Silvino Marques da Silva Junior

Escola de Algoritmos: Ferramenta Digital de Aprendizagem para o Auxílio do Ensino de Lógica de Programação na EAD

Recife

2016

Universidade Federal Rural de Pernambuco

Unidade Acadêmica de Educação a Distância e Tecnologia

Pró-Reitoria de Pesquisa e Pós-Graduação

Programa de Pós-Graduação em Tecnologia e Gestão em Educação a Distância

Escola de Algoritmos: Ferramenta Digital de Aprendizagem para o Auxílio do Ensino de Lógica de Programação na EAD

Dissertação apresentada ao Programa de Pós-Graduação em Tecnologia e Gestão em Educação a Distância, como exigência parcial à obtenção do título de Mestre em Tecnologia e Gestão em Educação a Distância.

Linha de Pesquisa: Ferramentas Tecnológicas para Educação a Distância

Orientadora: Profª. Dra. Sônia Virgínia Alves França

Recife

2016

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema Integrado de Bibliotecas da UFRPE
Biblioteca Central, Recife-PE, Brasil

S586e Silva Junior, Silvino Marques da
Escola de algoritmos: ferramenta digital de aprendizagem para o auxílio do ensino de lógica de programação na EAD / Silvino Marques da Silva Junior . – 2016.
100 f. : il.

Orientadora: Sônia Virgínia Alves França.
Dissertação (Mestrado) – Universidade Federal Rural de Pernambuco, Programa de Pós-Graduação em Tecnologia e Gestão em Educação a Distância, Recife, BR-PE, 2016.
Inclui referências, anexo(s) e apêndice(s).

1. Programação 2. Ensino-aprendizagem 3. Ferramentas tecnológicas I. França, Sônia Virgínia Alves, orient. II. Título

CDD 371.394422

Universidade Federal Rural de Pernambuco

Unidade Acadêmica de Educação a Distância e Tecnologia

Pró-Reitoria de Pesquisa e Pós-Graduação

Programa de Pós-Graduação em Tecnologia e Gestão em Educação a Distância

Escola de Algoritmos: Ferramenta Digital de Aprendizagem para o Auxílio do Ensino de Lógica de Programação na EAD

Silvino Marques da Silva Junior

Dissertação julgada adequada para obtenção do título de Mestre em Tecnologia e Gestão em Educação a Distância, defendida e aprovada por unanimidade em 14/10/2016 pela Banca Examinadora.

Orientadora:

_____

Prof(a). Dr(a). Sônia Virgínia Alves França
Programa de Pós-Graduação em Tecnologia e Gestão em Educação a Distância - UFRPE

Banca Examinadora:

_____

Prof. Dr. Rodrigo Nonamor Pereira Mariano de Souza
Membro Interno – Programa de Pós-Graduação em Tecnologia e Gestão em Educação a Distância - UFRPE

_____

Prof. Dr. Domingos Sávio Pereira Salazar
Membro Interno – Programa de Pós-Graduação em Tecnologia e Gestão em Educação a Distância - UFRPE

_____

Prof(a). Dr(a). Sandra de Albuquerque Siebra
Membro Externo – Programa de Pós-Graduação em Ciência da Informação – UFPE

A Deus, por tudo que fez e tem feito em minha vida.

A minha avó Terezinha de Jesus, pelos ensinamentos.

A minha esposa Jussara e filhos, Maria Cecília e Silvino Neto, pelo apoio, amor e compreensão.

AGRADECIMENTOS

Em primeiro lugar a Deus, que me proporcionou chegar até aqui, com saúde e coragem para superar os obstáculos da vida.

A minha avó Terezinha de Jesus e aos meus pais Silvino Marques da Silva e Isabel Madeira Coelho, por terem me ensinado os caminhos da sabedoria, ética e dignidade.

A minha esposa Jussara Rodrigues Barros, pelo amor, pela paciência e companheirismo nos momentos de dificuldade.

A meus filhos Maria Cecília e Silvino Neto, por todo o carinho e por serem o motivo de sempre lutar para atingir a meus objetivos.

A meus irmãos Igor e Isabela, por sempre estarem ao meu lado e ajudar nos momentos necessários.

A meus tios e tias, em especial a minha tia Socorro Maria, por todo apoio dado em vários momentos da minha vida.

A minha orientadora, Prof.ª. Dra. Sônia Virgínia Alves França, pelos ensinamentos, pela dedicação e pelo incentivo para conseguirmos completar este trabalho.

Ao Reitor do IFPI, Prof. Dr. Paulo Henrique Gomes de Lima, pelo apoio dado à nossa turma de mestrado.

Aos meus colegas de curso e de apartamento, pelas trocas de conhecimento e momentos de vivências em Recife - PE, em especial Marcelo Teixeira, Gildon César, Cláudio Rodrigues, Robson Almeida e Sebastião Assunção, pela parceria em diversos momentos.

E a todos que direta ou indiretamente contribuíram para realização deste trabalho. Muito obrigado!

RESUMO

A dificuldade de aprendizagem nas disciplinas de Algoritmos e Programação é uma realidade enfrentada por muitos alunos, seja no que tange ao conteúdo de raciocínio lógico ou conteúdo de codificação e linguagens de programação. Na Educação a Distância não é diferente, pelo contrário, tem-se a distância física entre professor-aluno como um agravante, o que dificulta o dirimir de dúvidas. Para o aprendizado significativo faz-se necessário que o aluno tenha força de vontade, autonomia e interesse pelos assuntos. Diante desse cenário, o advento das tecnologias de informação e comunicação trouxe novas perspectivas para a Educação a Distância, com a utilização de ferramentas digitais proporcionando uma nova forma de estudar e aprender. Neste contexto este trabalho tem por objetivo auxiliar o ensino de Lógica de Programação em cursos ofertados a distância, por meio de uma ferramenta tecnológica com recursos multimídia, interativa e de fácil manuseio, que seja agradável e possa despertar o interesse dos alunos, contribuindo para a aprendizagem. A presente pesquisa é de caráter qualitativo, pois surgiu da necessidade de se investigar o porquê dos altos índices de evasão e retenção em disciplinas de programação e analisar de forma descritiva possíveis soluções para esses problemas. O protótipo da ferramenta Escola de Algoritmos foi construído utilizando a solução *Adaptive eLearning Platform – AeLP*, plataforma para criação de atividades *e-Learning*, interativas, altamente visuais e adaptativas. Para isso, a primeira etapa do desenvolvimento envolveu a análise de ferramentas que tem como objetivo o ensino e a aprendizagem de programação. A segunda etapa envolveu o levantamento de informações por meio da aplicação de um questionário a um grupo de alunos do curso de Informática a distância do Instituto Federal do Piauí, para servir de base para o desenvolvimento da ferramenta proposta. Com a avaliação da ferramenta, pôde-se concluir que a mesma oferece meios para a aprendizagem de conteúdos teóricos e a desenvoltura na criação de algoritmos, reduzindo as dificuldades provenientes desse conteúdo.

Palavras-chave: programação; ensino-aprendizagem; ferramentas tecnológicas**.**

ABSTRACT

The difficulty in learning of algorithms and programming disciplines is a reality faced by many students, both in terms of logical reasoning and coding in programming languages. Distance Education is no different, on the contrary, there is an aggravating factor which is the physical distance between teacher and student, which imposes barriers to the solving of doubts and questions. Thus, it is necessary that the student has a strong will, autonomy and curiosity for the subjects, in order to achieve quality learning. Considering this scenario, the onset of information and communication technologies has brought new perspectives for Distance Education in the use of digital tools provides new ways to study and learn. In this context, this study aims to assist the teaching of Logic Programming courses offered on Distance Education platforms, through a technological tool with multimedia, interactive and easy to handle, which is pleasant and can encourage the interest of students, thus contributing to learning. This research is qualitative, it sprung from the need to investigate why the high rates of dropout and retention programming disciplines and analyze descriptively possible solutions to these problems. The prototype tool Escola de Algoritmos was built using the solution Adaptive eLearning Platform – AeLP it is a platform for building activities e-Learning, interactive, highly visual and adaptive. In order to achieve this, the first stage of development involved the analysis tools that aims at teaching and learning programming. The second step involves collecting information through the application of a questionnaire to a computer course the group of students away from Federal Institute of Piauí, to act as base for the development of the proposed tool. With the assessment tool, it could be concluded that it offers means to the learning of theoretical content and proficiency in the creation of algorithms, reducing the difficulties of this content.

Keywords: programming; teaching; technological tools**.**

LISTA DE FIGURAS

Figura 1 - Interface do *Feeper*. 29
Figura 2 - Tela do WH-IDE 30
Figura 3 - Interface do TutorICC 31
Figura 4 - Interface do *Portugol Studio* 32
Figura 5 - Interface do AAPW 33
Figura 6 - Tela do *TSTView* 34
Figura 7 - Página Inicial do *The Huxley* 35
Figura 8 - Interface do SOAP 36
Figura 9 - Interface do *ProgTest* 37
Figura 10 - Tela do *Codecademy* 38
Figura 11 - Interface do *Scratch*. 39
Figura 12 - Interface do Alice 40
Figura 13 - Fluxo de Trabalho 48
Figura 14 - Tela Inicial da *Smart Sparrow*. 57
Figura 15 - Tela Inicial da Escola de Algoritmos. 58
Figura 16 - Tela Inicial com descrição. 58
Figura 17 - Tela com Instruções. 59
Figura 18 - Tela de Conteúdo. 60
Figura 19 - Tela com *Feedback* de Erro 61
Figura 20 - Tela de Exercícios 62
Figura 21 - Tela com *Feedback* 62
Figura 22 - Tela de Revisão 63
Figura 23 - Tela com Exercícios sobre Fluxogramas 63
Figura 24 - Tela sobre Tipos de Dados e Variáveis 64
Figura 25 - Tela com Exercícios de Múltipla Escolha 64
Figura 26 - Tela com Código 65
Figura 27 - Tela com Animação 66
Figura 28 - Tela com Teste de Mesa. 66
Figura 29 - Tela com Exercício. 67
Figura 30 - Tela sobre Comandos de Leitura e Exibição. 68
Figura 31 - Exercício sobre Comandos de Leitura e Exibição. 68

Figura 32 - Tela de Finalização do Módulo.....69
Figura 33 - E-mail enviado ao aluno.....69

LISTA DE TABELAS

Tabela 1 - Resultados da disciplina Lógica de Programação 50

LISTA DE QUADROS

Quadro 1 - Nativos e Imigrantes Digitais. 24
Quadro 2 - Características Técnicas das Ferramentas. 41
Quadro 3 - Funcionalidades. 42
Quadro 4 - Competências Trabalhadas. 44

LISTA DE GRÁFICOS

Gráfico 1 - Aprendizado em programação. 50
Gráfico 2 - Qualidade e grau de interatividade do material 51
Gráfico 3 - Utilização de uma ferramenta tecnológica. 52
Gráfico 4 - Capacidade de codificar um programa 53
Gráfico 5 - Questões relacionadas a interface. 72
Gráfico 6 - Uso de texto, imagens, animações e vídeos 73
Gráfico 7 - Ferramenta apropriada, utilidade e evolução. 74
Gráfico 8 - Adaptatividade e *Feedback* Automático 75
Gráfico 9 - Satisfação e Considerações pós-uso 76

LISTA DE SIGLAS

ABED	Associação Brasileira de Educação a Distância
ABNT	Associação Brasileira de Normas Técnicas
AVA	Ambiente Virtual de Aprendizagem
CBIE	Congresso Brasileiro de Informática na Educação
EAD	Educação a Distância
IDE	*Integrated Development Environment* (Ambiente de Desenvolvimento Integrado)
IFPI	Instituto Federal de Educação, Ciência e Tecnologia do Piauí
NAD	Núcleo de Avaliação Diagnóstica
NAF	Núcleo de Avaliação Formativa
RBIE	Revista Brasileira de Informática na Educação
TIC	Tecnologias da Informação e Comunicação
UFRPE	Universidade Federal Rural de Pernambuco

SUMÁRIO

1. INTRODUÇÃO 14
1.1. Descrição do problema 15
1.2. Objetivo da pesquisa 16
1.2.1. Geral 16
1.2.2. Específicos 17
1.3. Justificativa 17
1.4. Estrutura do documento 18
2. REVISÃO BIBLIOGRÁFICA 20
2.1. Aprendizagem de Programação 20
2.2. EAD: Dificuldades de se estudar a Distância 24
2.3. Vantagens do Uso das Tecnologias no Ensino 27
2.4. Ferramentas Utilizadas no Ensino de Programação 29
2.4.1. Ambiente Virtual com *Feedback* Personalizado - Feeper 29
2.4.2. Ambiente Personalizável para Aprendizado de Programação – WH-IDE 30
2.4.3. Ambiente Interativo e Adaptável para Ensino de Programação - TutorICC 31
2.4.4. Portugol Studio 32
2.4.5. Ambiente de Aprendizagem de Programação Web - AAPW 33
2.4.6. Ferramenta de Acompanhamento de Exercícios de Programação - TSTView 34
2.4.7. Ferramenta para Avaliação de Aprendizado de Alunos em Programação de Computadores – *The Huxley* 35
2.4.8. Sistema On-line de Atividades de Programação - SOAP 36
2.4.9. Ambiente de Apoio ao Ensino de Programação e Teste de *Software* - *ProgTest* 37
2.4.10. *Codecademy* 38
2.4.11. *Scratch* 39
2.4.12. Alice 40
2.4.13. Comparativo das Ferramentas 41
3. METODOLOGIA 46
3.1. Materiais e Métodos 46
3.2. Coleta de Dados 49
4. PROTÓTIPO DA FERRAMENTA 55
4.1. Escola de Algoritmos 55
5. ANÁLISE E DISCUSSÃO 71
5.1. Avaliação da Ferramenta Escola de Algoritmos 71
6. CONCLUSÃO 78
REFERÊNCIAS 80

APÊNDICE A **89**
APÊNDICE B **91**
ANEXOS **94**

1. INTRODUÇÃO

É crescente a quantidade de pessoas que tem dificuldade para interpretar e solucionar problemas utilizando o computador, esta é observada nos cursos de informática e, principalmente, nas disciplinas de Lógica de Programação ou Algoritmos e Programação, que são extremamente importantes e que motivam os alunos a utilizar técnicas para resolução de problemas computacionais (BALDUINO; FERREIRA, 2015).

A aprendizagem de algoritmos e programação requer esforço e dedicação por parte dos alunos, uma vez que estes não possuem base sólida nos conhecimentos de matemática e raciocínio lógico, importantes para aprender conceitos de programação de computadores.

Jenkins (2002) enumera algumas causas dos altos índices de reprovação e evasão em disciplinas de programação: baixo nível de abstração, a falta de competências necessárias a resolução de problemas, métodos pedagógicos inadequados aos estilos de aprendizagem dos alunos, entre outros; e afirma ainda que a sintaxe da grande maioria das linguagens de programação não é adequada para alunos iniciantes.

Segundo Ferreira (2005), estas disciplinas são de extrema importância para um curso de Computação/Informática, pois são a base para o entendimento de muitas outras disciplinas e ferramenta indispensável para o egresso no mercado de trabalho. Desta forma, necessita-se de bons métodos para abordar o conteúdo e favorecer a motivação dos alunos.

É preciso que haja dedicação para entender os conceitos expostos e, também, prática para que eles possam ser assimilados. Outro grande problema enfrentado pelos alunos é o entendimento dos enunciados das questões, muitos deles não conseguem compreender o que é pedido e, consequentemente, não conseguem fazer o que é solicitado (ALVES, 2014).

A utilização da tecnologia tem o objetivo de entusiasmar os alunos e impulsionar o interesse acerca dos temas abordados, principalmente quando se trata de Educação a Distância (EAD), em que são fundamentais no aluno as características de autonomia e pró-atividade.

Assim, a tecnologia vem proporcionando oportunidades para práticas educacionais inovadoras. Estes espaços estão se definindo como possibilidades estratégicas para o desenvolvimento de programas de EAD, aliando a virtualidade à interatividade, em uma tentativa de superação de dois elementos básicos nos processos de ensino e aprendizagem: distância e tempo (AMORIM et al., 2016).

Ainda é possível citar, como referência em avanços tecnológicos, os dispositivos móveis e a comunicação sem fio, pois estes contribuem, também, de forma considerável para os processos de ensino e aprendizagem. A integração de diferentes tecnologias, de modo que as mesmas interajam entre si, oferece grande vantagem, que é o fato de todos os usuários poderem usufruir de vários aspectos dessa nova abordagem, em especial a comodidade de prover materiais, acessá-los e visualizá-los, em qualquer horário e lugar (YAU; JOY, 2011).

Diante do exposto, o presente trabalho descreve a Escola de Algoritmos[1], ferramenta digital para auxílio no ensino de Lógica de Programação na modalidade de ensino a distância. A ferramenta desenvolvida tem como principal objetivo estimular e motivar os alunos e, consequentemente, auxiliar o processo de ensino e aprendizagem da disciplina mencionada.

1.1 Descrição do Problema

O aprendizado de algoritmos e programação é uma parte importante da vida acadêmica do estudante de computação. E, também, é considerado fator crítico quando se trata das causas de reprovação e evasão nos cursos da área. Muitas são as causas e os estudos a respeito, mas devido à grande variedade de fatores que implicam nessas dificuldades, urge estabelecer técnica ou metodologia que possa resolver ou amenizar esses problemas (GIRAFFA; MORA, 2013).

Na Educação a Distância esse processo se torna ainda mais complexo, pois há separação física entre o professor e o aluno durante quase todo o

[1] Disponível em: https://aelp.smartsparrow.com/v/open/w8c53tpr

processo educativo, e o aluno tem que buscar o apoio das várias tecnologias e mídias para facilitar o aprendizado (TORRES; BEHRENS; MATOS, 2015).

Assim, o professor tem papel diferente, e nos novos espaços de comunicação utilizados, como ambientes educativos, passa a exercer a mediação pedagógica, “levando em consideração a consciência do trabalho em rede que é voltado para a coletividade, a colaboração e a partilha” (TSUKAMOTO; FIALHO; TORRES, 2014, p. 350).

Os problemas típicos da EAD, como a falta de motivação dos estudantes, resultando, muitas vezes, no abandono dos cursos, estimulam atualmente estudos sobre o uso de novas tecnologias (YESSAD; LABAT; KERMORVANT, 2010).

Diante desses fatos, este estudo inicia-se com a seguinte pergunta: como reduzir as dificuldades encontradas pelos alunos da EAD em aprender programação?

1.2 Objetivos da pesquisa

Esta seção aborda o objetivo geral e os objetivos específicos do trabalho.

1.2.1 Geral

Contribuir com a melhoria do aprendizado de Lógica de Programação em cursos ofertados na modalidade à distância, por meio da utilização de uma ferramenta tecnológica.

1.2.2 Específicos

- Identificar as dificuldades encontradas na disciplina de Lógica de Programação pelos alunos do curso técnico em Informática para Internet na modalidade a distância do IFPI;
- Analisar algumas ferramentas existentes que são utilizadas para auxiliar o ensino de programação;

- Desenvolver ferramenta tecnológica que auxilie o ensino-aprendizagem de Lógica de Programação, em consonância com os dados obtidos dos alunos e ferramentas analisadas;
- Avaliar a aceitação da ferramenta.

1.3 Justificativa

Segundo estudo da Associação Brasileira de Educação a Distância (ABED, 2015), a Educação a Distância aumentou 25%, no Brasil, entre 2013 e 2014, ultrapassando a marca de 3,8 milhões de alunos matriculados.

No IFPI, não foi diferente, a oferta de cursos a distância iniciou-se em 2008 com apenas dois cursos, divididos em quatro polos. Em 2010, a modalidade passou a ser ofertada em oito polos e o número de vagas aumentou de 350 para 800 (MORAIS, 2015).

Notadamente, foi em 2013 que a expansão do curso técnico de Informática para Internet se tornou mais significativa, passando de quatro para 33 polos, com oferta do respectivo curso, atingindo 1.750 alunos em diferentes regiões do estado do Piauí (MORAIS, 2015).

Com a rápida expansão dessa modalidade de ensino surgiram alguns problemas e dificuldades por parte das metodologias até então utilizadas no ensino presencial, que passaram a não mais suprir as necessidades dessa nova forma de “ensinar”.

E, nas disciplinas de programação, em que já existia alto grau de dificuldade por parte dos alunos em entenderem os conceitos estudados no ensino presencial, esse fato se agravou com o advento dos cursos de informática à distância. Isso desencadeou a realização de inúmeras pesquisas relacionadas ao estudo de melhores metodologias e técnicas para serem utilizadas no ensino dessas disciplinas (OLIVEIRA; OLIVEIRA, 2014).

O curso técnico em Informática para Internet na modalidade a distância do IFPI, contém três disciplinas relacionadas à programação: Lógica de Programação, Introdução à Programação para Web e Programação para Web.

Essas três disciplinas têm em seus conteúdos programáticos assuntos que demandam prática para serem entendidos e, em todas elas, são utilizadas

linguagens de programação para aplicação dos conceitos e visualização dos resultados. Essas práticas devem ser feitas utilizando computadores e *softwares* específicos, o que, muitas vezes, não acontece, por falta de infraestrutura física, ou até mesmo falta do conhecimento necessário por parte do tutor presencial para repassar aos alunos orientações.

O que se pretende é propor uma ferramenta que possa ser utilizada na disciplina de Lógica de Programação (ver anexo), em cursos de Informática a distância que facilite e estimule o aprendizado de assuntos considerados difíceis pelos alunos e, desta forma, possibilite ao professor mais uma opção metodológica em cursos a distância.

1.4 Estrutura do Documento

Os outros capítulos desta dissertação estão organizados da seguinte maneira:

- **Capítulo 2 – Revisão Bibliográfica:** discutem-se os problemas de aprendizagem de algoritmos e programação, as dificuldades no ensino a distância, as vantagens do uso da tecnologia a favor da educação e faz uma análise de ferramentas que auxiliam no ensino de programação;
- **Capítulo 3 – Metodologia:** descrevem-se as técnicas de pesquisa utilizadas e materiais e métodos para o desenvolvimento da ferramenta, além de apresentar dados, limitações e discussões acerca do problema, que foram utilizados como base para o desenvolvimento da ferramenta proposta;
- **Capítulo 4 – Protótipo da Ferramenta:** descreve-se o desenvolvimento da ferramenta proposta;
- **Capítulo 5 – Análise e Discussão:** discutem-se os resultados da avaliação da ferramenta por um grupo de alunos da modalidade a distância;
- **Capítulo 6 – Conclusão:** são feitas as considerações finais e propostas de trabalhos futuros.

2. REVISÃO BIBLIOGRÁFICA

Este capítulo descreve o estado da arte sobre os desafios na aprendizagem de programação, os benefícios do uso de ferramentas no apoio ao ensino e faz uma análise acerca de algumas das ferramentas disponíveis atualmente que auxiliam no ensino de programação.

2.1 Aprendizagem de Programação

A aprendizagem é um processo fundamental na vida do homem, por meio desta, desenvolvem-se comportamentos que o permitem viver. O processo de ensino requer a compreensão do processo de aprendizagem Neste sentido, desde a antiguidade, filósofos e pensadores estudam a aprendizagem e os fenômenos que a envolvem (VALENTIM, 2011).

> A educação do século XXI tem como objetivo primeiro o aprender a aprender, vivemos em um tempo de aceleração do ritmo das coisas, a sociedade apresenta mudanças em todas as áreas: política, tecnológica, social, cultural, econômica. Aprender neste contexto requer assimilar informação, ter conhecimentos, realizar operações, exercitar procedimentos e estratégias para tirar o melhor partido do que se conhece, conhecer mais, resolver problemas, tomar decisões. (GARCIA; VEIGA, 2007, p.92).

O professor, ao conduzir o processo ensino-aprendizagem, precisa estar por dentro de seu papel e possuir a percepção do que o aluno sente dificuldade e não consegue assimilar. Mais do que ensinar, é preciso reflexão sobre a prática e refazer as metodologias para que possa atingir o sucesso, que é o aprendizado do aluno.

O ensino de algoritmos e programação é fundamental para as disciplinas iniciais dos cursos de computação, como primeira etapa do desenvolvimento do raciocínio lógico e a prática da programação (RAABE; SILVA, 2005).

Segundo Petersen et al. (2011), a comunidade científica de educação em computação tem investido quantidade considerável de energia estudando características dos programadores iniciantes, uma vez que a disciplina

introdutória de algoritmos e programação apresenta altas taxas de reprovação e abandono.

O ensino de algoritmos objetiva a ordenação do pensamento do aluno, fazendo correlação com a mesma sequência lógica utilizada pelo computador (BORGES, 2016).

> A natureza específica da programação, substancialmente diferente da maioria das disciplinas, implica o ensino de muitos conceitos dinâmicos que é, normalmente, realizado através de materiais de natureza estática (apresentações projetadas, explicações verbais, diagramas, desenhos no quadro, textos, e assim por diante) não promovendo uma plena compreensão da dinâmica envolvida. Também a estrutura curricular das disciplinas de programação está na nossa ótica "virada ao contrário", seguindo uma aproximação *bottom-up* quando pensamos que o ideal seria uma abordagem *top-down*. É prática comum começar por ensinar os detalhes sintáticos de uma linguagem de programação, qualquer que ela seja, antes que os alunos percebam qual a finalidade e utilidade de aprender programação. Pensamos que a preocupação principal deveria ser antes de mais nada o desenvolvimento da capacidade de resolução de problemas, aparecendo a linguagem de programação apenas como um veículo para concretizar essa resolução, ou seja, para expressar o algoritmo ou estratégia de resolução (GOMES et al., 2008, p. 95).

De uma forma geral, os alunos têm dificuldade em esquematizar o problema, mesmo quando ele se relaciona com situações comuns da vida cotidiana (SOBRAL; PIMENTA, 2009).

É extremamente importante que os alunos tenham a compreensão completa dos dados do problema e do possível resultado, para que, em seguida, possam pensar no algoritmo ideal para essa transformação.

Mendes (2002) assinala algumas das razões para essas dificuldades, nomeadamente:

- Elevado nível de abstração envolvido;
- Metodologias tradicionais de ensino que privilegiam a aprendizagem de conceitos dinâmicos, utilizando principalmente abordagens e materiais de índole estática;
- Necessidade de um bom nível de conhecimentos e prática de técnicas de resolução de problemas;
- Existência de turmas com alunos com preparação básica, conhecimentos prévios diversificados acerca das matérias em questão, traduzindo-se em ritmos de aprendizagem muito diferenciados;

- Impossibilidade de acompanhamento individualizado ao aluno devido à existência de turmas demasiadamente grandes e ao tempo dentro do currículo escolar ser apertado;
- Exigência de um estudo muito baseado na prática e, por isso, bastante diferente do requerido pela maioria das disciplinas (mais baseado em noções teóricas, implicando muita leitura e alguma memorização);
- Linguagens de programação usuais apresentam sintaxes complexas e não têm representações visuais dos algoritmos, o que não contribui para melhor compreensão.

A metodologia de estudo utilizada pelos alunos também deve ser repensada, substituindo as práticas de leituras sucessivas, memorização de fórmulas e mecanização de procedimentos por uma prática intensiva e reflexão (GOMES et al., 2008).

Estudos realizados por várias instituições mostraram que, mesmo depois de frequentar várias disciplinas envolvendo programação, os alunos ainda ficavam com muitas dificuldades em assuntos básicos a elas relacionados (KINNUNEN; MALMI, 2006). Além disso, muitos alunos não conseguiam sequer compreender um processo de atribuição de valor a uma variável (SIMON, 2011).

Segundo Falckembach e Araújo (2013),

> A dificuldade está no reconhecimento dos procedimentos necessários para se chegar à solução do problema. Isso implica em ter que trabalhar de forma mais eficiente os processos cognitivos, em especial a abstração e a formalização, necessários à construção de um algoritmo, ou seja, à modelagem da solução do problema por meio da técnica de algoritmos pseudocódigos (FALCKEMBACH; ARAÚJO, 2013, p. 1).

Os estudantes também possuem dificuldades em utilizar habilidades prévias, o que provoca medo e frustração, culminando em reprovações, apatia, baixa autoestima, desistência da disciplina e, consequentemente, abandono do curso (CASTRO et al., 2011).

A fim de estimular a construção do conhecimento, é interessante se basear em contextos relacionados ao cotidiano dos alunos, tentando motivá-los a utilizar do conhecimento prévio para chegar à solução de um problema.

Segundo pesquisas realizadas por Giraffa, Muller e Moraes (2015), outros fatores que os alunos apontam como obstáculos para aprendizagem são: deficiências relacionadas aos pré-requisitos de Matemática, o não entendimento dos enunciados, exercícios dissociados de problemas “reais” e pouca disponibilidade extraclasse para estudos.

No ensino a distância, em que, muitas vezes, os alunos não estudam no ambiente correto e não há frequente supervisão do professor, o aprendizado dos conceitos iniciais de programação exigem mais empenho do aluno, devido à falta de conhecimentos específicos da área da computação, como o raciocínio lógico e a organização das ideias (PONTES, 2013).

Dessa forma, é preciso proporcionar estímulos aos estudantes ao estudo da disciplina, usando ferramentas que proporcionem interatividade e retorno imediato, tornando o aprendizado dinâmico (SILVA; PINTO, 2013).

Orehovački e Babić (2015) colocam o desejo de 60% dos alunos da área de Engenharia por disciplinas de programação com maior grau de interatividade, com interfaces amigáveis e divertidas.

> Grande parte do esforço dos desenvolvedores destas ferramentas está em possibilitar ao aluno um ambiente para experimentação das soluções desenvolvidas de forma que este possa expressar uma hipótese de solução, testá-la e depurá-la. Intrinsecamente esta abordagem busca auxiliar o aluno a desenvolver autonomia para construir algoritmos sem a dependência da correção do professor, habilidade esta que será fundamental nas demais disciplinas da área de programação (HOSTINS; RAABE, 2007, p. 97).

Muitas ferramentas utilizadas no auxílio do ensino de programação são geralmente desenvolvidas voltadas a necessidades de programadores profissionais, que já detêm muito conhecimento de conceitos e características que são confusos para os iniciantes.

É importante explicar todas as fases e o processo de aprendizagem a que os alunos serão submetidos, para que eles não se sintam desestimulados durante o seu desenvolvimento. A motivação é um elemento chave que precisa ser visado durante todo o processo.

Aureliano e Tedesco (2012) afirmam que a maior parte das pesquisas no Brasil sobre o ensino aprendizagem de programação para iniciantes, propõe o uso de ferramentas de *software* para o auxílio dos alunos. Nas seções seguintes abordaremos as dificuldades de estudar a distância e os benefícios da utilização da tecnologia no ensino.

2.2 EAD: Dificuldades de estudar a distância

Ao contrário do ensino tradicional, em que alunos e professores se encontram no mesmo ambiente, ao mesmo tempo, na EAD, professores e alunos estão separados física e, às vezes, temporalmente (MOORE et al., 2007).

Devido à flexibilidade dos alunos estudarem em locais e horários que forem mais adequados as suas possibilidades, a EAD tem a capacidade de atingir número maior de pessoas, desde que tenham condições de se comunicar por algum meio (LITTO; FORMIGA, 2012).

Nesse cenário, a EAD desponta como uma modalidade importante para o acesso ao conhecimento, provavelmente vivendo novas etapas, com ênfase na integração de meios, em busca da melhor e maior interatividade.

Com as Tecnologias da Informação e Comunicação (TIC) e a EAD caminhando juntas, as oportunidades de estudo passam a ser mais acessíveis e atingem redes de distribuição de conhecimentos e métodos de aprendizagem inovadores, podendo modificar conceitos ditos tradicionais e propiciar o surgimento de novos modelos de sistemas educacionais (ALMEIDA, 2011).

> O advento das TICs reavivou as práticas de educação a distância devido à flexibilização do tempo, a quebra de barreiras espaciais e a emissão e o recebimento instantâneo de material permitindo realizar tanto as tradicionais formas mecanicistas de transmitir conteúdos, agora digitalizados e hipermidiáticos, como explorar o potencial de interatividade das TICs e desenvolver atividades à distância com base na interação e na produção do conhecimento (ALMEIDA, 2011, p. 7).

Os alunos também tem o perfil modificado, vive-se em uma fase de transição, em que muitos nasceram e cresceram em um meio altamente tecnológico e outros têm pouco conhecimento na área. Pozo e Monereo (2010) nomeiam essa característica como Brecha Digital, afirmando que:

> Da mesma maneira que existem jovens que estabelecem uma relação distante com as TICs, podemos encontrar pessoas de idade mais avançada que desde o começo entram na rede e, atualmente, suas formas de trabalhar, comunicar-se e pensar estão firmemente mediadas por sistemas informatizados (POZO; MONEREO, 2010, p. 98).

Segundo Moore et al. (2007), às vezes, o aluno virtual não se adapta a essa modalidade de educação. Eles também afirmam que:

> Toda pessoa que participa da elaboração e do ensino de cursos precisa compreender as motivações do adulto para participar de um programa de aprendizado à distância e o que significa em termos de criação e veiculação de tais programas. Os adultos possuem muitas preocupações em suas vidas (trabalho, família, vida social), e a educação a distância precisa incluir essas preocupações como recursos na elaboração e instrução, e também como fontes potenciais de problemas que podem dificultar o estudo (MOORE et al., 2007, p. 197).

Além de questões pessoais, como as citadas anteriormente, outro fator que influencia e dificulta a aprendizagem dos alunos é a sua familiaridade com a tecnologia.

Como proposto por Prensky (2001), existem dois públicos distintos, que se relacionam com a tecnologia de diferentes formas: os nativos digitais e os imigrantes digitais, cujas características estão descritas no Quadro 1.

Quadro 1 – Nativos e Imigrantes Digitais

Imigrantes Digitais	**Nativos Digitais**
Se estiver impresso em papel, merece credibilidade	Tudo o que estiver on-line merece credibilidade
O mundo do conhecimento é particular	O mundo do conhecimento é público
Compram pra ler ou ouvir. Sem saber se vão gostar	Leem e ouvem – se gostam compram
São socialmente conservadores: conhecem pessoalmente para depois compartilhar acessos (são eternamente bairristas)	São socialmente liberais: acessam-se mutuamente para depois se conhecer pessoalmente (o mundo é sua vizinhança)
Respeitam e se submetem às hierarquias formais	Confiam primeiro em seus pares. Subvertem hierarquias
Elegem cada aparelho para uma única finalidade. Tem resistência a novidades e aplicativos complementares	Adoram a ideia de *pout-pourri* de funcionalidades num mesmo aparelho. Experimentam aplicativos e abraçam novidades
Reservam o lúdico para o lazer e a recreação	Apreciam o lúdico pra aprender e se sociabilizar
Twitter, Blog, Links: o que é isso?	Têm facilidade em *blogar, twittar e linkar*
São lineares e sequenciais: fazem uma coisa de cada vez	São multitarefas: fazem várias coisas ao mesmo tempo

Fonte: Adaptado de Santos e Domingues, 2015, p. 3.

Essas características dos jovens atuais mostram que existe nova forma de aprender, através de um mundo digital, mais atrativo e que possibilita a utilização de todas as ferramentas disponíveis pelos nativos digitais (CARVALHO; OLIVEIRA; SILVA, 2015).

Conforme Mattar (2010), o aprendizado dos jovens hoje não segue uma estrutura linear, como era antigamente, “eles possuem mentes hipertextuais” (p.10). Aprendem explorando, tentando, mexendo, não leem manuais, aprendem compartilhando.

Os alunos dessa geração, muitas vezes, chegam a sala de aula em um ritmo digital e se deparam com professores imigrantes digitais, muitas vezes, não familiarizados com as TIC, e a escola parece estar em um mundo alheio ao deles.

De acordo com Tori (2010), os professores imigrantes procedentes de uma cultura pré-internet, muitas vezes, não sabem valorizar e usufruir das características dos alunos nativos digitais.

Prensky (2010) conclui que “os nativos estão acostumados a receber informações com muito mais rapidez do que aquela que os imigrantes sabem passá-las" (p. 60).

> Não importa quanto os imigrantes desejem, os Nativos Digitais não voltarão atrás. Em primeiro lugar, não funcionaria: seus cérebros provavelmente já possuem padrões diferentes dos nossos. Em segundo lugar, seria um insulto a tudo que sabemos sobre migração cultural. (...) Adultos imigrantes inteligentes aceitam a ideia de que não sabem tanto a respeito deste novo mundo e aproveitam a ajuda de seus filhos para aprender e integrar-se. Imigrantes não tão inteligentes (...) passam a maior parte de seu tempo lamentando o quanto as coisas eram boas no “velho mundo” (PRENSKY, 2010. p. 60).

A inclusão de recursos tecnológicos que facilitam a interatividade e a colaboração no processo de aprendizagem pode contribuir em melhorias na educação, aproximando a relação entre o aluno Nativo Digital e o professor Imigrante Digital (SANTOS; DOMINGUES, 2015).

Dessa forma, é importante colocar o aluno a par das inovações e possibilidades da EAD, identificando as características necessárias a esse modelo educacional. Isso possibilitará prever e diminuir conflitos e problemas, fortalecendo a motivação, tão importante e necessária no processo de ensino e aprendizagem.

2.3 Vantagens do Uso das Tecnologias no Ensino

As TIC, alavancadas pela rede mundial de computadores, têm modificado o processo ensino-aprendizagem ao longo da história nas mais diversas formas e aspectos da sociedade. Com esses avanços surgem novas possibilidades de aprendizagem, exigindo mudanças das práticas educacionais e docentes (ALONSO; VASCONCELOS, 2012).

Baranauskas et al. (2013) afirmam que as TIC tem provocado profundas mudanças nas relações do homem em sociedade, principalmente no que diz respeito às possibilidades de acesso.

> Até alguns anos atrás, não se tinha noção da ubiquidade que a tecnologia de computadores teria na vida das pessoas, como acontece hoje. O impacto dessa presença é perceptível na forma como interagimos em sociedade, no acesso a serviços corriqueiros, como os de transações bancárias [...], ou no exercício da cidadania elegendo representantes de governo via urna eletrônica. Nossa relação com o conhecimento e com as demais pessoas é atualmente mediada pelas chamadas tecnologias da informação e comunicação (TICs) (BARANAUSKAS et al., 2013, p. 15).

Kenski (2013 p. 61) afirma que “as mudanças trazidas pelas tecnologias digitais transformam nossa cultura”, e que essa transformação tem ocorrido de forma cada vez mais rápida e em vários sentidos. Destaca também que ultimamente se utilizam cada vez mais as “redes digitais (internet) e sua interface gráfica (web) nos sistemas de intercomunicação e em nossas ações cotidianas”.

Segundo Nicolodi (2013):

> A busca pelo entendimento deste contexto tecnológico é fundamental para a compreensão e a readequação dos processos de aprendizagem na sociedade contemporânea. Não se trata mais de nos perguntarmos se devemos ou não utilizar as tecnologias digitais no processo educativo, presentes cotidianamente caracterizando uma nova cultura diretamente condicionada pelo seu desenvolvimento, mas aceitar sua presença e assumir sua relevância, pois é deste contexto que surgem novas práticas comunicacionais, novas formas de ser e estar no espaço e tempo, novas possibilidades de interação, consequentemente novas formas de aprender (NICOLODI, 2013, p. 9).

De acordo com Siemens (2004 p. 1), “a aprendizagem é uma atividade constante na vida humana, é contínua, permanente e fluida, e a tecnologia a potencializa cada vez mais, ligando os indivíduos e conectando-o a diversas áreas”.

É preciso refletir sobre o uso de *softwares* educativos e de que forma eles contribuirão para que os alunos ultrapassem os obstáculos e construam conhecimentos com flexibilidade, liberdade e prazer (RIOS, 2015).

Na Educação a Distância, é de extrema importância o uso de ferramentas que facilitem a comunicação e a interação com e entre os alunos. A maioria dos estudantes de hoje possui uma característica que os diferencia dos demais: eles cresceram cercados pela tecnologia e Internet. São os chamados “nativos digitais” (PRENSKY, 2001). São estudantes que estão habituados a utilizar redes sociais, aplicativos de mensagem instantânea (chats), buscas e compartilhamento de informações (*blogs, sites*) e é possível usar esse conhecimento sobre estas ferramentas a favor do ensino (PRENSKY, 2005).

De acordo com Kenski (2013), é importante que os cursos a distância se utilizem de princípios pedagógicos que promovam o interesse do educando pelas propostas de aprendizagem inovadoras. Caso contrário, os alunos perderão o interesse pelo curso, chegando até mesmo à evasão.

Para Tori (2010), os materiais disponibilizados através de atividades virtuais promovem maior aproximação entre aluno e conteúdo de aprendizagem, facilitando o atendimento a diferentes estilos e ritmos de aprendizagem e contribuindo com a produtividade do professor e do aprendiz.

> As tecnologias podem auxiliar no processo de aprendizado tanto quanto o papel já ajudou. Serão úteis se usadas de maneira apropriada. Creio que existem três frentes em que os novos recursos técnicos são relevantes: no acesso à informação, na comunicação (entendida como troca de conhecimento) e como meio para que as pessoas exerçam sua criatividade. Infelizmente, a maior parte do aparato moderno se limita a fazer coisas velhas de forma diferente. Com frequência, as pessoas pensam na educação como o repasse de informações para o estudante. Esquecem-se de que as mais importantes experiências ocorrem quando o aluno está ativamente engajado em projetar, criar e experimentar. Só aproveitaremos o potencial dos computadores quando pararmos de pensar neles como espécies de televisores e começarmos a enxergá-los como pincéis. Ou seja, como meios para a expressão criativa (RESNICK, 2006, p. 90).

É preciso engajar os alunos a utilizarem as tecnologias de forma inovadora e produtiva, promovendo experiências criativas e conduzindo esses jovens à novas e infinitas possibilidades de aprender (RESNICK, 2006).

O resultado positivo do uso das ferramentas computacionais, citado por vários autores estudados nesta seção, incita o desenvolvimento de uma

ferramenta tecnológica que se proponha a minimizar as dificuldades encontradas pelos alunos no aprendizado de conceitos básicos de programação. Para tanto, a seção seguinte apresenta *softwares* existentes que serviram de base para esse estudo.

2.4 Ferramentas Utilizadas no Ensino de Programação

Esta seção aborda algumas ferramentas tecnológicas existentes que são utilizadas no apoio ao ensino das disciplinas de programação, abordando suas principais características, pontos positivos e negativos.

As ferramentas foram selecionadas com base em Revisões Sistemáticas da Literatura realizadas por Silva et al. (2015) e Ramos et al. (2015), que analisaram iniciativas brasileiras na criação de ambientes de aprendizagem de programação, nos principais eventos brasileiros da área, que englobam o Congresso Brasileiro de Informática na Educação (CBIE) e a Revista Brasileira de Informática na Educação (RBIE).

Foi excluída deste trabalho a avaliação de *softwares* criados antes de 2010.

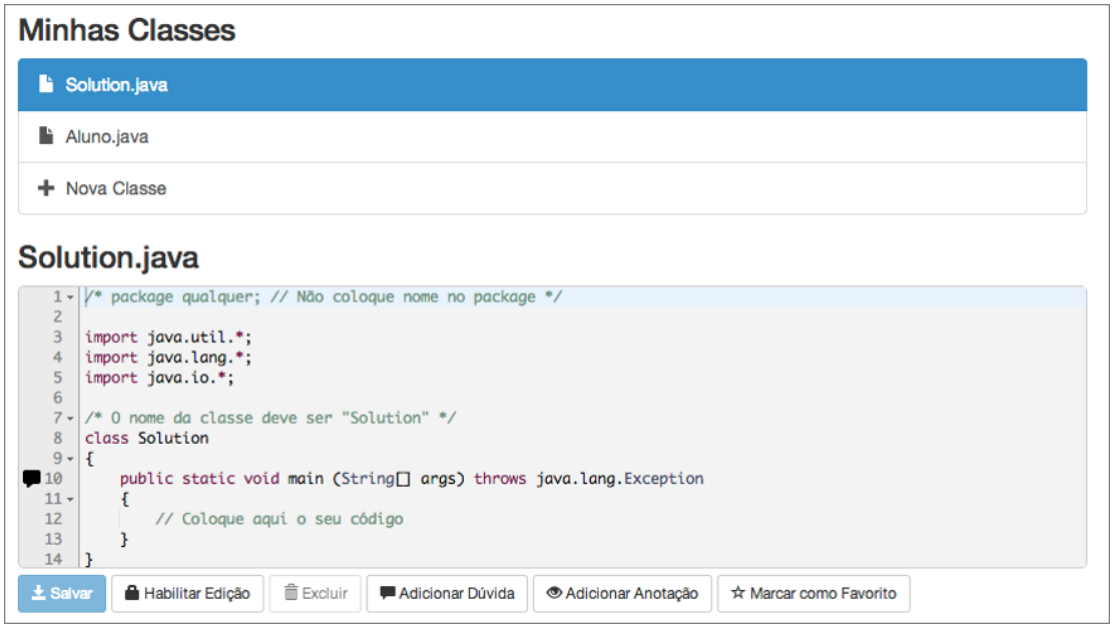
2.4.1 Ambiente Virtual Feedback Personalizável (Feeper)

O *Feeper* é um ambiente virtual para apoiar o ensino/aprendizagem de programação no ensino superior. Utiliza mecanismo para melhorar a comunicação entre aluno e professor, permitindo sinalizar partes do código fonte (ALVES; JAQUES, 2014).

A ferramenta funciona com o envio do código fonte produzido pelo aluno para validação de um Juiz On-line (Figura 1), este então solicita ao banco de dados os testes existentes para o exercício que o aluno está tentando responder. O Juiz On-line compara a resposta gerada pelo programa do aluno com as respostas esperadas, caso haja divergência, uma mensagem personalizada de alerta é emitida ao aluno, contendo orientações para que este consiga resolver o problema gerado, do contrário, então o programa é aceito e a resposta de sucesso é emitida ao aluno.

O *feedback* fornecido pelo Juiz On-line foi personalizado para apresentar mensagens que indiquem ao aluno onde aconteceu o erro, por que ele aconteceu e como corrigi-lo e para permitir a execução de diferentes tipos de teste no programa do aluno (ALVES; JAQUES, 2014).

Figura 1: Interface do *Feeper*.



Fonte: FEEPER, 2016.[2]

Através do *Feeper* o aluno pratica conceitos de programação orientada a objetos, utilizando a linguagem Java. Além da correção automática, a ferramenta emite dicas sobre os erros gerados pelo programa do aluno permitindo que o mesmo possa adicionar marcações em seu programa que serão exibidas ao professor. (ALVES; JAQUES, 2014).

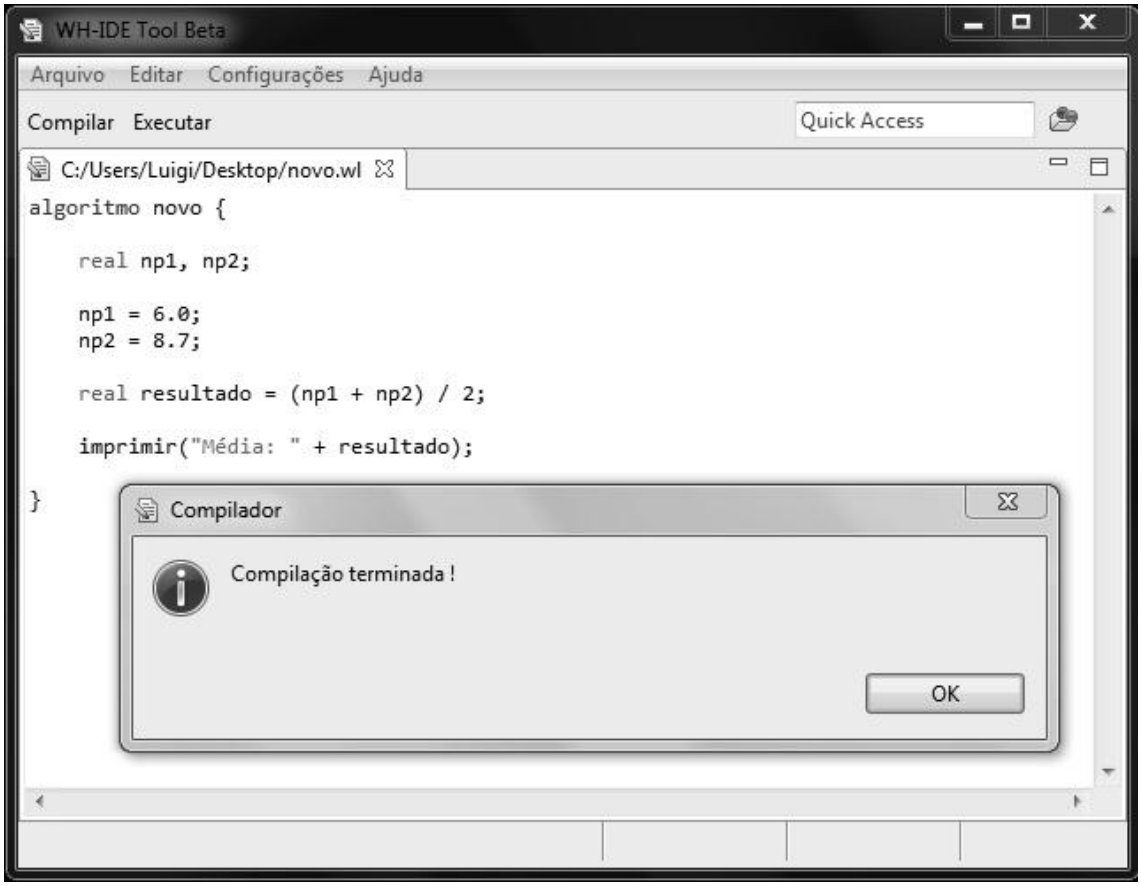
2.4.2 Ambiente Personalizável para Aprendizado de Programação – WH-IDE

É um ambiente para ensino de programação de computadores, no qual os alunos programam os algoritmos em português. O professor define todas as instruções que estarão à disposição dos alunos, personalizando, assim, a linguagem a ser utilizada e ignorando o máximo de detalhes técnicos. (FRANTZ; PONTES, 2014).

---

2 http://feeper.unisinos.br Acesso em: 21/02/2016.

Figura 2: Tela do WH-IDE



Fonte: FRANTZ, PONTES (2014) p. 5.

A ferramenta possui algumas limitações: ela não captura possíveis erros de sintaxe existentes no código, pelo fato de usar o compilador da linguagem Java para a geração do programa executável, além de não possuir depurador visual e área para exibição das variáveis criadas em memória, como se pode observar na Figura 2 (FRANTZ; PONTES, 2014).

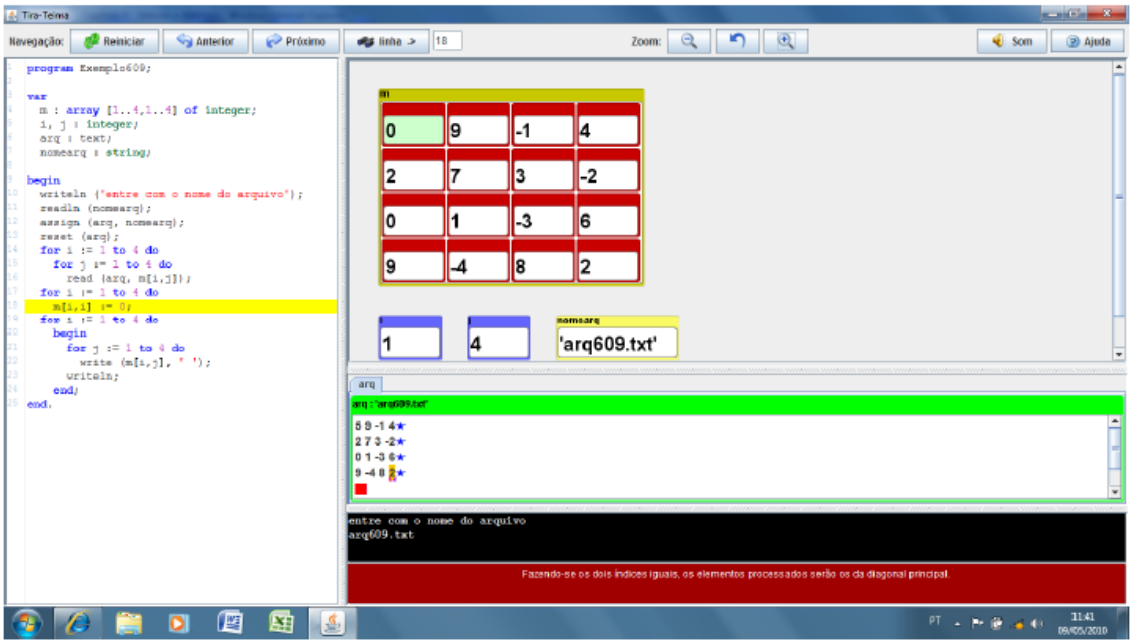
2.4.3 Ambiente Interativo e Adaptável para ensino de Programação - TutorICC

O TutorICC é um ambiente interativo e adaptável utilizado para o ensino de programação em Pascal. O conteúdo está dividido em níveis de dificuldade, permitindo ao aluno estabelecer o caminho mais conveniente para ele, dentro do conteúdo da disciplina (PÍCCOLO et al., 2010).

Há grande interatividade com o aluno, a cada passo é apresentado ao aluno um problema, para que ele construa um programa. Esses programas são

corrigidos imediatamente pelo TutorICC, que recomenda ao aluno o melhor caminho a seguir dentro do conteúdo da disciplina.

Figura 3: Interface do TutorICC



Fonte: PÍCCOLO et al, (2010) p. 5.

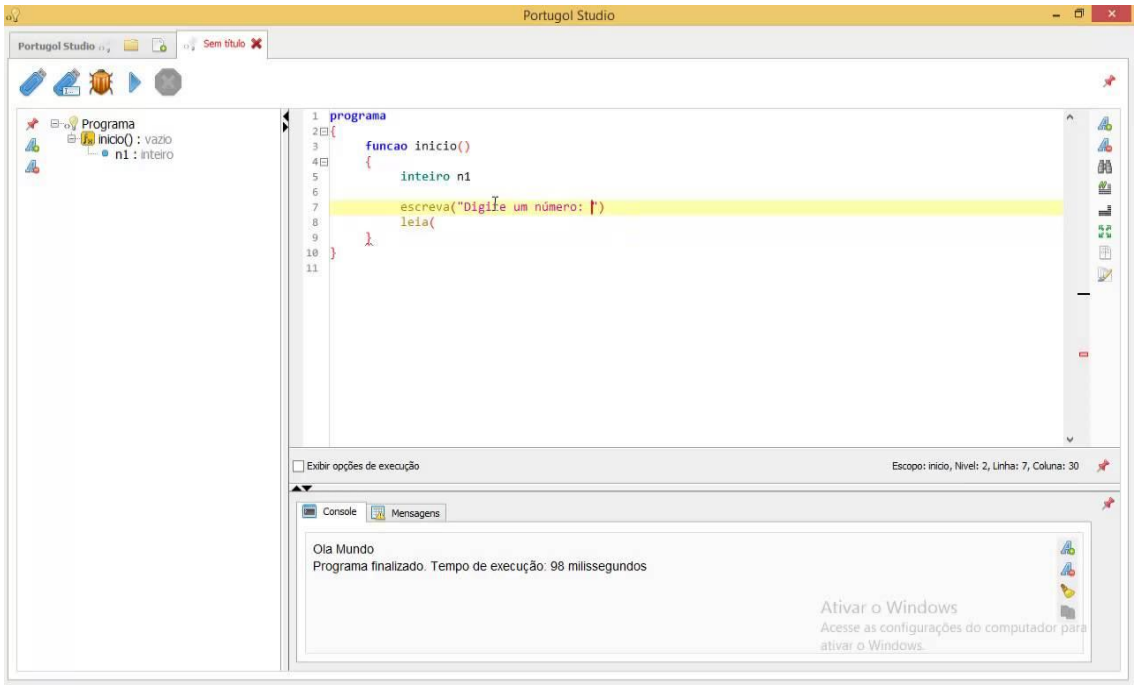
A Figura 3 apresenta a interface da ferramenta, que foi desenvolvida utilizando a linguagem Java. A principal contribuição desta em relação aos demais trabalhos encontrados na literatura, é que permite que o professor personalize a apresentação visual em cada exemplo. Pode escolher posição, tamanho e cor de cada uma das variáveis, dispondo-as da forma que achar mais didática para captar a atenção dos alunos (PÍCCOLO et al., 2010).

2.4.4 Portugol Studio

O Portugol Studio é um Ambiente Integrado de Desenvolvimento (IDE) para auxiliar a aprendizagem de algoritmos dos alunos iniciantes. Foi desenvolvida no idioma português, facilitando o acesso ao público com baixa fluência no idioma inglês, como se pode observar na Figura 4 (NOSCHANG et al., 2014).

A IDE possibilita o desenvolvimento, a execução e a depuração de programas escritos em *Portugol*, sendo eficaz para iniciantes em programação, principalmente por utilizar o português como língua nativa (ANIDO, 2015).

Figura 4: Interface do *Portugol Studio*



Fonte: *PORTUGOL STUDIO*, 2016. [3]

O fato de ser um aplicativo para *desktop* dificulta um pouco para que o aluno possa praticar em casa, pois é necessário instalar as ferramentas do ambiente (editor, compilador, IDE, etc.), o que pode ser uma tarefa complexa (ANIDO, 2015).

2.4.5 Ambiente de Aprendizagem de Programação Web - AAPW

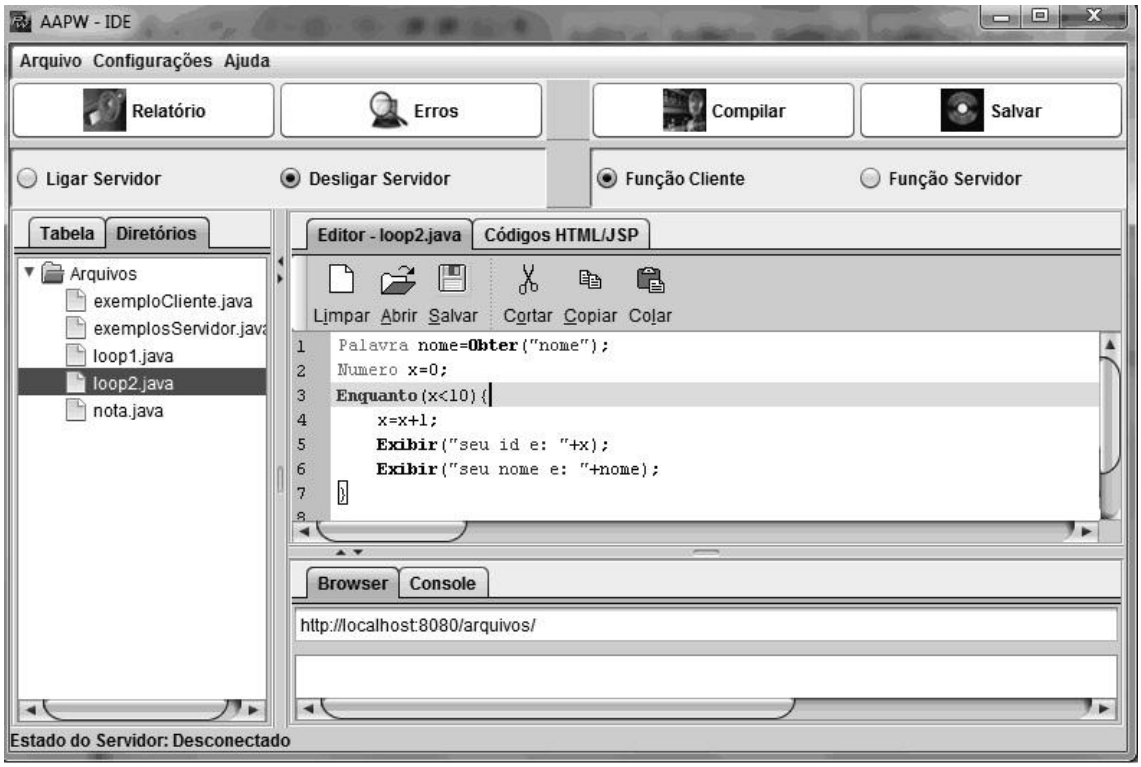
O AAPW permite o aprendizado de conceitos envolvidos na programação Web, de forma simples e objetiva, podendo ser utilizada como recurso auxiliar em disciplinas de cursos que abordam desenvolvimento Web. (GOMES; D'EMERY; FILHO, 2014).

A ferramenta (Figura 5) possui uma IDE de programação desenvolvida em Java, que é composta de barra de ferramentas; das telas de edição do programa fonte e do código compilado; de um *browser* para visualizar as páginas desenvolvidas; e de um console que exibe mensagens de erros presentes no código pertencente ao programa fonte.

---

3 http://lite.acad.univali.br/portugol/ Acesso em: 02/03/2016.

Figura 5: Interface do AAPW



Fonte: GOMES, D'EMERY, FILHO, (2014), p. 6.

A ferramenta trabalha com formulários, não necessita de *tags* no código e utiliza instruções simples com comandos em português. A IDE propicia ao usuário a visualização tanto do código construído em português quanto da codificação gerada, a qual pode ser interpretada em um *browser*. (GOMES; D'EMERY; FILHO, 2014).

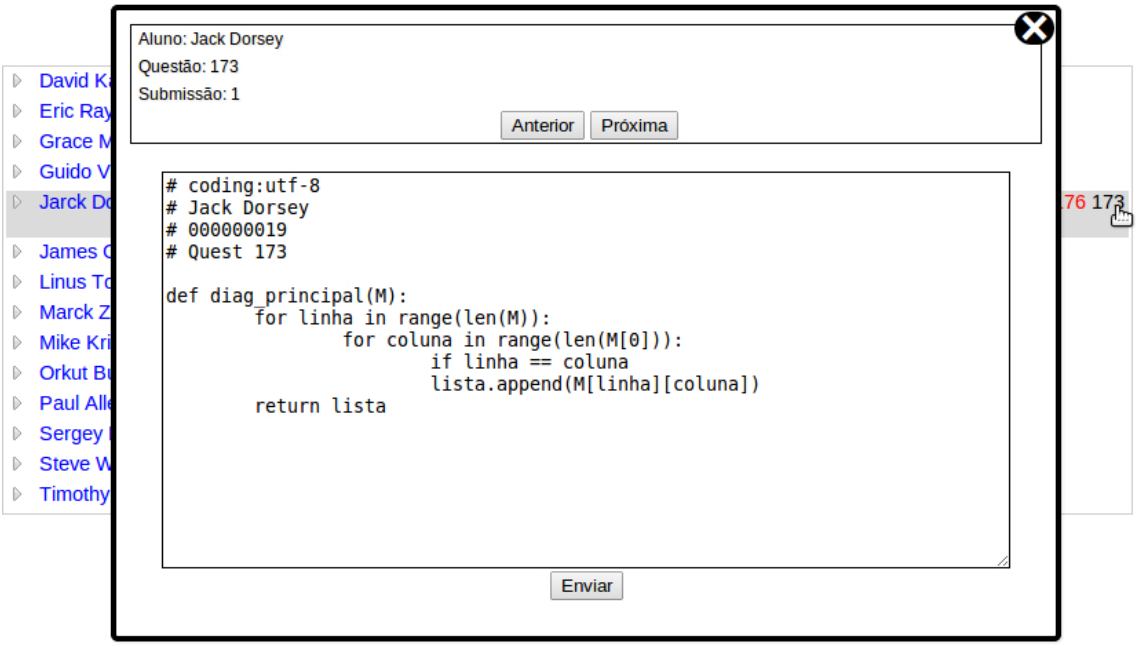
2.4.6 Ferramenta de Acompanhamento de Exercícios de Programação - TSTView

O TSTView permite acompanhar o desempenho dos estudantes em um certo período de tempo, além de monitorar em tempo real o desenvolvimento dos estudantes frente aos exercícios planejados possibilitando detectar, com mais facilidade e eficiência, dificuldades individuais dos estudantes, dificuldades comuns a grupos de estudantes e dificuldades gerais da turma (GAUDENCIO, 2013).

Os alunos respondem aos exercícios propostos utilizando a linguagem de programação *Python* e submetem a um testador automático que aplica os testes criados pelos professores para as questões propostas. Os dados de

controle e os resultados dos testes aplicados as submissões compõem a sua base de dados. Como se pode visualizar na Figura 6.

Figura 6: Tela do TSTView



Fonte: TSTView, 2016.[4]

A ferramenta possui dois módulos principais, o do professor e o do aluno. O primeiro exibe uma lista dos alunos que estão realizando alguma atividade no momento da consulta. Já o segundo, exibe um relatório histórico individual para cada aluno (GAUDENCIO, 2013).

Ao submeter uma solução, o estudante recebe feedback na forma de resultados de testes sobre seu código, podendo visualizar que tipos de erros ele cometeu e o número de casos de teste em que falhou.

2.4.7 Ferramenta para a Avaliação de Aprendizado de Alunos em Programação de Computadores - *The Huxley*

The Huxley é uma ferramenta web, como é exibido na Figura 7, que permite aos alunos responderem a exercícios de uma base de dados com centenas de problemas e submeterem às respostas através de código em diversas linguagens de programação.

O ambiente corrige os exercícios automaticamente através de análise sintática do código e testes de aceitação e emite *feedback* para o aluno. O

[4] http://tstview.appspot.com Acesso em: 02/03/2016.

professor tem acesso à quantidade de problemas resolvidos, porcentagem de acertos/erros, tipos de problemas com mais erros, detecção de plágio e erros específicos de cada aluno (PAES, 2013).

Figura 7: Página Inicial do The Huxley

Fonte: The Huxley, 2016.[5]

Através do ambiente, o professor faz uma avaliação personalizada do aluno, baseada na identificação de erros específicos durante as avaliações. A definição e correção de exercícios e provas, verificação de plágio, definição de notas e publicação dos resultados pode ser feita automaticamente através do *The Huxley* (PAES, 2013).

2.4.8 Sistema On-line de Atividades de Programação - SOAP

É um sistema *web* que possibilita ao professor disponibilizar tarefas para as turmas e, ao aluno, realizar submissões de exercícios, como se pode observar na Figura 8. Os exercícios submetidos, que são programas de computador em Linguagem C, são compilados e executados em um servidor remoto, gerando relatórios que são disponibilizados aos professores.

As atividades dos alunos são processadas pelos núcleos (Núcleo de Avaliação Diagnóstica NAD) e Núcleo de Avaliação Formativa (NAF). O NAD processa essas atividades, gerando planilhas e mapas de perfis de aprendizagem, o NAF, com base no diagnóstico dos alunos fornecido pelo

---

5 http://www.thehuxley.com Acesso em: 05/03/2016.

NAD, recomenda atividades diretamente para que os alunos com dificuldades possam melhorar os desempenhos (OLIVEIRA; OLIVEIRA, 2014).

Figura 8: Interface do SOAP

Tarefas

Tarefa 1 - 1a. Lista de Exercicios - Linguagem C

UFES SOAP Sistema Online de Atividades de Programação

Avaliação de desempenhos

Tarefa: 1

Questão	Situação	Arquivos submetidos	Saida	Nota	Feedback
A6-1	NÃO FEZ	0	*	*	Não Há
A6-2	NÃO FEZ	0	*	*	Não Há
A6-3	FEZ	1	Arqsaida	2	4
A6-4	NÃO FEZ	0	*	*	Não Há
A6-5	NÃO FEZ	0	*	*	Não Há
A6-6	NÃO FEZ	0	*	*	Não Há

Fonte: SOAP, 2016. [6]

O *software* combina técnicas de reconhecimento de padrões para classificar alunos e recomendar-lhes atividades conforme os seus perfis e possibilita ao professor realizar melhor gestão da aprendizagem individual de seus alunos (OLIVEIRA; OLIVEIRA, 2014).

2.4.9 Ambiente de Apoio ao Ensino de Programação e Teste de Software - ProgTest

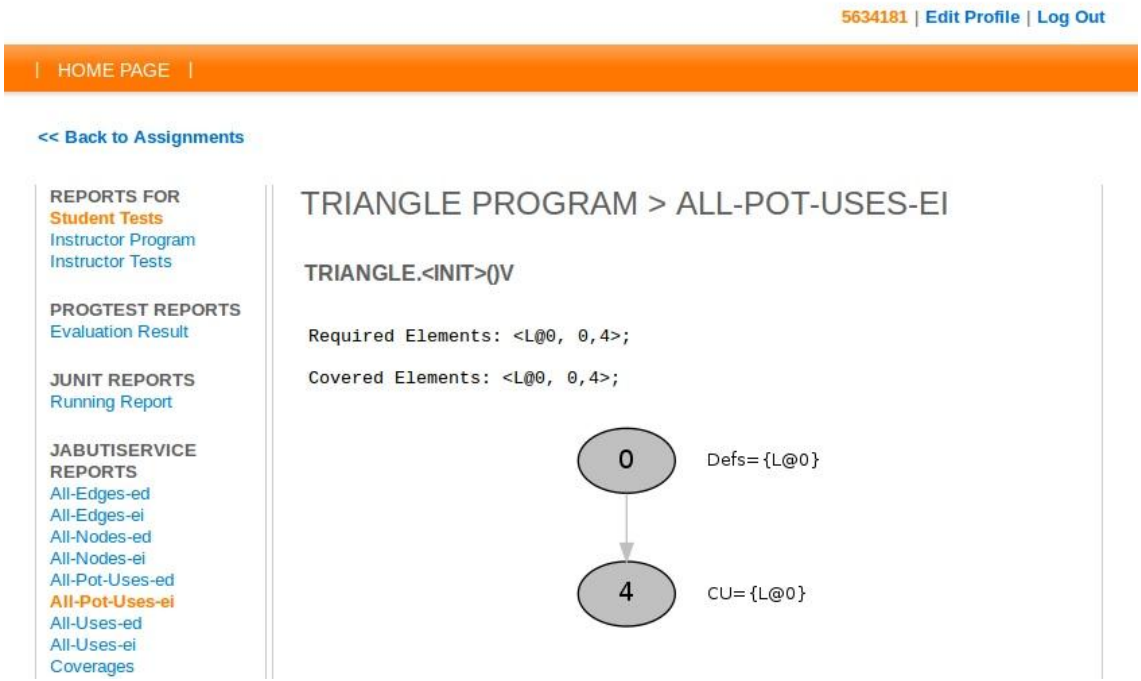
A *ProgTest* é um ambiente web (Figura 9) para submissão e avaliação automática de trabalhos de programação. A ideia é fornecer apoio automatizado para avaliar os programas e casos de teste submetidos pelos alunos (SOUZA; MALDONADO; BARBOSA, 2012).

O professor pode criar novos cursos, matricular alunos e definir trabalhos de programação, especificando os critérios de teste que deverão ser utilizados durante a avaliação, além do peso que este terá durante a avaliação. O aluno tem acesso a todos os trabalhos associados aos cursos em que ele está matriculado e ao submeter uma solução para um dado trabalho, ele envia o programa e o teste utilizado.

---

6 https://portal.ufes.fibre.org.br/soap Acesso em: 13/03/2016.

Figura 9: Interface do ProgTest



Fonte: ProgTest, 2016. [7]

A primeira versão da *ProgTest* apoiava o teste de programas escritos apenas em linguagem Java, dificultando o uso efetivo do ambiente em disciplinas introdutórias de programação. Por esse motivo, a ferramenta foi evoluída a fim de possibilitar o apoio a linguagens de programação procedimental (em especial, C) (SOUZA; MALDONADO; BARBOSA, 2012).

2.4.10 Codecademy

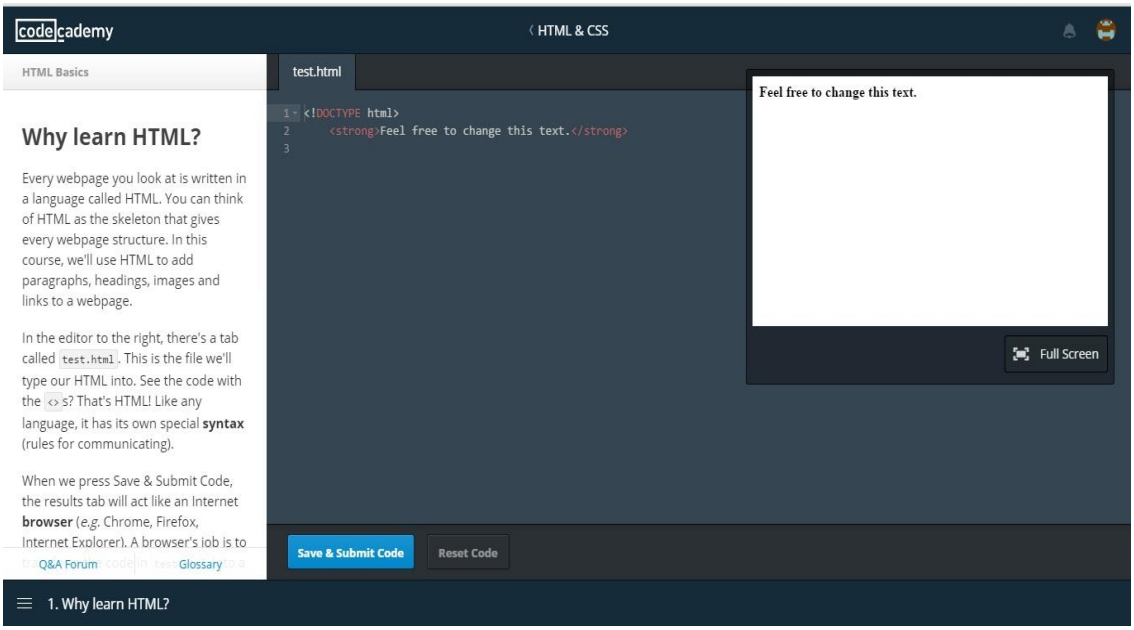
Um sistema baseado na web feito para seguir uma sequência lógica de "passo a passo", para ensinar linguagens de programação, como *Javascript, jQuery, Ruby* e *PHP*, promovendo uma intensa interação com o usuário através de desafios e exercícios com *feedback* de acertos e erros (SILVA, 2014).

A intenção é permitir aos usuários interagir com o sistema, como na Figura 10, digitando os comandos na linguagem de programação que ele está tentando aprender e analisando os resultados em tempo real na tela do browser (SILVA, 2014).

---

[7] https://www.sites.google.com/site/labcaed/ferramentas Acesso em: 13/03/2016.

Figura 10: Tela do Codecademy



Fonte: Codecademy, 2016.[8]

Segundo os criadores da ferramenta, mesmo pessoas com o mínimo conhecimento em programação pode obter êxito com o *Codecademy*. As aulas de *Python*, por exemplo, começam com lições bem simples, de sintaxe e variáveis, para depois abordar estruturas condicionais, funções, laços de repetição e assuntos um pouco mais avançados, como programação orientada a objetos e manipulação de arquivos.

2.4.11 *Scratch*

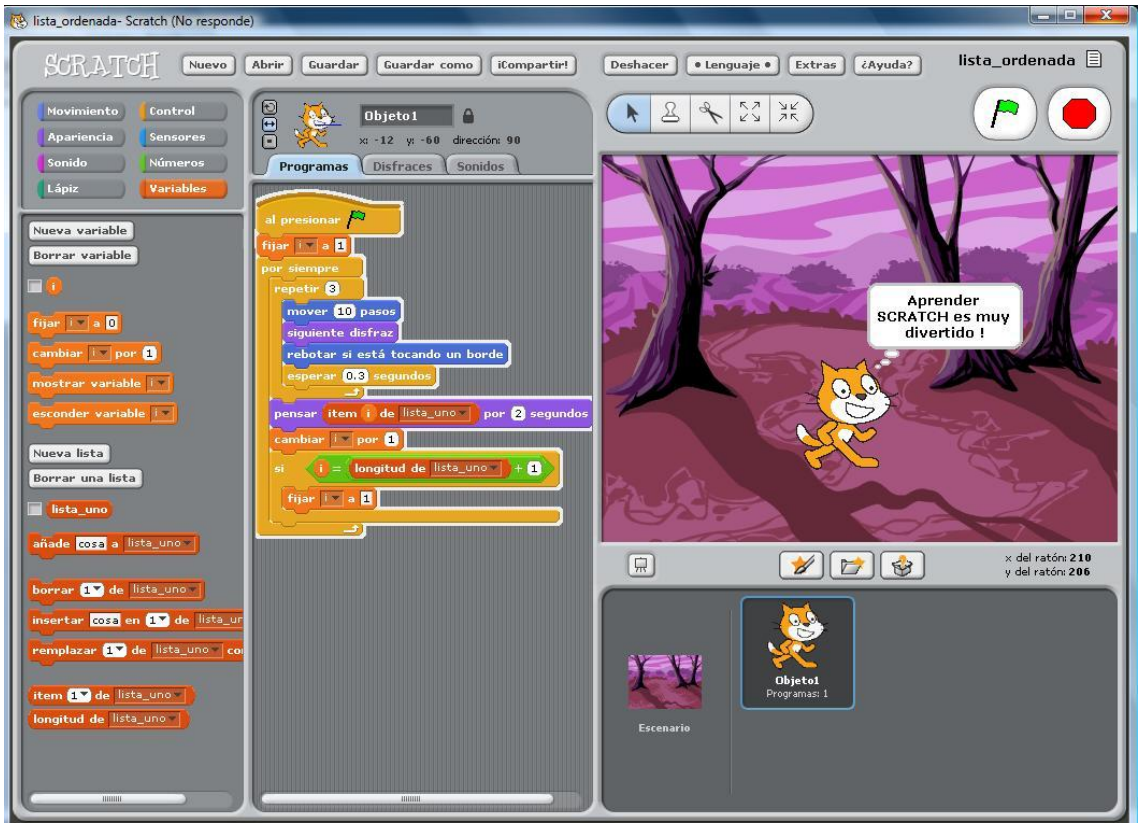
O ambiente de programação Scratch traz linguagem que contribui para a aprendizagem de programação através da criação de histórias, animações, simulações e outros produtos multimídia, arrastando e agrupando diferentes conjuntos de blocos de comandos coloridos (MARQUES, 2013).

Pelo fato de os comandos Scratch se basearem em uma estrutura de blocos de montar, o aluno pode concentrar esforço na busca pela construção do algoritmo, não se preocupando com erros de sintaxe. Desta maneira, pode-se afirmar que o caráter mais didático do *Scratch* projeta no aluno a possibilidade de ele se concentrar no exercício do pensamento algorítmico e na criatividade durante a construção das soluções (SCAICO et al., 2013).

---

8 www.codecademy.com Acesso em: 15/03/2016.

Figura 11: Interface do *Scratch*



Fonte: *Scratch*, 2016.[9]

Como se pode observar na Figura 11, o *Scratch* possui interface muito amigável e indicada para iniciantes ou pessoas com pouco conhecimento em Lógica de Programação. Há versão em português e é gratuito.

2.4.12 Alice

Alice é um ambiente de programação 3D que permite a criação de animações e possibilita o aprendizado de conceitos fundamentais da programação tradicional no contexto de criar vídeos animados ou jogos simples (CRUZ; LIMA, 2014).

Em Alice, os objetos 3D fazem parte de um mundo virtual, no qual os usuários criam programas para animar esses objetos (VALASKI; PARAISO, 2012).

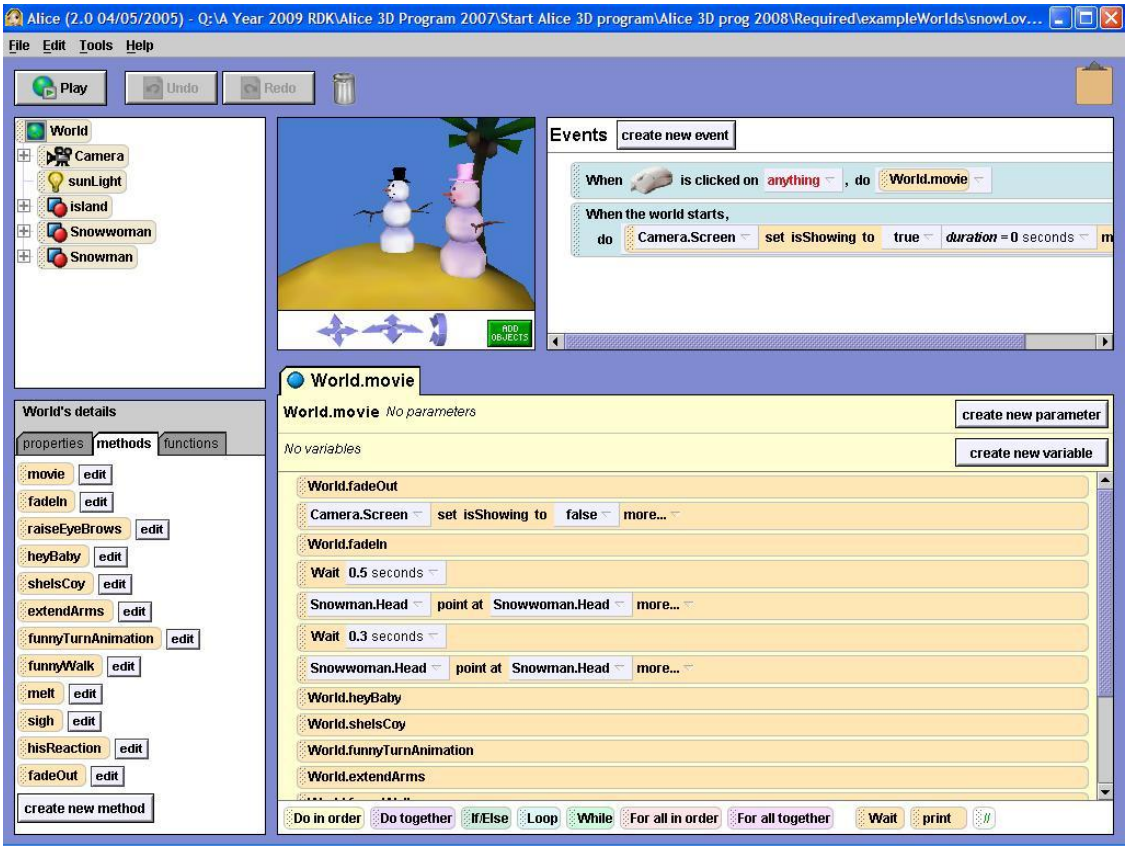
Assim como o *Scratch*, o Alice é um *software* de autoria que se baseia em um formato de montar blocos e pilhas de códigos, como se visualiza na Figura 12, mostrando sempre uma resposta visual do que determinado bloco

---

9 https://scratch.mit.edu/ Acesso em: 15/03/2016.

faz, permitindo aos usuários relacionar a programação com os seus resultados (CRUZ; LIMA, 2014).

Figura 12: Interface do Alice



Fonte: Alice, 2016.[10]

O ambiente Alice trata dos principais conceitos de programação orientada a objetos, como classes, objetos, métodos e funções. Não é necessário seguir uma sintaxe para programar e os alunos se sentem motivados, pois adoram contar histórias.

2.4.13 Comparativo das Ferramentas

Os *softwares* apresentados compartilham objetivos, como desenvolver a criatividade, o raciocínio, a descoberta e interação social. Todos exploram as interfaces visuais como meio de motivar a aprendizagem dos conceitos de programação. O que os difere, entretanto, é que uns são apenas IDE, onde o aluno pode escrever e testar códigos, alguns são voltados para o ensino de

---

10 www.alice.org Acesso em: 18/03/2016.

lógica de programação e outros para conceitos mais avançados de programação, como orientação a objetos.

Para uma efetiva comparação das ferramentas analisadas, foram utilizados alguns critérios como a forma de interação do usuário com o ambiente, a plataforma de execução da ferramenta, as linguagens de programação suportadas e se está disponível no idioma Português.

No Quadro 2, pode-se observar a relação das ferramentas e os critérios que foram utilizados na análise.

Quadro 2 – Características Técnicas das Ferramentas

Ferramenta	**Forma de Interação**	**Plataforma**	**Linguagem de Programação**	**Disponível em Português**
Feeper	Textual	Web	*Java*	Sim
WH-IDE	Textual	Java	*Portugol*	Sim
TutorICC	Visual e Textual	Java	*Pascal*	Sim
Portugol Studio	Visual e Textual	Java	*Portugol*	Sim
AAPW	Visual e Textual	Java	*Portugol*	Sim
TSTView	Visual e Textual	Web	*Phyton*	Sim
The Huxley	Textual	Web	Várias	Sim
SOAP	Visual e Textual	Web	C	Sim
ProgTest	Textual	Web	C	Não
Codecademy	Visual e Textual	Web	Várias	Sim
Scratch	Visual e Textual	Web	Blocos	Sim
Alice	Visual e Textual	Java	Blocos	Não

Fonte: Elaborado pelo autor, 2016.

Com exceção do ProgTest e do Alice, todos os demais possuem versões no idioma português. A maioria deles é composta de recursos visuais e textuais e podem ser executados na web através de um *browser*. Embora, todos os *softwares* apresentados sejam indicados para iniciantes, apenas três deles utilizam a linguagem *Portugol*, que é um fator considerado importante em disciplinas introdutórias.

Na literatura sobre aprendizagem de programação, encontram-se relatos de que o processo de conhecer e se habituar aos ambientes e às linguagens é uma tarefa árdua, com grande dispendiosidade de tempo por parte dos alunos e, por vezes, dos professores ou tutores (RIBEIRO, 2015).

No que se refere as funcionalidades implementadas, veriificou-se acerca da possibilidade do usuário visualizar os valores das variáveis durante a execução de um algoritmo, o uso de animações e vídeos na explicação dos conceitos e técnicas utilizadas, a forma utilizada para emitir *feedback* ao aluno e também se possuem ou não características de adaptatividade.

No Quadro 3, tem-se a comparação com relação às funcionalidades disponíveis:

Quadro 3 – Funcionalidades

Ferramenta	**Visualização de variáveis**	**Animações**	**Vídeos**	***Feedback***	**Adaptatividade**
Feeper	NÃO	NÃO	NÃO	Manual	NÃO
WH-IDE	NÃO	NÃO	NÃO	Manual	NÃO
TutorICC	SIM	SIM	NÃO	Automático	SIM
Portugol Studio	SIM	NÃO	NÃO	Manual	NÃO
AAPW	NÃO	NÃO	NÃO	Manual	NÃO
TSTView	NÃO	NÃO	NÃO	Automático	NÃO
The Huxley	NÃO	NÃO	NÃO	Automático	NÃO
SOAP	NÃO	NÃO	NÃO	Automático	NÃO
ProgTest	NÃO	NÃO	NÃO	Automático	NÃO
Codecademy	NÃO	SIM	SIM	Automático	SIM
Scratch	NÃO	SIM	SIM	Automático	NÃO
Alice	NÃO	SIM	SIM	Automático	NÃO

Fonte: Elaborado pelo autor, 2016.

Pode-se perceber que a maioria das ferramentas não utiliza recursos de animações e vídeos para apresentação de conteúdos, explicação dos códigos e funcionamento das variáveis.

A animação de algoritmos é importante, pois, muitas vezes, o professor tem a necessidade de usar representações visuais para ajudar os alunos a entenderem os algoritmos, o que na prática se concretiza, muitas vezes, mostrando o comportamento de programas (TAVARES et al., 2016).

Outra característica apresentada pela maioria dos *softwares* estudados é o *feedback* automático. Segundo Tavares et al. (2016), *feedback* imediato é crucial para a aquisição de conhecimento, independentemente da estratégia de aprendizagem adotada, pois motiva os alunos. Por outro lado, o *feedback* individual pode consumir muito tempo do professor com o risco de que os alunos possam não se beneficiar dele no devido tempo (QUEIRÓS; LEAL, 2015).

Com relação à adaptatividade, apenas as ferramentas TutorICC e Codecademy apresentaram essa funcionalidade, no qual o ritmo de aprendizagem se adapta ao perfil do aluno, podendo ser mais rápido ou mais lento, dependendo de sua maior ou menor facilidade em aprender programação.

A adaptatividade tem como propósito ajustar-se a diferentes níveis do educando, com diferentes conhecimentos prévios, facilitando o aprendizado e reforçando os conceitos apresentados nas aulas expositivas (BRAGA, 2015).

Segundo Gomes et al. (2008), para se atingir a capacidade de resolução de problemas são necessárias três fases: (1) resolução de problemas de diversos domínios, não utilizando algoritmos ou programação; em seguida, (2) uma amostra da utilidade da programação com aplicação dos conhecimentos adquiridos na fase anterior; e, finalmente, (3) passando para a construção dos algoritmos.

Na fase de resolução de problemas, a preocupação maior é com o tipo de problemas que serão apresentados aos alunos. A utilidade da programação tem por objetivo mostrar aos alunos para que serve e como se faz um programa. E na fase de construção os alunos teriam que analisar a resolução de problemas através de programas de computador e também propor e testar novas soluções.

O Quadro 4 analisa se as ferramentas estudadas possibilitam a aplicação dessas fases:

Quadro 4 – Competências trabalhadas (GOMES et al., 2008)

Ferramenta	**Resolução de problemas de domínios diversos**	**Utilidade da programação**	**Construção dos algoritmos**
Feeper	NÃO	NÃO	SIM
WH-IDE	NÃO	SIM	SIM
TutorICC	NÃO	NÃO	SIM
Portugol Studio	NÃO	SIM	SIM
AAPW	NÃO	SIM	SIM
TSTView	NÃO	NÃO	SIM
The Huxley	NÃO	NÃO	SIM
SOAP	NÃO	SIM	SIM
ProgTest	NÃO	SIM	SIM
Codecademy	NÃO	SIM	SIM
Scratch	SIM	SIM	NÃO
Alice	SIM	SIM	NÃO

Fonte: Elaborado pelo autor, 2016.

Observa-se que a grande maioria das ferramentas estudadas prioriza a etapa de construção dos algoritmos, sem a devida importância às competências anteriores.

Em disciplinas iniciais de programação o foco principal deve ser a capacidade de resolução de problemas, e o uso da linguagem de programação fica para fases posteriores, como meio de concretizar essa resolução através de um algoritmo (BORGES, 2016).

Embora todos os *softwares* apresentados sejam indicados para iniciantes, o *Codecademy* é preferencialmente aplicado para aprendizes que já tenham certa proficiência nos conceitos de lógica e desejem aprender alguma linguagem específica. Apenas o *software* Alice introduz conceitos de orientação a objetos.

Dentre as ferramentas citadas, as que mais se assemelham a discutida neste trabalho são *Portugol Studio* e *Codecademy*.

A ferramenta *Portugol Studio* prioriza o uso do idioma português, e possui diversos elementos presentes nas IDE profissionais, mas preservando sempre o enfoque na aprendizagem dos novatos em programação. Alguns elementos da interface auxiliam o docente a explicar processos como a depuração.

A Escola de Algoritmos também utiliza o idioma português e é voltada para iniciantes em programação, não funciona como IDE, mas possui recursos que possibilitam aos alunos a verificação e correção de blocos de códigos e animações que demonstrem a sua execução. Além de não necessitar da instalação da Máquina Virtual Java para utilização.

Com relação ao *Codecademy*, as ferramentas se assemelham bastante, pois apresentam conteúdos textuais e recursos interativos acessíveis através da web, mas é voltado para linguagens de programação específicas, como *HTML*, *Javascript, Ruby* e *Java*.

Embora alguns dos projetos encontrados na literatura sejam extremamente exitosos, ainda se pode notar algumas deficiências, como: (1) foco na capacidade de resolução de problemas, (2) uso de animações e vídeos para explicação de conteúdos complexos e (3) utilização do *Portugol*, que motivaram a busca por uma nova abordagem e o desenvolvimento de uma nova ferramenta que se encontra no Capítulo 4 deste trabalho.

3. METODOLOGIA

Este capítulo descreve a metodologia e as técnicas de pesquisa utilizadas. Ademais, apresenta alguns dados e discussões acerca do problema da pesquisa.

3.1 Materiais e Métodos

Como foi visto anteriormente neste trabalho, muitas são as dificuldades que os estudantes habitualmente enfrentam nas disciplinas iniciais de programação. As causas dessas dificuldades têm sido objeto de estudo, sendo apontadas diferentes razões.

Com vista a amenizar os problemas citados, surgiu a motivação para este trabalho. Desenvolver uma ferramenta que possa servir de ligação entre os conteúdos teóricos vistos nos slides e apostilas e a parte prática das disciplinas que envolvem conceitos de programação e a utilização desta para resolução de problemas.

A presente pesquisa é de caráter qualitativo, pois surgiu da necessidade de se investigar o porquê dos altos índices de evasão e retenção em disciplinas de programação e analisar, de forma descritiva, possíveis soluções para esses problemas.

Segundo Oliveira (2012, p. 37), a pesquisa qualitativa é

> Um processo de reflexão e análise da realidade através da utilização de métodos e técnicas para compreensão detalhada do objeto de estudo em seu contexto histórico e/ou segundo sua estruturação. Esse processo implica em estudos segundo a literatura pertinente ao tema, observações, aplicações de questionários, entrevistas e análise de dados, que deve ser apresentada de forma descritiva.

Inicialmente, realizou-se pesquisa bibliográfica sobre os *softwares* e ferramentas tecnológicas existentes que auxiliam os professores no ensino de algoritmos e programação. Dentre os *softwares* estudados, destacaram-se: *Feeper, WH-IDE, TutorICC, Portugol Studio, AAPW, TSTView, The Huxley, SOAP, ProgTest, Codecademy, Scratch e Alice.*

Leal (2014, p. 27) afirma em estudo sobre o Ensino de Programação que “a abordagem utilizada para estruturar a situação de aprendizagem influencia tanto os resultados esperados quanto as técnicas adotadas para alcançá-los”, portanto, é de grande importância este estudo e a seleção das abordagens e técnicas coerentes, para que futuramente seja possível gerar situação de aprendizagem que viabilize os resultados esperados.

Também foi feito levantamento, através da aplicação de um questionário (Apêndice A), sobre as principais dificuldades encontradas pelos alunos do polo de Santo Antônio dos Milagres, do curso Técnico em Informática para Internet à distância do IFPI, nas disciplinas de programação, e coletadas dicas e informações úteis que pudessem ser utilizadas para melhoria no processo de ensino e aprendizagem.

Após essa primeira etapa, iniciou-se o processo de desenvolvimento do protótipo da nova ferramenta tecnológica, levantando-se os requisitos com base no comparativo realizado entre os *softwares* existentes e nas informações repassadas pelos alunos. Durante e depois do processo de desenvolvimento, foram realizados testes buscando verificar a eficácia da ferramenta e possíveis erros e falhas no desenvolvimento.

As metodologias de análise, projeto e desenvolvimento de propósito geral não são direcionadas especificamente para produção de software educacional e Objetos de Aprendizagem, embora estejam sendo utilizadas para este propósito. Dentre os diversos métodos de propósitos gerais, é possível citar modelo em cascata, em espiral, por prototipação, orientados a objeto, ágeis, entre outros (BENITTI; SEARA; SCHLINDWEIN, 2005).

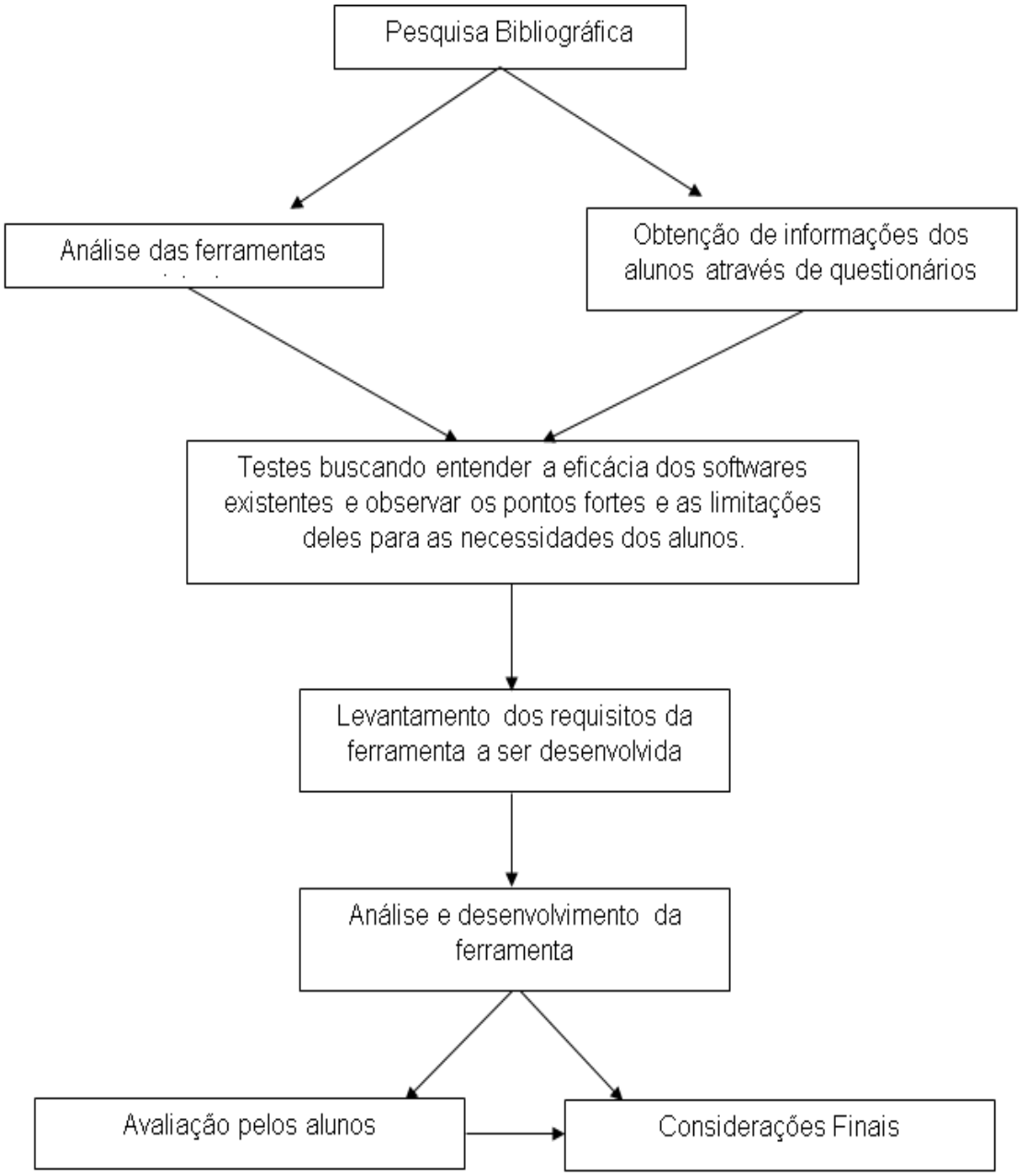
Os métodos denominados “ágeis” constituem proposta de contrapor às metodologias de desenvolvimento tradicionais usadas há décadas, pois visa criar um processo de desenvolvimento de *software* “leve”, não tão voltado a grande quantidade de documentação do projeto do *software* e adaptável às mudanças de requisitos ocorridas durante o desenvolvimento (LAPOLLI; GOMES, 2011).

Após o desenvolvimento do protótipo, a ferramenta foi submetida a avaliação por um grupo de alunos do polo de Uruçuí, que estavam em fase de conclusão do curso técnico de informática para Internet na modalidade a distância.

Com essa avaliação, foi possível mensurar o grau de aceitação dos alunos pela ferramenta proposta, além de comparar a metodologia anteriormente utilizada durante a disciplina de Lógica de Programação, com as novas possibilidades oferecidas pela Escola de Algoritmos.

O desenvolvimento do trabalho até o seu final seguiu o esquema apresentado na Figura 13.

Figura 13 - Fluxo de Trabalho



Fonte: Elaborado pelo autor, 2016.

A obtenção das informações dos alunos ocorreu de forma paralela a análise das ferramentas existentes, conforme pode-se observar na Figura 13.

Os alunos foram convidados a participar pelo autor deste trabalho, que atuou como Tutor no respectivo polo. Disponibilizou-se *link* no Ambiente Virtual de Aprendizagem - AVA do curso para o questionário, que foi confeccionado por meio da ferramenta de formulários *Google Docs* fornecida pela *Google*. Os dados obtidos foram submetidos a uma análise, que serviu de base para o desenvolvimento da ferramenta Escola de Algoritmos, conforme demonstra a seção 3.2 deste capítulo.

A avaliação também foi feita através da aplicação de um questionário (Apêndice B) do *Google Docs*, no qual os alunos, após terem utilizado a ferramenta, foram convidados a responderem questões acerca da interface, usabilidade, interatividade e aplicabilidade da ferramenta.

Os dados foram tabulados, e analisados e discutidos no capítulo 5 deste trabalho.

3.2 Coleta de Dados

Tradicionalmente, os cursos técnicos da área de informática possuem como componentes curriculares a introdução à lógica de programação, o desenvolvimento de algoritmos utilizando essa lógica e a aprendizagem de ambientes e linguagens para codificar os algoritmos desenvolvidos.

Em vários trabalhos sobre aprendizagem de programação, foram encontrados exemplos da dificuldade de conhecer e se habituar aos ambientes e às linguagens, tarefa considerada difícil e que necessita de muito tempo e esforço por parte dos alunos e, por vezes, dos professores ou tutores.

As disciplinas de Lógica de Programação e Algoritmos e Programação podem ser ensinadas de diversas formas, cuja metodologia tradicional ainda prevalece, na qual o professor transmite conhecimentos e soluções de diversos problemas e o aluno recebe essas informações sem discutir e/ou criar as próprias soluções.

No polo de Educação a Distância da cidade de Santo Antônio dos Milagres – PI, onde o autor deste trabalho atuou como Tutor a Distância, os resultados não foram diferentes, conforme pode ser visualizado na Tabela 1.

Tabela 1 – Resultados da disciplina Lógica de Programação

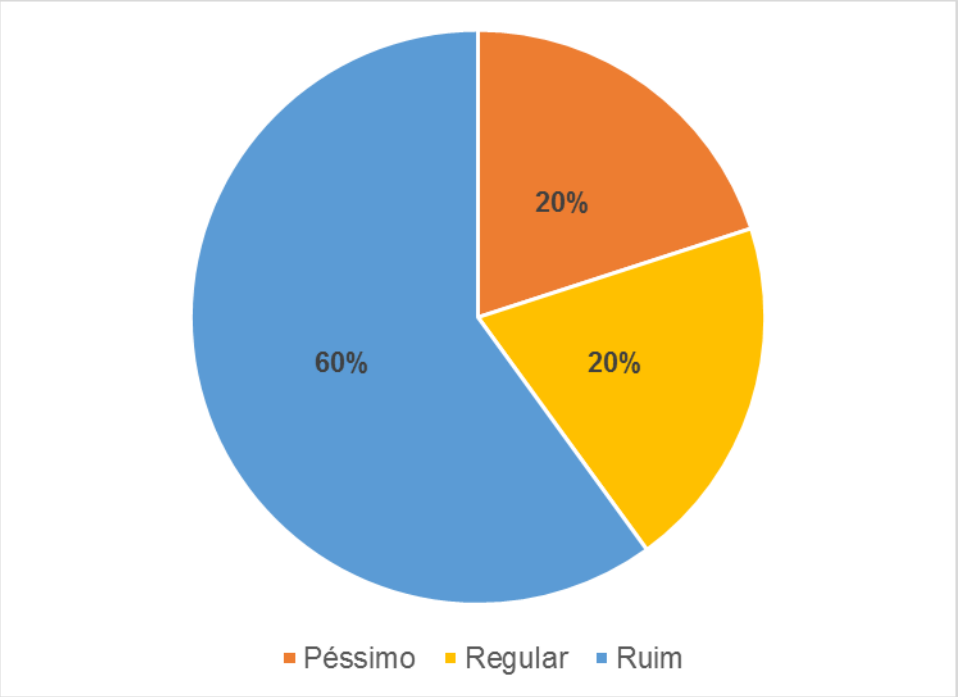
Item	Quantidade
Número de alunos que iniciaram o curso	51
Número de alunos aprovados	23
Número de alunos reprovados	7
Número de alunos evadidos	21

Fonte: Moodle IFPI.[11]

A fim de se obter informações acerca do aprendizado nessas disciplinas no curso de Informática para Internet a distância do IFPI, foi preparado questionário através do *Google Docs* e aplicado para 15 alunos do polo de Santo Antônio dos Milagres - PI, para confirmar o interesse dos mesmos por métodos inovadores de ensino e a utilização de ferramentas que potencializem o ensino.

Os alunos foram indagados sobre: como você considera seu aprendizado nas disciplinas de programação? Os mesmos responderam, conforme exposto no Gráfico 1.

Gráfico 1 – Aprendizado em programação



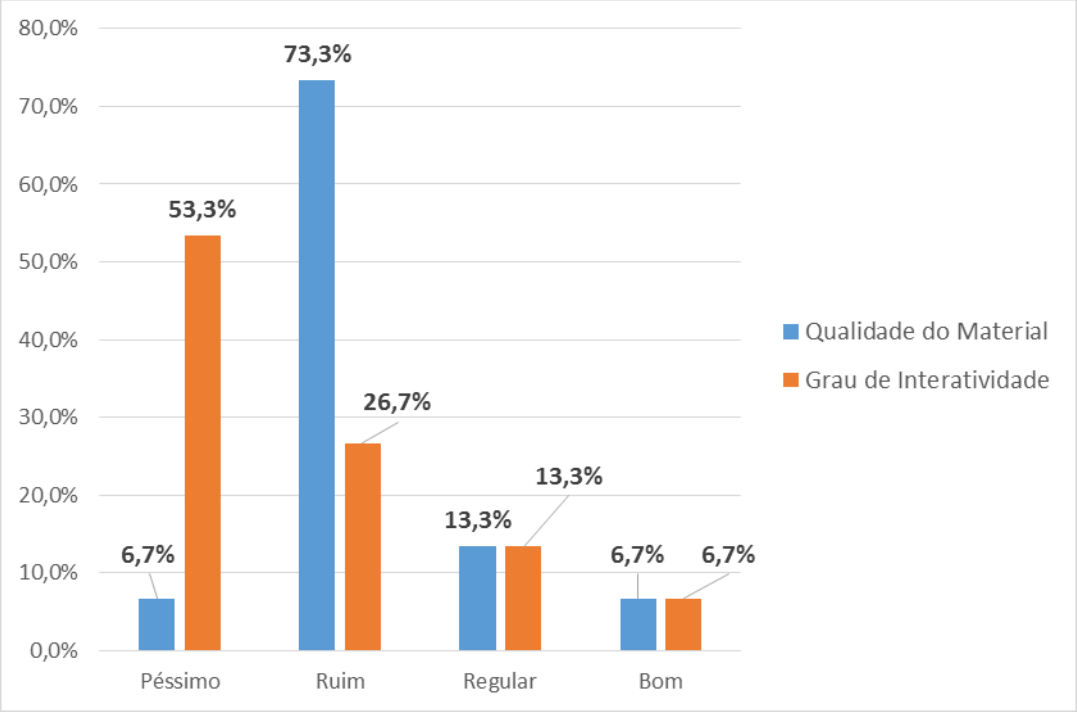
Fonte: Elaborado pelo Autor, 2016.

[11] http://etec.ifpi.edu.br Acesso em: 12/03/2015.

Como se pode observar no gráfico, a grande maioria dos alunos não conseguia obter aprendizagem plena nas disciplinas ligadas à programação, isso se deve a vários fatores como os citados e outros que vão da peculiaridade de cada aluno. Segundo Behar (2013), muitos alunos ainda estão em fase de alfabetização e letramento digital, o que prejudica a relação com os recursos digitais atuais. Portanto, faz-se necessária a promoção de novas tecnologias caracterizadas pela simplicidade, objetividade e precisão.

No Gráfico 2, questiona-se acerca do material de apoio utilizado e do grau de interatividade do mesmo.

Gráfico 2 – Qualidade e grau de interatividade do material



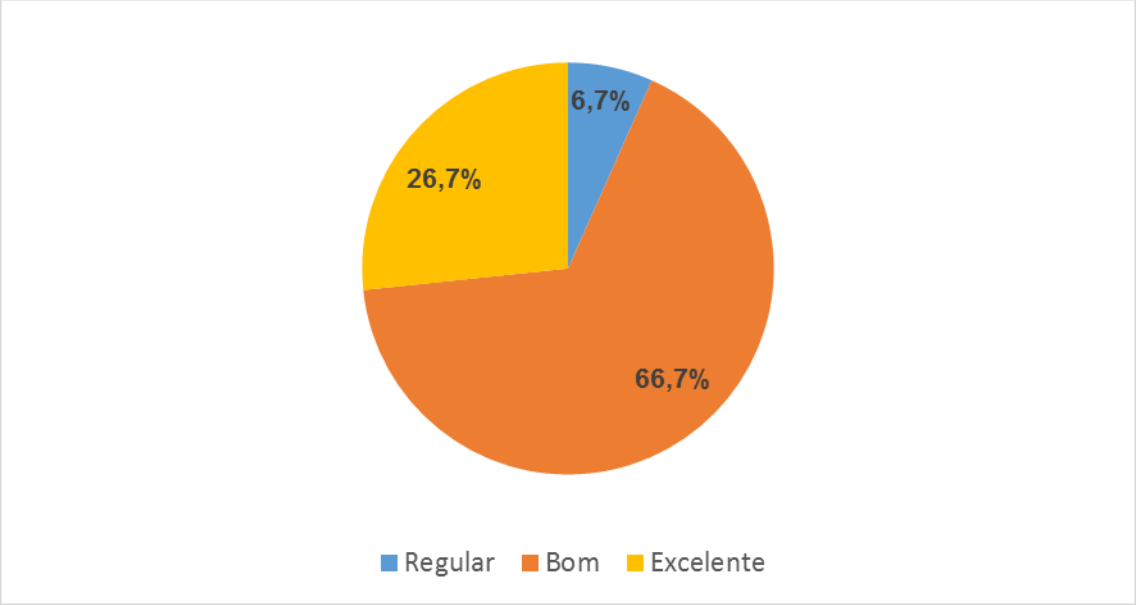
Fonte: Elaborado pelo Autor, 2016.

No segundo gráfico, observa-se que muitos alunos, além de não gostar do material utilizado nas disciplinas, não consideravam que este possibilitava a interatividade, muitas vezes, pelo fato da linguagem utilizada nos livros e apostilas ser uma linguagem de difícil entendimento pelos alunos. E, também, pelo fato desses materiais não sofrerem atualizações constantes, o que provoca a explicação de uma tecnologia que não condiz mais com o cenário atual.

De acordo com Lopes (2013), é necessária a utilização de um bom material didático, direcionado àquele que frequentará os cursos de EAD, bastante especializado e organizado dentro das necessidades particulares de cada assunto e de cada aluno, haja vista a necessidade de seus aprendizes em relação a essa situação específica.

No que se refere ao uso de ferramentas tecnológicas, foi observado que 86,7% dos alunos afirmaram que não utilizaram ferramenta que facilitasse o aprendizado, enquanto que 13,3% dos alunos afirmaram que utilizaram desse recurso. No Gráfico 3, foi perguntado aos alunos: como você considera a ideia da utilização de uma ferramenta tecnológica que favorecesse o aprendizado?

Gráfico 3 – Utilização de uma ferramenta tecnológica



Fonte: Elaborado pelo Autor, 2016.

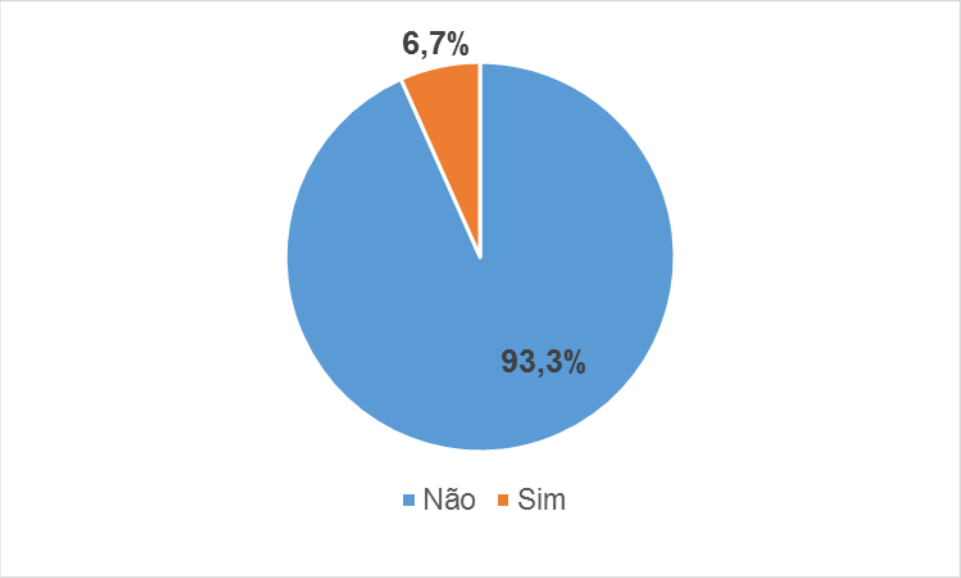
A maioria dos alunos considera importante a utilização de alguma ferramenta tecnológica que estimule o aprendizado e o torne mais interessante. Aprender é uma tarefa crucial no desenvolvimento pessoal e social, mas, em muitos casos, é percebida como algo difícil e aborrecido, repelindo a mente. No entanto, essa percepção pode ser alterada para algo divertido de se fazer, modificando a tarefa de forma a que esta se transforme em uma experiência lúdica, sem perder o aspecto sério da aprendizagem (GONÇALVES, 2013).

No que compete à forma de utilização dessas ferramentas de apoio ao ensino, 6,7% dos alunos responderam que o melhor seria utilizando o

celular/smartphone, 20% disseram preferir utilizar no computador, separadamente do AVA e 73,3% afirmaram que o ideal seria a utilização no computador, incorporado ao AVA.

Os alunos também foram questionados se tinham ou não capacidade de codificar um programa/*software*, conforme explicitado no Gráfico 4.

Gráfico 4 – Capacidade de codificar um programa



Fonte: Elaborado pelo Autor, 2016.

Mesmo eles já tendo cursado duas disciplinas da área, notou-se que a grande maioria deles não se julgava capaz de programar, ou seja, de desenvolver um *software*, o que deixa ainda mais notável a lacuna que há entre a forma como os assuntos são abordados e a verdadeira prática de programação por parte dos alunos.

A última pergunta do questionário trata-se de uma questão subjetiva, na qual os alunos poderiam citar sugestões de melhoria nas disciplinas de programação. As respostas foram variadas, com muitos problemas abordados, um dos alunos respondeu que "*Eu acho que o que poderia mudar em programação é unir teoria e prática*", outros mencionaram também problemas com relação à infraestrutura e falta de formação dos tutores.

Em muitos municípios brasileiros, ainda não se tem infraestrutura adequada para fornecer suporte a um curso a distância, problemas como computadores antigos, internet lenta ou indisponível acarretam a perda de interesse e desmotivação dos alunos. Outro fator importante é a falta de

profissionais qualificados para oferecer suporte “in loco” aos alunos e auxiliá-los na utilização dos computadores e programas.

> ..., é possível que existam outros fatores internos e externos aos alunos (e.g., instalações e o suporte oferecido pela instituição de ensino, práticas e metodologias utilizadas pelos professores, vida social do aluno, práticas e métodos de estudo, etc.) que ainda não estão muito claros, e que podem agravar ainda mais esse quadro de reprovação e evasão. Tais fatores podem variar de aluno para aluno, e a intensidade com a qual esses fatores podem influenciá-los no processo de ensino-aprendizagem ainda é desconhecida. (HENRIQUES, 2013, p. 14).

Com base nas respostas fornecidas pelos alunos, percebem-se os vários fatores que contribuem para a má aprendizagem das disciplinas de programação e, consequentemente, para o aumento dos índices de evasão e retenção. Através desses resultados, enfatiza-se o fato de que mais trabalhos devem ser desenvolvidos, no intuito de promover a criação de uma metodologia (seja ferramenta ou forma de ensino), a fim de amenizar os problemas relativos ao rendimento nas disciplinas mencionadas.

4. PROTÓTIPO DA FERRAMENTA

Este capítulo descreve o desenvolvimento da ferramenta, as funcionalidades e características.

4.1 Escola de Algoritmos

A Escola de Algoritmos é uma ferramenta digital de aprendizagem que propõem atividades diferenciadas para cada aluno, sob medida, de acordo com respostas e ações frente às inúmeras atividades que realiza, como leituras de textos e acesso a vídeos.

Dessa forma, não é mais o aluno que se adapta à ferramenta educacional como nas ferramentas tradicionais, mais se inverte o processo, ou seja, é o ambiente que se adapta ao desencadeamento da compreensão e da construção do conhecimento por parte de cada estudante.

A ferramenta tem como característica a utilização da pseudolinguagem Português Estruturado ou *Portugol* (simbiose do português com o ALGOL e o PASCAL), que possibilita ao aluno focar mais no problema a ser resolvido do que na máquina que executará o algoritmo.

Por meio do uso do *Portugol*, os alunos tendem a fixar mais as técnicas de programação, não se atendo a uma linguagem de programação e detalhes da sua sintaxe, além de favorecer a flexibilidade e a facilidade na interpretação dos algoritmos, devido à proximidade com a linguagem humana.

> Sabe-se que a programação orientada a objetos representa o estado da arte em programação, mas a iniciação à lógica de programação ainda é mais eficiente com a programação estruturada, pois, fica mais fácil ao aluno compreender e explicitar as ações que compõem um algoritmo e que, ao serem executadas, levam à solução do problema (FALCKEMBACH; ARAUÚJO, 2013, p. 2).

O objetivo principal da ferramenta é minimizar as dificuldades encontradas pelos alunos no aprendizado das técnicas básicas de programação, fazendo uso de linguagem natural e ambiente visual atrativo.

Ferramentas hipermídia possibilitam a visualização, o entendimento e a memorização de conceitos complexos de forma fácil, promovendo o aprendizado e diminuindo o tempo gasto pelos professores e alunos (FERREIRA, 2005).

A utilização de ambientes digitais e interativos de aprendizagem na educação a distância possibilita diminuir as distâncias espaço-temporais, permitindo ao aluno múltiplas opções e trajetórias, não focando apenas na disseminação de conteúdos e tarefas definidas antecipadamente (ALMEIDA, 2011).

A Escola de Algoritmos está acessível através da Internet, com uso de um navegador (*browser*). Optou-se pela plataforma web, pelo fato de ser de fácil distribuição e multiplataforma, uma vez que não é preciso instalar complemento e pode-se acessar a partir de qualquer computador com um navegador web e conectado à Internet.

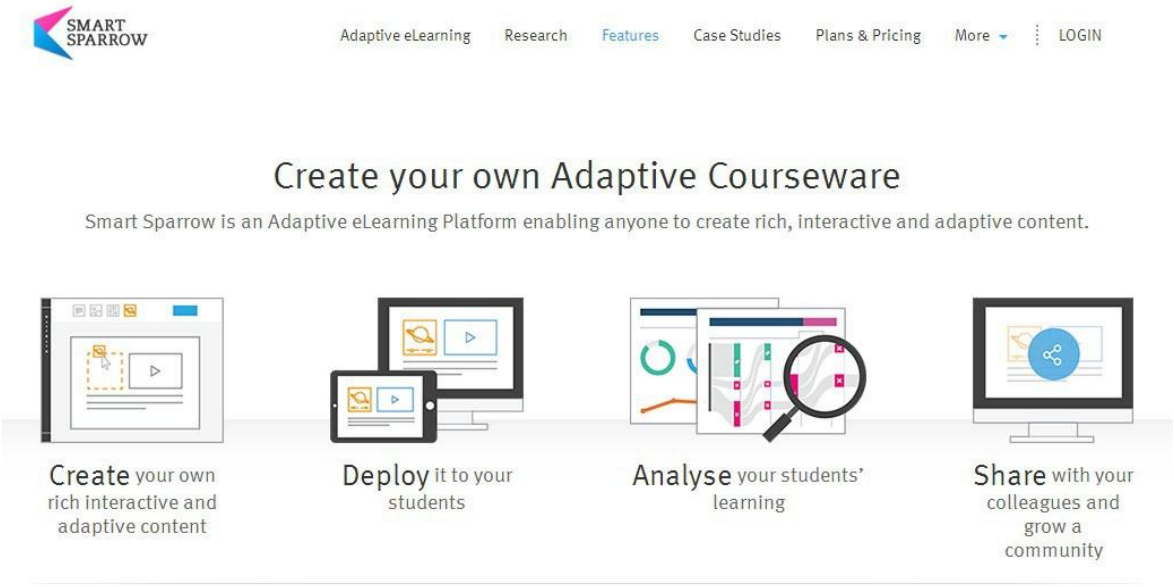
A ferramenta foi construída utilizando a solução *Adaptive eLearning Platform – AeLP*, plataforma para criação de atividades *e-Learning* ricas, interativas, altamente visuais e adaptativas.

O AeLP foi desenvolvido pela *Smart Sparrow*, uma *startup* incubada no Grupo de Pesquisas de Ensino Adaptativo da *University of New South Wales*, na Austrália. Ela foi criada em 2010, e é destinada para alunos do ensino médio e superior, sendo a primeira a permitir que qualquer pessoa crie seu curso interativo e adaptativo.

Ferramentas criadas pelo AeLP são adaptativamente organizadas para atender às necessidades individuais de aprendizagem dos alunos, fornecer *feedback* adaptativo para estudantes com dificuldades, bem como acompanhamento dentro do ambiente virtual (POLLY et al., 2014).

A Figura 14 apresenta a página inicial da *startup*.

Figura 14: Tela inicial da *Smart Sparrow*



Fonte: *Smart Sparrow*, 2016.[12]

A Escola de Algoritmos foi concebida para ser uma forma flexível de integrar informações introdutórias, exercícios com questões associadas e várias mídias de comunicação embutidas, como vídeos curtos, para melhorar aprendizagem dos alunos.

Na Figura 15, pode-se visualizar a página inicial da ferramenta. Cabe ressaltar que no protótipo desenvolvido para fins deste trabalho, foi implementado apenas o Módulo 1, Conceitos de Programação.

Os conteúdos estão divididos em quatro módulos que vão desde conceitos iniciais, como o de algoritmo e variáveis, a assuntos mais complexos, como estrutura de dados. Ao clicar no botão de informação, é exibida uma breve descrição de cada módulo, conforme Figura 16.

[12] http://www.smartsparrow.com Acesso em: 14/07/2016.

Figura 15: Tela inicial da Escola de Algoritmos

Fonte: Elaborado pelo Autor, 2016.

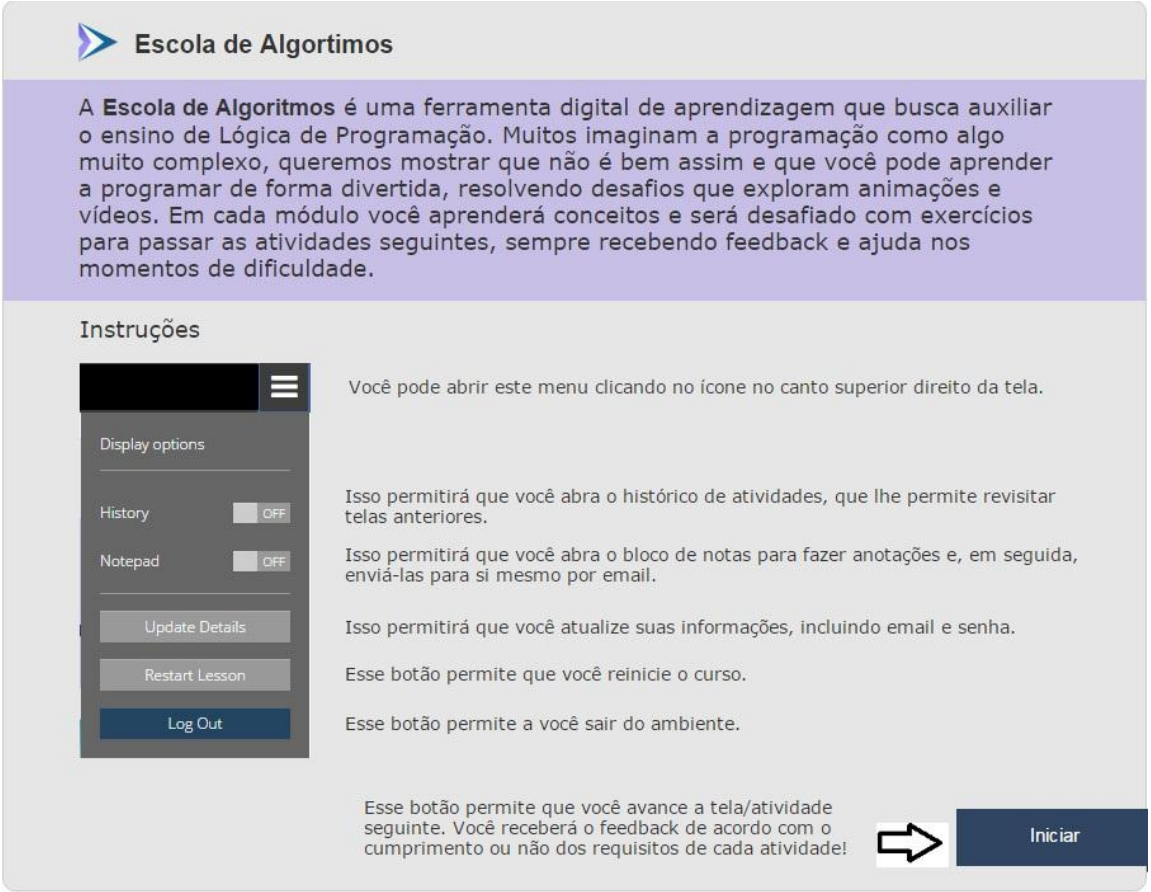
Figura 16: Tela Inicial com descrição

Fonte: Elaborado pelo Autor, 2016.

Ao clicarem no botão Avançar, os alunos tem acesso à descrição de funcionamento da ferramenta e às instruções de como navegar entre as páginas e obter o máximo proveito dos recursos disponíveis.

Na Figura 17, visualiza-se a página com as instruções iniciais.

Figura 17: Tela com Instruções



Fonte: Elaborado pelo Autor, 2016.

O usuário pode a qualquer momento retornar a uma tela anterior e rever conceitos ou refazer atividades através da opção "*History*". O recurso "*Notepad*" permite fazer anotações e enviá-las para o e-mail cadastrado, para consultas futuras. É possível também sair a qualquer momento da ferramenta (fechar a janela) e ao retornar, o aluno será direcionado para a tela em que se encontrava, assim como voltar para tela inicial através da opção "*Restart Lesson*".

A ferramenta se utiliza de estratégias adaptativas, que permitem indicar ao aluno quais telas e atividades atendem suas necessidades e sistematiza uma trilha de aprendizagem individualizada, visando desenvolvimento da autonomia.

O conteúdo da disciplina pode ser percorrido pelo aluno de forma variada, dependendo da capacidade de resolver os exercícios propostos. Desta forma, o ritmo de aprendizagem se adapta ao perfil do aluno, podendo ser mais

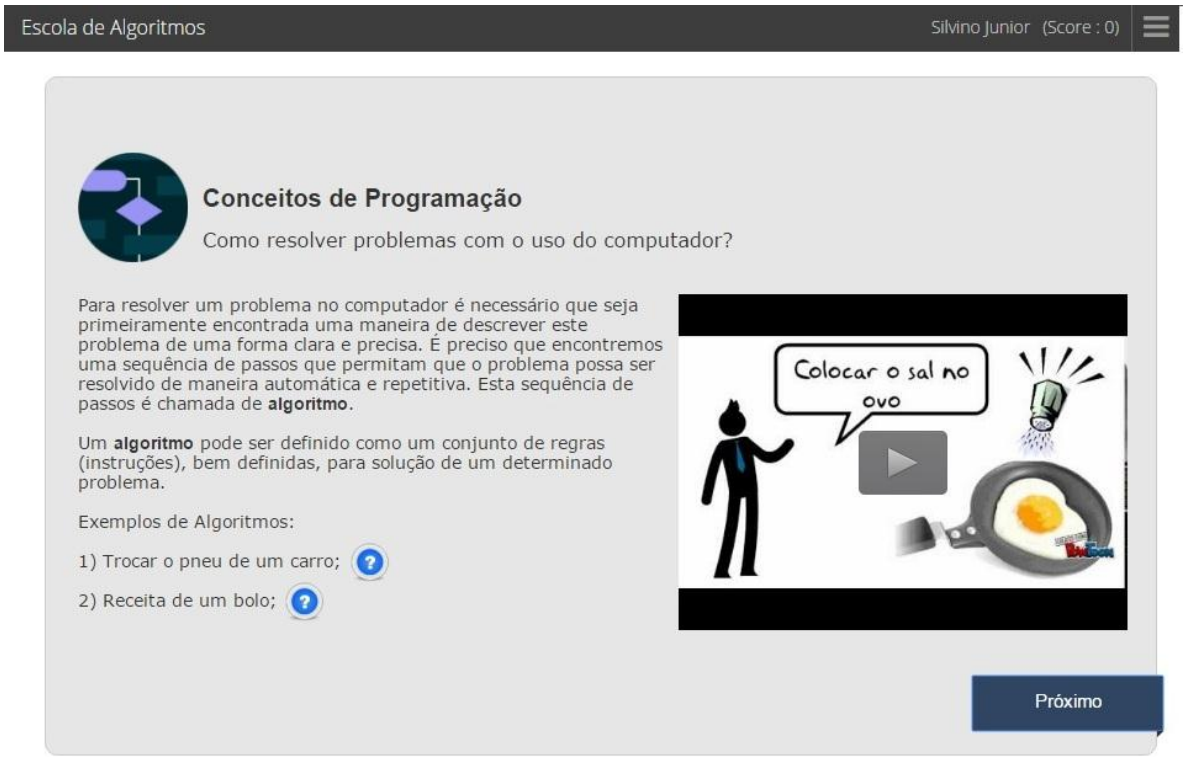
rápido ou mais lento, dependendo da maior ou menor facilidade em aprender os conceitos de programação.

Existem duas características utilizadas para prover a adaptação: a) o número de tentativas e erros e b) o tempo dispensado a resolver uma atividade. Essas características são utilizadas para adaptar a navegação e a apresentação do ambiente.

> Um sistema é dito adaptativo (adaptatividade) se é capaz de modificar suas características automaticamente de acordo com as necessidades do usuário. Modificações na apresentação da interface ou no comportamento do sistema dependem da maneira que o usuário interage com o mesmo. Desta forma, o sistema é quem inicia e executa as modificações apropriadas para o usuário (LIMA, 2007, p.17).

A Figura 18 mostra uma tela onde o aluno visualiza o texto relativo a um conteúdo e pode reproduzir um vídeo animado que contém explicações mais detalhadas acerca do assunto.

Figura 18: Tela de Conteúdo



Fonte: Elaborado pelo Autor, 2016.

O aluno somente pode avançar para a tela seguinte após cumprir os requisitos da tela atual. No caso da Figura 19, as condições seriam a visualização dos exemplos e a reprodução do vídeo. Se tentar avançar sem ter

feito o que foi solicitado é exibida uma mensagem de alerta pelo sistema, como mostra a Figura 19.

Figura 19: Tela com *Feedback* de Erro

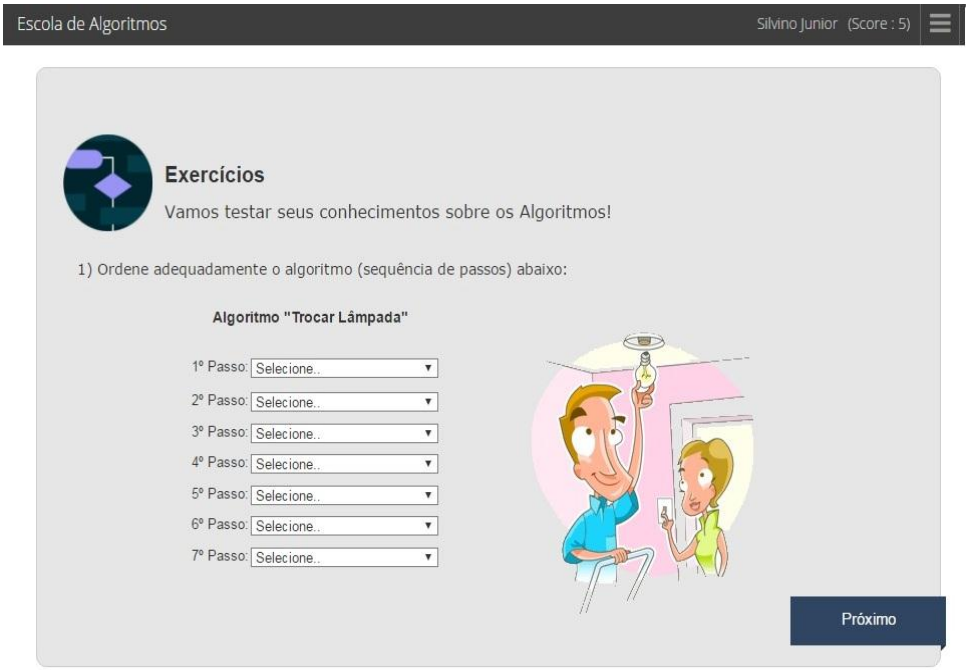
Fonte: Elaborado pelo Autor, 2016.

Além de conteúdos textuais e vídeos, a ferramenta também possui conteúdos dinâmicos que possibilitam ao aluno interagir com o ambiente, como é o caso da tela da Figura 20, que permite ao aluno testar os conhecimentos adquiridos em telas anteriores.

Antes de programar um computador, é necessário conhecer como desenvolver uma especificação formal que nada mais é do que um conjunto de instruções a serem seguidas, as quais possam indicar alguma solução de um problema (JÚNIOR et al., 2006).

Atividades como a da Figura 20, em que o usuário deve selecionar adequadamente os itens para colocar em uma sequência lógica, permitem ao aluno desenvolver novos esquemas mentais de forma gradativa que o habilita a chegar à solução de outros problemas a partir de situações equivalentes.

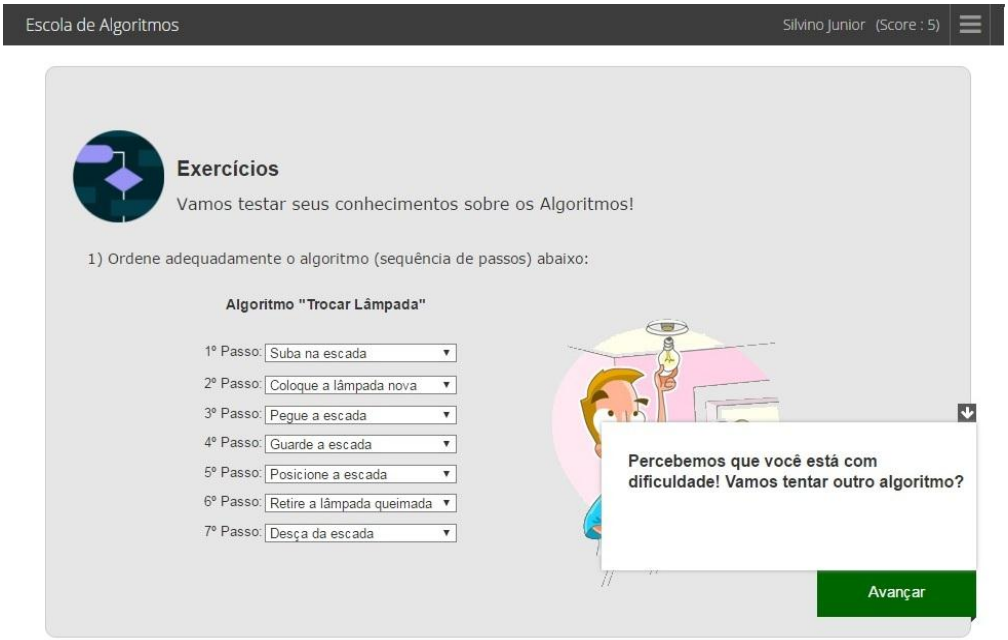
Figura 20: Tela de Exercícios



Fonte: Elaborado pelo Autor, 2016.

Caso o aluno tenha dificuldade em entender a lógica utilizada para resolver o problema proposto e supere o número de três tentativas com erro, o sistema identifica, emite um *feedback* automático e o redireciona a páginas com explicações adicionais e revisão do conteúdo visto, como nas Figuras 21 e 22.

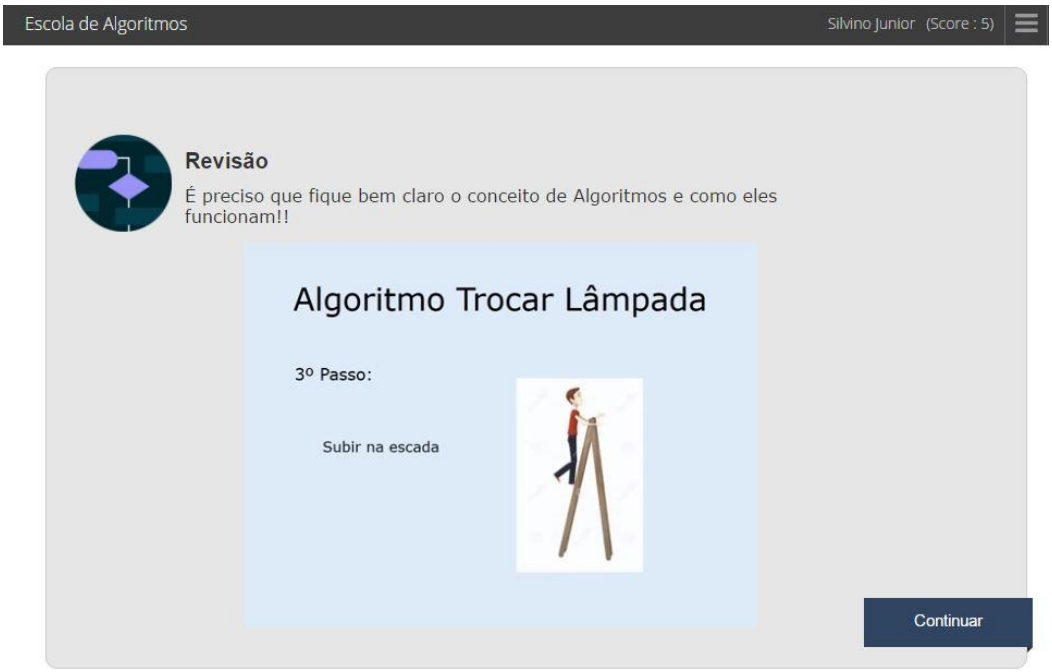
Figura 21: Tela com *Feedback*



Fonte: Elaborado pelo Autor, 2016.

Após a revisão (Figura 22) que é feita por meio de uma animação, o aluno é novamente testado, com outro exercício semelhante.

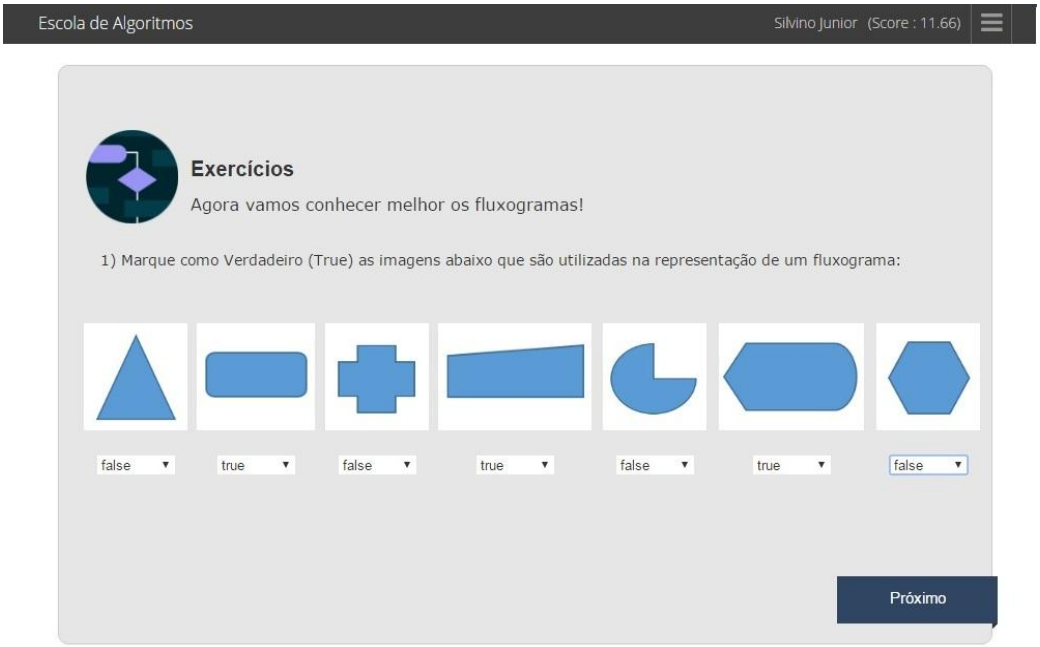
Figura 22: Tela de Revisão



Fonte: Elaborado pelo Autor, 2016.

Há também exercícios para estimular a representação gráfica dos algoritmos, utilizando fluxogramas, como se visualiza na Figura 23.

Figura 23: Tela com Exercícios sobre Fluxogramas



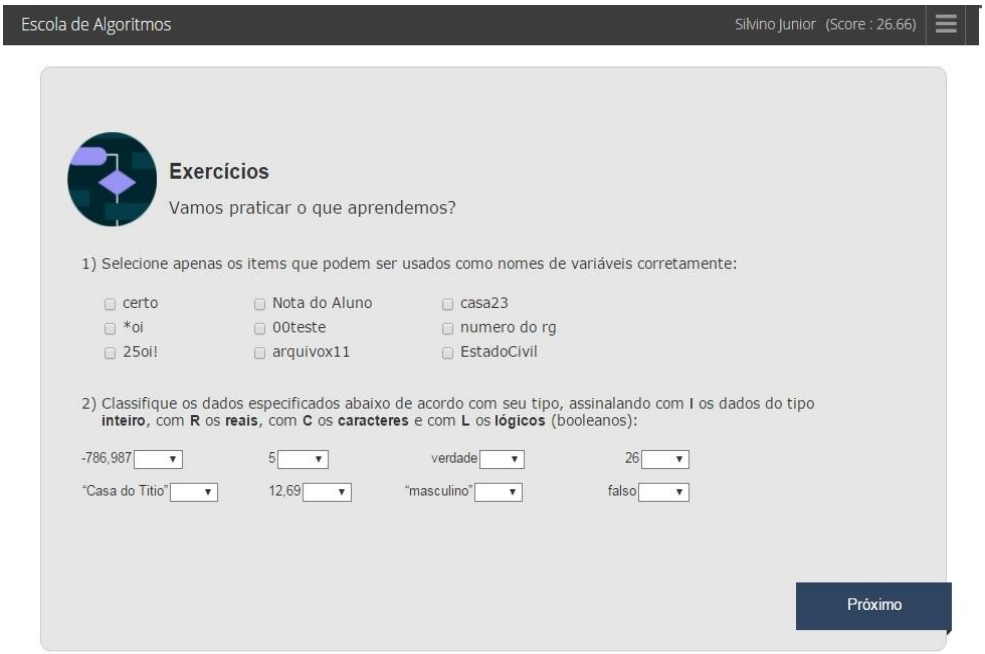
Fonte: Elaborado pelo Autor, 2016.

Na Figura 24, apresentam-se os conceitos de variáveis e tipos de dados. E na Figura 25, estão disponíveis exercícios de múltipla escolha, com os quais os alunos poderão testar os conhecimentos adquiridos e fixar conteúdos como regras para criação de variáveis e os tipos de dados primitivos.

Figura 24: Tela sobre Tipos de Dados e Variáveis

Fonte: Elaborado pelo Autor, 2016.

Figura 25: Tela com Exercícios de Múltipla Escolha



Fonte: Elaborado pelo Autor, 2016.

A ferramenta apresenta também trechos de códigos em linguagem natural, como mostrado na Figura 26, que possibilitam aos alunos contato com os algoritmos e animações que demonstrem a execução.

Figura 26: Tela com Código

Fonte: Elaborado pelo Autor, 2016.

As animações têm como finalidade aumentar a interação computador-aluno, incentivando, despertando a curiosidade e a atenção dos alunos. Outras características da ferramenta são a autoexplicação e didática, oferecendo nova experiência aos alunos.

Na Figura 27, tem-se uma animação que demonstra a estrutura de um algoritmo, explicando cada parte que o forma.

Outro ponto a ser explorado pela ferramenta é a demonstração da execução de testes de mesa, em que os estudantes podem acompanhar o valor das variáveis durante a execução do algoritmo, como demonstra a Figura 28.

A visualização de valores de variáveis faz com que o aluno possa inspecionar o valor que realmente está sendo utilizado pelo programa, independentemente do nível de compreensão do aluno sobre o funcionamento do código (NOSCHANG et al., 2014).

Figura 27: Tela com Animação

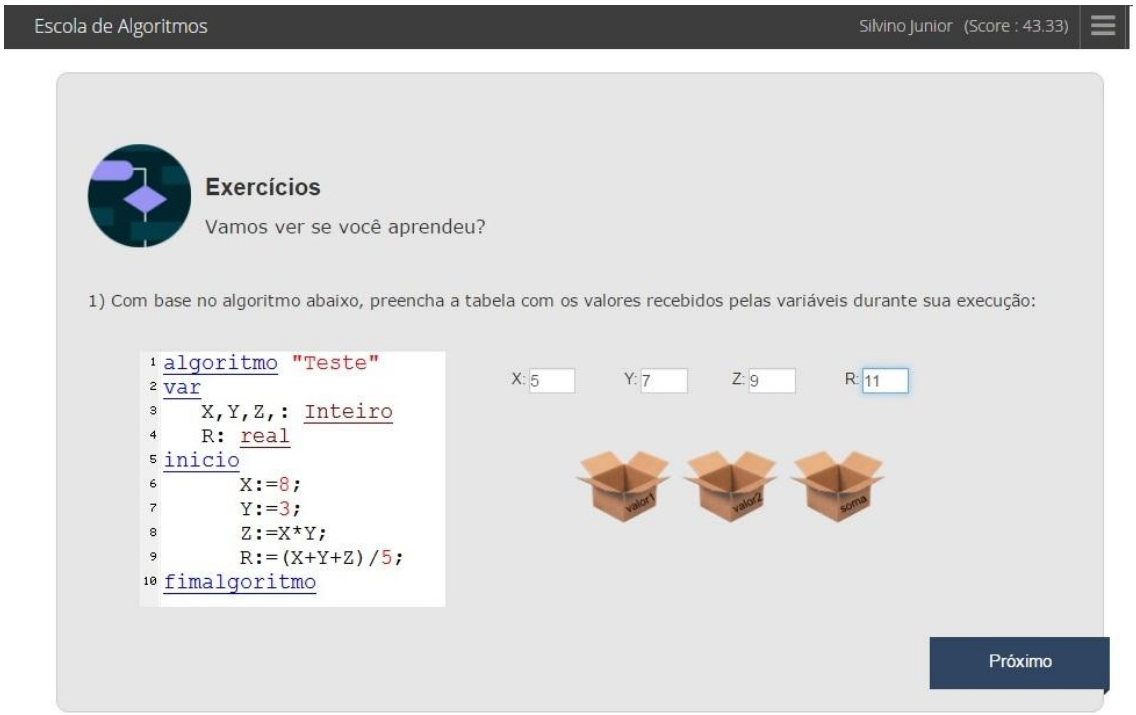
Fonte: Elaborado pelo Autor, 2016.

Figura 28: Tela com Teste de Mesa

Fonte: Elaborado pelo Autor, 2016.

Assim, entende-se que esse recurso acarreta benefícios para a aprendizagem e evita erros que poderiam ser cometidos por alunos iniciantes, ao anotar equivocadamente os valores gerados durante a execução.

Figura 29: Tela com Exercício



Fonte: Elaborado pelo Autor, 2016.

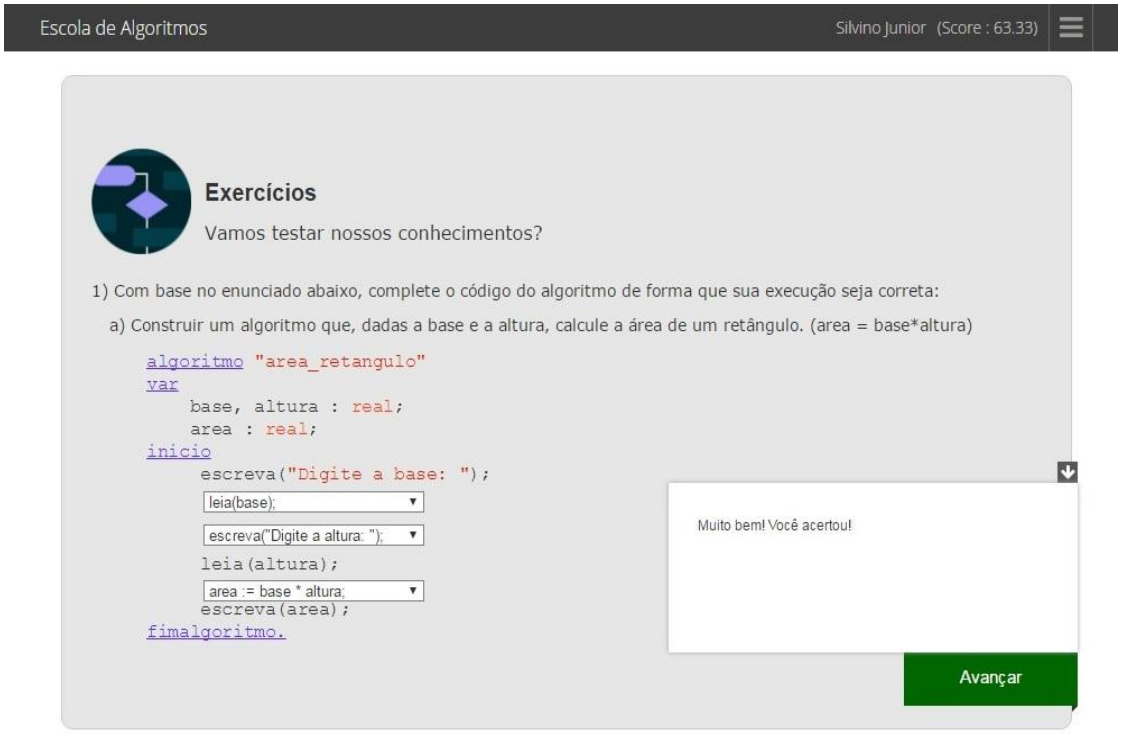
Após visualizarem como as variáveis têm seus valores modificados durante a execução de um algoritmo, os alunos são desafiados a responderem a um exercício (Figura 29) que possibilita analisar se houve realmente aprendizagem.

Os comandos de entrada e saída de dados também são frisados, com a explicação do processo dentro de um algoritmo (Figura 30) e exercícios (Figura 31), que permite ao aluno aprender em quais momentos do código ocorrem essas etapas.

Figura 30: Tela sobre Comandos de Leitura e Exibição

Fonte: Elaborado pelo Autor, 2016.

Figura 31: Exercício sobre Comandos de Leitura e Exibição



Fonte: Elaborado pelo Autor, 2016.

Ao chegar ao fim do módulo, o aluno recebe um aviso de que concluiu as atividades e também um e-mail avisando da conclusão do mesmo, como se observa na Figura 32.

Figura 32: Tela de Finalização do Módulo

Fonte: Elaborado pelo Autor, 2016.

Figura 33: E-mail enviado ao aluno

Fonte: Elaborado pelo Autor, 2016.

Apesar das inúmeras ferramentas existentes voltadas ao ensino de algoritmos, a ferramenta proposta neste trabalho apresenta diferencial frente as já existentes. Prioriza a utilização do *Portugol*, utiliza-se de recursos visuais, como vídeos e animações, que possibilitam aos alunos a resolução de problemas diversos, além da verificação e correção de blocos de códigos.

Segundo Anido (2015), a primeira e mais importante barreira para estudantes brasileiros usarem as ferramentas existentes e que elas foram desenvolvidas para um público que tem inglês como língua oficial e, apesar dos meios de internacionalização, a maioria dos ambientes exige algum conhecimento da língua inglesa, e não apenas referente às palavras-chave da linguagem de programação.

A ferramenta proposta não tem como objetivo ser utilizada para codificação dos algoritmos, ou seja, para o teste destes, mas para explicar a lógica computacional, recorrendo a trechos de código em pseudocódigo e permitindo que os alunos conheçam os passos necessários a dar antes de passar para a programação propriamente dita, utilizando uma ou mais linguagens de programação.

Assim, professores e alunos têm uma ferramenta fácil de usar, onde se encontram os conteúdos necessários da disciplina Lógica de Programação, em um ambiente lúdico e interativo.

Os alunos, ao estudarem todos os módulos da ferramenta, já devem ser capazes de desenvolver algoritmos simples com certo grau de formalização que não representam procedimentos difíceis de serem traduzidos para uma linguagem de programação.

5. ANÁLISE E DISCUSSÃO

Este capítulo descreve os resultados da avaliação da ferramenta Escola de Algoritmos por um grupo de alunos da modalidade de ensino a distância.

5.1 Avaliação da Ferramenta Escola de Algoritmos

Nesta subseção, estão apresentados os resultados da avaliação da ferramenta que foi utilizada por um grupo de 29 alunos do curso Técnico em Informática para Internet na modalidade a distância, do polo do IFPI, localizado na cidade de Uruçuí-PI.

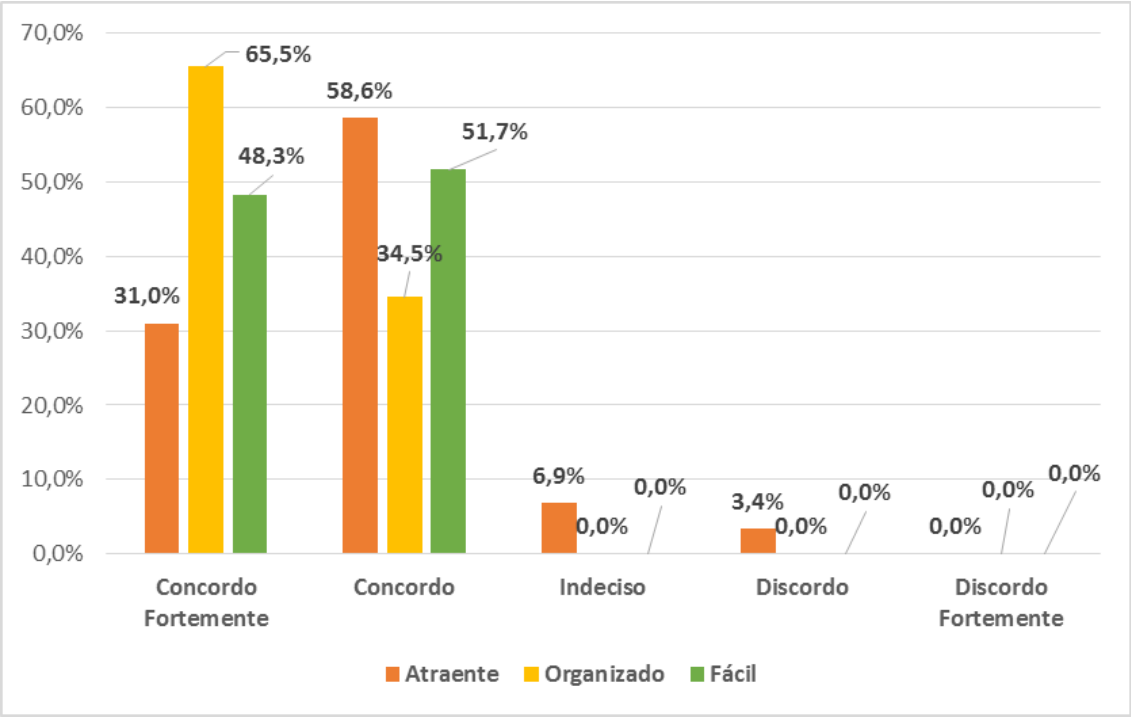
Os alunos que participaram estavam em fase de finalização do curso, sendo que todos cursaram e obtiveram aprovação na disciplina de Lógica de Programação. Possuíam idade entre 16 e 22 anos e 55,2% eram do sexo masculino e 44,8% do sexo feminino.

O grupo de alunos foi conduzido ao laboratório de informática, onde utilizaram a ferramenta durante aproximadamente 90 minutos. Após a conclusão do módulo por todos os alunos, foram convidados a responderem a um questionário (Apêndice B), contendo 16 questões sobre a ferramenta e informações sobre problemas e melhorias que pudessem ser feitas.

No questionário, que foi criado usando a ferramenta *Google Docs* e disponibilizado através de um *link* para o acesso *on–line,* utilizando o navegador *web*, foi utilizado a escala *Likert* (LIKERT, 1932) que é composta por um conjugado de frases (itens) em relação a cada uma das questões, para o aluno manifestar o grau de concordância desde o discordo totalmente (nível 1) até ao concordo totalmente (nível 5).

Inicialmente, os alunos foram questionados acerca da interface da ferramenta, se o ambiente era atraente, organizado e de fácil manuseio, os mesmos responderam, conforme exposto no Gráfico 5.

Gráfico 5 – Questões relacionadas à Interface

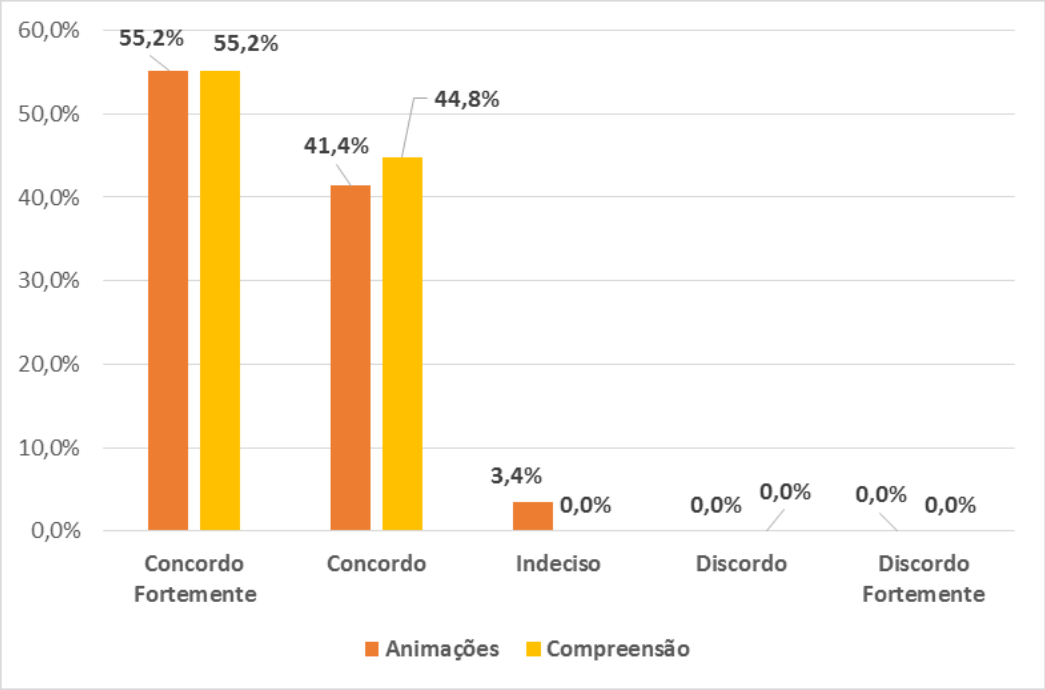


Fonte: Elaborado pelo Autor, 2016.

Pode-se observar no Gráfico 5, que a grande maioria dos alunos concordava que a ferramenta possuía interface atrativa, os conteúdos estavam bem organizados e favorecia fácil manuseio. Segundo Santos e Costa (2006), os alunos se interessam mais por uma aula dinâmica e diferenciada, e, o uso de um ambiente lúdico com movimentos em tempo real, cores e sons enriquecem ainda mais o poder de comunicação.

No Gráfico 6, questionou-se sobre os meios utilizados para apresentar as informações (textos, imagens e vídeos) aumentam a compreensão do conteúdo e se o uso de animações para explicação de procedimentos e códigos incentiva a utilização da ferramenta.

Gráfico 6 – Uso de texto, imagens, animações e vídeos



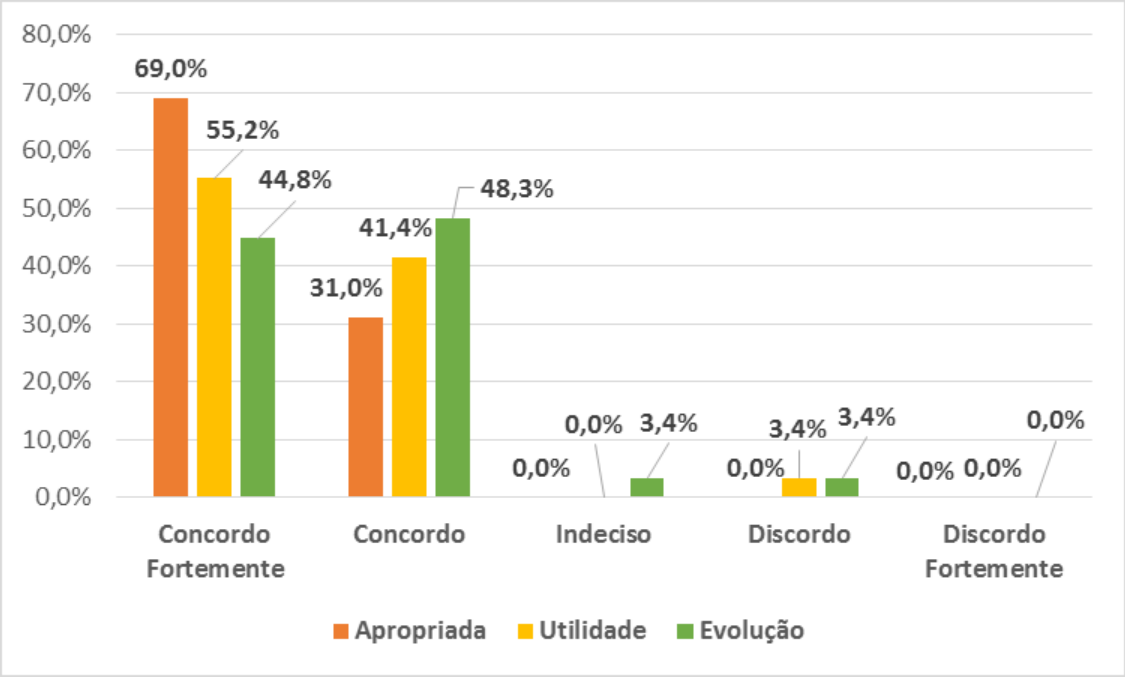
Fonte: Elaborado pelo Autor, 2016.

Observou-se que 55,2% dos alunos concordavam fortemente, 41,4% concordam e apenas 01 (um) ficou indeciso quanto ao uso de animações. Todos concordaram que os meios usados para exibição do conteúdo aumentam a compreensão do aluno.

A animação torna-se facilitador do processo de aprendizagem, visto que a apresentação de conceitos abstratos é mais didática, melhorando a qualidade do material de apoio à aula (TAVARES et al., 2016).

Os alunos também foram questionados se a ferramenta era apropriada para o auxílio no ensino de Lógica de Programação, se a ferramenta era útil para construção das habilidades necessárias na disciplina e se os conteúdos e exercícios evoluíam em ritmo adequado. Os resultados podem ser visualizados no Gráfico 7.

Gráfico 7 – Ferramenta apropriada, utilidade e evolução



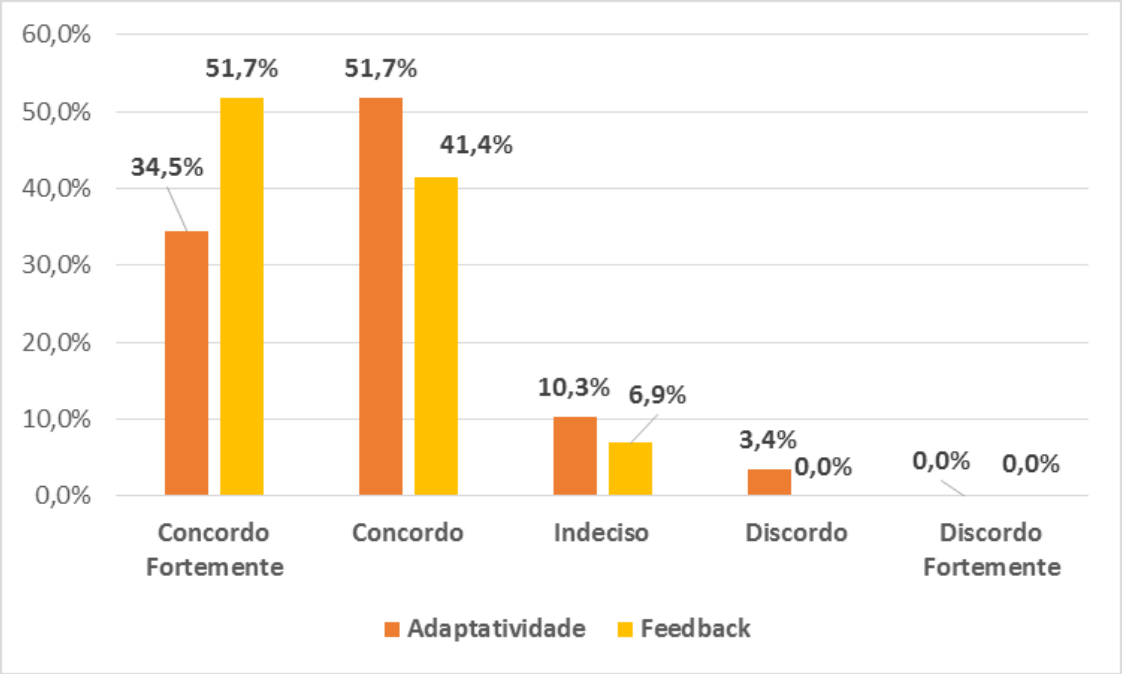
Fonte: Elaborado pelo Autor, 2016

Todos concordaram que a ferramenta utilizada era apropriada para auxiliar no ensino de Lógica de Programação. Do total de alunos entrevistados, 01 (um) discordou acerca da utilidade da ferramenta para construção das habilidades requisitadas pela disciplina, 01 (um) ficou indecisos e 01 (um) discordou que os conteúdos e atividades evoluíam em ritmo adequado ao aprendizado.

A utilização das TIC na EAD, somente apresentando informações, problemas e objetos de conhecimento para o aluno, pode não ser suficiente para envolvê-lo e despertar-lhe tal motivação pela aprendizagem. Além disso, é necessária organização do tempo para estudos e a participação efetiva independente de local ou horário (ALMEIDA, 2011).

No que se refere ao uso de características de adaptatividade para adequação ao estilo de aprendizagem do aluno e ao *feedback* automático emitido pela ferramenta como facilitador do aprendizado, os alunos responderam, conforme o Gráfico 8.

Gráfico 8 – Adaptatividade e *Feedback* Automático



Fonte: Elaborado pelo Autor, 2016.

A maioria dos alunos concordou com a utilização de estratégias adaptativas e a adequação do conteúdo aos diferentes estilos de aprendizagem, 10,3% ficaram indecisos e 01 (um) discordou. Com relação ao *feedback* emitido, apenas 6,9% ficaram indecisos e os demais concordaram com a utilização no auxílio ao processo ensino-aprendizagem.

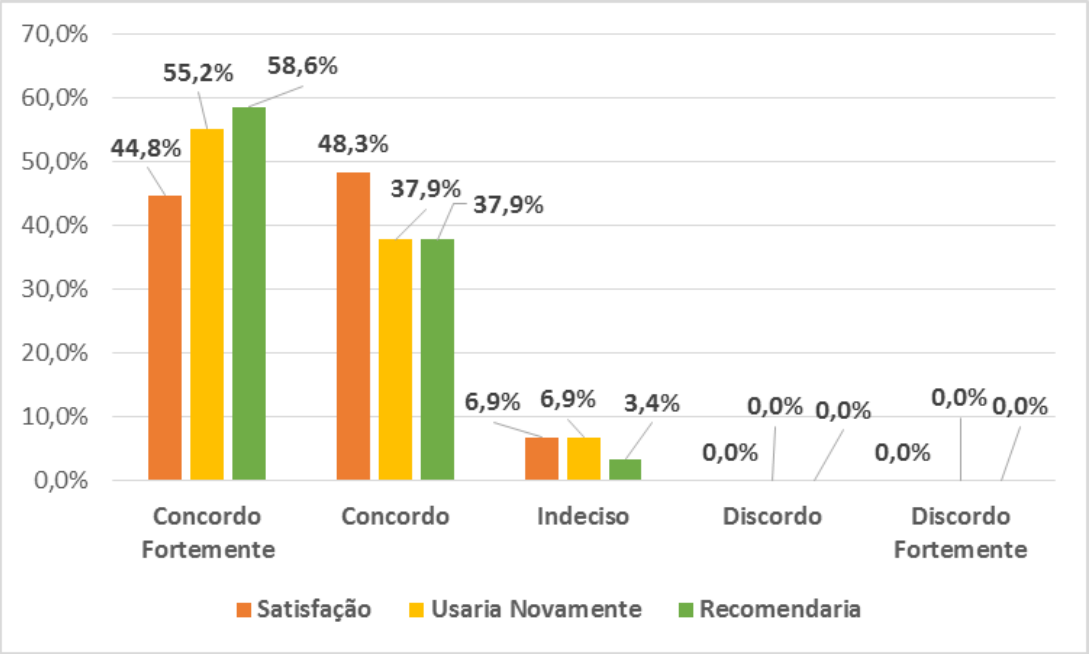
> Um aspecto relevante é o ritmo de andamento do estudo: tipicamente, o conteúdo de uma disciplina de programação é apresentado aos alunos através de aulas expositivas ou práticas, onde o ritmo de evolução do conteúdo é dado pelo professor, tendo-se em vista o cumprimento de um conteúdo programático. Ocorre que esse ritmo não necessariamente se ajusta à condição de cada indivíduo. Alunos com dificuldades em assuntos tratados anteriormente tendem a não acompanhar o conteúdo em sua plenitude, o que pode ocasionar perda de assimilação e, igualmente importante, de interesse, fazendo com que os mesmos passem a ver a disciplina como algo aversivo e preocupando-se essencialmente em conseguir a média mínima para seguir adiante no curso (ROCHA, 2010, p. 2).

Segundo Teixeira (2015), um *feedback* imediato mantém o aluno focado no problema a ser resolvido e pode ajudá-lo a tomar decisões ou corrigir as próximas ações.

Os alunos também foram questionados sobre a sua percepção após o uso da ferramenta, se ficaram satisfeitos com o conhecimento adquirido depois

de utilizá-la, se usariam novamente e recomendariam para outros alunos. Conforme é explicitado no Gráfico 9.

Gráfico 9 – Satisfação e Considerações pós-uso



Fonte: Elaborado pelo Autor, 2016.

A maioria dos alunos concordou que a utilização da ferramenta lhes acrescentou conhecimento, apenas 6,9% ficaram indecisos quanto a esse quesito, 6,9% e 3,4% ficaram indecisos se usariam novamente e recomendariam a utilização, respectivamente. A maior parte dos alunos utilizaria a ferramenta novamente e também recomendariam a utilização por outros.

A última pergunta do questionário tratou-se de uma questão subjetiva, na qual os alunos poderiam indicar sugestões de melhoria da ferramenta. As respostas mais frequentes sugeriram a total adequação para dispositivos móveis, e também a possibilidade de atividades e desafios a serem feitos em dupla ou grupo de alunos.

Os resultados obtidos foram positivos, uma vez que a maioria dos alunos entrevistados ficou satisfeita com a utilização da Escola de Algoritmos. Com relação à interface, os alunos a consideraram atrativa e com boa navegabilidade e animações adequadas à compreensão de conceitos complexos.

A experiência realizada veio confirmar a ideia de que um ambiente visual interativo pode tornar o processo de ensino e aprendizagem mais intuitivo e leve para os alunos, estimulando e motivando-os.

Com base nas respostas fornecidas pelos alunos, percebeu-se que as principais funcionalidades propostas foram úteis para os avaliados, e que de fato agregaram para a experiência de uso. Com a avaliação, também foi possível identificar pontos a serem melhorados, como a adequação para dispositivos móveis, que serão importantes para o desenvolvimento de trabalhos futuros.

6. CONCLUSÃO

Ao longo deste trabalho, abordaram-se as vantagens da utilização das tecnologias no ensino. O uso das TIC na EAD possibilita buscar alternativas que favoreçam o interesse e a representação do pensamento do aluno.

O ensino de algoritmos e programação é um dos grandes desafios na área de ensino de computação, pois são grandes as dificuldades encontradas pelos alunos.

Considerando a revisão de literatura, podemos verificar que são várias as dificuldades encontradas pelos alunos: como a deficiência em matemática, pouco ou nenhum conhecimento na língua inglesa e também problemas relacionados a metodologia utilizada pelos professores.

Esses alunos, considerados nativos digitais, não mais concebem a aprendizagem como algo estático, passivo e vivem em um mundo altamente "conectado", ou seja, dinâmico, sujeito a mudança e colaborativo. As TIC com suas características de dinamismo e compartilhamento rápido de informações induz a escola a repensar as metodologias tradicionais de seus professores.

Se adaptar a realidade desta nova geração, utilizando um ambiente mais lúdico e que considere as especificidades da aprendizagem dos alunos são características indisénsáveis as instituições de ensino. Portanto, é muito importante o desenvolvimento de ferramentas tecnológicas que busquem despertar o interesse do aluno, assim como facilitar o entendimento dos conteúdos e da lógica de programação. Por este motivo, foi proposto o desenvolvimento da ferramenta Escola de Algoritmos, como suporte à aprendizagem de Lógica de Programação, que permita que o aluno aprenda o conteúdo de forma gradual.

O protótipo desenvolvido da ferramenta tem a característica de ser disponibilizado na web, sem a necessidade de instalação de nenhum programa ou *plug-in* adicional podendo ser acessado por meio de qualquer sistema operacional que contenha navegador e esteja conectado à internet.

Ademais, estimula o aprendizado através do *Portugol*, pois o ensino de algoritmos deve independer de uma linguagem de programação específica, e

sim enfatizar a construção de algoritmos, etapa muito importante para o desenvolvimento passo a passo da solução de um problema.

Os resultados da avaliação da ferramenta em turmas de um curso a distância do Instituto Federal do Piauí indicam que a ferramenta é útil no auxílio ao aprendizado, enquanto viabiliza a adequação do conteúdo ao estilo de aprendizagem de cada aluno. A amostra, para outras pesquisas mais conclusivas, poderia ser bem mais numerosa, mas atendeu satisfatoriamente os objetivos deste trabalho.

A partir dos resultados obtidos, pode-se concluir que a ferramenta contribuiu de maneira significativa para a aprendizagem de conteúdos teóricos e a desenvoltura na criação de algoritmos, reduzindo as dificuldades provenientes desse conteúdo.

Espera-se que este trabalho incentive a reflexão acerca de metodologia inovadoras e utilização efetiva da tecnologia a favor da educação. Tornar o aluno mais participativo, motivado e autônomo da sua aprendizagem é um desafio diário a ser conquistado.

Como trabalhos futuros, pretende-se desenvolver os outros módulos da ferramenta e acrescentar recursos de acessibilidade e responsividade, para que possa ser melhor utilizada em dispositivos móveis, possibilitando o acesso por um público ainda maior.

REFERÊNCIAS

ABED – ASSOCIAÇÃO BRASILEIRA DE EDUCAÇÃO A DISTÂNCIA. **Censo ead.br**: relatório analítico da aprendizagem a distância no Brasil 2014. Curitiba: Ibpex, 2015. Disponível em: http://www.abed.org.br/censoead2014/CensoEAD2014_portugues.pdf. Acesso em 09 de dez. 2015.

ALMEIDA, M. E. B. Tecnologia e educação a distância: abordagens e contribuições dos ambientes digitais e interativos de aprendizagem. **Revista Brasileira de Educação a Distância**, n.110, p. 6, 2011.

ALONSO, K. M.; VASCONCELOS, M. A. M. As Tecnologias da informação e Comunicação e a Aprendizagem Colaborativa no Ensino Fundamental. **Revista Contrapontos - Eletrônica**, v. 12, n. 1, p. 58-67, jan./abr. 2012.

ALVES, F. **Gamification**: Como criar experiências de aprendizagem engajadoras. Um guia completo: do conceito à pratica. DVS Editora, 2014.

ALVES, F. P.; JAQUES, P. Um ambiente virtual com feedback personalizado para apoio a disciplinas de programação. In: **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**. 2014. p. 51.

AMORIM, M. C. M. S. et al. Learning and Digital Games: a dialogue with technical secondary education students. **Educação & Realidade**, v. 41, n. 1, p. 91-115, 2016.

ANIDO, R. Saci-ainda outro ambiente para o ensino de programação. In: **XXIII Workshop de Educação em Computação (WEI'2015)**. Recife, PE, Brasil. 2015.

AURELIANO, V. C. O.; TEDESCO, P. C. A. R. Ensino-aprendizagem de Programação para Iniciantes: uma Revisão Sistemática da Literatura focada no SBIE e WIE. In: **Anais do Simpósio Brasileiro de Informática na Educação**. 2012.

BALDUINO, J. O.; FERREIRA, F. S. S. Proposta de uma Nova Abordagem para Desenvolvimento de Algoritmos de Programação. **LINKSCIENCEPLACE-Revista Científica Interdisciplinar,** v. 2, n. 1, 2015.

BARANAUSKAS, M. C. C. et al. **Codesign de Redes Digitais**: Tecnologia e Educação a Serviço da Inclusão Social. Penso Editora, 2013.

BEHAR, P. A. et al. **Competências em educação a distância**. Porto Alegre: Penso Editora, 2013. 312p.

BENITTI, F. B. V; SEARA, E. F. R; SCHLINDWEIN, L. M. Processo de Desenvolvimento de Software Educacional: proposta e experimentação. **Revista Novas Tecnologias na Educação**, v. 3, n. 1, 2005.

BORGES, M. J. M. **Ferramenta Multimédia de Apoio à Programação Algoritmia e Programação Pascal**. Nº de Folhas 217, Dissertação de Mestrado em Tecnologias de Informação e Comunicação. Universidade de Trás-os-Montes e Alto Douro, 2016.

BRAGA, O. R. M. **Objeto de aprendizagem adaptativo no ensino da lógica booleana**. 2015. Nº de Folhas 93, Dissertação de Mestrado em Engenharia Elétrica. Universidade Presbiteriana Mackenzie, São Paulo, 2015.

CARVALHO, C. A.; OLIVEIRA, E. S. G.; SILVA, F. T. B. Aprendizagem e Tecnologias Digitais: novas práticas, jovens aprendizes. **Colóquio Internacional Educação, Cidadania e Exclusão (CEDUCE)**, v. 4, 2015.

CASTRO, T. C. et al. Utilizando programação funcional em disciplinas introdutórias de computação. In: **Anais do X WEI - Workshop sobre Educação em Computação**, 2011.

CRUZ, A. K. B. S.; LIMA, L. C. M. Estudo e Testes de Usabilidade em Sistemas de Autoria de Software: Scratch e Alice. **Blucher Design Proceedings**, v. 1, n. 4, 2014.

FALCKEMBACH, G. A. M.; ARAUJO, F. V. Aprendizagem de algoritmos: dificuldades na resolução de problemas. **Anais SULCOMP**, v. 2, 2013.

FERREIRA, M. A. S. **Proposta de Uma Metodologia Para Ensino-Aprendizagem de Algoritmos**. Trabalho de Conclusão de Curso: Curso de Informática (Bacharelado), Departamento de Ciências Exatas e Tecnológicas, Universidade do Planalto Catarinense. Lages, 2005.

FRANTZ, W. L. S; PONTES, H. P. Um Ambiente de Desenvolvimento Personalizável para o Ensino de Programação. In: **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**. 2014.

GARCIA, E. G.; VEIGA, E. C. O construtivismo e as funções mentais. **Diálogo Educacional**, Curitiba, v. 7, n. 20, jan./abr. de 2007.

GAUDENCIO, M. et al. Eu Sei o que Vocês Fizeram (Agora e) na Aula Passada: o TSTView no Acompanhamento de Exercícios de Programação. In: **Anais do Simpósio Brasileiro de Informática na Educação**. 2013. p. 204

GIRAFFA, L. M. M.; MORA, M. C. Evasão na Disciplina de Algoritmo e Programação: Um Estudo a Partir dos Fatores Intervenientes na Perspectiva do Aluno. In: **Tercera Conferencia sobre el Abandono en la Educación Superior (III CLABES)**, México. 2013.

GIRAFFA, L.; MULLER, L.; MORAES, M. C. Ensinado Programação apoiada por um Ambiente Virtual e Exercícios Associados a Cotidiano dos Alunos: compartilhando alternativas e lições aprendidas. In: **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**. 2015. p. 1330.

GOMES, M. E. N.; D'EMERY, R. A.; FILHO, G. A. A. C. AAPW: Uma Ferramenta para facilitar o Aprendizado de Programação Web. In: **Anais do XXII WEI - Workshop sobre Educação em Computação**. 2014.

GOMES, A. et al. Uma proposta para ajudar alunos com dificuldades na aprendizagem inicial de programação de computadores. In: **Educação, Formação & Tecnologias**; vol.1(1), pp. 93-103, 2008.

GONÇALVES, R. E. F. **Jogo digital para o ensino dos fundamentos da programação.** Nº de Folhas 73, Dissertação de Mestrado Integrado em Engenharia Informática e Computação. Universidade do Porto, 2013.

HENRIQUES, T. C. **Análise dos problemas enfrentados por alunos de programação**. Trabalho de Conclusão de Curso: Bacharelado em Ciência da Computação, Centro de Ciência e Tecnologia, Universidade Estadual da Paraíba. Campina Grande, 2013.

HOSTINS, H.; RAABE, A. Auxiliando a aprendizagem de algoritmos com a ferramenta Webportugol. In: **XV Workshop de Educação em Computação (WEI 2007)**. 2007.

JENKINS, T. On the difficulty of learning to program. In: **Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences**. 2002. p. 53-58.

JÚNIOR, J. C. R. P. et al. AVEP–Um ambiente de apoio ao ensino de algoritmos e programação. In: **XIV WEI - Workshop sobre Educação em Computação**. p. 51-60, 2006.

KENSKI, V. M. **Tecnologias e tempo docente**. Coleção Papirus Educação. Campinas, SP: Papirus, 2013.

KINNUNEN, P.; MALMI, L. Why students drop out cs1 course? **In Proceedings of the second international workshop on Computing education research**, ICER '06, páginas 97-108, New York, NY, USA. 2006. ACM. ISBN 1-59593-494-4.

LAPOLLI, F. R.; GOMES, J. O. **Desenvolvimento de Simulações Integrando Metodologias Ágeis de ES e Conceitos de IHC para o treinamento em Sistemas Complexos**. Nº de Folhas 187. Dissertação de Mestrado em Informática. Universidade Federal do Rio de Janeiro, 2011.

LEAL, A. V. A. **Ensino de programação no ensino médio integrado**: uma abordagem utilizando padrões e jogos com materiais concretos. Nº de Folhas 108. Dissertação do Mestrado em Ciência da Computação. Universidade Federal de Goiás, 2014.

LIKERT, R. **A Technique for the Measurement of Attitudes**. Archives of Psychology**.** Universidade da Virginia**,** 1932. 55p.

LIMA, G. C. B**. AdaptHA**: ambiente para autoria e ensino adaptativo. Nº de Folhas 113. Dissertação do Mestrado em Ciência da Computação. Universidade Federal do Rio Grande do Sul, 2007.

LITTO, F. M.; FORMIGA, M. **Educação a distância:** o estado da arte. Pearson, 2012.

LOPES, M. E. A Importância dos Recursos Didáticos na Educação Presencial e a Distância. **Revista Brasileira de Educação e Cultura**, n. 7, p. 87-99, 2013.

MARQUES, M. C. P. O. **O ensino da programação no desenvolvimento de jogos através do ambiente Scratch**. Nº de Folhas 177. Dissertação do Mestrado em Ensino de Informática. Universidade do Minho, 2013.

MATTAR, J. **Games em educação:** como os nativos digitais aprendem. Editora Prentice Hall, 2010. 208p.

MENDES, A. J. N. **Software educativo para apoio à aprendizagem de programação**. Universidade de Coimbra. Portugal, 2002.

MOORE, M. G. et al. **Educação a distância:** uma visão integrada. Cengage Learning, 2007.

MORAIS, M. A. C. **A importância da educação profissional na modalidade de educação a distância para o desenvolvimento territorial**. 2015. Tese de Doutorado - Universidade Estadual Paulista. 153 f.

NICOLODI, C. F., **Análise do projeto Mutirão pela inclusão Digital na Universidade de Passo Fundo sob a ótica das Arquiteturas Pedagógicas**. Dissertação de Mestrado Dezembro de 2013.

NOSCHANG, L. F. et al. Portugol studio: Uma ide para iniciantes em programaçao. In: **Workshop sobre Educação em Computação**. 2014. p. 1287-1296.

OLIVEIRA, M. M. **Como fazer pesquisa qualitativa**. Editora Vozes, 2012. 184p.

OLIVEIRA, M.; OLIVEIRA, E. Metodologia de Diagnóstico e Regulação de Componentes de Habilidades da Aprendizagem de Programação. In: **Anais do XXII WEI - Workshop sobre Educação em Computação**. 2014.

OREHOVAČKI, T.; BABIĆ, S. Inspecting Quality of Games Designed for Learning Programming. Lecture Notes in Computer Science - **17th International Conference on Human-Computer Interaction**, 2015.

PAES, R. B. et al. Ferramenta para a Avaliação de Aprendizado de Alunos em Programação de Computadores. In: **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**. 2013.

PETERSEN, A. et al. CS1 exam question content. **In: Proceedings of the 42nd ACM technical symposium on Computer science education**. ACM, 2011. p. 631-636.

PÍCCOLO, H. L. et al. Ambiente Interativo e Adaptável para ensino de Programação. In: **Anais do Simpósio Brasileiro de Informática na Educação**. 2010.

POLLY, P. et al. Evaluation of an adaptive virtual laboratory environment using Western Blotting for diagnosis of disease. **BMC Medical Education**, v. 14, n. 1, p. 1, 2014.

PONTES, H. Desenvolvimento de Jogos no Processo de Aprendizado em Algoritmos e Programação de Computadores. **In: Proceedings of the XII Simpósio Brasileiro de Games e Entretenimento Digital (SBGames),** 2013.

POZO, J. I; MONEREO, C. O aluno em ambientes virtuais: condições, perfil e competências. **COLL, C.; MONEREO, C. e colaboradores. Psicologia da educação virtual: aprender e ensinar com as tecnologias da informação e da comunicação**. Porto Alegre: Artmed, p. 97-117, 2010.

PRENSKY, M. Nativos digitais, imigrantes digitais. **De On the Horizon NCB University Press,** v. 9, n. 5, Out. 2001.

_____. Listen to the natives. **Educational leadership**, v. 63, n. 4, 2005.

_____. **Não me atrapalhe, mãe eu estou aprendendo**. São Paulo: Phorte, 2010.

QUEIRÓS, R.; LEAL, J. P. Ensemble: An Innovative Approach to Practice Computer Programming. In: QUEIRÓS, R., **Innovative Teaching Strategies and New Learning Paradigms in Computer Programming**. Hershey: Information Science, 2015. cap. 9.

RAABE, A. L. A.; SILVA, J. M. C. Um ambiente para atendimento as dificuldades de aprendizagem de algoritmos. **In: XXV Congresso da Sociedade Brasileira de Computação.** São Leopoldo, RS, Brasil. 2005.

RAMOS, V. et al. A Comparação da Realidade Mundial do Ensino de Programação para Iniciantes com a Realidade Nacional: Revisão sistemática

da literatura em eventos brasileiros. In: **Anais do Simpósio Brasileiro de Informática na Educação**. 2015. p. 318.

RESNICK, M. Repensando o aprendizado na era digital. In: **Workshop: Scratch e Cricket: Novos ambientes de aprendizagem e de criatividade.** Campinas. 2006.

RIBEIRO, R. S. **Construção e uso de ambiente visual para o ensino de programação introdutória**. Nº de Folhas 59. Dissertação do Mestrado em Ciência da Computação. Universidade de São Paulo, 2015.

RIOS, E. C. S. V. INFORMÁTICA APLICADA NA EDUCAÇÃO. **Conhecimento em Destaque**, v. 4, n. 8, 2015.

ROCHA, P. S. et al. Ensino e aprendizagem de programação: análise da aplicação de proposta metodológica baseada no sistema personalizado de ensino. **Revista Novas Tecnologias na Educação**, v. 8, n. 3, 2010.

SANTOS, C. P.; DOMINGUES, L. A. Atuação do Docente Imigrante Digital para com os Nativos Digitais. **Revista de Educação**, v. 18, n. 24, 2015.

SANTOS, R. P.; COSTA, H. A. X. Análise de metodologias e ambientes de ensino para algoritmos, estruturas de dados e programação aos iniciantes em computação e informática. **Journal of Computer Science**, v. 5, n. 1, p. 41-50, 2006.

SCAICO, P. D. et al. Ensino de Programação no Ensino Médio: Uma Abordagem Orientada ao Design com a Linguagem Scratch. **Revista Brasileira de Informática na Educação**, v. 21, n. 02, p. 92, 2013.

SIEMENS, G. **Conectivismo: uma teoria de aprendizagem para a idade digital**. Trad. Bruno Leite, 2004. Disponível em: <http://pt.scribd.com/doc/66317606/Conectivismouma-Teoria-Para-a-Era-Digital>. Acesso em 17 de mai. 2015.

SILVA, M. V. O. **A Metacomunicação em Interfaces Digitais de Aplicativos Educacionais:** repensando a experiência do aprendizado. Nº de folhas 160, Dissertação de Mestrado em Sistemas de Informação e Gestão do Conhecimento. Universidade FUMEC. 2014.

SILVA, P. et al. Um Mapeamento Sistemático sobre Iniciativas Brasileiras em Ambientes de Ensino de Programação. In: **Anais do Simpósio Brasileiro de Informática na Educação**. 2015. p. 367.

SILVA. J. G. S.; PINTO, C. Ambiente Virtual de Aprendizagem de Algoritmos (Avaa): A Construção de uma Ferramenta no Processo Ensino-Aprendizagem. **Seminário Estudantil de Produção Acadêmica**, v. 12, n. 1, 2013.

SIMON, S. Assignment and sequence: why some students can't recognise a simple swap. In: **Proceedings of the 11th Koli Calling International Conference on Computing Education Research**, Koli Calling '11, p. 10-15, New York, NY, USA. ACM. ISBN 978-1-4503-1052-9, 2011.

SOBRAL, S. R.; PIMENTA, P. C. O ensino da programação: exercitar a distância para combate às dificuldades. In: **4ª Conferência Ibérica de Sistemas e Tecnologias de Informação**. Póvoa de Varzim, 2009.

SOUZA, D. M.; MALDONADO, J. C.; BARBOSA, E. F. Aspectos de Desenvolvimento e Evolução de um Ambiente de Apoio ao Ensino de Programação e Teste de Software. In: **Anais do Simpósio Brasileiro de Informática na Educação**. 2012.

TORI, R. **Educação sem distância**: as tecnologias interativas na redução de distâncias em ensino e aprendizagem. São Paulo: Senac, 2010. 256p.

TAVARES, P. et al. Técnicas para aumentar o envolvimento dos alunos na aprendizagem da programação. In: **VII Congresso Mundial de Estilos de Aprendizagem**. Bragança, 2016. p. 1565-1577.

TEIXEIRA, K. L. **Contribuição do ensino adaptativo para aprendizagem de conceitos matemáticos**. Nº de Folhas 141. Dissertação de Mestrado em Educação e Tecnologias Digitais. Universidade de Lisboa, 2015.

TORRES, P. L.; BEHRENS, M. A.; MATOS, E. M. Prática pedagógica numa visão complexa na educação presencial e a distância: os ‘REAS’ como recurso para pesquisar, ensinar e aprender. **Revista Diálogo Educacional**, v. 15, n. 45, p. 443-471, 2015.

TSUKAMOTO, N. M. S.; FIALHO, N. E; TORRES, P. L. A face educacional do facebook: um relato de experiência. IN: PORTO, C.; SANTOS, E. **Orgs.**

Facebook e Educação: publicar, curtir, compartilhar [online]. Campina Grande: EDUEPB**,** p. 349-364 2014.

VALASKI, J.; PARAISO, E. C. Limitações da Utilização do Alice no Ensino de Programação para alunos de graduação. In: **Anais do Simpósio Brasileiro de Informática na Educação**. 2012.

VALENTIM, H. **Um Estudo sobre o Ensino-aprendizagem de Lógica de Programação**. Universidade Tecnológica Federal do Paraná. Programa de Pós-Graduação em Educação Científica e Tecnológica. Curitiba, 2011. Disponível em: <http://www.foco.fae.ufmg.br/pdfs/137.pdf>. Acesso em 16 de out. 2015.

YAU, J. Y.-K.; JOY, M. A context-aware personalised m-learning application based on m-learning preferences. **International Journal of Mobile Learning and Organisation, Inderscience**, v. 5, n. 1, p. 1–14, 2011.

YESSAD, A.; LABAT, J.; KERMORVANT, F. Segae: A serious game authoring environment. In: **2010 10th IEEE International Conference on Advanced Learning Technologies**. IEEE, 2010. p. 538-540.

APÊNDICE A

QUESTIONÁRIO SOBRE DIFICULDADES DE APRENDIZAGEM DE PROGRAMAÇÃO

UTILIZE COMO BASE AS DISCIPLINAS DE LÓGICA DE PROGRAMAÇÃO E INTRODUÇÃO A PROGRAMAÇÃO PARA WEB.

I. DADOS PESSOAIS

Nome (opcional): _____ Idade: ____
Sexo: M ☐ F ☐ Grau de Instrução: _____

II. DADOS RELACIONADOS A APRENDIZAGEM

1) Como você considera seu aprendizado nas disciplinas citadas?
   a) Excelente
   b) Bom
   c) Regular
   d) Ruim
   e) Péssimo

2) Como você considera o material (livros, fóruns, atividades, etc) utilizado nas disciplinas citadas?
   a) Excelente
   b) Bom
   c) Regular
   d) Ruim
   e) Péssimo

3) Após ter cursado as disciplinas você se considera apto a desenvolver/codificar programas (softwares)?
   a) Sim
   b) Não

4) Em relação a interatividade, como você considera o material utilizado nas disciplinas?
   a) Excelente
   b) Bom
   c) Regular
   d) Ruim
   e) Péssimo

5) Você utilizou alguma ferramenta adicional (site, software, curso on-line) para favorecer o aprendizado nas disciplinas?
   a) Sim
   b) Não

6) Como você considera a ideia da utilização de uma ferramenta tecnológica que favorecesse o aprendizado?
   a) Excelente
   b) Bom
   c) Regular
   d) Ruim
   e) Péssimo

7) Como você preferiria utilizar essa ferramenta?
   a) No computador, separadamente da plataforma Moodle
   b) No computador, incorporado a plataforma Moodle
   c) No celular/smartphone
   d) Outro meio

8) O que consideras que poderia melhorar os seus resultados nas disciplinas ligadas à programação.

APÊNDICE B

QUESTIONÁRIO DE AVALIAÇÃO DA FERRAMENTA ESCOLA DE ALGORITMOS

I. DADOS PESSOAIS

Nome (opcional): _____ Idade: ____
Sexo: M ☐ F ☐

II. DADOS RELACIONADOS AO APRENDIZADO

1) Como você considera seu aprendizado na disciplina Lógica de Programação.

☐ Muito difícil ☐ Regular ☐ Muito fácil
☐ Difícil ☐ Fácil

2) Você teve o auxílio de alguma ferramenta tecnológica na disciplina Lógica de Programação?

☐ Sim ☐ Não ☐ Se sim, qual:_____

III. DADOS RELACIONADOS A EXPERIÊNCIA DE USO DA FERRAMENTA

1) A ferramenta é apropriada para os estudos de Lógica de Programação.

☐ Concordo Fortemente ☐ Indeciso ☐ Discordo Fortemente
☐ Concordo ☐ Discordo

2) O conteúdo apresentado é bem organizado e fácil de entender.

☐ Concordo Fortemente ☐ Indeciso ☐ Discordo Fortemente
☐ Concordo ☐ Discordo

3) Os meios utilizados para apresentar as informações aumentam a compreensão do conteúdo.

☐ Concordo Fortemente ☐ Indeciso ☐ Discordo Fortemente
☐ Concordo ☐ Discordo

4) A ferramenta fornece conceitos/conteúdos e contribuição útil para a construção de habilidades de programação.

☐ Concordo Fortemente ☐ Indeciso ☐ Discordo Fortemente
☐ Concordo ☐ Discordo

5) A interface é atraente.

☐ Concordo Fortemente ☐ Indeciso ☐ Discordo Fortemente
☐ Concordo ☐ Discordo

6) A quantidade de informações por tela é adequada.

☐ Concordo Fortemente ☐ Indeciso ☐ Discordo Fortemente
☐ Concordo ☐ Discordo

7) Os aspectos de texto, imagem e vídeos utilizados me incentivam a utilizar a ferramenta.

☐ Concordo Fortemente ☐ Indeciso ☐ Discordo Fortemente
☐ Concordo ☐ Discordo

8) Ao concluir as atividades, senti-me realizado, satisfeito e com a certeza de que acrescentou conhecimento.

☐ Concordo Fortemente ☐ Indeciso ☐ Discordo Fortemente
☐ Concordo ☐ Discordo

9) Usaria a ferramenta novamente.

☐ Concordo Fortemente ☐ Indeciso ☐ Discordo Fortemente
☐ Concordo ☐ Discordo

10) Recomendaria para outros usuários.

☐ Concordo Fortemente ☐ Indeciso ☐ Discordo Fortemente
☐ Concordo ☐ Discordo

11) A ferramenta evolui em um ritmo adequado e não fica monótono, pois oferece novos obstáculos, situações ou variações de atividades.

☐ Concordo Fortemente ☐ Indeciso ☐ Discordo ☐ Discordo Fortemente ☐ Concordo

12) As características de adaptabilidade permitiram a adequação ao meu estilo de aprendizagem, enfatizando conteúdos em que apresentei mais dificuldade.

☐ Concordo Fortemente ☐ Indeciso ☐ Discordo ☐ Discordo Fortemente ☐ Concordo

13) As funções básicas da ferramenta são fáceis de usar, facilitando a interação e o aprendizado.

☐ Concordo Fortemente ☐ Indeciso ☐ Discordo ☐ Discordo Fortemente ☐ Concordo

14) O feedback oferecido auxiliou no entendimento das atividades e na melhoria do aprendizado.

☐ Concordo Fortemente ☐ Indeciso ☐ Discordo ☐ Discordo Fortemente ☐ Concordo

15) Depois da ferramenta, compreendo melhor os temas apresentados e consigo aplicá-los.

☐ Concordo Fortemente ☐ Indeciso ☐ Discordo ☐ Discordo Fortemente ☐ Concordo

16) Com base na sua avaliação, nos forneça dicas e sugestões de melhoria da ferramenta.

_____

_____

_____

ANEXOS

PLANO DE DISCIPLINA

Curso Técnico em Informática para Web		**Modalidade: Semipresencial**	
Ano: 2014	**Semestre: 1º**		
Disciplina: Lógica de Programação			
Carga Horária: 60 h	**Duração: 10 semanas**	**Início: 03/05/2014**	**Término: 19/07/2014**
Endereço Eletrônico: http://162.243.44.232/moodle2.6/			
Professor Regente/Formador: Jeanne de Souza Silva Leite			
Tutor a Distância: Silvino Marques da Silva Junior			
Tutor Presencial: Paulizangela Pereira da Silva			
Coordenador Curso: Ricardo Martins Ramos			

EMENTA:

- Introdução a Lógica de Programação, Conceitos de Programação; Estruturas de Decisão; Estruturas de Repetição, Estruturas de Dados, Funções.

OBJETIVO GERAL

- Proporcionar aos alunos uma compreensão de como resolver problemas utilizando o computador. Além disso, capacitá-los a escrever pequenos programas em uma linguagem de programação. Será utilizada a linguagem de programação Python.

UNIDADES DE ESTUDO

UNIDADE I - INTRODUÇÃO A LÓGICA DE PROGRAMAÇÃO

Objetivos Específicos:

- Entender o que é Lógica e Lógica de Programação;
- Entender como resolver problemas com o uso do computador;
- Saber o conceito de Algoritmos e suas formas de representação;
- Saber o conceito de Linguagem de Programação;
- Conhecer os conceitos de sintaxe e semântica de uma linguagem de programação;
- Conhecer os tipos de erros que podem aparecer em um programa;
- Entender os níveis de abstração e os processos de compilação e Interpretação das linguagens de programação.

Conteúdo Programático Semanal – 03/05/2014 a 10/05/2014 – CH: 06 HORAS

Aula 1: Introdução a Lógica de Programação
Aula 2: Algoritmos
Aula 3: Linguagem de Programação

UNIDADE II – CONCEITOS DE PROGRAMAÇÃO
Objetivos Específicos • Saber dos principais conceitos para construção de um programa; • Conhecer os tipos de dados, operadores, variáveis, expressões; • Conhecer os principais comandos de um programa: comandos de atribuição, de entrada e saída de dados.
Conteúdo Programático Semana II – 10/05/2014 a 17/05/2014– CH: 06 HORAS **Aula 4:** Construindo Programas **Aula 5:** Elementos de um programa **Aula 6:** Principais Comandos
UNIDADE III - INSTRUÇÕES SEQUENCIAIS
Objetivos Específicos • Construir programas sequenciais para resolução de problemas simples
Conteúdo Programático Semana III – 17/05/2014 a 24/05/2014 – CH: 06 HORAS **Aula 7:** Programas Sequenciais
UNIDADE IV - INSTRUÇÕES DE DESVIOS CONDICIONAIS
Objetivos Específicos • Entender como elaborar programas com tomadas de decisões; • Construir programas com as principais estruturas de decisão.
Conteúdo Programático Semana IV – 24/05/2014 a 31/05/2014 – CH: 06 HORAS **Aula 8:** Estruturas de Decisão Simples **Aula 9:** Estruturas de Decisão Composta
UNIDADE V - COMANDOS DE REPETIÇÃO
Objetivos Específicos • Entender como elaborar programas iterativos; • Construir programas com as principais estruturas de repetição.
Conteúdo Programático Semana V – 31/05/2014 a 07/06/2014 – CH: 06 HORAS **Aula 10:** Estruturas de Repetição Condicional; **Aula 11:** Estruturas de Repetição Automática; **Aula 12:** Acumuladores: Somatório e Produtório;
UNIDADE VI - BOAS PRÁTICAS DE PROGRAMAÇÃO
Objetivos Específicos • Entender as práticas e estilos de elaboração de um bom programa
Conteúdo Programático Semana VI – 07/06/2014 a 14/06/2014– CH: 06 HORAS **Aula 13:** Boas Práticas de Programação
UNIDADE VII - ESTRUTURAS DE DADOS
Objetivos Específicos

- Construir programas que armazenam dados em Memória

Conteúdo Programático Semana VII – 14/06/2014 a 21/06/2014– CH: 06 HORAS

Aula 14: Vetores
Aula 15: Matrizes

UNIDADE VIII - FUNÇÕES

Objetivos Específicos

- Entender os conceitos de modularização de programas, parametrização e reaproveitamento de código;
- Construir programas modulares através de funções.

Conteúdo Programático Semana VIII – 21/06/2014 a 28/06/2014– CH: 06 HORAS

Aula 16: Modularização e Método de Divisão e Conquista;
Aula 17: Funções.

UNIDADE IX - PRINCIPAIS PARADIGMAS DE PROGRAMAÇÃO

Objetivos Específicos

- Entender os principais paradigmas e características das linguagens de programação.

Conteúdo Programático Semana IX - 28/06/2014 a 05/07/2014– CH: 06 HORAS

Aula 18: Paradigmas de Programação;

UNIDADE X – REVISÃO

Objetivos Específicos

- Revisar os principais conceitos e práticas estudados no curso

Conteúdo Programático Semana X - 05/07/2014 a 12/07/2014 – CH: 06 HORAS

Aula 19: Atividades de revisão da parte teórica;
Aula 20: Atividades de revisão da prática de programação.

PROCEDIMENTOS DE ENSINO/ATIVIDADES

No Moodle:

Primeira Semana de 03/05/2014 a 10/05/2014

- Vídeo aula da Unidade I - **INTRODUÇÃO A LÓGICA DE PROGRAMAÇÃO**
- Atividade de Classe;
- Atividade de Casa;
- Fórum de discussão.

Segunda Semana de 10/05/2014 a 17/05/2014

- Vídeo aula da Unidade II – **CONCEITOS DE PROGRAMAÇÃO**
- Atividade de Classe;
- Atividade de Casa;
- Fórum de discussão.

Terceira Semana de 17/05/2014 a 24/05/2014

- Vídeo aula da Unidade III - **INSTRUÇÕES SEQUENCIAIS**
- Atividade de Classe;

- Atividade de Casa;
- Fórum de discussão.

Quarta Semana de 24/05/2014 a 31/05/2014

- Vídeo aula da Unidade IV - **INSTRUÇÕES DE DESVIOS CONDICIONAIS**
- Atividade de Classe;
- Atividade de Casa;
- Fórum de discussão.

Quinta Semana de 31/05/2014 a 07/06/2014

- Vídeo aula da Unidade V- **COMANDOS DE REPETIÇÃO**
- Atividade de Classe;
- Atividade de Casa;
- Fórum de discussão.
- **1ª Avaliação de Aprendizagem**

Sexta Semana de 07/06/2014 a 14/06/2014

- Vídeo aula da Unidade VI - **BOAS PRÁTICAS DE PROGRAMAÇÃO**
- Atividade de Classe;
- Atividade de Casa;
- Fórum de discussão.

Sétima Semana de 14/06/2014 a 21/06/2014

- Vídeo aula da Unidade VII - **ESTRUTURAS DE DADOS**
- Atividade de Classe;
- Atividade de Casa;
- Fórum de discussão.

Dia 14/07/2014 : Avaliação de 2ª Chamada

Oitava Semana de 21/06/2014 a 28/06/2014

- Vídeo aula da Unidade VIII - **FUNÇÕES**
- Atividade de Classe;
- Atividade de Casa;
- Fórum de discussão.

Nona Semana de 28/06/2014 a 05/07/2014

- Vídeo aula da Unidade IX - **PRINCIPAIS PARADIGMAS DE PROGRAMAÇÃO**
- Atividade de Classe;
- Atividade de Casa;
- Fórum de discussão.

Décima Semana de 05/07/2014 a 12/07/2014

- Vídeo aula de Revisão
- **2ª Avaliação de Aprendizagem;**
- Encerramento da Disciplina.

Dia 19/07/2014 : Avaliação de 2ª Chamada

AVALIAÇÃO DA APRENDIZAGEM

O aluno será avaliado durante as atividades realizadas no moodle observando a frequência; interesse; interatividade; participação no fórum; questionamentos, reflexões e críticas sobre os conteúdos abordados; clareza e objetividade dos textos/mensagens enviadas.

Durante os momentos presenciais: frequência ao polo, pontualidade na resolução das atividades, interatividade e participação das atividades em grupo e/ou individuais.

ACOMPANHAMENTO E AVALIAÇÃO

1 Acompanhamento:

O acompanhamento será feito continuamente pelo tutor de cada turma, para verificar a aprendizagem do aluno durante todo o desenvolvimento dos conteúdos da área de conhecimento, orientando-o nas suas dificuldades e dúvidas, e buscando aprofundar a sua motivação em todo o processo ensino/aprendizagem, numa perspectiva de avaliação formativa. A avaliação parte do estabelecimento de uma rotina de observação, descrição e análises contínuas da produção do aluno, devendo ser expressa em diferentes níveis e momentos. Esta avaliação mantém a condição de processo, visando ao acompanhamento do percurso de estudo do aluno, das interlocuções entre os pares e com os tutores e da produção de trabalhos escritos que possibilitem uma síntese dos conhecimentos trabalhados.

O acompanhamento e a avaliação, fundamentados nos princípios da participação ativa, dentro de numa perspectiva de aprendizagem cooperativa, fomentarão a construção da autonomia do educando e a sua inserção na dinamização dos processos de ensino e de aprendizagem, considerando as efetivas contribuições nas discussões engendradas nos chats e nos fóruns, bem como o desenvolvimento das atividades propostas, a capacidade de reflexão crítica, a produção de sínteses e a apropriação e domínio dos processos e recursos tecnológicos próprios da modalidade da EAD da era digital.

A participação do aluno será avaliada a partir das interações desencadeadas ao longo do curso, compreendendo a participação nos fóruns, a leitura dos conteúdos e textos indicados, o desenvolvimento das atividades propostas e/ou realização de exercícios e os seus questionamentos nos encontros presenciais; levando em conta não somente os aspectos quantitativos, mas fundamentalmente, os aspectos qualitativos que denotem compreensão, clareza de ideias, objetividade, criatividade, coerência intrínseca, visão crítica, relevância e pertinência aos conteúdos abordados, considerando, ainda, a pontualidade, o envolvimento e a assiduidade aos eventos educativos.

As provas serão aplicadas presencialmente nos Polos, pelo Tutor de cada turma, constando de questões objetivas e/ou subjetivas, nas datas definidas nesse Plano de Disciplina.

2 Avaliação:

Avaliação consistirá num processo contínuo onde se prevê quatro procedimentos principais:

Primeiro procedimento

O aluno estudará e realizará os exercícios propostos no material didático da disciplina. Além destes exercícios, há sugestões de sites para consultas dos assuntos abordados, vídeo aulas (que o ajudarão a revisar ou reforçar o conhecimento sobre um ou mais aspectos do conteúdo). O aluno poderá buscar orientações complementares de acordo com a necessidade, seja através do sistema de tutoria ou de outro sistema de apoio disponível. Os exercícios constantes no material didático são auto avaliativos, para que o aluno possa mensurar seu nível de compreensão dos textos lidos.

Segundo procedimento

O aluno será acompanhado de forma sistemática pelos tutores, através das ferramentas de avaliação oferecidas pela plataforma do curso. Serão observados e analisados, entre outros: o interesse e a iniciativa para a leitura, o estudo e a pesquisa; a participação nas atividades presenciais; a participação nos fóruns; a capacidade de questionar, refletir e criticar os conteúdos e abordagens propostas na disciplina; a interlocução com os tutores e colegas de curso; o acompanhamento das discussões e abordagens propostas no material didático.

Terceiro procedimento:

O aluno realizará atividades avaliativas, através das quais será verificado o progresso na aquisição de habilidades e competências previstas. Estas atividades serão elaboradas pelo professor conteudista da disciplina e discutidas com o professor regente e os tutores. A escolha dos instrumentos para obtenção de dados e informações deverá ser bastante criteriosa e ter em vista as características e objetivos da disciplina. Dentre eles, salientamos: trabalhos escritos individuais ou em grupo; pesquisa de campo, seminários; provas; estudo de caso, material didático, comentários sobre textos, vídeos e áudios, solução de casos práticos.

Quarto procedimento

Considerando a exigência legal do MEC para os cursos a distância, será realizada uma avaliação presencial ao final

de cada disciplina. Também nesse caso, os instrumentos e estratégias escolhidos deverão estar articulados com os objetivos, os conteúdos e as práticas pedagógicas adotadas. A avaliação será elaborada pelo especialista da disciplina e discutida com os tutores e coordenadores. O processo de impressão, empacotamento e transporte da avaliação será acompanhado por coordenadores que, também, estarão presentes nos polos no momento de sua aplicação. Este procedimento tem por objetivo garantir o sigilo e a segurança da avaliação, tendo em vista a confiabilidade e credibilidade dos resultados. Somente após a realização desses quatro procedimentos de avaliação é que será feita a valoração final do desempenho do aluno, traduzida em número de 0 a 10,0, por exigência de normas do IFPI.

VERIFICAÇÃO DE RENDIMENTO

Para efeito de aprovação e reprovação nos Cursos de Educação Técnica Subsequente na modalidade a distância, serão aplicados os critérios abaixo especificados:

Avaliações em escala de 0 (zero) a 10 (dez):
De 06 a 10 = Aprovado por Média
De 04 a 5,9 = Recuperação
De 0,0 a 3,9 = Reprovação
OBS: Caso o aluno fique de recuperação, a nota final será calculada da seguinte forma:

$$\textbf{Nota Final} = \frac{\text{nota obtida na recuperação} + \text{media obtida na disciplina}}{2}$$

A Nota Final será considerada aprovativa se for maior ou igual a 6.

DA PARTICIPAÇÃO EM FÓRUNS:
Nota: de 0 a 10 pontos
Critérios da avaliação quantitativa e qualitativa: Participação nas discussões, levando em consideração a fundamentação teórica do material didático e pesquisas; capacidade de argumentação e estruturação das idéias; análise crítica das temáticas propostas; a interação do aluno com os demais colegas a partir das observações e comentários realizados; contribuições sintetizadoras e/ou questionadoras; ética e respeito às opiniões apresentadas.
Quem avalia o aluno: tutor a distância.

DA REALIZAÇÃO DE EXERCÍCIOS E/OU PARTICIPAÇÃO EM SEMINÁRIOS:
Nota: de 0 a 10 pontos
Critérios da avaliação quantitativa e qualitativa dos exercícios: pontualidade na entrega da atividade e verificação de acertos nas questões propostas, observando o nível de profundidade e complexidade que o aluno apresenta.
Critérios da avaliação quantitativa e qualitativa dos seminários: pontualidade e freqüência; demonstração da compreensão do tema estudado; clareza e objetividade no desenvolvimento do tema proposto; organização e integração dos grupos de trabalho; qualidade e utilização dos recursos didáticos.
Quem avalia o aluno: tutor presencial.

DA PROVA:
Nota: de 0 a 10 pontos
Critérios da avaliação quantitativa e qualitativa: observação do nível de profundidade e complexidade que o aluno atingiu através das respostas às questões objetivas e/ou subjetivas.
Quem avalia o aluno: **tutor presencial.**

CÁLCULO DA MÉDIA POR DISCIPLINA (MD):

$$\textbf{MD} = \frac{(\text{nota da prova} \times 4) + (\text{nota da atividade/seminário} \times 3) + (\text{nota do fórum} \times 3)}{10}$$

BIBLIOGRAFIA BÁSICA:

- WATANABE, Edson. **Lógica de Programação de Computadores** - Florianópolis: Publicações do IF-SC, 2010.
- STEINMETZ, Ernesto; HENRIQUE, Radis. **Cartilha: Lógica de Programação** - Brasília, DF : IFB, 2012.
- MILLER Brad, RANUM David. **Como Pensar Como um Cientista da Computação Aprendendo com Python**. Disponível em: http://panda.ime.usp.br/panda/static/PensePython/
- Notas de Aula da Professora;

BIBLIOGRAFIA COMPLEMENTAR:

- ACHER, David. LUTZ, Mark. **Aprendendo Python**. 2ª. Ed. O'REILLY, 2007.

SITES NA INTERNET:

- Comunidade Python no Brasil: http://www.python.org.br/
- Página Oficial do Python: http://www.python.org