociação, o guia ágil para desenvolvimento da metodologia, e o modelo classificação *agentrank* baseado no *pagerank*, que podem ser reaproveitados. Dessa forma, ao utilizar estes artefatos, os usuários passarão a percebê-los como fatores que impulsionam o desenvolvimento e o raciocínio de como um sistema inteligente de apoio a decisão deve funcionar. A metodologia traz vantagens em termos de velocidade e adequação da arquitetura aos objetivos do sistema inteligente de apoio a decisão. Neste sentido, se mostrou perfeitamente aderente ao conceito ecossistema de negociação e a cenários com troca de mensagens entre as partes, o que corrobora com o uso de agentes de software. Para garantir a adequação, a metodologia disponibiliza uma fase de controle e além disso, disponibiliza e preconiza o controle contínuo, para que, enquanto o sistema estiver em uso ele possa ser adaptado. Espera-se que durante a fase de controle, ocorra uma execução iterativa de modo a garantir que cruzamentos entre fatores sociais e técnicos, como tendências e análises técnicas e/ou ranqueamentos, para que juntos possam apresentar resultados cada vez mais eficientes. A maior contribuição do artigo é a versão inicial de uma metodologia para desenvolvimento de sistemas inteligentes de apoio a decisão, apelidada de MoIDSS. A metodologia mostra que sistemas multi-agentes podem representar as incertezas do mundo real e com base nestas informações recomendar a realização do contrato com a empresa de menor fator de risco. Outra contribuição, é a modificação no algoritmo *pagerank* para o modelo *agentrank*, possibilitando que o algoritmo execute em ciclos de negociação, realizando assim uma classificação dos agentes e sem gerar confusão sobre os papéis de cada agente. Por fim, durante todo o ciclo iterativo da negociação a metodologia preconiza o enriquecimento dos dados de forma contínua o que apoia o processo de tomada de decisão.

Palavras-chave: Metodologia inteligente para suporte a decisão, Sistemas Multi-Agentes de apoio a Decisão, *pagerank*, Enriquecimento de Dados.

Responsável por publicações:

Rosane Teles Lins Castilho
Assessoria de Biblioteca, Documentação e Informação
PUC-Rio Departamento de Informática
Rua Marquês de São Vicente, 225 - Gávea
22453-900 Rio de Janeiro RJ Brasil
Tel. +55 21 3114-1516 Fax: +55 21 3114-1530
E-mail: bib-di@inf.puc-rio.br
Web site: <http://bib-di.inf.puc-rio.br/techreports/>

Sumário

1	Introdução	6
2	Metodologia para o desenvolvimento de sistemas inteligentes de apoio a tomada de decisão (MoIDSS)	9
2.1	Fases da metodologia	11
2.2	Artefatos da metodologia	13
2.2.1	Modelo de 3 camadas	13
2.2.2	Framework conceitual para o ecossistema de negociação	15
2.2.3	Guia ágil para desenvolvimento com a metodologia	16
2.2.4	Modelo do algoritmo <i>agentrank</i>	17
2.3	O ciclo da metodologia (MoIDSS) e o gerenciamento de projetos	19
2.4	Pontos relevantes do Metodologia (MoIDSS)	20
3	Aplicação (MASoIC) e a Visão Geral Contrato Internacional	21
4	Trabalhos Relacionados	25
5	Processos Utilizados na Pesquisa	27
6	Agentes do Sistema Inteligente de Apoio a Decisão (MASoIC)	30
6.1	Por que Agentes?	33
7	Discussão e Conclusão	34
8	Trabalhos Futuros	36
9	Referências	36

1 Introdução

Sem sombra de dúvidas a área de importação tem diversas facetas de atuação, e dependendo por onde seguimos podemos modificar, tanto o processo, quanto atores que realizam a importação e por isso representa um grande nicho para aplicações computacionais. Dessa forma, quando pensamos em contratos de importação, vemos que existem três grandes abordagens, já pré-estabelecidas e amplamente utilizados mundo a fora para este processo de negócio. Estes três fluxos são classificados como, importação por conta e ordem de terceiro, importação por encomenda e importação por conta própria. Para qualquer dos cenários a definição de importação é única, sendo assim temos que "a importação nada mais é que a entrada de produtos vindos de outros países, porém, perante a legislação brasileira, uma importação só se concretiza quando é realizado o desembaraço aduaneiro" [37]. A importância da importação se dá, por exemplo, na sua capacidade de diversificação seja de mercadoria seja de fornecedores, além disso ela pode servir como uma ferramenta de estímulo a concorrência entre produtos nacionais e importados, o que possibilita também um combate a inflação uma vez que a concorrência costuma fazer com que os preços baixem.

A importação por encomenda diferentemente da importação por conta e ordem, utiliza a *trading company* para arcar com os custos da operação e promover a venda da mercadoria já nacionalizada a um encomendante determinado. Sua definição legal consta da seguinte forma: "Art. 3º Considera-se operação de importação por encomenda aquela em que a pessoa jurídica importadora é contratada para promover, em seu nome e com recursos próprios, o despacho aduaneiro de importação de mercadoria estrangeira por ela adquirida no exterior para revenda a encomendante predeterminado." [23]. Além disso, em seu artigo 1º a lei também dispõe quem seria a pessoa encomendante: "Considera-se encomendante predeterminado a pessoa jurídica que contrata o importador por encomenda".

A importação direta ou importação por conta própria é aquela em que o adquirente da mercadoria é o responsável direto pela importação. Nesta hipótese, o adquirente busca o produto no exterior, realiza a negociação e a operação mercantil sem intermediação e efetua, em seu nome e com recursos próprios, o despacho aduaneiro. Nesta hipótese, o adquirente coincide com o importador [38].

A proposta em tela aborda a importação de produtos e a realização de um contrato de importação, para o fluxo de importação por conta e ordem de terceiro [24]. Sua definição legal consta da Instrução Normativa RFB N° 1861 em seu artigo 2º e define que: "Considera-se operação de importação por conta e ordem de terceiro aquela em que a pessoa jurídica importadora é contratada para promover, em seu nome, o despacho aduaneiro de importação de mercadoria estrangeira, adquirida no exterior por outra pessoa jurídica" [23]. De modo que, a *Trading Company* é uma importadora, pessoa jurídica, que realiza o processo de importação em nome próprio e depois repassa para o contratante. Ou seja, o desembaraço aduaneiro e demais exigências serão realizadas pela própria *Trading Company*. A figura 1 mostra um esquema gráfico para facilitar o entendimento e a visualização da definição e do funcionamento desse modelo de importação.

Este modelo de negócio é bastante utilizado por empresas que trabalham com a importação de produtos, pois oferece uma vantagem, que é deixar a logística a cargo de outra empresa, assim quando comparado com os demais fluxos, que é a possibilidade da empresa importadora focar na sua atividade fim, ou seja, negociar e realizar a operação de importação. Após a negociação uma terceira empresa chamada de trading

realiza os processos de compra, transporte e despacho aduaneiro das mercadorias desejada, entregando os produtos já nacionalizados ao cliente no final do processo [23]. Este fluxo de trabalho traz uma vantagem para ambas as partes que é a segregação por área de especialização das empresas, ou seja, uma irá delegar para outra aquilo que não é o seu foco, assim, permitindo que cada uma atue naquilo que é melhor. A segregação de função é um conceito que surgiu com Frederick Winslow Taylor (1856-1915) considerado o pai da Administração Científica e também foi muito difundido por Ford. Nesta época o conceito buscava a especialização extrema como forma de agilizar e maximizar os ganhos das empresas, porém as perdas com stress devido a repetição motivou uma flexibilização deste modelo fazendo com que nos dias de hoje o objetivo de maximizar os lucros sejam sustentáveis no que diz respeito a equipe [39][40].

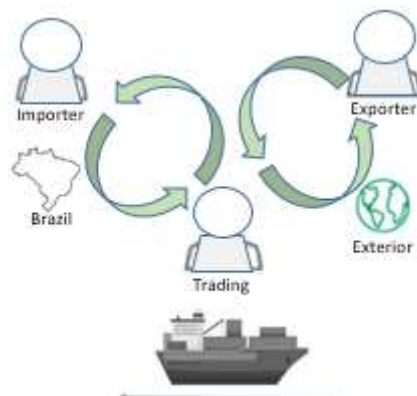


Figura 1. Fluxo de Importação por Conta e Ordem de Terceiro

Quando pensamos em aplicações e paradigmas computacionais percebemos que a abordagem de sistemas multi-agentes traz inerentemente ao seu conceito, a ideia de segregação de funções entre os agentes, dividindo o problema inicial em escopos menores a fim de alcançar um campo bem definido que facilita a compreensão e o desenvolvimento, de modo que cada agente realize aquela tarefa que melhor executa.

Uma definição comum para sistema multi-agente é de que trata-se de um sistema computacional onde pelo menos dois, mas geralmente mais, agentes trabalham em conjunto, cada um executando suas atividades, com o objetivo de concluir determinada tarefa global, ou um conjunto de objetivos globais [18]. Sendo assim, podemos observar que os sistemas multi-agentes auxiliam e permitem a criação de pequenas e grandes sociedades de agentes semiautônomos, capazes de interagir convenientemente através de troca de mensagens (que é uma das características do agente) [9][14]. Assim com bases (em características e propriedades) nas características (de) dos agentes de software (como autonomia, mobilidade, pró-atividade e o seu comportamento de troca de mensagens que permitiu a construção de uma rede *peer-to-peer*) entendemos que este paradigma pode apoiar de forma bastante realista e flexível a construção de um ecossistema de negociação para representar um cenário real de negociação em um fluxo de importação por conta e ordem [10][12][13].

Os frameworks disponíveis para viabilizar o paradigma de agentes também nos permite construir uma aplicação não monolítica, através da criação de agentes dentro de um container, o que permite uma boa escalabilidade, e traz assim uma noção de um ecossistema de negociação, permitindo realizar todas as simulações necessárias, mesmo sem uma grande infraestrutura.

Desta forma, o trabalho aqui apresentado traz algumas propostas que visam contribuir no sentido de agilizar e guiar o desenvolvimento de sistemas inteligentes de apoio a decisão. A primeira contribuição é a metodologia apelidada de MoIDSS que fornece

aos usuários, fases bem estruturadas e guiadas por perguntas que visam construir o conhecimento e a arquitetura do sistema que se pretende desenvolver. Para apoiar este processo de construção a metodologia disponibiliza uma arquitetura conceitual de 3 camadas, que tem como objetivo corroborar com a lógica para a construção de uma arquitetura geral do sistema. Outro ponto que podemos citar como diferencial da metodologia é que ela traz um conjunto de frameworks conceituais que guiam o usuário em um raciocínio de construção de um sistema inteligente de apoio a decisão. O framework de negociação entre agentes, entrega ao usuário uma estrutura para construção do ecossistema de negociação, já o modelo conceitual da arquitetura de 3 camadas proporciona ao usuário uma lógica de comunicação entre componentes do sistema através das camadas de negociação, dados e predição ou reputação, por fim, o modelo do algoritmo *agentrank* disponibiliza uma estrutura para classificar um agente que está imerso em um ciclo de negociação para a tomada de decisão. Além disso, a metodologia traz uma estrutura de uso baseada em um ciclo iterativo e interativo entre as fases do ciclo de vida o que leva a uma evolução natural e ágil de um sistema inteligente de apoio a decisão, os diferenciais da metodologia serão abordados com maior detalhe nas demais seções.

A metodologia traz também um framework conceitual para um ecossistema de negociação, que poderá ser totalmente utilizado ou poderá servir como base para estruturar a simulação, da negociação, das estratégias utilizadas para análise técnica como também análise de tendências. Ao estruturar esse ecossistema de negociação a metodologia sugere uma reflexão sobre como trazer para o sistema e sua arquitetura com mais consciência e aprendizagem iterativa. Com aprendizagem iterativa queremos seguir o conceito utilizado nos controladores por aprendizagem iterativa da engenharia elétrica, que tem como principal característica, para utilização na nossa metodologia, a utilização de informações prévias da resposta e do sinal de controle para o cálculo da ação de controle atual de modo a atingir o objetivo desejado: obter uma resposta desejada com grande precisão. Para isso as reuniões da fase de negociação da metodologia, devem tratar de questões como o enriquecimento de metadados, algoritmos de classificação e algoritmos de predição; e como estes algoritmos podem trabalhar de forma integrada com as estratégias (técnica e de tendência) utilizada na negociação de forma a melhorar o suporte a decisão que a feito pelo sistema. Neste sentido a metodologia propõem um algoritmo apelidado de *agentrank* que é uma adaptação do algoritmo de classificação *pagerank* [17], o algoritmo proposto reconhece ciclos de negociação entre os agentes e faz a classificação deles. A metodologia utilizou como referência métodos como scrum [33] e kanban [34] para apoiar na visão ágil.

Resumidamente podemos dizer que o Scrum é uma metodologia que reuni boas práticas utilizadas no gerenciamento de projetos complexos, onde geralmente não se conhece antecipadamente todas as etapas ou necessidades. Tem em seu ciclo de tratamento reuniões curtas e jogos, focando nos membros da equipe, para torna os processos mais simples e claros, pois assim, mantém registros visíveis sobre o andamento de todas as etapas.

Por sua vez o Kanban traz como principal foco o conceito de sinalizar as atividades através de cartões (*post-its*) posicionados em um quadro que é dividido por etapas, de modo indicar e acompanhar o andamento da produção dentro da indústria, tendo como objetivo complementar controlar o fluxo de produção para que ele seja sustentável no decorrer do tempo. Ou seja, o kanban é um sistema visual que busca gerenciar o trabalho conforme ele se move pelo processo.

Para ter uma experiência de uso da metodologia, desenvolvemos um sistema inteligente de apoio a decisão, apelidado de MASoIC que tem o objetivo de realizar o apoio a decisão em uma negociação de um contrato internacional de importação de produto

combustível, com isso foi possível avaliar como é realizar uma implementação de um sistema com a metodologia criada, pontuando suas vantagens de uso.

2 Metodologia para o desenvolvimento de sistemas inteligentes de apoio a tomada de decisão (MoIDSS)

A metodologia proposta foi idealizada para guiar o desenvolvimento de sistemas inteligentes de apoio a tomada de decisão, e a motivação principal foi a ausência na literatura de uma proposta unindo conceitos e paradigmas de uma forma ágil e integrada um ciclo de vida de gerencia de projetos. Nesta seção vamos abordar a metodologia, suas estruturas e conceitos relevantes para o entendimento e uso.

Em uma definição mais formal, a metodologia preconiza que sejam definidos no início do projeto, o processo, o objetivo, a meta, para que durante o seu uso, sejam percebidas as evoluções com as entregas atingindo assim uma mudança de estado que nos leva a um sistema inteligente de apoio a decisão.

A metodologia traz um protagonismo para o papel do Líder da equipe de TI, assim, há uma previsão de que o Líder seja o responsável por garantir o uso correto da metodologia, garantindo o ciclo correto das reuniões e a interação entre os atores. Dessa forma, o desenvolvimento deve seguir de forma ágil por todas as fases e iterações e por meio do guia, apoiar ou influenciar na escolha do processo a ser desenvolvido durante o uso da metodologia. O líder deve garantir também a apresentação dos pacotes desenvolvidos ao patrocinador, para que este dê sua anuência constando assim a aceitação do ponto de vista do usuário e da empresa.

Um ponto importante e que deve ocorrer no início do uso da metodologia, é a reunião de alinhamento, tratasse de uma reunião estritamente técnica de TI onde é realizado o alinhamento com toda a equipe incluindo o Líder, para que possam realizar um *brainstorm* e debater sobre paradigmas e tecnologias, que serão candidatas ao uso no desenvolvimento. A realização desta reunião torna ainda mais ágil todo o processo de construção da solução e por isso representa um ganho, no uso da metodologia e sempre deve ser utilizada como seu ponto de partida.

Na fase de iniciação o patrocinador deve trazer os processos que estão relacionados com o objetivo de apoio a decisão, nesta sequência o Líder de TI deve realizar os questionamentos de modo a convergir para os fluxos mais maduros e informatizados do processo, visando com isso acelerar o processo de construção do sistema. Dessa forma, na escolha do processo devemos perceber quais os processos e os objetivos estão intimamente relacionados e por fim as metas também devem estar bem definidas, pois a partir dela que definimos as tarefas que devem ser alcançadas para realização do objetivo. Assim, podemos entender que mudanças de estado são consequências direta da execução de processos e a realização das metas, sendo viabilizado através da execução de eventos gerados pelo processo escolhido.

Ainda na fase de iniciação, a metodologia disponibiliza um ciclo iterativo, representado por uma seta pontilhada, caso exista a necessidade de um refinamento daquilo que foi apresentado pelo patrocinador.

Na fase de análise e planejamento, a metodologia pretende que seja realizada toda as confirmações necessárias para garantir que os requisitos levantados estão com uma boa qualidade e a equipe pode seguir neste caminho sem muitos ajustes posteriormente.

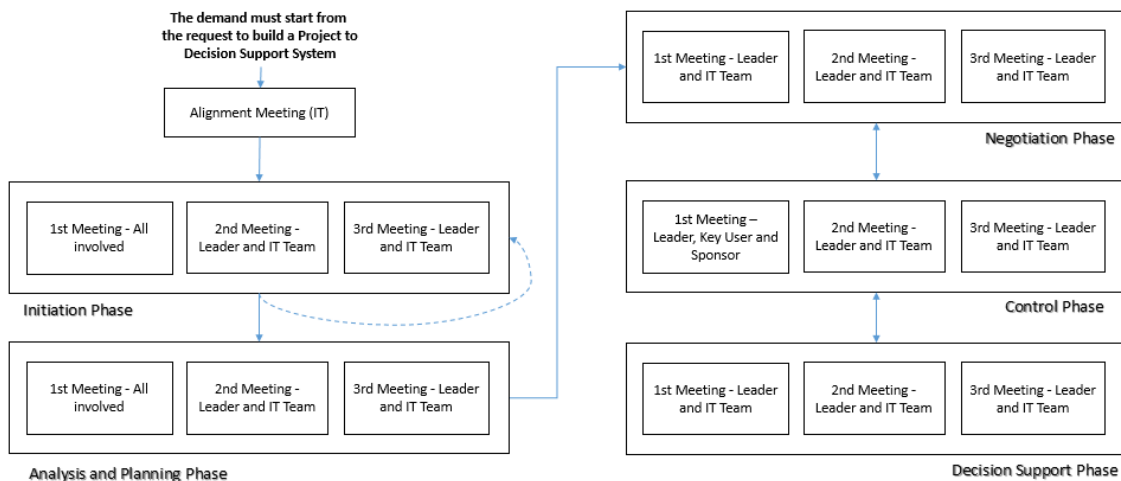


Figura 2. Metodologia MoIDSS, fluxo de fases e reuniões

Na fase de negociação temos a necessidade de definição de qual paradigma utilizar para que a negociação fique bem aderente ao que ocorre no mundo real. Além disso, boa parte da regra de negócio deve ser discutidas nas reuniões desta fase, para que as estratégias de análise técnica bem como a análise de tendências, possam ser implementadas de forma bem aderente ao negócio.

Na fase de negociação temos como centro da discussão à definição de qual paradigma será utilizado para a construção do ecossistema de negociação para que fique bem aderente ao que ocorre no mundo real. Além disso, boa parte da regra de negócio deve ser discutidas nas reuniões desta fase, para que as estratégias de análise técnica bem como a análise de tendências, possam ser implementadas de forma bem aderente ao negócio.

Na fase de controle a metodologia tem como principal objetivo realizar os ajuste necessários para a adequação a regra de negócio que foi levantada nas demais fases, bem como no nível de sistema discutir como podem ser desenvolvidas as adaptações, que possam surgir através de etapas como enriquecimento dos dados, previsões e classificações.

Na fase de suporte a decisão é onde deve ser feita a conclusão, ou seja, o sistema deve realizar a análises final com base nos dados obtidos a partir das análises realizadas nas demais etapas, e deve realizar o contato com o usuário de modo a informar claramente qual decisão o sistema entende ser a melhor.

Na fase de suporte a decisão é onde deve ser feita a conclusão a partir dos dados e as análises realizadas e deve realizar o contato com o usuário de modo a informar claramente qual a decisão o sistema entende ser a melhor.

O desenho básico na figura 1 mostra o fluxo das fases da metodologia e as reuniões que devem ser realizadas, além de, apresentar em cada caixinha os participantes de cada reunião. Deste fluxo podemos entender que o levantamento do requisito irá se concentrar na fase de iniciação e na fase de análise e planejamento, porém seus ajustes se darão durante a fase de controle, já na fase de negociação e apoio a decisão o foco será na regra de negócio e as estratégias de consciência de software.

Em todo a metodologia há a recomendação em utilizar conceitos que propiciem a construção de um software inteligente e adaptativo, para isso artefatos e métodos são disponibilizados aos usuários.

2.1 Fases da metodologia

A metodologia proposta deve guiar o seu usuário em um fluxo de desenvolvimento de sistemas inteligentes de apoio a decisão, e para o seu uso ganhe uma forma otimizada e interativa entre as fases preconiza-se utiliza-la conforme, apresentado nesta seção.

Conforme descrito nas seções anteriores o Líder de TI deve ser o protagonista e por isso vamos descrever as fases através desta visão.

O ponto de partida ocorre quando a equipe de TI recebe uma demanda, para desenvolvimento de um sistema inteligente de apoio a decisão, ao receber esta demanda o líder deve chamar uma reunião de alinhamento com a equipe técnica, primeiro para consolidar o entendimento da metodologia e seu fluxo de utilização, bem como utilizar esta primeira reunião para estudar sobre os paradigmas a serem utilizados no desenvolvimento e as tecnologias mais adequadas para realizar o desenvolvimento e a implementação.

Na fase de iniciação, há uma primeira reunião entre todos os envolvidos, ou seja, o líder de TI com toda a equipe e os patrocinadores com os usuários chave. Nesta reunião a parte interessada deve apresentar o processo que se quer implementar expondo também seus objetivos e metas. Após esta apresentação o líder, deve trazer, uma reflexão a respeito de qual deve ser o processo mais adequado para se iniciar o desenvolvimento de um sistema inteligente de suporte a decisão. Para isto a metodologia disponibiliza 3 perguntas fixas e sugere que o líder faça estas perguntas, ou que complementemente até um máximo de 5 perguntas de modo que as respostas a estas perguntas ajudem a convergir para um processo de negócio.

Caso a primeira reunião acabe sem convergir para um processo de negócio, o patrocinador terá um dia para consultar se existe um processo que responda as perguntas que foram realizadas, caso contrário será necessário realizar toda uma preparação do processo para que ele possa seguir no trabalho de uso da metodologia. Em resumo nesta etapa irão definir se existe algum processo com uma maturidade mínima de TI ou se o desenvolvimento deverá ser feito utilizando um processo que deverá ser trabalhado desde o início. Na sequência as duas outras reuniões da fase de iniciação são realizadas apenas com a equipe de TI, nesse sentido, o líder deve revisar o que foi discutido e exposto pelo patrocinador. Para estas reuniões o líder deve trazer uma discussão inicial sobre a arquitetura e garantir que não tenha um excesso de informação nem informação de baixa qualidade. Ao final das duas reuniões de TI o líder deve ter fechado uma primeira lógica para a solução e um desenho do protótipo para ser levado para a próxima fase (Análise e Planejamento). Nessa fase sugere-se que a duração das reuniões sejam de 15 minutos para exposição do *stakeholders*, mais 15 minutos para exposição do Líder e 5 para debate e conclusão já as duas outras reuniões com a equipe de TI devem demorar 15 minutos cada.

Na fase de análise e planejamento a primeira reunião segue com uma apresentação do patrocinador que tem 15 minutos para rever o processo, suas necessidades, o objetivo, suas prioridades e expectativa de funcionamento. Após esta apresentação o líder tem 10 minutos para apresentar a linha de raciocínio da solução e o desenho do protótipo que a equipe entendeu ser mais adequado para o sistema inteligente de suporte a decisão. O patrocinador tem mais 5 minutos para dar *feedback* sobre o que o líder de TI apresentou.

As outras duas reuniões desta fase ocorrem com o líder e a equipe de TI e duram 15 minutos. Nesta sequência durante a reunião o líder apresenta para o time, paradigmas, arquiteturas e tecnologias que podem apoiar no processo de construção da solução. O líder deve fomentar esta discussão com o time, que deve delimitar questões técnicas e

fazer a devida documentação. Por fim na última reunião o Líder deve orientar quanto a produção de diagramas e ao quanto ao desenvolvimento em si, de modo a garantir a funcionalidade do produto. Ao final das reuniões o líder deve ter a modelagem conforme sua orientação e o protótipo deve estar desenvolvido com funcionalidades básicas que permitam realizar a sua avaliação quanto as prioridades e os objetivos.

Na fase de negociação, a equipe de TI vai discutir e refletir a respeito de qual deve ser a melhor abordagem e paradigma para o fluxo de negociação dentro do processo escolhido nas fases anteriores, assim temos já definido aquele processo que é mais adequado para dar início ao desenvolvimento, e com este processo vamos desenvolver a simulação de um ambiente de negociação que estará embutido no sistema inteligente de apoio a decisão. Nesta fase, as três reuniões serão realizadas apenas pela equipe de TI e todas com duração de 15 minutos.

Na primeira reunião o líder deve trazer perguntas que viabilizem uma visão geral não apenas sobre o paradigma a ser utilizado, mas também sobre o que existe com relação aos dados, ou seja, no que diz respeito ao banco de dados e outras abordagens como LOG que possam apoiar no desenvolvimento de um ecossistemas de negociação. A primeira reunião deve resultar na ordenação dos paradigmas, sendo sugerido pela metodologia o uso de um paradigma que viabilize a troca de mensagens, como agentes de software, e se além disso verificar se o paradigma permite que a existência dos artefatos referentes a dados possam apoiar neste desenvolvimento do ecossistema. Na segunda reunião o líder deve guiar a reflexão no sentido de entender se as estruturas responsáveis pelos dados podem trazer um melhor resultado para negociação juntas ou se isoladamente, nesse sentido deve ser levado em consideração a necessidade do uso de estruturas como *datalake* e além disso, deve ficar definido nesta reunião qual paradigma será utilizado, sendo o paradigma de agentes uma sugestão da metodologia, por entender que é uma opção adequada. Na terceira reunião a equipe de TI junto com o líder deve realizar o desenho da arquitetura de negociação já no paradigma definido, este desenho deve incluir o *datalake* caso exista ou apenas o banco de dados a depender do resultado da segunda reunião. Nesta última reunião a equipe também deve definir um ciclo de negociação, ou seja, de quanto em quanto tempo o ecossistema de negociação é observado, a metodologia entende que 24 horas é um tempo adequado. Assim com esse ciclo será feito o processo de enriquecimento dos dados da negociação de modo a viabilizar a adaptação do sistema a partir de um *feedback*, ao final de tudo o protótipo deve ter evoluído suas funcionalidade e a modelagem também deve representar toda a evolução decorrida das fases.

A fase de controle, é uma fase transversal a todas as outras da metodologia e é a partir dela que surgem as adequações iniciais e a confirmação de homologação da versão inicial da solução. Desde o início da fase de controle o líder manter contato com o patrocinador de modo a entender o que deve ser ajustado no desenvolvimento da solução para que seja realizada uma entrega inicial. Ele também fica responsável por trazer para o patrocinador a visão de que o sistema inteligente de apoio a decisão só poderá trazer um bom suporte se os requisitos estiverem refletindo bem o processo real, ou seja, se ocorrerem mudanças no processo, essas mudanças devem ser refletidas no código do sistema. Dessa forma, a metodologia preconiza o controle contínuo para garantir que as mudanças no processo sempre estejam refletidas no sistema, o mesmo deve ocorrer quando o sistema estiver na etapa de manutenção. A metodologia sugere que sejam feitas reuniões de no máximo 15 minutos entre o líder e o patrocinador após cada ciclo de reunião (sequência de 3 reuniões) feita principalmente na fase de negociação.

Já na fase de suporte a decisão, o líder deve trazer questionamentos sobre que tipo de análise deve ser levada em consideração para realizar um bom suporte a decisão, ou seja, que sirva realmente como um apoio/suporte a decisão final do usuário. Este questionamento deve ser realizado inicialmente em reunião interna com a equipe de TI como *brainstorm* para na sequência realizar então um ciclo de reuniões (3 reuniões) com o patrocinador para entender e definir quais são as análises que podem apoiar a decisão.

A figura que segue mostra como as fases da metodologia devem ser relacionadas com as etapas levantamento de requisitos e planejamento, design da solução e a realização. O objetivo desta estrutura é apresentar, de forma resumida, ao usuário da metodologia como se dá a iteratividade e a relação com as etapas que devem ocorrer em tempo de reunião.

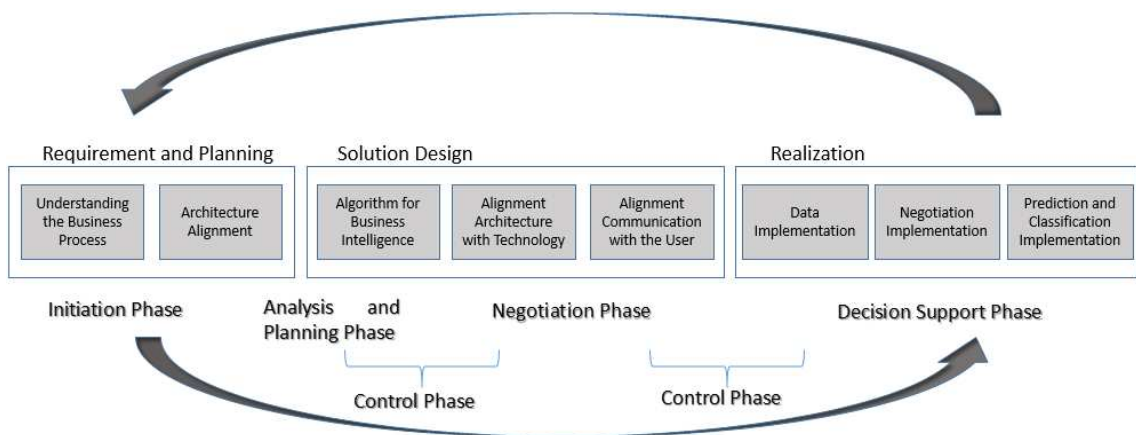


Figura 3. Visão conceitual do ciclo iterativo da metodologia

Por fim podemos concluir que a metodologia apresenta um ciclo iterativo e ágil para o desenvolvimento de sistemas inteligentes de apoio a decisão, esta iteração possibilita que as adequações sejam feitas tanto no momento de projeto quanto em tempo de uso do sistema.

2.2 Artefatos da metodologia

Para utilizar a metodologia MoIDSS de maneira mais assertiva foram construídos quatro artefatos para que o usuário pudesse utilizar com a intenção de impulsionar a construção de um sistema inteligente de apoio a decisão, o que permitirá uma melhor reflexão no que diz respeito ao raciocínio utilizado para análise e escolha da melhor decisão a ser tomada.

Os quatro artefatos disponibilizados são o modelo de 3 camadas para apoiar tanto no desenvolvimento quanto na organização da solução, o framework conceitual para apoiar o desenvolvimento do ecossistema de negociação, o guia ágil que visa tornar mais fluido o uso da metodologia e o modelo do *agentrank* que apresenta um algoritmo que permite realizar a classificação dos agentes envolvidos na negociação.

2.2.1 Modelo de 3 camadas

A metodologia disponibiliza um modelo em 3 camadas que entende-se ser bastante útil para o desenvolvimento, a divisão do modelo se dá em, camada de dados, camada de negociação, camada de classificação e predição.

Dessa forma, a Camada de Dados é a camada onde deve ser implementada toda a estrutura referente a acesso aos dados, seja apenas por banco de dados, seja por “*Data Lake*” e avaliar também o enriquecimento dos dados “que receberá os *inputs* da camada de classificação e predição” para que em seguida possa retro alimentar o sistema.

Na Camada de Negociação teremos o desenvolvimento do ecossistema de negociação com toda a regra necessária para que a interação sistêmica seja o mais próximo do real “, por isso, a metodologia preconiza o uso do paradigma de agentes de *software*”.

Por fim, a Camada de Classificação e Predição é onde serão implementados os algoritmos de análise (técnica ou de tendências) e os algoritmos para predições de dados pertinentes para esta avaliação; além disso teremos a adaptação do algoritmo *pagerank* para uma negociação entre agentes de *software* “conforme sugerido pelo método *agentrank*”.

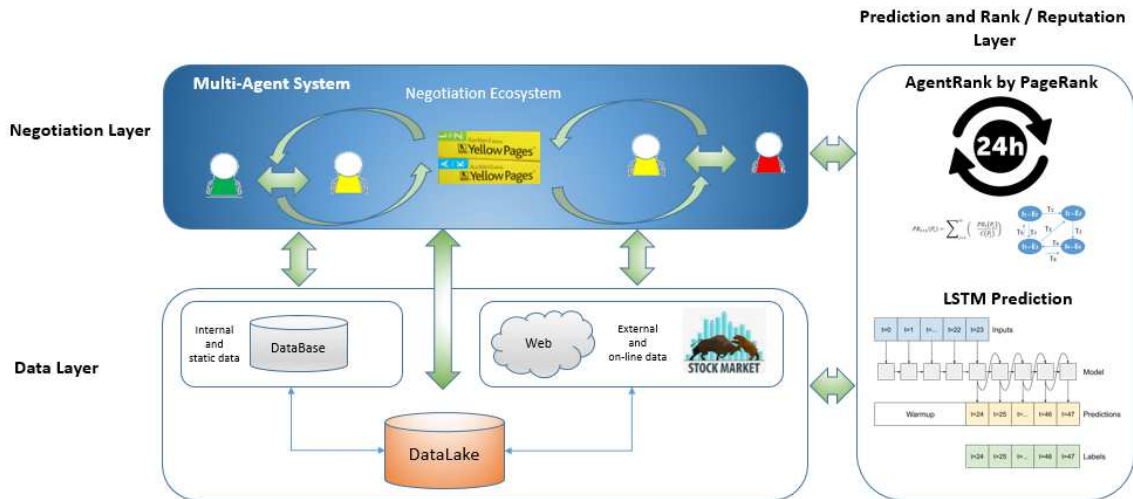


Figura 4. Modelo 3 camadas do MoIDSS, desenho conceitual

O modelo foi construído com três camadas, separadas por especialização, sendo eles, camada de negociação, camada de dados e camada de ranqueamento e predição, esta última sendo transversal as demais.

Na camada de negociação, deve ser construído um ecossistema de simulação seguindo as regras do negócio, com suas metas e objetivos, que será utilizada durante a implementação do modelo. Aqui vemos também a necessidade de definição de qual paradigma utilizar para que a negociação fique bem aderente ao que ocorre no mundo real.

A camada de dados, deve conter dados internos e dados externos que serão utilizados para construir um *datalake* que, além desses dados, vai receber dados vindos da camada de negociação e da camada de ranqueamento e predição. Assim com esta camada podemos manter dados históricos que poderão ser consultados. Na primeira iteração os dados históricos não estarão preenchidos, pois é a partir deste momento que a base histórica passa a ser construída. A partir do momento em que esta base histórica foi iniciada o modelo irá preconizar o enriquecimento dos dados através de um módulo de controle contínuo da informação, conforme a descreve.

Por fim precisamos entender o funcionamento da camada de predição e ranqueamento, nesta camada são feitas análises que vão gerar subsídios para as outras duas camadas de modo a apoiar através de ranqueamento e predição. Por sua vez estes resultados de ranqueamento e predição vão contribuir para o enriquecimento dos dados em todas as fases e camadas do processo.

Em toda a metodologia preconizasse a utilização de conceitos que propiciam a construção de um software inteligente e adaptativo, pensando por exemplo, em uma retroalimentação a partir do meio em que se encontra. Dessa forma, paradigmas como o de agentes de software e o conceito de computação autônômica, poderão agregar muito no uso da metodologia proposta.

2.2.2 Framework conceitual para o ecossistema de negociação

O framework conceitual, traz uma visão geral sobre agentes de software e nas subseções seguintes serão discutidas todas as questões referentes aos agentes de software utilizados no sistema além da descrição dos agentes e a questão do porquê dos agentes serem utilizados no sistema.

Os agentes de software são entidades independentes no sistema, cada qual com o seu objetivo [5]. Cada agente de software possui um comportamento pré-definido, mas podem sofrer influências em função do comportamento dos outros agentes que estão interagindo no sistema. A figura a seguir apresenta o framework conceitual de negociação entre agentes de software, mostrando como a dinâmica de negociação dos agentes pode ser realizada levando em consideração a influência que um agente pode ter sobre o outro dentro de um sistema de suporte a decisão conforme propõem a metodologia [12][13].

O framework apresentado neste trabalho disponibiliza dois sensores para o contato do ecossistema de negociação como mundo real, assim transmitirão aos agentes toda a percepção das mudanças, possibilitando então que o sistema multi-agente possa realizar o suporte a decisão mais adequado.

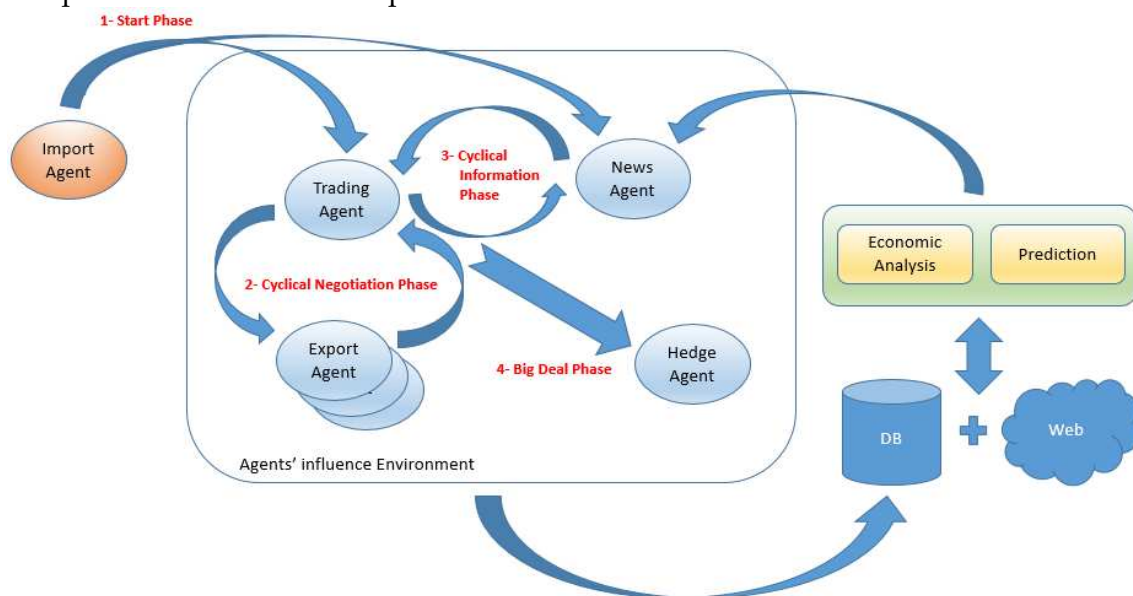


Figura 5. Framework conceitual de negociação

Dessa forma, no nosso cenário exemplo o sistema poderá sugerir a efetivação de um contrato de importação tomando como base as movimentações do mercado financeiro por exemplo ou ainda em conjunto com previsões e enriquecimento de dados.

Para que esses passos sejam realizados os sensores enviarão percepções referente a bolsa de valores com a variação cambial e também referente a evolução de tópicos de tendências de busca, ou seja, enquanto um traz uma visão econômica dos dados o outro traz uma visão social mostrando aquilo que um grupo de pessoas está procurando. Apesar de ambos trazerem para o sistemas fatores humanos, podemos perceber claramente que um tipo de dado é mais técnico que o outro, apesar disto, não se excluem, e são na verdade complementares.

Assim o método para construção de um ecossistema de negociação disponibiliza ao usuário da metodologia o conceito de comunicação entre agentes e a comunicação externa para apoiar na regulação interna visando o objetivo final de suporte a decisão.

2.2.3 Guia ágil para desenvolvimento com a metodologia

A metodologia traz também um método guia que proporciona grandes benefícios aos seus usuários, uma vez que traz predefinições para um uso mais fluido da metodologia, apresentando uma organização pelas fases da metodologia; onde cada fase apresenta um método que dará o tratamento adequado através do guia.

O guia apresenta uma abordagem desde a fase inicial onde a metodologia preconiza a reunião de alinhamento, para dar fluidez a todos os usuários é disponibilizado perguntas que são chamadas de questões essenciais. Neste sentido na reunião de alinhamento a equipe de TI e o líder além de trazerem uma lista de paradigmas e tecnologias, eles também devem ser fazer alguns questionamentos essenciais para a escolha do processo de negócio que será implementado.

Dessa forma, o primeiro questionamento essencial é feito apenas entre a equipe de TI, são perguntas sobre o entendimento do processo de negócio, sobre o gerenciamento do conhecimento e sobre o gerenciamento da arquitetura. Algumas das perguntas podem surgir novamente em outras fases. As perguntas sobre o Entendimento do Processo de Negócio dizem respeito a, Como realizar um suporte a decisão?, Qual algoritmo pode apoiar na inteligência do sistema?, Quais ferramentas utilizar para entregar e otimizar o suporte à decisão? Aqui o método prevê um alinhamento entre a equipe sobre os pontos apresentados.

As perguntas sobre o Gerenciamento do Conhecimento dizem respeito a, Como está o nível de detalhamento do processo de negócio?, Quais os problemas/riscos que podemos mapear?, Como o apoio a decisão pode ajudar na solução deste problema/risco? Aqui o método prevê uma verificação interna se alguém da equipe de TI tem algum conhecimento prévio sobre a área que será tratada neste sistema.

As perguntas essenciais sobre o Gerenciamento da Arquitetura dizem respeito a, Qual arquitetura é mais adequada para o desenvolvimento do sistema inteligente de apoio a decisão? Quais tecnologia serão utilizadas para cada camada da arquitetura? Aqui é o objetivo principal da reunião de alinhamento ajustar o conhecimento entre os membros da equipe visando a agilidade nas demais fases a serem tratadas.

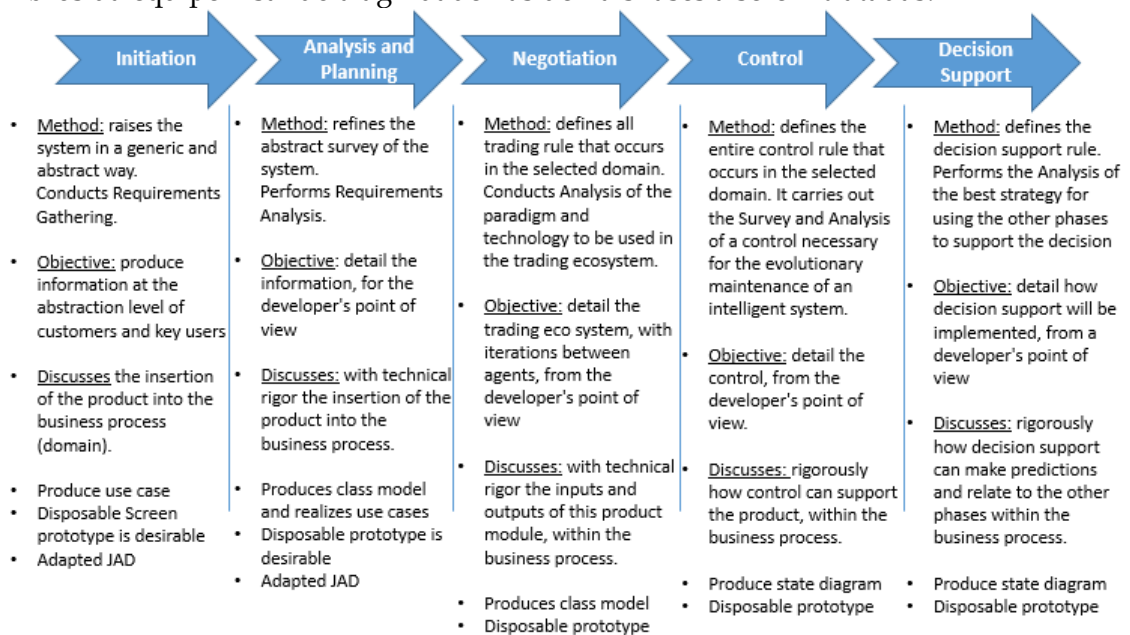


Figura 6. Guia esquematizado por fases

A metodologia define que na execução de suas fases os participantes serão o Time (equipe de TI), o Líder de TI e o Patrocinador (incluindo os usuários do sistema).

Para a fase de iniciação as perguntas essenciais devem ser realizadas pelo líder na reunião com o patrocinador, e tem o objetivo de convergir na escolha do processo de negócio mais preparado (na visão de TI). O líder deve questionar, Qual processo de negócio da empresa está melhor documentado? Qual processo de negócio é o mais conhecido? Qual processo de negócio é o mais informatizado?

Nesta fase o líder deve realizar estas perguntas essenciais que são ditas fixas, porém pode também acrescentar mais duas outras perguntas que apoiem na ideia de convergir o primeiro desenvolvimento focado em um processo que tenha respostas positivas sobre as perguntas essenciais, assim como um “processo de ouro”.

Na fase de negociação o método traz quatro perguntas essenciais, são elas: Existe um banco de problemas e soluções, para o processo escolhido? (o Patrocinador pode ser questionado). Existe arquivo de LOG no seu sistema atual, a arquitetura está prevendo isso? O banco de Dados de problemas e soluções é informatizado? A análise de tendência pode apoiar a análise técnica?

Neste ponto o objetivo é alinhar sobre o acesso aos dados e avaliar a possibilidade e enriquecimento, além da necessidade ou não de uma análise de tendência.

Na fase de controle existem duas perguntas essenciais e o líder deve questionar, como deve ser feita a detecção de uma necessidade de ajuste? Os algoritmos de predição e classificação podem apoiar nesse controle, ou seja, pode influir sobre o momento que o sistema deve ser ajustado?

Na fase de suporte a decisão o método traz apenas uma questão essencial, e preconiza que o líder deve questionar sobre, como deverá ocorrer o contato com o usuário interessado nesta análise realizada pelo sistema?

Com essas perguntas essenciais a metodologia visa guiar a equipe de TI por um caminho mais ágil e assertivo na construção de um sistema inteligente de apoio a decisão, além disso, a metodologia entende e preconiza que a figura 3 deve ser utilizada em conjunto com o guia, pois pode tornar ainda mais preciso o seu uso.

2.2.4 Modelo do algoritmo *agentrank*

O modelo *agentrank* propicia uma classificação dos agentes envolvidos em um ciclo de negociação sem a perda ou confusão de qual era o papel daquele agente em determinado momento, sem o uso deste algoritmo não seria possível realizar uma classificação sem tal confusão entre os papeis. Com uso deste algoritmo que foi modelado o usuário da metodologia pode ganhar muito tempo e manter o foco em outras partes do desenvolvimento.

Em um ecossistema de negociação, tendo como exemplo um cenário de importação ou exportação, devemos considerar que em um determinado lapso de tempo o importador I_1 pode ser também um exportador E_1 . Em alguns domínios esta dualidade é bastante comum, por isso, é importante considerar que quem importa pode também exportar em algum momento. Assim, assumindo que todo importador é também exportador, temos que, a cada ciclo de negociação dado um agente I_1 do conjunto de N agentes importadores (I_n), teremos então um conjunto $N-1$ de exportadores (E_{n-1}) para negociação, ou seja, \forall importador $I_1 \in N$, \exists um exportador $E_1 \in N$. Com esta premissa definida, passamos a utilizar o algoritmo *pagerank* adaptado para determinar, por exemplo, aquele agente exportador com mais utilização.

Em um ecossistema de negociação, percebemos que a variável tempo (T) é essencial para a adaptação do algoritmo *pagerank*, pois o tempo permitirá a realização de adaptações constantes no software através dos *feedbacks* recebidos.



Figura 7. Estrutura inicial de negociação

Dessa forma, a “variável tempo (T)” vai nos apoiar na determinação e construção do grafo entre os agentes importadores e exportadores para isso o algoritmo enviará e receberá *feedback* constante (a cada ciclo de negociação), dessa forma, a cada 24 horas o algoritmo executará o processo de análise para garantir que o grafo atende aos critérios de classificação, evitando assim problemas como ciclo e nó sem ligação. Caso ao final do ciclo alguns dos problemas ainda existam passamos a mais uma execução de 24 horas e caso seja resolvido paramos a execução e realizamos o ranking com base no grafo importador-exportador gerado. O ciclo de negociação é representado pela variável T, assim temos que o primeiro ciclo de negociação é dado por T_1 e assim sucessivamente até T_n onde $n \in \mathbb{N}$. Para esse método de classificação e ranqueamento demos o nome de *agentrank*.

Assim a negociação segue da seguinte forma, a cada ciclo o agente importador escolhe um agente exportador para realizar a negociação, no exemplo da imagem que segue temos um conjunto de 4 importadores (I) e de 4 exportadores (E). Dessa forma com base na premissa, temos que dado um importador a probabilidade de escolha de um exportador será de 75%.

Antes de aplicar o algoritmo vamos apresentar a sua formula, assim como no *pagerank*, ela contém a função de ranking dada por (PR), a iteração atual na negociação dada por (PR_t), a próxima iteração dada por (PR_{t+1}), o fechamento da negociação onde se origina a chamada (ou seja, primeiro negócio fechado ou negócio inicial a partir de um importador) dada por (P_i), e neste negócio inicial há quem recebe a chamada (primeiro negócio fechado ou negócio inicial que é recebido pelo seu destino) dada por (P_j) e por fim o número de links destinos dada por ($C(P_j)$).

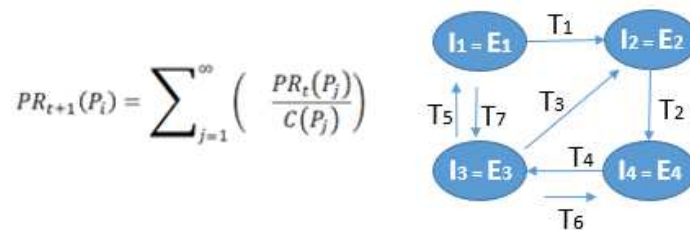


Figura 8. Fórmula *agentrank* e grafo de análise.

Neste caso, após a formação de um grafo, (grafo este que é construído através do método que executa de ciclos de negociação e processo de análise) que permite a execução do *ranking*, passaremos a executar as etapas do algoritmo *pagerank* que consiste nos seguintes passos. Como ponto de partida calculamos a probabilidade de cada elemento (agente) em uma iteração 0, nesta iteração todos tem a mesma probabilidade.

Já na iteração 1 vamos considerar o *start* com o agente I_1E_1 verificando os seguintes pontos: qual agente (importador) aponta para o agente I_1E_1 (exportador)? Neste caso, vemos que o agente I_3E_3 é o único que aponta para o agente I_1E_1 , vemos então que na rodada anterior o *agentrank* de I_3E_3 era de $\frac{1}{4}$. Na sequência teremos que saber para

quantos nós o agente I_3E_3 aponta? Podemos verificar que o agente I_3E_3 aponta para 3 outros, que será o valor de links destino, e assim conseguimos montar o *agentrank* do agente I_1E_1 na iteração 1.

Realizando o procedimento para todos os demais agentes e em cada uma de suas iterações ou ciclos de negociação, chegamos então ao desejado *ranking* dos agentes.

	Iteration 0	Iteration 1	Iteration 2	agentrank by pagerank
I_1E_1	1/4	1/12	1,5/12	
I_2E_2	1/4	2,5/12	2/12	
I_3E_3	1/4	4,5/12	4,5/12	Most relevant
I_4E_4	1/4	4/12	4/12	

Figura 9. Tabela de classificação dos agentes

Assim a cada execução do *agentrank* teremos uma classificação para os agentes criando assim o indicativo de qualidade para aquele agente.

2.3 O ciclo da metodologia (MoIDSS) e o gerenciamento de projetos

Referencias importantes para a metodologia são o guia de gerenciamento de projetos PMBoK e alguns conceitos de métodos ágeis. No processo de criação da metodologia, percebemos que as negociações poderiam ser melhoradas se fossem apoiadas em um processo iterativo que visa propiciar uma adaptabilidade frequente e constante durante todo o ciclo de negociação, que o modelo de 3 camadas disponibilizado pela metodologia limita em 24 horas de negociação.

Em uma visão mais gerencial da metodologia, percebemos que as fases, tem semelhanças com a definição de grupo de processos do PMBoK [36], e se assemelham a um ciclo de vida de um projeto, porém ao implementar a negociação conforme a metodologia determina, percebemos que o ciclo se torna iterativo, assim as características de métodos ágeis podem agregar ainda mais valor, no que diz respeito ao ciclo de vida do desenvolvimento de um sistema de apoio a decisão e suas manutenções.

Em cenários com várias iterações, percebemos a necessidade de ter um modulo de controle continuo onde é possível detectar desde a primeira iteração as perturbações do meio e com isso é possível atuar no sistema de forma a trazer para ele toda a consciência necessária para uma boa adaptação.

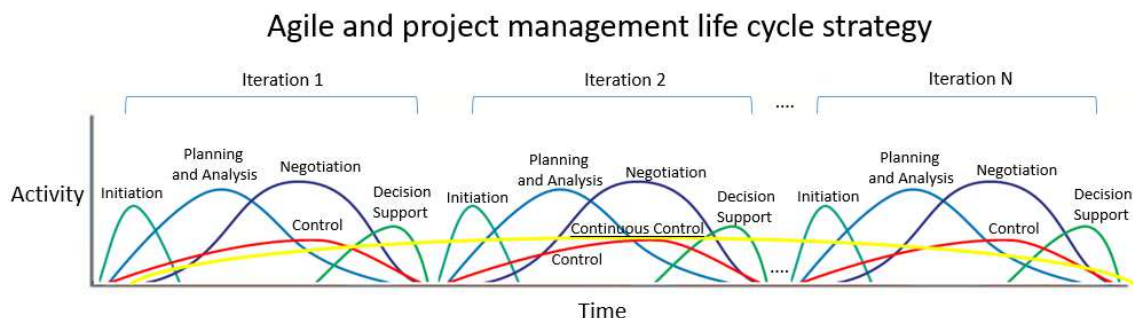


Figura 10. Ciclo de vida estratégico, com controle contínuo, para os desenvolvimentos baseados no MoIDSS

A metodologia para o desenvolvimento de sistemas inteligentes de apoio a tomada de decisão, expõem a sua relação com os processos do ciclo de vida de um projeto, buscando assim a aproximação dos profissionais especialistas em gerencia de projeto. Dessa forma, podemos verificar que, o processo de controle contínuo é um componente essencial para manter a evolução do sistema garantindo uma adaptação contínua, do

ponto de vista gerencial, o profissional poderá utilizar este processo juntamente com os demais da iteração, para poder ter uma estimativa de quantas iterações o projeto deverá ter, além de poder ter uma noção da maturidade do sistema.

Uma das definições essenciais é sobre o ecossistema de negociação onde se pretende realizar uma simulação do processo de negociação, olhando para o ciclo de vida vemos que esta estrutura deve ser tratada em todas as fase, porém bem discutida nas fases de planejamento e análise, na negociação e no controle, onde se pretende garantir que o entendimento para a primeira implementação está correto; e controle contínuo que visa garantir que a evolução para esta iteração foi a mais adequada, seja do ponto de vista tecnológico, seja do ponto de vista do negócio realizando assim sempre uma estimativa de quanto ainda precisa ser desenvolvido para atendimento completo da expectativa do cliente.

O tratamento dos dados deve ocorrer por meio de um banco de dados, onde teremos o tratamento de informações geradas internamente pela negociação, mas também informações externas que são obtidas via serviços web ou outros serviços on-line. Assim com base nessas informações podemos realizar o enriquecimento contínuo dos metadados garantindo assim a evolução contínua do *datalake*.

2.4 Pontos relevantes do Metodologia (MoIDSS)

Após uma referência aos processos do PMBoK podemos, de forma mais detalhada, apresentar as camadas e a concepção do uso iterativo do modelo proposto na metodologia.

A camada de ranqueamento e predição, vai apoiar de maneira transversal e produtiva as instancias do modelo durante os (n-1) ciclos iterativos limitados a 24 horas de um ciclo de negociação. Aqui vamos seguir em um modelo em tempo de execução uma vez que toda essa estrutura deve ser executada após as negociações entre os agentes já terem ocorrido. Por fim, nesta camada, o objetivo é determinar predições que possam apoiar na tomada de decisão e posteriormente no enriquecimento dos dados. Além disso, o modelo nos fornece o conceito de ranqueamento dos agentes que estão no ecossistema de negociação, visando pontuar aquele que mais tem negociações fechadas, apresentando assim um reputação do agente dentro daquele ecossistema.

Uma das definições essenciais é sobre o ecossistema de negociação onde se pretende realizar uma simulação do processo de negociação, sabemos que este desenvolvimento pode seguir através de diversos paradigmas, porém entendemos que atualmente o paradigma de agentes é perfeitamente aderente, pois viabiliza uma iteração por meio de troca de mensagens. A camada do ecossistema de negociação, vai tratar do processo de simulação utilizando fatores essenciais para a tomada de decisão que permitirá realizar uma apresentação do fluxo da negociação para o usuário final de modo que fique claro para ele os fatores que influenciaram para decisão do agente de software. O modelo determina que o conhecimento do negócio deve ser aprofundado, pois é necessário que o ecossistema seja modelado com base nas regras de negócios e nos atores envolvidos. Nesta camada do modelo, o objetivo é fornecer um ambiente de negociação, porém de forma a trazer para o ecossistema questões críticas que influenciem no fechamento. Fatores como dólar, pandemias, guerras e guerras comerciais (bloqueios econômicos) podem mudar o rumo de uma decisão de fechamento ou não uma negociação e por isso devem ser levados em consideração.

O tratamento dos dados deve ocorrer por meio de banco de dados que deve ser modelado para garantir que informações internas, relevantes ao processo de negócio sejam persistidas após as análises e os tratamentos adequados. Já os dados externos

devem ser obtidos por meio de serviços web e outros serviços on-line de forma a possibilitar análises técnicas e análises de tendências. Com base nestas fontes de informação deve ser avaliada a possibilidade de implementar um *datalake* que terá seus metadados enriquecidos continuamente. Dessa forma, ao pensar no ranqueamento e na predição, o líder de TI pode considerar a busca de informações históricas na base de dados e além de considerar que a informação é continuamente enriquecida, o que pode melhorar a acurácia do apoio a decisão.

A camada de dados, vai apoiar fornecendo informações de qualidade e assim suprir a necessidade de tomada de decisão, sendo isto realizado, através das análises de dados internos e externos e do enriquecimento dos dados. O modelo sugere que uma análise baseada no cruzamento de dados internos e fatores externos pode ser mais assertiva nas decisões de fechamento ou não do negócio. Após esta análise o modelo determina que um enriquecimento dos dados seja feito, agrupando tanto resultados das análises dessa camada quando da camada de negociação e predição, de modo a manter um histórico mais apropriado e assertivo e ainda apresentar esta informação para a decisão final do usuário.

Nesta camada o modelo preconiza que, devemos utilizar um banco de dados interno para armazenamento dos dados e resultados históricos dos ciclos de negociação dos agentes e dos fatores que influenciam na tomada de decisão. Além disso, o modelo prevê um acesso para a análise em tempo real de fatores de influência que são, por exemplo, dados da bolsa de valores para cotação de moeda, bem como dados de relevância das pesquisas web fornecidos pelo *google trends*.

Na camada de ranqueamento e predição, um ponto relevante foi o desenvolvimento de um algoritmo baseado no *pagerank* para agentes, apelidado de "*agentRank*" este desenvolvimento foi realizado com o intuito de facilitar o uso do ecossistema de negociação e melhorar a experiência do usuário, já que traz a possibilidade de apresentar ao usuário aquele agente melhor qualificado.

Na camada do ecossistema de negociação, um ponto relevante é a utilização de fatores "de risco social", como pandemias, guerras e guerras comerciais, como indicadores para apoiar na análise econômica. Estes indicadores de risco social são obtidos através do serviço do *google trends* e mostram que dentro de um determinado contexto o interesse social sobre um assunto pode antecipar algum movimento financeiro. Desta forma, esta análise segue complementarmente as análises técnicas, como o cruzamento de médias móveis, e sugere bons indícios de que o movimento social pode apoiar antecipando movimentos bruscos do mercado.

Na camada de base de dados, um ponto relevante é a implementação do enriquecimento de dados, uma vez que este desenvolvimento apoia na consciência do sistema trazendo para o modelo de decisão, dados históricos com um maior nível de tratamento, dando mais subsídios para as decisões finais, tomadas pelos usuários.

3 Aplicação (MASoIC) e a Visão Geral Contrato Internacional

A área de contrato internacional é um domínio extremamente vasto, dentro deste processo de negócio pode-se encontrar um extenso grupo de profissionais atuando num mesmo sentido e em sincronia. Um ponto importante é que para reduzir o risco de um contrato de importação, a equipe deve contar com profissionais multidisciplinares que atuando em equipe poderão realizar um contrato internacional com mais assertividade. Esta equipe poderá ser composta por pessoas especialistas em contrato internacional como, trading responsável por avaliar o mercado internacional, *hedge* responsável

por realizar a proteção cambial, além de responsáveis por realizar a análise de notícias que podem influenciar na tomada de decisão.

O *hedge* cambial consiste em um conjunto de instrumentos financeiros, os quais são usados para mitigar o risco financeiro de operações com duas ou mais moedas diferentes. Cenários de exportação, importação e investidores com ativos no exterior, são os principais exemplos de processos onde fazer o *hedge* cambial, se torna fundamental para se proteger da variação cambial e manter o valor de um contrato acordado.

Em um *hedge* de câmbio bem feito, gera pouca alteração no fluxo de caixa, pois esta proteção deverá promover uma compensação total ou parcial da diferença cambial da data da realização do contrato até a data do pagamento do contrato.

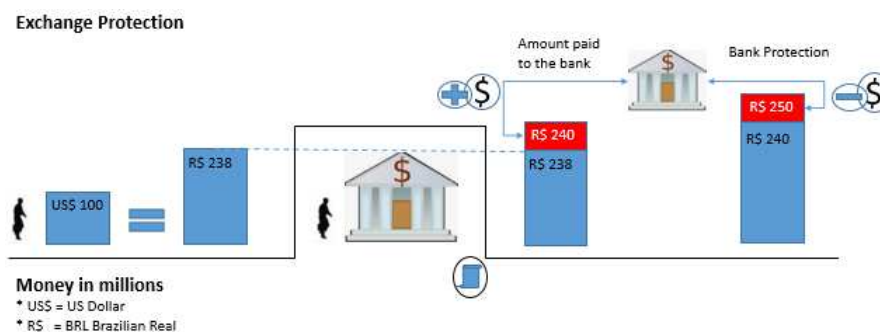


Figura 11. Contratação de hedge cambial

Em um *hedge* de câmbio bem feito, gera pouca alteração no fluxo de caixa, pois esta proteção deverá promover uma compensação total ou parcial da diferença cambial da data da realização do contrato até a data do pagamento do contrato.

Esta proteção é realizada de diversas formas, através de contrato, derivativos, entre outros, no sistema MASoIC vamos abordar aquela que consiste em trabalhar com compra e venda de opções. Para este caso o MASoIC irá recomendar uma opção de venda ou compra, por meio dos seguintes padrão.

Opção de venda, chamada também de PUT é aquela que confere ao seu titular o direito de vender um determinado ativo pelo preço estabelecido no contrato da opção, durante determinado período de tempo ou data pré-determinada.

- O titular é o investidor que comprou a opção de venda no mercado, adquirindo o direito de vender o ativo.
- O emissor é quem lançou a opção de venda no mercado e é aquele que assume a obrigação de comprar o ativo previsto no contrato, após ser comunicado que foi exercido pelo titular.

O objetivo de uma opção de venda é justamente prover proteção ao investidor que possui um determinado ativo, contra a desvalorização deste ativo no mercado.

Opção de compra, chamada também de CALL é uma opção que confere ao seu titular o direito de comprar um determinado ativo pelo preço estabelecido no contrato da opção, durante determinado período de tempo ou em uma data pré-determinada, dessa forma temos que.

- O titular é o investidor que comprou a opção de compra no mercado, adquirindo o direito de comprar o ativo.
- O emissor é quem lançou a opção de compra no mercado e é aquele que assume a obrigação de vender o ativo previsto no contrato, após ser comunicado que foi exercido pelo titular.

O objetivo de uma opção de compra é garantir que um investidor possa comprar determinado ativo no futuro pagando um valor pré-fixado, mesmo que este ativo valorize-se no mercado. Para a realização de um contrato internacional além da estratégia de proteção cambial é necessário observar a cotação do dólar, que é a moeda padrão para negociações internacionais [19]. Assim o objetivo do MASoIC é evitar, sempre que possível, negociação em cenários de grandes altas da moeda americana.

Variation of index, currency and search topics on, 01/27/2020

Date	Assets/Index	Reference	Low; High	Meaning
01/27/2020	Coronavirus	Top trend	+11%	Relevant social interest topic (2nd), may reflect on the stock exchange
01/27/2020	IBOV	MA9; MA21; Setup 9.1	28.435,4 ; 28676,6	Indecision Scenario
01/27/2020	DJI	MA9; MA21; Setup 9.1	114.137,29 ; 118.351,13	Indecision Scenario
01/27/2020	Dollar	MA9; MA21; Setup 9.1	\$4,18 ; \$4,21	Indecision Scenario

Fontes: <https://trends.google.com> | <https://br.tradingview.com> | <https://br.financas.yahoo.com/quote/USDBRL=X>

Figura 12. Tabela comparativa de fatores sociais e econômicos em 27/01/2020

Para atingir este objetivo utilizamos técnicas de análise gráfica onde podemos determinar tendência. Neste caso, o MASoIC utiliza as técnicas de média móvel e os setups de compra e venda, mais especificamente a média móvel de 9 e 21 dias e o setup 9.1 tanto para compra quando para venda.

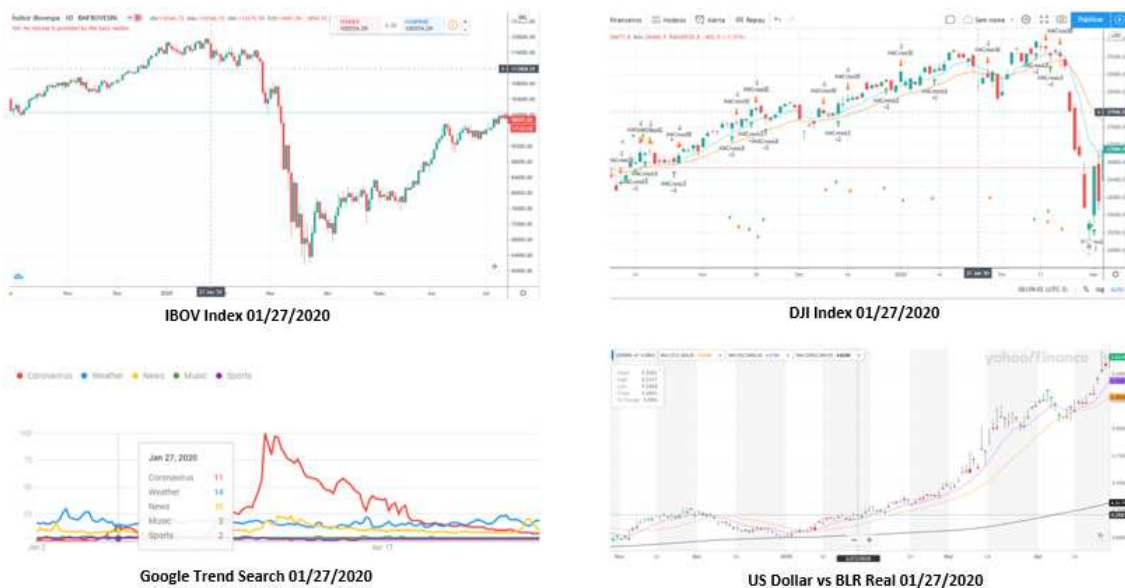


Figura 13. Gráficos de índices, dólar e interesse social em 27/01/2020

Como sabemos as negociações na bolsa de valores intrinsecamente representam um desejo humano, uma vez que mesmo que operadas por robôs, estes são programados por um ser humano para atingir seus objetivos. Assim todo o gráfico e as análises gráficas que fazemos refletem na verdade o comportamento humano [2].

Variation of index, currency and search topics on, 03/02/2020

Date	Assets/Index	Reference	Low; High	Meaning
03/02/2020	Coronavirus	Top trend	+25%	Topic of greatest social interest, great chance to reflect on the stock exchange and on the exchange
03/02/2020	IBOV	MA9; MA21; Setup 9.1	25.545.20; 26.687.77	Upward trend
03/02/2020	DJI	MA9; MA21; Setup 9.1	104.000.00 ; 106.955.55	Upward trend
03/02/2020	Dollar	MA9; MA21; Setup 9.1	\$4,44 ; \$4,50	Upward trend

Fontes: <https://trends.google.com> | <https://br.tradingview.com> | <https://br.financas.yahoo.com/quote/USDBRL=X>

Figura 14. Tabela comparativa de fatores sociais e econômicos em 02/03/2020

Por isso consideramos também na análise a questão social associada a busca de termos na internet baseada na ferramenta *google trends*, assim o aumento do interesse por algum tema pode representar uma confirmação de tendência [1].



Figura 15. Gráficos de índices, dólar e interesse social em 02/03/2020

Em uma análise histórica observamos que em 27/01/2020 a preocupação da sociedade com relação ao assunto (ex.: coronavírus) aumentou significativamente ultrapassando assuntos como esporte e música o que indica que provavelmente as bolsas em geral terão algum um reflexo.

Em 02/03/2020 percebemos que a preocupação da sociedade com relação ao assunto (coronavírus) aumentou ainda mais sendo o tópico mais procurado dentre assuntos como esporte, clima, música e notícias ultrapassando 25% do índice de busca o que indica que há uma possibilidade mais significativa que em 27/01/2020 de que esse interesse social gere um impacto nas bolsas em geral.

Em ambos os casos é possível observar uma queda nas bolsas e uma alta de ativos mais seguros como o dólar e ouro, sendo que em 02/03/2020 o período que segue teve um impacto mais acentuado. Assim o que a aplicação MASoIC nos mostra é que há uma forte correlação quando o interesse social por um determinado termo passa dos 25% [20][21]. Dessa forma, estas informações podem subsidiar uma tomada de decisão com maior segurança (com embasamento técnico e de notícias) para apoiar nesta tomada de decisão, utilizamos um fluxo iterativo (de ciclos de 24 horas) que irá agregar, trazendo um histórico mais consistente aos agentes onde ocorrerá o enriquecimento de metadado e a classificação dos agentes por meio do *agentrank* baseado no *pagerank*.

4 Trabalhos Relacionados

Neste tópico vamos falar brevemente de dois artigos correlacionados com este trabalho, são eles: “*A Multi-Agent Decision Support System for Stock Trading*” escrito por Yuan Luo, Kecheng Liu e Darryl N. Davis [22], o outro é o “*A decision support framework and prototype for aircraft dispatch assessment*” escrito por H.Koornneef, W.J.C.Verhagen, R.Curran [3].

O primeiro trata de um sistema de apoio a decisão para compra e venda de ações, onde um dos objetivos é a apoiar o investidor no sentido de definir qual momento é a hora certa de comprar uma ação, a aplicação (MASST) permite que o investidor defina a sua própria estratégia de compra e venda de ações e regras. Através de uma combinação de conhecimento humano e de máquina, usando tecnologias de agentes e inteligência artificial, o MASST visa reduzir sobrecarga de trabalho dos investidores no processo de análise de ações e tomada de decisão sobre investimentos.

Neste artigo os autores entendem que duas questões são as mais importantes, são elas, como agentes individuais devem ser interconectados para que suas capacidades sejam usadas eficientemente e seus objetivos sejam realizado de forma eficaz e eficiente. E a segunda questão refere-se a como deve ocorrer a transferência de informações entre agentes para permitir que respondam com sucesso a solicitação do usuário e situações inesperadas [22].

No segundo artigo vemos um sistema de apoio a decisão para determinar se uma aeronave que teve algum problema em voo pode realizar o próximo voo. O sistema traz informações para subsidiar a decisão do técnico que deve fazer essa análise, este é considerado um apoio essencial, pois a avaliação deve ser feita no intervalo entre um voo e outro, ou seja, é o tempo que a aeronave está no chão.

Neste artigo os autores entendem que três desafios são considerados os maiores no domínio do suporte à decisão: 1) a escassez de modelos e aplicativos de suporte à decisão para os processos operacionais em manutenção; 2) há poucos esforços na aplicação e avaliação de artefatos em ambientes operacionais experimentais e da vida real; e 3) falta de sistemática desenvolvimento, aplicação e avaliação de esforços de digitalização

e automação de processos complexos de decisão em manutenção. Para tratar esses desafios os autores propõem atuar no campo do design e a tomada de decisão em tempo real, trazendo um protótipo baseado na web acessível por soluções móveis que foi validada por meio de dois estudos representativos de aplicação e avaliação, um em um ambiente experimental e outro em um ambiente operacional onde os resultados, segundo os autores, mostram uma economia significativa de tempo e fortes indicações qualitativas para um maior incentivo ao uso da documentação e redução dos fatores de risco humanos que levam a erros de manutenção [3].

Três artigos (dentre vários) que serviram como inspiração no desenvolvimento do guia ágil do modelo desta pesquisa, são eles: “A survey of autonomic computing methods in digital service ecosystems” - Dhaminda B. Abeywickrama¹ · Eila Ovaska¹ [30]; “SOTA: Towards a General Model for Self-Adaptive Systems” - Dhaminda B. Abeywickrama, Nicola Biccchi, Franco Zambonelli [31]; “Adopting Autonomic Computing Capabilities in Existing Large-Scale Systems” - Heng Li, Tse-Hsun (Peter) Chen, Ahmed E. Hassan, Mohamed Nasser, Parminder Flora [32].

O primeiro artigo trata de uma pesquisa de diversos modelos de computação autônoma para um ecossistema de serviços digitais. Para seus autores este tipo de ecossistema pode ser associado a vários desafios, como mudança e evolução dos requisitos; coleta de requisitos de qualidade e avaliação. A incerteza causada por esta dinâmica de natureza e ambiente de implantação desconhecido (composição e usuários), nos leva portanto, a busca de soluções autônomas que venham suprir tal necessidade.

Neste artigo os autores entendem que um ecossistema de serviço digital o foi definido como um ambiente de agentes abertos, fracamente acoplados, agrupados por domínio, orientados pela demanda e auto organizados, onde cada entidade é proativa e responsiva para seu próprio benefício. Assim eles definem um framework de características para padronizar a análise dos modelos de computação autônoma; este framework foi baseado em quatro características, chamados de elementos essenciais para avaliar uma metodologia: método do contexto, método do usuário, método do conteúdo e método de avaliação. Ainda no primeiro artigo após analisarem diversos modelos os autores apresentam um fluxo de dependência entre as características definidas para construção do framework de análise de modelos que ocorrem a partir do decorrer do tempo [30].

No segundo artigo vemos um novo modelo geral denominado SOTA que segundo os autores servirá para modelar a adaptação dos requisitos no decorrer do tempo. Os autores também explicam no artigo que o SOTA permite: verificação antecipada de requisitos, identificação de requisitos de conhecimento para auto adaptação, e identificação dos padrões auto adaptativos mais adequados, pois, reuni as lições de modelagem orientada a objetivos e de modelagem de sistema ciente do contexto, assim tem potencial para lidar com algumas questões chave no design e desenvolvimento de sistemas complexos auto adaptativos [31]. Neste artigo um ponto interessante é a abordagem de hiperespaço (S_1, \dots, S_n), onde cada dimensão se dá em função da variável tempo e os objetivos têm suas pré e pós-condições, que no percurso de entre um e outro identifica a utilidade.

No terceiro artigo os autores, trazem uma abordagem de reengenharia através de um processo de adicionar recursos de computação autônoma para esses sistemas existentes [32]. Os autores relatam que este trabalho é particularmente desafiador, por causa de três pontos, são eles: a quantidade significativa de esforços que são necessários para investigar e refatorar o código base existente; o risco de adicionar

complexidade adicional; as dificuldades para alocar recursos enquanto os desenvolvedores estão ocupados adicionando recursos essenciais para o sistema.

Neste artigo os autores compartilham a experiência na indústria de reengenharia de recursos de computação autônoma para um sistema de software existente em grande escala. Uma abordagem interessante deste artigo é a definição de um ciclo de reengenharia e de inserção de recursos de computação autônoma.

Alguns pontos apresentados pelos autores também foram um desafio para o trabalho apresentado, no caso do guia da metodologia, o primeiro desafio foi realizar um estudo para viabilizar criação da metodologia que funcionasse como um guia e utilizasse conceitos de métodos ágeis, (ex.: Kanban [34], Scrum [33], CBL [35]), estruturando para isso perguntas fixas que irão direcionar ao início da utilização da metodologia e na sequência perguntas variáveis que deve seguir de forma iterativa em um processo evolutivo incremental, atualizando o backlog conforme os pacotes forem para o ambiente produtivo criando assim uma consciência conforme esperado. Já no caso da implementação da aplicação MASoIC, tratamos a comunicação entre os agentes de forma diversificada de modo que suas capacidades sejam aproveitadas da forma mais otimizada possível, conforme preconiza a metodologia MoIDSS, por isso em algum momento a comunicação é feita através de um agente coordenador e em outras diretamente de um agente para outro. Já a transferência de informações entre os agentes é realizada através dos tipos de mensagens disponibilizados pelo framework JADE, assim toda a estrutura pode ser aproveitada para viabilizar a troca de informação. Para uma tomada de decisão em tempo real a proposta é que os agentes buscam seus conjuntos de informações em tempo real, neste caso estes desafios passam pela busca de valores análise de tendência e setups da bolsa de valores referente ao dólar (moeda padrão para negociações internacionais) até a busca de informações percentuais referente ao interesse social de um dado termo de modo a complementar as análises econômicas clássicas e assim evitar um risco ainda maior. A fim de viabilizar uma tomada de decisão em tempo real o sistema retorna todas as análises para um sistema de mensagem instantânea (ex.: Telegram) para que então o usuário possa tomar sua decisão em qualquer lugar que esteja.

Para este estudo foram realizados experimentos com dados reais de datas passadas e também dados reais de datas atuais, onde tivemos fortes indicações qualitativas de que a análise de fatores de busca apoiam em uma confirmação de tendência, o que também apoia na tomada de decisão para efetivação de um contrato de importação.

5 Processos Utilizados na Pesquisa

Para o desenvolvimento do projeto foi utilizado o paradigma de Orientação a Objeto e de Orientação a Agentes, fazendo uso da UML para modelagem orientada a objeto e para a orientação a agentes. As seguintes tecnologias foram utilizadas para o desenvolvimento do sistema: Java, Python, mql5, RPA, JADE e Banco de Dados Mysql.

A implementação foi totalmente desenvolvida com a IDE eclipse jee Indigo SR2. A aplicação proposta tem a seguinte arquitetura, o sistema é composto de classes agentes de software e tanto os agentes quanto as classes comuns acessam o banco de dados, a bolsa de valores e a web.

Neste tópico vamos falar sobre a representação do fluxo iterativo da arquitetura baseada na metodologia MoIDSS. Onde na camada de negociação foi implementado um sistema multi-agente, onde foram depositadas as políticas e metas do sistema autoconsciente. Neste ponto podemos definir que novas metas podem ser inseridas no sis-

tema por agentes externos, ou seja, pessoas ou sistemas; ou ainda novas metas podem ser geradas em tempo de execução pelo próprio sistema.

Na camada de dados, onde repousa toda a instância de dados que representam o conhecimento de negócio para um certo domínio, podendo estar armazenadas em banco de dados, csv, txt ou frutos de serviços de plataformas que rodam em tempo real.

Na camada de predição e ranqueamento, é onde realizamos o controle por meio de proposições lógicas e algoritmos de ranqueamento e predição para que assim seja possível realizar as interferências na camada de dados e também na camada multi-agentes, porém na camada multi-agente a predição realizará consultas para identificar as políticas do sistema.

Por fim as interfaces de comunicação entre as três camadas do modelo, disposto na metodologia MoIDSS, funcionarão como "Modulo de Controle Contínuo" onde recebe o *feedback* do sistema monitorado e com base nisto aplica as mudanças necessários para a adequação.

Para executar este cenário vamos utilizar toda a estrutura básica provida pela plataforma JADE o que irá apoiar a solução e facilitar a sua aderência ao fluxo do cenário proposto no mundo real.

O modelo básico do *framework* traz três agentes, são eles, o agente MAS, o DF e o ACC [10]. O agente AMS (*Agent Management System*), é o responsável por supervisionar a plataforma e gerenciar o ciclo de vida dos agentes;

O agente DF (*Directory Facilitator*), é o responsável pelo serviço de páginas amarelas, que é o local onde é feito o registro de serviços realizados pelos agentes, os agentes também utilizam o DF para buscar e requisitar serviços e também para criação de confederações;

O agente ACC (*Agent Communication Channel*), é aquele que gerencia a comunicação entre os agentes dentro e fora da plataforma.

Os princípios de Uniformidade e Portabilidade dos agentes é preservado plataforma JADE, dessa forma, o *framework* provê um conjunto homogêneo de APIs (*Application Programming Interfaces*) que são essenciais para simplificar o desenvolvimento e a manutenção de uma aplicação além de possibilitar uma evolução simplificada no que diz respeito a plataforma da aplicação [10, 11].

A comunicação é reconhecida como tópico central de importantes estudos em ciência da computação onde o foco está em sistemas que interagem entre si, e o problema clássico é o de sincronizar vários processos.

Essencialmente, dois processos precisam ser sincronizados se houver possibilidade de que eles possam interferir um no outro de uma maneira destrutiva, como por exemplo no cenário clássico de perda de upload.

No paradigma de orientação a agente a comunicação é melhor explicada por meio de um exemplo, dado dois agentes "i" e "j", sabemos que eles têm capacidade de executar a ação "a", porém não se pode presumir que o agente "i" executará a ação "a" apenas porque outro agente "j" deseja, ou seja, executar a ação "a" pode não ser do interesse do agente "i" [9].

A teoria de John Austin (1962) *Speech Acts*, trata a comunicação como ação e é baseada no pressuposto de que as ações de fala são executadas por agentes como outras ações. Assim as comunicações são modeladas como ações que alteram o estado mental dos participantes da comunicação, com base em uma condição de felicidade [18].

Em 1969 John Searle fez uma extensão da *Speech Acts* classificando o atos de fala em *Representatives* que compromete o agente que fala (Speaker) com a verdade de uma proposição expressa por ele, neste sentido temos: *Directives* que é uma tentativa por parte do agente que fala (Speaker) em obter do Ouvinte (Hearer) alguma coisa, *Commissives* que compromete o agente que fala (Speaker) no fluxo da Ação, *Expressives* que é o agradecimento, *Declarations* que efetua uma mudança em um estado institucional mudando assim o comportamento do agente [18].

Neste sentido, o agente de software é uma metáfora natural para o cenário proposto no trabalho sendo facilmente representado e simulado através de um sistema multi-agente uma vez que, a negociação para realização de um contrato, se dá através da comunicação por meio de troca de mensagem que executam ações dentro de uma classificação e assim alteram o estado mental do agente o que pode resultar ou não em uma mudança de comportamento.

Para a implementação deste primeiro protótipo, baseado no modelo, utilizamos o framework JADE para agentes de software, pois com ele é possível utilizar uma estrutura de comunicação baseada em mensagens assíncronas que segue o padrão FIPA-ACL e em uma topologia de rede *peer-to-peer* [11]. Aliado a estes conceitos vamos utilizar o enriquecimento de metadados de forma cooperativa e iterativa aos resultados obtidos do algoritmo *agentrank* baseado no *pagerank* e as previsões baseadas em LSTM, e assim apoiar tanto no processo de análise e estruturação das informações para a tomada de decisão quanto para a própria tomada de decisão.

Pensando no desenvolvimento, vemos o framework JADE como uma boa opção, pois propicia a união destes primeiros conceitos permitindo a construção de um sistema adaptativo que irá suportar todo o ecossistema de simulação para a (construção de toda a informação que irá subsidiar a) tomada de decisão.

Para isso, utilizamos o MTS - *Message Transporte System* do padrão FIPA-ACL para troca de mensagens entre agentes, componente que já provê um grupo de mensagens pré-classificadas e um modelo de comportamento simples ou paralelo.

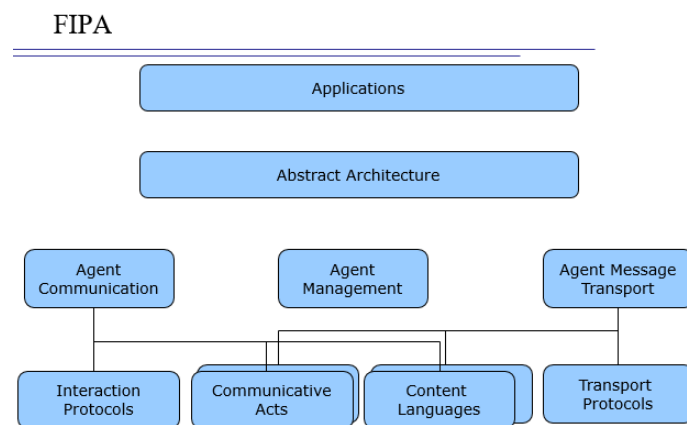


Figura 16. Arquitetura padrão FIPA para agentes

No trabalho proposto foram utilizadas as mensagens do tipo: REQUEST, INFORM, CFP, PROPOSE, ACCEPT_PROPOSAL, AGREE e REFUSE. Sendo utilizado também os comportamentos para o tratamento mais natural do ponto de vista de comportamento esperado.

- `addBehaviour(new CyclicBehaviour () { });`
- `parallelBehaviour.addSubBehaviour (new CyclicBehaviour () { });`

Para realizar a consulta de ativos na bolsa de valores o agente faz uso da linguagem MQL5 que permite uma consulta a bolsa de valores, em tempo real, mas para isso é necessário ter uma conta ativa e válida em uma corretora que realize operações na bolsa de valores desejada. Com isso, o agente é capaz de realizar as devidas análises econômicas referente ao dólar que é a moeda padrão em negociações internacionais.

Além disso, utilizamos um ranqueamento dos agentes “*agentrank*” baseado no algoritmo *pagerank* para determinar aquele agente mais representativo do ponto de vista do cenário de negociação de contratos de importação. Ainda para refinar as informações referentes a tomada de decisão, programas auxiliares ao sistema multiagente, desenvolvidos em Python visam dar uma previsão dos próximos preços do dólar, com base na série histórica e apoiado em um algoritmo de LSTM. Desta forma, o ciclo iterativo de aprendizagem e adaptação do sistema multiagente se completa com um processo de enriquecimento do metadado que deve ser realizado com base em informações previamente existentes e aquelas geradas ao final de cada ciclo, assim, com o enriquecimento de metadado, as informações para a tomada de decisão serão mais efetivas e objetivas.

No artigo *What is assessment for learning?* [2] onde é demonstrado que compreender o impacto que a avaliação tem na aprendizagem requer um foco mais amplo que o *feedback*, sendo necessário uma maior atenção para o meio de aprendizado, aqui chamamos a atenção para uma análise de risco social que pode apoiar e complementar a análise econômica. Para realizar as consultas a riscos sociais, o agente de software se utiliza da tecnologia RPA *Robotic Process Automation* para obter os dados referente ao interesse social de uma região por um termo específico. A variação do interesse em um termo tornará aquela região, em um dado momento, mais ariscada ou não para a realização de negócios e isto deverá ser levado em consideração na tomada de decisão final. O sensor social, baseado no *google trends* traz para o sistema multi-agente essa percepção e com isso é possível realizar gatilhos para evitar um risco excessivo, por exemplo, no caso de detectarmos 100% de interesse em um termo de pesquisa, o gatilho é ativado, pois neste caso, a probabilidade deste termo de pesquisa influenciar no contrato em questão é concreta.

Para gerenciar o desenvolvimento e auxiliar nos prazos de desenvolvimento da aplicação foi utilizada a metodologia MoIDSS que traz características de métodos ágeis como o SCRUM e o KANBAN. A metodologia foi utilizada justamente para pôr a prova sua usabilidade e agilidade no desenvolvimento de sistemas inteligentes de apoio a decisão.

Em um ambiente acadêmico, podemos perceber que a metodologia se adequa a necessidade, que é dar todo o ferramental ao desenvolvedor e mantê-lo focado em um trabalho de alto nível e concentrado em entregas de pacotes menores dedicados ao ponto central do processo escolhido. Com o controle contínuo que a metodologia traz temos também em mente o conceito de Produto em Desenvolvimento, ou seja, um software sempre em desenvolvimento, mas sempre utilizável e funcional, atingindo assim o objetivo final que é fazer com que a metodologia MoIDSS seja o motor que impulsiona as entregas com agilidade e com maior frequência.

6 Agentes do Sistema Inteligente de Apoio a Decisão (MASoIC)

Nesta seção, serão apresentados os tratamentos realizados por cada um dos agentes do sistema MASoIC. O sistema MASoIC conceitualmente é composto por sete (7) agentes, porém os agente exportador pode ser instanciado na medida que forem entrando mais

agentes exportadores no sistema. Os sete agentes implementados no sistema são denominados de *agente importador*, *agente exportador coordenador*, *agente exportador*, *agente trading*, *agente hedge*, *agente news* e por fim o *agente mensagem* utilizado para comunicação com o usuário final através do aplicativo de mensagem *Telegram*. Para o desenvolvimento dos agentes foi utilizado o framework JADE. O framework citado é utilizado para a construção de agentes de software e possui a característica de comunicação através da troca de mensagens, além de ter o comportamento de um *peer* de uma rede *peer-to-peer*, facilitando a distribuição do sistema.

Neste sistema foram exploradas as vantagens do paradigma de agentes através da troca de mensagens, para tanto foi utilizado o (DF) *Directory Facilitator*. O DF disponibilizado pelo framework JADE provê o serviço de páginas amarelas onde cada agente cadastra o nome do serviço prestado por ele, de forma que qualquer outro agente possa procurá-lo, ou seja, a priori não é necessário que os agentes se conheçam [6].

No cenário utilizado nos testes os agentes exportadores registram serviço de venda de gasolina indicando o número identificador da empresa, o nome da empresa, o produto que ele quer exportar, o preço de venda por metro cúbico e o país da empresa exportadora.

Nesse sentido na importação de produtos, observamos o conceito básico, onde um agente quer importar um produto e recebe diversas propostas de agentes exportadores, com isso o agente importador escolhe a melhor proposta seguindo alguns critérios, conforme descrito a seguir.

Em primeiro lugar o melhor preço, levando em consideração a variação cambial, este é um critério objetivo, que é recebido e enviado pela interface sensor do tipo *perform-operation* e pela interface atuadora do tipo *call-out request* garantindo assim o melhor valor para o contrato.

Em segundo lugar a melhor qualidade da informação, que é um critério subjetivo, e que pode ser retroalimentado no sistema pelo *feedback* do importador e do algoritmo *pagerank*. Neste caso, o importador responderá um breve questionamento no formato de formulário solicitado pelo sistema, já o algoritmo será executado de forma a realizar uma classificação dos agentes visando apresentar aquele que tem mais requisições o que implica de forma implícita na qualidade.

Por fim o terceiro critério, deve selecionar aquele candidato que traz o menor risco social, este também é um critério subjetivo, o qual recebe os tópicos de tendências que influenciam o mercado, como por exemplo, Pandemia, Guerra, Guerra Comercial, entre outros. Com estes tópicos determinados efetua a busca seguindo a heurística determinada no trabalho para dizer se o candidato exportador tem um risco social alto. Nesta primeira fase do trabalho o "*trend*" foi fixado, mas para as demais fases a ideia é que seja realizada uma observação do meio (*retrieve-state*) para perceber aquele "*trend*" que merece ser observado.

O *agente importador* é o agente que faz a interface com os usuários que tem a necessidade de importar e é o encarregado de solicitar ao *agente trading* uma importação indicando para isso os dados informados pelo usuário, quer seja, o número da empresa ao qual ele representa, o nome da empresa, o produto que deseja importar, o volume total referente ao produto que se quer importar, além do preço que se pretende vender aquele produto. Estas informações são armazenadas no banco de dados e utilizada pelo *agente trading* para realizar as negociações com os *agentes exportadores*.

Figura 17. Input do importador

O *agente trading* é o encarregado de solicitar periodicamente ao *agente DF* do JADE todos os serviços associados aos *agentes exportadores* para isso foi definida uma implementação no JADE para possibilitar a identificação e conseqüente segregação dos agentes que tem o serviço de exportação dos demais serviços de venda. Dessa forma, ciclicamente o agente trading conhecerá todos os agentes que prestam serviço de exportação e com base nestes agentes irá selecionar apenas aqueles que prestam serviços de exportação referente ao produto indicado pelo *agente importador*. O *agente trading* também será o responsável por realizar a chamada ao *agente hedge* e ao *agente news* para que eles então realizem as análises de risco inerente aos *agentes exportadores* listado naquele momento da simulação, para que assim sejam obtidas informações para a definição de um plano estratégico com relação ao contrato de importação [7].

O *agente hedge* é aquele que recebe a convocação do *agente trading* para realizar as análises de risco inerentes a moeda, quer seja, o dólar. Ele se encarregará de solicitar a bolsa de valores (IBOVESPA B3) a cotação do contrato de dólar que esteja vigente naquele momento, buscando sua variação de uma em uma hora nos últimos três meses até o mês corrente, dessa forma, obtendo um histórico que viabiliza a realização da média móvel de 9 dias e de 21 dias, além do setup de compra e venda 9.1, o que possibilita uma curva de tendência e a indicação de um ponto compra ou venda do ativo.

O *agente news* também receberá um chamado do agente trading para que realize a sua análise de risco que neste caso é o risco social que complementa a análise de risco econômico realizada pelo hedge.

```
parallelBehaviour.addSubBehaviour(new CyclicBehaviour() {
    private static final long serialVersionUID = 1L;
    public void action() {
        MessageTemplate messageTemplate = MessageTemplate.or
            (MessageTemplate.MatchPerformative(ACLMessage.REQUEST),
             MessageTemplate.or
                (MessageTemplate.MatchPerformative(ACLMessage.PROPOSE),
                 MessageTemplate.or
                    (MessageTemplate.MatchPerformative(ACLMessage.INFORM),
                     MessageTemplate.or
                        (MessageTemplate.MatchPerformative(ACLMessage.AGREE),
                         MessageTemplate.MatchPerformative(ACLMessage.REFUSE))));
        ACLMessage aclMessage2 = receive(messageTemplate);
        if(aclMessage2 != null){
            switch(aclMessage2.getPerformative()){
                case ACLMessage.INFORM:
```

Figura 18. Template utilizado pelo agente trading

Neste momento o *agente news* é o encarregado de realizar a busca por critérios que complementarmente indiquem uma probabilidade de variação na bolsa de valores e na moeda (dólar), neste estudo foram utilizados os termos (epidemia, pandemia, coronavírus e guerra comercial) executados na ferramenta Google Trends e a automação da sua avaliação foi feita com o uso de técnicas de RPA e tratamento de CSV. Neste caso, não nos preocupamos com a detecção de fraude ou *fakenews* [8][6], pois se uma fraude ou *fakenews* chegou até esse ponto ela irá se refletir em todo meio ambiente.

O *agente exportador* é o agente que faz a interface com os usuários que tem a necessidade de exportar e é o encarregado de solicitar ao *agente exportador coordenador* que realize o seu registro no serviço de páginas amarelas, indicando os dados informados pelo usuário, quer seja, o número da empresa ao qual ele representa, o nome da empresa, o produto que deseja exportar, o preço do produto por metro cúbico, o prazo de paga-

mento, além do seu país de origem. Estas informações são armazenadas no banco de dados e informadas ao *agente trading* para que ele realize as análises pertinentes para a realização de contrato internacional.

O *agente exportador coordenador* é o encarregado de receber os *agentes exportadores* que estão entrando no sistema e solicitar o registro do serviço destes ao *agente DF* do framework JADE. O *agente exportador coordenador* também faz o seu registro perante o *agente DF*. O *agente mensagem* é o encarregado de enviar as mensagens para o usuário final informando dos passos da simulação no sistema multi-agente de apoio a decisão em tempo real [3][4]. Nesta construção o *agente trading*, o *agente exportador coordenador* e o *agente mensagem* trabalham em conjunto para realizar a classificação, sendo o *trading* o responsável por saber dos importadores que vão negociar e o *exportador coordenador* o responsável pelos exportadores, a partir desse ponto o algoritmo *agentrank* começa a sua execução com um grafo inicial provisório que só poderá ser confirmado após resposta vinda do agente mensagem.



Figura 19. Mensagem enviada em tempo real para o importador

Após o retorno da mensagem o algoritmo pode confirmar o grafo da primeira iteração, e assim o algoritmo deverá seguir sua execução respeitando o ciclo de negociação estabelecido na metodologia.

Após a conclusão de um ciclo de negociação os dados de classificação dos agentes retomam para o sistema trazendo assim mais um critério para o apoio a decisão realizado pelo sistema.

6.1 Por que Agentes?

O porque utilizar ou não uma determinada abordagem é uma questão clássica e sempre nos ronda nas discussões sobre paradigma e a tecnologia. Um ponto primordial na abordagem de agentes de software, utilizando JADE é que este framework traz algumas características que propiciam a distribuição do sistema como, por exemplo, uma topologia de rede *peer-to-peer* e o paradigma de agentes através das trocas de mensagens [10, 18]. Antes mesmo dos microserviços, conceito que surgiu em 2012, o paradigma de agentes já ajudava a dividir o problema inicial em escopos menores onde podemos obter campos do conhecimento bem definidos, desta forma, facilitar a compreensão e o desenvolvimento do sistema.

Os sistemas multi-agentes comumente assumem alguma forma, ou de estrutura de dados, ou de conjunto de normas, ou de convenções, para assim viabilizar a articulação ou restrição, na interação entre os agentes e assim tornar mais eficaz o sistema que utiliza esta abordagem.

Nos dias de hoje, um desejo comum dos usuários é que um aplicativo tenha uma uniformidade de comportamento em qualquer uma das plataformas disponíveis. Este é um ponto positivo para o uso de agentes, pois duas importantes características desta abordagem são a Uniformidade e a Portabilidade de plataforma [5] que tornam possível atingir os anseios de dois grupos, quer sejam, os usuários que querem uma maior usabilidade e disponibilidade, quer sejam, os desenvolvedores que almejam por uma estrutura que facilite tanto a manutenção quanto a evolução de um dado projeto futuramente.

O cenário atual e previsões futuras da computação móvel é de uma vibrante mobilidade e sensibilidade ao contexto com os vestíveis (*wearable*), o que representa a abordagem de agentes uma outra vantagem, pois neste ambiente temos um cenário típico em que as normas ou políticas estão constantemente mudando, geralmente em função do contexto. Nesse cenário, vemos que os agentes podem se adaptar a cada novo conjunto de regras e políticas, que serão inseridas ao comportamento do agente.

Outra motivação para o uso de agentes de software são suas propriedades intrínsecas como, por exemplo, autonomia, mobilidade, pró-atividade. Tais propriedades são exploradas plenamente quando, por exemplo, informações vindas do contexto de um aplicativo móvel geram um conjunto de políticas para compor o comportamento do agente e assim definir um perfil, que irá estabelecer aquilo que é permitido ou proibido para um agente executar. Corroborando com o uso de agentes de software temos os modelos de computação autônoma que apoiam no desenvolvimento provendo toda a estrutura para que o software venha a ter um certo nível de consciência.

Além disso, o conceito da computação autônoma pode fornecer uma estrutura de base de conhecimento que se retro alimenta através do *feedback* que recebem do meio e é processado após um ciclo de uso e também após uma passagem de um ciclo de monitoramento, análise, planejamento e execução. Deste modo, as propriedades de auto* ou *self** como, por exemplo, autogerenciamento, autoproteção, podem garantir no decorrer do tempo que este software será menos afetado pela degradação.

Do ponto de vista da implementação do projeto proposto (Sistemas Multi-agentes para Contratos de Importação) a abordagem de agentes de software com JADE traz mais uma vantagem sobre a abordagem de desenvolvimento de software com classes thread, que é a implementação do padrão de projeto *Chain of Responsibility* realizada pelo framework JADE, esta cadeia de responsabilidade é uma importante estrutura do sistema proposto e representa uma grande vantagem no uso de agentes neste cenário.

7 Discussão e Conclusão

Esta seção retrata as discussões que foram feitas com base na aplicação e a conclusão com base nos testes realizados.

Neste trabalho tratamos da metodologia para o desenvolvimento de sistema inteligentes de apoio a decisão e para comprovar a eficácia do seu uso aplicamos no desenvolvimento de um sistema multi-agente baseado em JADE para apoio a decisão na efetivação de um contrato de importação. Para este tema utilizamos a comunicação entre os agentes de forma diversificada explorando os tipos de mensagem, para realizar a troca de informação, disponibilizadas pelo framework e as estruturas de gerenciamento e registro de serviços. Neste sentido, focamos em aproveitar as capacidades do agente da forma mais otimizada possível, por isso em algum momento a comunicação é feita através de um agente coordenador e em outras diretamente de um agente para outro.

Uma vantagem no uso da metodologia, é que ele facilita o raciocínio e guia para uma composição de arquitetura robusta, trazendo para o desenvolvimento conceitos como de agentes de software, que viabiliza uma estrutura de comunicação que propicia uma simulação de uma negociação real [10]. Já o enriquecimento dos dados, as predições e o ranqueamento vão apoiar na construção de informações adequadas para a tomada de decisão. Além disso, a metodologia traz quatro artefatos bastante elucidativos para a etapa de construção do sistema, o modelo em 3 camadas que serve como ponto de partida para o raciocínio e a organização do sistema, o framework conceitual de negociação que disponibiliza uma estratégia de comunicação que pode simular qualquer situação de negociação, o guia ágil para uso da metodologia, que traz para cada fase métodos com perguntas essenciais que visam dar maior celeridade e maior assertividade na execução da metodologia e o modelo do *agentrank* que apresenta um algoritmo que permite realizar a classificação dos agentes envolvidos na negociação. Estes artefatos apresentados pela metodologia se mostraram ser de fácil manuseio e bem úteis no sentido de agilizar o processo de construção da lógica de um sistema inteligente de apoio a decisão, assim, podemos dizer que a metodologia apoia bem tanto usuários iniciantes quanto avançados.

No que diz respeito ao ranqueamento, a proposta do *agentrank*, que é uma adaptação do *pagerank*, se mostrou bastante adequada uma vez que possibilitou a classificação dos agentes que estão negociando. Assim em cada ciclo, a execução do *pagerank* realiza uma classificação para os agentes criando assim o indicativo de qualidade para cada um dos agentes. Assim podemos concluir que esta modificação trouxe um ganho se comparada a versão original uma vez que conseguimos obter esse indicativo de qualidade sem gerar confusão nos papéis que cada agente assume em uma negociação.

Para uma tomada de decisão em tempo real a proposta é que os agentes busquem seus conjuntos de informações de fatores externos em tempo de execução, assim eles passam pela busca de valores, análise de tendência e setups da bolsa de valores referente ao dólar, moeda padrão para negociações internacionais. Em paralelo realizam também a busca de informações percentuais referente ao interesse social sobre um dado termo de modo a complementar as análises econômicas clássicas e assim evitar um risco ainda maior.

A fim de viabilizar uma tomada de decisão em tempo real o sistema retorna todas as análises para o usuário através de um sistema de mensagem instantânea (Telegram) para que então o usuário possa tomar sua decisão levando em consideração os fatores externos mais relevantes para sua decisão, quer seja o risco econômico e o risco social (tendências).

Com relação ao sensor social verificamos que quando o fator de pesquisa de um determinado termo passa de 25% a sua influência na economia já é considerável assim os agentes exportadores passam a indicar um risco moderado para o agente trading, subindo em uma classificação a cada 10 pontos percentuais até que o fator de pesquisa de um determinado termo chegue a 100% indicando um risco alto onde o agente exportador sai da lista.

Desta forma, o sistema multi-agente passa a internalizar os interesses sociais e econômicos e dinamicamente, acompanhar a evolução dos interesses sociais o que irá possibilitar a análise conjunta dos riscos inerentes a realização do contrato naquele momento.

Para este estudo foram realizados experimentos com dados reais de datas passadas e também dados reais de datas atuais, onde tivemos fortes indicações qualitativas de que a análise de percentuais referente ao interesse social sobre um termo buscado

apoiam em uma confirmação de tendência, o que também apoia na tomada de decisão para efetivação de um contrato de importação.

8 Trabalhos Futuros

Neste trabalho conseguimos demonstrar que a metodologia (MoIDSS) pode auxiliar de forma bem precisa e ágil, na construção de um sistema inteligente de apoio a decisão, para isso construí uma aplicação protótipo (MASoIC) como prova de conceito. Para os próximos passos vamos utilizar a metodologia em um projeto real para atestar a sua performance fora do ambiente acadêmico.

Os próximos passos vão contemplar, o que podemos chamar de, três *sprints*, sendo que a primeira contará com o alinhamento da metodologia entre os participantes que são integrantes da equipe do laboratório de engenharia de software (LES) da PUC-Rio, dessa forma, todos estarão nivelados quanto a utilização. A segunda *sprint* será a escolha do processo que será implementado, sendo um dos cenários o projeto de cidades inteligentes do estado do rio de janeiro. A terceira será a conclusão, ou seja, ao final do desenvolvimento o objetivo é concluir que a metodologia é completa, bem definida e utilizável, em projetos qualquer complexidade, por quem deseje desenvolver um sistema inteligente de apoio a decisão.

O objetivo é que ao final dessas 3 *sprints* a metodologia possa se provar como bem precisa e ágil também na construção de um sistema inteligente de apoio a decisão em um estudo de caso real. Assim seu critério de uso terá passo pela última etapa de avaliação garantindo assim que a metodologia é utilizável para todo o tipo de ambientes e por pessoas diferentes do criador.

9 Referências

- [1] W. Wang, W. Chen, K. Zhu, H. Wang (2020) Emphasizing the entrepreneur or the idea? The impact of text content emphasis on investment decisions in crowdfunding – Decision Support Systems Journal
- [2] D. Wiliam (2011) What is assessment for learning? – Decision Support Systems Journal
- [3] H. Koornneef, W. J. C. Verhagen, R. Curran (2020) A decision support framework and prototype for aircraft dispatch – Decision Support Systems Journal
- [4] M. Camargo, M. Dumas, O. González-Rojas (2020) Automated discovery of business process simulation models from event logs – Decision Support Systems Journal
- [5] D. G. Gregg, S. Walczak (2004) Auction Advisor: an agent-based on-line-auction decision support system – Decision Support Systems Journal
- [6] A. Fielder, E. Panaousis, P. Malacaria, C. Hankin, F. Smeraldi (2016) Decision support approaches for cyber security investment – Decision Support Systems Journal
- [7] S. D. Pinson, J. A. Louçã, P. Moraitis (1997) A distributed decision support system for strategic planning – Decision Support Systems Journal
- [8] T. Pourhabibi, K. L. Ong, B. H. Kam, Y. L. Boo (2020) Fraud detection: A systematic literature review of graph-based anomaly detection approaches – Decision Support Systems Journal
- [9] N. R. Jennings, M. Wooldridge (2001) “Agent-Oriented Software Engineering” - in Handbook of Agent Technology (ed. J. Bradshaw) AAAI/MIT Press. (to appear)
- [10] F. Bellifemine, G. Caire, A. Poggi, G. Rimassa (2003) “JADE A White Paper”.
- [11] G. Caire (2003) “JADE TUTORIAL JADE programming for Beginners”.
- [12] R. Choren, C. Lucena (2003) “Modeling Multi-agent systems with ANote” Springer-Verlag 2004.
- [13] R. Choren, C. Lucena (2003) “The ANote Modeling Language for Agent-Oriented Specification”.

- [14] H. O. Almeida, E. Costa, A. Perkusich (2005) "Desenvolvimento de Software para Sistemas Multiagentes" - XXV SBC.
- [15] M. E. Fayad, R. E. Johnson (2000) "Domain-Specific Application Frameworks: Frameworks Experience by Industry" - Wiley.
- [16] Skitmore, Martin and Thorpe, Tony and McCaffer, Ronald and Couzens, Alan (1992) Decision support system for contract bidding (interim report). Technical Report, Department of Surveying, University of Salford
- [17] Page, Lawrence; Brin, Sergey; Motwani, Rajeev and Winograd, Terry (1998), "The PageRank Citation Ranking: Bringing Order to the Web", Technical Report, Stanford University
- [18] M. Wooldridge (2002), "An Introduction to MultiAgent Systems" - Department of Computer Science, University of Liverpool, UK - JOHN WILEY & SONS, LTD
- [19] R. L. McKinnon (1974) A Neww Tripartite Monetary Agreement or a Limping Dollar Standard? - Eassy in International Finance, Princeton University
- [20] C. Grisse (2010) What Drives the Oil-Dollar Correlation? - Federal Reserve Bank of New York
- [21] J. Liao, Y. Shi, X. Xu (2018) Why is the Correlation between Crude Oil Prices and the US Dollar Exchange Rate Time-Varying? - Explanations Based on the Role of Key Mediators - International Journal of Financial Studies
- [22] Y. Luo, D. N. Davis, K. Liu (2002) A Multi-Agente Decision Support System for Stock Trading - IEEE Network Magazin, Special Issue on Enterprise Networking and Services, Vol.16, N° 1
- [23] Instrução Normativa RFB N° 1861, de 27 de Dezembro de 2018, acessado em 27/07/2020 às 01:49 <<http://normas.receita.fazenda.gov.br/sijut2consulta/link.action?visao=anotado&idAto=97727>>
- [24] Aduana e Comércio Exterior, Receita Federal - Ministério da Economia, acessado em 27/07/2020 às 01:55 <<http://receita.economia.gov.br/orientacao/aduaneira/manuais/despacho-de-importacao/topicos-1/importacao-por-conta-e-ordem-e-importacao-por-encomenda-1/importacao-por-encomenda>>
- [25] M. Oliver (2011) Contributo para uma introdução ao estudo dos sistemas homens-máquinas, Open Edition Journals - Editora Universidade do Porto
- [26] *An architectural blueprint for autonomic computing* (2005) - white paper IBM, acessado em 13/12/2020 às 01:05 <<https://www-03.ibm.com/autonomic/pdfs/AC%20Blueprint%20White%20Paper%20V7.pdf>>
- [27] Betty H.C. Cheng, Kerstin I. Eder, Martin Gogolla, Lars Grunske, Marin Litoiu, Hausi A. Muller, Patrizio Pelliccione, Anna Perini, Nauman A. Qureshi, Bernhard Rumpe, Daniel Schneider, Frank-Trollmann, and Norha M. Villegas (2014) *Using Models at Runtime to Address Assurance for Self-Adaptive Systems*, Springer Publisher.
- [28] Modelo ultra estável de Ashby, <<http://sisdinf.blogspot.com/2007/04/retro-alimentao-feedback.html>>, acessado em 06/12/2020 às 23:43
- [29] S. Franklin (2002) "Conscious" software: a computational view of mind (2002) - ACM Soft computing agents: new trends for designing autonomous systems pages 1-45
- [30] A survey of autonomic computing methods in digital service ecosystems (2016) - Dhaminda B. Abeywickrama1 · Eila Ovaska1- Springerlink.com
- [31] SOTA: Towards a General Model for Self-Adaptive Systems (2012) - Dhaminda B. Abeywickrama, Nicola Biccocchi, Franco Zambonelli - (WETICE'12), 25-27 June, Toulouse, France. IEEE, pp 48-53
- [32] Adopting Autonomic Computing Capabilities in Existing Large-Scale Systems - An Industrial Experience Report - Heng Li, Tse-Hsun (Peter) Chen, Ahmed E. Hassan, Mohamed Nasser, Parminder Flora
- [33] SCRUM Guide(2013) Guide to the Scrum Body of Knowledge(2013) SBOK GUIDE Ed
- [34] Essential Kanban Condensed(2016) Anderson, J. D.; Carmichael A., Lean Kanban University
- [35] CBL Guide(2011)
<https://images.apple.com/education/docs/CBL_Classroom_Guide_Jan_2011.pdf> acessado em 01 de nov. de 2020 às 22:30h
- [36] Pmbok guide seventh edition <<https://pmisp.org.br/pmbok-guide/>> acessado em 10 de nov. de 2021 às 00:47h
- [37] Lopes, José M. C.; Gama, Marilza Comércio Exterior Competitivo 4ª Ed São Paulo: Aduaneiras, 2013
- [38] Importação direta ou importação por conta própria
<[https://legislacao.fazenda.sp.gov.br/Paginas/RC24386_2021.aspx#:~:text=Importa%C3%A7%C3%A3o%20por%20encomenda%20\(article%203%C2%BA,para%20revenda%20a%20encomendante%20predeterminado](https://legislacao.fazenda.sp.gov.br/Paginas/RC24386_2021.aspx#:~:text=Importa%C3%A7%C3%A3o%20por%20encomenda%20(article%203%C2%BA,para%20revenda%20a%20encomendante%20predeterminado)> acessado em 16 de agosto de 2022 as 23:22h

- [39] CERVANTES, Caravantes, R; PANNO, Cláudia C., KLOECKNER, Mônica C. Administração: teoria e processos. São Paulo: Prentice Hall Brasil, 2005.
- [40] ROBBINS, Stephen Paul. Fundamentos do Comportamento Organizacional. São Paulo: Prentice Hall, 2004.