

Ensino de Algoritmos e Programação: Uma Experiência no Nível Médio

José Carlos Rocha Pereira Júnior¹, Clevis Elena Rapkiewicz¹, Carla Delgado²,
José Antonio Moreira Xexeo³

¹LEPROD - Universidade Estadual do Norte Fluminense (UENF)
Campos dos Goytacazes – RJ – Brasil

²Universidade Federal do Rio de Janeiro (UFRJ), Rio de Janeiro – RJ

³Centro Universitário Bennett, Rio de Janeiro – RJ

{zeca, clevis}@uenf.br, delgado@pobox.com, xexeo@bennett.br

Abstract. *The Licenciature in Computation Work Group (GT-3) of the Brazilian Computer Society (SBC) established between the goals for 2004-2005 develop experiences related to the programming teaching in high-school. This paper presents the realization of such proposal by UENF in a Cardoso Moreira's - Rio de Janeiro public school through a workshop that permitted to identify difficulty points in the process and the necessity of work more the informatics of interdisciplinary way.*

Resumo. *O Grupo de Trabalho de Licenciatura em Computação (GT-3) da Sociedade Brasileira de Computação (SBC) estabeleceu entre as metas para 2004-2005 desenvolver experiências relativas ao ensino de programação no nível médio. Este artigo apresenta a realização de tal proposta pela UENF numa escola pública de Cardoso Moreira - RJ através de uma oficina que permitiu identificar pontos de dificuldade no processo e a necessidade de trabalhar mais a informática de modo interdisciplinar.*

Palavras-chaves: *ensino de programação, ensino médio, algoritmos*

1. Introdução

Durante os três últimos séculos deslumbraram-se diferentes eras nas quais se constava o domínio de alguma tecnologia. O desenvolvimento tecnológico durante este período iniciou-se com a era dos grandes sistemas mecânicos até atingir a atual era da informação sobre a qual Castells (1999) refere-se como *Sociedade Informacional*. Para este autor, uma era que se caracteriza não pela centralidade de conhecimento e informação, mas pela aplicação destes conhecimentos e informações na geração de novos conhecimentos e de dispositivos de processamento/comunicação da informação, em um ciclo de realimentação cumulativo entre inovação e seu uso.

Nesta sociedade um elemento fundamental é a *educação*, que visa não somente capacitar o indivíduo a trabalhar com alguma das Tecnologias de Informação e Comunicação (TIC), mas investir na criação de competências para sua atuação efetiva no mercado de trabalho, na sua formação como auto-aprendiz para se adaptar a rápida mudança tecnológica e para aproveitar o conhecimento adquirido no complexo processo de tomada de decisão.

Nesta realidade, a escola necessita agir de forma diferente estando atenta às mudanças tecnológicas de forma a ser menos impactada por estas. No contexto deste ambiente escolar, os professores passam a repensar o processo de ensino e aprendizagem construindo novas formas de ação utilizando a informática como um novo instrumento de aprendizagem.

A informática deve habilitar e dar oportunidade ao aluno de adquirir novos conhecimentos, facilitar o processo ensino/aprendizagem, enfim ser um complemento de conteúdos curriculares visando o desenvolvimento integral do indivíduo (Flores, 2002 *apud* Lopes, 2004). Indo mais além, a informática representa um componente suscetível que influi a formação do indivíduo.

Uma vez que a informática tem representado uma peça elementar do processo educacional propõe-se incluir conteúdos da área de computação e informática no Ensino Médio. Esta idéia foi discutida durante o XXIV Congresso da SBC (2004) em Salvador pelo Grupo de Licenciatura em Computação (GT3) e aprovada em assembléia. O objetivo, entre outras coisas, é desenvolver competências desta área, desde cedo, de forma a facilitar a integração do computador nas mais diversas profissões e áreas do conhecimento. Além disso, a introdução da computação no ensino médio pode fomentar o interesse pela área aumentando o número de profissionais no país.

Os cursos de computação e informática são compostos de diferentes e interligadas áreas de conhecimento desenvolvendo diferentes competências, habilidades e conhecimentos tanto na área básica quanto na área tecnológica. Entre as competências mais difíceis de serem desenvolvidas estão as relacionadas com o desenvolvimento de algoritmos e programas. Esta é uma forte razão para incluir tal temática no ensino médio – assim, os futuros alunos dos cursos de computação teriam menos dificuldade nessa área. E aqueles que se dirigissem para outros ramos profissionais teriam desenvolvido competências de resolução de problemas e formalização dos mesmos que seriam úteis em suas respectivas áreas do conhecimento.

Para analisar o nível de dificuldade e possíveis resultados de introduzir o ensino de algoritmos e programação no ensino médio foi desenvolvido o projeto de uma “Oficina de Lógica de Programação”, o qual foi aceito e implementado no Colégio Baltazar Carneiro na cidade de Cardoso Moreira – RJ com alunos do segundo ano. Este projeto fundamentou-se na estratégia proposta em Delgado *et al* (2004) que difere muito do método tradicional de ensinar algoritmos e programação. A experiência realizada permitiu identificar alguns pontos de dificuldade a serem superados no processo, como também ocorrem no nível superior. Porém, também reafirmou a questão de que a informática na sala de aula precisa ser mais bem trabalhada e de modo interdisciplinar, isto é, indo além do conhecimento necessário para a utilização de equipamentos e promovendo um aprendizado de diversos outros conteúdos que no caso tratado neste artigo será através da programação de computadores.

Este artigo analisa a experiência realizada, estando dividido em cinco seções: a primeira relata alguns problemas relacionados ao ensino de programação; a segunda apresenta o projeto da oficina proposta; a terceira descreve a metodologia utilizada; a quarta o resultados obtidos com a oficina. Por fim, considerações finais são apresentadas.

2. Alguns Problemas no Ensino de Programação

A computação ocupa uma posição privilegiada na atual sociedade. Sachs (2000) observa que os países se dividem em três grupos – os que desenvolvem tecnologia, os que a absorvem e os excluídos – e que “sem acesso à tecnologia e à informação o país cai numa armadilha de pobreza”¹. Os autores deste artigo consideram que uma forma fundamental para que o país seja desenvolvedor de tecnologia de computação é o desenvolvimento de recursos humanos com amplas competências nessa área.

É fundamental que se aprimore a qualidade do processo de ensino e aprendizagem nos cursos da área de computação e informática. Um dos gargalos destes cursos é o alto índice de reprovação e/ou evasão, desde o início do curso, em disciplinas de algoritmo e programação. Este é um dos fatores que fazem com que a quantidade de egressos concluintes seja consideravelmente inferior à quantidade de entradas nesses cursos. Apenas a título de exemplo, até 2000, o curso de Informática da UFRJ, existente desde o início dos anos 80, havia formado apenas 911 bacharéis (Lacerda, 2002).

Outros problemas, alguns bastante objetivos, podem ser identificados a partir do levantamento na literatura.

1. Dificuldade de interpretação do próprio problema (Falkembach *et al*, 2003), muito antes da dificuldade de interpretação de algum tipo de representação; segundo os mesmos autores, além da dificuldade de interpretar enunciados,
2. Dificuldade de identificar os pré-requisitos necessários para o desenvolvimento das competências de construção de algoritmos e programação, de forma a poder trabalhá-los. A título de exemplo, alguns autores preconizam a necessidade de domínio de habilidades matemáticas prévias ou ao menos integradas (Henderson, 1987) ou pelo menos desejável (Koliver, Dorneles e Casa, 2004). Já para outros (Baeza-Yates, 1995), o próprio nível de conhecimento prévio de lógica matemática é discutido: Koliver, Dorneles e Casa(2004) afirmam que a apresentação de princípios básicos da lógica é suficiente para a resolução da maior parte dos problemas propostos para uma disciplina de nível introdutório.
3. No caso dos alunos, estes costumam ter muita dificuldade em aplicar suas habilidades prévias, criando fonte de medo e frustração (Chaves de Castro et al, 2003). Os próprios professores têm dificuldades em identificar/reconhecer nos alunos tais habilidades prévias (Schultz, 2003) de forma a aproveitá-las melhor, além dificuldades para trabalhar a capacidade de abstração do aluno, seja na busca de possíveis soluções seja na escolha das estruturas de dados (Nobre e Menezes 2002 *apud* Schultz, 2003). A necessidade do professor fazer o aluno compreender a abstração envolvida com toda simbologia utilizada é corroborada por Rodrigues (2002).
4. Necessidade (ou não) de definir o paradigma de programação utilizado, de forma a estruturar a soluções dos problemas já voltados para tal paradigma. No caso, discute-se tanto se a forma de representação da solução com suas ferramentas de apoio ou as próprias linguagens, quando utilizadas: devem estas ou não estar relacionadas a algum paradigma? O relatório da força tarefa da IEEE e ACM (2001)

¹ Jeffrey Sachs, economista mundialmente conhecido, é diretor do Centro de Desenvolvimento Internacional da Universidade de Harvard.

afirma que o estudo de algoritmos deve fornecer o discernimento da natureza intrínseca do problema e técnicas de solução independentes de qualquer paradigma, linguagem de programação ou hardware. Segundo a força tarefa, deve-se portanto, tratar o problema independente da forma de sua implementação. Esta visão é corroborada por autores como Henderson (1986), (Koliver, Dorneles e Casa, 2004). Alguns autores vão mais longe ao afirmar que a introdução de paradigmas como orientação a objetos nos cursos iniciais sobre algoritmos não fornece evidências significativas de facilitar o aprendizado (Marion, 1999; Perego, Lisboa e Bertagnolli, 2002; Burton e Bruhn, 2003). Já outros autores preconizam que certos paradigmas, como o funcional, podem facilitar o processo (Chaves de Castro *et al*, 2003). Outros defendem o próprio paradigma imperativo, preferencialmente orientado a objeto (Baeza-Yates, 1995) ou o estruturado clássico (Delgado *et al*, 2004).

Outra dificuldade, que pode ou não ser relacionada com a escolha ou não de algum paradigma, é o nível e a forma de formalização desejada, variando entre o uso apenas rudimentar de linguagem natural estruturada, o uso desta de forma mais estruturada na forma de pseudo-código e no uso de alguma linguagem para codificação, além do possível uso de formas gráficas de formalização como, por exemplo, fluxogramas ou diagramas de Chapin. Alguns autores como Gries (1974) defendem o uso de linguagens de programação mais didáticas como o Pascal, o que é contra-argumentado por Koliver, Dorneles e Casa (2004) ao lembrar que as linguagens têm grande número de detalhes sintático-semânticos, que nem sempre permitem abstração de aspectos da máquina quando da codificação da solução, além de em certa forma prender o aluno ao paradigma de programação da linguagem.

Outro ponto problemático abordado nos problemas relacionados ao ensino e aprendizagem de algoritmos e programação é a diversidade de ritmos de aprendizagem dos alunos conjugada, muitas vezes, com turmas grandes. Como ministrar atendimento individual, respeitando as individualidades, e, ao mesmo tempo, buscar trabalho cooperativo?

Finalmente, há que se considerar o envolvimento extra-classe necessário para o acompanhamento das dificuldades das disciplinas de algoritmos e programação.

Eventualmente, tais problemas podem ser até menores quando abordados numa faixa etária em que certas formas de abordar problemas podem ser mais intuitivas, podendo-se partir dessa base para levar a alguma forma de formalização. Caso esta hipótese se confirme, pode ser relevante para o país incluir, de alguma forma, o desenvolvimento de competências nesta área no ensino médio. Investigar, portanto, como abordar, estes problemas, no âmbito do ensino médio, pode fornecer subsídios para contribuir para que o Brasil não caia numa armadilha de pobreza, parafraseando Sachs (2000).

3. O projeto de uma Oficina

Visando analisar qual o nível de problemática de ensino de algoritmos e programação no nível médio e, portanto, aplicar a idéia do GT-3, foi desenvolvido o projeto da “Oficina de Lógica de Programação” com duração total de 36 horas divididas em 6 horas semanais. O desenvolvimento deste projeto foi realizado no final do ano letivo o que acarretou um curto prazo de tempo para sua implementação, justificando esta carga horária.

O projeto foi apresentado ao Colégio Estadual Baltazar Carneiro localizado em Cardoso Moreira, um pequeno município do Estado do Rio de Janeiro. O local foi escolhido principalmente pelo público alvo que é composto por alunos com pouco acesso a recursos tecnológicos. É um colégio pólo o qual possui um laboratório de informática composto por 10 computadores cedidos em parte pelo Programa Nacional de Informática Educativa (ProInfo) e outros adquiridos pela própria unidade escolar.

Para a oficina, trabalhar com os alunos do segundo ano é a melhor proposta uma vez que os do primeiro ano talvez tenham menor nível de maturidade e os dos terceiro ano estejam preocupados com o vestibular.

A argumentação do GT-3 em relação ao objetivo da oficina incluía desenvolver competências e habilidades para construir soluções estruturadas para problemas utilizando o computador. Com os alunos capacitados a aprender ensinando o computador, ou seja, a programar, possibilitam aos professores solicitarem a construção de programas de forma a fixar e aprofundar os conteúdos abordados, bem como, utilizar-se de tais programas em sala de aula de forma interativa contribuindo para a aprendizagem significativa e promovendo a interdisciplinaridade.

O colégio consentiu a realização da oficina a qual ocorreu entre os meses de novembro e dezembro de 2004. A meta era avaliar a capacidade dos alunos desenvolverem o conteúdo e a estratégia aplicada de forma a incluir o ensino de algoritmos e programação na matriz curricular, o que deverá ocorrer no segundo semestre de 2005.

De forma a diagnosticar o nível dos usuários do laboratório de informática foi desenvolvido pelos responsáveis do mesmo² um pequeno questionário aplicado aos alunos do Ensino Médio. Em relação aos alunos das três turmas do segundo ano, público-alvo da oficina, o questionário apontou que 56% possuem conhecimentos de informática, o que poderia atentar a participação da oficina e incrementar o currículo dos mesmos. Apesar deste resultado, apenas 21% dos alunos possuem noções de internet; entretanto este fator não é primordial, porém necessário, para o objetivo da oficina.

Foram ofertadas 16 vagas para a oficina. Qualquer aluno do segundo ano do ensino médio da escola podia se inscrever, dado que a oficina foi realizada no turno da tarde e as turmas existentes são nos períodos matutino e noturno. Todas as vagas foram preenchidas no primeiro dia de divulgação da oficina.

4. A Metodologia

A metodologia utilizada na oficina baseou-se na estratégia proposta em Delgado *et al* (2004) e adotou o seguinte conteúdo:

Primeira Fase – Iniciação Lúdica – Resolução de Problemas

- Resolução de problemas de diversos domínios

Segunda Fase – Falando Sério – Formalização

- Representação de soluções em linguagem natural

² Agradecimentos as diretoras e aos professores do colégio que colaboraram com a realização deste projeto e, em especial, as professoras e coordenadoras do laboratório de informática Silvane S. Medeiros e Maria Valdete R. Moraes por todo esforço realizado para que a oficina se concretizasse.

- Noções de lógica no cotidiano
- Conectivos lógicos
- Formalização de soluções (Construção de Algoritmos)
 - Divisão em casos (estrutura de seleção – “Se”)

Terceira Fase – E agora? – Construção de Programas

- Tradução dos Algoritmos para uma Linguagem de Programação (Pascal)
- Estruturas de Repetição

A primeira fase visa desenvolver o raciocínio lógico e a criatividade dos alunos a partir da resolução de problemas. A segunda, busca a formalização gradual de soluções até o desenvolvimento de algoritmos. Enfim, estes algoritmos serão traduzidos para uma linguagem de programação.

O conteúdo de lógica está intrínseco na metodologia original, pois os alunos do ensino superior já possuem tal conhecimento adquirido em uma disciplina a parte. Este não é o caso do ensino médio, onde se fez necessário explicitar tal conteúdo tornando-o um complemento à oficina.

Além da inclusão da lógica, foi necessário adaptar a metodologia para a exígua carga horária disponível tentando fazer com que os alunos utilizassem o computador a maior parte do tempo para que conhecessem também outras ferramentas computacionais e conceitos. Esta abordagem inclui a utilização da internet, jogos computacionais, fórum de discussão e da ferramenta de desenvolvimento dentro de uma fase apropriada cuja discussão será apresentada nas subseções a seguir.

A *priori* tenta-se propor situações que busquem motivar o aluno quanto ao aprendizado da programação de computadores e aproveitar o momento de ansiedade que o mesmo se encontra para aumentar o seu interesse pelo resultado final almejado. A motivação é um elemento chave que precisa ser buscado durante todo o processo. Para tanto, relembrar ao aluno a diferença e dependência intrínseca entre hardware e software, descrever o papel do software dentro de várias atividades profissionais e referenciar o diferencial do profissional que detêm o conhecimento de programação podem produzir a motivação primária proposta.

É importante fazer os alunos entenderem todo o processo de aprendizagem a ser vivenciado para não gerar um descontentamento durante o desenvolvimento do curso, pois a ânsia inicial se deve ao desejo da programação propriamente dita cuja atividade só será realizada na última etapa do processo.

Na metodologia original a construção de algoritmos se dava a partir da última fase e não constava a utilização de alguma linguagem de programação. Entretanto, devido à baixa carga horária e em se tratar de alunos adolescentes em uma oficina a parte de suas atividades escolares, decidiu-se agregar e antecipar o desenvolvimento de algoritmos na segunda fase para incluir a tradução dos mesmos para uma linguagem de programação na última fase.

4.1. As Fases

4.1.1. Primeira Fase – Iniciação Lúdica – Resolução de Problemas

Segundo Falkembach *et al* (2003), a diferença entre um exercício e um problema é que o primeiro exige uma mera aplicação de alguma habilidade e/ou conhecimento

matemático previamente assimilado; já o segundo é uma situação nova que necessariamente envolve a criação de algum procedimento para chegar à solução.

Num primeiro momento desta fase os alunos resolvem os problemas disponibilizados em uma lista de questões. Um exemplo simples de questão é calcular a média entre dois números. Também são usadas questões de raciocínio que não demandam conhecimento prévio de algum conceito ou propriedade matemática. Esta atividade proporciona ao aluno desenvolver novos esquemas mentais de forma gradual que o capacita a chegar à solução de outros problemas a partir de situações análogas. O professor atua como facilitador auxiliando em pontos que ficaram despercebidos e sugerindo formas alternativas de propor soluções.

Estimula-se a busca de soluções em grupo, no entanto, nada impede que um problema seja solucionado individualmente ou compartilhado com os colegas do grupo. Neste momento é interessante disponibilizar a internet para que os alunos a utilizem na busca de soluções e comecem a interagir com o computador. Caso os alunos recorram ao professor, este deve tentar se assegurar que foi investido certo grau de esforço desejado quanto à busca de soluções.

Após a resolução dos problemas, os alunos expõem oralmente para a turma as propostas de solução criando um ambiente de debate entre os alunos quanto à validação da solução e a melhor proposta. Neste momento inicial ocorre certa timidez em alguns alunos, porém, com a interação com os outros busca-se aumentar o contato com a turma deixando-os livres para expressar suas idéias, ou seja, os alunos percebem que uma solução errônea agrega valor para encontrar a melhor solução e que esta não é encarada como um fator que exprime suas capacidades intelectuais. O erro, no caso, não é necessariamente visto como um problema, mas como uma etapa para construção da solução.

Num segundo momento, os alunos desenvolvem suas atividades utilizando mais o computador. Eles passam para a busca de soluções de diversos problemas através de jogos lógicos de computador. Um exemplo de jogo é o clássico que envolve um lobo, uma ovelha e um repolho, sendo necessário transportar tais personagens para outra margem de um rio obedecendo a determinadas regras (disponível em <http://www.soft.aaanet.ru/soft/051104/logic1net.swf>). Assim, os alunos se sentem mais motivados no desenvolvimento das tarefas e, conseqüentemente, desenvolvem o raciocínio, pois estes jogos agregam um conjunto de elementos multimídia que prendem mais a atenção do que questões em papel. O professor deve solicitar a descrição do caminho feito pelos alunos na busca da solução e tentar comparar as diferentes descrições de forma a fazer com que os alunos perceberem a necessidade de se criar uma que seja entendida por qualquer pessoa e não somente pela pessoa ou grupo de pessoas que a desenvolveu. Este é um momento de transição para o aprendizado inserido na fase seguinte do processo.

4.1.2. Segunda Fase – Falando Sério – Formalização

Para programar um computador é necessário utilizar uma especificação formal que nada mais é do que um conjunto de instruções a serem seguidas, as quais possam indicar alguma solução de um problema. Neste caso, trata-se de utilizar uma linguagem que seja bem interpretada, sem ambigüidades, ou seja, uma linguagem formal.

Devido à formalização representar uma dificuldade do aluno, esta fase concentra-se neste processo de forma gradual. Ou seja, inicialmente os alunos devem propor soluções utilizando sintaxe proposta por eles próprios, a partir de simplificação da linguagem natural. A cada momento o aluno melhora sua especificação até conseguir descrever um algoritmo independente de seu estilo cognitivo e independente das regras estritas do português estruturado. Esse nível somente será desenvolvido um pouco mais adiante.

Um instrumento utilizado para propor soluções é fórum de discussão na Internet. É uma oportunidade para conhecer uma determinada tecnologia disseminada, além de criar um ambiente colaborativo para o desenvolvimento de bons procedimentos. Além disso, o fórum deixa registrado o histórico das contribuições, permitindo ao próprio aluno avaliar seu processo de crescimento.

A idéia é trabalhar com pequenos grupos de alunos para validar, questionar, comentar e propor descrições dos procedimentos para encontrar soluções de determinados problemas. O professor induz estas ações a serem aplicadas em todos os grupos até chegar a um consenso quanto à formalização.

Neste caso, é interessante o professor apresentar um exemplo do como seria um procedimento formal em linguagem natural para que os alunos não julguem erroneamente a melhor proposta. Estes geralmente tendem a se basear em exemplos de outros colegas ou do professor no desenvolvimento inicial. Assim, o exemplo dado definiria pontos estratégicos não percebidos pelos alunos quanto a definição de um bom procedimento.

Para a interação inicial busca-se o desenvolvimento de diferentes questões por diferentes grupos e avaliação das soluções pelos diferentes grupos e, assim, interagir questionando a avaliação destes diferentes grupos criando um ciclo interativo e motivador. O que se propõe é diminuir a competitividade e aumentar a colaboração visando à percepção gradativa do processo de formalização.

Nesta fase também é apresentada uma pequena introdução ao estudo de lógica matemática (conectivos lógicos: *e*, *ou* e *não*). É um conteúdo até então não explorado explicitamente no nível escolar dos alunos e de extrema importância para a oficina. O que se propõe neste ponto é capacitar estes alunos a desenvolver e avaliar condições lógicas que estarão presentes em seus futuros programas.

Num segundo momento desta fase, busca-se a formalização mais rígida e mais adequada para a meta final: programar o computador. O processo de aumentar a formalização de forma que seja adequada a máquina (computador) é gradual e lento. Busca-se utilizar um exemplo simples do cotidiano para descrever um procedimento no qual será aplicada esta formalização gradativa. A estratégia neste ponto implica em estimar que materiais ou elementos serão necessários para a realização do procedimento; em identificar tais materiais com um nome subjacente; em classificar os dados dentro de um procedimento (dados de entrada, dados calculados e dados de saída) e, finalmente, em utilizar uma linguagem matemática para representar tal procedimento conforme adota uma linguagem de programação. É interessante notar que este passo referencia a definição futura das variáveis de um programa. Estabelecem-se aqui também alguns tópicos para a organização de um procedimento ou especificação.

Seguindo o processo, a próxima etapa será definir tempo de elaboração e tempo de execução e discutir a divisão de casos formalizando a estrutura de seleção “Se”. O

conhecimento dos conectivos lógicos e a definição de condições lógicas são de extrema importância para a avaliação dentro da estrutura de seleção “Se”. A formalização desta estrutura não será uma atividade de alto grau de dificuldade, pois os alunos já possuem certo conhecimento da estrutura em linguagem natural no início desta fase.

Ao final desta fase os alunos já devem ser capazes de desenvolver algoritmos com certo grau de formalização que não representam procedimentos difíceis de serem traduzidos para uma linguagem de programação procedimental. Isso é realizado na fase a seguir.

4.1.3. Terceira Fase – E agora? – Construção de Programas

O objetivo principal desta fase é justamente permitir que os alunos vejam o resultado do trabalho desenvolvido. Neste momento, pode-se questionar a existência de ferramentas que apresentam tal resultado utilizando algoritmos, porém uma linguagem de programação propriamente dita irá representar a realidade de como é realizado o desenvolvimento profissional de programas de computador e permite aumentar a proficiência dos alunos. Também permite trabalhar, em parte, o inglês com os alunos exacerbando a interdisciplinaridade que a programação poderá propor. No caso da oficina optou-se em utilizar a linguagem Pascal por ser a mais comum para esta aprendizagem no meio acadêmico e pela facilidade da tradução dos algoritmos.

No momento inicial o professor poderá solicitar aos alunos a descrição formal de um procedimento simples ao qual ele utilizará para definir algoritmo e programa, além de introduzir a tradução para a linguagem de programação. Então, uma pequena descrição de como trabalhar com o compilador será feita e também uma apresentação da estrutura básica de um programa definindo alguns tipos de variáveis e alguns comandos ou instruções simples como as de leitura e escrita de dados. A formalização utilizada na linguagem de programação trará certas dificuldades para os alunos por ser bem mais rigorosa, mas em breve tempo estas serão superadas.

Neste momento é importante utilizar o algoritmo e realizar a tradução para deter o uso das instruções da respectiva linguagem. Após certo período utilizando-se de traduções progride-se para o desenvolvimento utilizando a linguagem somente. O número de instruções a serem aprendidas e a forma de apresentação destas varia com a necessidade e disponibilidade do professor.

Ao final da fase é desejado que os alunos detenham pelo menos a capacidade de desenvolver programas que envolvam comandos de entrada e saída de dados, estruturas de seleção e de repetição.

5. Análises de Resultados Obtidos

Embora o curso tenha iniciado com todas as 16 vagas preenchidas, o mesmo finalizou com uma turma composta por apenas 5 alunos. Este alto índice de evasão se deve a vários fatores:

- Alguns alunos comparecem apenas até os dois primeiros dias da oficina, indisponibilizando a vaga para outros alunos que realmente estavam interessados em participar. Não seria desejável incluir novos alunos, pois a aplicação da metodologia já estava em andamento e, caso fosse feito, poderia dificultar o andamento do processo.

- Na ficha de inscrição era informado que o certificado seria entregue somente aos alunos que tivessem presença mínima de 75%. Neste caso os alunos poderiam ter somente 3 faltas injustificadas, o que acarretou a desistência de alguns.
- Alguns alunos ingressaram na oficina já trabalhando e outros obtiverem um emprego ao decorrer do curso. Este fato contribuiu para a falta de estudos extra-classe e da presença na oficina.
- A época em que a oficina foi realizada prejudicou o aprendizado e contribuiu para alguns alunos não comparecerem na oficina. Como era final de ano letivo, muitos estavam preocupados com as provas finais e as recuperações.

O gráfico da frequência (Figura 1) mostra que a presença diária dos alunos começou a se estabilizar na transição da primeira para a segunda fase. Observar que cada uma das barras significa um dia de aula, com carga horária de 3 horas. O maior índice de não comparecimento ocorreu no final da segunda fase quando o colégio promoveu um evento no qual a maioria dos alunos estava participando. A realização deste evento não foi considerada no planejamento do cronograma da oficina e não havia mais como estendê-lo, pois o colégio entraria em recesso ao final, o que acabou prejudicando em parte o aprendizado de alguns alunos.

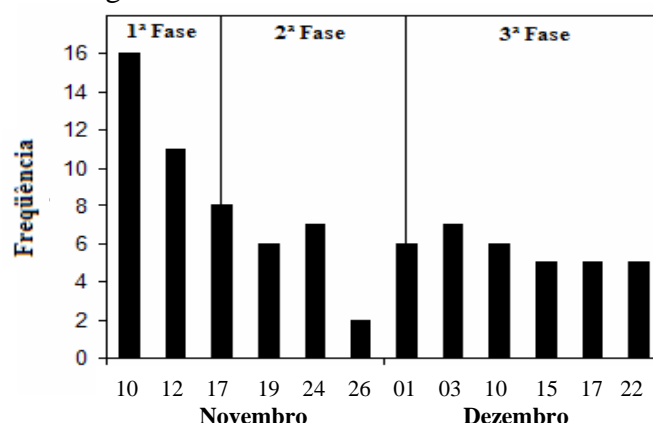


Figura 1: Gráfico da Frequência diária na oficina.

Durante a realização da primeira e segunda fase ocorreu elevado grau de interatividade principalmente durante a discussão das soluções propostas, na utilização dos jogos computacionais e no trabalho com o fórum. Esta abordagem de promover o trabalho coletivo aumentou o desejo pela resolução os problemas propostos, acarretando o desenvolvimento cognitivo quanto a este quesito sem a percepção dos alunos.

As dificuldades apresentadas pelos alunos não diferiram muito das apontadas na literatura e presentes no ensino superior. Destacam-se a dificuldade de descrever formalmente condições lógicas em uma estrutura de seleção, por exemplo; desenvolver programas que envolviam determinados conteúdos de matemática; e em se adaptarem a formalização rígida da linguagem de programação.

Ao final da oficina os alunos concluintes passaram por uma avaliação do aprendizado através da resolução de problemas utilizando programas desenvolvidos por eles próprios. O resultado desta avaliação mostra que dos 5 alunos concluintes, 4 possuem capacidade satisfatória de programar.

Em um questionário de avaliação do curso aplicado aos alunos mostra que o interesse pela computação aumentou excitando até mesmo o desejo de cursar o nível superior e frequentar outros cursos dentro da área. Os alunos também informaram que a oficina contribuiu para um melhor aprendizado da matemática e da física, fato este que levanta a questão da interdisciplinaridade proposta no ensino de programação.

6. Considerações Finais

Apesar da elevada evasão, o resultado obtido na avaliação dos concluintes da oficina permite projetar as expectativas da inclusão do tema algoritmos e programação na matriz curricular. Neste caso, os alunos terão, em princípio, maior comprometimento com o processo considerando-se a obrigatoriedade de cumprimento e o professor terá disponibilidade para abordar melhor o conteúdo devido à carga horária superior. O mais importante, porém, da integração na matriz curricular é a possibilidade de fazer algum trabalho envolvendo temas de outras áreas de domínio.

A oficina permitiu comprovar que a idéia do GT-3 é factível, ou seja, que alunos do ensino médio são capazes de desenvolver tal conteúdo sem necessariamente estarem inseridos num curso técnico, o que pode estimular o professor por aproveitar a nova competência do aluno para promover o aprofundamento de conteúdos de sua disciplina e, conseqüentemente, a interdisciplinaridade.

Além disso, a oficina contribuiu para ampliar o interesse pela área da computação e contribuiu para o aprendizado de outros conteúdos conforme relatado pelos próprios alunos e que poderá ainda ser mais explorado.

A próxima etapa deste trabalho será aplicar a metodologia apresentada no contexto da matriz curricular durante o ano de 2005, ainda no segundo ano do ensino médio, de modo a validar a projeção proposta. Também está desenvolver um ambiente que represente a metodologia proposta e integre as ferramentas utilizadas auxiliando nas aulas presenciais e possibilitando o ensino a distância.

Referências

- BAEZA-YATES, R. A. (1995) "Teaching Algorithms". SIGACT News 26, 4, 51-59.
- BAGERT, D.J., CALLONI, B.A. (1999) "Teaching Programming Concepts Using an Icon-Based Software Design Tool". IEEE Transactions on Education, Volume 42, ed. 4.
- BURTON, P. J. e BRUHN, R. E. (2003) "Teaching Programming in the OOP Era". SIGCSE Bulletin, 35(2): 111-114.
- CASTELLS, M., (1999), A era da informação: Economia, sociedade e cultura. Vol 1. A sociedade rede, Edit. Paz e Terra, São Paulo.
- CHAVES DE CASTRO, T., CASTRO JÚNIOR, A., MENEZES, C., BOERES, M. e RAUBER, M. (2003) "Utilizando Programação Funcional em Disciplinas Introdutórias de Computação", Anais do WEI 2003, Brasil.
- DELGADO, C., XEXEO, J. A. M., SOUZA, I. F., CAMPOS, M., RAPKIEWICZ, C. E. (2004) "Uma Abordagem Pedagógica para a Iniciação ao Estudo de Algoritmos". *XII Workshop de Educação em Computação (WEI'2004)*. Salvador, BA, Brasil.

FALKEMBACH, G. A. M., AMORETTI, M. S. M., TAROUCO, L. R., VIERO, F. (2003) “Aprendizagem de Algoritmos: Uso da Estratégia Ascendente de Resolução de Problemas”. 8º Taller Internacional de Software Educativo. Santiago, Chile.

FLORES, A. M. (2002) “A Informática na Educação: Uma Perspectiva Pedagógica”. *Monografia – Universidade do Sul de Santa Catarina (1996)*. Disponível em: <http://www.hipernet.ufsc.br/foruns/aprender/docs/monogr.htm> (nov/2002).

GRIES, D. (1974) “What should we teach in an introductory programming course?”. In *Proceedings of the fourth SIGCSE technical symposium on Computer science education*, ACM Press, pp. 81–89.

HENDERSON, P. (1987) “Modern Introductory Computer Science”. In *Proceedings of the Eighteenth SIGCSE Technical Symposium on Computer Science Education*, ACM Press, pp. 183-190.

HENDERSON, P. B. (1986) “Anatomy of an Introductory Computer Science Course”. In *Proceedings of the Seventeenth SIGCSE Technical Symposium on Computer Science Education*, ACM Press, pp. 257-264.

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

LACERDA, L. B. (2002) *Construção da empregabilidade em informática: um estudo de caso sobre o curso de informática da UFRJ*. Dissertação (Mestrado em Engenharia de Sistemas e Computação) - Universidade Federal do Rio de Janeiro. 159 f.

LOPES, J. J. (2004) “A Introdução da Informática no Ambiente Escolar”. Disponível em: <http://www.clubedoprofessor.com.br/artigos/artigojunio.htm> (dez/2004).

MARION, W. (1999) “CS1: What Should We Be Teaching?”. *Annual Joint Conference Integrating Technology into Computer Science Education , Working Group Reports From Iticse on Innovation and Technology in Computer Science Education*, pp 35-38, Polônia.

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

ON COMPUTING CURRICULA IEEE COMPUTER SOCIETY ASSOCIATION FOR COMPUTING MACHINERY, T. J. T. F. *Computing Curricula (2001) Computer Science*. Tech. rep., IEEE-CS and ACM, December 2001.

PEREGO, C. A., LISBOA, M. L. B., BERTAGNOLLI, S. C. (2002) “A Migração de Pascal para Java: Problemas e Propostas de Solução”. *X Workshop de Educação em Computação – WEI 2002*. Florianópolis, SC, Brasil.

SACHS, J. (2000) “O mapa da exclusão tecnológica”. *Jornal do Brasil*, Rio de Janeiro, 16/jun/2000.

SCHULTZ, M. R. O. (2003) *Metodologias para Ensino de Lógica de Programação de Computadores*. Monografia de Especialização. Universidade Federal de Santa Catarina (UFSC), Florianópolis, SC, Brasil.