



UNIVERSIDADE ESTADUAL DO NORTE DO PARANÁ
CAMPUS LUIZ MENEGHEL

IVAN CESAR DE CARVALHO

**DESENVOLVIMENTO DE UM OBJETO DE APRENDIZAGEM
PARA O ENSINO DOS CONCEITOS INICIAIS DE
PROGRAMAÇÃO UTILIZANDO A METODOLOGIA GIED**

BANDEIRANTES - PR

2012



UNIVERSIDADE ESTADUAL DO NORTE DO PARANÁ
CAMPUS LUIZ MENEGHEL

IVAN CESAR DE CARVALHO

**DESENVOLVIMENTO DE UM OBJETO DE APRENDIZAGEM PARA O ENSINO
DOS CONCEITOS INICIAIS DE PROGRAMAÇÃO UTILIZANDO A
METODOLOGIA GIED**

Trabalho de Conclusão de Curso apresentado à Universidade Estadual do Norte do Paraná *Campus* Luiz Meneghel como requisito para obtenção do grau de bacharel em Sistemas de Informação.

Orientador: Prof.^o Estevan Braz Brandt Costa.

BANDEIRANTES – PR

2012

IVAN CESAR DE CARVALHO

DESENVOLVIMENTO DE UM OBJETO DE APRENDIZAGEM PARA O ENSINO DOS CONCEITOS INICIAIS DE PROGRAMAÇÃO UTILIZANDO A METODOLOGIA GIED

Trabalho de Conclusão de Curso apresentado à Universidade Estadual do Norte do Paraná *Campus* Luiz Meneghel como requisito para obtenção do grau de bacharel em Sistemas de Informação, orientado pelo Prof.^o Estevan Braz Brandt Costa.

COMISSÃO EXAMINADORA

Prof. Me. Christian James de Castro Bussmann
UENP – *Campus* Luiz Meneghel

Prof. Me. Ederson Marcos Sgarbi
UENP – *Campus* Luiz Meneghel

Prof. Estevan Braz Brandt Costa
UENP – *Campus* Luiz Meneghel

Bandeirantes, _____ de _____ de 2012

Dedico este trabalho a minha esposa, a minha família e a todos que acreditaram que um dia eu chegaria até aqui.

AGRADECIMENTOS

Agradeço primeiramente a Deus, pois é a essência de tudo e que sempre esteve presente em todos os momentos da minha vida. Aprendi muito nesse tempo e agradeço por isso.

A minha amada esposa Graziéle, que soube entender as minhas ausências. A minha família que é a minha razão maior. E aos meus amigos (impossível citar todos), por acreditarem em mim nos momentos que nem mesmo eu acreditei.

Ao Grupo de Oração Jovem Rainha da Paz por fazer parte da minha vida e me ajudar a ser quem eu sou.

Agradeço ao professor Estevan pelo incentivo e paciência. Ao professor Adriano (amigo) pela amizade e apoio, aos professores da UENP - CLM que colaboraram com esse trabalho.

Aos meus amigos de sala que sempre estiveram ao meu lado, mostrando muito mais que uma amizade. Serão para sempre lembrados em meu coração.

Em palavras não conseguirei expressar tudo o que sinto nesse momento... Mas deixo a minha mensagem dizendo que existe um tempo para cada coisa e quando buscamos o reino de Deus, tudo mais nos é acrescentado. Confesso que quase desisti em muitos momentos, não foi fácil; mas hoje vejo que também não era impossível.

*Sem a curiosidade
que me move,
que me inquieta,
que me insere na busca,
não aprendo nem ensino.
(Paulo Freire)*

RESUMO

O presente trabalho foi realizado devido à grande dificuldade encontrada por alunos na área de computação com relação aos conceitos iniciais da disciplina de programação, que acabam levando o aluno à reprovação ou até mesmo à desistência do curso logo nos semestres iniciais. Foram realizadas entrevistas com alunos e professores para conhecer o seu ponto de vista diante dessa realidade. Também foram estudadas metodologias de desenvolvimento de objeto de aprendizagem e desenvolvido um objeto de aprendizagem para auxiliar no processo de ensino e aprendizagem de programação, abordando os conceitos iniciais da disciplina de programação I.

Palavras-chave: Programação. Ensino de Programação. Objeto de Aprendizagem. Lógica. Lógica de Programação.

ABSTRACT

The present work was done due to the great difficulty encountered by students in the area of computing with respect to the initial concepts of programming discipline, which lead the student to fail or even the withdrawal of the course just in the initial semesters. Interviews were conducted with students and teachers to know your view on this reality, were also studied methods of developing learning object and an object designed to assist in the learning process of teaching and learning programming, covering the initial concepts of the discipline.

Keywords: Programming. Teaching Programming. Learning Programming. Learning Object. Logic. Logic Programming.

LISTA DE FIGURAS

Figura 2.1 - Fluxo da metodologia de desenvolvimento de OA.....	25
Figura 3.1 – Mapa Conceitual	46
Figura 3.2 – Mapa Navegacional.....	47
Figura 3.3 – <i>Storyboard</i> tela 1	48
Figura 3.4 – <i>Storyboard</i> tela 2.....	48
Figura 3.5 – <i>Storyboard</i> tela 3.....	49
Figura 3.6 – <i>Storyboard</i> tela 4.....	50
Figura 3.7 – <i>Storyboard</i> tela 5.....	51
Figura 3.8 – <i>Storyboard</i> Tela 11.....	51
Figura 3.9 – <i>Storyboard</i> tela 28.....	52
Figura 3.4 - <i>Storyboard</i> tela 29.....	53
Figura 3.5 – <i>Storyboard</i> tela 33.....	53
Figura 3.6 – <i>Storyboard</i> tela 34.....	54
Figura 3.7 – <i>Storyboard</i> tela 35.....	54
Figura 3.8 – <i>Storyboard</i> tela 36.....	55
Figura 3.9 – <i>Storyboard</i> tela 37.....	56
Figura 3.10 – <i>Storyboard</i> tela 38.....	57
Figura 3.11 – <i>Storyboard</i> tela 39.....	57
Figura 3.12 – <i>Storyboard</i> tela 40.....	58
Figura 3.13 - Tela 1.....	59
Figura 3.14 - Tela 2.....	60
Figura 3.15 - Tela 4.....	60
Figura 3.16 - Tela 33.....	61
Figura 3.17 - Tela 34.....	61
Figura 3.18 - Tela 38.....	62

LISTA DE GRÁFICOS

Gráfico 3.1 – Faixa de idade dos alunos	32
Gráfico 3.2 – Motivo que o levou a fazer o curso de Sistemas de Informação.....	33
Gráfico 3.3 – Compromisso diário além do Curso.....	33
Gráfico 3.4 – Experiência com Programação.....	34
Gráfico 3.5 – Seu conhecimento sobre programação	35
Gráfico 3.6 – Conceitos que o aluno ainda não conhece ou com que se sente inseguro.	36
Gráfico 3.7 – Como o aluno se prepara para a disciplina de programação.....	37
Gráfico 3.8 – Frequência com que o aluno estuda em casa por conta própria	38
Gráfico 3.9 – Ambiente da sua casa com relação ao computador	38
Gráfico 3.10 – Além do estudo, o que mais gosta de fazer no tempo livre	39
Gráfico 3.11 – Como você considera a disciplina de programação.....	40
Gráfico 3.12 – Importância que o aluno atribui à disciplina de programação	40
Gráfico 3.13 – Uma boa abordagem para aprender os conceitos de programação ..	41
Gráfico 3.14 – Dependência na disciplina de programação	42

LISTA DE TABELAS

Tabela 3.1 – Critérios Técnico-Pedagógicos Parte 1	63
Tabela 3.2 – Critérios Técnico-Pedagógicos Parte 2	64
Tabela 3.3 - Usabilidade	65

LISTA DE SIGLAS

AVEA	Ambiente Virtual de Ensino e Aprendizagem
GIED	Grupo de Informática Educativa
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
LabVirt	Laboratório Didático Virtual
MEC	Ministério da Educação
OA	Objeto de Aprendizagem
ROA	Repositório de Objetos de Aprendizagem
SEED	Secretaria de Educação a Distância
UENP-CLM	Universidade Estadual do Norte do Paraná – <i>Campus</i> Luiz Meneghel
VRML	Linguagem para Modelagem de Realidade Virtual

SUMÁRIO

1 INTRODUÇÃO	13
1.2 Motivação e Justificativa	14
1.3 Objetivo Geral	15
1.3.1 Objetivos Específicos	15
1.4 Metodologia	15
1.5 Organização do Trabalho	16
2 FUNDAMENTAÇÃO TEÓRICA.....	17
2.1. Considerações Iniciais	17
2.2 Informática na Educação	17
2.3 O Ensino de Programação.....	18
2.3.1 Problemas de natureza didática:	18
2.3.2 Problemas de natureza cognitiva:	19
2.3.3 Problemas de natureza afetiva:.....	20
2.4 Objetos de Aprendizagem	21
2.5 Metodologias de Desenvolvimento de OAs	24
2.5.1 GIED	25
2.6 Software de autoria <i>Visual Class</i>	27
3 DESENVOLVIMENTO	30
3.1 Metodologia	30
3.2 Levantamento dos dados feito com os Alunos	32
3.3 Levantamento dos dados feito com os Professores	42
3.4 Fases do GIED	44
3.4.1 Análise	44
3.4.1 Projeto.....	45
3.4.2 Implementação.....	59
3.4.3 Validação	62
4 RESULTADOS OBTIDOS	67
5 CONCLUSÃO.....	68
6 TRABALHOS FUTUROS	69
7 REFERÊNCIAS.....	70
APÊNDICE A.....	74
APÊNDICE B.....	78
APÊNDICE C	80
APÊNDICE D	111
APÊNDICE E.....	118

1 INTRODUÇÃO

O modo tradicional de ensino, que consiste em apresentação teórica, exemplos e exercícios práticos simples e sequencialmente complexos, tem tornado um dos problemas em relação ao ensino de programação, pois ele não consegue motivar o aluno facilmente, em especial nos períodos iniciais do curso de computação (RODRIGUES, 2004).

Surge, então, um desafio para o professor: motivar os alunos e fazê-los entender que a disciplina e o seu conteúdo são importantes e que merecem uma atenção especial. Dentro, ainda, dos problemas do método de ensino tradicional da disciplina, outros pontos devem ser analisados como: a forma de avaliação, o relacionamento professor-aluno e a falta de um bom material didático (RODRIGUES, 2004).

Diante das dificuldades encontradas no ensino de programação, torna-se essencial a adoção de novas técnicas de aprendizado. Dentre os vários recursos utilizados em outras áreas da educação, podem-se destacar os Objetos de Aprendizagem (OAs), que surgem com a necessidade de potencializar o processo de ensino-aprendizagem (JESUS *et al.*, 2007). Podem-se destacar também, além dos OAs, os tutores inteligentes que já são usados em diversas instituições para colaborar no processo de ensino.

O ensino de programação é considerado por professores e pesquisadores da área uma tarefa árdua e desafiadora. Caracterizada como uma disciplina de conceitos abstratos, e sendo um dos pilares do curso de computação, independente do nível ou modalidade, é, para alguns alunos, uma barreira intransponível. Reflexo dessa barreira são a alta taxa de reprovação nessa disciplina e, algumas vezes, o abandono do curso (SILVA *et al.*, 2010).

As dificuldades podem ser diagnosticadas não somente pela alta taxa de reprovação, mas também pelos problemas demonstrados por alunos em disciplinas avançadas, que exigem o pré-requisito de programação (ROCHA, 1991).

Segundo Rodrigues (2004), sendo uma disciplina com um dos maiores índices de reprovação no Brasil, torna-se ponto de reflexão por parte dos professores preocupados com a qualidade do processo de ensino e aprendizagem

dos alunos, o que confirma a necessidade de mudanças quanto à didática e às metodologias de ensino empregadas.

No curso de Sistemas de Informação da Universidade Estadual do Norte do Paraná – *Campus* Luiz Meneghel (UENP-CLM), as dificuldades supracitadas seguem presentes. Este trabalho pretende colaborar com o processo de ensino e aprendizagem de programação por meio do desenvolvimento de um OA abordando alguns dos assuntos da ementa da disciplina de Programação I.

1.2 Motivação e Justificativa

Mediante uma revisão bibliográfica, pode-se notar que outros trabalhos foram desenvolvidos para auxiliar a disciplina de programação (SILVA *et al.*,2010), possibilitando uma melhoria da qualidade do ensino e aprendizagem da matéria por meio da adoção de novas técnicas de aprendizado.

No trabalho de Silva *et al.* (2010), é proposta uma abordagem colaborativa, baseada em desafios, que visa complementar e apoiar o aprendizado de programação. Para isso, esforços têm sido empregados para o desenvolvimento de estratégias e ferramentas que contribuem para o aprendizado das disciplinas afins.

No trabalho de Jesus *et al.* (2007), é apresentada uma breve introdução sobre os problemas que a educação atual está passando e mostra como os objetos de aprendizagem podem auxiliar os professores e alunos. Ainda em Jesus *et al.* (2007), são expostas as vantagens de sua utilização e exemplos de objetos de aprendizagem já utilizados como uma ferramenta de auxílio pedagógico.

No trabalho de Saraiva e Neto (2010), é apresentado o Monitor: um conjunto de trinta e três objetos de aprendizagem desenvolvidos para apoiarem o processo de ensino e aprendizagem dos conteúdos de introdução à programação. Para o desenvolvimento do OA, foi realizada uma investigação com a presença de alunos e professores da área de Informática.

O que motivou a escolha do tema e a pesquisa sobre o assunto foi notar que outros trabalhos que adotaram novas técnicas de aprendizado colaboraram com o processo de ensino e aprendizagem, em diversas áreas da educação. Justifica-se também pela necessidade em diminuir o número de reprovação na disciplina de

programação I, principalmente nos períodos iniciais e, possivelmente, em reduzir a taxa de evasão nos cursos de Sistemas de Informação da UENP-CLM.

1.3 Objetivo Geral

O objetivo desse trabalho é desenvolver um objeto de aprendizagem para auxiliar o professor no ensino dos conceitos iniciais de programação, utilizando a metodologia GIED e o Software de Autoria *Visual Class*.

1.3.1 Objetivos Específicos

- Conhecer o perfil do aluno na fase inicial do curso de Sistemas de Informação;
 - Investigar as dificuldades dos alunos na aula de programação;
 - Saber o ponto de vista do professor em relação ao ensino e aprendizagem de programação;
 - Estudar as metodologias de desenvolvimento de objetos de aprendizagem;
 - Desenvolver um objeto de aprendizagem para a disciplina de programação I;
- e
- Validar o objeto de aprendizagem desenvolvido verificando sua aceitação pelos alunos.

1.4 Metodologia

As pesquisas realizadas para esse trabalho foram do tipo exploratório, utilizando o procedimento técnico do tipo levantamento. Foi escolhido esse modelo de pesquisa, pois permite uma maior familiaridade com o problema, com vista a torná-lo mais explícito.

Quanto ao procedimento técnico do tipo levantamento, foi escolhido, pois permite uma análise quantitativa e permite tirar conclusões sobre os dados coletados. Levando em conta as suas vantagens, como conhecimento direto da realidade, rapidez; pois permite coletar muitos dados em um curto espaço de tempo e ainda quantificá-los e organizar os dados em forma de gráficos e tabelas.

1.5 Organização do Trabalho

A estrutura do trabalho está organizada da seguinte forma: A seção 2 apresenta a fundamentação teórica; sendo que a subseção 2.1 trás as considerações iniciais; a subseção 2.2 fala sobre o papel da informática na educação; a subseção 2.3 comenta sobre o ensino de programação; a subseção 2.4 explica detalhadamente sobre objeto de aprendizagem; a subseção 2.5 cita algumas metodologias para desenvolvimento OA e a subseção 2.6 apresenta o software de autoria *Visual Class*.

Na seção 3, procede-se ao desenvolvimento, sendo que a subseção 3.1 apresenta a descrição da metodologia, enquanto a subseção 3.2 o levantamento dos dados feito com os alunos, na subseção 3.3 o levantamento dos dados feito com os professores e na subseção 3.4 as fases da metodologia adotada.

A seção 4 apresenta os resultados obtidos, a seção 5, as conclusões, enquanto a seção 6 propõe os trabalhos futuros. Por último, a seção 7 termina com as referências.

2 FUNDAMENTAÇÃO TEÓRICA

2.1. Considerações Iniciais

Esta sessão procura demonstrar uma visão geral sobre informática na educação, o ensino de programação, os conceitos de objetos de aprendizagem e algumas metodologias de desenvolvimento.

2.2 Informática na Educação

Segundo Valente (1999, p.11), “Informática na Educação refere-se à inserção do computador no processo de ensino-aprendizagem de conteúdos curriculares de todos os níveis e modalidades de Educação”.

Cada vez mais a informática vem adquirindo relevância no cenário educacional. Para Oliveira *et al.* (2010), “sua utilização como instrumento de aprendizagem e sua ação no meio social vêm aumentando de forma vertiginosa”, fazendo com que a educação passe por mudanças estruturais e funcionais diante desse novo instrumento tecnológico.

A implantação da informática, como auxiliar do processo de construção de conhecimento, implica mudanças na instituição, as quais vão além da formação do professor. É necessário que toda instituição, alunos, professores, administradores estejam preparados e suportem as mudanças educacionais imprescindíveis à formação de um novo profissional. Nesse sentido, a informática é um dos elementos que deverá fazer parte desta mudança (VALENTE, 1999, p.13; COLELLO, 2002).

A informática na educação trouxe diversos benefícios para o processo de ensino, colaborando, de forma eficaz, com *softwares* destinados à educação. “Esses produtos trouxeram a necessidade de aprimorar as metodologias de desenvolvimento de software, buscando as particularidades das aplicações educacionais” (OLIVEIRA *et al.*, 2010). Mesmo com os diversos *softwares* educacionais e as diversas metodologias de desenvolvimento, todos têm em comum o objetivo de colaborar com o processo de ensino-aprendizagem.

2.3 O Ensino de Programação

Para Silva *et al.*(2010), aprender programação ainda é uma barreira para muitos alunos e essa disciplina é essencial para os cursos na área de computação. Pelo fato de muitos alunos encontrarem dificuldades, principalmente na fase inicial dos cursos de computação, muitos acabam desistindo do curso.

Raabe e Silva (2005), analisando as dificuldades existentes nesse campo de ensino, as classificam em três grandes grupos: problemas de natureza didática; problemas de natureza cognitiva; e problemas de natureza afetiva.

2.3.1 Problemas de natureza didática:

1. Grande número de alunos: normalmente as turmas da disciplina de programação são compostas de 40 a 50 alunos e acabam dificultando ao professor fazer uma avaliação individual do aluno, limitando também a quantidade de avaliações aplicadas durante o semestre;

2. Dificuldade do professor em compreender a lógica do aluno: após desenvolver uma lógica para um determinado problema, fica difícil pensar em outras soluções, o que acaba por dificultar ao professor compreender a lógica do aluno;

3. Diferença de experiência e ritmo de aprendizagem entre os alunos: alguns alunos ingressam na disciplina sem nenhum conhecimento prévio, enquanto outros alunos já trabalham na área, ou possuem conhecimento de programação;

4. Ambiente de realização das provas: no momento da realização das provas, o aluno percebe a diferença entre observar e fazer. Alguns alunos têm a sensação de estar entendendo, mas acabam percebendo sua incapacidade em fazer. Aliado a isso, têm o *stress*, gerado pela pressão, pelo tempo limitado para realização da prova que, normalmente, desfavorece o desenvolvimento do raciocínio;

5. Pouco uso dos monitores da disciplina: os alunos com dificuldades de aprendizagem procuram muito pouco a ajuda dos monitores da disciplina, mesmo com a divulgação sistemática dos horários e locais de atendimento. Aparentemente os alunos depositam pouca credibilidade às orientações provenientes dos monitores;

6. Dificuldade para compreensão dos materiais: há diversos livros relacionados ao ensino de programação, mas normalmente possuem conteúdo que os alunos têm dificuldade em compreender; e

7. Alunos desorientados a escolha do curso: sendo uma disciplina ministrada logo no início do curso, muitos alunos ainda não têm uma compreensão correta sobre o perfil do curso e acabam por descobrir isso durante o decorrer da disciplina, gerando em muitos alunos uma visão equivocada em relação ao curso a qual cria uma incompreensão e descaso frente aos desafios da disciplina.

2.3.2 Problemas de natureza cognitiva:

1. Alunos sem perfil para solução de problemas: alguns alunos não desenvolveram, de forma adequada, as estratégias necessárias para a solução de problemas durante o ensino médio e acabam tendo maiores dificuldades para realizar os algoritmos propostos;

2. Alunos sem base operatório-formal¹: aparentemente o aluno não desenvolve, adequadamente, o raciocínio operatório formal, que é a base para compreensão do raciocínio lógico²; no entanto, faltam dados empíricos que comprovem essa realidade; e

3. Conteúdo sem proximidade com o conteúdo escolar: para maior parte dos alunos, a lógica algorítmica é algo totalmente novo e, com

¹ condições para a construção de um novo modo de organizar o pensamento.

² O raciocínio lógico é aquele que se desvincula das relações entre os objetos e procede da própria elaboração do indivíduo

isso, eles não conseguem estabelecer relações com conceitos já aprendidos anteriormente, principalmente com a matemática.

2.3.3 Problemas de natureza afetiva:

1. Ocasionais: problemas casuais, de ordem pessoal, que afetam o aluno, gerando uma dificuldade para que o mesmo consiga se concentrar nas explicações ou, ainda, influenciar em seu desempenho nas avaliações; e

2. Constantes: são problemas de ordem afetiva que se manifestam durante todo o decorrer da disciplina em maior ou menor grau: pouca motivação, aversão ao conteúdo ou ao professor, baixa auto-estima, insegurança, são exemplos de emoções que podem afetar, de forma negativa, a aprendizagem do aluno.

Problemas que envolvem o ritmo do aprendizado do aluno, material utilizado, perfil comportamental ou mesmo problemas pessoais e afetivos são apontados como relevantes.

Para Rocha *et al.* (2010), ensino e aprendizagem de programação têm sido foco de muitas pesquisas, nas quais se pode destacar uma ênfase significativa na questão das dificuldades discentes e no entendimento de conceitos relacionados a esse aprendizado.

Nota-se uma preocupação crescente com o processo de ensino e aprendizagem de programação. Tais estudos são motivados, sobretudo, pela importância dos conceitos de programação na vida acadêmica e nos cursos de computação. Vale ressaltar, enfaticamente, que a apropriação ou não dos conceitos iniciais de programação possui relação direta com o desempenho do aluno no decorrer de todo o curso, já que disciplinas avançadas dependem fortemente desses conceitos

2.4 Objetos de Aprendizagem

Segundo Wiley (2000), um OA podem ser compreendido como “qualquer recurso digital que possa ser reutilizado para o suporte ao ensino”. Este conceito prima por criar objetos que possam ser reutilizados por contextos diferentes. Não há limite para o seu tamanho, podendo um projetista criar um pequeno curso, ou até um curso altamente completo, que pode ser usado e reusado por diferentes pessoas de diferentes lugares.

Eles podem ser criados em qualquer mídia ou formato, podendo ser simples como uma animação ou uma apresentação de *slides*, ou complexos como uma simulação: utiliza-se de imagens, animações e *applets*, documentos VRML (realidade virtual), arquivos de texto ou hipertexto, dentre outros. Não há um limite de tamanho para um OA. Porém, existe o consenso de que ele deve ter um propósito educacional definido, um elemento que estimule a reflexão do estudante e que sua aplicação não se restrinja a um único contexto (BETTIO; MARTINS, 2004).

No site da Rede Interativa Virtual de Educação (RIVED, 2011), encontra-se a seguinte definição para Objeto de Aprendizagem: “qualquer recurso que possa ser reutilizado para dar suporte ao aprendizado”. Já, para o *Institute of Electrical and Electronics Engineers* (IEEE, 2000), a definição do termo Objetos de Aprendizagem (*Learning Objects*) é mais amplo. Segundo o instituto, pode ser considerado um OA: “Qualquer entidade, digital ou não digital, que possa ser usada, reutilizada ou referenciada durante o uso de tecnologias que suportem ensino”.

Outra definição apresentada por Beck (2001) afirma que objetos de aprendizagem representam qualquer recurso digital que possa ser reutilizado para suporte ao ensino. Diz, ainda, que a principal intenção do OA é de possibilitar a divisão dos conteúdos educacionais em pequenos pedaços que possam ser reutilizados em diferentes ambientes de aprendizagem.

Para Muzio *et al.* (2001), objetos de aprendizagem são partições reutilizáveis independente da mídia utilizada. Já Oliveira *et al.* (2010), define como uma aplicação educacional com a principal característica: a possibilidade de reutilização de seus recursos em diferentes contextos, construídos para servir como uma ferramenta de apoio para professores em ambientes divergentes.

Behar (2009) diz que os objetos de aprendizagem são como recursos autônomos, utilizados como módulos de um determinado conteúdo e que podem ser incorporados a múltiplos aplicativos e adaptados a diversos contextos, considerando como objetos de aprendizagem quaisquer materiais eletrônicos (como imagens, vídeos, páginas *web*, animações ou simulações), desde que tragam informações destinadas à construção do conhecimento e guardem a característica de possibilidade de reutilização por meio da padronização.

Contudo, o Ministério da Educação (MEC, 2011) orienta que os OAs devem objetivar: o aprimoramento da educação presencial e/ou a distância, para incentivar a pesquisa e a construção de novos conhecimentos para melhoria da qualidade, equidade e eficiência dos sistemas públicos de ensino pela incorporação didática das novas tecnologias de informação e comunicação.

Outras características desejáveis aos objetos de aprendizagem são a acessibilidade, interoperabilidade, durabilidade e customização. Acessibilidade é a possibilidade de acesso remoto ao objeto, enquanto interoperabilidade se refere a sua articulação em diferentes plataformas e agregado a diferentes ferramentas. A característica de durabilidade é relativa à vida útil do material em termos tecnológicos, o que tem relação com a possibilidade de customização ou a flexibilidade do material em termos de adaptação para novas aplicações (TAROUÇO *et al.*, 2003; BETTIO; MARTINS, 2004).

Pode-se utilizar um OA para diferentes fins. Um fator a ser levado em consideração é o auxílio fundamental que fornecem aos crescentes AVEAs – Ambientes Virtuais de Ensino e Aprendizagem (POLSANI, 2003).

Uma das aplicabilidades propostas por Downes (2001) foi a de integração de diferentes instituições que oferecem, na modalidade a distância, o mesmo conteúdo, para usuários distintos e em diferentes casos de aplicação. O autor afirma que elaborar conteúdos e atividades instrucionais demanda custo financeiro, e os OAs podem ajudar a diminuir substancialmente estes investimentos.

Os objetos de aprendizagem têm se tornado uma importante ferramenta para aprimorar o ensino interativo de temas didáticos das mais diversas disciplinas. Parte-se de exemplos práticos para despertar a curiosidade dos estudantes, possibilitando ao professor ensinar com mais eficiência e os alunos aprenderem brincando (JESUS *et al.*, 2007).

De acordo com Colello (2002), o problema na aprendizagem remete também a outras dimensões dentro e fora da escola, em que se propõe a instituição uma mudança de concepção, postura e dinâmica de trabalho na busca da qualidade no ensino.

Segundo Wiley (2009), as principais características de um OA que justificam seu desenvolvimento são: interoperabilidade, reusabilidade, gerenciabilidade, granularidade, acessibilidade, durabilidade, as quais são enumeradas abaixo (PASSARINI, 2003), (WILEY, 2009):

- a) Interoperabilidade:** possibilidade de utilização em diferentes plataformas, ou seja, o mesmo objeto pode ser reutilizado em diferentes sistemas operacionais. Vale ressaltar que a infraestrutura básica do OA deve ser projetada para ser interoperável;
- b) Reusabilidade e Acessibilidade:** capacidade do objeto de ser utilizado e acessado em múltiplas aplicações, cursos e contextos que tenham similaridade. A descrição e catalogação dos conteúdos do OA são realizadas por meio de metadados - dados estruturados sobre dados -, portanto pode ser buscado, acessado e utilizado em disciplinas diferentes;
- c) Gerenciabilidade:** possibilidade de atualizar o objeto de aprendizagem no AVEA, conforme o objeto é atualizado na base. Portanto se mais de um ambiente de ensino estiver utilizando o mesmo objeto, estarão sempre atualizados, pois, ao ser alterado, um objeto no repositório de OAs, atualizará o objeto de aprendizagem utilizado nos diversos ambientes;
- d) Granularidade:** definição do tamanho ideal de um Objeto de Aprendizagem. Portanto, o OA deve conter um tamanho que apresente todo o conteúdo necessário para o ensino, mas também não pode ser muito grande, pois dificulta a assimilação do conteúdo; e
- e) Durabilidade:** garantia de reutilização caso mude a tecnologia de base, sem necessidade de reestruturação do OA. Dessa forma, o usuário tem a possibilidade de mudar seu sistema sem precisar alterar o objeto, devido ao fato de os objetos de aprendizagem possuírem um padrão neutro que se adapta a qualquer ambiente de ensino.

2.5 Metodologias de Desenvolvimento de OAs

Para o desenvolvimento de um software bem estruturado, exige-se uma metodologia de trabalho. Segundo Lucena *et al.* (2009), “esta metodologia deve ter fundamentos que forneçam a base para a construção de um produto de software de maneira eficaz”. Permite-se, também, um processo mais organizado, com maior controle, garantindo uma variabilidade mínima na criação do *software*, dentro dos níveis aceitáveis (LUCENA, 2009).

Várias metodologias de desenvolvimento de OAs podem ser encontradas na literatura. Nas seções a seguir, serão apresentadas algumas delas.

- **ADDIE:** A metodologia ADDIE utiliza cinco fases da engenharia de *software* para o desenvolvimento de OAs: análise, *design*, desenvolvimento, implementação e avaliação (MUSTARO, 2007).
- **LabVirt:** O Laboratório Didático Virtual (LabVirt) é uma ação da Universidade de São Paulo (USP), a fim de construir uma infra-estrutura pedagógica que dê suporte ao desenvolvimento de projetos das disciplinas de Física e Química (MUSTARO, 2007). Para isso, conta com a construção colaborativa de OAs por alunos e professores de escolas públicas e universidades, assim como pesquisadores. A metodologia aplicada no LabVirt se dá em quatro etapas (NUNES et al. ,2006):
 - Elaboração do roteiro de simulação;
 - Revisão e retorno do roteiro;
 - Produção; e
 - Apresentação.
- **Sophia:** O Sophia é um ROA da UNIVALI - Universidade do Vale do Itajaí, cujo foco consiste em dar suporte aos alunos do curso de Tecnologia em Análise e Desenvolvimento de Sistemas, apoiando os alunos quanto à realização de atividades e com conteúdo de apoio às disciplinas, podendo ser acessado em: <http://siaiacad17.univali.br/sophia>.

Segundo Pessoa e Benitti (2008), o processo Sophia de produção é composto por três etapas distintas:

- Projeto;

- Desenvolvimento; e
 - Distribuição;
- **RIVED:** RIVED é um programa da Secretaria de Educação a Distância (SEED) que tem por objetivo a produção de materiais pedagógicos digitais (RIVED, 2011). O processo de desenvolvimento da RIVED para objetos de aprendizagem é dividido em seis fases.

2.5.1 GIED

O GIED (Grupo de Informática Educativa) tem como objetivo realizar pesquisas na área de Informática na Educação. A atividade primordial é prover mecanismos baseados em Tecnologia da Informação e Comunicação para apoiar o processo de ensino-aprendizagem dos conteúdos curriculares de Escolas do Ensino Fundamental e a Inclusão Digital de Cidadãos da Região do Município de Bandeirantes.

O Grupo é formado por graduandos, graduados e professores do curso de Licenciatura em Sistemas de Informação da UENP – CLM, que, mediante pesquisas a diversas metodologias, criaram a sua própria metodologia de desenvolvimento de OA. A GIED é composta pelas seguintes fases: análise, projeto, implementação e validação (OLIVEIRA *et al.*, 2010). As fases são mostradas na figura 1.



Figura 2.1 - Fluxo da metodologia de desenvolvimento de OA

1. **Análise:** esta etapa contempla os aspectos relativos às atividades que antecedem os processos de modelagem e implementação. Neste ponto do desenvolvimento dos objetos de aprendizagem, é importante conhecer o público-alvo e as estratégias instrucionais, bem como o conteúdo.
 - a) Conteúdo Instrucional: seleção e organização dos conteúdos e da proposta pedagógica, evidenciando a boa prática na aplicação do objeto; e
 - b) Delimitação do público-alvo e seus pré-requisitos: análise dos conhecimentos exigidos para o usuário utilizar o OA.
2. **Projeto:** a segunda etapa constitui-se do desenvolvimento dos modelos que apoiarão a implementação.
 - a) Mapa conceitual: definição de mapas conceituais, base para o processo de desenvolvimento do objeto educacional, demarcando os conceitos estruturais do conteúdo;
 - b) Mapa navegacional: desenvolvimento do mapa de navegação apresentando a forma como será realizada a navegação entre as paginas do OA; e
 - c) *Storyboard*: elaboração do *layout* e orientações, roteiro a ser seguido na implementação.
3. **Implementação:** com base nos *storyboards*, a equipe de desenvolvimento assume essa etapa. A implementação do *software* educacional é concretizado com base em critérios de usabilidade e acessibilidade, procurando gerar produtos adequados ao público-alvo.
4. **Validação:** a etapa de validação é centrada na avaliação do objeto de aprendizagem por parte de uma parcela do público-alvo. Posteriormente à avaliação, são desenvolvidos os manuais do usuário final e do professor.
 - a) Avaliação: última fase, direcionada aos testes do produto, com o propósito de validar o funcionamento (se está adequado aos usuários e se cumpre os objetivos visados); e
 - b) Manuais: elaboração de todas as documentações necessárias para uma melhor utilização dos objetos desenvolvidos.

Vale ressaltar que esta metodologia proposta segue uma abordagem evolutiva. Desde a elaboração da primeira à última etapa, normalmente, surgem

alguns aspectos que necessitam de reformulação ou melhorias. Estas alterações devem ser incorporadas ao ciclo de vida do desenvolvimento, de modo que se estabeleça, assim, uma inter-relação circular entre as etapas distintas do ciclo de concepção e desenvolvimento de um OA (OLIVEIRA *et al.*, 2010).

2.6 Software de autoria *Visual Class*

Segundo Herculiani (2007, p. 40), “no meio educacional, um software de autoria é uma ferramenta ou programa de desenvolvimento que fornece elementos préprogramados que permitem desenvolver aplicações multimídia”. Enquanto Barcarolo (2007, p.06) define que softwares de autoria “são programas que permitem aos usuários a criação de seus próprios trabalhos para publicação ou aplicação em ambientes multimídia, internet e outros”.

Em 1995 a Faculdade de Engenharia de Minas da Escola Politécnica da USP iniciou projeto visando criar ferramentas multimídia para auxiliar professores, optando assim pelo desenvolvimento de um software com a seguinte característica: facilidade de uso, dispensando a necessidade de conhecimentos sobre programação. Este *software* então foi criado por um desenvolvedor chamado Celso Tatizana. A primeira versão denominou-se *Visual Class 1.0* (KASIM e SILVA, p 12373).

De acordo com Tatizana (2006, p. 14), o *Visual class*³ um *Software* de Autoria para criação de Projetos Multimídia, como Aulas, Palestras, TBC (Treinamento Baseado em Computador), Livros Eletrônicos, Quiosques Multimídia, Catálogos, CDs Institucionais, Cursos de Ensino a Distância e *Sites* na *Internet*.

O grande diferencial do *Visual Class* em relação aos *Softwares* de Autoria tradicionais é a sua facilidade de uso. Com o *Visual Class* é possível criar sofisticadas aplicações multimídia, por usuários não especializados em informática. A sua interface orientada a objetos elimina a necessidade de linguagem de programação, tornando o processo de criação intuitivo e acessível até mesmo a crianças com 7 anos de idade.

O *Visual Class* foi desenvolvido em 1996 pela Caltech Informática Ltda (www.class.com.br) e, atualmente, conta com mais de 1.000.000 de usuários no

³ http://www.classinformatica.com.br/3desc_1_1.htm

Brasil e vários em Angola, Peru, Argentina, Polônia, Japão e Estados Unidos, ocupando a liderança nacional no segmento de autoria educacional. Na área educacional, o *Visual Class* pode ser utilizado na perspectiva tradicional, em que o professor cria as aulas e as atribui aos alunos; ou na forma construtivista, na qual o aluno cria os conteúdos e o professor orienta e avalia o processo de criação.

Neste segmento, o *Visual Class* vem sendo utilizado em Universidades, Escolas do Ensino Regular Público e Privado (Infantil, Fundamental e Médio), Escolas Técnicas, Prefeituras, Núcleos de Tecnologia Educacional. Algumas Redes de Ensino, como Rede Pitágoras, Rede Marista, Vicentina e Ciesc, fornecem o *Visual Class* para suas unidades de ensino. Na área empresarial, o *Visual Class* vem sendo utilizado para treinamento e *marketing*.

Em 1998 o *Visual Class* foi selecionado a representar o Brasil no encontro binacional Brasil-Estados Unidos em Tecnologia Educacional, realizado nos Estados Unidos, integrando a comitiva do Ministério da Educação. Neste evento o Estado de Ohio se interessou pelo *software* e solicitou a tradução para o inglês.

A versão lançada em maio de 1999 já está traduzida para o inglês e foi lançada juntamente com o Livro do *Visual Class* publicado pela Editora Érica. Na feira Educar 2002, foi lançada a versão do *Visual Class* 2002 e o livro *Visual Class* 2002. Esta versão marcou a mudança da plataforma de 16 *bits* para 32 *bits*, com um conjunto de implementações superiores a todas as versões anteriores.

Na Feira Educar 2003, foi lançada a nova versão 2003 e o novo livro com implementações que otimizaram o desempenho e estabilidade do sistema. No ano de 2004, Celso Tatizana resolveu criar um evento especializado em Tecnologia Educacional, denominado *Tec Education* em parceria com a empresa Humus Consultoria. Neste evento foi lançada a nova versão do *Visual Class* 2004 e o Concurso Nacional Multimídia *Visual Class* para alunos da quarta série do ensino público municipal.

Em 2006 foi lançado o *Visual Class* FX, com uma maior integração com o *Flash*. No final de 2009, foi lançada a nova versão *Visual Class* FX SE, compatível com o *Windows* 7. Em 2009 o *Visual Class* foi selecionado pelo MEC (Ministério da Educação) para constar do Guia de Tecnologias Educacionais, publicado no portal do MEC (página 69):

http://portal.mec.gov.br/dmdocuments/guia_tecnologias_educacionais_2008_2.pdf

O *Visual Class* FX SE grava os projetos em formato próprio ou em formato executável com propaganda do software ou em formato PDF. Opcionalmente, o usuário pode adquirir o GeraHTML FX SE para converter em formato HTML/Javascript (publicação na WEB, compatível com "*browser*" *Internet Explorer* e *Mozilla/Firefox*. Ou adquirir o GeraCD FX SE para verter em formato executável sem propaganda do sistema(distribuição em CD). O GeraCD FX SE permite também gerar pequenos projetos(até 16 telas) em formato SWF (*Flash MX*) ou em um único arquivo executável, com algumas limitações de recursos.

3 DESENVOLVIMENTO

3.1 Metodologia

Todas as pesquisas realizadas com alunos e professores para esse trabalho foram do tipo exploratório, utilizando o procedimento técnico do tipo levantamento. Os questionários utilizados foram anônimos, para que os entrevistados tivessem total liberdade para responder às questões.

Para conhecer o perfil dos alunos do primeiro período do curso de Sistemas de Informação da UENP-CLM, foi realizada uma pesquisa com 69 alunos do respectivo curso, por meio de um questionário aos alunos (APÊNDICE A). Para o questionário, foi utilizado o *Google docs*⁴ ferramenta que permite a criação e compartilhamento de documentos, folhas de cálculos e apresentação *online*. O questionário é composto por 14 questões objetivas de múltipla escolha e foi aplicado em datas diferentes. No segundo semestre de 2011 foi aplicado para 38 alunos e no segundo semestre de 2012 foi aplicado para 31 alunos. Os alunos que participaram da pesquisa estão no período inicial do curso de Sistemas e as turmas são divididas em 2 salas com professores diferentes, sendo que, na UENP-CLM, o referido curso é oferecido apenas no período noturno.

Foi realizada também uma pesquisa com 4 professores, para saber o ponto de vista deles em relação ao ensino e aprendizagem de programação. Para eles, foi elaborado um questionário impresso contendo 6 questões do tipo dissertativas (APÊNDICE B). Dos professores que participaram da pesquisa, todos lecionam ou já lecionaram a disciplina supracitada.

O objeto desenvolvido foi aplicado para 29 alunos do primeiro ano do curso de Sistemas de Informação da UENP-CLM, para validação do AO.

Para o desenvolvimento do OA, foi escolhida a metodologia GIED, tendo em vista que a mesma já foi utilizada para o desenvolvimento de diversos OAs. Os objetos criados foram aplicados em várias escolas do ensino médio e também em projetos de inclusão digital na cidade de Bandeirante – PR, obtendo ótimos resultados.

⁴ Link: <http://docs.google.com>

No trabalho de Oliveira (2011, p.47), foi desenvolvido um objeto para o ensino da língua Portuguesa utilizando a metodologia GIED, sendo aplicado para 25 alunos do segundo ano do ensino médio do Colégio Integrado de Bandeirantes, no município de Bandeirantes - PR. Os alunos foram avaliados antes e depois de aplicado o objeto de aprendizagem, sendo que antes da aplicação, apenas 4 alunos tiveram notas entre 8,0 e 10,0; entretanto após aplicado o OA, 15 alunos tiveram notas entre 8,0 e 10,0.

Além do sucesso que a metodologia GIED teve com seus objetos desenvolvidos (VASCO *et al.*,2011), outro fator importante que levou a sua escolha é que ela foi criada a partir de estudos de outras metodologias, as quais foram citadas em Oliveira *et al.*(2010). Utilizar essa metodologia nesse trabalho foi encarado como um desafio em virtude da diferença do público-alvo.

O *software* utilizado para o desenvolvimento foi o *Visual Class*. A escolha deve-se ao seu sucesso como ferramenta de ensino, sendo, hoje, o *software* de autoria mais usado no país.

3.2 Levantamento dos dados feito com os Alunos

Para visualizar os dados levantados com a pesquisa realizada com os alunos, foram utilizados gráficos, pois permitem uma melhor compreensão do tema. Após cada gráfico, será apresentado um pequeno comentário.

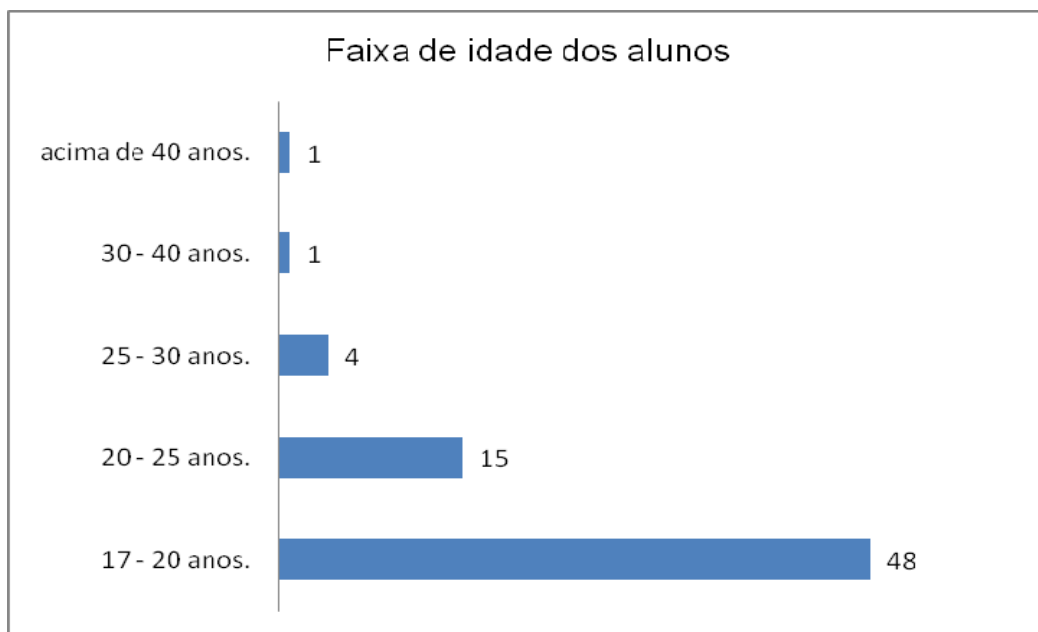


Gráfico 3.1 – Faixa de idade dos alunos

Observando o Gráfico 1, nota-se que a maior parte dos alunos, que corresponde a 48 deles, tem entre 17 e 20 anos. Mostra também que a maioria dos alunos ingressam no curso de Sistemas assim que terminam o ensino médio.

Outro dado relevante é que a soma dos alunos com idade entre 17 e 20 anos com os de 20 e 25 totalizam 63 alunos, o que revela um curso com a maior parte dos alunos com idade abaixo de 25 anos.

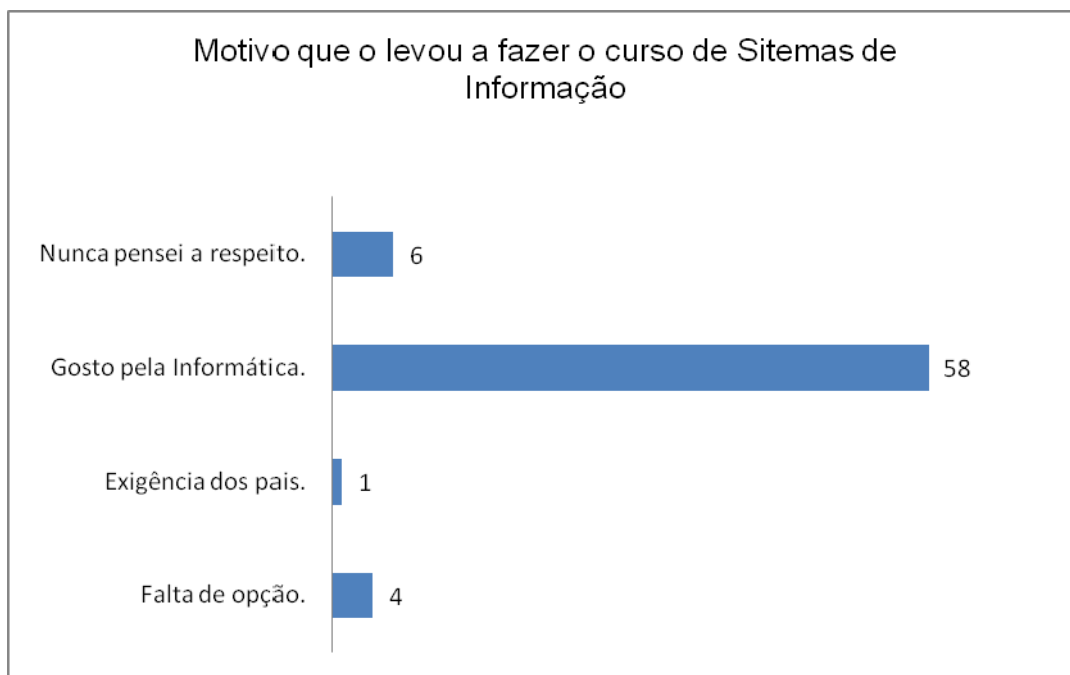


Gráfico 3.2 – Motivo que o levou a fazer o curso de Sistemas de Informação

O Gráfico 2, mostra que a grande parte dos alunos iniciou o curso de Sistemas de Informação motivados pelo gosto pela informática.

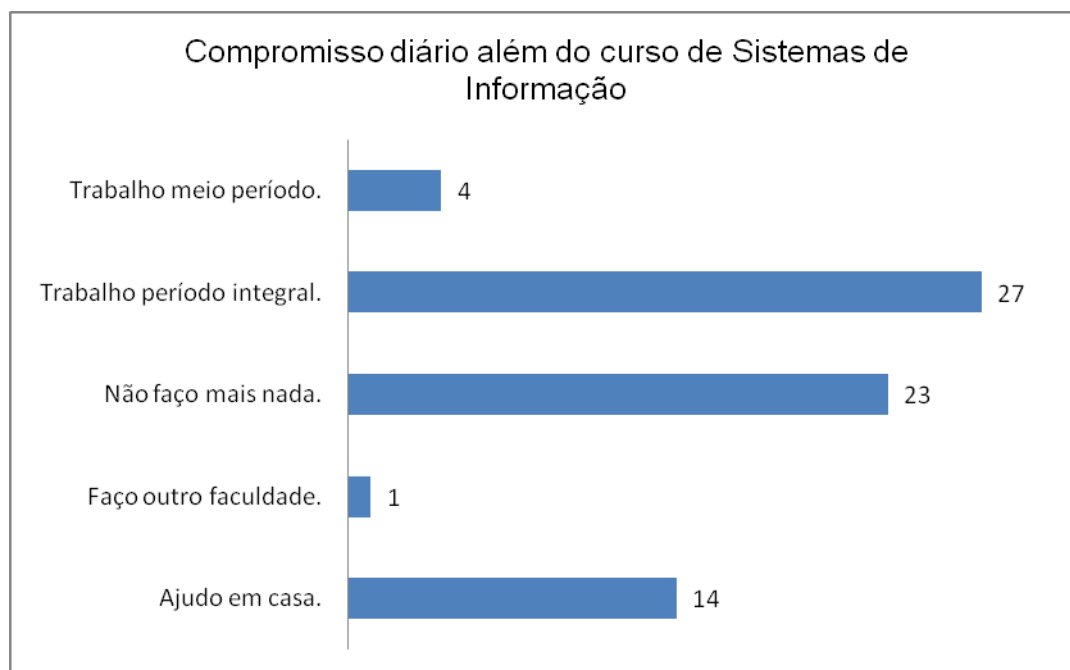


Gráfico 3.3 – Compromisso diário além do Curso

Analisando o Gráfico3.3, observa-se que 27 alunos trabalham em tempo integral, enquanto 23 deles não têm nenhuma outra atividade fora da sala de aula. Tal fato revela, uma grande diferença entre os alunos em relação ao tempo que eles têm para se dedicar aos estudos.

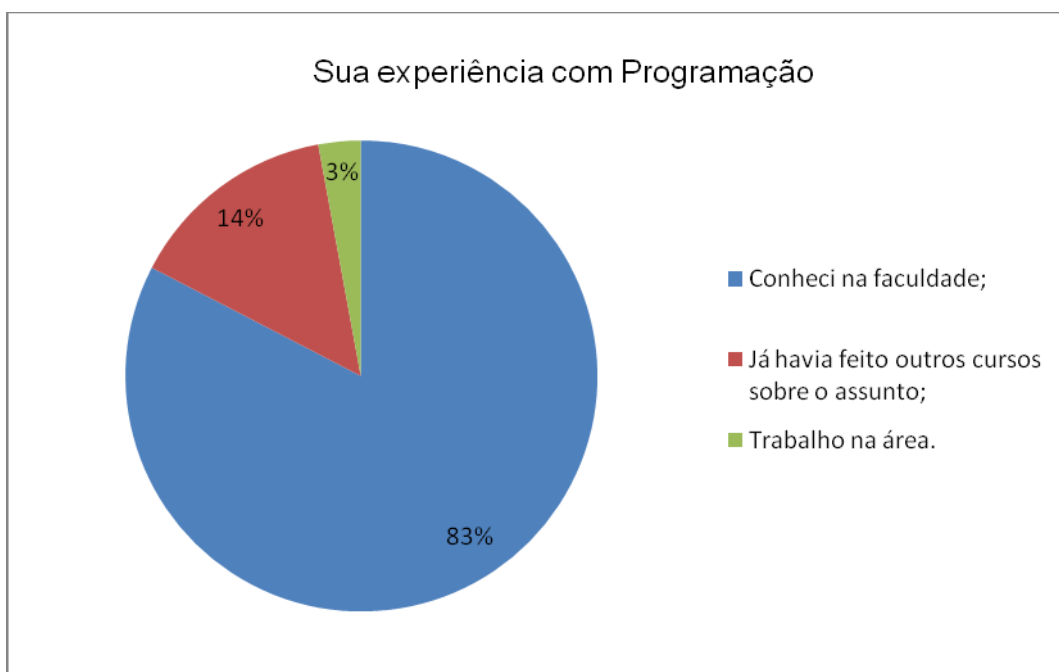


Gráfico 3.4 – Experiência com Programação

Por meio do Gráfico 3.4, pode-se ver que os alunos que iniciaram no curso de Sistemas de Informação, não possuem conhecimento em programação, pois 83% deles conheceram programação apenas na faculdade. Apenas 14% deles já haviam feito cursos sobre o assunto.

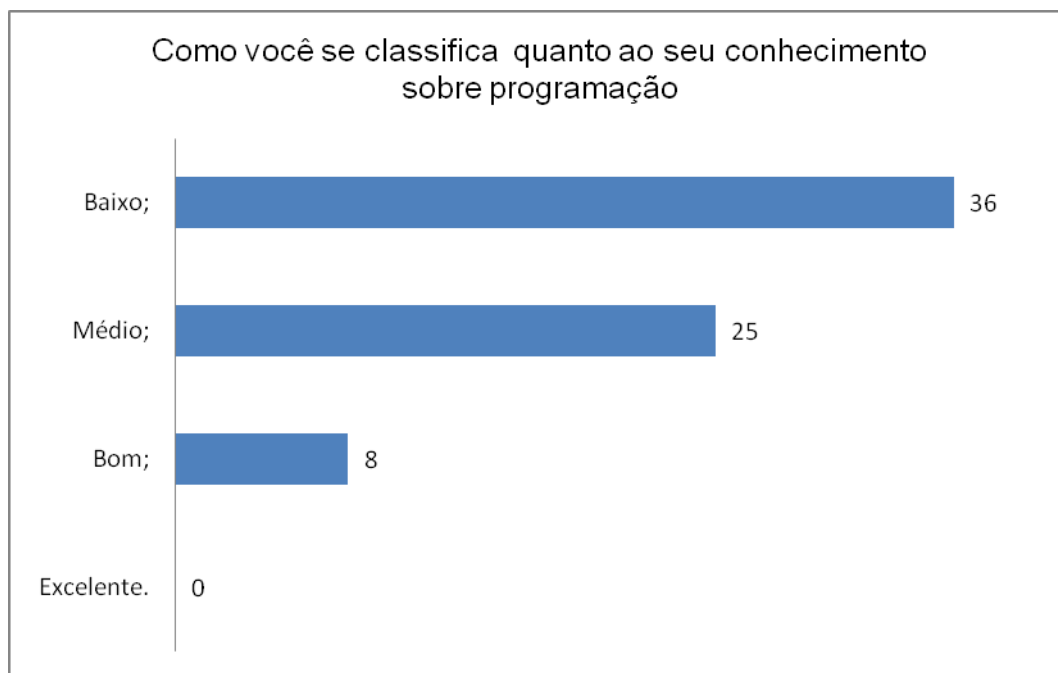


Gráfico 3.5 – Seu conhecimento sobre programação

O Gráfico 3.5 mostra que 36 alunos consideram seu conhecimento sobre programação baixo e 25 o consideram médio. Somados os 2 resultados, observa-se que 61 dos alunos consideram que seu conhecimento está entre baixo e médio.

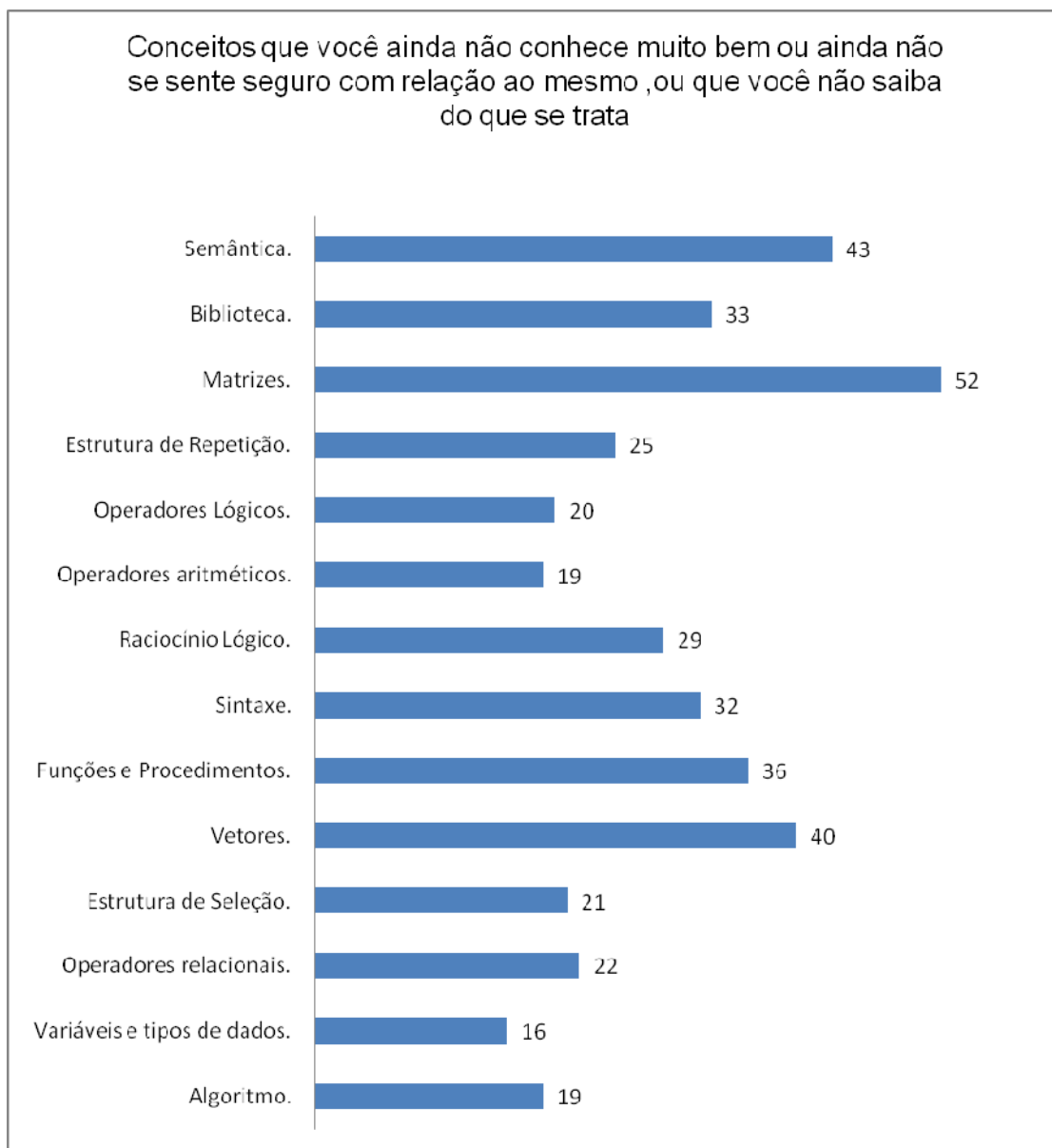


Gráfico 3.6 – Conceitos que o aluno ainda não conhece ou com que se sente inseguro.

Por meio dos resultados do Gráfico 3.6, observa-se, de um modo geral, que vários alunos possuem dificuldades ou não se sentem seguros em relação a conceitos iniciais da disciplina de programação I. Observa-se, também, que dos 69 alunos, 19 ainda não se sentem seguros com relação aos conceitos de algoritmo, 16 deles responderam à variáveis e tipos de dados e 22 disseram ter dificuldades com operadores relacionais.

Vale ressaltar que muito dos conceitos não haviam sido aplicados na disciplina, mas nota-se que alguns alunos já conheciam ou, pelo menos, ouviram falar de conceitos mais avançados de programação.

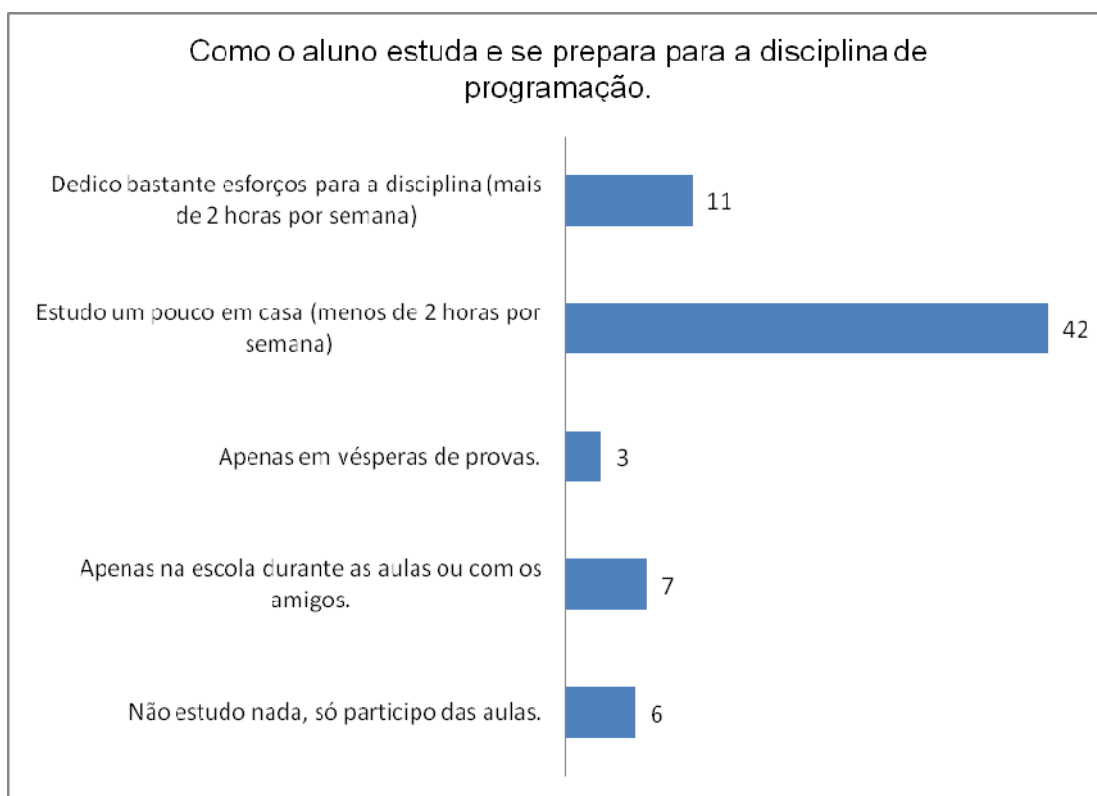


Gráfico 3.7 – Como o aluno se prepara para a disciplina de programação

Como pode ser observado no Gráfico 3.7, com relação ao tempo de estudo do aluno dedicado à disciplina de programação, nota-se que 42 deles estudam em casa menos de 2 horas por semana e 6 deles não estudam nada fora da sala, somente durante as aulas.

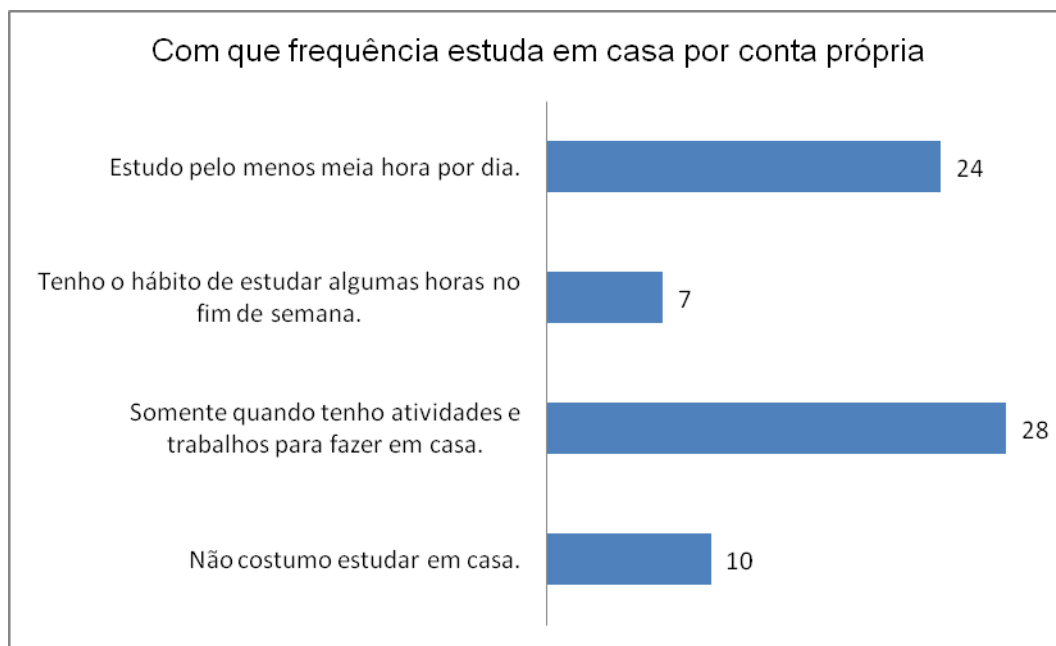


Gráfico 3.8 – Frequência com que o aluno estuda em casa por conta própria

Analisando o Gráfico 3.8, observa-se que 28 dos alunos de Sistemas de Informação estudam somente quando têm atividades e trabalhos para fazer em casa, outros 24 estudam menos de meia hora por dia. Já outros 10 não têm o costume de estudar em casa e apenas 7 deles estudam algumas horas no fim de semana.

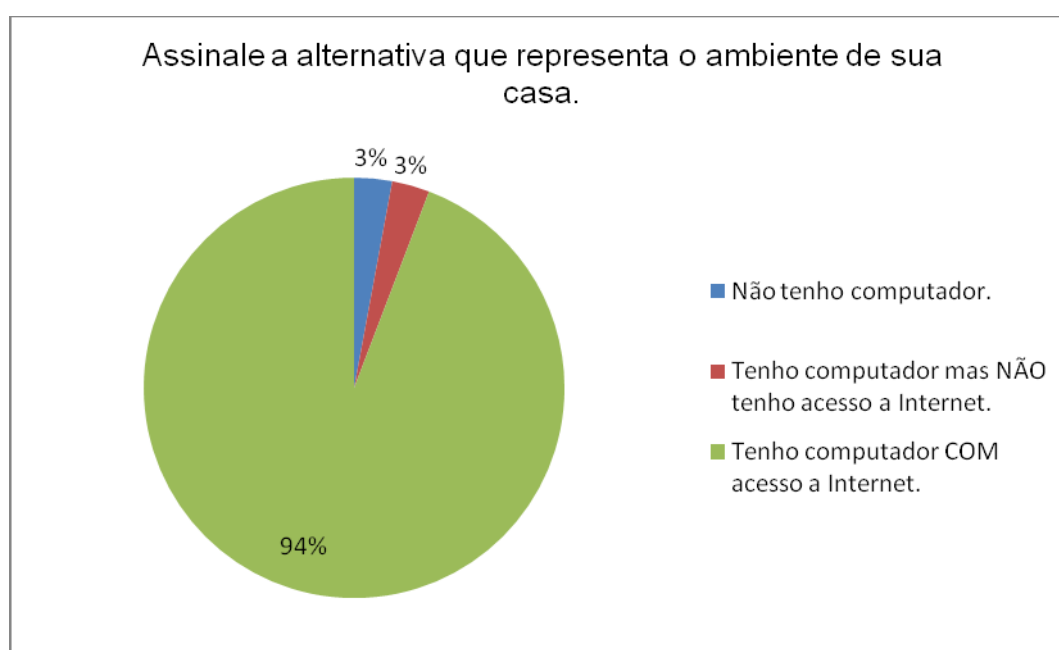


Gráfico 3.9 – Ambiente da sua casa com relação ao computador

Por meio do Gráfico 3.9, pode-se observar que quase todos os alunos possuem computador em casa com acesso à *internet*, totalizando 94%. Revelando que a maioria deles tem a opção de usar da rede para pesquisar ou estudar assuntos relativos ao curso.

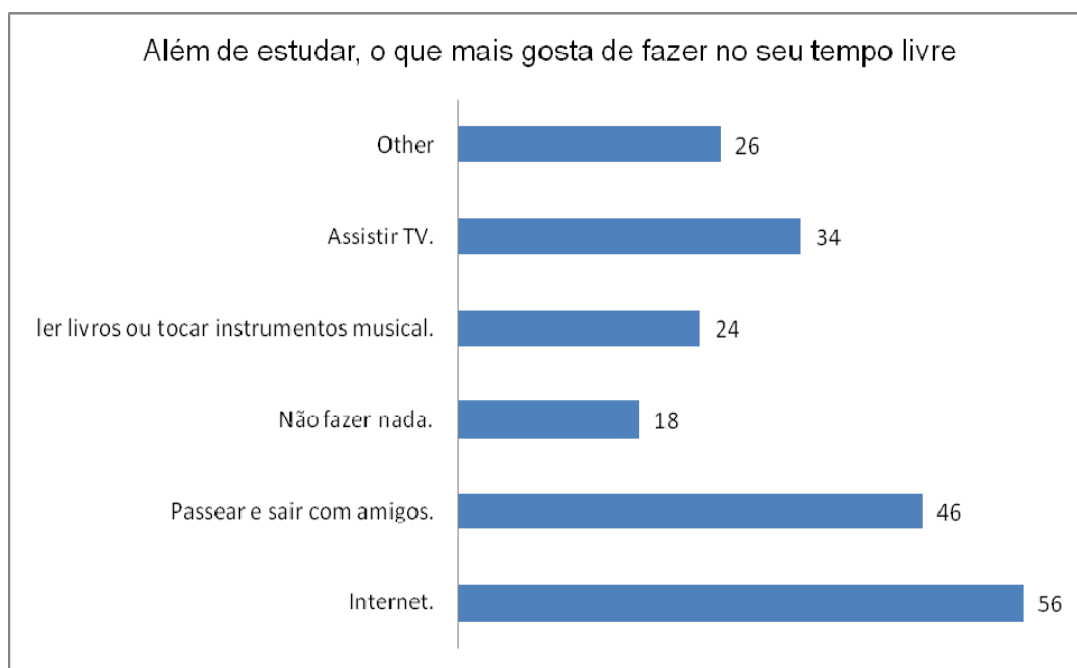


Gráfico 3.10 – Além do estudo, o que mais gosta de fazer no tempo livre

O gráfico 3.10 mostra que 56 dos alunos entrevistados gostam de ficar na *internet* no tempo livre. Mas, que a opção de passear com os amigos também teve uma escolha relevante, sendo escolhida por 46 alunos.

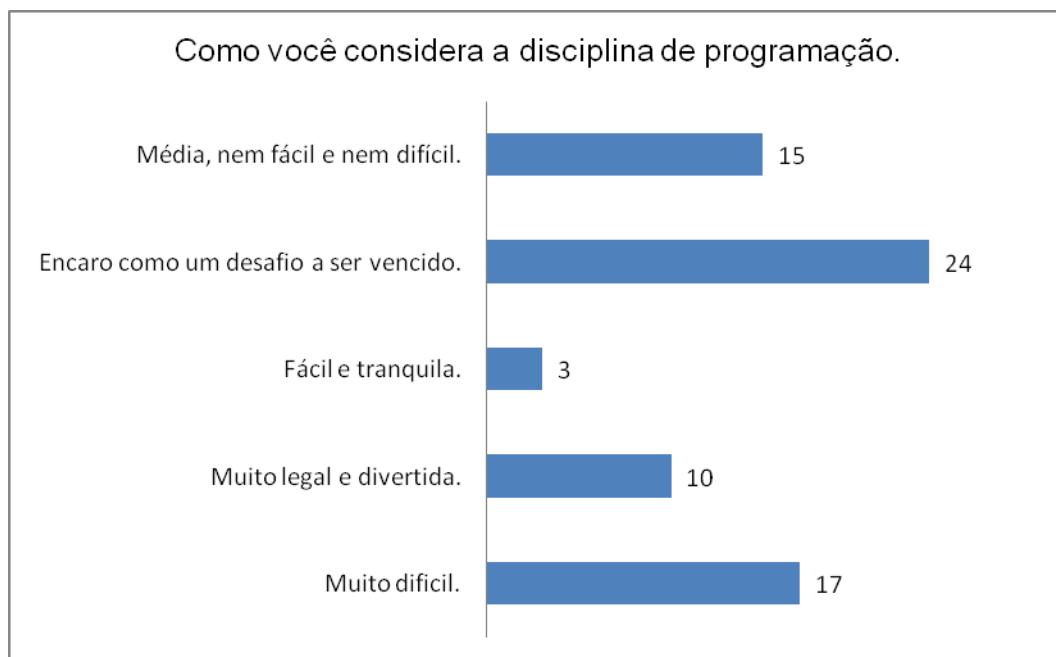


Gráfico 3.11 – Como você considera a disciplina de programação

Analisando o Gráfico 3.11, observa-se que 24 dos alunos entrevistados disseram encarar a disciplina de programação como um desafio a ser vencido, sendo a resposta da maioria. Outros 17 alunos a consideram uma disciplina muito difícil, 15 deles a encaram como média, nem fácil e nem difícil. Somente 3 alunos disseram que consideram a disciplina fácil e tranquila.

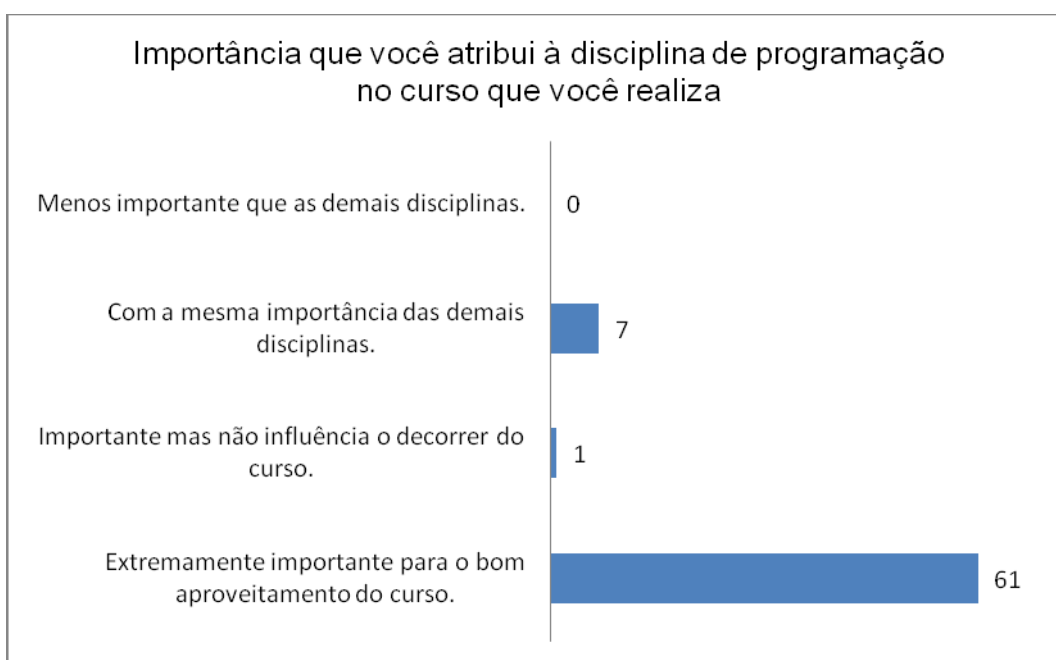


Gráfico 3.12 – Importância que o aluno atribui à disciplina de programação

Observando o Gráfico 3.12, quanto à importância que o aluno atribui à disciplina de programação, 61 alunos a consideram extremamente importante para o bom aproveitamento do curso; 7 dos alunos a consideram com a mesma importância das demais disciplinas e apenas 1 deles a considera importante, mas não influencia o decorrer do curso.

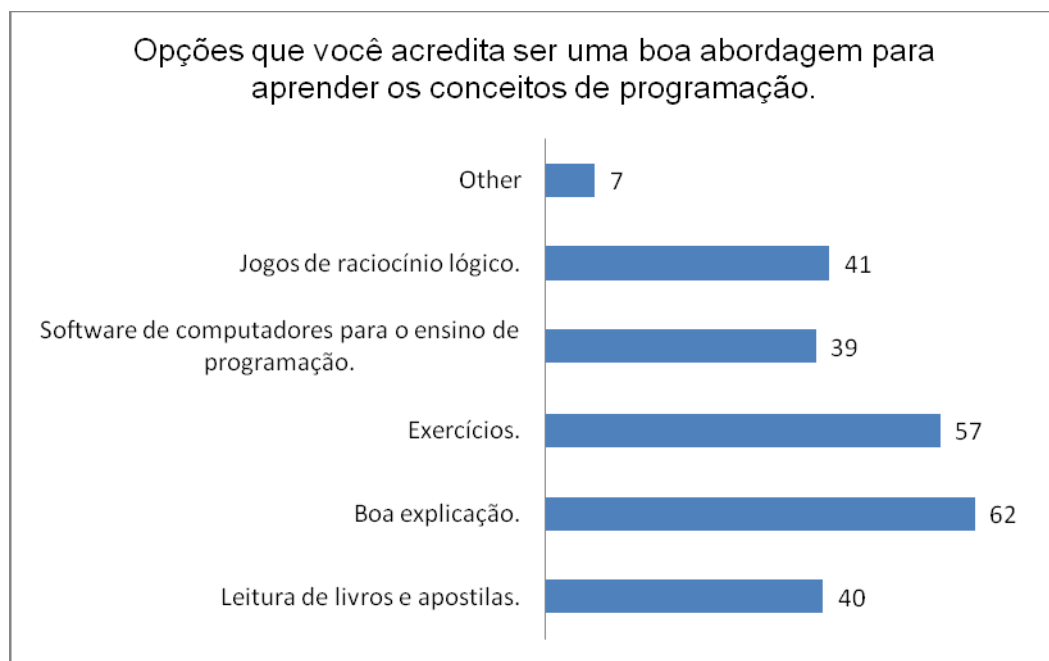


Gráfico 3.13 – Uma boa abordagem para aprender os conceitos de programação

Os alunos acreditam que uma boa explicação é uma melhor abordagem para aprender os conceitos de programação, como pode ser analisado pelo gráfico 3.13, visto que teve a escolha de 62 alunos. Mas também os exercícios são uma boa abordagem, ficando com 57 escolhas, ficando em seguida jogos de raciocínio, leitura de livros e apostilas, *software* de computadores para o ensino de programação com menos escolhas.

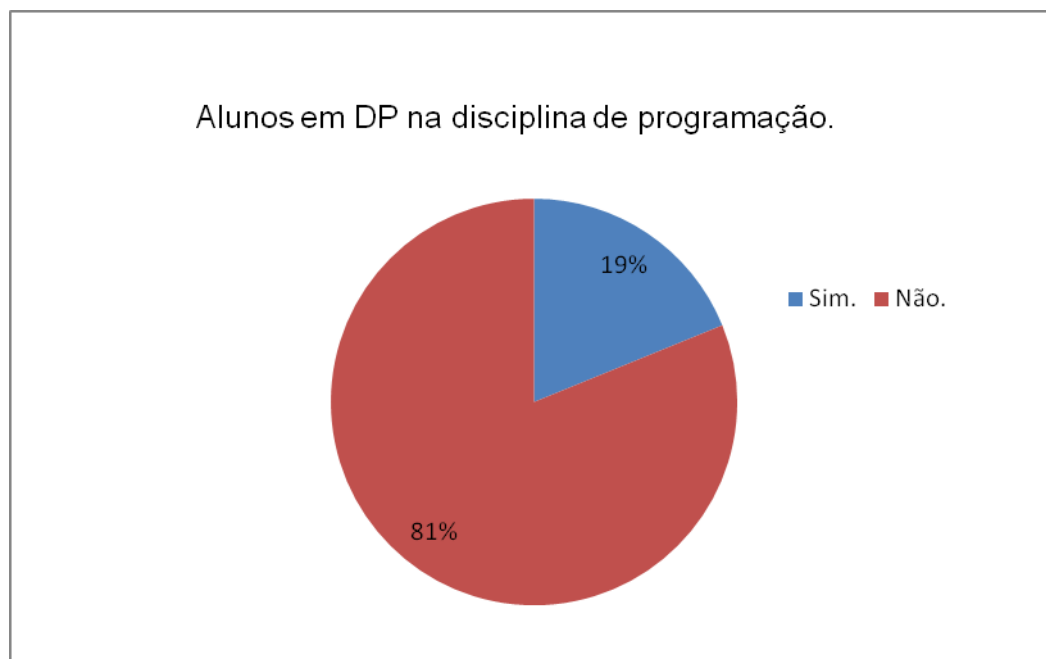


Gráfico 3.14 – Dependência na disciplina de programação

Como pode ser observado no gráfico 3.14, dos alunos que foram entrevistados, 19% deles estão em DP na disciplina de programação I.

3.3 Levantamento dos dados feito com os Professores

Em relação ao perfil dos alunos ingressantes no curso de Sistemas de Informação, os professores responderam que não há como traçar um perfil ou generalizar. Há casos de alunos que já possuem conhecimento na área, como existem aqueles que têm dificuldade em computação básica, em utilizar mouse e teclado. Assinalam ainda que o aluno está vindo do ensino médio com uma defasagem nos conceitos matemáticos mais básicos, e por isso tem dificuldade nas disciplinas iniciais no curso. Também falaram que por motivo do vestibular ser pouco concorrido, muitos entram sem a noção correta do que realmente é o curso, apenas têm interesse pelo uso do computador.

Quanto às principais dificuldades encontradas no ensino de programação, a concentração e o entendimento da lógica foram elencados. Já em relação à dificuldade com os conhecimentos básicos de matemática e interpretação de texto, destacam a empolgação para uso do computador e a desconcentração no raciocínio. Também existe um grande desinteresse dos alunos, que não estudam e

nem praticam fora da faculdade. Ressaltam, por fim, que o nível de conhecimento de um aluno para outro é muito grande.

Sobre a possibilidade de mudar o cenário atual local e se vê esse problema como algo específico, alguns consideram o problema genérico, mas acreditam que pode ser amenizado com aulas de matemática básica e trabalho com questões de raciocínio lógico. Há professor que considera o problema como algo específico, mas vê a possibilidade de mudanças, pois é possível verificar grandes feitos em programação em outras universidades. Alguns acreditam que a mudança deve partir do aluno, ao praticar e treinar o desenvolvimento dos exercícios passados pelos professores, pois só com muita prática o aluno aprenderá a programar, sendo que a mudança é a longo prazo e, talvez, com a consolidação da UENP como Universidade, haja uma melhora no quadro.

A questão referente à época em que o professor estudou, se ele ou seus amigos tiveram dificuldades para aprender os conceitos de programação, dois deles disseram que não tiveram problemas em relação à programação, mas que muita coisa só foi entender anos depois; e que alguns amigos chegaram até a desistir do curso, pois viram que não era o que pensavam. Já outros dois professores disseram que houve dificuldades para aprender programação, principalmente no início, mas que, com muita prática e fazendo os exercícios propostos, foi-se desenvolvendo a habilidade de programar.

Das alternativas encontradas/adotadas/desejáveis para resolver ou mesmo diminuir as dificuldades de aprendizado do aluno, um dos professores coloca que aulas de matemática básica e de raciocínio lógico é uma boa alternativa a ser adotada; outros dizem que adotar programas de apoio ao aprendizado seria a solução, mas lembraram que a principal atitude de aprender deve partir dos alunos. Outros, que a adoção de turmas menores permitiria o acompanhamento mais individualizado, com 80% das aulas práticas e acreditam que a mudança para periodicidade anual permitirá um aprendizado melhor.

3.4 Fases do GIED

3.4.1 Análise

Nessa fase será contemplado os aspectos relativos às atividades que antecedem os processos de modelagem e implementação. Neste ponto do desenvolvimento são definidas as estratégias instrucionais, bem como o conteúdo e o público-alvo.

O conteúdo instrucional foi definido juntamente com um dos professores da disciplina de programação I da UENP-CLM, tendo como proposta apresentar aos alunos, por meio de uma abordagem sucinta e natural, os conceitos iniciais que introduzirão o aluno no contexto da disciplina de programação I, com o intuito de familiarizar o aluno com a lógica, os algoritmos e seus métodos de construção.

O objetivo é apresentar os conceitos elementares de lógica e sua aplicação no cotidiano, definir algoritmo, estabelecer uma relação entre lógica e algoritmos: a lógica de programação, exemplificar a aplicação dos algoritmos utilizando situações diárias e comparar as principais formas de representação de algoritmos. O conteúdo é fundamental para a compreensão e construção de algoritmos básicos.

O objeto de aprendizagem ensina:

- Introdução à lógica de programação;
- Transformar a lógica em algoritmo;
- Conceitos e exemplos de algoritmos; e
- Noções de fluxo de controle.

Uma das estratégias deste objeto é abordar os tópicos passo a passo, permitindo uma aprendizagem gradual e consistente, permitindo que o usuário se aproprie das técnicas fundamentais e crie uma base sólida em lógica de programação o que facilita na sequência, o aprendizado de tópicos mais complexos.

Também são utilizados alguns exemplos e analogias provenientes do dia-a-dia, para facilitar a explicação dos conceitos e para aprimorar os temas abstratos a assuntos ligados ao cotidiano pelo usuário.

Contém uma série de exercícios de fixação, criada para sedimentar conhecimentos locais ao conteúdo em discussão. Permitindo interação do usuário com o programa.

Quanto ao público alvo, ele pode ser utilizado em cursos de técnicas de programação e construção de algoritmos. Devido ao seu caráter didático, o detalhamento dos assuntos e a abrangência de seu conteúdo, sendo indicado para disciplinas que necessitam de uma ferramenta de apoio, pedagogicamente concebida para facilitar o aprendizado de programação. Também é útil como fonte de estudo independente e de aprimoramentos técnicos para interessados em lógica de programação.

O OA é totalmente voltado para a lógica de programação, conferindo assim um alto grau de acessibilidade para estudantes e iniciantes na programação de computadores.

Os pré-requisitos necessários para o aluno utilizar o objeto é que tenham de preferência o ensino médio.

O nome do OA é 'Aprendendo Programação' e está no formato executável, que pode ser usado como ferramenta de ensino pelo professor, como uma forma de auxiliar o processo de ensino-aprendizagem.

Os materiais: teórico e prático, que foram utilizados para elaboração do objeto foram retirados do livro 'Lógica de programação: a construção de algoritmos e estruturas de dados' (FERBOLONE & EBERSPÄCHER, 2005). O vídeo 'algoritmo pode ser acessado: em http://www.youtube.com/watch?v=3hv5_hWPleo, último acesso em: 06 de abril 2012. O vídeo 'Torre de Hanói' pode ser acessado em: <http://www.youtube.com/watch?v=hLnuMXO95f8>, último acesso em 06 de abril de 2012.

3.4.1 Projeto

Nessa fase, serão desenvolvidos os modelos que apoiarão a implementação. Serão desenvolvidos também o mapa conceitual, o mapa navegacional e os *storyboards*.

- Mapa conceitual: a base para o processo de desenvolvimento do OA, por meio do mapa conceitual, é feita por um mapeamento dos conteúdos abordados pelo OA como pode ser visto na figura 3.1, sendo apresentados os conceitos de forma estrutural.

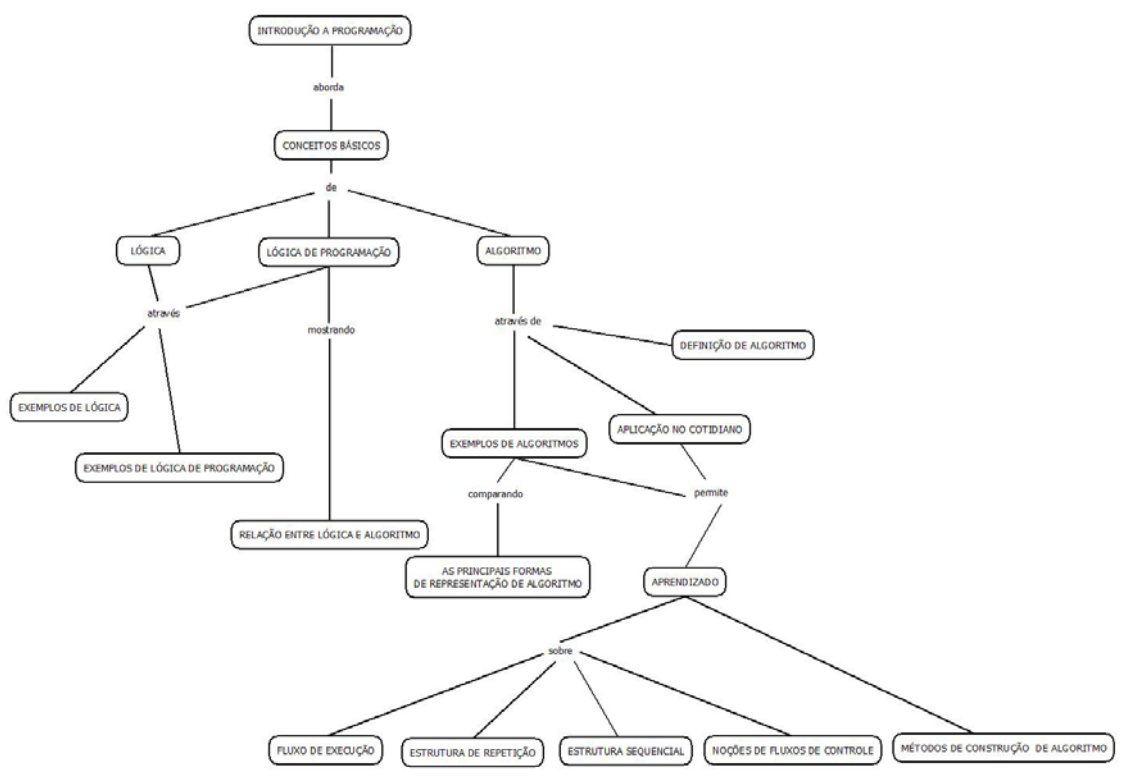


Figura 3.1 – Mapa Conceitual

- Mapa navegacional: o mapa navegacional representa a forma como será realizada a navegação entre as páginas do OA. Representa como o usuário irá navegar pelo objeto, podendo ser visto na Figura 3.2.

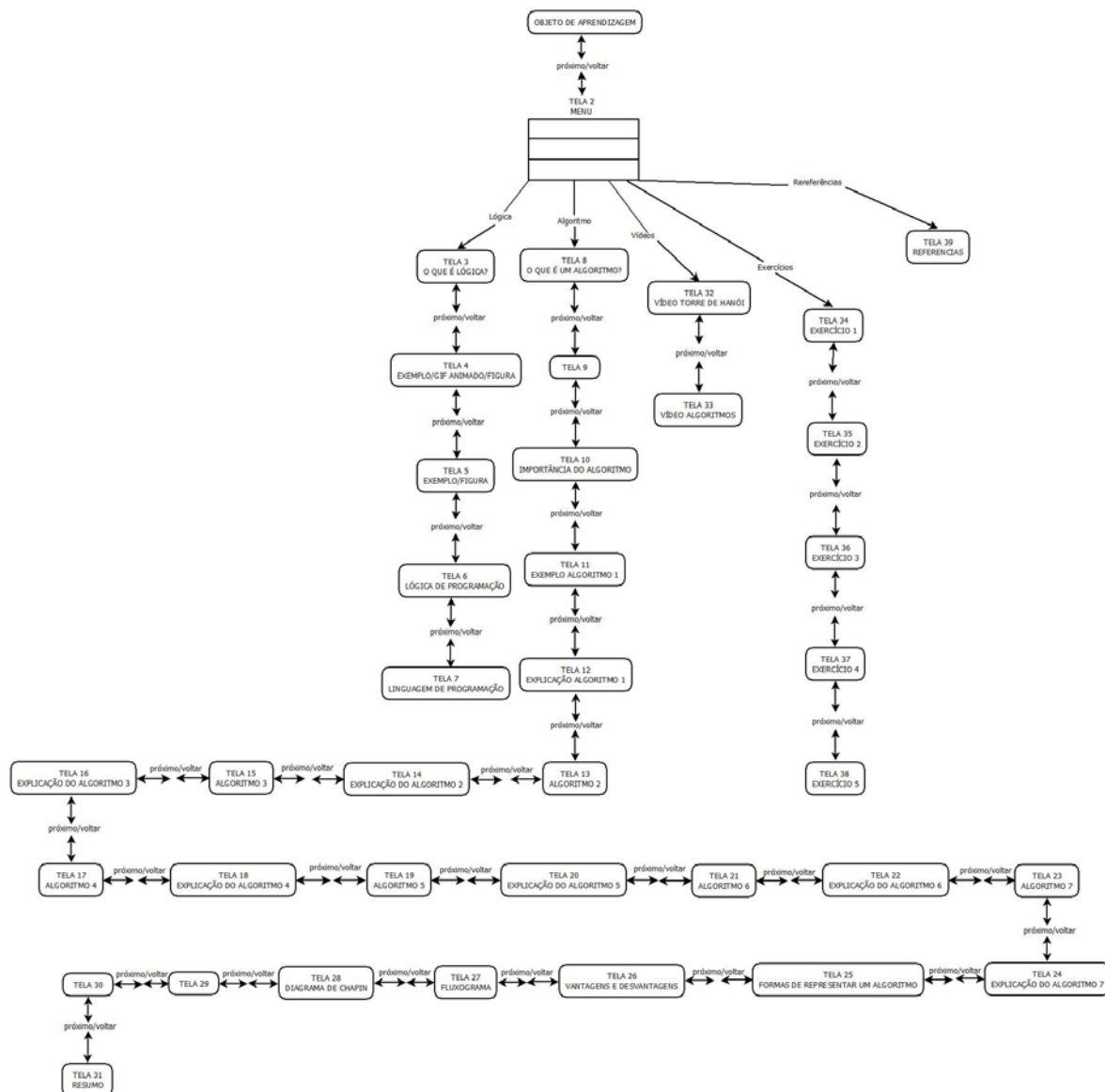


Figura 3.2 – Mapa Navegacional

- *Storyboard*: essa fase consiste na elaboração do *Layout* e orientações, roteiro a ser seguido na implementação. Veja abaixo alguns dos *storyboards* gerados para esse trabalho.



Universidade Estadual do Norte do
Paraná – *Campus* Luiz Meneghel

Storyboard: Aprendendo Programação

Disciplina: Programação I

Conteúdo: Teoria

Figuras	Título	Texto	Personagem	Cenário
 Centro da página	Objeto de Aprendizagem.	Uma forma diferente de aprender a programar. Obs.: Colocar o texto na parte inferior da tela.	Sem personagem.	

Figura 3.3 – *Storyboard* tela 1

A Figura 3.3 representa o *storyboard* da tela inicial do OA, sendo dividido em colunas. Cada coluna representa um componente da tela do objeto, que é composto por figuras, título, texto, personagem e cenário.



Universidade Estadual do Norte do
Paraná – *Campus* Luiz Meneghel

Storyboard: Aprendendo Programação

Disciplina: Programação I

Conteúdo: Teoria

Figuras	Título	Menu	Personagem	Cenário
 Imagem de fundo	Tela: Menu / Com o título Aprendendo Programação	<ul style="list-style-type: none"> - Lógica - Algoritmos - Vídeos - Exercícios - Referências 	 Obs.: fala da personagem em forma de balão, que aparece ao passar o mouse sobre a imagem da professora: Escolha uma das opções	

Figura 3.4 – *Storyboard* tela 2

No *storyboard* Figura 3.4 não aparece a coluna texto, que foi substituída pela coluna menu. Também na coluna personagem, aparece uma das personagens do OA e abaixo dela uma mensagem que aparece na forma de balão, quando se passa o mouse sobre a imagem.




Figuras	Título	Texto	Personagem	Cenário
Sem figura	O que é Lógica?	<p>O uso corriqueiro da palavra lógica está normalmente relacionado à coerência e à racionalidade. Frequentemente se associa lógica apenas à matemática, não se percebendo sua aplicabilidade e sua relação com as demais ciências.</p> <p>Podemos relacionar a lógica com a 'correção do pensamento', pois uma de suas preocupações é determinar quais operações são válidas e quais não são, fazendo análises das formas e leis do pensamento.</p> <p>Poderíamos dizer também que a lógica é a 'arte de bem pensar', que é a 'ciência das formas do pensamento'. Visto que a forma mais complexa do pensamento é o raciocínio, a lógica estuda a 'correção do raciocínio'.</p> <p>Podemos ainda dizer que a lógica tem em vista a 'ordem da razão'. Isto dá a entender que a nossa razão pode funcionar desordenadamente. Por isso a lógica estuda e ensina a colocar 'ordem no pensamento'.</p>	Sem personagem.	

Figura 3.5 – *Storyboard* tela 3

Na Figura 3.5 é apresentada o *storyboard* referente à tela 3 do OA, que tem pouco diferencial do anterior, apenas o conteúdo da coluna texto é um pouco mais extenso. O modelo acima não possui figuras nem mesmo personagens. O objeto aborda os assuntos de uma forma bem detalhada, usando muito os recursos de texto.





Figuras	Título	Texto	Personagem	Cenário
	Exemplo	<p>- Todo mamífero é um animal. - Todo cavalo é mamífero. - Portanto, todo cavalo é um animal.</p> <p>Esse exemplo ilustra silogismo, que no estudo da Lógica Proposicional (ou Cálculo Sentencial) representam um argumento composto de duas premissas e uma conclusão; e está estabelecendo uma relação, que pode ser válida ou não. Esse é um dos objetivos da lógica, o estudo de técnicas de formalização, dedução e análise que permitam verificar a validade de argumentos. No caso do exemplo acima citado é válido. Devemos ressaltar que, apesar da aparente coerência de um encadeamento lógico, ele pode ser válido ou não em uma estrutura.</p>	Sem personagem	

Figura 3.6 – *Storyboard* tela 4

Na Figura 3.6 são usadas figuras para auxiliar o aprendizado do aluno. São imagens que se relacionam com os exemplos usados no OA. Esse *storyboard* não possui personagem.



Universidade Estadual do Norte do
Paraná – *Campus* Luiz Meneghel

Storyboard: Aprendendo Programação
Disciplina: Programação I
Conteúdo: Teoria


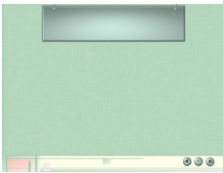
Figuras	Título	Texto	Personagem	Cenário
	Existe lógica no dia-a-dia	<p>Sempre que pensamos, a lógica ou a ilógica necessariamente nos acompanham. Quando falamos ou escrevemos, estamos expressando nosso pensamento, logo, precisamos usar a lógica nessas atividades. Podemos perceber a importância da lógica em nossa vida, não só na teoria, como na prática, já que, quando queremos pensar, falar, escrever ou agir corretamente, precisamos colocar 'ordem no pensamento', isto é, utilizar lógica.</p> <p>Exemplo</p> <ul style="list-style-type: none"> - A gaveta está fechada. - A caneta está dentro da gaveta. - Precisamos primeiro abrir a gaveta para depois pegar a caneta. 	Sem personagem.	

Figura 3.7 – Storyboard tela 5

Na Figura 3.7 também é utilizada uma figura para ajudar na assimilação com o exemplo que é apresentado. Assim como a Figura 3.6, ela não possui personagem.



Universidade Estadual do Norte do
Paraná – *Campus* Luiz Meneghel

Storyboard: Aprendendo Programação
Disciplina: Programação I
Conteúdo: Teoria


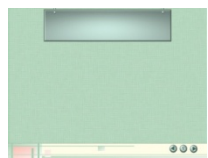
Figuras	Título	Texto	Personagem	Cenário
 <p>Imagem de fundo</p>	Vejam esse primeiro algoritmo, descrito passo a passo	<p>Algoritmo 1.1 - Troca de Lâmpada</p> <ul style="list-style-type: none"> -pegar uma escada; -posicionar a escada embaixo da lâmpada; -buscar uma lâmpada nova; - subir a escada; - retirar a lâmpada velha; -colocar a lâmpada nova; 	Sem personagem.	

Figura 3.8 – Storyboard Tela 11

Na Figura 3.8, a figura é utilizada como imagem de fundo, mesclada com o fundo. Também se relaciona com o exemplo proposto no *storyboard*.



Universidade Estadual do Norte do Paraná – Campus Luiz Meneghel

Storyboard: Aprendendo Programação

Disciplina: Programação I

Conteúdo: Teoria

Figuras	Título	Texto	Personagem	Cenário
	<p>Algoritmo 1.8 - Fluxograma</p>	<p>Sem texto</p>	<p>Sem personagem</p>	

Figura 3.9 – Storyboard tela 28

A Figura 3.9 traz a imagem de um fluxograma, que serve de exemplo para que o usuário compreenda os conceitos abordados.



Universidade Estadual do Norte do
Paraná – *Campus* Luiz Meneghel

Storyboard: Aprendendo Programação
Disciplina: Programação I
Conteúdo: Teoria

Figuras	Título	Texto	Personagem	Cenário
	Algoritmo 1.9 – Diagrama de Chapin	Sem texto	Sem personagem.	

Figura 3.4 - *Storyboard* tela 29

Na Figura 3.4 é usado a imagem do diagrama de Chapin que, assim como no na Figura 3.3, foi também necessário para explicar os conceitos abordados.



Universidade Estadual do Norte do
Paraná – *Campus* Luiz Meneghel

Storyboard: Aprendendo Programação
Disciplina: Programação I
Conteúdo: Teoria

Figuras	Título	Menu	Personagem	Cenário
	Vídeos	- Torre de Hanói - Algoritmo	<p>Obs.: fala da personagem em forma de balão, que aparece ao passar o mouse sobre a imagem do professor: Escolha um dos vídeos</p>	

Figura 3.5 – *Storyboard* tela 33

A Figura 3.5 é o *storyboard* composto pela tela menu dos vídeos. Também é utilizado o personagem do professor. Ao passar o mouse sobre o personagem, aparece a mensagem ‘escolha um dos vídeos’. Os vídeos foram utilizados para complementar o conteúdo do OA.



Universidade Estadual do Norte do
Paraná – *Campus* Luiz Meneghel

Storyboard: Aprendendo Programação
Disciplina: Programação I
Conteúdo: Teoria


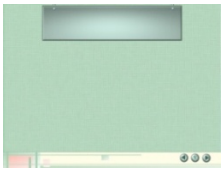
Figuras	Título	Vídeo	Personagem	Cenário
Sem figura	Torre de Hanói		Sem personagem	

Figura 3.6 – *Storyboard* tela 34

A Figura 3.6 compõe a tela com o vídeo ‘Torre de Hanói’, que, quando acessada, inicia a execução do vídeo.



Universidade Estadual do Norte do
Paraná – *Campus* Luiz Meneghel

Storyboard: Aprendendo Programação
Disciplina: Programação I
Conteúdo: Teoria



Figuras	Título	Vídeo	Personagem	Cenário
Sem Figura	Conteúdo Auxiliar		Sem personagem	

Figura 3.7 – *Storyboard* tela 35

A Figura 3.7 compõe a tela com o vídeo ‘Algoritmo’, que, quando acessada, também inicia a execução do vídeo.




Explicação	Exercício	Feedback	Alternativas	Cenário
<p>Três senhoras – dona Branca, dona Rosa e dona Violeta – passeavam pelo parque quando dona Rosa disse:</p> <p>- Não é curioso que estejamos usando vestidos de cores branca, rosa e violeta, embora nenhuma de nós esteja usando um vestido de cor igual ao seu próprio nome ?</p> <p>- Uma simples coincidência - respondeu a senhora com o vestido violeta. Qual a cor do vestido de cada senhora ?</p>	<p>Exercício 1: Selecione a alternativa correta</p>	<p>Negativo</p> <p>a) Se a senhora com o vestido violeta respondeu a dona Rosa, então ela não é a própria dona Rosa. Além disso, como ela não tem o vestido da mesma cor do seu nome, ela também não é a dona Violeta. Logo, é dona Branca que está com o vestido violeta.</p> <p>c) Nenhuma delas está usando o vestido com a cor igual ao seu próprio nome.</p> <p>d) Nenhuma delas está usando o vestido com a cor igual ao seu próprio nome.</p>	<p>a) Dona Branca está com o vestido rosa, Dona Rosa está com o vestido violeta e Dona Violeta está com o vestido branco.</p> <p>b) Dona Branca está com o vestido violeta, Dona Rosa está com o vestido branco e Dona Violeta está com o vestido rosa.</p> <p>c) Dona Branca está com o vestido rosa, Dona Rosa está com o vestido branco e Dona Violeta está com o vestido violeta.</p> <p>d) Dona Branca está com o vestido branco, Dona Rosa está com o vestido violeta e Dona Violeta está com o vestido rosa.</p>	
		<p>Positivo</p> <p>Muito Bem!</p>	<p>Respostas</p> <p>b) Dona Branca está com o vestido violeta, Dona Rosa está com o vestido branco e Dona Violeta está com o vestido rosa.</p>	

Figura 3.8 – *Storyboard* tela 36

A Figura 3.8 corresponde ao *storyboard* do primeiro exercício, contendo a explicação do exercício, o nome do exercício e um *feedback*, caso o usuário erre ao escolher a resposta. Contém também as alternativas, as respostas e o cenário.



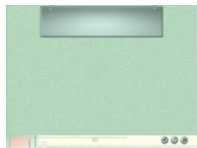
Explicação	Exercício	Alternativas	Cenário
Um homem precisa atravessar um rio com um barco que possui capacidade apenas para carregar ele mesmo e mais uma de suas três cargas, que são: um lobo, um bode e um maço de alfafa. O que o homem deve fazer para conseguir atravessar o rio sem perder suas cargas?	Exercício 2	<input type="checkbox"/> -levar a alfafa para o outro lado do rio. <input type="checkbox"/> -voltar sem carga nenhuma. <input type="checkbox"/> -levar o bode para o outro lado do rio. <input type="checkbox"/> - voltar com o bode. <input type="checkbox"/> -voltar sem carga alguma. <input type="checkbox"/> -levar o bode para o outro lado do rio. <input type="checkbox"/> -levar o lobo para o outro lado do rio.	
	Feedback	Respostas	
	Negativo		
	Reveja preenchimento	<input checked="" type="checkbox"/> 5 -levar a alfafa para o outro lado do rio. <input checked="" type="checkbox"/> 2 -voltar sem carga nenhuma. <input checked="" type="checkbox"/> 1 -levar o bode para o outro lado do rio. <input checked="" type="checkbox"/> 4 -voltar com o bode. <input checked="" type="checkbox"/> 6 -voltar sem carga alguma. <input checked="" type="checkbox"/> 7 -levar o bode para o outro lado do rio. <input checked="" type="checkbox"/> 3 -levar o lobo para o outro lado do rio.	
	Positivo		
	Parabéns		

Figura 3.9 – Storyboard tela 37

Os exercícios possuem a mesma estrutura, como pode ser visto na Figura 3.9, sendo compostos pelas mesmas colunas do *storyboard* da Figura 3.6.

A cada exercício são exigidos um pouco mais da interação do usuário e um pouco mais de concentração. Os conteúdos vão se tornando mais difíceis, mas ao mesmo tempo estimulantes e desafiador.



Universidade Estadual do Norte do
Paraná – *Campus* Luiz Meneghel

Storyboard: Aprendendo Programação
Disciplina: Programação I
Conteúdo: Prática

Explicação	Exercício	Alternativas	Cenário
<p>Elabore um algoritmo que mova três discos de uma Torre de Hanói, que consiste em três hastes (a - b - c), uma das quais serve de suporte para três discos de tamanhos diferentes (1 - 2 - 3), os menores sobre os maiores.</p> <p>Pode-se mover um disco de cada vez para qualquer haste, contanto que nunca seja colocado um disco maior sobre o menor.</p> <p>O objetivo é transferir os três discos para outra haste C.</p> <p>Os passos necessários para o algoritmo estão abaixo, apenas enumere-os.</p>	Exercício 3	<input type="checkbox"/> Mover o disco 3 da torre A para a torre C. <input type="checkbox"/> Mover o disco 1 da torre A para a torre C. <input type="checkbox"/> Mover o disco 2 da torre B para a torre C. <input type="checkbox"/> Mover o disco 2 da torre A para a torre B. <input type="checkbox"/> Mover o disco 1 da torre C para a torre B. <input type="checkbox"/> Mover o disco 1 da torre A para a torre C. <input type="checkbox"/> Mover o disco 1 da torre B para a torre A.	
	Feedback	Respostas	Figuras
	Negativo	<input checked="" type="checkbox"/> 4- Mover o disco 3 da torre A para a torre C. <input checked="" type="checkbox"/> 1- Mover o disco 1 da torre A para a torre C. <input checked="" type="checkbox"/> 6- Mover o disco 2 da torre B para a torre C. <input checked="" type="checkbox"/> 2- Mover o disco 2 da torre A para a torre B. <input checked="" type="checkbox"/> 3- Mover o disco 1 da torre C para a torre B. <input checked="" type="checkbox"/> 7- Mover o disco 1 da torre A para a torre C. <input checked="" type="checkbox"/> 5- Mover o disco 1 da torre B para a torre A.	
	Reveja preenchimento		Positivo

Figura 3.10 – *Storyboard* tela 38

O exercício, que é apresentado na Figura 3.10, utiliza também uma figura, sendo um diferencial dos demais exercícios. O usuário ordena os passos para resolver a ‘Torre de Hanói ‘.



Universidade Estadual do Norte do
Paraná – *Campus* Luiz Meneghel

Storyboard: Aprendendo Programação
Disciplina: Programação I
Conteúdo: Prática

Explicação	Exercício	Feedback	Alternativas	Cenário
<p>Até mesmo as coisas mais simples podem ser descritas por seqüências lógicas. Por exemplo: chupar uma bala.</p> <p>Ligue os pontos, colocando em ordem os passos necessários para chupar uma bala.</p>	Exercício 4	Negativo	<input type="button" value="Passo 1"/> <input type="button" value="Jogar o papel no lixo"/> <input type="button" value="Passo 3"/> <input type="button" value="Chupar a bala"/> <input type="button" value="Passo 2"/> <input type="button" value="Pegar a bala"/> <input type="button" value="Passo 4"/> <input type="button" value="Retirar o papel"/>	
		Positivo	Respostas	
		Parabéns	<input type="button" value="Passo 1"/> <input type="button" value="Jogar o papel no lixo"/> <input type="button" value="Passo 3"/> <input type="button" value="Chupar a bala"/> <input type="button" value="Passo 2"/> <input type="button" value="Pegar a bala"/> <input type="button" value="Passo 4"/> <input type="button" value="Retirar o papel"/>	

Figura 3.11 – *Storyboard* tela 39

A Figura 3.11 segue o mesmo padrão dos outros exercícios, com exceção do *feedback* negativo, pois quando o usuário erra não permite ligar os pontos. O usuário só consegue ligar os pontos quando a resposta estiver certa. Mas quando ele tenta ligar errado é calculado como erro.



Universidade Estadual do Norte do
Paraná – *Campus* Luiz Meneghel

Storyboard: Aprendendo Programação
Disciplina: Programação I
Conteúdo: Prática


Exercício	Feedback	Alternativas
Exercício 5 - Complete as frases com as Palavras abaixo.	Negativo Sem Feedback Negativo	estrutura sequencial, estrutura de seleção, lógica, estrutura de repetição, algoritmo, fluxo de execução
Frases	Positivo	Respostas
<p>1 - Um _____ pode ser definido como uma sequência de passos que visam a atingir um objetivo bem definido.</p> <p>2- A _____ se relaciona com a 'ordem da razão', com a correção do pensamento', e que é necessário utilizar processos lógicos de programação para construir algoritmos.</p> <p>3- A _____ significa que o algoritmo é executado passo a passo, sequencialmente, da primeira à última ação.</p> <p>4 – A _____ permite que uma ação seja ou não executada, dependendo do valor resultante da inspeção de uma condição.</p> <p>5 - A _____ permite que trechos de algoritmos sejam repetidos até que uma condição seja satisfeita ou enquanto uma condição não estiver satisfeita.</p> <p>6 - Através do exemplo das lâmpadas introduzimos o conceito de controle do _____ e mostramos a estrutura sequencial, de repetição e de seleção.</p>	<p>Parabéns</p> <p>Cenário</p> 	<p>1 - Um <u>algoritmo</u> que pode ser definido como uma sequência de passos que visam a atingir um objetivo bem definido.</p> <p>2- A <u>lógica</u> se relaciona com a 'ordem da razão', com a correção do pensamento', e que é necessário utilizar processos lógicos de programação para construir algoritmos.</p> <p>3- A <u>estrutura sequencial</u> significa que o algoritmo é executado passo a passo, sequencialmente, da primeira à última ação.</p> <p>4 – A <u>estrutura de seleção</u> permite que uma ação seja ou não executada, dependendo do valor resultante da inspeção de uma condição.</p> <p>5 - A <u>estrutura de repetição</u> permite que trechos de algoritmos sejam repetidos até que uma condição seja satisfeita ou enquanto uma condição não estiver satisfeita.</p> <p>6 - Através do exemplo das lâmpadas introduzimos o conceito de controle do <u>fluxo de execução</u> e mostramos a estrutura sequencial, de repetição e de seleção.</p>

Figura 3.12 – Storyboard tela 40

Como mostrado na Figura 3.12, o último exercício é um 'complete' que segue a mesma estrutura do exercício 4 e se o usuário errar, o programa não permite o usuário completar a frase, somente quando ele acerta é que se encaixa a palavra no lugar certo, mas que também é calculado como erro.

3.4.2 Implementação

O desenvolvimento do OA foi baseado no arquivo gerado na fase de projeto e *storyboard*. Com base nos critérios de usabilidade, de acessibilidade e de padrões de desenvolvimento destacados anteriormente por Wiley (2009), foi concretizada a implementação do objeto, procurando gerar produtos adequados ao público-alvo.

A partir de cada *storyboard* foi gerada uma tela do OA, propondo uma sequência para o conteúdo abordado no objeto, a fim de facilitar a navegação e torná-la mais organizada. A partir da tela inicial, segue a apresentação do menu, no qual são apresentados os tópicos abordados pelo OA. O usuário pode acessar o tópico que desejar, ou pode seguir o roteiro proposto, que consiste em conteúdo teórico, seguido por vídeos usados para complementar o conteúdo teórico e por fim, fazer as atividades referentes ao conteúdo explorado durante a apresentação da teoria.

Vejam-se a seguir algumas telas do objeto, podendo notar que os elementos definidos no *storyboard* são mantidos na implementação. Tudo que foi proposto no documento é realizado no objeto. Como personagem, figuras, cenário, enfim. Cada detalhe anteriormente definido é implementado nessa fase.



Figura 3.13 - Tela 1

A Figura 3.13 corresponde ao que foi proposto na Figura 3.3.



Figura 3.14 - Tela 2

A Figura 3.14 corresponde ao menu principal do OA, que foi proposto na Figura 3.4, permitindo ao usuário navegar com mais facilidade pelo objeto.

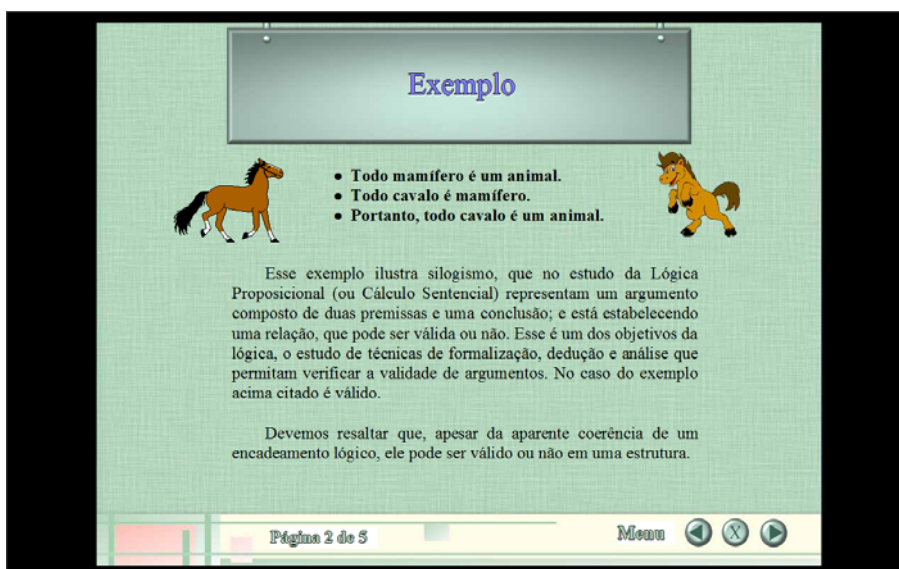


Figura 3.15 - Tela 4

A Figura 3.15 corresponde ao que foi proposto na Figura 3.6. Pode-se observar que as figuras foram adicionadas, como definido no *storyboard*.



Figura 3.16 - Tela 33

A Figura 3.16 corresponde ao menu dos vídeos como foi proposto na Figura 3.5. Pode-se notar a fala do personagem, os tópicos relativos aos vídeos, o personagem.



Figura 3.17 - Tela 34

A Figura 3.17 refere-se à tela de um dos vídeos, que foi proposto na Figura 3.6. O vídeo é executado na própria tela do OA, a fim de facilitar a navegação.

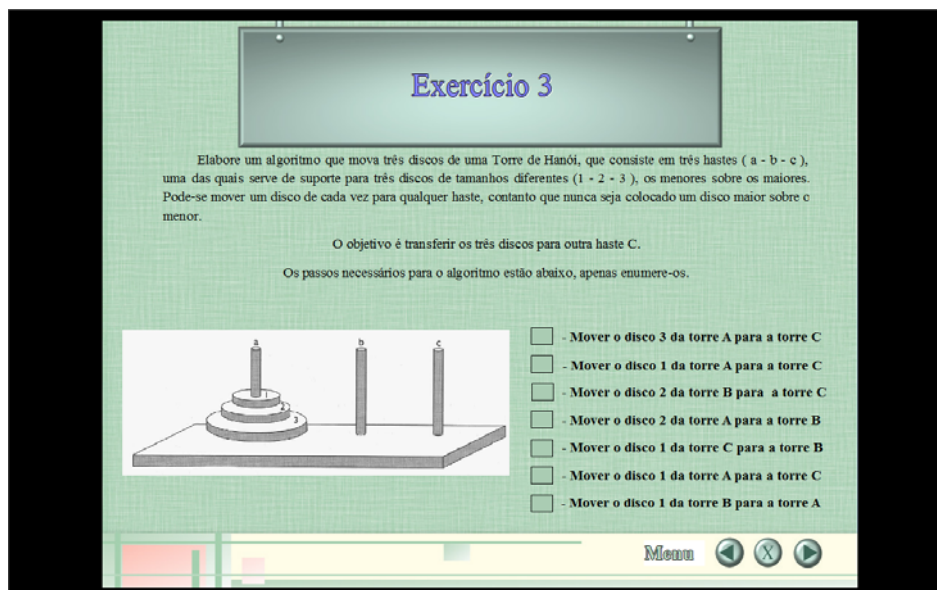


Figura 3.18 - Tela 38

A Figura 3.18 demonstra a tela de exercício proposto na Figura 3.10, que corresponde a um dos exercícios propostos pelo OA.

3.4.3 Validação

Nessa etapa foi realizada a avaliação do OA por parte de uma parcela do público-alvo. Posteriormente à avaliação, foram desenvolvidos os manuais do usuário final e do professor.

- **Avaliação:** para a avaliação foram realizados testes com o OA, com o propósito de avaliar o funcionamento. Verificou-se nessa fase se o Objeto estaria adequado aos critérios técnicos pedagógicos; usabilidade e interface.

Para avaliação foi utilizado um questionário de avaliação de *software* educacional por meio do formulário *google docs*. Teve a participação de 29 alunos do primeiro ano do curso de Sistemas de Informação da UENP-CLM. O questionário utilizado pode ser acessado em: <http://gied.ffalm.br>.

O objeto de aprendizagem obteve resultados positivos em relação aos critérios avaliados, como pode ser observado na Tabela 3.1 e na Tabela 3.2 que correspondem aos aspectos técnico-pedagógicos.

No quesito usabilidade, obteve-se uma média de 78%, como mostra a Tabela 3.3. Por fim, no critério interface, o resultado foi de 81,18%.

Os questionários com os resultados tabulados, juntamente com as fórmulas utilizadas para avaliação, estão anexados ao final do trabalho (APÊNDICE C).

Tabela 3.1 – Critérios Técnico-Pedagógicos Parte 1

Conformidade dos Critérios Avaliados	
Documentação	87,50%
Informações Pedagógicas - Observadas pelo Avaliador	94,33%
Aspectos Didáticos e de Conteúdo	87,93%
Aspectos Emocionais e Afetivos	85,52%
Aspectos Cognitivos	86,78%
Porcentagem Mínima para que um Critério seja Aceito	
Documentação	50,00%
Informações Pedagógicas - Observadas pelo Avaliador	70,00%
Aspectos Didáticos e de Conteúdo	70,00%
Aspectos Emocionais e Afetivos	60,00%
Aspectos Cognitivos	70,00%
Verificação se o <i>Software</i> foi Aceito ou Não Aceito	
Documentação	ACEITO
Informações Pedagógicas - Observadas pelo Avaliador	ACEITO
Aspectos Didáticos e de Conteúdo	ACEITO
Aspectos Emocionais e Afetivos	ACEITO
Aspectos Cognitivos	ACEITO

Tabela 3.2 – Critérios Técnico-Pedagógicos Parte 2

Critérios	Conformidade
1. Pertinência em relação a uma proposta pedagógica	91,72%
2. Interação	86,95%
3. Recursos motivacionais	81,23%
4. Gestão de erros	78,74%
5. Adaptabilidade	78,13%
6. Correspondência entre o <i>software</i> e o mundo real	84,48%
7. Conteúdo	75,00%
8. Avaliação da aprendizagem	85,06%
9. Controle e autonomia do usuário	82,47%
10. Utilização de recursos computacionais	86,21%
11. Carga de trabalho	78,02%
12. Significado de códigos e denominações	78,16%
13. Consistência e padrões	86,21%
14. Documentação	83,74%
Média de todos os critérios	82,58%
Porcentagem Mínima para que um Critério seja Aceito	
1. Pertinência em relação a uma proposta pedagógica	70%
2. Interação	50%
3. Recursos motivacionais	60%
4. Gestão de erros	50%
5. Adaptabilidade	60%
6. Correspondência entre o <i>software</i> e o mundo real	50%
7. Conteúdo	70%
8. Avaliação da aprendizagem	40%
9. Controle e autonomia do usuário	70%
10. Utilização de recursos computacionais	50%
11. Carga de trabalho	60%
12. Significado de códigos e denominações	60%
13. Consistência e padrões	50%
14. Documentação	40%
Verificação se o Critério foi Aceito ou Não Aceito	
1. Pertinência em relação a uma proposta pedagógica	ACEITO
2. Interação	ACEITO
3. Recursos motivacionais	ACEITO
4. Gestão de erros	ACEITO

5. Adaptabilidade	ACEITO
6. Correspondência entre o <i>software</i> e o mundo real	ACEITO
7. Conteúdo	ACEITO
8. Avaliação da aprendizagem	ACEITO
9. Controle e autonomia do usuário	ACEITO
10. Utilização de recursos computacionais	ACEITO
11. Carga de trabalho	ACEITO
12. Significado de códigos e denominações	ACEITO
13. Consistência e padrões	ACEITO
14. Documentação	ACEITO

Tabela 3.3 - Usabilidade

Crítérios	Conformidade(%)
1. Presteza	80,12%
2. Agrupamento por Localização	83,23%
3. Agrupamento por Formato	81,44%
4. <i>Feedback</i>	83,33%
5. Legibilidade	79,12%
6. Concisão	75,99%
7. Ações Mínimas	82,07%
8. Densidade Informal	81,03%
9. Ações Explícitas	71,98%
10. Controle de Usuário	71,98%
11. Flexibilidade	68,39%
12. Experiência do Usuário	77,59%
13. Proteção contra Erro	80,30%
14. Mensagem de Erro	75,10%
15. Correção de Erro	75,17%
16. Consistência	76,33%
17. Significados	81,90%
18. Compatibilidade	78,98%
MÉDIA =	
	78,00%

- **Manuais:** Nesta fase foram elaboradas as documentações necessárias para uma melhor utilização do OA, divididas em:

- **Manual do usuário:** o manual foi desenvolvido para auxiliar os alunos na utilização do OA, enfatizando o modo de navegação, leitura e realização das atividades apresentadas, caso tivessem alguma dúvida (APÊNDICE D).

- **Manual do professor:** nesse manual foi apresentada uma proposta de aula para o professor que pretender utilizar esse objeto de aprendizagem (APÊNDICE E). Destacam-se itens importantes para o decorrer dessa aula, sendo eles:
 - Objetivos da aula;
 - Duração mínima da aula;
 - Pré-requisitos didáticos;
 - Pré-requisitos técnicos;
 - Motivação;
 - Estratégia que deve ser adotada;
 - Métodos a serem seguidos para um melhor aproveitamento da aula; e
 - Forma de avaliação.

Ao término dessa fase, observou-se que o OA está em conformidade com os itens estabelecidos nos documentos gerados nas fases anteriores.

4 RESULTADOS OBTIDOS

O resultado obtido por meio desse trabalho foi o desenvolvimento do objeto de Aprendizagem 'Aprendendo a Programar', seguindo a metodologia de desenvolvimento GIED e utilizando como ferramenta de desenvolvimento o *software* de autoria *Visual Class*. O objeto é composto por 41 telas divididas em conteúdo teórico e prático, abordando alguns dos conceitos iniciais da disciplina de programação I, sendo eles:

- Conceitos de lógica;
- A aplicação da lógica no cotidiano;
- Definição de algoritmo;
- Lógica de programação; e
- Principais formas de representação de algoritmos.

O conteúdo foi apresentado ao aluno de uma maneira sucinta⁵ e natural⁶, familiarizando o aluno com a lógica, os algoritmos e seus métodos de construção. Os exemplos de algoritmos utilizados foram de situações do cotidiano, sendo abordados passo a passo, permitindo uma aprendizagem gradual e consistente. Permite também ao usuário se apropriar das técnicas fundamentais e criar uma base sólida em lógica de programação, facilitando na sequência o aprendizado de tópicos mais complexos.

A linguagem empregada foi bastante informal e acessível, contendo também 5 exercícios de fixação, criados para sedimentar conhecimentos locais ao conteúdo em discussão. Os exercícios são práticos, caracterizando um ambiente construtivista, razão pela qual permite ao usuário a interação com o programa. Possui ainda 2 vídeos explicativos, que foram adicionados para complementar o conteúdo abordado.

O OA foi avaliado por um público alvo de vinte e nove alunos do primeiro ano do curso de Sistemas de Informação da UENP-CLM. Também foram desenvolvidos manuais, para acompanhar o objeto e auxiliar quanto ao seu bom uso, sendo um manual para o professor e outro para o usuário.

⁵ que se diz ou se escreve resumidamente

⁶ de maneira simples

5 CONCLUSÃO

Com base nos resultados obtidos nesse trabalho, nota-se que o objeto desenvolvido pode ser utilizado para o ensino de conceitos iniciais de programação. Sendo utilizado juntamente com os manuais do professor e do usuário.

O objeto de aprendizagem desenvolvido teve boa aceitação pelos alunos, obtendo bons resultados com relação a critérios: técnico-pedagógicos, usabilidade e interface, mostrando que pode contribuir para o ensino de programação.

A opinião dos alunos que testaram a ferramenta foi de grande importância, pois permitiu que se conhecessem os pontos que ainda precisam ser melhorados, como o critério de usabilidade que obteve a média mais baixa entre os critérios avaliados.

O objeto foi desenvolvido para auxiliar no processo de ensino-aprendizagem, não descartando a presença do professor, mas sendo usado como ferramenta de apoio para auxiliar no processo de ensino-aprendizagem.

Observando os Gráficos 3.7 e 3.8, conclui-se que o aluno, na fase inicial do curso de Sistemas de Informação, estuda pouco para o curso e para a disciplina de programação I, revelando que o aluno precisa se dedicar mais para a disciplina de programação e para o curso de uma forma geral.

Os alunos, que consideram seu conhecimento em relação aos conceitos de algoritmos entre baixo e médio, revelaram também que não se sentem seguros e consideram a disciplina como um desafio a ser vencido. A conclusão a que se chega é de que as dificuldades dos alunos em relação a esses conceitos são vários, como mostrou o Gráfico 3.6, revelando a insegurança dos alunos quanto a programação.

Analisando os dados da entrevista com os professores, nota-se que as dificuldades dos alunos são várias, mas que algo pode ser feito, como aplicar reforço de matemática e informática básica; propor atividades que estimulem a concentração do aluno; e incentivar os alunos a procurarem mais os monitores.

Várias metodologias foram estudadas durante o desenvolvimento desse trabalho, cada uma delas com sua característica particular, mas todas com o mesmo propósito: colaborar com o ensino e aprendizagem. O objeto desenvolvido foi validado e, em todos os critérios avaliados, teve aceitação, revelando assim que está pronto para ser usado.

6 TRABALHOS FUTUROS

Para trabalhos futuros, sugere-se que sejam realizados testes de validações mais apuradas com o objeto desenvolvido, realizando assim melhorias no OA e que outros OAs possam ser desenvolvidos utilizando a metodologia GIED e o *software* de autoria *Visual Class*.

O objeto de aprendizagem também pode ser aplicado em plataformas de ensino a distância e em ambientes de rede de computadores.

7 REFERÊNCIAS

- BECK, R, J. . *Learning Objects: What?* Milwaukee: University of Wisconsin, 2001. Disponível em: <http://www.nwoet.org/tins/learning%20objects1.pdf>. Acesso em: 02/11/2011.
- BEHAR, P.A. (orgs.) *Modelos Pedagógicos em Educação a Distância*. Porto Alegre: Artmed, 2009. Disponível em: http://downloads.artmed.com.br/public/B/BEHAR_Patricia_Alejandra/Modelos_Pedagogicos_Educacao_Distancia/Liberado/cap_01.pdf . Acesso em: 02/11/2011.
- BETTIO, R. W. de; MARTINS, A. *Objetos de aprendizado: um novo modelo direcionado ao ensino a distância*. Document online publicado em 17/12/2004: Disponível em: <<http://www.universia.com.br/materia/materia.jsp?id=5938>>. Acesso em: 20/05/2006.
- BARCAROLO, Leandra Bolzane. *A informática na aquisição do conhecimento da matemática: o papel da psicopedagogia*. 2007, 47 f. Trabalho (Especialização em Educação e Psicopedagogia) – Pontifícia Universidade Católica de Campinas., PUC, Campinas, 2007. Disponível em: <<http://200.18.252.57/services/monografias/Leandra%20Bolzane%20Barcarolo.pdf>>. Acesso em: 31-05-12.
- COLELLO, S. M. G. . *A formação de professores na perspectiva do fracasso escolar*. Revista Internacional d'Humanitats, Barcelona, v. ano V, n. n 5, p. 39-46, 2002. Disponível em: <<http://www.hottopos.com/rih5/index.htm>>. Acesso em: 22-10-11.
- Downes, Stephen. *Learning Objects: Resources for distance education worldwide*, 2001. Disponível em : <http://www.irrodl.org/index.php/irrodl/article/viewArticle/32/378>. Acesso em: 02/11/2011.
- FORBELLONE, A. L. V.; EBERSPÄCHER, H, F. *Lógica de programação: a construção de algoritmos e estruturas de dados*. Pearson, São Paulo, 2005.
- HERCULIANI, Cristóvam Emílio. *Desenvolvimento de um software de autoria para alunos deficientes não-falantes nas atividades de contos e recontos de histórias*. 2007, 110 f. Dissertação (Mestrado em Educação) – Faculdade de Filosofia e Ciências, Marília. 2007. Disponível em: <http://www.marilia.unesp.br/Home/Pos-Graduacao/Educacao/Dissertacoes/herculiani_ce_ms_mar.pdf>. Acesso em: 31-05-12.
- IEEE. Learning Technology Standards Committee (LTSC P1484.12). *Learning Objects Metadata (LOM)*, 2000. Disponível em: <http://ltsc.ieee.org/wg12/> . Acesso em: 02/11/2011.
- JESUS, Alexandre Negrão de; LOPES, Daniel Lourenço, PERIN, Fernanda Rezende; CANTÃO, Juliana Martin; PINHEIRO, Edson. *Objetos de Aprendizagem no*

ensino da Lógica de Programação. Revista Informática Aplicada, v. III nº 2 jul/dez 2007.

KASIM, Valéria Mattos; SILVA, Otilia Marcacci Ribeiro da. *Software de autoria apoiando a aprendizagem*. Disponível em: <http://www.pucpr.br/eventos/educere/educere2008/anais/pdf/640_552.pdf>. Acesso em: 31-05-12.

KOLIVER, C.; DORNELES, R. V.; CASA, M. E. “*Das (muitas) dúvidas e (poucas) certezas do ensino de algoritmos*”. XII Workshop de Educação em Computação (WEI'2004). Salvador, 2004).

LUCENA, S.V. de Sá; DIAS, C. C. L.; KEMCZINSKI, A; HOUNSELL, M. da Silva. *Metodologias de desenvolvimento para a construção de objetos de aprendizagem*. CRICTE - Congresso Regional de Iniciação Científica e Tecnológica em Engenharia, 2009.

Disponível em: <http://www.mendeley.com/research/metodologias-desenvolvimento-para-construo-objetos-aprendizagem> . Acesso em: 03/11/2011.

MARTINS, B; WINCKLER, R. *Objetos de aprendizado: um novo modelo direcionado ao ensino a distância*, 2004. Disponível em: <<http://www.universia.com.br/ead/materia.jsp?materia=5938>>. Acesso em: 02/11/2011.

MEC. Ministério da Educação. Online: <http://portal.mec.gov.br> - Acessado em: 02/11/2011.

MUSTARO, P. N.; SILVEIRA, I. F.; OMAR, N.; STUMP, S. M. D.. *Structure of Storyboard for Interactive Learning Objects Development*. In: Learning Objects: and instructional design, Koohang, A. and Harman, K., Santa Rosa, Informing Science Press, p.253-279, 2007.

MUZIO, J.; HEINS, T.; MUNDELL, R. *Experiences with Reusable e-Learning Objects: From Theory to Practice*. Victoria, Canadá, 2001. Disponível em: <http://www.udutu.com/pdfs/eLearning-objects.pdf>. Acesso em: 02/11/2011.

NOBRE, I. A. M. N.; MENEZES, C. S. *Suporte à Cooperação em um Ambiente de Aprendizagem para Programação (SAmbA)* . Simpósio Brasileiro de Informática na Educação – SBIE 2002. São Leopoldo, 2002.

NUNES, C. A. A.; NEVES, D.; DEGANI, R. ; SANTOS, A. F.; GOUVEIA, P. ; OKUYAMA, R.; GÓES, R.; PAIDETI, E.; NAVAS, A. M.; FEJES, M. E. . *O processo de autoria/produção de objetos de aprendizagem de química: Uma experiência de trabalho colaborativo universidade-escola*. Virtual Educa Bilbao, 2006. Disponível em: <http://ihm.ccadet.unam.mx/virtualeduca2006/pdf/93-CN.pdf>. Acesso em: 04/11/2011.

OLIVEIRA, Kethure Aline. *Contribuição dos Objetos de Aprendizagem no ensino de Língua Portuguesa com foco na Aprendizagem Significativa*. 2011. 64p. Trabalho de

conclusão de curso (Graduação e Licenciatura em Sistemas de Informação) - Universidade Estadual do Norte do Paraná. Bandeirantes, 2011.

OLIVEIRA, Kethure Aline; AMARAL, Marília A.; BARTOLO, Viviane Fátima. *Uma experiência para definição de storyboard em metodologia de desenvolvimento colaborativo de objetos de aprendizagem*. Ciências & Cognição: Cien. Cogn, Rio de Janeiro, v. 15, n. 1, p.19-32, 20 abr. 2010. Disponível em: <<http://www.cienciasecognicao.org/revista/index.php/cec/issue/view/14>>. Acesso em: 30 set. 2011.

PASSARINI, R. F. *Objetos de aprendizagem: protótipo para módulo de ambiente de treinamento online*. 2003. 105f. Dissertação (Mestrado em Ciência da Computação), Universidade Federal de Santa Catarina, Florianópolis, 2003.

PESSOA, M. C.; BENITTI, F. B. V.. *Proposta de um processo para produção de objetos de aprendizagem*. Hífen, v. 32, p. 172-180, 2008.

POLSANI, Pithamber R., *Use and Abuse of Reusable Learning Objects*, 2003. Disponível em: <http://journals.tdl.org/jodi/article/viewArticle/89/88>. Acesso em: 02/11/2011.

RAABE, A. L. A.; SILVA, J. M. C. *Um Ambiente para Atendimento as Dificuldades de Aprendizagem de Algoritmos*. 2005. XXV Congresso da Sociedade Brasileira de Computação. São Leopoldo/RS.

RIVED. Rede Internacional Virtual de Educação. Disponível em: http://rived.mec.gov.br/site_objeto_lis.php. Acesso em: 02/11/2011.

ROCHA, Helena.V. *Representações computacionais auxiliares ao entendimento de conceitos de programação*. Campinas: Unicamp, 1991.

ROCHA, P. S.; FERREIRA, B.; MONTEIRO, D.; NUNES, D, S, C, N; GÕES, H, C, N. *Ensino e Aprendizagem de Programação: Análise da Aplicação de Proposta Metodológica Baseada no Sistema Personalizado de Ensino*. Revista Renote, v.8, n.3, 2010.

RODRIGUES, M. *Experiências positivas para o ensino de algoritmos*. IV Escola Regional de Computação Bahia-Sergipe. Feira de Santana, 2004.

SARAIVA, I ; NETTO, C. M. *Monitor: um conjunto de objetos de aprendizagem para apoio ao ensino de programação de computadores*. In: XVIII Workshop sobre Educação em Computação, 2010, Belo Horizonte. Anais do XVIII Workshop sobre Educação em Computação. Porto Alegre : SBC, 2010.

SILVA, A. R. ; GUIMARAES, M. P. ; MOREIRA, D. A. *Uma estratégia colaborativa para complementar o ensino e aprendizagem de lógica de programação*. In: VII Simpósio Brasileiro de Sistemas Colaborativos, 2010, Belo Horizonte. Anais do VII Simpósio Brasileiro de Sistemas Colaborativos, 2010. p. 133-139.

TAROUCO, L. M. R.; FABRE M. J. M.; TAMUSIUNAS, F. R.. *Reusabilidade de objetos educacionais*. RENOTE - Revista Novas Tecnologias na Educação. v.1, n.1, fev. 2003.

TATIZANA, Celso. *Visual Class FX*, Multimídia - Software para Criação. 2006. Disponível em: http://www.classinformatica.com.br/documentos/livro_1.pdf. Acesso em: 07 nov. 2011.

VALENTE, José Armando et al. (Org.). *Computadores na Sociedade do Conhecimento*. Campinas: Nied – Unicamp, 1999.

VASCO, A. M.; AMARAL, M. A.; MARTINS, N. S.; BARTHOLO, V. F.,. "Learning Objects" to Support the Teaching of Science. IEEE LATIN AMERICA TRANSACTIONS, VOL. 9, NO. 3, JUNE 2011. Disponível em: <http://www.ewh.ieee.org/reg/9/etrans/ieee/issues/vol09/vol9issue3June2011/9TLA321MendesVasco.pdf>. Acesso em: 13/06/2012.

XAVIER, C.M.G.; GARCIA, D.F; SILVA, Glaucimara Freitas e. ; SANTOS, A.C.S. *Estudo dos Fatores que Influenciam a Aprendizagem Introdutória de Programação*, 2004. Disponível em: <http://www.uefs.br/erbase2004/documentos/weibase/Weibase2004Artigo002.pdf>. Acesso em: 23/10/2011.

WILEY, D. Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. In D. A. Wiley (Ed.), *The Instructional Use of Learning Objects*, 2000. Disponível em: <http://reusability.org/read/>. Acesso em: 31/10/2011.

APÊNDICE A

Questionário sobre o ensino/aprendizagem de programação

Observações importantes:

- Quando duas ou mais alternativas forem válidas para as perguntas que requerem uma única resposta, marque a alternativa que mais se aplica ao seu caso.

- As avaliações não serão analisadas individualmente, deste modo, seus dados serão tratados com total sigilo.

Os resultados totalizados serão utilizados para fins acadêmicos e servirão de base para o desenvolvimento de uma ferramenta voltado para o ensino dos conceitos de programação, contribua para a pesquisa sendo sincero.

1. Qual a sua faixa de idade?

- 17 - 20 anos.
- 20 - 25 anos.
- 25 - 30 anos.
- 30 - 40 anos.
- acima de 40 anos.

2. Qual o motivo que o levou a fazer este curso ?

- Falta de opção.
- Exigência dos pais.
- Gosto pela Informática.
- Nunca pensei a respeito.

3. Qual o seu compromisso diário além do curso de Sistemas de Informação?

- Ajudo em casa.
- Faço outro faculdade.
- Não faço mais nada.
- Trabalho período integral.
- Trabalho meio período.

4. Qual a sua experiência com Programação?

- Conheci na faculdade;
- Já havia feito outros cursos sobre o assunto;
- Trabalho na área.

5. Quanto ao seu conhecimento sobre programação, classifique-a como:

- Excelente.
- Bom;
- Médio;
- Baixo;

6. Quais os conceitos que você ainda não conhece muito bem ou ainda não se sente seguro com relação ao mesmo? Caso você não saiba do que se trata assinale-o.

- Algoritmo.
- Variáveis e tipos de dados.
- Operadores relacionais.
- Estrutura de Seleção.
- Vetores.
- Funções e Procedimentos.
- Sintaxe.
- Raciocínio Lógico.
- Operadores aritméticos.
- Operadores Lógicos.
- Estrutura de Repetição.
- Matrizes.
- Biblioteca.
- Semântica.

7. Como você estuda e se prepara para a disciplina de programação.

- Não estudo nada, só participo das aulas.
- Apenas na escola durante as aulas ou com os amigos.
- Apenas em vésperas de provas.
- Estudo um pouco em casa (menos de 2 horas por semana)
- Dedico bastante esforços para a disciplina (mais de 2 horas por semana)

8. Com que frequência estuda em casa por conta própria.

- Não costumo estudar em casa.
- Somente quando tenho atividades e trabalhos para fazer em casa.
- Tenho o hábito de estudar algumas horas no fim de semana.
- Estudo pelo menos meia hora por dia.

9. Assinale a alternativa que representa o ambiente de sua casa.

- Não tenho computador.
- Tenho computador mas NÃO tenho acesso a Internet.
- Tenho computador COM acesso a Internet.

10. Além de estudar, o que mais gosta de fazer no seu tempo livre.

- Internet.
- Passear e sair com amigos.
- Não fazer nada.
- ler livros ou tocar instrumentos musical.
- Assistir TV.

11. Como você considera a disciplina de programação.

- Muito difícil.
- Muito legal e divertida.
- Fácil e tranquila.
- Encaro como um desafio a ser vencido.
- Média, nem fácil e nem difícil.

12. Qual a importância que você atribui à disciplina de programação no curso que você realiza.

- Extremamente importante para o bom aproveitamento do curso.
- Importante mas não influencia o decorrer do curso.
- Com a mesma importância das demais disciplinas.
- Menos importante que as demais disciplinas.

13. Assinale as opções que você acredita ser uma boa abordagem para aprender os conceitos de programação.

- Leitura de livros e apostilas.
- Boa explicação.
- Exercícios.
- Software de computadores para o ensino de programação.
- Jogos de raciocínio lógico.

14. Você está de DP na disciplina de programação.

- Sim.
- Não.

APÊNDICE B

Questionário sobre o ensino/aprendizagem de programação

- Os resultados serão utilizados para fins acadêmicos;
- Os resultados serão analisados individualmente, deste modo, suas informações serão tratadas com total sigilo;

- 1) Qual sua visão do perfil dos alunos ingressantes no curso de Sistemas de Informação?**

- 2) Quais as principais dificuldades encontradas (caso exista dificuldades) no ensino da disciplina de programação?**

- 3) Acredita na possibilidade de mudar o cenário atual local e se vê este problema como algo específico ou genérico?**

- 4) No tempo em que você cursou a graduação teve dificuldades na época com os conceitos de programação? E seus amigos?**

- 5) Quais as alternativas adotadas/encontradas/desejáveis para resolver ou diminuir as dificuldades de aprendizado do aluno?**

- 6) Tempo de experiência de docência e quais outras experiências relacionadas com o assunto.**

APÊNDICE C

Critérios Técnico-Pedagógicos Parte 1	
<p>Para avaliação dos <i>softwares</i> é necessário, que o avaliador responda a todas as questões consideradas como aplicáveis. O <i>software</i> pode atender à questão, atender parcialmente ou não atender. Para cada uma dessas respostas sugere-se um valor, conforme descrito abaixo.</p>	
Respostas das Questões:	
Sim -1.0	
Não - 0.0	
Parcialmente - 0.5	
<p>A Conformidade de cada critério se calcula pela soma dos resultados das perguntas, divide pelo total de questões de cada critério e multiplica por 100.</p> <p>Conformidade (critério) = $(\sum \text{res}/n^\circ \text{questões}) * 100$</p> <p>Critério de avaliação do ponto de vista do usuário.</p>	
Contexto da Análise.	
Duração: 3 horas aulas	
Data: 19/04/12	

Critério	Questões	S	P	N
Documentação	1. O programa tem manual de apoio?	27	2	0
	2. Está bem concebido graficamente?	18	11	0
	3. Contém informações úteis para o utilizador?	28	1	0
	4. Apresenta os passos para a instalação e desinstalação do produto?	19	5	5
	Conformidade	87,50%		

Critério	Questões	S	P	N
Informações Pedagógicas - Observadas pelo Avaliador	1. Vocabulário adequado?	29	0	0
	2. Ortografia e clareza de idéias?	25	4	0
	3. O conteúdo é atualizado?	27	2	0
	4. O <i>software</i> permite a identificação do ambiente educacional e do modelo de aprendizagem que ele privilegia?	25	4	0
	5. O <i>software</i> é adequado e pertinente a um dado contexto educacional ou disciplina específica?	27	2	0

6. O <i>software</i> atende a um objetivo educacional e para isso é fácil de usar, é amigável ao usuário, possui aspectos motivacionais e respeita as individualidades?	23	6	0
7. Inclui atributos como: conteúdos claros e corretos, recursos motivacionais, carga informacional e tratamento de erros?	24	5	0
Conformidade	94,33%		

Critério	Questões	S	P	N
Aspectos Didáticos e de Conteúdo	1. Inclui atributos como: conteúdos claros e corretos, recursos motivacionais, carga informacional e tratamento de erros?	24	5	0
	2. Os conceitos disponibilizaram exemplos, gráficos, imagens?	28	1	0
	3. O conteúdo apresentava congruência com as atividades propostas, os recursos ou serviços disponibilizados para tal?	25	4	0
	4. A redação do texto do conteúdo do programa estava adequada ao tipo e tempo de leitura?	24	4	1
	5. A forma de exposição das telas foi adequada para a leitura e/ou compreensão dos conteúdos?	27	2	0
	6. O vocabulário estava adequado ao público alvo?	26	3	0
	7. Os títulos de módulos e aulas estavam bem definidos e distribuídos?	25	4	0
	8. Os textos das aulas apresentavam boa ortografia, formulação e estrutura correta das frases proporcionando uma leitura confortável?	28	1	0
	9. O nível de linguagem e do conteúdo era apropriado ao nível de compreensão do aluno?	25	4	0
	10. Os tópicos de conteúdo apresentaram níveis crescentes de dificuldades para a assimilação de novos conhecimentos?	22	6	1
	11. As seqüências ou tópicos de apresentação do conteúdo foram variados?	23	6	0
	12. Os tópicos tratados eram inovadores e atuais na área?	21	6	2
	13. Cada etapa vencida constituiu-se como subestrutura para a etapa seguinte?	27	2	0
	14. O ambiente facilitou seu próprio estilo de aprendizagem?	20	9	0
	15. O ambiente permitiu o aprendizado num ritmo apropriado ao seu tempo?	23	5	1
	16. O ambiente/curso mostrou elementos motivadores?	20	7	2

17. O ambiente permitiu o desenvolvimento da aprendizagem de forma autônoma?	22	5	2
18. O ambiente possibilitou o aluno ser ativo no processo ensino-aprendizagem?	25	4	0
19. O ambiente estimulou o aluno a procurar recursos externos, tais como <i>links</i> , livros, artigos e outras fontes?	17	9	3
20. O ambiente proporcionou aprender em cooperação com colegas e professores?	23	3	3
21. O ambiente apresentou diferentes graus de dificuldade nas questões apresentadas?	23	5	1
22. As atividades apresentaram um bom grau de coerência entre os resultados alcançados e os objetivos propostos?	27	2	0
23. As atividades foram objeto de cobrança de prazo de entrega ou resposta?	21	1	7
24. O seu aprendizado foi individualmente adaptado para o ritmo de sua evolução no curso?	22	5	2
25. O ambiente motivou você a continuar?	24	3	2
26. As perguntas e respostas armazenadas no FAQ serviram as suas necessidades?	22	7	0
27. A navegação no ambiente/curso era fácil e rápida?	27	2	0
28. O <i>site</i> e as diferentes páginas do curso eram rápidas de carregar?	26	3	0
29. Os botões, os ícones e as metáforas eram significativos e eficazes à navegação?	19	10	0
30. Um atalho à página inicial do ambiente está sempre disponível em todas as páginas do curso?	21	4	4
31. Foi permitida a escolha de caminhos alternativos para aprendizagem em função de seu grau de dificuldade?	15	6	8
32. Foi permitida a escolha aleatória na seqüência das atividades?	17	4	8
33. Teve a clara informação da sua evolução no curso?	22	5	2
34. O ambiente permitiu que você pudesse retornar novamente ao ambiente no ponto em que atingiu no seu acesso anterior ou a qualquer ponto que queira visitar?	22	5	2
35. Exercícios que favoreceram o raciocínio lógico?	26	3	0
36. Exercícios que favoreceram o raciocínio numérico?	21	7	1
37. Exercícios que favoreceram o raciocínio verbal?	19	8	2
38. Exercícios de associação de elementos?	23	5	1
39. Exercícios de Interpretação de texto?	26	3	0
40. Elaboração de esquemas, esboços e projetos?	23	5	1
41. Exercícios de estimulação da memória?	28	0	1
42. Exercícios de solução de problemas?	25	3	1

	43. Exercícios de formulação e teste de hipóteses?	26	2	1
	44. Exercícios de experimentação?	27	2	0
	Conformidade	87,93%		

Critério	Questões	S	P	N
Aspectos Emocionais e Afetivos	1. No caso de erro ou acerto, o retorno da resposta (<i>feedback</i>) considerava o evento normalmente sem a atribuição de pressão ou carga emocional?	20	8	1
	2. As cobranças de atividades e prazos evitaram estimular sensações de medo ou ansiedade?	20	7	2
	3. O conteúdo evitou proporcionar a geração de expectativa exagerada (<i>suspense</i>)?	19	8	2
	4. O ambiente evitou trabalhar a opinião do usuário baseado em questões sócio políticas e religiosas?	25	3	1
	5. O ambiente evitou promover a sensação de frustração por dificuldade de manuseio do programa?	25	4	0
	Conformidade	85,52%		

Critério	Questões	S	P	N
Aspectos Cognitivos	1. O curso apresentou uma carga de informação de conteúdo teórico bem distribuído e adequado?	26	3	0
	2. O curso apresentou uma carga de atividades e trabalho prático adequada?	27	2	0
	3. O ambiente exigiu de você um nível confortável de atenção e concentração?	22	6	1
	4. As questões formuladas, bem como o conteúdo do programa, estão em nível de dificuldade aceitável?	17	11	1
	5. Os exercícios ou atividades promoveram a lembrança de idéias, materiais ou fenômenos já conhecidos?	25	4	0
	6. O curso exigiu a evocação de conhecimento prévio específico para a compreensão de seu conteúdo?	20	7	2
	7. Os exercícios promoveram a aplicação do conhecimento adquirido?	22	7	0
	8. O ambiente ofereceu a possibilidade de trabalhar com elementos, partes e combiná-los favorecendo a percepção de sua configuração ou estrutura?	17	11	1

	9. O ambiente disponibilizou algum tipo de processo de julgamento acerca do valor de idéias, trabalhos, soluções, métodos, materiais etc., fazendo um julgamento qualitativo e quantitativo?	22	6	1
	Conformidade	86,78%		

Conformidade dos Critérios Avaliados	
Documentação	87,50%
Informações Pedagógicas - Observadas pelo Avaliador	94,33%
Aspectos Didáticos e de Conteúdo	87,93%
Aspectos Emocionais e Afetivos	85,52%
Aspectos Cognitivos	86,78%

Porcentagem Mínima para que um Critério seja Aceito	
Documentação	50,00%
Informações Pedagógicas - Observadas pelo Avaliador	70,00%
Aspectos Didáticos e de Conteúdo	70,00%
Aspectos Emocionais e Afetivos	60,00%
Aspectos Cognitivos	70,00%

Verificação se o Software foi Aceito ou Não Aceito	
Documentação	ACEITO
Informações Pedagógicas - Observadas pelo Avaliador	ACEITO
Aspectos Didáticos e de Conteúdo	ACEITO
Aspectos Emocionais e Afetivos	ACEITO
Aspectos Cognitivos	ACEITO

Critérios Técnico-Pedagógicos Parte 2

Para avaliação dos *softwares* é necessário, que o avaliador responda a todas as questões consideradas como aplicáveis. O software pode atender à questão, atender parcialmente ou não atender. Para cada uma dessas respostas sugere-se um valor, conforme descrito na tabela 01.

Resposta da Questão
Sim - 1.0
Não - 0.0
Parcialmente - 0.5

Tabela 01: Tabela de pesos das questões.

A Conformidade de cada critério se calcula pela soma dos resultados das perguntas, divide pelo total de questões de cada critério e multiplica por 100.

$$\text{Conformidade (critério)} = (\sum \text{res}/n^{\circ} \text{questões}) * 100.$$

Critério	Questões	S	P	N
Pertinência em relação a uma proposta pedagógica	1. Permite a relação do ciclo descrição – execução – reflexão – depuração – descrição (<i>Software</i> Construtivista)?	23	6	0
	2. O <i>software</i> é adequado e pertinente em relação a uma disciplina específica?	27	2	0
	3. O <i>software</i> realmente auxilia os alunos na aquisição das habilidades e dos conteúdos propostos?	25	4	0
	4. Os docentes da escola teriam facilidade em adotar o <i>software</i> como parte das suas atividades?	22	5	2
	5. O objetivo do <i>software</i> é claro?	26	3	0
	Conformidade		91,72%	

Critério	Questões	S	P	N
Utilização de recursos computacionais	1. O mesmo conteúdo do <i>software</i> seria dificilmente ensinado sem o uso do recurso tecnológico do computador?	17	8	4
	2. Apresenta informações de forma interativa?	24	4	1
	3. Age com diversos níveis de inteligência adquirida?	23	6	0
	4. Viabiliza diferentes níveis de interação?	23	6	0
	5. Utiliza recursos motivacionais?	20	9	0

	6. Faz conexões com outros meios de aprendizagem?	24	5	0
	Conformidade	86,21%		

Critério	Questões	S	P	N
Avaliação da aprendizagem	1. O software possibilita algum recurso que permite avaliar o grau de compreensão dos alunos na resolução de problemas?	23	6	0
	2. O <i>software</i> armazena informações relativas à interação dos alunos tais como pontuações, níveis atingidos ou dificuldades enfrentadas?	23	5	1
	3. O <i>software</i> permite a impressão dos registros de desempenho dos alunos?	20	5	4
	Conformidade	85,06%		

Critério	Questões	S	P	N
Interação	1. O aluno encontra disponíveis meios (mensagens, alarmes) para aconselhá-lo, orientá-lo ou informá-lo na sua interação com o programa?	23	6	0
	2. O aluno sabe a qualquer momento se localizar numa sequência de interação ou na execução de uma tarefa?	28	1	0
	3. O aluno conhece as ações permitidas bem como suas consequências?	22	7	0
	4. O aluno pode obter informações suplementares (eventualmente a seu pedido)?	21	6	2
	5. O <i>software</i> informa os resultados do estado da ação, de forma que ele possa acompanhar a evolução do processamento da informação, usando recursos como, por exemplo, ampulhetas, relógio, barra de progressão?	21	6	2
		88,28%		
Interação – Reconhecimento no lugar de memorização	6. Objetos, ações e opções estão visíveis?	28	1	0
	7. O <i>software</i> minimiza a necessidade do aluno lembrar dados exatos de uma tela para outra?	22	6	1
		92,24%		
Interação- Qualidade das opções de ajuda	8. O <i>software</i> disponibiliza comandos de ajuda?	18	7	4
	9. Os dispositivos de ajuda abrangem a totalidade do sistema?	19	7	3

	10. O acionamento da opção de ajuda está estruturado no contexto da tarefa e da transação corrente?	22	5	2
	11. O <i>software</i> apresenta diferentes formas de acesso aos conteúdos de ajuda?	18	9	2
	14. Os diálogos/textos são pequenos?	19	8	2
		78,62%		
Interação – Legibilidade	12. O vocabulário usado propõe uma interpretação única no seu significado?	25	3	1
	13. Os diálogos/textos são de fácil legibilidade?	25	3	1
	14. Os diálogos/textos são pequenos?	24	2	3
	15. Os ícones são representativos das suas funções?	26	2	1
		90,52%		
Interação - <i>Feedback</i>	16. O <i>software</i> fornece <i>feedback</i> imediato de todas as entradas de dados do aluno?	24	5	0
	17. O <i>software</i> emite <i>feedback</i> encorajador, variado e isento de carga negativa mediante respostas inadequadas?	21	4	4
	18. O tempo de resposta do <i>software</i> é adequado?	26	3	0
	19. O usuário tem sempre a possibilidade de obter um <i>feedback</i> informando sobre o sucesso ou o fracasso da operação?	24	5	0
	20. O <i>software</i> fornece <i>feedback</i> sobre as mudanças de atributos dos objetos de interação, ou seja, ao selecionar um botão, ele muda de estado?	21	5	3
		87,59%		
Interação – Agrupamento e distinção de itens	21. O <i>software</i> apresenta distinção visual clara de áreas que possuem diferentes funções?	22	7	0
	22. Os dados que requerem atenção imediata são diferenciados de alguma forma?	21	5	3
	23. Nos agrupamentos de dados e informações, os itens estão organizados segundo critério lógico e facilitador?	21	7	1
	24. A informação é apresentada numa organização direcionada para o aluno?	21	7	1
		84,48%		
	Conformidade	86,95%		

Critério	Questões	S	P	N
Adaptabilidade – Flexibilidade	1. O <i>software</i> abrange alunos em vários níveis de conhecimento?	21	8	0
	2. O <i>software</i> permite que o acesso a qualquer nível seja feito diretamente, sem ter que passar pelos anteriores?	22	5	2
	3. O <i>software</i> permite diferentes formas para a realização de tarefas?	22	5	2
	4. O <i>software</i> permite a flexibilidade de ações (para <i>software</i> construtivista)?	20	8	1
	5. O <i>software</i> propõe formas variadas de apresentação das mesmas informações a diferentes alunos?	18	7	4
	6. O docente/pai pode alterar algumas configurações do <i>software</i> , adequando-o ao seu interesse?	17	7	5
	7. Os alunos têm a possibilidade de modificar ou eliminar itens?	11	7	11
	8. O <i>software</i> permite que usuários com algum tipo de deficiência (auditiva, visual) possa efetuar as atividades propostas?	18	7	4
	9. Caso o usuário tenha problemas visuais, o <i>software</i> possui sinais sonoros para poder orientar o usuário a fazer suas tarefas?	20	6	3
	10. Em caso do usuário ter problemas auditivos, o <i>software</i> é bem animado, tem caixa de diálogos explicando o que o usuário terá que fazer naquele momento, para que ele possa fazer as atividades do <i>software</i> sem nenhum problema?	22	3	4
Adaptabilidade – Nível de experiência	11. A sequência de apresentação dos conceitos evolui em grau de complexidade?	22	6	1
	12. O <i>software</i> abrange alunos que não tem familiaridade com computadores?	18	6	5
	13. O <i>software</i> possui atalhos?	21	6	2
	14. O <i>software</i> possibilita efetuar alterações em suas estruturas de modo a contemplar a experiência do aluno?	19	7	3
Adaptabilidade – Ambiente cooperativo	15. O <i>software</i> pode em vez de ordenar, sugere em vez de exigir ou persuade em vez de controlar?	21	6	2

16. O <i>software</i> possibilita que o(a) docente possa expandir o conteúdo do programa em seu trabalho?	20	7	2
Conformidade	78,13%		

Critério	Questões	S	P	N
Controle e autonomia do usuário	1. O processamento das ações é efetuado somente quando solicitado pelo aluno?	25	4	0
	2. O aluno pode controlar o ritmo das tarefas?	17	10	2
	3. Há possibilidade de acesso a todas as informações a todo momento (salvo em casos contra uma necessidade do conteúdo)?	22	5	2
	4. <i>Software</i> permite que o aluno abandone o programa a qualquer momento ou ir para outro item do menu?	20	6	3
	5. O <i>software</i> sinaliza as consequências da atividade do menu?	24	4	1
	6. <i>Software</i> permite que o aluno anule a última ação realizada?	19	4	6
	Conformidade	82,47%		

Critério	Questões	S	P	N
Recursos Motivacionais	1. O <i>software</i> possui recursos motivacionais para despertar e manter a atenção do aluno ao longo da interação?	18	9	2
	2. Os recursos motivacionais utilizados permanecem interessantes ao longo do tempo, sem tornarem-se irritantes?	19	8	2
	3. <i>Software</i> proporciona desafio ao aluno?	25	4	0
	4. São utilizados recursos sonoros para contribuir para a motivação?	23	6	0
	5. Os recursos sonoros são opcionais?	19	6	4
	6. O <i>software</i> usa de temporarização com cautela?	18	9	2
	7. O <i>software</i> estimula a imaginação do aluno, por meio do contexto ou simulação?	19	10	0
	8. Os recursos propostos pelo <i>software</i> podem contribuir para a melhoria do relacionamento professor/aluno e colegas/aluno?	21	7	1

	9. O aluno tem vontade de jogar de novo?	17	7	5
	Conformidade	81,23%		

Critério	Questões	S	P	N
Gestão de erros	1. <i>Software</i> previne a ocorrência de erros, por meio de um desenho cuidadoso?	17	12	0
	2. Na ocorrência de erros na resolução dos exercícios propostos, o <i>software</i> orienta e oferece ao aluno a possibilidade de tentar refazê-lo?	21	8	0
	3. Persistindo o erro durante uma tarefa, o <i>software</i> conduz o aluno, fornecendo-lhe explicações para a correção?	22	6	1
	4. O <i>software</i> fornece a solução após longa persistência do erro?	14	11	4
	5. O <i>software</i> permite a mudança automática de exercícios se o aluno persiste no erro, conduzindo-a a outro tipo de exercício, com menor grau de dificuldade?	16	7	6
	6. As mensagens de erro são assinaladas imediatamente ou o mais rápido possível?	25	4	0
	7. As mensagens de erro são expressas em linguagem simples (sem códigos), de uso comum e são curtas?	22	6	1
	8. As mensagens de erro adotam vocabulário neutro, não repreensivo e evitam o humor?	16	10	3
	9. As mensagens de erro são positivas e sugerem construtivamente uma solução ?	21	6	2
	10. As mensagens de erro indicam precisamente o problema?	21	5	3
	11. Perante uma dificuldade na resolução dos exercícios, o <i>software</i> evita a monotonia, oferecendo mensagens de erro variadas?	15	9	5
	12. O <i>software</i> permite que o aluno modifique o erro, corrigindo-o parcialmente ou totalmente?	19	6	4
	Conformidade	78,74%		

Critério	Questões	S	P	N
Carga de Trabalho	1. As informações evitam a poluição visual?	21	5	3
	2. Informações dos diálogos/textos são sempre relevantes?	17	8	4
	3. Somente as informações necessárias e utilizáveis são apresentadas?	20	4	5
	4. As tarefas são feitas em passos de acordo com o objetivo proposto?	22	4	3
	Conformidade	78,02%		

Critério	Questões	S	P	N
Conteúdo	1. O conteúdo do <i>software</i> é significativo para o aluno?	19	9	1
	2. O conteúdo do <i>software</i> educacional é rico em dimensões exploratórias?	16	11	2
	3. O conteúdo do <i>software</i> possibilita que o(a) docente possa explorá-lo durante a interação dos alunos com o programa?	17	9	3
	4. O conteúdo do <i>software</i> se relaciona com as vivências dos alunos, especialmente com a realidade brasileira?	18	10	1
	5. As explicações sobre um tema são opcionais?	16	9	4
	6. Novas palavras são explicadas?	15	11	3
	Conformidade	75,00%		

Critério	Questões	S	P	N
Significado de códigos e denominações	1. Os códigos e denominações estão de acordo com o que representam?	21	6	2
	2. É evitado o uso de siglas ou abreviaturas?	22	6	1
	3. As palavras, situações, objetos ou ações diferentes no <i>software</i> significam sempre a mesma coisa?	16	6	7
	Conformidade	78,16%		

Critério	Questões	S	P	N
Consistência e padrões	1. Os procedimentos, comandos ou botões do <i>software</i> têm formato estáveis?	22	6	1
	2. A posição ou sintaxe dos botões são compreendíveis pelos alunos?	23	4	2
	Conformidade	86,21%		

Critério	Questões	S	P	N
Correspondência com o mundo real	1. O <i>software</i> segue as convenções do mundo real?	21	6	2
	2. O <i>software</i> usa palavras, expressões e conceitos familiares ao aluno, no lugar de termos orientados para o programa?	23	6	0
	3. O <i>software</i> simula ambiente familiares aos alunos ou utiliza metáforas conhecidas por elas?	20	7	2
	4. O <i>software</i> segue convenções oriundas de outros programas populares, como jogos existentes no mercado?	22	5	2
	Conformidade	84,48%		

Critério	Questões	S	P	N
Documentação	1. O <i>software</i> fornece documentação direcionada tanto para pais/docentes quanto para os alunos?	21	5	3
	2. A documentação está acessível a todo momento para o usuário?	23	3	3
	3. A documentação aparece somente quando solicitada?	23	5	1
	4. A identificação da faixa etária a que se destina é identificada?	17	5	7
	5. A documentação de descrição do produto é compreensível, completa e está livre de inconsistências internas?	18	9	2
	6. É orientada para docentes/pais?	24	3	2
	7. Inclui dados sobre a utilização do <i>software</i> (regras) direcionadas a docentes/pais?	20	5	4
	8. É de fácil pesquisa?	23	2	4
	9. Focaliza a tarefa do usuário?	22	6	1
	10. Os termos técnicos ou abreviações são explicados?	20	6	3

11. As informações estão bem-organizadas e distribuídas?	24	5	0
12. Mantém a consistência e a homogeneidade em todas as suas ocorrências?	21	7	1
13. Existe documentação (regra direcionada para o aluno)?	20	8	1
14. Usa linguagem familiar ao aluno?	22	6	1
15. Estão visíveis ou facilmente recuperáveis?	22	6	1
16. São curtas?	15	8	6
17. São compreensíveis?	27	2	0
18. São transmitidas por animações?	16	13	0
19. Se forem escritas, são legíveis e sem erros?	25	4	0
20. Se forem escritas, são bem-organizadas e distribuídas?	25	5	0
21. Se forem escritas, estão disponíveis aos alunos?	22	7	0
Conformidade	83,74%		

Crítérios	Conformidade
1. Pertinência em relação a uma proposta pedagógica	91,72%
2. Interação	86,95%
3. Recursos motivacionais	81,23%
4. Gestão de erros	78,74%
5. Adaptabilidade	78,13%
6. Correspondência entre o <i>software</i> e o mundo real	84,48%
7. Conteúdo	75,00%
8. Avaliação da aprendizagem	85,06%
9. Controle e autonomia do usuário	82,47%
10. Utilização de recursos computacionais	86,21%
11. Carga de trabalho	78,02%
12. Significado de código e denominações	78,16%
13. Consistência e padrões	86,21%
14. Documentação	83,74%
Média de todos os critérios	82,58%

Porcentagem Mínima para que um Critério seja Aceito	
1. Pertinência em relação a uma proposta pedagógica	70%
2. Interação	50%
3. Recursos motivacionais	60%
4. Gestão de erros	50%
5. Adaptabilidade	60%
6. Correspondência entre o <i>software</i> e o mundo real	50%
7. Conteúdo	70%
8. Avaliação da aprendizagem	40%
9. Controle e autonomia do usuário	70%
10. Utilização de recursos computacionais	50%
11. Carga de trabalho	60%
12. Significado de códigos e denominações	60%
13. Consistência e padrões	50%
14. Documentação	40%

Verificação se o Critério foi Aceito ou Não Aceito	
1. Pertinência em relação a uma proposta pedagógica	ACEITO
2. Interação	ACEITO
3. Recursos motivacionais	ACEITO
4. Gestão de erros	ACEITO
5. Adaptabilidade	ACEITO
6. Correspondência entre o <i>software</i> e o mundo real	ACEITO
7. Conteúdo	ACEITO
8. Avaliação da aprendizagem	ACEITO
9. Controle e autonomia do usuário	ACEITO
10. Utilização de recursos computacionais	ACEITO
11. Carga de trabalho	ACEITO
12. Significado de códigos e denominações	ACEITO
13. Consistência e padrões	ACEITO
14. Documentação	ACEITO

CRITÉRIO USABILIDADE

A conformidade de cada critério será calculada seguindo os pesos da tabela

Resposta das questões	Peso
Sim (S)	1.0
Parcialmente (P)	0.5
Não (N)	0.0

Fórmula utilizada: Conformidade (critérios) = $(\sum \text{res}/n^{\circ} \text{questões}) * 100$

Critérios	Questões	S	P	N
Presteza	1. Os títulos de telas, janelas e caixas de diálogo estão no alto, centrados ou justificados à esquerda?	20	6	3
	2. Todos os campos e mostradores de dados possuem rótulos identificativos?	21	6	2
	3. Caso o dado a entrar possua um formato particular, esse formato encontra-se descrito na tela?	21	6	2
	4. As unidades para a entrada ou apresentação de dados métricos ou financeiros encontram-se descritas na tela?	20	7	2
	5. Os rótulos dos campos contêm um elemento específico, por exemplo, como convite às entradas de dados?	20	6	3
	6. Caso o dado a entrar possua valores aceitáveis esses valores encontram-se descritos na tela?	20	7	2
	7. Listas longas apresentam indicadores de continuação, e quantidade de itens e de páginas?	22	7	0
	8. As tabelas apresentam cabeçalhos para linhas e colunas consistentes e distinguíveis dos dados apresentados?	23	6	0
	9. Os gráficos possuem um título geral e rótulos para seus eixos?	20	7	2
	10. Os botões que comandam a apresentação de caixas de diálogo apresentam em seus rótulos o sinal "... " como indicador da continuidade do diálogo?	20	6	3
	11. As páginas de menus possuem títulos, cabeçalhos ou convites à entrada?	21	4	4
	12. As opções de menu que levam a outros painéis de menu apresentam algum sinal como indicador desse fato?	21	6	2
	13. O usuário encontra disponíveis as informações necessárias para suas ações?	19	7	3

	14. Nas caixas de mensagens de erro, o botão de comando "AJUDA" está sempre presente?	15	6	8
	15. A resposta para uma solicitação de ajuda do usuário está estruturada no contexto da tarefa e da transação corrente?	21	3	5
	16. Existe a possibilidade do usuário obter a lista de comandos básicos da linguagem?	19	5	5
	17. Na ocorrência de erros, o usuário pode acessar todas as informações necessárias ao diagnóstico e à solução do problema?	21	7	1
Conformidade		80,12%		

Critérios	Questões		P	N
Agrupamento por localização	1. O espaço de apresentação está diagramado em pequenas zonas funcionais?	21	7	1
	2. A disposição dos objetos de interação de uma caixa de dialogo segue uma ordem lógica?	24	5	0
	3. Nos agrupamentos de dados, os itens estão organizados espacialmente segundo um critério lógico?	19	9	1
	4. Os códigos das teclas aceleradoras de opções de menu estão localizados à direita do nome da opção?	19	8	2
	5. Nas listas de seleção, as opções estão organizadas segundo alguma ordem lógica?	21	7	1
	6. Os painéis de menus são formados a partir de um critério lógico de agrupamento de opções?	18	10	1
	7. Dentro de um painel de menu, as opções mutuamente exclusivas ou interdependentes estão agrupadas e separadas das demais?	22	5	2
	8. As opções dentro de um painel de menu estão ordenadas segundo algum critério lógico?	20	7	2
	9. A definição da opção de menu selecionada por <i>default</i> segue algum critério?	21	6	2
	10. Os grupos de botões de comando estão dispostos em coluna e à direita, ou em linha e abaixo dos objetos aos quais estão associados?	23	4	2
	11. O botão de comando selecionado por <i>default</i> está na posição mais alta, se os botões estão dispostos verticalmente, ou na mais à esquerda, se os botões estão dispostos horizontalmente?	19	9	1
Conformidade		83,23%		

Crítérios	Questões	S	P	N
Agrupamento por formato	1. Os controles e comandos encontram-se visualmente diferenciados das informações apresentadas nas telas?	21	6	2
	2. Códigos visuais são empregados para associar diferentes categorias de dados distribuídos de forma dispersa nas telas?	23	4	2
	3. Os diferentes tipos de elementos de uma tela de consulta (dados, comandos e instruções) são visualmente distintos uns dos outros?	23	3	3
	4. Os rótulos são visualmente diferentes dos dados aos quais estão associados?	17	10	2
	5. Os cabeçalhos de uma tabela estão diferenciados através do emprego de cores diferentes, letras maiores ou sublinhadas?	20	6	3
	6. Em situações anormais, os dados críticos e que requeiram atenção imediata são diferenciados através do uso de cores brilhantes como por exemplo, o vermelho ou o rosa?	17	9	3
	7. Sinais sonoros são empregados para alertar os usuários em relação a uma apresentação visual?	21	7	1
	8. Na apresentação de textos, os recursos de estilo, como itálico, negrito, sublinhado ou diferentes fontes são empregados para salientar palavras ou noções importantes?	23	6	0
	9. Os itens selecionados para alteração, atualização ou acionamento estão destacados dos outros?	19	9	1
	10. Nas situações de alarme e nas telas de alta densidade de informação, o recurso de intermitência visual é empregado para salientar dados e informações?	19	10	0
	11. Os campos obrigatórios são diferenciados dos campos opcionais de forma visualmente clara?	21	8	0
	12. Nas caixas de mensagens, o botão selecionado por <i>default</i> tem uma apresentação visual suficientemente distinta dos outros?	19	9	1
	13. Em situações em que se exija atenção especial do usuário, as mensagens de alerta e de aviso são apresentadas de maneira distinta?	22	5	2

	14. A forma do cursor do <i>mouse</i> é diferente da de qualquer outro item apresentado?	19	7	3
	15. As formas de cursores (dois ou mais) apresentados simultaneamente são suficientemente distintas umas das outras?	18	9	2
	16. As caixas de agrupamento são empregadas para realçar um grupo de dados relacionados?	18	10	1
	17. Quando apresenta opções não disponíveis no momento, o sistema as mostra de forma diferenciada visualmente?	19	7	3
Conformidade		81,44%		

Crítérios	Questões	S	P	N
Feedback	1. O sistema fornece <i>feedback</i> para todas as ações do usuário?	19	8	2
	2. Quando, durante a entrada de dados, o sistema torna-se indisponível ao usuário, devido a algum processamento longo, o usuário é avisado desse estado do sistema e do tempo dessa indisponibilidade?	17	8	4
	3. O sistema fornece informações sobre o estado das impressões?	21	3	5
	4. Os itens selecionados de uma lista são realçados visualmente de imediato?	22	5	2
	5. A imagem do cursor fornece <i>feedback</i> dinâmico e contextual sobre a manipulação direta?	25	3	1
	6. O sistema fornece ao usuário informações sobre o tempo de processamentos demorados?	23	2	4
	7. O sistema apresenta uma mensagem informando sobre o sucesso ou fracasso de um processamento demorado?	20	8	1
	8. O sistema fornece <i>feedback</i> imediato e contínuo das manipulações diretas?	22	6	1
	9. O sistema define o foco das ações para os objetos recém criados ou recém abertos?	23	5	1
	10. O sistema fornece <i>feedback</i> sobre as mudanças de atributos dos objetos?	23	3	3
	11. Qualquer mudança na situação atual de objetos de controle é apresentada visualmente de modo claro ao usuário?	23	6	0
	12. O sistema fornece um histórico dos comandos entrados pelo usuário durante uma sessão de trabalho?	22	3	4

Conformidade	83,33%
---------------------	---------------

Critérios	Questões	S	P	N
Legibilidade	1. As áreas livres são usadas para separar grupos lógicos em vez de tê-los todos de um só lado da tela, caixa ou janela?	22	7	0
	2. Os grupos de objetos de controle e de apresentação que compõem as caixas de diálogo e outros objetos compostos encontram-se alinhados vertical e horizontalmente?	18	10	1
	3. Os rótulos de campos organizados verticalmente e muito diferentes em tamanho estão justificados à direita?	21	7	1
	4. A largura mínima dos mostradores de texto é de 50 caracteres?	13	15	1
	5. A altura mínima dos mostradores de texto é de 4 linhas?	15	11	3
	6. Os parágrafos de texto são separados por, pelo menos, uma linha em branco?	19	8	2
	7. O uso exclusivo de maiúsculas nos textos é evitado?	18	8	3
	8. O uso do negrito é minimizado?	14	12	3
	9. O uso do sublinhado é minimizado?	13	13	3
	10. Nas tabelas, linhas em branco são empregadas para separar grupos?	18	8	3
	11. As listas de dados alfabéticos são justificadas à esquerda?	21	6	2
	12. As listas contendo números decimais apresentam alinhamento pela vírgula?	15	13	1
	13. As linhas empregadas para o enquadramento e segmentação de menus (separadores, delimitadores etc.) são simples?	23	5	1
	14. As bordas dos painéis dos menus estão suficientemente separadas dos textos das opções de modo a não prejudicar a sua legibilidade?	22	6	1
	15. O uso de abreviaturas é minimizado nos menus?	20	5	4
	16. Os nomes das opções estão somente com a inicial em maiúsculo?	18	9	2

17. Os números que indicam as opções de menu estão alinhados pela direita?	19	8	2
18. Quando a enumeração alfabética é utilizada, as letras para seleção estão alinhadas pela esquerda?	24	4	1
19. As opções das barras de menu horizontal estão separadas por, no mínimo, 2 caracteres brancos?	19	10	0
20. Os rótulos de campos começam com uma letra maiúscula, e as letras restantes são minúsculas?	18	9	2
21. Os itens de dados longos são particionados em grupos mais curtos, tanto nas entradas como nas apresentações?	20	8	1
22. Os códigos alfanuméricos do sistema agrupam separadamente letras e números?	18	8	3
23. Os ícones são legíveis?	24	4	1
24. O sistema utiliza rótulos (textuais) quando pode existir ambiguidade de ícones?	19	8	2
25. A informação codificada com o vídeo reverso é legível?	18	11	0
26. O uso de vídeo reverso está restrito à indicação de <i>feedback</i> de seleção?	18	7	4
27. Os dados a serem lidos são apresentados de forma contínua, ou seja, não piscantes ?	18	9	2
Conformidade	79,12%		

Critérios	Questões	S	P	N
Concisão	1. O sistema oferece valores <i>defaults</i> para acelerar a entrada de dados?	14	13	2
	2. A identificação alfanumérica das janelas é curta o suficiente para ser lembrada facilmente?	19	9	1
	3. Os nomes das opções de menu são concisos?	20	8	1
	4. Os ícones são econômicos sob o ponto de vista do espaço nas telas?	16	10	3
	5. As denominações são breves?	16	10	3
	6. As abreviaturas são curtas?	12	10	7
	7. Os códigos arbitrários que o usuário deve memorizar são sempre menores do que 4 ou 5 caracteres?	19	7	3
	8. Os rótulos são concisos?	23	4	2
	9. Códigos alfanuméricos não significativos para o usuário e que devem ser entradas no sistema são menores do que 7 caracteres?	15	13	1

10. Na entrada de dados alfanuméricos, o sistema considera as letras maiúsculas e minúsculas como equivalentes?	18	8	3
11. Na entrada de dados numéricos, o usuário é liberado do preenchimento do ponto decimal desnecessário?	18	9	2
12. Na entrada de dados numéricos, o usuário é liberado do preenchimento do zeros fracionários desnecessários?	16	11	2
13. Na entrada de valores métricos ou financeiros, o usuário é liberado do preenchimento da unidade de medida?	20	6	3
14. É permitido ao usuário reaproveitar os valores definidos para entradas anteriores, podendo inclusive alterá-los?	19	9	1
Conformidade	75,99%		

Crítérios	Questões	S	P	N
Ações Mínimas	1. Em formulário de entrada de dados o sistema posiciona o cursor no começo do primeiro campo de entrada?	22	6	1
	2. Na realização das ações principais em uma caixa de diálogo, o usuário tem os movimentos de cursor minimizados através da adequada ordenação dos objetos?	22	6	1
	3. O usuário dispõe de um modo simples e rápido (tecla TAB por exemplo) para a navegação entre os campos de um formulário?	15	11	3
	4. Os grupos de botões de comando possuem sempre um botão definido como <i>default</i> ?	18	7	4
	5. A estrutura dos menus é concebida de modo a diminuir os passos necessários para a seleção?	25	4	0
	Conformidade	82,07%		

Crítérios	Questões	S	P	N
Densidade Informal	1. A densidade informacional das janelas é reduzida?	17	10	2
	2. As telas apresentam somente os dados e informações necessários e indispensáveis para o usuário em sua tarefa?	19	10	0
	3. Na entrada de dados codificados, os códigos apresentam somente os dados necessários estão presentes na tela de uma maneira distinguível?	20	7	2

	4. O sistema minimiza a necessidade do usuário lembrar dados exatos de uma tela para outra?	24	4	1
	5. Na leitura de uma janela, o usuário tem seus movimentos oculares minimizados através da distribuição dos objetos principais segundo as linhas de um "Z" ?	21	8	0
	6. O sistema evita apresentar um grande número de janelas que podem desconcentrar ou sobrecarregar a memória do usuário?	24	4	1
	7. Na manipulação dos dados apresentados pelo sistema, o usuário está liberado da tradução de unidades?	21	5	3
	8. As listas de seleção e combinação apresentam uma altura correspondente a um máximo de nove linhas?	13	12	4
	9. Os painéis de menu apresentam como ativas somente as opções necessárias?	19	7	3
	Conformidade	81,03%		

Crítérios	Questões	S	P	N
Ações Explícitas	1. O sistema posterga os processamentos até que as ações de entrada do usuário tenham sido completadas?	15	12	2
	2. Durante a seleção de uma opção de menu o sistema permite a separação entre indicação e execução?	21	7	1
	3. Para iniciar o processamento dos dados, o sistema sempre exige do usuário uma ação explícita de "ENTER"?	11	11	7
	4. É sempre o usuário quem comanda a navegação entre os campos de um formulário?	16	11	2
	Conformidade	71,98%		

Crítérios	Questões	S	P	N
Controle de Usuário	1. O usuário pode terminar um diálogo seqüencial repetitivo a qualquer instante?	14	11	4
	2. O usuário pode interromper e retomar um diálogo seqüencial a qualquer instante?	17	10	2
	3. O usuário pode reiniciar um diálogo seqüencial a qualquer instante?	17	9	3

	4. Durante os períodos de bloqueio dos dispositivos de entrada, o sistema fornece ao usuário uma opção para interromper o processo que causou o bloqueio?	18	5	6
Conformidade		71,98%		
Crítérios	Questões	S	P	N
Flexibilidade	1. Os usuários têm a possibilidade de modificar ou eliminar itens irrelevantes das janelas?	15	9	5
	2. Ao usuário é permitido personalizar o diálogo, através da definição de macros?	15	8	6
	3. É permitido ao usuário alterar e personalizar valores definidos por <i>default</i> ?	16	10	3
Conformidade		68,39%		
Crítérios	Questões	S	P	N
Experiência do usuário	1. Caso se trate de um sistema de grande público, ele oferece formas variadas de apresentar as mesmas informações aos diferentes tipos de usuário?	19	7	3
	2. Os estilos de diálogo são compatíveis com as habilidades do usuário, permitindo ações passo-a-passo para iniciantes e a entrada de comandos mais complexos por usuários experimentados?	18	9	2
	3. O usuário pode se deslocar de uma parte da estrutura de menu para outra rapidamente?	20	7	2
	4. O sistema oferece equivalentes de teclado para a seleção e execução das opções de menu, além do dispositivo de apontamento (mouse,...)?	19	9	1
	5. O sistema é capaz de reconhecer um conjunto de sinônimos para os termos básicos definidos na linguagem de comando, isto para se adaptar aos usuários novatos ou ocasionais?	18	10	1
	6. O usuário experiente pode efetuar a digitação de vários comandos antes de uma confirmação?	16	8	5
Conformidade		77,59%		
Crítérios	Questões	S	P	N
Proteção contra erro	1. O sistema apresenta uma separação adequada entre áreas selecionáveis de um painel de menu de modo a minimizar as ativações acidentais?	21	8	0

2. Em toda ação destrutiva, os botões selecionados por <i>default</i> realizam a anulação dessa ação?	23	6	0
3. Os campos numéricos para entrada de dados longos estão subdivididos em grupos menores e pontuados com espaços, vírgulas, hífen ou barras?	22	7	0
4. Ao final de uma sessão de trabalho o sistema informa sobre o risco de perda dos dados?	15	10	4
5. O sistema emite sinais sonoros quando ocorrem problemas na entrada de dados?	18	10	1
6. As teclas de funções perigosas encontram-se agrupadas e/ou separadas das demais no teclado?	18	8	3
7. O sistema solicita confirmação (dupla) de ações que podem gerar perdas de dados e/ou resultados catastróficos?	17	9	3
Conformidade	80,30%		

Crítérios	Questões	S	P	N
Mensagem de erro	1. As mensagens de erro ajudam a resolver o problema do usuário, fornecendo com precisão o local e a causa específica ou provável do erro, bem como as ações que o usuário poderia realizar para corrigi-lo?	19	9	1
	2. As mensagens de erro são neutras e polidas?	14	13	2
	3. As frases das mensagens de erro são curtas e construídas a partir de palavras curtas, significativas e de uso comum?	19	7	3
	4. As mensagens de erro estão isentas de abreviaturas e/ou códigos gerados pelo sistema operacional?	19	9	1
	5. O usuário pode escolher o nível de detalhe das mensagens de erro em função de seu nível de conhecimento?	14	9	6
	6. A informação principal de uma mensagem de erro encontra-se logo no início da mensagem?	17	6	6
	7. Quando necessário, as informações que o usuário deve memorizar encontram-se localizadas na parte final da mensagem de erro?	20	5	4
	8. Em situações normais as mensagens de erro são escritas em maiúsculo/minúsculo?	17	10	2

9. As mensagens de erro têm seu conteúdo modificado quando na repetição imediata do mesmo erro pelo mesmo usuário?	19	8	2
Conformidade	75,10%		

Crítérios	Questões	S	P	N
Correção de erro	1. Qualquer ação do usuário pode ser revertida através da opção DESFAZER?	18	9	2
	2. Através da opção REFAZER, a regressão do diálogo, também pode ser desfeita?	17	6	6
	3. Os comandos para DESFAZER e REFAZER o diálogo estão diferenciados?	19	4	6
	4. O sistema reconhece e através de uma confirmação do usuário, executa os comandos mais freqüentes mesmo com erros de ortografia?	19	8	2
	5. Depois de um erro de digitação de um comando ou de dados, o usuário tem a possibilidade de corrigir somente a parte dos dados ou do comando que está errada?	19	7	3
	Conformidade	75,17%		

Crítérios	Questões	S	P	N
Consistência	1. A identificação das caixas, telas ou janelas são únicas?	21	8	0
	2. A organização em termos da localização das várias características das janelas é mantida consistente de uma tela para outra?	18	9	2
	3. A posição inicial do cursor é mantida consistente ao longo de todas as apresentações de formulários?	16	9	4
	4. Uma mesma tecla de função aciona a mesma opção de uma tela para outra?	18	8	3
	5. Os ícones são distintos uns dos outros e possuem sempre o mesmo significado de uma tela para outra?	20	7	2
	6. A localização dos dados é mantida consistente de uma tela para outra?	17	9	3
	7. Os formatos de apresentação dos dados são mantidos consistentes de uma tela para outra?	19	5	5

8. Os rótulos estão na mesma posição em relação aos campos associados?	17	10	2
9. O símbolo para convite à entrada de dados é padronizado (por exemplo " : ")?	18	7	4
10. As áreas de entrada de comandos estão na mesma posição de uma tela para outra?	16	9	4
11. Os significados dos códigos de cores são seguidos de maneira consistente?	20	6	3
Conformidade			76,33%

Critérios	Questões	S	P	N
Significados	1. As denominações dos títulos estão de acordo com o que eles representam?	22	7	0
	2. Os títulos das páginas de menu são explicativos, refletindo a natureza da escolha a ser feita?	23	6	0
	3. Os títulos das páginas de menu são distintos entre si?	20	7	2
	4. Os títulos das páginas de menu são combináveis ou componíveis?	16	11	2
	5. As denominações das opções de menu são familiares ao usuário?	20	8	1
	6. O vocabulário utilizado nos rótulos, convites e mensagens de orientação são familiares ao usuário, evitando palavras difíceis?	19	8	2
	7. O vocabulário utilizado em rótulos, convites e mensagens de orientação é orientado à tarefa, utilizando termos e jargão técnico normalmente empregados na tarefa?	24	4	1
	8. Os cabeçalhos de colunas de dados são significativos e distintos?	20	9	0
	9. O sistema adota códigos significativos ou familiares aos usuários?	14	14	1
	10. As abreviaturas são significativas?	22	5	2
	11. As abreviaturas são facilmente distinguíveis umas das outras, evitando confusões geradas por similaridade?	16	11	2
	12. A intermitência luminosa (pisca-pisca) é usada com moderação e somente para atrair a atenção para alarmes, avisos ou mensagens críticas?	20	8	1
Conformidade			81,90%	

Cr�terios	Quest�es	S	P	N
Compatibilidade	1. As telas s�o compat�veis com o padr�o do ambiente?	20	8	1
	2. A imagem do formul�rio na tela do terminal assemelha-se com o formul�rio de entrada em papel?	19	8	2
	3. O sistema prop�e uma caixa de di�logo modal, quando a aplica�o deve ter todos os dados antes de prosseguir ou quando o usu�rio tenha de responder a uma quest�o urgente?	20	6	3
	4. As caixas de di�logo do sistema apresentam um bot�o de valida�o, um bot�o de anula�o e, se poss�vel, um bot�o de ajuda?	19	7	3
	5. Os significados usuais das cores s�o respeitados nos c�digos de cores definidos?	19	8	2
	6. As op�es de codifica�o por cores s�o limitadas em n�mero?	16	10	3
	7. As informa�es codificadas atrav�s das cores apresentam uma codifica�o adicional redundante?	15	7	7
	8. A taxa de intermit�ncia para elementos piscantes est� entre 2 e 5 Hz (2 a 5 piscadas por segundo)?	16	10	3
	9. A apresenta�o sonora � compat�vel com o ru�do do ambiente?	19	8	2
	10. As mensagens s�o sempre afirmativas e na voz ativa?	22	5	2
	11. Quando uma frase descreve uma seq�ncia de eventos, a ordem das palavras na frase corresponde � seq�ncia temporal dos eventos?	24	4	1
	12. Ilustra�es e anima�es s�o usadas para completar as explica�es do texto?	19	8	2
	13. O sistema segue as conven�es dos usu�rios para dados padronizados?	18	9	2
	14. O sistema utiliza unidades de medida familiares ao usu�rio?	23	5	1
	15. Dados num�ricos que se alterem rapidamente s�o apresentados analogicamente?	18	11	0
	16. Dados num�ricos que demandam precis�o de leitura s�o apresentados digitalmente?	21	7	1
	17. Os itens s�o numerados com n�meros, n�o com letras?	17	11	1
	18. Os identificadores num�ricos de op�o de menu iniciam de "1", e n�o de "0"?	19	7	3

19. Os eixos de um gráfico apresentam escalas numéricas iniciando em zero, com intervalos padronizados, crescendo da esquerda para a direita e de cima para baixo?	19	6	4
20. Os itens de um grupo de botões de rádio são mutuamente exclusivos?	17	9	3
21. Os itens de um grupo de caixas de atribuição permitem escolhas independentes?	19	10	0
Conformidade	78,98%		

Crítérios	Conformidade(%)
1. Presteza	80,12%
2. Agrupamento por Localização	83,23%
3. Agrupamento por Formato	81,44%
4. <i>Feedback</i>	83,33%
5. Legibilidade	79,12%
6. Concisão	75,99%
7. Ações Mínimas	82,07%
8. Densidade Informal	81,03%
9. Ações Explícitas	71,98%
10. Controle de Usuário	71,98%
11. Flexibilidade	68,39%
12. Experiência do Usuário	77,59%
13. Proteção contra Erro	80,30%
14. Mensagem de Erro	75,10%
15. Correção de Erro	75,17%
16. Consistência	76,33%
17. Significados	81,90%
18. Compatibilidade	78,98%

MÉDIA =	78,00%
----------------	---------------

CRITÉRIO INTERFACE

A conformidade de cada critério será calculada seguindo os pesos da tabela

Resposta das questões	Peso
Sim (S)	1.0
Parcialmente (P)	0.5
Não (N)	0.0

Fórmula utilizada: Conformidade (critérios) = $(\sum \text{res}/n^{\circ} \text{questões}) * 100$

Questões	S	P	N
1. As listas para seleção de dados são organizadas logicamente?	18	9	2
2. Você soube identificar a todo instante onde se encontrava o cursor na tela?	21	6	2
3. As telas lhe parecem visualmente agradáveis?	21	8	0
4. Os contrastes das cores utilizadas nas telas lhe parecem agradáveis?	23	4	2
5. Você conseguiu visualizar bem o tipo de fonte empregada nas telas?	21	6	2
6. Você pode acessar qualquer tela do sistema sem precisar passar pelas demais?	17	8	4
7. O sistema lhe permitiu criar ou escolher seus próprios ícones para início de uma tarefa (diálogo)?	16	8	5
8. No diálogo com o sistema, você acha que o seu nível de conhecimento foi compatível com as telas?	24	4	1
9. O sistema lhe informava sobre a possível perda de dados quando passou de uma pasta ou tela do sistema para outra sem salvar?	19	7	3
10. Os ícones lhe conduziam para os mesmos caminhos em todas as telas?	20	6	3
11. A estrutura das informações disponíveis nas telas foi sempre apresentada da mesma maneira?	16	13	0
12. O vocabulário empregado no sistema lhe parece agradável ao seu entendimento?	25	4	0
Conformidade	81,18%		
TOTAL =	81,18%		

APÊNDICE D

*Objeto de
Aprendizagem*

Manual do Usuário

*Aprendendo
Programação*

CARACTERÍSTICAS DO OBJETO

1. INFORMAÇÕES GERAIS

Título: Aprendendo Programação;

Disciplina: Programação I;

Conteúdo Disciplinar: Conceitos Básicos de Programação;

Público alvo: Pode ser utilizado em cursos de técnicas de programação e construção de algoritmos.

Número de Atividades: 5

2. Requisitos técnicos

Sistema Operacional Windows;

3. INTERFACE DO USUÁRIO

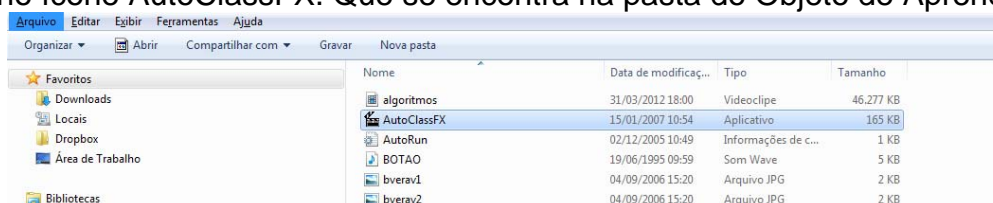
A interface do usuário foi projetada com o objetivo de ser manipulada facilmente, onde o público-alvo possa se orientar de modo intuitivo.



VOCÊ VAI PRECISAR DE:


- ✓ Mouse;
- ✓ Teclado;
- ✓ Programa de vídeo instalado;
- ✓ Flash instalado;
- ✓ Se for CD executável, irá precisar de um drive de CD.

COMO UTILIZAR: Para utilizar o Objeto de Aprendizagem, basta iniciar o CD ou clicar no ícone AutoClassFX. Que se encontra na pasta do Objeto de Aprendizagem.



Após concluir esta ação, aparecerá a tela abaixo:



Para iniciar a navegação é necessário clicar no botão  avançar que aparece na parte inferior direita.

Toda a navegação do OA se dá por apenas uma maneira:




Por meio do mouse, é possível utilizar todos os recursos que o objeto disponibiliza. Para acessar as opções de navegação, basta clicar com o mouse na opção desejada. A imagem a seguir mostra algumas das opções de navegação, basta o usuário escolher uma das opções usar o OA.




A tela seguinte mostra que na parte inferior das telas, temos as opções de navegação, que permite voltar ao menu inicial, voltar a tela anterior, fechar o Objeto ou avançar.

Exemplo



- Todo mamífero é um animal.
- Todo cavalo é mamífero.
- Portanto, todo cavalo é um animal.



Esse exemplo ilustra silogismo, que no estudo da Lógica Proposicional (ou Cálculo Sentencial) representam um argumento composto de duas premissas e uma conclusão; e está estabelecendo uma relação, que pode ser válida ou não. Esse é um dos objetivos da lógica, o estudo de técnicas de formalização, dedução e análise que permitam verificar a validade de argumentos. No caso do exemplo acima citado é válido.

Devemos resaltar que, apesar da aparente coerência de um encadeamento lógico, ele pode ser válido ou não em uma estrutura.

Página 2 de 5 Menu

Após concluir a parte teórica do Objeto, que constituiu também da parte textual e dos vídeos, o usuário poderá fazer os conhecimentos adquiridos por meio das 5 atividades propostas. Todas as atividades elaboradas são bastante peculiares. Isso implica que o presente documento tem a obrigação de relatar a sequência de ações necessárias para se efetuar cada um dos exercícios. A seguir, serão apresentados os exercícios e como prosseguir em cada um deles.

Exercício 1: Basta o usuário clicar sobre a resposta certa, caso acerte, receberá um parabéns. Caso erre, será apresentado um *feedback* para auxiliar o usuário. Permitindo ir para o próximo exercício somente se acertar o exercício que está fazendo.

Exercício 1

Três senhoras – dona Branca, dona Rosa e dona Violeta – passeavam pelo parque quando dona Rosa disse:

- Não é curioso que estejamos usando vestidos de cores branca, rosa e violeta, embora nenhuma de nós esteja usando um vestido de cor igual ao seu próprio nome ?
- Uma simples coincidência - respondeu a senhora com o vestido violeta.

Qual a cor do vestido de cada senhora ?

Selecione a alternativa correta.

<p>a) Dona Branca está com o vestido rosa, Dona Rosa está com o vestido violeta e Dona Violeta está com o vestido branco.</p> <p>c) Dona Branca está com o vestido rosa, Dona Rosa está com o vestido branco e Dona Violeta está com o vestido violeta.</p>	<p>b) Dona Branca está com o vestido violeta, Dona Rosa está com o vestido branco e Dona Violeta está com o vestido rosa.</p> <p>d) Dona Branca está com o vestido branco, Dona Rosa está com o vestido violeta e Dona Violeta está com o vestido rosa.</p>
---	---

Menu

Exercício 2: A atividade 2 basta que o usuário complete os campos com os passos necessários conforme pede o enunciado do exercício. Nessa atividade, se o usuário errar aparece apenas um *feedback* para rever o preenchimento, mas caso acerte aparecerá uma mensagem de muito bem.

Exercício 2

Um homem precisa atravessar um rio com um barco que possui capacidade apenas para carregar ele mesmo e mais uma de suas três cargas, que são: um lobo, um bode e um maço de alfafa. O que o homem deve fazer para conseguir atravessar o rio sem perder suas cargas?

Abaixo estão as ações necessárias para efetuar a travessia segura, coloque os números nos quadrinhos definindo a ordem das ações, montando assim o algoritmo.

- levar a alfafa para o outro lado do rio
- voltar sem carga nenhuma
- levar o bode para o outro lado do rio
- voltar com o bode
- voltar sem carga alguma
- levar o bode para o outro lado do rio
- levar o lobo para o outro lado do rio

Menu

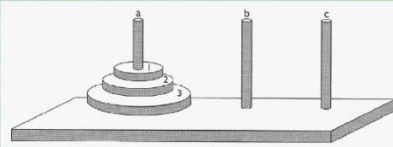
Exercício 3: Nesse exercício segue o mesmo padrão do exercício 2.

Exercício 3

Elabore um algoritmo que mova três discos de uma Torre de Hanói, que consiste em três hastes (a - b - c), uma das quais serve de suporte para três discos de tamanhos diferentes (1 - 2 - 3), os menores sobre os maiores. Pode-se mover um disco de cada vez para qualquer haste, contanto que nunca seja colocado um disco maior sobre c menor.

O objetivo é transferir os três discos para outra haste C.

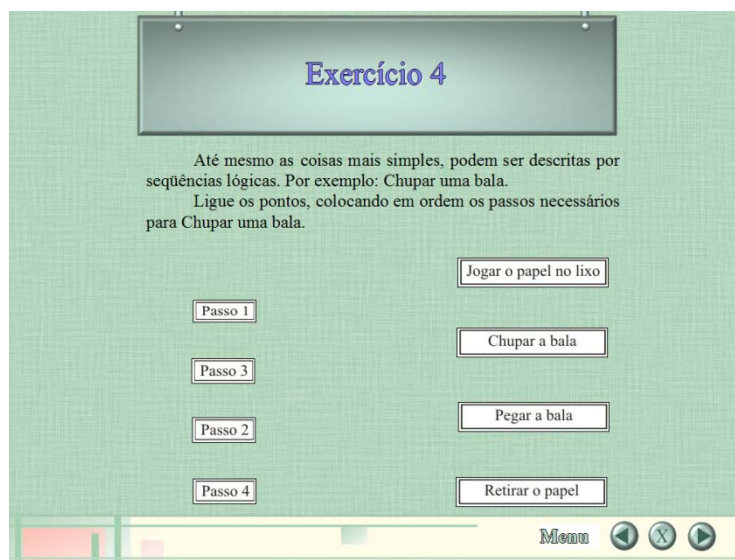
Os passos necessários para o algoritmo estão abaixo, apenas enumere-os.



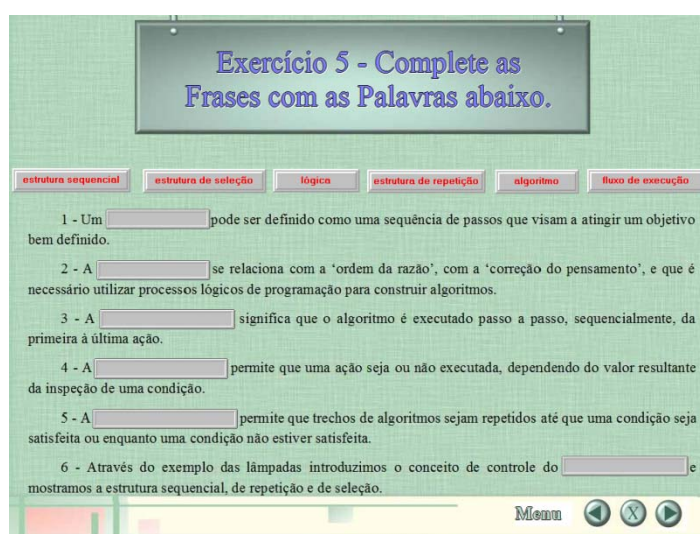
- Mover o disco 3 da torre A para a torre C
- Mover o disco 1 da torre A para a torre C
- Mover o disco 2 da torre B para a torre C
- Mover o disco 2 da torre A para a torre B
- Mover o disco 1 da torre C para a torre B
- Mover o disco 1 da torre A para a torre C
- Mover o disco 1 da torre B para a torre A

Menu

Exercício 4: No exercício 4 o usuário irá ligar a o passo com a sequência certa. Clicando uma vez sobre passo, por exemplo passo1 e depois clicar na opção correta. Quando acertar receberá uma mensagem de muito bem.



Exercício 5: Basta que o usuário clique sobre a resposta e arraste para completar o exercício. Ao final do exercício aparece uma mensagem de Parabéns.



Este manual refere-se ao uso do Objeto de Aprendizagem “Aprendendo Programação” como forma de apoiar o ensino-aprendizagem destinado ao público com interesses em técnicas de programação e construção de algoritmos

Seu objetivo é apresentar os conceitos elementares de lógica e sua aplicação no cotidiano, definir algoritmo, estabelecer uma relação entre lógica e algoritmos: a lógica de programação. Exemplificar a aplicação dos algoritmos utilizando situações do dia-a-dia, comparar as principais formas de representação de algoritmos. O conteúdo é fundamental para a compreensão e construção de algoritmos básicos.

APÊNDICE E

*Objeto de
Aprendizagem*

Manual do Professor

*Aprendendo
Programação*

OBJETIVOS DA AULA

O objetivo é apresentar os conceitos elementares de lógica e sua aplicação no cotidiano, definir algoritmo e estabelecer uma relação entre lógica e algoritmos: a lógica de programação. Exemplificar a aplicação dos algoritmos utilizando situações do dia-a-dia, comparar as principais formas de representação de algoritmos. O conteúdo é fundamental para a compreensão e construção de algoritmos básicos.

O Objeto de aprendizagem ensina:

- Introdução à lógica de programação;
- Transformar a lógica em algoritmo;
- Conceitos e exemplos de algoritmos
- Noções de fluxo de controle

DURAÇÃO DAS ATIVIDADES

Duas horas-aula de 45 minutos.

PRÉ-REQUISITOS

Os pré-requisitos necessários para o aluno utilizar o OA é que tenha de preferência o ensino médio, para que possa compreender a lógica de programação.

REQUISITO TÉCNICO

Sistema Operacional Windows;

VOCÊ VAI PRECISAR DE:

- ✓ Mouse;
- ✓ Teclado;
- ✓ Programa de vídeo instalado;
- ✓ Flash instalado;
- ✓ Se for CD executável, irá precisar de um drive de CD.

MOTIVAÇÃO

O conteúdo de conceitos básicos de programação deve ser apresentado de uma forma bastante clara, mostrando ao usuário que a facilidade nos conceitos de programação só será alcançada no decorrer da disciplina.

Deixe claro ao usuário o quanto será útil aprender sobre o tema.

Lembre-se de que o Objeto de Aprendizagem e o computador são também recursos motivacionais importantes.

ESTRATÉGIAS

Apresenta aos alunos por meio de uma abordagem sucinta e natural, os conceitos iniciais que introduzirão o aluno ao contexto da disciplina de programação I. Tendo como Objetivo dar ao aluno possibilidade de dar os primeiros passos para a familiarização com a lógica, os algoritmos e seus métodos de construção.

MÉTODOS

1º O objeto de aprendizagem utilizado nesta aula não dispensa aulas tradicionais sobre o assunto. É importante que antes de conduzir os alunos até o laboratório de informática, eles já tenham conhecimento sobre o objeto.

2º Se possível, cada aluno utilize um computador, caso não seja possível poderá ser em dupla ou mais.

3º Recomenda-se que antes de acomodar a turma no laboratório, os objetos já estejam salvos no computador ou em fácil acesso, a fim de adiantar algumas tarefas que poderiam tomar minutos da aula.

4º Após todos os alunos abrirem o Objeto, relembre as aulas anteriores, pois é importante que o aluno relacione o conteúdo aprendido em sala de aula com o objeto de aprendizagem.

5º Permita que os alunos naveguem pelo objeto e estimule-os a voltar às telas que por ventura não tenham sido compreendidas, assim como também animações que requerem mais atenção.

6º A navegação do Objeto pode ser consultada no manual que acompanha o material.

7º Quando todos tiverem alcançado a etapa das atividades, comente que serão testados os conhecimentos adquiridos. Isto é o suficiente. Dê liberdade a cada aluno de agir conforme julgar melhor. Apenas oriente-os e ofereça ajuda se necessário.

Observação: As instruções de como resolver as atividades do Objeto está contida no manual que acompanha o mesmo.

8º Ao final da aula, interaja com os alunos, incitado um debate informal sobre o tema aprendido.

AVALIAÇÃO

Observe o desempenho de cada aluno individualmente e se possível, compare-os com o da turma de modo genérico.

Analise as dúvidas, observe se já muita dificuldade em responder os exercícios e se eles trabalham em grupo, discutindo e argumentando sobre as respostas sugeridas, ou se utilizam o método de tentativa-e-erro para alcançar as atividades a fim de terminá-las rapidamente.

MATERIAL ADICIONAL

Utilize de livros durante as aulas, ou mesmo indique aos alunos como forma de complementar os assuntos abordados pelo OA.

Indique também filmes relacionados ao assunto e também alguns sites que possam servir de apoio até mesmo durante as aulas em laboratório.

ATIVIDADES COMPLEMENTARES

Utilize de outras ferramentas disponíveis no computador, com algo complementar, para que o aluno seja estimulado ao aprendizado. Não se limite apenas ao Objetos de Aprendizagem.