

MICHEL FIGUEIREDO DE SOUZA

CHRISTINE SERTÃ COSTA

SCRATCH



Guia Prático

para aplicação na Educação Básica

1ª Edição



SCRATCH:
Guia Prático para aplicação na
Educação Básica

MICHEL FIGUEIREDO DE SOUZA

CHRISTINE SERTÃ COSTA

SCRATCH:
Guia Prático para aplicação na
Educação Básica

1ª Edição



Rio de Janeiro, 2018

COLÉGIO PEDRO II
PRÓ-REITORIA DE PÓS-GRADUAÇÃO, PESQUISA, EXTENSÃO E CULTURA
BIBLIOTECA PROFESSORA SILVIA BECHER
CATALOGAÇÃO NA FONTE

S729 Souza, Michel Figueiredo de
SCRATCH: Guia Prático para aplicação na Educação Básica /
Michel Figueiredo de Souza; Christine Sertã Costa. - 1. ed. - Rio de
Janeiro: Imperial, 2018.

78 p.

Bibliografia: p. 78.

ISBN: 978-65-5930-093-8

1. Educação – Estudo e ensino. 2. Scratch (Linguagem de programação de computador). 3. Jogos eletrônicos. 4. Tecnologia educacional. 5. Teoria da aprendizagem significativa. I. Costa, Christine Sertã. II. Título.

CDD: 371.33

RESUMO

SOUZA, Michel Figueiredo de. **Scratch**: Guia Prático para aplicação na Educação Básica. 78 p. Produto Educacional – Colégio Pedro II, Pró-Reitoria de Pós-Graduação, Pesquisa, Extensão e Cultura, Programa de Mestrado Profissional em Práticas de Educação Básica, Rio de Janeiro, 2019.

O produto aqui apresentado foi desenvolvido visando ser uma possibilidade de promoção da autonomia do aluno e da aprendizagem significativa por meio do trabalho com tecnologia na escola. É fruto dos desdobramentos das atividades desenvolvidas junto aos alunos de pós-graduação do Programa de Residência Docente do Colégio Pedro II, na cidade do Rio de Janeiro, e junto aos alunos dos anos finais do Ensino Fundamental da Escola Municipal Rotary, no município de Duque de Caxias, no contexto da pesquisa de mestrado profissional cujo título é “Ensino e Tecnologias: aprendizagem significativa por meio da plataforma Scratch”. A pesquisa teve o objetivo geral de analisar o potencial de contribuição da plataforma Scratch para as práticas pedagógicas escolares. Definiu como objetivos específicos: analisar as opiniões de professores e alunos sobre a utilização da plataforma; desenvolver oficinas sobre o Scratch para instrumentalizar professores e alunos para o uso da plataforma; organizar um “Estúdio” na plataforma para concentrar a interação e a cooperação entre os participantes da pesquisa; elaborar um guia sobre a plataforma Scratch para orientar outros professores a respeito das possibilidades de sua inserção nas práticas de ensino. Tratou-se de um estudo de caso de cunho exploratório, a partir de uma abordagem descritiva e qualitativa. Utilizamos como instrumentos para obtenção de dados questionários mistos impressos e online e observações semiestruturadas ao longo do desenvolvimento das oficinas. Os resultados indicaram que, por meio da formação continuada docente e de propostas de ensino inovadoras envolvendo a tecnologia, é possível promover a autonomia, a expressão criativa e o trabalho colaborativo com suporte da tecnologia, com desdobramentos positivos potenciais sobre a qualidade do ensino.

Palavras-chave: Ensino e tecnologia; Práticas inovadoras; Scratch; Ensino Fundamental

O Scratch é um projeto da Fundação Scratch, em colaboração com o grupo Lifelong Kindergarten no MIT Media Lab. Está disponível gratuitamente em <https://scratch.mit.edu>.

SUMÁRIO

Apresentação	9
Introdução	10
I. Scratch: Origem, Estrutura e Funcionamento	13
1. Uma breve longa história.....	13
2. Jardim de infância para a vida toda.....	15
3. Começando no Scratch.....	16
3.1. Requisitos.....	16
4. Conhecendo a plataforma.....	17
4.1. Inscrição.....	18
4.2. Explorar.....	19
4.3. Ideias.....	20
4.4. Sobre.....	20
4.5. Editor.....	21
4.6. Programar no Scratch.....	23
4.6.1. Área de programação.....	23
4.6.2. Blocos de Movimento.....	24
4.6.3. Blocos de Aparência.....	25
4.6.4. Blocos de Som.....	26
4.6.5. Blocos de Eventos.....	27
4.6.6. Blocos de Controle.....	28
4.6.7. Blocos de Sensores.....	29
4.6.8. Blocos de Operadores.....	30
4.6.9. Blocos de Variáveis.....	33
4.6.10. Meus Blocos.....	34
II. Programando com o Scratch – Exemplos	36
Atividade 1: Animar seu nome.....	36
Atividade 2: Veloz e Furioso.....	41
Atividade 3: Astronauta Artista.....	46
Atividade 4: Desafio de Lógica.....	52

III. Scratch na Escola – Exemplo de Aplicação	60
Jogo: Passeio pela Amazônia	60
Apresentando	60
Ficha Técnica	61
Estrutura.....	62
Códigos do Palco	64
Códigos dos Atores	66
Referências.....	79

Apresentação

Esse Guia Prático é fruto da pesquisa de mestrado vinculada ao Programa de Mestrado Profissional em Práticas de Educação Básica, ligado ao Pró-Reitora de Pós-Graduação, Pesquisa, Extensão e Cultura do Colégio Pedro II.

Iniciamos o estudo da plataforma Scratch e das possibilidades de sua utilização integrada às práticas pedagógicas escolares durante a disciplina de Educação e Tecnologias no Espaço Escolar, frequentada no primeiro semestre de 2017. Consolidamos os conhecimentos a partir da organização do minicurso Introdução à Programação com Scratch, oferecido por nós, no segundo semestre de 2018, aos residentes do Programa de Residência Docente do Colégio Pedro II, uma pós-graduação *latu sensu* (especialização) oferecida pelo Colégio Pedro II a docentes da Educação Básica.

Posteriormente, realizamos o desenvolvimento da pesquisa em campo, durante o primeiro semestre de 2018. A pesquisa foi realizada junto aos alunos dos anos finais do Ensino Fundamental da Escola Municipal Rotary, no município de Duque de Caxias/RJ, onde pudemos avaliar, por meio de oficina oferecida aos alunos do 8º e 9º anos, os desdobramentos da utilização do Scratch como ferramenta de apoio ao trabalho pedagógico.

A partir do desenvolvimento do minicurso e da pesquisa na escola, decidimos elaborar um guia que permitisse aos professores interessados na temática das tecnologias aplicadas ao processo de ensino serem apresentados ao Scratch e suas possibilidades. Um guia que facilite a compreensão básica da estrutura e funcionamento da plataforma e que apresente exemplos e sugestões de desenvolvimento de objetos virtuais voltados direta ou indiretamente para o processo de aprendizagem.

Os pilares da metodologia que embasa esse guia são a teoria da aprendizagem significativa de David Ausubel (2003) e a teoria da aprendizagem baseada na construção de objetos e mediada pela tecnologia de Seymour Papert (1993), ao que se soma a concepção de aprendizagem criativa utilizando o trabalho por projetos de Mitchel Resnick (1998). A partir desse núcleo metodológico, apresentamos a plataforma Scratch para o professor da educação básica. Esta é uma ferramenta voltada para a aprendizagem a partir de projetos e da construção de objetos virtuais significativos pelos alunos e, portanto, contribui para uma aprendizagem também mais significativa.

Esperamos que esse Guia Prático contribua para que outros professores da Educação Básica, interessados pelo trabalho com tecnologias e preocupados com a qualidade da aprendizagem de seus alunos, vislumbrem novas possibilidades de articulação entre tecnologia, ensino e aprendizagem. E, principalmente, que forneça subsídios para uma prática pedagógica que contemple os anseios e se aproprie das ferramentas do mundo tecnológico contemporâneo.

Introdução

O Scratch¹ é uma plataforma criada pelo *Lifelong Kindergarten Group*, pertencente ao *Media Lab* do *Massachusetts Institute of Technology* (MIT), de Boston, EUA. Utiliza-se da programação por blocos para criação de objetos virtuais. Dentre as muitas aplicações possíveis, destacam-se as animações, os jogos e as apresentações, além de muitas outras possibilidades. É uma plataforma online, mas com possibilidade de desenvolvimento offline, além de ser gratuita e de código aberto.

Sua criação repousa sobre quatro pilares conceituais: em inglês, *projects*, *passion*, *peers* e *play*, que podemos, numa tradução livre, entender como: trabalho com projetos, paixão, cooperação e pensar brincando. Essa filosofia baseou-se na experiência de Mitchel Resnick (1998) à frente do *Lifelong Kindergarten Group* no *Media Lab*, com influência também das ideias de Seymour Papert (1993) a respeito da relação educação-tecnologia. Fazendo-se a leitura² do QR Code ao lado, é possível compreender essa filosofia nas palavras do próprio Resnick.



Nesse vídeo de 1986, o próprio Papert apresenta a fundamentação do seu trabalho com tecnologia na escola. MIT Media Lab. 1986. Duração: 2min.

A proposta desse Guia Prático é apresentar as características e o funcionamento do Scratch, com o olhar voltado para sua aplicação na Educação Básica. As diversas possibilidades de criação de objetos virtuais com utilização da plataforma permitem o desenvolvimento de projetos atrelados aos conceitos e conteúdos de uma disciplina específica do currículo escolar, mas também projetos interdisciplinares, agregando a colaboração dos conhecimentos das diversas disciplinas curriculares e até mesmo projetos que, partindo dos interesses de cada aluno ou grupo de alunos, extrapolem para uma abordagem transdisciplinar.

De toda forma, independente da abordagem que se pretenda (disciplinar, interdisciplinar ou transdisciplinar), acreditamos que a incorporação de tecnologia, no nosso caso o Scratch, ao processo de ensino tenha o potencial de contribuir para uma aprendizagem mais significativa por parte do aluno (AUSUBEL, 2003; LEITE, 2011; PELIZZARI, 2002; PISCHETOLA, 2016). A intenção é aprender por meio da criação de produtos (PAPERT, 1993), no nosso caso, virtuais, utilizando-se dos conhecimentos

¹ Disponível online em: <http://www.scratch.mit.edu>

² Dentre as muitas opções existentes de leitor de QR Code, sugerimos a utilização do app Scanner QR, gratuito, disponível para Android em: <http://play.google.com/store/apps/details?id=com.tohsoft.qrcode>.

prévios, mas também sendo instigado à busca de novos conhecimentos, aperfeiçoando o processo de aprendizagem gradualmente.

Além disso, acreditamos que o trabalho com o Scratch, se bem conduzido, pode promover, em paralelo à aprendizagem, uma consciência mais crítica dos alunos frente à sua relação com a(s) tecnologia(s). Pode despertar os alunos para se perceberem para além de meros consumidores de produtos tecnológicos (MOREIRA, 2000). Entenderem que também é possível criar, ter maior autonomia para se expressarem (FREIRE, 1996), apropriando-se de forma mais consciente dos recursos tecnológicos, sua estrutura e seu funcionamento.

Com esses objetivos em mente, organizamos esse Guia Prático com foco na figura do professor. Voltados para permitir que o professor possa apropriar-se da ferramenta e de seu funcionamento, posteriormente podendo utilizá-la atrelada à sua prática pedagógica, criando objetos virtuais voltados para a aprendizagem dos alunos ou, melhor e mais aconselhável ainda, lançando mão da apropriação da ferramenta por parte de seus alunos para, ao longo desse processo fazer os *links* com os objetivos de aprendizagem dos alunos.

Sendo assim, dividimos esse Guia em três etapas. Na primeira, se faz um breve histórico do Scratch, suas características e funcionamento. O objetivo é permitir que se aproprie da estrutura da plataforma e das diversas funcionalidades dos grupos de comandos existentes nos blocos de programação, suas combinações e como proceder à programação por blocos. Na segunda etapa apresentamos exemplos de alguns tipos de aplicações possíveis de serem desenvolvidas por meio da plataforma e mostramos o passo-a-passo de como criá-las. Aqui, se pretende despertar o olhar para como os objetos virtuais criados podem permitir a expressão da aprendizagem e/ou a busca pela mesma por parte do aluno. Na terceira etapa, concluímos com um exemplo de aplicação desenvolvida com utilização da plataforma Scratch, para aplicação no processo de ensino a respeito das questões socioambientais da Amazônia.

Esperamos sinceramente que esse Guia Prático possa contribuir para instigar os professores – e por tabela os alunos – a expandirem sua consciência crítica no tocante à sua relação com a tecnologia. Que nesse processo, os indivíduos possam despertar para as possibilidades que a tecnologia tem a oferecer para sua expressão, ação criativa e busca de mais conhecimento. Enfim, que com apoio do uso desse Guia Prático, em algum momento do processo, o usuário possa ter aquele instante de *insight*, onde o que antes parecia complexo e distante se torne assimilável, ao alcance das mãos e da mente.

Quando  virar página

Iniciar

Parte  I

I. Scratch: Origem, Estrutura e Funcionamento

1. Uma breve longa história

A plataforma Scratch é o resultado de um longo processo de trabalho com tecnologia e aprendizagem realizado pelo *Media Lab* do MIT, mais especificamente no *Lifelong Kindergarten Group*, um grupo criado e coordenado pelo pesquisador Mitchel Resnick. O principal objetivo do grupo é pesquisar e compreender melhor como o trabalho criativo com as tecnologias, especialmente usando bricolagem e programação, se relaciona com o processo de aprendizagem.



Resnick foi orientado por Seymour Papert, que desenvolveu a teoria do **construtivismo**, segundo a qual a aprendizagem baseada na construção de algo concreto favorece uma aprendizagem mais significativa (PAPERT, 1993). Papert estudou com Piaget, cuja teoria construtivista teve grande influência em seu pensamento. Também foi pioneiro na utilização da informática no ambiente escolar, ainda durante a década de 1970. Nesse processo, desenvolveu uma linguagem de programação chamada LOGO, voltada para a aprendizagem a partir da criação de objetos virtuais.

Nesse vídeo de 1986, o próprio Papert apresenta a fundamentação do seu trabalho com tecnologia na escola. MIT Media Lab. 1986. Duração: 2min.

No final da década de 1980, já no MIT *Media Lab*, Papert passou a utilizar a linguagem LOGO e os blocos de montar Lego para criar objetos que executassem determinadas ações, respondendo aos comandos programados a partir do computador. Foi o embrião dos projetos de robótica educacional que atualmente são sinônimo de modernidade nas escolas que optam por metodologias inovadoras.

Foto 1: Projeto Logo Lego



Fonte: <http://lukazi.blogspot.com/2014/07/lego-legos-first-programmable-product.html>

2. Jardim de infância para a vida toda

A estrutura da plataforma Scratch foi pensada de acordo com a filosofia do *Lifelong Kindergarten Group*, inspirada no funcionamento do jardim de infância, onde as crianças são estimuladas a imaginar, criar, brincar, compartilhar e refletir. A plataforma também é baseada em quatro pilares que norteiam sua estrutura e funcionamento: *projects, passion, peers e play*. Ou seja, trabalho com projetos, paixão, cooperação e pensar brincando, numa tradução livre.

- **Projetos (Projects):** toda criação de objetos virtuais no Scratch é realizada a partir de um projeto de trabalho. É imprescindível planejar minimamente o que se vai fazer, como será a estrutura do projeto visualmente e a nível de programação por blocos. É essencial se colocar no lugar de quem vai utilizar a criação para pensar na interface, na apresentação e no funcionamento.
- **Paixão (Passion):** estimula-se a criação a partir do que move o interesse do usuário, com o objetivo de que a produção seja significativa para o próprio usuário em primeiro lugar. Com isso, busca-se o engajamento e o estímulo à criatividade.
- **Cooperação (Peers):** o tempo todo há o incentivo às trocas, seja de ideias ou sugestões, seja das próprias criações. Todo produto virtual fica disponível na plataforma para a apreciação da comunidade de usuários. Qualquer usuário pode apropriar-se das produções dos demais para editar, adaptar e criar a partir das mesmas, o que é chamado de remixar.
- **Pensar brincando (Play):** criar e aprender enquanto se brinca e se diverte, de modo que o brincar não seja sinônimo e falta de seriedade na produção. Aliás, uma forma de encarar o trabalho que é adotada cada vez mais no mundo corporativo, em especial nas empresas de tecnologia.















3. Começando no Scratch

Antes de podermos criar objetos virtuais no Scratch, precisamos primeiramente compreender como funciona o Scratch e criar uma conta. Também é importante escolher o equipamento que vamos utilizar para acessar a plataforma e verificar se ele atende aos requisitos para funcionar bem. O ideal é utilizar um computador ou laptop, mas também é possível acessar, com algumas restrições, pelo celular.

Atualmente o Scratch está em sua versão 3.0, que trouxe uma reformulação na interface e algumas funcionalidades em relação à versão anterior. Há também disponível as versões 2.0 e 1.4. Quando a utilização é online, automaticamente a plataforma é carregada com a versão mais atual. Se a opção for utilizar a editor *offline*, é importante escolher entre as três versões disponíveis de acordo com as configurações do seu computador ou laptop. O download pode ser feito no link:

<http://scratch.mit.edu/download>

3.1. Requisitos

Versão 3.0	Versão 2.0	Versão 1.4
 Windows 10+	 Windows XP+	 Windows 2000+
 macOS 10.13+	 Mac OS X+ e Adobe Air	 Mac OS X 10.4+
 Navegador Puffin	 Linux Educacional 4+	 Ubuntu 12.04+
	 Linux 32bits	 Navegador Puffin
	 Navegador Puffin	

Apesar de haver a possibilidade de utilizar o editor Scratch *online* ou *offline*, as funcionalidades são basicamente as mesmas. Inclusive é possível fazer o *upload* das nossas criações no editor *offline* para a plataforma *online*. Assim, podemos aproveitar a íntegra das possibilidades que a plataforma proporciona, como compartilhar nossas criações, conversar com outros usuários, remixar suas criações, criar estúdios, etc.

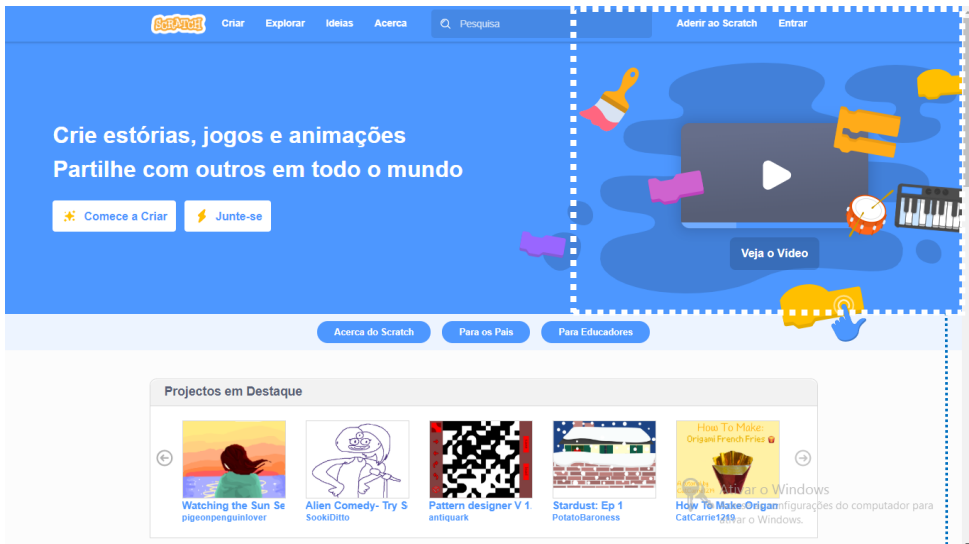
Nesse guia nosso referencial será a plataforma *online*. Faremos referência ao editor *offline* eventualmente, quando nos referirmos a alguma função ou comando que tenha alguma diferença ou restrição no editor *offline* em relação ao *online*. Nas próximas páginas vamos apresentar as etapas para se cadastrar na plataforma *online*, como é sua estrutura e funcionamento.

4. Conhecendo a plataforma

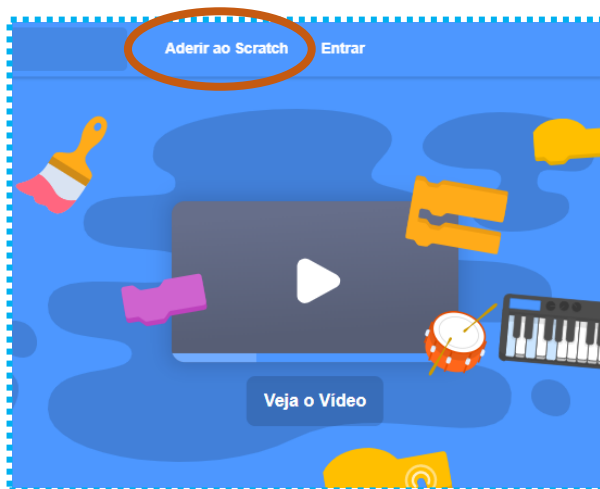
O primeiro passo para começar a se familiarizar com a plataforma Scratch é acessar o site:

<http://scratch.mit.edu>

Você entra na página de apresentação do site, que tem a aparência abaixo:



Clique em “Aderir ao Scratch”.



4.1. Inscrição

Nas janelas que surgirão, siga as etapas de inscrição na plataforma:

1ª Etapa

1. Digite um nome de usuário e uma senha

2ª Etapa

2. Preencha suas informações pessoais

3ª Etapa

3. Informe seu e-mail

4ª Etapa

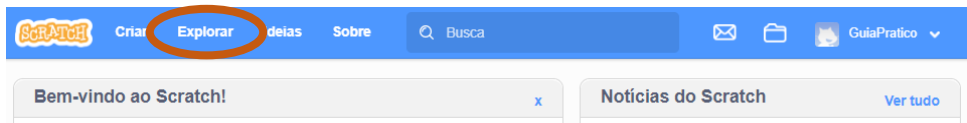
4. Vá até seu e-mail informado e confirme-o no link que vai receber do Scratch.

Vale ressaltar que se você não confirmar seu e-mail poderá utilizar a plataforma apenas para explorar as criações dos demais usuários e fazer suas criações, porém não poderá compartilhar e comentar, ou seja, não poderá utilizar todo o potencial da plataforma. Percorridas todas as quatro etapas mostradas, você receberá ao final a mensagem a seguir e estará tudo pronto para começar a utilizar o Scratch:

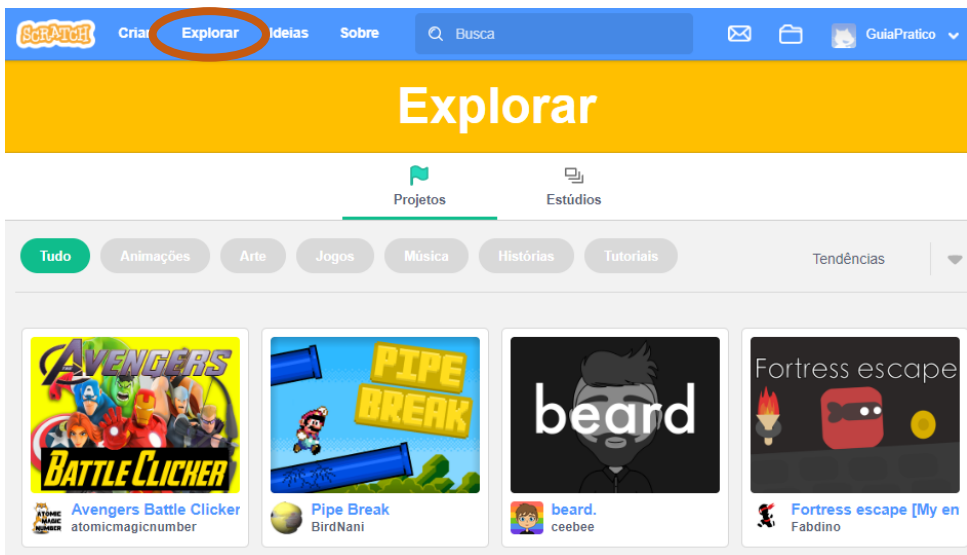
Bem-vindo ao Scratch! Depois de criar projetos e comentários, receberá mensagens sobre eles neste local. Vá a **Explorar** ou a **Criar um projeto**.

4.2. Explorar

Depois de realizar o cadastro, o ideal é explorar a plataforma, experimentar as criações dos demais usuários, verificar como montaram sua programação por blocos e, se alguma estiver de acordo com seus interesses, não há problema em “remixar”. Isso é permitido e incentivado pela plataforma. Vá até a parte superior da página e clique em “Explorar”:



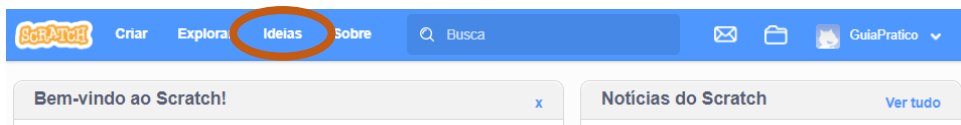
Você será direcionado para a página onde ficam disponibilizadas todas as criações e os estúdios dos usuários. Toda criação vai automaticamente para esse espaço quando é compartilhada e todo estúdio, ao ser criado, também aparece no mesmo. Há algumas possibilidades de filtros para ordenar, por exemplo, pelos mais populares, ou pelos mais recentes. Vale a pena navegar pela página e conhecer melhor, para se ambientar à plataforma.



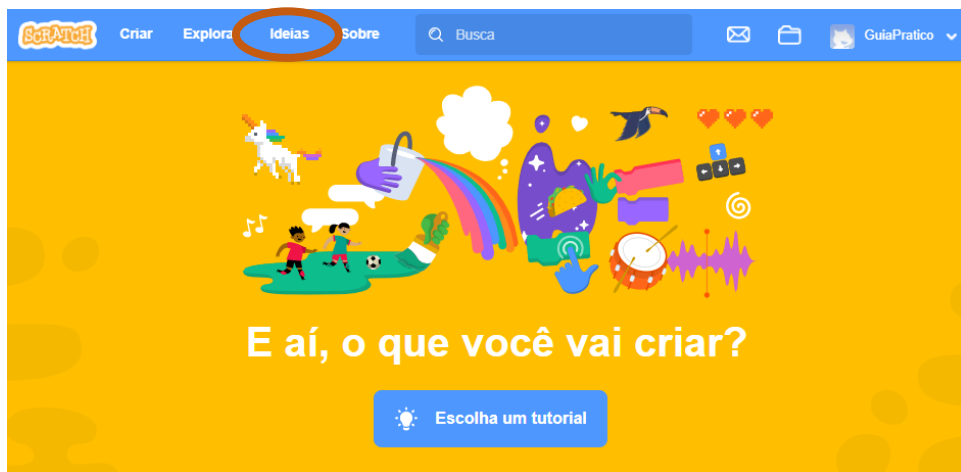
Ao explorar a plataforma você poderá perceber as possibilidades de criação que o Scratch permite. Há animações, jogos, quizzes, cartões animados, apresentações, arte, música, histórias, tutoriais e outras, a critério da criatividade e dedicação do usuário.

4.3. Ideias

Para continuar a exploração da plataforma, é interessante clicar em “Ideias”, na parte superior da página.



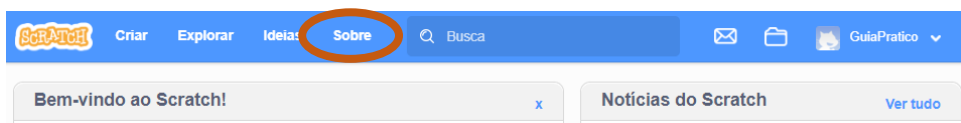
Nessa parte da plataforma você tem acesso a sugestões de atividades para começar a criar com auxílio de tutoriais.



Para o professor que está começando no Scratch essa seção é especialmente importante, pois fornece exemplos de atividades para desenvolver utilizando a plataforma, com opção de acompanhar o tutorial online ou baixar os cartões de atividade e/ou o guia do educador para a atividade para uso *offline*. Dessa forma é possível remixar, adaptar e criar a partir das propostas apresentadas.

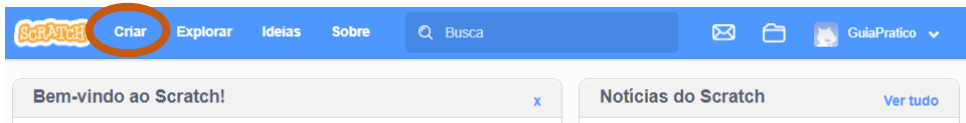
4.4. Sobre

Na seção “Sobre” há informações gerais sobre o Scratch e o link para duas seções que vale a pena serem exploradas: “Informações para os pais” e “Informações para educadores” - essa última por motivos óbvios, no caso desse Guia.

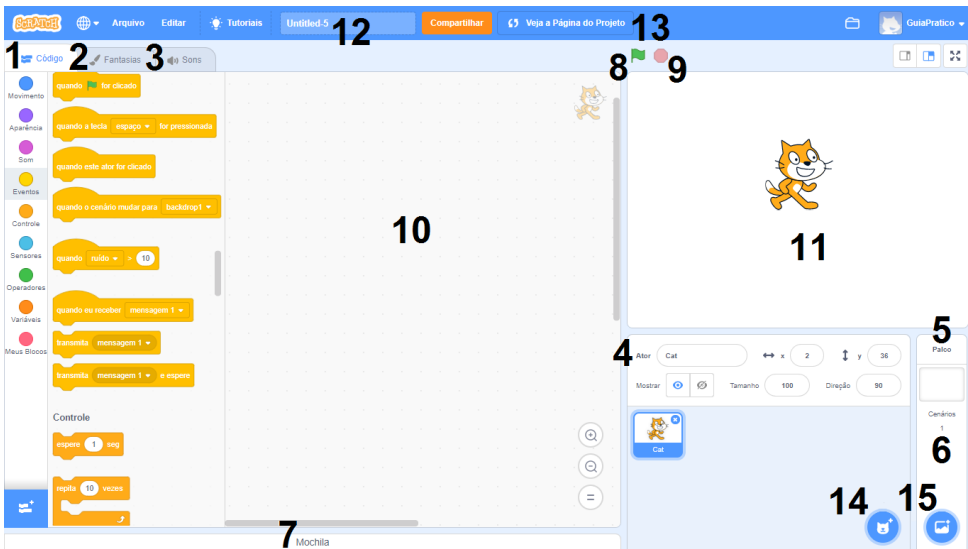


4.5. Editor






Finalmente, vamos explorar agora o editor online do Scratch. É nele onde realmente a mágica acontece, onde fazemos nossa imaginação ganhar vida, onde criamos nossos objetos virtuais. Mas, para isso, precisamos entender sua estrutura e, principalmente, como usar os blocos de programação (assunto que abordaremos no próximo capítulo). Na página do Scratch, é só clicar em “Criar”:



Você é direcionado para a página inicial do editor online do Scratch 3.0. Vamos compreender sua organização e funcionalidades. Ao explorar o editor você vai perceber que os itens têm algumas denominações bem típicas e não óbvias para quem está iniciando. A seguir listamos as principais denominações e suas funcionalidades, para facilitar a compreensão do editor.



1. **Aba Código**: é onde estão organizados os blocos de programação, formados por nove categorias distintas, organizadas por cores;
2. **Aba Fantasia**: é a área onde podemos editar as imagens dos personagens e cenários utilizados no projeto, fazendo as alterações que nos interessarem;
3. **Aba Sons**: é o local destinado à edição dos sons utilizados no projeto, sejam eles associados aos personagens ou aos cenários;

4. **Ator:** é cada personagem ou objeto que utilizamos no projeto, que podem ser modificados na aba “Fantasias”;
5. **Palco:** é a área que aciona os cenários para que possamos editá-los na área de programação;
6. **Cenários:** são os planos de fundo que utilizamos no nosso projeto. Quando a edição de cenários está ativa a aba “Fantasias” muda de nome para “Cenários”;
7. **Mochila:** é a área do editor que comporta as três abas mencionadas e mais o espaço de programação;
8. **Botão Ir** : é onde se inicia a execução do projeto na Área de visualização, para que possamos testar se tudo está saindo de acordo com o que planejamos;
9. **Botão Pare** : é o botão que para a execução do projeto na Área de visualização;
10. **Área de programação:** é o espaço onde podemos adicionar, ver e editar os blocos de programação utilizados para cada personagem ou cenário;
11. **Área de visualização:** é a área que funciona como uma mini tela, onde podemos visualizar e testar a execução do projeto;
12. **Nome do projeto:** é o espaço reservado para colocarmos o nome do nosso projeto – por padrão ele vem com a denominação *Untitled*;
13. **Botão**  **Veja a Página do Projeto:** é o botão que, quando clicado, alterna entre o modo de edição do projeto e o modo de compartilhamento, que mostra o projeto como os outros usuários irão vê-lo na plataforma;
14. **Botão Selecione um ator** : é o botão que mostra as opções que temos para escolher, criar ou adicionar personagens ao nosso projeto;
15. **Botão Selecionar Cenário** : é o botão que mostra as opções que temos para a escolher, criar ou adicionar cenários ao nosso projeto

Além de compreender as funcionalidades apresentadas, é importante compreender as funções de cada categoria de blocos de programação e, claro, como criamos o código que vai dar vida aos atores e cenários do nosso projeto. Também é essencial conhecer as características da área de programação para poder fazer uso correto dos blocos. Assim, evitamos que o projeto tenha algum erro.

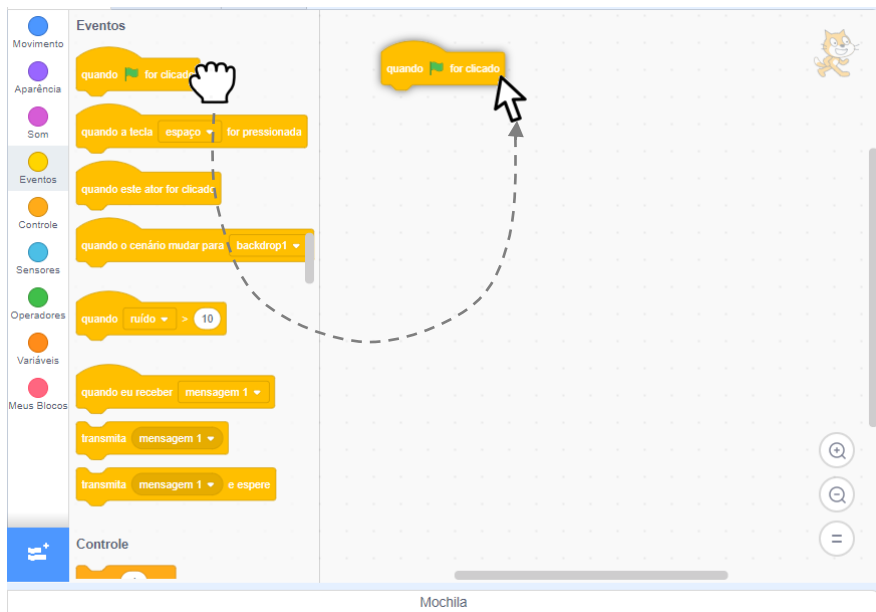
4.6. Programar no Scratch

O Scratch foi pensado para permitir que programar se tornasse fácil a ponto de uma criança conseguir fazer. Inspirados nos blocos Lego, foram criados os blocos de programação. Para “escrever” um comando para ser executado não é necessário escrever longas linhas de código numa linguagem enigmática para o leitor leigo. Basta juntar os blocos certos e associá-los ao ator ou cenário que se quer programar.

No início pode parecer complicado para quem está aprendendo, mas, depois de alguns experimentos torna-se um desafio até divertido. Para começar, precisamos conhecer a área de programação, os blocos e suas funções. Há nove categorias de blocos no editor do Scratch. Antes de mais nada, é importante entender a área de programação, pois muitos dos blocos fazem referência a ela. Vamos conhecê-la?

4.6.1. Área de programação

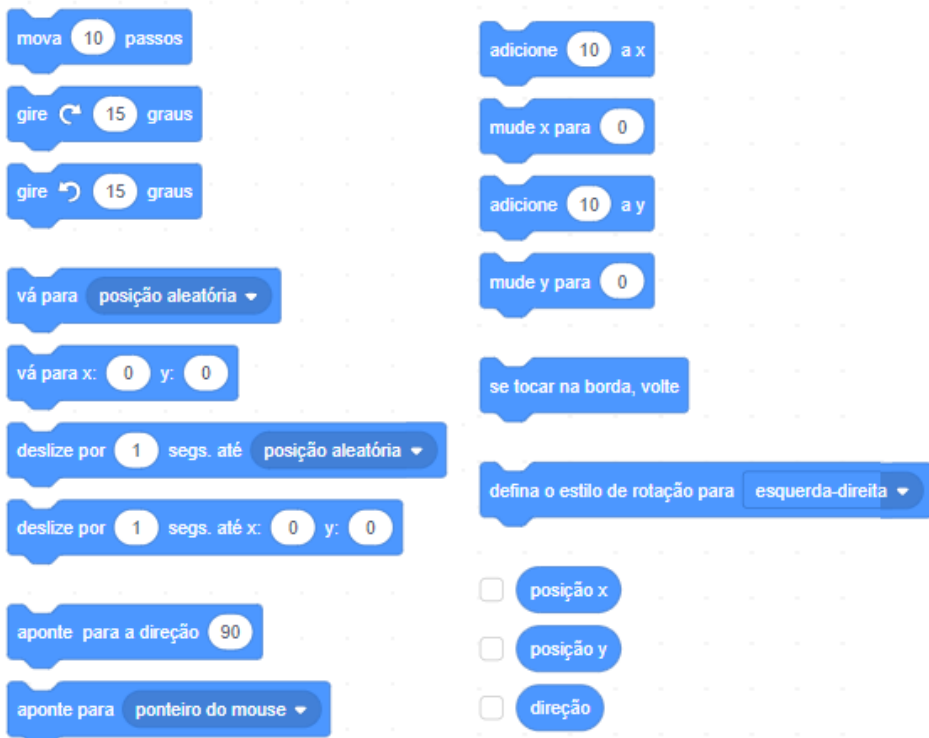
É o espaço logo à direita dos blocos de programação na mochila. Feita a escolha dos blocos que se vai utilizar, basta clicar, segurar com o mouse e arrastar cada um para essa área, “montando” a programação de acordo com o seu interesse.



Se por acaso mudar de ideia e quiser excluir algum bloco, basta arrastá-lo de volta para a área de blocos ou dar um clique nele e depois apertar a tecla **delete** no teclado e: pronto. O bloco foi apagado! Outra opção é clicar no bloco com o botão direito e escolher “apagar bloco” na caixa que aparece. Aprenderemos exemplos de como montar a programação na Parte II. Vamos agora aos blocos de programação.

4.6.2. Blocos de Movimento

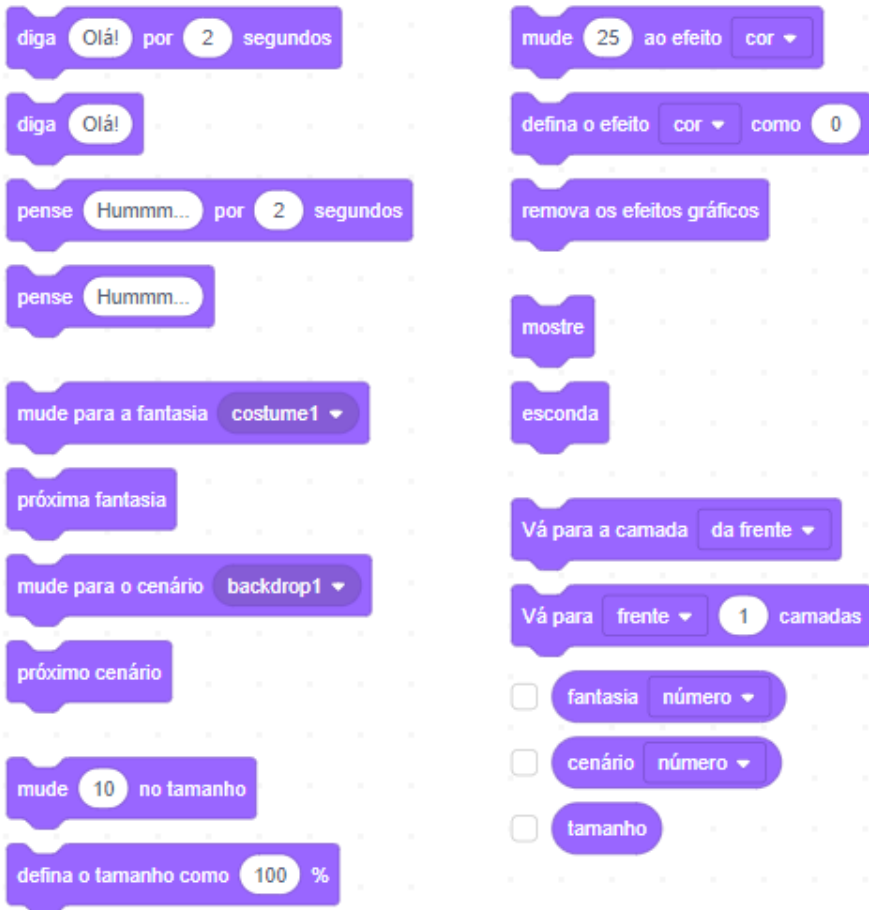
Os blocos de movimento são relacionados ao posicionamento e deslocamento dos componentes (atores, palcos) pelo espaço da tela. São no total dezoito blocos:



Utilizando esses blocos, podemos determinar a velocidade do movimento, para que parte da tela o componente deve se descolar, fazê-lo girar e determinar o que acontece quando o mesmo tocar na borda da tela. São informações muito importantes para se levar em consideração quando o objeto virtual que estivermos programando exigir movimentos. Na Parte II vamos entender na prática o funcionamento dos blocos, a partir dos exemplos de atividades que serão propostos para você criar.

4.6.3. Blocos de Aparência

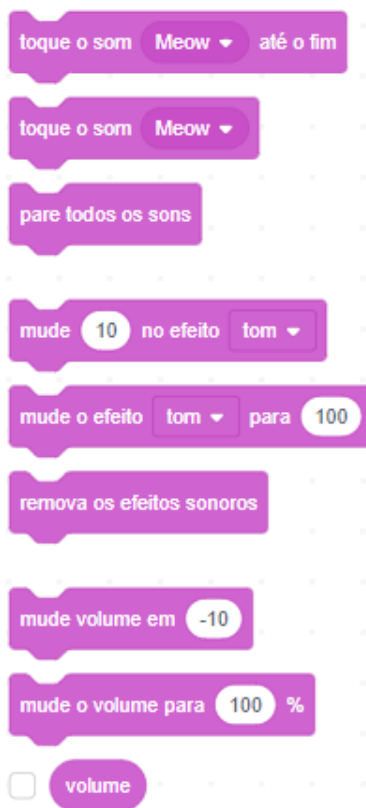
Os blocos de aparência são vinte e relacionam-se ao aspecto visual dos componentes, como tamanho, cor, aparecer ou desaparecer, camada de posicionamento, ações de fala escrita e mudanças de fantasia ou de cenário. Observe-os:



Com esses blocos, determinamos se um ator fala ou pensa alguma coisa de forma escrita e por quanto tempo o texto permanece na tela. Podemos também determinar uma mudança de fantasia do ator ou da imagem do cenário. É possível fazer o personagem aumentar ou diminuir de tamanho ou mudar sua cor. Enfim, uma diversidade de ações possíveis, claro, em associação com os demais blocos existentes.

4.6.4. Blocos de Som

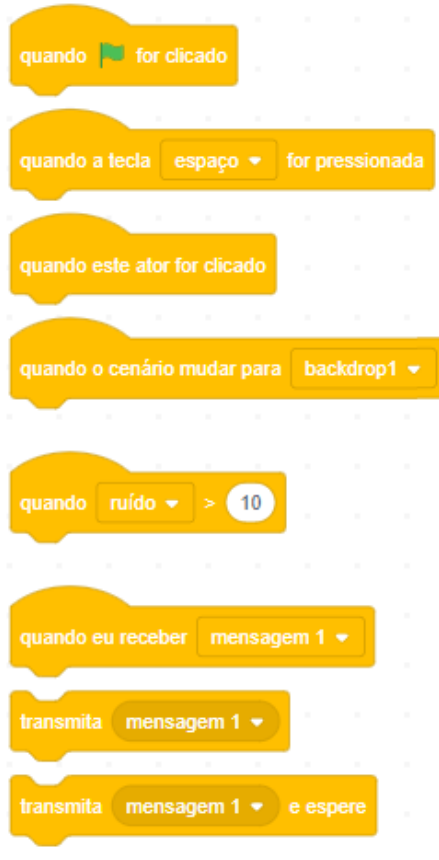
É bem provável que ao ler o item anterior, muitos leitores já tendam a se questionar se os atores só se expressam por meio escrito no Scratch. A resposta para esse questionamento é não! Além da expressão escrita, é possível também atribuir sons, tanto para simular a fala como para simular efeitos sonoros das ações e interações dos atores e dos cenários. Os blocos de som, num total de nove, são:



Com esses blocos podemos determinar que um ator ou cenário execute um som. O Scratch já traz embutido na plataforma alguns efeitos sonoros, mas também é possível importar um som externo, como, por exemplo, uma gravação de voz. Com os blocos adequados, é possível aumentar ou diminuir o volume e o tom dos efeitos sonoros. Além disso, a plataforma também tem um editor de som que, apesar de relativamente simples, permite modificar os sons. Por exemplo, podemos fazer o *upload* de uma música e editá-la para utilizar apenas um trecho específico.

4.6.5. Blocos de Eventos

Os blocos de evento são oito e são de extrema importância na hora de realizarmos a programação dos componentes de nossas criações. Utilizamos sempre um bloco de evento no início de um grupo de blocos para determinar a condição para que a programação em questão comece a ser iniciada, por exemplo. Utilizados em conjunto com os demais blocos, eles marcam a situação específica que ativa o código. Conheça-os abaixo:



Se não utilizarmos um bloco de evento, por exemplo, uma ação pode ocorrer de forma indefinida ou mesmo nem ocorrer. Na Parte II, ao praticar criando com os exemplos que serão dados, você vai perceber que o tempo todo vamos precisar lançar mão de algum bloco de evento para ajudar a estipular quais situações e interações entre os componentes ou entre esses e o usuário conduzem a determinada ação dos atores ou dos cenários.

4.6.6. Blocos de Controle

Esses blocos têm a função de organizar a execução das ações dos componentes, por exemplo, determinando quantas vezes determinada ação será realizada. É o caso do bloco denominado “repita ___ vezes”, o segundo na coluna da direita abaixo. Todos os demais blocos que estiverem englobados por ele repetirão suas ações tantas vezes quantas estiverem estipuladas por você. Eles ajudam também, em associação com outras categorias de blocos, a determinar o condicionante para que a ação aconteça. Repare abaixo em sua aparência e nos comandos que representam:



Os três últimos blocos dessa categoria têm uma função especial. São direcionados para lidar com os “clones”, que são as duplicatas de algum ator que pode surgir devido a alguma condicionalidade ou mesmo à programação dos blocos. Eles lembram os blocos de eventos, mas por uma determinação do programador do Scratch, ficaram agrupados nessa categoria de controle.

4.6.7. Blocos de Sensores

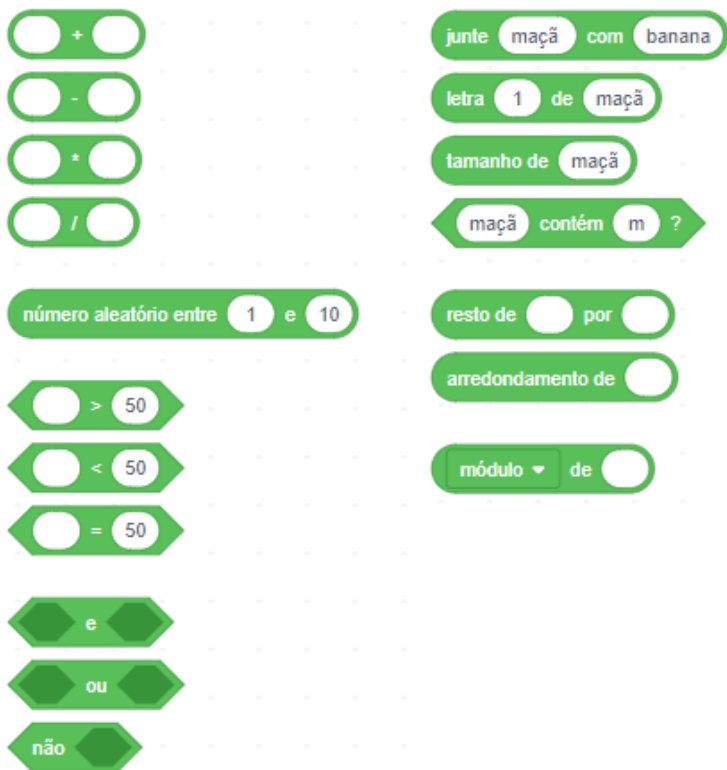
Pelo uso dos dezoito blocos que compõe essa categoria podemos estipular o tipo de interação que condiciona, associada com as demais categorias de blocos, as ações dos atores e cenários. Essas interações podem ser, por exemplo, um ator de um jogo se aproximar de outro ou tocá-lo. Também poderia ser o usuário clicando com o mouse ou apertando determinada tecla e isso ser o fator determinante para uma ação de algum componente. Confira os blocos de sensores logo abaixo:



Os formatos dos blocos de sensores já denunciam que alguns têm locais específicos para se encaixarem junto aos blocos de outras categorias. Por exemplo: o bloco “tocando na cor...?” necessariamente deve associar-se aos blocos de controle para determinar o que ocorre se o ator em questão tocar na cor escolhida por você. Alguns dos blocos de controle que têm formato arredondado podem ser acoplados a outros blocos das categorias anteriores para determinar um valor numérico específico, como por exemplo o ano atual. Como já foi citado, na Parte II, ao colocar a mão na massa e começar suas criações próprias ou a partir dos exemplos que serão apresentados, você compreenderá melhor as possibilidades que os blocos de sensores proporcionam.

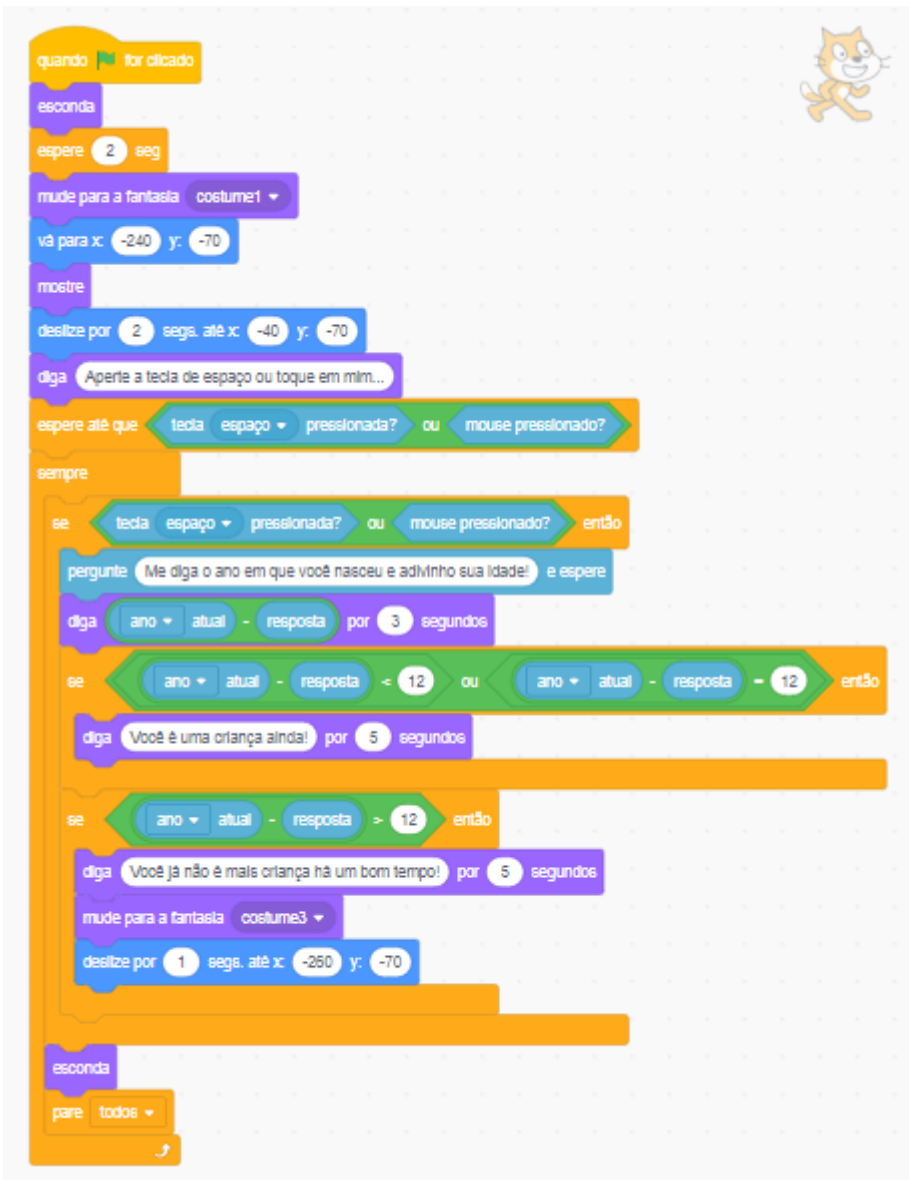
4.6.8. Blocos de Operadores

Os blocos de operadores têm função lógica e matemática, utilizada em associação com os demais grupos de blocos. Essa utilização associada dos grupos de blocos é importante quando se quer determinar números que dependem de outros componentes, sua posição na tela, entre outros fatores. São dezoito os blocos de operadores, que podemos conferir a seguir:



O uso de todos os grupos de blocos que vimos até aqui já permite desenvolver muitas criações interessantes. Associando os diversos blocos, podemos criar por exemplo objetos virtuais que interajam com o usuário. Mas esse já é um passo além do usuário iniciante, que você poderá alcançar sem problemas por meio de sua dedicação à prática.

Vamos ilustrar a seguir com o código pronto de um objeto interativo que aprenderemos mais à frente como fazer, nos exemplos deste guia. Você pode visualizar os diversos tipos de blocos trabalhando em conjunto para executar algumas ações a partir da interação com o usuário. Nesse objeto, criamos um código que faz o ator perguntar para o usuário o ano em que nasceu e, a partir da resposta, executar mais duas ações.



Conforme você pode conferir no código acima, utilizamos todos os grupos de blocos que apresentamos até agora, exceto os de som. Nesse código, que está associado ao personagem do gato (à mostra em cima, no canto direito), sempre que a aplicação for iniciada (🚩), ao apertar a tecla **espaço** ou clicar com o mouse, o gato diz:

Me diga o ano em que você nasceu e adivinhe sua idade!

Surge uma caixa de diálogo para o usuário digitar o ano no qual nasceu, que o código imediatamente diminui do ano atual. Então, o gato diz a idade da pessoa e, se a pessoa tem até 12 anos, o gato diz:

Você é uma criança ainda!

Se a pessoa tem mais de 12 anos, o gato diz a idade e complementa assim:

Você já não é mais criança há um bom tempo!

O código apresentado não é um que um iniciante construiria, mas também não é tão difícil de ser construído quanto aparenta. Com tempo e prática, vamos conseguindo desenvolver criações cada vez mais diversificadas e complexas. E não se intimide com as falhas e os erros: são parte do aprendizado. Não desanime caso eles ocorram. E, tenha certeza: eles vão ocorrer. Caso você queira testar a criação que apresentamos acima, utilize o leitor de QR Code para ler o código ao lado ou acesse pela *web*: <http://scratch.mit.edu/projects/311392152>.



Exemplo 1 Ver interior

Instruções

Esse é o primeiro exemplo do nosso Guia Prático.

Para começar, clique na bandeira verde!

Notas e créditos

Scratch: Guia Prático para aplicação na educação básica
 2019 © SOUZA, Michel Figueiredo de; COSTA, Christine Sertã
 Programa de Mestrado Profissional em Práticas de Educação Básica do Colégio Pedro II, Rio de Janeiro,

0 0 0 13 © 25 de mai de 2019 Adicionar ao Estúdio Copiar o Link

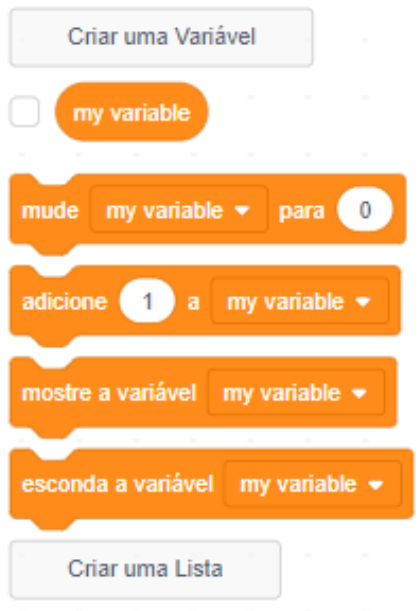
Comentários Comentários ativados

Publicar Cancelar 500caracteres restantes.

4.6.9. Blocos de Variáveis

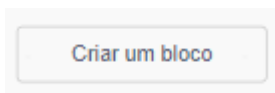
Além dos sete grupos de blocos de programação que vimos anteriormente, também existem os blocos de variáveis. Normalmente utilizamos variáveis em projetos mais complexos e avançados. Elas servem para armazenar números e sequência de letras na memória do computador, como o resultado de um cálculo matemático, ou o nome do usuário, por exemplo. Pode ser utilizada para um ator se referir ao usuário pelo nome, por exemplo.

Como esse guia tem um caráter introdutório e de conhecimentos básicos sobre o Scratch, não nos aprofundaremos na utilização das variáveis, apesar de também serem apresentadas na Parte II. Os blocos de variáveis básicos são os cinco mostrados abaixo. Conforme vamos criando variáveis, surgem outros blocos como o primeiro que vemos, mas com o nome que escolhemos para a variável. Dessa forma, podemos utilizar a variável em conjunto com os demais blocos de programação.





4.6.10. Meus Blocos

Nessa área, antes que criemos blocos novos, apenas aparece o seguinte botão:



Esse grupo de blocos serve para criarmos blocos novos e agruparmos outros grupos de blocos que executam comandos que vamos utilizar mais vezes, simplificando a programação posterior. Assim, quando formos reutilizar o mesmo tipo de comando, bastará adicionar apenas o novo bloco único criado, evitando ter de repetir todo um grupo enorme de blocos de programação novamente.

Por exemplo, criamos um longo grupo de blocos de programação para definir que um ator se move para a direita quando pressionamos a tecla , para a esquerda quando pressionamos a tecla  e pula quando pressionamos a tecla espaço. Se depois quisermos copiar o mesmo tipo de forma de movimentação para outros atores, podemos criar um bloco novo, nomear como “movimento” e agrupar todos os comandos de movimentação abaixo dele. Depois, bastará adicionar o bloco “movimento” para os outros atores e economizamos um tempo precioso fazendo o código longo.

Nos exemplos que serão apresentados na Parte II, vamos abordar algumas possibilidades de criação de novos blocos, o que pode facilitar muito nossa vida na hora de desenvolver alguma criação que tenha comandos que se repetem inúmeras vezes ao longo da programação.



Hora de colocar a mão na massa!

Agora que apresentamos a área de programação e os blocos, que tal partirmos para a prática e começarmos a aprender a fazer nossas próprias criações? Na próxima parte do guia, vamos apresentar alguns exemplos, partindo de exemplos simples a exemplos mais complexos de atividades que podem ser desenvolvidas no Scratch, sempre buscando alguma relação com os conhecimentos do currículo escolar.

Ao desenvolver essas atividades, você estará aprendendo pouco a pouco a utilizar a programação por blocos do Scratch e percebendo as possibilidades que a plataforma permite para criação. Obviamente, nosso foco será permitir que o leitor desenvolva a capacidade básica de programação com o Scratch e, a partir daí, lançar mão de sua imaginação, interesses e empenho para se aprofundar ainda mais.

Vamos lá!

Quando  virar página

Iniciar

Parte II

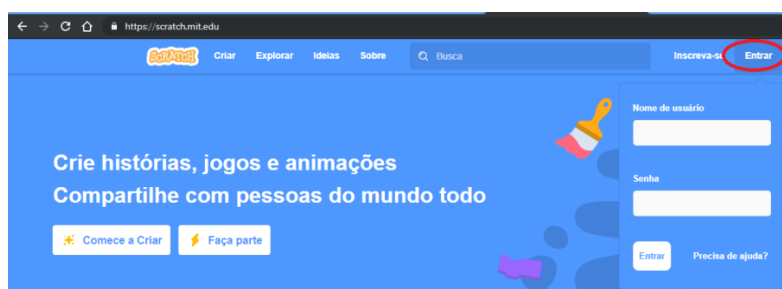
II. Programando com o Scratch – Exemplos

Atividade 1: Animar seu nome

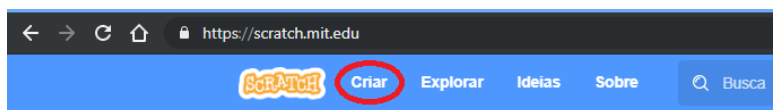
Uma atividade bem simples, para o usuário começar a utilizar a plataforma de maneira bem *light*. Consiste em inserir as letras do seu nome e fazê-las se mover pela tela e piscar, mudando de cor.

Passo-a-passo

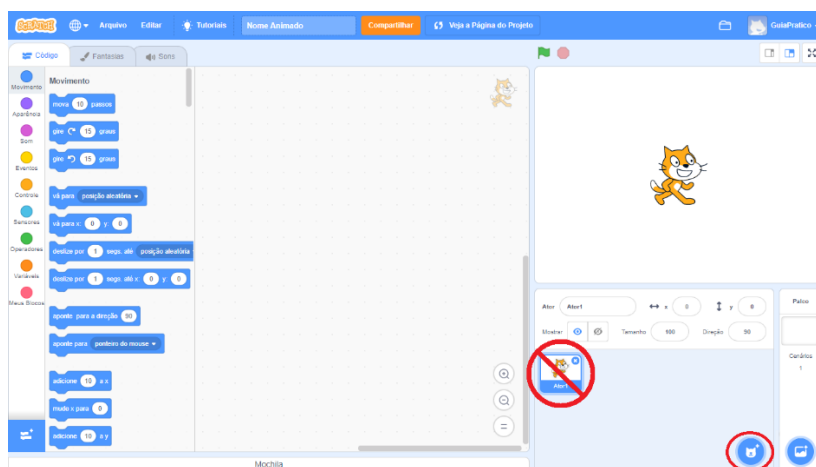
1. Abra a página do Scratch, faça login:



2. Login feito, clique em “Criar”:



3. Vai entra na página de programação do projeto (nomeamos como Nome Animado). O ator do gato sempre aparece por padrão. Apague-o. Em seguida clique no ícone para adicionar outros atores e escolher as letras do seu nome.



4. Na janela que vai abrir, escolha a aba letras e adicione as letras do seu nome. Infelizmente, precisa adicionar uma a uma. Quando terminar, sua tela aparecerá assim:

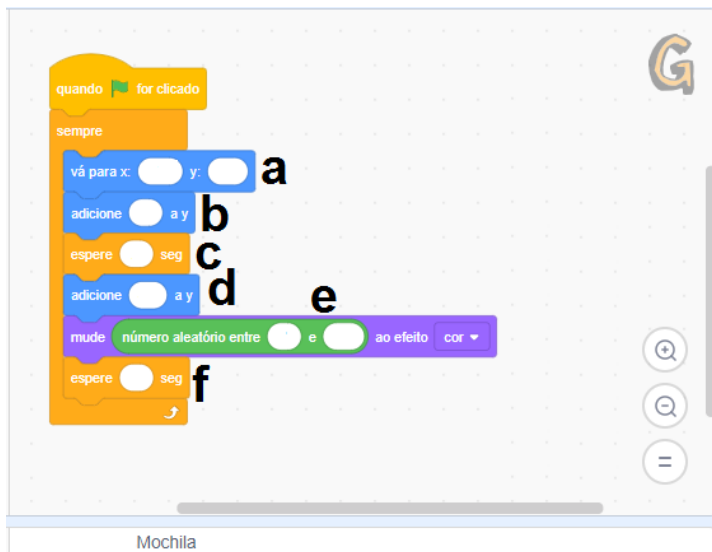


Agora é hora de criar a programação que fará os atores (letras) terem o efeito animado de se movimentarem e mudarem de cor.

5. Você vai precisar utilizar os seguintes blocos:

Movimento	vá para x: <input type="text"/> y: <input type="text"/> adicione <input type="text"/> a y
Aparência	mude <input type="text"/> ao efeito cor ▾
Som	Nenhum
Eventos	quando for clicado
Controle	sempre espere <input type="text"/> seg
Sensores	Nenhum
Operadores	número aleatório entre <input type="text"/> e <input type="text"/>

6. Selecione a primeira letra (G, no nosso exemplo) e organize os blocos na área de programação da maneira como é mostrado abaixo. Lembre-se que precisa clicar no grupo de blocos, escolher o que interessa, clicar e arrastá-lo para a área de programação. Em seguida, vá encaixando os blocos adequadamente.



Nessa programação, estamos dizendo para a letra G, quando a bandeirinha verde for clicada (ou seja, quando mandarmos executar), realizar “sempre” todas as ações que estão englobadas por ela. Sendo assim, as mesmas se repetem indefinidamente ou até que cliquemos no círculo vermelho (ou seja, quando mandarmos parar a execução).

Mas precisamos ainda preencher os blocos marcados com as letras de “a” a “f” com os valores adequados para que a programação das letras funcione corretamente. Siga as orientações abaixo.

- a- Depois de posicionar as letras na área de visualização, quando estiver montando a programação e for selecionar esse bloco, ele já virá preenchido com a posição da letra na tela, definida por uma coordenada “x” (posição horizontal) e uma coordenada “y” (posição vertical), basta mantê-la e o bloco sempre iniciará a execução na mesma posição. Abordaremos a configuração da área de visualização mais à frente;
- b- Nesse bloco vamos adicionar um valor que faça o valor da coordenada “y” aumentar ou diminuir. No nosso exemplo, colocamos o valor 20, que fará a letra subir um pouco na tela, mas você pode testar outros valores e personalizar;
- c- Esse bloco fornece um valor de tempo para que a programação faça uma pausa, para em seguida continuar os comandos seguintes, isso que dá o efeito de piscar na hora da mudança de cor. No nosso exemplo, colocamos o valor 0.5, ou seja, meio segundo (se você usar vírgula ocorrerá erro, pois o Scratch trabalha com

o ponto como marcador decimal). Você pode testar outros intervalos de tempo a seu critério;

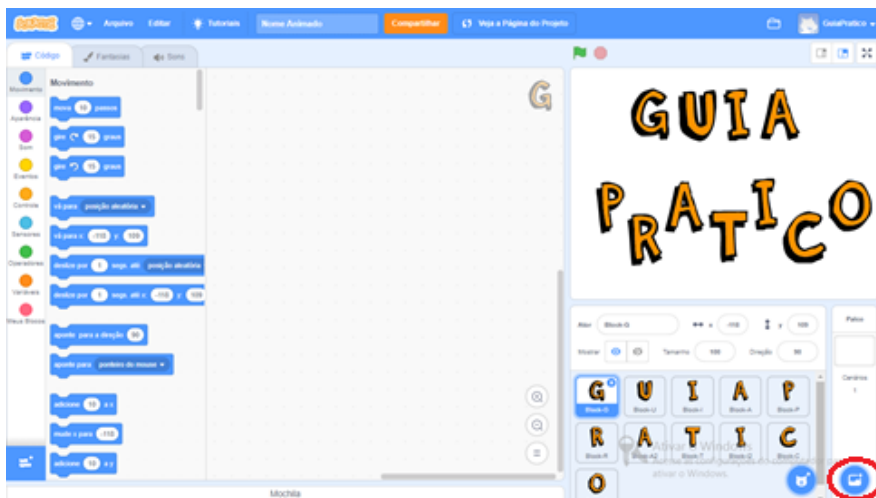
- d- Esse é o mesmo bloco da orientação **b**, mas agora vamos adicionar um valor que faça o valor da coordenada “y” diminuir. Colocamos o valor -20, que fará a letra descer um pouco na tela, voltando à posição original. Sinta-se à vontade também aqui para experimentar outros valores;
- e- No Scratch, os valores que correspondem às diversas cores variam de 1 a 100. Nesse caso, como queríamos que toda possibilidade de cor pudesse ocorrer sem restrições, preenchemos esse bloco operador com esses valores. Repare que esse bloco está encaixado no bloco de aparência que utilizamos. Você também pode testar outros intervalos se preferir e verificar o resultado;
- f- Aqui repetimos o que fizemos na orientação **c** e, claro, você pode testar à vontade se outros intervalos de tempo ficarão mais interessantes para você.

7. Agora que sua programação está completa, basta fazer o mesmo com cada letra e sua criação estará pronta para ser executada. Nesse ponto você deve ter pensado: “Esse trabalho todo vou precisar repetir em todas as letras?”. Só se você quiser. Preste atenção às dicas!

Atenção para as dicas!

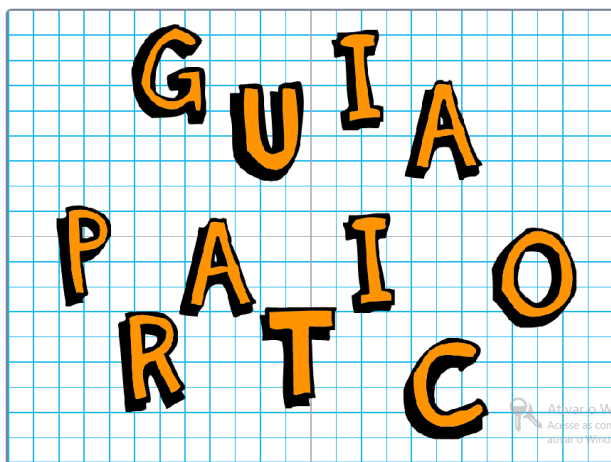
- I. Nesse exemplo todos os atores terão a programação igual, então, fique esperto: não tenha trabalho desnecessário! Basta fazer a programação para o primeiro ator, depois copiá-la direto para os demais! Clique no bloco superior da programação montada, isso a selecionará inteira e ativará a execução do código. Aperte no teclado as teclas **Ctrl** + **C** para copiá-la. Clique na próxima letra a ser programada e aperte no teclado as teclas **Ctrl** + **V** para colar o código na área de programação. Feito isso, é só ajustar os valores dos blocos copiados para as outras letras, quando for o caso.
- II. Quando copiar a programação completa para as demais letras ajuste o valor do bloco de movimento “vá para x... y...” para a posição da nova letra, caso contrário ela repetirá a posição da primeira. Ou seja: na hora que mandar executar, todas vão se sobre por na área de visualização e você precisará arrumar a bagunça! Repare que quando você clica na letra na relação de atores, a caixa de informações se preenche com seus dados. Preencha o bloco de movimento “vá para x... y...” com esses valores numéricos e a letra manterá essa posição.
- III. No nosso exemplo, alternamos para cada letra os valores dos blocos de movimento “adicione ... a y”, assim as letras alternam o movimento. Experimente os valores que preferir e veja o resultado que mais lhe agrada. Não tenha medo de experimentar!

8. Depois de terminar a programação de todos os blocos, se você quiser, pode adicionar um cenário à sua área de visualização, para incrementar sua criação. Basta clicar no botão “Selecionar cenário” e escolher o que mais lhe agrada:



Feito! Sua criação está pronta para ser executada e compartilhada com os demais usuários do Scratch!

Você pode conferir o nosso exemplo na íntegra no perfil do Guia Prático no Scratch. Lá você pode executar e testar como ficou nossa criação, ver o interior para conferir como ficou nosso código e até copiá-la para seu perfil no Scratch e remixar! Fique à vontade! Quanto mais você experimentar, mais vai aprender.



<http://scratch.mit.edu/projects/312064812>



*Se quiser visualizar
pelo celular, leia o
QR Code.*

Atividade 2: Veloz e Furioso

A segunda atividade vai um passo além nos conhecimentos da plataforma: será um jogo bem simples. Desta vez teremos dois atores (carro verde e carro roxo) e precisaremos adicionar um cenário (cidade com autopista). Cada ator será controlado por um usuário, logo, é um jogo para duas pessoas. O cenário também ganhará códigos para interagir com os atores.

A ideia é que os carros apostem uma corrida, na qual o primeiro que chegar ao lado oposto da tela vence. O código para ambos os carros é praticamente igual, só havendo diferença na tecla que deve ser pressionada para provocar o movimento e na mensagem que envia quando chega ao lado oposto da tela. Topa o desafio?

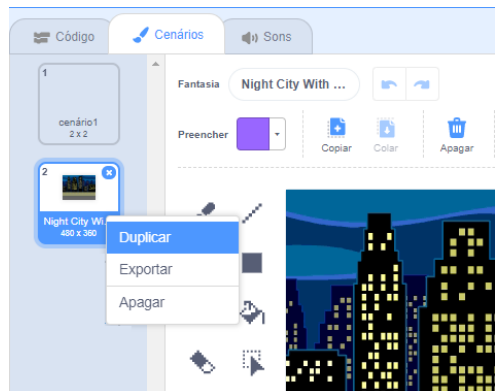
Passo-a-passo

1. Repita as etapas de 1 a 3 da Atividade 1;
2. Clique em “Selecionar um ator” e, no espaço de pesquisa da janela que vai abrir, digite “car”. Irão aparecer alguns atores relacionados.

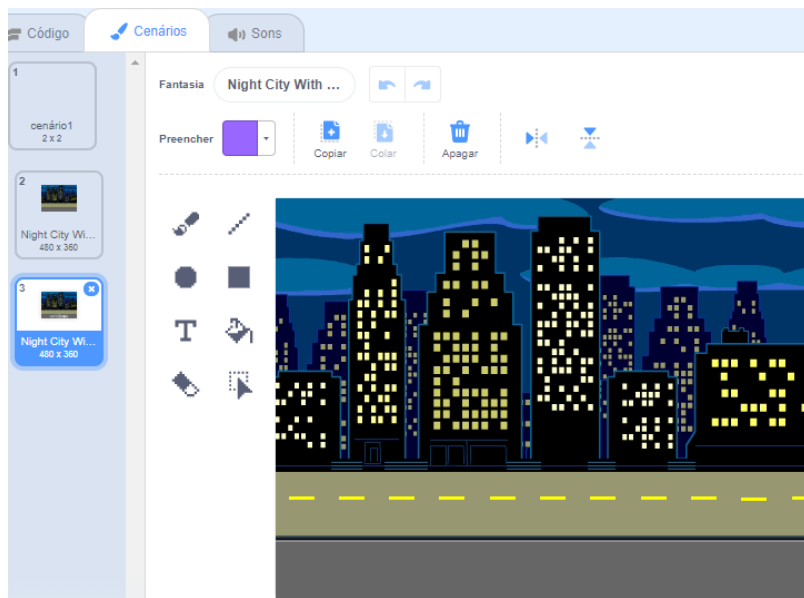


Adicione o ator “Convertible”. Em seguida, repita as ações e adicione também o ator “Convertible 2”. Serão os dois carros que disputarão nesse nosso primeiro jogo.

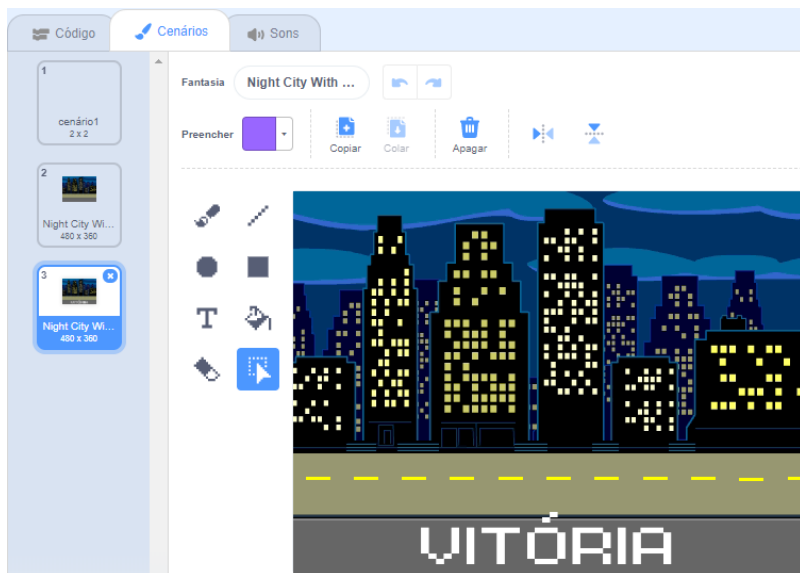
3. Clique em “Selecionar cenário” e, na janela que vai abrir, procure pelo cenário “Night City With Street” e adicione-o.
4. Ainda com o palco ativo, clique na aba “Cenários”;
5. Clique com o botão direito sobre a miniatura do cenário que escolhemos à esquerda e, no menu que aparecerá, selecione “Duplicar”. Agora você terá dois cenários iguais da cidade à noite. Agora nós vamos fazer uma alteração na cópia feita do cenário, que será importante para o funcionamento do jogo;




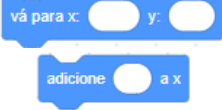

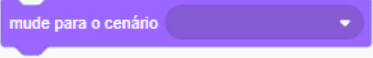
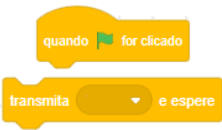
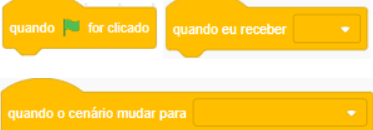


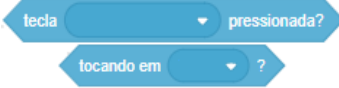

6. Selecione a cópia do cenário que você fez. Ela vai aparecer tanto na área de visualização como na área de edição do cenário, que surge onde seria a área de programação dos atores;



7. Utilizando o editor de texto e o botão de preenchimento, escreva “Vitória” (escolha a cor e a fonte de sua preferência – no nosso exemplo, usamos fonte Pixel e cor branca). Amplie o tamanho e posicione na tela (no exemplo, escolhemos posicionar na parte cinza escura na parte de baixo do cenário);



8. Feito isso, já temos nossos atores e nosso cenário. Agora é só criar os códigos. Você vai precisar dos seguintes blocos dessa vez:

	Atores	Cenário
Movimento		Nenhum
Aparência		
Som	Nenhum	Nenhum
Eventos		
Controle		
Sensores		Nenhum
Operadores		Nenhum

Na hora de fazer a programação, você repetirá o mesmo código para ambos os carros e um código à parte para o cenário. A única coisa que será diferente para cada carro é o valor da posição “x” e “y”, a referência da tecla que será pressionada no teclado para fazer com que se movimentem e a porcentagem de redução de tamanho de cada ator para ficarem adequados e de tamanho similar na tela.

9. Para cada ator, organize os blocos de código conforme nos exemplos abaixo. Fique atento para os valores indicados em cada bloco. Claro que, se você preferir, pode adaptar o projeto de acordo com suas preferências. Mas, para que funcione conforme criamos, é importante ficar atento aos detalhes.

Carro verde



```

quando for clicado
  defina o tamanho como 65 %
  vá para x: -186 y: -94
  mostre
  sempre
    se tecla seta para direita pressionada? então
      adicione 5 a x
      espere até que não tecla seta para direita pressionada?
    se tocando em borda ? então
      transmita Verde e espere
  
```

Carro roxo



```

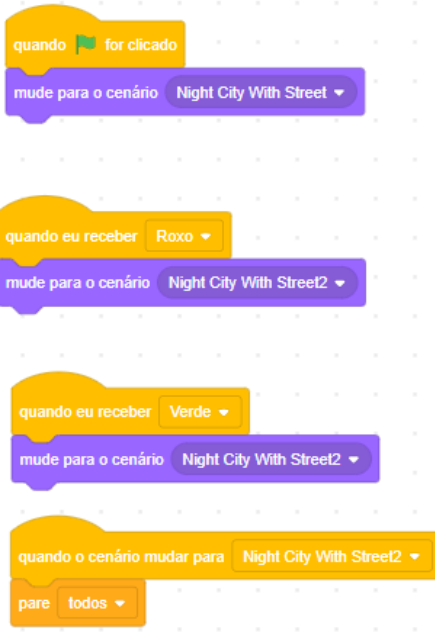
quando for clicado
  defina o tamanho como 50 %
  vá para x: -188 y: -70
  mostre
  sempre
    se tecla espaço pressionada? então
      adicione 5 a x
      espere até que não tecla espaço pressionada?
    se tocando em borda ? então
      transmita Roxo e espere
  
```

Passo-a-passo

- Quando colocamos os valores nos botões de aparência “defina tamanho como ... %”, o carro verde se reduz para 65% de seu tamanho original e o roxo, para 50%. Assim, nesse caso, ambos ficam de um tamanho adequado ao cenário escolhido e do mesmo tamanho na tela;
- Quando colocamos os valores para “x” e “y” nos botões de movimento “vá para x: ... y: ...”, determinamos que o carro roxo ficará na metade de cima da via e o verde na metade de baixo, ambos no limite da esquerda da tela, para se deslocarem para a direita durante o jogo;
- Nos botões de sensores “tecla ... pressionada?”, colocamos **seta para direita** no código do carro verde e **espaço** no código do carro roxo. Isso quer dizer que essas teclas devem ser pressionadas durante o jogo para que o movimento de cada ator aconteça;
- No botão de movimento “adicione ... a x”, colocamos um valor positivo que fará o carro se deslocar para a direita sempre que a tecla **espaço** ou **seta para direita** for pressionada, a depender do carro;
- Em seguida, acoplamos o botão de sensor “tecla ... pressionada?” no botão de operadores “não ...” e ambos no botão de controle “espere até que ...”. Isso quer dizer se o usuário mantiver a tecla **espaço** ou **seta para direita** pressionada, o carro não ficará se movimentando indefinidamente – é necessário apertar e soltar a tecla para que o carro se movimente de cada vez;
- Depois acoplamos o botão de sensor “tocando em ...?”, indicando a borda, e o botão de controle “se ... então ...” e agrupamos o botão de evento “transmita ... e espere”, criando

uma mensagem chamada “Verde”, no caso do carro verde, e “Roxo”, no caso do carro roxo. Essa mensagem é apenas um sinal virtual nomeado que serve para ser referência para outros comandos de código, sem estar visível para o usuário. Nesse caso, será referência para o código que será aplicado ao cenário, como veremos adiante.

10. Para programar o cenário, clique sobre ícone do cenário correspondente na área do palco, o organize os blocos conforme mostramos a seguir:



- Aqui, determinamos que quando o jogo for iniciado, ative o cenário sem a inscrição “Vitória” (lembre-se que nas orientações de 4 a 7 criamos dois cenários para esse palco);
- Aqui, determinamos que se o carro roxo chegar primeiro à borda (quando ele envia a mensagem “Roxo”), o cenário mude para o que tem a inscrição “Vitória”;
- Aqui, determinamos que se for o carro verde a chegar primeiro à borda (quando ele envia a mensagem “Verde”), o cenário também mude para o que tem a inscrição “Vitória”;
- Aqui, determinamos que se o cenário mudar para o que tem a inscrição “Vitória”, o jogo pare.

Chegando nesse ponto seu jogo já está pronto para ser testado e, se tudo estiver funcionando corretamente, nomeado e compartilhado na plataforma Scratch com os demais usuários!

Mais uma vez, fique à vontade para acessar o perfil do Guia Prático no Scratch e conferir como ficou o jogo por lá, ver o interior para verificar os códigos e detalhes.



<https://scratch.mit.edu/projects/312103513>

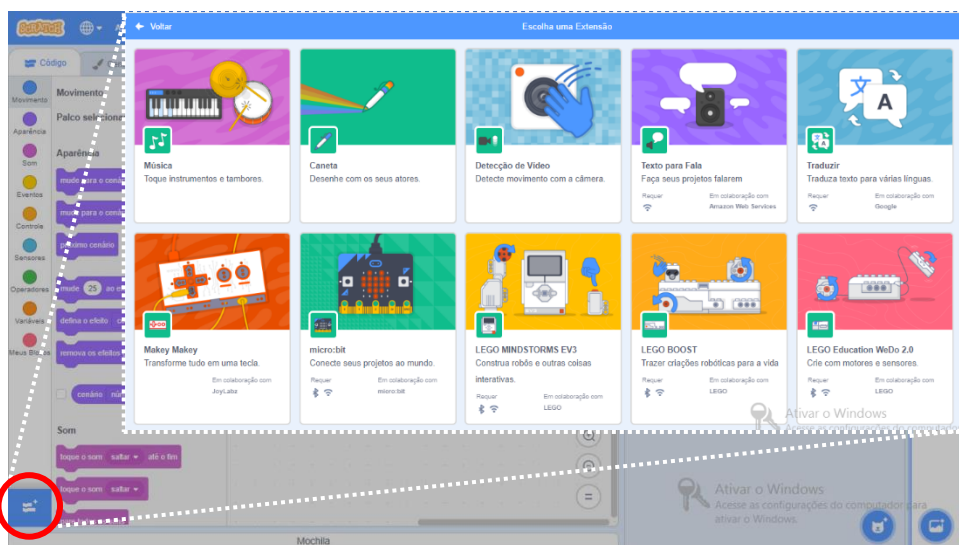
Leia o
QR Code para
visualizar no seu
aparelho celular
(vai precisar de
um teclado)



Atividade 3: Astronauta Artista

A terceira atividade apresenta uma possibilidade de utilizar os atores do Scratch para elaborarem desenhos e, se quisermos, arte. Nessa atividade, de acordo com nossos comandos, o astronauta elabora os desenhos de algumas figuras geométricas. Também é possível, com um pouco mais de prática e dedicação, elaborar desenhos artísticos, a critério de seu interesse.

Nessa atividade você irá um pouco mais além no conhecimento da plataforma Scratch. Terá uma primeira experiência com o uso de “extensões”, que são blocos extras que podem ser incorporados às suas criações. Existem diversas opções de extensões que já são disponibilizadas na plataforma, conforme a imagem abaixo. Mas qualquer pessoa com conhecimentos de programação e desenvolvimento pode criar extensões e submeter à plataforma.

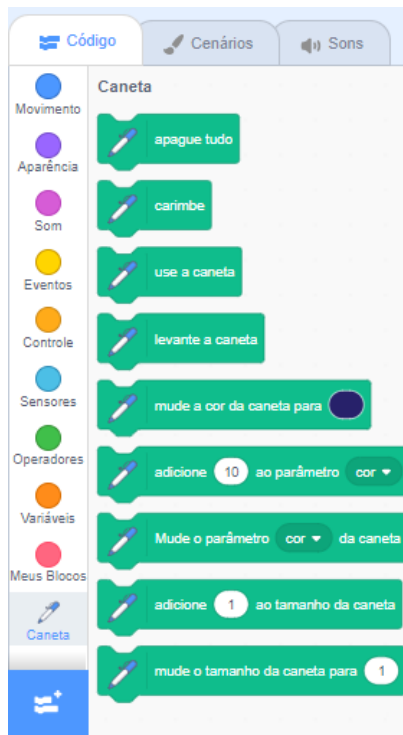


Para fazermos o ator dessa atividade desenhando, vamos utilizar a extensão “Caneta”. Basicamente o que essa extensão faz é deixar a marca do ator quando ele se movimenta pela tela, produzindo assim o desenho da trajetória do ator. Programando os movimentos do ator pela tela determinamos, então, qual tipo de apresentação gráfica será feita. Na atividade, nosso ator demonstrará o desenho de três figuras geométricas, a nossa escolha: círculo, quadrado ou triângulo.

Se estiver pronto para começar, abra a plataforma e inicie uma nova criação. O passo-a-passo a seguir vai te ajudar a experimentar esse novo conhecimento.

Passo-a-passo

1. Repita as etapas de 1 a 3 da Atividade 1;
2. Clique em “Selecionar um ator” e adicione o ator “Ripley”;
3. Clique em “Selecionar cenário”, procure pelo cenário “Stars” e adicione-o.
4. Ainda com o palco ativo, clique na aba “Cenários” e duplique o cenário *Stars*;
5. Clique no primeiro ícone do cenário *Stars* e adicione um título para a criação – no nosso exemplo, escolhemos o título “Astronauta Artista”;
6. Na área que mostra os blocos de código, na parte de baixo, clique em “Adicionar uma Extensão”. Na janela que vai abrir, selecione a extensão “Caneta”. Aparecerá na área de blocos um novo ícone, “Caneta”. Clique sobre ele para observar os novos blocos adicionados ao seu arsenal criativo (imagem abaixo);



Agora você tem todo o material de que precisa para desenvolver sua criação. Vai utilizar os blocos que já abordamos até aqui e mais os blocos novos da “Caneta”. Dessa vez, vamos mostrar as imagens com os conjuntos de códigos já construídos que vamos utilizar para o astronauta e para o cenário. Observe as próximas etapas atentamente e realize a construção dos códigos.

7. Para o nosso ator nessa atividade (Ripley, o astronauta), vamos precisar de seis conjuntos de códigos. Observe as imagens a seguir e crie os blocos conforme mostramos. Fique atento à utilização dos novos blocos de “Caneta”.

The image displays two Scratch code snippets for the character Ripley. The left snippet is triggered by a key press and sets up a drawing routine. The right snippet is triggered by a key press and performs a sequence of actions including drawing a yellow circle and asking the user for a key press.

Snippet 1 (Left):

- quando o cenário mudar para: **Itans2**
- apague tudo
- vá para x: **0** y: **0**
- defina o tamanho como: **0** %
- mostre
- repita: **5** vezes
 - mude para a fantasia: **10** no tamanho
 - espere: **1** seg
- espere: **1** seg
- mude para a fantasia: **ripley-b**
- diga: **Olá!** por **3** segundos
- mude para a fantasia: **ripley-a**
- espere: **5** seg
- mude para a fantasia: **ripley-b**
- diga: **Posso te mostrar algumas figuras geométricas.** por **4** segundos
- mude para a fantasia: **ripley-a**
- espere: **5** seg
- mude para a fantasia: **ripley-b**
- diga: **Quer ver?** por **4** segundos
- mude para a fantasia: **ripley-a**
- espere: **5** seg
- mude para a fantasia: **ripley-b**
- diga: **Aperte a tecla: 1- Círculo, 2- Quadrado, 3- Triângulo, 4- Não?**

Snippet 2 (Right):

- quando a tecla: **1** for pressionada
- mude para a fantasia: **ripley-a**
- espere: **5** seg
- mude para a fantasia: **ripley-b**
- diga: **Ok! Vamos lá!** por **4** segundos
- mude para a fantasia: **ripley-a**
- espere: **5** seg
- mude para a fantasia: **ripley-e**
- apague tudo
- repita: **300** vezes
 - mude para a fantasia: **ripley-f**
 - mude a cor da caneta para: **Amarelo**
 - use a caneta
 - mova: **1** passos
 - gire: **1** graus
 - levante a caneta
- mude para a fantasia: **ripley-e**
- espere: **5** seg
- mude para a fantasia: **ripley-b**
- diga: **Qual você quer que eu desene?** por **4** segundos
- mude para a fantasia: **ripley-a**
- espere: **5** seg
- mude para a fantasia: **ripley-b**
- diga: **Aperte a tecla: 2- Quadrado, 3- Triângulo, 4- Nenhum?**

The image displays two Scratch code snippets side-by-side, both triggered by a key press. The left snippet is triggered by key '2' and the right by key '3'. Both snippets follow a similar sequence of actions:

- quando a tecla [2] for pressionada (left) / [3] for pressionada (right)
- apague tudo
- mude para a fantasia riple-y-a
- espere .5 seg
- mude para a fantasia riple-y-b
- diga [Beleza! por 4 segundos] (left) / [Boa escolha! por 4 segundos] (right)
- mude para a fantasia riple-y-a
- espere .5 seg
- mude para a fantasia riple-y-e
- repita 4 vezes (left) / 3 vezes (right)
 - mude para a fantasia riple-y-f
 - mude a cor da caneta para []
 - use a caneta
 - repita 100 vezes
 - mova 1 passos
 - gire 90 graus (left) / 240 graus (right)
 - levante a caneta
- mude para a fantasia riple-y-e
- espere .5 seg
- mude para a fantasia riple-y-b
- diga [Quer que eu desenhe mais algum?] por 4 segundos (left) / [Quer ver outro desenho?] por 4 segundos (right)
- mude para a fantasia riple-y-a
- espere .5 seg
- mude para a fantasia riple-y-b
- diga [Aperte a tecla: 1- Círculo, 3- Triângulo, 4- Não?] (left) / [Aperte a tecla: 1- Círculo, 2- Quadrado, 4- Não?] (right)



Obviamente você pode personalizar à vontade os textos das falas do astronauta e as posições de movimento pela tela. Sempre é bom experimentar fazer alterações e observar os resultados. Encorajamos que você faça experiências para verificar os erros que podem ocorrer, além dos aperfeiçoamentos. Com os erros também aprendemos, não esqueça.

Analise bem os blocos para quando as teclas **1**, **2** e **3** são pressionadas. Neles você pode verificar os comandos que levam Ripley a desenhar, respectivamente, o círculo, o quadrado e o triângulo na tela.

8. Nessa atividade, a programação do cenário é bem resumida. Ao iniciar-se o objeto, determina mostrar o título e mudanças no efeito cor. Em seguida, retira o título e libera o cenário para a atuação de Ripley, quando muda para a segunda fantasia.



Atenção para as dicas!

- I. Ao fazer a montagem dos códigos de Ripley, fique atento àquelas partes que se repetem e, para agilizar o processo, copie e cole, ajustando valores e escritas onde for necessário. Vai economizar um tempo precioso.

- II. A ordem na qual os grupos de códigos ficam dispostos na área de programação não interfere na execução dos comandos. Até mesmo se ficarem sobrepostos uns aos outros, vai funcionar normalmente. Mas é claro que é sempre bom ordenar para que fiquem visíveis, para o caso de precisarmos fazer ajustes.

Se durante a criação e programação dessa atividade você tiver alguma dúvida, lembre-se de que sempre é possível acessar o perfil do Guia Prático no Scratch e testar nosso objeto concluído à vontade. Também é possível copiar o objeto para seu perfil e remixar à vontade. Abaixo vão os links direcionados ao objeto.



<https://scratch.mit.edu/projects/313072665>

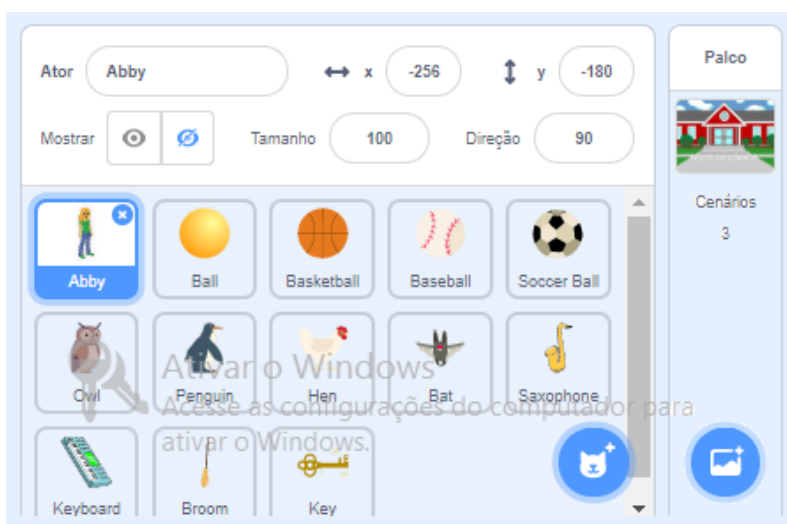


Leia o QR Code para visualizar no seu aparelho celular (vai precisar de um teclado conectado ao celular para funcionar corretamente).

Atividade 4: Desafio de Lógica

Na quarta atividade, além dos sete grupos de blocos com os quais já está familiarizado, vamos mostrar um exemplo de usos dos blocos de variáveis. Vamos criar duas variáveis que vão ser importantes para o funcionamento dessa atividade. Consiste num quis de lógica com três desafios para você descobrir.

O detalhe nessa atividade é que há programação para o cenário e para cada ator que aparece no palco, que são 13 – a professora e todos 12 itens do teste precisam de programação para executarem suas ações (imagem abaixo). Leia o passo-a-passo e siga atentamente as orientações para desenvolver a criação.



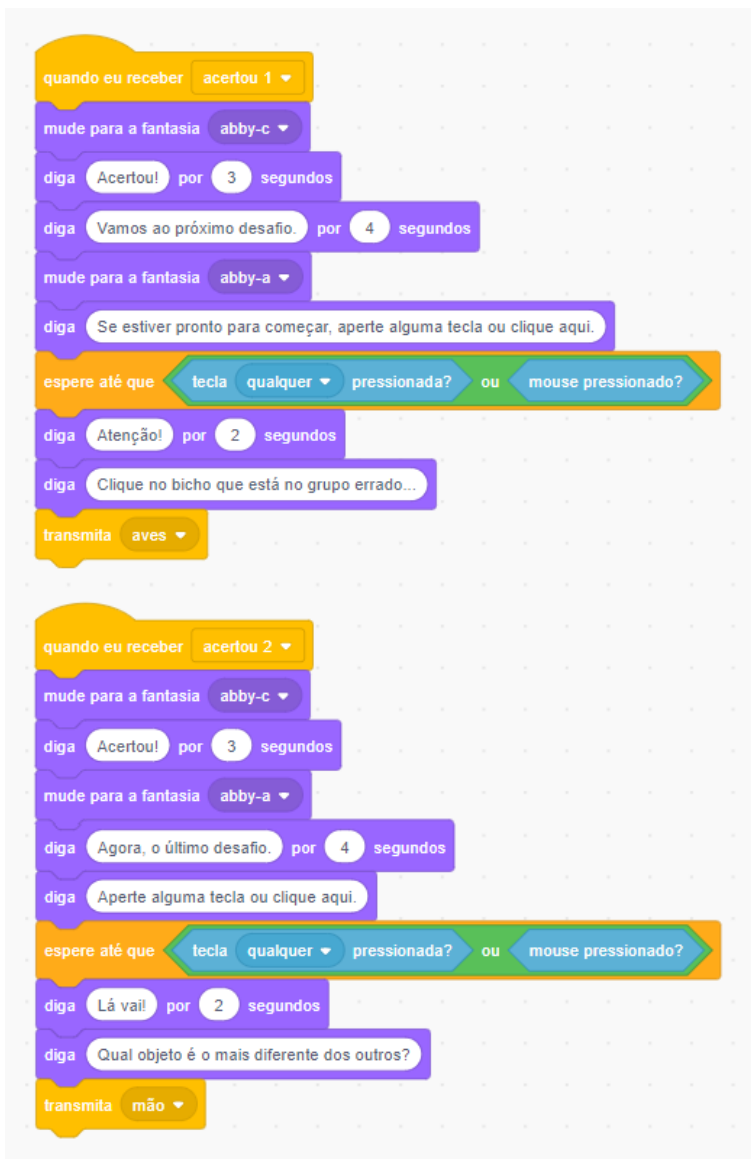
Passo-a-passo

1. Repita as etapas de 1 a 3 da Atividade 1;
2. Clique em “Selecionar um ator” e adicione “Abby”;
3. Clique em “Selecionar cenário”, procure pelo cenário “School” e adicione-o. Faça o mesmo com o cenário “Room 1”;
4. Clique no primeiro ícone do cenário *School* e adicione um título para a criação – escolhemos o título “Teste de Lógica”;
5. Na área que mostra os blocos de código, na parte de baixo, clique no ícone “Variáveis”. Aparecerão na área de blocos os blocos de variáveis. Clique no botão “Criar uma Variável” e crie uma variável chamada “Nome”. Repita a operação e crie uma variável chamada “Pontos”. Repare que surgirão dois botões logo abaixo, nomeados conforme você determinou;

6. Para programar a professora, vamos precisar determinar sua interação com o cenário e com os demais atores. Nomeamos a professora como Christine, em homenagem à orientadora desse trabalho, mas claro que você pode nomear como preferir. Para tanto você precisará criar seis grupos de códigos. Observe-os a seguir:

The image shows a Scratch script for a character named Christine. The script is composed of several blocks, each with an explanatory text box pointing to it:

- Block 1:** "quando for clicado" (when clicked) event block, followed by "esconda" (hide) block. *No início da ação, a professora fica oculta, para não aparecer na tela inicial.*
- Block 2:** "quando o cenário mudar para Room 1" (when scene changes to Room 1) event block, followed by "mude para a fantasia abby-a" (change costume to abby-a), "vá para x: -260 y: 0" (go to x: -260 y: 0), "mostre" (show), "deslize por 1.5 segs. até x: -160 y: -60" (slide for 1.5 seconds to x: -160 y: -60), and a series of "diga" (say) blocks with durations: "Olá! por 3 segundos", "Eu sou a ProF Christine, tudo bem? por 4 segundos", "Quero te fazer três desafios. por 4 segundos", "Cada vez que acertar, ganha dois pontos. por 4 segundos", "Cada vez que errar, perde um ponto. por 4 segundos", and "Quando estiver pronto para começar, aperte alguma tecla ou clique aqui." *A professora só surge quando o cenário muda para a sala.*
- Block 3:** "espere até que" (wait until) block with "tecla qualquer pressionada?" (any key pressed?) and "ou mouse pressionado?" (or mouse clicked?) options, followed by "diga Vamos lá! por 2 segundos" (say Vamos lá! for 2 seconds) and "diga Clique no item que não pertence a esse grupo..." (say Click on the item that does not belong to this group...). *Aqui o objetivo é a ativação se dar pelo toque, permitindo uso pelo computador ou pelo celular.*
- Block 4:** "transmita bolas" (send balloons) block. *A transmissão e o recebimento de mensagens marcam a comunicação com os demais atores.*
- Block 5:** "quando eu receber erro 1" (when I receive error 1) event block, followed by "mude para a fantasia abby-b" (change costume to abby-b) and "diga Não. Tente novamente..." (say No. Try again...). *A transmissão e o recebimento de mensagens marcam a comunicação com os demais atores.*



The image shows a Scratch script for a quiz game. The script starts with a 'quando eu receber' block set to 'acertou 3'. It then changes the costume to 'abby-c', says 'Acertou! Parabéns!' for 3 seconds, and changes the costume to 'abby-a'. This is followed by a 'se' block with the condition 'Pontos = 6', which says 'Você foi excelente! Parabéns!' for 4 seconds. Another 'se' block with the condition 'Pontos = 5' says 'Você foi bem! Parabéns!' for 4 seconds. A third 'se' block with the condition 'Pontos < 5 e [] > 2' says 'Você quase chegou lá! Tente de novo.' for 4 seconds. A fourth 'se' block with the condition 'Pontos < 3' says 'Você errou muito, precisa estudar mais!' for 4 seconds. The script then says 'Agora vou embora!' for 4 seconds, 'Até a próxima!' for 4 seconds, moves 1 second to x: -260, y: -180, hides, and finally stops all scripts.

Esses blocos determinam interação com o usuário. Dá feedback de acordo com o desempenho ao longo da atividade. Considera quatro níveis de desempenho: acertar tudo, errar apenas uma, acertar 3 ou 4 e acertar duas ou menos.

Na programação da professora, a interação com o cenário está determinada e os demais atores estão mostrados nos blocos que destacamos com comentários. Nossa opção por desenvolver interação entre a aplicação e o usuário ao longo dos exemplos até aqui visa tornar a utilização mais interessante e atrativa.

Ao planejar esse tipo de interatividade é importante levar em consideração as possibilidades de ações, para evitar erros na execução da programação. E não duvide: os erros ocorrerão. Cabe a você localizar as falhas e repensar o código para corrigi-las. Esse processo é normal e faz parte do aprendizado. Não se frustre quando acontecer!

7. Para programar os demais atores, você perceberá que há dois tipos de atores. Os três que são as respostas corretas dos três desafios e os nove que são respostas erradas nos desafios. A programação dos três corretos é igual, só mudando a mensagem que recebem da professora, a posição “x” e “y” na tela e a mensagem que enviam ao serem clicados. Basta criar o código para um e copiar para os outros dois, acertando os dados dos blocos onde se fizer necessário. O mesmo ocorre para os outros nove atores. A seguir mostramos a programação de cada tipo de ator.



The image shows a Scratch script for a character with the following blocks:

- quando for clicado** (when clicked)
 - esconda** (hide)
- quando eu receber** (when I receive)
 - defina o tamanho como** (define size as) [] %
 - vá para x:** [] **y:** [] (go to x: y:)
 - mostre** (show)
 - deslize por** (slide in) 3 segs. até x: [] y: [] (slide in for 3 secs. to x: y:)
- quando este ator for clicado** (when this actor clicked)
 - transmita** (send) [] (send [])
 - adicione** (add) 2 a Pontos (add 2 to Pontos)
 - espere** (wait) 3 seg (wait 3 secs)
 - esconda** (hide)

Callouts provide specific values for different characters:

- Todos ficam ocultos ao início, quando o cenário é o da escola.** (All are hidden at the start, when the scene is the school.)
- Bola = bolas**
Morcego = aves
Saxofone = mão
- Bola = 100%**
Morcego = 50%
Saxofone = 80%
- Para todos = -260, -134**
- Bola = 27, -134**
Morcego = 182, -134
Saxofone = 182, -134
- Bola = acertou 1**
Morcego = acertou 2
Saxofone = acertou 3



“quando eu receber”: *bolas*

“defina tamanho como...%”: *100, 100, 50*

“vá para x:, y:”: *-260, -134*

“deslize por 3 segs. até x:, y:”:

x: -46, 106, 182, y: -134

“transmita”: *errou 1*

“quando eu receber”: *acertou 1*



“quando eu receber”: *aves*

“defina tamanho como...%”: *60, 70, 50*

“vá para x:, y:”: *-260, -134*

“deslize por 3 segs. até x:, y:”:

x: -70, 6, 84, y: -134

“transmita”: *errou 1*

“quando eu receber”: *acertou 2*



“quando eu receber”: *mão*

“defina tamanho como...%”: *80, 90, 70*

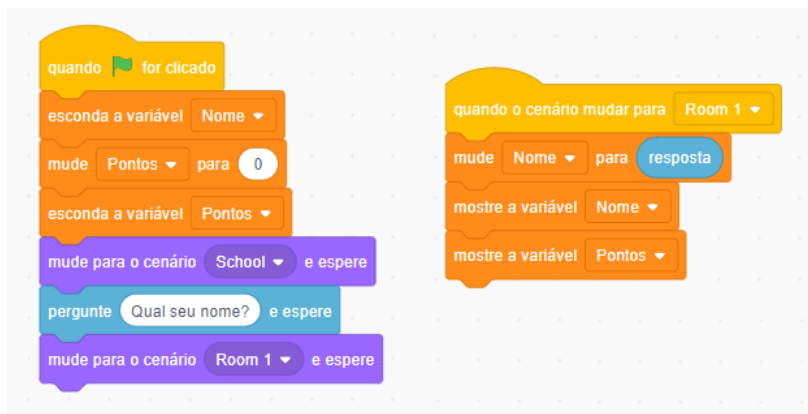
“vá para x:, y:”: *-260, -134 (teclado e chave); -260, -69 (vassoura)*

“deslize por 3 segs. até...”: *x: 20, 100, y: -134 (teclado e chave); x: -60, y: -69 (vassoura)*

“quando eu receber”: *acertou 3*

“transmita”: *errou 1*

8. Por fim, vamos programar os cenários. Para eles, o código é bem mais simples. Consiste em apenas dois grupos. Um para quando a aplicação for iniciada e outro para quando o cenário mudar. Confira a seguir.



Aqui utilizamos as variáveis que criamos no início do desenvolvimento da atividade. Estamos utilizando a variável “Nome” para personalizar o cenário de atuação dos atores com o nome do usuário, que é perguntado quando a aplicação é iniciada. A variável “Pontos” guarda a interação dos atores entre si, na forma de acréscimo de 2 em caso de acertos e -1 em caso de erros.

Atenção para as dicas!

- I. Determinamos que as variáveis Nome e Pontos fossem escondidas no início da aplicação por opção nossa. Se for sua vontade que elas fiquem à mostra desde o início da ação, não há qualquer problema.
- II. Quando a tela inicial é ativada, é importante não esquecer de zerar a variável Pontos, senão, a pontuação do último usuário fica registrada até que o próximo usuário conclua o desafio.
- III. Se a opção for também por esconder nome e pontos na tela inicial, é importante não esquecer de colocar os comandos para exibi-las no cenário da ação dos atores.



<https://scratch.mit.edu/projects/312952539>

Leia o QR
Code para visualizar
no seu aparelho
celular.



Quando  virar página

Iniciar

Parte III

III. Scratch na Escola – Exemplo de Aplicação

Jogo: Passeio pela Amazônia

Apresentando

Nessa terceira parte do guia, vamos apresentar um exemplo de um objeto virtual criado com foco no processo de aprendizagem escolar. Trata-se de um *tour* pela Amazônia, onde se apresentam aspectos gerais daquela região do país. Este objeto constitui-se numa dentre muitas possibilidades de aplicação direta do Scratch no trabalho pedagógico. No caso, foi um produto criado por um professor para utilização pelos alunos.

Uma atividade desse tipo pode ser pensada para ser aplicada antes, durante ou depois da abordagem dos conteúdos com os alunos. Ou seja, pode ter um caráter diagnóstico, de reforço/aprofundamento ou de avaliação. Além disso, é importante ter em mente que os alunos em geral apresentaram interesse em como criar algo do tipo também e que essa possibilidade também pode ser explorada pelo professor no sentido de contribuir para a aprendizagem à luz da teoria construcionista de Papert (1993).

Tanto quanto aprender por meio de atividades, desafios e exposições, o aluno pode desenvolver sua aprendizagem escolar a partir do processo de – termo tão na moda hoje em dia: “colocar a mão na massa”. Aqui, uma “massa” virtual, a bem da verdade. Mas ainda assim uma massa de conhecimentos que se cruzam e se complementam, com o foco no desenvolvimento da aplicação.

O que há de mais interessante na utilização do Scratch, do ponto de vista do ensino, não é o produto. Antes, ao longo de todo o processo há uma dinâmica de erros e acertos, somados a desafios e planejamento, que tem grande potencial para contribuir no sentido do desenvolvimento do raciocínio lógico, matemático, espacial, linguístico, etc. Nesse processo, nos deparamos também com as vantagens advindas de um trabalho a partir de projetos, conforme defende Resnick (1998), já que para o desenvolvimento das aplicações no Scratch o usuário é conduzido por um processo que minimamente desenvolve habilidades para o trabalho com o planejamento, prototipação, desenvolvimento e execução de um projeto.

Assim, a aprendizagem tem uma maior chance de ser tanto mais significativa quanto mais facilitada de ser assimilada pelo aluno, nos moldes do que defendem Ausubel, Novak e Hanesian (1983) a respeito das relações entre aprendizagem, método de ensino e estrutura cognitiva do aluno. Ou seja, trabalhar conhecimentos novos, mas a partir principalmente do que o aluno já domina, no nosso caso também com a introdução das tecnologias, na busca de uma facilitação da assimilação da aprendizagem e ao mesmo tempo a evolução da estrutura cognitiva discente.

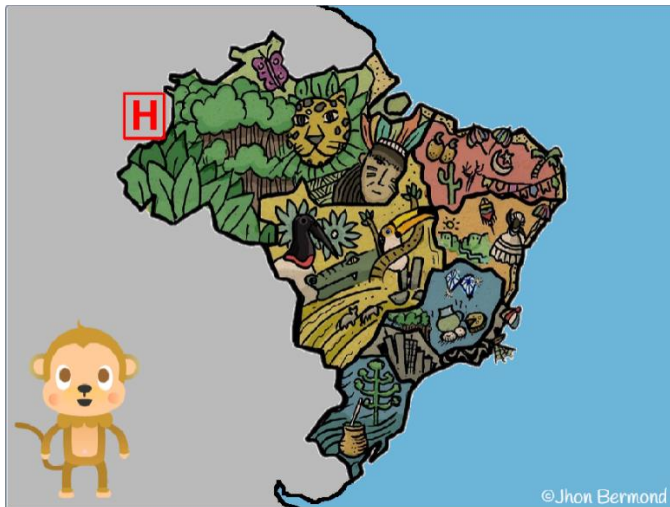
Ficha Técnica

A seguir, apresentamos a ficha técnica do objeto virtual e, após, passamos à sua apresentação pormenorizada.

1. <u>Nome:</u> Passeio pela Amazônia;
2. <u>Objeto:</u> Apresentação interativa + Quiz;
3. <u>Objetivo Geral:</u> Ler os resumos sobre as características gerais da Amazônia e responder as perguntas para testar os conhecimentos sobre a região;
4. <u>Objetivo Específico:</u> Refletir sobre a importância da região amazônica, sua biodiversidade e populações tradicionais;
5. <u>Conteúdos:</u> Questões socioambientais amazônicas;
6. <u>Disciplinas Relacionadas:</u> Tema transversal a diversas áreas do conhecimento;
7. <u>Recursos Materiais:</u> Computador ou smartphone, conexão com a internet;
8. <u>Recursos Humanos:</u> Professor, aluno;
9. <u>Metodologia:</u> O objetivo é percorrer toda a trilha sobre conhecimentos acerca da Amazônia, compreender os aspectos gerais que caracterizam a região, sua importância para o país e os desafios que enfrenta. Pode ser utilizado como uma introdução à abordagem da temática nas aulas, como um complemento ao estudo do tema ao longo das aulas ou como um indicador complementar de avaliação sobre o tema, após ter sido abordado nas aulas;
10. <u>Habilidades Desenvolvidas:</u> <ul style="list-style-type: none"> ▪ Leitura e compreensão textual; ▪ Desenvolvimento da memória; ▪ Gosto pelo desafio; ▪ Reflexão; ▪ Sensibilidade ambiental; ▪ Alteridade;
11. <u>Faixa Etária:</u> a partir de 10 anos;
12. <u>Avaliação:</u> <ul style="list-style-type: none"> ▪ Avaliar o interesse do aluno em participar do jogo; ▪ Concentração; ▪ Competência para reflexão sobre problemas a partir de situações lúdicas de aprendizagem.

Estrutura**Cenário 1**

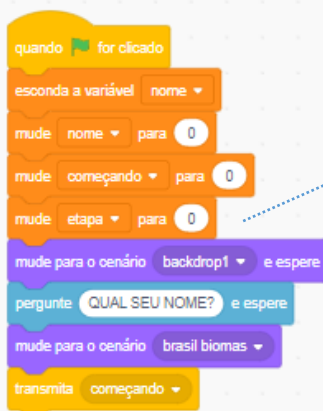
Título da aplicação e introdução do nome do usuário.

Cenário 2

Introdução da aplicação e apresentação dos comandos.

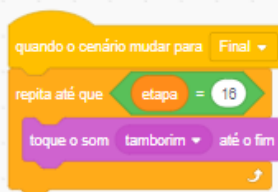
Códigos do Palco

O palco dessa aplicação tem quatro cenários, conforme vimos anteriormente. Para controlar as ações desses cenários e de sua interação com os atores e demais elementos (mensagens, variáveis e sons), há quatro conjuntos de blocos de códigos. Nem sempre as ações da aplicação estão relacionadas apenas aos comandos dos atores ou relacionado aos mesmos. É possível criar variáveis e mensagens ligadas ao palco, para que estejam disponíveis para todos os elementos que interagirem com o mesmo. Confira a seguir.



Comandamos aqui:

1. Ir para o cenário 1 quando a bandeira verde for clicada;
2. Zerar todas as variáveis utilizadas;
3. Perguntar o nome do usuário (que ficará armazenado na variável "nome");
4. Transmitir a mensagem "começando" (será recebida pelo ator Cauim, que desencadeará ações a partir disso).



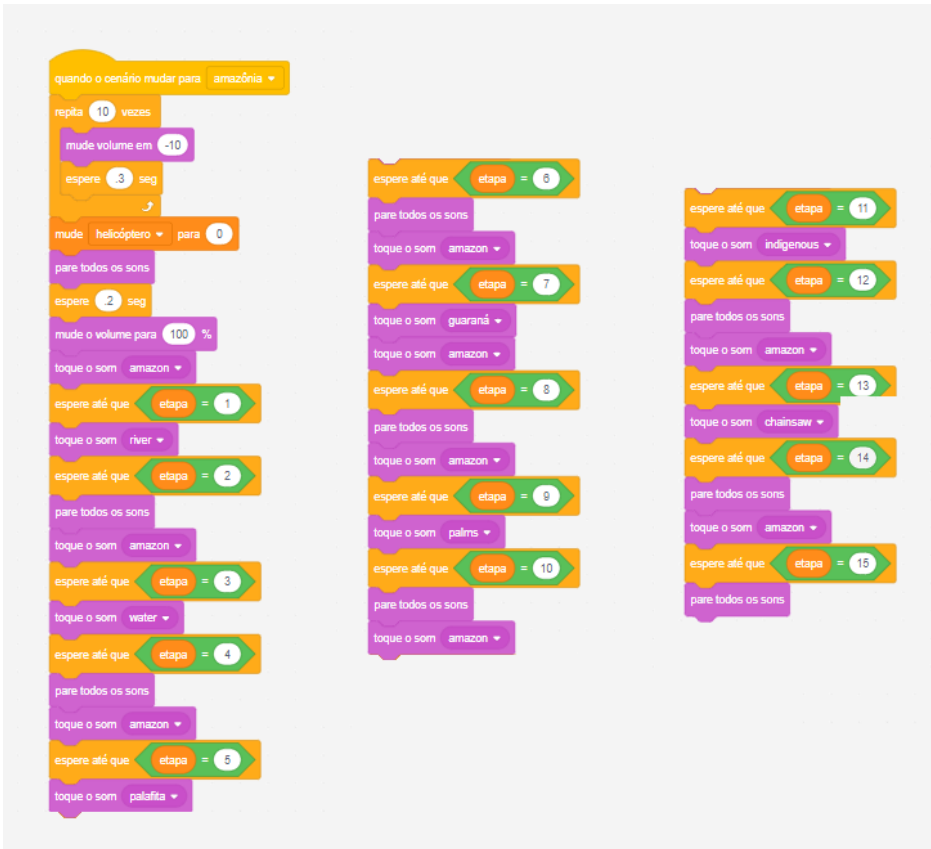
Com a mudança para o cenário 4:

1. Será emitido o som de tamborim.



Com a mudança para o cenário 2:

1. O nome do usuário fica registrado na tela até o final;
2. Com a variável "helicóptero" sendo mudada para 1, inicia-se a reprodução do som de helicóptero, até que a mesma variável receba comando para mudar novamente para 0.



Com a mudança para o cenário 3:

1. Percebemos o maior conjunto de blocos de código, que precisou ser dividido em três partes para se adequar ao espaço dessa página;

2. Basicamente, o que todo esse conjunto de blocos faz é controlar os sons que são acionados ao longo da mudança de atuação dos atores ao longo da apresentação, para ambientar o cenário:

- Amazon: ambiente de floresta
- River: águas correntes de rio
- Water: animal se movendo na água
- Palafita: águas nas margens do rio
- Guaraná: galhos de árvore sendo mexidos
- Palms: vento nas folhas de palmeiras
- Indigenous: fala ritual de um indígena
- Chainsaw: serra elétrica sendo ligada

3. A referência para o início ou fim da execução dos sons é o valor da variável “etapa”, que se relaciona com a ação do ator Cauim, como veremos mais adiante.

Códigos dos Atores

Os diversos atores utilizados na produção da aplicação têm seus próprios códigos e às vezes relacionam com os códigos dos demais atores ou do palco, por meio dos blocos de variáveis, de controle ou pelas mensagens enviadas. Foram utilizados nessa aplicação 11 atores. A seguir apresentamos cada um e a estrutura de seus códigos de comando.

Helicóptero

The image shows the Scratch code for a helicopter actor, organized into three columns. The first column contains initialization and scene change logic. The second and third columns contain movement logic for the four arrow keys (down, right, up, left).

- Column 1 (Initialization and Scene Change):**
 - quando for clicado → esconda
 - quando o cenário mudar para amazônia → transmite viajando → esconda
 - quando o cenário mudar para brasil biomas → espere até que helicóptero = 1 → vá para x: -187 y: -124 → mostre → repita até que helicóptero = 0 → próxima fantasia → espere .05 seg
- Column 2 (Down Arrow Movement):**
 - quando a tecla seta para baixo for pressionada → repita até que não tecla seta para baixo pressionada? → adicione -4 a y
- Column 3 (Right Arrow Movement):**
 - quando a tecla seta para direita for pressionada → repita até que não tecla seta para direita pressionada? → adicione 4 a x
- Column 4 (Up Arrow Movement):**
 - quando a tecla seta para cima for pressionada → repita até que não tecla seta para cima pressionada? → adicione 4 a y
- Column 5 (Left Arrow Movement):**
 - quando a tecla seta para esquerda for pressionada → repita até que não tecla seta para esquerda pressionada? → adicione -4 a x

Entra em cena com a mudança para o cenário 2:

- 1.O terceiro conjunto de blocos da esquerda determina a mudança de fantasia do ator, sempre que a variável helicóptero for considerada igual a 1. São três fantasias, cada uma com uma posição diferente das hélices. Com a troca de fantasia, ocorre o efeito de giro das hélices;*
- 2.Os quatro conjuntos de blocos da direita determinam a movimentação do helicóptero de acordo com as setas do teclado.*

Heliporto



quando for clicado

esconda

quando o cenário mudar para brasil biomas

mostre

quando o cenário mudar para brasil biomas

sempre

se tocando em Heliporter ? então

mude para o cenário amazônia

quando o cenário mudar para amazônia

esconda

Aparece apenas no cenário 2:

1.A ação executada por esse ator está vinculada unicamente ao fato de estar em contato ou não com o helicóptero. Havendo contato, ele direciona a mudança para o cenário 3, o que acaba afetando todos os demais atores.

Aparece apenas no cenário 3:

1.O conjunto de blocos da direita determina que o ator só apareça na tela quando receber a mensagem “Ok”, aumentando e diminuindo um pouco o tamanho enquanto não é clicado, conferindo movimento ao mesmo – efeito estético apenas;

2.O conjunto de blocos maior da esquerda determina que ao ser clicado o botão mude de aparência, para causar o efeito de ser apertado e, em seguida, acrescenta 1 à variável etapa e muda para o

Botão “OK”



quando for clicado

esconda

quando o cenário mudar para amazônia

mude amazônia para 1

quando este ator for clicado

mude para a fantasia button3-a

espere até que não mouse pressionado?

mude para a fantasia button3-b

adicione 1 a etapa

espere 1 seg

esconda

mude para o cenário Final

quando eu receber Ok

vá para x: 208 y: -160

mude para a fantasia button3-b

defina o tamanho como 50 %

mostre

repete até que mouse pressionado?

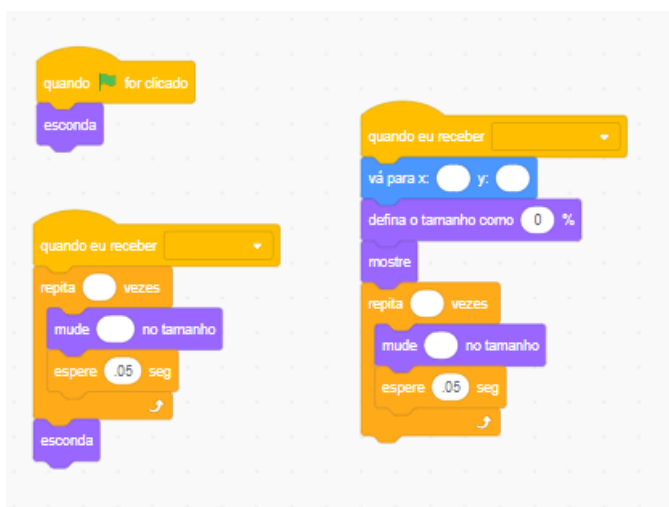
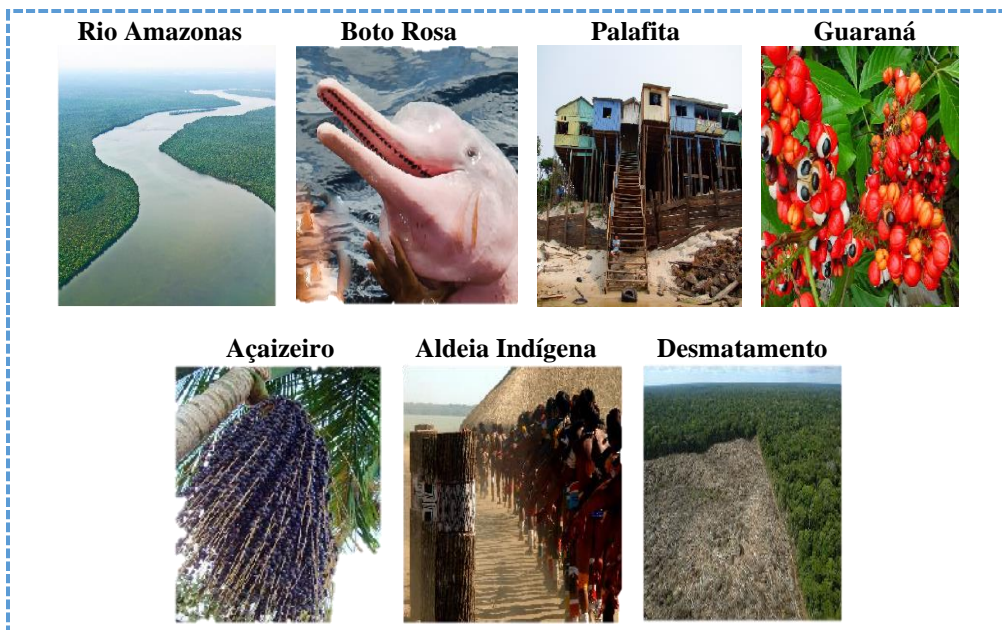
defina o tamanho como 55 %

espere 1 seg

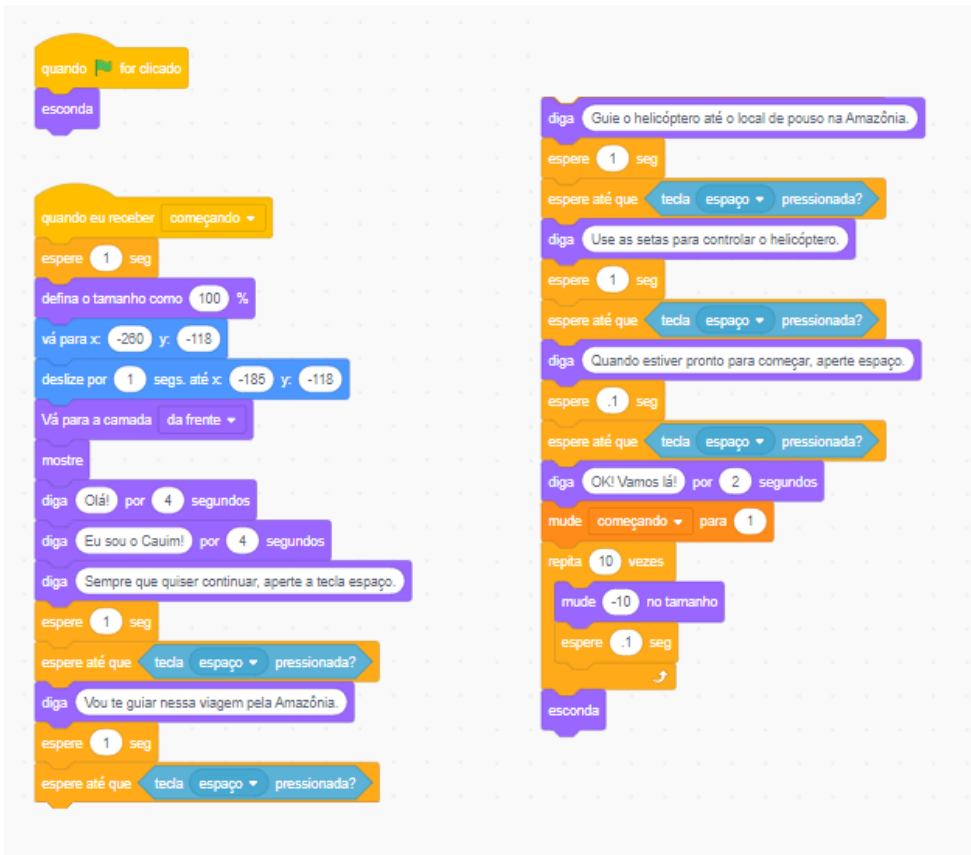
defina o tamanho como 50 %

espere 1 seg

Para todos os atores a seguir utilizamos os mesmos conjuntos de blocos de códigos. A diferença se dá unicamente pelo seu posicionamento na tela, pelas mensagens que os ativam e pelos valores das porcentagens de aumento e de diminuição quando, respectivamente, de seu aparecimento e desaparecimento. Esses valores são uma opção de quem desenvolve a atividade. Para verificar os valores que utilizamos, você pode conferir a aplicação na plataforma Scratch, conforme o link que indicamos ao final dessa parte do guia.



Para o último ator que falta considerarmos, é importante ficar atento, pois possui os mais longos conjuntos de blocos e, só por isso, já é mais fácil cometer enganos e gerar *bugs* nos códigos. São quatro grupos de códigos, que mostraremos e comentaremos parte a parte a seguir – fique atento quando o código aparece fragmentado aqui, por uma questão de espaço na página.



Acima, podemos perceber o menor comando, indicando que o ator deve permanecer oculto quando da inicialização da aplicação. Apenas quando recebe a mensagem “começando” (ligada diretamente ao palco) é que o ator aparece, nesse caso, após a mudança do cenário 1 para o cenário 2. Repare que o conjunto de blocos da direita na imagem é continuação do conjunto maior da esquerda.

Optamos por não colocar o texto de Cauim mudando automaticamente ao longo da programação, mas obedecendo ao comando do usuário, ao apertar a barra de espaço. Essa opção foi feita levando em consideração o tempo de leitura, que é diferente para cada indivíduo. Acharmos mais sensato, portanto, deixar o usuário escolher a hora de continuar.

```

quando o cenário mudar para amazonia
  espere 3 seg
  defina o tamanho como 100 %
  vá para x: -260 y: -118
  vá para a camada da frente
  mostre
  dizize por 2 segs. até x: -185 y: -118
  diga Chegamos à Amazônia.
  espere 1 seg
  espere até que tecla espaço pressionada?
  diga Essa é a maior floresta tropical do mundo.
  espere 1 seg
  espere até que tecla espaço pressionada?
  diga Ela recobre toda parte norte da América do Sul e a maior parte pertence ao Brasil.
  espere 1 seg
  espere até que tecla espaço pressionada?
  diga Aqui está o maior rio do mundo em volume de água. Você sabe qual é? Escolha entre as opções...
  espere 1 seg
  transmite amazonas vem
  adicione 1 a etapa
  espere até que tecla espaço pressionada?
  pergunte Aperte a tecla correspondente: 1- Nilo; 2- Araguaia; 3- Paraíba do Sul; 4- Amazonas. e espere
  
```

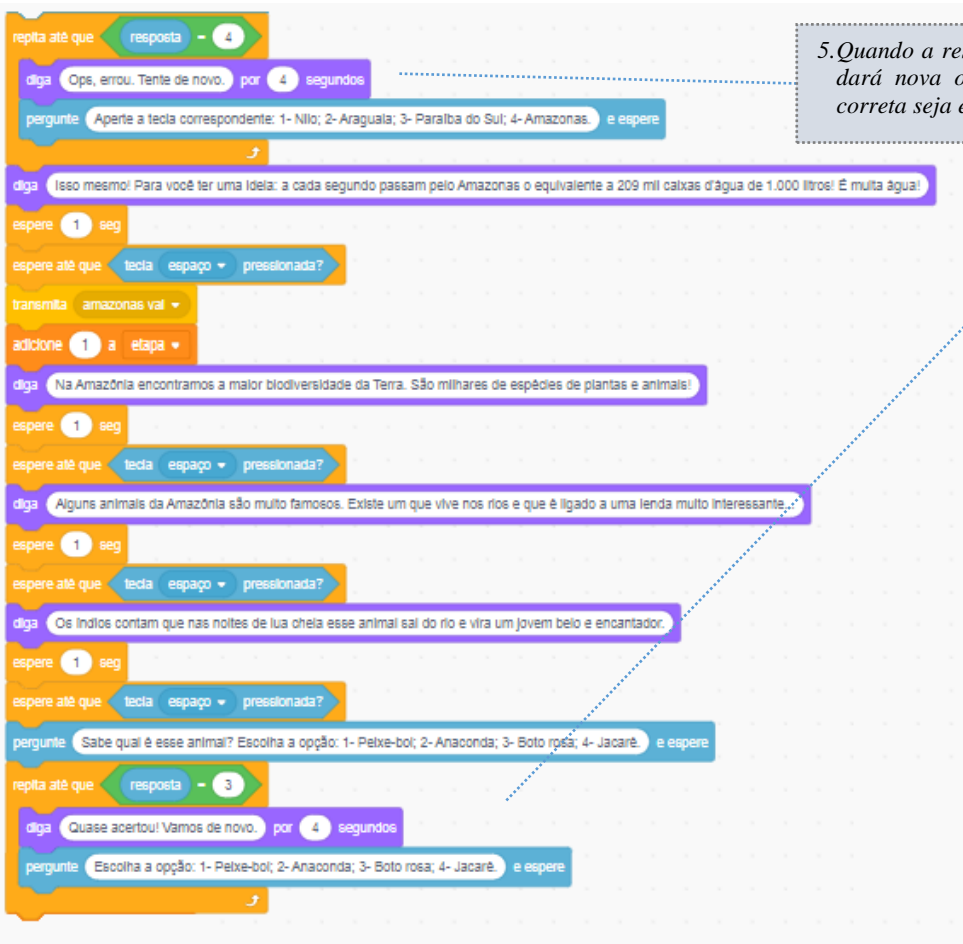
Quando o palco muda para o cenário 3:

4. Cauim reaparece e inicia a apresentação da região amazônica;

3. O avanço da apresentação permanece a partir do acionamento da barra de espaço pelo usuário, respeitando seu tempo de leitura;

2. O envio de mensagens ao longo do código de Cauim direciona o momento de entrada e saída de cada outro ator (pág. 67);

1. Também ao longo do código de Cauim vai ocorrendo interação com a variável “etapa”, que direciona a execução dos efeitos sonoros;



5. Quando a resposta dada for incorreta, Cauim indicará o erro e dará nova oportunidade para responder, até que a resposta correta seja escolhida;

The image shows a Scratch script with the following blocks:

- adicione 1 a etapa
- transmita boto emerge
- diga "Acertou! Segundo a lenda, o boto rosa encanta as moças solteiras e as leva para o fundo do rio. Depois as abandona. O problema é que algumas acabam grávidas. Que coisa, hein?!"
- espere 1 seg
- espere até que tecla espaço pressionada?
- transmita boto submerge
- adicione 1 a etapa
- diga "Na Amazônia também chove muito, praticamente todo dia. Algumas regiões passam boa parte do ano alagadas."
- espere 1 seg
- espere até que tecla espaço pressionada?
- diga "Por isso, os moradores de áreas próximas dos rios constroem casas bem típicas, elevadas, para evitar as inundações. Você sabe o nome desse tipo de construção?"
- espere 1 seg
- transmita vem palafita
- adicione 1 a etapa
- espere até que tecla espaço pressionada?
- pergunte "Então indique o nome desse tipo de casa: 1- Oca; 2- Palafita; 3- Taba; 4- Bangalô." e espere
- repita até que resposta = 2
- diga "Hmmm... Acho que não..." por 4 segundos
- pergunte "Escolha de novo: 1- Oca; 2- Palafita; 3- Taba; 4- Bangalô." e espere
- adicione 1 a etapa
- diga "Isso aí. As palafitas são típicas das áreas ribeirinhas da Amazônia. Um tipo de construção muito comum por lá."
- espere 1 seg
- espere até que tecla espaço pressionada?

The image shows a Scratch script designed as a quiz. It consists of several blocks connected in a sequence:

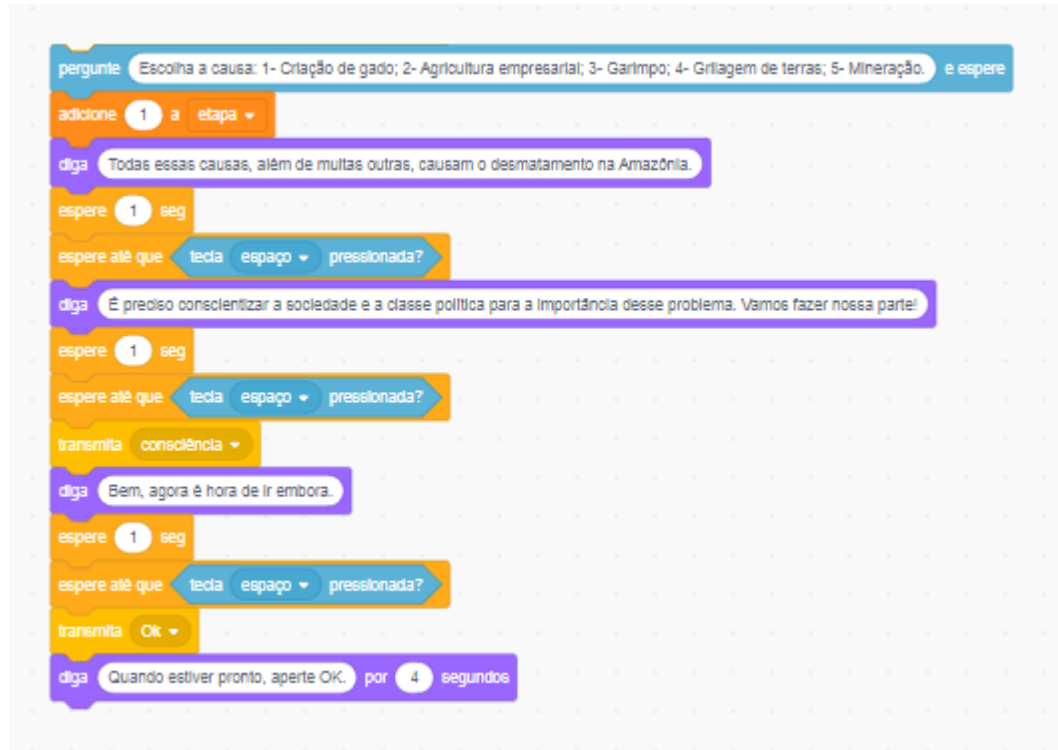
- transmita** val palavra
- adicione** 1 a etapa
- diga** Na nossa alimentação, há muitas influências da região amazônica. Uma fruta típica da Amazônia é usada para fazermos sucos e refrigerantes.
- espere** 1 seg
- espere até que** tecla espaço pressionada?
- pergunte** Será que você acertou que fruta é essa? Escolha: 1- Guaraná; 2- Laranja; 3- Morango; 4- Cacau. e espere
- repita até que** resposta = 1
- diga** Foi por pouco... por 4 segundos
- pergunte** Então, qual? 1- Guaraná; 2- Laranja; 3- Morango; 4- Cacau. e espere
- adicione** 1 a etapa
- transmita** guaraná no pé
- diga** Quem nunca tomou refrigerante de guaraná ou refresco com xarope de guaraná, não é mesmo?
- espere** 1 seg
- espere até que** tecla espaço pressionada?
- transmita** guaraná em pó
- adicione** 1 a etapa
- diga** Outro alimento originário da Amazônia faz sucesso por aqui. Na Amazônia costumam comê-lo com farinha e misturado à comida.
- espere** 1 seg
- espere até que** tecla espaço pressionada?
- pergunte** Duvido você acertar essa! Tente: 1- Cupuaçu; 2- Golaba; 3- Feijão; 4- Açaí. e espere
- repita até que** resposta = 4
- diga** Foi por pouco... por 4 segundos
- pergunte** Tente de novo: 1- Cupuaçu; 2- Golaba; 3- Feijão; 4- Açaí. e espere

The image shows a Scratch script designed as a quiz. It starts with a 'transmita' block for 'açai no pé', followed by an 'adicione' block to 'etapa'. A 'diga' block asks: 'Por aqui, a moda é misturar com frutas e granola. Como será que fica mais saboroso, hein?'. This is followed by a 'espere' block (1 seg) and an 'espere até que' block (tecla espaço pressionada?). Then, another 'transmita' block for 'açai no copo', an 'adicione' block to 'etapa', and a 'diga' block: 'Uma questão importante na Amazônia é a situação dos Índios. Devido ao contato com o "homem branco", cada vez mais problemas têm surgido.'. This is followed by another 'espere' block (1 seg), a 'transmita' block for 'Índios', an 'adicione' block to 'etapa', and an 'espere até que' block (tecla espaço pressionada?). A 'diga' block asks: 'Para tentar manter o isolamento e preservar o modo de vida dos Índios, uma importante medida costuma ser tomada pelo nosso governo.'. This is followed by another 'espere' block (1 seg) and an 'espere até que' block (tecla espaço pressionada?). A 'pergunte' block asks: 'Será que você acerta qual é essa medida? Escolha: 1- criar fazendas indígenas, 2- fornecer vestimentas e alimentos; 3- criar reservas indígenas; 4- transferir os Índios para outras regiões.' and includes an 'e espere' block. A 'repita até que' block (resposta = 3) contains a 'diga' block: 'Hmmm... Acho que não.' (por 4 segundos) and a 'pergunte' block: 'Escolha: 1- criar fazendas indígenas; 2- fornecer vestimentas e alimentos; 3- criar reservas indígenas; 4- transferir os Índios para outras regiões.' with an 'e espere' block. After the loop, a 'diga' block says: 'Isso mesmo. Com a criação de reservas indígenas espera-se garantir que os Índios possam continuar vivendo de forma tradicional, em meio à natureza. Assim, podem preservar seus hábitos e sua cultura.'. This is followed by an 'espere' block (1 seg), an 'espere até que' block (tecla espaço pressionada?), a 'transmita' block for 'reservas', and an 'adicione' block to 'etapa'.

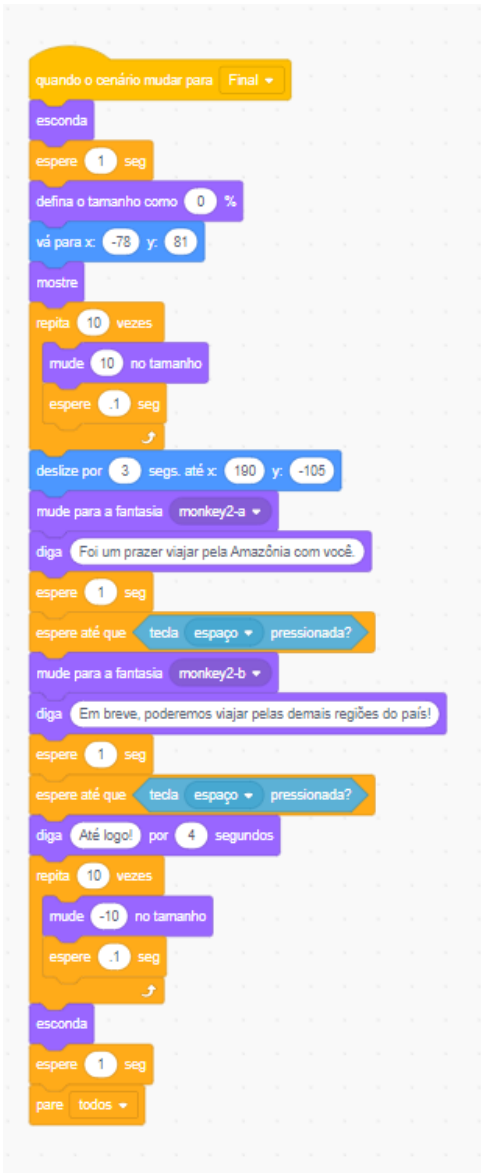
```
transmita açai no pé
adicione 1 a etapa
diga Por aqui, a moda é misturar com frutas e granola. Como será que fica mais saboroso, hein?
espere 1 seg
espere até que tecla espaço pressionada?
transmita açai no copo
adicione 1 a etapa
diga Uma questão importante na Amazônia é a situação dos Índios. Devido ao contato com o "homem branco", cada vez mais problemas têm surgido.
espere 1 seg
transmita Índios
adicione 1 a etapa
espere até que tecla espaço pressionada?
diga Para tentar manter o isolamento e preservar o modo de vida dos Índios, uma importante medida costuma ser tomada pelo nosso governo.
espere 1 seg
espere até que tecla espaço pressionada?
pergunte Será que você acerta qual é essa medida? Escolha: 1- criar fazendas indígenas, 2- fornecer vestimentas e alimentos; 3- criar reservas indígenas; 4- transferir os Índios para outras regiões. e espere
repita até que resposta = 3
diga Hmmm... Acho que não. por 4 segundos
pergunte Escolha: 1- criar fazendas indígenas; 2- fornecer vestimentas e alimentos; 3- criar reservas indígenas; 4- transferir os Índios para outras regiões. e espere
diga Isso mesmo. Com a criação de reservas indígenas espera-se garantir que os Índios possam continuar vivendo de forma tradicional, em meio à natureza. Assim, podem preservar seus hábitos e sua cultura.
espere 1 seg
espere até que tecla espaço pressionada?
transmita reservas
adicione 1 a etapa
```

The image shows a Scratch script designed for a presentation about Amazon deforestation. The script consists of several blocks: 'diga' (say) blocks with text, 'espere' (wait) blocks for 1 second, and 'espere até que' (wait until) blocks triggered by the 'tecla espaço' (space key) being pressed. The script is as follows:

```
1. diga: Vamos terminar falando de um problema muito sério que a Amazônia enfrenta cada dia mais: o desmatamento.
2. espere 1 seg
3. espere até que: tecla espaço pressionada?
4. diga: Por hora, a Amazônia perde em média 19 hectares de floresta.
5. espere 1 seg
6. transmite: desmatamento
7. adicione 1 a: etapa
8. espere até que: tecla espaço pressionada?
9. diga: Para você ter uma ideia, 1 hectare corresponde, aproximadamente, a 1 campo de futebol!
10. espere 1 seg
11. espere até que: tecla espaço pressionada?
12. diga: São 19 campos de futebol a menos da mata amazônica por hora. Pare e pense um pouco em como isso é grave.
13. espere 1 seg
14. espere até que: tecla espaço pressionada?
15. diga: Você imagina qual seja a causa desse problema?
16. espere 1 seg
17. espere até que: tecla espaço pressionada?
```



O uso de uma aplicação semelhante a essa poderia utilizar um código para atribuir uma nota de acordo com os erros ou acertos dos alunos, no caso da intenção de utilizar como forma de avaliação. Como nosso objetivo na elaboração não era esse, colocamos nas falas de Caum o *feedback* sobre erro ou acerto, porém, sem atribuição de notas.



Percorrida toda a trilha apresentada por Cauim a respeito da Amazônia, a aplicação direciona o usuário para a tela final, onde basicamente a ideia é arrematar a apresentação para finalizar. Chegando ao fim, o bloco “pare todos” determina o encerramento da execução da aplicação e, caso o usuário tenha intenção de percorrer novamente toda a trilha da apresentação, deverá clicar no símbolo da bandeira.

Como você sabe, pode conferir a íntegra da aplicação, testá-la, comentar e remixar acessando o perfil desse guia na plataforma Scratch:



<https://scratch.mit.edu/projects/313654792>

Se preferir, pode fazer a leitura do QR Code e acessar o produto pelo smartphone ou tablet.

Quando  virar página

Apresentar

Referências

Referências

- AUSUBEL, D. **Aquisição e Retenção de Conhecimentos: uma perspectiva cognitiva**. 1ª ed. Tradução de Lígia Teopisto. Lisboa: Plátano Edições Técnicas, 2003.
- AUSUBEL, D.; NOVAK, J. D.; HANESIAN, H. **Psicología Educativa: um ponto de vista cognoscitivo**. 2ª ed. Traducción al español de Mario Sandoval Pineda. México: Editorial Trillas, 1983.
- FREIRE, P. **Pedagogia da Autonomia: saberes necessários à prática educativa**. São
- LEITE, L. S. Mídia e a perspectiva da tecnologia educacional no processo pedagógico contemporâneo. In: FREIRE, Wendel (Org.). **Tecnologia e Educação: as mídias na prática docente**. Rio de Janeiro: Wak Ed., 2011. p. 61-67.
- MOREIRA, M. A. Aprendizagem significativa crítica. In: **ENCONTRO INTERNACIONAL SOBRE APRENDIZAGEM SIGNIFICATIVA, 3.**, 2000, Lisboa. Atas [...] Lisboa: Peniche, 2000. p. 33-35.
- PAPERT, S. **The Children's Machine: rethinking school in the age of the computer**. New York: BasicBooks, 1993.
- PELIZZARI, A. et al. Teoria da aprendizagem significativa segundo Ausubel. **Revista Psicologia Educação e Cultura**. Curitiba, V. 2, Nº 1, p. 37-42, jul. 2002.
- PISCHETOLA, M. **Inclusão Digital e Educação: a nova cultura da sala de aula**. Petrópolis: Vozes; Rio de Janeiro: Editora PUC-Rio, 2016.
- RESNICK, M. Technologies for lifelong kindergarten. In: **Educational Technology Research & Development**. Boston, Vol. 46, no. 4, p. 43-55, December, 1998. Disponível em: <http://web.media.mit.edu/~mres/papers.html>. Acesso em: 01/10/2017.