



# Introdução ao Pensamento Computacional

**Maristela Alcântara**

Junho 2015

## Aprender a programar, programar para aprender

Muito se tem discutido sobre a influência das tecnologias no desenvolvimento das habilidades e competências imprescindíveis para interagir com criatividade e pró-atividade neste mundo globalizado. O que é interagir com criatividade e pró-atividade? Quando falamos em interação podemos falar em interação entre pessoas e interação entre pessoas e máquinas (robôs, computadores, eletrônicos em geral). Para que haja interação é necessário que haja diálogo, estímulo, feedback, pensamento criativo, construção coletiva, enfim, são dois ou mais elementos em constante comunicação e tomando decisões diferenciadas de acordo com o grau de interação. E a criatividade? O que é ser criativo? Apenas pessoas com ideias inovadoras são criativas? E o que é inovação?

Inovar é buscar sempre novas soluções e caminhos diferenciados para a resolução de problemas. Não preciso “inventar a roda” para ser inovador. Preciso inventar novas funções para as rodas, ou incluir a roda em outras funções rotineiras de forma a aperfeiçoar minhas tarefas. E a criatividade? Ser criativo é deixar as ideias fluírem, é não parar de pensar em novas soluções para os problemas, mesmo que o problema já tenha sido solucionado de alguma maneira. Será que não podemos buscar outras formas mais eficazes para resolvermos este problema?

Atualmente, ouvimos muito o termo “Nativos Digitais” (termo criado pelo norte-americano Marc Prensky). Um nativo digital é aquele que nasceu e cresceu com as tecnologias digitais presentes em sua vida. E por ter convivido com as tecnologias digitais desde cedo se supõe que as habilidades com estes recursos tenham sido desenvolvidas de forma natural. Será mesmo?

A familiaridade com os recursos tecnológicos digitais ajudam, e muito, na fluência tecnológica, mas não garante o desenvolvimento de habilidades que transformem estes recursos em instrumentos de produção e criação. Ser fluente tecnologicamente é usufruir de tais recursos para ampliar suas possibilidades de produção, criação e comunicação. Quem produz algo novo sente o prazer da invenção, sente a vontade de criar sempre mais, aprendendo cada vez mais e compartilhando suas descobertas com as pessoas. Uma pessoa criativa e fluente em tecnologia aproveita todos os recursos disponíveis em suas mãos para buscar novas soluções para os problemas que surgem ao longo do seu caminho.

Uma das formas de desenvolver a fluência tecnológica é desenvolver nas crianças, jovens e adultos o pensamento computacional. E o que é isso?

Segundo *Paulo Blikstein* (Professor na Escola de Educação e no departamento de Ciência da Computação da *Stanford University*, EUA), pensamento computacional é saber usar o computador como um instrumento de aumento do poder cognitivo e operacional humano – em outras palavras, usar computadores, e redes de computadores, para aumentar nossa produtividade, inventividade, e criatividade.

Quando nos deparamos com um problema a ser resolvido precisamos “pensar computacionalmente”, ou seja, analisar o problema, dividi-lo em partes menores (**decomposição**), analisar se já não existem soluções para alguns destes pequenos problemas, encontrar os **padrões** nas diferentes soluções de cada um dos problemas menores, aprender a ignorar os detalhes (**abstração**) que são responsáveis pelas diferenças e encontrar uma solução que seja válida para mais de um problema. E, finalmente, procurar definir todos os passos que serão necessários para solucionar o problema (**algoritmo**).

Todas estas etapas necessárias neste método de resolução de problemas utilizado pelos cientistas da computação exigem habilidades específicas que precisamos desenvolver nos estudantes do século XXI. Independente se eles serão ou não, futuros cientistas da computação, as habilidades desenvolvidas serão fundamentais para se transformarem em pessoas mais criativas, produtivas e preparadas para enfrentarem os desafios que surgirão ao longo da vida.

Um dos caminhos que propomos para o desenvolvimento da habilidade Pensamento Computacional é aprender a programar. Entretanto, mais do que aprender a programar, queremos que crianças, jovens e adultos programem para aprenderem coisas novas. O objetivo não é simplesmente conhecer a sintaxe de uma linguagem de programação (Java, C, Delphi, Python....), mas conhecer a essência do programa, a forma como ele deverá ser construído, ou seja, as estruturas de programação que serão utilizadas para a resolução de um problema de forma eficiente e eficaz.

Desenvolver o raciocínio lógico, entender as funções das diferentes estruturas de programação são essenciais para a construção dos passos necessários para a resolução dos problemas que futuramente se tornarão programas escritos em qualquer linguagem de programação.

A essência do *aprender a programar, programar para aprender* está no fato de que ao desenvolver o raciocínio lógico e o pensamento computacional o estudante estará aprendendo a lidar com diferentes problemas, analisando-o, buscando soluções mais eficientes e, ao mesmo tempo, enriquecendo a aprendizagem daquele conteúdo que está sendo trabalhado no projeto desenvolvido. Por exemplo, ao criar um programa para que um robô percorra um trajeto em forma de trapézio retângulo, é essencial que o programador compreenda as características desse trapézio retângulo. Ou então, ao criar um programa que simule o sistema solar, é imprescindível que o programador compreenda esse sistema.

Hoje em dia existem vários recursos, sites, aplicativos e softwares que apoiam o ensino de programação.

A seguir alguns recursos para serem utilizados na educação básica, por crianças e jovens, com o objetivo de contribuir com o desenvolvimento do raciocínio lógico e conhecer toda a base estrutural de qualquer programa de computador.

Relação destes recursos disponíveis e gratuitos:

### **Site do Programaê – [www.programae.org.br](http://www.programae.org.br)**

Iniciativa que foi reforçada no Brasil ao final do ano passado, o Programaê tem foco em jovens e traz uma coletânea de materiais de parceiros – Codecademy, a KhanAcademy, Code.org, Scratch e a Fábrica de Aplicativos. As aulas, portanto, são mais introdutórias e ajudam a pegar os conceitos básicos de programação.



## **Code.org (parte do programaê) -**

**<http://studio.code.org/>**

A Code.org é uma referência mundial em ensino de programação e criou um currículo recheado de minigames (desafios), atividades “desplugadas” (sem auxílio do computador) e vídeos com celebridades que apresentam os conceitos fundamentais de programação.

## **Scratch (parte do programaê) -**

**<https://scratch.mit.edu/>**

O *Scratch* é uma nova linguagem de programação que permite criar histórias interativas, animações, jogos, música e arte. Indicado para quem está se aventurando em programação pela primeira vez, programar no *Scratch* é como brincar com blocos de montar. Você irá aprender os fundamentos da programação sem nem perceber que está escrevendo códigos!

## **Lightbot - <http://lightbot.com/>**

Um excelente recurso para resolver quebra-cabeças através da lógica de programação. A ideia é colocar comandos em um pequeno robô para fazê-lo andar, pular, virar, acender uma lâmpada e assim solucionar os diferentes desafios que são propostos.

## **Blockly Games - <https://blockly-games.appspot.com/>**

Blockly Games é uma série de jogos educativos que ensinam programação. Ele é projetado para crianças que não tiveram experiência prévia com programação de computadores.

## **HardCoder – aplicativo disponível no Google Play**

Um aplicativo criado para contribuir com o aprendizado das principais estruturas de programação: sequência, loops, funções, etc. São em torno de 180 níveis de dificuldades. O objetivo é você escrever códigos para mover-se em uma malha quadriculada. São vários caminhos que temos para explorar esta nova maneira de se comunicar na era digital. A proposta é que explorem ao máximo os recursos tecnológicos de forma a ampliarem as possibilidades de aprendizagem.

# **Programar é CRIAR!**

**Para saber mais...**

**O pensamento computacional e a reinvenção do computador na educação -**

*[http://www.blikstein.com/paulo/documents/online/ol\\_pensamento\\_computacional](http://www.blikstein.com/paulo/documents/online/ol_pensamento_computacional)*

**Aprender a programar, programar para aprender -**

*<http://porvir.org/porfazer/aprender-programar-programar-para-aprender/20130220>*

**Sites para aprender e programar, de graça -**

*<http://porvir.org/porfazer/para-aprender-programar-de-graca/20130902>*

**Programaê -**

*[http://fundacaolemann.org.br/novidades/programae?gclid=CjwKEAjwjd2pBRDB4o\\_ymcieoAQSJABm4egoXwdiez07tDilk\\_XxnXyU.9jJABoCpYTw\\_wcB](http://fundacaolemann.org.br/novidades/programae?gclid=CjwKEAjwjd2pBRDB4o_ymcieoAQSJABm4egoXwdiez07tDilk_XxnXyU.9jJABoCpYTw_wcB)*

**9 ferramentas gratuitas para você aprender a programar -**

*<http://super.abril.com.br/blogs/superlistas/9-ferramentas-gratuitas-para-voce-aprender-a-programar/>*