

## **DO SCRATCH AO ARDUINO: UMA PROPOSTA PARA O ENSINO INTRODUTÓRIO DE PROGRAMAÇÃO PARA CURSOS SUPERIORES DE TECNOLOGIA**

**Francisco Édson Nogueira de Mélo – emelo@ifsc.edu.br**

Professor, DAMM, IF-SC(\*)

**Raimundo Ricardo Matos da Cunha – rcunha@ifsc.edu.br**

Professor, DAMM, IF-SC(\*)

**Dyonad Renan Scolaro – dyonad@gmail.com**

Graduando do CST em Mecatrônica Industrial, Bolsista PET Mecatrônica, DAMM, IF-SC(\*)

**Jhonatan Luiz Campos – jho\_campos@hotmail.com**

Graduando do CST em Mecatrônica Industrial, Bolsista PET Mecatrônica, DAMM, IF-SC(\*)

(\*) Departamento Acadêmico de Metal-Mecânica

Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina

Av. Mauro Ramos, 950, Centro

CEP: 88020-300 – Florianópolis-SC

***Resumo:** Esse artigo apresenta uma proposta diferenciada de abordagem ao ensino introdutório de Programação em Cursos Superiores de Áreas Tecnológicas. Dificuldades na elaboração de um raciocínio estruturado para a solução de um problema computacional, de compreensão de conceitos, e a complexidade dos ambientes de desenvolvimento são, frequentemente, obstáculos ao aprendizado. Tais problemas podem ser bastante amenizados pelo uso didático de novas ferramentas computacionais, construídas para facilitar a aprendizagem pela experimentação. O Scratch revela-se uma excelente ferramenta para o ensino de conceitos de Lógica de Programação, com uma interface visual que remete intuitivamente às principais estruturas de uma linguagem. A plataforma Arduino, por outro lado, é um vasto repositório de recursos para experimentação prática, e um laboratório de baixo custo para testes de programas em C/C++. O Scratch for Arduino (S4A) uniu essas ferramentas, resultando num ambiente de programação de aprendizado fácil, com recursos de interfaceamento para sensores e atuadores. A aplicação coordenada das ferramentas Scratch, S4A e Arduino, no contexto de um curso introdutório de programação, pode conduzir o aluno a um aprendizado facilitado, consistente e voltado a aplicações práticas.*

***Palavras-chave:** Arduino, Ensino, Programação, Scratch*

### **1. DIFICULDADES E FERRAMENTAS PARA “APRENDER FAZENDO”**

Na geração atual, percebe-se uma convivência maior das pessoas, com relação à necessidade de lidar e interagir com raciocínios ou conceitos lógicos baseados em forma digital, e também a evolução das interfaces homem-máquina tem auxiliado nesse processo,

facilitando a operação, barateando os custos de aquisição de dispositivos eletrônicos, e tornando-os mais difundidos.

Todavia, no contexto da programação, o que acontece no nível superior, a necessidade de interagir com o computador ainda exige um elevado nível de raciocínio abstrato e de lógica, os quais não são trabalhados na escola durante o ensino fundamental ou médio, além de que podem ser dificultados também pelo uso de linguagens codificadas em baixo nível.

A utilização do computador como uma ferramenta para a busca do aprendizado pela experimentação, levando à criação do próprio conhecimento, teve um marco histórico em 1967 com o desenvolvimento do LOGO (ROCHESTER, 2011) por Seymour Papert, criador da teoria construcionista e um dos “pais” da inteligência artificial.

Essas “possíveis e/ou prováveis dificuldades” associadas ao ensino de programação num curso de tecnologia têm motivado à busca de alternativas. Mais de quatro décadas após o surgimento do LOGO, muitos recursos didáticos foram desenvolvidos, destacando-se neste trabalho três importantes iniciativas, a saber: o Scratch, o Arduino, e o Scratch for Arduino. A expectativa dos autores é que tais iniciativas podem contribuir substancialmente para o aprendizado de lógica de programação, além de inculcar na disciplina ferramentas e cenários operacionais estimulantes para implementar um laboratório de experimentação. Essa proposta encontra respaldo nas ideias de Papert (PAPERT, 1980), onde a construção constante e continuada do conhecimento é incentivada pelas necessidades do aluno, que por sua vez as tem com base no seu contexto de experimentação própria.

## 1.1 Scratch

O Scratch é um ambiente gráfico de programação concebido e desenvolvido pelo “Lifelong Kindergarten Group”, do MIT, que permite o desenvolvimento de aplicativos que integram recursos de multimídia, de forma intuitiva. Seu objetivo primário é facilitar a introdução de conceitos de matemática e de computação, enquanto também induzindo o pensamento criativo, o raciocínio sistemático e o trabalho colaborativo (SCRATCH, 2011). A proposta do projeto é levar o estudante a um processo de aprendizado contínuo, numa sequência que envolve criação, experimentação, compartilhamento e reflexão, que leva a mais criação, resultando expansão de suas habilidades, conhecimento e competências, na “*espiral do pensamento criativo*” (Figura 1) descrita por Resnick (RESNICK, 2011). No contexto da Programação, os autores idealizam a espiral correspondente, conforme ilustrada na Figura 1b. Entende-se que a atividade de Programação, no seu processo criativo, exige que após a análise de um determinado problema, o mesmo seja codificado, executado, depurado, e documentado para que assim outros possam reproduzi-lo ou mesmo melhorá-lo. O processo se repete, iniciando o ciclo num estágio mais evoluído.

A Figura 2 mostra a interface de usuário do Scratch, durante a execução de um programa. À direita na figura, a imagem de um gato, de um carrinho simples e de duas rodas são “sprites”, ou objetos, para os quais se podem escrever “*scripts*” (blocos de programa). Um programa pode ter diversos objetos, que se movimentam sobre uma região denominada

"palco". Um conjunto de scripts também pode ser associado ao palco, com recursos para modificar o seu aspecto, ou trocar a imagem de fundo, por exemplo.

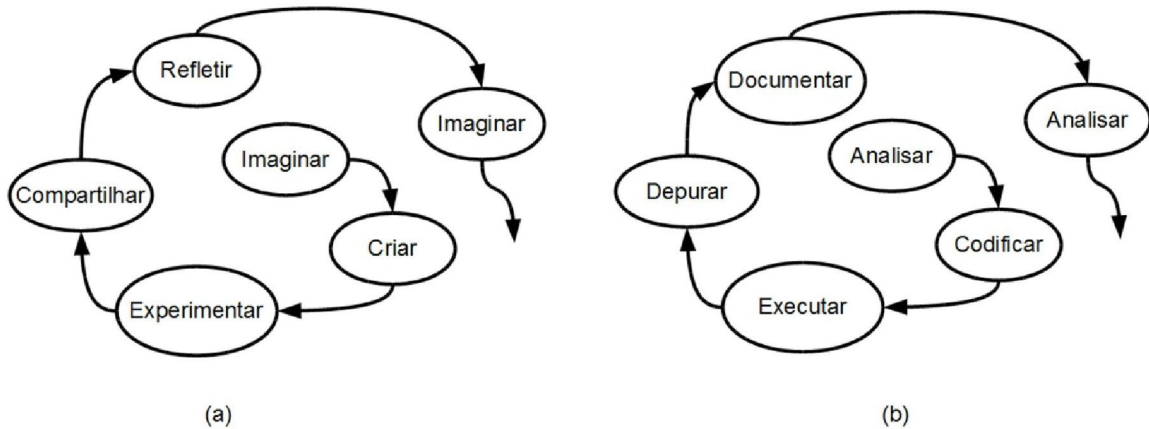


Figura 1 - Espiral do pensamento criativo, de Resnick (a) e Espiral correspondente, no contexto da programação (b).



Figura 2 - Ambiente de Desenvolvimento do Scratch.

A programação no Scratch é baseada em blocos, que implementam as estruturas básicas de um programa e se empilham para formar um script. Um aspecto muito interessante destes blocos é que o seu sistema de “encaixe” orienta o uso correto dos mesmos: uma operador relacional, por exemplo, tem um formato hexagonal, que pode ser inserido como condição de

teste em uma estrutura de decisão “se...”. Um conjunto bastante razoável de blocos de controle é disponível.

Um script do Scratch guarda similaridade com formas tradicionais de estruturas de algoritmo, com a indentação caracterizada nos blocos de estruturas de decisão e de repetição. Na figura 3, por exemplo, mostram-se aplicações dos blocos "se... senão" e o "sempre se..." do Scratch, com as suas formas correspondentes em algoritmo ("Português Estruturado"). O conceito de variáveis e matrizes unidimensionais também é introduzido de forma muito intuitiva, com comandos para a sua criação e blocos para operações básicas sobre elas.

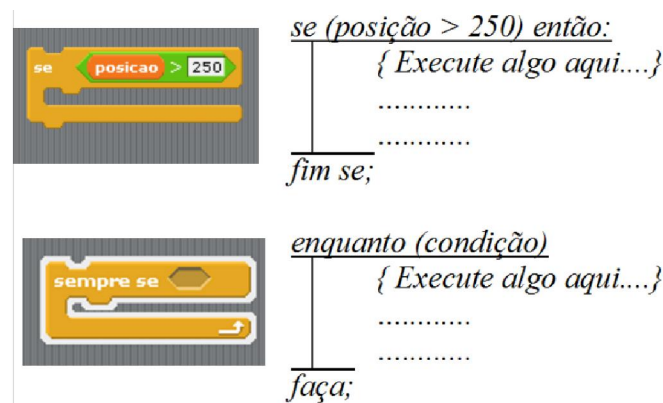


Figura 3 - Blocos do Scratch e fragmentos de algoritmo.

Segundo Resnick (RESNICK *et al.*, 2011), as características de “chão baixo” (que implica em facilidade de aprendizado inicial) e “teto alto” (implicando recursos avançados) preconizadas por Papert estão presentes no Scratch, e são complementadas pelo “espaço amplo”: o Scratch permite o desenvolvimento de aplicações de variados tipos como narração de histórias, recursos de animação e de som, recursos matemáticos e outros.

## 1.2 Scratch for Arduino

Sendo uma plataforma aberta, o Scratch deu origem a diversos outros ambientes de desenvolvimento, mais ou menos modificados do original, e voltados a diferentes objetos de aplicação. O Scratch for Arduino, ou S4A (CITILAB, 2011) é uma solução que merece destaque: é uma plataforma simples de desenvolvimento capaz de interagir com o "mundo exterior" ao PC, monitorando sensores e acionando atuadores diversos. Este recurso deverá atrair atenção e curiosidade do estudante, propiciando melhor aprendizado.

No ambiente S4A, o microcomputador deve ser conectado a uma placa de Arduino (descrita na seção seguinte), ou compatível, especialmente programada para impor funções específicas em seus pinos enquanto mantém comunicação com o sistema S4A/PC. Após reconhecer o Arduino conectado, o S4A disponibiliza um conjunto de blocos para interação com a placa, como parcialmente mostrado na Figura 4, no exemplo de um script simples que faz piscar um led conectado ao pino 13 do Arduino.

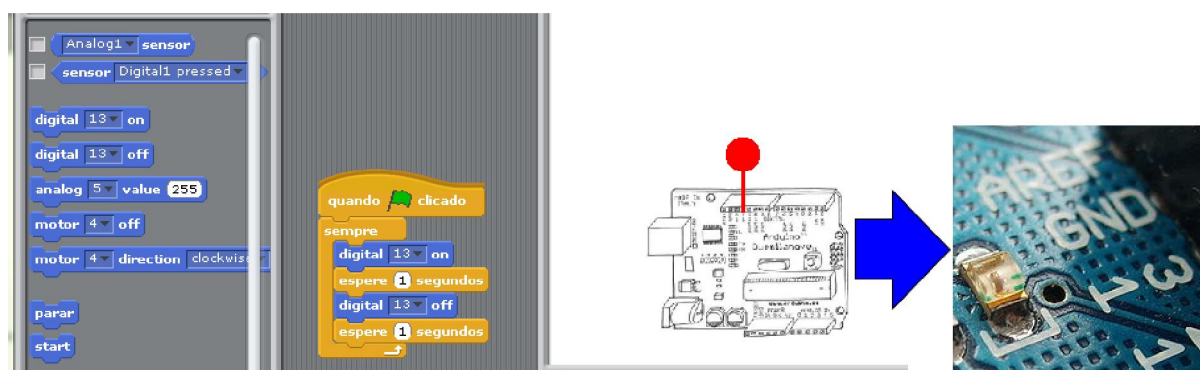


Figura 4 - Programa do S4A para piscar um led na placa do Arduino.

### 1.3 Arduino

O Arduino é descrito pelos seus construtores como “uma plataforma de prototipação de fonte aberta, baseada em hardware e software fácil de utilizar. É planejada para artistas, designers, hobbystas e qualquer um interessado em criar ambientes ou objetos iterativos” (ARDUINO, 2011). As suas características, e uma ativa comunidade na internet, lhe têm garantido um forte crescimento na sua base de usuários.

O site do Arduino, em <http://arduino.cc> oferece um excelente suporte ao sistema, com seções de referência, e inclui uma seção denominada “Playground”, em que conceitos de programação e de circuitos são explicados de forma simplificada.

#### Hardware do Arduino

Uma placa de Arduino é construída sobre um microcontrolador e um dispositivo de interface com o PC, para programação e comunicação. Nas versões mais populares da placa, estes componentes são, respectivamente, um AVR ATmega328, da ATMEL, e o CI FT232, da FTDI. O projeto da placa é bastante simples, e a maior parte da mesma ocupada pela conexão dos pinos do microcontrolador a um conjunto de conectores dispostos de uma forma bem definida, segundo o padrão do Arduino. Um outro aspecto importante é que o padrão Arduino permite a conexão de módulos periféricos, denominados “shields” (“escudos”), disponíveis comercialmente, para expansão do sistema, atendendo a uma grande variedade de aplicações.

A Figura 5 mostra uma placa de Arduino (a), na versão “Duemilanove” e uma placa compatível, denominada uCduino (b), desenvolvida por um dos autores (o Prof. Édson Mélo) para compatibilidade com matrizes de contato e com os seus módulos periféricos (c), formando um conjunto mais adequado aos recursos de sua prática pedagógica.

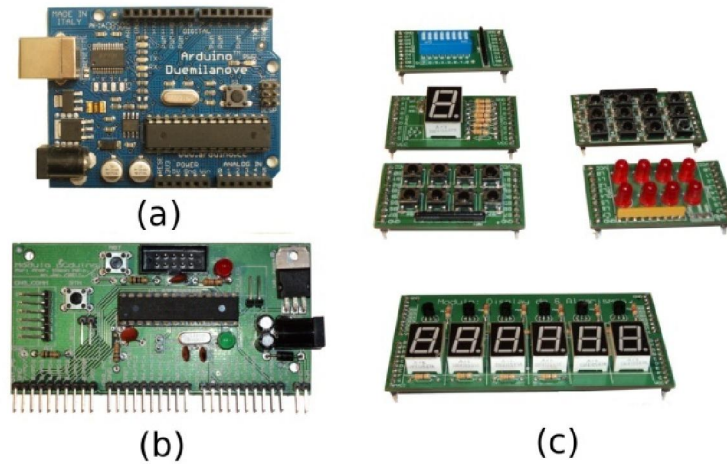


Figura 5 - Placas para experimentação com o Arduino.

### Ambiente de Desenvolvimento do Arduino

O ambiente de desenvolvimento do Arduino oferece uma interface de usuário simplificada. A linguagem do padrão, denominada Wiring, é derivada de C/C++, por expansão de seus recursos. A Figura 6 mostra o algoritmo (a) e a codificação em linguagem C (b) de um programa bastante simples, para acender e apagar alternadamente o led do pino 13 da placa. Verifique-se a “tradução” quase imediata, didática, do algoritmo elementar ao código-fonte em C, devido às funções disponíveis, adequadamente projetadas e nomeadas

```

Setup:
  |   Configura pino 13 para saída;
Fim_Setup.

Loop:
  |   “Acende o led”;
  |   “Espera 1 segundo”;
  |   “Apaga o led”;
  |   “Espera 1 segundo”;
fim_loop.
    
```

(a)

```

pisca
/*
  Pisca...
*/
void setup() {
  pinMode(13, OUTPUT); // Configura saída..
}

void loop() {
  digitalWrite(13, HIGH); // Acende o led
  delay(1000);           // Espera 1s
  digitalWrite(13, LOW); // Apaga o led
  delay(1000);           // Espera 1s
}
    
```

(b)

Figura 6 – Implementação de um exemplo simples no Arduino.

A descrição do Arduino, e o exemplo mostrado até aqui expõem a característica de “chão baixo” do Arduino, mas não se deve perder de vista o seu “teto alto”: os recursos do

microcontrolador Atmega328 podem ser plenamente explorados, e o desenvolvimento para ele pode ser feito com o uso de poderosas ferramentas de desenvolvimento.

## 2 UMA ABORDAGEM AO ENSINO INTRODUTÓRIO DE PROGRAMAÇÃO

Consideradas as dificuldades encontradas nas disciplinas iniciais de programação e as ferramentas amigáveis de desenvolvimento apresentadas (Scratch, S4A e Arduino), que oferecem melhores condições de aprendizagem dos conceitos básicos e abstratos de programação, propõe-se aqui uma sequência de abordagem ao tema, com a introdução gradual daqueles conceitos em contextos bastante “visuais”, práticos e estimulantes, evoluindo gradualmente para o tratamento de regras de sintaxe e estruturas mais complexas de dados.

### 2.1 Primeira Etapa: Scratch

Propõe-se aqui que o estudante tenha um primeiro contato com um ambiente de programação na interface amigável, simples, do Scratch. Enquanto fará os seus primeiros projetos, de pequenas animações e experimentos com multimídia, adquirirá conceitos como programa, estruturas de decisão e de repetição, variáveis, operadores, e outros, sempre experimentando-os dentro do ambiente intuitivo do Scratch.

### 2.2 Segunda Etapa: Scratch for Arduino (S4A)

As possibilidades de aplicação de interfaceamento com o “mundo real” serão tratados neste estágio. Aqui, a prática no uso do Scratch e o uso de um sistema computacional mínimo externo (Arduino) permitirá estabelecer conceitos como entrada, saída, sinais digitais e analógicos, em uma série de aulas práticas, com a implementação de aplicativos de acesso a sensores e comando de equipamentos básicos, como lâmpadas, motores, relés e outros.

### 2.3 Terceira Etapa: Arduino e Linguagem C

O Arduino, complementado por alguns elementos básicos de circuito, é uma plataforma bastante completa para experimentação com linguagem C, aplicável ao controle de sistemas simples, como leitura de sensores e acionamento de motores em pequenos automatismos, interfaces “homem-máquina”, ou dispositivos de supervisão e alarme. Observe-se aqui que nesta etapa os recursos do conjunto formado pelo kit uCduino e pelos seus módulos periféricos permitem uma grande variedade de experimentação.

### 2.4 Continuidade

A continuidade do aprendizado do aluno segue pelo tratamento da linguagem C em um ambiente de desenvolvimento para C/C++, explorando recursos da linguagem para plataformas de maior desempenho, e as possibilidades de E/S, de armazenamento, e outras, de um microcomputador PC.

### 3 EXEMPLO: PROJETO PARA IMPLEMENTAÇÃO NAS TRÊS FERRAMENTAS

Uma demonstração da aplicação das três ferramentas da abordagem proposta é apresentada aqui, na forma de um elevador didático simples, do tipo “monta-cargas”.

Inicialmente, o problema é proposto como um exercício de aplicação do Scratch, cuja solução é parcialmente mostrada na Figura 7. Ali, vê-se um “objeto” elevador, que deve se mover entre dois sensores, conforme comandado por dois botões, para subida e descida do elevador. Um botão de parada de emergência deve também ser verificado.

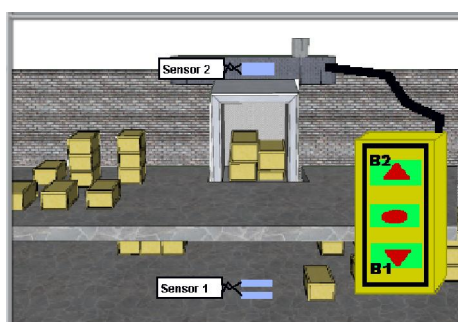


Figura 7 - Solução do elevador no Scratch.

Após o tratamento do Scratch for Arduino no curso, o problema do elevador é trazido novamente à apreciação do estudante. Ele agora saberá que o elevador real pode ser controlado pelo conjunto **computador + Arduino**, e poderá programar este controle. O equipamento, mostrado na Figura 8, com os seus detalhes técnicos como sensores de posição (b) e relés de comando do motor (a) são discutidos. Na interface de programação do S4A, o estudante deverá utilizar blocos de leitura de sensores digitais e de comando de saídas digitais para controlar o elevador.

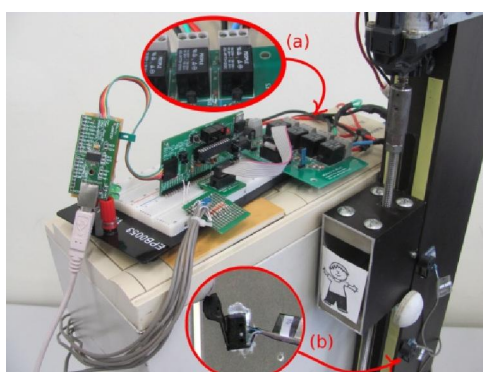


Figura 8 – Elevador didático para exercícios de programação.

Finalmente, com os seus primeiros conhecimentos de linguagem C, o estudante deverá desenvolver uma versão do elevador controlada apenas pelo Arduino.



#### 4 IMPLEMENTAÇÃO DA PROPOSTA E PRIMEIROS RESULTADOS

A sequência proposta, de introdução à programação em C, vem sendo adotada em um projeto piloto, em uma turma de trinta e quatro (34) alunos do Curso Superior de Tecnologia em Mecatrônica Industrial, do Campus Florianópolis–IF-SC, no primeiro semestre do curso, na disciplina de Programação I, que tem como objetivo a resolução de problemas de programação em linguagem C, de forma organizada, lógica e coerente.

As primeiras avaliações têm demonstrado um resultado positivo, verificando-se o interesse aumentado dos alunos pelas possibilidades de experimentação oferecidas pelas novas ferramentas. Os recursos de multimídia permitidos pelo Scratch e as interfaces de E/S do Arduino motivaram alunos, de forma que eles desenvolveram aplicativos interessantes como “tocadores de mp3”, automatismos simples, jogos elementares, e uma “casa inteligente” controlada por PC.

Uma pesquisa realizada entre os alunos do curso na disciplina de Programação I verificou uma aceitação positiva, conforme pontuaram e resumiram que:

- a metodologia implementada proporcionou uma maior contextualização com a mecatrônica, já nas primeiras fases, e isso aproxima os alunos do curso.
- as ferramentas atuais, comparado ao ambiente de desenvolvimento anteriormente utilizado, proporcionam uma maior facilidade de interação, por ser um programa “visual”, isso facilita a aprendizagem da lógica de programação. Importante frisar que o raciocínio lógico inerente à disciplina de programação, é uma das principais dificuldades apontadas pelos alunos.

#### 5 CONCLUSÕES

Os autores têm a expectativa de que os resultados da disciplina tenham reflexos nos semestres posteriores do CST em Mecatrônica Industrial, destacando os seguintes aspectos:

- uma melhor compreensão dos conceitos básicos envolvidos no processo de programação;
- os recursos do Scratch permitem o desenvolvimento de soluções em áreas diversas, incluindo multimídia, oferecendo possibilidades de aplicação;
- uma capacidade de visão “além do programa”, da sua aplicação num contexto de solução de um problema, interfaceando com sistemas de hardware externos;
- o conhecimento do Arduino no início do curso possibilitará aos alunos aplicá-lo em outras disciplinas, na aquisição simples de sinais e no comando de circuitos elétricos e eletrônicos básicos. No módulo IV do curso, o aluno terá uma disciplina de Sistemas Digitais, onde o microcontrolador Atmega328P será estudado em mais detalhes.

#### 6 REFERÊNCIAS BIBLIOGRÁFICAS

ARDUINO. ARDUINO. Disponível em <<http://arduino.cc>>. Acesso em 18 jun. 2011.

CITILAB. **Project Scratch: Scratch for Arduino (S4A)**. Disponível em <[http://seaside.citilab.eu/scratch?\\_s=0taALcOIKGKS3BcQ&\\_k=C5VhZQT5TKt8Dveq](http://seaside.citilab.eu/scratch?_s=0taALcOIKGKS3BcQ&_k=C5VhZQT5TKt8Dveq)>. Acesso em 18 jun. 2011.

PAPERT, Seymour. **MINDSTORMS: Children, Computers, and Powerful Ideas**. New York: Basic Books Inc. Publishers, 1980. 230 p.

RESNICK, Mitchel. **Sowing the seeds for a More Creative Society**. Disponível em <<http://web.media.mit.edu/~mres/papers/Learning-Leading-final.pdf>>. Acesso em 18 jun. 2011.

RESNICK, Mitchel; MALONEY, J.; HERNANDEZ, A. M.; RUSK, N.; EASTMOND, E.; BRENNAN, K.; MILLNER, A.; ROSEBAUM, A.; SILVER, J.; SILVERMAN, B.; KAFAI, Y. **Scratch: Programming for Everyone**. Disponível em: <<http://web.media.mit.edu/~mres/scratch/scratch-cacm.pdf>>. Acesso em 18 jun. 2011.

ROCHESTER, Clem Rutter. **LOGO (Programming Language)**. Disponível em: <<http://en.wikipedia.org/wiki/LOGO>>. Acesso em 3 jul.2011.

SCRATCH. **ABOUT Scratch (Scratch Documentation Site)**. Disponível em: <[http://info.scratch.mit.edu/About\\_Scratch](http://info.scratch.mit.edu/About_Scratch)>. Acesso em 18 jun. 2011.

## FROM SCRATCH TO ARDUINO: A PROPOSAL FOR INTRODUCTORY TEACHING OF PROGRAMMING IN UNDERGRADUATE COURSES OF TECHNOLOGY

***Abstract:** This paper presents a differentiated approach to the introductory teaching of Programming in Technology Colleges. Difficulties in elaborating a structured reasoning for solving a computing problem, understanding of concepts, and the complexity of development environment are often obstacles to learning. Such problems can be quite softened by the didactic use of new computational tools, built for facilitating learning by experimentation. Scratch reveals itself an excellent tool for teaching of programming logic concepts, with a visual interface that sends intuitively to the main structures of a language. The Arduino platform, on the other hand, is a great repository of resources for practical experimentation, and a low cost laboratory for testing of programs in C/C++. Scratch for Arduino (S4A) united those tools, forming a programming environment of very easy learning, with interface resources to sensors and actuators. The coordinated application of the tools Scratch, S4A and Arduino, in the context of an introductory programming course, can lead the student to a facilitated, consistent and focused to practical applications learning.*

**Key-words:** *Arduino, Programming, Scratch, Teaching*