

## Um estudo da fluência digital entre crianças e adultos na resolução de algoritmos

Mateus Madail Santin<sup>1</sup>, Silvia Silva da Costa Botelho<sup>2</sup>,

João Alberto Silva<sup>3</sup>

{mateus.santin<sup>1</sup>, silviacb<sup>2</sup>, joaosilva<sup>3</sup>} furg.br

### Resumo

Na literatura, define-se “Fluidez Digital” como a habilidade do indivíduo em projetar e criar mídias e programas. A introdução dos conceitos e metodologias associadas a tal habilidade constituem-se atualmente em um campo de pesquisa promissor na Educação em Computação. Em uma visão construtivista deste cenário, alguns autores pregam que a aprendizagem e o desenvolvimento desta habilidade passam por estratégias didático-pedagógicas que privilegiam a flexibilidade, a aprendizagem significativa e a sociabilidade. Dessa forma, esta pesquisa propõe um método para investigar a fluidez digital de adultos e crianças em atividades referentes ao desenvolvimento de algoritmos. Na resolução destes, é necessário o emprego de diferentes rotinas e fundamentos pertinentes ao desenvolvimento de programas computacionais. Durante o desenvolvimento, cada aluno tem diferentes abordagens para chegar a uma solução. Uma forma de medir estas características é proposta com base no grau de flexibilidade, significância e sociabilidade da abordagem desenvolvida contribuindo para o desenvolvimento e aprimoramento de novas propostas metodológicas para o Ensino-aprendizado da Computação.

**Palavras-chave:** linguagens de programação. lógica. Fluência Digital. Ensino de computação

### A study of digital fluency among children and adults in solving algorithms

#### Abstract

In the literature, "Digital Fluency" is define as an individual's ability to design and create media and programs. The introduction of the concepts and methodologies associated with such skill constitute currently in a promising area of research in Computer Educational. In a constructivist view of this scenario, some authors advocate the learning and development of this skill go through didactic and pedagogical strategies that emphasize flexibility, sociability and meaningful learning. Thus, this paper proposes a method to investigate the digital fluency of adults and children in activities related to the development of algorithms. In addressing these, the use of different routines and relevant to the development of computer programs fundamentals is necessary. During development, each student has different characteristics to reach a solution. One way of measuring these characteristics is proposed based on the degree of flexibility, sociability and significance of the approach developed contributing to the development and improvement of new methodological proposals for Teaching and Learning Computer.

**Keywords:** programming languages. logic. Digital Fluency. Computing education



## Introdução

A habilidade de programar e construir softwares se estabelece em função do pensamento lógico do sujeito que propõe tal tarefa. Nos últimos anos essa habilidade começou a ser desenvolvida através da exploração intuitiva e solitária, na qual o sujeito tateia formas de programar a partir de informações ou a partir do aprendizado formal em um curso específico da área. A primeira forma traz o inconveniente de depender de uma iniciativa pessoal, sem fomento de ordem da educação formal. Por outro lado, a segunda acontece apenas quando o candidato a programar se depara tardiamente em seu ciclo de formação com a necessidade de tratar problemas computacionais estruturados e complexos.

Há a hipótese de que atividades que envolvam o pensamento lógico, a estruturação de situações e a organização de ações favoreça o desenvolvimento da habilidade de programar e que isso possa ser explorado desde muito cedo junto ao aprendiz. Nesse sentido, o aprendizado da computação pode fazer parte do currículo escolar das crianças permitindo um contato mais direto com formas mais organizadas da lógica, da realização de experimentos e de contato com as tecnologias.

Assim como outras Ciências (Matemática, Física, e Química, por exemplo), busca-se tratar as questões pertinentes ao ensino precoce de fundamentos da Computação envolvendo seus conceitos e fundamentos associados as técnicas de programação e/ou de introdução de linguagens de baixo ou alto nível com as crianças.

Torna-se necessário o desenvolvimento de estratégias condizentes com o pensamento infantil e a necessidade das crianças que permitam, ao longo das diferentes etapas cognitivas do desenvolvimento infantil, a introdução das habilidades e fundamentos da Ciência da Computação. Entretanto, que habilidades são essas? O que precisa uma pessoa para aprender programação? Quais os procedimentos que podem ser usados na estruturação até o aprendiz chegar a um código de autoria.

Na opinião de Papert, as crianças devem ser capazes de projetar, criar e expressar-se com as novas tecnologias. Ao invés de apenas interagir com animações, jogos e simulações, as crianças devem aprender a programar suas próprias animações, jogos e simulações e, no processo, aprender importantes habilidades para resolução de problemas de design, e de projetos por elas mesmas (RESNICK, 2012). Estas habilidades estariam associadas ao conceito de *Fluidez Digital*. Este conceito representa a capacidade do indivíduo em criar e projetar mídias e algoritmos. Dentro da Educação em Computação a *Fluidez Digital* deveria ser explorada desde os primeiros anos do Ensino Fundamental, podendo ser tratada através de abordagens Didático-pedagógicas que contemplassem a *flexibilidade* e o imprevisto, a *aprendizagem significativa* e a *sociabilidade* do aprendiz (RESNICK, 2009).

A fim de contribuir para a área de Educação em Computação, nos propusemos a investigar como estudantes de cursos de nível superior em Computação abordam os problemas computacionais a serem solucionados, como organizam e constroem suas soluções e programas, comparativamente, utilizamos um grupo de crianças em um ambiente que favorece o desenvolvimento da criatividade e permite a conexão de peças e o arranjo destas para montar um robô, o qual pode ser programado manualmente através de um mecanismo de memória cinética, permitindo assim alterar o movimento deste robô utilizando as mãos.

Nesta pesquisa realizam-se experimentos onde o adulto necessita construir e projetar soluções em linguagem computacional. Nas oficinas, analisamos o comportamento dos aprendizes tendo em vista os aspectos de **flexibilidade**, **significância** e **sociabilidade** em suas abordagens para a solução dos desafios propostos, e que discutimos a importância de futuras estratégias didático-pedagógicas para o desenvolvimento e catalização da *fluidez digital* para os adultos.

Da mesma forma observamos estas variáveis quando as crianças estão desenvolvendo suas criações, como programadores, porém não em um ambiente descritivo como é natural do adulto, mas sim em um ambiente natural à elas, um ambiente lúdico o qual permite criar



através da construção e desconstrução das peças e por fim dando movimento à estas peças através da programação manual.

A pesquisa está estruturado em quatro seções. A seção seguinte aborda os aspectos associados ao ensino e aprendizado da computação, em seguida apresenta a metodologia aplicada com os adultos e com às crianças e por fim, a apresentação dos resultados e discussão dos resultados e as conclusões.

## 2. O ensino e aprendizado da Computação

O Ensino e aprendizagem da Computação têm sido foco de muitas pesquisas Pereira e Rapkiewicz (2004), Tais estudos são motivados, sobretudo, pela importância dos conceitos de programação na vida acadêmica dos cursos de computação. Pode-se destacar uma ênfase significativa na questão das dificuldades discentes no entendimento de conceitos relacionados a utilização de algoritmos para a solução de problemas (ROCHA, 2010). Justifica-se o foco do ensino da computação no conceito de Algoritmo, por ser este a estrutura formal para solução genérica de problemas.

Historicamente, o ensino de programação tem sido considerado de difícil entendimento para os alunos por diversos motivos: falta de preparo dos estudantes, ausência de uma didática adequada e de ferramentas computacionais que ajudem os atores (professores e estudantes) a superarem os problemas que se apresentam no processo ensino-aprendizagem (SOUZA, 2009).

Tem-se a compreensão que na medida em que o aluno compreende e desenvolve atividades relacionadas com algoritmos de modo satisfatório, apropriar-se de linguagens de programação torna-se muito mais fácil na medida em que o profissional precisa apenas apropriar-se das nuances específicas do software de autoria.

Compreender algoritmos de programação, constitui-se como um dos pilares para o desenvolvimento dos conhecimentos na área da Computação, principalmente nos aspectos lógicos trabalhados nesta Ciência. As disciplinas de algoritmos apresentam, geralmente, altos índices de reprovação, sendo parcialmente responsáveis pela evasão dos alunos do curso. Isso, na maioria dos casos, se deve a necessidade de estabelecer novas formas de analisar problemas e novas, e mais elevados, níveis de abstração para estudo e solução de problemas. Essa dificuldade de compreensão surge, em parte, pelo tardio contato do sujeito com os conceitos básicos associados ao projeto e construção de algoritmos, suas formas lógicas de estruturação e de organização, ocorrendo este somente na graduação. Ainda que muitos dos conceitos abstratos e lógicos, e suas representações, exijam níveis cognitivos mais elevados, é fundamental que identifique-se habilidades, estruturas e suas abordagens que possam ser gradualmente apresentadas e desenvolvidas ao longo da formação do aprendiz.

Recentemente Resnick (2009) aborda o ensino da Computação como um processo contínuo que deve ser trabalhado paralelamente a iniciação à outras Ciências. O trabalho propõe o uso da linguagem de programação Stratch, como ferramental Didático pedagógico no Ensino da Computação.

Os autores partem da premissa que o Ensino de Computação deve conduzir a *fluidez digital*, a qual estabelece-se como a habilidade para projetar e criar algoritmos representados através de diferentes mídias digitais. A proposta parte da Teoria Construcionista Papert (1986),

Seymour Papert (1986) adaptou os princípios do Construtivismo cognitivo de Piaget e da teoria interacionista de Vygotsky, propondo o Construcionismo o qual agrega um conjunto de premissas a serem utilizadas nas práticas de ensino, utilizando recursos tecnológicos como ferramenta de ensino/aprendizagem. Para Papert a apropriação do conhecimento acontece quando o aluno procura construir algo de seu interesse, tornando a aprendizagem significativa. O construcionismo é uma síntese da teoria de Piaget e das disponibilidades

oportunizadas pelo uso do computador para o desenvolvimento de uma aprendizagem contextualizada. Papert (1993), em seus estudos, percebeu a possibilidade de criar condições para mudanças significativas no desenvolvimento intelectual dos sujeitos com o uso dos computadores. De acordo com Papert (1993, p. 146), “a aprendizagem é facilitada e melhorada se o aluno construir algo de concreto, como por exemplo, uma maquete, um modelo, um programa de computador, algo que possa ser visto e analisado”.

Com base no Construcionismo, Resnick (2012) prega que as habilidades necessárias para o projeto e construção de mídias e programas passam pela presença de abordagens de aprendizado embasadas na *flexibilidade*, na capacidade de *aprendizado significativo* e na *sociabilidade*. Neste contexto, ao exibir estes TRÊS componentes no seu processo de aprendizado computacional, o indivíduo estaria apropriando de forma otimizada os conceitos e técnicas envolvidas. Busca-se neste trabalho, verificar se estas componentes estão presentes no processo de solução de problemas computacionais por jovens adultos de Cursos de Computação. Será realizado um conjunto de oficinas onde busca-se compreender como o adulto coordena as ações para resolver problemas relacionados ao desenvolvimento de algoritmos, de forma a classificar os diferentes perfis existentes no que tange os aspectos de flexibilidade, significado e sociabilidade da abordagem. Passamos a descrever a metodologia utilizada para investigar este processo.

### 3. Metodologia

O estudo foi realizado em duas etapas e a metodologia é apresentada em seções. A primeira seção discute os métodos desenvolvidos com os adultos. Já a segunda seção, discutimos os métodos adotados com as crianças.

#### 3.1 O estudo com os adultos

Nosso estudo realizou-se com alunos do último ano de um curso tecnólogo em processamento de dados o qual obteve conceito 4 na última avaliação do ENAD<sup>1</sup>, importante ressaltar que todos participantes possuem experiência em programação e muitos já trabalham na área de informática como desenvolvedores. Embora o ambiente seja favorável, tem-se a hipótese que mesmo sendo programadores com experiência como é o caso dos participantes adultos, a fluência digital não está diretamente relacionada à prática de programação.

Para este conjunto de experimentos 11 alunos do Curso Tecnólogo em Análise de Dados e Sistemas (TADS) foram voluntários. Após a realização dos experimentos foi analisado o resultado de 7 participantes. Os alunos do TADS têm contato com programação desde o primeiro semestre do curso, porém o ambiente de desenvolvimento ao qual eles são submetidos é diretamente relacionado com a linguagem Java. Os alunos não utilizam pseudocódigo para a iniciação do aprendizado. O ambiente de desenvolvimento no qual foram executados os algoritmos é desconhecido aos estudantes, bem como a linguagem Pascal<sup>2</sup> a qual o pseudocódigo é baseado.

O experimento compreende em apresentar uma seleção de problemas computacionais no âmbito da Maratona de Programação.

Estas maratonas, são eventos promovidos por diversas instituições no país, onde são postulados desafios computacionais que precisam ser resolvidos individualmente ou em grupo, levando em consideração o tempo para chegar a solução correta. Optou-se por utilizar o contexto das

---

<sup>1</sup> Exame Nacional de Desempenho de Estudantes

<sup>2</sup> <http://www.freepascal.org>

Maratonas como Objeto de Análise, uma vez que a competição promove nos alunos a criatividade, a capacidade de trabalho em equipe, a busca de novas soluções de software e a habilidade de resolver problemas sob pressão (SBC, 2012).

Para este estudo exemplificamos apresentando o primeiro desafio de um total de 3, a fim de comparar com o estudo realizado com as crianças. Uma vez que a intenção aqui é verificar a fluidez digital. Como os alunos fazem associações para a resolução de um problema?

O experimento é baseado nos procedimentos associados a Maratona. É oferecido um problema, juntamente com uma seleção de valores para o aluno testar o seu resultado. Uma entrada secreta não fornecida é utilizada pelo professor para validar se o exercício está correto. Os desafios computacionais fornecidos não exigem do aluno um grande volume de código, e podem ser resolvidos sem a necessidade de utilização de estruturas complexas de linguagem.

O desenvolvimento das aplicações utilizou um editor de pseudocódigo denominado *VisuAlg*, que é um software aplicativo que permite aos estudantes que se iniciam nas disciplinas de programação ferramentas para digitar, executar e depurar o pseudocódigo e resolver problemas propostos nas aulas e em exercícios, fornecendo também aos professores vários recursos didáticos para que expliquem como os programas funcionam, tais como execução passo a passo, visualização do conteúdo das variáveis, exame da pilha de ativação no caso de subprogramas, contador de execuções de cada linha do programa, etc. (SOUZA,2009).

O problema proposto: O algoritmo trata de um cálculo de horas e trás uma observação que objetiva facilitar e evitar interpretações que dificultem a execução, ver Tabela 1. O aluno neste exemplo mobiliza conceitos lógicos voltados para conversão de unidades, no caso, o tempo. A questão computacional é restrita a leitura de dados, cálculo de conversão das unidades e saída de dados para mostrar o resultado.

FAÇA UM ALGORITMO QUE LEIA UM HORÁRIO DO DIA (HORA:MINUTO:SEGUNDO). E MOSTRE NA TELA QUANTOS SEGUNDOS FALTAM PARA A MEIA-NOITE.

OBS: CONSIDERAR HORA DE 0 A 24 E QUE SEMPRE SERÃO INFORMADOS HORÁRIOS VÁLIDOS.

EX:

ENTRADA: 10 HORA: 40 MINUTOS: 0 SEGUNDOS SAÍDA: 48000

ENTRADA: 23 HORA: 50 MINUTOS: 10 SEGUNDOS SAÍDA: 590

ENTRADA: 23 HORA: 59 MINUTOS: 22 SEGUNDOS SAÍDA: 38

ESCONDIDA: ENTRADA: 0 HORA: 40 MINUTOS: 40 SEGUNDOS SAÍDA: 83960

**Tabela 1** – Descrição do Problema

**Instrumento de Coleta de Dados.** Neste trabalho, por se tratar de testes empíricos combinam-se diversos métodos de coleta de dados:

**Registro de vídeo:** uma câmera de alta definição é posicionada no fundo da sala e tem como objetivo filmar a interação entre os alunos e a atuação do professor.

Registro de Interação com Computador: A segunda forma de registro estabelece-se através do software livre recordmydesktop<sup>3</sup>. A utilização deste software permite que todas as ações realizadas pelo aluno sejam gravadas no computador e fiquem acessíveis através de um arquivo de vídeo. Desta forma consegue-se visualizar as tentativas e erros de cada aluno no desenvolvimento do algoritmo, permitindo analisar a forma com que cada aluno se organiza para resolver os problemas propostos. Cabe considerar que a estrutura computacional neste tipo de tarefa é bastante crítica, pois exige um computador com considerável processamento em função do vídeo ser processado simultaneamente com o desenvolvimento das atividades do aluno.

**Registro do Código:** o código final desenvolvido pelo aluno também serviu como elemento de análise.

### 3.2 O estudo com as crianças

O estudo com crianças utiliza o kit TOPOBO, que foi desenvolvido no MIT<sup>4</sup>, o TOPOBO é um kit educacional composto de 10 dez blocos primitivos básicos os quais se conectam em diversas orientações. Nove destas primitivas são chamadas “passivas”, pois formam ligações estáticas, sendo que 4 das formas geométricas são disponibilizados em dois tamanhos, as peças passivas e ativa são apresentadas na figura 1. A décima primitiva denominada “ativa”, é um circuito eletrônico envolvido por uma capa plástica cuja constituição é de um motor associado a componentes eletrônicos constituindo uma tecnologia embarcada, uma ideia de programação da peça ativa é mostrada na figura 2. Quando estas peças são unidas, cria uma rede peer-to-peer descentralizada (PARKS, 2008).

Para o estudo apresentado nesta pesquisa, adotamos um caso prático onde utilizamos um kit TOPOBO de 100 peças. No estudo, foram analisadas as práticas de 32 alunos de cinco anos de idade. O grupo foi composto por 15 meninos e 12 meninas de 3 escolas públicas da cidade de Salamanca na Espanha.

Para permitir que as crianças interajam em um ambiente onde a ideia é que elas desenvolvam a habilidade de programar, faz-se necessário que o ambiente esteja adequado para desenvolver está habilidade.

Para este estudo, iniciamos colocando as peças do TOPOBO sobre à mesa, incluindo duas partes ativas e um cabo que permitiu a interligação entre as peças ativas. Em seguida foi explicado aos alunos como é feito o procedimento de programação da peça ativa.

<sup>3</sup> <http://recordmydesktop.sourceforge.net/about.php>

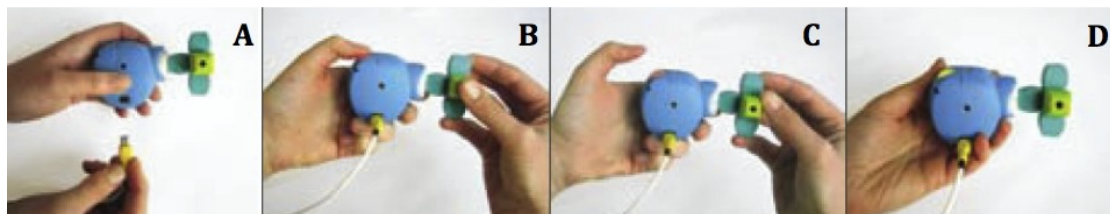
<sup>4</sup> <http://tangible.media.mit.edu/project/topobo/>





**Figura 1:** Conjunto de primitivas ativas e passivas que formam o TOPOBO

Na figura 2 é mostrada a forma de programação de uma peça ativa a qual é a responsável ativa do movimento das criações robóticas.



**Figura 2:** Programação tátil do TOPOBO (A) conecta a energia (B) Pressiona para programar (C) Move a estrutura (D) Pressiona para reproduzir

#### 4. Resultados e discussão dos resultados

Através dos instrumentos de coleta foi analisado o processo de desenvolvimento da solução por cada aluno do experimento. Ao longo do processo de escrita do código serão verificadas as abordagens desenvolvidas pelo aprendiz para a elaboração da solução.

As abordagens serão analisadas com base no grau de presença das seguintes características na solução:

- **Flexibilidade:** o quão flexível é a solução proposta, bem como o quão improvisado e espontâneo é o processo de desenvolvimento da solução pelo aprendiz;
- **Aprendizado Significativo:** o quão significativa são as Ações do aprendiz. Se este percebe claramente o problema a ser tratado, identificando o seu significado que o distingue de outras aplicações;
- **Sociabilidade:** estabelece o grau de colaboração do indivíduo com os demais da sala, ou mesmo, opostamente, o grau de individualismo na obtenção da solução.

Em cada experimento os instrumentos coletados servirão como subsídios para a quantificação de graus de *flexibilidade*, *significância* e *sociabilidade* associados a abordagem desenvolvida por cada aluno nas suas soluções propostas. Em função destes graus surgirão diferentes perfis de aprendizes os quais resultarão em classes de comportamentos existentes na resolução de problemas.

Com base nos vídeos, foram analisados os três indicativos de Fluidez Digital (Flexibilidade, Significância e Sociabilidade), a fim de elucidar o que estamos considerando para cada um, a tabela 1 nos trás os resultados do experimento realizado.

Adultos programando	Flexibilidade	Significância	Sociabilidade
1 – Supõe existir uma variável <i>data</i> como na linguagem Java; Insiste nesta premissa. Cola comando de C++ no ambiente; Pesquisa procurando por <i>data</i> .	Baixo	Alto	Baixo
2 – Declara variável do tipo <i>data</i> ; Declara as variáveis como na linguagem Java Procura como declarar um tipo <i>data</i>	Baixo	Alto	Baixo
3 – Iniciou pelo Segundo Algoritmo; Declara as variáveis; Procura no manual da linguagem a forma correta de definir	Alto	Alto	Baixo
4 – Procura apostila da linguagem; Cola exemplos da apostila no ambiente; Redefine variáveis;	Baixo	Médio	Alto
5 – Inicia procurando documentação do ambiente; Copia e cola modelos; Procura por variável do tipo <i>data</i> como no Java.	Baixo	Médio	Alto
6 – Pesquisa pelo ambiente, procura solução pronta, pesquisa exatamente o que é solicitado, cola comandos de c++ no ambiente.	Baixo	Baixo	Alto
7 – Iniciou procurando cálculo de tempo, Define variáveis sem tipo, procura comando do ambiente, altera as variáveis, refaz os cálculos.	Alto	Alto	Baixo

**Tabela 1:** Tabulação das Variáveis de Análise nos Experimentos.

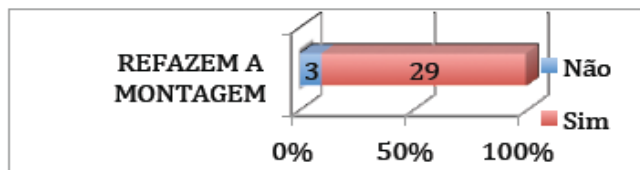
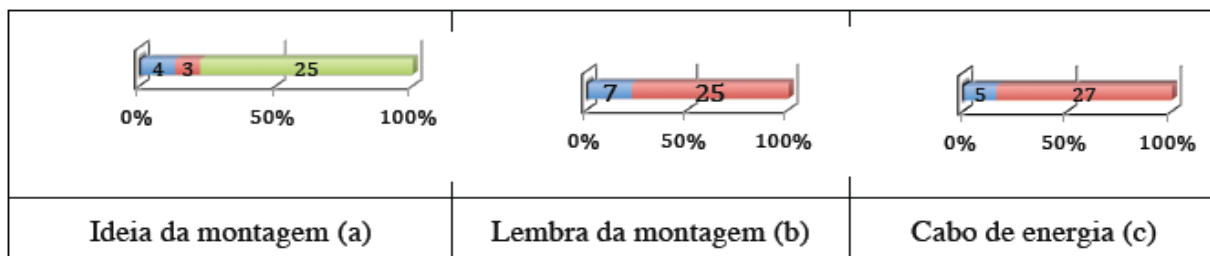
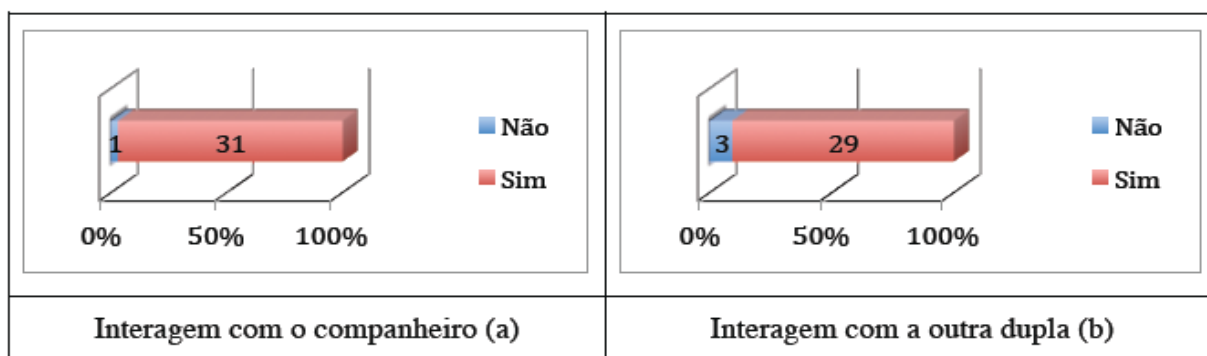
Apresentamos os dados de programação das crianças para os três pilares da Fluência digital, a **flexibilidade** é apresentada na figura 3, a **significância** na figura 4 foi analisada em três momentos distintos. Das três variáveis que adotamos para analisar a significância, a primeira foi a **ideia da montagem**, neste momento é perguntamos ao aluno se ele sabe o que quer montar com as peças, o resultado pode ser visto na figura 4 (a). No segundo momento analisamos a variável denominada, **lembra da montagem** figura 4 (b) que é uma variável onde é avaliado se após alguns minutos o aluno lembra da criação que se propôs a fazer.

E por fim analisamos a interconexão do cabo de energia figura 4 (c), onde verificamos se as crianças associam o processo de programação cinética à uma fonte externa para movimentar o motor da peça ativa.

No contexto da significância, estamos avaliando se o aluno mantém uma ideia do processo de desenvolvimento da criação/programa e as ações estão de acordo com o processo de montagem e não somente uma conexão sem sentido das peças.

E por fim analisamos a **sociabilidade** na figura 5. Neste momento analisamos na variável representada na figura 5 (a) a interação com o companheiro e na figura 5 (b) a interação com a outra dupla.



**Figura 3: Variáveis consideradas na flexibilidade****Figura 4: Variáveis consideradas na significância****Figura 5: Variáveis consideradas na sociabilidade**

Diferentes perfis de aprendizes surgem no momento que estes se propõem a resolver problemas computacionais. Com base nos pressupostos construcionistas, as questões referentes a flexibilidade, significância e sociabilidade são aspectos importantes no processo de aquisição de habilidades significativas para o projeto e construção de algoritmos e mídias digitais.

### Considerações Finais

Este trabalho insere-se na área de Educação em Informática, mais precisamente na definição de estratégias didático-pedagógicas associadas ao desenvolvimento de habilidade que potencializem a *fluidez digital* dos aprendizes. De forma mais precisa este estudo analisou a forma de programar de estudantes universitários e de crianças, traçando uma linha no desenvolvimento de algoritmos sobre a visão da **fluência digital**.

Podemos notar que a fluência digital é encontrada em crianças nos primeiros anos das séries iniciais e não necessariamente é observada em adultos com experiência em programação. O que nos leva a crer que desenvolver metodologias e ferramentas adequadas à diferentes níveis de aprendizes é um tema que precisa ser ampliado. Precisa-se promover mecanismos para o desenvolvimento da fluência digital para que as pessoas sejam integradas em uma cultura onde à informática é cada vez mais presente.

**BIBLIOGRAFIA**

- PERREIRA Junior, J.C.R., Rapkiewicz, C. O Processo de Ensino-Aprendizagem de Fundamentos de Programação: Uma Visão Crítica da Pesquisa no Brasil, WEI RJES, 2004
- MARCONI, M. de A.; LAKATOS, E. M. Metodologia científica. 5.ed. São Paulo: Atlas, 2007. 312p.
- PAPERT, S. Computadores e Conhecimento: repensando a educação. Campinas: Unicamp, 1993.
- \_\_\_\_\_. Constructionism: a new opportunity for elementary science education. Massachusetts Institute of Technology, The Epistemology and Learning Group. Massachusetts: National Science Foundation, 1986.
- PARKS, Amanda; RAFFLE Hayes; ISHII Hiroshi. Topobo in the wild: Longitudinal evaluations of educators appropriating a tangible interface. Paper presented at the Proceeding of the Twenty-Sixth Annual SIGCHI Conference on Human Factors in Computing Systems, pages 1129–1138, 2008.
- RESNICK, M., Maloney, J., Monroy-Hernandez, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). Scratch: Programming for All. *Communications of the ACM*, vol. 52, no. 11, pp. 60-67. 2009.
- RESNICK, M. Reviving Papert's Dream. *Educational Technology*, vol. 52, no. 4, pp. 42-46. 2012.
- ROCHA, P. S. ; B. Ferreira ; D. Monteiro . Ensino e Aprendizagem de Programação: Análise da Aplicação de Proposta Metodológica Baseada no Sistema Personalizado de Ensino. RENOTE. Revista Novas Tecnologias na Educação, v. 9, p. 1, 2010.
- SBC. MARATONA DE PROGRAMAÇÃO. – Definição sobre Maratona de Programação , Disponível em:  
[http://www.sbc.org.br/index.php?option=com\\_content&view=category&layout=blog&id=303&Itemid=180](http://www.sbc.org.br/index.php?option=com_content&view=category&layout=blog&id=303&Itemid=180), Acesso dia: 27/12/2012.
- SOUZA, Cláudio M. . VisuAlg - Ferramenta de Apoio ao Ensino de Programação. Revista TECCEN – v. 2, n 2 - setembro de 2009 – ISSN 1984-099