

UNIVERSIDADE FEDERAL DO PARANÁ

JOÃO BOSCO CAVALCANTE ALBUQUERQUE

ENSINO DO PENSAMENTO COMPUTACIONAL PARA CRIANÇAS: ANÁLISE DE  
FERRAMENTAS PARA CONSTRUÇÃO DE UMA PROPOSTA PEDAGÓGICA

JANDAIA DO SUL

2017

JOÃO BOSCO CAVALCANTE ALBUQUERQUE

**ENSINO DO PENSAMENTO COMPUTACIONAL PARA CRIANÇAS: ANÁLISE DE  
FERRAMENTAS PARA CONSTRUÇÃO DE UMA PROPOSTA PEDAGÓGICA**

Trabalho de Conclusão de Curso apresentado como requisito parcial para a obtenção do grau no curso de Licenciatura em Computação, Campus Avançado em Jandaia do Sul, da Universidade Federal do Paraná .

Orientador: Prof. Me. Carlos Eurico Galvão Rosa

JANDAIA DO SUL

2017

Ficha catalográfica elaborada pela Seção de Referência e Informação – UFPR –  
Biblioteca do Campus Jandaia do Sul (PR)

A345e Albuquerque, João Bosco Cavalcante  
Ensino do pensamento computacional para crianças: análise de  
ferramentas para construção de uma proposta pedagógica / João  
Bosco Cavalcante Albuquerque.  
Jandaia do Sul: 2017.  
92 p.;il.

Orientador Carlos Eurico Galvão Rosa.  
Trabalho de Conclusão de Curso (Graduação em Licenciatura em  
Ciência da Computação) Universidade Federal do Paraná.

1. Computação - Ensino. 2. Informática – Ensino. 3. Pensamento  
Computacional. I. Rosa, Carlos Eurico Galvão, orient. II. Universidade  
Federal do Paraná. Campus Jandaia do Sul. Licenciatura em Ciência  
da Computação. III. Título.

CDD: 370.7



UNIVERSIDADE FEDERAL DO PARANÁ

**PARECER Nº** 01/2017/UFPR/R/JA  
**PROCESSO Nº** 23075.217365/2017-50  
**INTERESSADO:** UFPR/R/JA/CCLC - COORDENAÇÃO DO CURSO DE LICENCIATURA EM COMPUTAÇÃO - JANDAIA

### TERMO DE APROVAÇÃO DE TRABALHO DE CONCLUSÃO DE CURSO

Título: Ensino do Pensamento Computacional para crianças: Análise de ferramentas para construção de uma proposta pedagógica

Autor: João Bosco Cavalcante Albuquerque

Trabalho de Conclusão de Curso apresentado como requisito parcial para a obtenção do grau no curso de Licenciatura em Ciência da Computação, aprovado pela seguinte banca examinadora.

- Prof. Me. Carlos Eurico Galvão Rosa
- Prof. Me. Robertino Mendes Santiago Junior
- Ma. Márcia Inês Schabarum Mikuska

Jandaia do Sul, 23/11/2017



Documento assinado eletronicamente por **CARLOS EURICO GALVAO ROSA, PROFESSOR DO MAGISTERIO SUPERIOR**, em 01/12/2017, às 15:53, conforme art. 1º, III, "b", da Lei 11.419/2006.



Documento assinado eletronicamente por **MARCIA INES SCHABARUM MIKUSKA, ASSISTENTE EM ADMINISTRACAO**, em 01/12/2017, às 20:14, conforme art. 1º, III, "b", da Lei 11.419/2006.



Documento assinado eletronicamente por **ROBERTINO MENDES SANTIAGO JUNIOR, PROFESSOR DO MAGISTERIO SUPERIOR**, em 05/12/2017, às 14:46, conforme art. 1º, III, "b", da Lei 11.419/2006.



A autenticidade do documento pode ser conferida [aqui](#) informando o código verificador **0592037** e o código CRC **AC259F1A**.

## **AGRADECIMENTOS**

Agradeço aos meus filhos que sempre me apoiaram durante esses quatro anos. A minha irmã, Patrícia, que mesmo longe sempre esteve comigo em todos os momentos. Aos meus queridos professores que de alguma forma me ajudaram a superar minhas limitações. A todos na minha turma, pelas vezes que nos ajudamos para vencer desafios. Ao meu orientador, que durante esse ano teve paciência e me ajudou a concluir esse trabalho. E a todos que de alguma forma, seja por incentivos ou críticas, são responsáveis por eu ter chegado até aqui.

*Sempre que te perguntarem se podes fazer um trabalho,  
respondas que sim e te ponhas em seguida a aprender como se faz.*

*F. Roosevelt*

## RESUMO

O Pensamento Computacional (PC) tem sido um tema debatido por muitos educadores ao redor do mundo. Sabe-se que o PC é a base para diversos fundamentos da Ciência da Computação, com aplicação na resolução de problemas em diversas áreas do conhecimento. A grande questão é saber quando PC deve começar a ser desenvolvido e qual deve ser o papel da escola. Outro ponto interessante deve ser o como fazer isso e quem estará habilitado a ministrar as disciplinas que promovem esse desenvolvimento. Este trabalho faz uma pesquisa da importância do desenvolvimento, as iniciativas em alguns países, as ferramentas que podem ser usadas no desenvolvimento do PC, apresentando sugestões de trabalho de acordo com a infraestrutura disponível.

**Palavras-chaves:** *Pensamento Computacional; Computação desplugada; ferramentas de ensino-aprendizagem na computação.*

## **ABSTRACT**

Computational Thinking (CT) has been a theme debated by many educators around the world. It is known that the CT is the basis for several foundations of Computer Science, with application in solving problems in several areas of knowledge. The big question is when CT should start to be developed and the role of the school. Another interesting point should be how to do this and who will be able to minister the disciplines that promote this development. This work does research of the importance of development, the initiatives in some countries, the tools that can be used in the development of the CT, presenting suggestions of work according to the available infrastructure.

**Key-words:** Computational Thinking; Unplugged Computing; Tools teaching-learning in Computing.

## LISTA DE ILUSTRAÇÕES

FIGURA 1 – DESENVOLVIMENTO DO PENSAMENTO COMPUTACIONAL . .	14
FIGURA 2 – My Interactive Gardem . . . . .	22
FIGURA 3 – CODE.ORG - A HORA DO CÓDIGO . . . . .	26
FIGURA 4 – IDE DO <i>SCRATCH</i> . . . . .	27
FIGURA 5 – <i>LIGHTBOT</i> . . . . .	28
FIGURA 6 – PROGRAMAÊ REÚNE PLATAFORMAS. . . . .	29
FIGURA 7 – NÚMEROS BINÁRIOS . . . . .	31

## LISTA DE TABELAS

TABELA 1 – ORGANIZAÇÃO DAS AULAS DO PROGRAMAÊ . . . . .	30
---	----

## LISTA DE ABREVIATURAS E SIGLAS

CT	<i>Computational Thinking</i>
IDE	<i>Integrated Development Environment</i>
MIT	<i>Massachusetts Institute of Technology</i>
PC	Pensamento Computacional

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>13</b>
1.1	OBJETIVOS	13
1.1.1	Objetivo Geral	13
1.1.2	Objetivos específicos	14
1.2	JUSTIFICATIVA	14
<b>2</b>	<b>METODOLOGIA</b>	<b>16</b>
<b>3</b>	<b>REVISÃO DE LITERATURA</b>	<b>17</b>
3.1	COMPUTAÇÃO E O DESENVOLVIMENTO COGNITIVO	17
3.2	DISCIPLINAS DE ALGORITMOS E SUAS DIFICULDADES	19
3.3	INICIATIVAS EM OUTROS PAÍSES	20
3.3.1	Estados Unidos	20
3.3.2	Alemanha	21
3.4	INICIATIVAS NO BRASIL	23
<b>4</b>	<b>ESCOLHA DA FERRAMENTA</b>	<b>25</b>
4.1	CODE.ORG	25
4.2	SCRATCH	26
4.3	LIGHTBOT	28
4.4	PROGRAMAÊ	28
4.5	COMPUTAÇÃO DESPLUGADA	29
<b>5</b>	<b>SUGESTÃO DE PROCEDIMENTO</b>	<b>32</b>
5.1	CENÁRIO OTIMISTA	32
5.2	CENÁRIO PESSIMISTA	32
5.3	CONSIDERAÇÕES	33
<b>6</b>	<b>CONSIDERAÇÕES FINAIS</b>	<b>35</b>
6.1	PROJETOS FUTUROS	35
	<b>REFERÊNCIAS</b>	<b>36</b>
	<b>ANEXOS</b>	<b>40</b>
<b>ANEXO A</b>	<b>A HORA DO CÓDIGO</b>	<b>41</b>

<b>ANEXO B</b>	<b>SCRATCH</b> . . . . .	<b>48</b>
<b>ANEXO C</b>	<b>LIGHTBOT</b> . . . . .	<b>73</b>
<b>ANEXO D</b>	<b>PROGRAMAE</b> . . . . .	<b>76</b>
<b>ANEXO E</b>	<b>CS UNPLUGGED</b> . . . . .	<b>84</b>

# 1 INTRODUÇÃO

O uso de computadores na educação não é nenhuma novidade, tendo iniciado já na década de 1950. Entretanto, seu uso era limitado ao armazenamento de uma série de informações posteriormente transmitidas ao aprendiz (VALENTE, 1999). Com o aumento do uso de tecnologia computacional, tudo está cada vez mais informatizado. As crianças são inseridas em um mundo tecnológico desde cedo, com fácil acesso a celulares e computadores. No entanto, na maioria dos casos, se tratam apenas de super usuários das tecnologias e desconhecem seu funcionamento. Com esse cenário, alguns países tem discutido a possibilidade de mudanças em currículos de ensino nas últimas décadas, a fim de que a informática deixe de ser apenas um meio e se torne o objeto de aprendizagem.

Em países como Alemanha, Estados Unidos e Israel, a computação tem sido inserida na rede regular de ensino (FROZZA; BRINGMANN; CRUZ, 2016). Mais recentemente o Reino Unido e a Austrália também tem adotado iniciativas semelhantes. Um ponto bastante interessante dessa discussão, é saber quando exatamente é preciso começar a ensinar programação e estimular o desenvolvimento do Pensamento Computacional (PC). Embora esse tema não seja algo novo, incluí-lo como matéria obrigatória em escolas é uma pauta importante que precisa ser considerada como estratégia de ensino.

As abordagens diversificadas, dos diferentes países, vão de encontro com o momento atual de importantes mudanças na educação. Neste, muitos apostam em uma quebra de paradigma na problemática de ensino e aprendizagem do PC (VALENTE, 1999). Com um número considerável de iniciativas mundiais que visam auxiliar a aplicação de um currículo base no ensino de programação para crianças, entender como essas ferramentas funcionam é vital para adotar métodos e estratégias que possam ser aplicadas em escolas brasileiras.

## 1.1 OBJETIVOS

### 1.1.1 Objetivo Geral

O objetivo desse trabalho é estudar diferentes modelos de ensino que visam o desenvolvimento do PC em crianças, destacando iniciativas de programas curriculares bem sucedidos, nos Estados Unidos e na Alemanha. Investigar algumas ferramentas existentes no Brasil, considerando seus objetivos e a relevância de seu uso para construir uma proposta pedagógica a ser implementada em escolas brasileiras.

### 1.1.2 Objetivos específicos

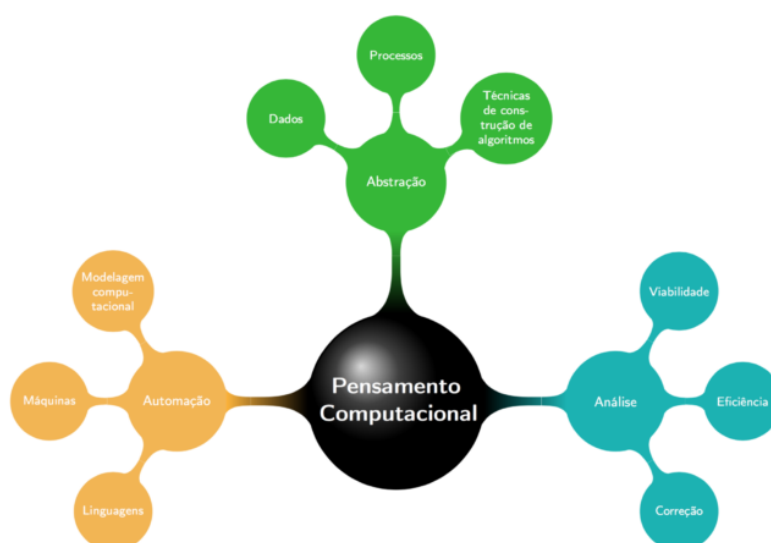
Pesquisar ferramentas que tenham o objetivo de ajudar professores no ensino do PC para crianças. Entender como essas ferramentas podem ajudar os professores, com materiais que podem ser utilizados em salas de aula tendo em vista a realidade das escolas públicas brasileiras. Para alcançar isso faz-se necessário pesquisar as plataformas para entender seus pontos fortes e fracos em conformidade com a realidade de nossas escolas.

## 1.2 JUSTIFICATIVA

Vivemos em mundo cercado de tecnologias: A era da informação mudou o mundo de forma acelerada. No entanto percebemos que nossos jovens chegam na vida adulta com dificuldades para interpretar problemas simples de Matemática e Análise de textos. Para Giraffa e Mora (2013, p. 6), “...os alunos possuem muita dificuldade em entender os enunciados e o que leem”. Isso leva a outras questões relacionadas a leitura prévia dos textos: Os alunos passam anos na escola, são selecionados para a entrada na universidade e ainda assim muitos não estão aptos para o nível de ensino que os espera (GIRAFFA; MORA, 2013).

O desenvolvimento do PC pode ser um importante aliado para trabalhar essas competências com os alunos. A programação é uma sequência de comandos para a resolução de um problema e estudar programação ajuda tanto o desenvolvimento cognitivo quanto o raciocínio lógico (CHAVES, 1983, p. 15). O PC ajuda a desenvolver uma série de habilidades que são base para diversos saberes, como ilustrado pela FIGURA 1.

FIGURA 1 – DESENVOLVIMENTO DO PENSAMENTO COMPUTACIONAL



FONTE: <<http://computacaonaescola.com.br/>>

Projetos como Codeorg<sup>1</sup>, Projectbloks<sup>2</sup>, Coderbyte<sup>3</sup> e Codeacademy<sup>4</sup> entre outros, disponibilizam conteúdo de programação de computadores com foco no público infantil. Felizmente estes projetos foram multiplicados e surgiram voluntários em várias partes do mundo, inclusive no Brasil, o que acelerou a tradução de muitas ferramentas para o português. É de interesse do licenciado em Computação entender como esses trabalhos são desenvolvidos e quais estratégias utilizadas para alcançar seus objetivos, sendo o desenvolvimento do PC uma importante área da atuação destes profissionais.

---

1 <<https://code.org>>

2 <<https://projectbloks.withgoogle.com/>>

3 <<https://coderbyte.com/>>

4 <<https://www.codecademy.com/>>

## 2 METODOLOGIA

No contexto desta pesquisa, define-se como “ferramenta” um conjunto de atividades, dentre vídeos, *softwares*, plataformas, métodos, etc, que podem ser usados com alunos, em turmas ou individualmente, e que auxiliam no desenvolvimento do PC. Aqui são apresentadas apenas algumas, tendo em vista que a computação é dinâmica e que a cada dia surgem aplicativos novos: Sabe-se que uma aplicação utilizada hoje pode ser facilmente substituída em pouco tempo por outra com mais recursos. Deste modo, não é foco deste trabalho uma relação pormenorizada de todas as possíveis e disponíveis ferramentas para o desenvolvimento do PC. Para escolha da abordagem, foram adotados os seguintes critérios:

- Disponíveis em português;
- Gratuitas ou com versões funcionais gratuitas;
- Fácil de aprender;
- Tem uma comunidade ativa para troca de conhecimentos.

Além dessas questões, foram considerados os cenários que podem ser encontrados nas escolas. Assim, julgou-se desejável que o trabalho contivesse soluções para as características particulares descritas:

- Usar sem instalação;
- Trabalhar elementos de som e imagens;
- Usar no *Smartphone*;
- Poder se combinar com outras ferramentas;
- Ter roteiros com planos de aulas;
- Permitir o desenvolvimento do PC sem o uso de computadores.

Cada ferramenta deste trabalho satisfaz pelo um dos itens dessa lista. O objetivo desta seleção é evidenciar a existência de alternativas para os mais diversos cenários que possam ser encontrados nas escolas, sem a pretensão de ser uma listagem exaustiva, mas uma sugestão de soluções.

### 3 REVISÃO DE LITERATURA

São apresentados a seguir algumas pesquisas nas quais se baseia o presente trabalho. É destacado por Santos (2017) o fato de que o ensino de programação está sendo ministrado para pessoas cada vez mais jovens. Iniciativas como Code.org, Kodu, *Game Lab* e ferramentas como *Scratch*, *Alice* e *ToonTalk* proporcionam que este tipo de conhecimento seja disseminado. As tecnologias são usadas em vários contextos, sendo que alguns conteúdos podem ser adaptados para o atividades curriculares extracurriculares. Para ele:

A programação de computadores é uma competência com uma crescente importância na formação de uma pessoa, em particular nos jovens. Mais do que habilitar para uma melhor utilização da enorme capacidade e funcionalidades computacionais hoje disponíveis, permite por si só desenvolver o raciocínio lógico agregado à capacidade de resolução de problemas (SANTOS, 2017, p. 1).

Isto está de acordo com a afirmação de Bastos e Cunha (2017) de que:

[...] os alunos, através da aprendizagem da programação, podem desenvolver a sua criatividade em ciências da computação; promover uma visão mais ampla e formas diferentes de utilização do computador e desenvolver o PC (BASTOS; CUNHA, 2017, p. 31).

#### 3.1 COMPUTAÇÃO E O DESENVOLVIMENTO COGNITIVO

O conceito de que a programação de computadores pode ajudar no desenvolvimento cognitivo das crianças também não é novo: Seymour Papert já defendia essa ideia em 1967, quando apresentou a linguagem de programação Logo. Valente (2016) afirma que:

A possibilidade de a computação auxiliar a criança a pensar melhor ficou clara no livro *Mindstorms* (PAPERT, 1980) no qual Papert propõe que a programação Logo pode estimular o que ele chamou de “*Powerful ideas*<sup>1</sup>” e “*Procedural knowledge*<sup>2</sup>”. Para esse pesquisador, os computadores deveriam ser utilizados para que as pessoas pudessem “pensar com” as máquinas e “pensar sobre” o próprio pensar (VALENTE, 2016, p. 869).

<sup>1</sup> Ideias Poderosas em português

<sup>2</sup> Conhecimento Procedural em português

Jeanette Wings define o termo *computational thinking* — PC em português — quando afirma que o PC “é o processo de pensamento envolvido na formulação de um problema, expressando sua solução, de uma maneira que um computador, humano ou máquina, pode efetivamente realizar” (WING, 2006, p. 7). Para ela além da leitura, da escrita e da matemática, as crianças precisam desenvolver o PC, pois este é uma habilidade fundamental para todas as pessoas (WING, 2014, p. 33).

Tem se consolidado o uso do computador na educação. No entanto, em muitos casos, o computador é restringido a uma “máquina de ensinar”. Valente (2016) aponta que o uso do computador pode ser tanto para transmitir conhecimento quanto para criar condições da construção do conhecimento pelo próprio aluno. No primeiro caso, o computador assume o papel de máquina de ensinar. No último, o aluno pode assumir o papel de ensinar a máquina:

[...] propiciando condições para o aluno descrever a resolução de problemas, usando linguagens de programação, refletir sobre os resultados obtidos e depurar suas ideias por intermédio da busca de novos conteúdos e novas estratégias (VALENTE, 1999, p. 2).

Sleeman (1986) afirma que a programação foi descrita por muitos autores como, “o novo latim do currículo escolar”, no sentido que ela pode ajudar no desenvolvimento do raciocínio lógico, embora de difícil aprendizagem. O mesmo autor destaca que ainda hoje estudar programação não é uma tarefa simples. Este é um dos pontos para explicar o alto número de evasão nos cursos de Computação das universidades brasileiras. As principais dificuldades apontadas envolvem entender a introdução à computação, lógica de programação a sintaxe das linguagens (RIBEIRO; BRANDÃO; BRANDÃO, 2012).

Em muitos casos o aluno tem o seu primeiro contato com os aspectos do PC durante a graduação, tornando o aprendizado ainda mais difícil. Segundo Scaico et al. (2012) a experiência dos Estados Unidos com K-12<sup>3</sup> tem o objetivo de abranger os seguintes aspectos:

[...] preparar os alunos para entender a natureza da Ciência da Computação, incluindo seus princípios, e desenvolver as habilidades necessárias para resolver problemas cotidianos, o que também é útil para o aprofundamento de assuntos na área de Computação. introdução da Computação inicia-se com os alunos na faixa etária de oito anos, os quais começam a estudar conceitos básicos da computação. (SCAICO et al., 2012, p. 7).

Estima-se que o PC deveria ser ensinado ainda nas séries iniciais, já que aumenta a capacidade de dedução e a resolução de problemas (SICA, 2011). Ademais, grande parte das profissões do século XXI exigem o conhecimento de competências

<sup>3</sup> K-12 é uma designação para a educação primária e secundária, usada nos Estados Unidos, em algumas partes da Austrália e no Canadá.

próprias de ciências da computação. Assim o PC, segundo [França, Silva e Amaral \(2012, p. 2\)](#) “[. . .] é saber usar o computador como um instrumento de aumento do poder cognitivo e operacional humano, aumentando a nossa produtividade, inventividade, e criatividade”.

### 3.2 DISCIPLINAS DE ALGORITMOS E SUAS DIFICULDADES

O alto índice de reprovação e, em alguns casos, a desistência nos cursos de Computação pode estar relacionado com a falta de compreensão do raciocínio lógico ([PRIETCH; PAZETO, 2010, p. 2](#)). Segundo [Neto e Schuvartz \(2007\)](#):

[. . .] os cursos da área de Computação e Informática enfrentam um grande problema com as disciplinas de introdução à programação de computadores, as quais visam ensinar como utilizar o computador para solucionar problemas. Acadêmicos iniciantes, ao se depararem com a disciplina, sentem-se incapazes de programar, devido ao conjunto de habilidades que a programação exige como capacidade para solucionar problemas, raciocínio lógico, habilidade matemática, capacidade de abstração, entre outras ([NETO; SCHUVARTZ, 2007, p. 521](#)).

Os professores entendem que a maioria dos alunos encararam essas habilidades de programação pela primeira vez durante a graduação e, por isso, se esforçam para aplicar uma estratégia de ensino adequada para tornar o aprendizado mais suave. Apesar de a desistência de um curso de graduação estar relacionada a uma série de fatores, a falta de um conhecimento prévio sobre PC acarreta um alto índice de reprovação nas disciplinas iniciais de programação, a aversão a linguagens de programação e até a evasão ao curso ([NETO; SCHUVARTZ, 2007, p. 2](#)). Um estudo realizado em cursos de Computação na cidade de Curitiba mostra que 22% dos estudantes acham as disciplinas de programação desestimulantes, sendo que 31% sentem grande dificuldades nessas disciplinas. Essa dificuldade cresce para 60% quando o estudante entrou na graduação por indicação familiar, enquanto é de 15% entre os que gostavam de informática e 19% entre os que já atuavam ([MORÃES; POMBEIRO, 2016, p. 9-10](#)).

As disciplinas introdutórias de Algoritmos não são restritas aos cursos de computação e afins, estando presentes na maioria das grades curriculares dos cursos da área de Ciências Exatas e Tecnologia. O processo mental para a aprendizagem de Algoritmos é novo para a maior parte dos alunos, o que representa dificuldades e muitas vezes resistência ao aprendizado ([FALKEMBACH et al., 2003](#)). Para compreender a abstração dos conteúdos de Algoritmos se faz necessário o desenvolvimento do PC, o que serve não somente para quem vai estudar computação, mas para todos os demais estudantes, visto que aumenta a capacidade de dedução e resolução de problemas ([FRANÇA; SILVA; AMARAL, 2012, p. 2](#)). O PC não é uma habilidade mecânica ou

utilitária: é uma habilidade fundamental que permite a resolução de problemas diversos utilizando computadores e que deveria ser ensinada para todas as pessoas.

### 3.3 INICIATIVAS EM OUTROS PAÍSES

Em várias partes do mundo, governos e educadores estão adotando estratégias para que o desenvolvimento do PC seja implantado em suas escolas, Santos (2017) recorda que alguns países saíram na frente, procurando implementar novas tecnologias que promovam avanços no ensino e no próprio ensino por meio de da inclusão de disciplinas em seu plano curricular oficial. Por exemplo, desde de 2014 o Reino Unido vem implementando o ensino de programação nas escolas. Outras nações vem promovendo iniciativas parecidas, provendo debates significativos sobre o ensino de programação para jovens, cada vez que existe alguma revisão curricular (SANTOS, 2017).

Heintz, Mannila e Färnqvist (2016) listam dez países que estão fazendo progresso no ensino do PC em suas escolas: Austrália, Inglaterra, Estônia, Finlândia, Nova Zelândia, Noruega, Polônia, Coreia do Sul, Suíça e Estados Unidos.

Existe um cenário de mudanças no ensino em escala global. Com isso em mente, são destacadas abaixo as iniciativas em alguns países, para o desenvolvimento do PC em crianças.

#### 3.3.1 Estados Unidos

Em 2006, quando Jeanette Wing lançou uma discussão sobre o papel do PC em todas as disciplinas, ela acendeu um profundo engajamento com as principais questões entre o computador e a ciência e como isso pode contribuir para resolver problemas da investigação humana. Ela argumentou que os avanços em Computação permitem que pesquisadores de outras disciplinas visualizem novas estratégias de resolução problemas e testem novas soluções. Wing também imaginou que o PC deveria se tornar essencial na educação de cada criança (BARR; STEPHENSON, 2011), fazendo com que a inclusão do ensino de programação para crianças no K-12 fosse algo a ser pensado.

Um dentre muitos projetos em desenvolvimento é o *iDREAMS (Integrative Design-based Reform-oriented Educational Approach for Motivating Students)* que visa usar o design de jogos para ensinar Ciência da Computação nas escolas com o currículo K-12. O objetivo é motivar e educar os alunos por meio de uma abordagem chamada de Design de Jogo Escalável, começando no nível médio. O projeto tem um alcance amplo, atingindo centros tecnológicos, áreas urbanas, rurais e remotas. Um objetivo importante é desenvolver recursos locais e capacidade de sustentabilidade. Fazendo isso por meio

do treinamento intensivo de professores e o envolvimento da comunidade local. Seus criadores recomendam que essas competências sejam ensinadas desde os primeiros anos do ensino regular, visto que este é o lugar onde se pode alcançar um número mais heterogêneo de indivíduos. Argumentam que os programas extra curriculares, apesar de bem sucedidos, na maioria das vezes se baseiam na auto-seleção e alcançam apenas um conjunto pequeno de crianças quando comparado com o número alcançado nos currículos regulares (IOANNIDOU et al., 2011).

### 3.3.2 Alemanha

Na Alemanha surgiram diversas ideias. Entre os anos de 2001 e 2003, foi implementado um modelo de projeto denominado ArtDeCom que reuniu arte, design e informática na educação primária, equivalente ao ensino fundamental I brasileiro. O projeto incluiu a identificação de tecnologias digitais para serem moldadas e programadas por alunos e não simplesmente a aprendizagem baseada no aluno na frente do computador. A abordagem foi financiada e através do programa alemão BLK de “Educação cultural na era da mídia” e com acompanhamento do *MediaArtLabSchool* da Universidade de Flensburg. Iniciativas mais amplas, como *Kids in media and motion (KiMM)* da Universidade de Lübeck, abordam e exploram a tecnologia do computador como um meio para processos criativos (GARCÍA-PEÑALVO et al., 2016).

Nos últimos anos, foram desenvolvidas abordagens no contexto das práticas “faça você mesmo” e *Maker-movement* (movimento criador) criando laboratórios de mídias e publicações como “*einfach Programmierung und digitale Erstellung*” (programação e criação digital fáceis) e “*Aktivitäten mit Kindern und Jugendlichen machen*” (como fazer atividades com crianças e jovens). Isso é uma tendência de aprendizagem transdisciplinar e informal, conceitos que foram além das concepções tecnocráticas de tecnologia, passando a integrar um ambiente criativo para aprendizagens significativas em contextos artísticos (GARCÍA-PEÑALVO et al., 2016).

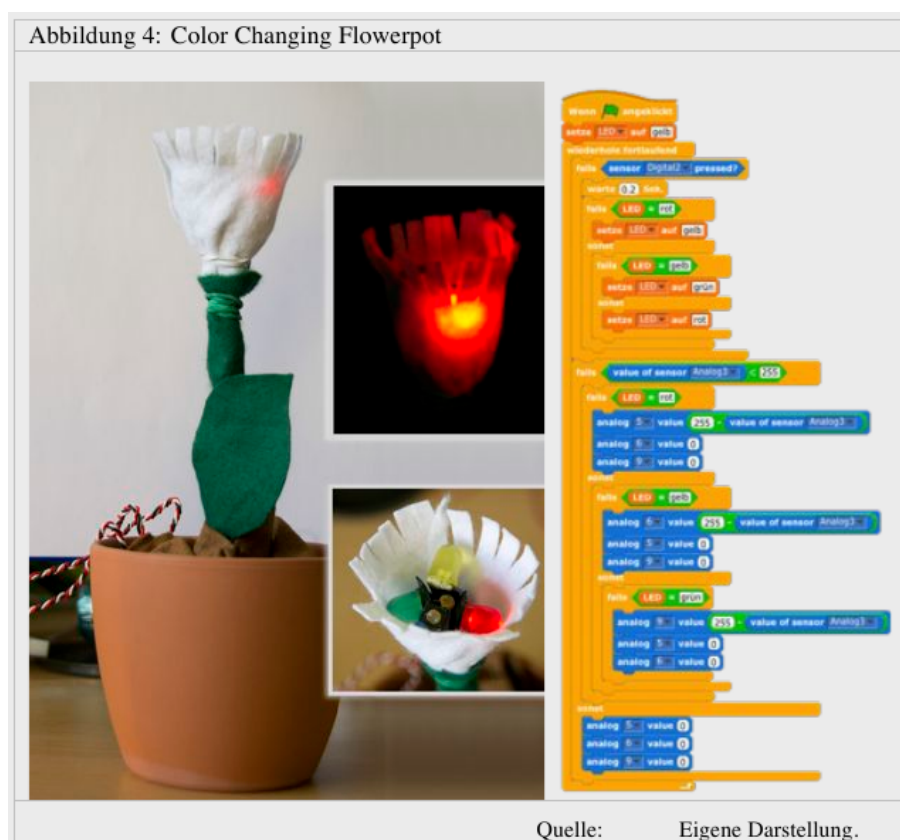
Existem algumas iniciativas para introduzir a computação física e abordagens mais criativas de design e artesanato para aprender sobre ciência da computação, por exemplo o projeto, “*My interactive garden*”<sup>4</sup> que é baseado tecnologia Arduino<sup>5</sup> e utiliza o *Scratch* para interagir com objetos onde os alunos desenvolvem e se juntam no quadro do jardim interativo (FIGURA 2). Com os computadores cada vez mais presentes na sociedade, devido aos muitos sistemas incorporados, existe a necessidade de tornar o conteúdo da Ciência da Computação menos abstrato. Uma forma interessante de fazer isso é desenvolver projetos que promovam as práticas, isso tornam esse

<sup>4</sup> Meu Jardim Interativo em português. <<https://www.youtube.com/watch?v=DpLnnHhcFNk>> e <<http://www.cs.uni-potsdam.de/~mprz/>>

<sup>5</sup> Arduino é uma plataforma eletrônica de código aberto baseada em hardware e software fáceis de usar. <<https://www.arduino.cc/en/Guide/Introduction>>

conteúdo acessível a uma parcela maior da sociedade. O conceito de ensino do “*My Interactive Garden*” foi desenvolvido com esse objetivo, tendo uma ambiente de design, programação e aprendizagem (PRZYBYLLA; ROMEIKE, 2013).

FIGURA 2 – My Interactive Gardem



Fonte: Captura de tela

Outra iniciativa colocada em prática são as *Smart textile* ou “roupas inteligentes”. Pode até ser questionável o uso dessa tecnologia para desenvolver o PC. No entanto se tem obtido sucesso no seu uso. São roupas e acessórios com microcomputadores embutidos que oferecem várias possibilidades de interação com o ambiente. Elas utilizam fios condutores, sensores, motores, luzes de LED e placas de circuito impresso. Nesse contexto o Arduino pode ser utilizado para controlar os componentes utilizados nos tecidos <sup>6</sup>. Assim os alunos utilizam esses componentes e aprendem a costurar, programar e projetar circuitos pelo caminho. É uma tecnologia para todas as idades e uma nova maneira de introduzir alunos à tecnologia por meios de aprendizagem baseada em artes. Um dos objetivos do projeto é estimular o interesse de meninos e meninas pela tecnologia. Essa combinação de arte com aprendizagem com conceitos de Ciência da Computação deverá desencadear maior interesse e compreensão mais profundos da tecnologia (GARCÍA-PEÑALVO et al., 2016).

<sup>6</sup> <<http://lilypadarduino.org/>>

Em um mundo abstrato como o da Ciência da Computação, atrair a atenção das crianças parece ser algo difícil. Mas utilizando a aprendizagem baseada em projetos por meio da física, artes, design, roupas inteligentes e Arduino, os alunos podem inventar, desenvolver, desenhar, conectar, construir, programar, controlar e depurar. Ao fazerem isso usam sua imaginação desenvolvendo o PC para a resolução de problemas (GARCÍA-PEÑALVO et al., 2016).

Na Alemanha existe um esforço para o desenvolvimento computacional nas escolas desde as séries iniciais, é interessante notar que lá existe um sistema de ensino com autonomia estadual e há diferenças em cada unidade da federação. Na Baviera por exemplo a Ciência Informática é obrigatória no 6º, 7º, 9º e 10º e o PC durante o 11º e o 12º anos<sup>7</sup>(HUBWIESER, 2012). No estado da Renânia do Norte-Vestefália é obrigatório o PC para o 7º, 8º, 9º, 12º e 13º anos <sup>8</sup>

### 3.4 INICIATIVAS NO BRASIL

No Brasil, o debate para a implantação de um currículo que visa o desenvolvimento do PC ainda está em fase inicial. Um projeto de lei, criado em 2011 no estado de Roraima, viabiliza a implantação da disciplina de computação a partir do ensino fundamental II. Outros estados tem promovido iniciativas semelhantes de forma experimental voltadas para alunos do fundamental e médio (FRANÇA; AMARAL, 2013).

Em uma pesquisa a respeito do cenário brasileiro a cerca do ensino de Computação nas educação básica, França e Amaral (2013), chegaram a seguinte conclusão:

A organização apresentada para os conteúdos sugere que, no Brasil, há pesquisas que acompanham as discussões internacionais sobre o que deve ser ensinado nas escolas, a exemplo dos estudos que abordam sobre os fundamentos da Computação, enquanto, ciências, a estudantes na educação básica. No entanto, há também aqueles que limitam suas atividades ao ensino de ferramentas básicas, a exemplo de Word e Excel. Em relação aos artefatos utilizados e/ou propostos, a robótica obteve maior citação. Do mesmo modo, a computação desplugada vem ganhando espaço, possibilitando que atividades que versem pelo ensino de Computação sejam executadas até mesmo em escolas carentes de recursos tecnológicos (FRANÇA; AMARAL, 2013, p. 431).

Uma iniciativa interessante foi a de Silva, Silva e França (2017), em que alunos do curso de Licenciatura em Computação da Universidade Federal Rural de Pernambuco (URRPE), em parceria com o Centro de Tecnologia em Educação (CETEC) da cidade de Recife, organizaram um curso para a formação de professores da educação básica para o ensino do PC. O curso teve 16 horas e seu objetivo foi apresentar os

<sup>7</sup> Os números se referem ao ciclo Educacional Alemão

<sup>8</sup> <<http://www.amanyadav.org/competencies-in-computational-thinking-perspectives-from-germany/>>

conceitos do PC e sua aplicabilidade em sala de aula por meio de atividades lúdicas (SILVA; SILVA; FRANÇA, 2017). Esse projeto aponta para a “disseminação do PC na formação continuada de professores da educação básica” frisando que ele é importante para todas as áreas do conhecimento, ou seja, é multidisciplinar.

Outro projeto foi de Rodrigues e Sousa (2017) que formulou uma proposta de curso que pudesse “desenvolver fundamentos da computação em crianças de 8 a 12 anos”, utilizando como base a Computação Desplugada e o Code.org, para desenvolver atividades lúdicas em forma de desafios. Os autores salientam que os resultados foram satisfatórios visto que auxiliaram os alunos a adquirir um maior nível de abstração, que é fundamental na resolução de exercícios (RODRIGUES; SOUSA, 2017).

O trabalho de Raabe et al. (2017) é bem mais direcionado. Trata-se de um processo de implantação do PC como disciplina obrigatória no ensino fundamental e médio em três colégios de aplicação da Universidade do Vale do Itajaí (Univali). A disciplina é ofertada para os alunos do 6º ano do ensino fundamental e do 3º ano do ensino médio, totalizando 25 turmas, em um total de 700 alunos. Os professores são todos bacharéis em Ciência da Computação ou Sistemas de Informação. O relato dos professores de outras disciplinas aponta que “as habilidades exploradas nas aulas de PC auxiliaram no progresso dos alunos em suas disciplinas específicas” (RAABE et al., 2017).

É importante destacar que os Licenciados em Computação reúnem as qualificações necessárias para atuarem como professores das disciplinas de Ciência da Computação. Durante a graduação esses profissionais adquirem conhecimentos que os habilitam para compreenderem a dinâmica de questões sociais, legais e éticas, tendo uma profunda percepção do impacto que o uso das tecnologias na vida do cidadão. As disciplinas teóricas são capazes de fornecer o conhecimento necessário para ajudar a desenvolver o PC em jovens e adultos. Ademais, as práticas durante o curso de graduação dão uma importante visão da necessidade da interdisciplinaridade, buscando fazer o egresso capaz de trabalhar em conjunto com outros profissionais a fim de aumentar o desempenho e a compreensão do aluno. Os conceitos pedagógicos presentes no curso, ajudam a relacionar as tecnologias com o contexto social, desempenhando um papel transformador não somente no desenvolvimento do PC, mas promovendo também, uma integração social do cidadão.

## 4 ESCOLHA DA FERRAMENTA

A escolha da ferramenta não é uma tarefa trivial. Isso se deve muito pela realidade que se encontra em grande parte das escolas públicas quando o assunto são as novas tecnologias (BIELSCHOWSKY, 2009). Existem diversas ferramentas que tem como meta o ensino de programação sendo que, nos últimos anos, tem havido um foco maior para que linguagens de programação sejam ensinadas para crianças. Deste modo podemos ver conteúdos especialmente preparados para elas. Seguem algumas ferramentas analisadas:

### 4.1 CODE.ORG

Code.org é uma organização sem fins lucrativos cujo objetivo é divulgar e ensinar programação a pessoas de todas as idades. É uma instituição que tem parcerias com algumas empresas de peso na área da tecnologia. Seu lema é “Todos os alunos, em todas as escolas, devem ter a oportunidade de aprender ciência da computação” (PARTOVI, 2017). Trata-se de um esforço comum para promover o ensino de programação para todas as pessoas, especialmente as crianças.

Na tela inicial do site o usuário tem a possibilidade de escolher um dentre os muitos idiomas disponíveis, inclusive português do Brasil. A plataforma usa especialmente jogos para ensinar fundamentos de programação e esta forma parece adequada, pois tem atraído a atenção das crianças para os aplicativos do site.

Em evento de 9 a 15 de dezembro de 2013, o Code.org lançou a “Hora do Código”<sup>1</sup> (FIGURA 3) nos Estados Unidos: um desafio realizado em escolas no qual os estudantes tentam resolver pequenos desafios computacionais durante uma hora, visando ensinar Ciência da Computação aos estudantes. A iniciativa teve tanto sucesso que logo se espalhou pelo mundo. Universidades selecionam alunos voluntários que ajudam a desenvolver os desafios e levam até as escolas. Todos são envolvidos nas atividades desde professores de todos os níveis educacionais até alunos da Educação Básica e os estudantes universitários. Todas as atividades propostas podem ser acessadas a partir do portal do projeto sendo necessário criar uma conta gratuita no site para a participação em algumas atividades.

O conteúdo do site tem diferenciação para alunos e educadores. Desta forma, quem tem interesse na própria aprendizagem é direcionado aos conteúdos específicos, com atividades desenvolvidas especialmente para quem deseja aprender a programar.

---

<sup>1</sup> Evento realizado anualmente durante A Semana da Educação em Ciência da Computação, em reconhecimento ao aniversário da pioneira da computação, a Almirante Grace Murray Hopper (9 de dezembro de 1906).

O nível das atividades vai aumentando de acordo com o progresso do aluno. Já no conteúdo para os educadores existem tutoriais com passos para a aplicação dos conteúdos de programação em salas de aulas. No [Anexo A](#) está disponibilizado parte do Guia do Usuário da Hora do Código.

FIGURA 3 – CODE.ORG - A HORA DO CÓDIGO



Fonte: Captura de tela

## 4.2 SCRATCH

O *Scratch* é um *software* que se utiliza de blocos lógicos, itens de som e imagem para que qualquer pessoa possa desenvolver sua própria história interativa, jogos e animações. É um projeto do grupo *Lifelong Kindergarten* no *Media Lab* do MIT (Massachusetts Institute of Technology<sup>2</sup>) idealizado por Mitchel Resnick, pensado especialmente para crianças entre 8 e 16 anos, mas pode ser usado por qualquer pessoas. Hoje o aplicativo está disponível em cerca de 40 idiomas em mais de 150 países. É disponibilizado gratuitamente para os principais sistemas operacionais - Windows, Linux e Mac ([RESNICK, 2017](#)).

Está disponível online no endereço [<https://scratch.mit.edu/>](https://scratch.mit.edu/) no qual as produções podem ser compartilhadas, mediante um cadastro. Também pode ser instalado como aplicativo em um computador para ser acessado sem conexão com a internet e existem as versões 1.4 para Linux, Windows e Mac, e a versão 2.0 que precisa do Adobe Air. O MIT fornece aos usuários do *Scratch* variados meios de criarem projetos educacionais com a ferramenta e são disponibilizados sites que tratam de iniciativas e comunidades que falam sobre a plataforma. No Brasil existem diversas iniciativas para o uso do *Scratch*, mas ainda há muito a ser feito. Algumas comunidades e canais oferecem tutoriais em português para o aprendizado da plataforma.

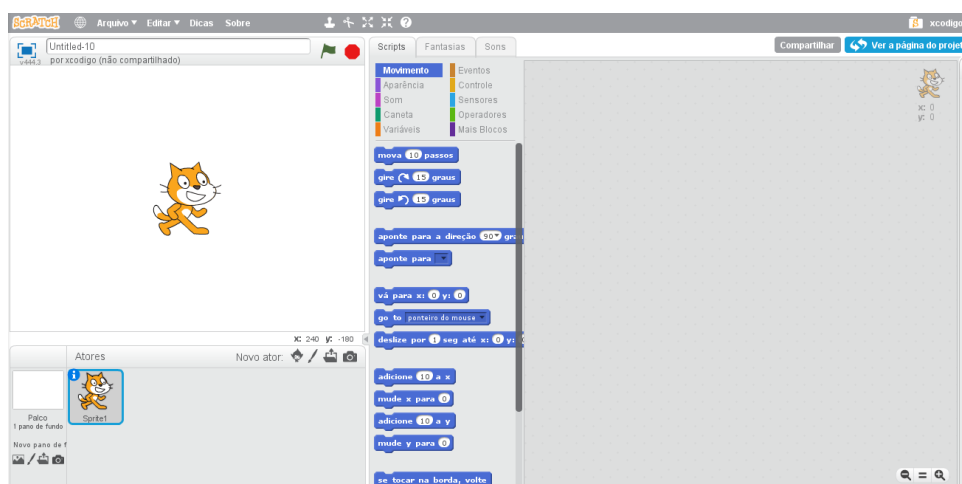
<sup>2</sup> Instituto de Tecnologia de Massachusetts em português

O *Scratch Day* é um dia especial no qual programadores reúnem-se em eventos, nas mais diferentes cidades ao redor do mundo, para compartilhar conhecimentos sobre o *Scratch*. Com ele, segundo [Voelcker, Fagundes e Seidel \(2008\)](#):

É possível trabalhar os seguintes conceitos específicos de programação: sequência, iteração, condição, variáveis, execução paralela, sincronia, interação em tempo real, lógica booleana, números randômicos, tratamento de evento e criação de interfaces ([VOELCKER; FAGUNDES; SEIDEL, 2008](#), p. 9).

A [FIGURA 4](#) mostra a IDE (Integrated Development Environment<sup>3</sup>) do *Scratch*: sua interface é muito prática, não é preciso digitar funções, endereços, etc. 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 ([RESNICK, 2017](#)).

FIGURA 4 – IDE DO SCRATCH



Extraída de: captura de tela (<https://scratch.mit.edu/>)

A interface vem com um grande número de personagens e itens, sendo possível adicionar outros. Em poucos minutos já é possível criar um programa simples. A lógica e os conceitos de programação são compreendidos a medida que se utiliza a aplicação.

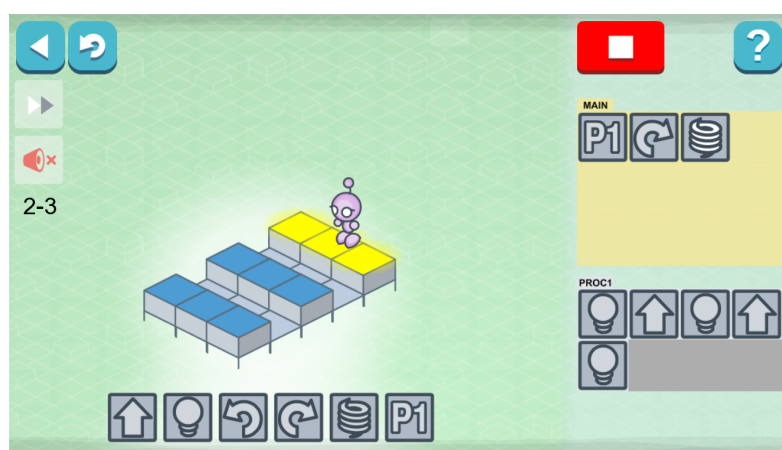
O *Scratch* usa a programação em blocos, na qual cada bloco precisa ser encaixado em sequência com outros, de forma semelhante a um quebra-cabeças. Esses encaixes formam a lógica da linguagem, que de um modo bastante intuitivo criam os eventos para serem executados. Trata-se de uma maneira fácil e divertida de programar. No *Scratch* os blocos estão dispostos em abas e cada um conjunto tem uma cor diferente. Isso facilita o aprendizado e deixa a interface mais intuitiva e organizada. Mais informações sobre o *Scratch* estão disponíveis no [Anexo B](#).

<sup>3</sup> Ambiente de Desenvolvimento Integrado em português

### 4.3 LIGHTBOT

O *Lightbot* <sup>4</sup> é um jogo desenvolvido por Danny Yaroslavski, que tem como foco principal o ensino da lógica computacional e noções básicas de algoritmos. Com o *Lightbot* é possível executar instruções em um robô para que ele caminhe na plataforma e atinja o objetivo de cada etapa do jogo. Conforme pode ser observado na FIGURA 5, o usuário do jogo é desafiado a conduzir o robô por um caminho e a medida que ele passa os quadrados vão acendendo. Para cumprir os desafios alguns objetos precisam ser utilizados, como lâmpadas e molas (SILVA; SILVA; FRANÇA, 2017). Além de lógica o jogo introduz as noções de laços de repetição e procedimentos.

FIGURA 5 – LIGHTBOT.



Extraída de: <http://lightbot.com/>

O aplicativo está disponível para Android e Iphone/iPad, Windows, MAC, Kindle e existem uma versão em flash para acesso web. Existem duas versões a *Lightbot Jr* é indicado para crianças de 4 a 8 anos de idade, já o *Lightbot* para crianças a partir do 9 anos. A versão paga oferece mais de 40 níveis com dificuldade crescente, enquanto que a versão gratuita contem 20 níveis, o que já trabalha bastante os conceitos propostos no jogo. O fato de poder ser usado em *Smartphone* é um ponto positivo, pois nem sempre existem computadores disponíveis para desenvolver as atividades. Por outro lado o acesso aos *Smartphones* tem sido cada vez mais comum. O Anexo C traz um relato do blog Caminhos e Espaços <sup>5</sup> sobre o *Lightbot*.

### 4.4 PROGRAMAÊ

Em 2014 a empresa Code Club Brasil, em parceria com a fundação Lemman, lançou a plataforma online de programação denominada "Programaê"<sup>6</sup>: uma ferramenta

<sup>4</sup> <<https://lightbot.com>>

<sup>5</sup> <<http://caminhosespacos.blogspot.com.br/2013/03/light-bot.html>>

<sup>6</sup> Disponível em <[programae.org.br](http://programae.org.br)>

gratuita que facilita o aprendizado de programação para crianças e jovens, oferecendo conteúdos de outras organizações como *Codecademy*, *Khan Academy*, *Scratch* e *Code.org* (LUCRÉCIO, 2017).

O Programaê funciona da seguinte forma: Os novos usuários são incentivados a fazer um cadastro, embora os conteúdos possam ser acessados sem o registro. Os conteúdos são diferenciados para quem deseja ensinar e para quem deseja aprender, bastando escolher entre “Quero Aprender” ou “Quero Ensinar”. Após sua escolha, o usuário pode entrar no roteiro apropriado. Existem as trilhas por áreas além de vídeos que enriquecem o conteúdo da plataforma. O Anexo D ilustra a página inicial da plataforma.

Uma característica interessante do Programaê é que não se reinventa a roda na plataforma, sendo utilizados recursos de outras plataformas em seus conteúdos (FIGURA 6). Por exemplo, no curso “Para Além o Computador” disponível no Programaê, são utilizados vídeos e atividades do Codeorg, do *Scratch* e também do Computer Science Unplugged<sup>7</sup>, ele de fato reúne plataformas. As atividades são bem organizadas e em muitos casos já contam com o material pronto. A organização das aulas pode ser vista na TABELA 1:

FIGURA 6 – PROGRAMAÊ REÚNE PLATAFORMAS.



Extraída de: <http://programae.org.br>

#### 4.5 COMPUTAÇÃO DESPLUGADA

Muitos conceitos computacionais podem ser ensinados sem uso do computador, o que pode ser interessante se levarmos em conta a estrutura presente em muitas escolas no Brasil. Para ensinar o pensamento computacional o computador não é um item indispensável. Segundo Brackmann et al. (2017):

<sup>7</sup> CS Unplugged: <http://csunplugged.org/> - computação desplugada em português

TABELA 1 – ORGANIZAÇÃO DAS AULAS DO PROGRAMAÊ

Seção	Meta
Objetivos	Define para o professor quais são os objetivos a serem alcançados nesta aula
Conteúdos	Mostra quais conteúdos de ensino são aplicados
Materiais	Lista todos os materiais utilizados durante a aula
Preparação	Deixa claro o que o professor precisa saber para ensinar com competência
Desenvolvimento	Apresenta a ficha da aula que deve ser seguida
Avaliação	Um roteiro com uma série de questionamento para elaborar uma Avaliação

A abordagem desplugada da CC [Ciência da Computação] introduz conceitos de hardware e software que impulsionam as tecnologias cotidianas até pessoas não-técnicas. Em vez de participar de uma aula expositiva, as atividades desplugadas ocorrem frequentemente através da aprendizagem cinestésica (e.g. movimentar-se, usar cartões, cortar, colar, desenhar, pintar, resolver enigmas, etc.) e os estudantes trabalham entre si para aprender conceitos da CC (BRACKMANN et al., 2017, p. 983).

O site CS Unplugged<sup>8</sup> traz uma coleção de atividades projetadas para auxiliar no ensino de computação sem o uso do computador. As atividades incluem jogos, quebra-cabeças, brincadeiras, bem como uso de cartas, cordas, lápis de cor entre outras. Os criadores do CS Unplugged, originalmente desenvolveram as atividades para que jovens estudantes pudessem ter o primeiro contato com a ciência da computação. Isto poderia levar a perguntas e desafios que os cientistas da computação também experimentam. Fazendo desta forma, os estudantes podem vivenciar essas práticas sem a necessidade aprender programação primeiro (BELL; WITTEN; FELLOWS, 2011).

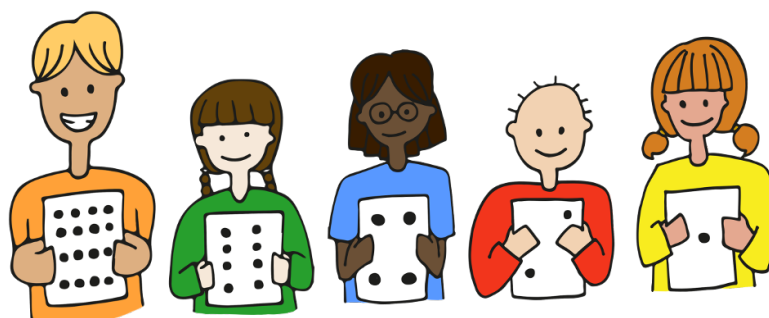
A coleção foi originalmente destinada a ser um recurso de divulgação e extensão, mas hoje é adotada para o ensino da computação e do PC em muitas salas de aula ao redor do mundo, sendo amplamente utilizada para o ensino.

O site reúne ideias associadas a cada atividade que podem ser encontradas nas páginas de atividade individuais. Na FIGURA 7 observamos a ilustração da atividade números binários, que utiliza cartas binários para fixar os conceitos base numérica, utilizada na Ciência da Computação, de forma divertida.

O material do CS Unplugged, é usado em muitos contextos fora da sala de aula, incluindo mostra de ciências, conversas para idosos e eventos especiais. Como é um projeto aberto, muitas pessoas podem contribuir com suas atividades, seja na tradução do conteúdo ou criando novas atividades. Existe um livro em formato PDF, disponível também em português, com diversas atividades que pode ser baixado diretamente no

<sup>8</sup> <[csunplugged.org](http://csunplugged.org)>

FIGURA 7 – NÚMEROS BINÁRIOS



Fonte: <http://csunplugged.org>

site do projeto.<sup>9</sup>

O CS Unplugged tem 11 princípios que regem o funcionamento do projeto: CS desconectado, Nenhum Computador necessário, Ciência da Computação real, Aprender fazendo, Diversão, Nenhum equipamento especializado, Modificações incentivadas, Para todos, Cooperativa, Atividades autônomas, Resiliente. No [Anexo E](#) é apresentada a forma de tratar o PC com CS Unplugged.

Muitas outras ferramentas poderiam ser citadas, mas não é objetivo desse trabalho listar todas o material disponível para o desenvolvimento do PC, porém é de interesse mostrar que existem ferramentas para diversas situações encontradas nas escolas, como descrito no [Capítulo 2](#). O importante é conhecer o que tem disponível e adaptar a cada realidade. O mundo da computação é muito dinâmico e novos aplicativos surgem o tempo todo. É imperioso ao bom profissional licenciado em Computação estar familiarizado com as novas tecnologias por meio de estudos continuados e atualizações.

<sup>9</sup> Link atual encurtado: <https://goo.gl/gVR6yg>

## 5 SUGESTÃO DE PROCEDIMENTO

Além de conhecer as possíveis ferramentas, como apresentado no [Capítulo 4](#), é essencial conhecer a realidade de onde se pretende aplicá-las. Isso é relevante devido a existência de um número considerável de aplicativos e plataformas com diferentes abordagens o que torna possível uma seleção adequada para cada realidade. Seria muito bom se todas as escolas tivessem a infraestrutura considerada mínima necessária, como projetor multimídia, laboratórios atualizados e conexão com a internet suficiente para o desenvolvimento das atividades. Sabemos, no entanto, que isso nem sempre é possível. Para a presente análise, são projetados dois possíveis cenários.

### 5.1 CENÁRIO OTIMISTA

Em um cenário otimista teremos projetor multimídia em pelo menos uma sala para apresentação de vídeos e um laboratório com computadores modernos que suporte os softwares descritos nesta pesquisa.

Neste caso, uma proposta seria a utilização das aulas do Programaê ([seção 4.4](#) e [Anexo D](#)) pois a plataforma conta com planos de aulas com um programa bem definido com níveis crescentes. Como esta ferramenta tem uma integração com diversas outras ferramentas, ela é ideal quando o cenário encontrado na escola permite a sua aplicação. O próprio programa disponibilizado no site é uma orientação para a execução das atividades.

Essa abordagem permite utilizar todo o potencial das diversas ferramentas, fazendo uma introdução ao aprendizado computacional. O uso de diversas plataformas em conjunto, como é a proposta do Programaê, favorece o trabalho em equipe e a cooperação dos alunos. O professor cumpre bem um papel de mediador da aprendizagem, enquanto trabalha os conceitos computacionais fortalecendo a ideia do uso do computador como ferramenta de conhecimento. Desta forma os alunos param de ser apenas super usuários de computadores, deixando de usá-lo apenas com fins de diversão e entretenimento, para aprender como programar o computador e extrair sua potencialidade para solução de problemas.

### 5.2 CENÁRIO PESSIMISTA

Por outro lado, a realidade de muitas escolas não é exatamente o cenário idealizado acima. [Nakagawa e Berghauser \(2017\)](#) salientam que “que as políticas públicas voltadas para a implantação de infraestruturas de informática em escolas públicas estão muito aquém do esperado e do considerado mínimo” e isto dificulta

o trabalho dos professores e limita o uso de algumas ferramentas. Em um cenário pessimista, não teríamos uma sala de projeção, nem computadores adequados à utilização de alguns aplicativos, o que inviabilizaria utilizar o Programaê, por exemplo.

Uma sugestão é o uso de CS Unplugged ([seção 4.5](#) e [Anexo E](#)), utilizando as atividades propostas que envolvem jogos e brincadeiras, com o objetivo de fixar os conceitos computacionais. Trata-se de uma importante ferramenta para o desenvolvimento do pensamento computacional, e ela pode ser utilizada de forma eficaz em situações como essas. As atividades envolvem cooperação e trabalho em equipe, que de uma forma lúdica, despertam os estudantes para explorar suas capacidades, sendo uma forma interessante de desenvolver o PC. Elas ajudam a desenvolver conceitos básicos de computação como sistemas de numeração, ordenação, sistemas de buscas, conceitos lógicos, criptografia entre outros. As possibilidades são grandes com o uso das atividades propostas no livro *Computer Science Unplugged*. Mais um ponto positivo é a possibilidade de adaptar, ampliar ou modificar as atividades para atender a diversas situações. É importante salientar, que embora o computador seja muito importante para o ensino de computação, ele não é fundamental ([SILVA et al., 2017](#)).

Vale ressaltar que independente do cenário, a Computação Desplugada é um elemento essencial para a fixação de alguns princípios básicos da computação. Naturalmente, por se tratar de uma abordagem sem o uso do computador, o aplicador precisa usar uma estratégia que possa despertar o interesse dos alunos, o quais podem sentir-se decepcionados a princípio, sem o uso dos equipamentos ([LENZ; CAMBRAIA, 2015](#)). É possível encontrar na internet vídeos de aulas reais, o que pode ser um bom ponto de partida para a aplicação das atividades. Torna-se necessário despertar o interesse dos alunos, de forma que os princípios computacionais sejam ensinados de forma eficaz.

### 5.3 CONSIDERAÇÕES

Em um país continental como o Brasil, é difícil encontrar um cenário único nas escolas em relação as Tecnologias da Informação e Comunicação (TICs). Este cenário não é realista, nem mesmo quando considerando os recursos tradicionais. Portanto, é necessário adotar uma postura desafiadora, no sentido de estar disposto a criar maneiras de contornar o problemas para ensinar o PC, mesmo que as condições não sejam as melhores.

Embora a literatura mostre alguns trabalhos acadêmicos, que tem como objetivo a formação de professores da educação básica para o ensino do PC ([SILVA; SILVA; FRANÇA, 2017](#)), o ideal é que profissionais formados especificamente para isso tenham essa responsabilidade. O ensino do PC deve ser tão importante como qualquer outra disciplina, como Português, Matemática e Geografia por exemplo.

Com crescimento dos cursos de Licenciatura em Computação no país, é esperado que em breve se tenham profissionais para atender essa demanda. Segundo [Santos, Hinterholz e Silva \(2017\)](#) os Licenciados em Computação são “capazes de operar no uso e desenvolvimento de tecnologias para a educação, bem como ensinar a Computação na escola”. Partindo das dificuldades apontadas nesse trabalho, se torna claro que o assunto exige que os responsáveis pelo desenvolvimento do PC sejam profissionais com formação específica. Os desafios exigem que os professores tenham conhecimento e habilidades relevantes para o desenvolvimento dessas competências. Visto que o PC é utilizado cada vez mais em diversas áreas da sociedade, e que ele fornece uma estrutura de trabalho em conjunto com o computador para resolver problemas complexos, o ensino de computação nas escolas desde as séries iniciais não é somente necessário, é fundamental ([HEINTZ; MANNILA; FÄRNQVIST, 2016](#)).

## 6 CONSIDERAÇÕES FINAIS

Neste trabalho, foi apresentado o estudo de algumas ferramentas que auxiliam no desenvolvimento do PC em crianças, considerando critérios de escolha e atendendo a possíveis situações escolares. Todas as ferramentas apresentadas são gratuitas e fazem parte de um esforço conjunto de várias organizações para a difusão do ensino de programação e do PC.

Foi exposta a importância do desenvolvimento do PC desde as séries iniciais e as iniciativas em alguns países para a inclusão de uma disciplina obrigatória no currículo escolar. Algumas iniciativas no Brasil mostram como a comunidade acadêmica está atenta as novas tendências na educação.

Um ponto interessante ao se planejar para uma disciplina como essa é não pensar em um cenário perfeito, onde as salas já estarão preparadas para o uso de tecnologias: Nem sempre existem equipamentos compatíveis com algumas ferramentas. Nesse caso precisamos de alternativas para o ensino do PC sem computador, como a Computação Desplugada ou adaptações para uso de celulares que, por mais que sejam populares, podem não ser acessíveis a todos os alunos de uma sala de aula, dependendo do contexto socioeconômico com o qual se trabalha.

O PC lida com o desenvolvimento de habilidades que envolvem a resolução de problemas e a compreensão do comportamento humano, além de fornecer ferramentas mentais capazes de abstrair os problemas, fragmentá-los e resolvê-los separadamente. Os conceitos trabalhados durante o desenvolvimento do PC preparam para os desafios que exigem raciocínio lógico na descoberta de novas soluções. O ensino de computação, programação e o desenvolvimento do PC, são habilidades tão necessárias quanto outras disciplinas (SANTANA; MARTINS, 2017). Tendo em vista o alto nível de complexidade para o desenvolvimento dessas habilidades, o ideal é que esta área do conhecimento seja ministrada por profissionais que foram preparados para isso.

### 6.1 PROJETOS FUTUROS

Este é o começo de um longo caminho que aponta para mudanças no ensino para crianças em muitos países. Algo necessário e muito desejável são pesquisas que direcionem ações. Em trabalhos futuros pode-se fazer uma aplicação das ferramentas disponíveis em escolas com o perfil médio no Brasil, relatando dificuldades encontradas e estratégias concretas de solução. Muito promissor também é fazer um estudo de diferentes abordagens e cenários. Espera-se que esse trabalho sirva de base para estudos aprofundados no assunto e seja relevante para ajudar no estudo do PC.

## REFERÊNCIAS

- BARR, V.; STEPHENSON, C. Bringing computational thinking to k-12: What is involved and what is the role of the computer science education community? *ACM Inroads*, ACM, New York, NY, USA, v. 2, n. 1, p. 48–54, fev. 2011. ISSN 2153-2184. Disponível em: <<http://doi.acm.org/10.1145/1929887.1929905>>. Acesso em: 28.out.2017.
- BASTOS, A. M. d. M. F.; CUNHA, J. A programação no 1º ciclo do ensino básico: o projeto-piloto em duas escolas do Concelho do Seixal, Portugal. *Revista Observatório*, v. 3, n. 4, p. 331–363, 2017. ISSN 2447-4266. Disponível em: <<https://doi.org/10.20873/uft.2447-4266.2017v3n4p331>>. Acesso em: 28.out.2017.
- BELL, T.; WITTEN, I.; FELLOWS, M. *Computer Science Unplugged*. 2011. Disponível em: <<http://csunplugged.org/about/>>. Acesso em: 28.out.2017.
- BIELSCHOWSKY, C. E. Tecnologia da informação e comunicação das escolas públicas brasileiras: o programa proinfo integrado. *Revista e-curriculum*, Pontifícia Universidade Católica de São Paulo, v. 5, n. 1, 2009. ISSN 1809-3876. Disponível em: <<http://www.redalyc.org/articulo.oa?id=76613025018>>. Acesso em: 28.out.2017.
- BRACKMANN, C. et al. Pensamento computacional desplugado: Ensino e avaliação na educação primária espanhola. In: *Anais*. [s.n.], 2017. v. 6, n. 1, p. 982. Disponível em: <<http://dx.doi.org/10.5753/cbie.wcbie.2017.982>>. Acesso em: 05.nov.2017.
- CHAVES, E. O. Computadores: máquinas de ensinar ou ferramentas para aprender? *Em Aberto*, v. 2, n. 17, 1983. ISSN 2176-6673. Disponível em: <<http://www.rbep.inep.gov.br/index.php/emaberto/article/viewFile/1473/1448>>. Acesso em: 28.out.2017.
- FALKEMBACH, G. A. M. et al. Aprendizagem de algoritmos: Uso da estratégia ascendente de resolução de problemas. In: *Anais*. Santiago, Chile: [s.n.], 2003. Disponível em: <[http://www.tise.cl/2010/archivos/tise2003/papers/aprendizagem\\_de\\_algoritmos.pdf](http://www.tise.cl/2010/archivos/tise2003/papers/aprendizagem_de_algoritmos.pdf)>. Acesso em: 28.out.2017.
- FRANÇA, R. S. de; AMARAL, H. J. C. do. Ensino de computação na educação básica no Brasil: Um mapeamento sistemático. In: *Anais*. [s.n.], 2013. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/wei/2013/009.pdf>>. Acesso em: 28.out.2017.
- FRANÇA, R. S. de; SILVA, W. C. da; AMARAL, H. J. C. do. Ensino de ciência da computação na educação básica: Experiências, desafios e possibilidades. In: . [s.n.], 2012. Disponível em: <<https://goo.gl/H73sk6>>. Acesso em: 28.out.2017.
- FROZZA, R.; BRINGMANN, P. A.; CRUZ, M. E. J. K. d. Construção de conhecimento em programação a partir de simulador on-line por crianças do ensino fundamental. In: *Anais*. Santa Cruz do Sul: [s.n.], 2016. p. 440. ISSN 2237-9193. Disponível em: <[https://online.unisc.br/acadnet/anais/index.php/salao\\_ensino\\_extensao/article/view/15471](https://online.unisc.br/acadnet/anais/index.php/salao_ensino_extensao/article/view/15471)>. Acesso em: 28.out.2017.
- GARCÍA-PEÑALVO, F. J. et al. An overview of the most relevant literature on coding and computational thinking with emphasis on the relevant issues for teachers. nov. 2016. Disponível em: <<https://doi.org/10.5281/zenodo.165123>>. Acesso em: 28.out.2017.

GIRAFFA, L. M. M.; MORA, M. da C. Evasão na disciplina de algoritmo e programação: um estudo a partir dos fatores intervenientes na perspectiva do aluno. In: *Anais*. Espanha: [s.n.], 2013. Disponível em: <<http://revistas.utp.ac.pa/index.php/clabes/article/view/888>>. Acesso em: 28.out.2017.

HEINTZ, F.; MANNILA, L.; FÄRNQVIST, T. A review of models for introducing computational thinking, computer science and computing in k-12 education. In: IEEE. *Anais*. 2016. p. 1–9. Disponível em: <<https://doi.org/10.1109/FIE.2016.7757410>>. Acesso em: 28.out.2017.

HUBWIESER, P. Computer science education in secondary schools – the introduction of a new compulsory subject. *ACM Transactions on Computing Education (TOCE)*, ACM, New York, NY, USA, v. 12, n. 4, p. 16:1–16:41, nov. 2012. ISSN 1946-6226. Disponível em: <<http://doi.acm.org/10.1145/2382564.2382568>>. Acesso em: 28.out.2017.

IOANNIDOU, A. et al. Computational thinking patterns. In: *Anais*. ERIC, 2011. Disponível em: <<https://eric.ed.gov/?id=ED520742>>. Acesso em: 28.out.2017.

LENZ, R. K.; CAMBRAIA, A. C. Ensino de programação no ensino fundamental. Santa Maria, 2015. ISSN 1984-9397. Disponível em: <<https://goo.gl/uLRMYG>>. Acesso em: 28.out.2017.

LUCRÉCIO, A. I. *Comparação e aplicação de diferentes ferramentas para ensino de programação para crianças*. 93 f. Monografia (Trabalho de Conclusão de Curso) — Tecnologias da Informação e Comunicação, Universidade Federal de Santa Catarina, Araranguá, 2017. Disponível em: <<https://repositorio.ufsc.br/handle/123456789/173903>>. Acesso em: 28.out.2017.

MORÃES, M. J. F.; POMBEIRO, O. J. Evasão nos cursos de graduação em computação de curitiba. In: *Anais*. [s.n.], 2016. v. 1, n. 4, p. 2088–2103. Disponível em: <<http://portaldeperiodicos.unibrasil.com.br/index.php/anaisevinci/article/view/1081>>. Acesso em: 28.out.2017.

NAKAGAWA, J. K. M.; BERGHAUSER, N. A. C. A tecnologia e a educação: Uma realidade levantada em três escolas de um município do norte do paraná. *Revista Eletrônica Científica Inovação e Tecnologia*, v. 2, n. 16, p. 8, 2017. Disponível em: <<https://revistas.utfpr.edu.br/recit/article/view/5187>>. Acesso em: 28.out.2017.

NETO, W. C. B.; SCHUVARTZ, A. A. Ferramenta computacional de apoio ao processo de ensino-aprendizagem dos fundamentos de programação de computadores. In: *Anais*. [s.n.], 2007. v. 1, n. 1, p. 520–528. Disponível em: <<http://br-ie.org/pub/index.php/sbie/article/view/601>>. Acesso em: 28.out.2017.

PARTOVI, H. *code.org code studio*. 2017. Disponível em: <<https://br.code.org/about>>. Acesso em: 28.out.2017.

PRIETCH, S. S.; PAZETO, T. A. Estudo sobre a evasão em um curso de licenciatura em informática e considerações para melhorias. Maceió, 2010. Disponível em: <<https://goo.gl/JZYzhB>>. Acesso em: 28.out.2017.

PRZYBYLLA, M.; ROMEIKE, R. Physical Computing mit “My Interactive Garden”. Department of Computer Science, CAU Kiel, Kiel, n. 2013/3, p. 87–91, 2013. ISSN

2193-6781. Disponível em: <[http://ddi.cs.fau.de/dl/INFOS13\\_Przybylla-Romeike\\_Physical-Computing-mit-My-Interactive-Garden.pdf](http://ddi.cs.fau.de/dl/INFOS13_Przybylla-Romeike_Physical-Computing-mit-My-Interactive-Garden.pdf)>. Acesso em: 28.out.2017.

RAABE, A. et al. A experiência de implantação de uma disciplina obrigatória de pensamento computacional em um colégio de educação básica. In: *Anais*. [s.n.], 2017. v. 6, n. 1, p. 1182. Disponível em: <<http://dx.doi.org/10.5753/cbie.wcbie.2017.1182>>. Acesso em: 05.nov.2017.

RESNICK, M. [https://scratch.mit.edu/ Scratch web site](https://scratch.mit.edu/). 2017. Disponível em: <<https://scratch.mit.edu/about>>. Acesso em: 28.out.2017.

RIBEIRO, R. d. S.; BRANDÃO, L. d. O.; BRANDÃO, A. A. F. Uma visão do cenário nacional do ensino de algoritmos e programação: uma proposta baseada no paradigma de programação visual. In: *Anais*. [s.n.], 2012. v. 23. Disponível em: <<http://www.br-ie.org/pub/index.php/sbie/article/view/1797/1558>>. Acesso em: 28.out.2017.

RODRIGUES, G.; SOUSA, L. O ensino do pensamento computacional como forma de inclusão tecnológica e motivação de crianças. In: *Anais*. [s.n.], 2017. v. 28, n. 1, p. 1784. Disponível em: <<http://dx.doi.org/10.5753/cbie.sbie.2017.1784>>. Acesso em: 05.nov.2017.

SANTANA, A. L. M.; MARTINS, P. Desenvolvimento e avaliação de modificação do jogo minecraft para estimular o pensamento computacional em estudantes do ensino médio. In: *Anais*. [s.n.], 2017. v. 6, n. 1, p. 92. Disponível em: <<http://dx.doi.org/10.5753/cbie.wcbie.2017.92>>. Acesso em: 05.nov.2017.

SANTOS, N. F. G. dos. Programação para jovens: Conteúdos, atividades estratégias e ferramentas. 2017. Disponível em: <<https://repositorio-aberto.up.pt/bitstream/10216/102443/2/178783.pdf>>. Acesso em: 28.out.2017.

SANTOS, W. O. dos; HINTERHOLZ, L.; SILVA, C. Licenciatura em computação: Desafios e oportunidades na perspectiva do professor. In: *Anais*. [s.n.], 2017. v. 23, n. 1, p. 705. Disponível em: <<http://dx.doi.org/10.5753/cbie.wie.2017.705>>. Acesso em: 05.nov.2017.

SCAICO, P. D. et al. Um relato de experiências de estagiários da licenciatura em computação com o ensino de computação para crianças. *RENOTE - Revista Novas Tecnologias na Educação*, UFRGS, v. 10, n. 3, 2012. ISSN 1679-1916. Disponível em: <<http://www.seer.ufrgs.br/renote/article/viewFile/36377/23471>>. Acesso em: 28.out.2017.

SICA, C. *Ciência da computação no ensino básico e médio*. 2011. Disponível em: <<http://blogs.odiaro.com/carlossica/2011.out.07/ciencia-da-computacao-no-ensino-medio/>>. Acesso em: 28.out.2017.

SILVA, N. C. et al. Raciocínio lógico nas escolas: Uma introdução ao ensino de algoritmos de programação. In: *Anais*. Recife: SBC, 2017. v. 6, p. 1011. Disponível em: <<http://br-ie.org/pub/index.php/wcbie/article/view/7490>>. Acesso em: 05.nov.2017.

SILVA, V.; SILVA, L. L. da; FRANÇA, R. Pensamento computacional na formação de professores: experiências e desafios encontrados no ensino da computação

em escolas públicas. In: *Anais*. [s.n.], 2017. v. 23, n. 1, p. 805. Disponível em: <<http://dx.doi.org/10.5753/cbie.wie.2017.805>>. Acesso em: 05.nov.2017.

SLEEMAN, D. The challenges of teaching computer programming. *Communications of ACM*, ACM, New York, NY, USA, v. 29, n. 9, p. 840–841, set. 1986. ISSN 0001-0782. Disponível em: <<http://doi.acm.org/10.1145/6592.214913>>. Acesso em: 28.out.2017.

VALENTE, J. A. (organizer). *O computador na sociedade do conhecimento*. Campinas: Unicamp/NIED, 1999. 11–18 p.

VALENTE, J. A. Integração do pensamento computacional no currículo da educação básica: Diferentes estratégias usadas e questões de formação de professores e avaliação do aluno. *Revista e-Curriculum*, v. 14, n. 3, p. 864–897, 2016. ISSN 1809-3876. Disponível em: <<http://www.redalyc.org/articulo.oa?id=76647706006>>. Acesso em: 28.out.2017.

VOELCKER, M. D.; FAGUNDES, L. da C.; SEIDEL, S. Fluência digital e ambientes de autoria multimídia. *RENOTE - Revista Novas Tecnologias na Educação*, v. 6, n. 2, 2008. Disponível em: <<http://dx.doi.org/10.22456/1679-1916.14624>>. Acesso em: 28.out.2017.

WING, J. M. Computational thinking. *Communications of the ACM*, v. 49, n. 3, p. 33–35, 2006. Disponível em: <<http://tech-insider.org/academia/research/acrobat/0603.pdf>>. Acesso em: 28.out.2017.

WING, J. M. *Computational thinking benefits society*. 2014. 40th Anniversary Blog of Social Issues in Computing. Disponível em: <<http://socialissues.cs.toronto.edu/category/general/index.html>>. Acesso em: 28.out.2017.

## **ANEXOS**

## **ANEXO A – A HORA DO CÓDIGO**



(/br/)

(/br/#join)



Português (Brasil) ▼

Organize um evento

# Como ensinar a Hora do Código

Junte-se ao movimento e apresente para um grupo de estudantes a sua primeira hora de informática com estas etapas. The Hour of Code is easy to run - even for beginners! If you'd like an extra set of hands to help out, you can find a **local volunteer** (<https://br.code.org/volunteer/local>) to help run an Hour of Code in your class.

## 1. Veja o guia prático em vídeo

How to Run an Hour of Code



## 2. Escolha um tutorial para sua hora

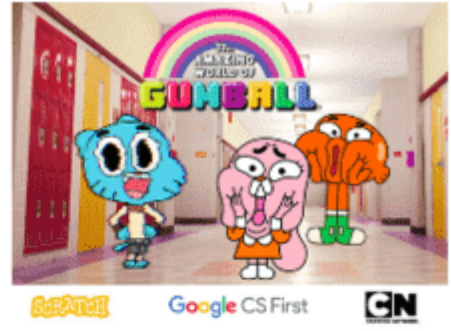
We provide a variety of fun, **student-guided tutorials** (</br/learn>) for all age groups and experience levels. Students do the activities on their own, though many activities include lesson plans for teachers (you'll see the link when you click the activity) to guide discussion or extend the activity.



**Star Wars: Building a Galaxy ...**  
All ages | Blocks, JavaScript



**Make Music**  
Grades 2-8 | Blocks, Scratch



**Gumball's Coding Adventure**  
Grades 6-8 | Blocks



**Minecraft Hour of Code**  
Grades 2+ | Blocks



**Code the News**  
Grades 6+ | Python



**Programming Lessons for Kids!**  
**Kodable (pre-readers welcome)**  
Pre-reader - Grade 5 | Blocks | Unplugged,  
iPad app

(/br/learn)

### 3. Promova sua Hora do Código

Promote your Hour of Code **with these tools (/br/promote/resources)** and encourage others to host their own events.

### 4) Avalie suas necessidades tecnológicas (computadores são opcionais)

A melhor experiência da Hora do Código inclui computadores conectados à Internet. Mas você **não** precisa de um computador para cada criança, e pode inclusive realizar a Hora do Código sem computador nenhum.

Certifique-se de testar tutoriais em computadores ou dispositivos para garantir que eles funcionem corretamente nos navegadores com som e vídeo. **Tem internet lenta?** Planeje mostrar vídeos na frente da classe, para que cada aluno não esteja baixando seus próprios vídeos. Ou tente os tutoriais desconectados / off-line.

Forneça fones de ouvido para sua turma, ou peça aos alunos que tragam seus próprios fones, se o tutorial escolhido funcionar melhor com som.

**Não tem dispositivos suficientes?** Use a **programação em duplas** (<https://www.youtube.com/watch?v=vgkahOzFH2Q>). Quando os alunos trabalham em equipe, eles ajudam uns aos outros e dependem menos do professor. Eles também verão que a ciência da computação é uma atividade social e colaborativa.



## 5. Comece sua hora de código com um áudio ou vídeo inspirador

**Invite a local volunteer** (<https://br.code.org/volunteer/local>) **to inspire your students by talking about the breadth of possibilities in computer science.** There are thousands of volunteers around the world ready to help with your Hour of Code through either a classroom visit or video chat with your students!

**Mostre um vídeo inspirador:**

- The original Code.org launch video, featuring Bill Gates, Mark Zuckerberg, and NBA star Chris Bosh. (Existem versões de **1 minuto** (<https://www.youtube.com/watch?v=qYZF6oIZtfc>), **5 minutos** (<https://www.youtube.com/watch?v=nKlu9yen5nc>) e **9 minutos** (<https://www.youtube.com/watch?v=dU1xS07N-FA>) disponíveis)
- Encontre mais **recursos** (<https://br.code.org/inspire>) e **vídeos** (<https://www.youtube.com/playlist?list=PLzdnOPi1iJNfpD8i4Sx7U0y2MccnrNZuP>) inspiradores.

**It's okay if both you and your students are brand new to computer science. Here are some ideas to introduce your Hour of Code activity:**

- Explique as formas em que a tecnologia afeta nossas vidas, com exemplos de meninos e meninas que se preocupam (falar sobre salvar vidas, ajudar as pessoas, conectar as pessoas etc).
- Em uma atividade com a turma toda, monte uma lista com as coisas que usam a programação na vida cotidiana.
- Veja dicas para estimular o interesse das meninas por ciência da computação **aqui** (<https://br.code.org/girls>).

## 6. É hora de programar!

### Direct students to the activity

- Escreva o link do tutorial em um quadro branco. Encontre o link listado nas **informações do seu tutorial selecionado (/br/learn)** sob o número de participantes.

### When your students come across difficulties it's okay to respond:

- "Eu não sei. Vamos descobrir juntos".
- "A tecnologia nem sempre funciona da maneira que queremos".
- "Aprender programação é como aprender uma nova língua, você não vai se tornar fluente imediatamente".

### What if a student finishes early?

- Os alunos podem ver todos os tutoriais e **tentar outra atividade do Hour of Code (/br/learn)**.
- Ou, peça aos alunos que terminarem mais cedo para que ajudem os colegas que estão tendo problemas com a atividade.



## Celebrando



- **Imprima certificados (<https://br.code.org/certificates>)** para seus alunos.

- Imprima adesivos com a mensagem **"Eu participei da Hora do Código!"** ([/br/promote/resources#stickers](#)) para seus alunos.
- **Encomende camisetas personalizadas** (<http://blog.code.org/post/132608499493/hour-of-code-shirts-and-more>) para sua escola.
- Compartilhe fotos e vídeos do seu evento da Hora do Código nas mídias sociais. Use #HoraDoCodigo e @codeorg, assim podemos divulgar sua experiência de sucesso também!



## Outros recursos da Hora do Código para educadores:

- Visite o **Fórum da Hora do Código para Professores** (<http://forum.code.org/c/plc/hour-of-code>) para conseguir conselhos, ideias e apoio de outros educadores.

## O que vem depois da Hora do Código?

A Hora do Código é apenas o primeiro passo de uma jornada para aprender mais sobre como a tecnologia funciona e como criar aplicativos de software. Para continuar essa jornada:

- Encourage students to continue to **learn online** (<https://br.code.org/learn/beyond>).
- **Assista** (<https://br.code.org/professional-development-workshops>) a uma oficina presencial, de um dia, para receber instruções de uma pessoa com experiência em ciência da computação. (Somente para educadores dos EUA)

([/br/#join](#))

Organize um evento

### Guias práticos

([/br/how-to](#))

Para educadores ([/br/how-to](#))

Para educadores em atividades extracurriculares ([/br/how-to/afterschool](#))

Para empresas (/br/how-to/companies)

Para cidades (/br/how-to/districts)

Para a escola toda (/br/how-to/events)

Para pais (/br/how-to/parents)

Para funcionários públicos (/br/how-to/public-officials)

Para voluntários (/br/how-to/volunteers)



(/br/promote/resources)

## **ANEXO B – SCRATCH**



## Tutorial Scratch – Conceitos básicos (versão XO-OLPC)



Scratch é uma linguagem de programação desenvolvida por *Lifelong Kindergarten Group no Media Lab, MIT*(com financiamento da *National Science Foundation, Intel Foundation, Nokia e do consórcio de pesquisa do MIT Media Lab*). Este aplicativo possibilita a criação de histórias interativas, jogos e animações bem como o compartilhamento das criações na Web. Tudo pode ser feito a partir de comandos que devem ser agrupados de modo lógico.

Neste tutorial vamos conhecer alguns comandos do Scratch, como executar um movimento do Objeto (Gato), como iniciar um projeto, salvar, abrir um projeto já existente.

**Última atualização: 31/10/2011**

### Índice

1. Tela inicial
2. Primeiro movimento no Scratch
3. Salvando o projeto
4. Abrir projeto
5. Anexo I – componentes básicos de um projeto scratch
6. Outros materiais

## 1. Tela inicial

A tela inicial do Scratch é composta por (A) uma área que apresenta e possibilita a escolha dos grupos de comandos desta linguagem de programação, (B) uma área edição que possibilita a criação do projeto, ou a programação de eventos (ou "scripts"), (C) uma área de definição dos objetos (ou "sprites") e cenários (ou "palcos") que integram um dado projeto, (D) uma área que lista miniaturas dos "sprites" utilizados no projeto, (E) uma área de apresentação, que viabiliza a execução do projeto criado.

Assim a criação de um projeto no Scratch requer a escolha de comandos da linguagem de programação, a edição de um projeto que envolve a programação utilizando elementos gráficos para compor o "palco" da história, a definição de scripts ou "rotinas de ações" a partir do uso de comandos, especificação de parâmetros, sprites (objetos), trajes e sons.

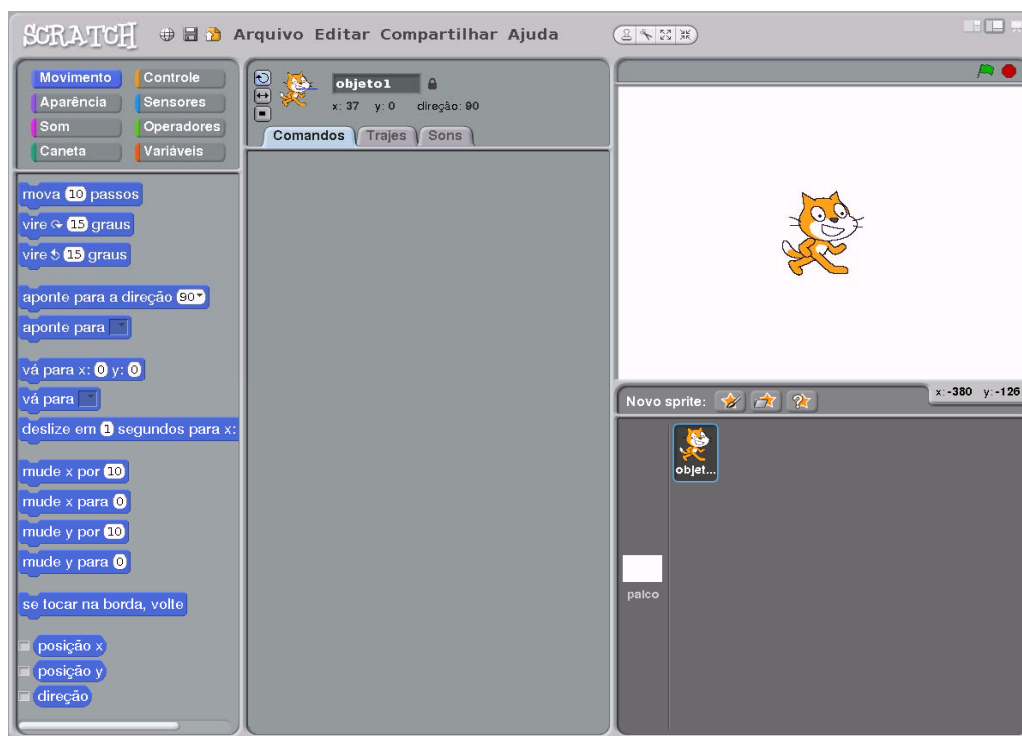


Figura 1: Tela Inicial do Scratch

## Nome dos campos da Tela Inicial

Vamos conhecer agora os nomes dos campos, abas, botões da Tela Inicial para facilitar o uso deste aplicativo. Nesta figura inicial aparece um objeto ("sprite") com a imagem do gato que pode ser movimentado na tela a partir da utilização de alguns comandos desta linguagem de programação.

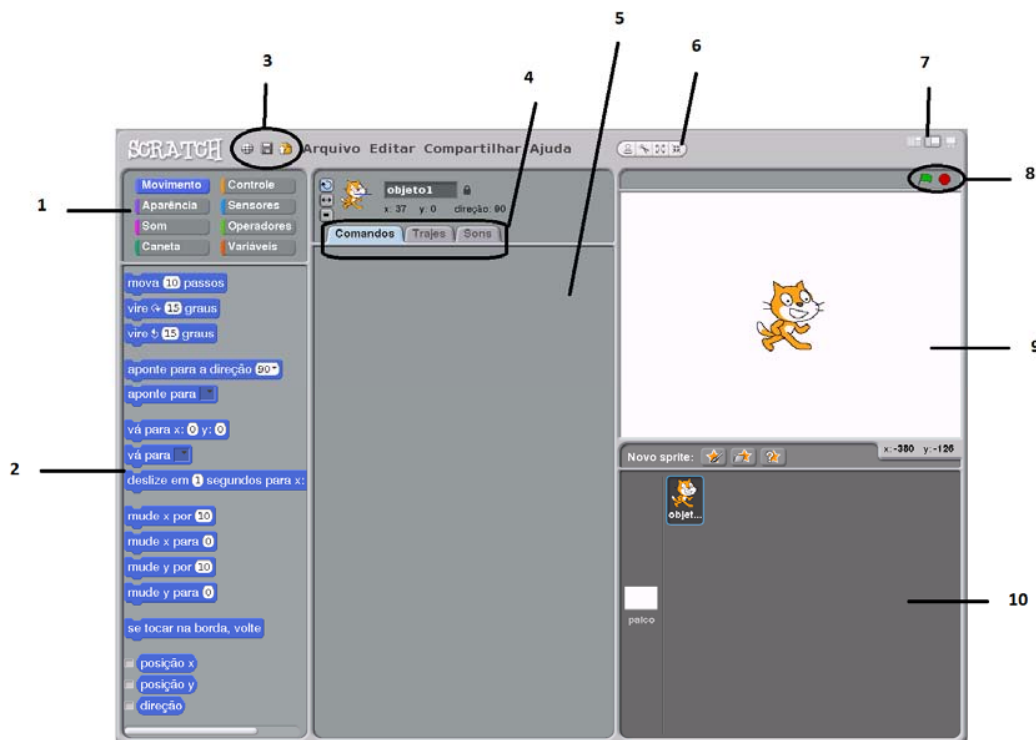


Figura 2: Tela Inicial do Scratch: áreas, botões, abas e elementos gráficos

- 1) Categoria de comandos: Movimento, Aparência, Som, Caneta, Controle, Sensores, Operadores, Variáveis;
- 2) Bloco de comandos: ao escolher uma categoria de comando temos várias opções de comandos;
- 3) Botões de atalho: selecionar idioma, salvar este projeto e compartilhar este projeto;
- 4) Abas para opções comandos, trajes e sons;
- 5) Área de edição (depende da aba selecionada – item anterior (4));
- 6) Botões para editar o objeto no palco: Duplicar, Apagar, Crescer objeto e Encolher objeto.
- 7) Botão para alterar o tamanho de visualização do palco;
- 8) Botões para iniciar (bandeira verde) e parar o script (círculo vermelho);
- 9) Palco: local onde visualizamos o Sprite ;
- 10) Área que disponibiliza a seleção/criação dos Sprites e do palco.

## 2. Primeiro Movimento no Scratch

Agora vamos iniciar um projeto que apenas desloca o Gato ("sprite" na tela ("palco").

Ao abrir o Scratch já aparece selecionada na opção **Movimento** no campo (1) – Categoria de comandos.

Utilizando o cursor do mouse arraste um dos comandos que está no campo (2) - Bloco de comandos para a área de Edição (campo 5). Clique e segure com o botão esquerdo do mouse no comando **mova 10 passos**. Arraste para o Campo (5) – Área de Edição. Solte o botão direito. Você obterá a seguinte imagem:

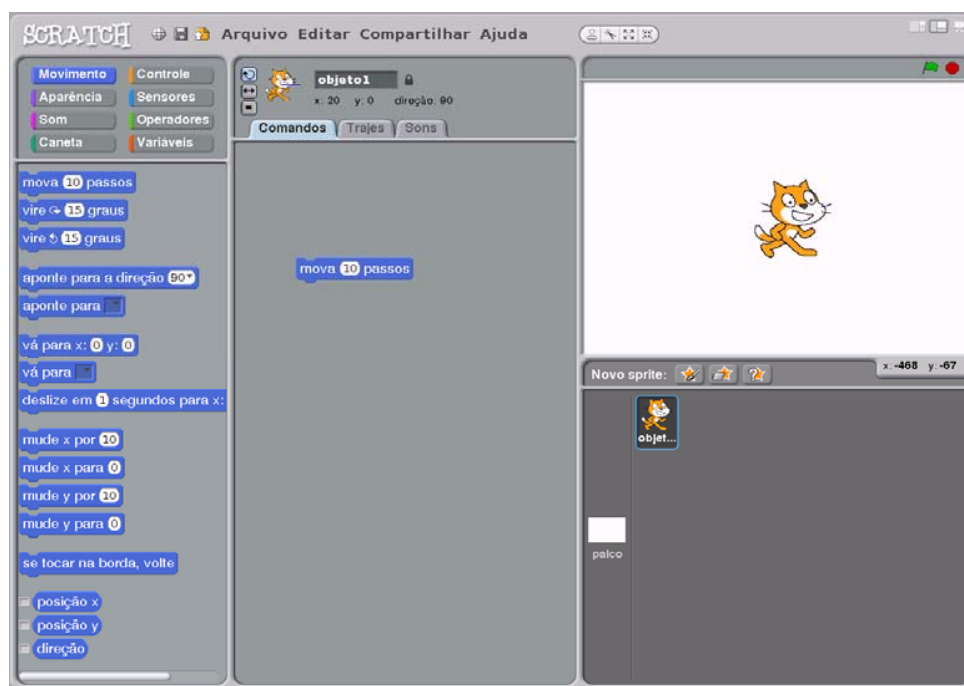


Figura 3: Uso do comando mova e número 10 para deslocamento do objeto na tela

Para executar este comando clique duas vezes com o botão esquerdo do mouse sob o botão que você acabou de arrastar. Veja que o Objeto (Gato) se mover para direita no campo (9) – Palco. Neste exemplo estamos programando o deslocamento do Objeto

(Gato) 10 passos para direita, ou seja, o parâmetro utilizado no comando “mova” foi o número 10.

Vamos agora experimentar outros números como parâmetro do comando “mova”!!!! Clique apenas uma vez com o botão esquerdo do mouse sob o número 10 e altere o valor do parâmetro de entrada do comando “mova”.

Altere o valor para 40, aperte ENTER ou clique duas vezes com o botão esquerdo do mouse sob o comando “MOVA”.

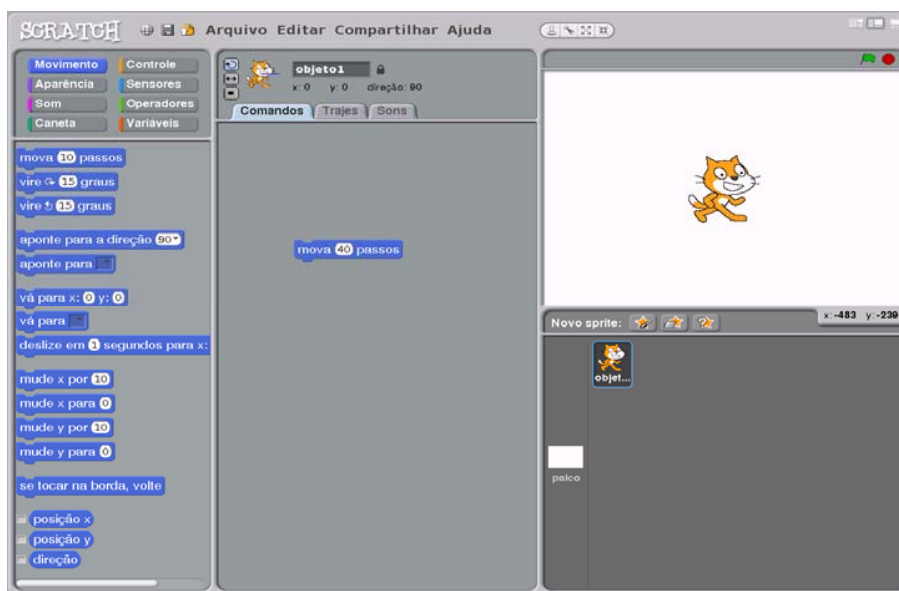


Figura 4: Uso do comando mova e número 10 para deslocamento do objeto na tela

Observe que desta vez o Objeto (Gato) caminhará 40 passos para a direita.

## Situando movimentos de objetos no Plano cartesiano

Ao abrir o Scratch, na tela inicial não existem comandos na Área de Edição (campo 5) e o Objeto (Gato) está localizado no Palco (campo 9) que se encontra na origem do plano cartesiano.

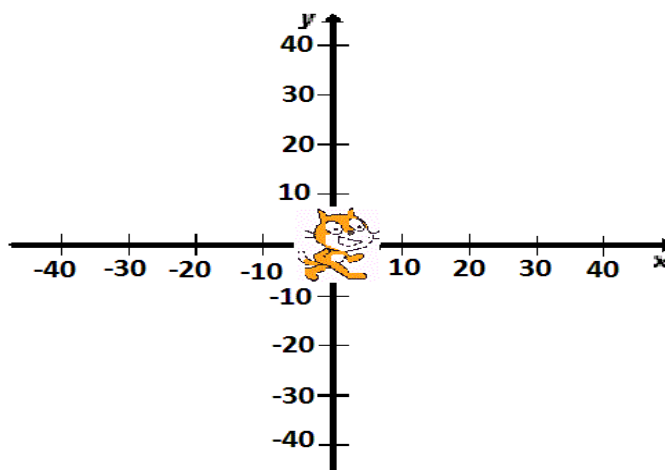


Figura 5: Coordenadas Cartesianas – pontos [0,0] dos Eixos XY da área "Palco" do Scratch

No primeiro exemplo quando foi utilizado o parâmetro 10 no comando "mova" o Objeto (Gato) foi movimentado no eixo das abscissas. Ou seja, o Objeto (Gato) foi deslocado da posição 0 para a posição 10 na coordenada X. Já com o parâmetro 40 no comando mova o Objeto (Gato) foi movimentado 40 passos para a direita ou 40 unidades de medida no eixo das abscissas(eixo x)

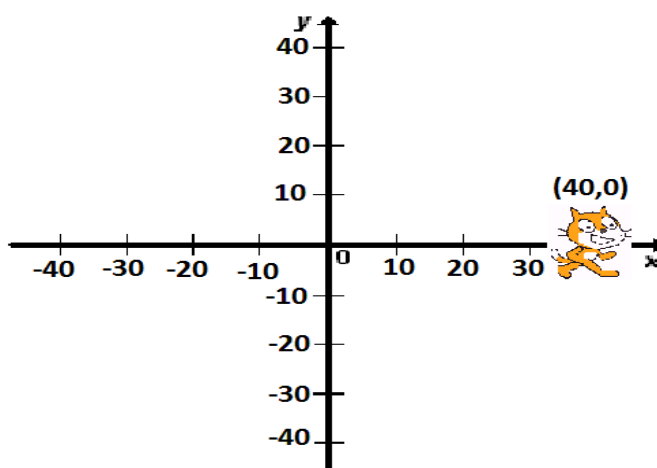
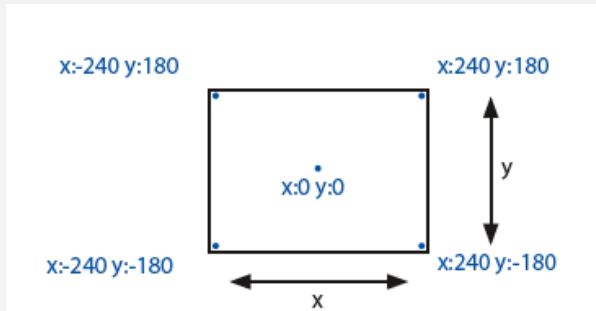


Figura 6: Coordenadas Cartesianas – pontos [40,0] dos Eixos XY da área "Palco" do Scratch

O Palco é onde ocorre a execução de eventos que podem programar a movimentação de objetos na tela. Tem 480 unidades de largura por 360 de altura e é associado a um sistema de eixos coordenados cartesianos x-y .



Para saber a coordenada de um ponto na tela coloque o mouse sobre esse ponto e olhe a mensagem à direita e abaixo do palco.

mouse x: -353  
 mouse y: -122


Considerando que o Objeto (Gato) está inserido no plano cartesiano podemos dizer o Gato realizou um movimento linear, com  $y = 0$ ,  $x$  variando de  $x=0$  até  $x=10$  e também  $x$  variando de  $x=10$  até  $x=40$ .

Agora experimente colocar parâmetros negativos no comando “mova”. Veja o que acontecer com o Objeto (Gato).








<b>mova 10 passos</b>	Movimenta o sprite para frente ou para trás(nº positivo ou negativo)
-----------------------	--

Experimente também outros comandos da categoria “movimento” que consideram as coordenadas cartesianas para referenciar a posição e o deslocamento de objetos na tela.

<b>mude x por 10</b>	Muda a abscissa usando o incremento especificado
<b>mude x para 0</b>	Leva o sprite para a abscissa especificada
<b>mude y por 10</b>	Muda a ordenada y usando o incremento especificado
<b>mude y para 0</b>	Leva o sprite para a ordenada especificada
<b>vá para x: 0 y: 0</b>	Move o sprite para a posição x e y especificada (posição no Palco )
<b>deslize em 1 segundos para x: 0 y: 0</b>	Leva p/posição especificada no tempo determinado
<b>posição x</b>	Reporta a abscissa do sprite (-240 a 240)




 posição y	Reporta a ordenada do sprite (-180 a 180)
---	---

Outros comandos da categoria “movimento” permitem GIRAR os objetos na tela a partir da especificação de parâmetros que variam de 0 a 360 graus, sentido horário e anti-horário. Experimente!!!!

	Gira o sprite no sentido horário
	Gira o sprite no sentido anti-horário
	Aponta o sprite para uma direção especificada 0=p/cima, 90=direita, 180=p/baixo, -90=esquerda
	Reporta a direção do sprite
	Muda a direção do sprite em 180º qdo tocar na borda
	Movimenta o sprite para a posição do cursor do mouse ou de outro sprite
	Aponta o sprite para o mouse ou para outro sprite

### 3. Salvando o Projeto

Após executar e combinar vários tipos de comandos você pode salvar o seu projeto.

Na parte superior da tela - Campo 3 há os seguinte Botões de atalho:    .

Para saber a função/nome dos botões passe o cursor do mouse sobre as figuras e veja o contexto.



: seleciona idioma. Permite escolher qual idioma queremos trabalhar no Scratch



: Compartilhar este projeto.



: Salvar este projeto. Permite salvar o projeto no próprio diário, no computador ou em um pendrive.

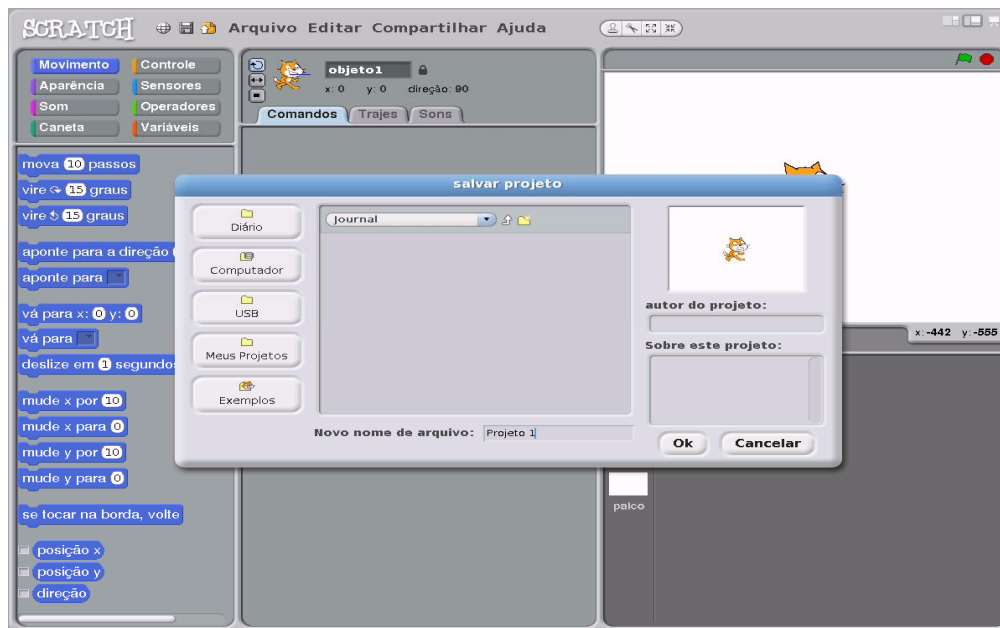


Figura 8: Tela "Salvar projeto"

Experimente utilizar a opção "Diário". Aparecerá um campo para preencher o "Novo nome de arquivo" que no nosso exemplo será "Projeto 1".

#### 4. Abrir o projeto

A opção "abrir" projetos apresenta 2 possibilidades: 1º) Abrir projeto salvo; 2º) Abrir novo projeto.

Entendendo o 1º caso - Abrir projeto salvo:

Na tela inicial do XO ao passar o cursor sobre o ícone do Scratch podemos ver o nome do último projeto salvo e clicando com o botão esquerdo do mouse já podemos acessar o projeto. Veja imagem:



Figura 9: Acesso aos Projetos do Scratch no laptop XO

Entendendo o 2º caso - Abrir novo projeto:

Na Figura 09 acima, podemos colocar o mouse sobre o retângulo preto e aguarde, veja que irá aparecer várias opções para acessar o Scratch



Figura 10 Acesso aos Projetos do Scratch no laptop XO

A Figura 10 mostra que podemos acessar todos os projetos salvos anteriormente (Projeto 1 e Projeto 2) bem como iniciar um novo projeto (Iniciar).

## 5. ANEXO I - COMPONENTES BÁSICOS DE UM PROJETO SCRATCH

### BANDEIRA VERDE



Para iniciar qualquer script referenciado por este bloco clique na bandeira verde. No modo de Apresentação a Bandeira Verde aparece como um ícone brilhante no canto superior-direito da tela, logo acima do Palco. Atalho: pressionar a tecla Enter produz o mesmo resultado que clicar na bandeira verde.

### MODO DE APRESENTAÇÃO



Botão **Modo de Apresentação** Para ver os projetos ocupando toda a tela, clique neste botão. Para sair do Modo de Apresentação aperte a tecla ESC.



Clique nestes botões para passar de palco miniatura para normal (cheio) e vice-versa

### BARRA DE FERRAMENTAS



Clique na barra para selecionar um instrumento e depois nos objetos para ser executada a ação



**Flecha:** Pinça e move sprites e blocos.

**Copiar:** Duplica sprites, trajes, sons, blocos e scripts. (Shift + clique)

**Deletar:** Deleta sprites, trajes, sons, blocos e scripts. (Shift + clique)

**Aumentar:** Torna o sprite maior. (Shift + clique )

**Diminuir:** Torna o sprite menor. (Shift + clique )

**SPRITES:** Os projetos Scratch são formados por objetos chamados **sprites**.




A aparência de um sprite pode ser modificada pela sua apresentação em um traje diferente. Qualquer imagem pode ser usada como um traje; pode-se desenhar uma imagem no editor de Pintura, importar de uma lista ou baixá-la de um website.

Você pode dar instruções a um Sprite, mandar que ele se mova, que toque música, reaja a outros sprites, etc.

Para determinar o que cada sprite deve executar, você deve criar para cada um, sequências de comandos, arrastando blocos gráficos e juntando-os em pilhas chamadas **script**. Basta um duplo clique em qualquer bloco para fazer rodar um script. O Scratch executa esses scripts a partir do topo e depois, sucessivamente, bloco a bloco, os de baixo.

### BOTÕES - NOVO SPRITE

Um novo projeto começa com um único sprite do gato. Para criar novos sprites clica-se em um dos seguintes botões, conforme o que se deseje:

-  Desenhar novo sprite
-  Escolher sprite na biblioteca de imagens
-  Ganhar um sprite surpresa

A **LISTA DE SPRITES** contém miniaturas de todos os sprites do projeto, mostrando o nome, quantos trajes e quantos scripts cada um possui. Para ver e editar o script de um sprite clicar na miniatura na Lista de Sprites ou dar um duplo-clique no sprite na tela. O Palco pode mudar de aparência assumindo outros fundos de tela. Para ver e editar scripts, fundos de tela e sons associados ao Palco, clique no ícone do Palco (retângulo), à esquerda da Lista de Sprites.



Para mostrar um sprite que não esteja no Palco, pressione Shift , clique na miniatura na Lista de sprites, e o sprite aparece no Palco.

### APARÊNCIA ATUAL DO SPRITE



Aqui aparecem as seguintes características do sprite no momento: nome, abscissa e ordenada, direção e o estado de bloqueio. Para dar um

novo nome basta digitá-lo no retângulo escuro. A direção indica para onde ele se moverá quando executar um comando de deslocamento. ( 0=para cima; 90=direita; 180=para baixo;

-90=esquerda). A linha azul na miniatura também mostra a direção do sprite. Arrastando essa linha com o mouse, a direção do sprite, no Palco, será mudada; um clique duplo na miniatura devolverá a direção original (90). Clicando no cadeado modifica-se sua situação de bloqueio. Se o sprite estiver desbloqueado poderá ser arrastado no Modo de Apresentação. Ao lado do cadeado aparece a cor da caneta que estiver sendo usada

Pode-se exportar um sprite clicando com o botão direito do mouse no sprite do Palco ou da Lista de sprites. Será guardado como um arquivo do tipo "sprite" e poderá ser importado em outro projeto..

## ESTILO DE ROTAÇÃO

Estes três botões controlam o giro do traje quando o sprite muda de direção



Rotação: o traje gira à medida que o sprite muda de direção



Esquerda-Direita: o traje se movimenta apenas para a direita ou para a esquerda



Sem Rotação: o traje não gira (nem se o sprite mudar de direção)

**EDITOR de PINTURA** : usado para criar ou editar trajes e fundos de palco. Tem as ferramentas:

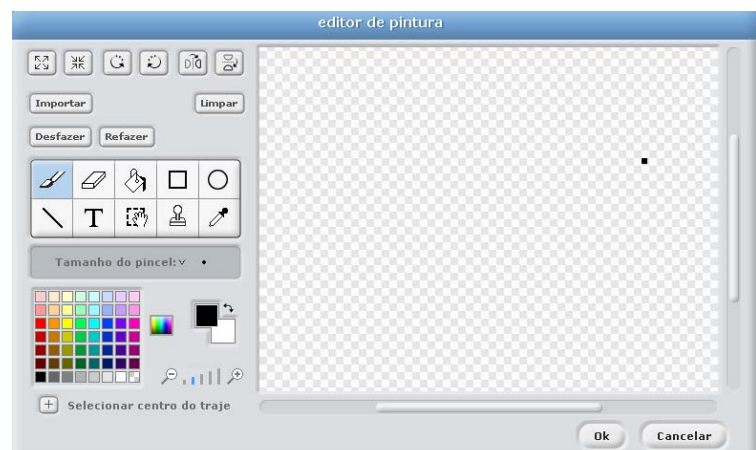
**Pincel** : pinta usando uma cor selecionada na paleta de cores. Ao clicar na ferramenta a área de Opções mostra o tamanho do pincel. Clicando em  escolhe-se um tamanho diferente.

**Borracha**: apaga desenho, tornando transparentes as áreas apagadas. Ao clicar nesta ferramenta a área de opções mostra o tamanho da borracha.  permite modificar o tamanho.

**Balde de Tinta**: preenche áreas fechadas com uma única cor ou com um gradiente de cores. Ao clicar na ferramenta aparecerá na área de opções o estilo de preenchimento (cor única, gradiente horizontal, gradiente vertical, gradiente radial)

Barra de Ferramentas →

Área de Opções →



## Paleta de Cores

**Retângulo:** desenha um retângulo . Para obter quadrado basta pressionar Shift e arrastar o cursor. Ao clicar na ferramenta aparece na área de Opções o estilo a ser usado (cheio ou contorno)

**Elipse:** desenha uma elipse. Para circunferência basta pressionar Shift e arrastar o cursor.

**Linha:** desenha linhas; para verticais ou horizontais basta pressionar Shift e arrastar o cursor.



**Texto:** para colocar um texto no desenho. Na área de opções escolhe-se o tipo e tamanho da fonte. Cada traje pode ter um único bloco de texto.


**Seleção:** permite selecionar uma região retangular e depois deslocá-la para uma nova posição. (o botão delete do teclado apaga o que estiver dentro da seleção e Shift + Del ou Shift + Backspace apaga o que estiver fora da área selecionada).


**Carimbo:** Seleciona uma região retangular e depois a copia para novas localizações.


**Conta Gotas:** para escolher uma cor de um desenho no quadriculado ou na paleta de cores. As cores (ativa e de fundo) são mostradas abaixo da área de Opções.


O botão "Selecionar centro do traje" localiza no desenho o centro de rotação quando esse traje for movimentado no Palco.


As lupas  e  servem para aumentar ou diminuir a escala do desenho enquanto for trabalhado no quadriculado. O zoom não altera a dimensão final da imagem.


 Botão **Importar:** traz uma imagem para ser colocada na tela de edição.

 Botão **Limpar:** para apagar a imagem que estava sendo editada.

 Botão **Aumentar e Diminuir** : para redimensionar o tamanho da imagem toda ou apenas de uma região selecionada. Shift + clique permite indicar a % desejada.

 Botões de rotação: para girar todo o desenho ou apenas uma região selecionada na tela de edição. Shift + clique permite indicar o ângulo exato desejado.

 Para obter uma figura simétrica vertical

 Para obter uma figura simétrica horizontal

## Programação de Sprite e área de Scripts

Para programar um sprite, arraste os blocos de comando da Paleta de Blocos até a Área de Projetos. Para rodar um bloco isolado basta clicar sobre ele.



Os scripts (programas) são criados por pilhas de blocos reunidos. Um clique em qualquer ponto do script (a pilha) determina sua execução, do topo até o último bloco.

Para descobrir o que faz cada bloco basta clicar sobre ele com o botão direito do mouse e selecionar Ajuda no menu que aparece.

Para mover uma pilha basta arrastar o bloco de cima. Arrastando um bloco do meio todos os de baixo são arrastados juntos.

Para copiar uma pilha de blocos, de um sprite para outro, basta arrastá-la para a miniatura do outro sprite na Lista de Sprites ( bastidores).

Alguns blocos apresentam uma região branca na qual podem ser colocados valores

 . Blocos arredondados como  podem ser colocados nessas áreas.

Blocos como  ,  , tem menus pull-down .

Com um clique aparece o menu e com outro é feita a seleção.

Se clicar com o botão direito do mouse na Área de Scripts aparecerá um menu do tipo:

limpe  
salvar uma imagem dos comandos  
adiciona comentário

Conforme a escolha, a Área de Scripts será limpa, será salva uma imagem dos comandos ou será aberta uma área

amarela para colocação de comentários,

adicionar  
comentário  
aqui...


Comentários podem ser postos em qualquer lugar da Janela de Projetos e arrastados até o topo do bloco pertinente.

## TRAJES

Clicando no separador **Trajes** aparecem os trajes escolhidos para os sprites e que podem ser editados. Este sprite tem 2 trajes. O que está contornado em azul é o usado no momento.

Há quatro maneiras para criação de novos trajes:



- Clicar em **Pintar** para pintar um novo traje no Editor de Pintura;
- Clicar em **Importar** para importar uma imagem do disco rígido;
- Clicar em **Câmera** para tirar fotos com uma webcam conectada a seu computador. Cada vez que for pressionada a barra de espaços ou clicado o botão  será tirada uma foto.
- Arrastando uma ou mais imagens de um site ou da **Área de Trabalho**.

O Scratch reconhece os formatos de imagem: JPG, BMP, PNG, GIF (animado também)  
A ordem dos trajes pode ser modificada arrastando-se as miniaturas. Clicar com o botão direito do mouse em uma miniatura de traje o transforma em um novo sprite.

## SONS






Clicando no separador Sons aparecem os sons que estão sendo usados.



Podem ser gravados novos sons ou importados de arquivos de som. Scratch pode ler arquivos MP3, WAV não zipados, AIF e AU (desde que não sejam de 24 bits).













## ANEXO II: DESCRIÇÃO DOS BLOCOS DE COMANDOS DO SCRATCH

Os blocos de comandos do Scratch estão dispostos em oito categorias (Movimento, Aparência, Som, Caneta, Controle, Sensores, Operadores, Variáveis), cada uma com sua cor específica.

<b>Movimento</b>	<b>Blocos De Motor</b>	Só aparecem na escolha de <b>Exibir Blocos de Motor</b> no Menu <b>Editar</b> . <b>Ou se conectar com um motor Lego WeDo</b>
		Faz o motor funcionar pelo tempo designado
		Faz o motor funcionar
		Interrompe o funcionamento do motor
		Estabelece a potência indicada e liga ( varia de 0 a 100)
		Muda o sentido mas não liga. (para cá = sentido horário; para lá = sentido anti-horário; reverso = inverte o sentido )

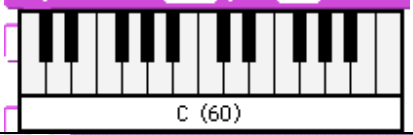
Os comandos Bloco de Motor funcionam com um motor LEGO WeDo (Acesse <http://www.legoeducation.com>)

### Aparência


	Muda a aparência do sprite assumindo o traje especificado
	Mostra o valor do traje em uso
	Muda para o traje seguinte da lista de trajes
	Mostra o especificado, em um balão de conversa, pelo tempo especificado
	Mostra algo especificado em um balão de conversa
	Mostra o escrito, em um balão de pensar, pelo tempo especificado
	Mostra o especificado, em um balão de pensamento
	Muda o efeito marcado usando o incremento especificado
	Muda o efeito especificado para o marcado na 2ª janelinha
	Limpa todos os efeitos gráficos dos sprites (ou do palco)
	Dá ao tamanho um incremento de 10
	Dá ao tamanho do sprite a % especificada de seu tamanho original

<input type="checkbox"/> tamanho	Mostra o valor do tamanho do sprite (em % do tamanho original)
apareça	Faz o sprite aparecer no palco
desapareça	Faz o sprite desaparecer (não será detectado por outro sprite)
vá para a camada de cima	Põe o sprite na frente de outros
desça 1 camadas	Muda para trás de outros sprites o nº de camadas especificado
Abaixo estão blocos específicos que só aparecem quando o Palco é selecionado	
mude para o fundo de tela fundo de tela1	Muda para o fundo de tela cujo nome for especificado
próximo fundo de tela	Muda para o cenário seguinte
fundo de tela #	Indica o número do <b>fundo de tela</b> , conforme a lista de cenários



## Som








toque o som pop	Toca o som selecionado do menu e depois vai para o próximo bloco (mesmo que o som ainda esteja tocando)
pare todos os sons	Pára a execução de todos os sons
pare por 0.2 batidas	Para de tocar pelo tempo estipulado
toque o tambor 48 por 0.2 batidas	Toca o som do tambor(selecionado do menu) durante o tempo determinado
toque a nota 60 por 0.5 batidas	Toca a nota (selecionada do menu) durante o tempo determinado.
toque a nota 60 por 0.5 batidas 	Ao ser clicado o espaço reservado ao nº da nota aparece um teclado que, quando clicado, produzirá um som e mostrará o valor da nota correspondente.
toque o som miau até terminar	Toca o som selecionado até que termine e então vai p/outro bloco
mude o instrumento para 1	Muda para o instrumento escolhido
mude o volume por -10	Muda usando o incremento especificado
<input type="checkbox"/> volume	Mostra o volume no momento
mude o volume para 100 %	Muda na % especificada
mude o ritmo por 20	Muda usando o incremento especificado
mude o ritmo para 60 bpm	Muda o ritmo para o número estipulado de batidas por minuto
<input type="checkbox"/> ritmo	Mostra o ritmo usado no momento

## Caneta














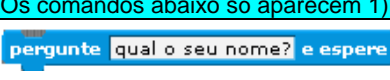



limpe	Limpa todos os traços e carimbos do Palco
abaixe a caneta	Abaixa a caneta do sprite, permitindo-o desenhar no Palco
levante a caneta	Tira a caneta do sprite, não permitindo que desene no Palco
mude a cor da caneta para 	Coloca a cor especificada no quadradinho
mude a cor da caneta por 10	Muda a cor da caneta pelo incremento especificado
mude a cor da caneta para 0	Muda a cor da caneta para o valor especificado
mude o tom da caneta por 10	Muda o tom da caneta pelo incremento especificado
mude o tom da caneta para 50	Muda o tom da caneta para o valor especificado
mude o tamanho da caneta por 1	Muda a espessura do traço da caneta pelo incremento especificado
mude o tamanho da caneta para 1	Muda a espessura do traço da caneta para o valor especificado
carimbe	Carimba a imagem do sprite no Palco

## Controle

quando  clicado	Roda o script debaixo quando a bandeira verde for clicada
quando tecla <input type="text" value="space"/> pressionada	Executa o script debaixo quando a tecla especificada for pressionada
quando objeto1 clicado	Roda o script debaixo quando o prite especificado for clicado no Palco
espere 1 segundos	Aguarda o tempo especificado e passa para o bloco seguinte
sempre	Executa ininterruptamente o conjunto de blocos abrangidos em seu interior ( só pára se for clicado o  ou se houver algum comando específico para parada)
repita 10	Repete, o nº de vezes especificado, o conjunto de blocos abrangidos
anuncie <input type="text" value=""/> para todos	Avisa a todos os sprites o que deve ser feito
anuncie <input type="text" value=""/> para todos e espere <input type="text" value=""/>	Alerta todos os sprites para a execução de uma ordem, só passando para outro bloco quando todos os sprites tiverem executado essa ordem.
quando eu ouvir <input type="text" value=""/>	Espera pelo recebimento de uma mensagem para só então executar os comandos do bloco que encima

	Enquanto a condição dada for satisfeita continua executa o script interno
	Sómente executará o script interno se a condição colocada estiver satisfeita
	Se a condição dada for satisfeita executa o script do laço <b>se</b> ; em caso contrário executará o script do laço <b>senão</b>
	Espera até que a condição seja verdadeira para executar o bloco seguinte
	Repete a execução do bloco interno até que a condição especificada esteja satisfeita
	Pára a execução do script em que estiver inserido
	Pára a execução de todos os scripts de todos os sprites

## Sensores




	Indica a abscissa do cursor do mouse
	Indica a ordenada do cursor do mouse
	"verdadeiro se o botão do mouse estiver pressionado e "falso se não estiver
	Relata "verdadeiro se a tecla especificada estiver pressionada e "falso se não estiver
	Relata "verdadeiro se o sprite está tocando em 1 sprite especificado e "falso se não estiver
	Relata "verdadeiro se o sprite está tocando na cor mostrada no quadradinho e "falso se não estiver
	Relata "verdadeiro se a 1ª cor(a do sprite)tocar a 2ª(do fundo ou de outro sprite) e "falso se não estiver
	Devolve a distância do sprite até outro sprite ou o cursor do mouse
	Dá o valor de uma característica do sprite especificado (posição, neste caso)
	Faz iniciar a contagem de tempo
	Relata o valor do tempo (em segundos)
	Relata o valor do volume do som detectado pelo microfone( de 1 a 100)
	Relata "verdadeiro se detectar um volume superior a 30; "falso se não
<b>Os comandos abaixo só aparecem 1) no caso de uma pergunta 2) se estiver conectado um Scratch Board</b>	
	Apresenta uma pergunta na tela e espera pela resposta digitada no teclado. Se continua o programa se for pressionado o <b>Enter</b> ou se for clicado o sinal de checagem
	Devolve o valor entrado pelo teclado no caso do comando acima (pergunta)
	Relata o valor do sensor especificado (é necessário estar com o Scratch board conectado)
	Relata "verdadeiro se um determinado sensor está pressionado e "falso se não estiver

## Operadores

	Soma os 2 números escritos
	Subtrai o 2º número do 1º
	Multiplica os 2 números escritos
	Retorna o quociente(c/2 casas) do 1º pelo 2º número
	Sorteia um nº entre os extremos especificados
	Verifica se o 1º nº é menor que o 2º
	Verifica se o 1º nº é igual ao 2º
	Verifica se o 1º nº é maior que o 2º
	Verifica se as 2 condições dadas estão satisfeitas
	Verifica se pelo menos uma das cond está satisfeita
	"verdadeiro qdo a condição não for satisfeita
	Concatena 2 elementos no formato frase
	Dá a letra da posição pedida da palavra especificada
	Devolve o número de letras da palavra especificada
	Devolve o resto da divisão do 1º pelo 2º número
	Dá o resultado da função selecionada(abs, sen,log...)
	Arredonda o nº dado para o inteiro mais próximo

## Variáveis



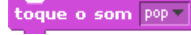


	Permite criar e nomear uma variável; com a variável são criados automaticamente os 6 blocos abaixo
	Deleta os 5 blocos associados à variável
	Relata o valor atual da variável <b>var</b> (ou do nome que estiver especificado)
	Dá à variável indicada o incremento/decremento especificado
	Muda o valor da variável indicada para o valor especificado
	Mostra, no Palco, o monitor da variável indicada
	Retira, do Palco, o monitor da variável indicada
	Permite criar e nomear uma lista; com a lista são criados automaticamente os 9 blocos abaixo
	Deleta os 8 blocos associados à lista
	Relata o valor atual da lista <b>lis</b> (ou com o nome que estiver especificado)
	Coloca o item digitado na lista especificada
	Retira, da lista, o elemento cuja posição estiver indicada
	Insere o elemento escrito na posição indicada da lista especificada
	Substitui o elemento da posição indicada pelo item especificado

	Dá o valor do item especificado
	Dá o número de elementos da lista
	"verdadeiro" se a lista contiver o elemento especificado e falso se não contiver








Lembrar que os blocos de cantos pontiagudos contêm comandos booleanos, isto é, que assumem o valor **verdadeiro** ou o valor **falso**.


## BLOCOS DO SCRATCH : TIPOS

Nas oito categorias existentes na Paleta de Comandos há três tipos principais de blocos de comando:

- **Blocos Empilháveis.(Simples)**. Estes blocos têm uma cavidade na parte de cima e uma saliência na base, tal como: . Podem ser conectados e reunidos em pilhas. Alguns blocos têm uma área de **entrada** onde pode ser digitado um número (como ) , ou escolhido um item de um menu pull-down  (tal como "pop"). Outros têm a forma de "boca" em C, como este,  permitindo a inserção de outro Bloco de comando.
- **Chapéus**. Estes blocos têm o topo arredondado como 

São colocados no topo das pilhas; esperam que ocorra algum evento (como, por ex, uma tecla pressionada ou o clicar na bandeira verde), para fazer rodar os outros blocos da pilha

- **Repórteres/Valores**. Estes blocos, tal como  e  , são feitos para preencher a área de input de outros blocos. Podem ter 2 formatos e apenas se encaixam nas cavidades de mesmo formato. Reporteres com extremidades arredondadas (tal como ) devolvem valores numéricos e encaixam-se nos blocos que também possuem partes arredondadas dentro (tal como ). Repórteres com extremidades pontudas (como ) devolvem valores "booleanos" (verdadeiro ou falso) e cabem nos que têm, dentro, reentrâncias com extremidades em triângulo (tal como ) ou retangulares (como ).

Alguns blocos-reporter têm uma caixa de checagem (ex: ). Se v. clicar nessa caixa, aparecerá no palco um **monitor**, mostrando o valor do reporter na ocasião e atualizando esse valor, à medida que for mudando. Um monitor pode mostrar o valor em 3 formatos: uma pequena



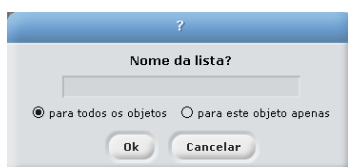
janela com o nome do repórter uma grande janela sem qualquer nome um slider que permite a manipulação do valor do repórter e que só está disponível para variáveis criadas pelo usuário.

Um clique duplo no slider (ou do lado direito do mouse), ajusta os limites máximo e mínimo da variável.

## LISTAS

No Scratch podemos criar e manipular listas formadas por números, letras, frases. Ou outros caracteres.

Para criar uma lista clicar em **Variáveis** e depois em **Cria uma lista**.



Aparece uma janela onde deve ser colocado o nome da lista.

Imediatamente aparecerão diversos blocos na Paleta de blocos e no Palco fica visível a lista com todos seus itens.



Esses itens podem ser editados diretamente no Palco. Para incluir itens basta clicar no botão + no extremo inferior esquerdo do mostrador e preencher os retângulos numerados que aparecem.

Também podem ser usados os blocos tipo



Botão +

Um clique com o botão direito do mouse acessa a opção de exportar a lista para um arquivo.TXT que será criado diretamente na pasta de instalação do Scratch, para importar uma lista salva (com todos seus valores) ou para fazer desaparecer.

Podem ser formados conjuntos compostos de letras, palavras ou outros caracteres (ex: eleição; outubro de 2010; Já acabou!) Esses conjuntos podem ser estocados em variáveis

ou listas ( como **mude resultado para 20** ou **adiciona triangulo a polígonos** ). Esses

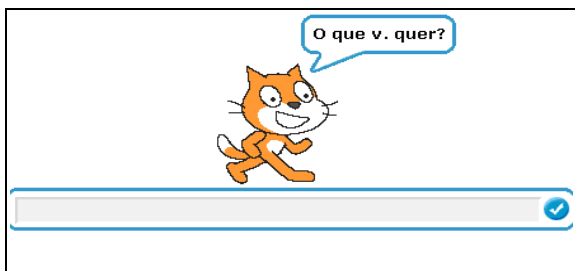
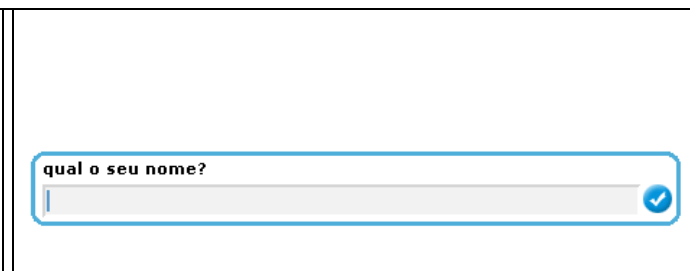
conjuntos podem ser reunidos com blocos do tipo **junte** e comparados com blocos de comandos como **=**, **<**, ou **>**. Nos blocos de operações

matemáticas ( como **+** ) ou nos que aguardam números ( como **mova 10 passos** ou

**mude o tamanho da caneta para 1** ), conjuntos de caracteres não numéricos são interpretados como sendo o valor 0 .

### Entradas do Teclado:

Pode ser usado o bloco **pergunte e espere** para que o usuário digite uma resposta com o teclado. A resposta será armazenada em **resposta**. A pergunta aparece na tela e o programa espera até que a tecla Enter seja pressionada ou que o sinal de checagem seja clicado.

	
A pergunta aparecerá em um balão de voz quando for usada para um sprite.	A pergunta aparecerá na parte de baixo da tela quando for usada no Palco.

**resposta** será compartilhada por todos os sprites e modificada cada vez que for executado um bloco **pergunte e espere**. Se quisermos salvar uma resposta, ela deverá ser armazenada em uma variável ou em uma lista.

## 6. Outros materiais

Outros materiais sobre o Scratch podem ser obtidos no próprio site do aplicativo:

- Material de apoio: <http://scratch.mit.edu>
- FAQs <http://info.scratch.mit.edu/Support>
- Getting Started Guide: <http://scratch.mit.edu/GettingStarted.pdf> ou <http://scratch.mit.edu/files/ScratchGettingStarted.pdf>.
- EduScratch: Projeto que visa promover a utilização educativa do Scratch através do apoio, formação e partilha de experiências na comunidade educativa portuguesa: <http://eduscratch.dgidc.min-edu.pt/index.php>
- Computação criativa: uma introdução ao pensamento computacional baseada no conceito de design: <http://projectos.ese.ips.pt/cctic/wp-content/uploads/2011/10/Guia-Curricular-ScratchMIT-EduScratchLPpdf.pdf>

Documento elaborado para o Projeto XO-UNICAMP

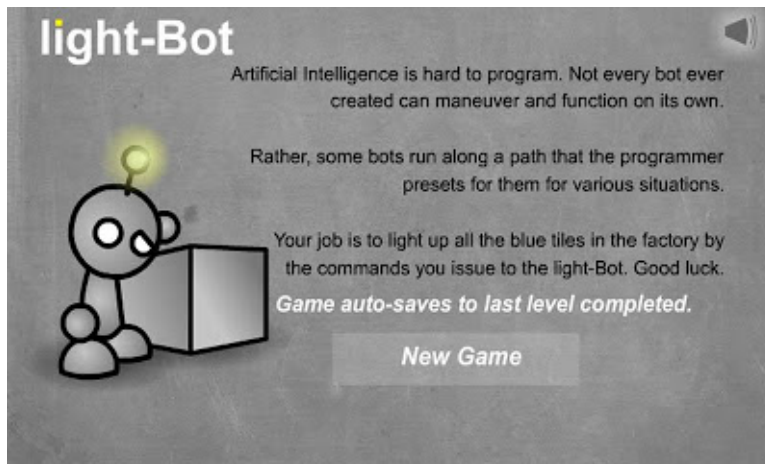
Coordenação: Maria Cecília Calini Baranauskas

Elaboração: Eduardo Mauricio Moreno Pinto e Maria Cecília Martins, Setembro 2011

Este tutorial adaptou informações traduzidas em Materiais de Apoio gerado por Maria Ercilia C. Rolim para o Curso Tecnologias Interativas Aplicadas à Educação, COGEAEPUC-SP (2009)



**ANEXO C – LIGHTBOT**

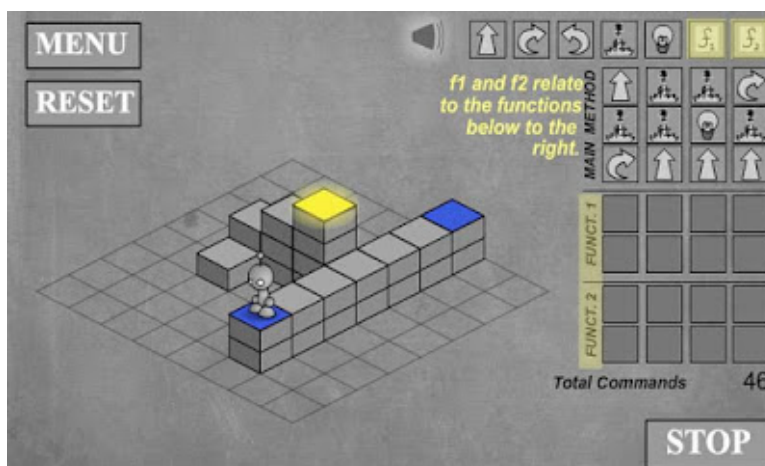


[<http://1.bp.blogspot.com/->

[CkaxehX40aE/UVZm2VvZkpl/AAAAAAAAAzo/6CTFW0JZdml/s1600/800250031d64a189f68aa88b6d0f4666.png.jpeg](http://1.bp.blogspot.com/-CkaxehX40aE/UVZm2VvZkpl/AAAAAAAAAzo/6CTFW0JZdml/s1600/800250031d64a189f68aa88b6d0f4666.png.jpeg)]

Light Bot 1.0 é o que podemos chamar de jogo pra nerd (mas que é apreciado por lerdos, geeks e afins também). Na categoria “jogos que parecem bobinhos, mas não são”, essa mistura de 'passatempo && uso de cachola' tem me conquistado nessas últimas horas. Quem é desenvolvedor ou programador pode se identificar muito com esse jogo.

Num resumo geral, é o seguinte: usando a lógica de um programador-estilo, deve ser passada uma sequência de comandos para fazer com que o robô ande, pule, vire e acenda sua lâmpada. Isso se consegue arrastando e encaixando ( nos quadrados vazios) os quadrados com os comandos e depois dando “permissão de execução” ao robô.



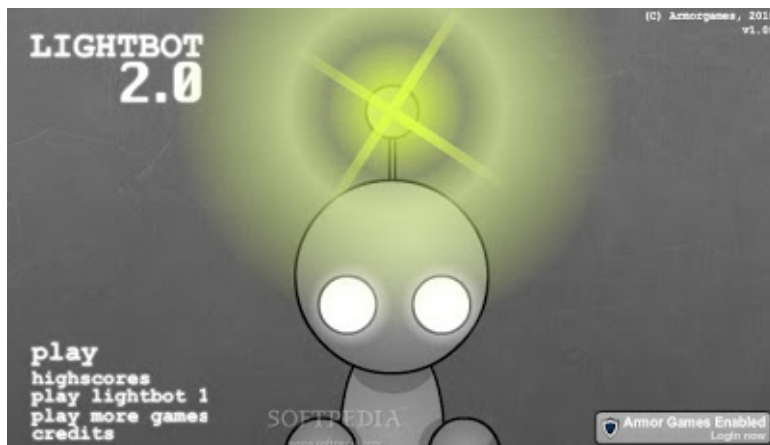
[<http://2.bp.blogspot.com/->

[Qgm0U8LF5XQ/UVZm2CJ36qI/AAAAAAAAAzW/4S4kiAnLGKA/s1600/1352518262\\_screen.jpg](http://2.bp.blogspot.com/-Qgm0U8LF5XQ/UVZm2CJ36qI/AAAAAAAAAzW/4S4kiAnLGKA/s1600/1352518262_screen.jpg)]

O jogador tem como objetivo acender todos os quadrados azuis para passar da fase em que se encontra. Pouco a pouco, conforme se passa de fase, os comandos vão ficando mais complexos e passam a incluir funções para reutilização ( $f1$  e  $f2$ ). As diretrizes devem ser precisas para desviar de obstáculos e acender a lâmpada na hora certa. As fases são de níveis diferentes e desafiantes.

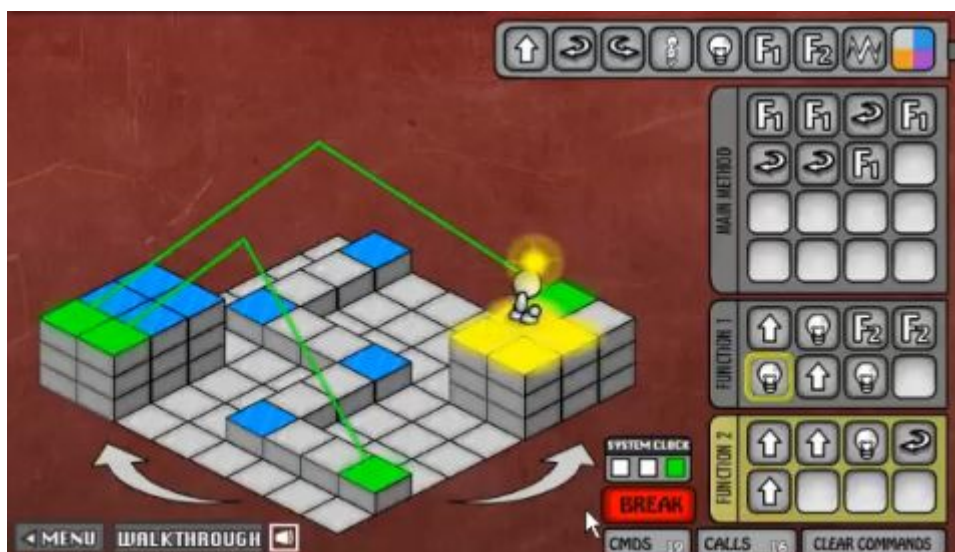
Antes que eu me esqueça: este jogo é on-line.

E muito importante: o jogador não deve esquecer de que existe um limite no número de ações que o robô pode realizar.



[[http://1.bp.blogspot.com/-nas2pojItYM/UVZm2zwWZyl/AAAAAAAAA0A/jHU-UCv0Rqw/s1600/Light-Bot-2-0\\_1.jpg](http://1.bp.blogspot.com/-nas2pojItYM/UVZm2zwWZyl/AAAAAAAAA0A/jHU-UCv0Rqw/s1600/Light-Bot-2-0_1.jpg)]

Na versão 2.0 o jogador continua tendo de direcionar um pequeno robô para acender todas as luzes do cenário. Claro que a nova versão exibe diversas novidades como poder ver os comandos que estão sendo executados (deve ser útil saber onde o robô se encontra no programa em tempo de execução) e o robô pode se teletransportar, porém eu não pude “brincar”, pois a única coisa que funcionava nos sites por onde passei procurando jogar essa nova versão era a de “user levels”, o que pra mim não serviu de nada, já que se tratava da “cola dos comandos” para resolver o jogo.



Então eu parei de procurar e voltei lá no 1.0 só pra matar algumas horas do dia de “folga” (e da leitura de artigos, por que não?).

Acesse o link: <http://www.newgrounds.com/portal/view/459508> [<http://www.newgrounds.com/portal/view/459508>] liste seus comandos e clique em “GO!” Para fazer o robô executar!

Postado há 30th March 2013 por [Narayana Ribeiro](#)

Marcadores: [Jogos](#), [Puzzles](#)

0 Adicionar um comentário

**ANEXO D – PROGRAMA E**

entrar  
comusuário  
ou  
senha

entrar

criar conta

Powered by  
OneAll Social Login[esqueci minha senha](#)[PARCEIROS](#)[CONTATO](#)[TRILHAS](#)[PLANOS DE AULA](#)

## Quer ajudar o Programaê! a ficar cada vez melhor?

Clicaê e vem bater um papo com a gente. ;)

ACREDITAMOS QUE A TECNOLOGIA TEM O PODER DE TRANSFORMAR E USÁ-LA PARA EDUCAÇÃO PODE FAZER A DIFERENÇA. EMBARQUE COM O PROGRAMAÊ! NESSA JORNADA PELO CONHECIMENTO! VOCÊ QUER APRENDER OU ENSINAR USANDO A TECNOLOGIA?

QUERO  
APRENDER!



QUERO  
ENSINAR!



OU, EXPERIMENTE UMA DE NOSSAS TRILHAS



Página na  
Internet



Programando  
com blocos



Para começar  
a aprender



Meu primeiro  
aplicativo

Desenhos e  
Animações

Frozen

+

VEJA MAIS

# Saiba +



**Bia Granja**, fundadora do **Youpix**, fala sobre como a programação pode ser a melhor resposta quando se trata de realizar grandes feitos a partir de ideias simples.



Durante o **Transformar** – evento que discutiu o futuro da educação –, conversamos com especialistas no assunto em um espaço cheio de ideias inspiradoras! Neste vídeo, você confere o resultado.



**Jessica Grecco e Ariane Freitas**, fundadoras do **Indiretas do Bem**, falam sobre como o site nasceu, quando a brincadeira virou trabalho e como a programação faz parte do dia a dia delas.



## POR QUE

A tecnologia tem um poder transformador incrível. Usá-la para a Educação pode fazer a diferença para muita gente.



## O QUE

Um movimento que quer aproximar a programação do cotidiano de jovens de todo o Brasil.



## COMO

Um portal prático e agregador de ideias, soluções e dicas de gente experiente e inspiradora.



ALGUNS PROJETOS  
SELECIONADOS PARA  
TE INSPIRAR

**Quando as linguagens de programação devem ser explicadas para alunos?**

*leia o artigo completo*

**A programação é uma linguagem inclusiva**

*leia o artigo completo*

**Sites e apps para inserir tecnologia em sala de aula**

*leia o artigo completo*

**Dicas do especialista: o que iniciantes em programação devem aprender primeiro?**

*leia o artigo completo*

**Games em sala de aula: é hora de aprender a programar**

*leia o artigo completo*

**Hora do Código: conheça o projeto que incentiva a programação!**

*leia o artigo completo*



VEJA MAIS

PARCEIROS

Olha só quem são nossos parceiros. Que tal aprender mais com os cursos que eles oferecem?



O PROGRAMAÊ! É UMA INICIATIVA



CONTATO

Tá com dúvidas? Mandaê que a gente responde!

Nome

assunto

e-mail

UF

telefone Cidade

mensagem

Quero receber notícias sobre o Programaê!

En'

[política de  
privacidade](#)



Programaê! está licenciado com uma Licença Creative Commons Atribuição-NãoComercial-Compartilhável 4.0 Internacional. Podem estar disponíveis autorizações adicionais às concedidas no âmbito desta licença em [www.fundacaolemann.org.br](http://www.fundacaolemann.org.br)

Desenvolvido com  pela FIB | Fábrica de Ideias Brasileiras

**ANEXO E – CS UNPLUGGED**

---

[Início](#) / [Computacional de Pensamento e CS Unplugged](#)

# Pensamento computacional e CS desconectado

## Sobre páginas

[Sobre o](#)

[pensamento computacional e CS Desconectado](#)

[Pessoas](#)

[Contacte-nos](#)

[Princípios](#)

## O que é o pensamento computacional?

O mundo em que vivemos tornou-se digital, cheio de tecnologia e conduzido pela Ciência da Computação. Software e tecnologia transformaram todo o assunto e área de trabalho, desde ciência e medicina até a história da arte e psicologia. A tecnologia digital é onipresente. Para serem cidadãos informados e capacitados, a próxima geração de estudantes precisa entender esse mundo digital em que vivem.

É por isso que o Pensamento computacional foi chamado de 'Conjunto de habilidade do século 21', e é importante para todos aprenderem. É fundamental para entender como o mundo digital funciona, para aproveitar o poder dos computadores para resolver problemas difíceis e fazer grandes coisas acontecer! Isso também nos permite pensar criticamente sobre não apenas os benefícios de certas tecnologias, mas também os possíveis danos, implicações éticas ou consequências não desejadas.

Mas o que exatamente é o pensamento computacional? Vamos dar uma olhada em uma definição técnica ...

"O pensamento computacional é o processo de pensamento envolvido na formulação de problemas e suas soluções para que as soluções sejam representadas de uma forma que possa ser efetivamente realizada por um agente de processamento de informações".

Cuny, Snyder, Wing, 2010 —

Phew, é um bocado, não é? Mas, como gostamos de dizer no CS Unplugged, são apenas grandes palavras para idéias simples! "Agente de processamento de informações" significa qualquer coisa que seque um conjunto de instruções para completar uma tarefa

---

Nós nos referiremos a isso como um computador para tornar as coisas um pouco mais simples. Para representar as soluções de forma que um computador possa realizá-las, devemos representá-las como um processo passo a passo - um **algoritmo**. Para criar essas soluções algorítmicas, aplicamos algumas habilidades especiais para resolver problemas. Essas habilidades são o que compõem o pensamento computacional! E são habilidades que são transferíveis para qualquer campo.

O pensamento computacional poderia ser descrito como "pensar como um cientista da computação", mas agora é uma habilidade importante para todos aprenderem, quer eles desejem ser cientistas da computação ou não! É interessante e importante notar que o Computational Thinking, e Computer Science, não são inteiramente sobre computadores, eles são mais sobre **pessoas**. Você pode pensar que escrevemos programas para computadores, mas realmente escrevemos programas para pessoas - para ajudá-los a se comunicar, encontrar informações e resolver problemas.

Por exemplo, você pode usar um aplicativo em um smartphone para obter direções para a casa de um amigo; o aplicativo é um exemplo de um programa de computador, e o smartphone é o "agente de processamento de informações" que administra o programa para nós. Quem projetou o algoritmo para trabalhar a melhor rota e todos os detalhes como a interface e como armazenar o mapa, aplicou o pensamento computacional para projetar o sistema. Mas eles não o projetaram por causa do smartphone; Eles projetaram isso para ajudar a pessoa a usar o smartphone.

## Pensamento computacional no CS desconectado

Ao longo das aulas e unidades no CS Unplugged, existem muitos links para o Pensamento Computacional. Ensinar o pensamento computacional através das atividades CS Unplugged ensina aos alunos como:

- descreva um problema,
- identifique os detalhes importantes necessários para resolver esse problema,
- romper o problema em pequenos passos lógicos,
- use estas etapas para criar um processo (algoritmo) que resolva o problema,
- e depois avalie esse processo.

Essas habilidades são transferíveis para qualquer outra área curricular, mas são particularmente relevantes para o desenvolvimento de sistemas digitais e a resolução de problemas usando as capacidades dos computadores.

Esses conceitos de pensamento computacional estão todos conectados uns aos outros e se apoiam, mas é importante notar que nem todos os aspectos do pensamento computacional acontecerão necessariamente em cada unidade ou aula. Em cada unidade e lição destacamos as conexões importantes para você observar seus alunos em ação

conjunto de 5 ou 6 habilidades de resolução de problemas que o pensamento computacional incorpora. Para o projeto desconectado, identificamos as seguintes seis habilidades de CT que muitas vezes são mencionadas na literatura; eles são descritos abaixo e, no final de cada lição desconectada, identificamos maneiras que essas habilidades apareceram na lição, para ajudá-lo a ver a conexão de CT com as lições.

## Habilidades de pensamento computacional



### ▼ Pensamento algorítmico

Algoritmos estão no coração do pensamento computacional e da ciência da computação, porque na Ciência da Computação as soluções para os problemas não são simplesmente uma resposta (por exemplo, "42", ou um fato), são algoritmos. Um algoritmo é um processo passo a passo que resolve um problema ou completa uma tarefa. Se você seguir as etapas do algoritmo corretamente, você chegará a uma solução correta, mesmo para diferentes entradas. Por exemplo, podemos usar um algoritmo para encontrar a rota mais curta entre dois locais em um mapa; O mesmo algoritmo pode ser usado para qualquer par de pontos de partida e de acabamento, de modo que a solução depende da entrada do algoritmo. Se conhecemos o algoritmo para resolver um problema, podemos resolver esse problema com facilidade, sempre que quisermos, sem ter que pensar! Nós podemos seguir os passos.

O pensamento algorítmico é o processo de criação de algoritmos. Quando criamos um algoritmo para resolver um problema, chamamos isso de solução algorítmica.

Os algoritmos computacionais (o tipo que pode ser executado em dispositivos digitais) têm relativamente poucos ingredientes, porque os dispositivos digitais têm apenas alguns tipos de instruções que podem seguir; as principais coisas que eles podem fazer são receber entrada, fornecer saída, armazenar valores, seguir instruções em uma seqüência, escolher entre opções e repetir instruções em um loop. Apesar de quão limitada é esta gama de instruções, descrevemos tudo o que os dispositivos digitais podem calcular, e é por isso que os algoritmos são descritos restritos a esses elementos.



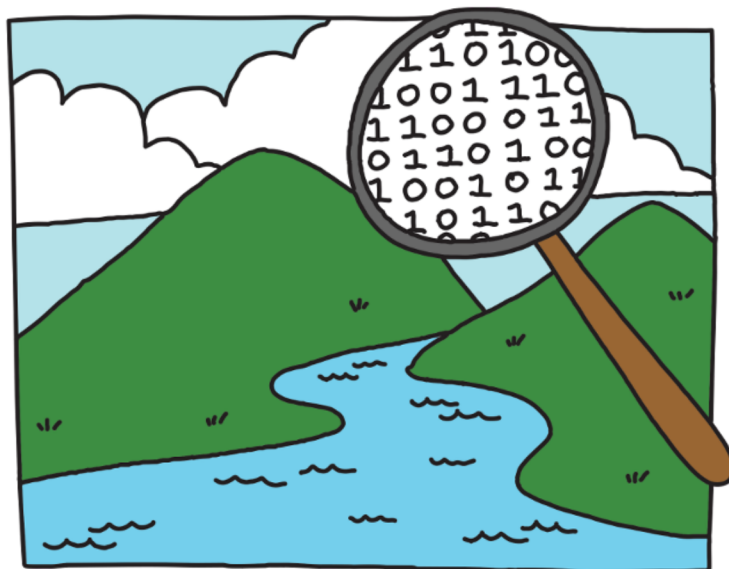
### ▼ Abstração

A abstração consiste em simplificar coisas para nos ajudar a gerenciar a complexidade. Ele exige identificar quais os aspectos mais importantes de um problema e esconder os outros detalhes específicos sobre os quais não precisamos nos concentrar. Os aspectos importantes podem ser usados para criar um modelo, ou representação simplificada, da coisa original com a qual lidamos. Podemos então trabalhar com este modelo para

abstração.

Usamos abstração muitas vezes em nossas vidas diárias, por exemplo, quando usamos mapas. Os mapas nos mostram uma versão simplificada do mundo, deixando de lado detalhes desnecessários, como onde cada árvore individual em um parque é, e apenas mantendo as informações mais relevantes que o leitor de mapa precisará, como estradas e nomes de ruas.

Os dispositivos digitais usam a abstração o tempo todo; Eles tentam esconder o máximo de informações desnecessárias do usuário. Por exemplo, digamos que você tirou uma foto panorâmica agradável em sua última viagem de acampamento, e agora você deseja editá-la no seu laptop e ajustar as cores nele. Normalmente, poderíamos fazer isso abrindo um programa de edição de imagem, ajustando alguns controles deslizantes de cores ou talvez escolhendo um filtro. Quando você faz isso, há muitas operações complicadas que o computador está escondendo de você.



A imagem que você tirou é armazenada no computador como uma grande lista de pixels, que são cada uma de cores diferentes, e cada cor é representada por um conjunto de números e cada um desses números é armazenado como dígitos binários! Isso é muita informação. Imagine se, quando ajustou as cores, você teve que passar e ver todos os valores de cor de cada pixel e mudar todos e cada um desses! Isso é o que o computador está fazendo para você, mas como você não precisa saber isso para atingir seu objetivo, o computador esconde essa informação.

### ▼ Decomposição



A decomposição consiste em quebrar problemas em partes mais pequenas, mais

mais pequenos e mais simples se acumulam em uma solução para o grande problema com o qual começamos. Decomposição ajuda a fazer grandes problemas muito menos intimidante!

A decomposição é uma habilidade importante para criar algoritmos e processos que podem ser implementados em um dispositivo de computação, porque os computadores precisam de instruções muito específicas. Eles precisam ser informados de cada uma das pequenas etapas que eles precisam seguir para fazer as coisas.

Por exemplo, a tarefa geral de fazer um bolo pode ser decomposta em várias tarefas menores, cada uma das quais pode ser realizada com facilidade.

### Fazer bolo

1. Assar bolo
  1. Coloque os ingredientes na tigela (manteiga, açúcar, ovo, farinha)
  2. Misturar
  3. Despeje em lata
  4. Coloque no forno por 30 minutos
  5. Tirar da lata
2. Faça glacê
3. Coloque o bolo



### ▼ Generalização e padrões

A generalização também é referida como "reconhecimento e generalização de padrões". A generalização está levando uma solução (ou parte de uma solução) a um problema e generalizando-a para que ela possa ser aplicada a outros problemas e tarefas semelhantes. Uma vez que as soluções em Ciência da Computação são algoritmos, isso significa que nós tomamos um algoritmo e tornamos geral o suficiente para que ele possa ser usado para uma variedade de problemas. Este processo envolve a abstração, porque para fazer algo mais geral, temos que remover detalhes desnecessários que estão relacionados a um problema ou situação específica, mas não são importantes para o funcionamento do algoritmo.

Os padrões de mancha são uma parte importante deste processo, quando pensamos em problemas, podemos reconhecer semelhanças entre eles e que podem ser resolvidos de maneiras semelhantes. Isso é chamado de correspondência de padrões, e é algo que fazemos naturalmente o tempo todo em nossa vida diária.

## ▼ Avaliação



A avaliação é sobre a identificação das possíveis soluções para um problema e julgar qual é o melhor para usar, se eles funcionarem em algumas situações, mas não em outras, e como elas podem ser melhoradas. Ao julgar nossas soluções, precisamos pensar em uma série de fatores. Por exemplo, quanto tempo levará esses processos (algoritmos) para resolver o problema e ele resolverá com segurança o problema, ou se houver certas situações em que ele irá realizar de forma muito diferente. A avaliação é algo que fazemos muito em nossa vida cotidiana.

Existem diferentes maneiras de avaliar nossas soluções algorítmicas. Podemos testar sua velocidade implementando-os em um computador; ou podemos analisá-los contando ou calculando quantos passos eles provavelmente tomarão. Podemos testar que eles funcionam corretamente, dando a nossa solução muitas entradas diferentes e verificando que funciona como esperado. Quando fazemos isso, precisamos pensar sobre os diferentes insumos que testamos, porque não queremos verificar todas as entradas possíveis (muitas vezes há um número infinito de entradas possíveis diferentes!), Mas ainda precisamos saber se isso funcionará para eles. O teste é algo que os cientistas da computação e os programadores fazem o tempo todo. Mas porque geralmente não podemos testar todas as entradas possíveis, também tentamos avaliar um sistema usando o raciocínio lógico.

## ▼ Lógica



CS **UNPLUGGED** 2.0 Alfa 5

As vezes, ao tentar resolver problemas, precisamos pensar logicamente. O raciocínio lógico é tentar entender as coisas observando, coletando dados, pensando nos fatos que você conhece e depois descobre as coisas com base no que você já conhece. Isso nos ajuda a usar nossos conhecimentos existentes para estabelecer regras e verificar factos.

Por exemplo, suponha que você esteja escrevendo software que funcione o caminho mais curto para um local da sua casa. No mapa a seguir, fica a 2 minutos da biblioteca, se você se dirige para o norte da sua casa, mas se você se dirige para o sul é de 3 minutos para o próximo cruzamento. Você pode se perguntar se há uma rota melhor para a biblioteca se você começar pela direção sul, mas logicamente não pode haver porque você já caminhou por 3 minutos para chegar ao cruzamento.

Em um nível mais profundo, os computadores são construídos inteiramente na lógica. Eles usam valores "Verdadeiros" e "Falso", e usam algo chamado "expressões

---

onde e por que alguma coisa no programa está errada.

---

 **COMPUTER  
SCIENCE EDUCATION**



 Microsoft  
**Philanthropies**

### Links Úteis

[Sobre](#)  
[Tópicos](#)  
[Recursos](#)

### Comunidade

[Twitter](#)  
[Youtube](#)  
[GitHub](#)

### Socorro

[Glossário](#)  
[Comentários](#)  
[Contato](#)

À procura de algo para as escolas secundárias? Confira o Guia de Campo de Ciência da Computação .

O material CS Unplugged é de código aberto no GitHub e o conteúdo deste site é compartilhado sob uma licença Creative Commons Attribution-ShareAlike 4.0 International . O CS Unplugged é um projeto do Computer Science Education Research Group da Universidade de Canterbury , Nova Zelândia.

2-0.0-alfa.5