

UNIVERSIDADE ESTADUAL DO MARANHÃO  
CENTRO DE CIÊNCIAS TECNOLÓGICAS  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DA COMPUTAÇÃO E  
SISTEMAS

FERRAMENTA DE AUTORIA PARA ATIVIDADES DE APRENDIZAGEM  
ADAPTATIVAS COM USO DE OBJETOS DE APRENDIZAGEM

**ALAN CLOVES SILVA BARRETO**

São Luís-MA

2017

FERRAMENTA DE AUTORIA PARA ATIVIDADES DE APRENDIZAGEM  
ADAPTATIVAS COM USO DE OBJETOS DE APRENDIZAGEM

**ALAN CLOVES SILVA BARRETO**

Dissertação apresentada junto ao Programa de Pós-Graduação em Engenharia da Computação e Sistemas da Universidade Estadual do Maranhão – UEMA, para a obtenção do grau de Mestrado em Engenharia da Computação e Sistemas.

**Orientadora:** Prof<sup>a</sup>. Dr<sup>a</sup>. Eveline de Jesus Viana Sá

São Luís-MA

2017

Barreto, Alan Cloves Silva.

Ferramenta de autoria para atividades de aprendizagem adaptativas com uso de objetos de aprendizagem / Alan Cloves Silva Barreto. – São Luís, 2017.

86 f.

Dissertação (Mestrado) – Programa de Pós-Graduação em Engenharia da Computação e Sistemas, Universidade Estadual do Maranhão, 2017.

Orientador: Profa. Dra. Eveline de Jesus Viana Sá.

1. ACO. 2. Ferramenta de autoria. 3. Objetos de aprendizagem. 4. Estilo de aprendizagem. I. Título.

CDU 004:37.018.43

**ALAN CLOVES SILVA BARRETO**

**FERRAMENTA DE AUTORIA PARA ATIVIDADES DE APRENDIZAGEM  
ADAPTATIVAS COM USO DE OBJETOS DE APRENDIZAGEM**

Dissertação apresentada junto ao Programa de Pós-Graduação em Engenharia da Computação e Sistemas da Universidade Estadual do Maranhão – UEMA, para a obtenção do grau de Mestrado em Engenharia da Computação e Sistemas.

**Aprovado em: 14 / 08 / 2017**

**BANCA EXAMINADORA**

---

Profª. Drª Eveline de Jesus Viana Sá (orientadora)

Doutora em Engenharia Elétrica e Computação

Instituto Federal de Educação, Ciência e Tecnologia do Estado do Maranhão

---

Prof. MSc. Antonio F. L. Jacob Junior

Universidade Estadual do Maranhão

---

Profª. Drª. Jeane Silva Ferreira

Instituto Federal de Educação, Ciência e Tecnologia do Estado do Maranhão

## **AGRADECIMENTOS**

Agradeço primeiramente à Deus, por todas as coisas que provém dele, por mais essa conquista na minha carreira acadêmica.

Agradeço também a minha família, pelo apoio incondicional, me motivando todos os dias para concluir este trabalho, me acompanhando noite e dia, que não foi fácil essa caminhada até aqui.

Agradeço aos professores do Mestrado, em especial a minha orientadora, Eveline Sá, pela paciência, confiança e pelos ensinamentos durante a carreira acadêmica, bem como fora dela também.

Agradeço aos meus amigos que apoiaram e também fazem parte dessa etapa.

Agradeço a todos que, direta e indiretamente, contribuíram para a realização deste trabalho.

Agradeço à FAPEMA pelo apoio financeiro desta pesquisa.

## RESUMO

No mundo da tecnologia da informação, é possível notar diversos de sistemas (e/ou aplicações) voltados para a atividade pedagógica, com o intuito de auxiliar o processo de ensino-aprendizagem, como podemos citar os jogos educativos, os AVAs (Ambientes Virtuais de Aprendizagem), ou simplesmente, uma apresentação de slides no ambiente de sala de aula.

No decorrer das pesquisas de ferramentas para aplicação no que tange ao processo educacional, é possível notar as evoluções, como no caso do conceito de Objetos de Aprendizagem.

Nesse campo há certas dificuldades, e em virtude disso, foram realizados estudos a fim de identificar abordagens, vistos na literatura, que minimizassem essa dificuldades, assim sendo, a aplicação de algoritmos para adaptações de formas de materiais de acordo com o perfil definido do aprendiz, tem se mostrado uma alternativa. Dentre essas abordagens, podemos citar a utilização de Algoritmos Evolucionários, como o ACO (*Ant Colony Optimization*). E partindo dessas premissas, nesta pesquisa foi adotado algoritmo evolucionário ACO, para fins de adaptação de objetos (mídias) de conteúdos de acordo com o Estilo de Aprendizagem do aprendiz, integrado ao processo de pré-autoria pré-3AJC, como proposta para a prototipação de ferramenta de autoria de atividades de aprendizagem adaptativas.

**Palavras-Chave:** ACO. Ferramenta de Autoria. Objetos de Aprendizagem. Estilo de Aprendizagem

## **ABSTRACT**

In the world of information technology, it is possible to notice several systems (and / or applications) aimed at pedagogical activity, with the purpose of assisting the teaching-learning process, such as educational games, LMSs (Learning Management System), or simply a slide show in the classroom environment.

In the course of researching tools for application regarding the educational process, it is possible to notice the evolutions, as in the case of the concept of Learning Objects.

In this field there are certain difficulties, and because of this, studies were carried out in order to identify approaches, seen in the literature, that minimize such difficulties, thus, the application of algorithms for adapting forms of materials according to the defined profile of the apprentice, has been shown to be an alternative. Among these approaches, we can cite the use of Evolutionary Algorithms, such as the ACO (Ant Colony Optimization). Based on these premises, in this research was adopted an evolutionary algorithm ACO, for purposes of adaptation of content objects (media) according to the Apprentice Learning Style, integrated to the pre-3AJC pre-authorship process, as a proposal for prototyping of adaptive learning activities.

**Keywords:** ACO. Authorship tool. Learning Objects. Learning Style

## FIGURAS

|  |    |
|--|----|
| Figura 1 - Fases da Etapa de Desenvolvimento.....  | 17 |
| Figura 2 - Estrutura da AA-Jogo .....  | 18 |
| Figura 3 - Módulos do processo pré-3AJC.....   | 19 |
| Figura 4 - Estrutura da UoL (Unit of Learning) .....   | 22 |
| Figura 5 - Categorias do domínio cognitivo conhecido como Taxonomia de Bloom.....                        | 26 |
| Figura 6 - Fluxograma de um Algoritmo Evolucionário Genérico .....                                       | 27 |
| Figura 7 - Pseudocódigo do ACO .....   | 30 |
| Figura 8 - Esquema da Proposta de Pesquisa .....   | 42 |
| Figura 9 - Funcionamento do <i>AntStudy</i> .....  | 43 |
| Figura 10 - Módulos do Processo de pré-autoria com o novo módulo de adaptação.....                       | 48 |
| Figura 11 - Módulo de Adaptação.....   | 49 |
| Figura 12 - Diagrama de casos de uso da Ferramenta de Autoria visão Professor e Aprendiz                 | 50 |
| Figura 13 - Processo de adaptação integrado ao processo pré-3AJC.....                                    | 51 |
| Figura 14 - Diagrama de Classes.....   | 52 |
| Figura 15 - Modelagem Banco de Dados .....   | 53 |
| Figura 16 - Representação de exemplo de busca dos objetos pelo ACO adaptado em cima do grafo criado..... | 54 |
| Figura 17 - Tela inicial para o Professor .....  | 72 |
| Figura 18 - Tela apresentando o módulo de Cadastro de Alunos.....  | 72 |
| Figura 19 - Tela para Criar Roteiro de Aula.....   | 73 |
| Figura 20 - Tela para Cadastro de Assuntos.....  | 74 |
| Figura 21 - Tela do caso de Uso de Cadastrar Objetos de Aprendizagem .....                               | 74 |
| Figura 22 - Tela Inicial para o Aprendiz .....   | 75 |
| Figura 23 - Tela para Acessar Aula .....   | 75 |
| Figura 24 - Tela da segunda parte do caso de Uso Acessar Aula.....                                       | 76 |
| Figura 25 - Classe Aresta .....  | 77 |
| Figura 26 - Classe Grafo.....  | 78 |
| Figura 27 - Classe Formiga .....   | 79 |
| Figura 28 - Classe ACO .....   | 80 |
| Figura 29 - Trecho aplicado para definição do caminho a ser percorrido.....                              | 81 |
| Figura 30 - Método para executar o algoritmo .....   | 82 |

|   |    |
|---|----|
| Figura 31 - Exemplo de entrada de dados, feito manualmente para testar o funcionamento lógico do algoritmo..... | 84 |
| Figura 32 - Etapa de adição das arestas .....   | 85 |
| Figura 33 - Instanciação da Classe ACO.....   | 86 |

## LISTA DE SIGLAS

AA – Atividade de Aprendizagem  
ACO – Ant Colony Optimization (Otimização por Colônia de Formigas)  
AEs – Algoritmos Evolucionários  
AG – Algoritmo Genético  
APP – Application (Aplicação)  
AS - *Ant System*  
AVA – Ambiente Virtual de Aprendizagem  
CEA – Combinação Estilo Aprendizagem  
EA – Estilo de Aprendizagem  
EML – *Education Modeling Language*  
FA – Ferramenta de Autoria  
HTML – *HyperText Markup Language*  
IEA – Invetário de Estilo de Aprendizagem  
IMs – Inteligências Múltiplas  
IMS-LD – IMS Learning Design  
LMS – Learning Management System  
LOM – *Learning Object Metadata*  
ME – Modelo do Estudante  
MU – Modelo de Usuário  
NC – Nível de Conhecimento  
OA – Objeto de Aprendizagem  
PPT – Apresentação do PowerPoint  
Pré-3AJC – Pré-Autoria de Atividades de Aprendizagem com Jogos e Cooperação  
PSO – Particle Swarm Optimization  
SH - Sistemas Hipermídias  
SHA - Sistemas Hipermídias Adaptativos  
STI - Sistemas de Tutoria Inteligentes  
UoL – *Unit of Learning*  
W3C - World Wide Web Consortium

## SUMÁRIO

|   |           |
|---|-----------|
| <b>1 INTRODUÇÃO.....</b>  | <b>11</b> |
| <b>1.1 Justificativa .....</b>  | <b>12</b> |
| <b>1.2 Objetivos.....</b>   | <b>12</b> |
| 1.2.1 Específicos .....   | 13        |
| <b>1.3 Metodologia.....</b>   | <b>13</b> |
| <b>2 FUNDAMENTAÇÃO TEÓRICA .....</b>  | <b>14</b> |
| <b>2.1 Ferramentas de Autoria.....</b>  | <b>14</b> |
| <b>2.2 Processo de Pré-Autoria pré-3AJC .....</b>   | <b>15</b> |
| <b>2.3 Padrões de Aprendizagem (Learning Design/IMS-LD).....</b>  | <b>21</b> |
| 2.3.1 Learning Design.....  | 21        |
| <b>2.4 Sistemas hipermédia adaptativos.....</b>   | <b>24</b> |
| <b>2.5 Algoritmos Evolucionários .....</b>  | <b>27</b> |
| 2.5.1 PSO ( <i>Particle Swarm Optimization</i> ).....   | 28        |
| 2.5.2 ACO ( <i>Ant Colony Optimization</i> ).....   | 29        |
| <b>3 TRABALHOS RELACIONADOS/CORRELATOS.....</b>   | <b>34</b> |
| <b>3.1 Detecção e Correção Automática de Estilos de Aprendizagem em Sistemas Adaptativos para Educação .....</b>  | <b>34</b> |
| <b>3.2 Em direção à recuperação automática de objetos de aprendizagem em repositórios através da associação dos estilos de aprendizagem de estudantes com metadados no padrão IEEE-LOM.....</b> | <b>36</b> |
| <b>3.3 Uma proposta de um modelo computacional que usa PSO para a escolha de Objetos de Aprendizagem baseado na Espiral de Kolb e nas Inteligências Múltiplas .....</b>                         | <b>39</b> |
| <b>3.4 <i>AntStudy</i>: Proposta de definição de sequências de estudo utilizando <i>Ant System</i> .....</b>  | <b>42</b> |
| <b>3.5 Modelo de Apresentação Adaptativa de Objeto de Aprendizagem baseada em Estilos de Aprendizagem .....</b>   | <b>44</b> |
| <b>4 FERRAMENTA DE AUTORIA PARA ATIVIDADES DE APRENDIZAGEM ADAPTATIVAS COM USO DE OBJETOS DE APRENDIZAGEM .....</b>   | <b>47</b> |
| <b>4.1 Integrando Adaptação ao Processo pré-3AJC.....</b>   | <b>47</b> |
| 4.1.1 Módulo de Adaptação .....   | 47        |
| <b>4.2 Modelagem da Ferramenta de Autoria .....</b>   | <b>49</b> |
| <b>4.3 O Algoritmo ACO Adaptado.....</b>  | <b>53</b> |
| <b>4.4 Prototipação da Ferramenta de autoria .....</b>  | <b>58</b> |
| <b>4.5 Avaliação da Ferramenta de Autoria .....</b>   | <b>59</b> |
| <b>5 CONSIDERAÇÕES FINAIS .....</b>   | <b>62</b> |

|                          |           |
|--------------------------|-----------|
| <b>REFERÊNCIAS .....</b> | <b>64</b> |
| <b>APÊNDICES .....</b>   | <b>71</b> |

## 1 INTRODUÇÃO

No mundo da tecnologia da informação, é possível notar uma grande gama de sistemas (e/ou aplicações) voltados para a atividade pedagógica, com o intuito de auxiliar o processo de ensino-aprendizagem, como podemos citar os jogos educativos, os AVAs (Ambientes Virtuais de Aprendizagem), ou simplesmente, uma apresentação de slides no ambiente de sala de aula.

No decorrer das pesquisas de ferramentas para aplicação no que tange ao processo educacional, é possível notar as evoluções, como no caso do conceito de Objetos de Aprendizagem, materiais disponíveis e que permite sua reutilização, os Metadados, como base da definição de roteiros do processo, e possibilitando, também, a reutilização desses objetos.

Tem-se também as ferramentas de autoria, onde permite a produção de materiais de diversos formatos, a fim de que os mesmos sejam utilizados e posteriormente reutilizados, podendo ser definidos como Objetos de Aprendizagem. Estes produzidos pelo professor/educador, sendo disponibilizados aos aprendizes.

O processo de pré-autoria, pré-3AJC (SÁ, 2011), é baseado no padrão IMS-LD, para desenvolvimento de Atividades de Aprendizagem cooperativas ou adaptativas, a partir do uso das Melhores Práticas do IMS-LD para representar aspectos pedagógicos, onde propõe um modelo de documentação ativa de *framework* formado pelos seguintes componentes, a saber: Estrutura, Processo e Ferramenta de Apoio. Esses componentes possuem interação entre si trocando informações em um ambiente de desenvolvimento apoiado por uma ferramenta.

No decorrer da evolução destas ferramentas e abordagens, foi notada uma melhora no que se refere a gestão do processo pedagógico, como pode ser visto em Dorça et al. (2011). Porém, ainda há dificuldade em lidar com as particularidades de perfil de cada aprendiz com Estilo de Aprendizagem distintos.

Em virtude dessa dificuldade, foram realizados estudos a fim de identificar técnicas que minimizasse essa dificuldade de lidar com os Estilos de Aprendizagem heterogêneos. Portanto, a aplicação de algoritmos para adaptações de formas de materiais de acordo com o perfil definido do Aprendiz, tem se mostrado uma alternativa. Dentre essas técnicas, podemos citar a Taxonomia de Bloom (DORÇA et al., 2011) como técnica pedagógica, bem como a utilização de Algoritmos Evolucionários, como PSO (*Particle Swarm Optimization*) (MOURA E FERNANDES, 2012); o algoritmo ACO (*Ant Colony Optimization*) utilizado para definir as sequências de estudo em um ambiente virtual de aprendizagem, tendo como entradas informações do objeto de aprendizagem definidas pelo padrão LOM e modelos de usuários (Araújo e Fagundes, 2011), entre outras formas, como adaptações das abordagens já existentes.

Neste contexto, este trabalho propõe o desenvolvimento de uma aplicação/sistema baseado no processo de pré-autoria citado, com utilização de uma técnica adaptativa a ser integrada na autoria de objetos de aprendizagem, tendo como resultado a autoria de atividades de aprendizagem adaptadas ao perfil de cada aprendiz. Importante ressaltar que, apesar do processo pré-3AJC ser um processo de pré-autoria que provê atividades que usam jogos e cooperação, será considerado apenas objetos de aprendizagem que não sejam jogos nessa primeira etapa. Uma vez que os jogos possuem características específicas, e que estas não serão contempladas no escopo dessa pesquisa.

Este documento está organizado da seguinte forma: Capítulo 1, contém Introdução, Justificativa, Objetivos e Metodologia da pesquisa; Capítulo 2, contém a Fundamentação Teórica como referência da Literatura para embasamento da pesquisa; Capítulo 3, possui os Trabalhos Relacionados ao contexto da pesquisa; Capítulo 4, a Ferramenta de autoria proposta, suas características, modelagens e prototipação, por fim, Capítulo 5, possui as Considerações Finais em relação à pesquisa.

## **1.1 Justificativa**

Ao analisar o processo de pré-autoria pré-3AJC, é possível notar uma limitação, na qual não foi explorada a capacidade para adaptações no decorrer da atividade de aprendizagem, ou seja, não há consideração dos perfis dos aprendizes na execução da atividade.

A necessidade de ter que considerar os diferentes tipos de perfis de aprendizes, adotando medidas pedagógicas para cada um deles, tem sido um grande desafio para os professores/educadores, afinal, cada aprendiz apresenta um perfil distinto, dificultando o processo de ensino-aprendizagem. Devido a isso, tem-se como proposta aprimorá-lo nesse aspecto, a fim de adicionar essa característica ao pré-3AJC, através de técnicas que provém adaptações, mais especificamente a utilização de algoritmo inteligente, nesse caso escolhido o algoritmo ACO, e que através disso, seja possível maximizar a transmissão de conhecimento e o nivelamento do conhecimento entre os aprendizes.

## **1.2 Objetivos**

O objetivo geral é o desenvolvimento da ferramenta de autoria para prover atividades de aprendizagem adaptáveis segundo o processo de pré-autoria Pré-3AJC proposto em Sá(2011), a fim de que se insira capacidade de adaptação de conteúdo de acordo com o perfil Aprendiz.

### 1.2.1 Específicos

- Definir modelo de adaptação para inserir capacidade de adaptação ao processo pré-3AJC;
- Utilizar de algoritmos evolucionários, no caso o ACO (*Ant Colony Optimization*), para adaptar os objetos de aprendizagem (conteúdos) disponibilizado ao aprendiz;
- Integrar a função de adaptação de conteúdo no processo de pré- autoria pré-3AJC.

### 1.3 Metodologia

Para a realização deste projeto será adotada a pesquisa documental e bibliográfica para o embasamento de conceitos sobre métodos adaptativos, modo experimental com o desenvolvimento do sistema baseado em um modelo de pré-autoria adicionando novo recurso. Após esta reunião de informações, estes serão modelados e aplicados com o intuito de ter um novo recurso para o sistema a ser implementado.

Segue as etapas utilizadas pela metodologia de desenvolvimento da proposta:

- Estudo dos conceitos de Ferramentas de Autoria, Processo pré-3AJC, Padrões de Aprendizagem, Sistemas Adaptativos;
- Especificar o perfil do aprendiz e suas particularidades para definição de uma base de dados;
- Estudo das técnicas de aplicação em adaptação de conteúdo para auxiliar no processo de Atividade de Aprendizagem adicionando a capacidade de adaptação de acordo com o perfil do aprendiz;
- Modelagem dos requisitos para adaptabilidade;
- Prototipação do processo de pré-autoria pré-3AJC com a inclusão da funcionalidade de adaptação.

## 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo será abordada a fundamentação teórica para o embasamento da produção científica.

### 2.1 Ferramentas de Autoria

Segundo o W3C (2015), ferramenta de autoria é qualquer aplicativo, parte de um aplicativo ou coleção de aplicativos com os quais o autor interage para produzir, alterar ou montar conteúdo Web, que será utilizado por outras pessoas.

Segundo Nôleto (2014), as ferramentas de autoria são focadas em usuários que nem sempre necessitam possuir conhecimentos aprofundados de programação. Visa atingir um público mais leigo para auxiliar no processo de produção de conteúdo/material, em sua maioria didáticos.

Para a produção de objetos de aprendizagem, é necessário a utilização de uma ferramenta de autoria. De acordo com Leffa (2006), se refere a um tipo de software que objetiva a produção de objetos educacionais, seja eles: textuais, visuais, auditivos, etc. Sua disponibilidade é de forma *online* ou *offline*, sendo seu público alvo aprendizes e professores.

Silva (2013) também afirma que, a ferramenta (ou software) de autoria, é um aplicativo equipado com diversas ferramentas de multimídia que, a partir de modelos pré-definidos, possibilita ao usuário inserir informações, imagens e sons, promovendo a autonomia e estimulando o desenvolvimento cognitivo de quem cria atividades a partir dele.

Segundo Nolêto, Viana e Trinta (2014), tem como exemplo de ferramenta de autoria é do APP Inventor do Google, onde afirma que a mesma possui uma interface amigável e de fácil usabilidade. Esse APP permite aos usuários a criação de suas aplicações móveis com pouco ou nenhum conhecimento de programação.

Flores (2011) afirma que, as ferramentas de autoria facilitam a atuação do professor, pois viabiliza a criação de material educacional digital sem a necessidade que o professor seja um programador, usando estruturas e procedimentos já programados, reunindo-os, agregando conteúdo e forma de tratamento aos dados que dependem de sua estratégia pedagógica.

Para Silveira (1999), as Ferramentas de Autoria incluem funcionalidades a fim de criar, editar e importar tipos específicos de dados, juntar dados brutos em uma sequência de reprodução, e fornece um método ou linguagem estruturada para responder às entradas do usuário.

Ao selecionar a ferramenta de autoria mais adequada para criar um OA específico, o professor deve avaliar suas necessidades, criar uma lista de recursos possíveis, determinar a funcionalidade mais importante para o seu objeto e analisar qual ferramenta dispõe dessa funcionalidade (Flores, 2011).

Conta também com recurso de espaço para compartilhamento de conteúdo, como afirma Goudouris (2013), elas podem oferecer espaços para a disseminação de conteúdos e informações; interações dos aprendizes entre si e/ou com seus professores; ambientes que facilitam o processo de ensino-aprendizagem, podendo portar diferentes estilos e objetivos de aprendizagem; estimular a busca ativa e a partilha de conhecimento; e recursos de gerência do processo educativo. Sendo um ambiente de ensino, mediado por computador, até mesmo podendo ser concebido pela combinação de ferramentas distintas.

Nienow e Bez (2009) e Leffa(2006) afirmam que, algumas vantagens desse tipo de software deve-se citar o alto nível de interação que o usuário pode vir a ter com o objeto de aprendizagem – facilitando o aprendizado –, o baixo custo de geração do material, a sua capacidade de ofertar uma maneira fácil de criar conteúdos educacionais e a liberdade do professor de poder criar e gerenciar o conteúdo.

De acordo com Silva Filho (2000), no processo de desenvolvimento de uma ferramenta ou sistema de autoria, os critérios a serem observados ou adotados são os seguintes:

- a) Facilidade de uso;
- b) Funcionalidade da barra de ferramentas;
- c) Habilidade de adicionar objetos na biblioteca;
- d) Recursos multimídia;
- e) Ajuda ao usuário;
- f) Controle dos objetos;
- g) Interfaces.

## **2.2 Processo de Pré-Autoria pré-3AJC**

É comum notar a interação de pessoas com alguma atividade utilizando jogos, seja por lazer, bem como jogos (sérios) educativos, onde uma pessoa acaba reservando um tempo para um jogo casual, por exemplo.

No âmbito educacional, os Ambientes Virtuais de Aprendizagem (AVAs) oferecem aos aprendizes aplicações educacionais de vários tipos. Aos professores, os AVAs oferecem ferramentas para o desenvolvimento de tais aplicações, a fim de que os objetivos instrucionais traçados possam ser alcançados pelos aprendizes. (SÁ, 2011).

Visando à definição de um modelo de atividade de aprendizagem com uso de jogos e cooperação, que não pode ser realizada sem uma orientação e parametrização de todos os aspectos relevantes; a formatação do processo cooperativo da atividade de aprendizagem; e, especificação do uso de jogos como fatores instrucionais dentro de etapas de uma atividade de aprendizagem, a fim de possibilitar a utilização de diferentes jogos por diferentes aprendizes em diferentes etapas de uma atividade de aprendizagem.

O professor poderá modelar uma AA-Jogo<sup>1</sup> formada por subatividades que possibilitam preparar o aprendiz, inferir informações do aprendiz, aquecer, assimilar e avaliar o aprendiz usando o jogo como recurso pedagógico, bem como discutir os resultados dessas subatividades, a fim de verificar o aprendizado alcançado.

São definidas 3 fases para a estrutura de uma AA-Jogo (Etapa de Desenvolvimento Pré-3AJC), a saber, Preliminar, Intermediária e Final, cada uma das fases é constituída por subatividades. Cada uma dessas fases é descrita a seguir:

A Fase Preliminar é a fase inicial da atividade de aprendizagem (AA). É considerada uma fase opcional, pois a remoção da mesma não compromete a AA propriamente dita; porém, sua inclusão pode proporcionar um melhor rendimento na aprendizagem. Isso ocorre devido as subatividades relacionadas a esta fase podem servir como feedback ao professor com informações dos aprendizes, bem como para estimular os aprendizes no início da realização da AA. Essas subatividades podem ser chamadas de Atividades de Aquecimento (AAq) e podem ser submetidas aos aprendizes de forma coletiva ou individual. O objetivo das Atividades de Aquecimento é estimular os aprendizes com atividades de desafio e entretenimento relacionados ao conteúdo que será abordado na AA, para que se sintam confortáveis com o mesmo.

A Fase Intermediária é a fase onde se encontra a AA propriamente dita. Nela são trabalhadas atividades que desenvolvem competências e habilidades nos aprendizes, relacionadas à aquisição de conhecimento sobre o conteúdo tratado pela AA. As competências e habilidades também podem ser trabalhadas nas outras duas fases, bastando apenas que estejam relacionadas com o escopo de cada fase.

Entende-se por competências um conjunto de conhecimentos, atitudes, capacidades e aptidões que habilitam o aprendiz em um dado contexto. As habilidades são ações relacionadas ao saber fazer, saber conviver e saber ser. As competências especificam a capacidade para usar

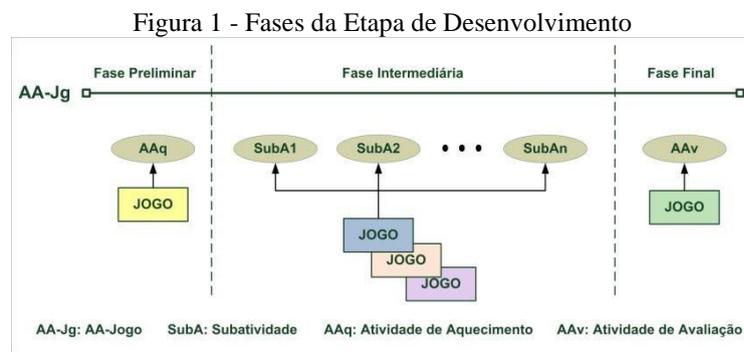
---

<sup>1</sup> Atividade de Aprendizagem com Jogo

as habilidades na realização de tarefas e aquisição de conhecimentos. As habilidades devem ser desenvolvidas para atingir as competências, uma vez que estão associadas à ação física ou mental que indica a capacidade adquirida. Como exemplo de habilidades, tem-se: identificar variáveis, relacionar informações, analisar situações-problema, sintetizar, correlacionar e manipular, etc. As competências e habilidades definidas são baseadas nas diretrizes curriculares orientadas pelo MEC (Pareceres CNE/CES nº 776/1997, CNE/CES nº 436/2001, CNE/CP nº 29/2002).

A Fase Final é a fase onde são aplicados nos aprendizes subatividades para inferir o conhecimento do aprendiz referente ao conteúdo trabalhado, bem como, avaliar se as habilidades e competências foram alcançadas. Vale ressaltar, que a avaliação é processual, pode ocorrer a qualquer momento no processo de aprendizagem, e não necessariamente apenas no final.

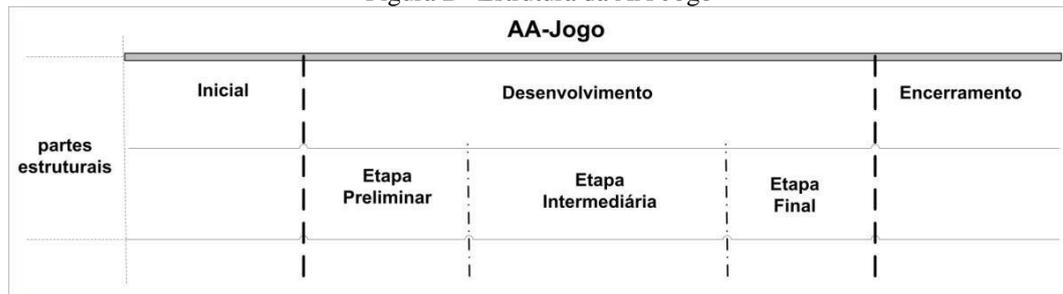
As subatividades previstas em cada uma dessas fases podem elencar vários recursos e serviços de aprendizagem. O jogo pode ser utilizado como um desses recursos como ilustrado na Figura 1.



Fonte: Sá et al. (2007)

Observa-se que o jogo pode ser utilizado como uma atividade de aquecimento (AAq) para estimular o aprendiz no assunto que será tratado (Sá et al., 2007) na fase Preliminar. Pode ser usado na fase Intermediária como um recurso motivador e interativo, bem como, pode ser usado para testes e avaliações, na fase Final. Este comportamento define uma AA-Jogo padrão, que possui no mínimo três momentos de aplicação do jogo definidos pelo professor, um em cada fase da AA, sendo que estes jogos podem ser usados de forma individual ou cooperativa.

Figura 2 - Estrutura da AA-Jogo



Fonte: Sá, 2011.

Verificando a Figura 2, uma AA-Jogo é formada pela sequência das 5 partes: Inicial, Desenvolvimento/Preliminar, Desenvolvimento/Intermediário, Desenvolvimento/Final e Encerramento. No entanto, as etapas que compõem a parte de Desenvolvimento podem ser utilizadas pelo professor sequencialmente ou de forma independente, ou seja:

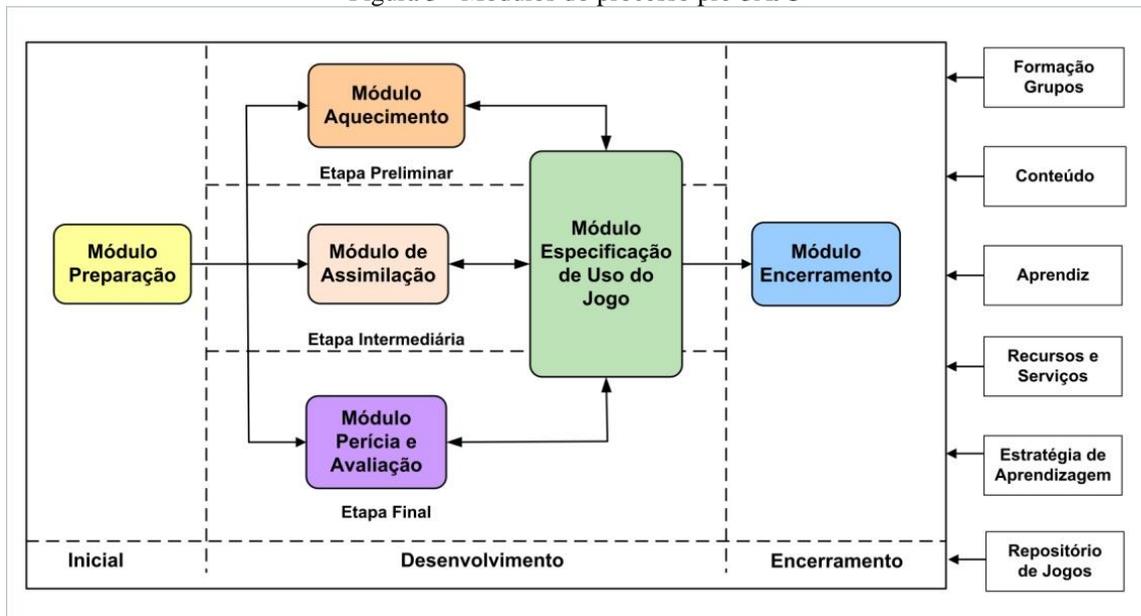
- A etapa preliminar pode ser utilizada apenas para preparar o aprendiz em um dado assunto, para uma próxima atividade a ser aplicada em outro momento.
- A etapa intermediária pode ser utilizada sem a necessidade de que o aprendiz tenha sido estimulado anteriormente ou sem a meta de avaliar ao final.
- A etapa final pode ser utilizada em qualquer momento onde o professor necessite efetuar uma avaliação ou diagnóstico dos aprendizes.

Vale ressaltar que a independência da parte de Desenvolvimento/Final, predispõe a aplicação do processo avaliativo, não apenas no final completo de uma atividade, podendo este ser utilizado em diferentes momentos da AA, possibilitando estimar o desempenho do aprendiz em todo o processo de aprendizagem.

A estrutura da AA-Jogo foi utilizada como base para definição dos jogos que serão recuperados pelo mecanismos de busca e recuperação de jogos proposto por Teixeira et. al. (2013).

A Figura 3, apresenta os módulos que constituem o Processo pré-3AJC. Nela é apresentada cada etapa e fase, e sua relação com cada módulo presente no modelo proposto por Sá (2011).

Figura 3 - Módulos do processo pré-3AJC



Fonte: Sá (2011).

A seguir serão descritos os Módulos do Processo pré-3AJC, a saber:

Fase Inicial - Módulo de Preparação:

- Monta-se a estrutura inicial da AA-Jogo, na qual são definidos: a estratégia de ensino da AA, o conteúdo a utilizar, o objetivo instrucional principal da AA e a competência a ser desenvolvida. Apresenta estas informações aos aprendizes que comporão a meta de aprendizagem, bem como os professores e monitores envolvidos, a lista de aprendizes inscritos na AA-Jogo e a metodologia que será utilizada para a realização da mesma. Estas informações permitem a familiarização do aprendiz com o assunto abordado, bem como com o procedimento utilizado para a aprendizagem. O resultado deste módulo é um roteiro instrucional da fase inicial.

Fase de Desenvolvimento:

a) Etapa Preliminar - Módulo de Aquecimento

- Utilizar jogos (OA) para estimular o aprendiz como se o jogo fosse uma atividade de aquecimento (Sá et al., 2007a), provendo um roteiro instrucional da etapa preliminar.

b) Etapa Intermediária – Módulo de Assimilação

- Possibilita o planejamento da AA a partir das condições de aprendizagem e das informações provenientes tanto do aprendiz como da AA. Essas informações delimitam e suporta a especificação dos procedimentos didáticos, objetivos instrucionais e suas divisões, competências, habilidades a serem trabalhadas nesta fase, bem como, os recursos, conteúdos, os jogos, tipos de uso dos jogos. Define também, quando utilizado o jogo, a formação de grupos e as regras de cooperação com uso de jogos, quando precisar. O resultado deste módulo é um roteiro instrucional da etapa intermediária.

#### c) Etapa Final – Módulo de Perícia e Avaliação

- Verifica a aquisição das competências e habilidades estabelecidas na fase de planejamento pelos aprendizes com o uso de jogos. Neste ponto, os jogos podem ser aplicados individualmente aos aprendizes para mensurar o grau de aprendizado utilizando-se dos desafios proporcionados pelos jogos como estímulo para o processo de avaliação. O professor tem opção de planejar a utilização de outras estratégias de avaliação. Nesse caso, é permitida a inserção de uma avaliação. O resultado deste módulo é um roteiro instrucional da etapa final.

#### d) Módulo de Especificação de Uso do Jogo

- Este módulo pertence à fase de Desenvolvimento podendo ser acionado em uma das suas etapas. Permite definir os requisitos para a seleção dos jogos a serem utilizados na AA, bem como agregar valor educacional ao jogo e definir o fluxo de aprendizagem caso o jogo seja utilizado como promotor da cooperação.

#### Fase de Encerramento – Módulo de Encerramento:

- Neste módulo realiza-se o planejamento do momento em que os resultados são disponibilizados e socializados aos aprendizes, visando prover argumentações para a eliminação de dúvidas referente aos conteúdos abordados e aos objetivos instrucionais traçados, para posterior discussão entre aprendizes e professores, a fim de analisar o sucesso ou a possível falha no processo de aprendizagem. Estas argumentações são baseadas nos

resultados das auto-avaliações submetidas aos aprendizes e nos resultados alcançados pelos processos anteriores. Tais interações entre professor e aprendizes, previamente planejados, podem ocorrer presencialmente ou via ferramentas de comunicação quando estiverem à distância. O resultado deste módulo é um roteiro instrucional da etapa de encerramento.

Diante deste contexto, na análise do processo pré-3AJC, foi possível notar dentre algumas limitações do Processo pré-3AJC, a ausência da capacidade de adaptação em relação ao perfil do aprendiz, podendo notar que a escolha do jogo, bem como materiais, são definidas pela ação do educador/professor e aprendiz, sem uma certa automatização. Portanto, a adaptabilidade poderia ser uma forma de otimizar o processo da atividade de aprendizagem.

### **2.3 Padrões de Aprendizagem (Learning Design/IMS-LD)**

No Ambiente Educacional, é possível notar o surgimento de diversas ferramentas para o auxílio no trabalho de ensino-aprendizagem do educador. Muitas TICs a disposição, utilização de mídias para facilitar na execução da atividade educativa. Tendo em vista essa realidade, foi necessário definir padrões para um melhor aproveitamento de todo o material produzido, a fim de ter um menor desperdício possível de materiais.

Os ambientes de ensino-aprendizagem via web estão cada vez mais presentes nos meios educacionais. O desenvolvimento de material didático a ser disponibilizado nesses ambientes exige criatividade e conhecimento de tecnologias adequadas. Visando minimizar o tempo e esforço aplicado na produção de material educacional, e visando uma forma de reutilizar esse material, o conceito de Objetos de Aprendizagem (OA) foi concebido.

No decorrer das evoluções de tecnológicas, foram surgindo diversos padrões que auxiliam na produção de objetos de aprendizagem. Estes padrões são utilizados como um conjunto de especificações que orientam o processo de empacotamento.

Para Oliveira, Nelson e Ishitani (2007), a tecnologia utilizada para desenvolvimento de ensino a distância deve utilizar padrões internacionalmente conhecidos e aceitos, para que qualquer pessoa possa ter acesso à informação e para possibilitar com mais facilidade a reutilização.

#### **2.3.1 Learning Design**

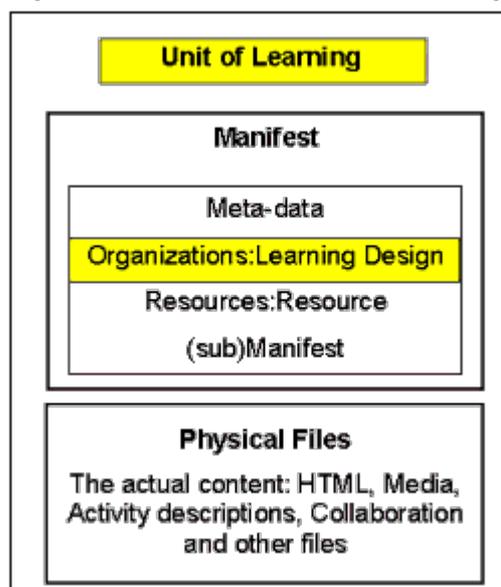
De acordo com Dutra (2005), o Learning Design é uma linguagem de modelagem para definição de objetos e atividades de aprendizagem definida pela IMS com base no EML

(*Educational Modeling Language*). Tem suporte ao uso de uma grande variedade de abordagens de ensino-aprendizagem, tais como: behavioristas, cognitivistas e construtivistas. Isso é possível através de uma linguagem genérica e flexível, arquitetada para abranger diversos tipos de abordagens pedagógicas com a mesma tecnologia.

Toda unidade de aprendizagem produzida a partir de objetos de aprendizagem pode ser guiada por *learning design* que é mais genérico que a própria unidade, onde sua estrutura é independente.

O IMS Learning Design, aqui denominado como IMSLD ou apenas LD, é um padrão que contém uma metalinguagem para construção de unidades de aprendizado (*Units of Learning – UoLs*) independente de modelos pedagógicos. A fim de auxiliar os professores na especificação e desenvolvimento de UoLs, no conteúdo e modelo instrucional desejados, estão disponíveis os editores LD, que são ferramentas de autoria. (Silva, 2013)

Figura 4 - Estrutura da UoL (Unit of Learning)



Fonte: IMS-LD (2003)

Na Figura 4 é demonstrada a estrutura da UoL, onde é composta por dois núcleos: manifesto, com o dados referente: organização, locais de recursos e (sub)manifesto; e, os arquivos a serem utilizados por essa UoL, como PPTs, páginas HTML, etc.

Kolper e Oliver (2004, apud Silva e Barreto, 2009) afirmam que, a sumarização das teorias do aprendizado (ex., construtivismo, aprendizagem colaborativa, baseada em problemas etc.) é feita pelo uso da tríade: papel – atividade – ambiente. O ambiente é representado pela analogia de uma peça de teatro.

Santana e Xavier (2011) afirmam que, o padrão IMS Learning Design torna possível a descrição formal do processo de ensino-aprendizagem, independente da modalidade ou teoria de aprendizado a ser aplicado, por via dos elementos que compõem uma unidade de aprendizagem.

O Learning Design possui 3 níveis de especificação (IMS-LD, 2003), denominados de níveis A, B e C. A saber: (BOARETTO; NUNES; FILATRO, 2006).

- No Nível A é possível representar um sequenciamento simples, pré-determinado, no qual as atividades são apresentadas da mesma maneira para todos os usuários atribuídos a um papel. Ele contém todo o vocabulário de base necessários para apoiar a diversidade pedagógica
- O Nível B possibilita maior customização da aprendizagem e um sequenciamento mais complexo das atividades pela adição de propriedades e condições, através das quais se pode definir, por exemplo, que determinada atividade deve ser feita por determinado papel somente se determinada condição for satisfeita. Além disso, as propriedades permitem armazenar dados de cada usuário (aprendiz ou professor), facilitando o rastreamento e a análise processual da aprendizagem.
- O Nível C agrega um recurso de envio de notificações, adicionando uma flexibilidade ainda maior com relação à personalização da aprendizagem. Aumenta significativamente a capacidade.

Segundo Filatro (2008), um *Learning Design* pode ser reutilizado quantas vezes se quiser, na íntegra, parcialmente ou com ajustes e adaptações. Nesse sentido, o Learning Design é um produto, um resultado bem identificável - mesmo virtual - da representação de um processo. Em última instância, no Learning Design, o produto final que representa determinado processo educacional onde esteja sempre de acordo com conceitos e regras de modelagem determinados, no caso a especificação IMS LD.

Segundo Fernandes (2012), o *Learning Design* possibilita na construção de UoLs que se especifique o seguinte: (a) as atividades de aprendizagem a serem realizadas; num curso seria equivalente, por exemplo, aos módulos de curso ou ainda a aulas específicas; (b) os recursos que serão utilizados, tais como texto, imagens e apresentações; (c) os papéis que realizarão determinada atividade, tais como professor, tutor, mediador e aprendiz; (d) a dinâmica dos participantes das atividades, a saber, quando se pode terminar uma atividade ou uma unidade e,

caso tenha terminado, quando se pode continuar ou esperar por outros membros do grupo envolvido; (e) os serviços externos, tais como e-mail, bate-papo, fórum e outros existentes em AVAs.

Em *Learning Design*, segundo Bruno (2011), os conhecimentos podem ser adquiridos de três formas:

1. Pela visão do Design Instrucional, no qual o conhecimento é modelado por teorias que consistem em um arranjo de princípios de design;
2. Na identificação de exemplos, ou melhores práticas, de processos de ensino-aprendizagem;
3. Através de Padrões de design pedagógico, sendo tais Padrões um intermediário entre teorias e melhores práticas.

#### **2.4 Sistemas hipermídia adaptativos**

No campo da informática, é possível notar a evolução dos sistemas, dentre eles destacam-se os sistemas adaptativos. Tais sistemas têm a capacidade de mudar e aprender com a experiência (DORÇA et al, 2011). Essa capacidade do sistema de adaptação, permite individualizar o ensino de acordo com o perfil de cada aprendiz.

Segundo ITO, FERREIRA e SANT'ANNA (2006), as interfaces adaptativas se mostram como soluções para alguns dos problemas atuais na interação homem-computador. Sendo necessárias interfaces capazes de se ajustar às necessidades do usuário.

Havendo a necessidade de melhorar a satisfação dos usuários, como no caso do ambiente pedagógico entre educador e aprendiz, tem sido uma grande preocupação dos desenvolvedores buscar uma adaptação sendo a melhor possível.

Tendo em vista isso, de acordo com Wu (2002), os sistemas de hipermídia adaptativa funcionam, trivialmente, armazenando um modelo de usuário (MU) onde coleta as características classificadas como relevantes do usuário e, a partir destas, produzindo adaptação de conteúdo ou adaptação de navegação de hipermídia.

Segundo Farias et al. (2014), os Sistemas Hipermídias Adaptativos (SHA), baseados em STI (Sistemas de Tutoria Inteligentes) e SH (Sistemas Hipermídias), elevam a capacidade de adequação ao perfil do aprendiz, visto que seu princípio fundamental é o fato de considerar a heterogeneidade entre as pessoas, onde estes podem aprender em ritmos diferentes, bem como possuir necessidades diferentes de aprendizagem. Sendo dessa forma, um forte aliado à eficiência do processo de ensino-aprendizagem.

Os SHAs podem ser entendidos como sistemas de hipertexto e hiperímia combinados com STIs que, por meio de características do usuário refletidas no modelo de usuário, promovem a adaptação de conteúdos e recursos de hiperímia, vindos de qualquer fonte e apresentados em qualquer formato que seja adequado ao perfil de seus usuários.

Para Cunha, Emer e Amaral (2015), no desenvolvimento de Sistemas Hiperímia Adaptativos é interessante identificar aspectos genéricos existentes e modularizá-los. Isso também conhecido como modelo de referência. Estes modelos facilitam a implementação de um SHA, já que fornecem um framework que expressa as funcionalidades mais comuns em um SHA.

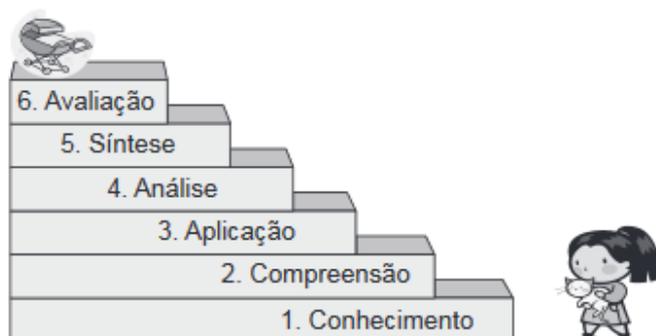
Pode-se citar também, a Taxonomia de Bloom, de acordo com (Marzano e Kendal, 2007; Bloom e Krathwohl, 1956; Starr e Stalvey, 2008, Krathwohl, 2002), descreve objetivos educacionais como uma formulação explícita das mudanças que ocorrem nos estudantes mediante processos educacionais, através de um sistema de classificação de objetivos de aprendizagem e do desenvolvimento de capacidades e habilidades intelectuais. Além disto, permite expressar qualitativamente o nível de conhecimento do estudante em determinado conceito do domínio de forma qualitativa.

De acordo com Ferraz e Belhot (2010), os processos categorizados pela Taxonomia dos Objetivos Cognitivos de Bloom, além de representarem resultados de aprendizagem esperados, são cumulativos, o que deixa caracterizado uma relação de dependência entre os níveis e são organizados em termos de complexidades dos processos mentais.

De acordo com Krathwohl, Bloom et al. (2002;1956 apud OLIVEIRA et. al, 2015) viram a teoria de taxonomia como uma ferramenta que, entre outros pontos:

- Padronizaria a linguagem sobre os objetivos de aprendizagem para facilitar a comunicação entre pessoas (docente, coordenadores etc.), conteúdos, competências e grau de instrução desejado;
- Serviria como base para que em determinados cursos fossem definidos, de forma clara e particular, objetivos e currículos de acordo com as necessidades e diretrizes contextual, regional, federal e individual (perfil do discente/curso);
- Determinaria a congruência dos objetivos educacionais, atividade e avaliação de uma unidade, curso ou currículo; e
- Definiria um panorama para outras oportunidades educacionais (currículos, objetivos e cursos), quando comparado às existentes antes dela ter sido escrita.

Figura 5 - Categorias do domínio cognitivo conhecido como Taxonomia de Bloom.



Fonte: Ferraz e Belhot (2010).

É possível observar na Figura 5, uma analogia da definição da Taxionomia de Bloom, onde o mesmo é uma escada e cada nível do processo seria semelhante a um degrau, chegando a um determinado objetivo que está ao seu topo.

Segundo Bloom et. al. (1977), os níveis da Figura 5, tem as seguintes características:

- **Nível Conhecimento:** são classificados os objetivos que exigem do aprendiz um baixo nível de habilidades e desenvolvimento cognitivo; nesse nível se classificam os objetivos ou comportamentos relacionados à memória e a evocação de ideias, materiais ou fenômenos.
- **Nível Compreensão:** se refere aos objetivos, comportamentos ou respostas que representam um entendimento da mensagem literal contida em uma comunicação.
- **Nível Aplicação:** exige que o aprendiz aplique o conhecimento compreendido em uma nova situação.
- **Nível Análise:** inclui os objetivos e comportamentos nos quais o aprendiz deve saber segregar em partes um conteúdo ou um material aprendido e conseguir perceber as relações destas partes para formar o todo e analisar como determinado material se organiza.
- **Nível Síntese:** inclui os objetivos educacionais que desenvolvem no aprendiz a habilidade de produzir um material empregando o que aprendeu; classifica objetivos relacionados ao “pensamento criador” do estudante.
- **Nível Avaliação:** classifica os objetivos educacionais que exigem uma maior capacidade cognitiva do aprendiz; nele se define a avaliação como o processo de julgamento acerca do valor de ideias, trabalhos, soluções, métodos, materiais, etc. realizados com um determinado propósito.

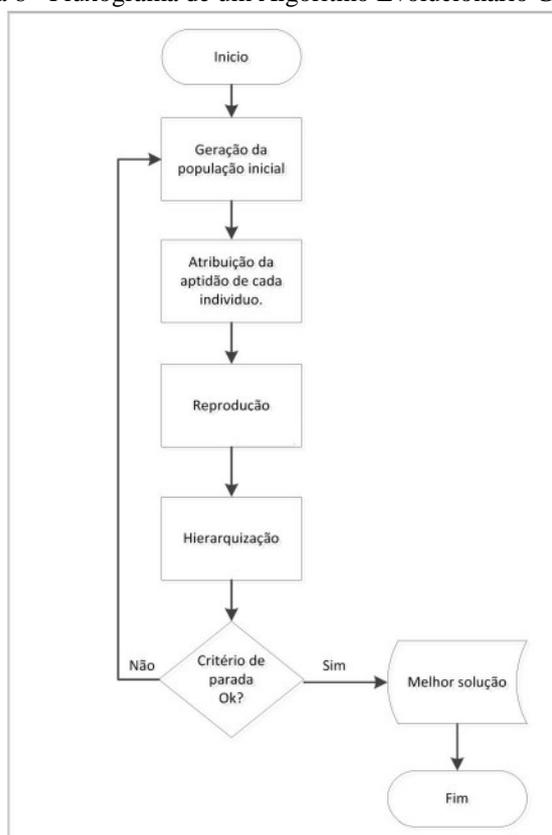
## 2.5 Algoritmos Evolucionários

A Computação Evolucionária é uma abordagem que se inspira em mecanismos evolucionários naturais para desenvolver algoritmos computacionais. No contexto computacional “evolução” é entendida como um método de busca dentro de um espaço de soluções viáveis. (GOLDBARG, GOLDBARG e MEDEIROS NETO, 2005).

Algoritmos Evolutivos (AEs) são meta-heurísticas populacionais de otimização que utilizam mecanismos tendo a Teoria da Evolução como inspiração; seleção natural e sobrevivência de indivíduos mais adaptados, e na Genética, como mutação e recombinação (EIBEN e SMITH, 2003)

A vantagem dos AEs está na possibilidade de modelar um problema pela simples descrição de uma potencial solução do mesmo. Sendo assim, possibilita que AEs possam ser adaptados para uma grande diversidade de problemas complexos. (GABRIEL, MELOY e DELBEM, 2012). Na Figura 6 é demonstrado o fluxograma do funcionamento genérico de um algoritmo Evolucionário.

Figura 6 - Fluxograma de um Algoritmo Evolucionário Genérico



Fonte: Martinez (2014).

Como exemplos de algoritmos evolucionários, podem ser citados os Algoritmos Genéticos (AG) e colônias de insetos sociais (inteligência coletiva), tais como formigas, abelhas

e cupins. (PRATA, 2012). É ilustrado na Figura 6, o fluxograma do algoritmo evolucionário de forma genérica.

### 2.5.1 PSO (*Particle Swarm Optimization*)

Os grupos de animais, ao procurarem por comida ou ao fugirem de predadores, possuem um tipo de consciência coletiva. Cada elemento do grupo aprende com o comportamento dos seus demais membros. Considerando esse comportamento, diversos pesquisadores buscaram simular esses tipos de interação por intermédio de computadores. (PRATA, 2012).

Kennedy e Eberhart (1995a) propuseram uma nova meta-heurística, destinada à resolução de problemas irrestritos de otimização não-linear, denominada *Particle Swarm Optimization* (PSO). Inspirado no comportamento de aves ou peixes, é um algoritmo pertencente à classe de algoritmos denominada “inteligência de enxame (coletiva)” (BARBOSA, 2012)

PSO é uma metaheurística com base no comportamento social e comunicação de um bando de pássaros ou cardumes de peixes (KENNEDY & EBERHART, 1995). PSO pode ser considerado como um algoritmo evolutivo, pois explora o espaço de soluções através de uma população (enxame) e da estrutura de vizinhança de cada indivíduo (partículas) e aproveita a informação geracional adquirida. (SANTIBANEZ-GONZALEZ, MATEUS, LUNA, 2012)

Cada partícula caracteriza-se por sua aptidão (valor da função objetivo), sua posição (conjunto de valores das variáveis independentes do problema) e por um vetor que regula a mudança de posições. Este vetor é chamado ‘velocidade’. (DRÉO et al., 2006).

O algoritmo que descreve o PSO é a seguir: (BARBOSA, 2012)

- 1: Inicializa o enxame aleatoriamente
- 2: Armazena  $p_{best}$  e  $g_{best}$
- 3:  $t = 0$
- 4: **enquanto**  $t <$  iterações máxima **faça**
- 5: para todo pássaro faça
  - 6: Atualiza a velocidade atual
  - 7: Movimenta os pássaros
- 8: Avalia a posição segundo:  $\sqrt{(x_{atual} - x_m)^2} + \sqrt{(y_{atual} - y_m)^2}$
- 9: fim
- 10: Armazena  $p_{best}$  e  $g_{best}$
- 11: fim

Segundo Prata (2012), no PSO a filosofia é a seguinte: gera-se uma população inicial de partículas (indivíduos), cada qual numa posição inicial do espaço de soluções e atribuída de uma velocidade inicial. Cada partícula irá percorrer o espaço de busca n-dimensional e terá sua velocidade atualizada de acordo com a velocidade das demais partículas. Partículas que estiverem distantes das regiões promissoras do espaço de busca, terão sua velocidade elevada, enquanto partículas que estiverem próximas das regiões promissoras do espaço de busca, terão sua velocidade moderada.

### 2.5.2 ACO (*Ant Colony Optimization*)

O *Ant Colony Optimization* (ACO) é uma meta-heurística baseada no comportamento de colônias de formigas em busca de comida. As formigas se comunicam através de um feromônio, uma substância química, utilizada para criar as trilhas que orientam até a comida. A partir do feromônio deixado pelas formigas, as demais formigas seguem essa trilha, depositando também feromônios, aumentando assim a probabilidade que este caminho seja utilizado.

A primeira versão do ACO conhecida como *Ant System* (AS) definida por (Dorigo et al, 1991) foi aplicado a problemas de otimização combinatória em domínios discretos, como por exemplo, caixeiro viajante, roteamento, etc. Este algoritmo teve três diferentes variações: *AS-Density*, *AS-Quantity* e *AS-Cycle*, cuja diferença principal entre eles refere-se a atualização da trilha dos feromônios (Dorigo et al., 1996). Os algoritmos de otimização por colônia de formigas propostos atualmente tem sido originados do algoritmo *AS-Cycle*, pois foi o que se mostrou mais eficiente (Serapião, 2009).

O pseudocódigo do algoritmo ACO é demonstrado a seguir (TAVARES e GODINHO, 2013) na Figura 7.

Figura 7 - Pseudocódigo do ACO

|   |  |
|---|--|
| 1 | Inicialize   |
| 2 | <b>Repita</b> Neste nível, cada execução é chamada <i>iteração</i>                   |
| 3 | <b>Repita</b> Neste nível, cada execução é chamada <i>passo</i>                      |
| 4 | Cada formiga aplica uma regra de transição para construir a próxima etapa da solução |
| 5 | Aplica-se a atualização local de feromônios  |
| 6 | <b>Até que</b> todas as formigas tenham criado uma <i>solução completa</i>           |
| 7 | Aplica-se o procedimento de busca local  |
| 8 | Aplica-se o procedimento de atualização global de feromônios                         |
| 9 | <b>Até que</b> o critério de parada seja satisfeito                                  |

Fonte: TAVARES e GODINHO (2013)

No algoritmo apresentado na Figura 7, na primeira iteração, cada formiga é posicionada no nó inicial, definido por uma regra específica. As formigas podem iniciar todas no mesmo nó inicial ou serem alocadas de forma aleatória nos diversos nós. Depois de posicionadas, movem-se conforme uma *regra de transição* até que se tenha uma solução completa. Para que a formiga defina qual o próximo nó seja o escolhido, a *regra de transição* se baseia no cálculo de uma probabilidade, tendo com um dos parâmetros o feromônios depositados. Depois de movimentação da formiga, os feromônios são atualizados conforme uma *regra de atualização local*. Quando todas as formigas terminam de construir suas soluções, são executados dois procedimentos: procedimento de busca local, para melhorar a solução criada pelas formigas, e que é opcional; aplicação de uma *regra de atualização global de feromônios*. Uma nova iteração será realizada e ciclo continuará até que o critério de parada seja satisfeito, como por exemplo, o tempo computacional tenha sido atingido ou o número máximo de iterações tenha sido alcançado.

O algoritmo AS continuou sofrendo variações, de onde se originou o ACS (*Ant Colony System*). A seguir são apresentados as características dos algoritmos AS e ACS (Neto, 2010).

a) Ant System

**- Fase de Construção da Solução:** cada formiga escolhe a cidade para qual irá baseada em uma probabilidade definida pela distância da cidade e da quantidade de feromônio presente na aresta que conecta a esta cidade. Cidades já visitadas não podem ser visitadas novamente até que a solução esteja completa, devido a memória que grava as cidades que já visitou.

Para definir a próxima cidade a ser visitada, é calculado a probabilidade de todas as cidades ainda não visitadas e a cidade com maior probabilidade é a escolhida. Para isso, usa-se seguinte fórmula:

$$p_{ij}^k = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}]^\alpha [\eta_{il}]^\beta}$$

Onde

$\alpha$  e  $\beta$  importância do feromônio e da informação heurística.

$\tau_{ij}$  é a quantidade de feromônio na aresta  $ij$ ,

$\eta_{ij}$  é o inverso de  $d_{ij}$

$N_i^k$  o conjunto das cidades ainda não visitadas pela formiga  $k$ .

Neste algoritmo, não há depósito de feromônio nesta fase, bem como não realiza nenhuma ação global e nem busca local.

**- Fase de Atualização:** onde é realizado tanto o incremento do feromônio, quanto a sua evaporação por todas as formigas. A atualização é definida pela fórmula:

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \sum_{k=1}^m \Delta \tau_{ij}^k$$

onde  $\rho$  é um parâmetro para regular a taxa de evaporação do feromônio,  $m$  o número de formigas e  $\Delta \tau_{ij}^k$  é definida por:

$$\frac{1}{L^k},$$

se aresta  $ij$  faz parte do caminho da formiga  $k$  e

0, caso contrário.

onde  $L^k$  é o custo do caminho percorrido pela formiga  $k$

Para se obter um bom desempenho do algoritmo AS é necessário ajustar alguns parâmetros, a fim de se obter o equilíbrio entre intensificação e diversificação, a saber:

- a importância do feromônio;
- informação heurística;
- taxa de evaporação;
- tamanho da população de formigas.

Exemplos de valores para esses parâmetros citados por Neto (2010) são:

- a)  $\rho$ , deve ser menor que 1, para que não ocorra uma acumulação ilimitada do feromônio, prejudicando a diversificação;

- b)  $\alpha$ , deve ser maior que 0, pois quando o seu valor é alto, significa que o feromônio é muito importante e as formigas tendem a estas arestas, ocasionando o efeito de intensificação; se for baixo, a construção comporta-se a uma construção gulosa aleatória.

#### b) ACS – Ant Colony System

O ACS foi desenvolvido baseado no AS, tendo como diferença os seguintes aspectos:

##### - Fase de Construção da Solução

- a. Um parâmetro  $q_0$  foi adicionado para auxiliar na probabilidade de definir qual aresta cada formiga deve seguir. A cada iteração um número é sorteado de forma aleatória e se for maior que  $q_0$ , utiliza-se a regra de decisão do algoritmo AS, senão, utiliza-se regra de decisão definida por:

$$p_{ij}^k = \tau_{ij} \eta_{ij}^\beta$$

- b. É realizado a atualização local de feromônio, onde a cada iteração as formigas atualizam o feromônio da aresta pela qual acabaram de atravessar. Utilizado para balancear a regra de decisão inserida acima, que comporta-se um tanto gulosa. A expressão é dada por:

$$\tau_{ij} = (1 - \varphi) \cdot \tau_{ij} + \varphi \cdot \tau_0$$

onde  $\varphi$  é um parâmetro definido entre 0 e 1, e  $\tau_0$  é o valor inicial dos feromônios das arestas.

##### - Fase de Atualização

É executada uma atualização global feita somente pela formiga que construiu a melhor solução até o momento. Por ser global, todas as arestas são atualizadas conforme a expressão dada por:

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \rho \cdot \Delta \tau_{ij}^{melhor}$$

A literatura (Neto, 2010) cita que os melhores valores para os parâmetros, para que a construção gulosa seja priorizada, com o intuito de favorecer a intensificação no ACS são:

- $m$  deve ser igual a 10,
- $\alpha$  e  $\rho$  devem ser iguais a 0.1,
- $\beta$  deve ser igual a 2,

- $q_0$  deve valer 0.9.
- inicialização dos feromônios das arestas deve-se utilizar a expressão  $1/n.L1nn$ .

A adoção da utilização do ACO para o desenvolvimento da pesquisa, levou-se em consideração o estudo realizado por Breve (2007), onde apresenta o comparativo de desempenho entre ACO e PSO no problema do caixeiro viajante.

Como nesta pesquisa tem uma analogia à busca de um caminho, logo o ACO se mostrou uma opção melhor em relação ao PSO.

### 3 TRABALHOS RELACIONADOS/CORRELATOS

Neste capítulo serão apresentados os trabalhos relacionados ao propósito da pesquisa, sendo que estão dispostos nas subseções. Por envolver o Estilo de Aprendizagem, Algoritmos Inteligentes, e até mesmo abordagens diferentes, se tornou interessante a análise dos trabalhos correlatos selecionados, pois visam considerar o perfil de cada Aprendiz de forma distinta.

Tem-se o trabalho na seção 3.1, que apresenta abordagem para ajuste do Estilo de Aprendizagem com base na cadeia de Markov; o trabalho na seção 3.2, aborda a utilização de metadados IEEE-LOM nos Objetos de Aprendizagem para associar ao Estilo de Aprendizagem mapeado do Aprendiz; no trabalho na seção 3.3, apresenta a utilização de algoritmo inteligente (PSO) para escolha de Objetos de Aprendizagem, baseando na utilização da Espiral de Kolb e Inteligências Múltiplas; no trabalho na subseção 3.4, aborda a utilização em conjunto do IEEE-LOM (caracterizar Objeto de Aprendizagem), e com algoritmo ACO; e por último, no trabalho na seção 3.5, propõe um modelo de apresentação adaptativa de Objetos de Aprendizagem baseando-se no Estilo de Aprendizagem.

#### 3.1 Detecção e Correção Automática de Estilos de Aprendizagem em Sistemas Adaptativos para Educação

Neste artigo de Dorça et al. (2011), é apresentado uma abordagem para a identificação e ajuste automático de estilos de aprendizagem (EA) baseada em cadeias de Markov. Indica a dificuldade das plataformas *e-learning* não considerarem as características particulares de cada estudante/usuário, onde o mesmo conteúdo e estratégia pedagógica são, geralmente, aplicados de forma genérica. Uma outra aplicação desta pesquisa aplicada para educação a distância é descrita em Dorça et al. (2014).

Aponta a possibilidade de proporcionar mecanismos personalizados para que o estudante possa selecionar caminhos dentro dos mecanismos gerais disponibilizados pelo professor/tutor em ambientes informatizados para facilitação da sua aprendizagem.

Um trabalho baseado na combinação de estilos de aprendizagem (CEA) e na correção dinâmica das inconsistências no modelo do Estudante (ME), levando em conta o forte aspecto não-determinístico do processo de aprendizagem. Adotando como fatores importantes, a saber: os objetivos de aprendizagem, o nível de conhecimento, os interesses, as preferências, os estereótipos, as preferências cognitivas e os estilos de aprendizagem (EA).

Através dessa definição, é estabelecido um modelo probabilístico baseado na combinações de estilos de aprendizagem (CEA) para detecção e correção automática das

preferências do estudante, utilizando a Cadeia de Markov. Afirma possibilitar que sistemas adaptativos para educação existentes possam, de forma eficiente, automaticamente detectar os estilos de aprendizagem dos estudantes com alto índice de precisão. Para isso, foi adotado uma abordagem baseada em um processo estocástico, que se caracteriza principalmente pela representação da variabilidade e das incertezas inerentes ao processo de aprendizagem de cada estudante.

Esta abordagem foi testada sob técnicas de simulação computacional, onde foi observado o comportamento do modelo em diferentes contextos e situações. Então, foi efetuada a implementação e utilizado em LMSs (Learning Management System) existentes.

Foi adotada a definição do Estilo de aprendizagem segundo as características ressaltadas por Felder e Siverman (1988) (FSLSM). Este utiliza o conceito de dimensões e, então, os estilos de aprendizagem são descritos em mais detalhes, baseado numa escala de quatro dimensões, a saber: ativo-reflexivo, sensitivo-intuitivo, visual-verbal, sequencial-global. O FSLSM combina a maioria dos modelos de estilo de aprendizagem.

A característica ressaltada no trabalho é que, no FSLSM, os estudantes não são classificados em tipos, é baseado na ideia de que cada estudante possui uma preferência em cada uma das suas quatro dimensões, estimado em valores dentro de um intervalo. Utiliza escalas ao invés de tipos, os pesos das preferências podem ser descritos, permitindo distinguir-se entre preferências fortes, moderadas e leves.

Outro ponto considerado é que o FSLSM considera o estilo de aprendizagem como tendências (estudantes possuem uma tendência EA), entretanto, podem agir diferente em casos isolados (sendo caso não-determinístico). Sendo assim, como o modelo de tendências, a descrição do EA considera exceções e situações extraordinárias, sendo tratado de forma probabilística ao invés de determinístico, sendo baseada nos pesos das preferências em cada uma das quatro dimensões do FSLSM, dentre outras variáveis.

É apresentado um modelo de estudante (ME) utilizado para avaliar o nível de conhecimento do estudante, assim podendo fazer inferências a respeito de suas características, concepções e estratégias de raciocínio empregadas para atingir o nível de conhecimento atual.

Sendo também permitido mudar vários aspectos do sistema, para os casos particulares, usando para definir o estudante. Este modelo, adotado no trabalho, é baseado na Taxonomia de Bloom para níveis cognitivos com o FSLSM para estilos de aprendizagem. A abordagem utilizada nessa modelagem de estudante é a de sobreposição, onde o modelo do estudante é visto como um subconjunto da base de conhecimento do sistema.

Para definir a preferência de cada estilo de aprendizagem, é armazenado, no ME, como um valor real dentro do intervalo [0...100] ao invés de valores inteiros no intervalo [-11...+11]. Nesse ponto de conversão que afirma a base para implementação de adaptação estocástico através de cadeias de Markov, no que tange à construção das matrizes de transição de estados.

Em cada seção de aprendizagem do curso, o estudante é submetido a um conjunto de objetos de aprendizagem que atendam a uma CEA específica, gerada estocásticamente de acordo com os valores dos estilos de aprendizagem armazenados no ME. O processo de composição do CEA é modelado através de 4 cadeias de Markov concorrentes. Cada estado de uma cadeia de Markov representa um estilo de aprendizagem dentro de uma dimensão FSLSM.

Os valores das matrizes de probabilidade são alteradas constantemente ao longo do processo de ensino-aprendizagem, corrigindo as preferências no ME. Sendo um processo particular a cada estudante. Esses valores armazenados no ME são modificados sempre que o sistema detectar problemas de aprendizagem, quando o desempenho do estudante for considerado insatisfatório na avaliação de uma sessão de aprendizagem. Ocorrendo quando não é atingido um limiar.

Foi estabelecido um conjunto de regras de produção, que levam em consideração os estilos de aprendizagem armazenados no ME e CEA considerada durante a sessão de aprendizagem. Estas regras são executadas para cada uma das 4 dimensões do FSLSM sempre que o estudante obter um desempenho baixo. Se as regras não forem satisfeitas para alguma dimensão, o ME não é alterado na dimensão respectiva.

A abordagem proposta se mostrou eficiente e eficaz na detecção e correção automática, dinâmica e contínua dos estilos de aprendizagem do estudante, através de um processo não-determinístico e evolutivo, que atua ao longo de todo o processo de ensino-aprendizagem. Proporcionando, também, uma redução de custo computacional e complexidade menor em relação a outras abordagens.

### **3.2 Em direção à recuperação automática de objetos de aprendizagem em repositórios através da associação dos estilos de aprendizagem de estudantes com metadados no padrão IEEE-LOM**

Este trabalho de Resende et al. (2014), apresenta uma proposta para personalização do processo de ensino, que se baseia no mapeamento automático de características de estilos de aprendizagem de estudantes em metadados de objetos de aprendizagem.

Para tal, foi apresentada pelo autor a necessidade de produzir uma solução para adaptação de conteúdo em diferentes formas de materiais de estudo, em vista que cada estudante pode ter perfil diferenciado. Assim sendo necessário considerar a particularidade de cada indivíduo, de acordo com suas preferências no que tange aprendizagem, e a partir disso ter melhor aproveitamento no decorrer da apresentação do conteúdo a ser estudado.

Outra problemática apontada, é os Sistemas de Gerenciamento de Aprendizagem (Learning Management Systems – LMS), dão suporte ao professor no que tange criar, administrar e manter cursos online, com vários recursos, porém, não apresentam o recurso de adaptabilidade, assim carecem de apoio para professores realizarem adaptações, o que pode ser uma das razões pelos quais sistemas adaptativos são raramente utilizados em instituições de ensino. Sendo assim, a interatividade dos sistemas existentes não se tem disponível o desejado.

O objetivo apontado pelo autor é de prover o aprendiz com serviços e informações que sejam relevantes ao processo de ensino aprendizagem e que ao mesmo tempo atendam o perfil do respectivo aprendiz.

A inviabilidade de se obter, manualmente, os OAs baseado no estilo de aprendizagem do estudante, em qualquer que seja o repositório, é sustentada no fato de haver uma grande variação de combinações de características de aprendizagem de estudantes neste tipo de curso. Ainda maior no que se refere a repositórios com número elevados de objetos a serem analisados. Nesse ponto o autor aponta uma alternativa para auxiliar nessa dificuldade operacional, para fazer a correlação automática entre as características de aprendizagem de estudantes e de OAs.

Então, é apresentada no trabalho uma abordagem para mapeamento de características de OAs em estilos de aprendizagem considerando o modelo definido por Felder e Silverman (1988). Considerando o modelo de estilos de aprendizagem de FS e o padrão IEEE-LOM para metadados de OA, e buscando relacionar as características deste em relação àquele e, a partir disto, propõe a recuperação de OAs de acordo com os estilos de aprendizagem de um estudante específico. O objetivo principal apresentada neste trabalho foi definir associações entre características de EAs no modelo de Felder e Silverman (FS) e metadados de OAs no padrão LOM.

O modelo adotado é baseado na ideia de que cada estudante possui uma tendência a um dos estilos em cada uma das suas quatro dimensões. A seguir, apresenta-se as descrições das dimensões deste modelo, sendo: Percepção (podendo ser sensorial/sensitivo, ou, intuitivo), Entrada (sendo visual, ou, verbal/textual), Processamento (sendo ativo, ou, reflexivo) e Organização (classifica-se em sequencial, ou, global).

O desenvolvimento de material didático a ser disponibilizado em ambientes de ensino-aprendizagem via web exige criatividade, disponibilidade de tempo e conhecimento de tecnologias adequadas. Visando minimizar o tempo e esforço despendido no desenvolvimento de material educacional, e pensando em uma forma de reutilizar esse material, surgiu o conceito de Objetos de Aprendizagem (OA).

Os metadados de um objeto apresentam informações sobre o próprio objeto, seja físico ou digital. Devido a oferta de OAs é crescente, a falta de informação ou metadados sobre objetos ocupa um ponto crítico e fundamental na habilidade de descobrir, gerenciar e utilizar OAs.

Os metadados definem o conjunto mínimo de propriedades necessárias para permitir o gerenciamento, a localização e a avaliação destes objetos, utilizados para contextualizar as características de um determinado elemento de informação, no caso os objetos de aprendizagem. O padrão adotado é o IEEE-LOM, onde possui um esquema de dados conceituais que define a estrutura de instâncias de metadados para OAs. Sendo um facilitador no que tange a análise com menor custo de tempo.

Para efetuar o relacionamento entre o modelo de FS e o padrão LOM, foi apresentado pelo autor um estudo de conceitos aplicadas em ambas teorias, buscando encontrar pontos que elas se relacionam.

Foram apresentadas tabelas que relacionam as duas teorias conforme discutida pelo autor. Dentre elas, considerando-se as características de EAs apresentadas, foram analisados todos os campos do padrão IEEE-LOM cujos valores atendiam as características das dimensões do modelo de Felder e Silverman e, assim, foi definido quais destes campos seriam utilizados para realizar o mapeamento entre o modelo do estudante e metadados de OAs.

Apresenta-se os campos que têm seus valores associados às características de perfis de estudantes mapeados pelas dimensões de Estilo de Aprendizagem. Cada coluna representa uma dimensão do modelo de Felder e Silverman (1988). Nelas, o estilo atendido é descrito, seguido por justificativa entre parênteses, que indica qual característica de Estilo de Aprendizagem é afetada.

Desta forma, através deste trabalho, o autor propôs preencher estas lacunas ao apresentar uma abordagem para mapeamento entre estilos de aprendizagem e metadados de objetos de aprendizagem que leva em consideração o modelo de estilos de aprendizagem mais conhecido, completo e amplamente utilizado na implementação de sistemas educacionais, e um padrão de metadados amplamente conhecido e adotado em repositórios de objetos de aprendizagem.

Então, uma das características inerentes à proposta apresentada neste trabalho é a facilidade de aplicação em qualquer ambiente de aprendizagem, já que não é levada em consideração características específicas de um ambiente específico. Desta forma, ambientes virtuais de aprendizagem existentes passariam a ter a possibilidade do reuso em larga escala de objetos de aprendizagem através de catalogações no padrão IEEE LOM e da possibilidade de personalização do conteúdo de acordo com os estilos de aprendizagem dos estudantes.

### **3.3 Uma proposta de um modelo computacional que usa PSO para a escolha de Objetos de Aprendizagem baseado na Espiral de Kolb e nas Inteligências Múltiplas**

Neste artigo de Moura e Fernandes (2012), é descrita uma proposta de um modelo computacional que utiliza a técnica *Particle Swarm Optimization* para recomendar Objetos de Aprendizagem de acordo com a teoria das Inteligências Múltiplas (IMs) e a Espiral de Aprendizagem de Kolb, tentando abstrair uma correlação entre eles. Sugere a utilização destes conceitos tanto para o ensino presencial e/ou à distância.

Visou a adaptação da forma de apresentação do conteúdo a ser mostrado ao estudante. Esta necessidade é apontada devido as pessoas terem formas diferentes de aprender, ou seja, ter que trabalhar individualmente com cada estudante de modo mais apropriado. Dentre as dificuldades, tem-se as técnicas empregadas e a forma como é apresentado o conteúdo.

Ainda neste trabalho propõe um modelo que relaciona duas teorias bastante difundidas na educação e na informática na educação: os estilos de aprendizagem e as IMs. Pretendendo oferecer ao estudante Objeto de Aprendizagem ou material didático que se adeque melhor às suas características. Sendo fundamental, onde estudantes diferentes aprendem o mesmo conteúdo que são dispostos de formas diferentes.

Nesta proposta utilizou a teoria de David Kolb, onde este é uma espiral de aprendizagem que evolui de acordo com quatro fases: o sentir, o observar, o refletir e o experimentar. Também foi abordada a teoria das Inteligências Múltiplas(IM) de Gardner, onde indicou que todo ser humano possui oito inteligências em níveis diferentes. Os Estilos de Aprendizagem mostram a preferência que o estudante opta para visualizar o material de estudo, e esta teoria (Inteligências Múltiplas) dão o formato deste material, estas são ligadas a fatores culturais particularizados de cada indivíduo.

Foi proposto um modelo computacional que, durante o processo de aprendizagem baseado na espiral de Kolb, seleciona os OA's mais adequados ao estudante através do *Particle Swarm Optimization* (PSO). Assim, visou auxiliar o processo de ensino e aprendizagem através

de conhecimento das necessidades particulares de cada estudante, analisando sua evolução na espiral. No final do processo, é associada a relação da espiral com o aprimoramento do nível de inteligências múltiplas do estudante.

Ainda foi efetuada uma pesquisa com 5 profissionais da educação (pedagogos), aos quais foram submetidos questionários de pesquisa onde atribuem pesos ao Inventário de Estilos de Aprendizagem (IEA) de Kolb relacionando as sentenças às quatro IM escolhidas para este trabalho. Com esse questionário, baseou a criação de um modelo computacional que ajude um sistema *e-learning*, ou até mesmo um professor no ensino presencial a escolher o OA ou material didático mais adequado a ser aplicado a um estudante específico.

No modelo computacional proposto, o IEA da espiral de Kolb e o questionário de detecção de IM foram, previamente, respondidos pelo estudante. Foi elaborado um fluxograma para mostrar a estrutura do algoritmo desenvolvido para o sistema adaptativo de aprendizagem usando PSO, denominado “Sistema Adaptativo Educacional usando PSO (SAEP)”.

No fluxograma, o módulo “Inicia Kolb” define uma matriz (12x4) onde cada linha representa uma sentença do IEA e cada coluna representa uma dimensão da espiral (EC, OR, CA, EA). A partir desta matriz, encontra-se o valor inicial das quatro áreas. Essas áreas são calculadas de acordo com o somatório do IEA de Kolb respondido pelo estudante. Cada somatório origina um cateto sobre um dos semieixos do plano cartesiano, e estes catetos gerarão as áreas em cada uma das dimensões. Das 4 áreas, a de maior valor definirá o estilo a partir do qual a espiral se iniciará e se seguirá no sentido horário aos estilos seguintes.

O módulo “Inicia Gardner” define uma matriz (4x9) onde cada questionário referente à uma IM é representada por uma linha. Cada pergunta específica da referida IM é representada por uma coluna de uma linha que pode receber os valores 1,2,3 ou 4. Neste trabalho, foram usadas apenas quatro inteligências múltiplas: Linguístico-verbal (LV), Lógico-matemática (LM), Visual-espacial (VE) e Cinestésico-corporal (CC).

Definido o estilo de Kolb (pela maior área calculada), o módulo “PSO” será acionado para que se busque dentro deste estilo a ordenação de partículas do PSO que atribuiriam OA ao estilo do estado mais adequado àquele momento. Este módulo inicia-se com a definição de uma matriz de partículas ótimas (16x30) onde as 4 primeiras linhas denotam as partículas ótimas do estilo Divergente, as 4 linhas seguintes denotam partículas ótimas do estilo Assimilador, as outras 4 linhas denotam partículas ótimas do estilo Convergente e as últimas 4 linhas denotam partículas ótimas do estilo Acomodador. Dentro de cada um desses estilos, cada partícula ótima representa uma sugestão para oferecer um OA referente à uma IM (LV, LM, VE, CC).

A representação de uma partícula (indivíduo) é dada por um vetor de 30 posições. As primeiras 24 posições referem-se às 12 questões do questionário de Kolb (as posições pares indicam a primeira dimensão e as ímpares a segunda), as duas posições seguintes fornecem as somas das duas dimensões (posição 25 é a soma da primeira dimensão e a posição 26 a soma da segunda) e as quatro últimas identificam o percentual das quatro inteligências.

O algoritmo PSO utilizou uma matriz (50x30) onde 50 é o número de partículas, tendo como a convergência destas para um dos 4 ótimos do estilo atual da espiral. Para cálculo de proximidade, foi utilizada a distância Euclidiana de cada partícula em relação aos ótimos.

Assim, identifica-se o OA mais adequado para dado momento do processo de aprendizagem. A primeira partícula representa a resposta do estudante aos questionários e as demais são produzidas aleatoriamente.

Em relação aos parâmetros de inicialização do PSO, os seguintes valores foram definidos:  $c_1=c_2=2$  (parâmetros cognitivo e social, também nomeado de Taxa de Aprendizado),  $w_{\min}=0.4$  (ponderação mínima de inércia) e  $w_{\max}=0.9$  (ponderação máxima de inércia). Segundo o autor, estes valores são sugeridos em uma série de problemas de otimização e que de modo geral mostraram bons resultados. A saída do módulo “PSO” é um vetor com quatro posições, onde se identifica da primeira para quarta posições a ordem em que os tipos de OA’s devem ser apresentados ao estudante. Foi considerado para a melhor partícula global do PSO como sendo aquela que representa a partícula do estudante (primeira linha da matriz 50x30).

Em seguida, o módulo “Oferece Módulo” simula o uso dos OA’s conforme saída do PSO, avaliando os resultados de aprendizagem obtidos através de uma heurística do módulo “Avalia”. Se a nota obtida nesta heurística for favorável, o estudante seguirá para o próximo módulo (próximo estilo na espiral). Caso contrário, o estudante permanece no mesmo módulo e o próximo OA é sugerido e avaliado. Caso os quatro OA’s sejam sugeridos e avaliados de forma desfavorável, automaticamente o estudante passa para o próximo módulo do curso.

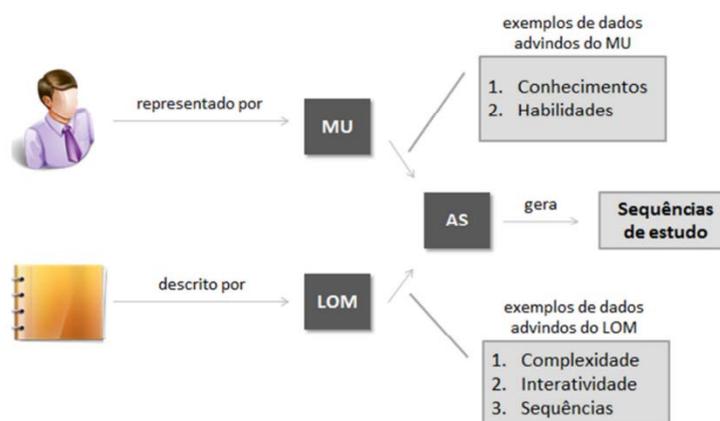
A avaliação que determina se haverá a passagem de um estilo para outro na espiral é composta de três fatores normalizados entre zero e um. O primeiro fator de avaliação identifica se o estudante gostou (nota = 1) ou não (nota = 0) do OA proposto, o segundo o conceito obtido pelo estudante em atividades realizadas após o uso do OA e o terceiro verifica o tempo total que o estudante gastou para fazer essas atividades. Para dar a nota ao fator tempo, utilizou-se uma penalização. Uma vez calculados os fatores de avaliação do OA, o módulo “Avalia” finaliza com a média ponderada desses fatores. O primeiro fator tem peso igual a 2, o segundo peso 4 e o terceiro peso 1.

Por fim, após o módulo “Avalia” é verificado se o OA é considerado bom ou não. Se a nota obtida pelo o OA foi maior que 0.4 (pode atingir o valor máximo de 1) então é considerado um objeto bom, caso contrário é considerado um objeto ruim. O processo todo se repete conforme o número máximo de ciclos na espiral definido no próprio sistema.

### 3.4 *AntStudy*: Proposta de definição de sequências de estudo utilizando *Ant System*

A proposta de Araújo e Fagundes (2011) é utilizar o padrão LOM e modelos de usuários como entradas para o algoritmo ACO, do tipo *Ant System*, que terá como objetivo definir as sequências de estudo em um ambiente virtual de aprendizagem. O padrão LOM é utilizado como modelo de metadados para caracterizar objeto de aprendizagem, onde pode-se identificar as informações: complexidade, interatividade, sequências, entre outros. O modelo de usuário (MU) especifica os conhecimentos e as habilidades do aprendiz. A Figura 8 ilustra o esquema da proposta.

Figura 8 - Esquema da Proposta de Pesquisa



Fonte: Araújo e Fagundes (2011)

O algoritmo ACO, chamado nesta pesquisa por *AntStudy*, baseia-se no comportamento das formigas. Na sua fase inicial, considera as sequências pré-determinadas no contexto de trabalho, como os planos de ensino propostos pelos professores.

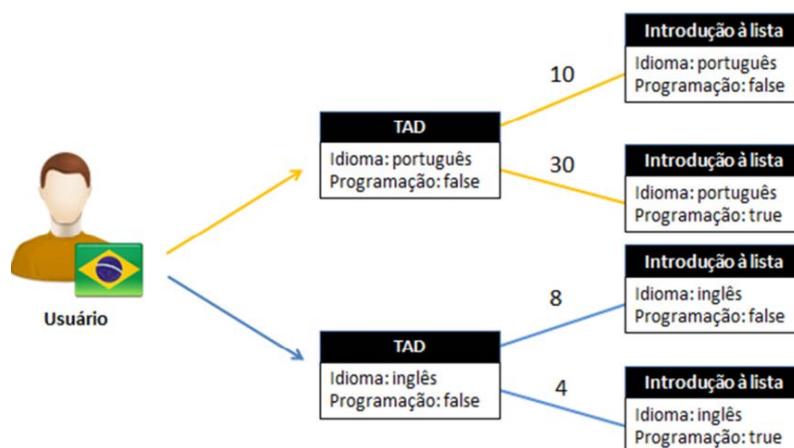
As taxas de feromônio são atualizadas a partir do conhecimento abstraído do modelo de usuário e o conhecimento requerido para o estudo de um determinado recurso didático. As sequências de estudo são formadas pelo conjunto de objetos de aprendizagem dispostos em forma de grafo, e as ligações entre os nós desse grafo, representam os possíveis segmentos de estudo, tomando como base as informações dos metadados do LOM, a fim de possibilitar uma sequência lógica nos assuntos abordados. Com o uso dos grafos e os metadados do LOM, é possível relacionar os objetos de aprendizagem de forma a respeitar particularidades de cada

estereótipo do usuário, sendo assim, propicia-se a oferta de objetos de aprendizagem de acordo com as características do usuário.

A escolha dos objetos de aprendizagem considera a quantidade de feromônio presente nas ligações entre os nós do grafo (ou seja, entre os objetos de aprendizagem) e o estereótipo do usuário. Se existir somente uma possibilidade, ela sempre será escolhida e apresentada ao usuário.

Considerando o exemplo da Figura 9, e a primeira inicialização do algoritmo com sequencias pré-definidas, é possível identificar dois materiais sobre “Tipos Abstratos de Dados (TAD)”. No entanto, para o aprendiz brasileiro o objeto de aprendizagem com o estereótipo “Idioma” com valor “português”, torna-se a única possibilidade para iniciar, independentemente da quantidade de feromônio (se houvesse). Já para acessar o segundo objeto de aprendizagem que trata de “Introdução a listas”, observa-se ainda na Figura 9, que existem duas possibilidades: uma ligação com 10 unidades de feromônio e outra com 30 unidades de feromônio. A probabilidade de escolher o segundo objeto de aprendizagem é maior, uma vez que a quantidade de feromônio nessa ligação também é maior.

Figura 9 - Funcionamento do *AntStudy*



Fonte: Araújo e Fagundes (2011)

Em seguida, ocorre a evaporação dos feromônios e estes são atualizados em todo o grafo, independente do estereótipo do usuário. Ao concluir a sequência de estudos, o aprendiz avalia o aprendizado obtido. De posse dessa informação, associada às características do estereótipo e do objeto de aprendizagem, calcula-se a taxa de atualização do feromônio, que será aplicada à ligação que o usuário acabou de percorrer. Para isso, deve-se levar em consideração os conteúdos que estão sendo abordados, a complexidade do objeto de aprendizagem em relação ao nível de conhecimento que um usuário possui, entre outros.

Os resultados alcançados com a pesquisa foram favoráveis, pois conseguiram utilizar um algoritmo inicialmente voltado a problemas objetivos, numéricos, também em problemas que envolvem certa subjetividade, como é o estabelecimento de uma sequência de estudos. O que foi alcançado a partir da definição de parâmetros que permitam quantificar as informações utilizadas para a atualização do feromônio. Mas, ainda assim, o modelo de usuário necessita ser mais representativo.

### **3.5 Modelo de Apresentação Adaptativa de Objeto de Aprendizagem baseada em Estilos de Aprendizagem**

O modelo de apresentação adaptativa de objetos de aprendizagem proposto em Silva et. al, (2016) teve como base as propriedades dos estilos de aprendizagem do modelo de Felder-Silverman (1988), que a partir de um estudo e investigação, pode-se definir as “características de apresentação para OA” em relação a sequenciamento, apresentação e forma/formato de conteúdo e recursos que compõem o OA, dos quais pode-se estabelecer os parâmetros e atributos necessários para definição do modelo.

Segundo o modelo de adaptação, o OA será formado por “elementos de composição de conteúdo” que constituem as seguintes etapas:

- Resumo (Res): fornece uma visão geral e clara do conteúdo a ser abordado;
- Introdução (Int): possui um conteúdo com a breve descrição do assunto a ser abordado;
- Desenvolvimento (Des): possui um conteúdo com descrição mais abrangente do assunto de uma forma mais completa;
- Atividade (Ati): formado por um conteúdo para fixação do assunto; e,
- Avaliação (Ava): avaliação do conteúdo abordado.

Os “elementos de composição de conteúdo” listados são organizados em relação aos parâmetros e atributos definidos no modelo de apresentação adaptativa. Estes parâmetros e atributos combinados aos elementos de criação e composição do OA estabelecem como o OA pode adaptar-se em função de cada EA. A seguir uma descrição desses parâmetros e atributos:

- Recurso (R): define os tipos de recursos que podem ser utilizados nos elementos da composição de conteúdos. Tipos: Vídeo (Vid); ou Exercício (Exe); ou Gráfico (Gra); ou Figura (Fig); ou Palestra (Pal); ou Slide (Sli); ou Auto-Avaliação (Aav); ou Tabela (Tab); ou Experimento (Exp); ou Simulação (Sim); ou Questionário (Que); etc.

- Forma de Exploração (FE): define como o conteúdo pode ser estruturado em relação a forma de exploração do conteúdo pelo aprendiz. Tipos: Rede (Red): não há um roteiro, investigação aleatória; ou Linear (Lin): há um roteiro a seguir, investigação dirigida.
- Ordem de Detalhamento (OD): estabelece como o aprendiz prefere abordar os conteúdos apresentados no OA. Tipos: Geral-paraEspecífico (Ger-p-Esp): inicia na parte geral e prossegue para a parte específica; ou Específico-para-Geral (Esp-p-Ger): inicia na parte específica e prossegue para a parte geral.
- Ordem de Composição (OC): define a organização das etapas usadas na composição dos conteúdos de um OA, para apresenta aos aprendizes. São três ordens de composição definidas:
  - ordem 1 – 1º Introdução, 2º Desenvolvimento, 3º Resumo, 4º Atividade, 5º Avaliação;
  - ordem 2 - 1º Introdução, 2º Desenvolvimento, 3º Atividade, 4º Resumo, 5º Avaliação; e
  - ordem 3 - 1º Resumo, 2º Introdução, 3º Desenvolvimento, 4º Atividade, 5º Avaliação.

Foram criadas regras de adaptação para composição da interface do OA de acordo com os estilos do modelo de Felder-Silverman (1988). O Quadro 1 ilustra um recorte da tabela original que contém os atributos e parâmetros do modelo preenchidos com os valores referentes a cada estilo.

Quadro 1 - Recorte com Estilos Aprendizagem X Parâmetros X Modelo de Adaptação

| Estilo    | Recurso (R)  | Forma de Exploração (FE) | Ordem de Detalhamento (OD)   | Ordem de Composição (OC) |
|-----------|--|--------------------------|------------------------------|--------------------------|
| Ativo     | <i>vid, dia, gra, fig, aav, exe, sim, esq, pag, map, exem.</i>                         | <i>rede</i>              | <i>geral-para-especifico</i> | <i>ordem 2</i>           |
| Reflexivo | <i>dia, gra, pal, sli, aav, tab, exe, sim, esq, ani, pag, dem, exem.</i>               | <i>linear</i>            | <i>especifico-para-geral</i> | <i>ordem 1</i>           |
| Visual    | <i>vid, dia, gra, fig, sli, aav, tab, exp, exe, sim, ani, fot, pag, map, dem, exem</i> | <i>rede</i>              | <i>geral-para-especifico</i> | <i>ordem 3</i>           |
| Verbal    | <i>dia, nar, pal, sli, aav, tab, exe, que, pag, dem, exem.</i>                         | <i>linear</i>            | <i>especifico-para-geral</i> | <i>ordem 3</i>           |

Fonte: Silva et al, (2016)

Foi definido uma regra de adaptação para cada estilo do modelo de adaptação. Quando a regra é acionada, o objetivo é preencher com as informações cada parâmetro do estilo selecionado. Por exemplo, a regra para o estilo *ativo*:

Nome da Regra = “*estiloAtivo*”

Condições:

Estilo = “*Ativo*”

Ações:

- a) Crie uma *OC* na seguinte ordem: “*Introdução*”, “*Desenvolvimento*”, “*Atividade*”, “*Resumo*” e “*Avaliação*”;
- b) Disponibilize os seguintes *Rs*: “*Vídeo*”, “*Diagrama*”, “*Figura*”, “*Gráfico*”, “*Auto-avaliação*”, “*Exercício*”, “*Simulação*”, “*Esquema*”, “*Página-Web*”, “*Mapa*” e “*Exemplo*”;
- c) Indique a *FE* “*Rede*”;
- d) E indique a *OD* “*Geral para Específico*”.

Observa-se que inicialmente, aloca-se a *OC*, depois os *Rs*, em seguida a *FE* e por fim a *OD*, conforme modelado.

As contribuições desta pesquisa foi a definição de um modelo de apresentação adaptativa do OA, que pode ser integrado a criação de ferramentas de autoria de fácil uso, com redução na complexidade e esforço quando na composição de OA adaptados ao EA do aprendiz, facilitando o trabalho do professor/autor, bem como, gerar um acréscimo na motivação em utilizar o OA por atender as preferências individuais de aprendizagem do aprendiz.

## **4 FERRAMENTA DE AUTORIA PARA ATIVIDADES DE APRENDIZAGEM ADAPTATIVAS COM USO DE OBJETOS DE APRENDIZAGEM**

Neste capítulo será abordada a ferramenta de autoria de atividades de aprendizagem adaptativas, sua modelagem, integração do processo de adaptação ao processo pré-3AJC, algoritmo ACO adaptado e prototipação da ferramenta.

### **4.1 Integrando Adaptação ao Processo pré-3AJC**

O processo de pré-autoria pré-3AJC (Sá, 2011), visto no capítulo 2, dá origem a uma atividade de aprendizagem (AA) que pode ser executada por diferentes aprendizes, ou por grupo de aprendizes, no entanto, será a mesma atividade de aprendizagem, mesmo que estes aprendizes possuam características que os diferenciam.

Um dos objetivos desta pesquisa é a prototipação da ferramenta de autoria baseada no processo de pré-autoria pré-3AJC integrando capacidade de adaptação de conteúdo de acordo com o perfil do aprendiz ao interagir com a atividade de aprendizagem. Neste processo, em cada etapa que o professor planejar inserir um objeto de aprendizagem, este objeto de aprendizagem será analisado pela ferramenta de autoria como um conteúdo a ser adaptado conforme características do aprendiz. As características consideradas nesta pesquisa serão o estilo de aprendizagem e o nível de conhecimento.

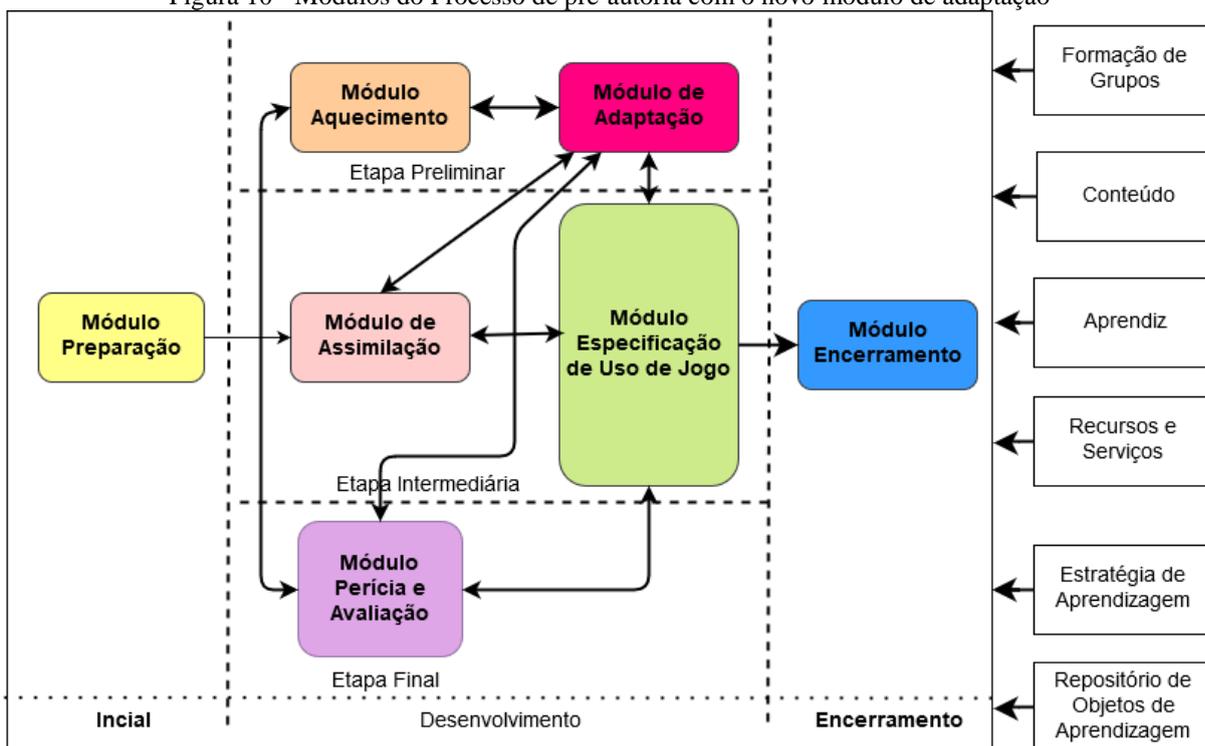
Dentro das soluções que podem ser adotadas para prover a adaptação de conteúdos, tem-se a aplicação de algoritmos inteligentes, como o algoritmo *Ant Colony Optimization* (ACO). O algoritmo ACO necessitou de adequações para contemplar tal problemática e será explicado nas próximas seções, bem como, a prototipação da ferramenta de autoria.

#### **4.1.1 Módulo de Adaptação**

Como a base da pesquisa é o processo pré-3AJC, foram analisados os módulos de sua estrutura original, apresentada na seção 2.2, e posteriormente, adicionado o módulo de adaptação, que visa inserir a capacidade de adaptação no processo de autoria da Atividade de Aprendizagem, como ilustrado no diagrama da Figura 10.

Para efeito de adaptação, são consideradas características do Aprendiz, bem como do Objeto de Aprendizagem. Essas características serão parâmetros no módulo de Adaptação.

Figura 10 - Módulos do Processo de pré-autoria com o novo módulo de adaptação



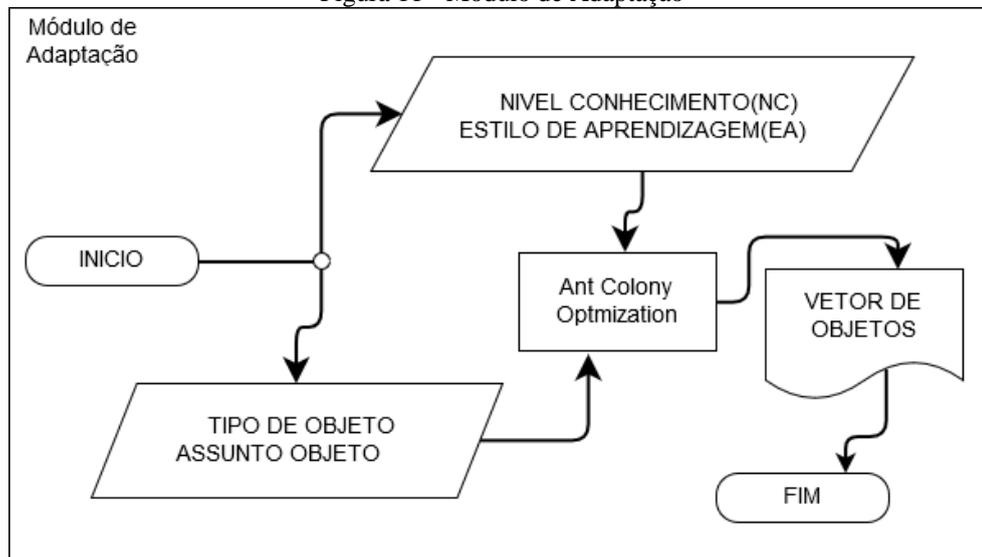
Fonte: Adaptado de Sá (2011)

Pode-se analisar na Figura 10, a inserção do módulo de adaptação no esquema de módulos do processo de pré-autoria, onde este módulo terá comunicação com os demais módulos para, assim, efetuar o processo de adaptação.

Dentre as atividades previstas para o módulo de aquecimento, tem-se as de inferência sobre informações relacionadas ao aprendiz. No caso do módulo de adaptação, o nível de conhecimento (NC) é uma informação necessária para seu funcionamento.

O módulo de adaptação, ilustrado na figura 11 terá como entrada de dados, os dados referente ao Nível de Conhecimento do Aprendiz (NC), onde este será aferido através de uma rápida avaliação com perguntas e respostas de “certo” ou “errado”, onde o número de acertos dará margem para o percentual de conhecimento no assunto abordado nas perguntas, além do Estilo de Aprendizagem previamente informado pelo Professor responsável pela Atividade de Aprendizagem.

Figura 11 - Módulo de Adaptação



Fonte: Autor (2017)

A respeito do Objeto de Aprendizagem, o módulo de adaptação irá considerar o seu tipo de objeto, a fim de adequar ao Estilo de Aprendizagem do Aprendiz, bem como o assunto que é abordado nesse dado objeto, a fim de selecionar o objeto respectivo ao assunto que foi definido no roteiro da Atividade de Aprendizagem.

Portanto, o módulo de adaptação é representado por características pertinentes ao Objeto de Aprendizagem e ao Aprendiz, a fim de possibilitar a personalização da sua Atividade de Aprendizagem.

#### 4.2 Modelagem da Ferramenta de Autoria

Para a atividade de adaptação, são necessários dados pertinentes a criação do roteiro de aula, a saber: Número de Etapas, Assunto de cada Etapa, Perguntas e Respostas referente a cada assunto, e o Objeto Instrucional de cada Etapa (assunto) da Atividade de Aprendizagem, e as Etapas Inicial e Final, com seus objetos específicos.

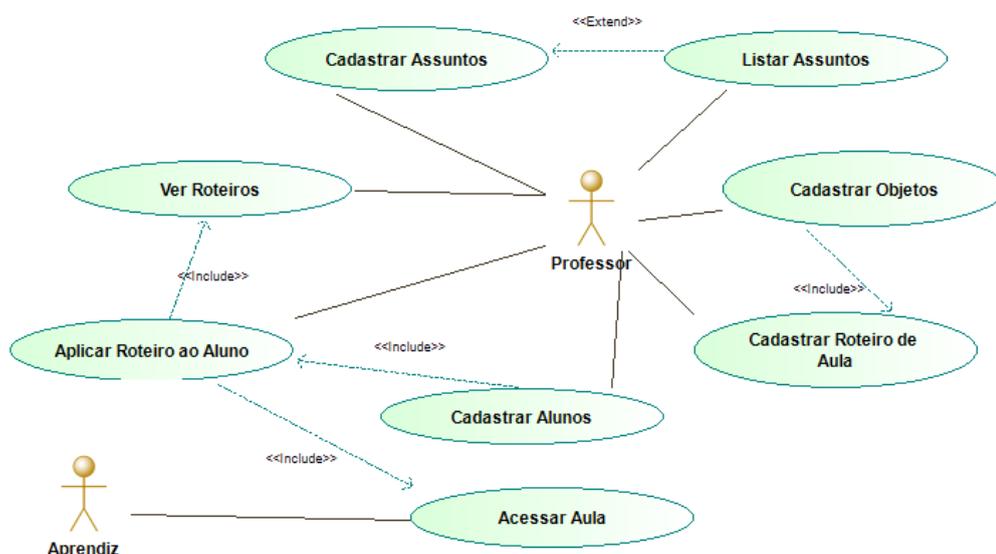
No ato de cadastro do Objeto de Aprendizagem, é necessário dar um nome ao Objeto, declarar o Tipo do Objeto (vídeo, texto, etc.), e qual o Assunto que é abordado neste Objeto que está sendo cadastrado.

Após essas entradas, o Algoritmo de Colônia de Formigas (ACO), receberá os valores referente as etapas, pois será necessário para ordenar os assuntos na Atividade de Assimilação, onde o assunto da primeira etapa será executada primeiramente, e assim sucessivamente. Através dos dados referente a Etapa, ele agrupa nós (de grafos), onde, se os nós forem de mesmo

assunto, os nós não podem se comunicar e participarão do mesmo grupo, e cada nó, nesse dado grupo, é a representação de Objetos de Tipos diferentes, ou seja, em um grupo terá objetos diversos (vídeo, áudio, texto, etc.), como explanado anteriormente.

Os requisitos da Ferramenta de Autoria segundo processo pré-3AJC são ilustrados nos diagramas de casos de usos apresentados na Figura 12, diagrama de classes na Figura 14, e a modelagem do base de dados na Figura 15, utilizada para efetuar os registros inerentes ao funcionamento da ferramenta de autoria.

Figura 12 - Diagrama de casos de uso da Ferramenta de Autoria visão Professor e Aprendiz



Fonte: Autor (2017)

O Quadro 2 a seguir apresenta um breve descrição dos casos de uso da Figura 12.

Quadro 2 – Descrição resumida dos casos de uso

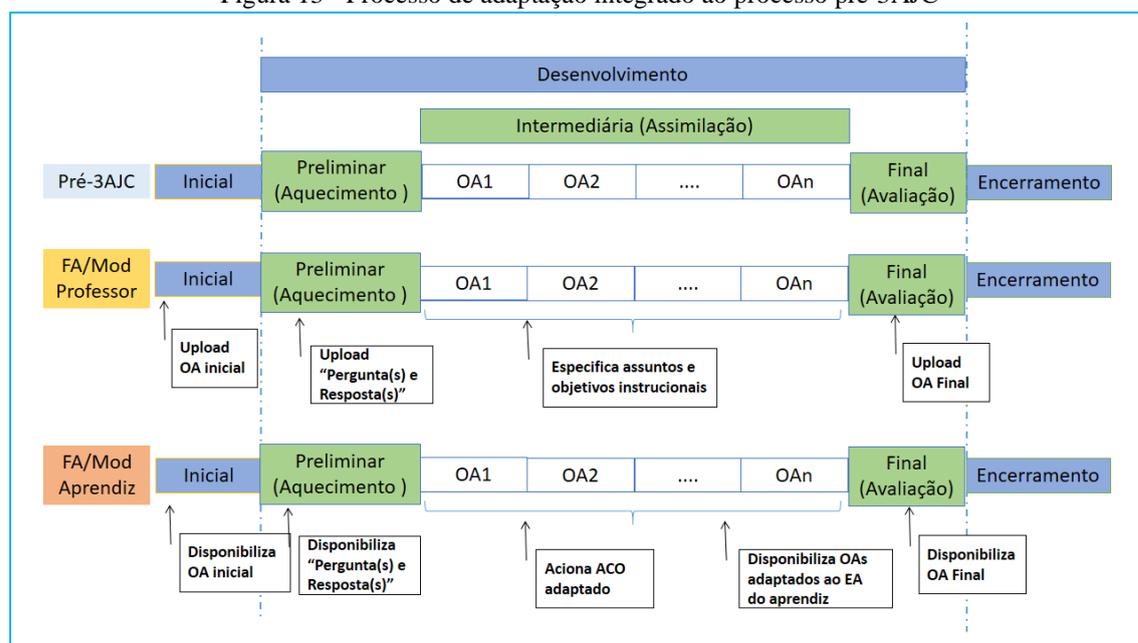
| Caso de Uso                       | Descrição   |
|-----------------------------------|---|
| Cadastrar Assuntos                | Inserir os assuntos que poderão ser abordados na aula que será feita a autoria, ficando armazenado na base de dados.  |
| Cadastrar Objetos de Aprendizagem | Inserir os objetos que poderão ser utilizados na aula que será feita a autoria, bem como seus tipos (vídeo, áudio, texto, etc.). Ficando armazenado na base de dados.   |
| Cadastrar Alunos                  | Inserir os alunos (aprendizes) que participaram da aula que será feita a autoria. Com isso, será acessado na base de dados o estilo de aprendizagem do aluno.   |
| Listar Alunos                     | Consulta os alunos (aprendizes) cadastrados.  |
| Cadastrar Roteiro de Aula         | Elabora a autoria da aula de acordo com o processo pré-3AJC, inserindo os objetos de aprendizagem pré-definidos e as informações necessárias para que a adaptação dos objetos de aprendizagem ocorra. Como pré-requisito para criação, objetos de aprendizagem e os assuntos que serão abordados. |
| Aplicar Aula ao Aluno             | Seleciona o aluno e o roteiro de aula para este aluno.  |

|              |  |
|--------------|--|
| Acessar Aula | O aluno (aprendizes) acessa a aula que foi elaborada pelo professor e realiza as ações que lhe são solicitadas. Nesse caso de uso, o algoritmo de adaptação é acionado para encontrar os conjuntos de OAs que mais se adequam ao estilo de aprendizagem do aprendiz. |
|--------------|--|

É necessário explanar que ao se integrar o processo de adaptação ao processo pré-3AJC, todas as fases da estrutura da AA foram mantidas, inclusive as subdivisões da Fase de Desenvolvimento: Aquecimento, Assimilação e Avaliação.

A Figura 13, tem-se duas visões de interação com as funcionalidades da ferramenta acordo com o Diagrama de Casos de Uso mostrados na Figura 12, a saber: Ferramenta de Autoria/Módulo Professor e Ferramenta de Autoria/Módulo Aprendiz.

Figura 13 - Processo de adaptação integrado ao processo pré-3AJC



Fonte: Autor (2017)

Como pode ser visto na Figura 13, as etapas Inicial e Encerramento, e etapas Preliminar/Aquecimento e Final/Avaliação do processo pré-3AJC correspondem as mesmas fases e etapas na Ferramenta de Autoria em ambos os Módulos Professor e Aprendiz, respeitando as funcionalidades dos casos de uso Cadastrar Roteiro de Aula e Acessar Aula, respectivamente. A etapa Preliminar/Aquecimento, no processo pré-3AJC prevê o uso de jogo como recurso para essa dinâmica. No contexto atual, será aplicado “Perguntas e Respostas” em substituição ao jogo, para sondar o nível de conhecimento do aprendiz nos assuntos a serem abordados pela atividade de aprendizagem, uma vez que não faz parte do contexto desta pesquisa considerar o jogo como objeto de aprendizagem, como explicado anteriormente.

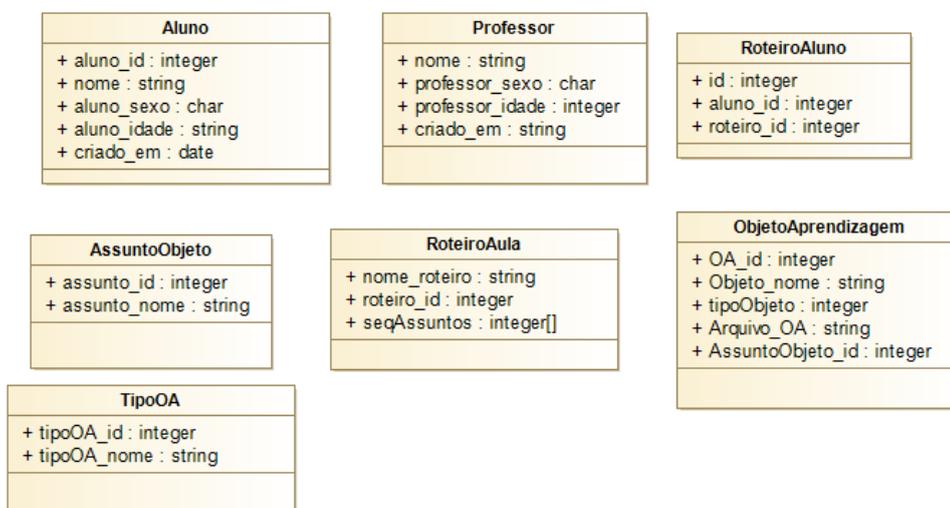
Ainda na Figura 13, a etapa Intermediária/Assimilação no pré-3AJC, que possibilita a inserção de N objetos de aprendizagem, definidos tomando como base os assuntos e objetivos instrucionais a serem trabalhados nesta atividade de aprendizagem, é correspondente à etapa em que o Professor insere os assuntos e objetivos instrucionais na Ferramenta de Autoria/Módulo Professor; e correspondente também a etapa em que o processo de adaptação através do algoritmo ACO, a partir das informações do aprendiz (nível de conhecimento e estilo de aprendizagem) e do objeto de aprendizagem (assunto, tipo-objeto, estilo aprendizagem do objeto), e com o intuito de buscar do melhor conjunto de objetos de aprendizagem que melhor se adaptam ao perfil do aprendiz.

A fase Inicial e etapa Final/Avaliação da fase de Desenvolvimento, no processo pré-3AJC, é prevista a seleção de um objeto de aprendizagem pré-definido. Nesse caso, na fase Inicial, é uma introdução sobre o assunto e como será a aula, os objetivo geral, os assuntos abordados, se terá uma avaliação no final, etc. Caso tenha uma avaliação no final, esta será o objeto de aprendizagem a ser carregado na etapa Final/Avaliação.

A fase de Encerramento, não foi contemplada na Ferramenta de Autoria, pois não foi inserida dentro do escopo da prototipação, por ser uma etapa de socialização dos resultados e requisitar o suporte de ferramentas de comunicação e colaboração.

Para construção da Ferramenta de Autoria, foram definidas características da Ferramenta, na qual podem ser representadas pelas classes e atributos das respectivas classes ilustradas na Figura 14. Onde essas classes é referente aos módulos utilizados no desenvolvimento por meio do Django.

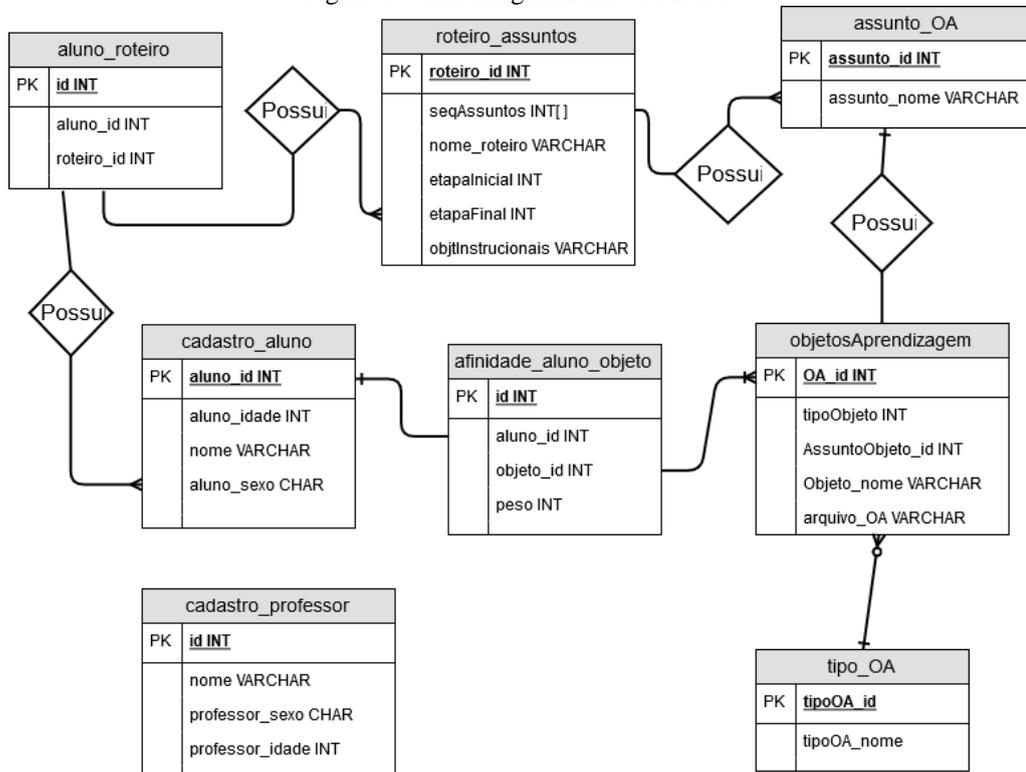
Figura 14 - Diagrama de Classes



Fonte: Autor(2017)

Para efeito de armazenamento de dados, foram construídas as tabelas na base de dados, por meio do PostgreSQL, que são ilustradas na Figura 15, onde estão dispostas as entidades, o relacionamento entre as entidades, e seus respectivos atributos e chaves primárias.

Figura 15 - Modelagem Banco de Dados



Fonte: Autor (2017)

Após essas definições e levantamentos, foi possível modelar a Ferramenta de Autoria, de forma que possibilite receber os dados inerentes ao Aprendiz e da Atividade de Aprendizagem a ser criada, sendo armazenada e, também, manipular esses respectivos dados com a utilização do Algoritmo ACO, a fim de prover as adaptações ao Aprendiz de forma que as aulas se tornem, o máximo possível, agradáveis.

### 4.3 O Algoritmo ACO Adaptado

No algoritmo de Otimização de Colônia de Formigas (ACO) para atender a realidade aqui abordada, foram necessárias algumas extensões através da definição de novos parâmetros, assim como na busca pelo melhor caminho, o qual irá colaborar na tomada de decisão sobre quais objetos em um determinado assunto de aula se adequam melhor ao aprendiz de acordo com seu Estilo de Aprendizagem.

O algoritmo ACO adaptado é apresentado a seguir:

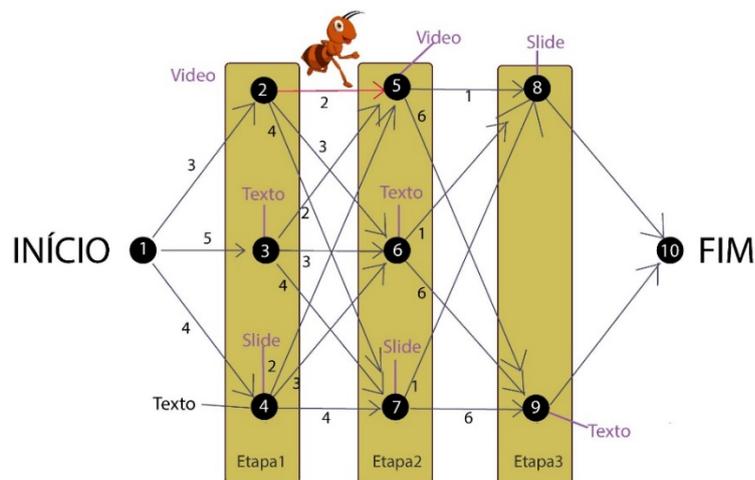
### Algoritmo ACO Adaptado

1. Inicializar Modelo Aprendiz (NC, EA\_AP)
2. Inicializar Modelo OA (tipo-objeto, assunto, EA\_OA)
3. Inicializar parâmetros do ACO (taxa evaporação, alfa, beta)
4. Inicializar Matriz de Afinidades
  - a. Para EA-AP iguais ou próximos de EA\_OA setar com menor custo
  - b. Para EA-AP diferentes de EA\_OA setar com maior custo
5. enquanto (número de iterações não finalizado) OU (solução desejada não encontrada)
  - a. calcula feromônios
  - b. para cada nó visitado até chegar no nó FINAL
    - i. calcula probabilidade para visitar o próximo nó
  - c. fim-para
  - d. aplica taxa evaporação
  - e. avalia solução encontrada
  - f. armazena solução encontrada se melhor que solução anterior
6. fim-enquanto

Para o ACO adaptado, é importante a definição da sequência de assuntos definidos pelo professor, que o aprendiz irá acessar. O algoritmo fará as buscas dos objetos (mídias) que serão utilizados pelo aprendiz com base nesses assuntos, tipos de objetos e estilos de aprendizagem do aprendiz. Como resultado, tem-se um roteiro representado por um grafo com o sequenciamento dos objetos.

Os objetos são escolhidos inicialmente, formando uma matriz de afinidades, entre aprendiz e objeto, constituída da seguinte forma: os assuntos definidos pelo professor relacionam-se aos tipos de objetos, e os tipos de objetos relacionam-se aos estilos de aprendizagem do aprendiz. A Figura 16 ilustra o grafo desses relacionamentos. A matriz de afinidade foi baseada no Modelo de adaptação desenvolvido por Silva et. al. (2016), que considera os estilos de categorizados por Felder e Silverman (1988).

Figura 16 - Representação de exemplo de busca dos objetos pelo ACO adaptado em cima do grafo criado



Fonte: Autor (2017)

Na Figura 16, é possível ilustrar que, as ‘Etapa1’, ‘Etapa2’ e a ‘Etapa3’ como etapas de Assimilação da fase de “Desenvolvimento” do modelo pré-3AJC. É perceptível as arestas do grafo, na qual possuem pesos, sendo um fator para tomada de decisão de qual nó irá, ou seja, a aresta terá o custo para chegar até o próximo nó, considerando que: menor custo, melhor opção. Nos nós são atribuídos as identificações (do objeto), na imagem indo de 1 até 10, bem como o Tipo do Objeto que representa cada nó do grafo. Considerando que: nó ‘1’ é o nó que representa a Etapa Inicial, e nó ‘10’ que representa a Etapa Final. Com essas informações, o ACO já tem o nó para iniciar a busca, utilizando os dados como: Tipo de Objeto e Peso das Arestas, ele vai traçando o caminho a ser percorrido no grafo, e quando finalizada as iterações, ele traz a melhor solução para ser aplicado ao Aprendiz.

O ACO adaptado é executado após a Etapa de Aquecimento definida no Roteiro de Aula produzido pela Ferramenta de Autoria, que a partir das iterações do aprendiz no “Perguntas e Respostas” é coletado o Nível de Conhecimento (NC) do aprendiz e repassado ao ACO adaptado. Já o EA do Aprendiz, é informado pelo Professor. O ACO é acionado após a sondagem do NC, quando o aprendiz inicia as atividades, bem como as buscas por quais objetos se adequam ao perfil do Aprendiz que está utilizando a ferramenta, conforme matriz de afinidades. Portanto, cada Aprendiz terá em sua tela o melhor roteiro disponível para se adequar ao seu estilo de aprendizagem.

Um exemplo de situação de adequação ao EA, é quando o usuário possui a capacidade mais elevada de assimilar um conteúdo através de um vídeo, ou imagem, ou slide, sendo essa característica do Aprendiz com Estilo de aprendizagem “Visual” conforme Felder e Silverman (1988) afirma, assim todo Aprendiz que tiver um perfil de EA Visual, será ofertado o OA que se adequa ao perfil do Aprendiz que possui o “EA: Visual” conforme a seleção em (Silva et. al, 2016), ou seja, será o parâmetro a ser utilizado como critério de construção do Roteiro (representado por um ‘caminho’ de grafo), a fim de traçar o melhor percurso de Ensino-Aprendizagem.

Outra característica da Ferramenta de Autoria, é agrupar diferentes Tipos de Objetos para um determinada Etapa (relacionado ao Assunto da Etapa em questão), para isso o Professor deve deixar bem definido os Assuntos(contéúdos) a serem estudados, e adicionando os objetos, previamente no repositório através da opção “Cadastrar Objetos de Aprendizagem), e o seu respectivo Tipo de Objeto, para assim, poder fazer os agrupamentos de O.A. de mesmo assunto, porém, de tipos diferentes, assim possibilitando um maior leque de opções na construção do Roteiro para o Aprendiz, a fim de possibilitar a adaptação do Roteiro a ser aplicado ao Aprendiz,

dessa forma, sendo traçado de acordo com seu Estilo de Aprendizagem, ofertando os Objetos que se adequam a mesma.

Também foi necessário definir uma Matriz de Adjacências para os nós, ou seja, uma matriz para possibilitar a construção do grafo dirigido e efetuar o agrupamento de objetos do mesmo assunto e de tipos diferentes, de forma que possibilite Roteiros diversificados, onde, como exemplo: Supondo que, um Roteiro de Aula na Etapa de Desenvolvimento onde, pode-se identificar três etapas de assuntos, aqui denominados Etapa1, Etapa2 e Etapa3, considerando que cada grupo de assuntos é relacionado com uma etapa do roteiro de aula, e cada grupo de assuntos é um assunto distinto, e conseqüentemente composto de Objetos diferentes.

- Na Etapa1, sendo equivalente a Etapa de Assimilação do pré-3AJC (Sá, 2011), e Objetos com tipo-objeto: Vídeo, Slides e Texto
- Na Etapa2 sendo equivalente a Etapa de Assimilação do pré-3AJC (Sá, 2011), Objetos com tipo-objeto: Vídeo, Slide e Texto
- Na Etapa3 sendo equivalente a Etapa de Assimilação do pré-3AJC (Sá, 2011), Objetos com tipo-objeto: Slide e Texto

Através disso, define-se um grafo, possibilitando oferecer aos aprendizes essas opções de objetos de forma que, após as adequações ao Estilo de Aprendizagem do Aprendiz, o objeto que o aprendiz tiver maior afinidade de acordo com a Tabela relacional entre Tipos de Objetos e Estilo de Aprendizagem (Silva et. al, 2016), os Objetos que se adequem ao Estilo de Aprendizagem em questão, serão os objetos com maior probabilidade de aplicações em relação aos demais.

Como outro fator, tem-se o Nível de Conhecimento (NC), onde através dele, extraído por via de um “Perguntas e Respostas” antes da inicialização do processo de Ensino-Aprendizagem, é possível definir, também, se ofertará mais um Objeto de Aprendizagem a fim de que ele fixe em um assunto, especificado no Roteiro, a fim de um tratamento diferenciado para aprimorar o nível de conhecimento no assunto em questão que apresentou dificuldade no “Perguntas e Respostas”.

Na etapa de avaliações das soluções encontradas pelo ACO (conjunto de objetos de aprendizagem), foi adotado um esquema de escalas para poder atribuir os valores de pesos na construção do Roteiro de Aula para aplicar ao Aprendiz. A escala de aproveitamento é definida com as seguintes regras que são abordadas a seguir, e que avalia a solução utilizando os EAs

(do Aprendiz e do Objeto) como parâmetro, cujos valores serão aplicados nas atualizações dos pesos referente a Afinidade do Aprendiz com o roteiro criado pela Ferramenta de Autoria.

Algoritmo Avalia solução encontrada com as regras abaixo,

1. se ((quantidade de EA\_OA dos OAs da solução encontrada = EA\_AP) => 90%)
  - a. STATUS solução encontrada = EXCELENTE
2. se (89% <= (quantidade de EA\_OA dos OAs da solução encontrada = EA\_AP) => 80%)
  - a. STATUS solução encontrada = MUITO BOM
3. se (79% <= (quantidade de EA\_OA dos OAs da solução encontrada = EA\_AP) => 70%)
  - a. STATUS solução encontrada = BOM
4. se (69% <= (quantidade de EA\_OA dos OAs da solução encontrada = EA\_AP) => 50%)
  - a. STATUS solução encontrada = REGULAR
5. se (quantidade de EA\_OA dos OAs da solução encontrada = EA\_AP) =< 49%)
  - a. STATUS solução encontrada = INSATISFATORIO
6. Se (número\_de\_Iterações chegou ao limite)
  - a. STATUS = Finaliza Execução

O algoritmo ACO adaptado foi testado em ambiente de simulação, onde as entradas são simuladas manualmente para fins de teste de funcionamento lógico, e mostrou-se funcional referente ao objetivo de gerar roteiros que possuem maior afinidade com o usuário, onde a saída é a indicação de um conjunto de Objetos de Aprendizagem, feito através dos relacionamentos entre o Estilo de Aprendizagem do Aprendiz, e os Tipos de Objetos de cada Assunto (Aplicado nas Etapas do Roteiro de Ensino-Aprendizagem), com base nas escalas citadas acima, gerando o grafo que se adequou da melhor forma possível com esses critérios.

Os valores convencionais dos parâmetros utilizados para os testes de simulação do algoritmo de ACO integrado a ferramenta de autoria são apresentados a seguir:

- O valor do feromônio, foi definido através de uma fórmula, onde este valor é atualizado no decorrer da execução do algoritmo com as seguintes fórmulas:
  - $\text{feromonio} = 1.0 / (\text{numero\_de\_vertices} * \text{custo\_guloso})$ , define feromonio da aresta =  $\text{setFeromonioAresta}(\text{chave\_aresta}_0, \text{chave\_aresta}_1, \text{feromonio})$ , onde  $\text{chave\_aresta}_n$  são nós que possuem arestas de ligação, e  $\text{setFeromonioAresta}$  é a função para atualizar o feromônio da aresta.

- informação heurística, com o custo guloso este formado: pelos custos atualizados, onde no ACO adaptado para o problema de Adaptação de Roteiros produzidos pela Ferramenta de Autoria, estes custos são atualizados baseados no Estilo de Aprendizagem do Aprendiz.
- taxa de evaporação, foi definida com valor de 0.5.
- tamanho da população, o tamanho da população inicial, para início de testes, foram definidas 2 formigas.
- probabilidade de qual aresta a formiga pode visitar (matriz de afinidades), de acordo com o custo guloso da aresta analisada, a probabilidade de escolha é maior ou menor.
- usa atualização local, que é definida por:  $\text{novo\_feromonio} = (1.0 - \text{self.evaporacao}) * \text{self.grafo.obterFeromonioAresta}(\text{aresta}[0], \text{aresta}[1]) + \text{somatorio\_feromonio}$ , onde  $\text{aresta}[ ]$  é a aresta, e a posição 0 é correspondente ao nó de origem, e a posição 1 é correspondente ao destino da aresta.
- usa atualização global, onde após a execução, é escolhido o melhor caminho, isto é, após comparar o melhor custo obtido por cada formiga, a melhor solução dentre as formigas é selecionada como melhor global, após as comparações.

#### 4.4 Prototipação da Ferramenta de autoria

Para a prototipação da FA, foi escolhida a linguagem de programação Python voltado para Web, através da utilização do *framework* Django, por conta de várias opções de bibliotecas que otimizam a implementação. E para o armazenamento dos dados, foi adotado o banco de dados PostgreSQL, por conta do seu desempenho e robustez para uma expansão futura da Ferramenta de Autoria. Todas as ferramentas citadas acima são *open-source* (ou seja, de código-fonte aberto).

Com a utilização do Django na construção da Ferramenta, é definido *templates* (modelos) de páginas para exibição, onde é possível reutilizar suas formas, a fim de agilizar a produção do sistema em formato voltado para Web, possibilitando, também, uma melhor organização para futuras manutenções e expansões com novos módulos. Todo o material a respeito das tecnologias em utilização está disponível no Apêndice.

#### 4.5 Avaliação da Ferramenta de Autoria

Como cenário de uso foi definido um universo de objetos de Aprendizagem, no qual possuem diversos tipos distintos, a fim de dar opções diferentes para adequar-se ao estilo do aprendiz que irá participar da atividade de aprendizagem (aula) com seu referido roteiro de aula.

No roteiro de aula são definidos os assuntos, que será abordados em cada etapa, nesse caso a Etapa de Assimilação. Também são definidos os objetivos instrucionais da referida etapa, além dos assuntos, com descrição breve das atividades a serem executadas.

O cenário escolhido foi a aplicação de uma aula de operações básicas de matemática, onde terá sua subdivisão entre: Apresentação de aula (Etapa Inicial), Aplicação de uma aferição rápida com Perguntas e Respostas (Etapa de Aquecimento); e a Atividade de Aprendizagem propriamente dita, onde serão definidos os assuntos, nesse caso, Adição, Subtração, Multiplicação e Divisão (Etapa de Assimilação), distribuídas nas etapas nessa mesma ordem; e finalizando com a Etapa de Encerramento, onde após sua exibição, finaliza a execução.

Para execução de teste de utilização da ferramenta, foram definidos valores a fim de verificar o resultado final da execução de acordo com as entradas de dados recebidas pela ferramenta. Esses valores estão descritos no *data set* apresentado na Tabela 1 a seguir.

Tabela 1 - Dados Iniciais

| INICIALIZAÇÃO          |  |
|------------------------|--|
| Parâmetros             | Valores  |
| Assuntos[]             | Soma,Subtração,Multiplicação,Divisão                         |
| formigas               | 2  |
| taxa de evaporação     | 0.5  |
| feromônio              | $1.0 / (\text{numero\_de\_vertices} * \text{custo\_guloso})$ |
| alfa                   | 1  |
| beta                   | 5  |
| Tipos de Objetos       | 3  |
| Estilo de Aprendizagem | Visual   |
| Nível de Conhecimento  | 50%  |
| NC < 50% (NOVA ETAPA?) | FALSO  |
| Objetos no Repositório | 30   |

Fonte: Autor (2017)

Para qualificar um dado conjunto de objetos (por meio do roteiro de aula) selecionados pelo agente do algoritmo, que no caso do ACO é a formiga, é considerada a quantidade total dos objetos de aprendizagem, e a quantidade de objetos de aprendizagem que se adequam ao perfil do aprendiz referente ao estilo de aprendizagem do mesmo, sendo assim, de acordo com o percentual de objetos, é possível classifica-los.

As classificações do conjunto de objetos, estes conjuntos gerados pelo ACO, é possível analisar sua verificação após a execução, de acordo com os dados previamente informados no data set na Tabela 1 acima. Com isso, após execuções, tem-se os comparativos descritos na Tabela 2.

Tabela 2 - Comparativos após execuções

| <b>EXECUÇÃO ACO</b>           |                              |
|-------------------------------|------------------------------|
| % de adequação OBJ_EA         | Valor                        |
| >=90% (EXCELENTE)             | FALSO                        |
| =<89% OU >=80% (MUITO BOM)    | VERDADEIRO                   |
| <=79% OU >=70% (BOM)          | FALSO                        |
| <=69% OU >= 50% (REGULAR)     | FALSO                        |
| <=49% (INSATISFATORIO)        | FALSO                        |
| <b>APÓS EXECUÇÃO</b>          |                              |
| Tipos de Objetos Selecionados | Video, Grafico, Video, Slide |

Fonte: Autor (2017)

Após essas execuções, foi possível notar a tendência da convergência, desde o início das iterações, do algoritmo ACO, para seleção de objetos que atendem ao Estilo de Aprendizagem, ou se aproxime ao Estilo de Aprendizagem do Aprendiz, de forma que as etapas, e os respectivos objetos, sejam adequados ao Estilo de Aprendizagem, quando o objeto verificado possui relação com o Estilo de Aprendizagem do Aprendiz, este objeto é atribuído um valor de custo baixo (ou seja, quanto menor o valor do custo, maior a afinidade com o Estilo de Aprendizagem), e na busca local é verificada a probabilidade de escolha do objeto, após as buscas locais, são feitas as comparações do conjunto de objetos, como pode-se verificar na Tabela 2, onde observa-se que as soluções apresentadas pelas formigas, considerando a % de adequação na faixa de 80% à 89%, logo foram executadas todas as iterações previstas, sendo considerado “Muito Bom” e disponibilizado o conjunto de objetos (Roteiro de Aula) de maior afinidade de acordo com o EA do Aprendiz, a fim de maximizar o possível para tornar a Atividade de Aprendizagem agradável e motivadora.

Inicialmente o algoritmo proposto é capaz de definir as preferências a partir do Tipo de Objeto, e estas preferências sendo armazenadas, na execução, através da utilização de matrizes, e seu armazenamento para usos futuros, através da inserção ao Banco de Dados.

Considerando a pesquisa bibliográfica Quadro 3.

Quadro 3 - Comparativo entre a pesquisa executada e os trabalhos relacionados

| Características                              | Trabalhos Relacionados                       |  |   |                                       |                                       | Pesquisa   |
|--|--|--|---|---------------------------------------|---------------------------------------|--|
|  | 3.1  | 3.2                                      | 3.3   | 3.4                                   | 3.5                                   |  |
| Técnica de Adaptação                         | Processo Estocástico(Cadeia de Markov)       | Associação entre EA e Metadados IEEE-LOM | Algoritmo PSO/Espiral de Kolb/Inteligências Múltiplas | Algoritmo ACO/LOM                     | Associação entre EA e Tipo de Objetos | Algoritmo Evolucionário (ACO)/ Associação entre EA e Tipo de Objetos |
| Parâmetros de adaptação                      | Nível de Conhecimento/Estilo de Aprendizagem | Estilo de Aprendizagem                   | Estilo de Aprendizagem /Tipos de Materiais            | Modelo de Usuário                     | Estilo de Aprendizagem                | Estilo de Aprendizagem/Tipos de Objetos/Nível de Conhecimento        |
| Integrado a Autoria (Ferramenta ou Processo) | Não  | Não                                      | Não   | Não                                   | Não                                   | Sim  |
| Algoritmo Inteligente                        | Não  | Não                                      | Sim   | Sim                                   | Não                                   | Sim  |
| Geração de O.A.                              | Não identificado na visão do Aprendiz        | Não identificado na visão do Aprendiz    | Não identificado na visão do Aprendiz                 | Não identificado na visão do Aprendiz | Não identificado na visão do Aprendiz | Sim  |

Fonte: Autor (2017)

Como é possível observar no Quadro 3 acima, levam-se em conta os seguintes aspectos: Técnica de Adaptação, na qual descreve qual abordagem foi adotada; Parâmetros de adaptação, onde são os dados extraídos do aprendiz e dos objetos de aprendizagem; Se é integrado à Autoria, sendo uma Ferramenta ou Processo; Verifica-se se há utilização de algoritmos inteligentes, como caso do PSO, ACO, etc; e por fim, se há geração de Objeto de Aprendizagem a ser disponibilizado ao aluno.

## 5 CONSIDERAÇÕES FINAIS

O ambiente educacional tem trazido muitas inovações e, também, tem mostrado novas situações adversas, dentre elas a particularidade de cada aprendiz, onde cada um apresenta forma distinta de assimilação de conteúdo. Essa heterogeneidade causa uma dificuldade para o educador, no que tange a aplicação de estratégias pedagógicas para uma melhor transmissão de conhecimento para o aprendiz. Como o caso do aprendiz que apresenta dificuldades em lidar com uma forma de material disponibilizada para sua instrução educacional, onde uns assimilam melhor com imagens, há outros que se adequam melhor a utilização de textos. Tendo em vista algumas dessas dificuldades apresentadas, pode-se aplicar algoritmos que permitam personalizar e adaptar a forma de abordagem de acordo com a particularidade de cada aprendiz, visando o aproveitamento melhor e mais produtivo, a fim de chegar ao objetivo final de assimilação do conteúdo.

Este trabalho teve como desenvolvimento de um modelo de adaptação com a utilização do algoritmo de Colônia de Formigas (ACO) como técnica para prover adaptação de conteúdo na autoria de atividades de aprendizagem baseadas no processo de pré-autoria pré-3AJC (Sá, 2011). Os conteúdos, são considerados nesta pesquisa objetos de aprendizagem que podem ser de diferentes tipos de mídias: texto, vídeo, áudio, etc. A utilização do algoritmo ACO baseado no perfil de afinidades do aprendiz propõe o melhor “caminho/roteiro” que contém o conjunto de objetos de aprendizagem mais adequados a esse perfil. Um protótipo de parte da ferramenta de autoria baseada no processo pré-3AJC foi implementado usando dados simulados dos perfis dos aprendizes e de OAs.

No presente documento foram apresentados os conceitos sobre Ferramentas de Autoria, Padrões de Aprendizagem, Sistemas Adaptativos, Trabalhos Relacionados com técnicas adaptativas, algumas com uso de algoritmos ACO e PSO, para um bom entendimento da proposta.

Como resultados de pesquisa, obteve-se o protótipo de uma ferramenta para o desenvolvimento de atividades de aprendizagem com a utilização de objetos de aprendizagem que se adaptam ao perfil do aprendiz, definidos pelo estilo de aprendizagem e nível de conhecimento. Leva-se em consideração também, características dos objetos de aprendizagem, tais como, os assuntos abordados e os tipos de objetos (vídeo, texto, etc.).

Uma contribuição deste trabalho de pesquisa tem-se a integração de um algoritmo ACO (que foi adaptado) a uma ferramenta de autoria, pois nos trabalhos relacionados que usam técnicas adaptativas, através de algoritmos de busca e otimização ACO (Araújo e Fagundes,

2011) e PSO (Moura e Fernandes, 2012) encontrados, estes são utilizados ou separadamente ou em conjunto com ambientes virtuais de aprendizagem, mas ainda não integrados a ferramentas de autoria como parte da implementação.

O resultado obtido através da execução do algoritmo ACO adaptado à situação do processo de ensino-aprendizagem baseado no pré-3AJC, tem como retorno um vetor com o conjunto de objetos que definem o sequenciamento, ou seja, o roteiro a ser percorrido por determinado aprendiz. Esse conjunto de objetos integrado a ferramenta de autoria produz um roteiro de pré-autoria que é dividido em etapas.

Quando o aprendiz tem acesso a atividade de aprendizagem (roteiro de pré-autoria/aula), é disponibilizada uma lista ordenada de objetos de aprendizagem a serem seguidos, bem como o acesso a cada objeto, de acordo com as etapas e seus respectivos assuntos determinados pelo professor. O algoritmo de adaptação (ACO adaptado), é responsável por definir quais objetos serão dispostos ao aprendiz.

Os testes realizados com o protótipo usaram de dados simulados para os OAs e para os aprendizes. Em sua maioria foram satisfatórios, encontrando o conjunto de objetos de aprendizagem mais adequados ao estilo de aprendizagem do aprendiz.

Como sugestão sobre possíveis trabalhos futuros, tem-se:

- a) Suportar o objeto de aprendizagem do tipo jogo;
- b) Contemplar a fase de Encerramento do processo pré-3AJC;
- c) Inserir a avaliação realizada na etapa Final/Avaliação como parâmetro para o ACO atualizar as informações do aprendiz;
- d) a implementação de computação paralela a fim de acelerar o processamento, bem como a exportação das saídas resultantes para AVAs existentes no mercado.

## REFERÊNCIAS

- ARAÚJO, Lucas Moreno, FAGUNDES, Fabiano. AntStudy: Proposta de definição de sequências de estudo utilizando Ant System. In: **ENCONTRO DE COMPUTAÇÃO E INFORMÁTICA DO TOCANTINS**, 13., 2011, Palmas. Anais... Palmas: CEULP/ULBRA, 2011. p. 98-108.
- BARBOSA, Leandro Zavarez. **Técnicas de otimização baseadas no paradigma de enxames de partículas e sua aplicação ao projeto de equipamentos eletromagnéticos**. 2012. Dissertação (Mestrado em Engenharia Elétrica). Escola Politécnica da Universidade de São Paulo, São Paulo.
- BARKER, Phil; CAMPBELL, Lorna M.; ROBERTS, Anthony; SMYTHE, Colin. **Resolves the drift between the IMS Meta-data and the IEEE LOM standard, which began as a combination of IMS Meta-data and ARIADNE collaboration**. 2006. Disponível em: <[https://www.imsglobal.org/metadata/mdv1p3/imsmd\\_bestv1p3.html](https://www.imsglobal.org/metadata/mdv1p3/imsmd_bestv1p3.html)>. Acesso em 16 de novembro de 2015.
- BLOOM, B. e KRATHWOHL, D. **Taxonomy of educational objectives: The classification of educational goals. Handbook I: Cognitive domain**. pp. 64–81, 1956.
- BLOOM, Benjamin S. et al. **Taxionomia de objetivos educacionais**. 6 ed. Porto Alegre. 1977.
- BOARETTO, Rogério; NUNES, César A. A.; FILATRO, Andrea. Representação de Uma Ação de Aprendizagem Através do IMS-Learning Design e Implicações para o Desenvolvimento de LMSs. In: **Anais do 4º Seminário Nacional ABED de EAD**. 2006. Disponível em: <<http://www.abed.org.br/seminario2006/pdf/tc043.pdf>>. Acesso em: 18 de outubro de 2015.
- BREVE, Fabrício Aparecido. **Otimização por enxame de partículas (PSO) e otimização por colônias de formigas (ASO) aplicadas ao problema do caixeiro viajante (TSP)**. Universidade de São Paulo, São Paulo, 2007. Disponível em: <<https://www.fabriciobreve.com/trabalhos/compbio3.pdf>>. Acesso em 20 de janeiro de 2017.
- CUNHA, Andressa Caroline Portes da; EMER, Maria Claudia Figueiredo Pereira; AMARAL, Marília Abrahão. Modelo de mediação para hiperfídia adaptativa em ambiente virtual de

- aprendizagem. **Revista Ciências & Cognição v.20 n. 2**. 2015. Disponível em: <<http://www.cienciasecognicao.org/revista/index.php/cec/article/view/1008>>. Acesso em 1 de março de 2016.
- DORÇA, Fabiano Azevedo; LIMA, Luciano Vieira; FERNANDES, Márcia Aparecida; LOPES, Carlos Roberto. Detecção e Correção Automática de Estilos de Aprendizagem em Sistemas Adaptativos para Educação. **Revista de Informática Teórica e Aplicada v.18 n.2**. 2011. Disponível em: <[http://www.seer.ufrgs.br/rita/articleb/view/rita\\_v18\\_n2\\_p178](http://www.seer.ufrgs.br/rita/articleb/view/rita_v18_n2_p178)>. Acesso em: 16 de fevereiro de 2016.
- Dorigo, M.; Maniezzo, V.; Colomi, A. (1991). Positive feedback as a search strategy. Dipartimento di Elettronica, Politecnico di Milano, Italy, **Technical Report TR 91-016**.
- Dorigo, M.; Maniezzo, V.; Colomi, A. (1996). Ant system: optimization by a colony of cooperating agents. **IEEE Transactions on Systems, Man and Cybernetics, Part B**, Vol. 26, pp. 29–41.
- DRÉO, J.; PÉTROWSKI, A.; SIARRY, P.; TAILLARD, E. **Metaheuristics for hard optimization**. Springer: Heidelberg, 2006.
- EIBEN, A. . E.; SMITH, J. E. **Introduction to Evolutionary Computing**. Springer. 2003.
- FARIAS, Adelito Borba; LEITE, Lucas Sampaio; GOMES, Tancicleide Carina; FIGUEIREDO, Bruno Nascimento; XAVIER, Mariana Pereira. Potencialização da Educação a Distância com Sistemas Hipermídia Adaptativos: uma Proposta Baseada em Tipos de Personalidade Acadêmica e Recomendação de Conteúdo. Em: **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**. 2014. Disponível em: <<http://www.br-ie.org/pub/index.php/wcbie/article/view/3222>>. Acesso em 10 de dezembro de 2015.
- Felder, R., Silverman, L. (1988). Learning and teaching styles in engineering education. **Journal of Engineering education**, 78(7), 674–681
- FERRAZ, A. P. C. M. e BELHOT, Renato Vairo. Taxionomia de Bloom: revisão teórica e apresentação das adequações do instrumento para definição de objetivos instrucionais. **Gest. Prod.**, São Carlos, v. 17, n. 2, p. 421-431, 2010.

FERNANDES, Clovis Torres; SILVA JÚNIOR, Aladir F.; VARCHAVSKY, Márcio. Editor LD: Uma Ferramenta de Apoio à Autoria de Atividades de Aprendizagem no Padrão IMS Learning Design. **II APPLETS - Applications to Provide Learning and Teaching Support**. 2012. Disponível em: <<http://www.br-ie.org/pub/index.php/wcbie/article/view/1881/1646>>. Acesso em 5 de setembro de 2015.

FILATRO, A. **Learning Design como fundamentação teórica pratica para design instrucional contextualizado**. Tese (Doutorado). Programa de Pós-Graduação em Educação da Universidade de São Paulo. São Paulo, 2008. Disponível em: <[www.teses.usp.br/teses/disponiveis/48/48134/tde-12062008-142556/](http://www.teses.usp.br/teses/disponiveis/48/48134/tde-12062008-142556/)>. Acesso em 10 de setembro de 2015.

GABRIEL, Paulo H. R.; MELOY, Vinícius V. de; DELBEM, Alexandre C. B. Algoritmos Evolutivos e Modelo HP para Predição de Estruturas de Proteínas. **Revista Controle & Automação**. Vol.23 no.1. 2012. Disponível em: <[www.scielo.br/pdf/ca/v23n1/a03v23n1.pdf](http://www.scielo.br/pdf/ca/v23n1/a03v23n1.pdf)>. Acesso em: 23 de Março de 2016.

GOLDBARG, Marco César; GOLDBARG, Elizabeth Ferreira Gouvêa; MEDEIROS NETO, Francisco Dantas de. Algoritmos evolucionários na determinação da configuração de custo mínimo de sistemas de co-geração de energia com base no gás natural. **Pesqui. Oper.**, Rio de Janeiro, v. 25, n. 2, p. 231-259, Aug. 2005. Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S0101-74382005000200005&lng=en&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0101-74382005000200005&lng=en&nrm=iso)>. Acesso em: 27 de março de 2016.

IMS-LD. **IMS Learning Design Information Model**. 2003. Disponível em: <[https://www.imsglobal.org/learningdesign/ldv1p0/imslld\\_infov1p0.html](https://www.imsglobal.org/learningdesign/ldv1p0/imslld_infov1p0.html)>. Acesso em 14 de setembro de 2015.

ITO, G. C.; FERREIRA, M. G.; SANT'ANNA, Nilson. Uma ferramenta para geração de interfaces adaptativas. In: **CONFERÊNCIA IBERO AMERICANA WWW/INTERNET**. Disponível em: <[http://www.simpep.feb.unesp.br/anais/anais\\_13/artigos/1207.pdf](http://www.simpep.feb.unesp.br/anais/anais_13/artigos/1207.pdf)>. Acesso em: 15 de dezembro de 2015.

KRATHWOHL, D. A revision of Bloom's taxonomy: An overview. **Theory into practice**. vol. 41, no. 4, pp. 212–218, 2002.

- Leffa, V. J. Uma ferramenta de autoria para o professor: o que é e o que faz, **Em Letras de Hoje**. 2006. Páginas 189-214, PUCRS
- MARTINEZ, Carlos Antonio Echeverria. **Calibração Automática de Modelo Hidrológico com Algoritmo Baseado no Comportamento de Formigas**. Fev. 2014. Dissertação (Programa de Pós-Graduação em Recursos Hídricos e Saneamento Ambiental). Universidade Federal do Rio Grande do Sul, Porto Alegre.
- MARZANO, R. e KENDALL, J. **The new taxonomy of educational objectives**. Corwin Pr, 2007.
- MOURA, Fábio F. de; FERNANDES, Márcia A. Uma proposta de um modelo computacional que usa PSO para a escolha de Objetos de Aprendizagem baseado na Espiral de Kolb e nas Inteligências Múltiplas. **Anais do 23º Simpósio Brasileiro de Informática na Educação (SBIE)**. 2012. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/sbie/2012/0080.pdf>>. Acesso em: 21 de janeiro de 2016.
- NETO, R.F.T. **Proposta de solução de problemas scheduling considerando a possibilidade de terceirização usando a técnica de otimização por colônia de formigas**. Tese de Doutorado apresentada ao Programa de Pós-Graduação em Engenharia de Produção da Universidade de São Carlos. 2010.
- Nienow, A. L. e Bez, M. R. 2009 . Ferramenta de autoria para construção de objetos de aprendizagem para a área da saúde, Em **Anais do VIII Seminário de Informática - RS** (Seminfo 2009), Ulbra.
- NOLÊTO, Carleandro O.; VIANA, Windson; TRINTA, Fernando A.M.. **Uma Ferramenta de Autoria para o Desenvolvimento de Jogos Pervasivos Baseados em Realidade Aumentada**. 2014. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/wtdwebmedia/2014/005.pdf>>. Acesso em 20 de janeiro de 2016.
- OLIVEIRA, Rodrigo Guerra de; et al. Problematização como método ativo de ensino-aprendizagem em um Curso de Odontologia. **Revista da ABENO**, v. 15, n. 2, p. 74-81, 2015. Disponível em <<http://revabeno.emnuvens.com.br/revabeno/article/view/180>>. Acesso em 13 de janeiro de 2016.

- OLIVEIRA, Erica R.; NELSON, Maria Augusta V.; ISHITANI, Lucila. Ciclo de vida de objetos de aprendizagem baseado no padrão SCORM. **Workshop em Informática na Educação (SBIE)**. 2007. Disponível em: <<http://www.br-ie.org/pub/index.php/sbie/article/download/571/557>>. Acesso em: 27 de outubro de 2015.
- PRATA, Bruno de Athayde. Um algoritmo enxame de partículas para uma variante do problema de máxima cobertura. **Revista GEPROS**. 2012. Disponível em: <<http://revista.feb.unesp.br/index.php/gepros/article/view/417>>. Acesso em: 25 de março de 2016.
- RESENDE, Daniel Teixeira; DORÇA, Fabiano Azevedo; CATTELAN, Renan G.; ARAÚJO, Rafael D.. Em direção à recuperação automática de objetos de aprendizagem em repositórios através da associação dos estilos de aprendizagem de estudantes com metadados no padrão IEEE-LOM. **Anais dos Workshops do Congresso Brasileiro de Informática na Educação (CBIE)**. 2014. Disponível em: <<http://www.br-ie.org/pub/index.php/wcbie/article/view/3265>>. Acesso em: 04 de março de 2016.
- Serapião, Adriane Beatriz de Souza. FUNDAMENTOS DE OTIMIZAÇÃO POR INTELIGÊNCIA DE ENXAMES: UMA VISÃO GERAL. **Revista Controle & Automação**. Vol.20 no.3/Julho, Agosto e Setembro. 2009.
- SÁ, Eveline de Jesus Viana. **Processo de pré-Autoria para Atividades de Aprendizagem com Uso de Jogos e Cooperação**. 2011. Tese (Doutorado em Engenharia Eletrônica e Computação). Instituto Tecnológico da Aeronáutica, São José dos Campos.
- \_\_\_\_\_, TEIXEIRA, Jeane S. F.; FERNANDES, Clovis Torres. Uso de Jogos Cooperativos em Atividades de Aquecimento em Cursos Hipermedia Adaptativos. 2007. **WIE'2007 – Workshop de Informática na Educação**, Rio de Janeiro.
- SANTANA FILHO, Antônio Ferreira; XAVIER, Neila Batista. Objetos de Aprendizagem: análise crítica, aplicabilidade e estudo de caso utilizando o software *Reload Learning Design*. **Revista IGAPÓ**. 2011. Disponível em: <<http://200.129.168.183/ojs/index.php/igapo/article/view/235>>. Acesso em: 25 de outubro de 2015.
- SILVA, D. H.; DORÇA, F. A. (2014). Uma Abordagem Automática para Personalização do Processo de Ensino Baseada em Estilos de Aprendizagem em Sistemas Adaptativos e Inteligentes para Educação a Distância. **Revista Brasileira de Informática na Educação (RBIE)**, v. 22, n. 2, p.1–15.

- SILVA, José Cláudio Moreira da. **O Uso do Software de Autoria JCLIC como Ferramenta Pedagógica no Processo de Ensino-Aprendizagem de Conteúdos da Matemática nas Séries Finais do Ensino Fundamental por Meio da Construção de Jogos Educativos**. 2013. Dissertação (Mestrado Profissional em Ensino de Ciências e Matemática) Universidade Federal do Ceará, Fortaleza. Disponível em: <[http://www.repositorio.ufc.br/bitstream/riufc/7324/1/2013\\_dis\\_jcmsilva.pdf](http://www.repositorio.ufc.br/bitstream/riufc/7324/1/2013_dis_jcmsilva.pdf)>. Acesso em: 10 de fevereiro de 2016.
- SILVA, Luiz Augusto Matos da; BARRETO, Luciano Porto. Interoperabilidade de Unidades de Aprendizagem do IMS Learning Design em Ambientes Virtuais de Aprendizagem. Em: **SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO - SBIE**. 2009. Disponível em: <[http://www.proativa.virtual.ufc.br/sbie/CD\\_ROM\\_COMPLETO/sbie\\_artigos\\_completo/Interoperabilidade%20de%20Unidades%20de%20Aprendizagem%20do%20IMS.pdf](http://www.proativa.virtual.ufc.br/sbie/CD_ROM_COMPLETO/sbie_artigos_completo/Interoperabilidade%20de%20Unidades%20de%20Aprendizagem%20do%20IMS.pdf)>. Acesso em: 16 de novembro de 2015.
- SILVA, Zenaide Carvalho da; FERREIRA, Leandro Rodrigues; e PIMENTEL, Andrey Ricardo. 2016. Modelo de Apresentação Adaptativa de Objeto de Aprendizagem baseada em Estilos de Aprendizagem. **Anais do XXVII Simpósio Brasileiro de Informática na Educação (SBIE 2016) V Congresso Brasileiro de Informática na Educação (CBIE 2016)**.
- SILVA FILHO, Vasco Pinto da. **Ferramenta de Autoria Multimídia para Aprendizado em Língua Estrangeira**. 2000. Dissertação (Mestrado em Ciência da Computação) - Universidade Federal de Santa Catarina, Florianópolis. Disponível em: <<https://repositorio.ufsc.br/handle/123456789/82096>>. Acesso em: 21 de outubro de 2015.
- SILVEIRA, Sidnei Renato. **Estudo de uma Ferramenta de Autoria Multimídia para a Elaboração de Jogos Educativos**. 1999. 128f. Dissertação (Mestrado em Ciência da Computação). Universidade Federal do Rio Grande do Sul, Porto Alegre.
- STARR, C.; MANARIS, B. e STALVEY, R. “Bloom’s taxonomy revisited: specifying assessable learning objectives in computer science,” *ACM SIGCSE Bulletin*, vol. 40, no. 1, pp. 261–265, 2008.
- TAVARES Neto, Roberto Fernandes; GODINHO Filho, Moacir. Otimização por colônia de formigas para o problema de sequenciamento de tarefas em uma única máquina com

terceirização permitida. **Gest. Prod.**, São Carlos, v. 20, n. 1, p. 76-86, Mar. 2013. Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S0104-530X2013000100006&lng=en&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0104-530X2013000100006&lng=en&nrm=iso)>. Acesso em: 16 de janeiro de 2016.

TEIXEIRA, J. S. F. ; SA, E. J. ; FERNANDES, C. T. . Busca e Recuperação de Jogos Educacionais: Uma Ontologia Representativa do Contexto de Aprendizagem. In: **SBIE 2013/WSWED 2013/V Workshop Brasileiro de Web SEMÂNTICA e Educação**. 2013. Campinas. Anais do SBIE 2013. Porto Alegre: SBC (Sociedade Brasileira de Computação), 2013.

**W3C. Browsers and Authoring Tools**. Disponível em:

<<https://www.w3.org/standards/agents/authoring>>. Acessado em 19 de outubro de 2015.

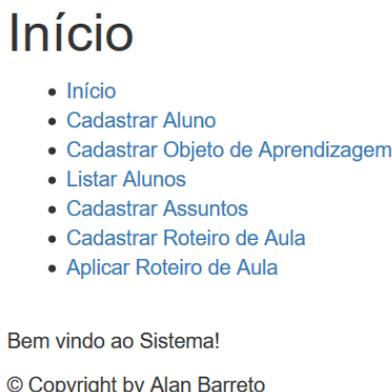
XU, Qi; MAO, Jun; e JIN, Zhihong. Simulated Annealing-Based Ant Colony Algorithm for Tugboat Scheduling Optimization. **Mathematical Problems in Engineering**. 2012. Disponível em: <<https://www.hindawi.com/journals/mpe/2012/246978/fig3/>>. Acesso em 02 de abril de 2017.

## **APÊNDICES**

## APÊNDICE A – Manual do Usuário

A tela Inicial do Professor, foi construída da forma ilustrada na Figura 17, onde dispõe de todas as opções que poderá ser utilizada.

Figura 17 - Tela inicial para o Professor



Fonte: Autor (2017)

A Figura 18 ilustra a tela referente ao caso de uso Cadastrar Aluno, onde os dados inseridos serão armazenados no banco de dados, para fins de identificação (Nome), bem como, a produção de estatísticas baseado em cada característica extraída através destas entradas de dados (Idade e Sexo), tanto quanto a entrada do Estilo de Aprendizagem (EA), esta informada pelo Professor, para detecção de suas afinidades com os Objetos durante a decorrência de uma Atividade de Aprendizagem, aplicados no Roteiro da Atividade de Aprendizagem.

Figura 18 - Tela apresentando o módulo de Cadastro de Alunos



Fonte: Autor (2017)

A Figura 19 ilustra a tela referente ao caso de uso Cadastrar Roteiro Aula, onde o Professor especifica os Assuntos a serem abordados em cada etapa do processo de Ensino Aprendizagem, bem como “Perguntas e Respostas” que serão aplicados ao Aprendiz a fim de sondar o seu respectivo NC, onde a quantidade de “Perguntas e Respostas” é baseada no número de etapas, ou seja, serão duas “Perguntas e Respostas” para cada Assunto abordado, e de acordo com o NC extraído, determinar se necessita, ou não, de um objeto extra em uma determinada Etapa, e posteriormente com essas informações, o ACO buscar os objetos desses assuntos e aplicar na Etapa específica.

Figura 19 - Tela para Criar Roteiro de Aula

## Criar Roteiro de Aula

- Início
- Cadastrar Aluno
- Cadastrar Professor
- Cadastrar Objeto de Aprendizagem
- Listar Alunos
- Cadastrar Assuntos
- Cadastrar Roteiro de Aula
- Aplicar Roteiro de Aula
- Cadastrar Aula

Nesta seção, você escolhe os assuntos e suas respectivas etapas, escolhendo, também, o número de etapas necessárias. Lembrando que:

- A Etapa Inicial é a etapa de introdução, logo você deve escolher um arquivo para este tipo de situação;
- A Etapa Final é a etapa de finalização, podendo executar posteriormente (ou nesta etapa) uma avaliação, logo você deve escolher um arquivo para este tipo de situação.

**Nome roteiro:**

**Etapa Inicial:**

**Quiz (1) Etapa Aquecimento:**

**Quiz (2) Etapa Aquecimento:**

**Resposta para Quiz(1) Etapa Aquecimento:**

**Resposta para Quiz(2) Etapa Aquecimento:**

**Assunto Etapa (1) Assimilação:**

**Objetivo Instrucional Etapa (1) Assimilação:**

**Assunto Etapa (2) Assimilação:**

**Objetivo Instrucional Etapa (2) Assimilação:**

**Etapa Final:**

© Copyright by Alan Barreto

Fonte: Autor (2017).

A Figura 20 ilustra a tela referente ao Cadastro de Assuntos, onde o Professor deverá informar o nome do Assunto, e este será armazenado na base de dados da Ferramenta de Autoria.

Figura 20 - Tela para Cadastro de Assuntos

## Cadastro de Assuntos

- Início
- Cadastrar Aluno
- Cadastrar Professor
- Cadastrar Objeto de Aprendizagem
- Listar Alunos
- Cadastrar Assuntos
- Cadastrar Roteiro de Aula
- Aplicar Roteiro de Aula
- Cadastrar Aula

Nome:

© Copyright by Alan Barreto

Fonte: Autor (2017).

Na Figura 21, é ilustrada a tela que representa o caso de Uso para Cadastrar Objetos de Aprendizagem, onde nesta opção, tem como entradas: Nome, que será atribuído à um Objeto para fácil identificação; Tipo de Objeto, onde terá as opções como, 'Vídeo, Texto, Slide e Áudio', a opção para o envio do arquivo, onde este será armazenado no repositório da Ferramenta de Autoria, e por último o Assunto Abordado, que é referente ao assunto abordado no objeto, lembrando que, o Assunto deve ser cadastrado antes para aparecer como opção na hora de escolher o assunto do Objeto, conforme a Figura 20.

Figura 21 - Tela do caso de Uso de Cadastrar Objetos de Aprendizagem

## Cadastro de Objeto de Aprendizagem

- Início
- Cadastrar Aluno
- Cadastrar Professor
- Cadastrar Objeto de Aprendizagem
- Listar Alunos
- Cadastrar Assuntos
- Cadastrar Roteiro de Aula
- Aplicar Roteiro de Aula

Nome:

Tipo Objeto:

Selecione o arquivo a ser adicionado:

Nenhum arquivo selecionado.

Assunto Abordado:

© Copyright by Alan Barreto

Fonte: Autor (2017).

Para a tela Inicial da Ferramenta de Autoria para o Aprendiz, a mesma é ilustrada na Figura 22, onde é disposto as opções para o Aprendiz, sendo este o Caso de Uso (Acessar Aula).

Figura 22 - Tela Inicial para o Aprendiz

# Início

- [Acessar Aula](#)

Bem vindo ao Sistema!

© Copyright by Alan Barreto

Fonte: Autor (2017).

Em relação ao caso de uso Acessar Aula, esta é subdividida em duas telas, onde na Figura 23, é ilustrado a primeira tela referente às aulas disponíveis para acesso do Aprendiz.

Figura 23 - Tela para Acessar Aula

## Acessar Aula

- [Início](#)
- [Acessar Aula](#)

### Aulas Disponíveis

| Aula              | Acesar              |
|-------------------|---------------------|
| Operações Básicas | <a href="#">Ver</a> |

© Copyright by Alan Barreto

Fonte: Autor (2017).

A segunda tela, ainda do caso de uso para Acessar Aula, é ilustrada na Figura 24, onde é disposto todo o conteúdo a ser abordado no Roteiro de Aula, subdividido em Assuntos para cada Etapa do processo de Ensino-Aprendizagem.

Figura 24 - Tela da segunda parte do caso de Uso Acessar Aula

## Acessar Aula

- Início
- Acessar Aula

### Operações Básicas

| Assuntos   | Acessar             |
|------------|---------------------|
| Introdução | <a href="#">Ver</a> |
| Soma       | <a href="#">Ver</a> |
| Subtração  | <a href="#">Ver</a> |
| Final      | <a href="#">Ver</a> |

© Copyright by Alan Barreto

Fonte: Autor(2017).

Bem como implementado o algoritmo ACO, a fim de servir como módulo para a execução da adaptação no protótipo de ferramenta de autoria, afim de mostrar a otimização da adaptação de acordo com o estilo de aprendizagem do aprendiz (aluno), com o intuito de tornar a aula mais atraente no ponto de vista do aprendiz (aluno).

A implementação dessas desses casos de uso foi desenvolvida com uso do Framework Django, onde este possibilita integrar com a linguagem de Programação Python (até mesmo o Framework foi baseado no Python), onde tornou-se possível disponibiliza-la através da forma Web (hospedagem na internet) em servidores locais, possibilitando o acesso de qualquer navegador de internet (browser).

### APENDICE B - Codificação

O algoritmo ACO apresentado na seção 4.2 foi integrado ao protótipo da ferramenta de autoria. A seguir são descritos alguns aspectos principais de implementação relacionados a essa integração e as questões de adaptação definidas no próprio algoritmo.

Na Figura 25, é demonstrada a implementação da Classe Aresta, que possui a representação da aresta do grafo, com suas devidas características, onde tem-se o nó de origem do grafo a ser percorrido e o nó destino. Com isso, é possível representar a estrutura que a formiga irá percorrer.

Figura 25 - Classe Aresta

```

# encoding:utf-8

import random, math
CONST_CUSTO =1
# classe que representa uma aresta
class Aresta:

    def __init__(self, origem, destino, custo):
        self.origem = origem
        self.destino = destino
        self.custo = custo
        # self.feromonio = None

    def __init__(self, origem, destino, custo=CONST_CUSTO):
        self.origem = origem
        self.destino = destino
        self.custo = custo
        # self.feromonio = None

    def obterOrigem(self):
        return self.origem

    def obterDestino(self):
        return self.destino

    def obterCusto(self):
        return self.custo

    #def obterFeronomio(self):
    #    return self.feromonio

    #def setFeromonio(self, feromonio):
    #    self.feromonio = feromonio

```

Fonte: Autor(2017)

Como pode ser observado na Figura 26, a classe Grafo foi definida para representar a estrutura do grafo, onde tem-se o número de vértices como parâmetro do método construtor, um dicionário com as arestas, ou seja, onde será adicionado as instancias das arestas representadas na Figura 25. No atributo “self.vizinhos”, ficará armazenado todos os nós vizinhos de um determinado nó, a fim de construir as possíveis combinações de caminho a ser percorrido a partir desse determinado nó.

Figura 26 - Classe Grafo

```

class Grafo:

    def __init__(self, num_vertices):
        self.num_vertices = num_vertices # número de vértices do grafo
        self.arestas = {} # dicionário com as arestas
        self.vertice = {}
        self.arestas.items()
        self.vizinhos = {} # dicionário com todos os vizinhos de cada vértice
        self.arestas.items()

    def adicionarAresta(self, origem, destino, custo=CONST_CUSTO):

        aresta = Aresta(origem=origem, destino=destino, custo=custo)

        self.arestas[(origem, destino)] = aresta
        if origem not in self.vizinhos:
            self.vizinhos[origem] = [destino]

        else:

            self.vizinhos[origem].append(destino)

    def obterCustoAresta(self, origem, destino):
        return self.arestas[(origem, destino)].obterCusto()

```

Fonte: Autor (2017)

Na Figura 26 também é demonstrada a definição do método para adicionar arestas ao grafo, aqui denominada “adicionarAresta”, tendo como parâmetros os nós de origem e destino, bem como o custo que terá em cima dessa aresta. Esse custo, é considerado um parâmetro para a tomada de decisão para qual nó (destino) a formiga irá.

E por último, na Figura 26, foi definido o método “obterCustoAresta”, devido a necessidade de obter o custo da aresta para definir os pesos de afinidade, este retorna o valor de custo atual na aresta quando chamado esse método durante a execução do algoritmo, a fim de cálculo de probabilidade de escolha.

Na Figura 27, é demonstrada a Classe “Formiga”, onde está ilustra a estrutura definida para a representação da formiga que será aplicada como agente de decisões durante a execução do algoritmo.

Figura 27 - Classe Formiga

```

class Formiga:

    def __init__(self, objetoAprendizagem):
        self.objetoAprendizagem = objetoAprendizagem
        self.solucao = []
        self.custo = None

    def obterObjetoAprendizagem(self):
        return self.objetoAprendizagem

    def setObjetoAprendizagem(self, objetoAprendizagem):
        self.objetoAprendizagem = objetoAprendizagem

    def obterSolucao(self):
        return self.solucao

    def setSolucao(self, solucao, custo):
        # atualiza a solução
        if not self.custo:
            self.solucao = solucao[:]
            self.custo = custo
        else:
            if custo < self.custo:
                self.solucao = solucao[:]
                self.custo = custo

    def obterCustoSolucao(self):
        return self.custo

```

Fonte: Autor (2017)

Os atributos principais da Classe Formiga, estão definidas dentro do método construtor da classe, onde estes são:

“self.objetoAprendizagem”, este atributo é o nó onde a formiga estará presente, ou seja, um nó do grafo representa cada Objeto de Aprendizagem adicionado ao grafo que será percorrido durante o processo de aprendizagem;

“self.solucao”, este atributo se refere a solução, até então, construída pela formiga durante a execução do algoritmo, a fim de, no final, apresentar sua solução local, e posteriormente ser comparada, e caso satisfaça a condição, como solução global;

“self.custo”, este atributo se refere ao custo que a formiga está tendo até a posição que ela está, esse dado é importante para a construção do caminho a ser percorrido no grafo durante a execução do algoritmo.

A classe ACO está representada na Figura 28, onde todos os seus atributos são definidos no método construtor. Dentre os atributos definidos, temos:

“self.grafo”, onde este se refere ao grafo construído para ser percorrido;

“self.num\_formigas”, que se refere ao número de formigas a serem utilizadas, ou seja, o número de agentes de tomada de decisão;

“self.tamanhoMatVert”, que é o tamanho da matriz de vértices que será construída, a fim de possibilitar a construção de outro dado durante a execução, como a matriz de adjacências;

“self.alfa”, que é a importância do valor do feromônio;

“self.beta”, que é a definição heurística a ser utilizada;

“self.iteracoes”, o número de iterações a ser executada;

“self.evaporacao”, que é a taxa de evaporação do feromônio;

“self.formigas”, que serão as formigas a serem aplicadas no grafo a fim de encontrar o melhor caminho para as afinidades do aprendiz; e,

“self.visitados”, que são os nós que já foram visitados pelos agentes aplicado.

Figura 28 - Classe ACO

```
# classe do ACO
class ACO:
    def __init__(self, grafo, num_formigas,tamanhoMatVert, alfa=1.0, beta=5.0,iteracoes=10, evaporacao=0.5):
        self.grafo = grafo
        self.num_formigas = num_formigas
        self.tamanhoMatVert = tamanhoMatVert
        self.alfa = alfa # importância do feromônio
        self.beta = beta # importância da informação heurística
        self.iteracoes = iteracoes # quantidade de iterações
        self.evaporacao = evaporacao # taxa de evaporação
        self.formigas = [] # lista de formigas
        self.visitados = []
```

Fonte: Autor (2017)

Na classe ACO, também são efetuadas várias instruções, bem como definidos métodos que serão utilizadas nela. Como poderemos ver a seguir.

Nesta etapa na Figura 29, é definido uma variável chamada “custo\_guloso”, onde está irá adicionar a soma dos custos “mínimos” de cada nó, ou seja, “min\_custo”, e a partir disso, será acumulado os valores, somando todos os mínimos de cada nó percorrido durante a execução.

Figura 29 - Trecho aplicado para definição do caminho a ser percorrido

```

custo_guloso = 0.0 # custo guloso
#vertice_inicial = random.randint(1, grafo.num_vertices) # seleciona um vértice aleatório

vertice_inicial = 1 # inicia no vertice 1
vertice_corrente = vertice_inicial
visitados = [vertice_corrente] # lista de visitados
while vertice_corrente!=self.tamanhoMatVert:
    if self.grafo.vizinhos[vertice_corrente] == '':
        break
    vizinhos = self.grafo.vizinhos[vertice_corrente][:]

    #print "VIZ",vizinhos
    custos, escolhidos = [], {}
    for vizinho in vizinhos:
        if vizinho not in visitados:
            custo = self.grafo.obterCustoAresta(vertice_corrente, vizinho)
            escolhidos[custo] = vizinho
            custos.append(custo)

    #if len(visitados) == self.grafo.num_vertices:
    #    break
    #if self.grafo.vizinhos[vertice_corrente] == self.grafo.num_vertices:
    #    break

    min_custo = min(custos) # pega o menor custo da lista
    custo_guloso += min_custo # adiciona o custo ao total
    vertice_corrente = escolhidos[min_custo] # atualiza o vértice corrente
    visitados.append(vertice_corrente) # marca o vértice corrente como visitado

```

Fonte: Autor(2017)

É definido como “vertice\_inicial” com o valor ‘1’, ou seja, ele iniciará o percurso do grafo no nó ‘1’, este sendo o que denominamos “nó inicial”, onde será dada a partida para definição do caminho a ser percorrido de acordo com as preferências detectadas do aprendiz.

O vértice corrente é uma variável que irá percorrer os vértices (nós) no grafo, ou seja, ele identificará em qual nó a formiga estará no momento, para assim traçar os nós adjacentes (vizinhos) a ele, iniciando pelo nó ‘1’.

A lista “visitados”, recebe todo o nó que estará na variável “vertice\_corrente”, ou seja, essa lista será composta por todos os nós onde a formiga for passando, para no final apresentar o caminho seguido no grafo (ou seja, o caminho respectivo de cada aprendiz em um determinado grafo com os devidos objetos de Aprendizagem).

Ainda na Figura 29, é possível notar uma estrutura de repetição, onde está construirá a lista de nós visitados, adicionando-os cada um no final da lista “visitados”, bem como os vizinhos de cada nó que, extremamente importante para a construção do caminho a ser seguido, bem como efetuar as somas do custo guloso.

Ainda na Classe “ACO”, é definido o método “rodar”, como pode ser ilustrado na Figura 30, onde este método é chamado para iniciar a execução do algoritmo.

Figura 30 - Método para executar o algoritmo

```

def rodar(self):
    for it in range(self.iteracoes):

        # lista de listas com as objetosAprendizagem visitadas por cada formiga
        objetosAprendizagem_visitadas = []
        for k in range(self.num_formigas):
            # adiciona a objetoAprendizagem de origem de cada formiga
            objetosAprendizagem = [self.formigas[k].obterObjetoAprendizagem()]
            objetosAprendizagem_visitadas.append(objetosAprendizagem)

        # para cada formiga constrói uma solução
        for k in range(self.num_formigas):

            # somatório do conjunto de objetosAprendizagem não visitadas pela formiga "k"
            # servirá para utilizar no cálculo da probabilidade
            somatorio = 0.0

            # probabilidades de escolher um caminho
            probabilidades = {}

        # atualiza quantidade de feromônio
        for aresta in self.grafo.arestas:
            # somatório dos feromônios da aresta
            somatorio_feromonio = 0.0
            # para cada formiga "k"
            for k in range(self.num_formigas):
                arestas_formiga = []
                # gera todas as arestas percorridas da formiga "k"
                # adiciona a última aresta
                arestas_formiga.append((objetosAprendizagem_visitadas[k][-1], objetosAprendizagem_visitadas[k][0]))
                # verifica se a aresta faz parte do caminho da formiga "k"
                if aresta in arestas_formiga:
                    somatorio_feromonio += (1.0 / self.grafo.obterCustoCaminho(objetosAprendizagem_visitadas[k]))
            # calcula o novo feromônio
            novo_feromonio = (1.0 - self.evaporacao) * self.grafo.obterFeromonioAresta(aresta[0], aresta[1]) + somatorio_feromonio
            # seta o novo feromônio da aresta
            self.grafo.setFeromonioAresta(aresta[0], aresta[1], novo_feromonio)

        # percorre para obter as soluções das formigas
        solucao, custo = None, None
        for k in range(self.num_formigas):
            if not solucao:
                solucao = self.formigas[k].obterSolucao()[:]
                custo = self.formigas[k].obterCustoSolucao()
            else:
                aux_custo = self.formigas[k].obterCustoSolucao()
                if aux_custo < custo:
                    solucao = self.formigas[k].obterSolucao()[:]
                    custo = aux_custo
        #print('Solução final: %s | custo: %d\n' % (' -> '.join(str(i) for i in solucao), custo))
        soma = 0
        for k in self.visitados:
            for j in self.visitados:
                if ((k,j) in self.grafo.arestas.keys()):
                    #print "FOI",self.grafo.obterCustoAresta( k, j)
                    soma = soma + self.grafo.obterCustoAresta( k, j)

        print "caminho: ", self.visitados, " CUSTO TOTAL",soma

```

Fonte: Autor (2017)

Na Figura 30 no método “rodar”, é possível notar uma estrutura de repetição, onde é feita a interação de acordo com o número de iterações definidas nos parâmetros da classe ACO. Diante disso, ele constrói a lista de Objetos de Aprendizagem visitados, através da lista “objetosAprendizagem\_visitadas”. Outra estrutura de repetição foi definida para construir para cada formiga uma solução, e suas interações são de acordo com o número de formigas.

Ainda na Figura 30, também é definida a soma de feromônio de cada aresta através de uma estrutura de repetição onde, através de uma variável “somatorio\_feromonio”, é armazenada a soma de todos os feromônios das arestas, e seu número de iterações é baseada na quantidade de arestas. E ainda com uma estrutura de repetição aninhada, com o número de iterações limitado pelo número de formigas, é criada uma lista de arestas da formiga (aqui como lista “arestas\_formiga”), onde é composta da adição de todas as arestas visitadas pela respectiva

formiga que a visitou, e com um estrutura condicional, verifica se a aresta já foi visitada pela formiga. Posteriormente faz o cálculo do novo feromônio, e define o novo feromônio da aresta através do método “setFeromonioAresta” do grafo.

Ainda no método “rodar”, é definida uma estrutura para percorrer em todas as formigas, e obter as soluções de cada uma das formigas, e armazenar o custo total na variável “custo”. Após isso, é definida uma nova estrutura de repetição, onde o número de iterações é limitada ao número de nós visitados, assim efetuando a soma do custo de cada aresta, e armazenando o valor total em “soma”. Após todo esse processo, é exibido o caminho final, ou caminho percorrido, pela formiga com base nas preferências de cada aprendiz, que foi baseada no custo de cada destino. E como característica, o peso é decrescido, ou seja, é em caráter de minimização, portanto, o caminho traçado de acordo com a afinidade do aprendiz, é com o menor valor de custo do caminho final.

Para efeito de testes, foi definida uma instância chamada “\_\_main\_\_”, ou seja, a instância principal, onde foi feito testes manualmente. Como foi dito anteriormente, para efeito de funcionamento da lógica do algoritmo, e posteriormente implementar de forma automática.

A ilustração pode ser vista na Figura 31 que será apresentada a seguir.

Figura 31 - Exemplo de entrada de dados, feito manualmente para testar o funcionamento lógico do algoritmo.

```

if __name__ == "__main__":

    # cria um grafo passando o número de vértices

    # mapeando objetosAprendizagem para números
    d = {'A':1, 'B':2, 'C':3, 'D':4, 'E':5, 'F':6, 'G':7, 'H':8}

    # adiciona as arestasadicionarAresta
    #for x in grafo.arestas:

        tipo_objeto = 1
        peso=0
        if tipo_objeto == 1:
            peso = 1

        linha0 = [1,0,1,2]
        linha1 = [2,1,1,2]
        linha2 = [3,1,2,2]
        linha3 = [4,1,3,2]
        linha4 = [5,2,1,8]
        linha5 = [6,2,2,90]
        linha6 = [7,2,3,1]
        linha7 = [8,3,1,1]
        linha8 = [9,3,2,1]
        linha9 = [10,3,3,1]
        linha10 = [11,4,1,1]
        linha11 = [12,4,2,1]
        linha12 = [13,4,3,1]
        linha13 = [14,5,1,1]

        #posição [x][0] <- nó Grafo
        #posição [x][1] <- Assunto do Objeto
        #posição [x][2] <- Tipo do Objeto
        #posicao [x][3] <- custo

        matriz = []
        matriz.append(linha0)
        matriz.append(linha1)
        matriz.append(linha2)
        matriz.append(linha3)
        matriz.append(linha4)
        matriz.append(linha5)
        matriz.append(linha6)
        matriz.append(linha7)
        matriz.append(linha8)
        matriz.append(linha9)
        matriz.append(linha10)
        matriz.append(linha11)
        matriz.append(linha12)
        matriz.append(linha13)
        #print "MATRIZ",len(matriz)
        #print "MATRIZ 4,0",matriz[4][0]
        #print "ELEMENTO 0,0:", matriz[1][0]

        matriz_tipoPeso = []
        #matriz_tipoPeso.append([1,2])
        #matriz_tipoPeso.append([2,10])
        #posicao 0 tipoObjeto
        #posicao 1 peso

```

Fonte: Autor (2017)

Podemos perceber na Figura 31, que foi mapeada de letras para números os nomes dos Objetos de Aprendizagens, através da utilização de um tipo de dado dicionário, aqui declarada como “d”.

Foi feita a declaração de linhas que serão adicionadas a matriz “matriz”, onde será os nós do grafo, onde cada posição se refere a um dado de cada nó do grafo, ou seja, a posição 0 da linha da matriz será o nome do nó do grafo, a posição 1 será o Assunto abordado no objeto através de uma numeração que será a identificação única de cada assunto. Na posição 2 é definido o tipo de objeto, ou seja, cada número representa um tipo, seja Vídeo, Texto ou Áudio. E finalmente o peso do nó do grafo, definido na posição 3, com a denominação de “custo”.

E para efeito de definição da preferência do aprendiz, também foi definida uma nova matriz que, levará o tipo de objeto (na posição 0 da linha), que o aprendiz tem maior preferência, bem como o peso atribuído a ela de acordo com a preferência (na posição 1 da linha).

Na Figura 32, é notada a definição de uma estrutura de repetição com mais duas outras estruturas de repetição aninhadas, ainda na instância de “\_\_main\_\_”, a fim de adicionar as arestas por onde a formiga terá possibilidade de transitar, usando a regra estabelecidas nas estruturas condicionais, é estabelecida baseada na tabela de adjacências, ele adiciona as arestas apenas com os nós que possuem relacionamentos definidos em “tabelaAdj” e seu respectivo custo do nó de destino.

Figura 32 - Etapa de adição das arestas

```
contADD=0
for TESTE in range(0,tamanhoMat):

    for TESTE2 in range(tamanhoMat):
        #print "",TESTE2
        if(tabelaAdj[TESTE][TESTE2] == 1):
            #print TESTE+1,",",TESTE2+1,": 1"
            for indices in range (tamanhoMat):
                if matriz[indices][0] == TESTE2+1:

                    #print matriz[indices][3]
                    aux = matriz[indices][3]
                    #print "AUX", aux

                    #if (matriz[indices]==matriz[-1]):
                    #    break
                    #print "ADICIONADO",TESTE+1,",",TESTE2+1,",",aux
                    contADD=contADD+1
                    #print "CONTA PRA MIM", contADD
                    grafo.adicionarAresta(TESTE+1,TESTE2+1,aux)
```

Fonte: Autor (2017)

E finalmente, na Figura 33, é criada uma instância da classe “ACO”, com os parâmetros para sua execução, e é chamada a classe “rodar” da respectiva instância, e é feita a execução do algoritmo de acordo com a entrada de dados, e posteriormente é exibido o caminho percorrido e seu respectivo custo.

Figura 33 - Instanciação da Classe ACO

```
# cria uma instância de ACO  
  
aco = ACO(grafo=grafo, num_formigas=1,tamanhoMatVert=tamanhoMat, alfa=1.0, beta=5.0,  
          iteracoes=10, evaporacao=0.5)  
  
# roda o algoritmo  
aco.rodar()
```

Fonte: Autor (2017)