

Desenvolvimento de OAs como Estratégia para o Ensino Técnico de Computação.

Luiz F Dias Pereira ¹, Fabio Lapolli ², Fábio Ferrentini Sampaio ², Claudia L. R. Motta ², Carlo Emmanoel Tolla de Oliveira ².

¹ Colégio Pedro II
Rio de Janeiro – RJ – Brazil
{Luiz.dias.pereira@gmail.com}

² NCE– Universidade Federal do Rio de Janeiro (UFRJ)
Rio de Janeiro – RJ – Brazil
{lapollimaster@gmail.com}, {ffs,claudiam,cetoli}@nce.ufrj.br}

Abstract: *The way that young people deal with technology makes many students, who attend to Computer Science lectures, losing their motivation when confronted with teaching methods and the high abstraction level of the contents. This lack of interest leads to learning difficulties and class evasion. To minimize these effects, a teaching approach where learners can engage in a real-life software development project is proposed, collaborating in and between groups to create learning objects, which will be able to be used afterwards by other students.*

Resumo: *A familiaridade com que os jovens lidam com a tecnologia faz com que muitos alunos, ingressos em cursos de computação, sintam-se desmotivados ao se deparar com o nível de abstração dos conteúdos e com os métodos utilizados para transmiti-lo. Esta falta de interesse leva à dificuldade na aprendizagem e à evasão das aulas. Para minimizar estes efeitos é proposta uma abordagem de ensino onde os aprendizes possam se engajar em um projeto real de software, colaborando em e entre grupos para a criação de objetos de aprendizagem que poderão ser utilizados posteriormente por outros estudantes da escola.*

1. Introdução

As disciplinas técnicas básicas de computação habitualmente não recebem a atenção necessária por parte dos professores. Segundo GAL-EZER (1998), os educadores na área de computação não são os mais preocupados com a utilização de ferramentas computacionais. A maioria utiliza a Internet no ensino à distancia ou para distribuição de notas, exercícios e material didático, isto é, como meio de comunicação extra-classe com os seus alunos. O interesse dos jovens na tecnologia que os cerca contraposto por métodos antiquados de ensino de computação, que trabalham com exemplos pouco motivadores, privilegiando a abstração, contribuem para o aumento da evasão e do desinteresse dos aprendizes. Com o objetivo de aumentar o interesse e a autonomia dos alunos e potencializar a aprendizagem, é proposto o desenvolvimento de um curso de ensino de programação, onde os participantes integrarão um ateliê para construção de objetos de aprendizagem (OA).

O artigo está organizado da seguinte maneira; na Seção 2 revisamos algumas soluções encontradas na literatura; na Seção 3 elicitamos a proposta; na Seção 4 relatamos

a metodologia; na Seção 5 descrevemos alguns resultados da pesquisa e na Seção 6 descrevemos as conclusões.

2. Revisão da Literatura

A comunidade acadêmica vem buscando soluções que possam melhorar a aprendizagem de disciplinas de computação, especificamente focando minimizar as dificuldades do estudo de programação. Ao longo de nossa pesquisa nos foi possível acompanhar muitas soluções cujo foco era o ensino de computação e todas elas puderam ser categorizadas em uma ou mais abordagens de ensino, das quais identificamos as seguintes: Simplificação de conteúdo, projetos de softwares, aprendizagem colaborativa.

É apresentado no Quadro 1, a relação entre propostas pesquisadas e abordagens de ensino e modelos para colaboração.

Quadro 1. Comparação entre propostas e métodos utilizados

	Conteúdo Simplificado	Projeto real de <i>software</i>	Colaboração
Ambiente de Apoio ao Aprendizado de Programação. (Almeida et al., 2002)	✓		
JAVATOOL: Uma Ferramenta Para Ensino De Programação. (Mota et al 2008)	✓		
Portugol IDE – Uma ferramenta para o ensino de programação. (Manso et al 2009)	✓		
O uso do Lego Mindstorms no apoio ao Ensino de Programação de Computadores. (Vahldick et al.2009)	✓		
Protagonismo Juvenil no Ensino de Computação da UFS (Costa et al., 2004)		✓	
Bancada Experimental Robótica para o Ensino de Computação (Alves et al., 2008)		✓	✓
B-Learning em disciplinas introdutórias de programação. (Sobral, 2008)			✓
Análise de um Estudo de Caso para Aprendizagem de Programação em Grupo. (Castro et al ,2009)			✓
Ensino de Programação em um Ambiente Colaborativo. (Marselino et. al. 2008)			✓
Um sistema de Aprendizado Colaborativo de Programação Baseado em Agentes Chamado Learn In Group. (Faria et al.2008)			✓

Nos tópicos abaixo tentamos elicitare o que cada estratégia se propõe a fazer.

2.1–Simplificação de Conteúdo

Boa parte das abordagens utilizadas para o ensino de computação se vale de tentativas de simplificação de conteúdo para facilitar o entendimento dos conceitos, diminuindo o grau de abstração, e posteriormente fazer uso desses conceitos como subsunçores na aquisição de novos conhecimentos.

O conhecimento pode ser simplificado (1) selecionando-se, de todo o conteúdo, as idéias fundamentais e primárias para o melhor entendimento da disciplina. Segundo Brunner(1960), idéias fundamentais são invariantes que podem ser ensinadas e entendidas em qualquer fase do aprendizado do indivíduo. Ele afirma ainda "Que qualquer assunto

pode ser ensinado eficazmente, de alguma forma intelectualmente honesta para qualquer criança em qualquer estágio de desenvolvimento”, com isso quer dizer que, uma vez determinada a invariância, os conceitos fundamentais, da disciplina a qual se deseja lecionar, essa poderá ser ensinada e posteriormente utilizada como ancora para o entendimento dos outros conceitos mais complexos ou mais abrangentes.

(2) Tentando eliminar a complexidade inerente ao processo de aprendizagem para permitir que o aprendiz concentre a maior parte de seus esforços no entendimento dos conceitos que lhe estão sendo apresentados. Para tanto, pode-se fazer uso de uma melhor contextualização, de softwares que facilitem o processo de aprendizagem e outras estratégias que permitam ao aluno abstrair informações secundárias e se concentrar no conteúdo apresentado pelo educador.

Embora simplificando o conteúdo se consiga diminuir o grau de abstração, não é possível garantir o aumento da motivação do aluno ou mesmo trabalhar melhor sua autonomia. A simplificação de conteúdo permite que os conceitos estejam em um formato que facilite o entendimento, mas ainda é necessário fazer com que o aluno deseje aprender os conceitos a ele apresentados e fazer com que ele se sinta confiante para atingir essa meta.

2.2 – Projetos Reais de Software

Outra abordagem para o ensino de computação são os projetos Reais, onde os educadores elaboram o contexto e a idéia de um projeto o qual um ou mais alunos tomarão parte. Assim, pretende-se aumentar, através da contextualização, o interesse do aprendiz, seu empenho e participação no processo de aprendizagem.

Fazendo uso de projetos reais de software permite-se que o aprendiz produza benefícios para sua comunidade, uma vez que o produto desenvolvido através do projeto poderá ser usado por seus membros. Exatamente por isso, segundo Costa (2004), o aprendiz que protagoniza esse processo, se sente valorizado e estimulado sabendo que está proporcionando benefícios e softwares que terão um uso real dentro de sua comunidade. Essa estratégia propicia ao aluno a oportunidade de se envolver em uma atividade que lhe dará reconhecimento por parte de seus pares. Aumentando sua motivação e comprometimento, principalmente por que este aluno sabe que suas ações estarão sendo observadas e avaliadas. No entanto, se não fizermos esse aprendiz acreditar ser capaz de vencer os desafios a ele impostos, mesmo motivado ele irá desistir e não cumprirá os objetivos propostos.

2.3 – Aprendizagem Colaborativa

Muitos dos esforços em melhorar a aprendizagem se apóiam em colaboração e troca de experiências entre os aprendizes. Ao agruparem-se em equipes, os alunos dividem os problemas e somam os esforços, conseguindo resolver desafios cujo nível de dificuldade seria intransponível para um indivíduo trabalhando de forma isolada.

Para Vygotsky (1991) “O sujeito é ativo, ele age sobre o meio. Para ele, não há a natureza humana, ou a essência humana. Somos primeiro sociais e depois nos individualizamos.”

Por tanto, é natural agir em grupo, agir socialmente. Da mesma forma é natural participar em grupo do processo de aprendizagem, colaborando para alcançar os objetivos. Os aprendizes sentir-se-ão, a medida que o grupo avança em seus objetivos, confiantes de sua capacidade. Esta confiança atuará como fator motivacional para que o aprendiz persista em alcançar a meta estipulada, e conseqüentemente obtenha êxito em seu aprendizado.

Entretanto, apenas a confiança pode não ser o suficiente, se o conteúdo, a ser compreendido, estiver além do alcance do grupo ou se este, não se sentir inserido no contexto do trabalho. Nestas situações, os aprendizes podem acabar por desistir do aprendizado, transformando o tempo do grupo em um momento de troca de amenidades e conversas particulares.

3. Proposta

Do exposto, propomos trabalhar as abordagens citadas no Quadro 1 apresentado na Seção 2, adicionando-se a essas a possibilidade de contribuição e avaliação entre grupos de alunos e utilizando boas práticas de desenvolvimento ágio para nortear a colaboração, como demonstrado no Quadro 2.

Quadro 2. Contribuição da proposta

	Conteúdo Simplificado	Projeto real de software	Colaboração	Métodos Ágeis para Colaboração	Contribuição/avaliação entre equipes
Proposta	√	√	√	√	√

Nossa hipótese é de que ao utilizar as supracitadas estratégias através da criação de um ateliê para desenvolvimento de objetos de aprendizagem, onde os aprendizes formarão grupos de trabalho que colaboram para elaboração dos OAs e contribuem e avaliam uns aos outros, poderemos aumentar a autonomia e a motivação dos alunos, minimizar a evasão das aulas e potencializar a aprendizagem.

É possível observar na Figura 1 a relação entre a proposta e as teorias que a fundamentam e como são trabalhados os aspectos individuais e coletivos do processo de aprendizagem.

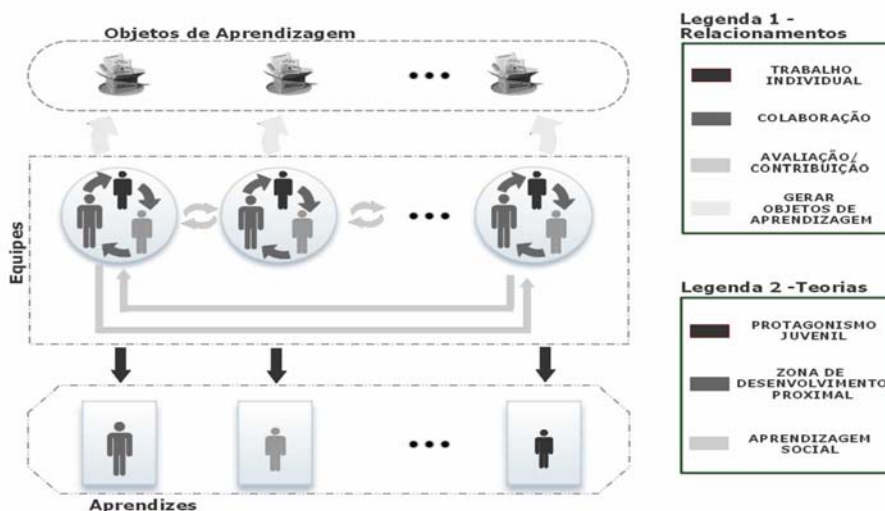


Figura 1. Modelo de Interação

Quando trabalhamos os aspectos individuais nos apoiamos no (1) Protagonismo Juvenil (Costa, 2004) que afirma ser possível despertar o interesse dos aprendizes ao torná-los partícipes de um projeto real que resultará em benefícios para a comunidade da qual o mesmo faz parte, aumentando com isso sua motivação, e (2) na crença na auto-eficácia (Bandura, 1977), a qual afiança que ao trabalhar em um contexto o qual domina, o aprendiz se vê capaz de solucionar os problemas e com isso se envolve e dedica com mais afinco às tarefas que lhe são imputadas.

Quando trabalhamos os aspectos coletivos nos apoiamos na idéia de (1) Zona de Desenvolvimento Proximal (Vygotsky,1991) que postula ser possível para o aprendiz desenvolver seu conhecimento com maior rapidez e abrangência, quando o mesmo trabalha colaborativamente com um orientador ou com outros aprendizes. E também (2) na teoria da aprendizagem social (Bandura, 1977) que afirma que as pessoas ao observarem comportamentos, que resultem em estímulos positivos, tendem a dublá-los, copiando-os com o objetivo de receber os mesmos estímulos positivos observados. Para Bandura (1977), o aprendizado seria excessivamente trabalhoso, para não mencionar perigoso, se as pessoas dependessem somente dos efeitos de suas próprias ações para informá-las sobre o que fazer. Por sorte, a maior parte do comportamento humano é aprendido pela observação de modelos comportamentais. Ao observar os outros, uma pessoa forma uma idéia de como novos comportamentos são executados e, em ocasiões posteriores, esta informação codificada serve como um guia para a ação.

Para que as equipes colaborem, como sugerido no modelo de iteração exposto na figura 1, propomos que, as mesmas, utilizem técnicas de desenvolvimento ágil. Desta forma, tentamos garantir que os aprendizes estivessem realmente compartilhando o aprendizado ao trabalharem no Ateliê. Nesse escopo, a metodologia ágil de Extreme Programming (XP), viabiliza uma maior troca de conhecimentos entre os aprendizes através da observação de modelos comportamentais para atender as demandas variacionais de modelagem das funcionalidades do aplicativo. Ela contribui também para o aumento da autonomia dos alunos, tornando-os participantes ativos dentro de todas as etapas do processo de desenvolvimento.

O desenvolvimento dos OAs foi apoiado, em um conjunto de boas práticas em XP definido a seguir de acordo com Beck (2000):

Equipe Coesa (*Whole Team*): No projeto recomenda-se a composição de equipes não muito numerosas com um representante de cada área de desenvolvimento. A equipe é composta de alunos que se revezam entre os papéis do cliente, do programador, do designer e do designer instrucional. A constante troca de papéis, tem o propósito de além de permitir que os alunos estejam sempre em sintonia com o professor de forma coesa, objetiva também proporcionar maior motivação aos alunos no processo de aprendizagem ao fazer com que ele acompanhe o processo de desenvolvimento do início ao fim e assim perceba a importância daquele conhecimento que ele adquiriu e sua aplicação prática

Posse Coletiva (*Collective Ownership*): O código-fonte é compartilhado entre os desenvolvedores podendo ser modificado ao mesmo tempo e construído de forma colaborativa. O objetivo com isto é fazer a equipe conhecer todas as partes do sistema podendo ser feita a atualização do desenvolvimento através de repositórios de versões compartilhado pela equipe. Esse processo aplicado dentro da proposta de aprendizagem, permite aos alunos construir o conhecimento de forma colaborativa através do acesso ao modelo referencial de sucesso. Eles podem compartilhar soluções e estudar como esse modelo foi construído por outros alunos de seu grupo para que posteriormente elaborem seu próprio modelo. Esse processo capacita o aluno a perceber a possibilidade de replicar e adaptar o conhecimento a outras etapas do desenvolvimento, promovendo a autonomia no processo de aprendizagem, tornando-o aluno mais reflexivo.

Reuniões em Pé (*Stand-up Meeting*): Realização de reuniões de curta duração entre os membros da equipe para discutir tarefas realizadas e tarefas a serem realizadas pela equipe. Esse formato tem o objetivo de conseguir maior concentração dos membros do

grupo durante as reuniões. Esse modelo permite a constante interação entre alunos na definição e integração contínua de novas funcionalidades.

Integração Contínua (*Continuous Integration*): A integração de novas funcionalidades é realizada de maneira contínua e imediata evitando a possibilidade de conflitos e erros no código fonte, permitindo acompanhar o status real da programação pelo cliente, facilitando a identificação de ajustes a serem realizados nas funcionalidades do aplicativo. A integração contínua de novas funcionalidades acontece não só na programação, como também no design, a fim atender a proposta pedagógica.

4. Metodologia

Com o objetivo de validar a proposta foi oferecido um curso livre de programação orientada a objetos no Colégio Pedro II no Rio de Janeiro. A oferta de curso ocorreu através da Seção de Supervisão e Orientação Pedagógica (SESOP) que divulgou o mesmo entre as turmas do ensino médio. Foram oferecidas e preenchidas quarenta e duas (42) vagas para os estudantes do colégio. Para aceitar a matrícula foram seguidos os seguintes pré-requisitos: (1) Não ter conhecimento prévio de programação orientada a objetos, para esse corte avaliou-se a ficha de inscrição onde os alunos eram questionados a cerca de seus conhecimentos em informática. (2) A ordem de inscrição dos candidatos.

Observando as inscrições evidenciou-se que as vagas, em sua totalidade, foram preenchidas por alunos de duas turmas, A e B, do segundo ano do ensino médio integrado e para minimizar a contaminação optou-se por criar duas turmas no curso, preenchendo uma com os alunos selecionados da turma A e outra com os alunos selecionados da turma B. Que, com efeito dos cortes, ficaram com vinte e um (21) alunos cada.

De forma a por em prática o quase experimento optou-se por ministrar aulas segundo o método tradicional para a turma A, grupo controle, e aplicar a metodologia de ensino na turma B, grupo experimental.

No grupo controle, turma A, a cada aula, antes da expor o conteúdo, o professor dividia a turma em grupos com três (3) alunos. Então, ao longo do curso, os aprendizes receberam aulas sobre orientação a objetos e actionscript, onde foi demonstrada a sintaxe da linguagem e como programar, além disso, foi apresentado todo o funcionamento do programa Flash na versão CS3. Sempre intercalando a apresentação do conteúdo com exercícios de fixação que permitissem a prática de programação. Ao final do curso foi efetuada a avaliação dos exercícios feitos pelos aprendizes e ministrado um teste prático que tinha por objetivo por a prova os conceitos apresentados até então. Também foi pedido aos alunos que respondessem um questionário onde fariam sua avaliação pessoal do curso.

No grupo do experimento, turma B, na primeira aula, a turma foi dividida em equipes com três (3) integrantes, para essa divisão listamos os alunos em ordem, segundo seu rendimento no ano anterior (CR) e separamos esta lista em três grupos de 7 alunos, sorteando cada equipe com um alunos de cada grupo. Estas equipes trabalharam em conjunto ao longo de todo o curso dividindo os mesmos computadores. Foram, então, demonstrados os conceitos básicos sobre Objetos de Aprendizagem, orientação a objetos e actionscript, o suficiente para que o aprendiz compreendesse a linguagem que seria encontrada em suas pesquisas e projetos. Em um segundo momento, tendo demonstrado os supracitados conceitos, foi apresentada as equipes a proposta para compor um ateliê que iria desenvolver objetos de aprendizagem para o colégio. Também foi demonstrado o modelo para esse processo de desenvolvimento com as etapas que deveriam ser seguidas conduzindo ao cumprimento das metas de produção.

Ao longo de suas atividades no ateliê de objetos de aprendizagem, as equipes de desenvolvimento elegeram dois ou três temas que gostariam de trabalhar e os mesmos foram votados pela turma, que escolheu aqueles que consideravam mais interessantes. Como resultado desta votação os OAs trabalhados foram: Física – Atrito; História – Guerra fria; Biologia – Sistema Digestório; Biologia – Sistema Imunológico; Geografia – Placas Tectônicas; Química - Geometria molecular; Física – MUV. Os alunos com o apoio de um professor, convidado por eles, definiram o design instrucional dos objetos e detalharam o script do OA. Acreditamos que essas atividades, embora não diretamente relacionadas com a programação, permitem que os aprendizes comecem a participar mais de seu processo de aprendizagem aumentando a crença na autoeficácia e, conseqüentemente, aumentando sua motivação na fase de desenvolvimento.

Ao final do processo com o acompanhamento do professor orientador, os alunos desenvolveram, sem que lhes fosse ensinado, um objeto de aprendizagem interativo, programado na linguagem actionscript. Esse Objeto aliado a uma prova e um questionário idênticos aos que foram exigidos do grupo controle foram os instrumentos de avaliação utilizados no grupo experimental. Um exemplo do que foi desenvolvido é apresentado na Figura 2



Figura 2. OA sobre o Sistema Digestório

5. Resultados

Ao fim do curso foi possível avaliar os resultados obtidos pelo grupo controle e pelo grupo experimental, que podem ser observados na Tabela 1.

Tabela 1. Resultados das Avaliações

Grupo Controle				Grupo Experimental			
Equipes	Nº	Média	Trabalhos	Equipes	Nº	Média	Trabalhos
I	3	6	6	I	3	7	6
II	0	0	2.6	II	3	6.6	6
III	2	7	6.5	III	3	8.3	6
IV	0	0	4	IV	3	6.3	7
V	2	7.6	8	V	3	6.6	6
VI	0	0	3.3	VI	2	8	7
VII	2	7	4.6	VII	3	8	6

O que podemos inferir observando a Tabela 1 é que, a princípio, temos uma performance melhor entre as equipes do grupo experimental e que no grupo controle o resultado das médias sofre interferência pelo alto número de evasões, uma vez que apenas uma equipe conseguiu concluir o curso com os três integrantes enquanto no grupo experimental apenas uma não conseguiu.

É Fato que a diferença entre as médias das equipes do grupo controle e do grupo experimental não é acentuada, sendo os resultados até parecidos, no entanto, esse fenômeno é facilmente explicado uma vez que, em função do grande número de desistências do grupo controle, apenas os alunos resilientes persistiram até o fim do curso e esse processo de seleção natural, geralmente, implica em alunos com bom aproveitamento e comprometimento. Já no grupo experimental praticamente todos os alunos terminaram o curso, estando entre eles alunos com mais dificuldade e aqueles cujo desempenho normalmente é acima da média.

Podemos entender que, como as médias do grupo experimental ficaram dentro do limite do Colégio, embora não tenha havido um aumento significativo nas notas, temos indícios de que os alunos compreenderam os conceitos avaliados na prova e obtiveram desempenho aceitável embora pouco abaixo do esperado e mais importante, os alunos que normalmente desistiriam ante as dificuldades, persistiram e obtiveram resultados satisfatórios.

Foi feito ainda um levantamento de dados sob forma de pesquisa, que apontou para a satisfação dos alunos com o curso, aulas e com a forma como agiram durante o processo de aprendizagem. Apontando ainda para o interesse dos alunos em reproduzir, em outras aulas, o comportamento colaborativo predominante nas atividades do Ateliê.

Durante todo esse processo foi possível observar dois aspectos importantes na participação dos aprendizes em sala de aula. A motivação e mobilização para cumprir as metas estipuladas ao longo do curso e a autonomia necessária para tomar a frente e resolver os defeitos dos softwares que por ventura eles iriam desenvolver.

5.1 - Autonomia

Para comprovar o aumento da autonomia dos aprendizes, foram observadas informações referentes a quantidade de vezes nas quais os alunos requisitaram a ajuda do professor. Esses dados, anotados pelo professor, nos dão indícios de que a metodologia melhora a confiança dos alunos estimulando-os a buscar as repostas para suas dúvidas por conta própria. O gráfico abaixo demonstra o número de vezes nas quais os aprendizes solicitaram a ajuda do professor para retirar alguma dúvida.

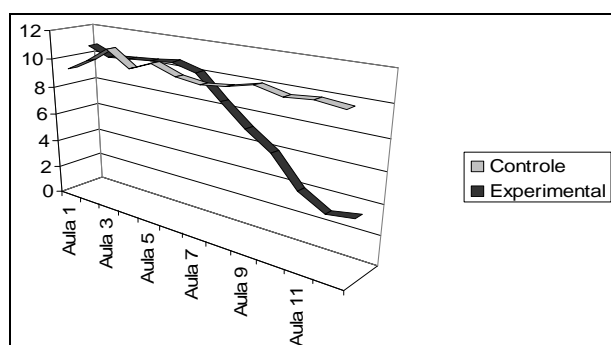


Gráfico 1 – Requisições ao professor

Pode-se observar que no Gráfico 1, que o número de vezes que os estudantes do grupo controle solicitaram ao professor não se alterou de forma significativa ao longo das aulas do curso. No entanto, no grupo experimental é perceptível a redução gradativa do número de vezes nas quais o professor é requisitado. Este fato nos dá indícios de que ao longo das aulas do curso os aprendizes do Ateliê desenvolveram confiança em suas habilidades e potencialidades o que os impulsionou a desvendar grande parte de suas dúvidas, deixando para o professor apenas as dúvidas realmente difíceis.

5.2 – Evasão

A comparação da taxa de evasão do grupo controle em relação ao grupo experimental nos mostrou que enquanto no primeiro a evasão chegou a 60% do total que iniciou o curso no grupo experimental ela não passou dos 5%. Em números significa que dos 21 alunos, em cada grupo, que iniciaram as atividades do ateliê, no grupo controle terminaram 9 e no grupo experimental 19.

Aceitamos esse fato como prova de que a prática de ensino ora aplicada foi capaz de aumentar a confiança e motivação dos aprendizes, tornando-os capazes de, de forma resiliente, se empenhar nas tarefas propostas ao longo do curso.

6. Conclusões

Com esse estudo concluímos ser possível incentivar os aprendizes a protagonizar sua aprendizagem, atribuindo ao professor o papel de mediador no processo de aquisição do conhecimento e dando ao aluno a autonomia necessária para que ele possa resolver os problemas envolvidos no aprendizado de programação. A metodologia de ensino se mostrou bem sucedida, uma vez que os alunos a ela expostos reagiram da forma esperada, ou seja, demonstraram um crescente aumento em sua autonomia, no que tange a tomar decisões sobre o que e como aprender, e em sua motivação, quando persistiram em aprender, o que lhes foi proposto, onde os alunos do grupo controle optaram por desistir.

Foi possível notar que no grupo controle, apenas alunos com algum grau de resiliência conseguiram levar o curso até o seu fim e mesmo nesses casos a falta de motivação e as dificuldades para entender o conteúdo era uma constante entre a maioria deles. Já os aprendizes do grupo experimental se mostraram envolvidos com a proposta do Ateliê e, a partir da metade das atividades do curso, demonstraram uma motivação e autonomia que os permitiram seguir adiante até a conclusão do curso, com o mínimo de desistências e um bom aproveitamento.

7. Referências

- Almeida, E. S., Costa, E. B., Silva, K. S., Paes, R. B., Almeida, A. A. M., Braga, J. D. H.: AMBAP: Um Ambiente de Apoio ao Aprendizado de Programação. In: Workshop de Educação Em Computação, Florianópolis, 2002
- Alves, S. F. R., Ferasoli Filho, H.: Bancada Experimental Robótica para o Ensino de Computação, Congresso de Iniciação Científica, São Carlos. Eventos da UFSCar, v. 4, p. 1195, 2008.
- Bandura, A.: Self-efficacy: Toward a Unifying Theory of Behavioral Change, *Psychological Review*, 1977.
- Beck, K.: *Extreme Programming Explained: Embrace Change*; Addison-Wesley, 2000
- Bruner, J. (1960). *O processo da educação*. S. Paulo: Companhia Editora Nacional.
- Castro, T., Fuks, H., Spósito, M. & Castro, A. Análise de um Estudo de Caso para Aprendizagem de Programação em Grupo. *IEEE-RITA: Revista Iberoamericana de Tecnologia del Aprendizaje*. ISSN: 1932-8540. V. 4, N.2, pp. 155-160., 2009
- Costa Neto, A, Cox, K. K.: *Protagonismo Juvenil no Ensino de Computação da Universidade Federal de Sergipe*, 2004
- Faria, E. S. J., Vilela, J. M.; Adán Coello, J. M. Um sistema de Aprendizado Colaborativo de Programação Baseado em Agentes Chamado Learn In Group. XIII

Workshop de Educação em Computação (WEI), Anais do XXV Congresso da Sociedade Brasileira de Computação, pp. 2278-2290, 22 a 29 de Julho 2005.

GAL-EZER, J. e HAREL, D. What (Else) Should CS Educators Know? Communications of The ACM, September 1998, Vol. 41, N°9, 77-84.

González, L. A. G., Ruggiero, W. V.: Um Modelo Conceitual para Aprendizagem Colaborativa Baseada na Execução de Projetos pela Web, IEEE-RITA, 2008.

Marcelino, Eduardo Rosalém; Rosatelli, Marta Costa – Ensino de Programação em um Ambiente Colaborativo – Workshop sobre Educação em Computação, XXVIII Congresso da SBC, 2008.

Manso, A. L. Oliveira , C. Marques - Portugal IDE – Uma ferramenta para o ensino de programação - PAEE'2009 - Project Approaches in Engineering Education - Guimarães , Portugal, Julho 2009.

Mota, M.P., Pereira, L.W.P., Fávero, E.L.: JAVATOOL: Uma Ferramenta Para Ensino De Programação. In: Workshop de Educação Em Computação, Belém do Pará, 2008

Sobral, Sónia R. 2008. B-learning em disciplinas introdutórias de programação. ed. 1. Guimarães: DSI-UMinho. Tese de Doutorado.

Vahldick, A., Benitti, F.B.V., Urban, D. L., Krueger, M. L., Halma, A.: O uso do Lego Mindstorms no apoio ao Ensino de Programação de Computadores, In: Workshop de Educação Em Computação, Bento Gonçalves, 2009

Vygotski, L. S.: A Formação Social da Mente. Livraria Martins Fontes Editora Ltda. São Paulo, 1991