

Arthur Branco Costa

# **Ensino de Computação no Ensino Fundamental**

Brasil

16 de Novembro de 2016

Arthur Branco Costa

## **Ensino de Computação no Ensino Fundamental**

Trabalho monográfico acadêmico apresentado como exigência de conclusão da disciplina MAC0499, coordenado pela Prof<sup>ca</sup>. Dr<sup>a</sup>. Nina S. T. Hirata.

Universidade de São Paulo

Instituto de Matemática e Estatística

Bacharelado em Ciência da Computação

Orientador: Prof. Dr. Leônidas de Oliveira Brandão

Brasil

16 de Novembro de 2016

# Agradecimentos

Aos meus pais, pelo amor, incentivo e apoio incondicional.

Meus agradecimentos aos amigos e amigas que fizeram parte da minha formação e que vão continuar presentes em minha vida.

E a todos que direta ou indiretamente fizeram parte da minha formação, o meu muito obrigado.

# Resumo

As inovações tecnológicas movem a economia e a sociedade, porém a alta taxa de desistência dos estudantes de TI e Ciência da Computação e a escassez desses profissionais no mercado, bem como, a necessidade de dominar tecnologias emergentes com proficiência, impulsionam a introdução do ensino de Computação cada vez mais cedo.

O presente trabalho visa a introduzir conceitos de programação e pensamento computacional para alunos do Ensino Fundamental, através de uma abordagem pedagógica que estimule o trabalho em equipe e a criatividade, utilizando ferramentas pedagógicas lúdicas, colaborativas e gratuitas.

Apresenta também o desempenho escolar obtido, relatando as experiências de cada aula, coletando os resultados das atividades diagnósticas e avaliando a adequação das atividades e ferramentas para a faixa etária abrangida.

**Palavras-chaves:** Computação para crianças. Ensino de programação. Code.org. Scratch.

# Abstract

Technological innovations move the economy and society, however the high dropout rate of IT students and computer science and the shortage of these professionals in the market, as well as the need to master emerging technologies with proficiency, drive the introduction of Computing ever earlier.

The present work aims to introduce concepts of programming and computational thinking for elementary school students, through a pedagogical approach that stimulates teamwork and creativity, using playful, collaborative and free pedagogical tools.

It also presents the obtained school performance, reporting the experiences of each class, collecting the results of the diagnostic activities and evaluating the adequacy of the activities and tools for the age group covered.

**Key-words:** Computing for children. Programming teaching. Code.org. Scratch.

# Sumário

<b>Introdução</b> .....	<b>7</b>
<b>I Sobre as pesquisas</b> .....	<b>9</b>
<b>1 Ferramentas pedagógicas</b> .....	<b>10</b>
1.1 LightBot .....	11
1.2 Code.org .....	13
1.3 A linguagem Scratch .....	14
1.4 MIT App Inventor .....	15
1.5 Computação Desplugada .....	16
<b>2 Computação na Educação Básica</b> .....	<b>17</b>
2.1 Diretrizes curriculares .....	17
2.2 Inclusão digital nas escolas .....	18
2.2.1 ProInfo .....	18
2.2.2 ACESSA São Paulo .....	19
<b>3 O projeto</b> .....	<b>21</b>
3.1 Letramento digital .....	21
3.2 Onde aplicar .....	22
3.3 Público-alvo .....	22
3.3.1 Quadro socioeconômico .....	23
3.3.2 Perfil dos participantes .....	23
<b>II As aulas</b> .....	<b>30</b>
<b>4 Módulo I - <i>Conhecendo a Computação</i></b> .....	<b>31</b>
4.1 Aula 1: <i>apresentações, Computação e a história do computador</i> .....	31
4.1.1 Apresentação e questionário .....	31
4.1.2 Roda de conversa .....	35
4.1.3 História do computador .....	36
<b>5 Módulo II - <i>LightBot</i></b> .....	<b>39</b>
5.1 Aula 2: <i>guiando os robôs</i> .....	39
5.1.1 Atividade desplugada: guiando o computador .....	39
5.1.2 Desafio LightBot .....	40

5.2	Aula 3: <i>atividade diagnóstica I</i>	44
<b>6</b>	<b>Módulo III - <i>Code.org</i></b>	<b>49</b>
6.1	Aula 4: <i>labirinto</i>	49
6.2	Aula 5: <i>artista</i>	54
6.2.1	Atividade desplugada: o computador que desenha	54
6.2.2	Code.org: o artista	56
6.3	Aula 6: <i>repetições no Minecraft</i>	59
<b>7</b>	<b>Módulo IV - <i>Scratch</i></b>	<b>64</b>
7.1	Aula 7: <i>polígonos e mandalas</i>	64
7.2	Aula 8: <i>atividade diagnóstica II</i>	66
7.3	Aula 9: <i>labirinto</i>	70
7.4	Aula 10: <i>Pong</i>	71
7.5	Aula 11: <i>Space Invaders</i>	73
7.6	Aula 12: <i>esconde-esconde</i>	74
<b>8</b>	<b>Módulo V - <i>MIT App Inventor</i></b>	<b>78</b>
8.1	Aulas 13 e 14: <i>bola de cristal</i>	78
	<b>III Resultados</b>	<b>81</b>
<b>9</b>	<b>Considerações finais</b>	<b>82</b>
9.1	Inclusão digital nas escolas	82
	<b>Conclusão</b>	<b>84</b>
	<b>Referências</b>	<b>85</b>
	<b>Apêndices</b>	<b>86</b>
	<b>APÊNDICE A Letramento digital</b>	<b>87</b>
	<b>APÊNDICE B Questionário</b>	<b>92</b>
	<b>APÊNDICE C Atividade diagnóstica: LightBot</b>	<b>95</b>
	<b>APÊNDICE D Atividade diagnóstica: repetição</b>	<b>102</b>

# Introdução

## Justificativa

Atualmente a Computação impulsiona a criação de empregos e a inovação em toda a economia e a sociedade. É importante que os estudantes adquiram uma clara compreensão do que é a Computação e como esse conhecimento pode impulsionar sua inserção no mercado de trabalho, independentemente da carreira profissional, pois, além da grande escassez de profissionais de tecnologia da informação (TI), (segundo a Softex, o Brasil pode deixar de arrecadar R\$ 115 bilhões em receitas, em 2020, devido a essa falta), profissionais em outras áreas precisam entender sobre Computação para se destacarem e serem mais produtivos.

No entanto, os cursos superiores de TI e Ciência da Computação possuem elevadas taxas de desistência dos estudantes. Entre as razões dessa evasão abundante encontram-se a falta de conhecimento dos alunos sobre a carreira e as expectativas não realistas em relação aos cursos, causadas entre outros fatores por uma falha na formação dos estudantes no Ensino Fundamental e Médio.

Como reflexo disso, nota-se que vários países têm introduzido Computação dentro das diretrizes de currículo na Educação Infantil, Fundamental e de Nível Médio - (CSTA, 2011) - visando a abranger o pensamento computacional e práticas de programação de maneira inspiradora, motivando estudantes a utilizá-la como ferramenta criativa.

## Objetivo

Este projeto educacional propõe a criação de um modelo de curso de Computação para crianças no Ensino Fundamental, 6º e 7º anos, com base em padrões internacionais de diretrizes curriculares, bem como expor os desafios encontrados ao longo dessas aulas, avaliar a efetividade e a qualidade das ferramentas utilizadas e o desempenho dos alunos em termos qualitativos.

## Organização do trabalho

Este trabalho divide-se em três partes.

Na Parte I apresentam-se: as ferramentas pedagógicas utilizadas no projeto, as diretrizes curriculares de alguns países que incluem Computação como disciplina, uma



---

breve contextualização comentando sobre a inserção dessa disciplina no currículo nacional, uma descrição do modelo de curso elaborado e o público escolhido para sua aplicação.

A Parte II consiste em expor detalhadamente cada uma das aulas ministradas durante a aplicação do projeto, incluindo seus objetivos, as atividades envolvidas e como desempenhá-las, bem como as considerações sobre o decorrer de cada aula.

Por fim, a Parte III contém considerações finais e sugestões de adaptações do projeto para trabalhos futuros.

# Parte I

## Sobre as pesquisas

# 1 Ferramentas pedagógicas

Em 2015, o autor começara a lecionar programação para crianças de 7 a 11 anos e introdução à tecnologia criativa para crianças de 5 e 6 anos, ambos na forma de cursos extracurriculares em uma escola de programação e robótica. Além de lecionar durante as férias escolares em uma escola adepta do *Movimento Maker* e robótica. No início de 2016, começou a trabalhar em um colégio particular que adotou Computação no currículo, com aulas para adolescentes de 13 a 16 anos.

A partir das experiências de ensino adquiridas nesses três colégios diferentes, ao longo de um ano e meio, foi possível ao autor entrar em contato com diversas ferramentas de ensino de lógica e programação, entre sites, programas, aplicativos e mesmo a Computação Desplugada (sem o uso do computador, utilizando-se de papéis, canetas etc., para ensinar Computação).

Serão descritas detalhadamente as ferramentas que puderam ser aplicadas e testadas com os alunos neste projeto educacional. As demais encontram-se tabeladas como sugestão para um projeto de maior duração e que inclua outras faixas etárias.

A maioria dos programas de ensino em questão buscam construir a noção de como programar um personagem, designando-lhe diversos comandos de acordo com o objetivo da fase, introduzindo conceitos básicos de Computação como: sequenciamento, repetição, procedimento, depuração de erros e condição.

Na Tabela 1 há uma lista de programas voltados para a Educação Infantil, desenvolvidos para crianças de uma faixa etária de cinco a oito anos, cuja habilidade de leitura está em desenvolvimento e que ainda estão adquirindo conhecimento alfabético e consciência fonológica. Atualmente, essas crianças interagem rotineiramente com a tecnologia, seja navegando, conversando ou jogando as mais diversas aplicações nos *tablets*, computadores e celulares. Esses recursos procuram mostrar às crianças, sempre de maneira lúdica, a possibilidade de elas criarem e se expressarem através dessas tecnologias.

Ferramenta	Faixa etária	Preço	Plataformas
<a href="#">LightBot Jr</a>	4 - 8	R\$ 6,99	Android, iOS, Kindle
<a href="#">Kodable</a>	5 - 10	Teste grátis	Web, Android, iOS
<a href="#">The Foos</a>	4+	Teste grátis	Android, iOS, Kindle
<a href="#">Code.org (curso 1)</a>	4 - 6	Gratuito	Web
<a href="#">Cargo-Bot</a>	5+	Gratuito	iOS
<a href="#">Scratch Jr</a>	5 - 7	Gratuito	Android, iOS, Chrome Book
<a href="#">Robozzle</a>	6+	Gratuito	Android, iOS, Web
<a href="#">CS Unplugged</a>	4+	Gratuito	Não utiliza o computador

Tabela 1: Ferramentas pedagógicas para a Educação Infantil

Os três primeiros itens da tabela compõem-se de diversas fases, nas quais o usuário deve cumprir missões para avançá-las, contudo há um momento do jogo em que se deve pagar para desbloquear novas etapas. O presente trabalho considerou para a modelagem do curso apenas ferramentas total ou parcialmente gratuitas, que já apresentam uma gama de conceitos computacionais bastante abrangente.

Para crianças que tenham compreensão de leitura, por volta dos sete anos de idade, o leque de ferramentas computacionais se expande. Além disso, os conceitos anteriores são aprofundados e novas habilidades são introduzidas. A tabela 2, a seguir, contém uma lista de programas mais avançados, com maior potencial de criação.

Ferramenta	Faixa etária	Preço	Plataformas
<a href="#">Tynker</a>	7+	Grátis (plano básico)	Web
<a href="#">Scratch</a>	8+	Gratuito	Web, Windows, Mac, Linux
<a href="#">Hopscotch</a>	9 - 11	US\$ 6,67/mês	iOS
<a href="#">Kodu</a>	9+	Gratuito	Windows
<a href="#">App Inventor</a>	12+	Gratuito	Web
<a href="#">Alice</a>	13+	Gratuito	Windows, Mac, Linux
<a href="#">Stencyl</a>	14+	Grátis (plano básico)	Windows, Mac, Linux

Tabela 2: Ferramentas pedagógicas no Ensino Fundamental

Nas próximas seções encontram-se descrições dos programas e *sites* utilizados neste projeto: *LightBot*, *Code.org*, *Scratch*, *App Inventor* e *Computação Desplugada*.

## 1.1 LightBot

Criado em 2008 por Danny Yaroslavski, o jogo LightBot não tinha a princípio um propósito educacional, porém com a aceitação e as sugestões dos usuários ele foi se moldando mais como uma ferramenta de ensino de programação e menos como um simples desafio estilo quebra-cabeça.

Em 2013, o jogo passou por uma reestruturação, responsável por adequá-lo melhor à curva de aprendizado, amenizando a elevação da dificuldade entre uma fase e a seguinte. A grande ausência de textos contribui para seu formato lúdico e bastante intuitivo, tornando-o possível de ser jogado até por crianças ainda não-alfabetizadas. O jogo tradicional é indicado para crianças a partir dos nove anos de idade, além disso há a versão junior, que é voltada para crianças de quatro a oito anos.

Atualmente, o jogo foi traduzido para 28 idiomas, embora as partes textuais sejam escassas. Elas aparecem na forma de um breve tutorial, conforme uma nova instrução é incorporada.

O LightBot é pago e está disponível para Android, iOS, Windows, Mac, bem como para navegadores web. Este último, trata-se de uma versão demonstrativa gratuita, que

será utilizada no projeto.

A intenção atual do jogo, além do entretenimento, é estimular o pensamento lógico e introduzir conceitos de programação nos usuários de uma maneira sutil. Seu visual, observado na Figura 1, mostra-se simples, não-poluído e sem publicidade, ajudando a criança a manter o foco na tela.

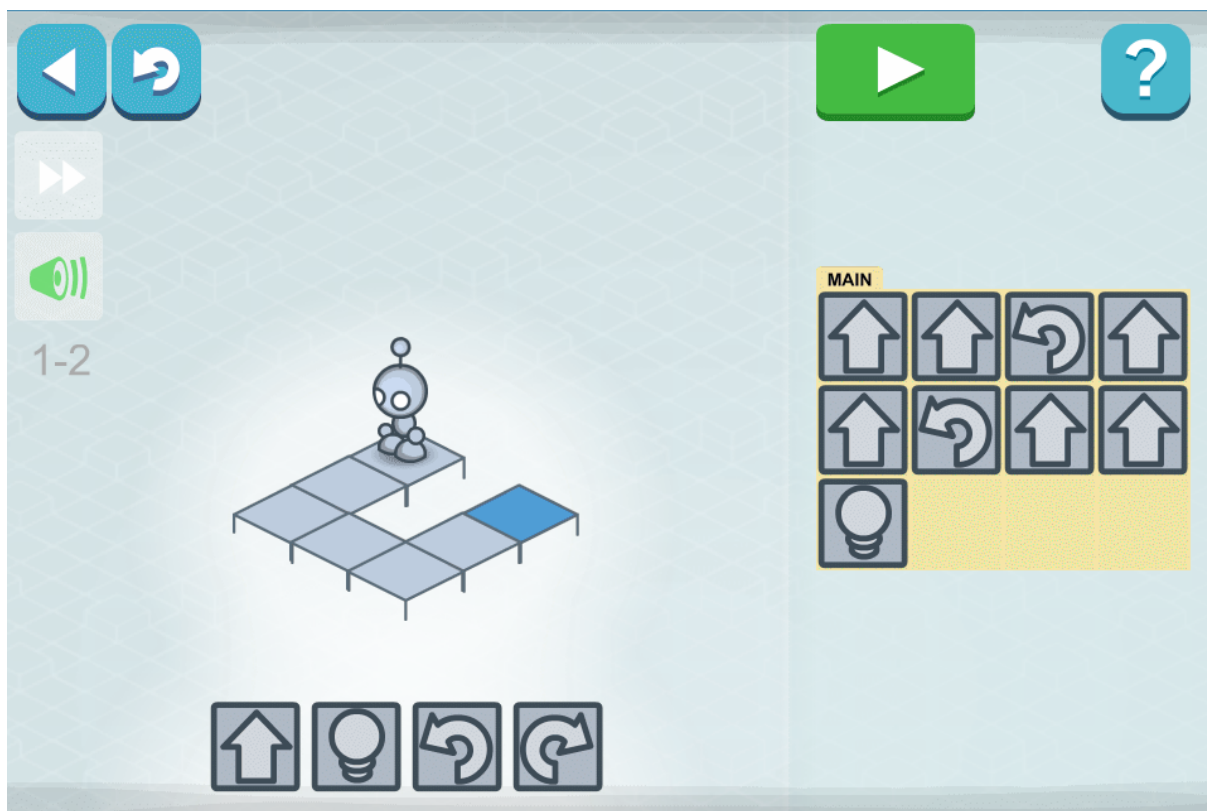


Figura 1: Jogo LightBot (versão demo) - Fase 1-2

No jogo, o usuário controla o robô mostrado na Figura 1, transmitindo-lhe uma série de instruções pré-determinadas, de modo a cumprir as missões estilo quebra-cabeça, guiando-o através dos labirintos de cada fase.

Após iniciar, a área amarela, *Main*, deverá ser preenchida com as instruções, bastando clicar sobre uma ou arrastá-la para adicionar um comando à fila de execução, introduzindo os conceitos de *instrução* e *algoritmo*.

Para movimentar o robô, o usuário deverá optar pelos comandos: mover para a frente, virar para o lado e subir ou descer um degrau. Para concluir uma fase, deve-se acender todos os ladrilhos azuis presentes nela, para isso utiliza-se o comando de acender ou apagar um ladrilho.

Para rodar o algoritmo feito basta apertar o botão *Play* e observar as instruções serem executadas sequencialmente. Todavia, este processo ocorre rapidamente, sendo por vezes difícil acompanhar a rápida execução dos comandos. Uma sugestão de aperfeiçoa-

mento, seria adicionar um botão de execução passo a passo, permitindo ao usuário decidir quando executar o comando seguinte.

No primeiro módulo, passar de fase requer algoritmos bastante simples, porém à medida que o jogo avança, novas instruções e conceitos de programação são incorporados, como procedimentos e inclusive recursão. Além disso, os labirintos ficam mais complexos, exigindo cada vez mais do pensamento analítico dos usuários.

## 1.2 Code.org

Iniciado em 2013, Code.org é uma organização sem fins lucrativos, que se dedica a expandir o acesso da Ciência da Computação, divulgando e ensinando programação para pessoas de todas as idades. Seus criadores acreditam que esta disciplina deveria ser incluída no currículo escolar, com o mesmo peso de outras matérias como Biologia, Química ou Matemática, de modo que todos os estudantes de qualquer escola possam ter acesso a essa oportunidade de aprendizado.

O site [code.org](https://code.org) é uma referência mundial, amplamente apoiado por grandes personalidades desde célebres programadores como Bill Gates e Mark Zuckerberg, a celebridades de outros ramos: Ashton Kutcher (ator) e Snoop Dogg (rapper), que defendem a importância de se implantar essa matéria nas escolas. Neste *link* <https://code.org/quotes/> é possível ler os comentários desses e outros famosos.

A plataforma possui quatro grandes cursos:

- Curso 1: voltado para crianças que ainda não dominam a leitura, de 4 a 6 anos;
- Curso 2: ponto de início para os alunos que sabem ler. A partir de 6 anos. Algumas etapas deste curso serão utilizadas neste projeto.
- Curso 3: continuação do curso 2. De 8 a 18 anos;
- Curso 4: continuação do curso 3. De 10 a 18 anos.

Além disso, possui módulos de aulas *offline* (sem uso do computador) e diversos desafios temáticos de uma hora de duração, ambientados com os cenários e personagens de *Frozen*, *Star Wars*, *Minecraft* etc.

A linguagem de programação empregada pelo site é orientada a eventos, ou seja, as execuções dos comandos apenas se iniciam após determinado evento ocorrer. Ela é bastante intuitiva, pois encontra-se em formato de blocos que se encaixam uns nos outros, como peças de montar e, além disso, já vêm com os comandos escritos, evitando que o aluno precise se preocupar com a sintaxe da linguagem.

### 1.3 A linguagem Scratch

Desde o início da década de 1960, pesquisadores criaram diversas linguagens de programação e ambientes com a intenção de tornar a programação mais acessível para o maior número de pessoas (Kelleher e Pausch 2005). Em seu artigo, Kelleher e Pausch descrevem o alto nível organizacional da taxonomia dessas linguagens, bem como as táticas utilizadas para tornar a programação mais acessível aos novatos, citando quase todas as linguagens que influenciaram na criação do Scratch: Logo (criada em 1967), Smalltalk, HyperCard, StarLogo e Etoys, ficando de fora somente a AgentSheets (uma linguagem paga, voltada para crianças, com uma interface de arrastar e soltar comandos).

A linguagem Scratch foi desenvolvida pelo MIT Media Lab *Lifelong Kindergarten Group* sob a liderança de Mitchel Resnick, com sua primeira aparição em 2002 (para teste) e sendo liberada oficialmente em 2005, contudo a versão utilizada neste projeto é a 2.0, liberada somente em 2013. A linguagem possibilita a criação de jogos e aplicações, assim como histórias e animações. Implementada em Squeak e ActionScript, ela é muito fácil de se operar, por possuir uma tipagem dinâmica e por aproximar-se bastante da linguagem natural, uma vez que fora criada sob o paradigma imperativo e orientada a eventos.

A Figura 2 apresenta a interface do usuário, que divide a tela em diversos painéis, incluindo: o palco (com os cenários) e a lista de personagens (*sprites*), a paleta de blocos e o editor de *scripts*. Tais blocos consistem em fragmentos de código que podem ser arrastados para a área de *scripts* e criar programas, ao se encaixarem uns nos outros. A paleta organiza-se em dez grupos diferentes de blocos: **movimento** (trabalhando com direção, graus e plano cartesiano), **aparência** (inclui trocas de cenários e fantasias, efeitos de cor e brilho), **som** (inclusão de músicas e sons, instrumentos musicais e batidas), **caneta** (módulo similar à linguagem Logo, desenhando na tela o caminho por onde o personagem deve passar), **variáveis**, **eventos** (contém todos os gatilhos para iniciar os *scripts*), **controle** (laços e condicionais), **sensores** (de toque, de cor, de ruído etc.), **operadores** (aritméticos e lógicos) e **mais blocos** (procedimentos e extensões para integração com Lego WeDo e Arduíno).

Segundo dados estatísticos do site [scratch.mit.edu/statistics/](https://scratch.mit.edu/statistics/), a comunidade Scratch engloba mais de 14,5 milhões de usuários espalhados por mais de 150 países (a linguagem foi traduzida para 40 idiomas), que interagem entre si abrindo estúdios de criação, fóruns de dúvidas e compartilhando programas e seus códigos. Durante o cadastro, os usuários são incentivados a não colocarem o nome real, por questões de segurança e exposição. Além disso, caso sejam menores de idade, devem vincular sua conta ao e-mail de um responsável, para que ele aprove ou não o compartilhamento dos programas.

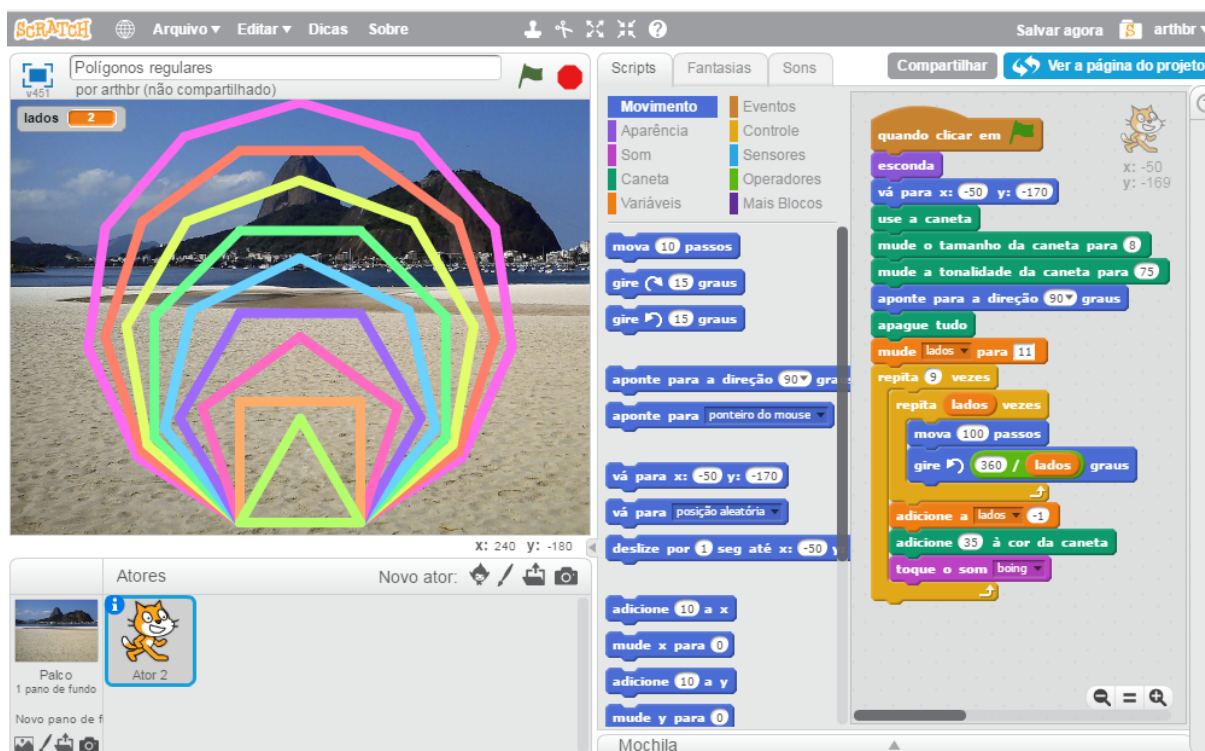


Figura 2: Exemplo de programa em Scratch

## 1.4 MIT App Inventor

Também conhecido como **App Inventor for Android**, esta aplicação de código aberto em Java foi lançada publicamente em 2010 pela Google, porém hoje é mantida pelo Massachusetts Institute of Technology (MIT). Esta ferramenta possibilita a criação de aplicativos para o sistema operacional Android. Para este projeto será utilizada a versão 2.0 criada em 2013 já pela equipe do MIT.

Sua interface gráfica assemelha-se à do StarLogo TNG e, assim como no Scratch, permite aos usuários arrastar e soltar os blocos de comandos que desejam utilizar, o próprio encaixe entre blocos assemelha-se à junção de peças em um quebra-cabeça. Para desenvolver qualquer programa é necessário utilizar os dois módulos da plataforma: o *designer* e o dos blocos.

No primeiro, há vários painéis: o **visualizador** consiste na disposição que os componentes aparecerão na aplicação; a **paleta** trata-se de uma lista contendo todos os *wid-gets* (componentes) que podem ser arrastados e soltos no visualizador, como botões, caixas de textos, listas, calendários, sensores etc.; **componentes** mostra todos os componentes que estão sendo utilizados, bem como sua hierarquia e, por último, as **propriedades** do componente em destaque, podendo personalizar cor, fonte, texto etc.

Na visão dos blocos há apenas dois painéis: o de **blocos** contém todos os comandos que podem ser utilizados no código, desde lógica e aritmética até condicionais; já o painel **visualizador** consiste no editor de *scripts*: área onde se manipula o código da aplicação.



## 1.5 Computação Desplugada

Trata-se de ensinar Computação sem utilizar computadores, com a exposição de diversos conceitos, explorados de uma forma divertida e cooperativa, mostrando sua palpabilidade, ou seja, trazendo-os mais próximo do universo dos alunos.

Há inúmeras atividades possíveis de serem realizadas que, em geral, requerem algum material (papel, canetinhas, recortes, fita adesiva) e um ambiente amplo, com espaço para movimentação, dependendo de quantas crianças houver.

As atividades desplugadas foram incluídas neste curso como uma alternativa para mostrar que mesmo sem um computador é possível se transmitir esse conteúdo computacional. “Na realidade, por vezes, o computador torna-se apenas uma distração no processo de aprendizagem” (Bell et al. 2011).

Neste projeto foram utilizadas várias ideias e adaptações do *CS Unplugged*, que consiste em uma coleção de atividades desplugadas gratuitas, além de ser uma comunidade para compartilhamento de ideias e dúvidas, vídeos explicativos e novidades. As atividades 5.1.1, 5.2, 6.2.1 e 7.2 têm como base alguma ideia do *CS Unplugged*.

## 2 Computação na Educação Básica

A Computação já está bem estabelecida com o caráter de Ciência, principalmente nos cursos de graduação e pós-graduação. Sob essa ótica, seria plausível dar a essa disciplina um tratamento mais próximo ao das matérias presentes no currículo da Educação Básica, como Biologia, Matemática e Física.

A Ciência da Computação de acordo com o *Model Curriculum for K-12 Computer Science 2011* (Seehorn et al. 2011) auxilia o aluno em diversos processos cognitivos, desenvolvendo sua capacidade de resolução de problemas e raciocínio lógico. Além disso, ela fornece um bom suporte às outras disciplinas, motivando o estudante a se relacionar com as outras ciências utilizando uma nova abordagem e a fluência no uso de tecnologias.

### 2.1 Diretrizes curriculares

O ensino de programação nas escolas tem crescido no mundo (Saeli et al. 2011), nesse ínterim, surgem novas diretrizes sobre o ensino de Computação, almejando que o aluno do nível Fundamental aprenda não só sobre a informática básica, já presente em grande parte dos colégios, mas também sobre o pensamento computacional, resolvendo problemas algorítmicos e computacionais através do uso de ferramentas de softwares adequadas. Para tal, foram seguidas, neste trabalho, as diretrizes curriculares do modelo americano: (Tucker et al. 2003), estabelecidas nos seguintes pilares:

- **Pensamento computacional:** engloba a resolução de problemas de forma analítica, incluindo conceitos computacionais como iteração e condição, de forma a auxiliar o aluno a analisar os problemas, aplicando melhores estratégias e ferramentas, ampliando sua produtividade e explorando sua criatividade;
- **Colaboração:** incentivo ao trabalho em equipe e à comunicabilidade;
- **Programação:** trata-se da competência de criar programas, de modo que os alunos aprendam a desenvolver e publicar projetos, na forma de animações, jogos e aplicativos;
- **Computadores e dispositivos de comunicação:** abordam os elementos do computador, dispositivos de comunicação e a internet;
- **Impactos éticos, globais e na comunidade:** englobam princípios de privacidade, direitos autorais, segurança, veracidade de informação, bem como o impacto da tecnologia na sociedade.

Nos moldes curriculares da CSTA (Computer Science Teachers Association), apoiada pela ACM (Association for Computing Machinery), há três níveis de aprendizagem, divididos em: nível I (do Ensino Infantil ao Fundamental I), nível II (Fundamental II) e nível III (Ensino Médio). A faixa etária dos alunos deste projeto corresponde ao nível II, entretanto, como eles não passaram pelo nível I foi feita uma mescla das competências de ambos os níveis para compor os objetivos de aprendizagem, descritos a seguir:

- Identificar de quais formas a tecnologia é empregada nos mais variados setores, compreendendo a onipresença e o impacto da Computação na vida pessoal e na sociedade;
- Resolução de problemas de pensamento lógico utilizando recursos tecnológicos;
- Construir algoritmos para efetuar tarefas (implementação e teste) usando diferentes linguagens de programação visual, orientadas a eventos;
- Trabalhar de maneira cooperativa e colaborativa, identificando como a programação em pares pode beneficiar a resolução de um problema;
- Saber identificar diferentes tipos de comportamentos éticos e sociais, discernindo sobre o uso responsável da tecnologia, bem como, as consequências de uma utilização inapropriada.

## 2.2 Inclusão digital nas escolas

Em decorrência dos avanços tecnológicos, a sociedade contemporânea vive em constante adaptação a novas ferramentas, que cada vez mais incorporam-se em nosso cotidiano. Contudo, o advento dessas novas tecnologias causa grande impacto nas relações sociais, por um lado, ampliando fortemente as possibilidades de informação e comunicação, por outro, distanciando pessoas de classes sociais mais baixas cuja acessibilidade virtual é precária. Nesse contexto, o Governo Federal vem tentando, desde a década de 90, disseminar a informática no sistema público de ensino brasileiro, com experiências e programas de inclusão digital.

### 2.2.1 ProInfo

Criado pelo Ministério da Educação (MEC) no final da década de 90, o Programa Nacional de Informática na Educação (ProInfo) objetiva a disseminação e o incentivo do uso pedagógico das tecnologias de informação e comunicação na rede pública de educação básica, instalando laboratórios de informática e levando conteúdos educacionais às escolas públicas.

Para receber um laboratório, escolas urbanas devem atender a algumas condições, como oferecer Ensino Fundamental completo (do 1º ao 9º ano), possuir mais de 50 alunos e energia elétrica. Dessa forma, após adesão e seleção da escola, o MEC oferece-lhe o equipamento para montar um laboratório: oito CPU's e um servidor, impressora a *laser*, terminais de acesso e estabilizadores, com garantia de três anos, sendo responsabilidade das prefeituras e dos governos estaduais providenciar a infraestrutura necessária para receber os laboratórios.

### 2.2.2 Acessa São Paulo

Criado em meados do ano 2000 pelo Governo do Estado, o programa Acessa São Paulo nasceu da necessidade de fornecer acesso virtual à população paulista, em especial acesso à internet. Este programa consiste na implantação de espaços públicos com computadores e acesso gratuito à internet, denominados infocentros, além de contribuir com projetos de inclusão digital nas escolas estaduais.

O programa já espalhou mais de 3.500 computadores pelo estado, dos quais oito situam-se na sala de informática da Escola Estadual Jornalista Wandyck de Freitas, palco da aplicação deste projeto, que pode ser observado na Figura 3, equipado com oito computadores, um servidor, estabilizadores, quadro-branco e cadeiras de escritório.



Figura 3: Sala de informática do Acessa São Paulo

Optou-se, assim, por atender 14 alunos por vez, de modo a aproveitar o máximo potencial da sala.

### Incertezas

Tais programas de inclusão digital, embora oficializados e operantes, trazem diversas dúvidas quanto à maneira como as escolas fazem uso dessas tecnologias e a formação dos professores para a melhoria do ensino e da aprendizagem. Uma vez que, de nada adianta possuir um laboratório de informática se os alunos não obtêm acesso a ele ou

se o treinamento dado aos professores apenas incluir noções de informática, sem de fato indicar maneiras de se utilizar o computador almejando a melhoria do ensino.

## 3 O projeto

O desafio deste projeto se traduz em propor um modo de aplicação do conhecimento computacional no ensino público, adequado a diretrizes curriculares internacionais e utilizando-se de ferramentas pedagógicas digitais, atuais e voltadas para o público infantil, atendendo às necessidades de inclusão digital da escola. Também descrevem-se as dificuldades identificadas na preparação e no decorrer das aulas.

Dessa forma, foi elaborado um curso contendo 14 aulas de Computação para crianças de 11 e 14 anos de idade, nas quais serão apresentados conceitos básicos de programação, como: sequência de comandos, repetição, condição, com a finalidade de mostrar e estimular a visão de que a programação pode e deve fazer parte do universo deles, para que eles possam criar e desenvolver suas ideias, como uma nova forma de se expressar, podendo aplicá-las no dia a dia. Além disso, o projeto caminha atrelado a conceitos matemáticos, mais especificamente: graus, polígonos e plano cartesiano, temas que fazem parte da grade curricular da disciplina.

### 3.1 Letramento digital

A proposta do curso enviada às possíveis escolas de aplicação pode ser lida no Apêndice A. Contudo, ela sofreu algumas alterações no decorrer do semestre, devido a variados fatores como: comportamento das turmas, adequação ao calendário escolar, desempenho e compreensão dos alunos. Basicamente, o curso divide-se em módulos de acordo com a ferramenta pedagógica utilizada, listados a seguir na mesma ordem e estrutura como as aulas serão apresentadas na Parte II, acompanhados do conteúdo computacional introduzido em cada módulo.

- Módulo I: introdução (1 aula)
  - O que é Computação e para que serve
  - História do computador
- Módulo II: LightBot (2 aulas)
  - Instrução
  - Algoritmo
- Módulo III: Code.org (3 aulas)
  - Desenho (similar ao da linguagem **Logo**)

- Repetição
- Módulo IV: Scratch (7 aulas)
  - Linguagem de programação, orientação a eventos
  - Condição
  - Edição de imagem, som, sensores etc.
- Módulo V: App Inventor (2 aulas)
  - Aplicativos

## 3.2 Onde aplicar

Com o curso em mão, restava encontrar uma escola onde aplicá-lo. Assim, entrando em contato com a Faculdade de Educação da Universidade de São Paulo (FEUSP), obteve-se o direcionamento necessário para procurar por escolas.

A FEUSP mantém projetos pedagógicos desenvolvidos por seus docentes, vinculados ao programa de formação de professores, que faz parte de uma política nacional de incentivo e colaboração à formação de professores, bem como uma [lista atualizada anualmente de escolas recomendadas](#) pelo programa.

Assim, se deu o contato inicial com diversas escolas, até se obter uma resposta positiva e, após algumas visitas e alinhamentos, a aplicação teve início na [Escola Estadual Jornalista Wandyck de Freitas](#).

Ela situa-se próxima ao Parque das Hortênsias em Taboão da Serra, funcionando em um prédio com dois pavimentos: térreo e primeiro andar. Tem capacidade para atender até mil alunos, distribuídos nas nove salas de aulas disponíveis. A Escola oferece os cursos: Ensino Fundamental (anos finais) e Ensino Médio regular. Além disso, seu funcionamento divide-se em três períodos: manhã, tarde e noite. Neste projeto, será considerado apenas o turno vespertino.

## 3.3 Público-alvo

O público participante do projeto é composto por 56 alunos, compreendidos na faixa etária de 11 a 16 anos, que cursam 6º e 7º anos (28 em cada) do Ensino Fundamental da Escola Estadual Jornalista Wandyck de Freitas. Para integrar este grupo, a coordenação pedagógica priorizou os alunos que possuem maior dificuldade na disciplina de Matemática, na tentativa de estimular seu aprendizado utilizando uma nova abordagem, mais interativa, fugindo dos moldes do ensino tradicional.

### 3.3.1 Quadro socioeconômico

Analisando o projeto pedagógico de 2015 da escola, observa-se o quadro social e econômico das famílias dos educandos do Ensino Fundamental, apresentados na forma de gráficos na Figura 4.

A maior parte das famílias é composta por cinco ou mais pessoas, dentre as quais mais de dois integrantes trabalham (em 57% dos casos são mais de quatro trabalhadores). Elas sobrevivem com uma renda mensal entre um e quatro salários mínimos, sendo que apenas 13% sustentam-se com mais do que cinco salários mínimos por mês, o equivalente a R\$ 4.400,00 reais hoje. Além disso, o nível de escolaridade dos responsáveis compreende-se em geral até o Ensino Médio completo e a pequena parcela de 17% possui Ensino Superior completo.

Cerca de um terço dos educandos não mora com ambos os pais e um quarto deles costuma ficar por pelo menos oito horas sem supervisão em casa. Um índice alarmante é que 15% dos estudantes alegam não ter quem acompanhe seus estudos. Além disso, quase metade deles estuda por somente uma hora diária. Esses últimos fatores (falta de acompanhamento e baixo tempo de estudo) são importantes requisitos da vida escolar, influenciando diretamente no mal desempenho na escola.

Outro ponto observado, apontado na Figura 5, foi: o compromisso com a leitura das famílias, das quais ao menos um integrante mantém esse hábito. Entre esses leitores, 75% optam por livros e quase a metade deles acompanha revistas e/ou jornais.

Quase todas as famílias declaram-se religiosas, sendo que metade delas são católicas e a outra metade compõe-se de evangélicas.

Quanto às atividades culturais mais realizadas, a grande maioria das famílias prefere ir ao cinema. Pequenas parcelas optam por teatro, shows e exposições. Além disso, mais de um quarto dos alunos dedicam no mínimo quatro horas diárias à TV, ao videogame e à internet.

### 3.3.2 Perfil dos participantes

Na primeira aula foi passado um breve questionário para os alunos preencherem, a partir do qual foi traçado o perfil de cada um. Ele se divide em duas partes: a primeira aborda questões socioeconômicas, cujas respostas localizam-se mais adiante; já a segunda busca analisar de que forma as crianças utilizam o computador, estas respostas relacionam-se mais diretamente com outras etapas do projeto, por isso se encontram na seção 4.1.1.

No dia desta atividade, faltaram três alunos do 6º ano e seis do 7º ano. Assim, excetuando-se a primeira questão (sobre gênero), em que a Escola forneceu esse dado dos alunos, obteve-se um total de 47 fichas respondidas.



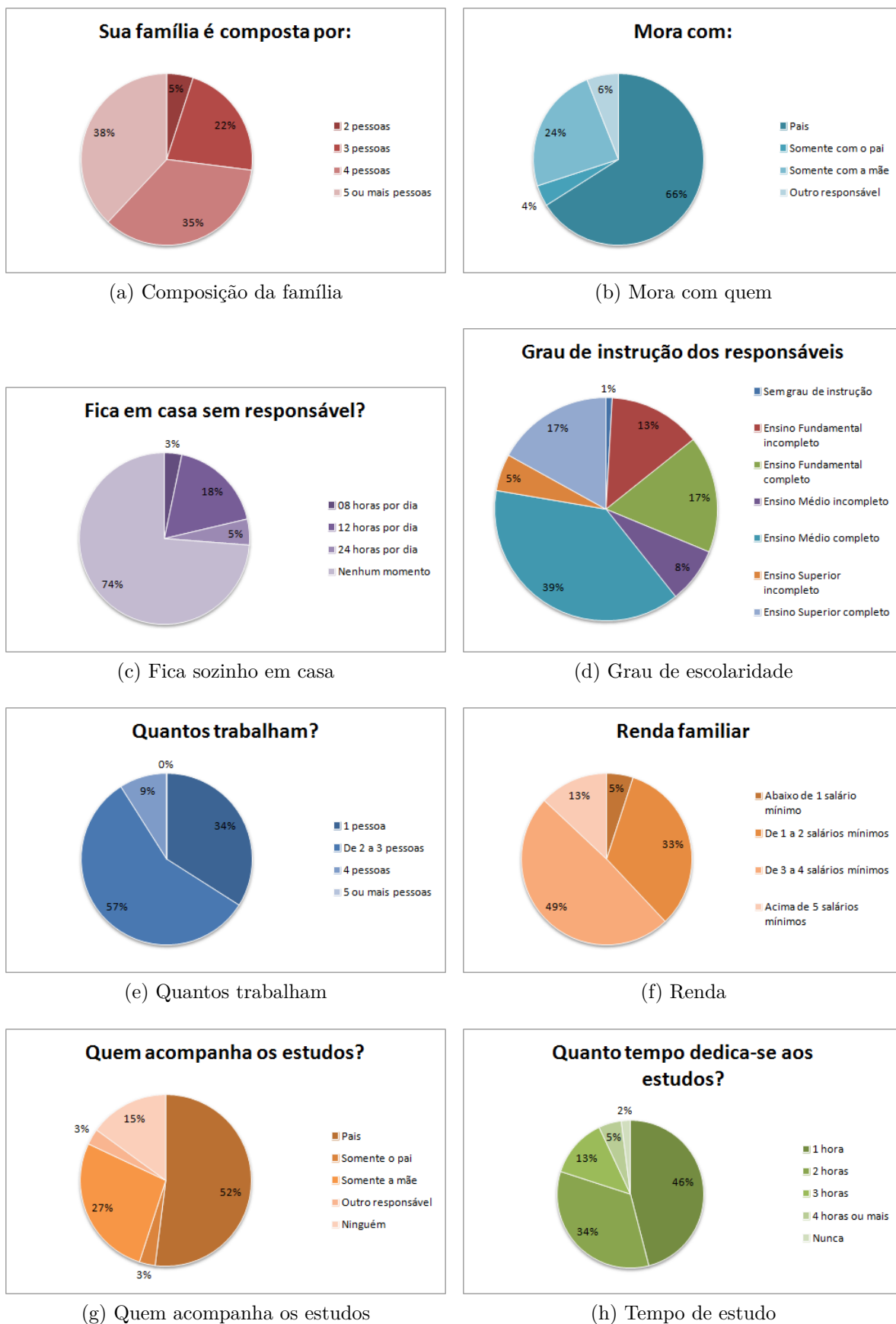


Figura 4

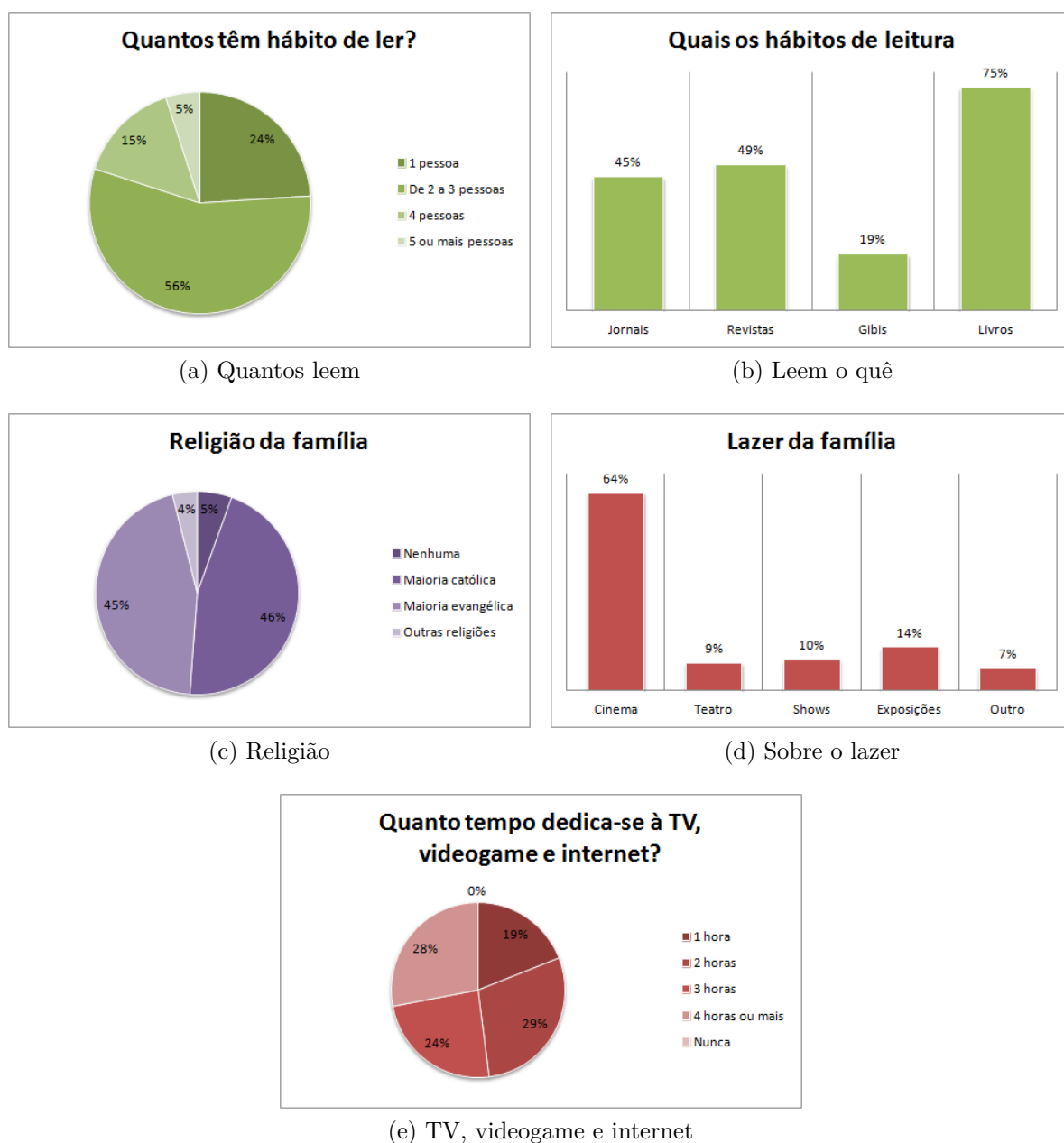


Figura 5

### Gênero

Segundo estatísticas do Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP) sobre cursos superiores na área de Computação, o índice de mulheres matriculadas no cursos de TI, SI, Ciência e Engenharia da Computação, vem caindo gradativamente. Em 1991 no Brasil a porcentagem feminina nesses cursos girava em torno de 50%, enquanto em 2014 o número encontra-se em 15%. Na Figura 6, podemos observar que cerca de um quarto dos participantes se identificou como sendo do gênero feminino.

Inúmeras vezes, meninas requerem permissão para utilizar computadores sozinhas, devido à preocupação dos responsáveis em relação com quem ela pode se comunicar vir-

tualmente, enquanto meninos da mesma faixa etária, são incentivados, sem qualquer permissão explícita, a explorar os meios tecnológicos. “Nesse cenário dominado pelo sexo masculino, as mulheres se sentem meras coadjuvantes e menos confortáveis nas áreas relacionadas à Computação. Tal estigma pode fazer com que as mulheres se sintam menos capacitadas para atuar nessas áreas. Por isso, o acesso e a exposição igualitários de experiência e conhecimento computacional durante o período escolar é imprescindível para contornar esse paradigma.” Mattos (3).

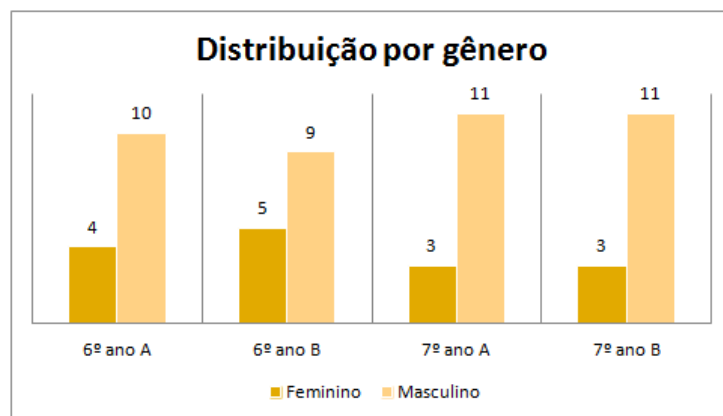


Figura 6: Divisão por gênero

### Idade

De acordo com o Conselho Nacional de Educação (CNE), uma criança no sexto ano do Ensino Fundamental deveria ter idealmente entre 11 e 12 anos, assim, sucessivamente, para uma criança no sétimo ano seria ideal ter entre 12 e 13 anos de idade. O gráfico da Figura 7 evidencia no mínimo três alunos de cada série considerados em idade escolar atrasada. Foi perguntada a data de nascimento em vez da idade, para evitar comentários maliciosos apontando a defasagem de outros alunos em relação à idade escolar.

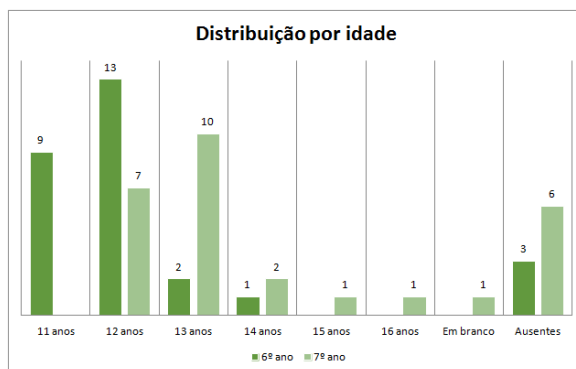


Figura 7: Divisão por idade

### Facilidade de acesso

A Figura 8 aborda um aspecto mais econômico da família, a disponibilidade de computadores, celulares e *tablets*, bem como de internet, na casa da criança. A partir do gráfico a seguir observa-se que a esmagadora maioria possui acesso à internet em casa, apenas dois alunos do 6º e um do 7º não possuem (equivalente a 8% e 5% das amostras). Esta conexão se dá a partir de um computador, celular ou *tablet*, ou ainda, de ambos (caso mais frequente).

Este índice está muito acima da média nacional, em que metade dos brasileiros está conectada à internet, segundo o Instituto Brasileiro de Geografia e Estatística (IBGE).

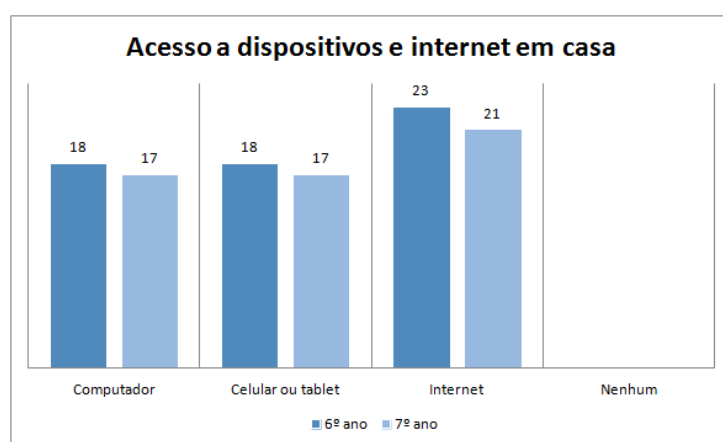


Figura 8: Possui computador, celular, *tablet* e internet em casa

Nascidos entre 2000 e 2005, desde muito pequenos já se mostram familiarizados com aparelhos tecnológicos e as mais variadas possibilidades da internet. Desse seletivo grupo da Geração Z, metade já utilizava o computador desde os sete anos de idade ou antes, vide Figura 9.

Contudo, ao perguntar-lhes verbalmente sobre o uso de *smartphones* ou *tablets*, as idades iniciais diminuíram ainda mais, pois, devido à portabilidade, às telas e menus mais intuitivos e à funcionalidade desses dispositivos, estes aparelhos dominam o mercado cada vez mais.

### Frequência de utilização

Na Figura 10, nota-se que o uso do computador por eles é bastante frequente ao longo da semana e, mesmo os que quase nunca se utilizam do computador, quando o usam ficam em geral de uma a quatro horas conectados.

### Locais onde acessam

Na Figura 11 [a], aproximadamente 70% dos alunos de ambas as séries já utilizaram o computador da escola. Porém, quando perguntado oralmente quantas vezes, todos que

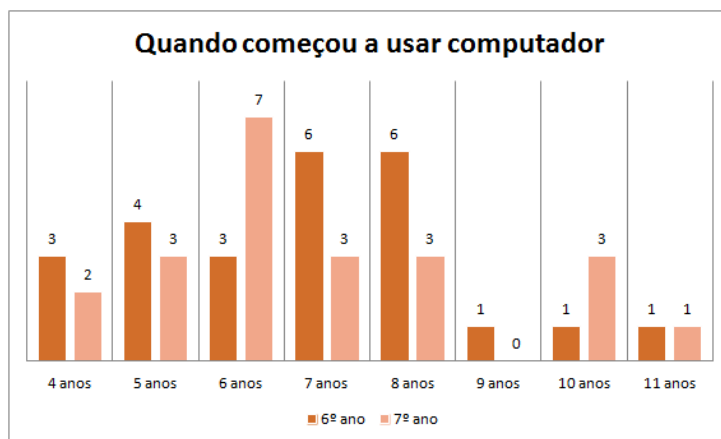
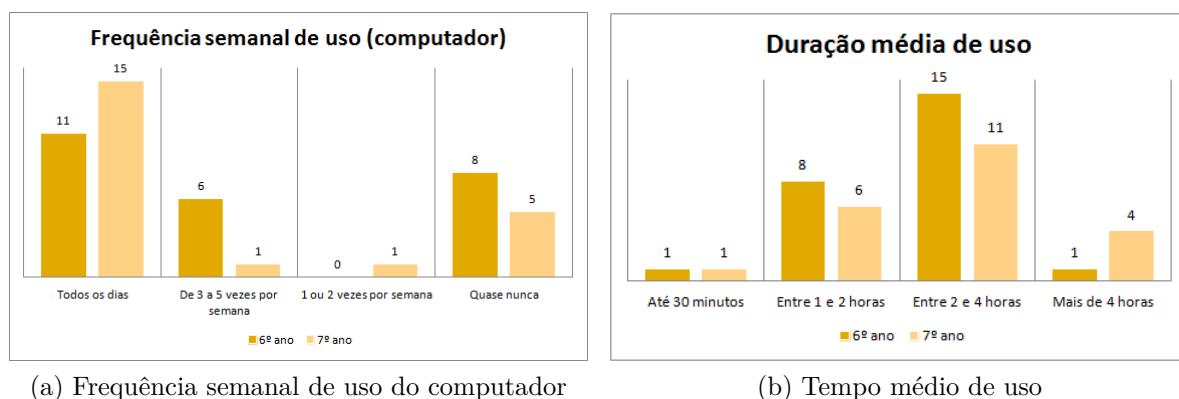


Figura 9: Idade quando começou a usar o computador

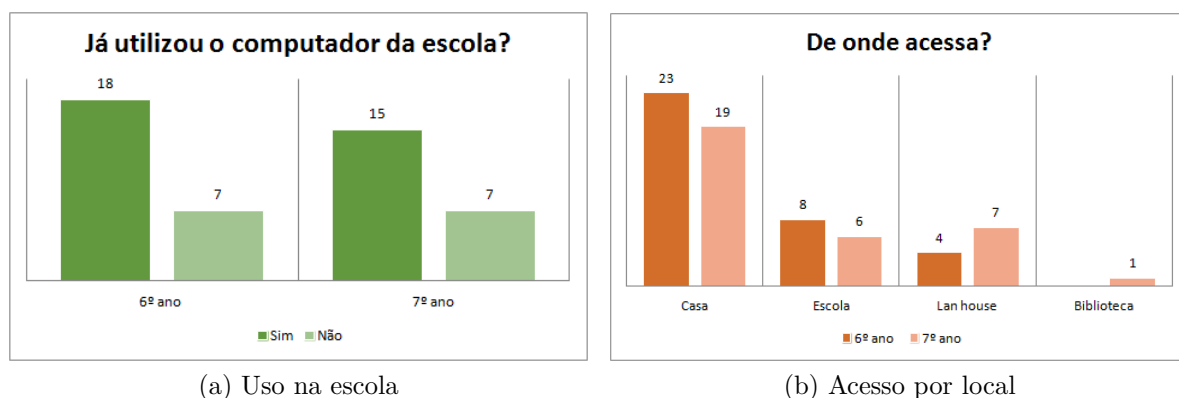


(a) Frequência semanal de uso do computador

(b) Tempo médio de uso

Figura 10

havia entrado naquela sala responderam que isso ocorreu de duas a quatro vezes neste ano.



(a) Uso na escola

(b) Acesso por local

Figura 11: Locais onde acessa

A Figura 11 [b] também engloba o local físico de onde eles costumam acessar a internet. Foi indicado que eles só assinalassem no máximo dois locais, os mais frequentes.

Embora 44 estudantes tenham alegado possuir internet em casa (Figura 8), dois desses não a utilizam, utilizando a escola e *lan houses* como alternativas. Somente um

aluno preencheu a opção “outros” acrescentando biblioteca.

Apesar de afirmarem, que só visitaram a sala do ACESSA no máximo quatro vezes naquele ano, a opção “escola” ainda se classificou como um local acessado por 32% e 27% dos alunos do 6º e 7º anos respectivamente. Isso se explica, devido a vários deles possuírem um *smartphone* e utilizarem seus próprios planos de internet durante os intervalos.

Parte II

As aulas

## 4 Módulo I - *Conhecendo a Computação*

### 4.1 Aula 1: *apresentações, Computação e a história do computador*

Esta primeira aula teve como foco conhecer brevemente os estudantes, seus hábitos e conhecimentos de informática, se eles sabem o que é programação e quando surgiu o computador.

#### Objetivos

- Observar a grande influência que a tecnologia exerce em nosso cotidiano;
- Conhecer a definição de **Computação**;
- Entender a relação entre o computador e contas aritméticas;
- Conhecer a origem do computador;
- Compreender os problemas e as limitações enfrentados em diferentes épocas;
- Observar a evolução das máquinas de calcular.

#### 4.1.1 Apresentação e questionário

Cada um dos alunos recebeu um questionário (Apêndice B) para completar durante a aula. Ele foi formulado a fim de coletar informações sobre aspectos tecnológicos presentes no cotidiano dos estudantes, para se obter uma base de pesquisas, que pode apontar, por exemplo: quais seus hábitos de uso do computador, para qual finalidade eles o utilizam e qual a sua familiaridade com o mundo digital.

As questões foram ditadas em voz alta e de forma espaçada, disponibilizando tempo suficiente de resposta para cada uma. A última pergunta foi a única dissertativa, que serviu de gancho para a atividade seguinte. Solicitou-se que apenas respondessem a esta questão ao final da próxima atividade.

#### Compilação das respostas

A compilação da primeira parte do questionário, foi alocada na seção 3.3.2, salientando um aspecto mais socioeconômico e a facilidade de acesso ao computador. A segunda parte relaciona-se mais com outras atividades desta aula e localiza-se a seguir.



Nas próximas duas questões, eles foram indagados sobre como utilizam seu tempo tanto com os programas de computador, como com aplicativos de celular. Buscando fazê-los começar a se indagar sobre a maneira que usam a tecnologia e para quê ela serve.

No gráfico [a] da Figura 12, nota-se que as atividades campeãs de público são assistir a vídeos no YouTube e navegar nas redes sociais, com destaque para o Facebook. A partir do gráfico [b], observa-se que dentre os aplicativos de celular, o YouTube, o Facebook e o Whatsapp dominam a atenção deles, seguidos por aplicativos de jogos e pela rede social Instagram. Isso mostra a imprescindibilidade da tecnologia para comunicação, entretenimento e compartilhamento de informação para eles.

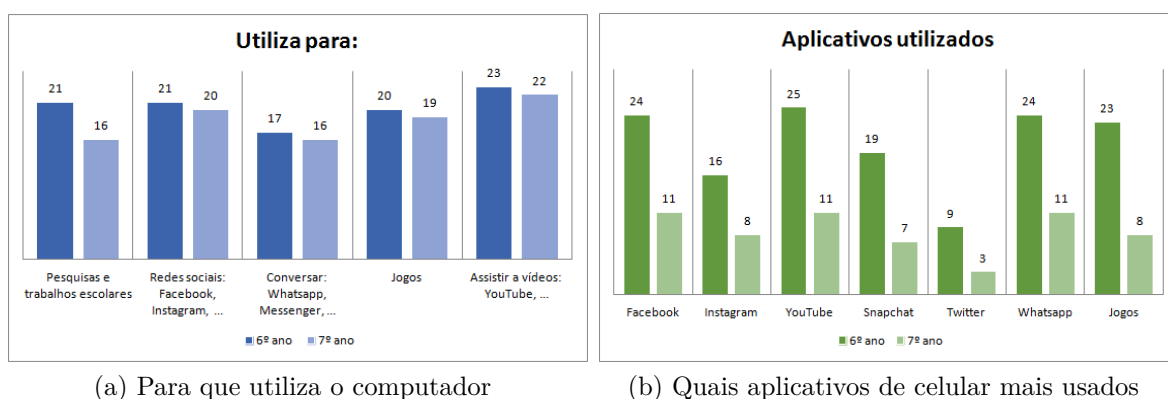


Figura 12

Nos gráficos da Figura 13, perguntou-se se eles possuem alguma conta de e-mail e se era comum acessarem-na semanalmente. Esta informação é relevante para as próximas aulas, nas quais serão criadas contas nas plataformas: Code.org, Scratch e App Inventor e o campo de e-mail é um requisito.

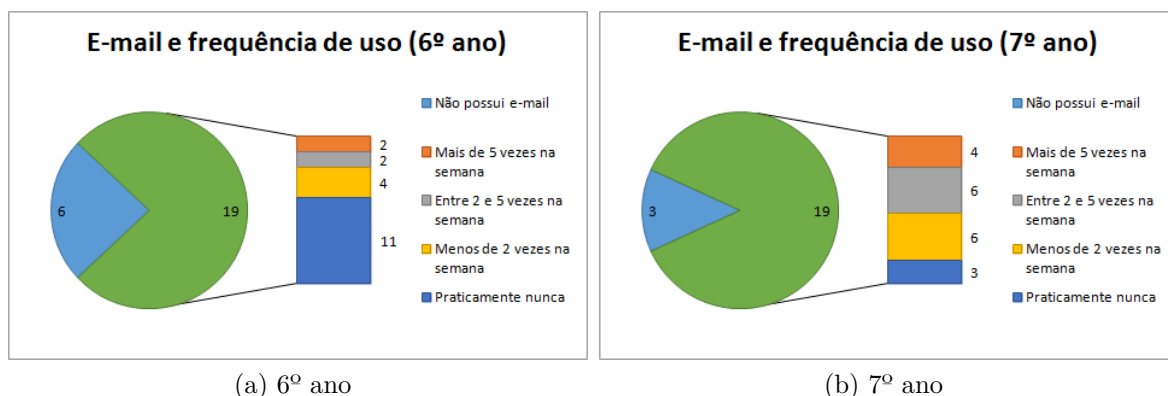


Figura 13: Utiliza o e-mail com que frequência

Esta pergunta visa a medir levemente seu entusiasmo com a proposta de algumas aulas futuras: a criação de jogos. No gráfico da Figura 14, observa-se o 6º ano bastante entusiasmado com essa possibilidade e o 7º ano com opiniões bem divididas. Quem responde “não”, em geral, tenta se justificar falando que prefere brincar com os jogos a criá-los.

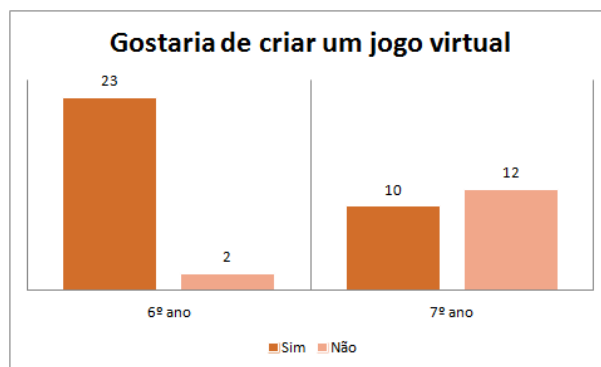


Figura 14: Gostaria de criar seu próprio jogo de computador

### Questão dissertativa

Seguem algumas respostas da última pergunta do questionário (“Se você pudesse criar um aplicativo de celular, você gostaria que ele fizesse o quê?”), sucedidas por suas transcrições:

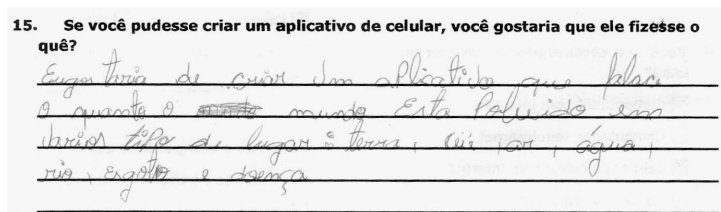


Figura 15

Transcrição da Figura 15:

“Eu gostaria de criar um aplicativo que falasse o quanto o mundo está poluído em vários tipos de lugar: terra, céu, ar, água, rio, esgoto e doença.”

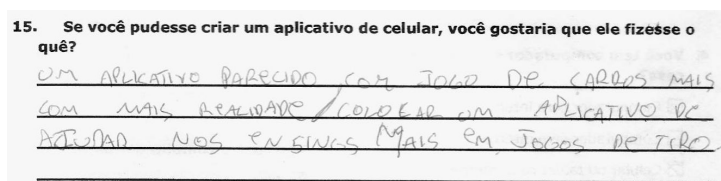


Figura 16

Transcrição da Figura 16:

“Um aplicativo parecido com jogo de carros, mas com mais realidade. Colocar um aplicativo de ajuda nos *engines*, mais em jogos de tiro.”

Transcrição da Figura 17:

“Um jogo que você cria seus personagens e sua casa, sua família e outras coisas.

Um aplicativo que tem um monte de teclados, papéis de parede e os apps ficar com temas.”

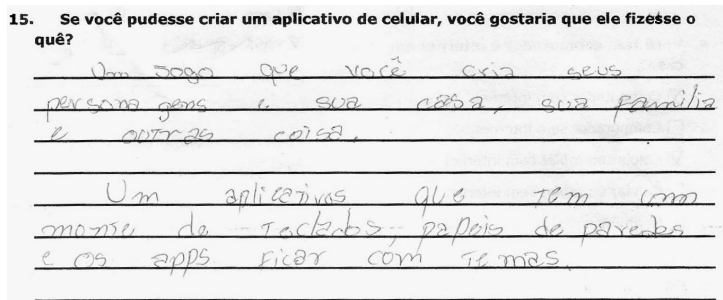


Figura 17

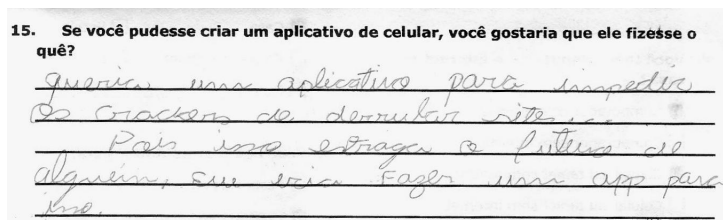


Figura 18

Transcrição da Figura 18:

“Queria um aplicativo para impedir os crackers de derrubar sites...

Pois isso estraga o futuro de alguém. Eu iria fazer um app para isso.”

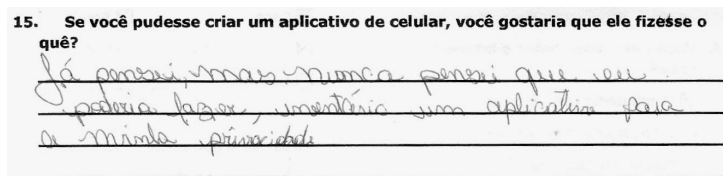


Figura 19

Transcrição da Figura 19:

“Já pensei, mas nunca pensei que eu poderia fazer. Inventaria um aplicativo para minha privacidade.”

### Respostas frequentes

A resposta que mais apareceu foi a da Figura 20, transcrita da seguinte maneira: “Eu faria um aplicativo de ajuda a fazer lições de casa.”, e suas derivações lexicais, como: deveres, provas e trabalhos da escola.

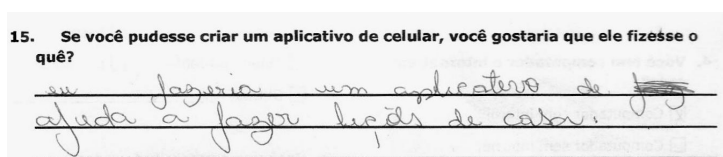


Figura 20: Lição de casa

As duas próximas representam o segundo tipo de resposta mais comum separadas por gênero, evidenciando algumas de suas diferenças comportamentais e sociais.

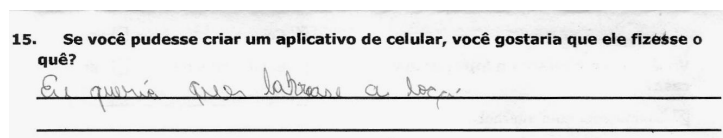


Figura 21: Resposta comum entre as meninas

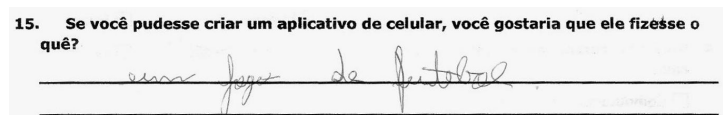


Figura 22: Resposta comum entre os meninos

Transcrições dos textos:

- Gênero feminino (Figura 21):

“Eu queria que lavasse a louça.”

Além de similares, envolvendo: limpar e lavar a casa;

- Gênero masculino (Figura 22):

“Um jogo de futebol.”

Respostas similares incluem termos como: jogo de animais, monstros; jogo com zumbis especiais; jogo legal, cooperativo e competitivo; jogo estilo Minecraft ou Pokémon Go.

#### 4.1.2 Roda de conversa

A proposta desta atividade é observar o conhecimento dos alunos sobre tecnologia, programação e Computação. Nela, manteve-se um aspecto descontraído, deixando os alunos à vontade para se expressarem, sem se sentirem pressionados a falar.

##### Tecnologia

Um ponto abordado foi a familiaridade deles com tecnologia, ressaltando a sua importância e formas de uso. Inicialmente, foi perguntado: “quais ações do dia a dia são ou podem ser controladas por computadores?”.

Em algumas turmas houve certa dificuldade com esta questão, necessitando, assim, de um direcionamento, como, por exemplo, indicações de setores: transporte, saúde, comunicação etc. Para complementar a discussão, buscou-se apresentar exemplos atraentes em diversas áreas, alguns tópicos citados foram: sondas espaciais, veículos que dirigem sozinhos; cirurgias médicas à distância, órgãos artificiais; comunicação imediata.

## Programação

Outro ponto presente foi a parte de programação: se eles sabem o que é, se conhecem alguém que trabalha com isso e, principalmente, foi perguntado o que eles gostariam de ensinar o computador a fazer por eles, mostrando-lhes que essa possibilidade existe.

Assim, dentre a profusão de ideias surgidas, eles deveriam refletir melhor e responder à última pergunta do questionário, dessa vez, com mais exemplos em mente sobre aplicações da tecnologia.

## Computação

Ao final, pergunta-se a eles se sabem qual a definição de Computação e para que ela serve. Com a diminuição dos palpites, as respostas devem ser dadas, de forma a explicar que a origem do termo Computação significa contagem, cálculo; uma operação aritmética ou lógica feita com operações pré-definidas. Assim, lhes perguntamos: “quando o homem começou a contar as coisas?”, fazendo uma conexão com a atividade seguinte.

### 4.1.3 História do computador

Neste momento da aula foi relatada brevemente a história da Computação, para os alunos se familiarizarem com a origem do computador, bem como situar o avanço da tecnologia através do tempo, constatando as dificuldades industriais e tecnológicas de cada época.

#### Surgimento do cálculo

A contagem e o uso de operações aritméticas surgiu na Antiguidade, quando o homem começou a sentir a necessidade de contabilizar pessoas, animais e seus pertences. A fim de registrar e facilitar os cálculos, diversos recursos foram utilizados no decorrer dos séculos: além dos dedos, eram utilizados pedras, gravetos, ossos etc.

É interessante não tornar essa aula completamente expositiva e, frequentemente, abrir brechas para os alunos participarem, deixando-os levantar suposições sobre como os povos primitivos contabilizavam pequenas quantidades, para controlar um rebanho, por exemplo.

Séculos depois, surgiu a primeira engenhoca de calcular: o ábaco. Questionou-se se eles já ouviram falar desse aparelho e se conhecem seu funcionamento, simples e repetitivo de se fazer contas.

Finalmente, foi explicada a relação entre o computador e as operações aritméticas: o cerne do computador encontra-se na ULA (unidade lógica e aritmética), onde ocorrem

inúmeros cálculos extremamente simples de lógica e aritmética, somando zeros com uns, por exemplo.

### Calculadoras Mecânicas

Da Antiguidade damos um salto até o século XVII, quando começa a surgir o ancestral do computador: a calculadora. Por volta do ano 1640, o matemático Blaise Pascal inventa a primeira calculadora mecânica (Pascalina, ilustrada na Figura 23 [a]), que funcionava por meio do movimento de engrenagens e realizava somente as operações de adição e subtração. Nas décadas seguintes as máquinas de calcular foram se aperfeiçoando, ganhando novas operações e tornando-se programáveis.

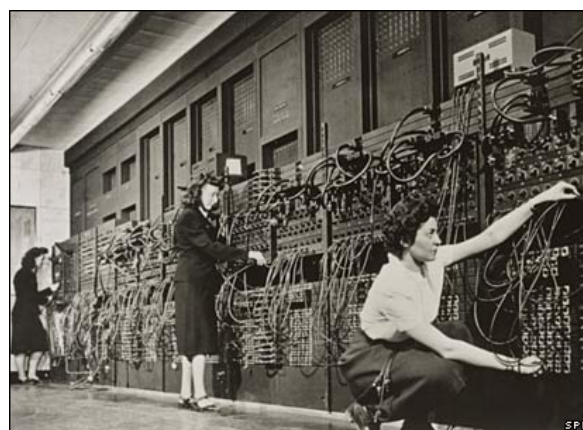
### Computadores eletrônicos

Retomando a interação, perguntou-se se sabem quando os primeiros computadores foram inventados. Explicou-se em seguida que os computadores eletrônicos foram criados por volta da Segunda Guerra Mundial para fins bélicos.

O ENIAC é considerado por muitos como o primeiro computador eletrônico digital de larga escala. Eles foram indagados sobre suas dimensões e, após alguns palpites, mostrou-se a Figura 23 [b], acrescentando que ele pesava 30 toneladas (equivalente a 11 elefantes asiáticos) e que ocupava uma área de 180 metros quadrados (equivalente a cinco salas de aula do Wandyck), buscou-se aqui tornar dimensões mais palpáveis, de modo que pudessem comparar pesos e medidas. Por fim, mencionou-se sobre sua capacidade de cálculo: inferior a uma calculadora atual de bolso.



(a) A Pascalina



(b) ENIAC e programadoras

Figura 23

Ao final da aula, foram disponibilizados alguns vídeos sobre os assuntos vistos para quem se interessar assistir em casa:

- [A história do computador](#)

- [Pascalina](#)
- [O tear de Jacquard](#)

## Observações

Os alunos empolgaram-se, por ser uma aula diferente, na qual puderam frequentar a sala do Acessa e utilizar os computadores, algo que raramente costumam fazer na escola.

Durante a atividade [4.1.1](#) houve algumas dúvidas para preencher o questionário, que foram indagadas na hora ou então percebidas no momento de compilar as respostas:

- Na questão sobre quando nasceram, houve confusão na hora de escrever o dia, mês e ano. Pelo menos três alunos alternaram alguns deles, fazendo surgir os meses: 25 e 2002, por exemplo;
- Muitos não tinham certeza sobre quando começaram a usar o computador pela primeira vez, assim houve vários palpites e aproximações;
- Na questão sobre a frequência com que utilizam o computador, muitos dos que preencheram a opção “quase nunca”, marcaram também a opção “todos os dias” e acrescentaram a palavra “celular” na frente. Talvez seja interessante dar um destaque maior para o celular em uma próxima pesquisa;
- Na questão seguinte, foi solicitado que acrescentassem na hora a opção “mais de quatro horas”, assim que alguns comentaram esse valor;
- Aparentemente a única vez que uma das turmas utilizou a sala do Acessa naquele ano foi para fazer um relatório de matemática;
- Na questão sobre o local onde acessam, foi pedido que ignorassem a opção “celular ou *tablet*”, ao ver que muitos se mostraram confusos por não se tratar de uma localidade física.

A atividade [4.1.2](#) foi bem dinâmica, com a maioria dos alunos tentando responder em voz alta às questões, alguns inclusive aos berros, devido à empolgação e a vontade de ser escutado, tanto que várias vezes foi preciso acalmá-los. Além disso, alguns pareciam bastante receosos e inseguros. Procurou-se, então, incentivar estes a responder alguma pergunta ou outra, na tentativa (muitas vezes sem sucesso) de quebrar a timidez. A resposta de todas as turmas é a que faz parte do senso comum: que Computação é quando fazemos qualquer tarefa utilizando um computador. Um palpite interessante foi o de um garoto do 7º ano, arriscando que é construir máquinas para elas fazerem o nosso trabalho, misturando a definição de Computação com a de Robótica.

## 5 Módulo II - *LightBot*

### 5.1 Aula 2: *guiando os robôs*

Nesta aula são ensinados os conceitos de instruções e algoritmo, com os quais eles devem comandar robôs através de diferentes labirintos até chegar no destino esperado.

#### Objetivos

- Definir de modo simples o que é um robô;
- Aprender o conceito de **instrução**;
- Aprender a definição de **algoritmo**;
- Expressar a ideia de movimento por meio de uma sequência de comandos;
- Definir basicamente o que é um *bug*;
- Mostrar que computadores possuem comandos limitados;
- Introduzir o conceito de **programa**.

#### 5.1.1 Atividade desplugada: guiando o computador

Inicialmente, o professor pergunta se eles conseguem *definir o que é um robô* e ouve as diversas respostas. Em seguida, explica que é um mecanismo que realiza operações pré-definidas, ordenadas e repetitivas e que, de repente, ele acabou de se tornar um.

Passando a desempenhar esse papel, ele desafia os alunos a guiarem-no até a porta da sala por meio de comandos específicos. Ele só está programado para entender três comandos. Dessa forma, pede aos alunos que tentem descobrir quais são eles para conseguirem cumprir a tarefa.

As instruções aceitas são:

- **F** (frente): o robô dá um passo para a sua frente;
- **D** (direita): o robô vira 90 graus para a sua direita;
- **E** (esquerda): o robô vira 90 graus para a sua esquerda.



Ao descobrirem todas elas, o robô demonstra o que cada uma faz e, antes de iniciar a atividade, pede ajuda de dois voluntários: um irá escrever na lousa os comandos executados pelo robô (**F**, **D** ou **E**) e o outro ficará encarregado de usar as **cartas bug**. O professor entrega três cartas para esse aluno e explica quando ele deve utilizá-las.

*Bug* é uma falha ou erro no código de um programa cujo resultado não era bem o esperado, provocando seu mau funcionamento. Assim, se um dos comandos que os alunos passaram para o robô resultar em uma ação inesperada como: virar para o lado errado ou colidir com uma parede, a **carta bug** pode ser acionada, levando o robô a desfazer sua última ação. Esse recurso só pode ser usado até três vezes.

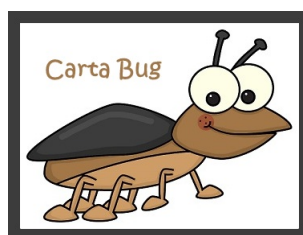


Figura 24: Modelo de **carta bug**

Com tudo estabelecido, a atividade começa: o robô se posiciona a alguns metros da porta olhando para outra direção e conforme os alunos forem falando os comandos, o robô os executa. Lembrando que um aluno anota na lousa a ordem passada e outro, se perceber que o comando escolhido não resulta no movimento esperado, pode usar a **carta bug**, para o robô cancelar sua última movimentação.

Ao chegar na porta, o professor pede aos alunos que observem o registro das instruções que foi anotado na lousa, explicando que eles acabaram de fazer um **algoritmo**, ou seja, uma série de instruções ordenadas. É interessante fazê-los perceber que diferentes algoritmos poderiam ser anotados e que atingiriam o mesmo objetivo de levar o robô do ponto inicial até a porta.

Neste momento, pode-se refazer a atividade em um ponto inicial diferente. Além disso, é interessante deixar que uma das crianças assuma o papel do robô, papel que elas costumam se mostrar animadas em desempenhar.

### 5.1.2 Desafio LightBot

Divididos em duplas, os alunos acessaram o seguinte *link*: <https://lightbot.com/hocflash.html>, para iniciar a atividade, envolvendo o jogo LightBot, detalhado na seção 1.1. Solicitou-se que eles fossem passando de fase, até a mais avançada que conseguissem chegar.

No início, foi dada a explicação sobre o funcionamento do jogo e demonstrou-se como completar a fase inicial. Comentando também que os algoritmos para o robô andar

seriam escritos de maneira que o computador entendesse, diferentemente daqueles escritos na lousa na atividade anterior, formando, assim, um **programa de computador**.

A partir desse momento, os alunos ficaram livres para tentar passar por conta própria as fases seguintes. Mesmo assim, o professor manteve-se à disposição para sanar eventuais dúvidas, por exemplo, para explicar novas instruções, incorporadas conforme o avanço das fases. Descreve-se abaixo em quais momentos do jogo cada instrução é introduzida, para que o professor a explique aos alunos.

- Fase 1-1: desde o início o robô possui dois comandos: mover para a frente e acender (ou apagar) um ladrilho;
- Fase 1-2: comandos de rotação: virar 90° para a esquerda e virar 90° para a direita;
- Fase 1-3: comando relacionado à altitude do personagem, fazendo-o subir ou descer um degrau à sua frente;
- Fase 2-1: no segundo módulo aparece o conceito de procedimento, que é efetuado com a instrução P1;
- Fase 2-4: nesta fase surge a necessidade de criar duas sub-rotinas e, assim, o comando P2 é incorporado junto aos demais.



Figura 25: Comandos do LightBot em ordem de aparição

A partir deste ponto não há novos comandos, porém um novo conceito será introduzido: a recursão, porém é explorado no jogo na forma de laço. Remetendo ao obsoleto comando *goto*, criando *loops* infinitos e sem abordar a condição de saída do laço.

## Observações

A atividade 5.1.1 foi breve, variando de 10 a 15 minutos de duração. Apesar disso, foi bastante agitada, com alguns alunos bem impetuosos para assumir um dos papéis de apoio ou então ordenar os comandos ao robô.

A **carta bug** não chegou a ser utilizada pelas turmas do sétimo ano, deixando seus portadores visivelmente frustrados. Já no sexto ano, ela foi acionada uma vez em uma turma e duas na outra, todas por girar no sentido contrário.

Em apenas uma turma os alunos não quiseram interpretar o papel de robô programável. Em todas as outras, porém, quem se voluntariou demonstrou bastante entusiasmo ao interpretá-lo.

Os alunos se ocuparam até o final da aula com o jogo LightBot, durante o qual diversas duplas apresentaram as **dificuldades** listadas adiante.

- Virar para o lado correto:

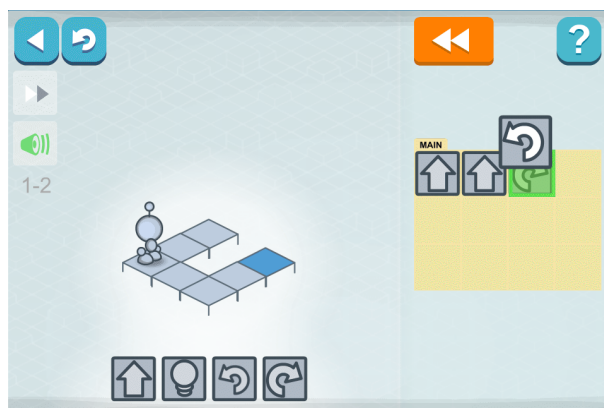


Figura 26: Fase 1-2

Logo na segunda fase eles se depararam com este desafio, o qual diversas vezes foi resolvido na base da tentativa e erro, com os alunos escolhendo aleatoriamente um dos lados, testando e dependendo do resultado, substituindo pelo outro lado, conforme consta na Figura 26. Muitos porém, giravam na cadeira, inclinavam a cabeça, gesticulavam com as mãos, ficavam em pé e viravam para os lados, na busca de encontrar o lado correto.

- Subir um degrau e andar:

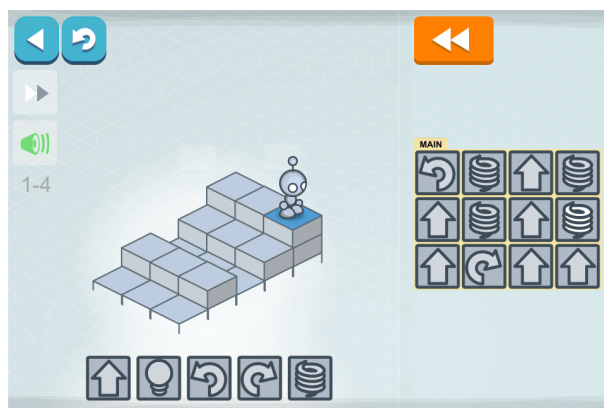


Figura 27: Fase 1-4

Assim que o comando do degrau foi introduzido, diversos alunos não associaram, que ao utilizá-lo, o robô além de subir ou descer já se move para a frente. Assim, combinavam esses comandos, ocupando um espaço extra a cada tentativa, conforme

consta na Figura 27. Este erro era percebido ao ocuparem todos os espaços disponíveis, constatando que eles não seriam suficiente para concluir seus algoritmos.

- Algoritmos corretos, porém demasiado extensos:

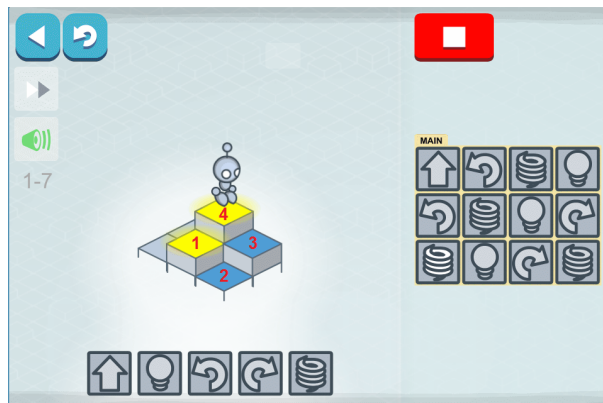


Figura 28: Fase 1-7

Algumas duplas se irritaram com esta fase, pois pensaram no algoritmo mostrado na Figura 28, em que o robô saltita pelos ladrilhos  $1 \rightarrow 4 \rightarrow 3 \rightarrow 2$ , faltando apenas um comando para completarem a fase, alegando que o jogo estava errado. Neste momento, foi explicado que eles precisariam fazer um caminho diferente, mais curto, para conseguir finalizar ( $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$ , no caso). Eles tentaram novamente, mas um pouco desanimados. No fim, todos concluíram este desafio, seja refazendo e conseguindo, seja apelando para as duplas ao lado.

- Girar 180°:

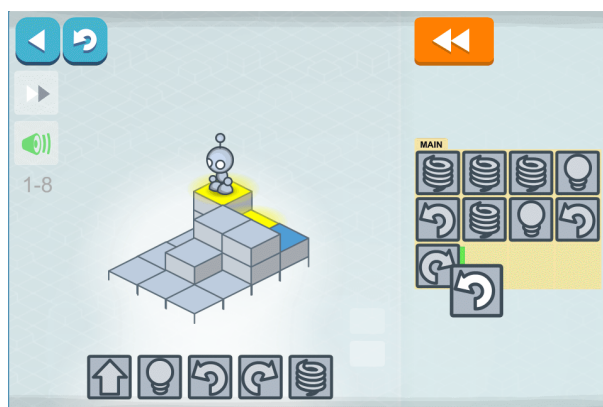
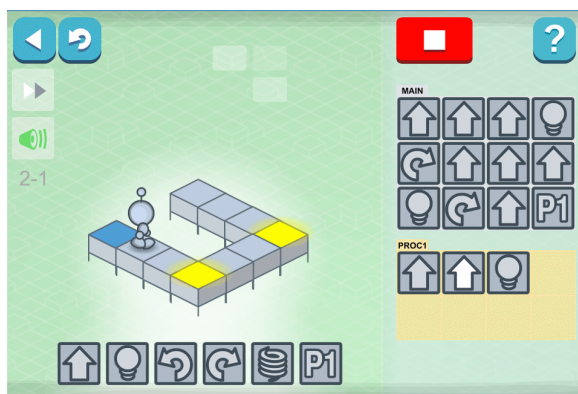


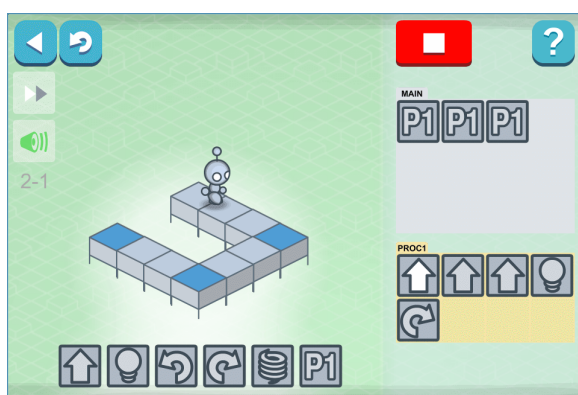
Figura 29: Fase 1-8

Houve certa dúvida quando surgiu a necessidade de se girar o personagem a 180°. A Figura 29 mostra um exemplo das várias setas que eles colocavam, inclusive uma de virar 90° para a direita, imediatamente seguida por uma para a esquerda. Nas turmas do sexto, solicitou-se um voluntário para participar da demonstração desse conceito. Ela se deu nos moldes da atividade anterior, porém o aluno foi posicionado de costas para a porta e, assim, os outros começaram a direcioná-lo.

- Assimilar a finalidade de um procedimento:



(a)



(b)

Figura 30: Fase 2-1

Quando o conceito de procedimento foi introduzido, vários o utilizaram como uma extensão do *Main*, como mostrado na Figura 30 [a], sem explorar sua característica fundamental: a reutilização. Um exemplo desse reuso apresenta-se na Figura 30 [b], notando-se também a redução do código pela metade. Contudo, este conceito não foi avaliado, apenas foi apontada a possibilidade de se reduzir a quantidade de comandos para os mais curiosos.

A maior parte dos alunos terminou a fase 2-3 e ficou empacada na seguinte e, apenas duas duplas chegaram na fase 2-5. Sugeriu-se que tentassem terminar todas as fases em casa e que trouxessem as dúvidas na semana seguinte.

## 5.2 Aula 3: atividade diagnóstica I

### Objetivos

- Avaliar se os conceitos de **instrução** e **algoritmo** foram fixados;
- Mostrar que diferentes algoritmos podem levar ao mesmo resultado;

- Estimular a troca e a construção coletiva do conhecimento.

## Atividade diagnóstica: LightBot

A ficha para realizar esta atividade encontra-se no Apêndice C, assim como algumas resolvidas com casos mais interessantes. Nesta aula, cada um recebeu uma caneta, um tubo de cola branca, dezenas de comandos do LightBot, impressos e recortados, e uma ficha com os seguintes conteúdos: dois campos para serem preenchidos com o nome e a turma, uma imagem de uma nova fase do LightBot, instruções e uma área com doze espaços numerados.

Cada aluno deveria tentar resolver o desafio de sua ficha (há dois tipos de fichas, com labirintos do LightBot diferentes, vistos na aula 5.1.2). Primeiramente, eles foram orientados a escrever os comandos necessários em ordem nos quadrados na área de resposta e, quando estivessem certos do algoritmo, deveriam colar os respectivos comandos impressos e recortados no lugar adequado.

### Correção pareada

Assim que quaisquer dois alunos terminavam a atividade 5.2, eram orientados a sentarem-se em duplas. Eles deveriam trocar as fichas entre si e cada um deveria corrigir a do outro.

Os alunos foram orientados a procurar todos os *bugs* (os erros que encontrassem) no código do outro e, assim que ambos terminassem de corrigir, deveriam mostrar os apontamentos que fizeram, contudo sem mencionar qual a natureza dos *bugs*, deixando para o próprio autor a oportunidade de identificá-los e, então, corrigir seu código.

Ambas as fichas possibilitavam mais do que um algoritmo para resolver o desafio. Após as correções, foram solicitados dois voluntários com uma ficha de cada para escrever seus códigos na lousa. Enquanto isso, o resto da turma deveria avisar se tivesse feito algum algoritmo diferente daqueles. Em apenas uma das turmas não houve variedade de códigos. No Apêndice C encontram-se quatro fichas resolvidas deste exercício, exemplificando a diversidade de códigos mencionada anteriormente.

### Observações

A primeira atividade durou de 15 a 25 minutos, com algumas duplas demorando bastante para encontrar os comandos que precisavam e outras para colar as instruções. Durante a correção pareada a duração variou ainda mais: para as duplas em que ambas as fichas estavam corretas, ela foi rápida, porém, nas fichas que continham erros, uma minoria teve bastante dificuldade para encontrá-los, chegando a entregar a ficha no final da aula sem as devidas correções. Mais adiante nas páginas 48 e 49 tem-se dois exemplares

de códigos entregues sem corrigir as falhas apontadas, ambos contendo erros comuns: virar para o sentido contrário e andar mais ladrilhos do que o necessário. Houve também diversos casos em que o corretor da dupla não conseguia se conter e acabava mostrando para o parceiro como resolver os *bugs*.

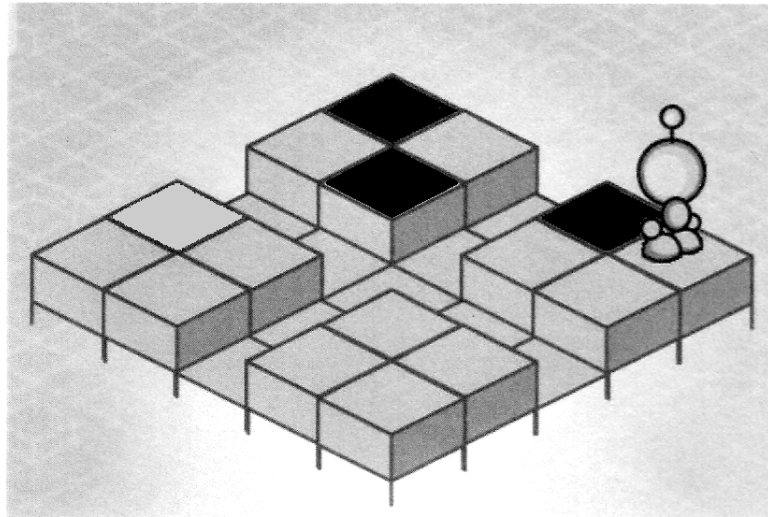
Durante a correção da atividade 5.2 percebeu-se que poderia haver na ficha um campo adicional para o corretor da questão incluir seu nome. Além disso, seria mais eficiente ter padronizado a correção com eles, pois cada um corrigiu do seu jeito particular: com xis grandes ou pequenos, círculos e setas ao redor do erro, marcações por cima, embaixo ou ao lado do desenho, alguns arrancaram os comandos colados que estavam errados, alguns desenharam um campo extra no meio de outros para colar um comando faltante sem precisar alterar todos, enfim, houve de tudo.

Nome: Sorella C.

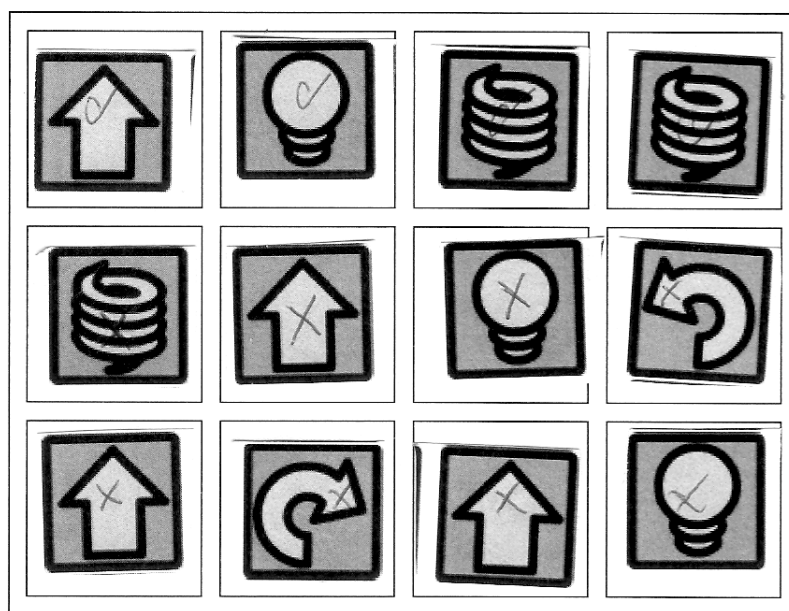
Turma: 6<sup>o</sup> B

### Programando o robô

- Ajude o robô a acender todos os quadrados pretos no caminho abaixo.



- Use no máximo doze comandos.
- Quando tiver certeza dos comandos, cole-os em ordem aqui embaixo.
- Você **não** precisa usar todos os comandos.



X

X



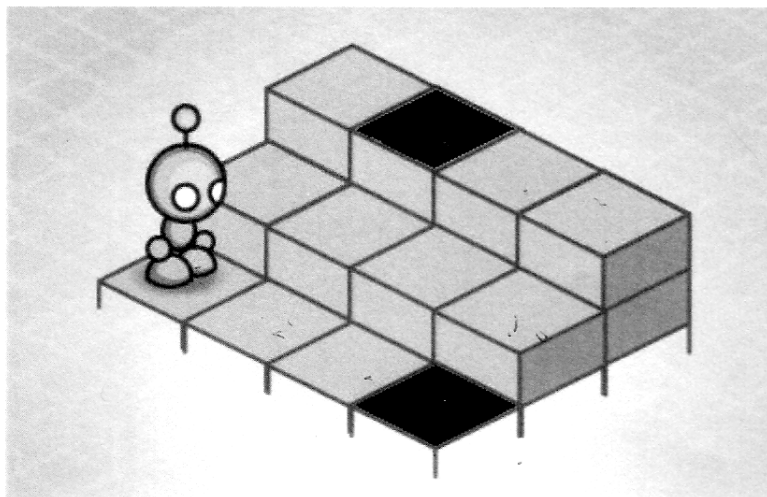
Nome: Geovanna Russo

Turma: 6º 2

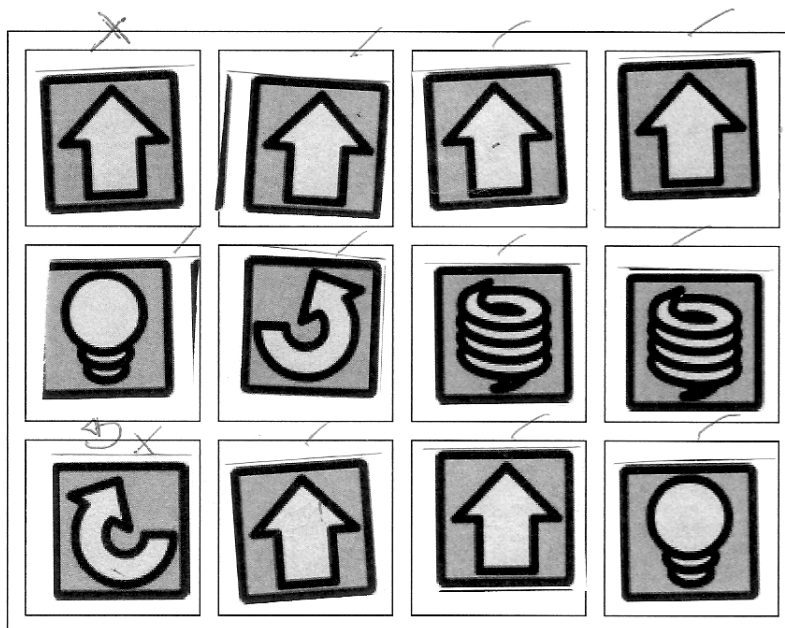
Geovanna  
Russo

### Programando o robô

- Ajude o robô a acender todos os quadrados pretos no caminho abaixo.



- Use no máximo doze comandos.
- Quando tiver certeza dos comandos, cole-os em ordem aqui embaixo.
- Você **não** precisa usar todos os comandos.



## 6 Módulo III - Code.org

### 6.1 Aula 4: *labirinto*

Nesta aula, os alunos deixam de lado o LightBot e começam a trabalhar com comandos descritos na forma de texto, em vez de ícones intuitivos. A plataforma em questão é o Code.org, detalhada na seção 1.2.

#### Objetivos

- Conhecer uma linguagem de programação em blocos;
- Expressar movimento por meio de uma sequência de comandos;
- Representar um algoritmo como um programa de computador;
- Aplicar técnicas de programação pareada para completar as tarefas de maneira colaborativa.

#### Área do professor

O Code.org disponibiliza uma área de controle do professor, para que ele possa acompanhar o progresso dos alunos e de diversas turmas simultâneas, representada na Figura 31. Nela podemos observar o ano escolar das turmas, qual curso estão aprendendo, quantidade de alunos e o tipo de *login* que o aluno deve fazer.

Página inicial do professor > Contas e progresso do aluno

Nova turma

Seção	Tipo de login	Grau	Curso	Fases extras	Alunos	Código da seção	
<b>Wandyck - 6A</b> Progresso Gerenciar alunos	word	6	Curso 2	Sim	15	XTHKQP	<input type="button" value="Editar"/> <input type="button" value="Imprimir certificados"/>
<b>Wandyck - 6B</b> Progresso Gerenciar alunos	word	6	Curso 2	Sim	15	RRVGYV	<input type="button" value="Editar"/> <input type="button" value="Imprimir certificados"/>
<b>Wandyck - 7A</b> Progresso Gerenciar alunos	email	7	Curso 2	Sim	8	CNXGHR	<input type="button" value="Editar"/> <input type="button" value="Imprimir certificados"/>
<b>Wandyck - 7B</b> Progresso Gerenciar alunos	email	7	Curso 2	Sim	10	PKQYMY	<input type="button" value="Editar"/> <input type="button" value="Imprimir certificados"/>

Figura 31: Área de controle do professor

Sobre o login, há três possibilidades diferentes:

- (a) Imagem;
- (b) Palavra;
- (c) E-mail.

Escolheu-se aplicar a segunda opção no sexto ano e a última no sétimo. Nas duas primeiras, o professor inclui o nome de todos os alunos na turma, e cada um recebe uma senha, que será ou uma imagem (opção **a**) ou duas palavras em inglês (opção **b**). Já na opção **c**, cada um dos alunos deve criar sua própria conta, fornecendo nome e e-mail.

Esta escolha foi feita com base nas respostas que eles forneceram sobre como utilizam seus e-mails (Figura 13), que aponta uma frequência maior do sétimo ano utilizando-o.

Ao optar por imagem ou palavra, o site gera uma página para se imprimir um cartão de login para cada aluno, contendo o *link* em que eles precisam entrar e a sua senha, como mostrado na Figura 32.



Figura 32: Cartões de login

Neste projeto, os cartões foram impressos, plastificados e distribuídos para cada um dos respectivos alunos do sexto ano, que foram orientados a guardá-los consigo e sempre levá-los às aulas. No entanto, essa prática mostrou-se deveras ingênua, pois na semana seguinte apenas cerca de 5% dos alunos portavam seus cartões.

### Etapa 3: Labirinto

Nesta aula os alunos tentaram completar a 3ª etapa do curso 2. Cada turma recebe um *link* diferente, gerado na área do professor (seção 6.1). Ilustrada com os personagens do jogo Angry Birds, esta etapa contém 11 desafios ao todo, sendo que nos nove primeiros os estudantes devem montar um algoritmo guiando o personagem através de um labirinto, utilizando as mesmas instruções da atividade 5.1.1:

- Mova para a frente;
- Vire à direita;
- Vire à esquerda.

Dessa vez, porém, o algoritmo será escrito na forma de um programa de computador, para isso, eles devem encaixar os blocos de instruções uns no outros em uma sequência linear ordenada. A dificuldade dos desafios aumenta gradativamente a cada fase e, nas duas últimas, há questões de múltipla escolha para ver se o estudante consegue associar qual dos algoritmos liberta o pássaro do labirinto. Na Figura 33 tem-se um exemplo de uma das fases.

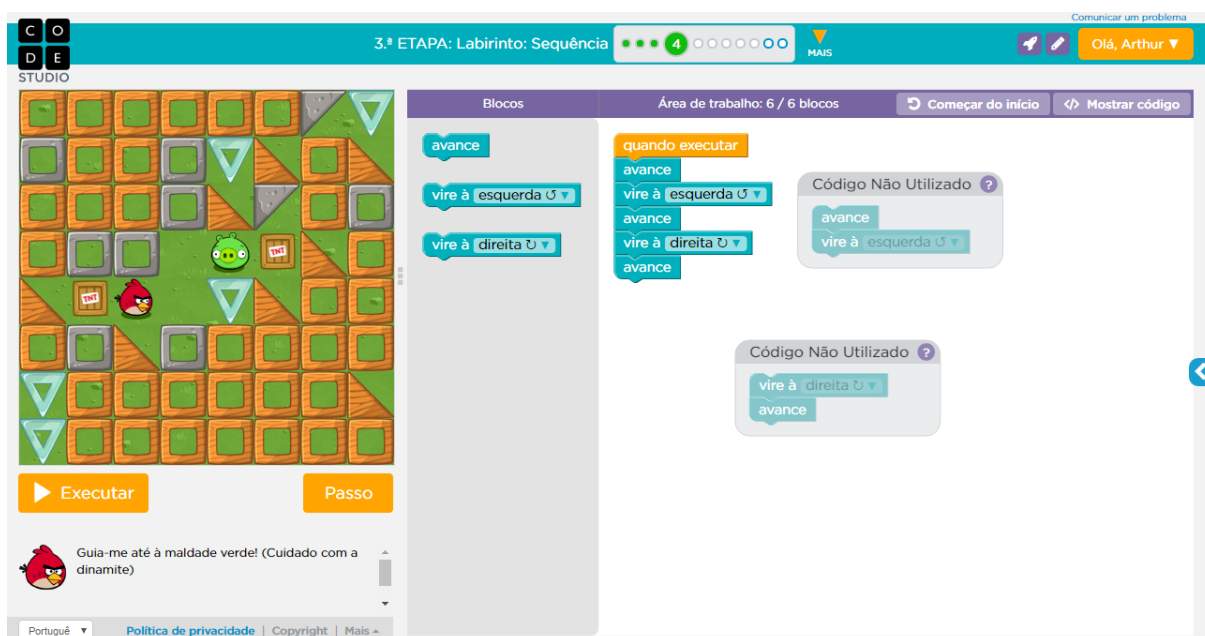


Figura 33: Labirinto: Fase 4

Para iniciar a atividade, o professor perguntou aos alunos se eles conhecem o jogo Angry Birds, em seguida, explicou que eles deveriam escrever diversos programas, de modo a ajudar o pássaro a chegar até o porco. Para isso eles precisaram organizar os blocos em uma sequência bem específica, tentando utilizar a menor quantidade de blocos possível (esse valor aparece na parte superior da área de código).

Nesse ponto, é importante comentar que ao utilizar os comandos de virar à esquerda ou direita, eles devem ter em mente que essa direção refere-se ao ponto de vista do pássaro e não ao deles próprios. Conforme forem avançando nas fases, é interessante retomar a fala da aula anterior, comentando que a partir da quinta fase há diferentes estratégias de caminho que eles devem optar, mostrando que diferentes algoritmos podem resolver um mesmo desafio.

Os alunos foram solicitados a completar as 11 fases durante a aula, e os que não conseguiram foram incentivados a terminá-las em casa.

## Observações

Os alunos em geral apresentaram dificuldades para digitar o *link* da sua turma, frequentemente caindo em algum endereço inexistente. Ainda sobre digitação, observou-se que para o sexto ano foi difícil digitar a senha para acessar o site. Como seu tipo de login era por palavras, a senha gerada para cada aluno consistia em uma combinação de duas palavras em inglês, por exemplo: “those except”, “noun high”, “word only”. Palavras que são, em sua maioria, desconhecidas para eles. Como o foco do projeto não é treinar a capacidade de digitação deles, acredita-se que numa próxima vez seja melhor utilizar o tipo de login por imagem, dessa forma eles precisariam apenas memorizar uma figura semelhante a um *emoji*, exemplos: um feiticeiro, um sapo, um carro etc.

Na área do professor é possível acompanhar o progresso de cada aluno ao longo da etapa. Isso se dá na forma de bolinhas numeradas e coloridas, o número indica a fase e as cores explicam o progresso dos alunos, como ilustrado na Figura 34.

### 6º ano A

Esta turma teve excelente desempenho, mantendo durante a aula um comportamento calmo, porém bastante atento. Logo no início, eles foram incentivados a completar as fases tentando utilizar o limite mínimo de blocos necessários. Duas duplas não conseguiram finalizar os desafios, porém o fizeram em casa.

Há 15 alunos (ao invés dos 14 originais) porque um saiu da escola após a turma já ter sido criada e um outro foi chamado para participar do projeto em seu lugar. A imagem mostra apenas oito alunos passando pelos desafios, neste dia um aluno faltou e eles se dividiram em duplas.

### 6º ano B

Antes de cada aula começar, o professor deve se encaminhar até a sala dos alunos, pedir autorização do professor da classe para liberá-los e, então, descer com eles até a sala do ACESSA. Neste dia, o professor da vez demorou para liberar a turma, comprometendo cerca de 10 minutos da aula para que todos se acomodassem. Além disso, eles estavam mais agitados neste dia, sem prestar muita atenção na explicação.

Assim como no 6º B, um aluno saiu da escola e outro se juntou ao projeto. Apenas uma dupla finalizou a atividade, a segunda que foi mais longe apenas não fez as duas questões de múltipla escolha.

Nome	Progresso Q
Fase 3: Labirinto: Sequência	
Nicolau Pereira	3 1 2 3 4 5 6 7 8 9 10 11
João Pedro	3 1 2 3 4 5 6 7 8 9 10 11
Gustavo dos Santos	3 1 2 3 4 5 6 7 8 9 10 11
Guilherme Teodoro	3 1 2 3 4 5 6 7 8 9 10 11
Kayke Jordan	3 1 2 3 4 5 6 7 8 9 10 11
Matheus Ryan	3 1 2 3 4 5 6 7 8 9 10 11
Maria Eduarda	3 1 2 3 4 5 6 7 8 9 10 11
Livia Regina	3 1 2 3 4 5 6 7 8 9 10 11
Kevin Souza	3 1 2 3 4 5 6 7 8 9 10 11
Geovana Russo	3 1 2 3 4 5 6 7 8 9 10 11
Erick Mark	3 1 2 3 4 5 6 7 8 9 10 11
Eduardo Fernandes	3 1 2 3 4 5 6 7 8 9 10 11
Danilo Souza	3 1 2 3 4 5 6 7 8 9 10 11
Carlos Eduardo	3 1 2 3 4 5 6 7 8 9 10 11
Agata Elen	3 1 2 3 4 5 6 7 8 9 10 11

(a) Turma 6A

Nome	Progresso Q
Fase 3: Labirinto: Sequência	
Vitor Pena	3 1 2 3 4 5 6 7 8 9 10 11
Victória Arruda	3 1 2 3 4 5 6 7 8 9 10 11
Renata Silva	3 1 2 3 4 5 6 7 8 9 10 11
Pedro Henrique	3 1 2 3 4 5 6 7 8 9 10 11
Matheus de Oliveira	3 1 2 3 4 5 6 7 8 9 10 11
Maria Eduarda Borges	3 1 2 3 4 5 6 7 8 9 10 11
Marcela Zembruski	3 1 2 3 4 5 6 7 8 9 10 11
Isabela Ariadina	3 1 2 3 4 5 6 7 8 9 10 11
Ighor Garcia	3 1 2 3 4 5 6 7 8 9 10 11
Guilherme Santos	3 1 2 3 4 5 6 7 8 9 10 11
Guilherme Henrique	3 1 2 3 4 5 6 7 8 9 10 11
Gabriel Silveira	3 1 2 3 4 5 6 7 8 9 10 11
Gabriel Nakamura	3 1 2 3 4 5 6 7 8 9 10 11
Camila Gabriela	3 1 2 3 4 5 6 7 8 9 10 11
Vinicius Poirier	3 1 2 3 4 5 6 7 8 9 10 11

(b) Turma 6B

Nome	Progresso Q
Fase 3: Labirinto: Sequência	
Luiz Gustavo	3 1 2 3 4 5 6 7 8 9 10 11
joao pedro	3 1 2 3 4 5 6 7 8 9 10 11
kaique e murilo	3 1 2 3 4 5 6 7 8 9 10 11
isaac.kaique	3 1 2 3 4 5 6 7 8 9 10 11
diego darcy	3 1 2 3 4 5 6 7 8 9 10 11
Eric w.f.l. jhonatan	3 1 2 3 4 5 6 7 8 9 10 11
kaique murilo	3 1 2 3 4 5 6 7 8 9 10 11
eric e gabriela	3 1 2 3 4 5 6 7 8 9 10 11

(c) Turma 7A

Nome	Progresso Q
Fase 3: Labirinto: Sequência	
Rafael.Joker	3 1 2 3 4 5 6 7 8 9 10 11
lucas alves	3 1 2 3 4 5 6 7 8 9 10 11
diego/arthur	3 1 2 3 4 5 6 7 8 9 10 11
feipe winicius	3 1 2 3 4 5 6 7 8 9 10 11
renato	3 1 2 3 4 5 6 7 8 9 10 11
KAYLANI E MARIA	3 1 2 3 4 5 6 7 8 9 10 11
william e lucas	3 1 2 3 4 5 6 7 8 9 10 11
rayssa	3 1 2 3 4 5 6 7 8 9 10 11
maria e kaylani	3 1 2 3 4 5 6 7 8 9 10 11
Winicius	3 1 2 3 4 5 6 7 8 9 10 11

(d) Turma 7B

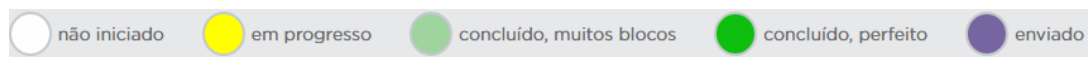


Figura 34: Progresso das turmas na etapa 3: Labirinto

7º ano A

No total, 11 alunos estavam presentes. Como o tipo de login dessa turma era por e-mail, eles mesmos criaram suas contas e acabaram colocando o nome da dupla no campo “nome”. Eles tiveram um ótimo desempenho nesta atividade, com apenas duas duplas que em algumas fases poderiam ter usado menos blocos.

## 7º ano B

No dia, nove alunos estavam presentes. Ao criar a conta, colocaram erroneamente o nome da dupla no *login*, fator que repercutiu na formação das duplas das aulas seguintes. Apenas uma dupla mostrou grande desinteresse, demorando bastante para fazer a atividade. Um aluno queixou-se de seu nome vinculado à dupla e, em sua casa, criou sua própria conta, refazendo todas as fases.

## 6.2 Aula 5: *artista*

### Objetivos

- Observar que o computador obedece às instruções que lhe são passadas ao pé da letra;
- Desenhar triângulos, quadrados e retângulos e observar seus atributos particulares;
- Criar programas utilizando instruções sequenciais para completar imagens;
- Selecionar **argumentos** para alguns comandos
- Diferenciar triângulos, quadrados e retângulos, pela abertura dos ângulos, quantidade e tamanho dos lados;
- Diferenciar um ângulo interno de um ângulo externo em um polígono;
- Juntar ou decompor polígonos e criar novas figuras a partir disso.

### 6.2.1 Atividade desplugada: o computador que desenha

Este exercício baseia-se em uma atividade do *CS Unplugged* (Bell et al. 2011). Primeiramente, iniciou-se uma conversa informal com a turma, explicando que para se programar um computador é necessária uma linguagem própria, com instruções limitadas que ele consegue entender.

Os alunos, então, formaram duplas: um deles recebeu um desenho e o outro, uma folha em branco e uma canetinha colorida porosa. A criança com o desenho foi o “programador” da atividade, enquanto que a outra fez o papel de “computador”. O programador deve descrever a figura (sem mostrá-la) para que o computador tente reproduzi-la no papel em branco. Eles poderiam perguntar coisas entre si, mas sem mostrar qualquer um dos papéis para o parceiro, na tentativa de reproduzir o mais fielmente os desenhos.

Para exemplificar, o professor pegou um desenho e solicitou um voluntário para ir até a lousa tentar reproduzir o desenho conforme as instruções. É interessante procurar

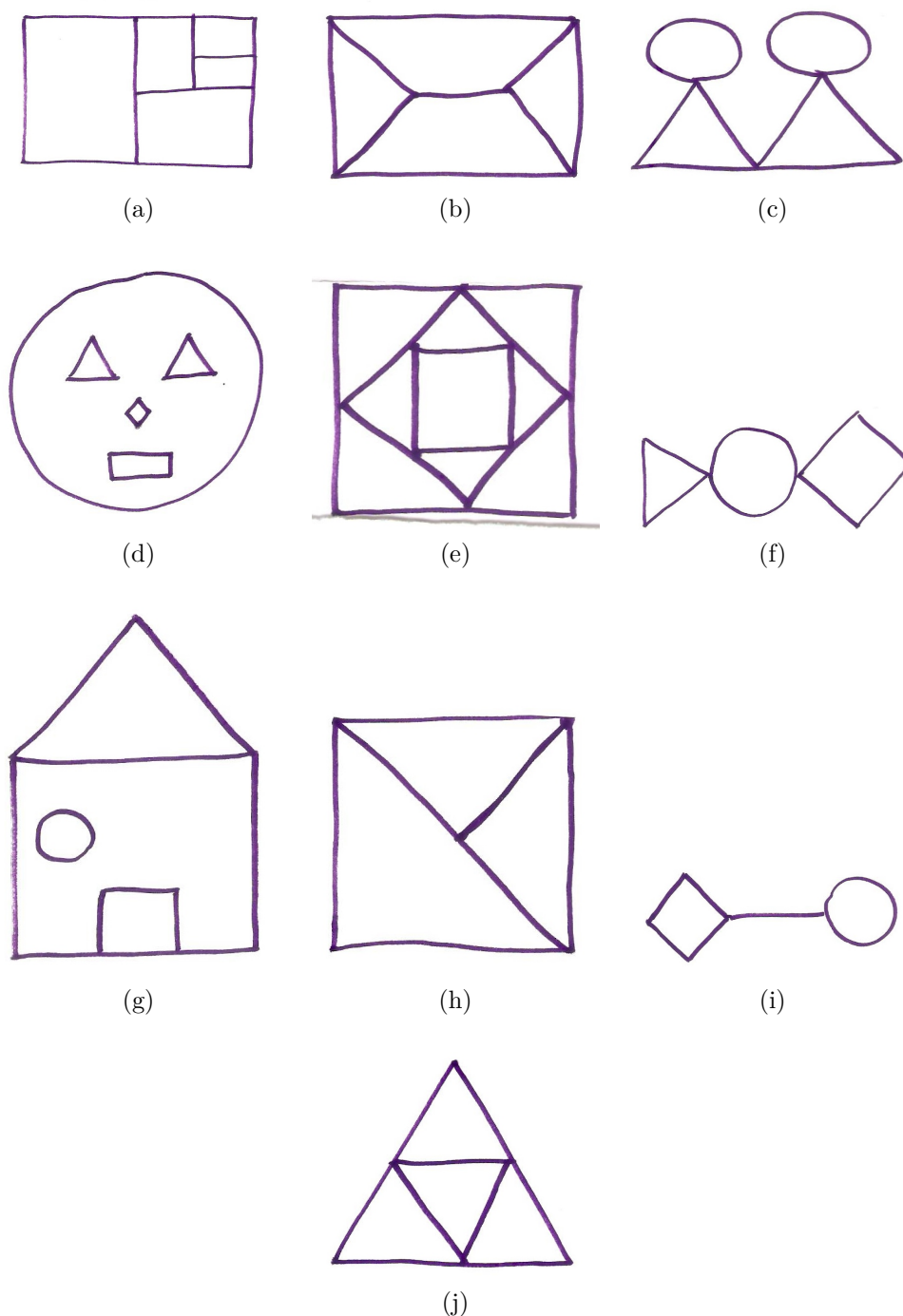


Figura 35: Desenhos-modelo

utilizar os nomes dos polígonos na descrição, explicar o posicionamento da figura, para onde ela aponta, qual o tamanho aproximado etc.

Foram utilizadas, nesta atividade, dez imagens diferentes, representadas na Figura 38, todas contendo ao menos um dos polígonos: triângulo, quadrado, losango e retângulo, que serão retomados na atividade seguinte.

Ao final, os alunos deveriam comparar os desenhos, observando as semelhanças e quais partes não saíram conforme o esperado. Após essa reflexão, os papéis foram invertidos.



dos e um desenho diferente foi entregue ao novo programador. Comentou-se com eles que programadores escrevem seus códigos dessa forma, fornecendo uma série de instruções ao computador e, só depois de rodar o programa, veem de fato seu efeito.

## Explicação de ângulos

Para o sexto ano foi introduzido este conceito ainda não aprendido em sala de aula. Assim, explicou-se o conceito de ângulo, descrevendo-o como o encontro das duas travessas do gol, com uma abertura entre elas. Explicou-se também que as arestas de um polígono também contêm aberturas de ângulos entre si, e que essa abertura é medida em graus e que em um triângulo equilátero todos os ângulos internos medem  $60^\circ$ .

Os alunos do sétimo ano aprenderam esse conteúdo no bimestre anterior, dessa forma, ele apenas foi lembrado, além disso conversou-se um pouco sobre ângulos externos antes de prosseguir para a outra atividade.

### 6.2.2 Code.org: o artista

Novamente os alunos acessaram sua respectiva turma no site [code.org](https://code.org), e começaram a resolver a etapa 4: o artista, na qual vários desafios de desenhos incompletos são fornecidos para serem finalizados utilizando as instruções pré-definidas: mova  $x$  pixels, vire  $x$  graus à direita ou à esquerda. Ao todo são nove desafios, nos quais eles devem programar um código que complete corretamente o desenho, além de uma fase livre para se desenhar à vontade e duas questões de múltipla escolha ao final. No desafio representado na Figura 36 observa-se que o programador deve completar o desenho da pipa, riscando em cima do quadrado faltante.

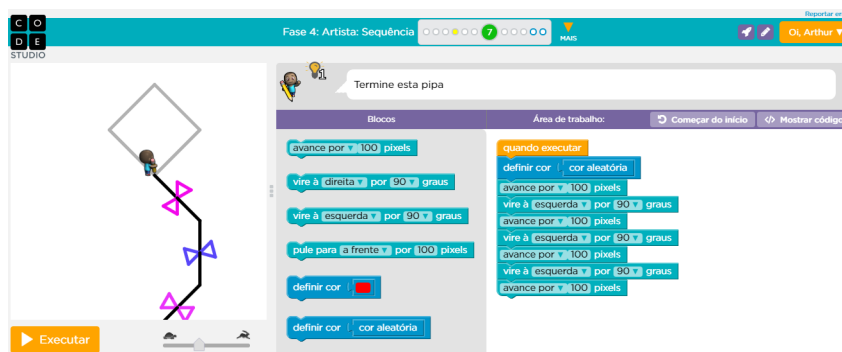


Figura 36: Artista: fase 7

Para ajudá-los em caso de dúvidas sobre quantos graus virar, há um recurso no Code.org que demonstra o que irá ocorrer dependendo do ângulo escolhido, bastando clicar no argumento dos graus para aparecer um quadro ilustrado (como mostrada na Figura 37), facilitando a resolução dos exercícios.

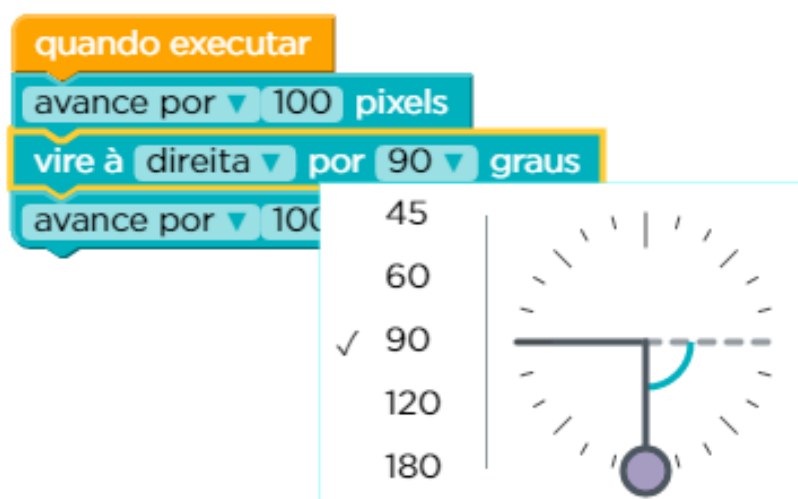


Figura 37: Quadro de apoio para selecionar ângulos

Os alunos foram instruídos a ultrapassar o máximo de fases possível em aula e tentar encerrar toda a etapa 4 em casa.

## Observações

### Atividade 6.2.1

A princípio, eles se entusiasmaram com a atividade “o computador que desenha”, ao verem que receberiam canetinhas coloridas. Apenas em seu decorrer, perceberam a dificuldade do exercício, principalmente os programadores, que deveriam descrever um desenho precisamente apenas utilizando palavras. Em seu desenrolar observou-se que algumas duplas tentavam trapacear, mostrando discretamente o papel para o outro. Apenas uma turma não apresentou esse comportamento.

Os desenhos feitos pelos “computadores” estão representados na Figura 38. Alguns aproximam-se bastante dos originais, como o das letras: [c], [d] e [j]. Já as imagens: [a], [b], [e] e [h] foram as mais diferentes. Os desenhos [d], [g] e [i] foram os mais criativos, apresentando em alguns casos complementos ao desenho original, apresentando os seguintes atributos extras: sorrisos; sol, nuvens, montanha ou chaminé; menina.

### Atividade 6.2.1

Já na segunda atividade, os alunos do sétimo ano mostraram-se bastante participativos, tentando relembrar esses conceitos vistos recentemente. No entanto, o sexto ano por não ter aprendido ainda essa matéria, esboçou certa apreensão de início, mas que só durou até a atividade seguinte começar, na qual eles se viram amparados pelo quadro de ajuda da Figura 37, indicando a quantidade de graus a ser girada.

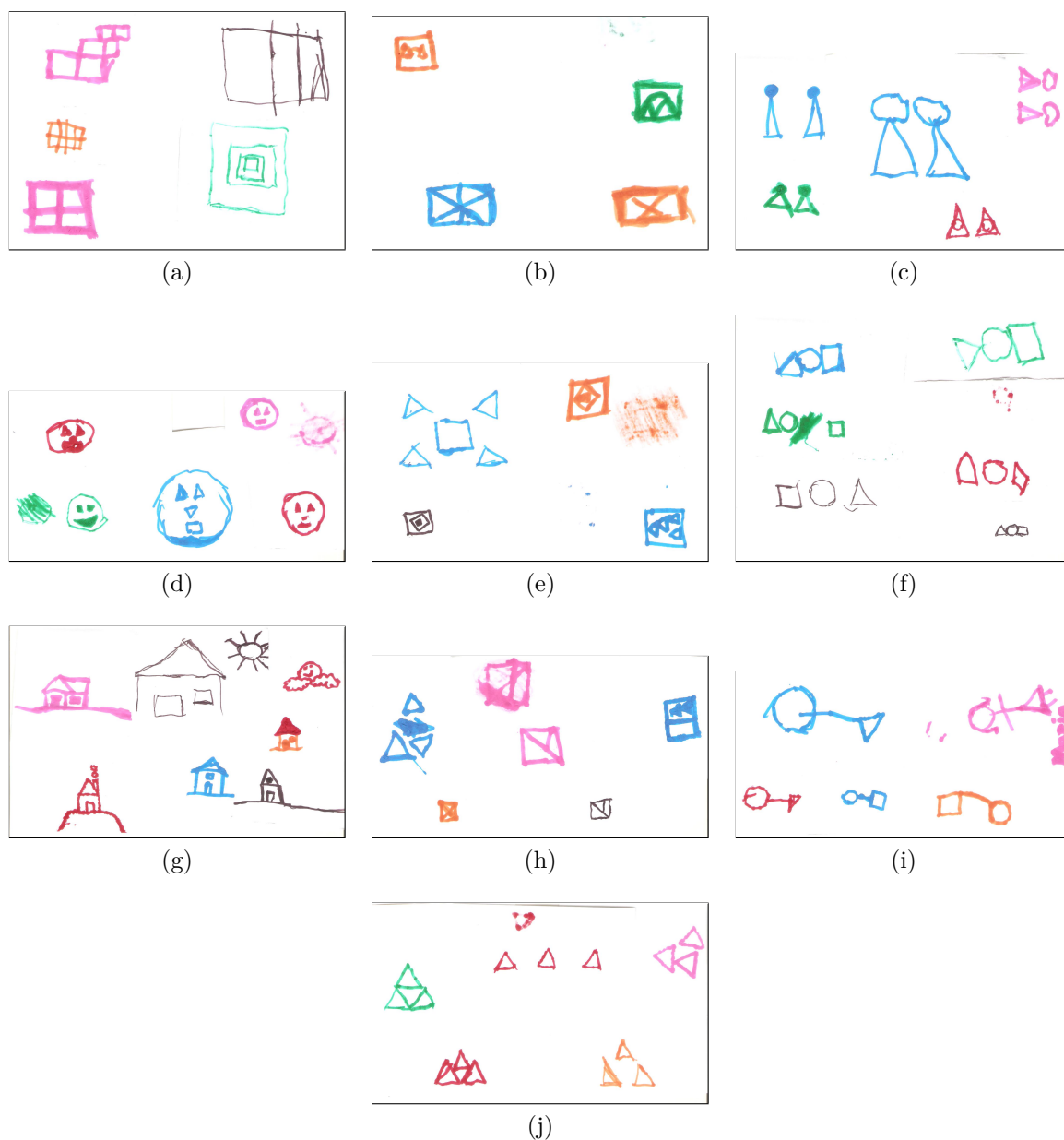
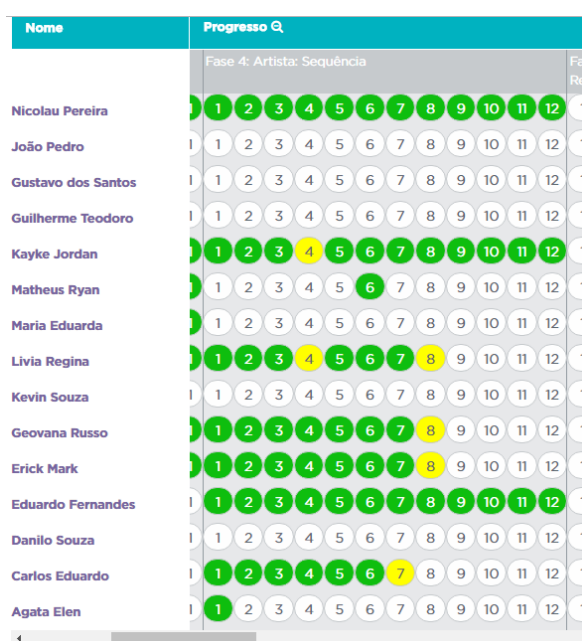


Figura 38: Desenhos dos “computadores”

As duas primeiras atividades duraram juntas cerca de 25 minutos e a última durou até o fim da aula, na qual os alunos não conseguiram passar da fase 8, devido ao tempo. Em média, pararam na sétima questão. Apenas cinco alunos continuaram a atividade em casa. Seus desempenhos podem ser observados na Figura 39.

Na décima fase aparece pela primeira vez o bloco de comando de repetição, porém como poucos chegaram até aí e como essa é uma fase de desenho livre, essa instrução ainda não foi explorada.



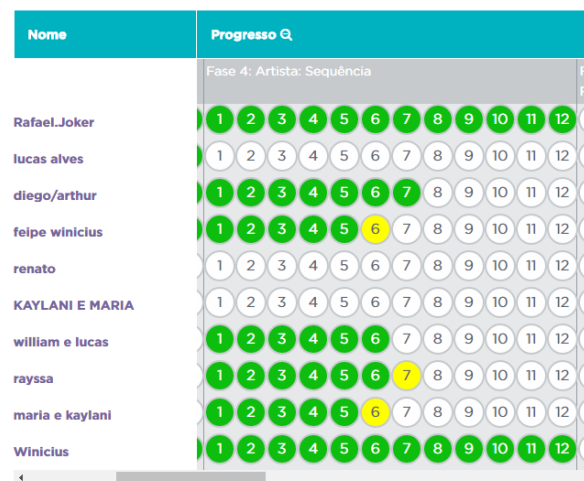
(a) Turma 6A



(b) Turma 6B



(c) Turma 7A



(d) Turma 7B

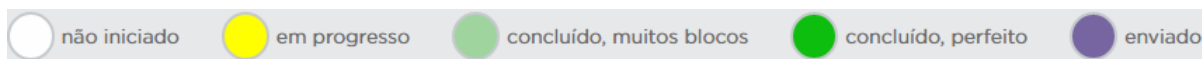


Figura 39: Progresso das turmas na etapa 4: Labirinto

### 6.3 Aula 6: repetições no Minecraft

#### Objetivos

- Associar algoritmos com atividades variadas, como plantar e construir casas, além de resolver labirintos e desenhar polígonos;
- Introduzir o comando de **repetição**;
- Comparar códigos com e sem o uso da instrução de repetição;

- Analisar as vantagens de se usar laços de repetição;
- Mostrar que um bloco de repetição pode conter outro dentro de si;
- Introduzir a ideia de **condição**.

## Code.org: Minecraft

Com os cenários e personagens do jogo Minecraft, este módulo foi bastante agradável aos alunos já familiarizados com o jogo. São 14 fases no total, com desafios temáticos do jogo, que requerem a criação de algoritmos para que o personagem tosque ovelhas, corte árvores, construa uma casa etc.

Na quinta fase o comando de repetição é introduzido por meio de um vídeo, porém por ser em inglês coube uma explicação melhor aos alunos, dessa forma, quando todos chegaram nessa fase foi dada uma pausa no Minecraft e demonstrou-se a utilidade deste recurso.

Foi solicitado um voluntário para ficar na frente da sala e obedecer às seguintes instruções: andar até o outro lado da sala, dar uma pirueta, retornar ao ponto de início, andar até o outro lado da sala, dar uma pirueta, retornar ao ponto de início, andar até o outro lado da sala, dar uma pirueta e retornar ao ponto de início. Cada nova instrução era passada após a conclusão da anterior e as repetidas foram declaradas exatamente com as mesmas palavras. Ao final, perguntou-se à turma se haveria um outro jeito, mais prático, de transmitir essas instruções, sem demorar tanto para falar.

Em geral, eles mesmos comentaram que se poderia ter falado: “faça aquelas três instruções três vezes seguidas” ou “repita três vezes as seguintes instruções...”. Neste momento, foi apresentado o bloco de repetição mostrado na Figura 40 e demonstrou-se como utilizá-lo.



Figura 40: Minecraft: fase 5

Na décima primeira fase introduziu-se o conceito de condição, por meio do comando: “Se lava à frente, faça” (i.e., se o ladrilho à frente contiver lava vulcânica), conforme a Figura 41.

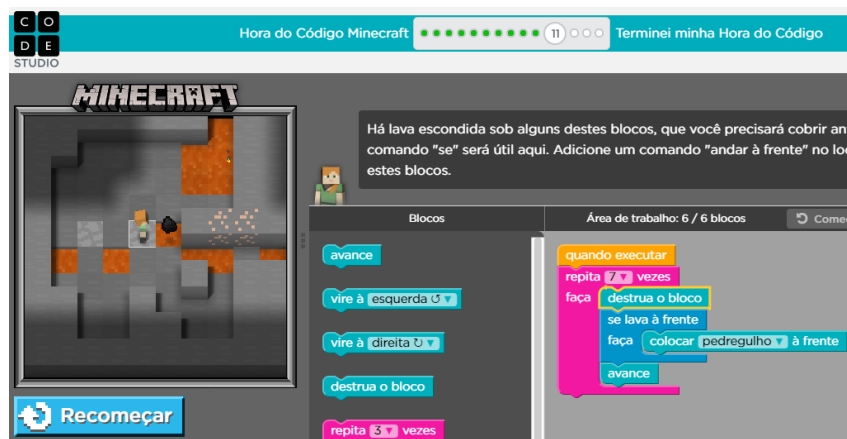


Figura 41: Minecraft: fase 11

A décima quarta fase é livre para a construção de cenários, semelhante ao modo de construção do Minecraft, porém utilizando a programação em seu favor.

## Observações

Na Figura 42 é possível avaliar o desempenho dos alunos pelas fases do módulo Minecraft. Ressaltando que os alunos sentam-se majoritariamente em duplas, os nomes em branco ou faltaram no dia ou fizeram a atividade junto com outra pessoa.

Durante a aula nenhuma dupla conseguiu completar a fase 12. Os alunos que ultrapassaram esse nível o conseguiram porque continuaram a atividade em casa.

A partir da quinta fase, quando o comando de repetição foi introduzido, o professor analisou o código de cada dupla e perguntou se era possível utilizar repetição a fim de reduzir a quantidade total de blocos. Se fosse, pedia para eles procurarem trechos repetidos e caso encontrassem separavam os blocos conforme a Figura 43. Assim, eles notavam claramente a repetição e substituíam-na pelo bloco adequado, encurtando o código.

Na fase número 11, muitos alunos ficaram confusos com o comando condicional (se lava à frente, faça) e mostraram-se hesitantes em prosseguir com a tarefa, já que a aula estava próxima do final. Não houve uma pausa para explicar a todos seu funcionamento, pois algumas duplas estavam muito defasadas em relação às outras. Dessa maneira, apenas quem atingiu essa fase recebeu uma explicação sobre a tomada de decisões que o computador precisa fazer, assim como nós decidimos nossas ações cotidianas dependendo de algumas condições. Exemplifiquei utilizando frases simples, como: “se chover, então eu abro o guarda-chuva” e pedi que eles completassem algumas: “se eu for ao parque, então...”, “se eu for à praia, então...” e similares.

Nome	Progresso Q
	Hora do Código Minecraft
Nicolau Pereira	1 2 3 4 5 6 7 8 9 10 11 12 13 14
João Pedro	1 2 3 4 5 6 7 8 9 10 11 12 13 14
Gustavo dos Santos	1 2 3 4 5 6 7 8 9 10 11 12 13 14
Guilherme Teodoro	1 2 3 4 5 6 7 8 9 10 11 12 13 14
Kayke Jordan	1 2 3 4 5 6 7 8 9 10 11 12 13 14
Matheus Ryan	1 2 3 4 5 6 7 8 9 10 11 12 13 14
Maria Eduarda	1 2 3 4 5 6 7 8 9 10 11 12 13 14
Livia Regina	1 2 3 4 5 6 7 8 9 10 11 12 13 14
Kevin Souza	1 2 3 4 5 6 7 8 9 10 11 12 13 14
Geovana Russo	1 2 3 4 5 6 7 8 9 10 11 12 13 14
Erick Mark	1 2 3 4 5 6 7 8 9 10 11 12 13 14
Eduardo Fernandes	1 2 3 4 5 6 7 8 9 10 11 12 13 14
Danilo Souza	1 2 3 4 5 6 7 8 9 10 11 12 13 14
Carlos Eduardo	1 2 3 4 5 6 7 8 9 10 11 12 13 14
Agata Elen	1 2 3 4 5 6 7 8 9 10 11 12 13 14

(a) Turma 6A

Nome	Progresso Q
	Hora do Código Minecraft
Camila Gabriela	1 2 3 4 5 6 7 8 9 10 11 12 13 14
Gabriel Nakamura	1 2 3 4 5 6 7 8 9 10 11 12 13 14
Gabriel Silveira	1 2 3 4 5 6 7 8 9 10 11 12 13 14
Guilherme Henrique	1 2 3 4 5 6 7 8 9 10 11 12 13 14
Guilherme Santos	1 2 3 4 5 6 7 8 9 10 11 12 13 14
Ighor Garcia	1 2 3 4 5 6 7 8 9 10 11 12 13 14
Isabela Ariadina	1 2 3 4 5 6 7 8 9 10 11 12 13 14
Marcela Zembruski	1 2 3 4 5 6 7 8 9 10 11 12 13 14
Maria Eduarda Borges	1 2 3 4 5 6 7 8 9 10 11 12 13 14
Matheus de Oliveira	1 2 3 4 5 6 7 8 9 10 11 12 13 14
Pedro Henrique	1 2 3 4 5 6 7 8 9 10 11 12 13 14
Renata Silva	1 2 3 4 5 6 7 8 9 10 11 12 13 14
Victória Arruda	1 2 3 4 5 6 7 8 9 10 11 12 13 14
Vinicius Poirier	1 2 3 4 5 6 7 8 9 10 11 12 13 14
Vitor Pena	1 2 3 4 5 6 7 8 9 10 11 12 13 14

(b) Turma 6B

Nome	Progresso Q
	Hora do Código Minecraft
Luiz Gustavo	1 2 3 4 5 6 7 8 9 10 11 12 13 14
Joao pedro	1 2 3 4 5 6 7 8 9 10 11 12 13 14
kaique e murilo	1 2 3 4 5 6 7 8 9 10 11 12 13 14
isaac.kaique	1 2 3 4 5 6 7 8 9 10 11 12 13 14
diego darcy	1 2 3 4 5 6 7 8 9 10 11 12 13 14
Eric w.f.l. Jhonatan	1 2 3 4 5 6 7 8 9 10 11 12 13 14
kaique murilo	1 2 3 4 5 6 7 8 9 10 11 12 13 14
eric e gabriela	1 2 3 4 5 6 7 8 9 10 11 12 13 14

(c) Turma 7A

Nome	Progresso Q
	Hora do Código Minecraft
Rafael Joker	1 2 3 4 5 6 7 8 9 10 11 12 13 14
lucas alves	1 2 3 4 5 6 7 8 9 10 11 12 13 14
diego/arthur	1 2 3 4 5 6 7 8 9 10 11 12 13 14
felpe winicius	1 2 3 4 5 6 7 8 9 10 11 12 13 14
renato	1 2 3 4 5 6 7 8 9 10 11 12 13 14
KAYLANI E MARIA	1 2 3 4 5 6 7 8 9 10 11 12 13 14
william e lucas	1 2 3 4 5 6 7 8 9 10 11 12 13 14
rayssa	1 2 3 4 5 6 7 8 9 10 11 12 13 14
maria e kaylani	1 2 3 4 5 6 7 8 9 10 11 12 13 14
Winicius	1 2 3 4 5 6 7 8 9 10 11 12 13 14

(d) Turma 7B

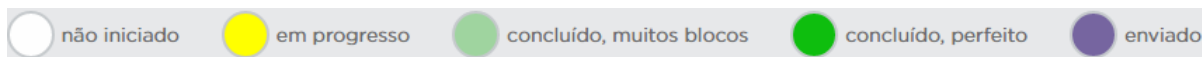
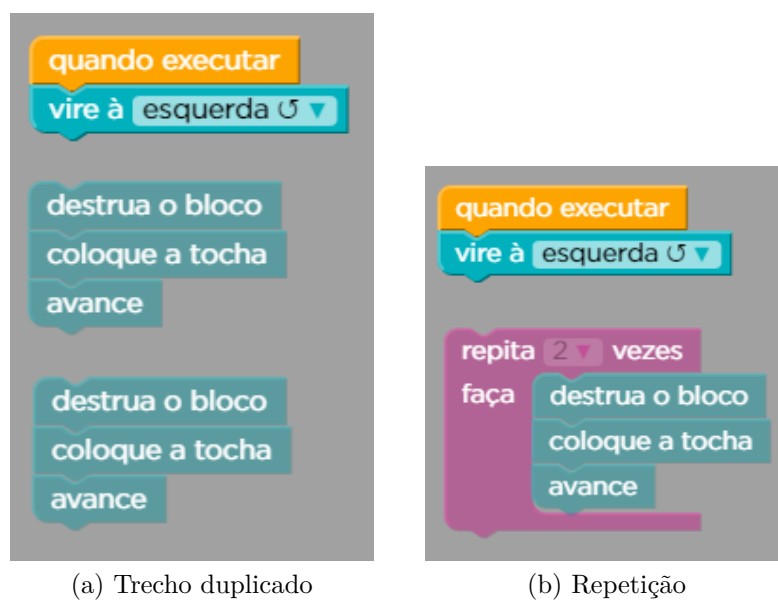


Figura 42: Progresso das turmas no módulo: Minecraft



(a) Trecho duplicado

(b) Repetição

Figura 43



## 7 Módulo IV - *Scratch*

### 7.1 Aula 7: *polígonos e mandalas*

#### Objetivos

- Manipular uma linguagem de programação completa;
- Familiarizar-se com um novo ambiente de programação;
- Perceber que esta nova plataforma dá liberdade ao aluno para criar o que quiser, sem prendê-lo a desafios específicos;
- Reproduzir figuras geométricas a partir de comandos;
- Retomar as vantagens de se utilizar o comando de repetição.

#### Usando a Caneta do Scratch

Nesta aula os alunos começaram a utilizar a linguagem de programação Scratch, detalhada na seção 1.3. Logo no início, eles precisaram criar uma conta no site [scratch.mit.edu](https://scratch.mit.edu), utilizando um nome fictício e uma conta de e-mail. O site sugere a conta de um dos responsáveis, para que ele autorize ou não o compartilhamento dos programas da criança.

Em duplas, eles foram desafiados a lembrarem-se da aula 6.2.2, em que tiveram que desenhar vários polígonos no [code.org](https://code.org). Iniciou-se com o quadrado, pois seus ângulos internos e externos são iguais e, assim, recriou-se na lousa o algoritmo para desenhá-lo. Em seguida, iniciaram-se as tentativas de transpor as instruções contidas na lousa para a linguagem Scratch.

Por ser o primeiro contato com essa linguagem, buscou-se facilitar o desafio, para que eles não se sentissem perdidos com a grande quantidade de comandos existentes e acabassem traumatizando-se com o salto de uma plataforma de ensino para a outra. Dessa forma, eles foram ajudados a localizar os comandos necessários entre as dez abas diferentes.

Assim que o quadrado foi desenhado, eles foram desafiados a modificar esse algoritmo incluindo o comando recém-aprendido na aula 6.3: a repetição. A aula então prosseguiu dessa mesma forma, com desafios periódicos, sendo os próximos o triângulo equilátero e o hexágono.

Após algum tempo, alguns alunos já estavam cansados dos polígonos, então foi sugerido que tentassem desenhar mandalas, com o intuito de tornar a aula mais atraente. Assim, eles foram instruídos a alterar a quantidade de graus girados, bem como aumentar bastante o número de repetições, obtendo desenhos simétricos. A partir daí, introduziram-se outros recursos da aba “Caneta” como: novas cores, tonalidade e espessura.

A Figura 44 apresenta os códigos básicos para se desenhar os polígonos: triângulo equilátero e quadrado, com e sem o comando de repetição. Além disso, contém o código de uma mandala.

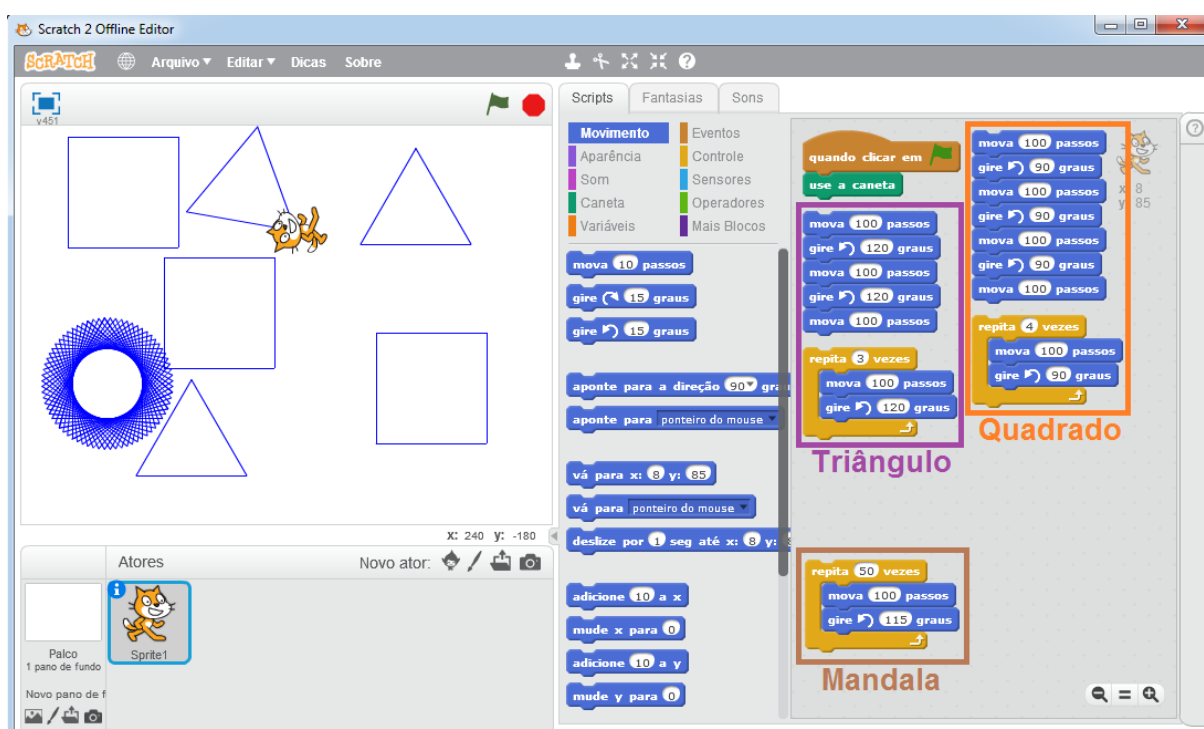


Figura 44: Polígonos e mandalas

## Observações

Os alunos tiveram mais dificuldade em criar a própria conta do que em qualquer outro momento da aula, ora porque o nome de usuário escolhido já existia, ora porque não sabiam nenhum e-mail para preencher nesse campo e acabavam inventando qualquer um, gerando erro no cadastro. Eles foram insistentemente aconselhados a gravarem seus usuários e senhas, registrando-os em papel ou em fotos no celular. Contudo, poucos o fizeram.

Após todos finalmente se acertarem com o cadastro, a atividade dos polígonos teve início. Todas as turmas conseguiram se lembrar de quais instruções deveriam utilizar para desenhar um quadrado, inclusive conseguiram transferir com certa facilidade esse algoritmo da lousa para o Scratch. No entanto, na parte de refatorar o código utilizando

comandos de repetição houve várias dúvidas, assim como na hora de formar o triângulo equilátero e calcular quantos graus são necessários na hora de girar.

Apesar das dificuldades mencionadas anteriormente, em cerca de 25-30 minutos todos estavam com diversos quadrados e triângulos desenhados pela tela além de outros polígonos. De vez em quando se mencionavam alguns ângulos notáveis, como:  $72^\circ$ ,  $60^\circ$  e  $144^\circ$ , para desenharem um pentágono, hexágono e pentagrama, respectivamente. Após fazerem vários polígonos, desenhar as mandalas foi fácil e a parte mais apreciada da aula, na qual eles brincaram com as diversas figuras, cores e tonalidades.

## 7.2 Aula 8: *atividade diagnóstica II*

Nesta aula aplicou-se uma ficha avaliativa nos mesmos moldes da atividade 5.2, colando comandos no algoritmo da ficha.

### Objetivos

- Avaliar se o conceito de repetição foi compreendido;
- Observar a preferência deles entre um algoritmo com e sem o uso de repetição;
- Compreender a diferenciação entre ângulos externos de polígonos diferentes.

### Atividade diagnóstica: *repetição*

No Apêndice D apresentam-se as fichas para realização desta atividade. Nesta aula, os alunos organizaram-se em duplas, as quais receberam os seguintes materiais: uma ficha avaliativa, diversos comandos recortados, um tubo de cola branca e duas canetas esferográficas.

Cada aluno deveria tentar resolver o desafio de sua ficha, havendo dois tipos de fichas: uma para se desenhar um triângulo e outra para se desenhar um quadrado. Dessa forma, eles foram orientados a completar o código iniciado na ficha, com o respectivo algoritmo.



Figura 45: Comandos da atividade

Foram entregues diversos blocos de comandos recortados, ilustrados na Figura 45, e comentou-se que os alunos poderiam utilizar uma quantidade máxima de blocos

em cada ficha. Quando eles tivessem certeza do algoritmo, deveriam colar os comandos em ordem no algoritmo da ficha e, caso optassem por utilizar o comando de repetição, deveriam preenchê-lo com o número mínimo de vezes necessárias para conseguir desenhar o polígono inteiro.

## Observações

A atividade, a princípio, pareceu simples para todos, porém houve códigos errados ou não tão eficientes quanto poderiam.

Durante a correção, as 27 fichas recebidas foram separadas nas categorias: o algoritmo desenha o polígono ou não desenha. Também foram contabilizados: quantos confundiram o ângulo da figura, quantos não utilizaram repetição e quantos repetiram o laço mais vezes do que o necessário. Essas quantidades encontram-se na Figura 46. Além disso, nas páginas 69 e 70 apresentam-se duas fichas resolvidas de alunos, a primeira utilizando repetição e a segunda sem usar esse recurso, ambas, porém, desenhando a mesma figura.

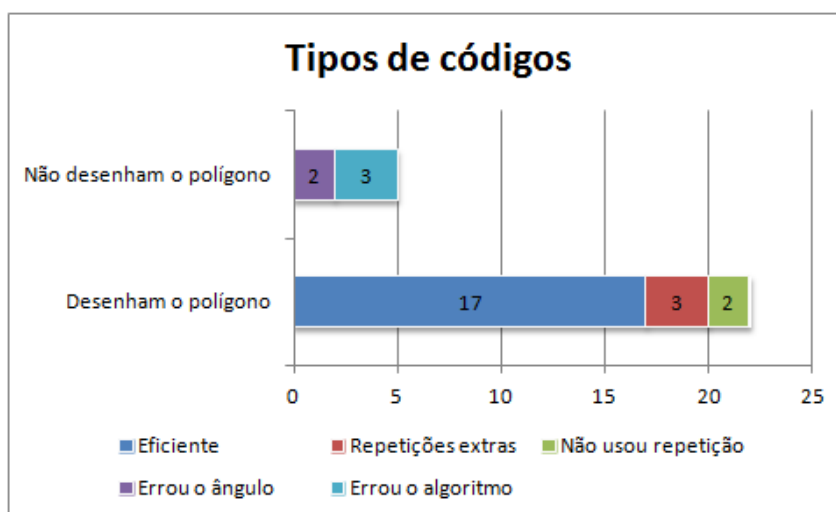


Figura 46: Algoritmos categorizados


O segundo e o terceiro comandos, já preenchidos da ficha, servem para modificar o posicionamento inicial do personagem. No entanto, percebeu-se que essa informação foi completamente desnecessária para os alunos, fazendo alguns se questionarem a respeito dessas instruções. Assim, recomenda-se a remoção desses dois blocos, uma vez que a posição do personagem não é relevante para esta avaliação.

Nome: Ézio Milha Correia Jr Turma: 7 B

Nome: MONATA DA ROCHA

## Polígonos

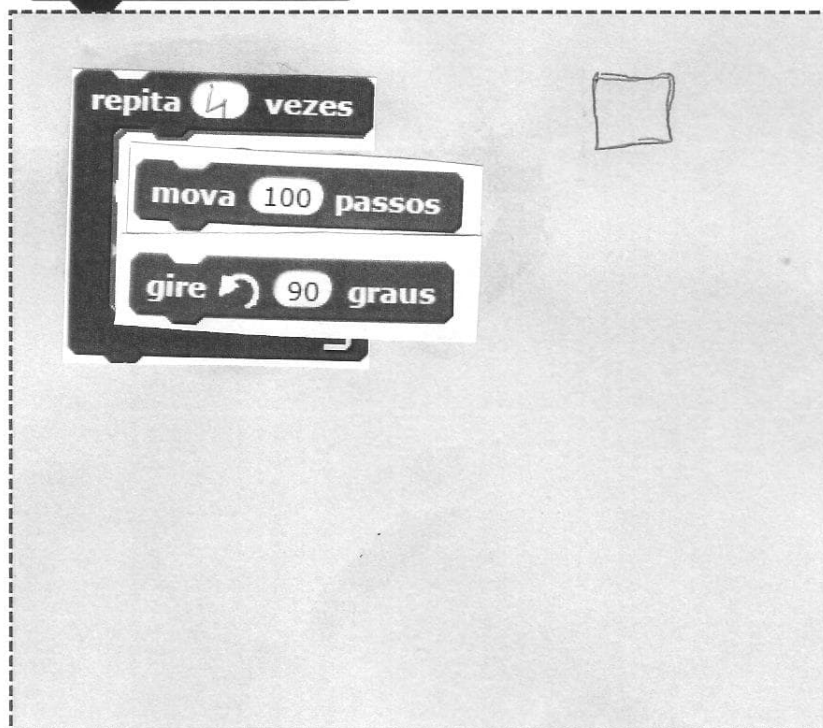
- Complete o código abaixo com os blocos de comando
- O programa deve desenhar um **quadrado**
- Quando tiver certeza dos comandos, cole-os em ordem aqui embaixo
- Quantidade ideal de comandos: TRÊS
- Quantidade máxima: OITO (não precisa usar todos)

quando clicar em 

aponte para a direção  graus

vá para x:  y:

use a caneta



repita  vezes

mova  passos


gire  graus

Nome: VANILLO S. OLIVEIRA Turma: 6<sup>o</sup> ANO A

Nome: Guilherme Costa

## Polígonos

- Complete o código abaixo com os blocos de comando
- O programa deve desenhar um **quadrado**
- Quando tiver certeza dos comandos, cole-os em ordem aqui embaixo
- Quantidade ideal de comandos: TRÊS
- Quantidade máxima: OITO (não precisa usar todos)


**quando clicar em** 

**aponte para a direção**  **graus**


**vá para x:**  **y:**

**use a caneta**


**mova**  **passos**

**gire**   **graus**

**mova**  **passos**

**gire**   **graus**

**mova**  **passos**

**gire**   **graus**

**mova**  **passos**

## 7.3 Aula 9: *labirinto*

Nesta aula retomou-se a ideia de labirinto, porém oferecendo-lhes a liberdade de criarem os próprios caminhos e cenários.

### Objetivos

- Transmitir a ideia de posicionamento do personagem na tela;
- Movimentar o personagem pelas quatro direções, trabalhando novamente com o conceito de graus;
- Trabalhar com edição de imagens;
- Utilizar o comando de **condição**;
- Começar a trabalhar com **plano cartesiano**.

### Jogo: Labirinto

Em duplas, eles abriram o Scratch *online* e a aula começou utilizando a seção de edição de imagem, uma área que se assemelha ao programa *Paint* e, assim, desenharam a entrada, a saída e as paredes do labirinto, utilizando diversas ferramentas de desenho, como: o pincel, linha, retângulo, borracha e balde de tinta, inclusive mesclando cores ou utilizando os cenários do acervo do Scratch.

Após esse momento descontraído iniciou-se a parte dos códigos, na qual o professor e os alunos construíram juntos os primeiros *scripts*, que moviam o personagem para a direita e para baixo. Para tal, estão envolvidos o evento de se pressionar uma tecla, o comando de apontar para uma direção e o comando de mover o personagem por determinado número de *pixels*. Deixou-se que eles terminassem as demais direções sozinhos.

Depois disso, foi abordada a condição de o personagem não poder encostar na parede e, assim, construímos um laço que sempre checava isso e, caso ocorresse, o personagem deveria voltar para a posição inicial do jogo.

Na Figura 47 apresenta-se o código do ator principal, sendo as partes destacadas em vermelho essenciais e construídas com todas as turmas.

### Observações

Diversos alunos já não se lembravam das contas criadas na aula anterior e acabaram criando uma nova conta, perdendo um tempo desnecessário no cadastro tentando encontrar um nome de usuário ainda não-utilizado ou digitando um e-mail válido, já que muitos simplesmente inventavam qualquer e-mail.



Figura 47: Código do labirinto

Em algumas turmas, mais tranquilas, foi possível trabalhar com recursos extras que eles próprios solicitavam, como: a troca de fantasias dos personagens, passando uma impressão mais realista de movimento; que ele falasse algo ou que a tela mudasse ao chegar na saída do labirinto; que o personagem não ficasse de ponta-cabeça ao girar para a esquerda, entre outras coisas que os animavam bastante sempre que as conseguiam realizar. Tais exemplos, estão ilustrados na Figura 47 sem os destaques em vermelho.

Devido aos frenéticos pedidos de melhorias no programa, esta aula foi muito agitada em todas as turmas, dessa forma não foi possível registrar os programas realizados pelos alunos.

## 7.4 Aula 10: *Pong*

### Objetivos

- Trabalhar com um intervalo de ângulos;
- Introduzir o conceito de **números aleatórios**;
- Aprender a utilizar um **laço condicional**;
- Fazer um programa com vários atores;
- Trabalhar com os sensores de toque do Scratch.



## Jogo: Pong

Divididos em duplas, ao abrir o Scratch, eles tiveram que escolher as *sprites* de dois personagens para o jogo: uma bola e uma plataforma. Além disso, precisaram desenhar uma faixa vermelha no cenário ou em um novo ator, na parte inferior da tela (novamente utilizando o editor de imagem do Scratch).

Pediu-se que iniciassem montando sozinhos os *scripts* da plataforma, uma vez que já fizeram isso na aula anterior com o personagem do labirinto, porém ela só deveria se movimentar para a esquerda e para a direita.

Em seguida, começamos a construir juntos o *script* da bola, ilustrado na Figura 48. Procurou-se aqui induzi-los a falar o que a bola precisaria fazer e, assim que um comando correto era falado, indicava-se em qual aba ele se localizava. Desse modo, começou a se construir o algoritmo do bola com os comandos: “sempre”, “mova 10 passos” e “se tocar na borda, volte”.

Depois foi acrescentada a condição que, ao tocar na plataforma, a bola deveria quicar para cima, com os blocos: “se ... então”, “tocando em Plataforma”, “aponte para a direção ...” e “número aleatório entre -30 e 30”. Aqui escolheu-se um intervalo de direções que apontassem para cima.

Após isso, o jogo deveria parar caso a bola tocasse a barra inferior vermelha, nesse momento, substituiu-se o comando “sempre” pela repetição com condição: “repita até que” e “tocando na cor vermelha” e acrescentou-se o bloco “pare todos” ao final.

Para encerrar, foram feitos alguns ajustes como o posicionamento inicial, para a bola não começar o jogo já presa na barra vermelha fazendo-a apontar para cima.



Figura 48: Pong

## Observações

Vários alunos não se lembravam mais das contas criadas e, dessa vez, foram distribuídos papéis e canetas esferográficas para que os alunos que precisassem criar uma conta nova anotassem o usuário, a senha e o e-mail fornecidos.

Esta aula foi extensa e alguns alunos não conseguiram fazer os ajustes finais, como o posicionamento inicial. Sempre procurou-se atender aos pedidos deles de melhorias no jogo, assim, com os alunos que completaram a aula, foi possível ir além: criando uma segunda plataforma para defender a parte de cima da tela, incluindo sons ou cenários.

## 7.5 Aula 11: *Space Invaders*

Nesta aula, foram disponibilizados todos os comandos que seriam utilizados no programa, com os quais eles deveriam tentar montar o código sozinhos.

### Objetivos

- Assimilar cada comando a uma determinada ação;
- Diferenciar blocos de condições dos demais;
- Rever repetição com condição;
- Conectar os comandos na ordem correta.

### Jogo: *Space Invaders*

Primeiramente, foi solicitado às duplas que entrassem em um *link* para o usuário do professor no Scratch, do qual eles deveriam compartilhar consigo o jogo *Space Invaders*. Dessa vez, o código foi estruturado de maneira a deixar todos os blocos que seriam utilizados já no editor de *scripts*, devidamente embaralhados e desconectados uns dos outros, para que os alunos os encaixassem nos respectivos eventos e na ordem correta.

Em seguida, explicaram-se as ações de cada personagem, procurando pronunciar palavras semelhantes às dos comandos.

- Nave: deve se mover para a direita e para esquerda ao serem pressionadas as respectivas setas do teclado (*scripts* já trabalhados com o Labirinto e o Pong);
- Bala: quando a tecla “espaço” for pressionada, o personagem deve ir para a mesma posição da Nave e, enquanto não tocar na cor verde, deve subir repetidamente, porém, após tocar, esconde-se;

- Alien: sempre se move para um lado e, ao tocar na borda, move-se para o outro; se ele tocar na Bala, então todos os personagens param e o jogo acaba.

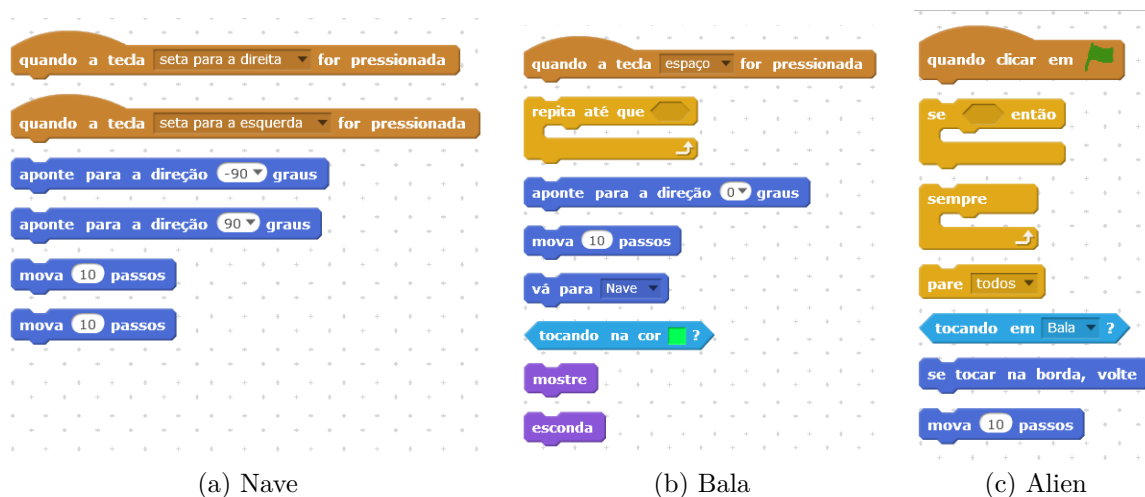


Figura 49: Blocos de comandos desconexos de cada personagem

## Observações

Todos os alunos começaram pelos *scripts* da Nave, com bastante sucesso. Em seguida, alguns tentaram a Bala primeiro, outros o Alien, nos dois casos apresentaram certa dificuldade, com certas duplas muito resistentes em prosseguir com a atividade. Algumas jogavam os blocos fora e não conseguiam fazer mais nada; em compensação, alguns alunos completaram com grande facilidade o desafio.

A maioria apresentou mais dificuldade para completar o código da Bala, muitos por não entenderem o comando “vá para Nave” e o colocarem dentro do laço, acabando com a movimentação retilínea uniforme para cima.

## 7.6 Aula 12: *esconde-esconde*

Os comandos necessários para o funcionamento do programa foram previamente separados, com exceção de quatro blocos. Os alunos deveriam conectá-los, bem como localizar e incluir os comandos faltantes, de modo a fazer o programa funcionar conforme a especificação.

### Objetivos

- Entender a especificação de cada personagem;
- Conectar os comandos na ordem correta, atendendo à especificação;

- Descobrir quais comandos faltantes completam o código;
- Perceber a necessidade de usar repetição;
- Diferenciar os eixos horizontal e vertical a partir da movimentação do personagem.

## Jogo: esconde-esconde

As duplas entraram no usuário do professor no Scratch e *remixaram* o programa esconde-esconde (criaram uma cópia dele em seus próprios usuários). Assim, iniciaram a montagem do código.

Neste programa o personagem deveria aparecer em uma posição aleatória da tela, permanecer lá por um segundo, desaparecer por um segundo e depois reaparecer em outra posição e, assim, repetidamente. Quando o personagem fosse clicado pelo *mouse* o jogo deveria terminar.



Figura 50: Blocos de comandos desconexos do programa

Os comandos iniciais do programa foram previamente destacados, mostrados na Figura 50. A princípio, as duplas tentaram sozinhas desenvolver o programa, contudo, à medida que travavam, novas orientações eram passadas. Dessa forma, foram orientados a dividir o problema em partes:

1. Fazer o personagem sumir e reaparecer em um intervalo de dois segundos;
2. Tornar os sumiços e as aparições constantes (incluindo-os em um laço de repetição);
3. Fazer o personagem trocar de posição horizontal;
4. Localizar quais comandos o fazem trocar de posição vertical;
5. Fazer o programa terminar assim que o personagem for clicado.

Os blocos faltantes sugeridos podem ser vistos na Figura 51 [a]. No entanto, eles poderiam utilizar blocos diferentes, mas que produziriam o mesmo resultado. Na Figura 51 [b] encontram-se exemplos de blocos alternativos bastante frequentes: o primeiro substituiu o bloco “sempre” e o segundo, os blocos “mude x para número aleatório entre -240 e 240” e “mude y para número aleatório entre -180 e 180”



(a) Mais frequentes



(b) Alternativas

Figura 51: Blocos de comandos faltantes

A fim de auxiliá-los com o posicionamento do personagem, foi escolhido como cenário o tema **xy-grid** nativo do Scratch, que consiste em um plano cartesiano, conforme Figura 52.

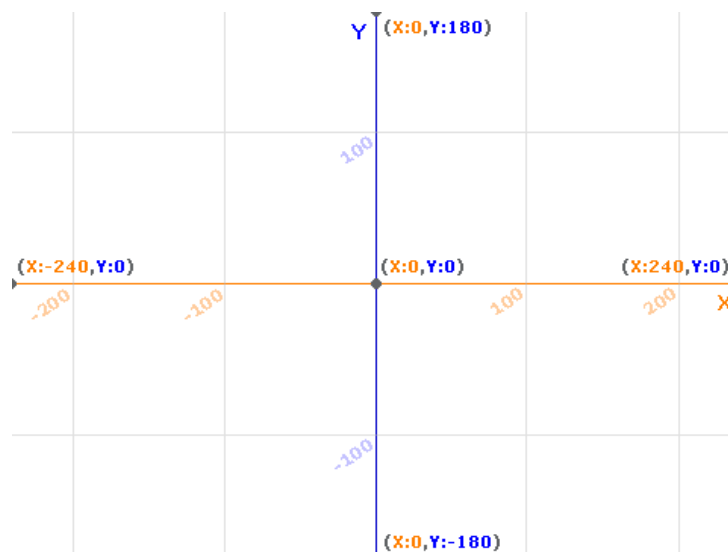


Figura 52: Pano de fundo “xy-grid”

## Observações

Este programa mostrou-se simples, uma vez que a grande maioria conseguiu acabá-lo antes do término da aula, além disso tiveram tempo para realizar suas próprias alterações, como: mudança de cenário, troca de fantasias do personagem, alterações no tempo de aparição, inclusão de novos personagens, utilização da ferramenta **carimbo** e inclusão de sons.

As duplas que não conseguiram terminar haviam jogado fora o código inicial e

perderam tempo tentando recuperá-lo, ou, então, simplesmente tentavam qualquer combinação de códigos sem se importar com o resultados.

## 8 Módulo V - *MIT App Inventor*

### 8.1 Aulas 13 e 14: *bola de cristal*

Nestas aulas uma nova ferramenta foi introduzida aos alunos: o App Inventor, mais bem detalhado em [1.4](#).

#### Objetivos

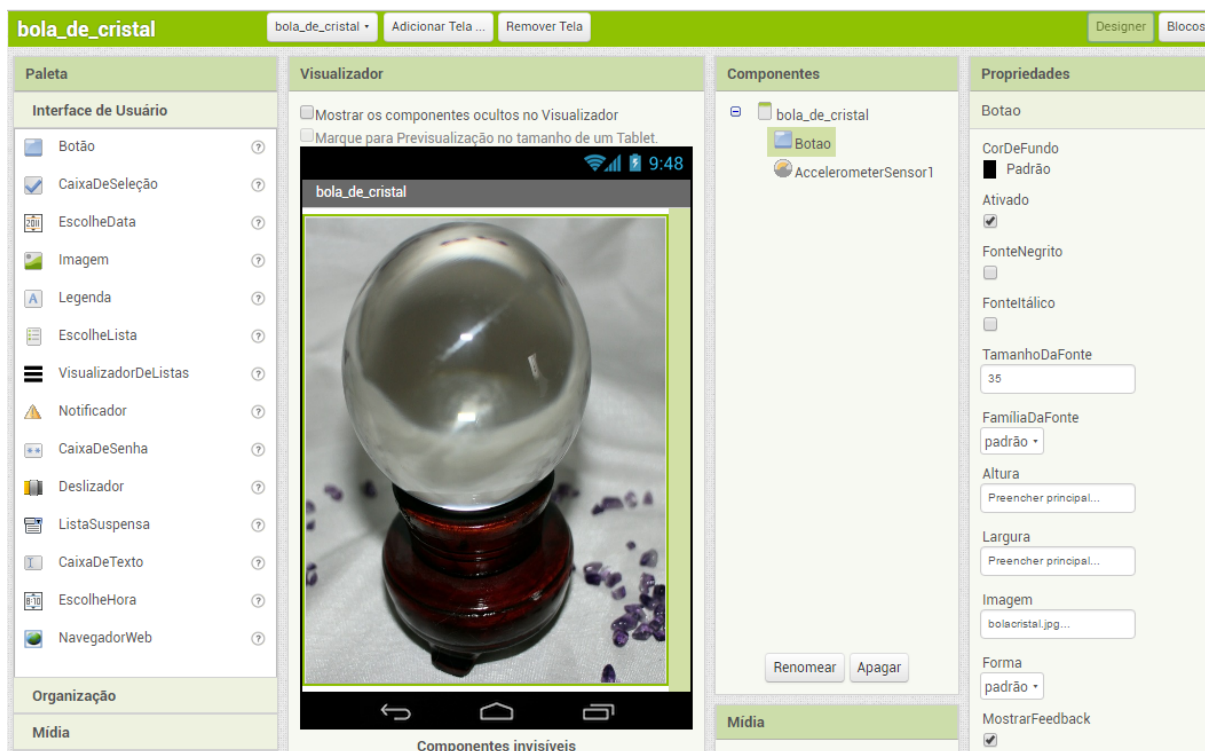
- Conhecer uma nova ferramenta de programação;
- Atentar para a parte gráfica do programa;
- Trabalhar com *widgets* como botões e legendas (*labels*);
- Trabalhar com sensores do celular como o acelerômetro;
- Introduzir o conceito de **lista**;
- Criar um aplicativo para sistema Android;
- Baixar e testar o aplicativo no próprio celular.

#### Aplicativo: Bola de cristal

Nesta atividade os alunos deveriam, em duplas, montar um aplicativo para sistema Android utilizando a plataforma App Inventor. O *app* consiste em uma única tela, contendo uma imagem de bola de cristal. Se o usuário chacoalhar o celular, ativando o acelerômetro, uma das entradas de uma lista de respostas deve aparecer na tela, além disso, quando a tela for clicada, esse texto deve ser apagado. O programa se assemelha ao brinquedo da década de 1950 conhecido como *Magic 8 Ball*.

Primeiramente, os alunos tiveram que se cadastrar no site: [appinventor.mit.edu](http://appinventor.mit.edu), criando uma conta. Processo que demorou para alguns, pois não se lembravam de seu e-mail ou sua senha, tendo, assim, que criar uma conta nova.

Ao iniciar um novo projeto, o editor abre no modo *designer*, conforme Figura 53. Neste modo, o usuário deve adicionar os elementos que irão compor a tela da aplicação, como imagens, botões, sensores, caixas de entrada etc. e personalizá-los. No caso deste *app* será preciso apenas acrescentar: um botão e o acelerômetro.

Figura 53: Modo *designer*

Na segunda aula, utilizou-se o editor de blocos, mostrado na Figura 54. Nele foi criada uma lista de respostas para serem mostradas ao chacoalhar o dispositivo, bem como um comando que apaga esse texto ao se clicar na imagem da bola de cristal.



Figura 54: Editor de blocos

Depois dessas etapas, foi o momento de instalar o aplicativo nos celulares. Para isso, foi fornecida a senha do Wi-Fi para que eles baixassem o *app MIT AI2 Companion*, um assistente para testar o programa e gerar o APK para instalá-lo.



## Observações

Nas duas aulas destinadas para a criação desse aplicativo, pode-se observar os seguintes empecilhos:

- Esquecimento da conta: alguns alunos não se lembravam mais da conta que haviam criado na primeira aula e tiveram que refazer o *design* do aplicativo na segunda aula;
- Falta de celular: metade dos alunos conseguiu instalar o aplicativo nos próprios celulares. Os outros não possuíam celular ou possuíam um com sistema operacional diferente do Android.

Apesar disso, foi uma aula boa para se concluir o curso, uma vez que os alunos saíram bastante animados, brincando com o aplicativo criado.

Parte III

Resultados

## 9 Considerações finais

O presente trabalho proporcionou diversos resultados interessantes, dentre os quais a própria experiência de tê-lo aplicado foi o principal.

A elaboração do curso não foi uma tarefa fácil, dado que, por mais que existam recentes currículos internacionais tratando do assunto, nenhum deles aborda quais ferramentas pedagógicas são as mais indicadas para se ensinar determinados assuntos. Dessa forma, foram realizados testes com diversas plataformas de ensino, buscando localizar as que abordavam os assuntos escolhidos de forma mais didática, já que simplesmente selecioná-las por adequação à idade não é um medidor preciso, uma vez que os alunos em questão nunca tiveram contato algum com programação.

Além disso, grande parte dos livros e dos artigos pesquisados trata sobre os benefícios da tecnologia na educação ou sobre *softwares* educacionais como apoio a outras disciplinas.

Outro ponto a se considerar é tempo de aula (45 minutos semanais), que se mostrou curto demais para alguns que após uma semana já haviam esquecido por completo os ensinamentos passados. No entanto, por serem todas as atividades *online*, alguns deles continuaram os desafios e programas em casa, dessa forma seria válido incluir atividades para serem feitas em casa ao longo da semana.

No mais, o curso mostrou-se bastante divertido para o público-alvo e, aplicando-se as devidas alterações para melhorar o controle sobre as atividades realizadas, é possível torná-lo ainda mais proveitoso.

### 9.1 Inclusão digital nas escolas

A sala de informática da escola, proveniente do programa de inclusão digital ACESSA São Paulo, foi imprescindível para o projeto, contudo esse espaço com bons computadores mostrou-se completamente sub-utilizado. Desconsiderando este projeto, os alunos foram nessa sala apenas uma ou duas vezes neste ano.

Compreende-se que apenas oito computadores não podem atender simultaneamente uma sala com mais de trinta alunos, dessa forma esse programa mostra-se insuficiente para atender às demandas tecnológicas da escola.

Contudo, entende-se que é possível elaborar atividades dividindo-se a turma, ficando parte em sala de aula, parte no laboratório de informática, ou, ainda, manter o laboratório aberto durante os intervalos para pesquisas e elaboração de trabalhos, uma

vez que nesse ínterim, vários alunos vinham até a sala pedindo para utilizá-la. Todavia, percebe-se que faltam profissionais para supervisionar a sala dividida ou durante os intervalos.

Em consequência disso, os programas de inclusão social aparentam ser mera fachada, desencadeados por modismos educacionais, sem uma proposta efetiva de melhoria no ensino.

## Conclusão

Ninguém desconhece a gravidade e magnitude das falhas que a educação brasileira pública deve solucionar, nem os pontos em que elas precisam ser repensadas: professores enfrentando condições de trabalho adversas, recebendo salários baixos, com uma carreira desprestigiada, sem incentivo para uma formação continuada, ao lado de uma realidade social em que os jovens não entendem bem o papel da cultura, do ensino e da escola em suas vidas. Assim, tem-se clareza das limitações do alcance do projeto desenvolvido, frente a uma realidade tão árdua. Apesar disso, o contato direto dos estudantes com a programação na educação básica pode ser um bom começo para sanar o desestímulo apresentado pelos jovens frente a metodologias engessadas e abundância de conteúdo que diversas vezes não se mostra útil para os estudantes.

# Referências

BELL, T. et al. Ensinando ciência da computação sem o uso do computador. 2011. Citado 2 vezes nas páginas 16 e 54.

KELLEHER, C.; PAUSCH, R. Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys (CSUR)*, ACM, v. 37, n. 2, p. 83–137, 2005. Citado na página 14.

MATTOS, F.; FERREIRA, V.; ANACLETO, J. O ensino de programação com scratch e seu impacto na opção profissional para meninas. *Simpósio Brasileiro de Informática na Educação*, v. 27, n. 1, p. 300, 2016. Citado na página 26.

SAELI, M. et al. Teaching programming in secondary school: a pedagogical content knowledge perspective. *Informatics in Education-An International Journal*, Vilniaus Universiteto Leidykla, n. Vol 10\_1, p. 73–88, 2011. Citado na página 17.

SEEHORN, D. et al. Csta k–12 computer science standards: Revised 2011. ACM, 2011. Citado na página 17.

TUCKER, A. et al. A model curriculum for k-12 computer science. Citeseer, 2003. Citado na página 17.

# Apêndices

## APÊNDICE A – Letramento digital

Nas próximas quatro páginas encontra-se a proposta inicial do curso enviada às escolas de aplicação.



# Letramento Digital: primeiros bytes

---

O projeto de Letramento Digital visa trazer inovação para a sala de aula através do ensino de programação para os alunos de escolas públicas, utilizando formas lúdicas e colaborativas de aprendizado, desenvolvendo o pensamento lógico e a capacidade de resolver problemas.

As crianças estão cada vez mais conectadas, porém essa interação com a tecnologia ainda é básica, não indo além de jogos, conversas e navegação na internet. Com este projeto, os alunos conseguirão ultrapassar essas fronteiras digitais, projetando e criando seus próprios jogos, se expressando por meio das tecnologias atuais.

A fase piloto foi planejada idealmente para durar **um semestre**, com aulas semanais **entre 45min e 1h30min** (conforme disponibilidade da escola).

Todo o **material** utilizado é gratuito e está disponível online, sendo as principais ferramentas: o site [code.org](http://code.org), a plataforma de criação e programação [Scratch](http://Scratch) e o [applInventor](http://applInventor), ambiente desenvolvedor de aplicativos. Todas elas foram desenvolvidas para se ensinar computação para crianças a partir de oito anos de idade e se utilizam de uma linguagem de programação leve, visual e lúdica. Seu formato em blocos passa a ideia de que as instruções para o computador são peças de encaixar e que, juntando umas nas outras, torna-se possível elaborar desenhos, contar histórias, desenvolver jogos e aplicativos.

O **número máximo de alunos** depende da quantidade de computadores disponíveis na escola, sendo possível juntar uma dupla por máquina.

Para a viabilidade do projeto, o processamento das máquinas e a velocidade da internet não podem ser muito lentos. Além disso, seria interessante se houvesse algum projetor para uma melhor visualização das demonstrações e modelos.

Anexo segue o resumo das aulas da fase piloto (um semestre), entretanto o cronograma dos conteúdos passados pode sofrer alterações, conforme o ritmo de aprendizado dos alunos.

---

Projeto de TCC – Agosto/2016

Bacharelado em Ciência da Computação, pelo Instituto de Matemática e Estatística – USP

Graduando: Arthur Branco Costa - [arthbr@gmail.com](mailto:arthbr@gmail.com) - (11) 99497-9534

Orientador: Prof. Dr. Leônidas de Oliveira Brandão

# Conteúdo programático

## Aula 1 - Introdução

- Apresentações pessoais e um resumo do projeto, mostrando as ferramentas que serão utilizadas.
- Conversa descontraída em roda, com perguntas e respostas sobre: o que é programação? Para quê ela serve? Onde a vemos no dia a dia?
- “Programando a Máquina”: atividade desplugada (sem uso do computador), que coloca os alunos no lugar da máquina, tentando obedecer a uma série de instruções ao pé da letra para replicar um desenho e observando que o resultado pode ser bem distante do esperado.
- LightBot [Figura 1 - LightBot]: neste aplicativo os alunos devem fazer o robô cumprir determinadas tarefas através dos comandos disponíveis, explorando conceitos como: instrução (comandos), direção (esquerda e direita) e sequenciamento (a ordem dos comandos afeta o resultado final).

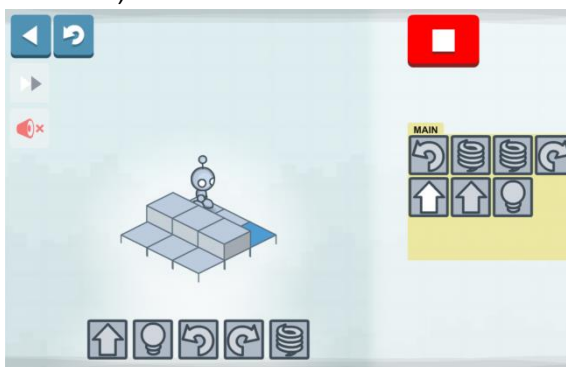


Figura 1 - LightBot

## Aulas 2 a 4 - Desafios

- Trabalharemos com o site [code.org](http://code.org)
- O site possui uma gama de atividades direcionadas para estudantes que já sabem ler e não têm experiência em programação. Nele realizaremos o curso 2.
- Ele utiliza uma linguagem de programação visual e bastante intuitiva.
- Neste módulo os alunos resolverão diversos desafios lógicos, explorando aspectos computacionais como: algoritmo (sequência de comandos), repetição (como otimizar comandos repetidos) e condição (como a máquina toma uma decisão).



Figura 2 - Code.org: curso 2

- O progresso de cada estudante é avaliado automaticamente a cada etapa. É possível conferir se algum nível foi pulado, bem como se o desempenho foi satisfatório.

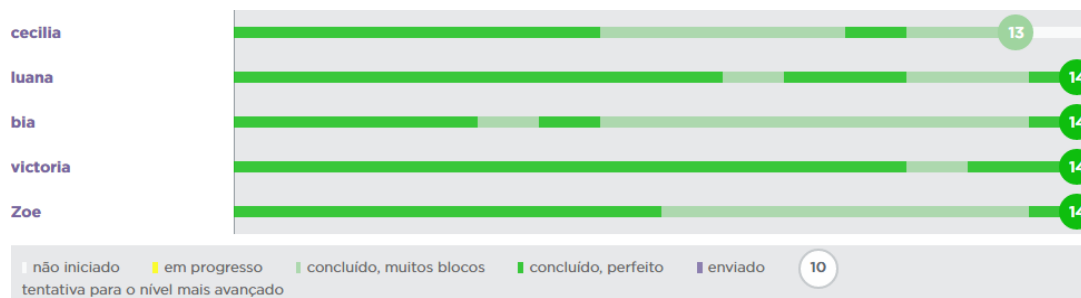


Figura 3 - Code.org: progresso da turma

## Aulas 5 a 12 - Criação

Nessas aulas trabalharemos com o [Scratch](#), uma linguagem de programação visual, desenvolvida no MIT, com intuito de ajudar crianças no aprendizado de conceitos matemáticos e computacionais.

Na primeira aula deste módulo, os alunos elaborarão uma história e deverão desenvolvê-la, adquirindo familiaridade com a ferramenta e suas funcionalidades. Ao final, apresentarão suas histórias para os colegas.

Nas aulas seguintes eles desenvolverão jogos simples, cuja dificuldade de implementação deve aumentar gradativamente. As aulas abordarão elementos:

- Lógico-matemáticos: posição, direção, aritmética (as quatro operações), plano cartesiano, comparação (maior, menor, igual), variáveis, etc.
- De informática: ferramentas de seleção, preenchimento, desenho, rotação, etc.
- Computacionais: operadores lógicos, repetição, condição, paralelismo, etc.



Figura 4 - Scratch: jogo Pong

## Aulas 13 até o final - Aplicação

Nessas aulas trabalharemos com o [App Inventor](#), um site para desenvolvimento de aplicativos para Android.

É possível que eles instalem os aplicativos criados nos próprios celulares (Android).

Também trabalha com uma linguagem visual, derivada do Scratch, porém com diversos recursos extras e um pouco mais de texto. Há ainda poucos comandos em inglês, mas todos bem simples, a maior parte do site está traduzida.

Neste módulo os alunos aplicarão todo o conhecimento adquirido na parte de criação, e deverão desenvolver um trabalho próprio, que poderá ser instalado e utilizado em seus próprios dispositivos eletrônicos (sistema *Android*). Os alunos irão lidar com recursos mais concretos como: botões, textos, listas e imagens, e juntamente com os elementos computacionais vistos no módulo anterior conseguirão desenvolver aplicações mais robustas.

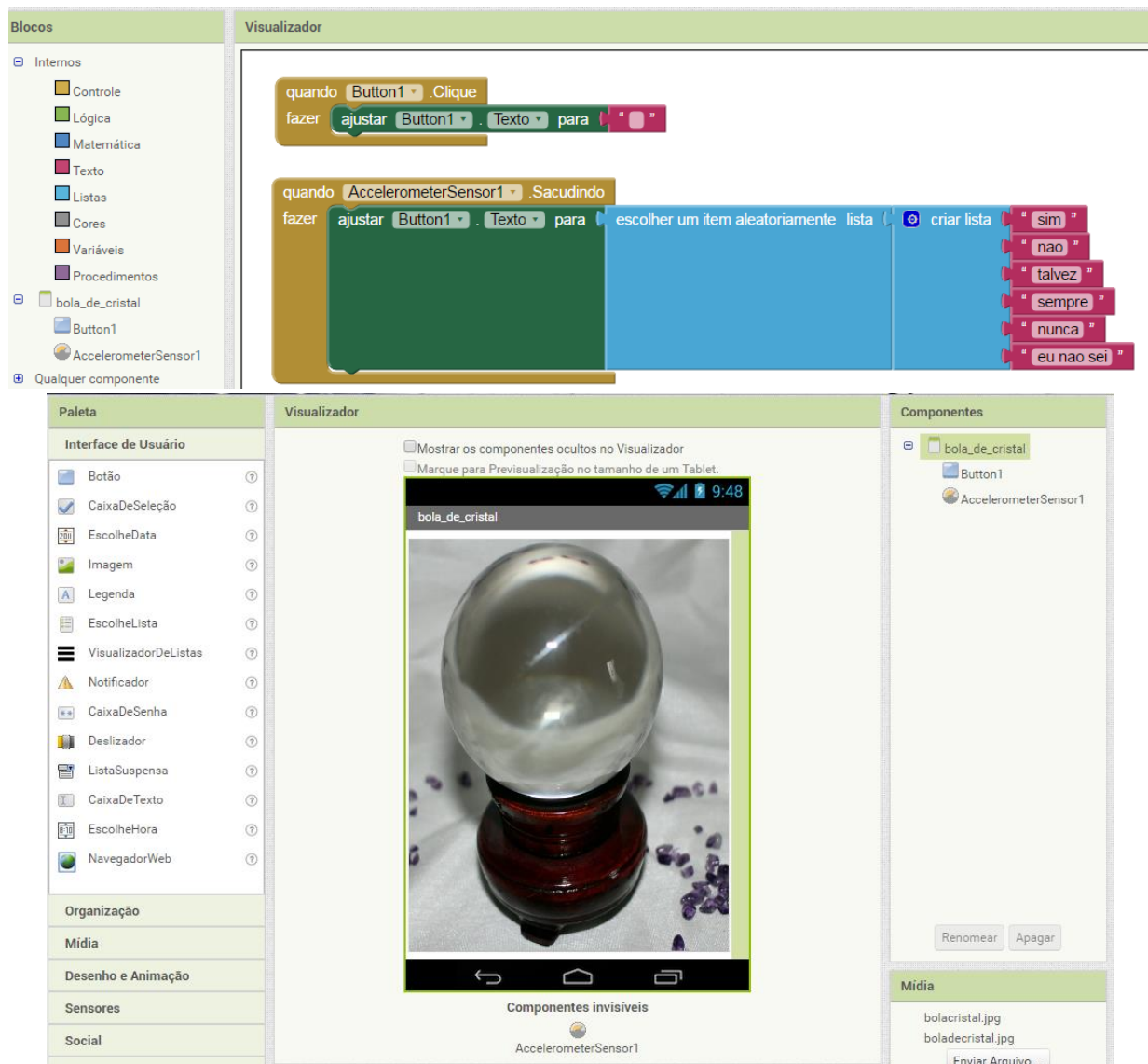


Figura 5 - AppInventor: modelo de aplicativo

## APÊNDICE B – Questionário

Referente à atividade [4.1.1](#).

As 15 perguntas do questionário encontram-se nas duas páginas a seguir.

---

# QUESTIONÁRIO DO ALUNO

---

**1. Nome:**

\_\_\_\_\_

**2. Qual o seu sexo?**

Masculino       Feminino

**3. Quando você nasceu?**

Dia: \_\_\_\_\_ Mês: \_\_\_\_\_ Ano: \_\_\_\_\_

**4. Você tem computador e internet em casa?**

- Computador com internet  
 Computador sem internet  
 Celular ou *tablet* com internet  
 Celular ou *tablet* sem internet  
 Nenhum

**5. Com que idade você começou a usar o computador?**

\_\_\_\_\_

**6. Você usa o computador com que frequência?**

- Todos os dias  
 De 3 a 5 vezes por semana  
 1 ou 2 vezes por semana  
 Quase nunca

**7. Você já usou o computador da escola?**

Sim, nas aulas de:

\_\_\_\_\_

Sim, para pesquisas e trabalhos

Não

**8. De onde você acessa a internet?**

Casa                                       Escola

Lan house                                 Outros:

\_\_\_\_\_

**9. Você usa o computador para:**

- Pesquisas e trabalhos escolares  
 Acessar redes sociais (Facebook, Twitter, Instagram, ...)  
 Conversar e enviar mensagens (Whatsapp, MSN, Messenger, ...)  
 Jogos  
 Assistir vídeos (YouTube)  
 Outros:

\_\_\_\_\_

**10. Em média, qual a duração de seus acessos por dia?**

- Até 30 minutos  
 Entre 1 e 2 horas  
 Mais de 2 horas

**11. Você tem e-mail?**

Sim, é este aqui:  
\_\_\_\_\_

Não

**12. Com qual frequência você acessa seu e-mail?**

Mais de 5 vezes na semana

Entre 2 e 5 vezes na semana

Menos de 2 vezes na semana

Praticamente nunca

**13. Quais programas / aplicativos de celular você mais usa?**

Facebook       YouTube

Instagram       Snapchat

Twitter       Whatsapp

Jogos

Outros:  
\_\_\_\_\_

**14. Você já pensou em criar o seu próprio jogo de computador?**

Sim

Não

**15. Se você pudesse criar um aplicativo de celular, você gostaria que ele fizesse o quê?**

---

---

---

---

---

---

---

---

---

---

# APÊNDICE C – Atividade diagnóstica: Light-Bot

Referente à aula 5.2.

Conteúdo presente neste apêndice:

- Duas fichas de atividade em branco, a partir da página 98.
- Quatro fichas resolvidas, a partir da página 99. O exercício contém duas fichas com desafios diferentes entre si, que os alunos resolveram de dois jeitos diferentes:
  - Ficha I
    - \* Código preferido entre a grande maioria alunos (página 99);
    - \* Código pouco frequente (página 100), optado por apenas dois estudantes em todas as turmas;
  - Ficha II
    - \* Cerca de 70% dos alunos utilizaram este algoritmo (página 101);
    - \* Opção menos frequente (página 102).

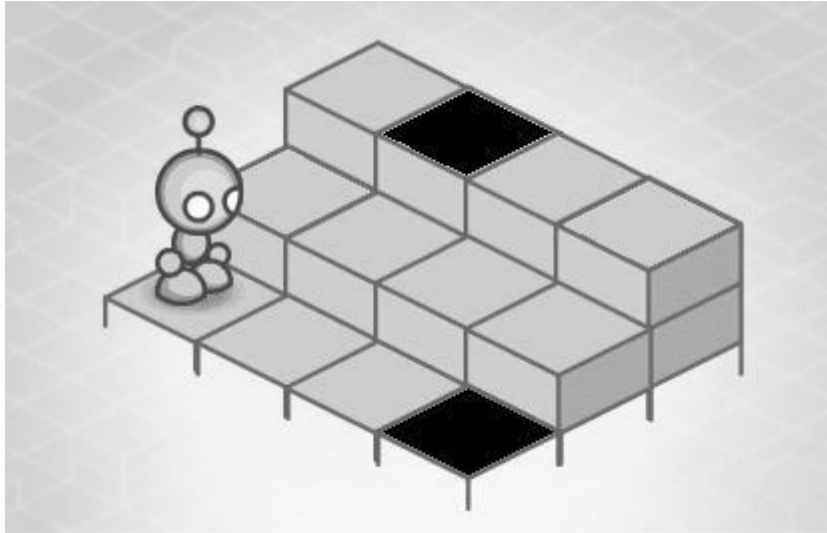


Nome: \_\_\_\_\_

Turma: \_\_\_\_\_

## Programando o robô

Ajude o robô a acender todos os quadrados pretos no caminho abaixo.



Use no máximo doze comandos.

Quando tiver certeza dos comandos, cole-os em ordem aqui embaixo. Você **não** precisa usar todos os comandos.

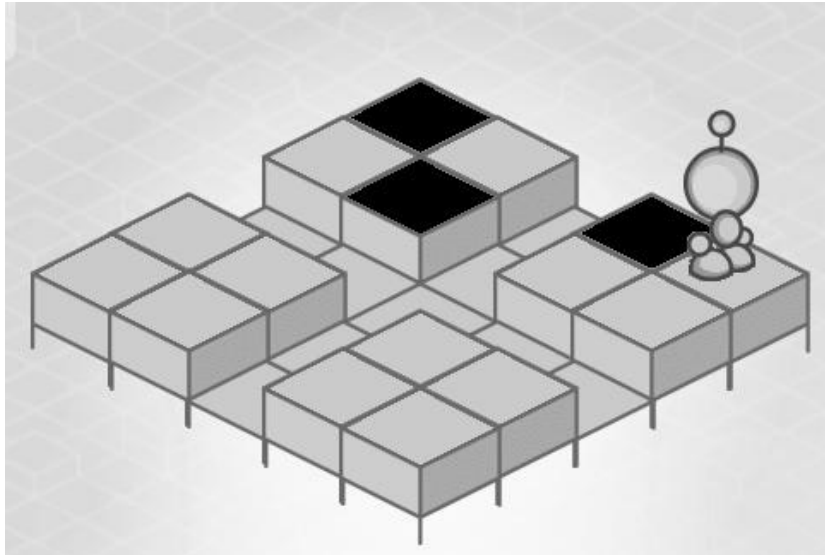
1	2	3	4
5	6	7	8
9	10	11	12

Nome: \_\_\_\_\_

Turma: \_\_\_\_\_

## Programando o robô

Ajude o robô a acender todos os quadrados pretos no caminho abaixo.



Use no máximo doze comandos.

Quando tiver certeza dos comandos, cole-os em ordem aqui embaixo. Você **não** precisa usar todos os comandos.

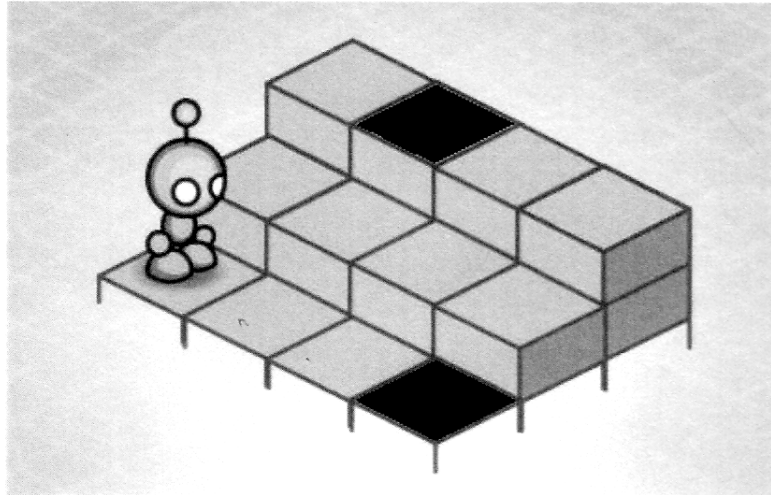
1	2	3	4
5	6	7	8
9	10	11	12

Nome: Gabriela

Turma: 7º A

### Programando o robô

- Ajude o robô a acender todos os quadrados pretos no caminho abaixo.



- Use no máximo doze comandos.
- Quando tiver certeza dos comandos, cole-os em ordem aqui embaixo.
- Você **não** precisa usar todos os comandos.

			12

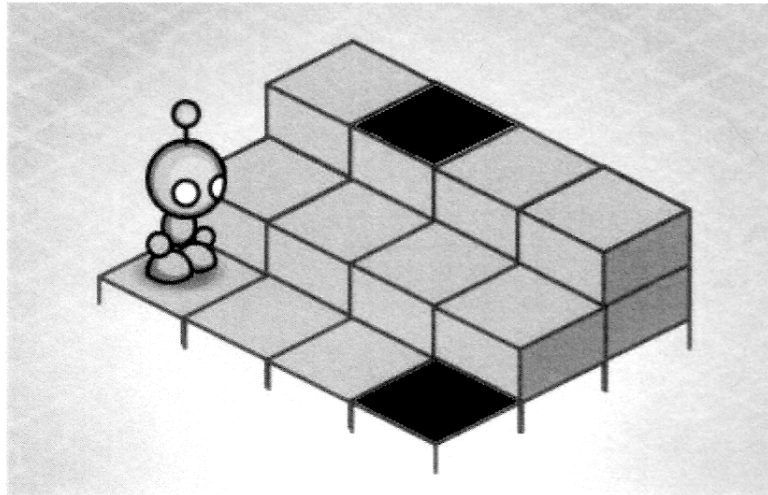
↑  
↑  
↑  
💡  
↪  
🌀  
🌀  
↩  
↑  
↑  
💡

Nome: Lucas Celso de Lima

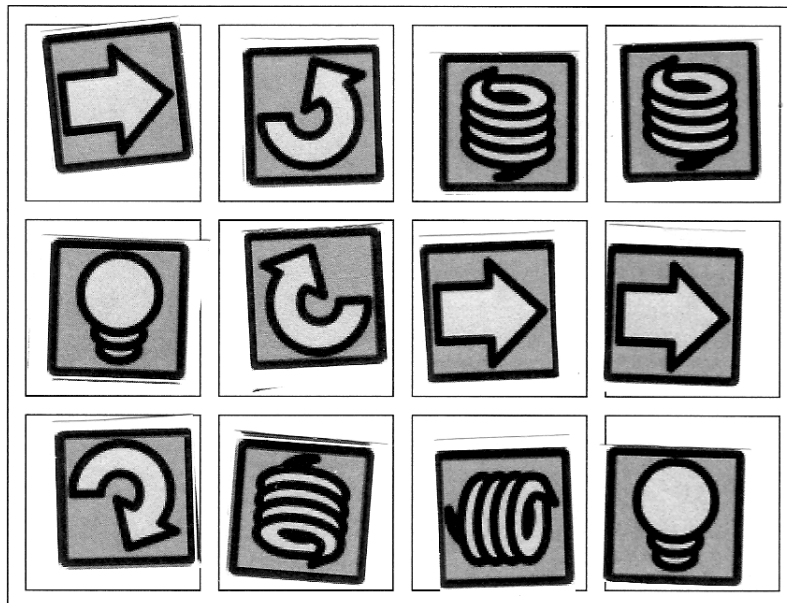
Turma: 7ºB

## Programando o robô

- Ajude o robô a acender todos os quadrados pretos no caminho abaixo.



- Use no máximo doze comandos.
- Quando tiver certeza dos comandos, cole-os em ordem aqui embaixo.
- Você **não** precisa usar todos os comandos.

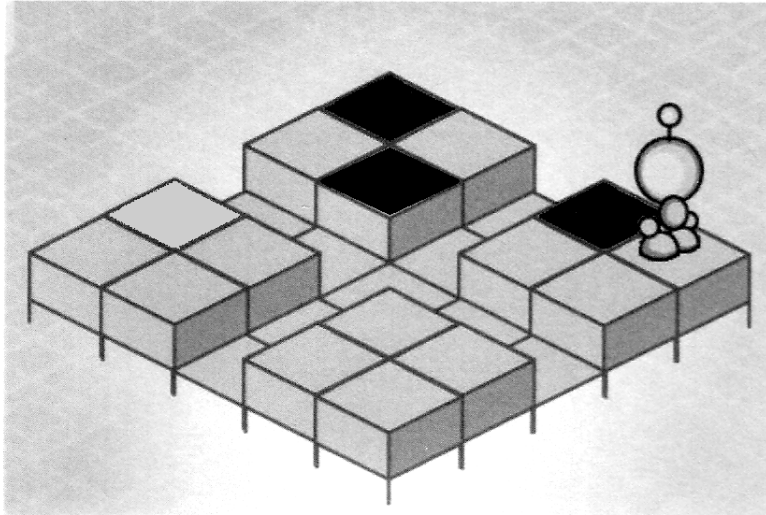


Nome: Guilherme Mendes

Turma: 6<sup>º</sup>B

### Programando o robô

- Ajude o robô a acender todos os quadrados pretos no caminho abaixo.



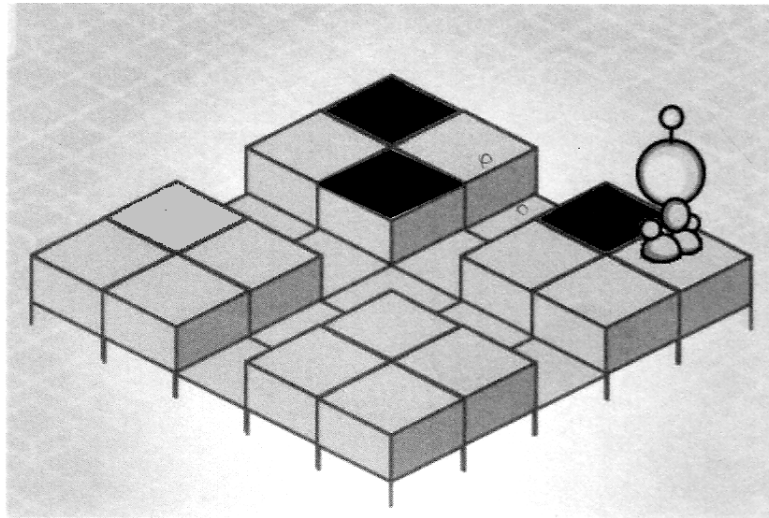
- Use no máximo doze comandos.
- Quando tiver certeza dos comandos, cole-os em ordem aqui embaixo.
- Você **não** precisa usar todos os comandos.

			12

Nome: Micelone Lorenna do Lago Junior Turma: 6<sup>o</sup> A

### Programando o robô

- Ajude o robô a acender todos os quadrados pretos no caminho abaixo.



- Use no máximo doze comandos.
- Quando tiver certeza dos comandos, cole-os em ordem aqui embaixo.
- Você **não** precisa usar todos os comandos.


# APÊNDICE D – Atividade diagnóstica: repetição

Referente à aula [7.2](#).

Nome: \_\_\_\_\_ Turma: \_\_\_\_\_

Nome: \_\_\_\_\_

## Polígonos

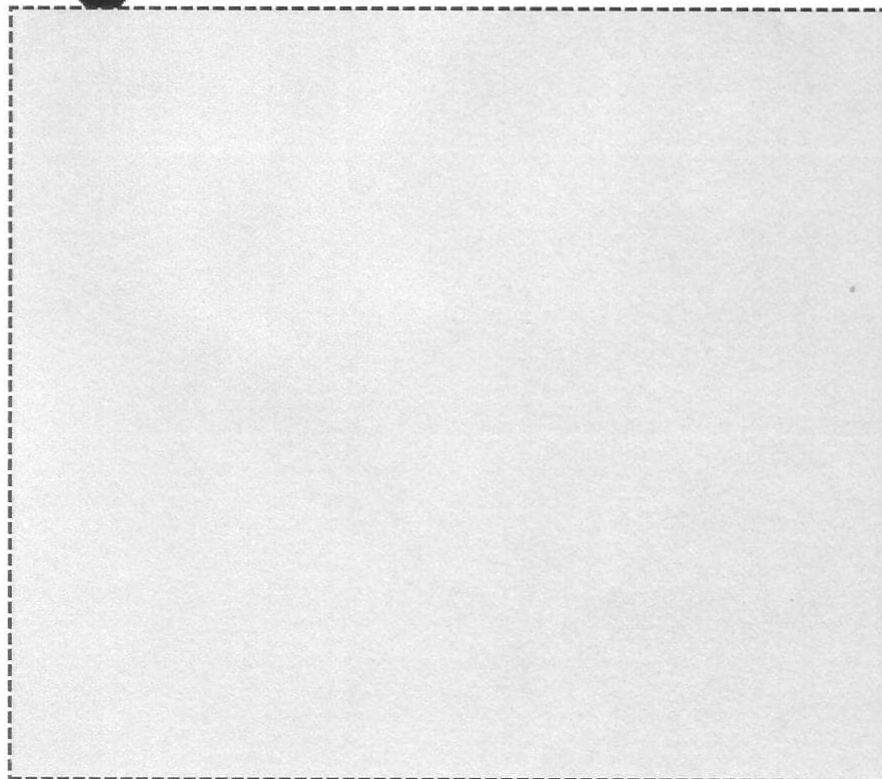
- Complete o código abaixo com os blocos de comando
- O programa deve desenhar um **triângulo equilátero**
- Quando tiver certeza dos comandos, cole-os em ordem aqui embaixo
- Quantidade ideal de comandos: TRÊS
- Quantidade máxima: SEIS (não precisa usar todos)

quando clicar em 

aponte para a direção  graus

vá para x:  y:

use a caneta





Nome: \_\_\_\_\_ Turma: \_\_\_\_\_

Nome: \_\_\_\_\_

## Polígonos

- Complete o código abaixo com os blocos de comando
- O programa deve desenhar um **quadrado**
- Quando tiver certeza dos comandos, cole-os em ordem aqui embaixo
- Quantidade ideal de comandos: TRÊS
- Quantidade máxima: OITO (não precisa usar todos)

quando clicar em 

aponte para a direção  graus

vá para x:  y:

use a caneta

