



**UNIVERSIDADE ESTADUAL DA PARAÍBA
CAMPUS I CAMPINA GRANDE
CENTRO DE CIÊNCIAS E TECNOLOGIA
CURSO DE LICENCIATURA EM COMPUTAÇÃO**

HÉLDER NUNES DE ALMEIDA

LogicSort: Um Objeto de Aprendizagem para o Ensino de Algoritmos de Ordenação

CAMPINA GRANDE - PB
2013

HÉLDER NUNES DE ALMEIDA

LogicSort: Um Objeto de Aprendizagem para o Ensino de Algoritmos de Ordenação

Relatório apresentado ao Curso de Graduação em Licenciatura em Computação da Universidade Estadual da Paraíba, em cumprimento à exigência para obtenção do grau Licenciado em Computação.

Orientador: Prof. Ms. Edson Holanda Cavalcante Junior

CAMPINA GRANDE - PB
2013

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL-UEPB

A4471 Almeida, Hélder Nunes de.
LogicSort [manuscrito] : Um objeto de aprendizagem para o ensino de algoritmos de ordenação / Hélder Nunes de Almeida. – 2013.
23 f. : il. color.

Digitado
Trabalho de Conclusão de Curso (Graduação em Computação) – Universidade Estadual da Paraíba, Centro de Ciências e Tecnologias, 2013.
“Orientador: Prof. Me. Edson Holanda Cavalcante Junior, Departamento de Computação”.

1. Objeto de aprendizagem. 2. Algoritmos. 3. Softwares.
I. Título.

21. ed. CDD 005

HÉLDER NUNES DE ALMEIDA

LogicSort: Um Objeto de Aprendizagem para o Ensino de Algoritmos de Ordenação

Relatório apresentado ao Curso de Graduação em Licenciatura em Computação da Universidade Estadual da Paraíba, em cumprimento à exigência para obtenção do grau Licenciado em Computação.

Aprovada em 6 / 9 / 2013.



Prof. Ms. Edson Holanda Cavalcante Junior / UEPB
Orientador



Prof. Dr. Frederico Moreira Bublitz / UEPB
Examinador



Prof. Dr. Eanes Torres Pereira / UFCG
Examinador

RESUMO

A utilização de Objetos de Aprendizagem (OA) aliada a uma metodologia de ensino promete diminuir as dificuldades enfrentadas por alunos em disciplinas como algoritmos e estrutura de dados. Por serem disciplinas que exigem bastante capacidade de abstrair o conteúdo, o uso de um OA em sala de aula pode ajudar a solucionar este problema. Assim, este relatório apresenta o processo de desenvolvimento de um software, denominado de LogicSort, que tem como objetivo principal demonstrar, de maneira visual, o funcionamento dos algoritmos de ordenação: Bubble Sort, Insertion Sort e Selection Sort.

PALAVRAS-CHAVE: Algoritmos de Ordenação. Ensino. Objeto de Aprendizagem. Processo de Desenvolvimento.

LISTA DE FIGURAS

Figura 1 - WBS da concepção do projeto.....	14
Figura 2 - Diagrama de Casos de Uso	17
Figura 3 - Diagrama de Classes	17
Figura 4 - Interface do LogicSort	18
Figura 5 - Algoritmos	18
Figura 6 - Modo de exibição: Números.....	18
Figura 7 - Modo de edição.....	19
Figura 8 - Modo de exibição: Barras	20
Figura 9 - Modo de exibição: Gráfico	20
Figura 10 - Informações dos algoritmos.....	21

LISTA DE TABELAS

Tabela 1 - Cronograma do desenvolvimento do software	14
Tabela 2 - Ferramentas	15

SUMÁRIO

1. INTRODUÇÃO.....	10
2. FUNDAMENTAÇÃO TEÓRICA	11
3. TRABALHOS RELACIONADOS.....	12
4. METODOLOGIA.....	13
4.1. FERRAMENTAS UTILIZADAS	14
5. ANÁLISE DO SISTEMA.....	15
5.1. LEVANTAMENTO DE REQUISITOS.....	15
5.2. MODELAGEM DA FERRAMENTA	16
6. DESCRIÇÃO DO SISTEMA	17
7. CONSIDERAÇÕES FINAIS.....	21
REFERÊNCIAS BIBLIOGRÁFICAS	22

AGRADECIMENTOS

À Deus, por estar sempre conosco, tanto nos bons, quanto nos momentos de luta.

Aos meus pais Valberto José Araújo de Almeida e Margarida Maria Nunes de Almeida, pela contribuição que me proporcionaram para realização de meus objetivos.

Ao professor Edson Holanda pelas leituras sugeridas ao longo dessa orientação e pela dedicação.

Ao professor Adriano Santos pelas contribuições dadas ao longo do curso.

Aos demais professores do curso de Licenciatura em Computação da UEPB, que muito contribuíram com sua experiência e dedicação.

À minha namorada, por ter me incentivado a concluir o curso e a não desistir nos momentos difíceis.

Aos colegas de classe pelos momentos de amizade e apoio.

1. INTRODUÇÃO

O presente Trabalho de Conclusão de Curso (TCC), em forma de relatório, foi realizado como requisito para a conclusão do curso de Licenciatura em Computação da Universidade Estadual da Paraíba (UEPB). Ele descreve o processo de desenvolvimento de um software, mais especificamente um Objeto de Aprendizagem (OA) voltado para o ensino de algoritmos de ordenação.

A proposta de criação desse OA surgiu, inicialmente, na disciplina de “Software Educacional”. Como produto da disciplina foi desenvolvido uma primeira versão do LogicSort (ALMEIDA *et al.*, 2012).

Já com foco no TCC foi pensado em uma nova versão do LogicSort com a implementação de novas funcionalidades, tais como recursos de animação e gráficos, além da melhoria da sua interface gráfica. Isso foi feito buscando tornar o OA mais eficiente no processo de auxílio das disciplinas.

As dificuldades no ensino de algoritmos se dão pelo fato de exigir não só o aprendizado de uma tecnologia, mas de um conjunto de habilidades como raciocínio lógico, competência para resolver problemas, além da capacidade de abstrair soluções fazendo uso de uma representação formal (SILVA *et al.*, 2011).

A não utilização de metodologias adequadas, ou de alguma ferramenta de apoio ao ensino do professor diante de um conteúdo que apresenta um alto grau de complexidade pode causar problemas como a desmotivação, contribuindo dessa maneira para um maior desinteresse por parte dos alunos.

Assim, é importante buscar alternativas para o ensino de conteúdos que exigem maior concentração do aluno, como o uso de uma ferramenta de ensino que permite ao aluno visualizar mais facilmente o assunto abordado contribuindo assim para diminuir o grau de abstração.

Porém, não se pode afirmar que somente diminuir o grau de abstração de uma disciplina ou conteúdo implicaria, de fato, na motivação do aluno. Para isso, faz-se necessário o uso de uma metodologia atrativa tal como a realização de projetos reais de software fazendo com que o aluno vivencie na teoria e prática o seu processo de desenvolvimento (PEREIRA *et al.*, 2010).

Tomando como base o problema exposto, foi colocado em prática o desenvolvimento de um Objeto de Aprendizagem (OA) que é definida, segundo o *Institute of Electrical and Electronics Engineers (IEEE)*¹, como qualquer entidade, digital ou não, que pode ser usada, reutilizada ou referenciada durante o processo de ensino como suporte tecnológico, auxiliando alunos e professores no processo ensino-aprendizado. Um OA é uma ferramenta capaz de prender a atenção dos alunos e ao mesmo tempo facilitar a aprendizagem (SCOLARI *et al.*, 2007).

Pela experiência obtida na disciplina de Estrutura de dados, algoritmos de ordenação requer um alto grau de abstração. Muitas vezes a complexidade de alguns métodos de ordenação pode se tornar uma grande barreira para que o aluno compreenda o conceito do algoritmo.

2. FUNDAMENTAÇÃO TEÓRICA

Algoritmos de ordenação podem ser definidos como quaisquer algoritmos que, dado um conjunto de elementos, irá colocá-los em uma determinada ordem (crescente ou decrescente). A ordenação pode ser feita de forma completa ou parcial dos elementos. Existem vários métodos de ordenação, como por exemplo: o *Bubble Sort*, o *Insertion Sort* e o *Selection Sort*.

O *Bubble Sort*, ou ordenação pelo método da Bolha, percorre uma lista de n elementos comparando os adjacentes (dois a dois) até que o maior elemento da lista chegue ao final. Este procedimento é repetido nos $n-1$ primeiros elementos até que a lista esteja completamente ordenada.

O *Insertion Sort* é um método de ordenação eficiente quando utilizado um conjunto pequeno de elementos. Basicamente, o algoritmo percorre uma lista de n elementos da esquerda para a direita, deixando ordenados os elementos mais a esquerda à medida que vai avançando até chegar ao último elemento.

O *Selection Sort*, ou ordenação por seleção, também é um método recomendado para trabalhar com um conjunto pequeno de elementos. O algoritmo percorre a lista de n elementos a procura do menor valor colocando-o na primeira posição. Em seguida, repete esse

¹ Institute of Electrical and Electronics Engineers (IEEE) - <http://www.ieee.org/portal/site/iptables/>

procedimento com os próximos n-1 elementos da lista e assim recursivamente até o fim da lista.

Na modelagem do LogicSort, foi utilizado UML², uma linguagem visual usada para modelar sistemas computacionais e que vem sendo adotada como linguagem padrão de modelagem na engenharia de softwares. O uso de UML ajuda a definir as características do software como requisitos, comportamentos, estrutura lógica, dinâmica dos processos, etc. Segundo a OMG³, os diagramas UML dividem-se em duas grandes categorias: estruturais e comportamentais.

Para descrever o cenário que mostra as funcionalidades do sistema segundo o ponto de vista do usuário é aconselhável o uso do diagrama de casos de uso, um diagrama UML que, de acordo com Sampaio (2007), tem como objetivo auxiliar a comunicação entre o analista e o usuário. Já diagrama de classes é um diagrama estrutural que tem como objetivo descrever os vários tipos de classes utilizadas pelo sistema, bem como os seus relacionamentos. É o mais usado e conhecido diagrama UML.

3. TRABALHOS RELACIONADOS

Em “Desenvolvimento de Objetos de Aprendizagem para o Apoio à Disciplina de Algoritmos e Programação” é descrito o desenvolvimento de um objeto de aprendizagem para o ensino da linguagem de programação C (CHECHINEL *et al.*, 2008). O artigo também apresenta algumas observações sobre as dificuldades enfrentadas pelos estudantes no processo de ensino-aprendizagem dos conteúdos de programação e quais materiais didáticos os alunos apontam como úteis para o apoio a esse processo.

Já em “SORTIA - Um Jogo para Ensino de Algoritmo de Ordenação - Estudo de caso na Disciplina de Estrutura de Dados” é apresentado o jogo de ordenação onde o estudante realizará o papel do algoritmo Heapsort afim de vivenciar o processo de ordenação do mesmo e apresenta a metodologia, bem como os resultados obtidos do estudo de caso realizado (BATTISTELLA *et al.*, 2012). O artigo também apresenta o impacto do jogo na motivação, experiência de usuário e aprendizagem com a utilização do jogo.

² Unified Modeling Language ou Linguagem de Modelagem Unificada.

³ Organização Internacional que aprova padrões abertos para aplicações orientadas a objetos.

Enquanto em “A Avaliação do uso de Objetos de Aprendizagem na Educação de Jovens e Adultos” é destacado a satisfação do usuário com o uso de OA’s para o apoio no processo de ensino de jovens e adultos, mostrando assim a possibilidade de reutilização da ferramenta em públicos alvos distintos (OLIVEIRA *et al.*, 2011).

O que diferencia o LogicSort dos demais é o fato de ser um software que aborda o aprendizado focando na ideia central dos métodos de ordenação de uma forma simplificada e de fácil entendimento. Ele utiliza técnicas como passo-a-passo, animações e gráficos, buscando assim a constatação prática do que foi absorvido em sala de aula ou como forma de revisar e fixar melhor a proposta de cada método. Sendo assim, o LogicSort utiliza de um teste de mesa, que consiste em ilustrar todos os passos de um algoritmo, para realizar a demonstração das funcionalidades de algoritmos de ordenação tais como *Bubble Sort*, *Insertion Sort* e *Selection Sort*.

4. METODOLOGIA

Tendo em vista o trabalho de Conclusão de Curso (TCC) foi proposto o desenvolvimento de um sistema, inicialmente idealizado na disciplina de Software Educacional, onde foram utilizadas técnicas de Engenharia de Software focadas no desenvolvimento de Software Educacional com o intuito de motivar os alunos no processo de ensino-aprendizagem (PEREIRA *et al.*, 2009). Assim, foi pensada a criação de um Objeto de Aprendizagem (OA) voltado para o ensino de algoritmos de ordenação.

Definido o tema, realizou-se a etapa de concepção do projeto, ilustrado na Figura 1, na qual foram elencados os requisitos constituintes e pertinentes ao projeto, definindo o público alvo, e questões referentes à acessibilidade do software.

Tendo finalizado essa etapa, iniciou-se a fase de desenvolvimento do sistema e testes da ferramenta. O projeto teve início em maio de 2013 e teve o fim de seu desenvolvimento em junho de 2013, como ilustra a Tabela 1.

Etapas	2013				
	Maio	Junho	Julho	Agosto	Setembro
Análise de Requisitos	x				
Desenvolvimento do Software	x	x			
Testes			x	x	
Produção do Relatório			x	x	
Entrega do Relatório				x	
Apresentação					x

Tabela 1 - Cronograma do desenvolvimento do software

4.1. FERRAMENTAS UTILIZADAS

Para o desenvolvimento do projeto foram utilizadas ferramentas como o WBS Chart PRO e o Astah Community, para a fase de planejamento, e o Eclipse, para a fase de desenvolvimento do software.

O WBS PRO é uma ferramenta que possibilita a organização e o gerenciamento de projetos e ideias com grande facilidade. Com isso, foi utilizado para definir e agrupar os elementos tanto de projeto quanto de modelagem de requisitos do software, como por exemplo, a definição dos itens que compõem a janela do aplicativo, definição das etapas do projeto. A Figura 1 representa o WBS da concepção do projeto.

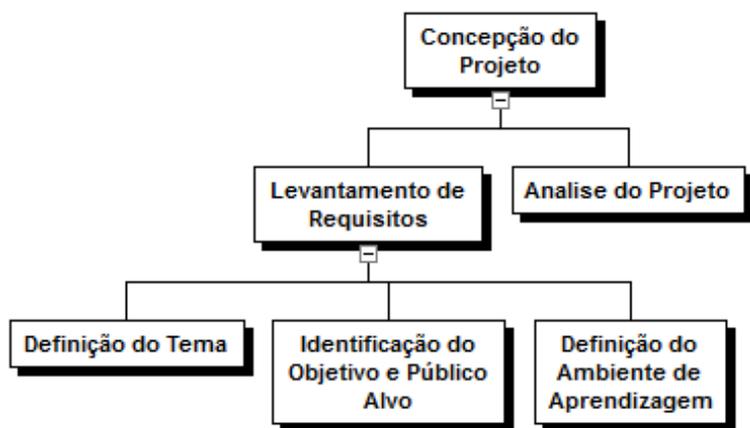


Figura 1 - WBS da concepção do projeto

O Astah Community é uma ferramenta de modelagem UML gratuita que possui diversas funcionalidades, como a criação de vários tipos de diagramas (Classe, Caso de Uso,

etc.), impressão e exportação das imagens dos diagramas, entre outras, além de possuir uma interface de fácil manuseio.

Para o desenvolvimento do aplicativo foi necessária a instalação do JDK 7, esta é fornecida livremente pela Oracle, já que é o kit básico para o desenvolvimento de aplicações Java que engloba compilador, interpretador e utilitários. Junto a este kit, foi utilizado o ambiente de desenvolvimento Integrado (IDE) do Eclipse, por ser uma ferramenta que permite desenvolver aplicações em Java, além de diversas outras linguagens de programação.

O Eclipse apresenta os recursos mínimos para o desenvolvimento de aplicações, além de possuir a funcionalidade de personalização com a instalação de plug-ins adicionais, como é o caso do WindowsBuilder, plug-in utilizado para criação da interface gráfica.

Na Tabela 2, estão dispostas as ferramentas utilizadas para a construção do OA.

Ferramenta	Tipo	Descrição
WBS Chart Pro	Objeto	Modelagem de requisitos
Astah Community	Objeto	Modelagem da ferramenta
JDK 7	Objeto	Kit básico de desenvolvimento Java
Eclipse	Objeto	Ambiente de desenvolvimento
WindowsBuilder	Objeto	Plug-in do Eclipse para o desenvolvimento de interface gráfica

Tabela 2 – Ferramentas

5. ANÁLISE DO SISTEMA

5.1. LEVANTAMENTO DE REQUISITOS

O LogicSort é um objeto de aprendizagem para o ensino de algoritmos de ordenação que tem como objetivo auxiliar no processo de ensino-aprendizagem, transmitindo a ideia central do algoritmo. Para isso foram utilizadas de técnicas como passo-a-passo, podendo ser visualizados em forma de números, barras e gráficos, bem como a possibilidade de navegar entre as interações do algoritmo. Para permitir uma maior interação com o usuário, o software permite que ele altere os valores da sequência que será ordenada.

O software foi desenvolvido para ser aplicado em aula por professores das disciplinas de estrutura de dados ou algoritmos em cursos de computação, porém também pode ser utilizado pelo aluno como forma de tirar dúvidas. O software é indicado para professores que desejam uma metodologia diferente da abordagem tradicional de passar o conteúdo como também para aqueles alunos que desejam exercitar de forma mais pensativa.

Diante destes fatores foram definidos os requisitos do programa para o desenvolvimento da interface gráfica do OA de modo que o ajude a torna-lo mais eficiente no processo de ensino-aprendizagem de algoritmos de ordenação.

5.2. MODELAGEM DA FERRAMENTA

O LogicSort foi desenvolvido na linguagem de programação Java por ser uma linguagem robusta, que possui suporte aos principais conceitos de orientação à objetos, favorecendo para a reusabilidade de código, sendo também, uma linguagem compatível com a maioria dos sistemas operacionais existentes com o uso da sua máquina virtual, além de ser uma das linguagens mais populares do mundo (Deitel, 2010).

A Figura 2 apresenta o diagrama das funcionalidades do LogicSort, na qual representa as seguintes ações que o usuário poderá fazer no sistema:

- “Escolher Algoritmo” - indica que o usuário poderá escolher entre os algoritmos apresentados no programa, o que será aplicado na sequência de números;
- “Escolher Modo de Exibição” - indica que o usuário poderá escolher o modo que será exibido a sequência de números;
- “Escolher Velocidade de Exibição” - indica que o usuário poderá escolher a velocidade de exibição das iterações do algoritmo;
- “Navegar pelos Estados” - indica que o usuário poderá navegar entre as iterações realizadas pelo algoritmo na sequência;
- “Editar sequência de números” - indica que o usuário poderá alterar os valores da sequência de números.

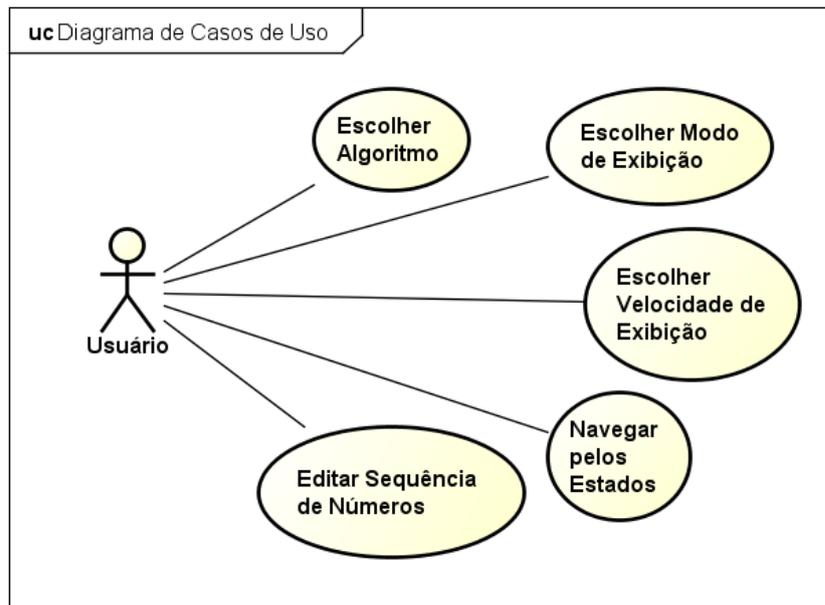


Figura 2 - Diagrama de Casos de Uso

O diagrama de classes do LogicSort foi abordado com a perspectiva voltada para implementação, sendo representado apenas a parte lógica do sistema. Como pode ser visto no diagrama apresentado na Figura 3, a classe “Algoritmo” representa que cada algoritmo possui um nome e pode ter nenhum ou vários Estados associados a ele. Já a classe “Estado” representa a sequência de números, representados por um array, e os identificadores de quais itens estão sendo analisados pelo algoritmo que são representados pelas variáveis, “flagA” e “flagB”, do tipo inteiro.

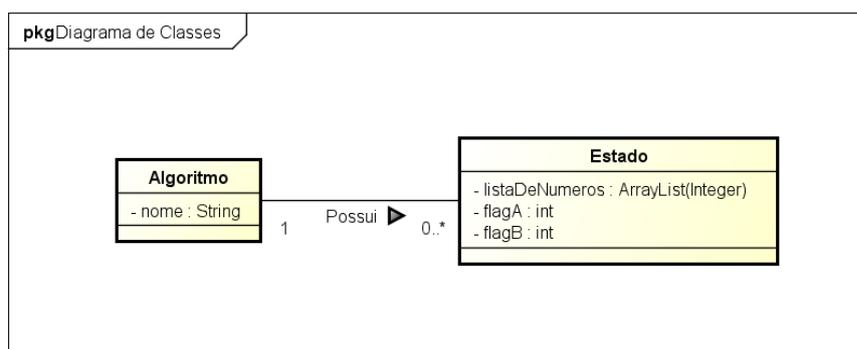


Figura 3 - Diagrama de Classes

6. DESCRIÇÃO DO SISTEMA

O LogicSort possui uma interface de fácil entendimento, com cores neutras para evitar um possível cansaço visual, importante para a aceitação do sistema (MOREIRA *et al.*, 2010), como ilustrado na Figura 4. Na qual consiste na tela de abertura do programa.

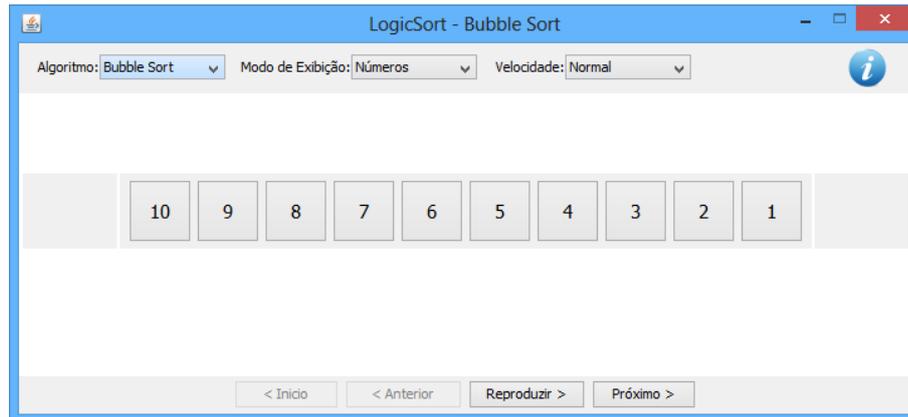


Figura 4 - Interface do LogicSort

A interface do LogicSort contém 3 elementos básicos: Uma barra de propriedades, uma barra de navegações e um painel de exibição do conteúdo. A barra de propriedades, localizada na parte superior da janela, possui um campo de seleção de algoritmo, onde possibilita ao usuário escolher os algoritmos de ordenação que serão aplicados na lista de itens, como é mostrado na Figura 5.

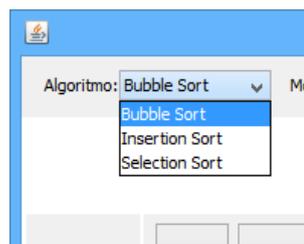


Figura 5 - Algoritmos

Ainda na barra de propriedades, o usuário poderá definir o modo de exibição, escolhendo entre o “modo de Barras”, o “modo gráfico” e o “modo números”, como ilustrado na Figura 6. O terceiro item da barra de propriedades permite controlar a velocidade em que a lista de itens será ordenada.

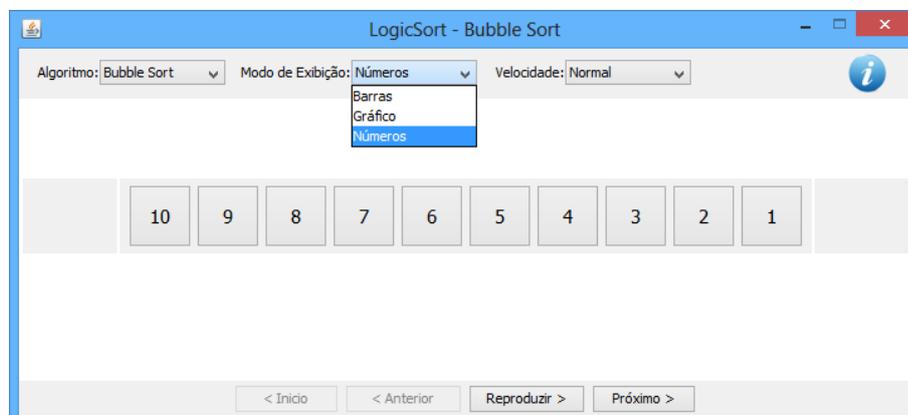


Figura 6 - Modo de exibição: Números

Localizada na parte inferior da janela, a barra de navegação possibilita ao usuário navegar entre as iterações à medida que o algoritmo selecionado vai interagindo com a sequência de itens. Assim, permite ao usuário tanto a reprodução automática, na qual a navegação ocorre automaticamente de acordo com a velocidade escolhida, como em um tocador de vídeo, como também a navegação manual, na qual o usuário escolhe quando irá avançar para o próximo passo e se quiser voltar para o passo anterior ou para o passo inicial.

Na parte central da janela é onde se encontra o painel de exibição. Nele são exibidas as iterações da lista de itens de acordo com o modo de exibição selecionado. Quando selecionado o “modo números”, ilustrado na Figura 6, são exibidos no painel 10 botões, de modo que cada um representa um item da lista, cada um com o seu valor. Clicando em qualquer um deles, é ativado o modo de edição, como mostra a Figura 7, que possibilita a alteração dos seus valores, podendo alterar os valores individualmente, definir números aleatórios para todos os botões ou determinar uma sequência decrescente apenas acionando os botões correspondentes. Feita a alteração, cabe ao usuário confirmar, caso queira manter a sequência clicando no botão “Ok” ou caso queira desistir da nova sequência e voltar para os valores anteriores, bastando apenas clicar no botão “Cancelar”.

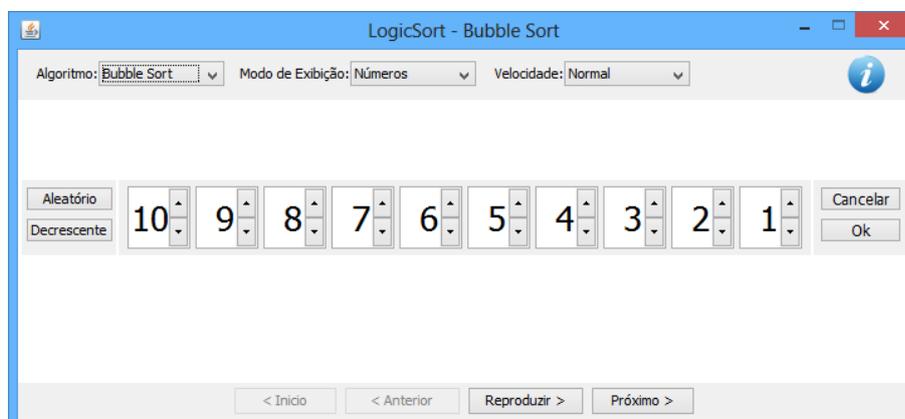


Figura 7 - Modo de edição

O “modo barras”, como mostra na Figura 8, consiste em dez barras verticais posicionadas lado a lado, cada uma com a altura correspondente ao valor do item. Foram utilizadas barras com cores de borda e preenchimento diferentes para destacar a iteração do algoritmo.

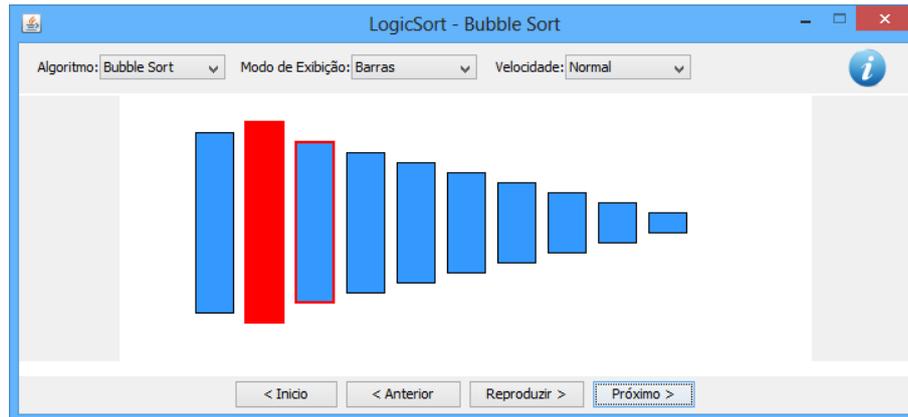


Figura 8 - Modo de exibição: Barras

Assim como o “modo barras”, o “modo gráfico” apresenta o mesmo princípio, consiste em dez barras verticais posicionadas lado a lado, cada um com o tamanho correspondente ao valor do item, porém são posicionadas com a base inferior no plano cartesiano, ilustrado na Figura 9.

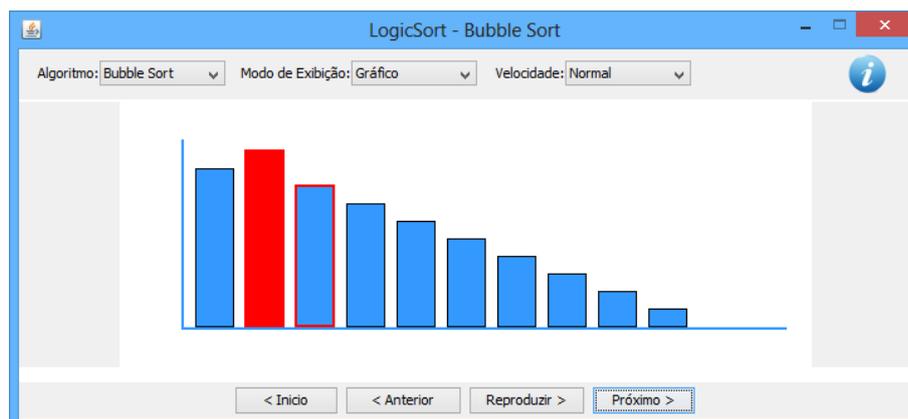


Figura 9 - Modo de exibição: Gráfico

Por fim, o botão localizado no ponto superior direito abre um menu composto de dois itens: algoritmos e sobre, como é mostrado na Figura 10. Clicando nesses itens são exibidas no painel de exibição as informações sobre os algoritmos presentes, referentes ao item Algoritmo, e as informações de software e contato no caso do usuário ter dúvidas ou sugestões, referentes ao item Sobre.



Figura 10 - Informações dos algoritmos

7. CONSIDERAÇÕES FINAIS

Com o desenvolvimento desta nova versão do LogicSort se procurou aplicar tanto novas maneiras de demonstrar o conteúdo abordado quanto poder tornar sua interface gráfica mais atraente e fácil de usar. Essas alterações foram feitas com o objetivo de inserir um OA mais significativo no ensino de Estrutura de Dados.

Contudo, um OA sozinho não terá resultados expressivos se o professor não souber manipulá-lo (FERNANDES *et al.*, 2008), espera-se um aproveitamento efetivo e eficiente tanto por parte dos professores como por parte dos alunos. Ou seja, não se pode avaliar uma ferramenta como o LogicSort quando o professor que a gerencia não é capacitado para utilizá-la de modo correto em seu ambiente de ensino.

Com o uso desta ferramenta, espera-se obter melhorias no processo ensino-aprendizado, fazendo com que o conceito do algoritmo abordado na disciplina de Estrutura de Dados seja melhor entendido pelos alunos e, possivelmente, contribuindo para que o índice de evasão e reprovação se torne cada vez mais reduzido.

Com isso, pretende-se, como trabalho futuro, a realização da validação do LogicSort, na qual consiste na sua aplicação em sala de aula. Em seguida, será realizada uma pesquisa, de forma que se possa avaliar a qualidade do software identificando o índice de aproveitamento bem como seu impacto no processo de ensino-aprendizagem.

REFERÊNCIAS BIBLIOGRÁFICAS

ALMEIDA, H. N. D. et al. Logic SORT: Uma Oportunidade de Aprendizagem Alternativa sobre Algoritmos de Ordenação. **CSBC 2012 - WEI - XX Workshop sobre Educação em Computação**, Curitiba, 2012.

BATTISTELLA, P. E.; WANGENHEIM, A. V.; WANGENHEIM, C. G. V. SORTIA - Um Jogo para Ensino de Algoritmo de Ordenação - Estudo de caso na Disciplina de Estrutura de Dados. **Anais do 23º Simpósio Brasileiro de Informática na Educação**, Rio de Janeiro, 26-30 Novembro 2012.

CECHINEL, C. et al. Desenvolvimento de Objetos de Aprendizagem para o Apoio a disciplina de Algoritmos e Programação. **WA4P - II Workshop de Ambientes de apoio à Aprendizagem de Algoritmos e Programação - SBIE 2008**, Fortaleza, 2008. 1-9.

DEITEL, P. J.; DEITEL, H. M. **Java Como Programar**. 8. ed. São Paulo: Pearson Prentice Hall, 2010.

FERNANDES, A. C. et al. Implementação e observação de práticas pedagógicas com o uso de Objetos de Aprendizagem na Escola. **Anais do XVIII Congresso da SBC - Workshop sobre Informática na Escola - WIE**, Belém do Pará, 2008. 127-133.

IEEE. Learning Technology Standards Committee. **IEEE - WG12: Learning Object Metadata**. Disponível em: <<http://ltsc.ieee.org/wg12/>>. Acesso em: 24 Julho 2013.

MOREIRA, N. S.; MELARÉ, A. V. S.; MICALI, D. L. C. A Interação do Homem com o Computador (IHC) - Análise dos preceitos semióticos e psicológicos como instrumentos para a compreensão e estruturação de Interfaces Homem-Computador. **Revista Sapere**, 2010. Disponível em: <http://www.revistasapere.inf.br/download/segunda/MOREIRA_MELARE_MICALI.pdf>. Acesso em: 20 Julho 2013.

OLIVEIRA, K. A.; AMARAL, M. A.; DOMINGOS, G. R. A Avaliação do uso de Objetos de Aprendizagem na Educação de Jovens e Adultos. **Revista Brasileira de Informática na Educação**, v. 19, n. 3, p. 53-64, Dezembro 2011.

PEREIRA, L. F. D. et al. Ateliê de Objetos de Aprendizagem - Uma metodologia para o Ensino de Computação em Cursos Técnicos. **Anais do CSBC 2010 - XVIII Workshop sobre educação em Computação - XVIII WEI**, Belo Horizonte, 2010. 907-916.

SAMPAIO, M. C. Formalismo UML de Análise de Projetos, 2007. Disponível em: <<http://www.dsc.ufcg.edu.br/~sampaio/cursos/2007.1/Graduacao/SI-II/Uml/>>. Acesso em: 9 Agosto 2013.

SCOLARI, A. T.; BERNARDI, G.; CORDENONSI, A. Z. O Desenvolvimento do Raciocínio Lógico através de Objetos de Aprendizagem. **RENOTE - Revista Novas Tecnologias na**

Educação, v. 5, 2007. ISSN 1679-1916. Disponível em: <<http://www.cinted.ufrgs.br/ciclo10/artigos/4eGiliane.pdf>>. Acesso em: 15 Agosto 2013.

SILVA, G. C. et al. Uma experiência na aplicação de práticas de apoio no ensino-aprendizado de algoritmos. **Anais do XXII SBIE - XVII Workshop sobre Informática na Escola - WIE**, Aracaju, 21-25 Novembro 2011. 1378-1381.