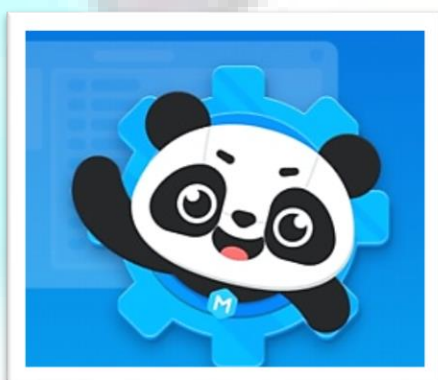


UMA PROPOSTA METODOLÓGICA PARA O DESENVOLVIMENTO DE OBJETOS DE APRENDIZAGEM DIGITAIS USANDO PROGRAMAÇÃO POR BLOCOS

E - BOOK



**PRINCIPAIS CONCEITOS TEÓRICOS E TÉCNICOS PARA A
ELABORAÇÃO DE OBJETOS DE APRENDIZAGEM DIGITAIS USANDO
A PLATAFORMA MBLOCK**

**PONTA GROSSA
2021**

**UM ESPECIAL AGRADECIMENTO A TODOS OS PROFESSORES QUE
ENSINAM, EDUCAM, FORMAM E ORIENTAM.**



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite que outros remixem, adaptem e criem a partir do trabalho para fins não comerciais, desde que atribuam o devido crédito e que licenciem as novas criações sob termos idênticos. Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

1. INTRODUÇÃO

Por mais que a tecnologia tenha avançado, verifica-se que pouco se sabe sobre as tecnologias quando se foca o seu uso no ensino e na aprendizagem (BRITO; PURIFICAÇÃO, 2015). Apesar das incertezas, fazer uso das tecnologias como meio educacional, é incorporar instrumentos que passam a ser uma excelente via pela qual diferentes conhecimentos possam chegar às salas de aula (ALMEIDA *et al.*, 2014).

Diante de um cenário pandêmico e de excepcionalidade, vivenciado em decorrência da mudança de atendimento nas escolas, o momento exige instrumentalizar os professores para que possam planejar suas aulas com o apoio de Tecnologias Educacionais Digitais (TED), no sentido de contemplar programas, dispositivos e aplicativos que estejam ao alcance de todos (SILVA; ANDRADE; BRINATTI, 2020). Partindo dessas premissas, a presente pesquisa visa tomar como referência o modelo Conhecimento Tecnológico e Pedagógico do Conteúdo (CTPC). Ainda segundo Silva; Andrade; Brinatti (2020, p. iv), algumas considerações acerca desse modelo podem “colaborar no processo de elaboração de propostas didáticas afim de que se possa abordar tópicos específicos abrangendo todas as inter-relações possíveis entre conhecimento disciplinar, tecnológico e também pedagógico”.

Buscando integrar as Tecnologias Educacionais Digitais (TED), as Tecnologias Digitais de Informação e Comunicação (TDIC), o modelo de Conhecimento Tecnológico e Pedagógico do Conteúdo (CTPC) e o desenvolvimento da competência Cultura Digital suas dimensões e subdimensões, encontrada na BNCC (2017), este projeto de pesquisa consiste em ofertar um curso online, a um público alvo de professores da Educação Básica frente a necessidade do ensino híbrido imposto pela pandemia da COVID-19.

A plataforma mBlock foi escolhida para ser utilizada nesta pesquisa por estar ancorada a Teoria Construcionista de Seymour Papert e sua Linguagem LOGO, bem como ser disponibilizada para *smartphones*, dispensando o uso de espaços específicos com conexão à internet. Acredita-se que por meio desta plataforma, os

professores possam elaborar projetos de OAs contemplando diferentes disciplinas e conteúdos escolares.

Além do mais, por meio do desenvolvimento de Objetos de Aprendizagem (OAs) digitais usando programação por blocos, os professores terão acesso a conceitos computacionais e contato com os quatro pilares do Pensamento Computacional (PC), alicerces estes para a resolução de problemas escolares ou da vida real, sendo que, o Pensamento Computacional (PC) consiste em reconhecer os aspectos computacionais no mundo ao nosso redor, a fim de que se possa abordá-los usando métodos e técnicas encontrados na Ciência Computacional. (BRACKMANN; 2017; WING, 2016).

Atualmente um novo e vasto campo de pesquisa surge a respeito da utilização de tecnologias no processo de ensino e aprendizagem. Esse novo campo, necessariamente interdisciplinar, tem de considerar a utilização cada vez maior das tecnologias em nossa sociedade e o redimensionamento do papel do professor nesse processo. Portanto, não se trata simplesmente por meio de um curso propor uma nova metodologia, trata-se de entender de como são criadas novas formas de ter acesso ao conhecimento e também de como produzi-lo (BRITO; PURIFICAÇÃO, 2015).

2. A BNCC E A COMPETÊNCIA CULTURA DIGITAL

Segundo a BNCC (2017), a quinta competência, denominada **Cultura Digital**, na qual se embasa este projeto, faz referência ao despertar nos alunos a utilização das **Tecnologias Educacionais Digitais (TED)**. Segundo a BNCC (2017), com a introdução da **Cultura Digital** nas escolas, pretende-se que os alunos compreendam, utilizem e criem tecnologias de forma crítica, significativa e ética para que possam comunicar-se, acessar e produzir informações e conhecimentos além de resolver problemas, exercer protagonismo e autoria. A **Cultura Digital** é composta pelas dimensões: **Computação e Programação**, **Pensamento Computacional** e **Cultura e Mundo Digital**. A figura 1 abaixo ilustra a Cultura Digital e suas dimensões:

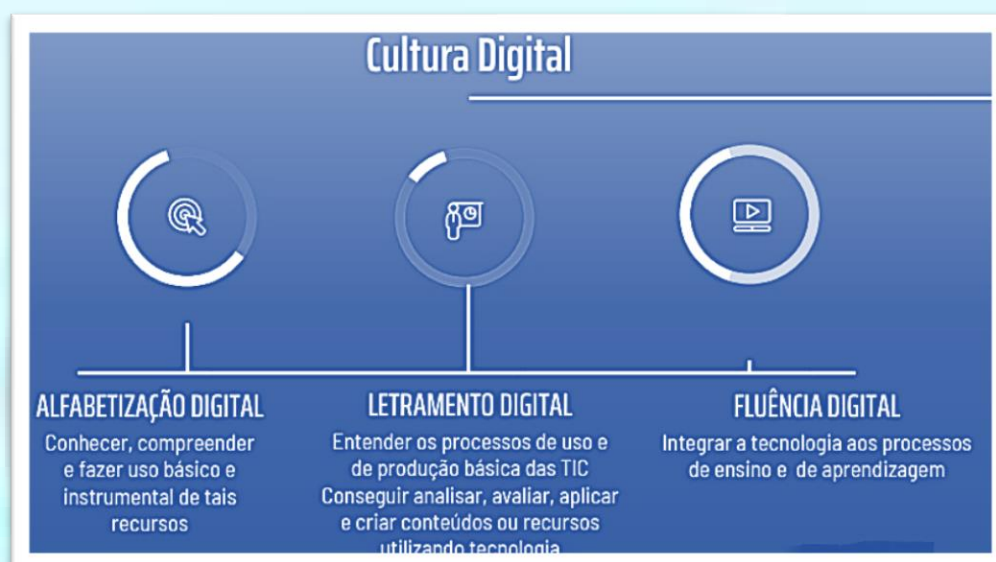
Figura 1 - A Cultura Digital e suas Dimensões



Fonte: Adaptado de (SATHLER, 2020)

Nesse contexto, com a introdução da **Cultura Digital** nas escolas, pretende-se desenvolver nos alunos a **Alfabetização Digital**, o **Letramento Digital** e uma maior proficiência digital, entendida na BNCC (2017) como **Fluência Digital**. A descrição destes três elementos está ilustrada na figura 2:

Figura 2 - Alfabetização, Letramento e Fluência Digital



Fonte: Adaptado de (SATHLER, 2020)

As dimensões da **Cultura Digital** possuem **Subdimensões** que detalham os objetivos propostos em cada dimensão. O quadro 1 resume a **Competência Cultura Digital**, suas **Dimensões** e o detalhamento de suas **Subdimensões**:

Quadro 1 - A Competência Cultura Digital e suas Dimensões e Subdimensões

COMPETÊNCIA	DIMENSÃO	SUBDIMENSÕES
CULTURA DIGITAL	COMPUTAÇÃO E PROGRAMAÇÃO	<ul style="list-style-type: none"> Utilização de Ferramentas digitais Produção Multimídia Linguagens de Programação
	PENSAMENTO COMPUTACIONAL	<ul style="list-style-type: none"> Domínio de Algoritmos Visualização e Análise de Dados.

	CULTURA E MUNDO DIGITAL	<ul style="list-style-type: none"> • Mundo Digital • Uso Ético.

Fonte: BNCC (2017)

Cada uma das **três dimensões** contidas na competência **Cultura Digital**, bem como o detalhamento das suas **subdimensões**, podem ser observadas no quadro 2:

Quadro 2 - As Dimensões e o detalhamento das Subdimensões

DIMENSÃO	SUBDIMENSÕES	DETALHAMENTO
COMPUTAÇÃO E PROGRAMAÇÃO	UTILIZAÇÃO DE FERRAMENTAS DIGITAIS	Utilização de ferramentas multimídia e periféricos para aprender e produzir.
	PRODUÇÃO MULTIMÍDIA	Utilização de recursos tecnológicos para desenhar, desenvolver, publicar, testar e apresentar produtos para demonstrar conhecimento e resolver problemas.
	LINGUAGENS DE PROGRAMAÇÃO	Utilização de Linguagens de Programação para solucionar problemas.
DIMENSÃO	SUBDIMENSÃO	DETALHAMENTO
PENSAMENTO COMPUTACIONAL	DOMÍNIO DE ALGORITMOS	Compreensão e escrita de algoritmos. Avaliação de vantagens e desvantagens de diferentes algoritmos. Utilização de classes métodos, funções e parâmetros para dividir e resolver problemas.
	VISUALIZAÇÃO E ANÁLISE DE DADOS	Utilização de diferentes representações e abordagens para visualizar e analisar dados.

DIMENSÃO	SUBDIMENSÃO	DETALHAMENTO
CULTURA E MUNDO DIGITAL	MUNDO DIGITAL	Compreensão do impacto das tecnologias na vida das pessoas e na sociedade, incluindo nas relações sociais, culturais e comerciais.
	USO ÉTICO	Utilização das tecnologias, mídias e dispositivos de comunicação modernos de forma ética, comparando comportamentos adequados e inadequados.

Fonte: BNCC (2017)

A intenção de compreender melhor a competência **Cultura Digital** constante na BNCC (2017), é a de incentivar os professores a inseri-la em sua metodologia de aula. As **Dimensões** e **Subdimensões** da **Competência Cultura Digital** e sua abrangência nas séries da Educação Básica, enfoque desta dissertação, estão descritas a partir do quadro 3, prosseguindo em sequência por ano escolar, até o quadro 9:

Quadro 3 - Cultura Digital (Utilização de Ferramentas Digitais)

Competência 5. Cultura Digital				
Dimensão: COMPUTAÇÃO E PROGRAMAÇÃO				
Subdimensão: Utilização de Ferramentas Digitais				
	Até 3 Ano do Ensino Fundamental	Até 6 Ano do Ensino Fundamental	Até 9 Ano do Ensino Fundamental	Até 3 Ano do Ensino Médio
UTILIZAÇÃO DE FERRAMENTAS E PERIFÉRICOS PARA APRENDER A PRODUZIR	Utiliza recursos multimídia adequados (por exemplo livros)	Utiliza ferramentas de finalidade geral e periféricos para ampliar	Utiliza uma variedade de ferramentas multimídia e periféricos para auxiliar	Utiliza uma variedade de ferramentas multimídia e periféricos para auxiliar na

	interativos e softwares educativos) para apoiar a sua aprendizagem.	a sua produtividade pessoal, suprir déficits de habilidades e facilitar a aprendizagem.	na produtividade e pessoal e na aprendizagem ao longo de todo o currículo.	produtividade pessoal e na aprendizagem ao longo de todo o currículo e utiliza dispositivos móveis/emuladores para projetar, desenvolver e implementar aplicativos de computação móvel.
--	---	---	--	---

Fonte: BNCC (2017)

Quadro 4 - Cultura Digital (Multimídia)

Competência 5. Cultura Digital				
Dimensão: COMPUTAÇÃO E PROGRAMAÇÃO				
Subdimensão: Multimídia				
	Até 3 Ano do Ensino Fundamental	Até 6 Ano do Ensino Fundamental	Até 9 Ano do Ensino Fundamental	Até 3 Ano do Ensino Médio
UTILIZAÇÃO DE RECURSOS TECNOLÓGICOS PARA DESENHAR, DESENVOLVER, PUBLICAR, TESTAR E APRESENTAR PRODUTOS PARA DEMONSTRAR CONHECIMENTO E RESOLVER PROBLEMAS	Cria produtos multimídia adequados ao seu desenvolvimento, com o apoio de professores, familiares ou colegas	Utiliza ferramentas tecnológicas (por exemplo, ferramentas de autoria multimídia e de textos, apresentação, ferramentas para web, câmeras digitais e scanners) para escrita, comunicação e atividades de	Utiliza recursos tecnológicos para desenhar, desenvolver, publicar e apresentar produtos (por exemplo, páginas web, aplicativos móveis e animações), para demonstrar conhecimentos e resolver problemas	Utiliza recursos tecnológicos para desenhar, desenvolver, publicar e apresentar produtos cada vez mais sofisticados, para demonstrar conhecimentos e resolver problemas mais complexos

		publicação individuais e colaborativas		
--	--	--	--	--

Fonte: BNCC (2017)

Quadro 5 - Cultura Digital (Linguagens de Programação)

Competência 5. Cultura Digital				
Dimensão: COMPUTAÇÃO E PROGRAMAÇÃO				
Subdimensão: Linguagens de Programação				
	Até 3 Ano do Ensino Fundamental	Até 6 Ano do Ensino Fundamental	Até 9 Ano do Ensino Fundamental	Até 3 Ano do Ensino Médio
UTILIZAÇÃO DE LINGUAGENS DE PROGRAMAÇÃO PARA SOLUCIONAR PROBLEMAS	Reúne e organiza informações utilizando ferramentas de mapeamento de conceitos, inclusive fluxogramas.	Constrói um programa como um conjunto de instruções passo a passo a serem executadas, implementa soluções para problemas utilizando uma linguagem de programação visual baseada em blocos.	Implementa soluções para problemas utilizando uma linguagem de programação, inclusive looping, expressões condicionais, lógica, expressões variáveis e funções.	Utiliza interfaces de Programação de Aplicações (Application Programming Interfaces – API) e bibliotecas para aprimorar soluções de programação, utiliza diversos métodos para identificar e corrigir problemas de programação (por exemplo, casos de teste, teste de unidades, caixa branca, caixa preta, testes de integração).

Fonte: BNCC (2017)

Quadro 6 - Cultura Digital (Domínio de Algoritmos)

Competência 5. Cultura Digital				
Dimensão: PENSAMENTO COMPUTACIONAL				
Subdimensão: Domínio de Algoritmos				
	Até 3 Ano do Ensino Fundamental I	Até 6 Ano do Ensino Fundamental	Até o 9 Ano do Ensino Fundamental I	Até o 3 Ano do Ensino Médio
COMPREENSÃO E ESCRITA DE ALGORITMOS	Utiliza recursos tecnológicos (por exemplo, enigmas e programas de raciocínio lógico) para solucionar problemas adequados a sua idade.	Desenvolve a compreensão simples de um algoritmo (por exemplo, pesquisa, sequência de eventos, organização), utilizando exercícios sem computador.	Escreve um algoritmo como uma sequência de instruções que podem ser processadas por um computador.	Explica como sequência, seleção, iteração e recursão são blocos que compõem os algoritmos.
AVALIAÇÃO DE VANTAGENS E DESVANTAGENS DE DIFERENTES ALGORITMOS	---	---	Descreve vantagens e desvantagens associadas a diferentes algoritmos utilizados para solucionar o mesmo problema.	Avalia as vantagens e desvantagens de diferentes algoritmos utilizados para solucionar o mesmo problema.
UTILIZAÇÃO DE CLASSES, MÉTODOS, FUNÇÕES E PARÂMETROS PARA DIVIDIR E RESOLVER PROBLEMAS	---	Compreende e utiliza os passos básicos da solução de problemas por algoritmos (por exemplo, questão-problema e exploração, avaliação de amostras de instâncias,	Utiliza os passos básicos da solução de problemas por algoritmos para resolver questões.	Utiliza classes e métodos, funções e parâmetros pré-definidos para dividir um problema complexo em partes mais simples e resolvê-lo.

		design, implementação e testes).		
--	--	----------------------------------	--	--

Fonte: BNCC (2017)

Quadro 7 - Cultura Digital (Visualização e Análise de Dados)

Competência 5. Cultura Digital				
Dimensão: PENSAMENTO COMPUTACIONAL				
Subdimensão: Visualização e Análise de Dados				
	Até 3 Ano do Ensino Fundamental I	Até 6 Ano do Ensino Fundamental	Até 9 Ano do Ensino Fundamental I	Até 3 Ano do Ensino Médio
UTILIZAÇÃO DE DIFERENTES REPRESENTAÇÕES E ABORDAGENS PARA VISUALIZAR E ANALISAR DADOS	Compreende como ordenar (organizar) informações de uma forma útil (por exemplo, ordenar alunos por data de nascimento), com o uso do computador.	Utiliza representações visuais de problemas, estruturas e dados (por exemplo, gráficos, tabelas, diagramas de rede, fluxogramas).	Representa dados de diversas maneiras, inclusive em textos, sons, imagens e números.	Utiliza diferentes abordagens para visualizar e analisar grandes conjuntos de dados.

Fonte: BNCC (2017)

Quadro 8 - Cultura Digital (Mundo Digital)

Competência 5. Cultura Digital
Dimensão: CULTURA E MUNDO DIGITAL
Subdimensão: Mundo Digital

	Até 3 Ano do Ensino Fundamental I	Até 6 Ano do Ensino Fundamental I	Até 9 Ano do Ensino Fundamental	Até 3 Ano do Ensino Médio
COMPREENSÃO DO IMPACTO DAS TECNOLOGIAS NA VIDA DAS PESSOAS E NA SOCIEDADE, INCLUINDO NAS RELAÇÕES SOCIAIS, CULTURAIS E COMERCIAIS	Identifica a presença e os efeitos positivos e negativos da tecnologia na vida das pessoas.	Identifica o impacto da tecnologia na vida das pessoas e da sociedade (por exemplo, redes sociais, cyberbullying, computação e comunicação móvel, tecnologias web, segurança cibernética e virtualização).	Demonstra conhecimento sobre as mudanças nas tecnologias de informação com o passar do tempo e os efeitos dessas transformações sobre a educação, o mundo do trabalho e a sociedade, analisa os impactos positivos e negativos da computação sobre a cultura humana.	Discute o impacto das tecnologias sobre as relações sociais (por exemplo, cidadania digital, exclusão digital, governo digital), comerciais (por exemplo, rastreamento automático de produtos, transações financeiras automatizadas, e-commerce, computação em nuvem) e culturais (por exemplo, redes sociais, mídias, comunicação intercultural).

Fonte: BNCC (2017)

Quadro 9 - Cultura Digital (Uso Ético)

Competência 5. Cultura Digital				
Dimensão: CULTURA E MUNDO DIGITAL				
Subdimensão: Uso Ético				
UTILIZAÇÃO DAS TECNOLOGIAS, MÍDIAS E DISPOSITIVOS	Até 3 Ano do Ensino Fundamental	Até 6 Ano do Ensino Fundamental	Até 9 Ano do Ensino Fundamental	Até 3 Ano do Ensino Médio

<p>DE COMUNICAÇÃO MODERNOS DE FORMA ÉTICA, COMPARANDO COMPORTAMENTOS ADEQUADOS E INADEQUADOS</p>	<p>Identifica comportamentos sociais e éticos positivos e negativos no uso de tecnologias e reconhece práticas de cidadania digital responsáveis (comportamentos legais e éticos).</p>	<p>Discute e compreende questões éticas relativas ao uso das tecnologias e das redes sociais (por exemplo, igualdade de acesso, segurança, privacidade, direitos autorais e propriedade intelectual) e as consequências do uso inadequado e antiético.</p>	<p>Exibe comportamentos legais e éticos no uso das tecnologias e das redes sociais e discute as consequências do mau uso.</p>	<p>Utiliza as tecnologias, mídias e dispositivos de comunicação modernos de forma ética e compara comportamento adequados e inadequados em redes sociais.</p>
---	--	--	---	---

Fonte: BNCC (2017)

3. OBJETOS DE APRENDIZAGEM (OAs)

Afinal, o que é um Objeto de Aprendizagem (OA)?

David Wiley em seu artigo intitulado *Learning objects need instructional design theory*, (WILEY, 2002, p. 115), define os Objetos de Aprendizagem (OAs) como “sendo qualquer recurso digital que possa ser reutilizado e aplicado como apoio a aprendizagem.”, ou segundo (SÁ FILHO; MACHADO, 2004, p. 3-4) que definem os Objetos de Aprendizagem (OAs) como sendo “[...] recursos digitais que podem ser usados, reutilizados e combinados com outros objetos para formar um ambiente de aprendizagem rico e flexível”.

A definição atribuída por Wiley (2002) sobre os Objetos de Aprendizagem (OAs) é bastante ampla, já que considera um Objeto de Aprendizagem (OA) qualquer tipo de recurso digital que possa ser reutilizado para apoiar a aprendizagem. Outros autores, como Braga *et al.* (2012), Galafassi *et al.* (2014), Flôres (2008), Rebouças; Maia,

Scaico (2021), consideram os Objetos de Aprendizagem (OAs) como sendo aqueles materiais que podem interferir diretamente na prática pedagógica do professor refletindo na aprendizagem dos alunos.

As definições de Objetos de Aprendizagem são as mesmas em alguns aspectos, entretanto, considera-se relevante adotar o conceito adequado de acordo com o objetivo que o professor pretende alcançar no processo de ensino e aprendizagem (AGUIAR; FLÔRES, 2014; TAROUÇO *et al.*, 2003; TAROUÇO *et al.*, 2014).

3.1 Tipos de Objetos de Aprendizagem (OAs)

Braga *et al.* (2012) aponta que existem diversos tipos de recursos digitais que podem ser considerados um Objetos de Aprendizagem (OAs) como sendo:

- **Imagem:** uma imagem pode ser utilizada para apoiar a aprendizagem, pois uma imagem pode ser considerada como representação de uma pessoa ou coisa;
- **Áudio:** um áudio pode atuar de forma isolada como um Objeto de Aprendizagem desde que seja direcionado ao ensino;
- **Vídeo:** um vídeo nada mais é do que a gravação de imagens em movimento ou uma animação constituída por fotos em uma dada sequência. Um vídeo utilizado para dar apoio e suporte ao ensino pode ser considerado um Objeto de Aprendizagem;
- **Animações:** animar significa dar a sensação de vida ou de movimento a objetos estáticos, que no caso podem ser imagens, textos, avatares, etc., acompanhadas ou não de sons. Animações interativas podem vir a ser ferramentas didáticas valiosas no auxílio àqueles alunos com alguma dificuldade na abstração de conceitos pois seu uso estimula os processos cognitivos como a memória, percepção e linguagem. Proporcionam ainda um ambiente lúdico para o desenvolvimento de atividades escolares de ensino-aprendizagem. Sendo assim, uma animação pode ser considerada um Objeto de Aprendizagem; Existem três tipos básicos de animações

com alguns subtipos que são classificados de acordo com a forma como as imagens são criadas, os materiais envolvidos no processo criativo e na tecnologia utilizada: a animação tradicional, a animação stop-motion e a animação por computador;

- **Simulação:** é uma técnica de estudar o comportamento e as reações de determinados sistemas naturais ou artificiais por meio de modelos. As simulações executam animações que representam esse modelo da natureza e com isso podem ser muito utilizados como Objetos de Aprendizagem;
- **Hipertexto:** Um hipertexto é mais conhecido como página da internet. O mesmo pode ser utilizado como apoio pedagógico, portanto, pode ser considerado como um Objeto de Aprendizagem;
- **Softwares:** os softwares são programas computacionais que permitem realizar determinadas tarefas e também resolver problemas de forma automatizada e muito rapidamente. Diversos softwares podem ser utilizados para apoiar a aprendizagem de maneira direta e por isso podem ser considerados Objetos de Aprendizagem (BRAGA *et al.* 2012, p. 14 - 17).

3.2 Uso e Projeto de um Objeto de Aprendizagem (OA)

Aguiar; Flôres (2014, p. 23), apontam que “o projeto e a construção de um Objeto de Aprendizagem (OA), envolvem um arranjo de habilidades multidisciplinares”. Gagné *et al.* (2005, citado por AGUIAR; FLÔRES, 2014, p. 23), relata que para produzir Objetos de Aprendizagem (OAs), pode-se usar os princípios do projeto instrucional:

- a) os objetivos do material pedagógico;
- b) o público alvo (suas habilidades, conhecimentos, estilos preferenciais de aprendizagem, estilos cognitivos);
- c) a interface (com vistas a maximizar a usabilidade);
- d) as estratégias de interatividade;
- e) as ferramentas que serão utilizadas para sua construção;
- f) e os recursos humanos e financeiros disponíveis.

Segundo Aguiar; Flôres (2014, p. 23), os fatores elencados acima, “poderão influenciar de forma direta no tamanho e na qualidade do material que será desenvolvido”. Segundo as autoras supracitadas, antes de iniciar o projeto instrutivo, o professor deve levar em conta os seis princípios listados acima. Além do mais, o Objeto de Aprendizagem (AO) deve ser atraente e possuir o potencial de envolver o aluno na atividade pedagógica e ocorra aprendizagem.

3.3 Perspectiva Pedagógica e Técnica dos OAs

Dias *et al.* (2009) aponta que os Objetos de Aprendizagem (OAs) possuem duas perspectivas: **a perspectiva pedagógica** e **a perspectiva técnica**. Segundo o mesmo autor, as características relacionadas à dimensão pedagógica fazem referência à elaboração de Objetos de Aprendizagem (OAs) que facilitem o trabalho dos professores e alunos, visando à aquisição do conhecimento. Tendo isso em mente, Galafassi *et al.* (2014, p. 100) relata que podemos considerar como **aspectos pedagógicos** importantes:

- **Interatividade:** indica se há suporte às consolidações e ações mentais, requerendo que o aluno interaja com o conteúdo do Objeto de Aprendizagem (OA) de alguma forma, podendo ver, escutar ou responder algo;
- **Autonomia:** indica se os Objetos de Aprendizagem (OAs) apoiam a iniciativa e tomada de decisão;
- **Cooperação:** indica se há suporte para aos alunos trocarem opiniões e trabalhar coletivamente sobre o conceito apresentado;
- **Cognição:** refere-se às sobrecargas cognitivas alocadas na memória do aluno durante o processo de ensino-aprendizagem;

- **Afetividade:** refere-se aos sentimentos e motivações do aluno com sua aprendizagem e durante a interação com o Objeto de Aprendizagem.

De acordo com Braga *et al.* (2012, p. 27), as **características técnicas** dos Objetos de Aprendizagem (OAs), são definidas de acordo com **três teorias**:

- a. Normas de qualidade de software ISO/IEC 9126;
- b. Itens de avaliação sugeridos pela Learning Object Review Instrument (LORI),
- c. Índices de satisfação sugeridos pela Computer Education Management Association (CEdMA, 2001).

Então, seguindo estas três teorias, as **características técnicas** dos Objetos de Aprendizagem (OAs), segundo a autora supracitada, ficam definidas como:

- **Acessibilidade:** indica se o Objeto de Aprendizagem pode ser acessado por diferentes tipos de usuários (ex.: deficientes visuais e auditivos etc.), em diferentes lugares (ex. lugares com acesso à internet e sem acesso à internet, etc.), e por diferentes tipos de dispositivos (ex. computadores, notebooks, dispositivos móveis, etc.);
- **Agregação:** indica se os componentes do Objeto de Aprendizagem podem ser agrupados em conjuntos maiores de conteúdo (ex. estruturas tradicionais de um curso);
- **Confiabilidade:** indica que o Objeto de Aprendizagem não apresenta defeitos técnicos ou problemas no conteúdo pedagógico;
- **Disponibilidade:** indica se o Objeto de Aprendizagem está disponível para ser utilizado;

- **Durabilidade:** indica se o Objeto de Aprendizagem se mantém intacto quando o repositório em que ele está armazenado muda ou sofre problemas técnicos;
- **Facilidade de instalação:** indica se o Objeto de Aprendizagem pode ser facilmente instalado caso ele exija esse recurso;
- **Granularidade:** é a extensão à qual um Objeto de Aprendizagem é composto por componentes menores e reutilizáveis.
- **Interoperabilidade:** medida do esforço necessário para que os dados dos Objetos de Aprendizagem (OAs) possam ser integrados a vários sistemas;
- **Manutenibilidade:** é a medida de esforço necessária para alterações do Objeto de Aprendizagem (OA);
- **Reusabilidade:** indica as possibilidades de reutilizar os Objetos de Aprendizagem (OAs) em diferentes contextos ou aplicações. Essa é a principal característica de um Objeto de Aprendizagem e pode ser influenciada por todas as demais (BRAGA *et al.* 2012, p. 27-28).

Braga *et al.* (2012) adverte que nem todo Objeto de Aprendizagem (OA) possui todas as características citadas acima, no entanto, quanto mais dessas características o Objeto de Aprendizagem (OA) apresentar, maior será a sua capacidade de reutilização em diversos contextos pedagógicos.

3.4 Repositórios de Objetos de Aprendizagem (OAs)

Segundo Braga *et al.* (2012, p. 66), após serem produzidos, os Objetos de Aprendizagem (OAs) precisam ser “publicados em um local em que os usuários possam facilmente buscá-los e recuperá-los para uso futuro [...]”. No Brasil, o maior acervo de Objetos de Aprendizagem (OAs) é mantido por universidades públicas

(SILVA; SOARES; SOUZA, 2021). O quadro 10 exemplifica alguns repositórios de Objetos de Aprendizagem (OAs) existentes no Brasil e em outros países:

Quadro 10 - Repositórios de Objetos de Aprendizagem (OAs)

Repositório (Instituição) Endereço URL	Área do Conhecimento
Repositórios Nacionais	
Laboratório Didático Virtual – LABVIRT (USP) http://www.labvirtq.fe.usp.br/indice.asp	Química
Laboratório Virtual de Matemática (UNIJUÍ) http://www.projetos.unijui.edu.br/matematica	Matemática
PROATIVA (UFC) http://www.proativa.virtual.ufc.br	Diversas Áreas
LAPREN (PUC-RS) http://www.pucrs.br/logos/lapren	Biologia, Física, Matemática, Química, Língua Estrangeira – Inglês, Língua Portuguesa
Portal do Professor (MEC, MCTIC) http://www.portaldoprofessor.mec.gov.br	Diversas Áreas
LABVIRT (USP) http://www.labvirt.fe.usp.br/	Física e Química
CESTA (UFRGS) http://www.cesta2.cinted.ufrgs.br/xmlui	Diversas Áreas
Escola Digital (Escola Digital) https://rede.escoladigital.org.br/	Diversas Áreas
Portal Educacional do Estado do Paraná (Secretaria de Educação – PR) http://www.diaadia.pr.gov.br/	Diversas Áreas
Portal Domínio Público (MEC) http://www.dominiopublico.gov.br	Diversas Áreas
Repositórios Internacionais	

Khan Academy (Khan Academy) http://www.pt.khanacademy.org	Matemática, Ciências, Artes, Humanidades, Economia
MIT Opencourseware (Massachusetts Institute of Technology) http://www.ocw.mit.edu/index.htm	Diversas Áreas
BBC Learning English (BBC) http://www.bbc.co.uk/worldservice/learningenglish	Língua Inglesa
MERLOT (California State University) http://www.merlot.org/merlot/index.br	Diversas Áreas

Fonte: Adaptado de (SILVA; SOARES; SOUZA, 2021)

4. O PENSAMENTO COMPUTACIONAL (PC)

O termo Pensamento Computacional (PC), foi cunhado originalmente pela pesquisadora Jeannet Wing em meados de 2006. O termo traduz-se como sendo o “processo de pensamento envolvido na formulação de um problema e na expressão de sua(s) solução(s) de forma que um computador - humano ou máquina - possa efetivamente executar”. Em seus trabalhos, Wing (2014) expressa a ideia do “Computational Thinking”, ou seja, o Pensamento computacional é o processo de pensamento envolvido na formulação de um problema e na expressão de sua(s) solução(ões) de tal maneira que um computador - humano ou máquina - possa executá-la(s) de forma efetiva (WING, 2014, p.1)

Nesse sentido, o Pensamento Computacional (PC), segundo Wing (2014), expressaria a ideia de aspectos lógicos e computacionais no mundo ao redor, a fim de que se possa abordá-los usando métodos e técnicas provenientes da ciência computacional.

4.1 Os Quatro Pilares do Pensamento Computacional (PC)

O Pensamento Computacional (PC) envolve identificar um problema complexo e quebrá-lo em pedaços menores e mais fáceis de manipular (Decomposição). Cada problema menor resultante pode ser analisado individualmente com maior profundidade, identificando problemas semelhantes que já foram solucionados anteriormente (Reconhecimento de Padrões), focando apenas nos detalhes mais importantes, enquanto informações irrelevantes são ignoradas (Abstração).

A figura 3 ilustra uma síntese dos **Quatro Pilares** do Pensamento Computacional (PC)

Figura 3 - Os Quatro Pilares do Pensamento Computacional (PC)



Fonte: Adaptado de (SANTOS, 2019)

A utilização destes elementos que compõe os Quatro Pilares do Pensamento Computacional (PC), contribui para o desenvolvimento do raciocínio lógico e para a

capacidade de resolução de problemas da vida diária, individualmente ou coletivamente (BRACKMANN, 2017; PASQUAL JÚNIOR 2019; SANTOS, 2019; WING, 2014).

4.2 Os Benefícios do Pensamento Computacional (PC)

Para Brackmann (2017, p. 44-46), entre os diversos benefícios levantados a favor do Pensamento Computacional (PC), destacam-se os seguintes:

- Empregos: demanda por profissionais melhor qualificados;
- Compreender o mundo em que vive: a realidade atual é informatizada;
- Transdisciplinaridade: o conhecimento computacional em outras áreas;
- Proficiência digital: aprender e aumentar os seus domínios cognitivos;
- Ampliação e produtividade: oportunidade para mais pessoas;
- Diminuição de fronteiras: acessível em diversas partes, a todo tempo;
- Trabalho em equipe: construções conjuntas e compartilhamento;
- Inclusão de Minorias: Informática para todos

Apesar de estudos a favor, o Pensamento Computacional (PC) ainda é um tema recente no Brasil. É possível encontrar diversos materiais relacionados ao Pensamento Computacional (PC) na Code.org¹. Este site conta com um fórum de discussão além de uma seção destinada a educadores com sugestões para cada nível de ensino. Existe também o site brasileiro do CIEB – Centro de Educação para a

¹ A Code.org é uma organização sem fins lucrativos dedicada a expandir o acesso à ciência da computação em escolas e aumentar a participação das mulheres e das minorias não representadas. Nossa visão é de que todo estudante em toda escola tenha a oportunidade de aprender ciência da computação, assim como aprende biologia, química ou álgebra. Disponível em: <https://code.org/international/about>. Acesso em: 28 de jul. de 2021.

Inovação Brasileira² que traz diversos conteúdos sobre os temas Pensamento Computacional (PC) e Tecnologias Digitais aplicadas na educação. Sobre o Pensamento Computacional (PC).

4.3 O Pensamento Computacional (PC) na BNCC (2017)

A BNCC (2017) demonstra preocupação com as constantes transformações sociais que as tecnologias impõem e também com os impactos das mesmas sobre a Educação Básica brasileira, pontua Brackmann (2017). Ela aponta três características já citadas anteriormente, mas que devem ser articuladas com os objetivos de ensino e aprendizagem e, descreve pela primeira vez, sobre o que envolve o Pensamento Computacional (PC), [...] envolve as capacidades de compreender, analisar, definir, modelar, resolver, comparar e automatizar problemas e suas relações de forma metódica e sistemática, por meio do desenvolvimento de algoritmos (BNCC, 2017, p. 473-474)

A BNCC (2017) traz uma seção intitulada: “A progressão das aprendizagens essenciais do Ensino Fundamental para o Ensino Médio”, na qual existe a primeira menção, agora de forma explícita, sobre o Pensamento Computacional (PC) na transição do Ensino Fundamental para o Ensino Médio: [...] o Ensino Fundamental, centra-se na compreensão de conceitos e procedimentos em seus diferentes campos e no desenvolvimento do **pensamento computacional**, visando à resolução e formulação de problemas em contextos diversos.

No Ensino Médio, na área de Matemática e suas Tecnologias, os estudantes devem consolidar os conhecimentos desenvolvidos na etapa anterior e agregar novos, ampliando o leque de recursos para resolver problemas mais complexos, que exijam maior reflexão e abstração (BNCC, 2017, p. 471).

Constata-se uma tímida tendência à inclusão na BNCC (2017) de habilidades e competências relativas ao Pensamento Computacional (PC) na Educação Básica. A

² O CIEB é uma associação sem fins lucrativos, criada em 2016, com o intuito de promover a cultura de inovação na educação pública brasileira. Disponível em: <https://cieb.net.br/#o-que-fazemos>. Acesso em: 28 de jul. de 2021.

figura 4 e na sequência a figura 5, mostram com detalhes o avanço do **Pensamento Computacional (PC)** e suas subdimensões **Domínio de Algoritmos** e **Visualização e Análise de Dados** nas etapas da Educação Básica:

Figura 4 – Subdimensão: Domínio de Algoritmos

Cultura Digital. Dimensão: Pensamento Computacional				
Subdimensão: Domínio de algoritmos				
	Até 3º Ensino Fundamental	Até 6º Ensino Fundamental	Até 9º Ensino Fundamental	Até 3º Ensino Médio
Compreensão e escrita de algoritmos.	Utiliza recursos tecnológicos (por exemplo, enigmas e programas de raciocínio lógico) para solucionar problemas adequados a sua idade.	Desenvolve compreensão simples de um algoritmo (por exemplo, pesquisa, sequência de eventos, organização), utilizando exercícios sem computador.	Escreve um algoritmo como uma sequência de instruções que podem ser processadas por um computador.	Explica como sequência, seleção, iteração e recursão são blocos que compõem os algoritmos.
avaliação de vantagens e desvantagens de diferentes algoritmos.	Fonte: Ministério da Educação - MEC		Descreve vantagens e desvantagens associadas a diferentes algoritmos utilizados para solucionar o mesmo problema.	Avalia as vantagens e desvantagens de diferentes algoritmos utilizados para solucionar o mesmo problema.
Utilização de classes, métodos, funções e parâmetros para dividir e resolver problemas.	Luciano Sathler	Compreende e utiliza os passos básicos da solução de problemas por algoritmos (por exemplo, questão problema e exploração, avaliação de amostras de instâncias, design, implementação e testes).	Utiliza os passos básicos da solução de problemas por algoritmo para resolver questões.	Utiliza classes e métodos, funções e parâmetros pré-definidos para dividir um problema complexo em partes mais simples e resolvê-lo.

Fonte: Adaptado de (SATHLER, 2020)

Figura 5 – Subdimensão: Visualização e Análise de Dados

Subdimensão: Visualização e análise de dados				Luciano Sathler
	Até 3º Ensino Fundamental	Até 6º Ensino Fundamental	Até 9º Ensino Fundamental	Até 3º Ensino Médio
Utilização de diferentes representações e abordagens para visualizar e analisar dados.	Compreende como ordenar (organizar) informações de uma forma útil (por exemplo, ordenar alunos por data de nascimento), com o uso do computador.	Utiliza representações visuais de problemas, estruturas e dados (por exemplo, gráficos, tabelas, diagramas de rede, fluxogramas).	Representa dados de diversas maneiras, inclusive em textos, sons, imagens e números.	Utiliza diferentes abordagens para visualizar e analisar grandes conjuntos de dados.
Fonte: Ministério da Educação - MEC				

Fonte: Adaptado de (SATHLER, 2020)

Como visto, o Pensamento Computacional (PC) está atrelado a competência **Cultura Digital** na BNCC (2017) como uma de suas **três dimensões** e possuindo as subdimensões **Domínio de Algoritmos** e **Visualização e Análise de Dados**.

4.4 A Escola e o Pensamento Computacional (PC)

Segundo aponta André (2018, p. 104), a escola tem um papel significativo na formação dos indivíduos, na sua cultura, nas suas relações sociais e, decididamente, em seu processo de interpretação da informação. É uma instituição que reúne condições propícias para desempenhar esse papel. Na escola, os alunos podem ser apresentados a um conjunto organizado e planejado de temas e situações de aprendizagem que pode ser sistematizado gerando oportunidades para que sejam autores do próprio conhecimento.

Diante desse contexto, Santella (2013, citado por ANDRÉ (2018, p. 104), aponta que: “[...] é possível desenvolver o pensamento computacional de forma dinâmica, orientando o trabalho escolar para o conhecimento sobre fenômenos da natureza, incluindo o ser humano e as tecnologias mais próximas e mais distantes, no espaço e no tempo.

Pode-se contemplar os diversos eixos temáticos por meio das bases do Pensamento Computacional (PC). Apesar das muitas possibilidades em usar essa metodologia em sala de aula, infelizmente ainda são tímidas as menções sobre as diretrizes do Pensamento Computacional (PC) na Educação Básica. Mesmo sem muitas orientações a respeito, segundo Pasqual Júnior (2019), existem formas de trabalhar o Pensamento Computacional (PC) com alunos desde a Educação Infantil até o Ensino Superior. O autor cita a Computação Desplugada³, na qual, os alunos trabalham conceitos computacionais na solução de problemas e atividades sem necessariamente utilizarem um computador.

Outra maneira seria através da programação por blocos, onde as características essenciais do Pensamento Computacional (PC) seriam trabalhadas programando robôs em forma de bichos ou personagens de desenho animado, resolvendo problemas, o que poderia contribuir até em processos de alfabetização ajudando crianças a escolher as letras para formar uma palavra (PASQUAL JÚNIOR., 2019).

O Pensamento Computacional (PC) não possui aplicação apenas ao ensino das disciplinas da área das ciências exatas, mas em diversas áreas do conhecimento e

³ Disponível em: <http://desplugada.ime.unicamp.br/>. Acesso: 28 de jul. de 2021.

também da vida social. Nesse sentido, Pasqual Júnior, (2019) relata que o PC se encaixa em todas as áreas do conhecimento e permite aos professores se aproximarem mais dos estudantes. Nas humanidades, auxilia a planejar e construir narrativas, por exemplo. Em matemática, permite materializar diversos conceitos, o que facilita o aprendizado. O estudante que desenvolver habilidades de planejamento e resolução de problemas tem a ganhar na vida escolar, no trabalho e até na vida social (PASQUAL JÚNIOR, 2019).

4.5 Os Benefícios do Pensamento Computacional (PC)

Para Brackmann (2017, p. 44-46), entre os diversos benefícios levantados a favor do Pensamento Computacional (PC), destacam-se os seguintes:

- Empregos: demanda por profissionais melhor qualificados;
- Compreender o mundo em que vive: a realidade atual é informatizada;
- Transdisciplinaridade: o conhecimento computacional em outras áreas;
- Proficiência digital: aprender e aumentar os seus domínios cognitivos;
- Ampliação e produtividade: oportunidade para mais pessoas;
- Diminuição de fronteiras: acessível em diversas partes, a todo tempo;
- Trabalho em equipe: construções conjuntas e compartilhamento;
- Inclusão de Minorias: Informática para todos

Segundo aponta Wing (2014), os benefícios de pensar computacionalmente aprimoram e reforçam as habilidades intelectuais e, portanto, podem ser transferidos para diversas áreas do conhecimento e também do convívio social. Logo, a ciência, a sociedade e nossa economia se beneficiarão das descobertas e inovações produzidas por uma força de trabalho treinada para pensar computacionalmente.

Ainda segundo Wing (2014, citado por ANDRÉ, 2018, p. 100), [...] favorecer o pensamento computacional ter por principal função a formação de pessoas capazes de, não apenas identificar informações, mas principalmente produzir artefatos a partir

da compreensão de conceitos e utilizá-los para enfrentar desafios e refletir sobre o seu cotidiano.

Nesse sentido, o Pensamento Computacional (PC) versa em saber usar o computador como um artefato que estimule o aumento do poder cognitivo e operacional humano, para dessa forma contribuir para o aumento da produtividade, inventividade e criatividade das pessoas (BLIKSTEIN, citado por SANTOS, 2019).

Apesar de estudos a favor, o Pensamento Computacional (PC) ainda é um tema recente no Brasil. É possível encontrar diversos materiais relacionados ao Pensamento Computacional (PC) na Code.org⁴. Este site conta com um fórum de discussão além de uma seção destinada a educadores com sugestões para cada nível de ensino. Existe também o site brasileiro do CIEB – Centro de Educação para a Inovação Brasileira⁵ que traz diversos conteúdos sobre os temas Pensamento Computacional (PC) e Tecnologias Digitais aplicadas na educação.

Sobre o Pensamento Computacional (PC), André (2018, p. 97) finaliza dizendo que nos últimos anos, as concepções a respeito do pensamento computacional passaram por profundas modificações que permitiram aproximar essa proposta ao dia a dia do aluno, ou seja, ao seu mundo real, tornando-a cada vez mais presente e concreta.

5. O MBLOCK COMO SOFTWARE EDUCATIVO

Um Software Educativo (SE) ou aplicativo educacional (app) consiste em um sistema computacional cujo objetivo principal é auxiliar no processo de ensino e aprendizagem ou de autoaprendizagem. No software educativo deve existir uma relação entre a proposta pedagógica, decisões de design e os processos de aprendizagem (SILVA; SOARES; SOUZA, 2021).

⁴ A Code.org é uma organização sem fins lucrativos dedicada a expandir o acesso à ciência da computação em escolas e aumentar a participação das mulheres e das minorias não representadas. Nossa visão é de que todo estudante em toda escola tenha a oportunidade de aprender ciência da computação, assim como aprende biologia, química ou álgebra. Disponível em: <https://code.org/international/about>. Acesso em: 28 de jul. de 2021.

⁵ O CIEB é uma associação sem fins lucrativos, criada em 2016, com o intuito de promover a cultura de inovação na educação pública brasileira. Disponível em: <https://cieb.net.br/#o-que-fazemos>. Acesso em: 28 de jul. de 2021.

Seguindo a linha de software educativo encontra-se a plataforma mBlock. Trata-se de uma linguagem de programação computacional gráfica, utilizando blocos visuais, os quais permitem aos usuários criar as suas próprias histórias animadas, jogos e animações. Foi desenvolvida para que crianças e jovens pudessem desenvolver os seus próprios projetos, aprender a pensar de maneira criativa e sistemática.

Baseia seus objetivos na linguagem de programação em blocos Scratch⁶, criada pelo grupo de pesquisa *Lifelong Kindergarten, do MIT Media Lab do Massachusetts Institute of Technology (MIT)*, liderado por *Mitchel Resnick*⁷. O Scratch segue as ideias construcionistas, estreitamente ligadas à linguagem LOGO, desenvolvida por *Seymour Papert* em 1967. O mBlock por sua vez, também segue a linha construcionista do Scratch e da linguagem LOGO de Papert. Foi criada para ensinar os conceitos de programação para crianças e adolescentes de forma fácil e divertida.

Em uma comparação direta, os dois softwares educativos diferem apenas em poucas funcionalidades ligadas a opções de conexão e controle de robôs educativos e a conexão com a plataforma eletrônica Arduino⁸. O mBlock possui uma interface gráfica e amigável, colocando o usuário em contato direto com suas funções básicas. Nele é possível testar o código inserido em tempo real, ou seja, à medida que se encaixam os blocos lógicos de programação é possível ver o resultado dos comandos imediatamente nos atores, palco e demais elementos que compõe a cena do projeto.

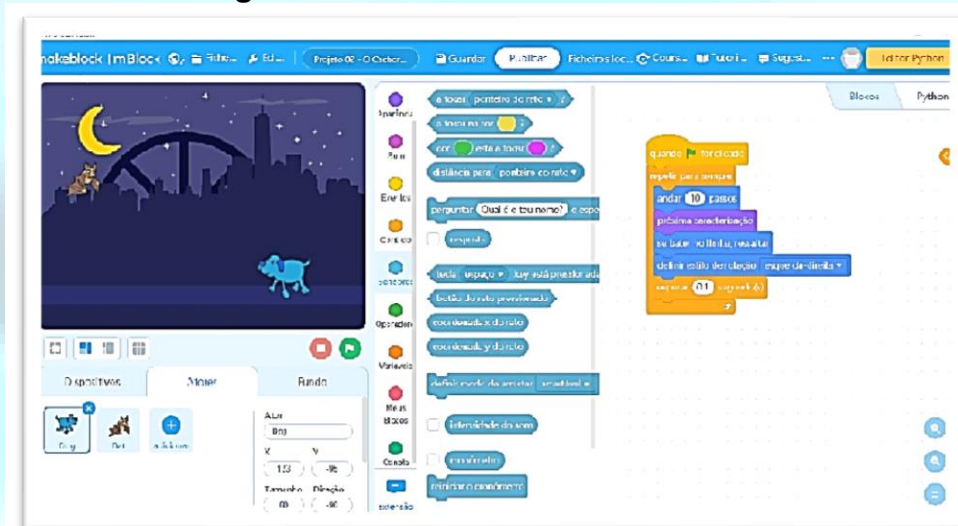
A figura 6, mostra a interface gráfica da plataforma mBlock para computadores e notebooks:

⁶ Scratch é uma linguagem de programação criada em 2007 no Media Lab do **MIT**. Disponível em: <https://pt.wikipedia.org/wiki/Scratch>. Acesso em: 27 de jul. de 2021.

⁷ Disponível em: <https://novaescola.org.br/conteudo/905/mitchel-resnick-a-tecnologia-deve-levar-o-aluno-a-ser-um-pensador-criativo>. Acesso em: 21 de ago. de 2021.

⁸ Arduino é uma plataforma eletrônica de código aberto baseada em hardware e software fáceis de usar. É destinado a qualquer pessoa que faça projetos interativos. Disponível em: <https://www.arduino.cc/>. Acesso em: 29 de jul. de 2021.

Figura 6 - Interface Gráfica do mBlock



Fonte: Elaborado pelo autor, 2021

Apesar da semelhança funcional entre a plataforma mBlock e o Scratch, a escolha do mBlock se deu pelo mesmo possuir as seguintes características específicas:

- Múltiplas opções de Sistema Operacional⁹ (SO) para Computadores e Notebooks: *Windows 32/64 bits, Linux 32/64 bits, Chrome 32/64 bits, OS X 32/64 bits*;
- Opções de Plataformas *Mobile: Android e iOS*;
- Objetivos pedagógicos de um Software Educacional;
- Não requer computadores ou dispositivos móveis robustos para rodar;
- Conexão com modelos diferentes de robôs educativos utilizando Inteligência Artificial (IA);
- Possibilidade de elaborar projetos educativos em consonância com os Quatro Pilares do Pensamento Computacional (PC);
- Transposição da codificação em blocos para a linguagem *Python*¹⁰;

⁹ Sistema Operacional é um conjunto de programas cuja função é gerenciar os recursos do sistema, fornecendo uma interface entre o computador e o usuário. Disponível em: https://pt.wikipedia.org/wiki/Sistema_operativo. Acesso em: 29 de jul. de 2021.

¹⁰ *Python* é uma linguagem de programação de alto nível, interpretada de *script*, imperativa, orientada a objetos, funcional, de tipagem dinâmica e forte. Foi lançada por Guido van Rossum em 1991. Disponível em: <https://pt.wikipedia.org/wiki/Python>. Acesso em: 10 de ago. de 2021.

- Coleta e análise de Dados Científicos visualizados em tabelas e gráficos por meio de sensores de temperatura, umidade, luminosidade, distância, etc.;
- Possibilidade de usar a Educação *STEAM*¹¹ em seus projetos;
- Em conformidade com o modelo de conhecimento CTPC;
- Projetos com a possibilidade de conexão com a Internet das Coisas (IoT)¹²
- Permite também o desenvolvimento da Computação Criativa¹³
- Aprendizagem baseada no conceito de *Design*¹⁴

Uma das características que mais chamou a atenção para a escolha do mBlock em contrapartida ao Scratch neste projeto, foi de o mBlock estar disponível também para **dispositivos móveis Android ou iOS**. Esta característica facilita em o seu uso, pois acredita-se que dessa forma o mBlock possa ser utilizado em qualquer ambiente, offline, diretamente dos *smartphones* dos alunos e professores, dispensando a necessidade de computadores fixos e com acesso à internet, como por exemplo, o laboratório de informática ou de tecnologia.

5.1 O mBlock e a Programação por Blocos

A plataforma mBlock permite a programação de projetos através do encaixe de blocos visuais. A programação por encaixe de blocos, possibilita que cada projeto seja desenvolvido utilizando blocos coloridos, o que remete ao sistema do brinquedo LEGO¹⁵, cujo conceito principal também é o encaixe de blocos. A programação por blocos, facilita a elaboração de projetos simples ou complexos, tanto por professores

¹¹ O modelo STEAM prevê a integração de conhecimentos de Artes, Ciências, Tecnologia, Engenharia e Matemática, possibilitando ao aluno se preparar para desafios como cidadão e também no mercado de trabalho. Disponível em: <https://j.pucsp.br/artigo/educacao-steam-o-que-e-para-que-serve-e-como-usar>. Acesso em: 29 de jul. de 2021.

¹² Internet das coisas (em inglês: Internet of Things, IoT), é um conceito que se refere à interconexão digital de objetos cotidianos com a internet. Disponível em: https://pt.wikipedia.org/wiki/Internet_das_coisas. Acesso em: 11 de ago. de 2021.

¹³ Disponível em: <https://liag.ft.unicamp.br/computacaocriativa/>. Acesso em: 15 de ago. de 2021.

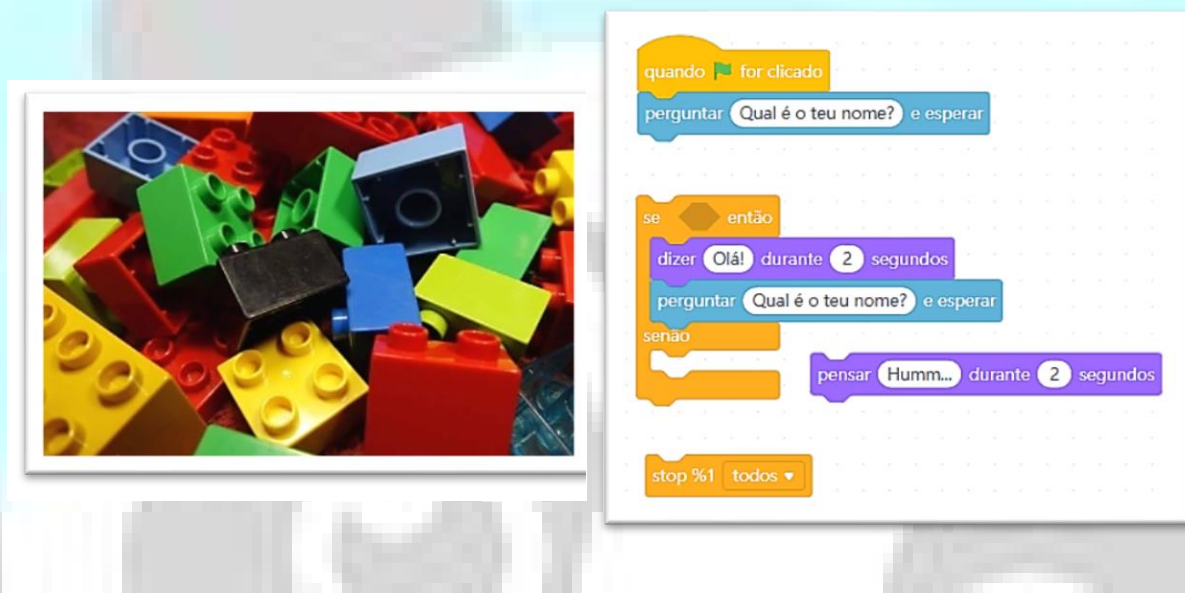
¹⁴ Disponível em: http://pdf.blucher.com.br/s3-sa-east-1.amazonaws.com/designproceedings/cidi2015/cidi_217.pdf. Acesso em: 15 de ago. de 2021.

¹⁵ LEGO Group é uma empresa familiar com sede na Dinamarca. É mais conhecido pela fabricação de brinquedos da marca LEGO, que consistem principalmente em blocos de montar de plástico interligados. Disponível em: https://pt.wikipedia.org/wiki/The_Lego_Group. Acesso em: 15 de jul. de 2021.

quanto por alunos de diversas faixas etárias, com ou sem experiência em linguagens de programação.

A figura 7 ilustra a semelhança dos blocos coloridos de encaixar presente no mBlock semelhante ao conceito de blocos coloridos do brinquedo educativo LEGO:

Figura 7 - Blocos LEGO e a Programação por Blocos do mBlock



Fonte: Compilado pelo autor¹⁶, 2021

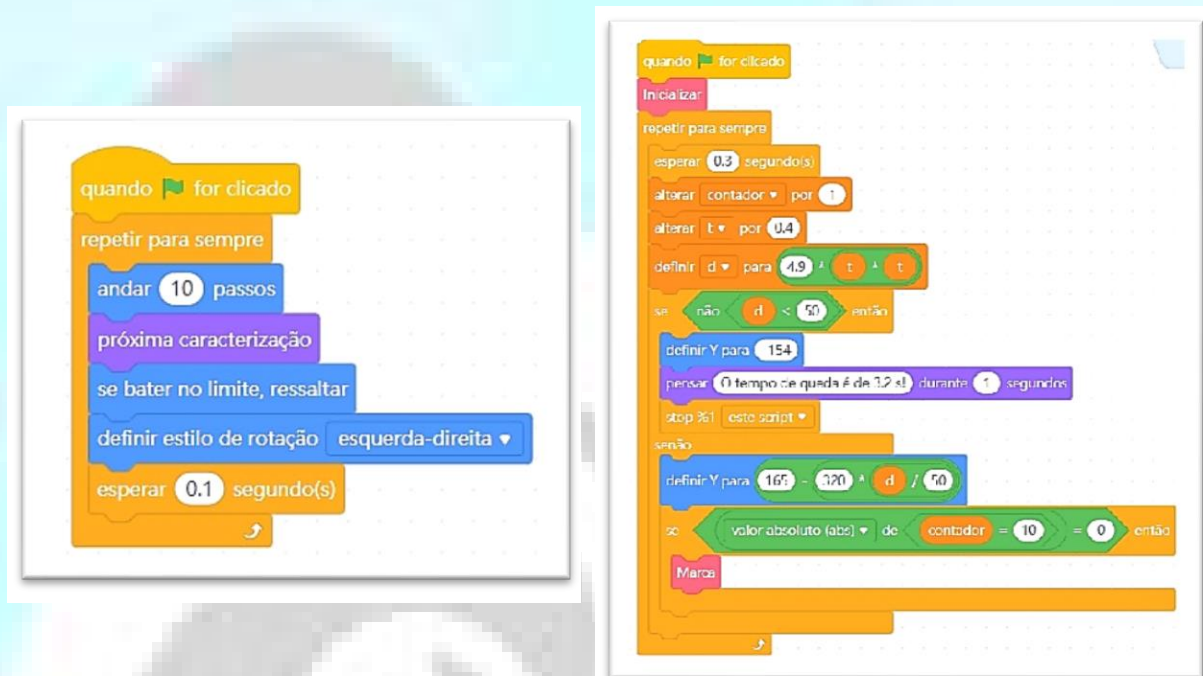
Uma das vantagens em se programar encaixando blocos, característica intrínseca do mBlock, reside em eliminar erros sintáticos presentes em outras linguagens usuais de programação. Dessa forma, acredita-se que docentes e alunos, poderão programar facilmente os seus Objetos de Aprendizagem (OAs) digitais, sem necessitar escrever centenas de linhas de código sujeitas a erros sintáticos.

A programação por blocos permite construir estruturas lógicas de diferentes complexidades, tudo vai depender do conhecimento e objetivo do professor ou do aluno que está por trás da ferramenta de construção. Pode-se iniciar com uma estrutura de blocos simples e avançar para uma estrutura de blocos mais complexa.

¹⁶ Disponível em: <https://www.reviewbox.com.br/lego-melhores-modelos/>. Acesso em: 28 de jul. de 2021.

Exemplos de estruturas de blocos simples e complexa, respectivamente, estão exemplificadas na figura 8:

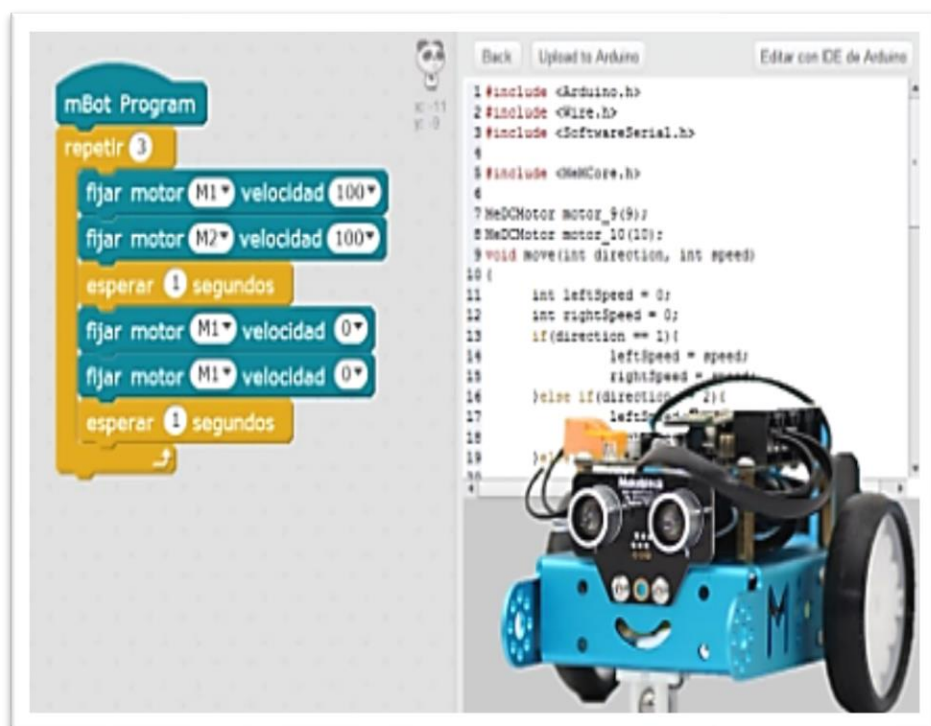
Figura 8 - Estruturas de Blocos Simples e Complexa



Fonte: Elaborado pelo autor, 2021

A plataforma mBlock permite também interagir com novas tecnologias, utilizando diversos recursos disponíveis, como por exemplo, o uso de Inteligência Artificial (IA) para controlar robôs educativos. Esta característica intrínseca do mBlock está ilustrada na figura 9 abaixo:

Figura 9 - Projeto de Robótica (IA) usando o mBlock



Fonte: Plataforma mBlock¹⁷

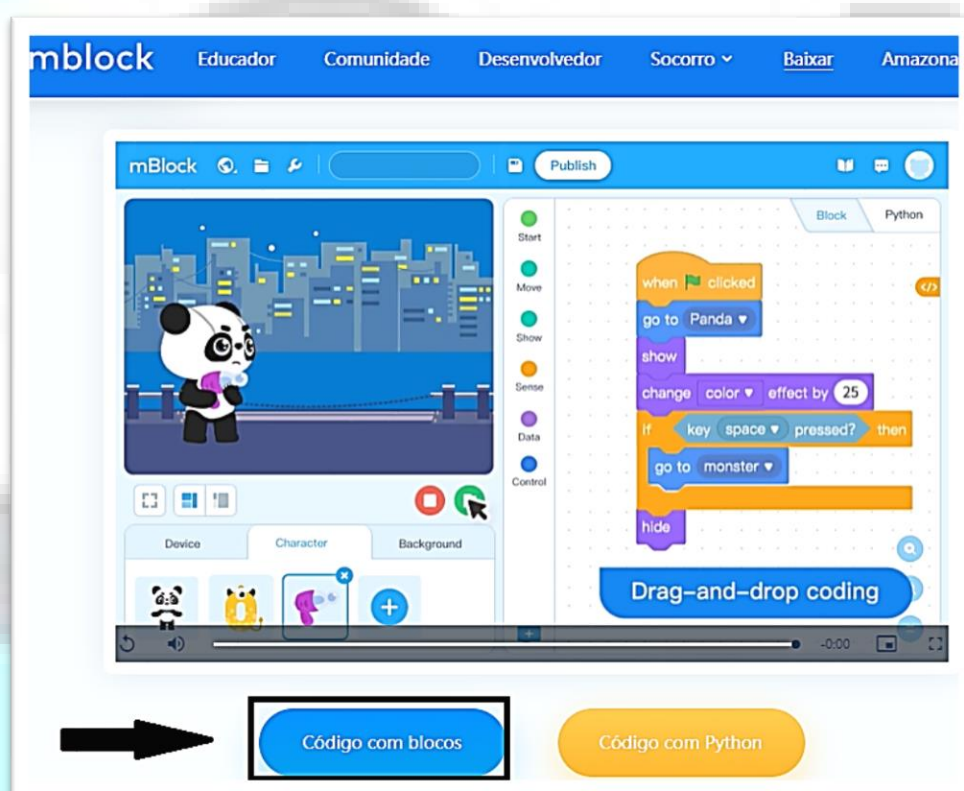
Portanto, acredita-se que utilizando a programação por blocos na elaboração de Objetos de Aprendizagem por meio do mBlock, o professor poderá desenvolver metodologias de ensino voltadas para o uso de tecnologias de ponta, possibilitando conduzir o aluno na busca do conhecimento para uma aprendizagem mais ativa, dinâmica, criativa e também divertida.

¹⁷ Disponível em: <https://mblock.makeblock.com/en-us/>. Acesso em: 10 mar. de 2021.

6. O MBLOCK PARA COMPUTADORES E NOTEBOOKS

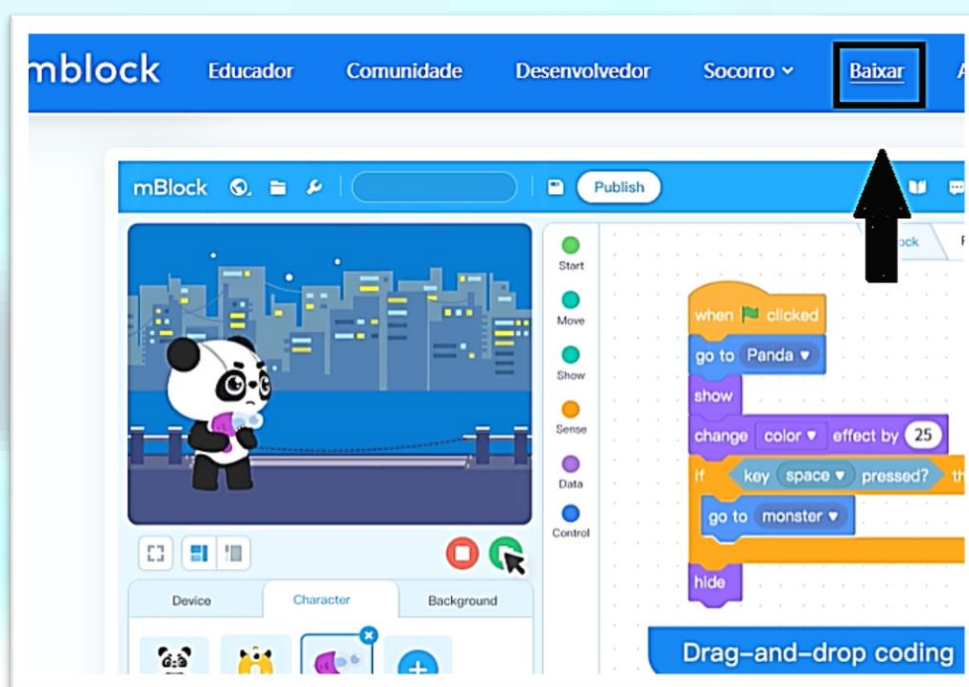
A versão da plataforma mBlock para computadores pode ser utilizada de duas maneiras distintas: uma é online, visitando o site do desenvolvedor em <https://mblock.makeblock.com/en-us/>, clicando no botão “Código com blocos”, conforme ilustra a figura 10, e a outra maneira é *offline*, clicando em “Baixar” para realizar o download do aplicativo por meio do *link* <https://mblock.makeblock.com/en-us/download/> (como mostra a figura 11). A presente pesquisa utilizou a **versão 5.3.0**, *offline*, para computadores equipados com o **Sistema Operacional (SO) Microsoft Windows de 64 bits**.

Figura 10 - Acesso Online à Plataforma mBlock



Fonte: Elaborado pelo autor, 2020

Figura 11 - Acesso Offline à Plataforma mBlock



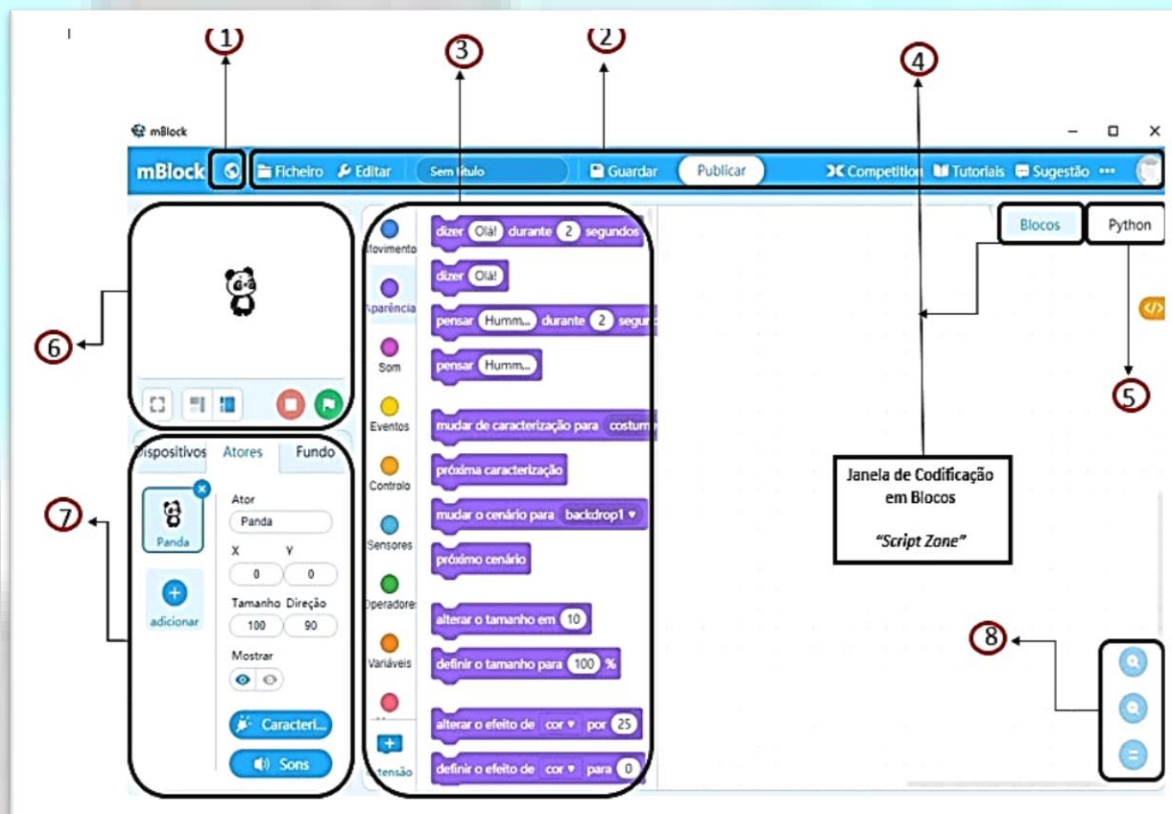
Fonte: Elaborado pelo autor, 2020

A possibilidade de utilizar a plataforma de codificação mBlock online oferece recursos que estão disponíveis somente nessa modalidade. Por meio de sua conta, é permitido ao usuário compartilhar seus projetos, usar e reutilizar projetos de outros usuários, além da possibilidade de que todas as suas criações serem salvas automaticamente em sua conta. Todavia, vale lembrar que os projetos criados **OFFLINE** também podem ser carregados na plataforma **ONLINE** e compartilhados com a comunidade no próprio site da plataforma.

Para utilizar a plataforma mBlock não é necessário digitar nenhum código complicado, pois a codificação “pesada” está “embutida” nos blocos visuais manipuláveis. Ou seja, o usuário precisa apenas se concentrar na lógica das conexões desses blocos, **ARRASTÁ-LOS E SOLTÁ-LOS (Drag-and-Drop)** e encaixá-los de forma coerente e estruturada quando está criando seus projetos. Portanto, não é necessário conhecimento prévio da sintaxe de nenhuma linguagem de programação específica para usar a programação por blocos do mBlock. A codificação utilizando a linguagem

de programação **Python**¹⁸ não será abordada nesta pesquisa. A tela inicial ou interface da plataforma mBlock, na sua versão offline para computadores equipados com o **Sistema Operacional (SO) Microsoft Windows de 32 ou 64 bits**, está organizada como ilustra a figura 12:

Figura 12 - Interface da Plataforma mBlock para Computadores



Fonte: Elaborado pelo autor, 2020

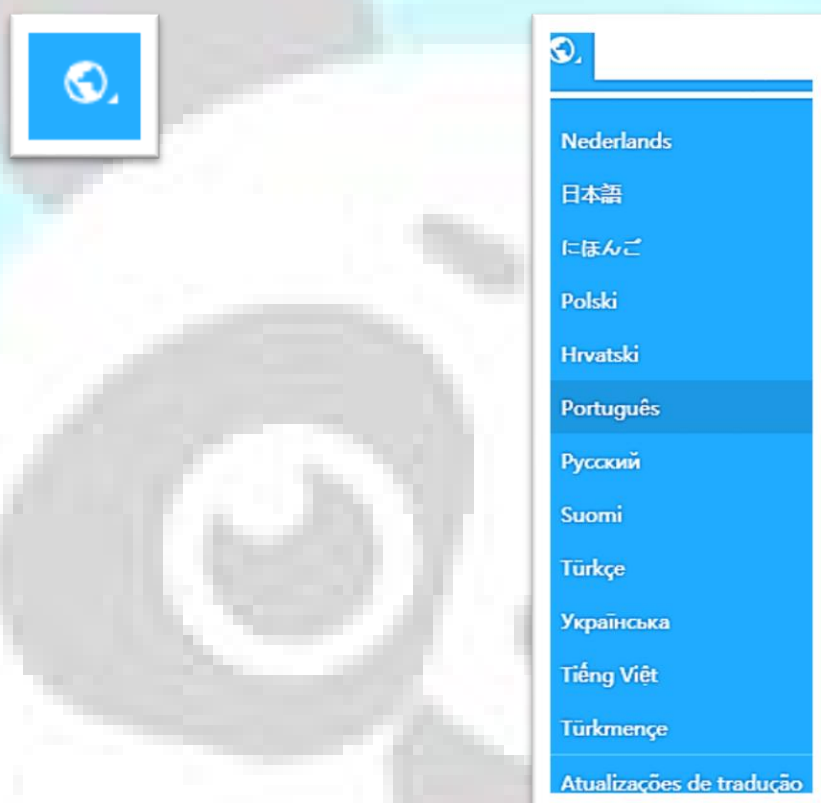


IDIOMA: Por meio do ícone em forma de globo terrestre, pode-se alterar o idioma da plataforma que está disponível em diversas línguas. O idioma inglês é default. Algumas ações na plataforma estão disponíveis somente neste

¹⁸ *Python* é uma linguagem de programação de alto nível, interpretada de *script*, imperativa, orientada a objetos, funcional, de tipagem dinâmica e forte. Foi lançada por Guido van Rossum em 1991. Disponível em: <https://pt.wikipedia.org/wiki/Python>. Acesso em: 10 de ago. de 2021.

idioma, inclusive o aplicativo mBlock para **Smartphones e Tablets**. Usaremos nesta pesquisa o idioma português de Portugal pois a opção para o idioma Português do Brasil ainda não disponível para a atual versão 5.3.0. A opção para troca de idioma da plataforma mBlock está ilustrada na figura 13:

Figura 13 - Opções de Idioma na Plataforma mBlock

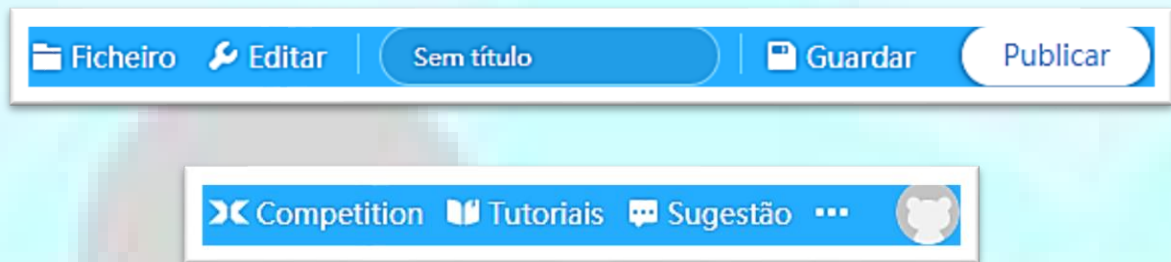


Fonte: Elaborado pelo autor, 2020



BARRA DE MENUS: Oferece basicamente a possibilidade de criar, salvar, abrir, compartilhar, importar, renomear, publicar e acessar um projeto. Possibilita ainda o acesso a tutoriais, programas exemplos e a procura de versões atualizadas da plataforma. A barra de menus está ilustrada na figura 14:

Figura 14 - Barra de Menus na Plataforma mBlock

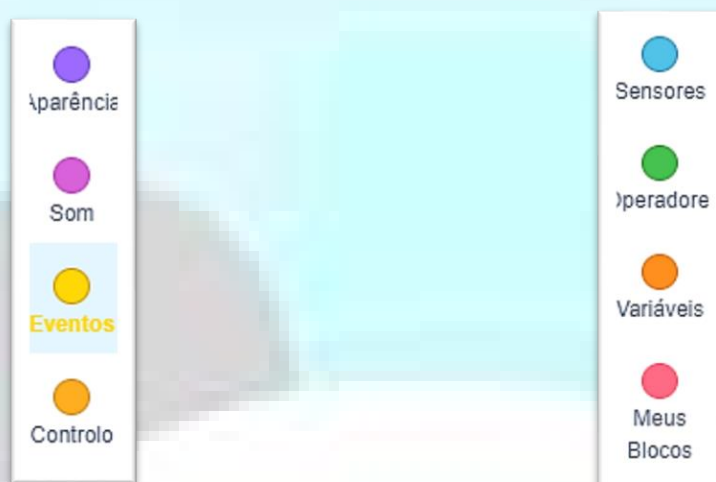


Fonte: Elaborado pelo autor, 2020



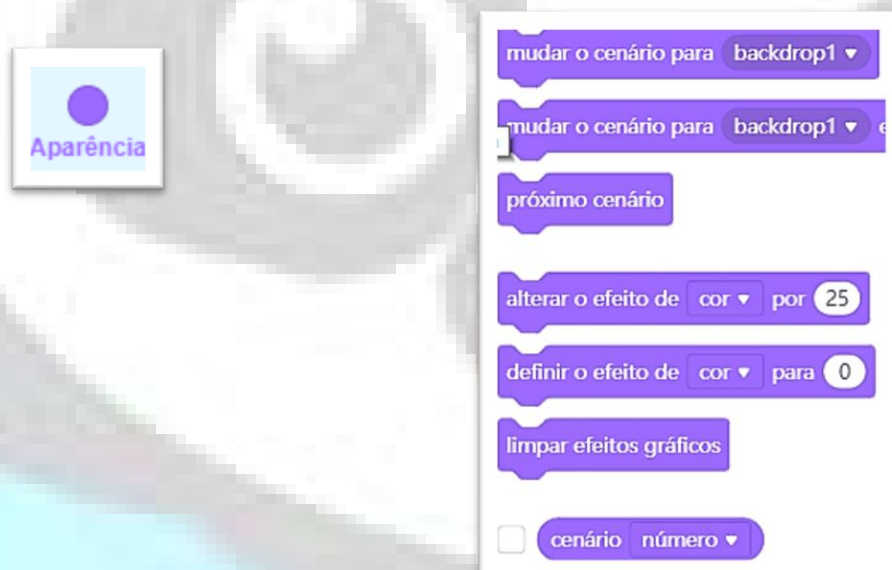
PALETA DE BLOCOS: Este item está dividido (na sua forma mais básica) em uma coluna com oito círculos identificados por cores diferentes e outra coluna contendo os respectivos blocos de comando. As cores dos círculos representam o que se pode construir com os blocos e a sua correspondência funcional e lógica. Sendo eles: **Aparência, Som, Eventos, Controle, Sensores, Operadores, Variáveis e Meus Blocos**. Observe a disposição dos Tipos de Blocos ilustrado na figura 15 bem como a correspondência entre a cor dos círculos e os blocos de comando, como ilustrado na figura 16 na sequência:

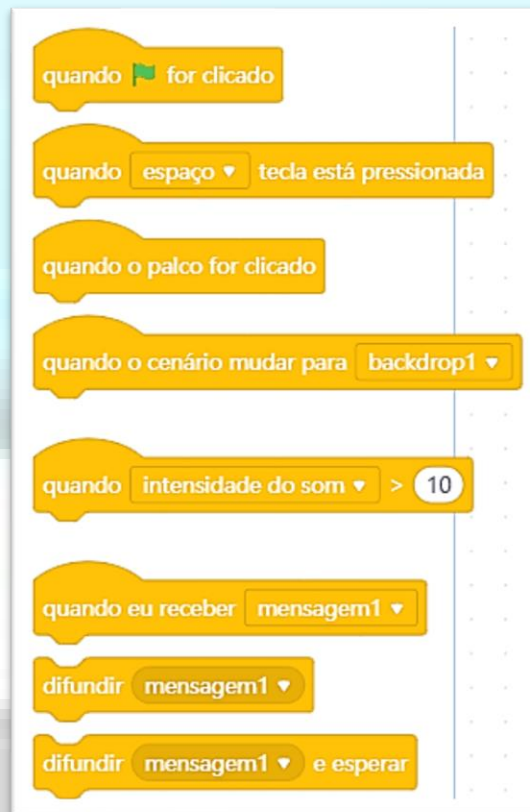
Figura 15 - Tipos de Blocos na Plataforma mBlock



Fonte: Elaborado pelo autor, 2020

Figura 16 - A Cor dos Círculos e os Blocos de Comando

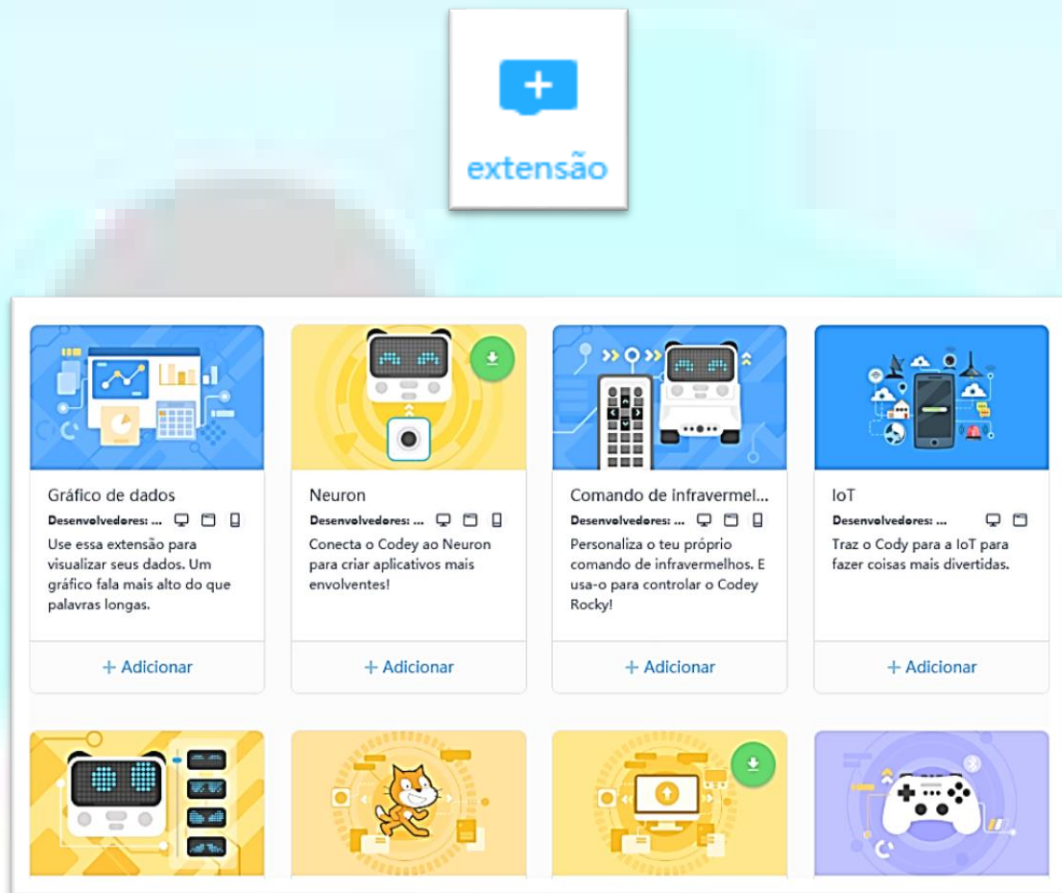




Fonte: Elaborado pelo autor, 2020

Neste item existe também, no final da coluna dos círculos, a opção **+EXTENSÃO** que amplia as funções do mBlock com outros dispositivos compatíveis e codificáveis por meio de blocos visuais e/ou a linguagem *Python*. Podemos citar: **Arduino Uno, Arduino Mega, Arduino Leonardo, Arduino Nano, Arduino Micro, OrangeBoard, Neuron, Codey, mBot, mBot Ranger, microbit, Bluetooth Controller, MotionBlock, HaleCode, Raspberry Pi, Real STEAM** e outros. Alguns exemplos de extensões disponíveis na plataforma mBlock estão ilustradas na figura 17:

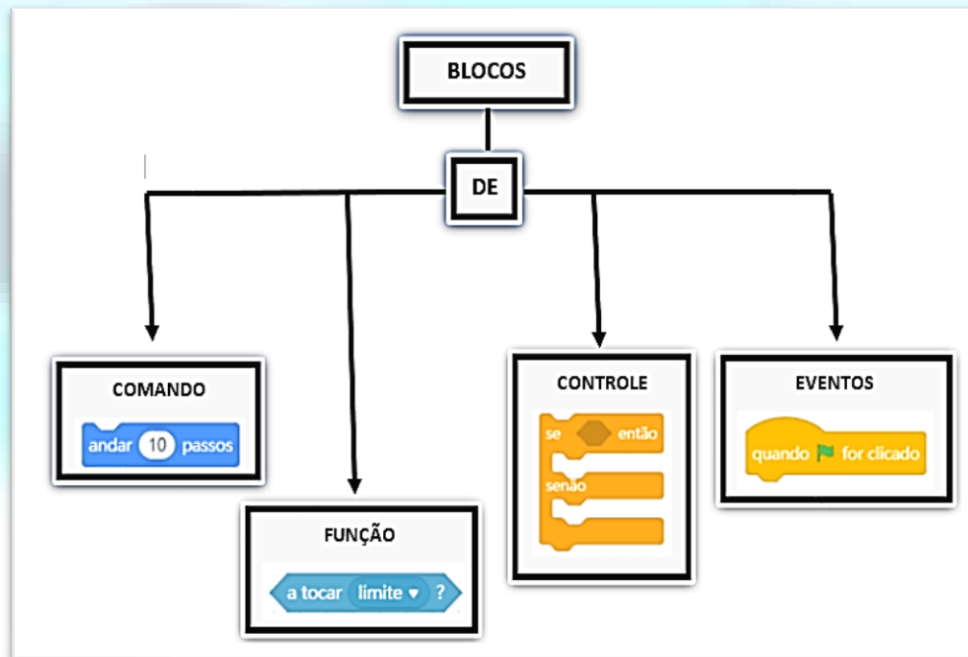
Figura 17 - Extensões Disponíveis na Plataforma mBlock



Fonte: Elaborado pelo autor, 2020

Ainda na **Paleta de Blocos**, destaca-se que além da organização por categorias e cores distintas dos blocos de comando, o mBlock os classifica em quatro tipos gerais como mostrado na figura 18 a seguir:

Figura 18 - Classificação dos Blocos de Comando



Fonte: Elaborado pelo autor, 2020

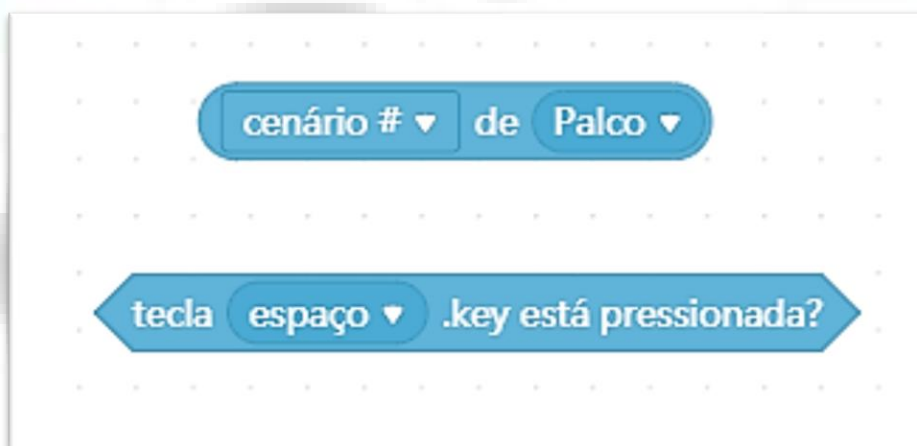
Os **Blocos de Controle** e os **Blocos de Comando** possuem em seu design uma pequena reentrância na parte superior do bloco e uma saliência na parte inferior permitindo ser encaixados no formato de pilhas. Os **Blocos de Eventos** são permitidos apenas no início de um **SCRIPT**¹⁹. Por sua vez, os **Blocos de Função** não possuem saliências e retornam algum valor. Estes blocos são usados como entradas para outros blocos e não podem ser utilizados de forma independente para compor um **Script** do mBlock. Os **Blocos de Função** possuem duas características básicas que podemos destacar:

¹⁹ Em Ciência da Computação a Linguagem de *Script* é uma linguagem interpretada que age de dentro de um programa, como forma de extensão. Disponível em: https://pt.wikipedia.org/wiki/Linguagem_de_script. Acesso em: 22 de mar. de 2020.

- **Blocos** que possuem suas extremidades arredondadas podem retornar números ou **Strings**;
- **Blocos** que possuem as extremidades pontiagudas informam se alguma sentença ou expressão é verdadeira ou falsa, ou seja, retornam valores **Booleanos**.

A figura 19 exemplifica as características dos **Blocos de Função** encontrados na plataforma mBlock:

Figura 19 - Característica dos Blocos de Função



Fonte: Elaborado pelo autor, 2020

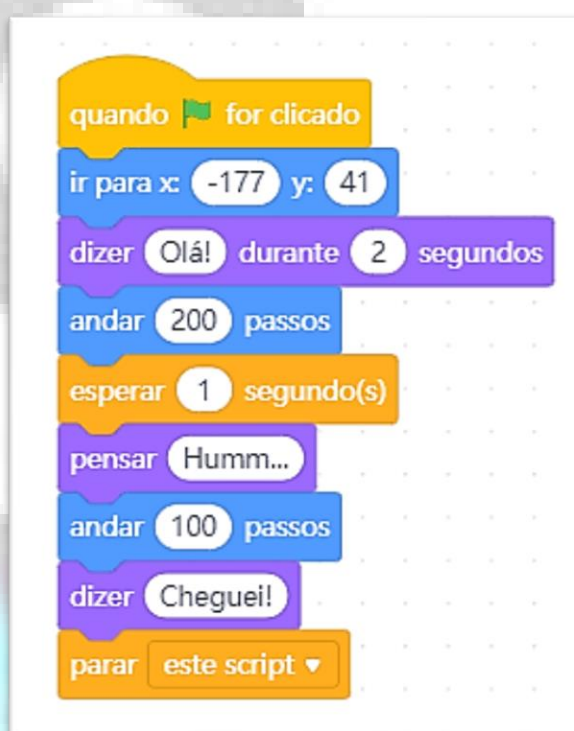


JANELA DE CODIFICAÇÃO OU SCRIPT ZONE: Configura-se como o espaço onde a codificação é construída, ou seja, a programação é feita por meio de **Scripts**. Chamam-se **Scripts** as rotinas sequenciais de instruções escritas para um sistema que automatiza a execução de determinadas tarefas. Na Plataforma mBlock, esses **Scripts** são vistos como o **encaixe lógico-sequencial de blocos visuais**.

Esses fragmentos de código, ou seja, os **Scripts**, são montados por meio dos blocos de cor específica e em seguida testados, para posteriormente serem combinados em unidades de blocos maiores. Ao codificar usando o mBlock, as mensagens de erro não são prontamente visíveis ao usuário. Essa característica do mBlock permite realizar a alteração e a substituição de blocos específicos ou de conjuntos de blocos no decorrer do processo de codificação **incentivando à aprendizagem do tipo “mãos-à-obra”**.

Na figura 20, ilustra-se o exemplo de um **Script** usando a **Codificação por Blocos** do mBlock. A partir para a solução de um dado problema, um **Script** longo pode ser fracionado em **Scripts** menores, ou seja, pode ser dividido em um conjunto de blocos menores, então, cada bloco pode ser testado de forma independente.

Figura 20 - Conjuntos de Blocos formando Comandos

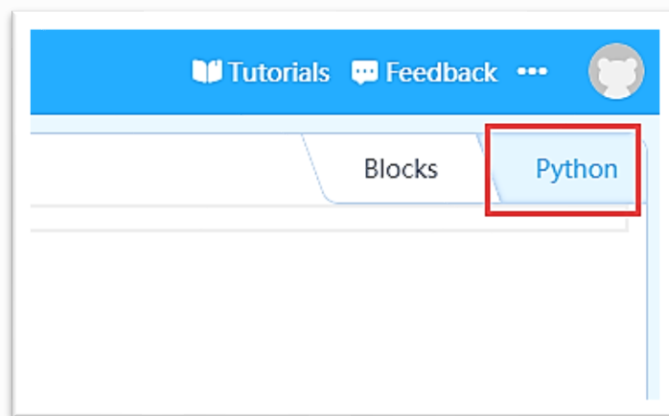


Fonte: Elaborado pelo autor, 2020



O AMBIENTE DE PROGRAMAÇÃO *PYTHON*: O mBlock oferece um ambiente auxiliar de programação na linguagem *Python*. Existe a possibilidade em alternar o modo de programação de "**Blocos para *Python***" com apenas um clique na aba ilustrada na figura 21:

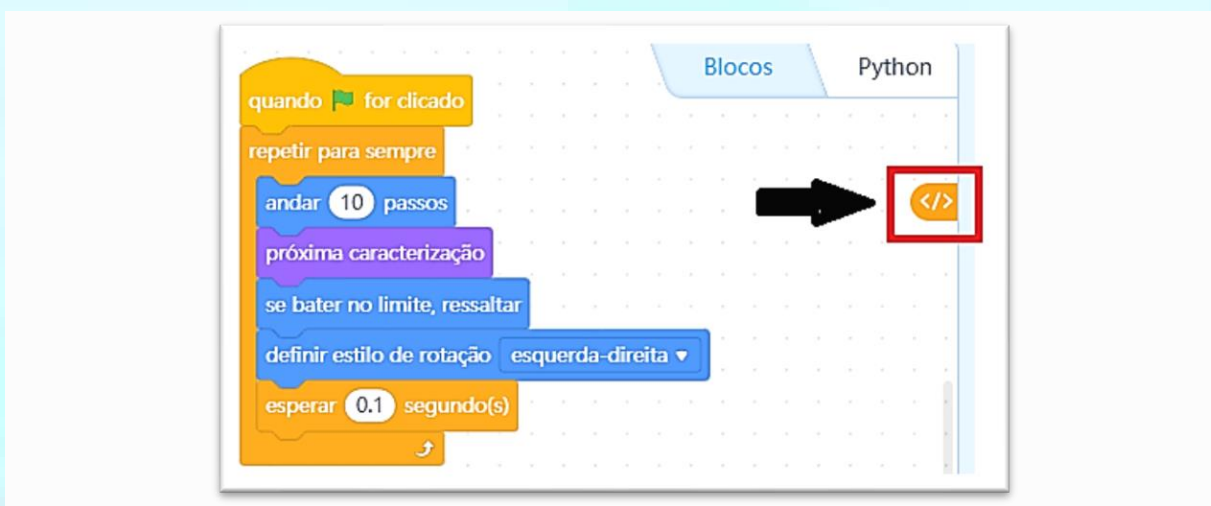
Figura 21 - Aba de Acesso à Linguagem *Python*



Fonte: Elaborado pelo autor, 2020

Existe também a possibilidade em transpor uma estrutura de blocos superpostos para a linguagem *Python*, simplesmente clicando no ícone à direita da área de *Scripts*, como ilustrado na figura 22:

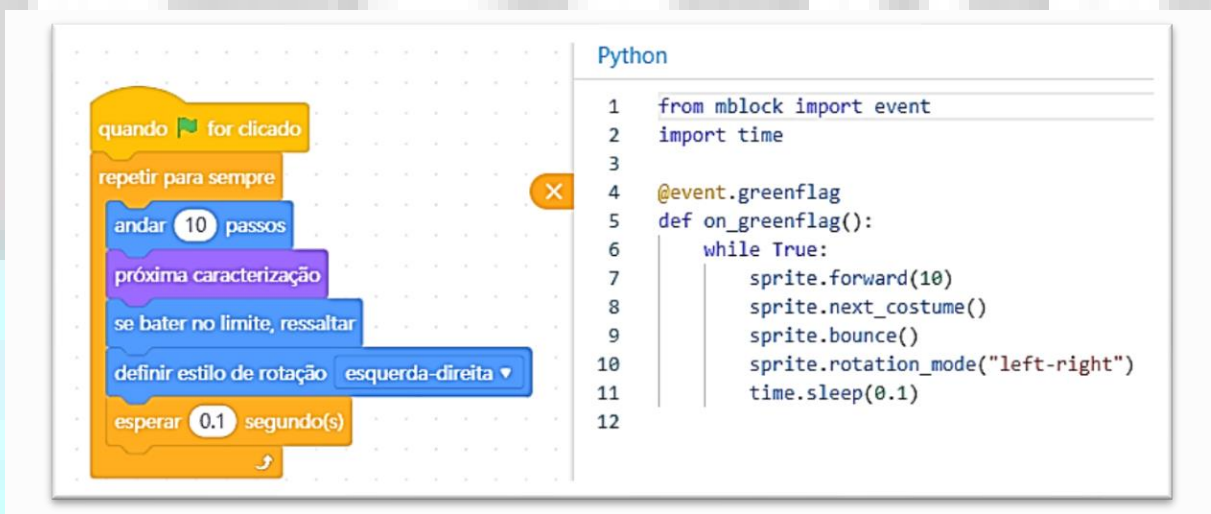
Figura 22 - Transposição de Blocos para código *Python*



Fonte: Elaborado pelo autor, 2020

O simples ato de “**clicar**” no ícone correspondente resulta na conversão completa e imediata do código que estava no formato de blocos para a forma estruturada e sintática presente na linguagem ***Python***. Observe na figura 23, um conjunto de Blocos convertidos em **Código *Python***:

Figura 23 - Blocos de Comando Convertidos em Código *Python*



Fonte: Elaborado pelo autor, 2020

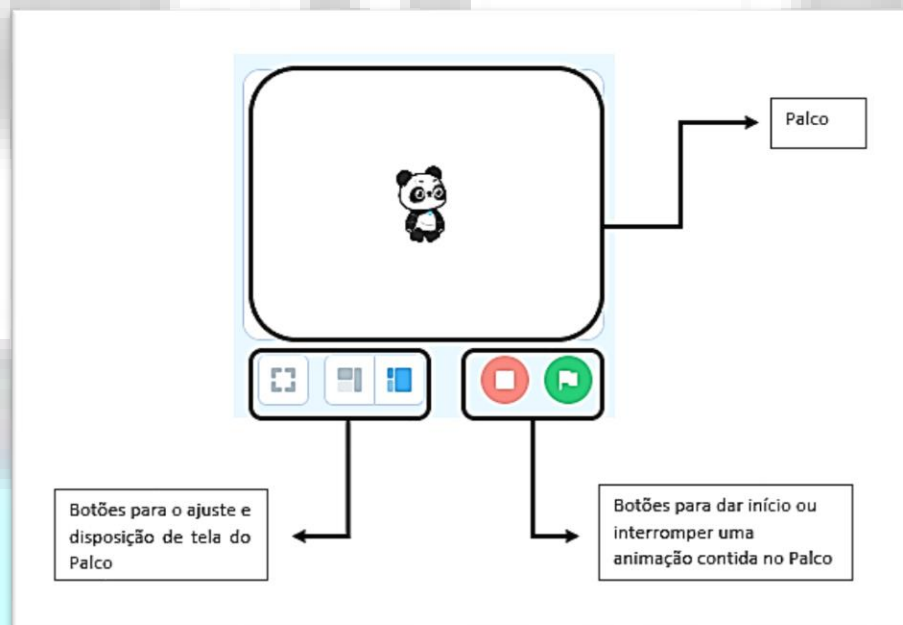
Este procedimento é válido para a maioria das funções e projetos que envolvam atores, palcos e dispositivos como robôs e placas eletrônica que venham a ser

acoplados e utilizados com a plataforma de codificação mBlock. Neste projeto, não faremos uso desse recurso.



O PALCO: É utilizado para realizar desenhos e colocar imagens em movimento. Essas imagens são denominadas atores ou **Sprites**. É o ambiente em que os atores são desenhados, se movem e realizam interações. O palco inicia com o fundo branco e com um único ator, **o urso Panda, a mascote símbolo do mBlock**. Na parte inferior do palco existem cinco botões, dos quais, três à esquerda servem para modificar a aparência e disposição do palco e os outros dois botões à direita servem para iniciar ou interromper uma animação, observe um **exemplo de Palco** ilustrado na figura 24:

Figura 24 - Palco com o Ator Mascote da Plataforma mBlock: O Panda



Fonte: Elaborado pelo autor, 2020



PAINEL DE DISPOSITIVOS, ATORES E FUNDO: Nesse ambiente temos a possibilidade de selecionar entre três abas. Na primeira aba da esquerda para a direita, temos acesso via cabo **USB²⁰** a dispositivos e acessórios que podem ou não ser incorporados ao projeto, como por exemplo: **Codey, Arduino, mBot, Raspberry Pi** e outros. O Painel contendo os **Dispositivos, Atores e Fundo de Palco** está exemplificado na figura 25:

Figura 25 - Painel de Dispositivos, Atores e Fundo de Palco

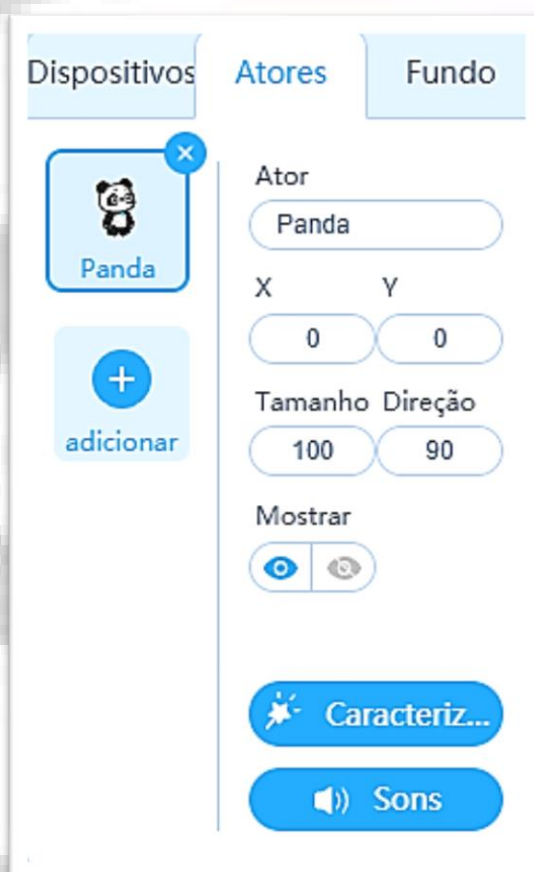


Fonte: Elaborado pelo autor, 2020

²⁰ USB (abreviatura de Universal Serial Bus, em português, porta serial universal) é um padrão de industrial que estabelece especificações para cabos, conectores, e protocolos de comunicação para conexão, comunicação e provimento de energia entre computadores pessoais e seus dispositivos periféricos. Disponível em: <https://pt.wikipedia.org/wiki/USB>. Acesso em: 28 de abr. de 2020.

Na segunda aba, o ator é definido mediante a escolha de **Sprites**²¹ distribuídas numa biblioteca do mBlock e onde são mostradas miniaturas de todos os atores do projeto, seus respectivos nomes, categorias, fantasias e demais informações. A escolha de **Sprites** para os atores é exemplificada na sequência conforme ilustram a figura 26 e na sequência a figura 27:

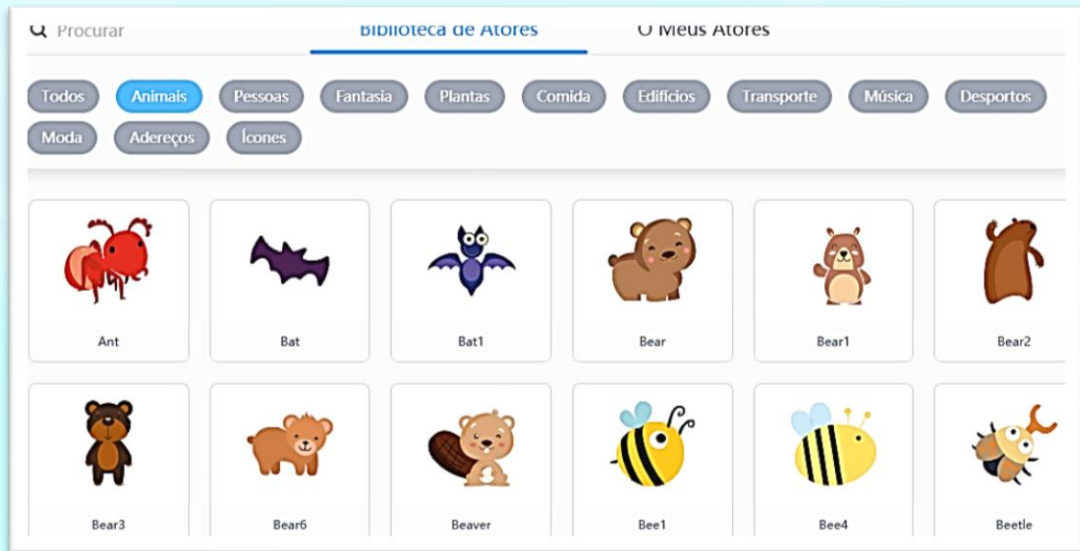
Figura 26 - Sprite do ator Panda



Fonte: Elaborado pelo autor, 2020

²¹ Em computação gráfica, um *Sprite* é um objeto gráfico bidimensional ou tridimensional que se move numa tela sem deixar traços de sua passagem. Disponível em: [https://pt.wikipedia.org/wiki/Sprite_\(computa%C3%A7%C3%A3o_gr%C3%A1fica\)](https://pt.wikipedia.org/wiki/Sprite_(computa%C3%A7%C3%A3o_gr%C3%A1fica)). Acesso em: 25 de jun. de 2020.

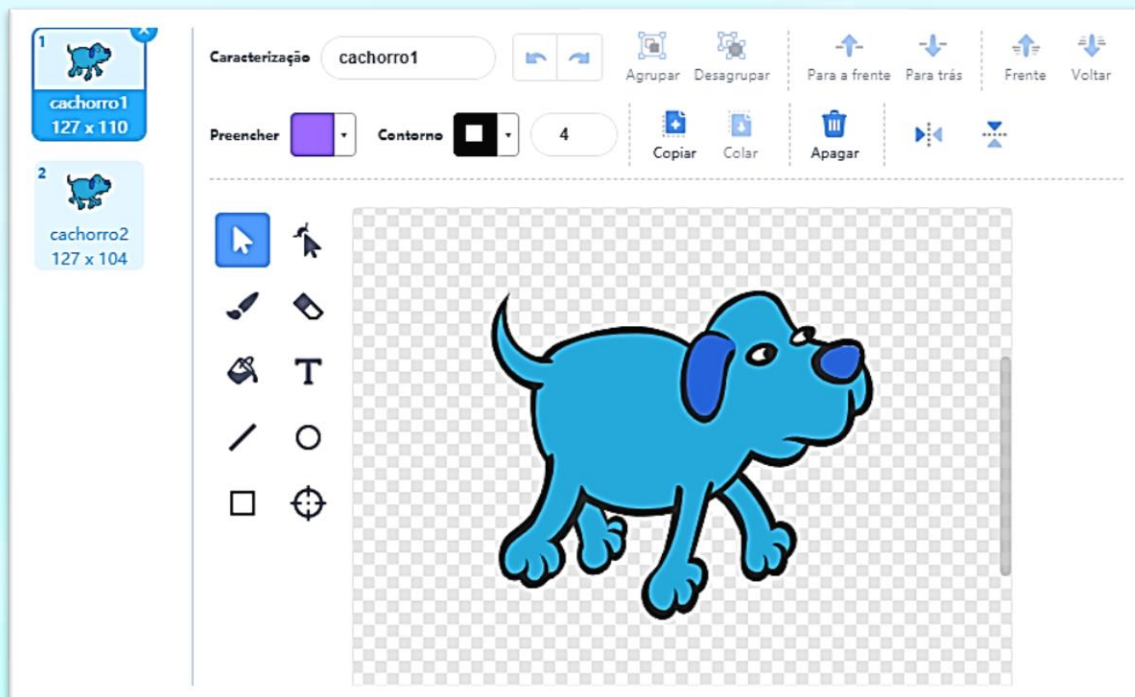
Figura 27 - Biblioteca de Sprites dos Atores



Fonte: Elaborado pelo autor, 2020

Existe também a possibilidade de o ator ser criado pelo próprio usuário no **Editor de Caracterização** onde o usuário pode conferir e manipular os trajes (costumes) do ator. Muitos atores (ou *Sprites*) são disponibilizados com diferentes trajes. Esse recurso é muito utilizado quando se quer dar a impressão de movimento no ator, por exemplo: caminhar, saltar, correr, bater as asas, etc. **O Editor de Caracterização de Atores** da plataforma mBlock está exemplificado segundo ilustra a figura 28:

Figura 28 - Editor de Caracterização de Atores

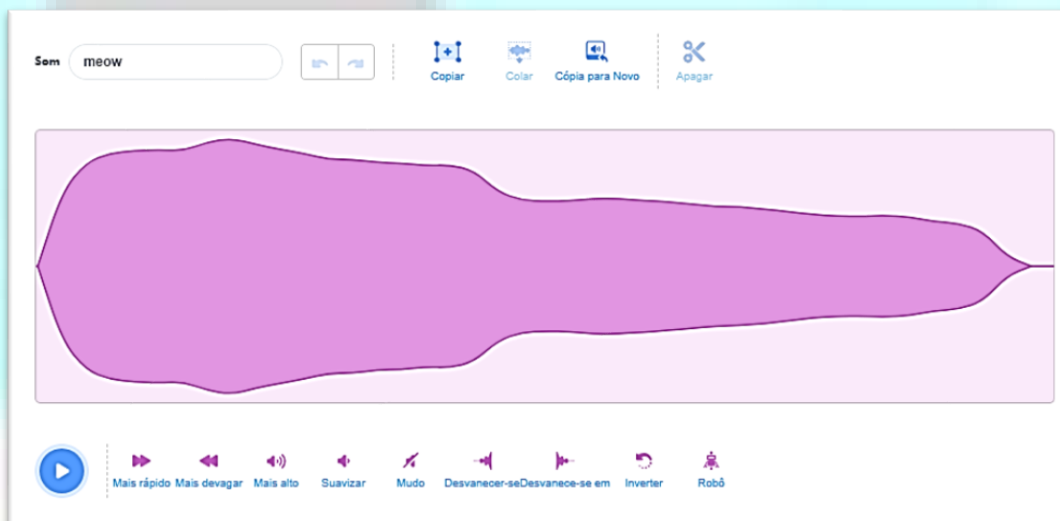


Fonte: Elaborado pelo autor, 2020

Na mesma aba, podemos incluir sons da biblioteca do mBlock ou criar nossa própria trilha sonora ou os mais variados efeitos sonoros que poderão ser utilizados com os atores e cenários nos projetos compilados no mBlock.

A **Biblioteca de Sons** da Plataforma mBlock está exemplificada conforme ilustra a figura 29:

Figura 29 - Biblioteca de Sons



Fonte: Elaborado pelo autor, 2020

Na terceira e última aba, temos a opção de criar ou manipular o **Backdrop**, que pode ser entendido como o **Palco (Pano de Fundo)** em Português, onde os atores irão se atuar e interagir. Nessa aba também existe um **Editor de Caracterização de Palco (Backdrop)**, estilo o **Editor Microsoft Paint**²² e o mesmo Editor de Sons da aba anterior.

Esse editor de imagens serve para que o usuário libere a sua criatividade na elaboração de lindos cenários para os seus mais diversos projetos e tenha a possibilidade de manipular a imagem no formato **Bitmap**²³ ou no formato de **Imagem Vetorial**²⁴.

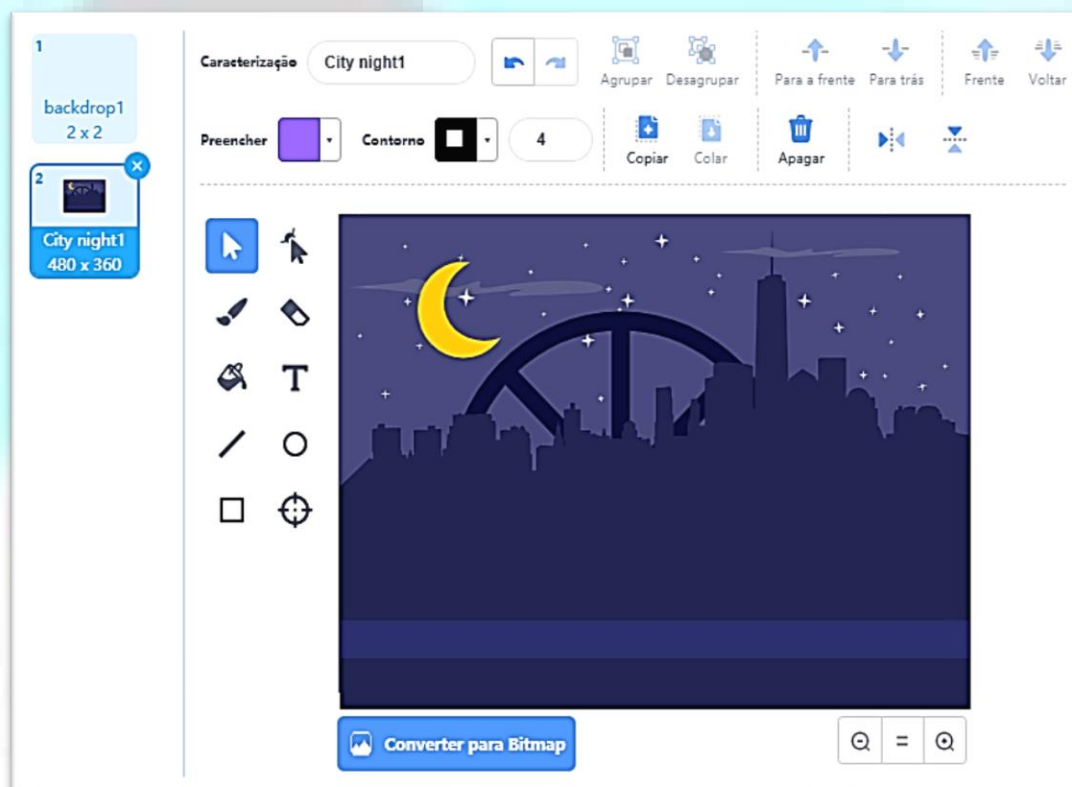
²² *Microsoft Paint* é um software utilizado para a criação de desenhos simples e também para a edição de imagens. O programa é incluso, como um acessório, no sistema operacional *Windows* da *Microsoft*.

²³ *Bitmap*, do inglês mapa de *bits*, em computação gráfica são imagens que contêm a descrição de cada pixel em um desenho. Disponível em: https://pt.wikipedia.org/wiki/Microsoft_Paint. Acesso em: 03 de fev. de 2021.

²⁴ Imagem vetorial em computação gráfica é o uso de primitivas geométricas como pontos, linhas, curvas e formas ou polígonos - todos os quais são baseados em expressões matemáticas para representar imagens. Disponível em: https://pt.wikipedia.org/wiki/Desenho_vetorial. Acesso em: 03 de fev. de 2021.

O usuário também tem a sua disposição uma biblioteca contendo cenários separados em diversas categorias prontos para serem usados nos projetos realizados na plataforma mBlock. O **Editor de Caracterização de Palco (*Backdrop*) ou Imagem**, está ilustrado na figura 30 abaixo:

Figura 30 - Editor de Caracterização de Palco (*Backdrop*)



Fonte: Elaborado pelo autor, 2020



FERRAMENTA DE AMPLIAÇÃO OU REDUÇÃO: Como o próprio nome diz, serve para ampliar ou reduzir os blocos superpostos colocados na **Janela de Codificação ou *Script Zone***. Diante de todos os recursos presentes na plataforma de programação visual mBlock, podemos concluir que o mesmo não transforma o usuário em uma espécie de programador profissional, mas permite por meio da lógica da programação que os **usuários possam se expressar de forma criativa e independente**.

Desde a época em que **Seymour Papert** idealizou as crianças como programadoras desenvolvendo a **linguagem de programação** para crianças **LOGO**, bem como Mitchel Resnick²⁵ quando imaginou o SCRATCH como um software de programação que fosse acessível a todas às pessoas, possuir a liberdade em criar e de saber fazer por si só, torna as pessoas mais ativas e conscientes e as próprias produtoras de mídias ao invés de serem apenas suas consumidoras.

Nesse contexto, e por ser inspirado no Scratch, o mBlock caracteriza-se por possuir fortes características construcionistas da Linguagem LOGO de Papert, acredita-se que ele se apresente como um recurso pedagógico digital válido para o desenvolvimento da autonomia tecnológica e a proficiência digital de professores e alunos.

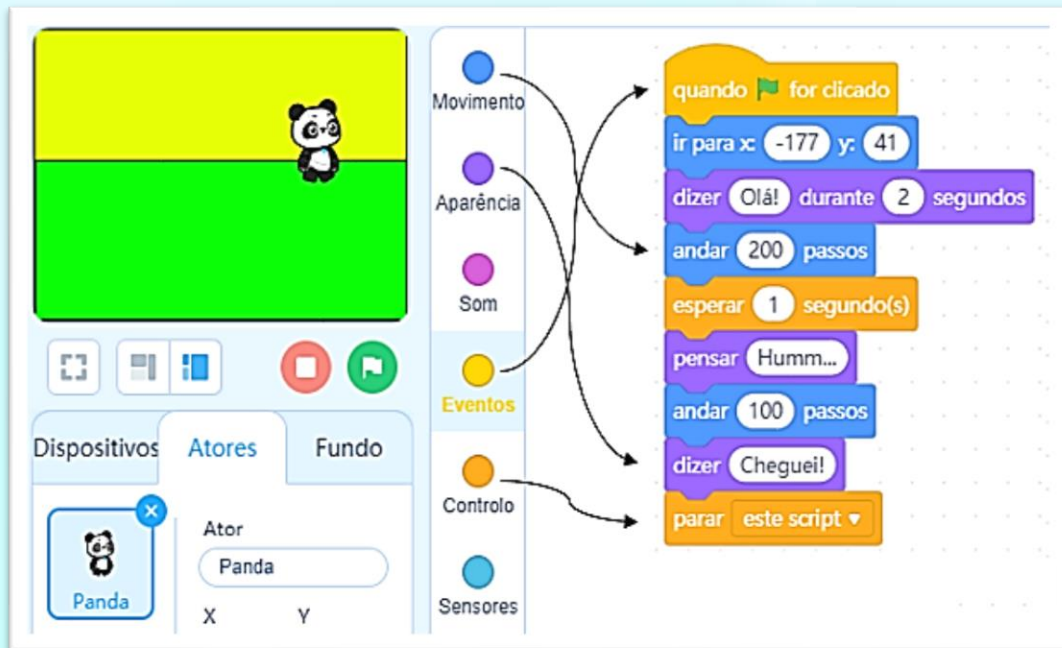
5.1 *Scripts* em Blocos de Arrastar e Soltar

Scripts são os comandos fornecidos pelo usuário, de forma lógica e estruturada que atendam determinadas funções e objetivos. Eles são construídos por meio do encaixe de blocos visuais coloridos. Os blocos visuais devem ser arrastados e soltos na janela de **Scripts**. Essa ação denomina-se **Drag-and-Drop**, ou seja, **Arrastar e Soltar**.

A figura 31 apresenta a construção de um bloco de **Script** que faz com que o mascote da plataforma mBlock, o Panda, diga frases curtas e desloque-se pelo palco:

²⁵ Disponível em: <https://novaescola.org.br/conteudo/905/mitchel-resnick-a-tecnologia-deve-levar-o-aluno-a-ser-um-pensador-criativo>. Acesso em: 09 de ago. de 2021.

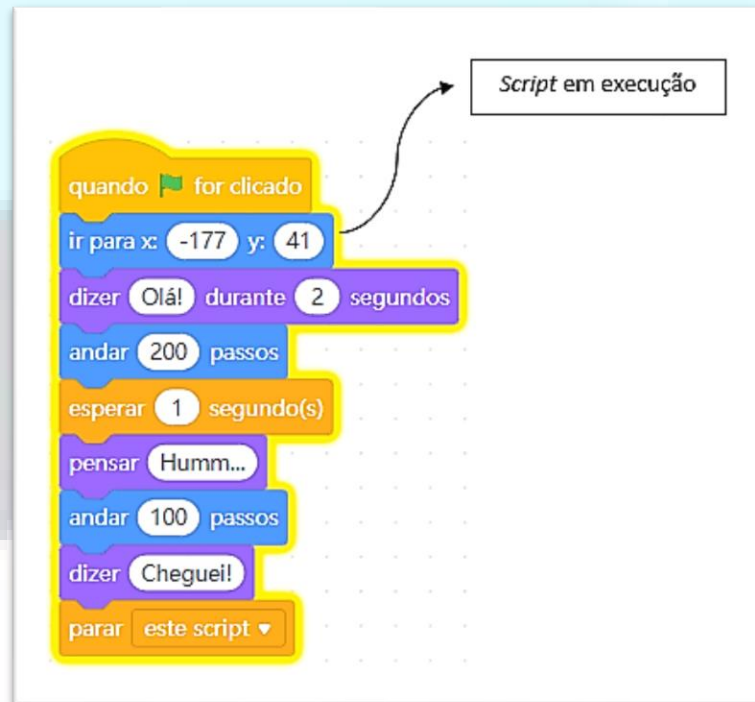
Figura 31 - Exemplo de Bloco de *Script*



Fonte: Elaborado pelo autor, 2020

Para que o ator, no caso o Panda, execute os comandos de dizer algumas frases e deslocar-se pelo palco, deve-se clicar sobre o **Script** na **Janela de Scripts** ou clicar na bandeira verde na parte inferior direita do palco. Todos os blocos de comando ficarão com uma linha amarela brilhante indicando que o **Script** está em execução, caso contrário, o comando não está sendo executado. Esta ação está ilustrada na figura 32:

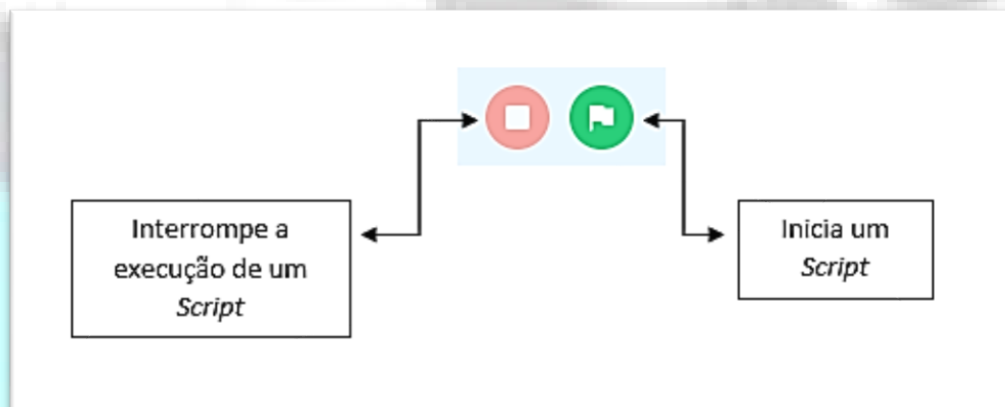
Figura 32 - Exemplo de um Bloco de *Script* em Execução



Fonte: Elaborado pelo autor, 2020

Para interromper o **Script**, basta clicar no quadrado branco inserido em um círculo vermelho ao lado esquerdo da bandeira verde como ilustrado na figura 33 a seguir:

Figura 33 - Botões para Iniciar ou Interromper um *Script* em Execução



Fonte: Elaborado pelo autor, 2020

A plataforma mBlock oferece um **processo de depuração** que é realizado durante a construção da programação. As correções dos erros podem ser realizadas sem a

necessidade de interromper o **Script**. Dessa forma, há uma promoção do retorno imediato e a possibilidade de o usuário repensar sobre os seus erros ou sobre a estrutura do programa. Essa característica difere o mBlock das demais linguagens de programação usuais em que o retorno do comando só é obtido após findar-se a sua execução.

5.2 Projeto Exemplo de OA Digital para Computadores e Notebooks: Simulador de Queda Livre

Este projeto exemplo de Objeto de Aprendizagem (OA) digital simula a sequência de instantes em que uma bola abandonada cai em queda livre. Nele poderão ser explorados conceitos físicos, matemáticos, computacionais dentre outros. O código de blocos poderá ser modificado, reutilizado ou adaptado para a realidade das disciplinas ou conteúdos conforme o professor considerar necessário. Este tipo de Objeto de Aprendizagem (OA) digital envolve o desenvolvimento de habilidades e da competência Cultura Digital encontradas na BNCC (2017), bem como, diversos aspectos pedagógicos inerentes as atividades utilizando um Software Educacional além de delinear o modelo de conhecimento CTPC.

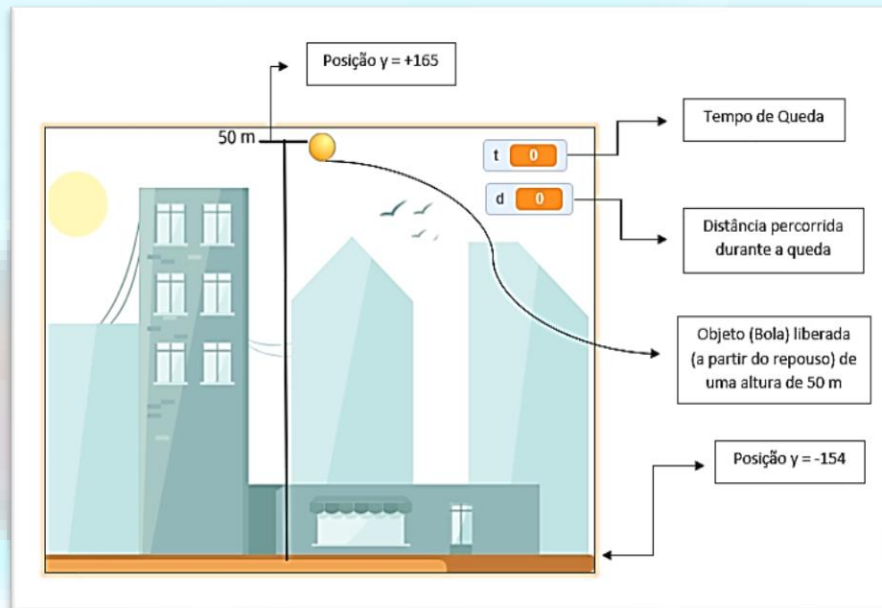
- **Queda Livre**

Ignorando os efeitos da resistência do ar, quando um objeto, a partir do repouso é solto de uma determinada altura, a distância **d (em metros)** percorrida pelo objeto durante um certo intervalo de tempo **t (em segundos)** é dada pela fórmula:

$$d = (0.5 \cdot g \cdot t^2)$$

A aceleração da gravidade terrestre é de aproximadamente **9,81 m/s²**. O objetivo dessa simulação é mostrar a posição de uma bola em queda livre nos instantes **0,5 s, 1,0 s, 1,5 s, 2,0 s, 2,5 s** e assim sucessivamente, até atingir o solo. O Palco desta simulação foi elaborado conforme ilustra a figura 34:

Figura 34 - Projeto Simulador de Queda Livre



Fonte: Adaptado de (MARJI, 2014)

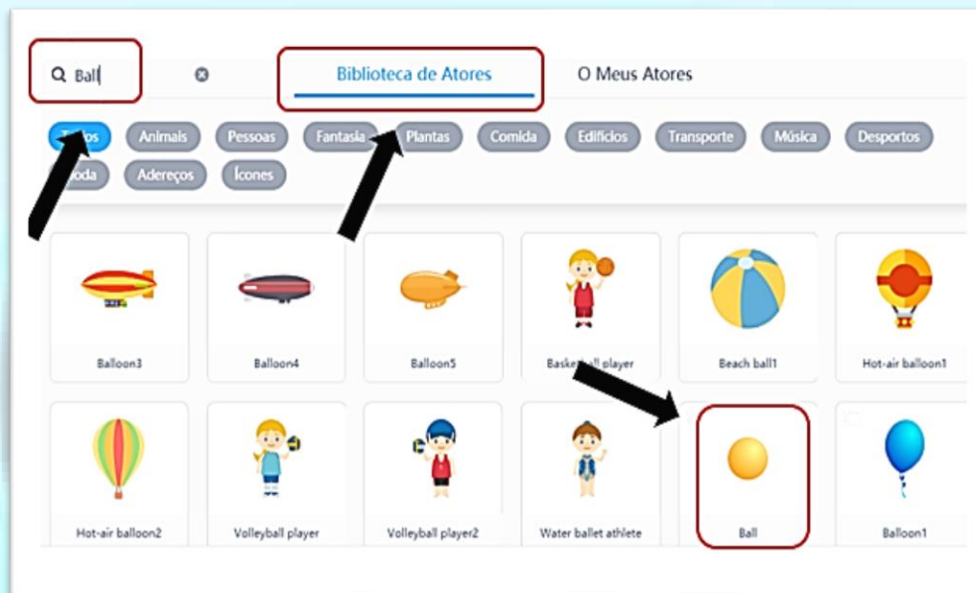
O objeto em repouso (a bola da figura 48) cairá de uma altura de **50 metros**. Uma substituição simples na fórmula anterior mostra que o objeto atingirá o solo após aproximadamente:

$$t = \frac{2 \cdot 50}{9,81 \cdot 3,20}$$

$$t = 3,18 \text{ s}$$

O **projeto**, contém um **Ator** ou **Sprite (chamado Ball)** com apenas uma fantasia que pode ser facilmente encontrada na biblioteca de atores (**Sprites**) conforme foi descrito com detalhes anteriormente. A figura 35 reforça a localização da **Sprite Ball (Bola)** para ser utilizada no projeto de simulação de Queda Livre a partir da **Biblioteca de Atores** pertencentes ao mBlock:

Figura 35 - A Localização do *Sprite Ball* na Biblioteca de Atores



Fonte: Elaborado pelo autor, 2020

No momento de mostrar a posição da bola em queda, o **Sprite Ball (Bola)** fará um “carimbo” da sua própria fantasia durante a queda em alguns trechos d do percurso. A simulação é iniciada quando a bandeira verde é clicada. Em resposta, o **Sprite Ball** executa o **Script** que está sendo ilustrado na figura 36:

Figura 36 - Script para o ator Bola no Projeto Queda Livre

```

quando for clicado
  Inicializar
  repetir para sempre
    esperar 0.3 segundo(s)
    alterar contador por 1
    alterar t por 0.4
    definir d para 4.9 * t * t
  se não d < 50 então
    definir Y para -154
    pensar O tempo de queda é de 3.2 s! durante 1 segundos
    parar este script
  senão
    definir Y para 165 - 320 * d / 50
  se valor absoluto (abs) de contador = 10 = 0 então
    Marca
  Marca
  
```

```

definir Inicializar
  definir t para 0
  definir d para 0
  definir contador para 0
  eliminar tudo
  ir para x: 3 y: 165
  mudar de caracterização para ball-a
  
```

```

definir Marca
  mudar de caracterização para ball-b
  carimbar
  mudar de caracterização para ball-a
  
```

Fonte: Elaborado pelo autor, 2020

1

Durante a inicialização, o **Sprite** se move para a sua posição inicial, muda para a fantasia **Ball (Bola)**, limpa o balão de voz da última execução e remove qualquer marca anterior do palco. Então **t** e o **contador** são inicializados com 0. A **variável t** representa a duração da queda e a variável **contador** controla o número de repetições do laço.

2

O **Script** então entra em um laço infinito para calcular os parâmetros da simulação em diferentes intervalos de tempo. Esses cálculos são feitos e a posição

3

da bola é atualizada a cada **0,4 (s) segundos** para garantir um movimento suave da bola. A cada **0,4 (s) segundos**, o valor da **variável t** referente ao tempo é atualizado e a distância percorrida pela bola na queda (**d**) é calculada. O valor da variável contador também é **incrementado de 1**. Se a bola atingir o solo (o que acontece quando **(d = 50 m)**, o **Script** definirá a **posição y (final)** da bola para a posição do solo, exibirá a duração do trajeto e interromperá o **Script** para finalizar a

4

simulação. Do contrário, o **Script** definirá a posição vertical da bola de acordo

5

com a distância percorrida na queda.

Como a altura de **50 metros** corresponde a **324 pixels** no palco, uma altura de d metros podem ser facilmente determinada pela expressão $324 \cdot (d / 50)$. A posição **y (final)** é determinada ao subtrair esse número da posição **y₀ (inicial)** que é igual a **170 pixels**. O **Sprite Ball** fará um carimbo da sua fantasia pela trajetória enquanto cai

6

em queda livre para mostrar a posição da bola em queda em alguns instantes.

A figura 37 ilustra o resultado da execução desta simulação. Observe como a distância percorrida na queda a cada intervalo de tempo aumenta à medida que a bola cai. Por causa da gravidade, a bola sofre uma aceleração, ou seja, a sua velocidade aumenta a uma **taxa de 9,81 m/s por segundo**.

Figura 37 - Resultado Final da Simulação de Queda Livre



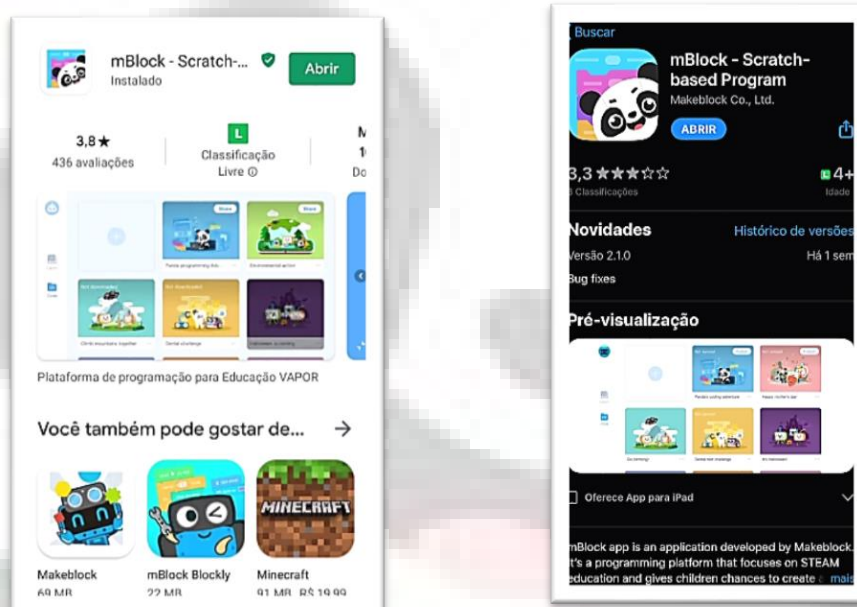
Fonte: Adaptado de (MARJI, 2014)

A construção dessa simulação física para o assunto Queda Livre, ilustrou a concepção de um projeto na plataforma mBlock para computadores e notebooks. Deve-se salientar neste momento que não é a construção em si que torna esse projeto um Objeto de Aprendizagem (OA) digital em si mesmo, mas a sua contextualização em um ambiente de aprendizagem escolar é o que o transformará em um rico recurso educativo o qual o professor poderá explorar de diversas maneiras com os seus alunos.

7. O MBLOCK PARA DISPOSITIVOS MÓVEIS

Além de ser disponibilizado como software para diversos sistemas operacionais computacionais como por exemplo, **Windows 32/64 bits** e **OS X 32/64 bits**, o mBlock também está disponível como aplicativo para **dispositivos móveis**. O mesmo pode ser baixado na **Google Play** para *smartphones* e *tablets* com a plataforma mobile **Android**. O mBlock também está disponível na **Apple Store** para dispositivos móveis embarcados na plataforma mobile **iOS**. A figura 38 ilustra estas duas opções:

Figura 38 - Opções da Plataforma mBlock para Dispositivos Móveis

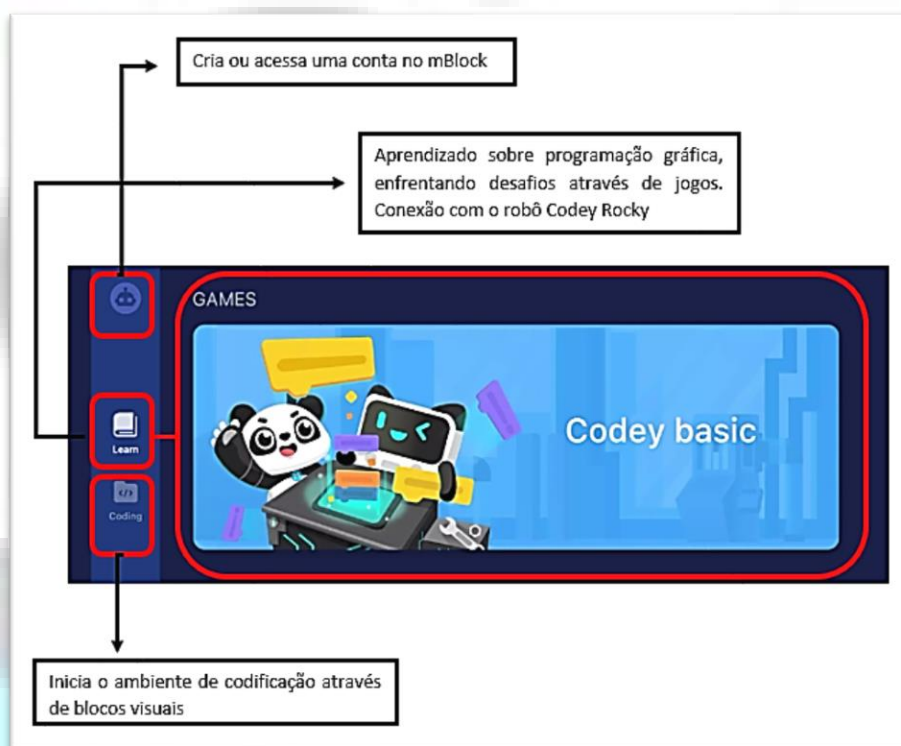


Fonte: Elaborado pelo autor, 2020

A disponibilidade da plataforma mBlock para dispositivos móveis se faz útil, pois, professores e alunos mesmo sem computador disponível podem ter acesso a plataforma de programação e praticamente usufruir das mesmas ferramentas da versão para computador. Nesta pesquisa se fez uso das duas versões, para computador e para dispositivos móveis. O que foi relatado acima, praticamente orientou a escolha pelo mBlock.

A versão desta plataforma para dispositivos móveis somente disponibiliza o aplicativo na língua inglesa. Observa-se na figura 39, que a tela inicial da plataforma mBlock na sua versão para dispositivos móveis, equipados na plataforma mobile **Android**, é bem simples, intuitiva e está disposta da seguinte forma:

Figura 39 - Plataforma mBlock para Dispositivos Móveis com *Android*

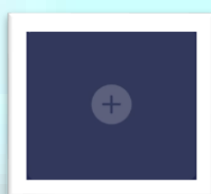


Fonte: Elaborado pelo autor, 2020

Após clicar no ícone **Coding (Código)** para iniciar a codificação, caso seja o primeiro projeto, o usuário verá uma tela pedindo para que o mesmo inicie o projeto imediatamente (**start immediately, create your first work**) escrito abaixo do símbolo de **“mais”**. Essa mensagem não aparecerá mais depois dessa etapa apenas o

símbolo de “mais” para quando o usuário for criar um novo projeto. A figura 40, ilustra o botão para iniciar um novo projeto na plataforma mBlock para dispositivos móveis:

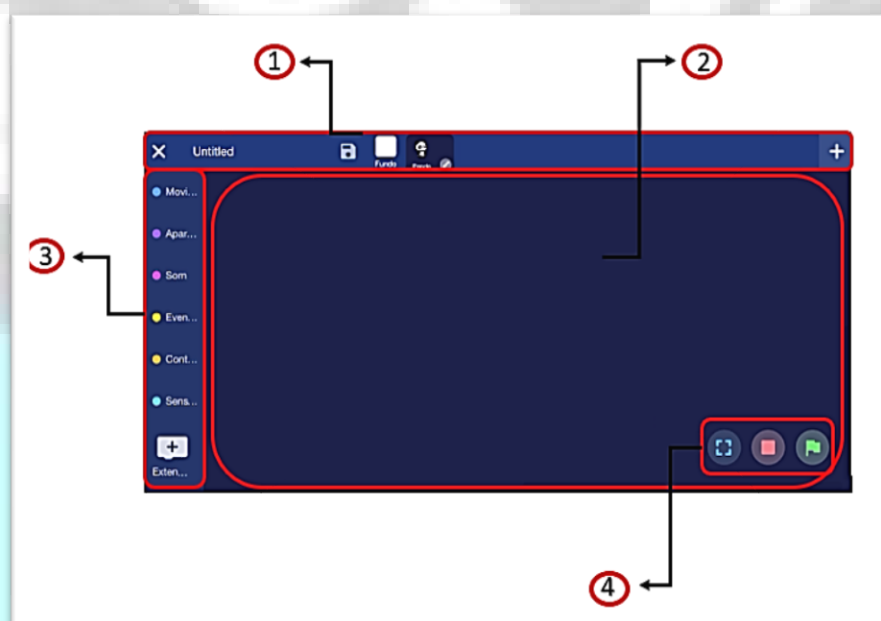
Figura 40 - Botão para iniciar um Novo Projeto



Fonte: Elaborado pelo autor, 2020

Depois de iniciar o primeiro projeto, temos acesso ao ambiente onde todos os **Scripts** serão desenvolvidos por meio do encaixe sucessivo dos blocos visuais. Esse procedimento é muito similar ao descrito anteriormente para a versão do mBlock para computadores. A figura 41, ilustra os principais recursos que estão disponíveis no ambiente de codificação do mBlock para dispositivos móveis.

Figura 41 - Recursos Disponíveis no mBlock para Dispositivos Móveis

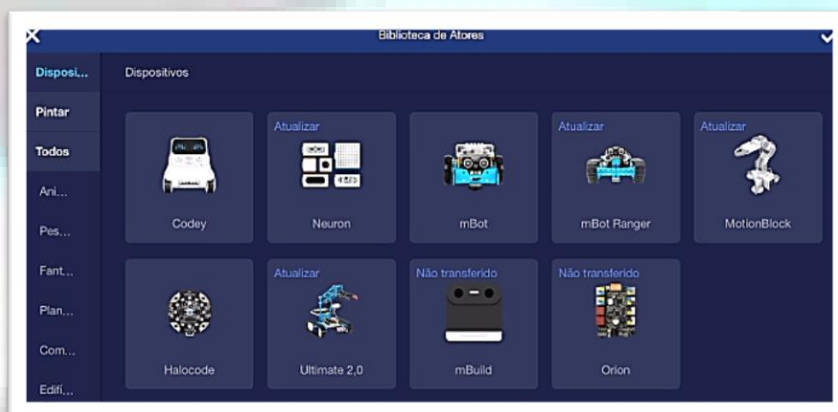


Fonte: Elaborado pelo autor, 2020

1

BARRA DE MENUS: Aqui o usuário pode criar, salvar, nomear ou renomear um projeto. Clicando no sinal de “mais” à direita da tela, o usuário pode acessar a **Biblioteca de Atores** e de dispositivos compatíveis com o mBlock, conforme vemos na figura 42:

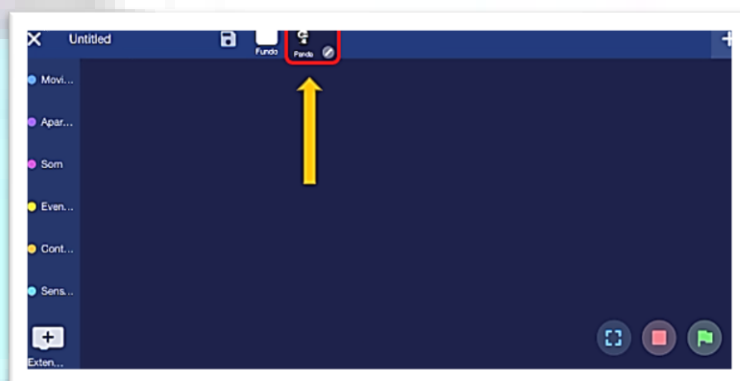
Figura 42 - Biblioteca de Atores mBlock



Fonte: Elaborado pelo autor, 2020

Também na barra de menus, temos acesso ao **Ambiente de Configuração** do(s) **Ator(res) (Sprites)** clicando no desenho de um pequeno lápis que fica à frente do **ursinho Panda** como é ilustrado na figura 43:

Figura 43 - Acesso ao Ambiente de Configuração dos Atores



Fonte: Elaborado pelo autor, 2020

Nesse ambiente é possível editar os **parâmetros**, a **caracterização** ou **trajes (costumes)** e também os **sons** que irão compor ou já compõe a cena ou o projeto todo. Observe o que ilustram a sequência das figuras 44, 45 e 46:

Figura 44 - Acesso ao Editor de Parâmetros



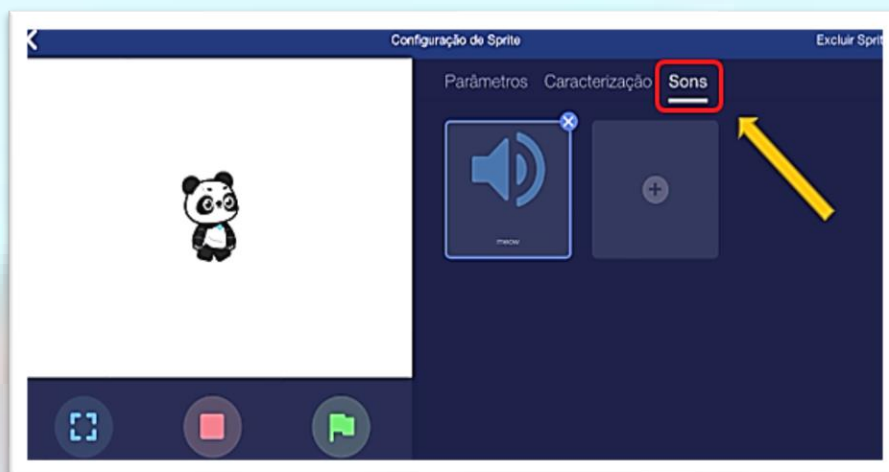
Fonte: Elaborado pelo autor, 2020

Figura 45 - Acesso ao Editor de Caracterização



Fonte: Elaborado pelo autor, 2020

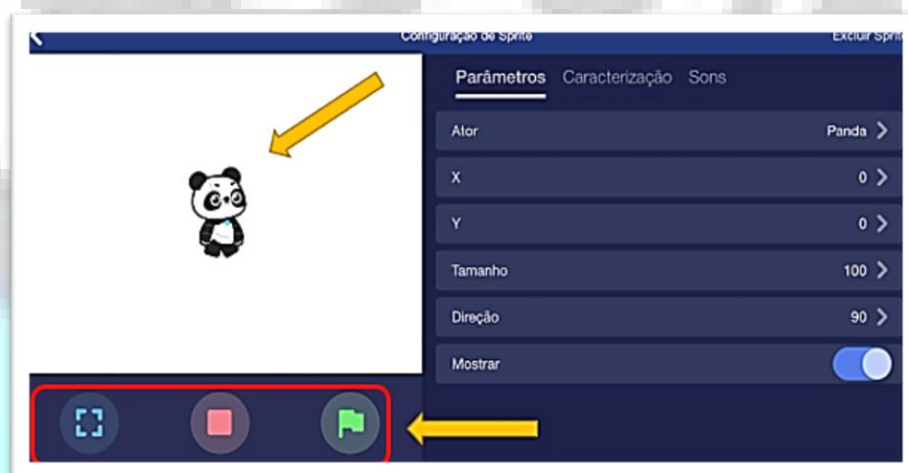
Figura 46 - Acesso ao Editor de Sons



Fonte: Elaborado pelo autor, 2020

Todas as mudanças poderão ser acompanhadas e testadas em tempo real observando-as no urso Panda situado em um **MINIPALCO** à esquerda da tela desse ambiente. Observe na figura 47 que existe também um pequeno painel contendo dois botões de controle da animação como iniciar e parar e um botão para deixar o palco no modo tela cheia:

Figura 47 - Minipalco para alterações feitas no Ator



Fonte: Elaborado pelo autor, 2020

Ainda na barra de menus, temos o ambiente de definição de **Palco (Background)** da animação e ou do projeto. É possível acessar esse ambiente clicando na figura de um pequeno lápis na frente de um quadradinho branco como ilustra a figura 48:

Figura 48 - Acesso ao Ambiente de Definição de Palco (Background)



Fonte: Elaborado pelo autor, 2020

Nesse ambiente é possível editar o **Fundo (Backdrop1)** e também os **Sons** que irão compor ou já compõe a cena ou o projeto todo. Similarmente às configurações das **Sprites (Atores)** no ambiente anterior, aqui nesta tela todas as mudanças poderão ser acompanhadas e testadas em tempo real observando-as no urso Panda situado em um minipalco à esquerda da tela desse ambiente.

Existe também um pequeno painel contendo dois botões de controle da animação como iniciar e parar e um botão para deixar o palco no modo tela cheia. As configurações para a inclusão ou edição de sons são exatamente as mesmas vistas anteriormente. Tudo o que foi descrito acima está ilustrado na figura 49:

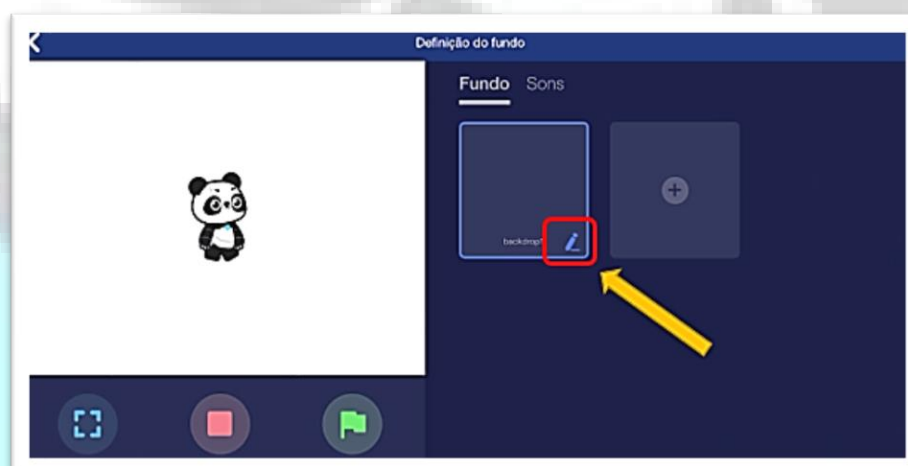
Figura 49 - Acesso ao Editor de Palco e Sons



Fonte: Elaborado pelo autor, 2020

Nesse mesmo ambiente é possível soltar a criatividade e criar o seu próprio desenho que poderá ser utilizado no projeto como fundo, sem a necessidade de usar os fundos pré-definidos contidos na biblioteca do mBlock. Para isso, basta clicar no desenho de um pequeno **lápiz azul** como ilustrado pela figura 50, para acessar o **Ambiente de Edição de Imagens** ou simplesmente **Editor de Imagens**:

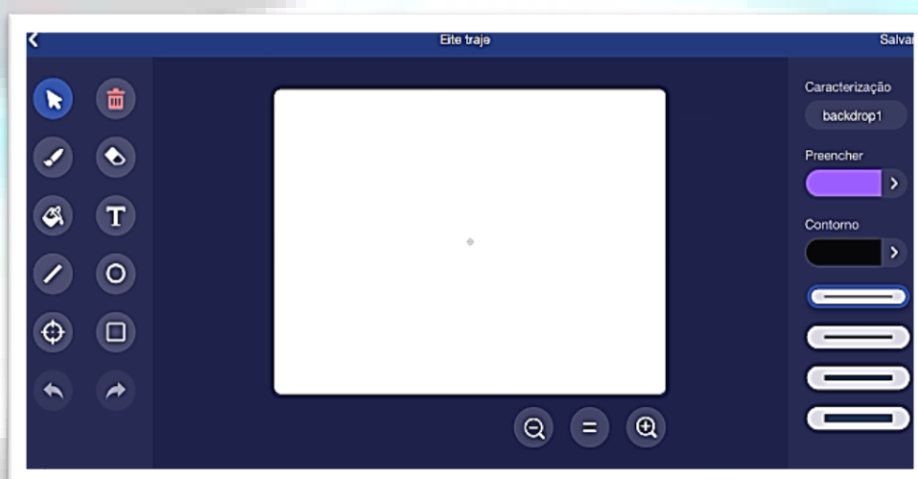
Figura 50 - Acesso ao Ambiente de Edição de Imagens



Fonte: Elaborado pelo autor, 2020

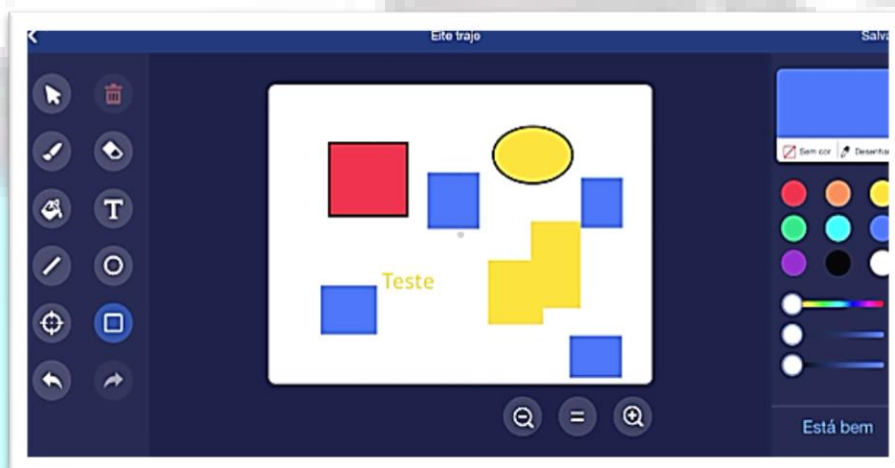
Trata-se de um ambiente de edição completo. Possui ferramentas para traçado de linhas com várias espessuras, preenchimento de cores, desenho a mão livre, inserção de figuras geométricas como círculos e retângulos além da edição de cores por meio de uma paleta específica e ferramentas para ampliação e diminuição da imagem. Podemos observar o **Ambiente de Edição de Imagens** ou simplesmente **Editor de Imagens** e as ferramentas, ilustrados em sequência nas figuras 51 e 52:

Figura 51 - Ambiente de Edição de Imagens



Fonte: Elaborado pelo autor, 2020

Figura 52 - Criação de Desenhos no Ambiente de Edição de Imagens

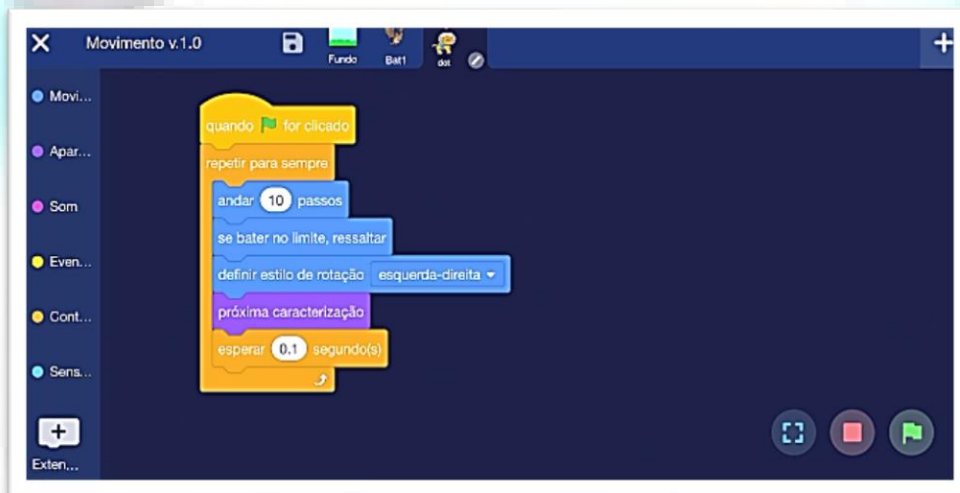


Fonte: Elaborado pelo autor, 2020

2

JANELA DE SCRIPTS OU SCRIPT ZONE: É o ambiente onde os **Scripts** são elaborados, ou seja, é o lugar onde é feita toda a programação do projeto. Os **Scripts** são construídos **arrastando e soltando (Drag and Drop)** os blocos de comando para esse ambiente e os encaixando como blocos de montar seguindo uma lógica estruturada. A Janela de **Scripts** ou **Script Zone** é ilustrada na figura 53:

Figura 53 - Janela do *Script Zone*



Fonte: Elaborado pelo autor, 2020

3

PALETA DE BLOCOS: Está subdividida em nove paletas identificadas por cores. São elas: **Movimento, Aparência, Som, Eventos, Controle, Sensores, Operadores, Variáveis e Meus Blocos**. Cada paleta designa um determinado conjunto de blocos com encaixes e reentrâncias específicas para qual foi desenvolvido.

Seguem-se as mesmas regras de design dos blocos da versão do mBlock para computadores, ou seja, os **Blocos de Controle** e os **Blocos de Comando** possuem em seu design uma pequena reentrância na parte superior do bloco e uma saliência na parte inferior permitindo ser encaixados no formato de pilhas.

Os **Blocos de Eventos** são permitidos apenas no início de um **Script**. Por sua vez, os **Blocos de Função** não possuem saliências e retornam algum valor. Estes blocos são usados como entradas para outros blocos e não podem ser utilizados de forma independente para compor um **Script do mBlock**. Os **Blocos de Função** também possuem duas características básicas:

- **Blocos** que possuem suas extremidades arredondadas podem retornar **números** ou **strings**²⁶
- **Blocos** que possuem as extremidades pontiagudas informam se alguma sentença ou expressão é verdadeira ou falsa, ou seja, retornam valores **booleanos**²⁷

Ao final dessa paleta existe a opção **Extensões** nas quais o usuário poderá adicionar mais blocos à paleta de blocos conforme a sua necessidade. Um bom exemplo é o bloco caneta se o usuário precisar desenhar em seu projeto como ilustrado na figura 54:

²⁶ Na programação de computadores, uma cadeia de caracteres ou *string*, é uma sequência de caracteres, geralmente utilizada para representar palavras, frases ou textos de um programa. Disponível em: https://pt.wikipedia.org/wiki/Desenho_vetorial. Acesso em: 25 de jan. de 2021.

²⁷ Em ciência da computação, booleano, ou lógico, é um tipo de dado primitivo que possui dois valores, que podem ser considerados como 0 ou 1, falso ou verdadeiro. Chamado booleano em homenagem a George Boole, que definiu um sistema de lógica algébrica pela primeira vez na metade do século XIX. Disponível em: <https://pt.wikipedia.org/wiki/Boolean>. Acesso em: 30 de jan. de 2021.

Figura 54 - Acesso às Extensões de Blocos

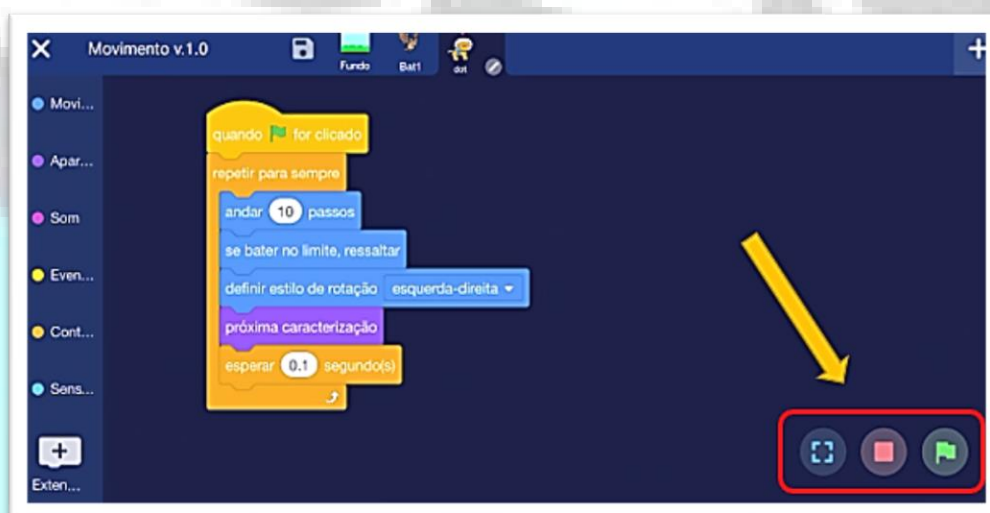


Fonte: Elaborado pelo autor, 2020



BOTÕES DE CONTROLE: É um pequeno painel contendo dois botões de controle da animação como iniciar e parar e um botão para deixar o palco no modo tela cheia. Podemos ver isso ilustrado na figura 55:

Figura 55 - Painel contendo os Botões de Controle



Fonte: Elaborado pelo autor, 2020

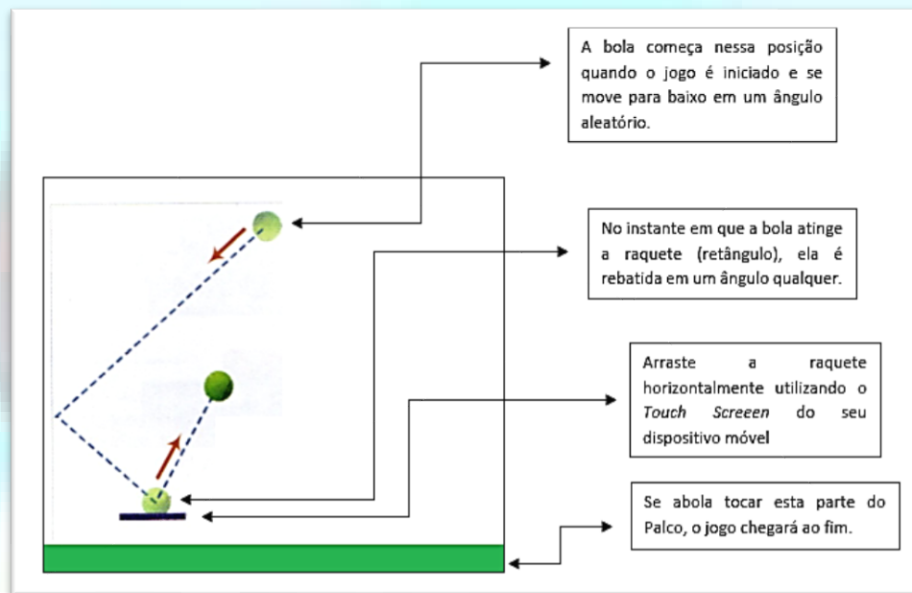
7.1 Projeto Exemplo de OA Digital para Dispositivos Móveis: O Jogo PONG

Neste projeto desenvolvemos uma versão do **Jogo PONG** para apenas um jogador no qual o mesmo tem a função de deslocar lateralmente uma raquete (em forma de um simples retângulo) para evitar que a bola atinja a parte inferior do Palco e com isso encerre o jogo, de acordo com o clássico jogo **Arcade PONG**²⁸.

Para este projeto, será utilizado um **smartphone com a plataforma Android** em todas as etapas do desenvolvimento. Salienta-se que o mesmo projeto pode ser construído com dispositivos móveis com a plataforma iOS. Esse projeto serve para exemplificar como é fácil e divertido utilizar a plataforma mBlock em dispositivos móveis sem precisar de um computador sofisticado para elaborar esse tipo de Objeto de Aprendizagem (OA) digital. A figura 56, ilustra uma ideia conceitual do **Jogo PONG**:

²⁸ PONG é o primeiro videogame lucrativo da história, dando origem a um novo setor da indústria. Foi de importância fundamental na história do videogame. Foi criado por Nolan Bushnell e Ted Dabney na forma de um console ligado a um monitor, movido a moedas. A primeira instalação em um bar de San Francisco, Califórnia, mostrou aos dois a possibilidade de lucro da criação. Assim, em 27 de Junho de 1972, a empresa Atari foi fundada. Disponível em: <https://pt.wikipedia.org/wiki/Pong>. Acesso em: 31 de jan. de 2021.

Figura 56 - Tela Conceitual do Jogo PONG



Fonte: Adaptado de (MARJI, 2014)

Como ilustrado na figura 56, a bola inicia na parte superior do Palco e se move para baixo em um ângulo aleatório, sendo rebatida ao tocar nas bordas do Palco. O jogador move a raquete horizontalmente (**usando o Touch Screen²⁹**) do seu dispositivo móvel para enviar a bola de volta para a parte superior do Palco. Se a bola tocar na parte inferior do Palco, o jogo chegará ao fim. Existe também a opção de o usuário utilizar um mouse adaptado ao seu dispositivo móvel por meio de um **Cabo Conector OTG³⁰**, o que melhoraria significativamente o seu controle sobre o jogo e também na confecção do mesmo e de muitos outros projetos no mBlock para dispositivos móveis.

Com esse cabo é possível conectar mouses, teclados, controles **USB**, *pen drives* e muito mais no *smartphone* ou *tablet* ou transferir dados do *pen drive* via entrada USB. As figuras 57 e 58, ilustram o conector **OTG** e um *smartphone* conectado a periféricos utilizando o **conector OTG**, respectivamente:

²⁹ Película tátil que pode ser ativada com a pressão de um dedo ou de uma caneta de feltro (sem tinta). Esta tela é ideal para jogos, para desenho no computador, ou outras atividades pedagógicas. Disponível em: https://pt.wikipedia.org/wiki/Ecr%C3%A3_t%C3%A1til. Acesso em: 01 de fev. de 2021.

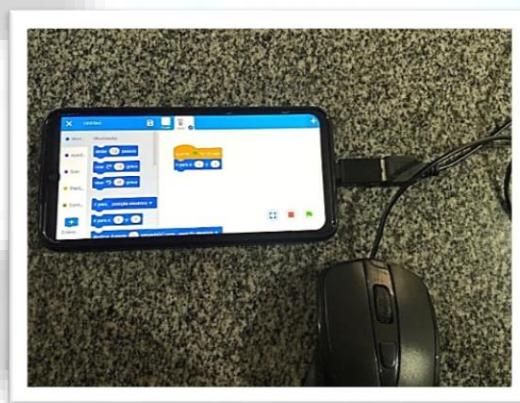
³⁰ OTG significa "On-The-Go" (em movimento), é um cabo ou adaptador usado em *Smartphones* para permitir que outros dispositivos sejam conectados ao celular. Disponível em: <https://canaltech.com.br/hardware/funcoes-cabo-otg/>. Acesso em: 29 de jul. de 2021.

Figura 57 - Conector ou Cabo OTG



Fonte: Elaborado pelo autor, 2020

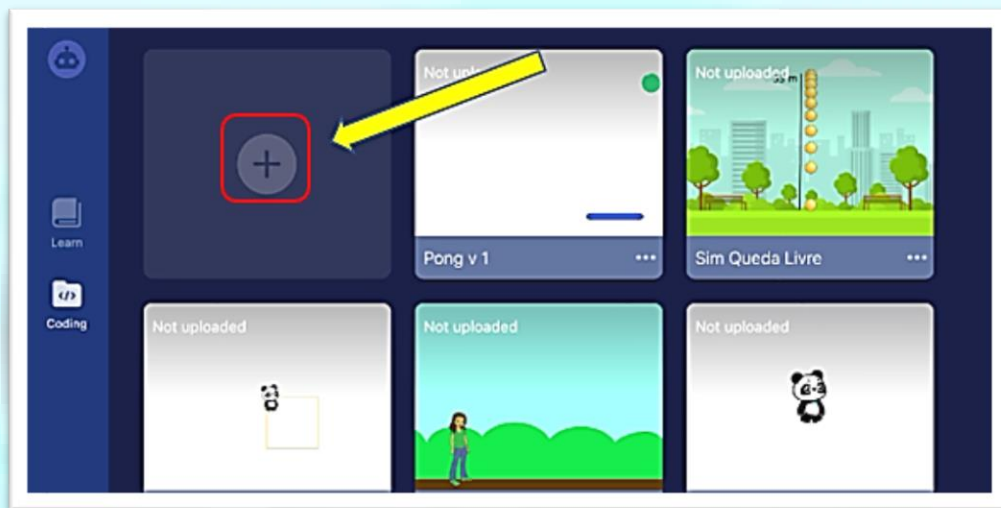
Figura 58 - Smartphone conectado a periféricos por meio do cabo OTG



Fonte: Elaborado pelo autor, 2020

Para criar esse jogo, devemos iniciar um novo projeto no mBlock para dispositivos móveis como ilustra a figura 59:

Figura 59 - Tela para Iniciar o Projeto do Jogo PONG



Fonte: Elaborado pelo autor, 2020

Em seguida deve-se apagar o **Sprite** do Panda clicando no símbolo de um pequeno lápis. Observe o que ilustra a figura 60:

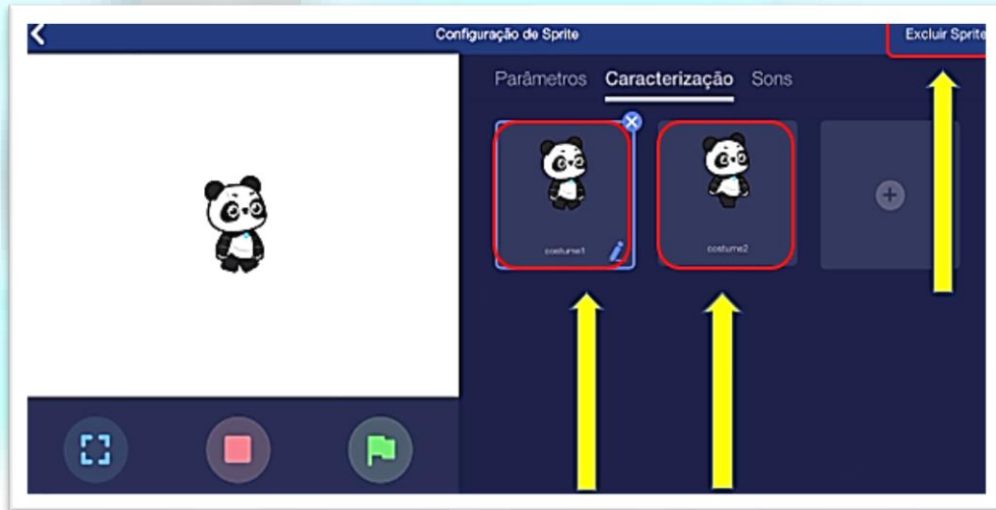
Figura 60 - Procedimento para Deletar o *Sprite* do Panda



Fonte: Elaborado pelo autor, 2020

Após deve-se clicar em **Excluir Sprite** (canto direito superior da tela) e em seguida excluindo as duas fantasias do Panda, conforme ilustra a figura 61:

Figura 61 - Procedimento para Excluir Sprite



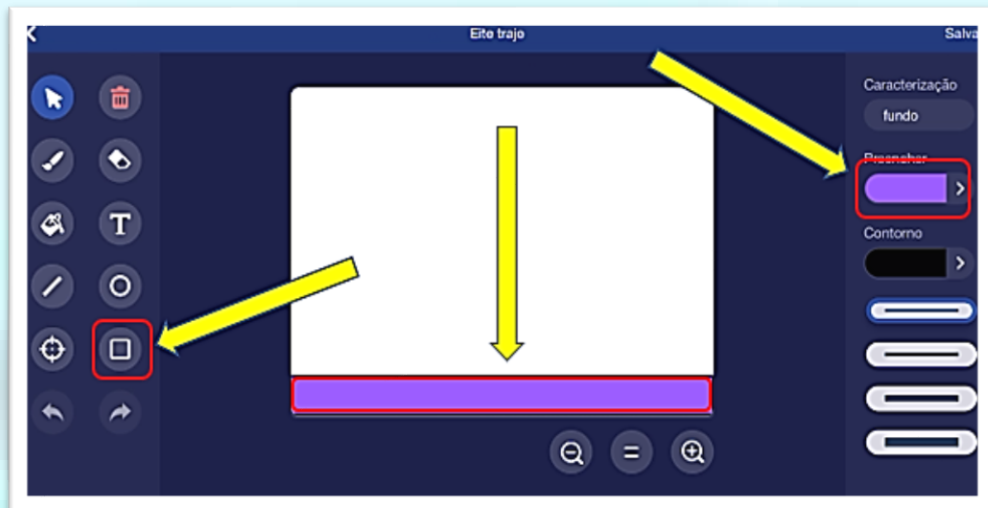
Fonte: Elaborado pelo autor, 2020

Para detectar se a bola não foi rebatida pela raquete (retângulo), a parte inferior do Palco será marcada com uma determinada cor. Para isso, utilizaremos o bloco



, da paleta **Sensores** para nos informar quando a bola interagir com essa cor. Nosso **Palco (Backdrop1)** atual é completamente branco, portanto, podemos simplesmente selecionar a ferramenta para desenhar um retângulo estreito e colorido na parte inferior do **Palco**, conforme é ilustrado na figura 62:

Figura 62 - Desenho de um Retângulo como Palco



Fonte: Elaborado pelo autor, 2020

O projeto contém dois **atores** ou **sprites**, um chama-se **bola** e o outro, a **raquete**, pode ser desenhada pelo usuário utilizando o **Editor de Imagens** do mBlock. Os dois possuem apenas uma fantasia. A **sprite** bola pode ser facilmente encontrada na biblioteca de atores (**Sprites**) com o nome **Ball**, conforme foi descrito com detalhes anteriormente. A figura 63 ilustra a localização da **Sprite Ball (Bola)** para ser utilizada no projeto do **Jogo PONG** a partir da **Biblioteca de Atores** pertencentes ao mBlock para dispositivos móveis:

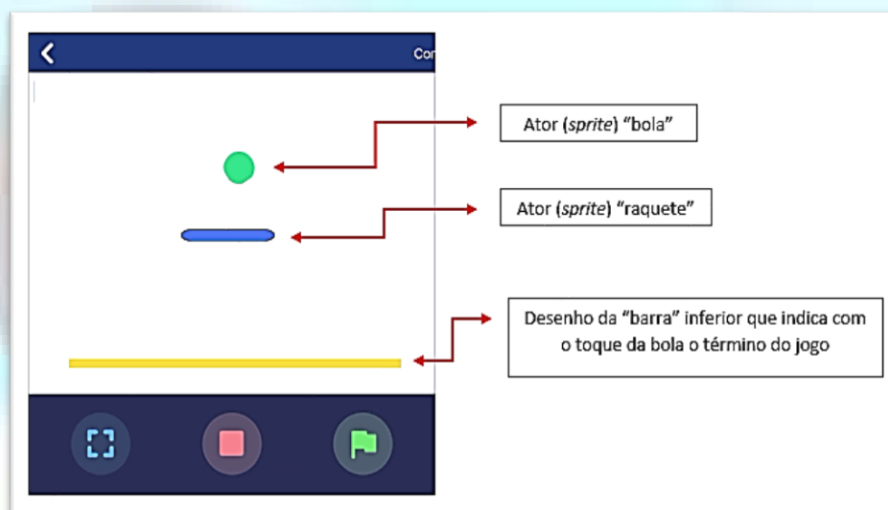
Figura 63 - Localização do **Sprite Bola** na Biblioteca de Atores



Fonte: Elaborado pelo autor, 2020

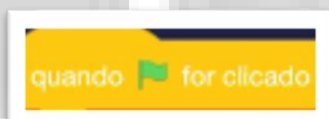
A construção final do Palco, com os atores **bola** e **raquete** juntamente com a **barra amarela** na parte inferior do Palco pode ser vista na figura 64:

Figura 64 - Construção Final do Palco com os Atores



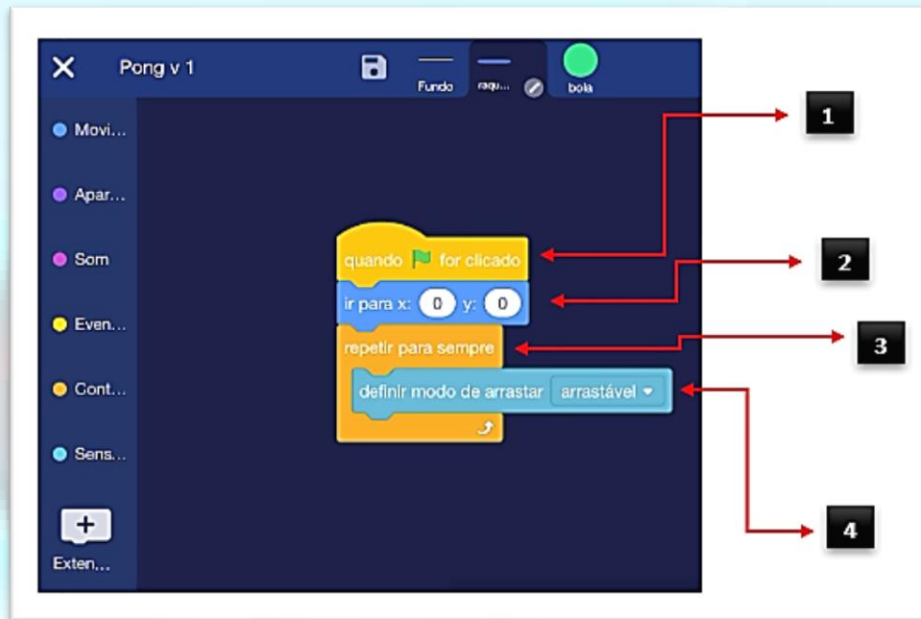
Fonte: Elaborado pelo autor, 2020

A ideia do código é bem simples. Qualquer **script** que inicie com o bloco **trigger**



(**Gatilho**), iniciará a execução quando esse botão for pressionado. A bandeira assumirá a cor verde brilhante e permanecerá assim até o que o **script** termine a sua execução. O **script** da **raquete** foi construído conforme ilustra a **Figura 65**:

Figura 65 - *Script* da Raquete para o Jogo PONG



Fonte: Elaborado pelo autor, 2020

Quando a bandeira verde for clicada **1**, o bloco **2** definirá a posição vertical da raquete para **x e y igual a 0**. A raquete deve flutuar imediatamente acima do retângulo amarelo que está na parte inferior do Palco. O **Script** então utiliza um

3 bloco **Forever (Para Sempre)** para verificar constantemente a posição da raquete na área do Palco utilizando a correspondência direta de posição (**x, y**) por

4 meio do **Touch Screen** do dispositivo móvel. O **Script** para o **Ator Bola (Sprite Ball)** é um pouco mais longo, mas de lógica simples. A bola deve começar a se mover para baixo segundo uma descrição resumida da sequência lógica construída para o ator Bola e a estrutura geral de pontuação e condição do término do jogo. O **Script** completo para o **Jogo PONG** está ilustrado na figura 66:

Figura 66 - Script Completo para o Jogo PONG

```
quando for clicado
  definir Pontos para 0
  dizer Prepare sua raquete! durante 2 segundos
  ir para x: -10 y: 150
  apontar na direção escolher aleatório entre 135 e 225
  repetir para sempre
    andar 12 passos
    se bater no limite, ressaltar
  se a tocar raquete? então
    apontar na direção escolher aleatório entre -30 e 30
    tocar o som pop até terminar
    alterar Pontos por 1
  se a tocar na cor? então
    tocar o som Alien Creak2 até terminar
    dizer Você perdeu! Tente novamente! durante 2 segundos
    parar todos
  se Pontos = 10 então
    Iniciar o som Clapping
    dizer Parabéns! Você venceu o jogo! durante 2 segundos
    parar todos
```

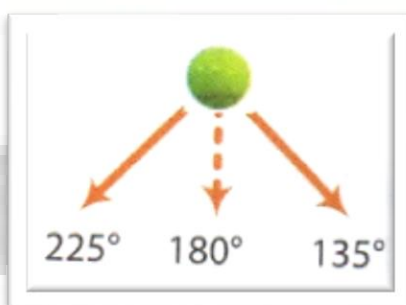
The image shows a Scratch script for a Pong game, organized into four numbered sections. Section 1 (labeled '1' in a black box) starts with a 'quando for clicado' event, sets 'Pontos' to 0, says 'Prepare sua raquete!' for 2 seconds, moves to x: -10, y: 150, and chooses a random direction between 135 and 225. It then enters a 'repetir para sempre' loop with 'andar 12 passos' and 'se bater no limite, ressaltar'. Section 2 (labeled '2') has a 'se a tocar raquete?' condition, which triggers 'apontar na direção' (random between -30 and 30), 'tocar o som pop até terminar', and 'alterar Pontos por 1'. Section 3 (labeled '3') has a 'se a tocar na cor?' condition, which triggers 'tocar o som Alien Creak2 até terminar', 'dizer Você perdeu! Tente novamente! durante 2 segundos', and 'parar todos'. Section 4 (labeled '4') has a 'se Pontos = 10' condition, which triggers 'Iniciar o som Clapping', 'dizer Parabéns! Você venceu o jogo! durante 2 segundos', and 'parar todos'. The script ends with a 'j' block.

Fonte: Elaborado pelo autor, 2020

1

Após um clique na bandeira verde que dá início ao processamento do **Script**, a bola será movida para a parte superior do palco por meio de valores de coordenadas cartesianas pré-definida pelo usuário, de forma que ela aponte para baixo em um ângulo aleatório entre **135°** e **225°**. Os ângulos possíveis para o **Ator Bola (Ball)** estão ilustrados na figura 67:

Figura 67 - Ângulos Possíveis para o Ator Bola



Fonte: Adaptado de (MARJI, 2014)

O restante desse fragmento de **Script** estabelece que a bola irá andar pelo Palco a uma **taxa de 12 passos** e quando ela tocar em qualquer uma das quatro bordas do Palco ela será rebatida em uma direção aleatória. Esse movimento contínuo da bola é definido pelo laço de controle repetir para sempre. O usuário poderá modificar a dificuldade do jogo alterando o número de passos que a bola pode dar no Palco. O contador de pontos é zerado cada vez que o usuário clicar na bandeira verde para iniciar uma nova partida.

2

Nesse fragmento de **Script**, se a bola tocar na raquete, a mesma irá rebater a bola num ângulo aleatório entre **- 30° e + 30°**. A cada rebatida, o **contador será alterado em +1 (um) ponto** e será tocado um som característico, ilustrando que a bola realmente tocou a raquete.

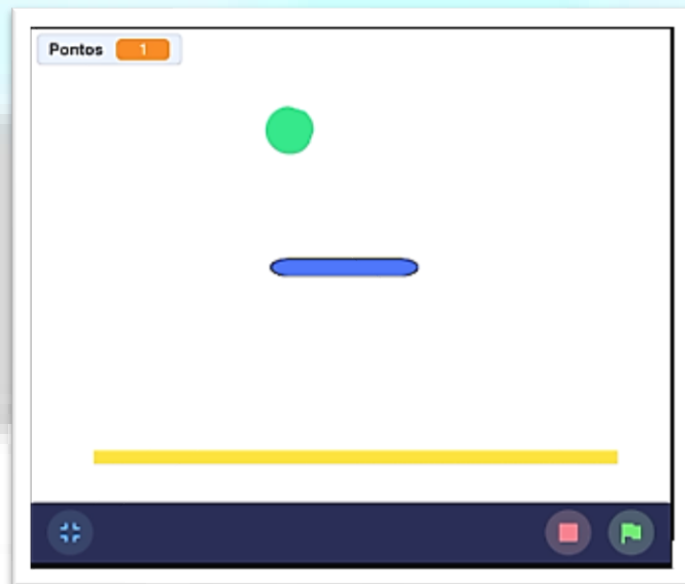
3

Nesse trecho do código, se a bola tocar no retângulo amarelo na parte inferior do cenário será tocado um som característico de erro e o jogo será encerrado. Por meio de uma caixa de diálogo será oferecida a oportunidade de o jogador iniciar uma nova partida e contabilizar os pontos ganhos até o momento da interrupção.

4

Finalmente na parte final do **Script**, o **contador de Pontos**, definido como dez, encerra a partida e estabelece por meio do som de palmas e de uma caixa de diálogo o jogador como vencedor do jogo se este limite é alcançado com êxito. **O bloco parar todos** encerra todos os **Scripts** em execução no momento. A figura 68 mostra a tela do **Projeto do Jogo PONG** finalizado e pronto para ser jogado. A tela mostra um contador para os pontos, os atores **Bola e Raquete**, **o Palco e o Retângulo** que serve de sensor para finalizar o jogo.

Figura 68 - Tela do Projeto do Jogo PONG Finalizado



Fonte: Elaborado pelo autor, 2020

Consideramos que a construção do **Jogo PONG**, ilustrou a concepção de um projeto de Objeto de Aprendizagem (OA) digital na plataforma mBlock para dispositivos móveis. Deve-se salientar também que nesse processo não é a construção ou a concepção do jogo em si que torna esse projeto com a finalidade educativa, mas a sua contextualização em um ambiente de aprendizagem escolar e um rico recurso educativo que o professor poderá explorar junto de seus alunos como um **Objeto de Aprendizagem (OA) digital educativo**.8. **CONSIDERAÇÕES FINAIS**

A utilização pelos professores das informações e teorias educacionais contidas neste *e-book* para o planejamento e/ou a elaboração de um Objeto de Aprendizagem (OA) digital deve ser muito bem planejada. Um OA digital não pode ser utilizado sem uma intenção pedagógica concreta e bem definida, apenas com o objetivo de tornar a aula mais agradável. Deve se pensar em que medida o OA digital apoiará o professor em sua tarefa de ensinar e quais os objetivos pedagógicos e metodológicos devem ser alcançados com a sua utilização.

Faça um bom uso deste *e-book*!

9. REFERENCIAL TEÓRICO

AGUIAR, E. V. B.; FLÔRES, M. L. P. **Objetos de Aprendizagem: Conceitos Básicos**. In: TAROUCO, L. M. R. **Objetos de Aprendizagem: teoria e prática**. Organizadores: Liane Margarida Rockenbach Tarouco, Bárbara Gorziza Ávila, Edson Felix dos Santos, Marta Rosecler Bez e Valeria Costa. Porto Alegre: Centro Interdisciplinar de Novas Tecnologias na Educação (CINTED – UFRGS), 2014. Disponível em: <https://www.lume.ufrgs.br/bitstream/handle/10183/102993/000937201.pdf>. Acesso em: 26 de jul. 2021.

BRAGA, J. C., **Desafios para o desenvolvimento de objetos de aprendizagem reutilizáveis e de qualidade**. In: DESAFIE! 2012, Curitiba. Anais... Curitiba/PR: CEIE/SBC, 2012. p. 90-99. Disponível em: <https://www.br-ie.org/pub/index.php/desafie/article/view/2779>. Acesso em: 20 fev. 2021.

BRAGA, J. (Org.). **Objetos de Aprendizagem Volume 1: introdução e fundamentos**. Santo André, SP: UFABC, 2015. 157 p. Disponível em: <https://www.interaufabc.com.br/>. Acesso em: 15 dez. 2021.

BRASIL. Ministério da Educação. **Base Nacional Comum Curricular: Educação é a base**. Versão final. Brasília, DF, 2017. Disponível em: http://basenacionalcomum.mec.gov.br/images/BNCC_EI_EF_110518_versaofinal_sit_e.pdf. Acesso em: 22 de maio de 2019.

BRASIL, Ministério da Educação. **Diretrizes Curriculares Nacionais para Educação Básica**. Brasília, DF: MEC, SEB, DICEI, 2013. Disponível em: <http://portal.mec.gov.br/docman/julho-2013-pdf/13677-diretrizes-educacao-basica-2013-pdf/file>. Acesso em: 09 de ago. de 2020.

BRASIL, Ministério da Educação. **Plano nacional de educação: PNE**. Brasília, DF: Inep, 2014. Disponível em: <http://portal.inep.gov.br/documents/186968/485745/Plano+Nacional+de+Educa%C3%A7%C3%A3o+PNE+2014-2024++Linha+de+Base/c2dd0faa-7227-40ee-a520-12c6fc77700f?version=1.1>. Acesso em: 09 de ago. de 2020.

BRACKMANN, C. P. **Desenvolvimento do pensamento computacional através de atividades desplugadas na educação básica**. Tese (Doutorado) – Universidade Federal do Rio Grande do Sul (UFRGS), Porto Alegre, 2017. Disponível em: <https://lume.ufrgs.br/handle/10183/172208>. Acesso em: 12 de jun. 2020.

BRITO, G. da Silva., PURIFICAÇÃO, I. da., **Educação e novas tecnologias: um (re)pensar**. Curitiba, PR: Editora InterSaberes, 2015.

BULEGON, A. M.; MUSSOI, E. M. **Objetos de Aprendizagem: Conceitos Básicos**. In: TAROUCO, L. M. R. **Objetos de Aprendizagem: teoria e prática**. Organizadores: Liane Margarida Rockenbach Tarouco, Bárbara Gorziza Ávila, Edson Felix dos Santos, Marta Rosecler Bez e Valeria Costa. Porto Alegre: Centro Interdisciplinar de Novas Tecnologias na Educação (CINTED – UFRGS), 2014. Disponível em: <https://www.lume.ufrgs.br/bitstream/handle/10183/102993/000937201.pdf>. Acesso em: 26 de jul. 2021.

TAROUCO, L. M. R. **Objetos de Aprendizagem: teoria e prática**. Organizadores: Liane Margarida Rockenbach Tarouco, Bárbara Gorziza Ávila, Edson Felix dos Santos, Marta Rosecler Bez e Valeria Costa. Porto Alegre: Centro Interdisciplinar de Novas Tecnologias na Educação (CINTED – UFRGS), 2014. Disponível em: <https://www.lume.ufrgs.br/bitstream/handle/10183/102993/000937201.pdf>. Acesso em: 26 de jul. 2021.

DIAS, C.L. *et al.* **Padrões abertos: aplicabilidade em Objetos de Aprendizagem (OAs)**. In: Simpósio Brasileiro de Informática na Educação (SBIE), n.20, 2009, Florianópolis. Anais... Florianópolis, SC. 2009. Disponível em: http://www.niee.ufrgs.br/eventos/SBIE/2009/conteudo/artigos/completos/61812_1.pdf. Acesso em: 22 out. 2020.

ELECTROFUN, 2021. Disponível em: <https://www.electrofun.pt/blog/mblock-scratch-arduino/>. Acesso em: 04 de jul. de 2021.

FLÔRES, M. L. P.; TAROUCO, L. M. R. **Diferentes tipos de objetos para dar suporte a aprendizagem**. Novas Tecnologias na Educação, v.6, n.1, p.1-10, 2008. Disponível em: <https://www.lume.ufrgs.br/bitstream/handle/10183/102993/000937201.pdf>. Acesso em 20 out. 2020.

GALAFASSI, F. P.; GLUZ, J. C.; GALAFASSI, C.; **Análise Crítica das Pesquisas Recentes sobre Tecnologias de Objetos de Aprendizagem e Ambientes Virtuais de Aprendizagem**. Revista Brasileira de Informática na Educação, v.21, n.3, p.100, 2014. Disponível em: <https://www.br-ie.org/pub/index.php/rbie/article/view/2351>. Acesso em: 10 set. 2020.

KENSKI, V. M., **Educação e tecnologias: O novo ritmo da informação**. Campinas, SP: Papirus, 2012.

PROGRAMAÇÃO EM BLOCOS: APRENDENDO DE MANEIRA DIVERTIDA. I **Do Code - Escola de Programação e Tecnologia**. Florianópolis, 02 de out. de 2018. Disponível em <https://idocode.com.br/blog/programacao/programacao-em-blocos/>. Acesso em: 28 de jul. de 2021.

MARJI, M., **Aprenda a programar com Scratch: Uma introdução visual à programação com jogos, arte, ciência e matemática**. 1 ed. São Paulo: Novatec Editora Ltda., 2014. 284 p.

MENEZES, C. S.; FERRETI, C.; LINDNER, E. L.; LIRA, A. F. Aplicando arquiteturas pedagógicas em objetos digitais interativos. In: **RENOTE – Revista Novas Tecnologias para a Educação**. Porto Alegre: Centro Interdisciplinar de Novas Tecnologias na Educação (CINTED – UFRGS), v. 4, n° 2, 2006. Disponível em: <https://seer.ufrgs.br/renote/article/view/14138>. Acesso em: 26 de jul. 2021.

PAPERT, S. **A máquina das crianças: repensando a escola na era informática**. Porto Alegre, RS: Artes Médicas, 1994.

PASCAL JÚNIOR., P. A. Pensamento Computacional em sala de aula: muito além da programação. **Fundação Telefônica Vivo**, São Paulo, 28, out. de 2018. Disponível em: <https://fundacaotelefonicavivo.org.br/noticias/pensamento-computacional-em-sala-de-aula-muito-alem-da-programacao/>. Acesso em: 28 de jul. de 2021.

REBOUÇAS, Ayla Dantas; MAIA, Dennys Leite; SCAICO, Pasqueline Dantas. **Objetos de Aprendizagem: da Definição ao Desenvolvimento**, Passando pela Sala de Aula. In: PIMENTEL, Mariano; SAMPAIO, Fábio F.; SANTOS, Edméa O. (Org.). **Informática na Educação: ambientes de aprendizagem, objetos de aprendizagem e empreendedorismo**. Porto Alegre: Sociedade Brasileira de Computação, 2021. (Série Informática na Educação, v.5) Disponível em: <https://ieducao.ceie-br.org/objetos-aprendizagem/>. Acesso em: 23 de jul. 2021.

RESNICK, M. **Technologies for lifelong kindergarten**. Educational Technology Research and Development, v. 46, n.4, 1998. Disponível em: <https://web.media.mit.edu/~mres/papers.html>. Acesso em: 22 de jun. 2020.

RESNICK, M. **Behavior construction kits**. Communications of the ACM. v. 36, n. 7, p. 64-71, jul. 1993. Disponível em: <https://web.media.mit.edu/~mres/papers.html>. Acesso em: 22 de jun. 2020.

SÁ FILHO, C. S.; MACHADO, E. C. **O computador como agente transformador da educação e o papel do objeto da aprendizagem**. Universia, 17 dez. 2004. Disponível em: <https://www.universia.net/br/actualidad>. Acesso em: 15 mar. 2020.

SANTOS, P. R. de S. **O Professor e sua Prática: do planejamento as estratégias pedagógicas**. 2011. Disponível em: <https://docplayer.com.br/22769674-O-professor-e-sua-pratica-do-planejamento-as-estrategias-pedagogicas-1-o-professor-e-sua-pratica-do-planejamento-as-estrategias-pedagogicas.html>. Acesso em: 11/10/2020.

SANTOS, C. G. dos. **Guia das Atividades desplugadas para o desenvolvimento do pensamento computacional** [recurso eletrônico] / Cícero Gonçalves dos Santos, Maria Augusta Silveira Netto Nunes, Margarida Romero. – Porto Alegre: SBC, 2019. Disponível em: <http://almanaguesdacomputacao.com.br/gutanunes/publications/S12V2.pdf>. Acesso em: 07 de ago. de 2021.

SATHLER, L. **A Cultura digital e o pensamento computacional na BNCC**. Associação Brasileira de Educação a Distância – ABED. Sucesu Minas. Inforuso. 14 de ago. de 2020. Disponível em: https://www.researchgate.net/publication/343658070_A_cultura_digital_e_o_pensamento_computacional_na_BNCC. Acesso em: 01 de ago. de 2021.

SILVA, S. L. R. da.; ANDRADE, A. V. C. de; BRINATTI, A. M. **Ensino remoto emergencial** [livro eletrônico]. Ponta Grossa, PR: Ed. Dos Autores, 2020. Disponível em: http://www1.fisica.org.br/mnpef/sites/default/files/anexosnoticia/EnsinoRemotoEmergencial_SilvaAndradeBrinatti.pdf. Acesso em: 24 de mai. de 2021.

SILVA, Alzira Ferreira; SOARES, Cláudia Vivien Carvalho de Oliveira; SOUZA, Elmara Pereira. **Construção de software educativo, objeto de aprendizagem e recurso educacional aberto para o desenvolvimento do pensamento computacional**. In: SAMPAIO, Fábio F.; PIMENTEL, Mariano; SANTOS, Edméa O. (Org.). *Informática na Educação: pensamento computacional, robótica e internet das coisas*. Porto Alegre: Sociedade Brasileira de Computação, 2021. (Série Informática na Educação CEIE-SBC, v.6). Disponível em: <https://educacao.ceie-br.org/pensamentocomputacional>. Acesso em: 03 de ago. de 2021.

TAROUCO, L. M. R. **Objetos de Aprendizagem: teoria e prática**. Organizadores: Liane Margarida Rockenbach Tarouco, Bárbara Gorziza Ávila, Edson Felix dos Santos, Marta Rosecler Bez e Valeria Costa. Porto Alegre: Centro Interdisciplinar de Novas Tecnologias na Educação (CINTED – UFRGS), 2014. Disponível em: <https://www.lume.ufrgs.br/bitstream/handle/10183/102993/000937201.pdf>. Acesso em: 26 de jul. 2021.

TAROUCO, L. M. R.; FABRE, M. J.; TAMUSIUNAS, F. R. Reusabilidade de objetos educacionais. In: **RENOTE – Revista Novas Tecnologias para a Educação**. Porto

Alegre: Centro Interdisciplinar de Novas Tecnologias na Educação (CINTED-UFRGS), v. 1. n° 1, 2003. Disponível em: <https://lume.ufrgs.br/handle/10183/12975>. Acesso em: 26 de jul. de 2021.

WILEY, D.; **Learning objects need instructional design theory**. The ASTD e-Learning handbook, p. 115-126, 2002. Disponível em: [https://www.scirp.org/\(S\(lz5mqp453edsnp55rrgict55\)\)/reference/ReferencesPapers.aspx?ReferenceID=589634](https://www.scirp.org/(S(lz5mqp453edsnp55rrgict55))/reference/ReferencesPapers.aspx?ReferenceID=589634). Acesso em: 20 de set. 2020.

WING, J. M. **Computacional Thinking**. Communications of the ACM, v. 49, n. 3, p. 33-35, 2006. Disponível em: <https://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf>. Acesso em: 30 mar. 2020.

WING, J. M. **Computational Thinking Benefits Society**. Social Issues in Computing. 2014. Disponível em: <http://socialissues.cs.toronto.edu/2014/01/computational-thinking/>. Acesso em: 24/11/2020.

GUIA TÉCNICO E PEDAGÓGICO DO PROJETO-EXEMPLO 1 - O PANDA

O produto educacional deste projeto de mestrado caracteriza-se por meio de um curso online. O curso online versa sobre a elaboração de Objetos de Aprendizagem (OAs) digitais usando a programação por blocos visuais oferecida pela plataforma mBlock. O curso oferece um material digital auto instrutivo (um *e-book*, cinco videoaulas de aproximadamente vinte minutos de duração cada uma e também um Guia Técnico Pedagógico específico para cada projeto-exemplo). Este material digital está disponível para leitura e download na plataforma *Moodle* institucional da UTFPR por meio do link (<https://moodle.utfpr.edu.br/login/index.php>). Não haverá intervenção dos pesquisadores ou encontros virtuais programados para realizar as atividades do curso. O participante poderá utilizar os Fóruns disponíveis na área do curso na plataforma *Moodle* para postar e compartilhar com os outros cursistas as suas dúvidas e comentários. Os projetos-exemplo de Objetos de Aprendizagem (OAs) digitais contidos no produto educacional são independentes entre si e servem apenas como exemplo básico para futuras implementações e adaptações, por parte dos professores, adequando-os aos objetivos pedagógicos e de planejamento metodológico de sua disciplina de atuação. Salienta-se que não é necessário e nem obrigatório aplicar com os alunos os projetos-exemplo no período vigente da pesquisa.



O professor poderá consultar a BNCC (2017) para adotar uma habilidade que espera que os alunos de determinada etapa da Educação Básica possam condições de desenvolver por meio dos Objetos de Aprendizagem (OAs) digitais exemplificados nos projetos exemplo contidos no material digital.



[4.0 Internacional](#)

Esta licença permite que outros remixem, adaptem e criem a partir do trabalho para fins não comerciais, desde que atribuam o devido crédito e que licenciem as novas criações sob termos idênticos. Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

1. OBJETIVOS GERAIS

- Compreender e praticar a Programação por Blocos por meio da construção de estruturas lógicas simples ou complexas ajudando a desenvolver os conceitos básicos do Pensamento Computacional (PC);
- Desenvolver os quatro pilares do Pensamento Computacional (PC): abstração, decomposição, reconhecimento de padrões e algoritmos;
- Promover o desenvolvimento da competência Cultura Digital (BNCC, 2017), bem como as suas dimensões e subdimensões em diversas etapas da Educação Básica;
- Criar e manipular mídias exercitando a criatividade e a autoria;
- Compartilhar ideias e projetos de forma ética;
- Desenvolver o raciocínio lógico matemático;
- Despertar o interesse dos educadores com relação ao uso das Tecnologias Digitais Educacionais (TED) no ensino e aprendizagem de conteúdos;
- Utilizar estratégias, conceitos e procedimentos para interpretar situações em diversos contextos;
- Propor soluções ou participar de ações para investigar desafios do mundo contemporâneo;
- Interpretar, construir modelos em diversos contextos, analisando a adequação das soluções propostas;
- Adaptar a metodologia proposta à realidade de cada disciplina e/ou conteúdo escolar;
- **Educação Especial** – Metodologia com o potencial em promover o desenvolvimento das habilidades dos alunos com algum tipo de deficiência, priorizando o atendimento especializado. Possibilidade da elaboração de projetos de Objetos de Aprendizagem (OAs) digitais direcionados a um público-alvo de alunos com algum tipo de deficiência (auditiva, visual, intelectual, física ou múltipla), com distúrbios de aprendizagem ou com altas habilidades;
- **Educação Inclusiva** – Metodologia com o potencial em promover o ensino e a aprendizagem na qual o processo educativo deve ser considerado como um processo social, ou seja, possibilitar a todos os alunos, com deficiência ou não,

total direito à escolarização. Prioridade a uma educação voltada para a formação completa dos alunos e livre de preconceitos, reconhecendo as diferenças e dando a elas o seu devido valor.

2. INTEGRAÇÃO ENTRE HABILIDADES PREVISTAS NA BNCC (2017)

Os projetos-exemplo de Objetos de Aprendizagem (OAs) digitais presentes neste projeto de mestrado pretendem integrar as habilidades de diversas áreas do conhecimento com as habilidades referentes a Matemática e Matemática e suas Tecnologias listadas abaixo:

- **EF06MA04:** construir algoritmo em linguagem natural e representa-lo por fluxograma que indique a resolução de um problema simples
- **EF07MA05:** Resolver um mesmo problema utilizando diferentes algoritmos
- **EF07MA06:** Reconhecer que as resoluções de um grupo de problemas que têm a mesma estrutura pode ser obtidas utilizando os mesmos procedimentos
- **EF07MA07:** Representar por meio de um fluxograma os passos utilizados para resolver um grupo de problemas
- **EF07MA20:** Reconhecer e representar, no plano cartesiano, o simétrico de figuras em relação aos eixos e à origem
- **EF07MA13:** Compreender a ideia de variável, representada por letra ou símbolo, para expressar relação entre duas grandezas, diferenciando-a da ideia de incógnita
- **EF07MA18:** Resolver e elaborar problemas que possam ser representados por equações polinomiais de primeiro grau, redutíveis à forma $ax + b = c$, fazendo uso das propriedades de igualdade
- **EF08MA06:** Resolver e elaborar problemas que envolvam cálculo de valor numérico de expressões algébricas, utilizando as propriedades das operações
- **EF06MA04:** construir algoritmo em linguagem natural e representa-lo por fluxograma que indique a resolução de um problema simples
- **EF07MA05:** Resolver um mesmo problema utilizando diferentes algoritmos
- **EF07MA06:** Reconhecer que as resoluções de um grupo de problemas que têm a mesma estrutura pode ser obtidas utilizando os mesmos procedimentos

- **EF07MA07:** Representar por meio de um fluxograma os passos utilizados para resolver um grupo de problemas
- **EF07MA20:** Reconhecer e representar, no plano cartesiano, o simétrico de figuras em relação aos eixos e à origem
- **EF07MA13:** Compreender a ideia de variável, representada por letra ou símbolo, para expressar relação entre duas grandezas, diferenciando-a da ideia de incógnita
- **EF07MA18:** Resolver e elaborar problemas que possam ser representados por equações polinomiais de primeiro grau, redutíveis à forma $a x + b = c$, fazendo uso das propriedades de igualdade
- **EF08MA06:** Resolver e elaborar problemas que envolvam cálculo de valor numérico de expressões algébricas, utilizando as propriedades das operações
- **EF08MA19:** Resolver e elaborar problemas que envolvam medidas de áreas de figuras geométricas, utilizando expressões de cálculo de área (quadriláteros, triângulos e círculos), em situações como determinar a medida de terrenos
- **EF09MA04:** Resolver e elaborar problemas com números reais, inclusive em notação científica, envolvendo diferentes operações
- **EF09MA06:** Compreender as funções como relações de dependência unívoca entre duas variáveis e suas representações numérica, algébrica e gráfica e utilizar esse conceito para analisar situações que envolvam relações funcionais entre duas variáveis
- **EM13MAT302:** Construir modelos empregando as funções polinomiais de primeiro e segundo graus, para resolver problemas em contextos diversos, com ou sem apoio de tecnologias digitais
- **EM13MAT315:** Investigar e registrar, por meio de um fluxograma, quando possível, um algoritmo que resolve um problema
- **EM13MAT405:** Utilizar conceitos iniciais de uma linguagem de programação na implementação de algoritmos escritos em linguagem corrente e/ou matemática

3. RECURSOS GERAIS

Recursos envolvidos na elaboração deste projeto-exemplo de Objeto de Aprendizagem (OA) digital:

- Programação por Blocos Visuais
- Criação e Manipulação de *Script*
- Noção sobre Fluxo de Programa
- Manipulação do Ator (*Sprite Panda1*)
- Manipulação de blocos de Eventos
- Manipulação de blocos de Sensores
- Manipulação de blocos de Controle
- Manipulação de blocos de Aparência
- Manipulação do Cenário (*Blue Sky*)
- Utilização de Comandos Básicos: Iniciar *Script*, Movimento (*Andar*), Posição (*Plano Cartesiano*), Interação (*Falar*), Rotação, Ângulos
- Criação e Manipulação Digitalmente de Imagens

4. CARACTERÍSTICAS PEDAGÓGICAS

Principais características pedagógicas desejáveis na elaboração deste projeto-exemplo de Objeto de Aprendizagem (OA) digital:

- **Interatividade:** indica se há suporte às consolidações e ações mentais, requerendo que o aluno interaja com o conteúdo do Objeto de Aprendizagem (OA) de alguma forma, podendo ver, escutar ou responder algo;
- **Autonomia:** indica se os Objetos de Aprendizagem (OAs) apoiam a iniciativa e tomada de decisão;
- **Cooperação:** indica se há suporte para aos alunos trocarem opiniões e trabalhar coletivamente sobre o conceito apresentado;
- **Cognição:** refere-se às sobrecargas cognitivas alocadas na memória do aluno durante o processo de ensino-aprendizagem;

- **Afetividade:** refere-se aos sentimentos e motivações do aluno com sua aprendizagem e durante a interação com o Objeto de Aprendizagem.

5. CARACTERÍSTICAS TÉCNICAS

Principais características técnicas desejáveis na elaboração deste projeto-exemplo de Objeto de Aprendizagem (OA) digital:

- **Acessibilidade:** indica se o Objeto de Aprendizagem pode ser acessado por diferentes tipos de usuários (ex.: deficientes visuais e auditivos etc.), em diferentes lugares (ex. lugares com acesso à internet e sem acesso à internet, etc.), e por diferentes tipos de dispositivos (ex. computadores, notebooks, dispositivos móveis, etc.);
- **Agregação:** indica se os componentes do Objeto de Aprendizagem podem ser agrupados em conjuntos maiores de conteúdos (ex. estruturas tradicionais de um curso);
- **Confiabilidade:** indica que o Objeto de Aprendizagem não apresenta defeitos técnicos ou problemas no conteúdo pedagógico;
- **Disponibilidade:** indica se o Objeto de Aprendizagem está disponível para ser utilizado;
- **Durabilidade:** indica se o Objeto de Aprendizagem se mantém intacto quando o repositório em que ele está armazenado muda ou sofre problemas técnicos;
- **Facilidade de instalação:** indica se o Objeto de Aprendizagem pode ser facilmente instalado caso ele exija esse recurso;
- **Granularidade:** é a extensão à qual um Objeto de Aprendizagem é composto por componentes menores e reutilizáveis.
- **Interoperabilidade:** medida do esforço necessário para que os dados dos Objetos de Aprendizagem (OAs) possam ser integrados a vários sistemas;
- **Manutenibilidade:** é a medida de esforço necessária para alterações do Objeto de Aprendizagem (OA);
- **Reusabilidade:** indica as possibilidades de reutilizar os Objetos de Aprendizagem (OAs) em diferentes contextos ou aplicações. Essa é a

principal característica de um Objeto de Aprendizagem e pode ser influenciada por todas as demais.

6. A CULTURA DIGITAL BNCC (2017)

Características da competência Cultura Digital BNCC (2017) encontradas na elaboração deste projeto-exemplo de Objeto de Aprendizagem (OA) digital:

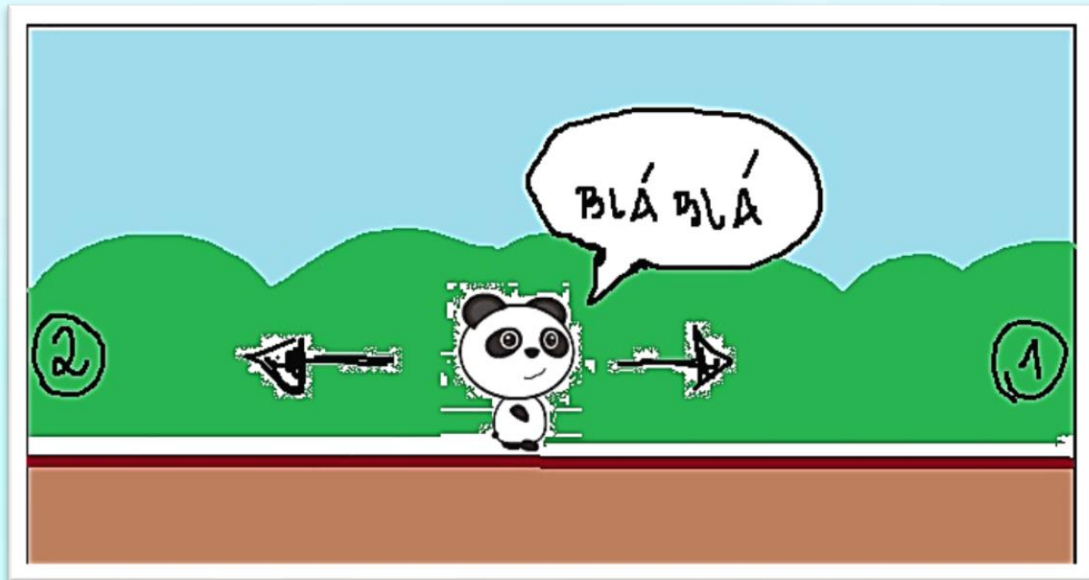
- **Computação e Programação** – Utilização de Ferramentas Digitais, Produção de Multimídia, Linguagens de Programação;
- **Pensamento Computacional** – Domínio de Algoritmos e Resolução de Problemas;
- **Cultura e Mundo Digital** – Mundo Digital, Uso Ético.

7. ELABORAÇÃO DO PROJETO-EXEMPLO 1

7.1 Sketch do Design e Objetivos Básicos do Projeto-Exemplo 1

Antes de começar a programação de um novo projeto é importante fazer um sketch para se ter uma ideia geral do design e dos objetivos a serem alcançados. O projeto-exemplo **O Panda** é introdutório e servirá para o professor estabelecer o primeiro contato com os recursos operacionais e didáticos da plataforma mBlock. O objetivo deste projeto é muito simples, criar um cenário para o urso mascote do mBlock, o Panda andar e falar alguma frase. O Panda é o único ator deste projeto-exemplo. O movimento do ator Panda é unidimensional no sentido horizontal, ou seja, somente esquerda-direita sem o controle do usuário. Chegando à borda do cenário o ator Panda bate e volta e vai até o outro limite do cenário repetindo o processo. Não há limite de tempo e nem pontuação a ser alcançada. O objetivo deste projeto-exemplo é servir como base para futuras implementações e adaptações conforme o professor julgar necessárias para a metodologia de ensino e aprendizagem de determinados conteúdos da sua disciplina em diferentes etapas da Educação Básica.

Sketch do Projeto O Panda

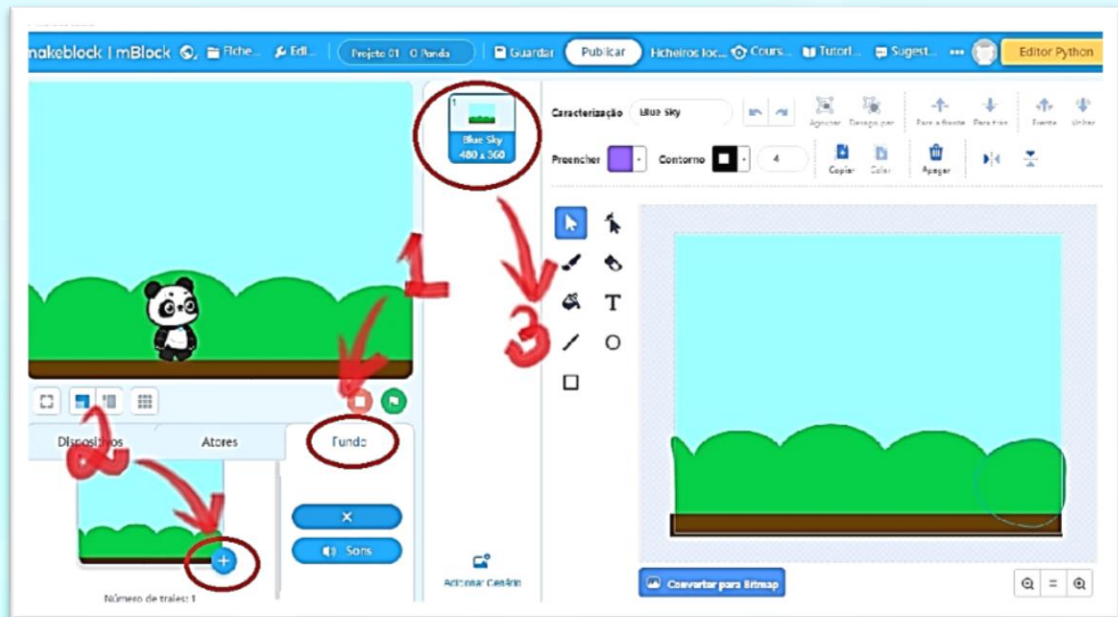


Fonte: Elaborado pelo autor, 2021

7.2 O Palco (Cenário)

- 1 – Ao selecionar a aba **FUNDO** pode-se carregar a imagem do Palco (Cenário)
- 2 – Clicar no símbolo **+** permite abrir a Biblioteca de Cenários. Pode-se digitar na caixa de busca pelo cenário **BLUE SKY** ou outro cenário.
- 3 – Ícone indicando que o cenário **BLUE SKY** foi carregado no Editor de Imagem, caso queira fazer edições nesta imagem.

Configuração do cenário Blue Sky



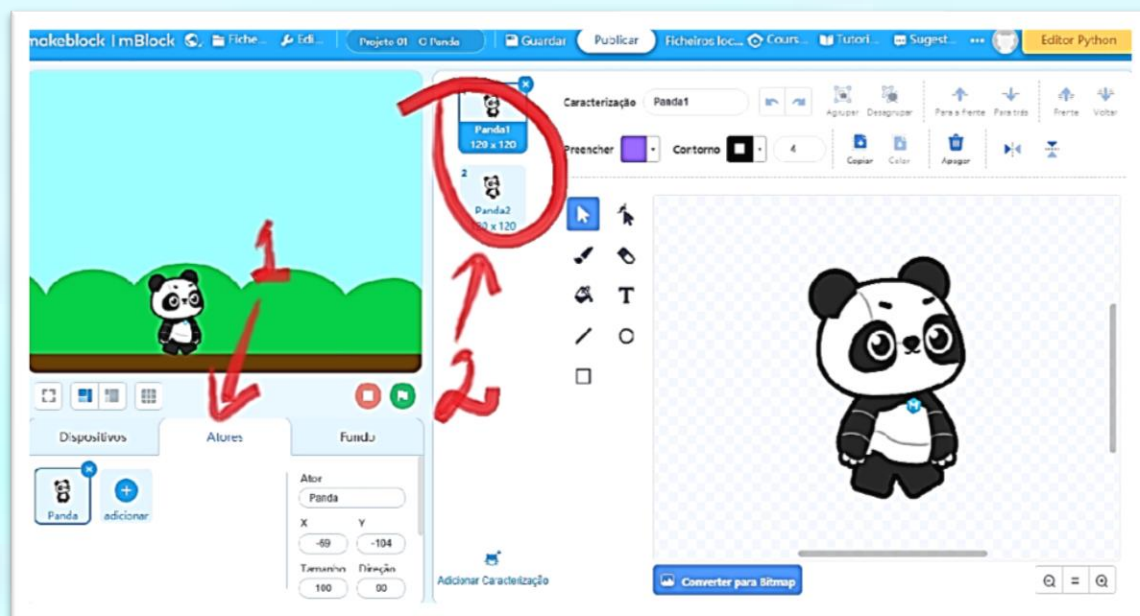
Fonte: Elaborado pelo autor, 2021

7.3 O Ator Panda1

1 – Ao selecionar a aba **ATOR** pode-se clicar no símbolo **+** para abrir a Biblioteca de Atores. digitar na caixa de busca pelo *sprite* do ator **Panda1** ou outro *sprite*. Nesse momento pode-se editar o *sprite* do ator Panda se preferir;

2 – Existem duas fantasias para o *sprite* Panda (**Panda1** e **Panda 2**). Elas são usadas neste projeto-exemplo para dar a impressão de movimento quando programarmos o ator Panda para andar pelo Palco (Cenário) como nos desenhos animados.

Configuração do ator Panda1



Fonte: Elaborado pelo autor, 2021

7.4 Blocos de código para o Ator Panda1

1 – Seleccionamos o ator Panda1 para que se possa atribuir blocos de código a este ator:

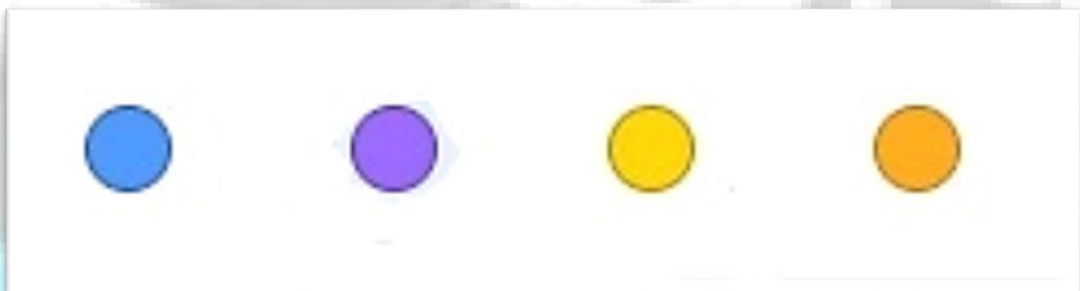
Blocos de código para o ator Panda1



Fonte: Elaborado pelo autor, 2021

2 – Cor dos blocos usados para o ator Panda foram: **Movimento, Aparência, Eventos e Controle.**

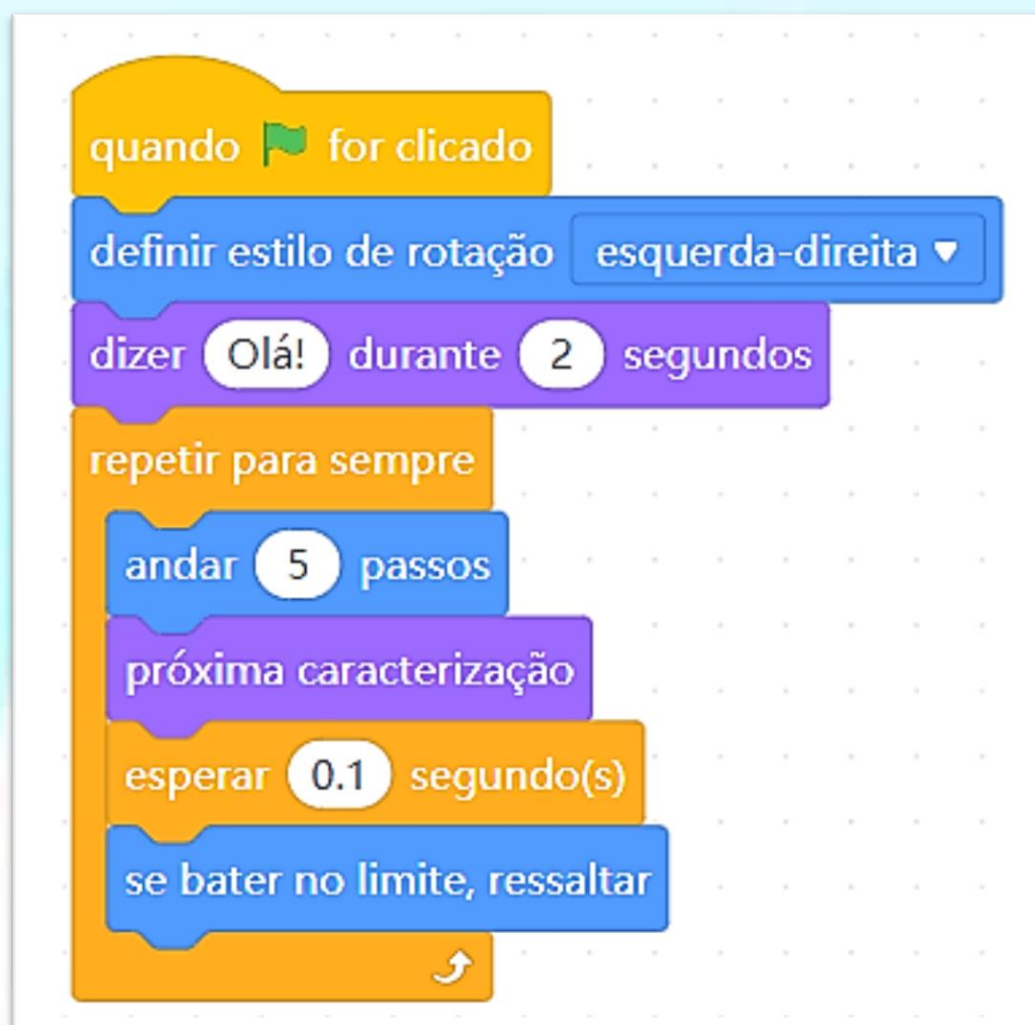
Cor dos blocos para o Ator Panda1



Fonte: Elaborado pelo autor, 2021

3 – Os blocos serão arrastados para a **ÁREA DE SCRIPT** e encaixados de forma que se estabeleça um fluxo lógico e coerente de comandos. Nesta área é que a mágica da programação por blocos visuais acontece!

Encaixe de blocos para o ator Panda1



Fonte: Elaborado pelo autor, 2021

7.5 Resultado Final do Projeto-Exemplo O Panda

Projeto-Exemplo O Panda



Fonte: Elaborado pelo autor, 2021

8. SUGESTÕES PEDAGÓGICAS, TÉCNICAS E METODOLÓGICAS PARA ALTERAÇÕES E/OU ADEQUAÇÕES NO PROJETO-EXEMPLO 1: O PANDA

- Professores e alunos podem criar seus próprios *sprites* dos atores e do cenário no Editor de Imagens do mBlock;
- Incluir tempo limite para o ator Panda1 percorrer as laterais do cenário;
- Incluir outros atores para interagirem com o ator Panda1;
- Possibilidade da portabilidade deste projeto-exemplo para dispositivos móveis (*Smartphones* e *Tablets*);
- Incluir sons por meio do Editor de Sons e novas cores ao cenário(os) e ator(res);

- Planejar e/ou realizar modificações neste projeto-exemplo focando na **Educação Especial** e na **Educação Inclusiva**.

Críticas, sugestões ou relato de erros serão bem recebidos!

Favor enviar e-mail para Prof. André

endriuscampos@gmail.com

BONS ESTUDOS E ATÉ O PRÓXIMO PROJETO-EXEMPLO!



GUIA TÉCNICO E PEDAGÓGICO DO PROJETO-EXEMPLO 2 – CLASSES GRAMATICAIS

O produto educacional deste projeto de mestrado caracteriza-se por meio de um curso online. O curso online versa sobre a elaboração de Objetos de Aprendizagem (OAs) digitais usando a programação por blocos visuais oferecida pela plataforma mBlock. O curso oferece um material digital auto instrutivo (um e-



book, cinco videoaulas de aproximadamente vinte minutos de duração cada uma e também um Guia Técnico Pedagógico específico para cada projeto-exemplo). Este material digital está disponível para leitura e download na plataforma Moodle institucional da UTFPR por meio do link (<https://moodle.utfpr.edu.br/login/index.php>). Não haverá intervenção dos pesquisadores ou encontros virtuais programados para realizar as atividades do curso. O participante poderá utilizar os Fóruns disponíveis na área do curso na plataforma Moodle para postar e compartilhar com os outros cursistas as suas dúvidas e comentários. Os projetos-exemplo de Objetos de Aprendizagem (OAs) digitais contidos no produto educacional são independentes entre si e servem apenas como exemplo básico para futuras implementações e adaptações, por parte dos professores, adequando-os aos objetivos pedagógicos e de planejamento metodológico de sua disciplina de atuação. Salienta-se que não é necessário e nem obrigatório aplicar com os alunos os projetos-exemplo no período vigente da pesquisa.

O professor poderá consultar a BNCC (2017) para adotar uma habilidade que espera que os alunos de determinada etapa da Educação Básica possuam condições de desenvolver por meio dos Objetos de Aprendizagem (OAs) digitais exemplificados nos projetos exemplo contidos no material digital.



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite que outros remixem, adaptem e criem a partir do trabalho para fins não comerciais, desde que atribuam o devido crédito e que licenciem as novas criações sob termos idênticos. Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

1. OBJETIVOS GERAIS

- Compreender e praticar a Programação por Blocos por meio da construção de estruturas lógicas simples ou complexas ajudando a desenvolver os conceitos básicos do Pensamento Computacional (PC);
- Desenvolver os quatro pilares do Pensamento Computacional (PC): abstração, decomposição, reconhecimento de padrões e algoritmos;
- Promover o desenvolvimento da competência Cultura Digital (BNCC, 2017), bem como as suas dimensões e subdimensões em diversas etapas da Educação Básica;
- Criar e manipular mídias exercitando a criatividade e a autoria;
- Compartilhar ideias e projetos de forma ética;
- Desenvolver o raciocínio lógico matemático;
- Despertar o interesse dos educadores com relação ao uso das Tecnologias Digitais Educacionais (TED) no ensino e aprendizagem de conteúdos;
- Utilizar estratégias, conceitos e procedimentos para interpretar situações em diversos contextos;
- Propor soluções ou participar de ações para investigar desafios do mundo contemporâneo;
- Interpretar, construir modelos em diversos contextos, analisando a adequação das soluções propostas;
- Adaptar a metodologia proposta à realidade de cada disciplina e/ou conteúdo escolar;
- **Educação Especial** – Metodologia com o potencial em promover o desenvolvimento das habilidades dos alunos com algum tipo de deficiência, priorizando o atendimento especializado. Possibilidade da elaboração de projetos de Objetos de Aprendizagem (OAs) digitais direcionados a um público-alvo de alunos com algum tipo de deficiência (auditiva, visual, intelectual, física ou múltipla), com distúrbios de aprendizagem ou com altas habilidades;
- **Educação Inclusiva** – Metodologia com o potencial em promover o ensino e a aprendizagem na qual o processo educativo deve ser considerado como um processo social, ou seja, possibilitar a todos os alunos, com deficiência ou não,

total direito à escolarização. Prioridade a uma educação voltada para a formação completa dos alunos e livre de preconceitos, reconhecendo as diferenças e dando a elas o seu devido valor.

2. INTEGRAÇÃO DE HABILIDADES BNCC (2017)

Os projetos-exemplo de Objetos de Aprendizagem (OAs) digitais presentes neste projeto de mestrado pretendem integrar as habilidades de diversas áreas do conhecimento com as habilidades referentes a Matemática e Matemática e suas Tecnologias listadas abaixo:

- **EF06MA04:** construir algoritmo em linguagem natural e representa-lo por fluxograma que indique a resolução de um problema simples
- **EF07MA05:** Resolver um mesmo problema utilizando diferentes algoritmos
- **EF07MA06:** Reconhecer que as resoluções de um grupo de problemas que têm a mesma estrutura pode ser obtidas utilizando os mesmos procedimentos
- **EF07MA07:** Representar por meio de um fluxograma os passos utilizados para resolver um grupo de problemas
- **EF07MA20:** Reconhecer e representar, no plano cartesiano, o simétrico de figuras em relação aos eixos e à origem
- **EF07MA13:** Compreender a ideia de variável, representada por letra ou símbolo, para expressar relação entre duas grandezas, diferenciando-a da ideia de incógnita
- **EF07MA18:** Resolver e elaborar problemas que possam ser representados por equações polinomiais de primeiro grau, redutíveis à forma $ax + b = c$, fazendo uso das propriedades de igualdade
- **EF08MA06:** Resolver e elaborar problemas que envolvam cálculo de valor numérico de expressões algébricas, utilizando as propriedades das operações
- **EF06MA04:** construir algoritmo em linguagem natural e representa-lo por fluxograma que indique a resolução de um problema simples
- **EF07MA05:** Resolver um mesmo problema utilizando diferentes algoritmos
- **EF07MA06:** Reconhecer que as resoluções de um grupo de problemas que têm a mesma estrutura pode ser obtidas utilizando os mesmos procedimentos

- **EF07MA07:** Representar por meio de um fluxograma os passos utilizados para resolver um grupo de problemas
- **EF07MA20:** Reconhecer e representar, no plano cartesiano, o simétrico de figuras em relação aos eixos e à origem
- **EF07MA13:** Compreender a ideia de variável, representada por letra ou símbolo, para expressar relação entre duas grandezas, diferenciando-a da ideia de incógnita
- **EF07MA18:** Resolver e elaborar problemas que possam ser representados por equações polinomiais de primeiro grau, redutíveis à forma $a x + b = c$, fazendo uso das propriedades de igualdade
- **EF08MA06:** Resolver e elaborar problemas que envolvam cálculo de valor numérico de expressões algébricas, utilizando as propriedades das operações
- **EF08MA19:** Resolver e elaborar problemas que envolvam medidas de áreas de figuras geométricas, utilizando expressões de cálculo de área (quadriláteros, triângulos e círculos), em situações como determinar a medida de terrenos
- **EF09MA04:** Resolver e elaborar problemas com números reais, inclusive em notação científica, envolvendo diferentes operações
- **EF09MA06:** Compreender as funções como relações de dependência unívoca entre duas variáveis e suas representações numérica, algébrica e gráfica e utilizar esse conceito para analisar situações que envolvam relações funcionais entre duas variáveis
- **EM13MAT302:** Construir modelos empregando as funções polinomiais de primeiro e segundo graus, para resolver problemas em contextos diversos, com ou sem apoio de tecnologias digitais
- **EM13MAT315:** Investigar e registrar, por meio de um fluxograma, quando possível, um algoritmo que resolve um problema
- **EM13MAT405:** Utilizar conceitos iniciais de uma linguagem de programação na implementação de algoritmos escritos em linguagem corrente e/ou matemática

3. RECURSOS GERAIS

Recursos envolvidos na elaboração deste projeto-exemplo de Objeto de Aprendizagem (OA) digital:

- Programação por Blocos
- Noção de Script
- Noção de Fluxo de Programa
- Manipulação do Ator (Sprite Panda1)
- Manipulação do Cenário (Forest)
- Criação e Manipulação de Variáveis
- Manipulação dos blocos de Variáveis
- Manipulação dos blocos de Eventos
- Manipulação dos blocos de Sensores
- Manipulação dos blocos de Operadores
- Manipulação dos blocos de aparência
- Criação e Manipulação de Imagens Digitalmente por Computador

4. CARACTERÍSTICAS PEDAGÓGICAS

Principais características pedagógicas desejáveis na elaboração deste projeto-exemplo de Objeto de Aprendizagem (OA) digital:

- **Interatividade:** indica se há suporte às consolidações e ações mentais, requerendo que o aluno interaja com o conteúdo do Objeto de Aprendizagem (OA) de alguma forma, podendo ver, escutar ou responder algo;
- **Autonomia:** indica se os Objetos de Aprendizagem (OAs) apoiam a iniciativa e tomada de decisão;
- **Cooperação:** indica se há suporte para aos alunos trocarem opiniões e trabalhar coletivamente sobre o conceito apresentado;
- **Cognição:** refere-se às sobrecargas cognitivas alocadas na memória do aluno durante o processo de ensino-aprendizagem;

- **Afetividade:** refere-se aos sentimentos e motivações do aluno com sua aprendizagem e durante a interação com o Objeto de Aprendizagem.

5. CARACTERÍSTICAS TÉCNICAS

Principais características técnicas desejáveis na elaboração deste projeto-exemplo de Objeto de Aprendizagem (OA) digital:

- **Acessibilidade:** indica se o Objeto de Aprendizagem pode ser acessado por diferentes tipos de usuários (ex.: deficientes visuais e auditivos etc.), em diferentes lugares (ex. lugares com acesso à internet e sem acesso à internet, etc.), e por diferentes tipos de dispositivos (ex. computadores, notebooks, dispositivos móveis, etc.);
- **Agregação:** indica se os componentes do Objeto de Aprendizagem podem ser agrupados em conjuntos maiores de conteúdos (ex. estruturas tradicionais de um curso);
- **Confiabilidade:** indica que o Objeto de Aprendizagem não apresenta defeitos técnicos ou problemas no conteúdo pedagógico;
- **Disponibilidade:** indica se o Objeto de Aprendizagem está disponível para ser utilizado;
- **Durabilidade:** indica se o Objeto de Aprendizagem se mantém intacto quando o repositório em que ele está armazenado muda ou sofre problemas técnicos;
- **Facilidade de instalação:** indica se o Objeto de Aprendizagem pode ser facilmente instalado caso ele exija esse recurso;
- **Granularidade:** é a extensão à qual um Objeto de Aprendizagem é composto por componentes menores e reutilizáveis.
- **Interoperabilidade:** medida do esforço necessário para que os dados dos Objetos de Aprendizagem (OAs) possam ser integrados a vários sistemas;
- **Manutenibilidade:** é a medida de esforço necessária para alterações do Objeto de Aprendizagem (OA);
- **Reusabilidade:** indica as possibilidades de reutilizar os Objetos de Aprendizagem (OAs) em diferentes contextos ou aplicações. Essa é a

principal característica de um Objeto de Aprendizagem e pode ser influenciada por todas as demais.

6. CULTURA DIGITAL BNCC (2017)

Características da competência Cultura Digital BNCC (2017) encontradas na elaboração deste projeto-exemplo de Objeto de Aprendizagem (OA) digital:

- **Computação e Programação** – Utilização de Ferramentas Digitais, Produção de Multimídia, Linguagens de Programação;
- **Pensamento Computacional** – Domínio de Algoritmos e Resolução de Problemas;
- **Cultura e Mundo Digital** – Mundo Digital, Uso Ético.

7. ELABORAÇÃO DO PROJETO-EXEMPLO 2

7.1 Sketch do Design e Objetivos Básicos do Projeto-Exemplo 2

Antes de começar a programação de um novo projeto é importante fazer um sketch para se ter uma ideia geral do design e dos objetivos a serem alcançados. O objetivo principal do projeto-exemplo Classes Gramaticais é muito simples, criar um cenário para que através de comandos de fala, o ator Panda1 (ou qualquer outro personagem), una três classes gramaticais (um artigo, um substantivo e um verbo) para formar uma frase. O ator Panda1 não fará nenhum movimento pelo cenário. O objetivo deste projeto-exemplo é servir como base para futuras implementações e adaptações conforme o professor julgar necessárias para a metodologia de ensino e aprendizagem de determinados conteúdos da sua disciplina em diferentes etapas da Educação Básica.

Figura 1 - Sketch do Projeto-Exemplo Classes Gramaticais

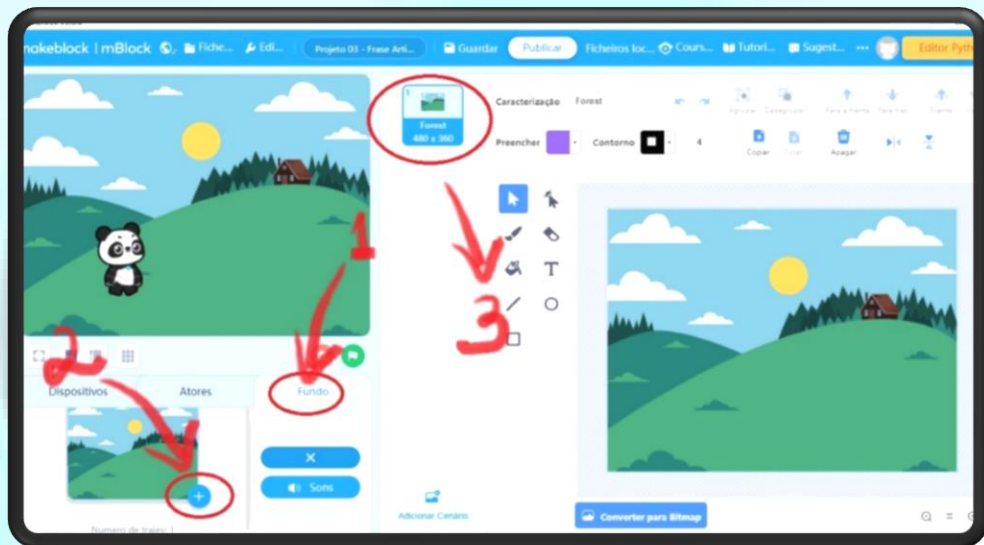


Fonte: Elaborado pelo autor, 2021

7.2 O Palco (Cenário)

- 1 – Ao selecionar a aba **FUNDO** pode-se carregar a imagem do Palco (Cenário);
- 2 – Clicar no símbolo **+** para abrir a Biblioteca de Cenários. Pode-se digitar na caixa de busca pelo cenário **FOREST** ou outro;
- 3 – Ícone indicando que o cenário **FOREST** foi carregado no Editor de Imagem, caso queira fazer edições nesta imagem.

Figura 2 - Configuração do cenário Forest



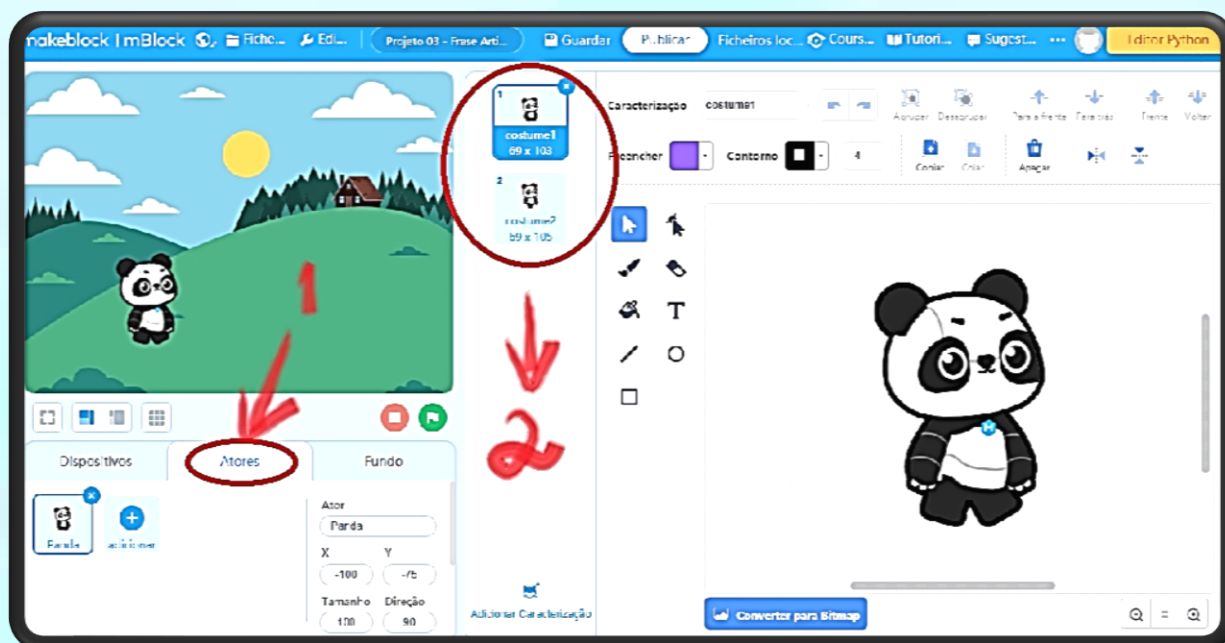
Fonte: Elaborado pelo autor, 2021

7.3 O ator Panda1

1 – Ao selecionar a aba **ATOR** pode-se clicar no símbolo **+** para abrir a Biblioteca de Atores. digitar na caixa de busca pelo sprite do ator **Panda1** ou outro sprite. Nesse momento pode-se editar o sprite do ator Panda se preferir;

2 – Existem duas fantasias para o sprite Panda (**Panda1** e **Panda 2**). Neste exemplo será usada apenas uma fantasia para o ator Panda, pois não existe a necessidade de colocar o Panda em movimento.

Figura 3 - Configuração do ator Panda1

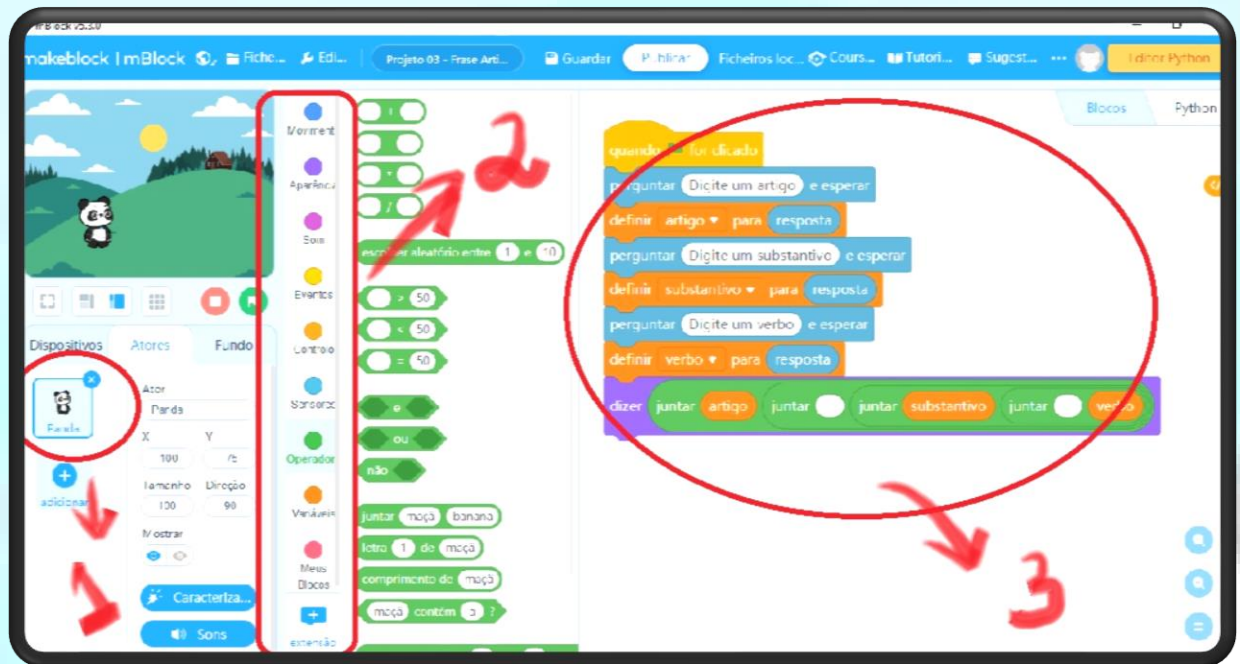


Fonte: Elaborado pelo autor, 2021

7.4 Blocos de código para o ator Panda1

1 – Seleccionamos o ator Panda1 para que se possa atribuir blocos de código a este ator:

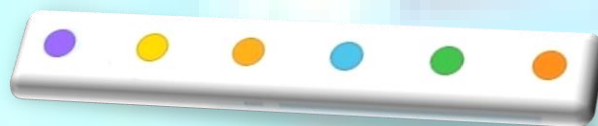
Figura 4 - Blocos de código para o ator Panda1



Fonte: Elaborado pelo autor, 2021

2 – Cor dos blocos usados neste projeto em sequência: **Aparência, Eventos, Controle, Sensores, Operadores e Variáveis.**

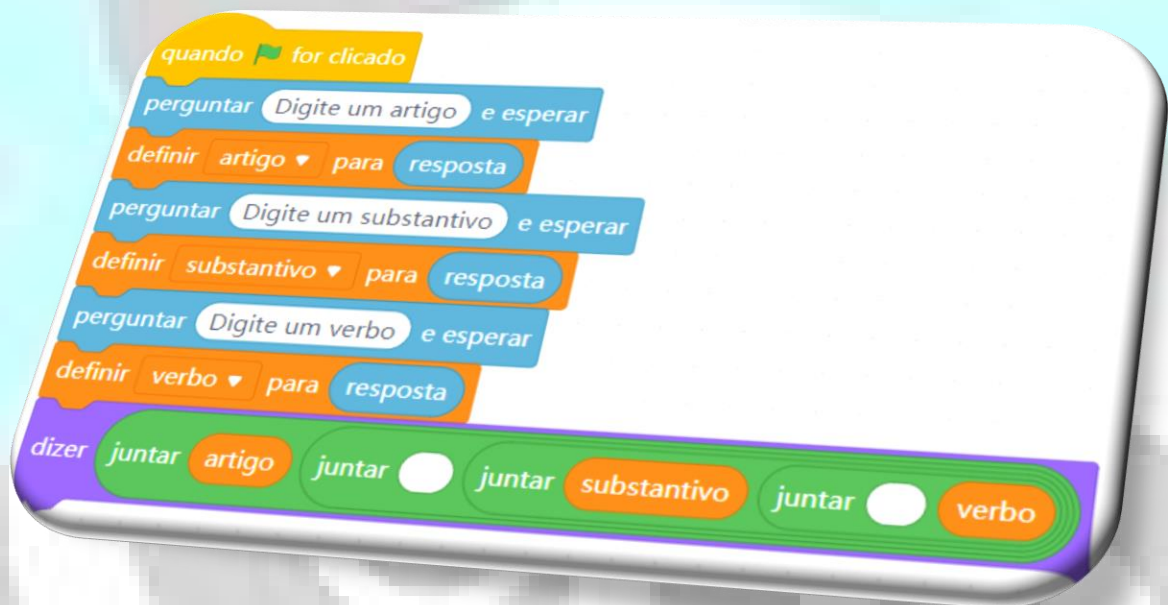
Figura 5 - Cor dos blocos para o ator Panda1



Fonte: Elaborado pelo autor, 2021

3 – Os blocos serão arrastados para a **ÁREA DE SCRIPT** e encaixados de forma que se estabeleça um fluxo lógico e coerente de comandos. Nesta área é que a mágica da programação por blocos visuais acontece!

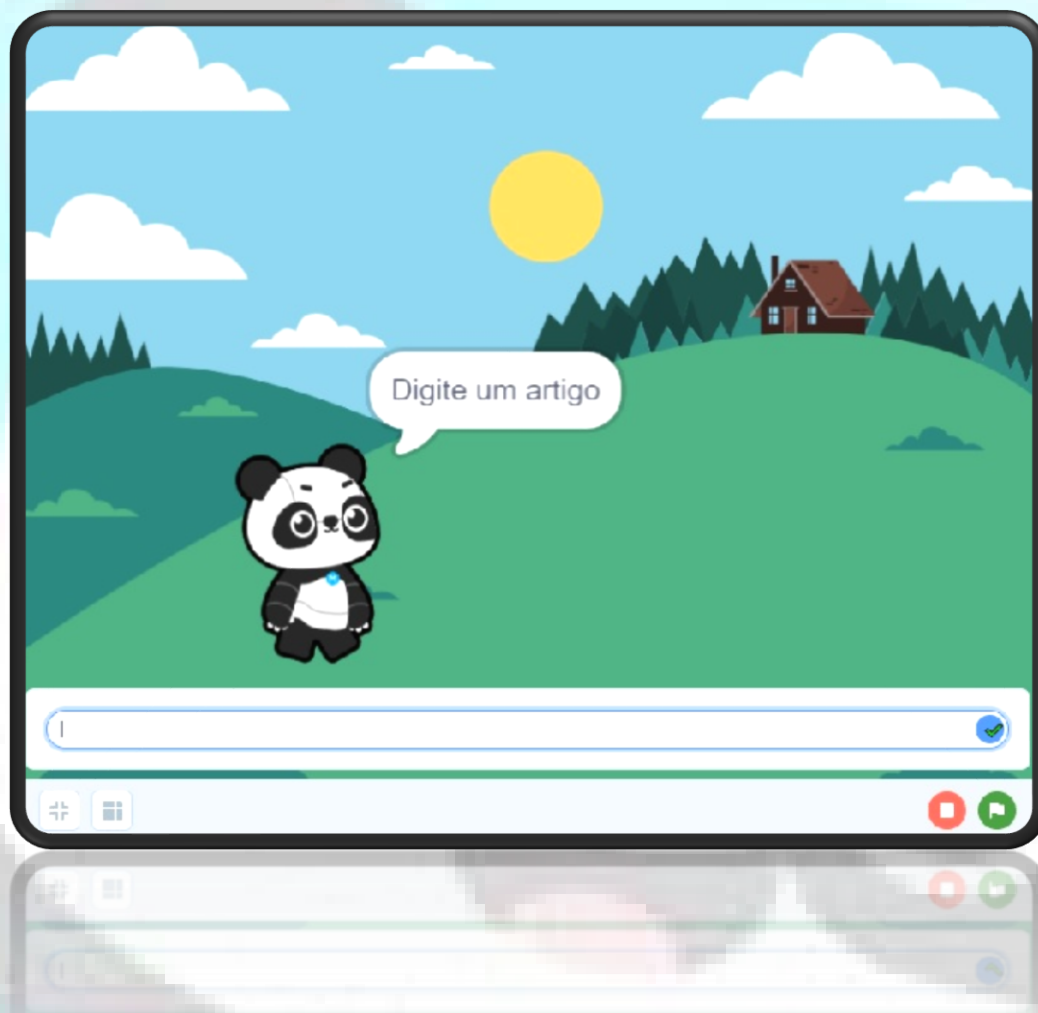
Figura 6 – Encaixe dos blocos para o ator Panda 1



Fonte: Elaborado pelo autor, 2021

7.5 Resultado Final do Projeto-Exemplo Classes Gramaticais

Figura 7 – Projeto-Exemplo Classes Gramaticais



Fonte: Elaborado pelo autor, 2021

8. SUGESTÕES PEDAGÓGICAS E TÉCNICAS PARA ALTERAÇÕES E/OU ADEQUAÇÕES NO PROJETO-EXEMPLO CLASSES GRAMATICAIS

- Professores e alunos podem criar seus próprios sprites dos atores e do cenário no Editor de Imagens do mBlock;
- Incluir tempo limite para o ator Panda1 formar a frase;
- Incluir outras classes gramaticais;
- Incluir outros atores para interagirem com o ator Panda1;
- Possibilidade da portabilidade deste projeto-exemplo para plataformas móveis (Smartphones e Tablets);
- Incluir sons por meio do Editor de Sons e novas cores ao cenário(os) e ator(res);
- Planejar e/ou realizar modificações neste projeto-exemplo focando na **Educação Especial** e na **Educação Inclusiva**.

Críticas, sugestões ou relato de erros serão bem recebidos!

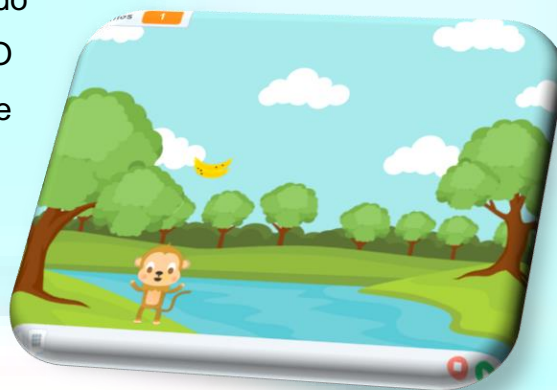
Favor enviar e-mail para Prof. André
endriuscampos@gmail.com

BONS ESTUDOS E ATÉ O PRÓXIMO PROJETO-EXEMPLO!



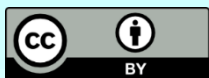
GUIA TÉCNICO E PEDAGÓGICO DO PROJETO-EXEMPLO 3 – PEGUE A FRUTA

O produto educacional deste projeto de mestrado caracteriza-se por meio de um curso online. O curso online versa sobre a elaboração de Objetos de Aprendizagem (OAs) digitais usando a programação por blocos visuais oferecida pela plataforma mBlock. O curso oferece um material digital auto instrutivo (um e-book, cinco videoaulas de aproximadamente vinte minutos de duração cada



uma e também um Guia Técnico Pedagógico específico para cada projeto-exemplo). Este material digital está disponível para leitura e download na plataforma Moodle institucional da UTFPR por meio do link (<https://moodle.utfpr.edu.br/login/index.php>). Não haverá intervenção dos pesquisadores ou encontros virtuais programados para realizar as atividades do curso. O participante poderá utilizar os Fóruns disponíveis na área do curso na plataforma Moodle para postar e compartilhar com os outros cursistas as suas dúvidas e comentários. Os projetos-exemplo de Objetos de Aprendizagem (OAs) digitais contidos no produto educacional são independentes entre si e servem apenas como exemplo básico para futuras implementações e adaptações, por parte dos professores, adequando-os aos objetivos pedagógicos e de planeamento metodológico de sua disciplina de atuação. Salienta-se que não é necessário e nem obrigatório aplicar com os alunos os projetos-exemplo no período vigente da pesquisa.

O professor poderá consultar a BNCC (2017) para adotar uma habilidade que espera que os alunos de determinada etapa da Educação Básica possuam condições de desenvolver por meio dos Objetos de Aprendizagem (OAs) digitais exemplificados nos projetos exemplo contidos no material digital.



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite que outros remixem, adaptem e criem a partir do trabalho para fins não comerciais, desde que atribuam o devido crédito e que licenciem as novas criações sob termos idênticos. Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

1. OBJETIVOS GERAIS

- Compreender e praticar a Programação por Blocos por meio da construção de estruturas lógicas simples ou complexas ajudando a desenvolver os conceitos básicos do Pensamento Computacional (PC);
- Desenvolver os quatro pilares do Pensamento Computacional (PC): abstração, decomposição, reconhecimento de padrões e algoritmos;
- Promover o desenvolvimento da competência Cultura Digital (BNCC, 2017), bem como as suas dimensões e subdimensões em diversas etapas da Educação Básica;
- Criar e manipular mídias exercitando a criatividade e a autoria;
- Compartilhar ideias e projetos de forma ética;
- Desenvolver o raciocínio lógico matemático;
- Despertar o interesse dos educadores com relação ao uso das Tecnologias Digitais Educacionais (TED) no ensino e aprendizagem de conteúdos;
- Utilizar estratégias, conceitos e procedimentos para interpretar situações em diversos contextos;
- Propor soluções ou participar de ações para investigar desafios do mundo contemporâneo;
- Interpretar, construir modelos em diversos contextos, analisando a adequação das soluções propostas;
- Adaptar a metodologia proposta à realidade de cada disciplina e/ou conteúdo escolar;
- **Educação Especial** – Metodologia com o potencial em promover o desenvolvimento das habilidades dos alunos com algum tipo de deficiência, priorizando o atendimento especializado. Possibilidade da elaboração de projetos de Objetos de Aprendizagem (OAs) digitais direcionados a um público-alvo de alunos com algum tipo de deficiência (auditiva, visual, intelectual, física ou múltipla), com distúrbios de aprendizagem ou com altas habilidades;
- **Educação Inclusiva** – Metodologia com o potencial em promover o ensino e a aprendizagem na qual o processo educativo deve ser considerado como um processo social, ou seja, possibilitar a todos os alunos, com deficiência ou não, total direito à escolarização. Prioridade a uma educação voltada para a

formação completa dos alunos e livre de preconceitos, reconhecendo as diferenças e dando a elas o seu devido valor.

2. INTEGRAÇÃO DE HABILIDADES BNCC (2017)

Os projetos-exemplo de Objetos de Aprendizagem (OAs) digitais presentes neste projeto de mestrado pretendem integrar as habilidades de diversas áreas do conhecimento com as habilidades referentes a Matemática e Matemática e suas Tecnologias listadas abaixo:

- **EF06MA04:** construir algoritmo em linguagem natural e representa-lo por fluxograma que indique a resolução de um problema simples
- **EF07MA05:** Resolver um mesmo problema utilizando diferentes algoritmos
- **EF07MA06:** Reconhecer que as resoluções de um grupo de problemas que têm a mesma estrutura pode ser obtidas utilizando os mesmos procedimentos
- **EF07MA07:** Representar por meio de um fluxograma os passos utilizados para resolver um grupo de problemas
- **EF07MA20:** Reconhecer e representar, no plano cartesiano, o simétrico de figuras em relação aos eixos e à origem
- **EF07MA13:** Compreender a ideia de variável, representada por letra ou símbolo, para expressar relação entre duas grandezas, diferenciando-a da ideia de incógnita
- **EF07MA18:** Resolver e elaborar problemas que possam ser representados por equações polinomiais de primeiro grau, redutíveis à forma $a x + b = c$, fazendo uso das propriedades de igualdade
- **EF08MA06:** Resolver e elaborar problemas que envolvam cálculo de valor numérico de expressões algébricas, utilizando as propriedades das operações
- **EF06MA04:** construir algoritmo em linguagem natural e representa-lo por fluxograma que indique a resolução de um problema simples
- **EF07MA05:** Resolver um mesmo problema utilizando diferentes algoritmos
- **EF07MA06:** Reconhecer que as resoluções de um grupo de problemas que têm a mesma estrutura pode ser obtidas utilizando os mesmos procedimentos

- **EF07MA07:** Representar por meio de um fluxograma os passos utilizados para resolver um grupo de problemas
- **EF07MA20:** Reconhecer e representar, no plano cartesiano, o simétrico de figuras em relação aos eixos e à origem
- **EF07MA13:** Compreender a ideia de variável, representada por letra ou símbolo, para expressar relação entre duas grandezas, diferenciando-a da ideia de incógnita
- **EF07MA18:** Resolver e elaborar problemas que possam ser representados por equações polinomiais de primeiro grau, redutíveis à forma $a x + b = c$, fazendo uso das propriedades de igualdade
- **EF08MA06:** Resolver e elaborar problemas que envolvam cálculo de valor numérico de expressões algébricas, utilizando as propriedades das operações
- **EF08MA19:** Resolver e elaborar problemas que envolvam medidas de áreas de figuras geométricas, utilizando expressões de cálculo de área (quadriláteros, triângulos e círculos), em situações como determinar a medida de terrenos
- **EF09MA04:** Resolver e elaborar problemas com números reais, inclusive em notação científica, envolvendo diferentes operações
- **EF09MA06:** Compreender as funções como relações de dependência unívoca entre duas variáveis e suas representações numérica, algébrica e gráfica e utilizar esse conceito para analisar situações que envolvam relações funcionais entre duas variáveis
- **EM13MAT302:** Construir modelos empregando as funções polinomiais de primeiro e segundo graus, para resolver problemas em contextos diversos, com ou sem apoio de tecnologias digitais
- **EM13MAT315:** Investigar e registrar, por meio de um fluxograma, quando possível, um algoritmo que resolve um problema
- **EM13MAT405:** Utilizar conceitos iniciais de uma linguagem de programação na implementação de algoritmos escritos em linguagem corrente e/ou matemática.

3. RECURSOS GERAIS

Recursos envolvidos na elaboração deste projeto-exemplo de Objeto de Aprendizagem (OA) digital:

- Programação por Blocos Visuais;
- Criação e Manipulação de Script;
- Noção sobre Fluxo de Programa;
- Manipulação dos Atores (Sprite Monkey2 e Banana);
- Manipulação de blocos de Eventos;
- Manipulação de blocos de Sensores;
- Manipulação de blocos de Controle;
- Manipulação de blocos de Aparência;
- Manipulação de blocos de Operadores;
- Manipulação do Cenário (Lake3);
- Utilização de Comandos Básicos: Iniciar Script, Movimento (Teclado), Posição (Plano Cartesiano), Rotação, Ângulos;
- Contagem de Pontos;
- Incremento para Pontuação;
- Criação e Manipulação de Imagens Digitais.

4. CARACTERÍSTICAS PEDAGÓGICAS

Principais características pedagógicas desejáveis na elaboração deste projeto-exemplo de Objeto de Aprendizagem (OA) digital:

- **Interatividade:** indica se há suporte às consolidações e ações mentais, requerendo que o aluno interaja com o conteúdo do Objeto de Aprendizagem (OA) de alguma forma, podendo ver, escutar ou responder algo;
- **Autonomia:** indica se os Objetos de Aprendizagem (OAs) apoiam a iniciativa e tomada de decisão;

- **Cooperação:** indica se há suporte para aos alunos trocarem opiniões e trabalhar coletivamente sobre o conceito apresentado;
- **Cognição:** refere-se às sobrecargas cognitivas alocadas na memória do aluno durante o processo de ensino-aprendizagem;
- **Afetividade:** refere-se aos sentimentos e motivações do aluno com sua aprendizagem e durante a interação com o Objeto de Aprendizagem

5. CARACTERÍSTICAS TÉCNICAS

Principais características técnicas desejáveis na elaboração deste projeto-exemplo de Objeto de Aprendizagem (OA) digital:

- **Acessibilidade:** indica se o Objeto de Aprendizagem pode ser acessado por diferentes tipos de usuários (ex.: deficientes visuais e auditivos etc.), em diferentes lugares (ex. lugares com acesso à internet e sem acesso à internet, etc.), e por diferentes tipos de dispositivos (ex. computadores, notebooks, dispositivos móveis, etc.);
- **Agregação:** indica se os componentes do Objeto de Aprendizagem podem ser agrupados em conjuntos maiores de conteúdos (ex. estruturas tradicionais de um curso);
- **Confiabilidade:** indica que o Objeto de Aprendizagem não apresenta defeitos técnicos ou problemas no conteúdo pedagógico;
- **Disponibilidade:** indica se o Objeto de Aprendizagem está disponível para ser utilizado;
- **Durabilidade:** indica se o Objeto de Aprendizagem se mantém intacto quando o repositório em que ele está armazenado muda ou sofre problemas técnicos;
- **Facilidade de instalação:** indica se o Objeto de Aprendizagem pode ser facilmente instalado caso ele exija esse recurso;
- **Granularidade:** é a extensão à qual um Objeto de Aprendizagem é composto por componentes menores e reutilizáveis.
- **Interoperabilidade:** medida do esforço necessário para que os dados dos Objetos de Aprendizagem (OAs) possam ser integrados a vários sistemas;

- **Manutenibilidade:** é a medida de esforço necessária para alterações do Objeto de Aprendizagem (OA);
- **Reusabilidade:** indica as possibilidades de reutilizar os Objetos de Aprendizagem (OAs) em diferentes contextos ou aplicações. Essa é a principal característica de um Objeto de Aprendizagem e pode ser influenciada por todas as demais.

6. CULTURA DIGITAL BNCC (2017)

Características da competência Cultura Digital BNCC (2017) encontradas na elaboração deste projeto-exemplo de Objeto de Aprendizagem (OA) digital:

- **Computação e Programação** – Utilização de Ferramentas Digitais, Produção de Multimídia, Linguagens de Programação;
- **Pensamento Computacional** – Domínio de Algoritmos e Resolução de Problemas;
- **Cultura e Mundo Digital** – Mundo Digital, Uso Ético.

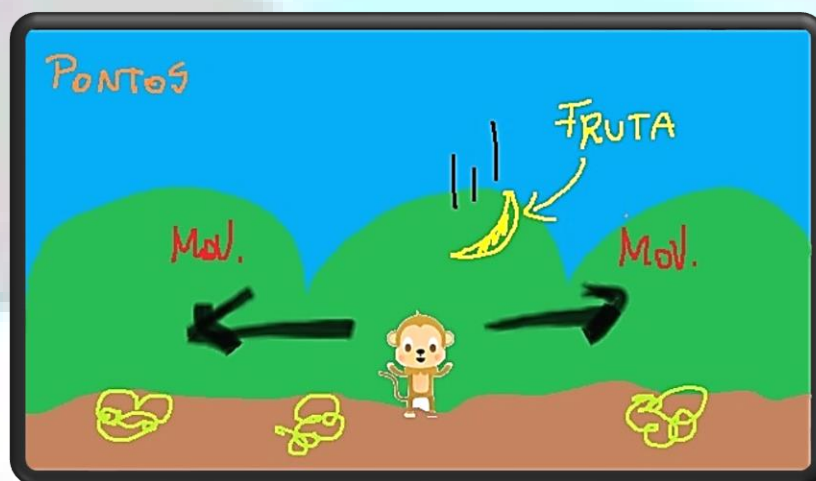
7. ELABORAÇÃO DO PROJETO-EXEMPLO 3

7.1 Sketch do Design e Objetivos Básicos do Projeto-Exemplo 3

Antes de começar a programação de um novo projeto é importante fazer um sketch para se ter uma ideia geral do design e dos objetivos a serem alcançados. O projeto-exemplo **Pegue uma Fruta** é um jogo simples. O objetivo é fazer com que o ator **Monkey2** pegue o maior número de frutas (bananas no caso) possíveis (ator **Banana**). Cada banana colhida pelo ator Monkey2 contará como um ponto no placar. O movimento do ator Monkey2 é unidimensional e horizontal, ou seja, somente esquerda-direita e se dará pelo uso das setas do teclado. O movimento do ator Banana também é unidimensional, mas ocorrerá na vertical com surgimento aleatório no espaço útil do Cenário. Não há limite de tempo e nem pontuação máxima a ser alcançada. O objetivo deste projeto-exemplo é servir como base para futuras implementações e adaptações conforme o professor julgar necessárias para a

metodologia de ensino e aprendizagem de determinados conteúdos da sua disciplina em diferentes etapas da Educação Básica.

Figura 8 - Sketch do Projeto

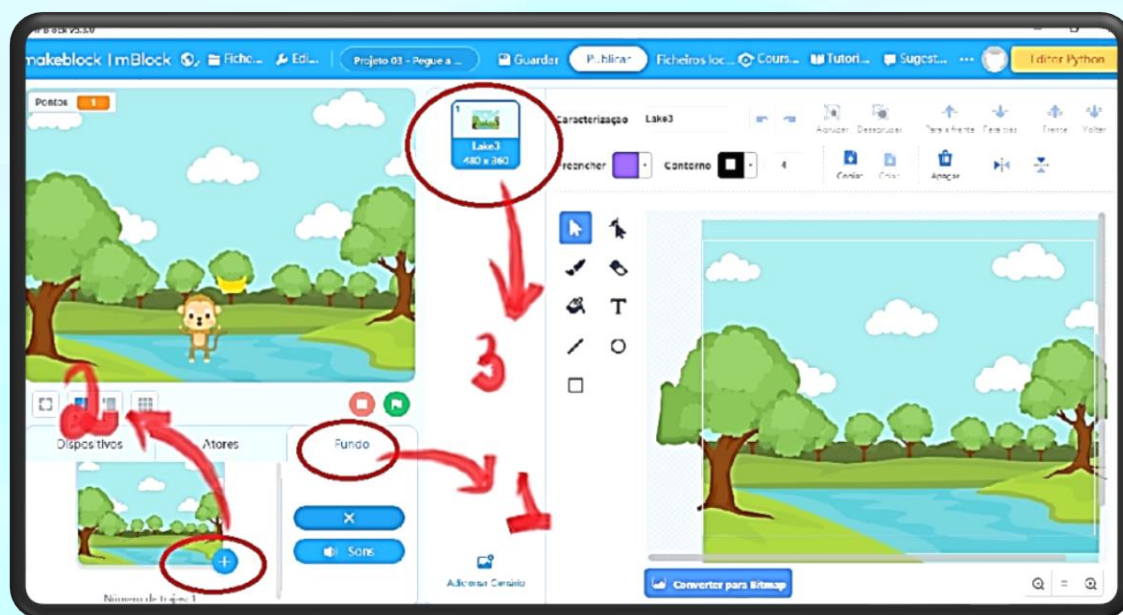


Fonte: Elaborado pelo autor, 2021

7.2 O Palco (Cenário)

- 1 – Clicar na aba **FUNDO** para carregar a imagem do Palco ou Cenário;
- 2 – Clicar no símbolo **+** para abrir a Biblioteca de Cenários. Digitar na caixa de busca pelo cenário **LAKE3** ou outro como desejar;
- 3 – Ícone indicando que o cenário **LAKE3** foi carregado no Editor de Imagem caso queira fazer edições nesta imagem.

Figura 9 - Configuração do cenário Lake3



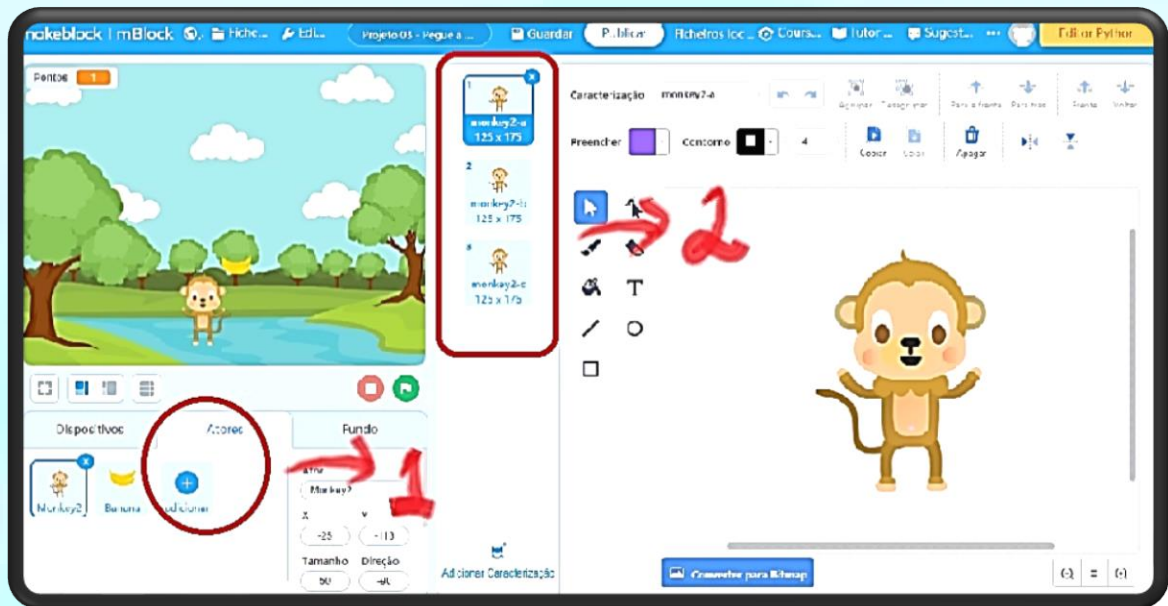
Fonte: Elaborado pelo autor, 2021

7.3 O ator Monkey2

1 – Ao se selecionar a aba **ATOR**, pode-se clicar no símbolo **+** para abrir a Biblioteca de Atores. digitar na caixa de busca pelo sprite do ator **Monkey2** ou outro sprite como desejar. Nesse momento pode-se editar o sprite **Monkey2** como preferir;

2 – Existem três fantasias para o sprite **Monkey2**. Elas poderão ser usadas para dar a impressão de movimento como nos desenhos animados. Para simplificar usaremos apenas uma sprite de **Monkey2** neste projeto.

Figura 10 - Configuração do ator Monkey2

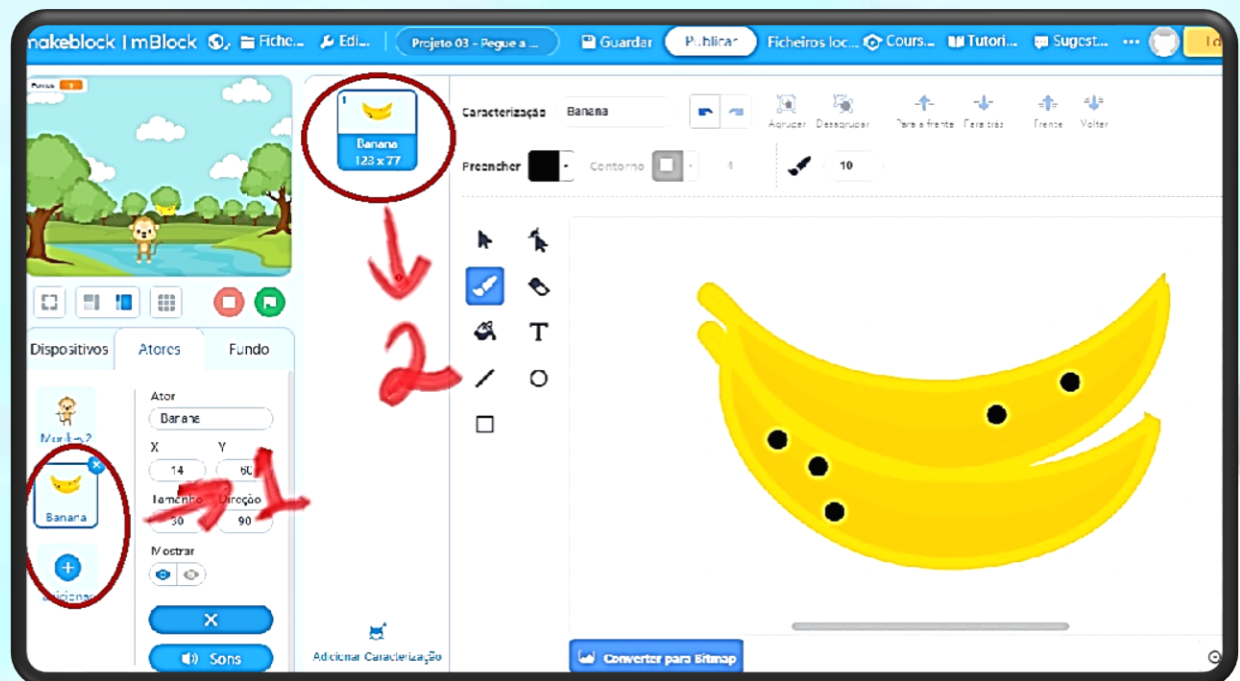


Fonte: Elaborado pelo autor, 2021

7.4 O ator Banana

- 1 – Ao selecionar a aba **ATOR** pode-se clicar no símbolo **+** para abrir a Biblioteca de Atores. Pode-se digitar na caixa de busca pelo sprite do ator **Banana** ou escolher outro sprite. Nesse momento pode-se editar o sprite do ator **Banana** como preferir;
- 2 – Existe apenas uma fantasia para o sprite **Banana**.

Figura 11 - Configuração do ator Banana

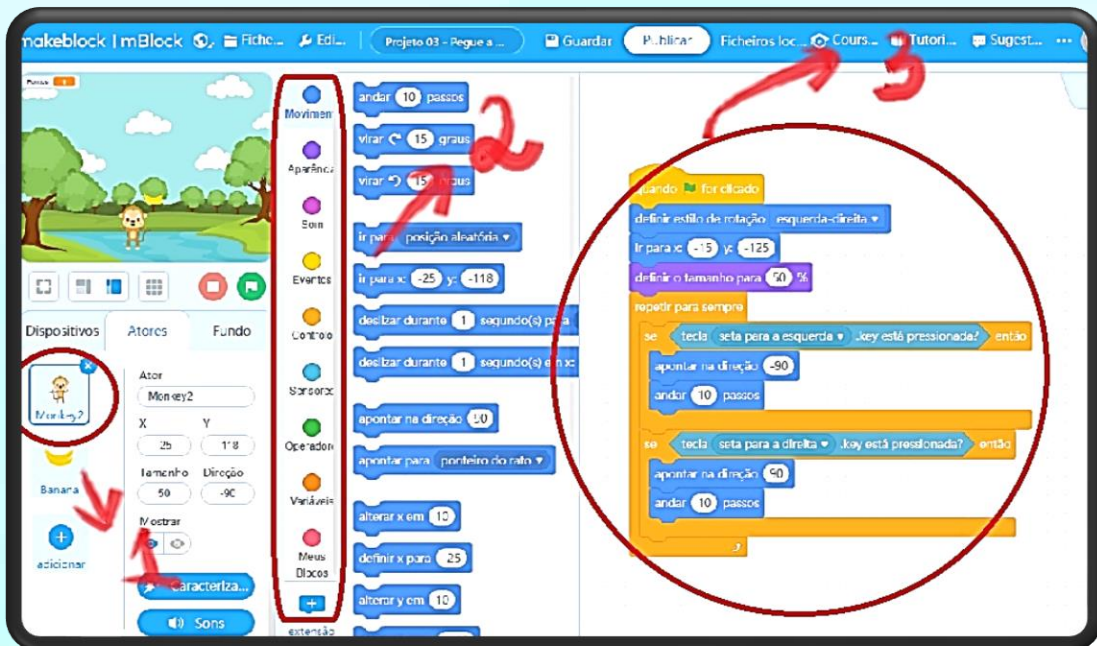


Fonte: Elaborado pelo autor, 2021

7.5 Blocos de código para o ator Monkey2

1 – Selecionamos o ator **Monkey2** para que se possamos atribuir blocos de código a este ator:

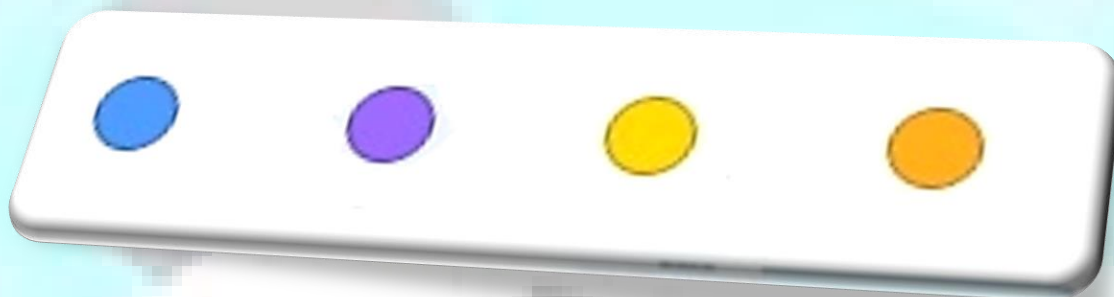
Figura 12 - Blocos de código para o ator Monkey2



Fonte: Elaborado pelo autor, 2021

2 – Cor dos blocos usados no ator **Monkey2** em sequência: **Movimento, Aparência, Eventos e Controle**:

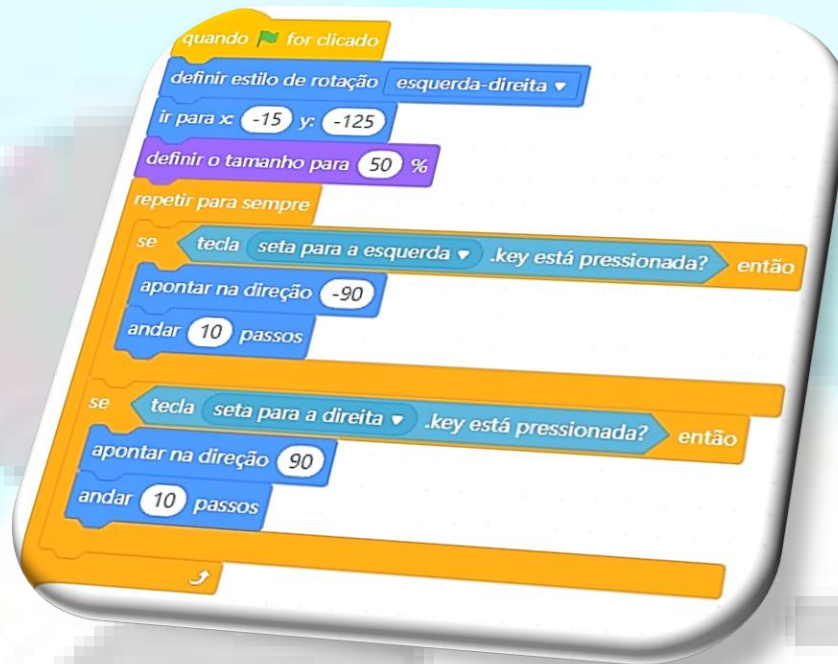
Figura 13 - Cor dos blocos usados no ator Monkey2



Fonte: Elaborado pelo autor, 2021

3 – Os blocos serão arrastados para a **ÁREA DE SCRIPT** e encaixados de forma que se estabeleça um fluxo lógico e coerente de comandos. Nesta área é que a mágica da programação por blocos visuais acontece!

Figura 14 - Encaixe de blocos para o ator Monkey2

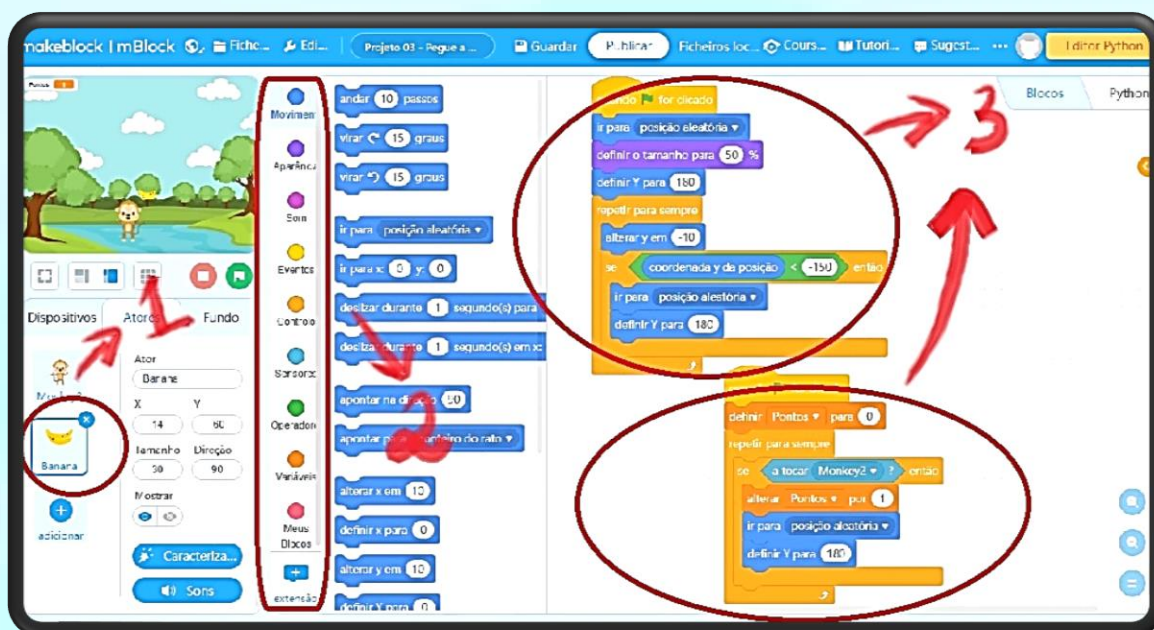


Fonte: Elaborado pelo autor, 2021

7.6 Blocos de código para o ator Banana

1 – Selecionamos o ator **Banana** para que possamos atribuir blocos de código a este ator:

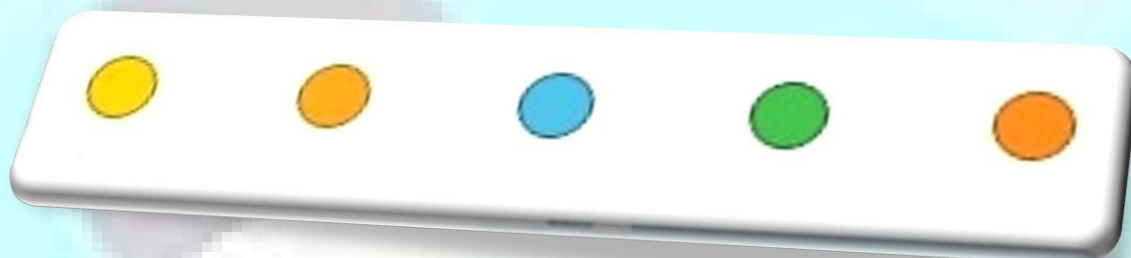
Figura 15 - Blocos de código para o ator Banana



Fonte: Elaborado pelo autor, 2021

2 – Cor dos blocos usados no ator **Banana** em sequência: **Eventos, Controle, Sensores, Operadores e Variáveis**:

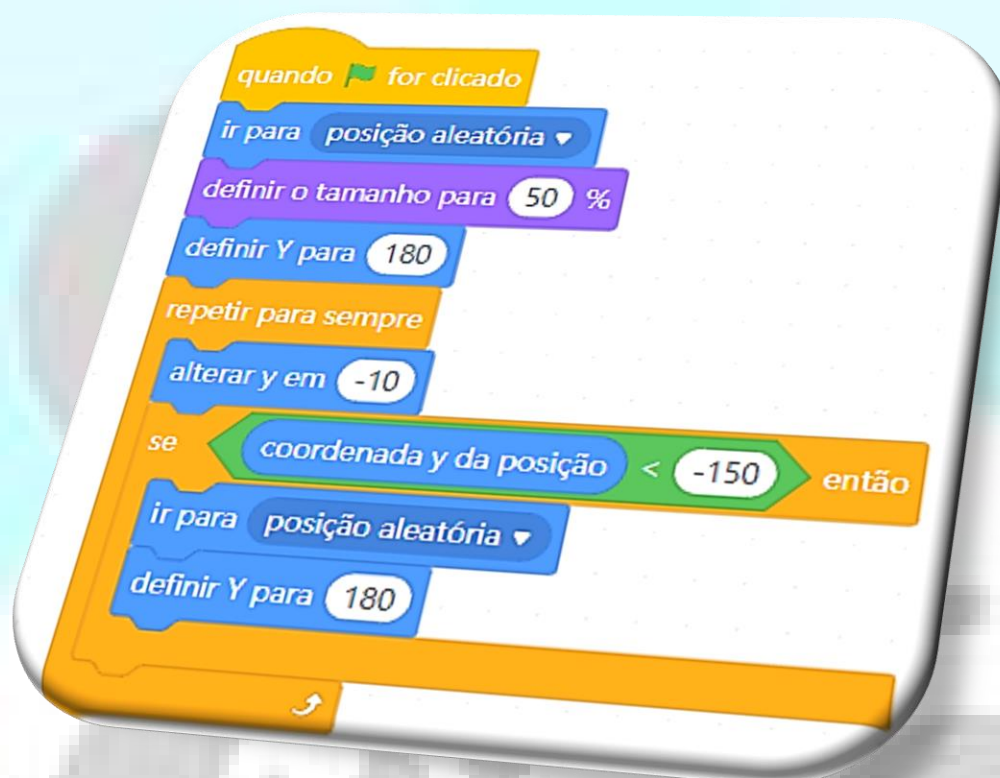
Figura 16 - Cor dos blocos para o ator Banana



Fonte: Elaborado pelo autor, 2021

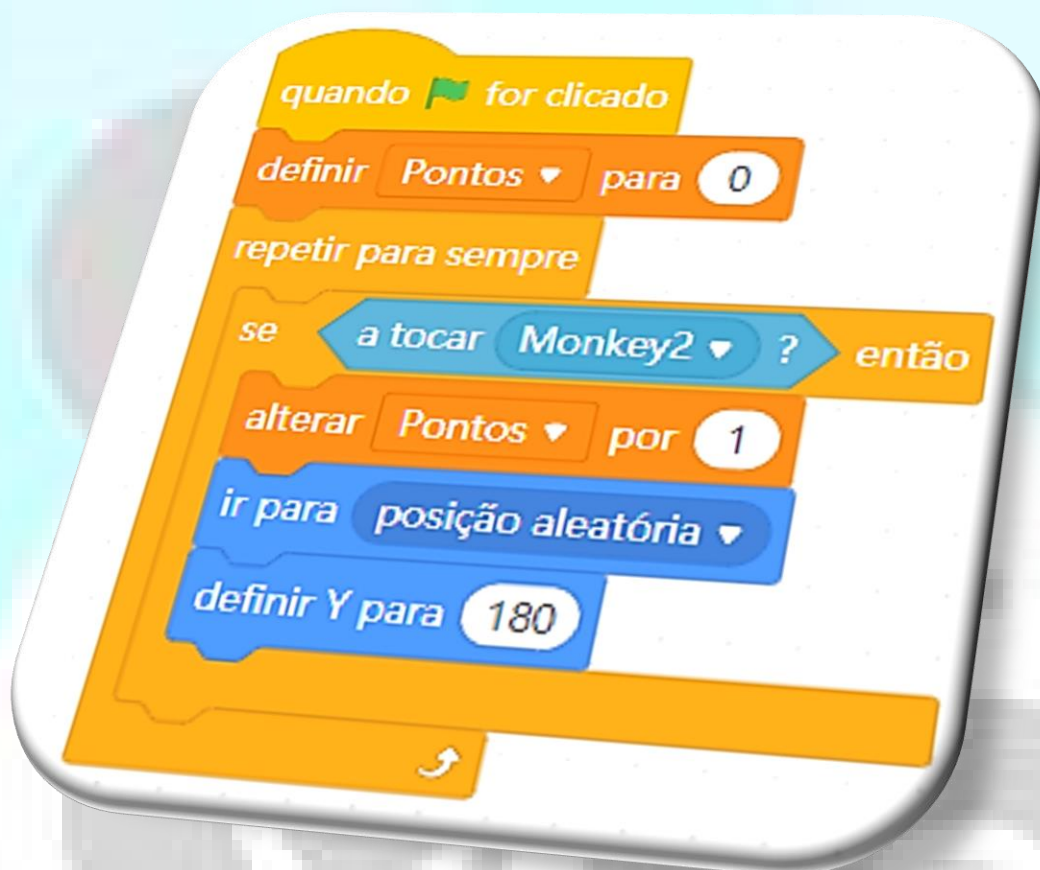
3 – Os blocos serão arrastados para a **ÁREA DE SCRIPT** e encaixados de forma que se estabeleça um fluxo lógico e coerente de comandos. Os blocos foram divididos em duas partes: a primeira parte ilustrada na **Figura 10** comanda o ator **Banana** desde o seu surgimento no cenário e interação com o ator **Monkey2** e a segunda parte ilustrada na **Figura 11** comanda a pontuação quando o ator **Monkey2** interage com o ator **Banana**.

Figura 17 – Primeira Parte: Conjunto de blocos para o ator Banana



Fonte: Elaborado pelo autor, 2021

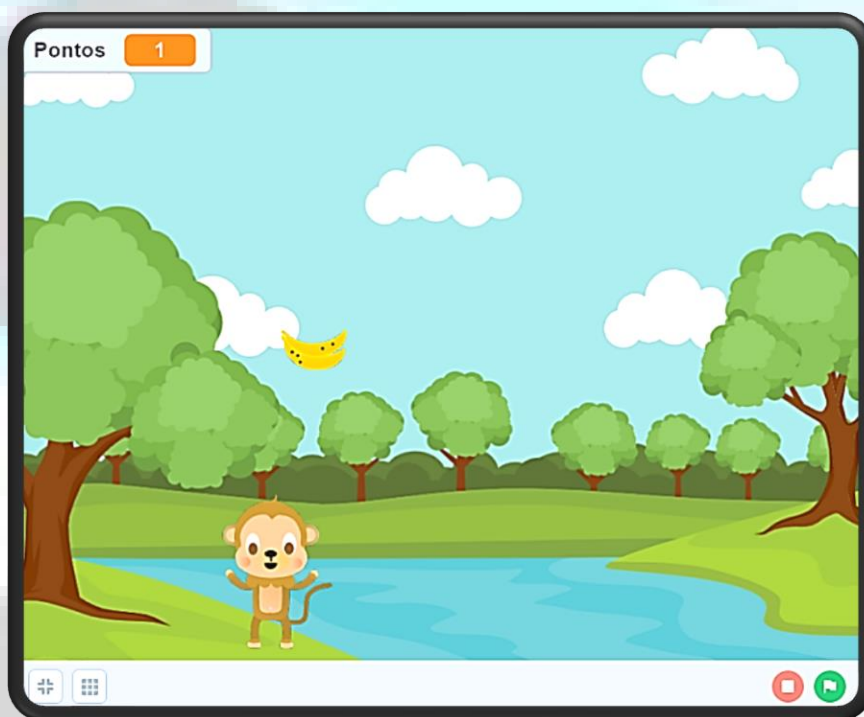
Figura 18 - Segunda Parte: Conjunto de blocos para registrar a Pontuação



Fonte: Elaborado pelo autor, 2021

7.7 Resultado Final do Projeto-Exemplo Pegue a Fruta

Figura 19 - Projeto-Exemplo Pegue a Fruta



Fonte: Elaborado pelo autor, 2021

8. SUGESTÕES PEDAGÓGICAS E TÉCNICAS PARA ALTERAÇÕES E/OU ADEQUAÇÕES NO PROJETO-EXEMPLO PEGUE A FRUTA

- Inclusão de diferentes frutas, por exemplo, maçã, laranja, uva, etc., com diferentes pontuações;
- Inclusão de diferentes alimentos com diferentes pontuações, objetivando mostrar aos alunos uma alimentação mais saudável proporciona maior pontuação;

- Os próprios professores ou alunos poderão criar os sprites dos atores e do cenário no Editor de Imagens do mBlock;
- Incluir tempo limite para o ator Monkey2 colher as frutas ou outros alimentos;
- Incluir número máximo de frutas ou diferentes alimentos para colher;
- Incluir diferentes contextos de ensino e aprendizagem ao projeto, por exemplo, ao invés de colher frutas, o ator Monkey2 poderá colher diferentes números, letras, cores, datas históricas, etc. Dessa forma o projeto poderá privilegiar diversos conteúdos presentes em diferentes etapas da Educação Básica;
- Incluir sons por meio do Editor de Sons e novas cores ao cenário(os) e ator(res);
- Possibilidade da portabilidade deste projeto-exemplo para plataformas móveis (Smartphones e Tablets);
- Planejar e/ou realizar modificações neste projeto-exemplo focando na **Educação Especial** e na **Educação Inclusiva**.

Críticas, sugestões ou relato de erros serão bem recebidos!

Favor enviar e-mail para Prof. André
endriuscampos@gmail.com

BONS ESTUDOS E ATÉ O PRÓXIMO PROJETO-EXEMPLO!



GUIA TÉCNICO E PEDAGÓGICO DO PROJETO-EXEMPLO 4 – JOGO QUIZ

O produto educacional deste projeto de mestrado caracteriza-se por meio de um curso online. O curso online versa sobre a elaboração de Objetos de Aprendizagem (OAs) digitais usando a programação por blocos visuais oferecida pela plataforma mBlock. O curso oferece um material digital auto instrutivo (um e-book, cinco videoaulas de aproximadamente vinte minutos de duração



cada uma e também um Guia Técnico Pedagógico específico para cada projeto-exemplo). Este material digital está disponível para leitura e download na plataforma Moodle institucional da UTFPR por meio do link (<https://moodle.utfpr.edu.br/login/index.php>). Não haverá intervenção dos pesquisadores ou encontros virtuais programados para realizar as atividades do curso. O participante poderá utilizar os Fóruns disponíveis na área do curso na plataforma Moodle para postar e compartilhar com os outros cursistas as suas dúvidas e comentários. Os projetos-exemplo de Objetos de Aprendizagem (OAs) digitais contidos no produto educacional são independentes entre si e servem apenas como exemplo básico para futuras implementações e adaptações, por parte dos professores, adequando-os aos objetivos pedagógicos e de planejamento metodológico de sua disciplina de atuação. Salienta-se que não é necessário e nem obrigatório aplicar com os alunos os projetos-exemplo no período vigente da pesquisa.

O professor poderá consultar a BNCC (2017) para adotar uma habilidade que espera que os alunos de determinada etapa da Educação Básica possam condições de desenvolver por meio dos Objetos de Aprendizagem (OAs) digitais exemplificados nos projetos exemplo contidos no material digital.



[4.0 Internacional](#)

Esta licença permite que outros remixem, adaptem e criem a partir do trabalho para fins não comerciais, desde que atribuam o devido crédito e que licenciem as novas criações sob termos idênticos. Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

1. OBJETIVOS GERAIS

- Compreender e praticar a Programação por Blocos por meio da construção de estruturas lógicas simples ou complexas ajudando a desenvolver os conceitos básicos do Pensamento Computacional (PC);
- Desenvolver os quatro pilares do Pensamento Computacional (PC): abstração, decomposição, reconhecimento de padrões e algoritmos;
- Promover o desenvolvimento da competência Cultura Digital (BNCC, 2017), bem como as suas dimensões e subdimensões em diversas etapas da Educação Básica;
- Criar e manipular mídias exercitando a criatividade e a autoria;
- Compartilhar ideias e projetos de forma ética;
- Desenvolver o raciocínio lógico matemático;
- Despertar o interesse dos educadores com relação ao uso das Tecnologias Digitais Educacionais (TED) no ensino e aprendizagem de conteúdos;
- Utilizar estratégias, conceitos e procedimentos para interpretar situações em diversos contextos;
- Propor soluções ou participar de ações para investigar desafios do mundo contemporâneo;
- Interpretar, construir modelos em diversos contextos, analisando a adequação das soluções propostas;
- Adaptar a metodologia proposta à realidade de cada disciplina e/ou conteúdo escolar;
- **Educação Especial** – Metodologia com o potencial em promover o desenvolvimento das habilidades dos alunos com algum tipo de deficiência, priorizando o atendimento especializado. Possibilidade da elaboração de projetos de Objetos de Aprendizagem (OAs) digitais direcionados a um público-alvo de alunos com algum tipo de deficiência (auditiva, visual, intelectual, física ou múltipla), com distúrbios de aprendizagem ou com altas habilidades;
- **Educação Inclusiva** – Metodologia com o potencial em promover o ensino e a aprendizagem na qual o processo educativo deve ser

considerado como um processo social, ou seja, possibilitar a todos os alunos, com deficiência ou não, total direito à escolarização. Prioridade a uma educação voltada para a formação completa dos alunos e livre de preconceitos, reconhecendo as diferenças e dando a elas o seu devido valor.

2. INTEGRAÇÃO DE HABILIDADES BNCC (2017)

Os projetos-exemplo de Objetos de Aprendizagem (OAs) digitais presentes neste projeto de mestrado pretendem integrar as habilidades de diversas áreas do conhecimento com as habilidades referentes a Matemática e Matemática e suas Tecnologias listadas abaixo:

- **EF06MA04:** construir algoritmo em linguagem natural e representa-lo por fluxograma que indique a resolução de um problema simples
- **EF07MA05:** Resolver um mesmo problema utilizando diferentes algoritmos
- **EF07MA06:** Reconhecer que as resoluções de um grupo de problemas que têm a mesma estrutura pode ser obtidas utilizando os mesmos procedimentos
- **EF07MA07:** Representar por meio de um fluxograma os passos utilizados para resolver um grupo de problemas
- **EF07MA20:** Reconhecer e representar, no plano cartesiano, o simétrico de figuras em relação aos eixos e à origem
- **EF07MA13:** Compreender a ideia de variável, representada por letra ou símbolo, para expressar relação entre duas grandezas, diferenciando-a da ideia de incógnita
- **EF07MA18:** Resolver e elaborar problemas que possam ser representados por equações polinomiais de primeiro grau, redutíveis à forma $a x + b = c$, fazendo uso das propriedades de igualdade
- **EF08MA06:** Resolver e elaborar problemas que envolvam cálculo de valor numérico de expressões algébricas, utilizando as propriedades das operações

- **EF06MA04:** construir algoritmo em linguagem natural e representá-lo por fluxograma que indique a resolução de um problema simples
- **EF07MA05:** Resolver um mesmo problema utilizando diferentes algoritmos
- **EF07MA06:** Reconhecer que as resoluções de um grupo de problemas que têm a mesma estrutura pode ser obtidas utilizando os mesmos procedimentos
- **EF07MA07:** Representar por meio de um fluxograma os passos utilizados para resolver um grupo de problemas
- **EF07MA20:** Reconhecer e representar, no plano cartesiano, o simétrico de figuras em relação aos eixos e à origem
- **EF07MA13:** Compreender a ideia de variável, representada por letra ou símbolo, para expressar relação entre duas grandezas, diferenciando-a da ideia de incógnita
- **EF07MA18:** Resolver e elaborar problemas que possam ser representados por equações polinomiais de primeiro grau, redutíveis à forma $a x + b = c$, fazendo uso das propriedades de igualdade
- **EF08MA06:** Resolver e elaborar problemas que envolvam cálculo de valor numérico de expressões algébricas, utilizando as propriedades das operações
- **EF08MA19:** Resolver e elaborar problemas que envolvam medidas de áreas de figuras geométricas, utilizando expressões de cálculo de área (quadriláteros, triângulos e círculos), em situações como determinar a medida de terrenos
- **EF09MA04:** Resolver e elaborar problemas com números reais, inclusive em notação científica, envolvendo diferentes operações
- **EF09MA06:** Compreender as funções como relações de dependência unívoca entre duas variáveis e suas representações numérica, algébrica e gráfica e utilizar esse conceito para analisar situações que envolvam relações funcionais entre duas variáveis
- **EM13MAT302:** Construir modelos empregando as funções polinomiais de primeiro e segundo graus, para resolver problemas em contextos diversos, com ou sem apoio de tecnologias digitais

- **EM13MAT315:** Investigar e registrar, por meio de um fluxograma, quando possível, um algoritmo que resolve um problema
- **EM13MAT405:** Utilizar conceitos iniciais de uma linguagem de programação na implementação de algoritmos escritos em linguagem corrente e/ou matemática

3. RECURSOS GERAIS

Recursos envolvidos na elaboração deste projeto-exemplo de Objeto de Aprendizagem (OA) digital:

- Programação por Blocos Visuais
- Criação e Manipulação de Script
- Noção sobre Fluxo de Programa
- Manipulação do Ator (Sprite Boy19)
- Manipulação de blocos de Eventos
- Manipulação de blocos de Sensores
- Manipulação de blocos de Controle
- Manipulação de blocos de Aparência
- Manipulação de blocos Operadores
- Manipulação de blocos de Variáveis
- Manipulação de blocos de Sons
- Manipulação do Cenário (Classroom3)
- Criação e Manipulação Digitalmente de Imagens

4. CARACTERÍSTICAS PEDAGÓGICAS

Principais características pedagógicas desejáveis na elaboração deste projeto-exemplo de Objeto de Aprendizagem (OA) digital:

- **Interatividade:** indica se há suporte às consolidações e ações mentais, requerendo que o aluno interaja com o conteúdo do Objeto de

Aprendizagem (OA) de alguma forma, podendo ver, escutar ou responder algo;

- **Autonomia:** indica se os Objetos de Aprendizagem (OAs) apoiam a iniciativa e tomada de decisão;
- **Cooperação:** indica se há suporte para aos alunos trocarem opiniões e trabalhar coletivamente sobre o conceito apresentado;
- **Cognição:** refere-se às sobrecargas cognitivas alocadas na memória do aluno durante o processo de ensino-aprendizagem;
- **Afetividade:** refere-se aos sentimentos e motivações do aluno com sua aprendizagem e durante a interação com o Objeto de Aprendizagem.

5. CARACTERÍSTICAS TÉCNICAS

Principais características técnicas desejáveis na elaboração deste projeto-exemplo de Objeto de Aprendizagem (OA) digital:

- **Acessibilidade:** indica se o Objeto de Aprendizagem pode ser acessado por diferentes tipos de usuários (ex.: deficientes visuais e auditivos etc.), em diferentes lugares (ex. lugares com acesso à internet e sem acesso à internet, etc.), e por diferentes tipos de dispositivos (ex. computadores, notebooks, dispositivos móveis, etc.);
- **Agregação:** indica se os componentes do Objeto de Aprendizagem podem ser agrupados em conjuntos maiores de conteúdos (ex. estruturas tradicionais de um curso);
- **Confiabilidade:** indica que o Objeto de Aprendizagem não apresenta defeitos técnicos ou problemas no conteúdo pedagógico;
- **Disponibilidade:** indica se o Objeto de Aprendizagem está disponível para ser utilizado;
- **Durabilidade:** indica se o Objeto de Aprendizagem se mantém intacto quando o repositório em que ele está armazenado muda ou sofre problemas técnicos;

- **Facilidade de instalação:** indica se o Objeto de Aprendizagem pode ser facilmente instalado caso ele exija esse recurso;
- **Granularidade:** é a extensão à qual um Objeto de Aprendizagem é composto por componentes menores e reutilizáveis.
- **Interoperabilidade:** medida do esforço necessário para que os dados dos Objetos de Aprendizagem (OAs) possam ser integrados a vários sistemas;
- **Manutenibilidade:** é a medida de esforço necessária para alterações do Objeto de Aprendizagem (OA);
- **Reusabilidade:** indica as possibilidades de reutilizar os Objetos de Aprendizagem (OAs) em diferentes contextos ou aplicações. Essa é a principal característica de um Objeto de Aprendizagem e pode ser influenciada por todas as demais.

6. CULTURA DIGITAL BNCC (2017)

Características da competência Cultura Digital BNCC (2017) encontradas na elaboração deste projeto-exemplo de Objeto de Aprendizagem (OA) digital:

- **Computação e Programação** – Utilização de Ferramentas Digitais, Produção de Multimídia, Linguagens de Programação;
- **Pensamento Computacional** – Domínio de Algoritmos e Resolução de Problemas;
- **Cultura e Mundo Digital** – Mundo Digital, Uso Ético.

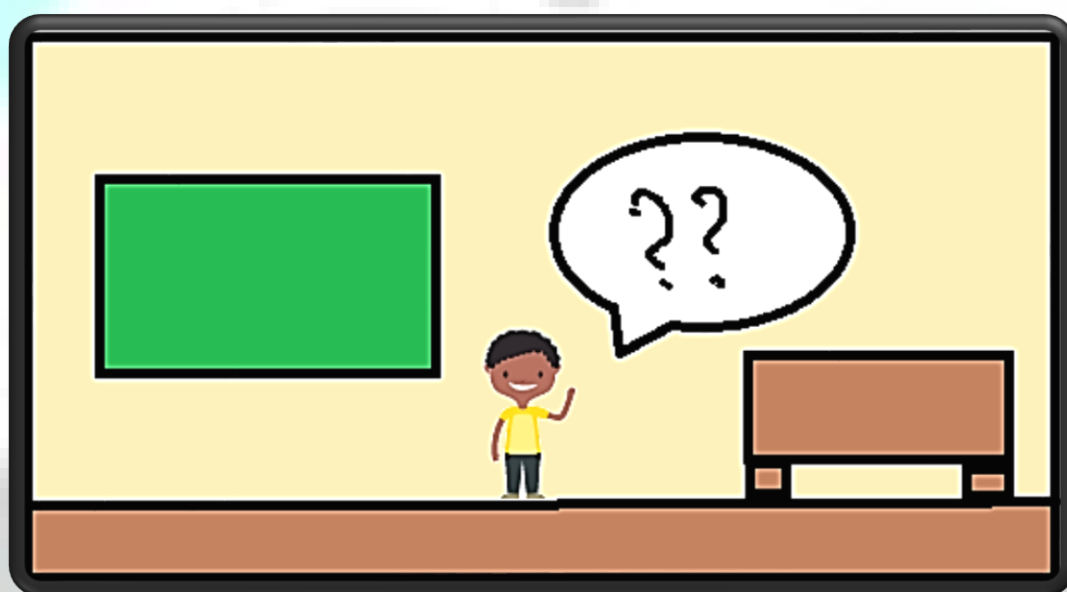
7. ELABORAÇÃO DO PROJETO-EXEMPLO 4

7.1 Sketch do Design e Objetivos Básicos do Projeto-Exemplo 4

Antes de começar a programação de um novo projeto é importante fazer um sketch para se ter uma ideia geral do design e dos objetivos a serem alcançados. O projeto-exemplo **JOGO QUIZ** é intermediário e servirá para o professor estabelecer contato com recursos operacionais e didáticos mais elaborados da

plataforma mBlock. O objetivo deste projeto é criar um cenário para o personagem de um menino interagir fazendo algumas perguntas que deverão ser respondidas pelo usuário. O menino é o único ator deste projeto-exemplo. Não há movimento do ator menino pelo cenário. Não há limite de tempo, mas existe o registro de pontos para cada pergunta respondida corretamente. O objetivo deste projeto-exemplo é servir como base para futuras implementações e adaptações conforme o professor julgar necessário para a metodologia de ensino e aprendizagem de certos conteúdos de sua disciplina para diferentes etapas da Educação Básica.

Figura 20 - Sketch do Projeto JOGO QUIZ

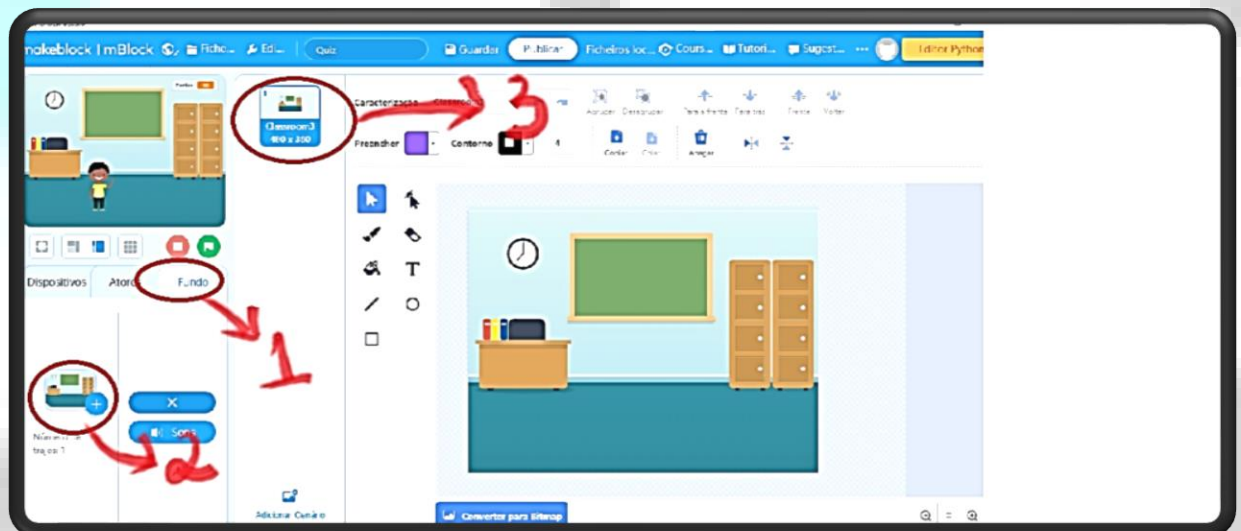


Fonte: Elaborado pelo autor, 2021

7.2 O Palco (Cenário)

- 1 – Ao selecionar a aba **FUNDO** pode-se carregar a imagem do Palco (Cenário)
- 2 – Clicar no símbolo **+** permite abrir a Biblioteca de Cenários. Pode-se digitar na caixa de busca pelo cenário **CLASSROOM3** ou outro cenário.
- 3 – Ícone indicando que o cenário **CLASSROOM3** foi carregado no Editor de Imagem, caso queira fazer edições nesta imagem.

Figura 21 - Configuração do cenário Classroom3



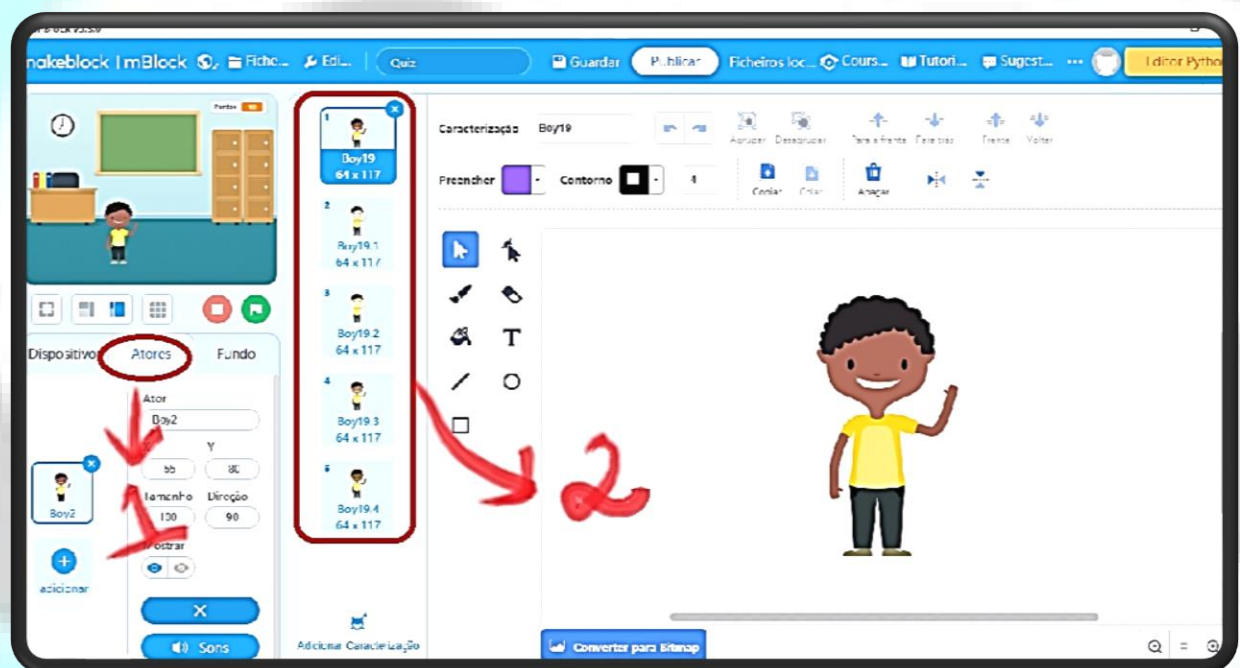
Fonte: Elaborado pelo autor, 2021

7.3 O ator Boy19

1 – Ao selecionar a aba **ATOR** pode-se clicar no símbolo **+** para abrir a Biblioteca de Atores. digitar na caixa de busca pelo sprite do ator **Boy19** ou outro sprite. Nesse momento pode-se editar o sprite do ator Panda se preferir;

2 – Existem quatro fantasias para o sprite **Boy19** (**Boy19.1**, **Boy19.2**, **Boy19.3**, **Boy19.4**). Somente a sprite **Boy19** será usada neste projeto pois o personagem não precisará se deslocar pelo cenário. As outras sprites podem ser deletadas.

Figura 22 - Configuração do ator Boy19

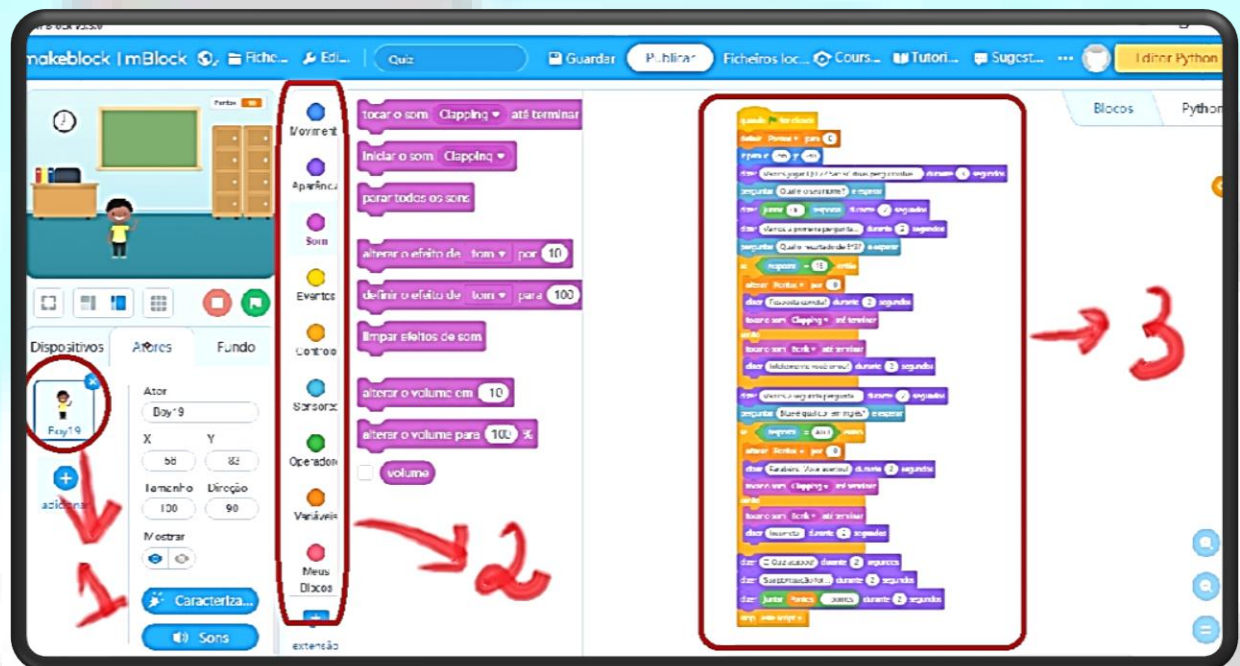


Fonte: Elaborado pelo autor, 2021

7.4 Blocos de código para o ator Boy19

1 – Selecionamos o ator Boy19 para que se possa atribuir blocos de código a este ator:

Figura 23 - Blocos de código para o ator Boy19



Fonte: Elaborado pelo autor, 2021

2 – Cor dos blocos usados para o ator Panda foram em sequência: **Movimento, Aparência, Som, Eventos, Controle, Sensores, Operadores e Variáveis.**

Figura 24 - Cor dos blocos para o ator Boy19



Fonte: Elaborado pelo autor, 2021

3 – Os blocos serão arrastados para a **ÁREA DE SCRIPT** e encaixados de forma que se estabeleça um fluxo lógico e coerente de comandos. Nesta área é que a mágica da programação por blocos visuais acontece!

Figura 25 - Encaixe de blocos para o ator Boy19

```
quando for clicado
definir Pontos para 0
ir para x: -56 y: -83
dizer Vamos jogar QUIZ? São só duas perguntinhas ... durante 3 segundos
perguntar Qual é o seu nome? e esperar
dizer juntar Oi resposta durante 2 segundos
dizer Vamos a primeira pergunta ... durante 2 segundos
perguntar Qual o resultado de 5*3? e esperar
se resposta = 15 então
  alterar Pontos por 10
  dizer Resposta correta! durante 2 segundos
  tocar o som Clapping até terminar
senão
  tocar o som Bonk até terminar
  dizer Infelizmente você errou! durante 2 segundos
dizer Vamos a segunda pergunta ... durante 2 segundos
perguntar Blue é qual cor em inglês? e esperar
se resposta = Azul então
  alterar Pontos por 10
  dizer Parabéns! Você acertou! durante 2 segundos
  tocar o som Clapping até terminar
senão
  tocar o som Bonk até terminar
  dizer Incorreto! durante 2 segundos
dizer O Quiz acabou! durante 2 segundos
dizer Sua pontuação foi ... durante 2 segundos
dizer juntar Pontos pontos! durante 2 segundos
stop este script
```

7.5 Resultado Final do Projeto-Exemplo JOGO QUIZ

Figura 26 - Projeto-Exemplo JOGO QUIZ



Fonte: Elaborado pelo autor, 2021

8. SUGESTÕES PEDAGÓGICAS E TÉCNICAS PARA ALTERAÇÕES E/OU ADEQUAÇÕES NO PROJETO-EXEMPLO JOGO QUIZ

- Professores e alunos podem criar seus próprios sprites do ator e do cenário no Editor de Imagens do mBlock;
- Incluir tempo limite para que o usuário responda as perguntas;
- Incluir perguntas específicas da sua disciplina de atuação;
- Interromper o JOGO QUIZ caso ocorram três respostas erradas;
- Estimular os alunos a elaborarem as perguntas e respostas do JOGO QUIZ sobre determinado conteúdo escolar;
- A cada pergunta respondida corretamente o ator dá alguns passos à frente ganhando pontos, caso contrário retrocede e perde pontos;
- Implementar o JOGO QUIZ para ser jogado com mais jogadores;
- Possibilidade da portabilidade deste projeto-exemplo para plataformas móveis (Smartphones e Tablets);
- Incluir sons por meio do Editor de Sons e novas cores ao cenário(os) e ator(res);
- Planejar e/ou realizar modificações neste projeto-exemplo focando na **Educação Especial** e na **Educação Inclusiva**.

Críticas, sugestões ou relato de erros serão bem recebidos!

Favor enviar e-mail para Prof. André

endriuscampos@gmail.com

BONS ESTUDOS E ATÉ O PRÓXIMO PROJETO-EXEMPLO!



PROJETOS-EXTRA DE OBJETOS DE APRENDIZAGEM DIGITAIS (OAs) ELABORADOS USANDO A PROGRAMAÇÃO POR BLOCOS DO MBLOCK

PROJETO DE MESTRADO – UMA PROPOSTA METODOLÓGICA PARA O DESENVOLVIMENTO DE OBJETOS DE APRENDIZAGEM DIGITAIS USANDO PROGRAMAÇÃO POR BLOCOS

Prof. André de Campos

Esta coleção de nove Projetos-Extra de Objetos de Aprendizagem Digitais tem o propósito de servir como material de apoio ao professor cursista. Nestes projetos podem ser encontrados diversos exemplos do uso da programação por blocos em diferentes situações. O professor poderá adequá-los e inseri-los em sua metodologia de aula da forma que entender ser a mais adequada sem esquecer de priorizar o ensino e a aprendizagem dos conteúdos escolares. Poderá também realizar modificações nestes projetos-exemplo focando na Educação Especial e na Educação Inclusiva.



Importante: Os nove projetos-extra estão disponibilizados por meio do seu código-fonte compactado. Os códigos-fonte poderão ser abertos e estudados usando o software mBlock. Todos os projetos-extras podem ser adaptados para plataformas móveis (Smartphones e Tablets).

Críticas, sugestões ou relato de erros serão bem recebidos!

Favor enviar e-mail para Prof. André

andriuscampos@gmail.com

1. INTRODUÇÃO

O produto educacional deste projeto de mestrado caracteriza-se por meio de um curso online. O curso online versa sobre a elaboração de Objetos de Aprendizagem (OAs) digitais usando a programação por blocos visuais oferecida pela plataforma mBlock. O curso oferece um material digital auto instrutivo (um e-book, cinco videoaulas de aproximadamente vinte minutos de duração cada uma e também um Guia Técnico Pedagógico específico para cada projeto-exemplo). Este material digital está disponível para leitura e download na plataforma Moodle institucional da UTFPR por meio do link (<https://moodle.utfpr.edu.br/login/index.php>). Não haverá intervenção dos pesquisadores ou encontros virtuais programados para realizar as atividades do curso. O participante poderá utilizar os Fóruns disponíveis na área do curso na plataforma Moodle para postar e compartilhar com os outros cursistas as suas dúvidas e comentários. Os projetos-exemplo de Objetos de Aprendizagem (OAs) digitais contidos no produto educacional são independentes entre si e servem apenas como exemplo básico para futuras implementações e adaptações, por parte dos professores, adequando-os aos objetivos pedagógicos e de planejamento metodológico de sua disciplina de atuação. Salienta-se que não é necessário e nem obrigatório aplicar com os alunos os projetos-exemplo no período vigente da pesquisa.

O professor poderá consultar a BNCC (2017) para adotar uma habilidade que espera que os alunos de determinada etapa da Educação Básica possam condições de desenvolver por meio dos Objetos de Aprendizagem (OAs) digitais exemplificados nos projetos exemplo contidos no material digital.



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite que outros remixem, adaptem e criem a partir do trabalho para fins não comerciais, desde que atribuam o devido crédito e que licenciem as novas criações sob termos idênticos. Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

2. OBJETIVOS GERAIS

- Compreender e praticar a Programação por Blocos por meio da construção de estruturas lógicas simples ou complexas ajudando a desenvolver os conceitos básicos do Pensamento Computacional (PC);
- Desenvolver os quatro pilares do Pensamento Computacional (PC): abstração, decomposição, reconhecimento de padrões e algoritmos;
- Promover o desenvolvimento da competência Cultura Digital (BNCC, 2017), bem como as suas dimensões e subdimensões em diversas etapas da Educação Básica;
- Criar e manipular mídias exercitando a criatividade e a autoria;
- Compartilhar ideias e projetos de forma ética;
- Desenvolver o raciocínio lógico matemático;
- Despertar o interesse dos educadores com relação ao uso das Tecnologias Digitais Educacionais (TED) no ensino e aprendizagem de conteúdos;
- Utilizar estratégias, conceitos e procedimentos para interpretar situações em diversos contextos;
- Propor soluções ou participar de ações para investigar desafios do mundo contemporâneo;
- Interpretar, construir modelos em diversos contextos, analisando a adequação das soluções propostas;
- Adaptar a metodologia proposta à realidade de cada disciplina e/ou conteúdo escolar;
- **Educação Especial** – Metodologia com o potencial em promover o desenvolvimento das habilidades dos alunos com algum tipo de deficiência, priorizando o atendimento especializado. Possibilidade da elaboração de projetos de Objetos de Aprendizagem (OAs) digitais direcionados a um público-alvo de alunos com algum tipo de deficiência (auditiva, visual, intelectual, física ou múltipla), com distúrbios de aprendizagem ou com altas habilidades;
- **Educação Inclusiva** – Metodologia com o potencial em promover o ensino e a aprendizagem na qual o processo educativo deve ser

considerado como um processo social, ou seja, possibilitar a todos os alunos, com deficiência ou não, total direito à escolarização. Prioridade a uma educação voltada para a formação completa dos alunos e livre de preconceitos, reconhecendo as diferenças e dando a elas o seu devido valor.

3. INTEGRAÇÃO DE HABILIDADES BNCC (2017)

Os projetos-exemplo de Objetos de Aprendizagem (OAs) digitais presentes neste projeto de mestrado pretendem integrar as habilidades de diversas áreas do conhecimento com as habilidades referentes a Matemática e Matemática e suas Tecnologias listadas abaixo:

- **EF06MA04:** construir algoritmo em linguagem natural e representa-lo por fluxograma que indique a resolução de um problema simples
- **EF07MA05:** Resolver um mesmo problema utilizando diferentes algoritmos
- **EF07MA06:** Reconhecer que as resoluções de um grupo de problemas que têm a mesma estrutura pode ser obtidas utilizando os mesmos procedimentos
- **EF07MA07:** Representar por meio de um fluxograma os passos utilizados para resolver um grupo de problemas
- **EF07MA20:** Reconhecer e representar, no plano cartesiano, o simétrico de figuras em relação aos eixos e à origem
- **EF07MA13:** Compreender a ideia de variável, representada por letra ou símbolo, para expressar relação entre duas grandezas, diferenciando-a da ideia de incógnita
- **EF07MA18:** Resolver e elaborar problemas que possam ser representados por equações polinomiais de primeiro grau, redutíveis à forma $a x + b = c$, fazendo uso das propriedades de igualdade
- **EF08MA06:** Resolver e elaborar problemas que envolvam cálculo de valor numérico de expressões algébricas, utilizando as propriedades das operações

- **EF06MA04:** construir algoritmo em linguagem natural e representa-lo por fluxograma que indique a resolução de um problema simples
- **EF07MA05:** Resolver um mesmo problema utilizando diferentes algoritmos
- **EF07MA06:** Reconhecer que as resoluções de um grupo de problemas que têm a mesma estrutura pode ser obtidas utilizando os mesmos procedimentos
- **EF07MA07:** Representar por meio de um fluxograma os passos utilizados para resolver um grupo de problemas
- **EF07MA20:** Reconhecer e representar, no plano cartesiano, o simétrico de figuras em relação aos eixos e à origem
- **EF07MA13:** Compreender a ideia de variável, representada por letra ou símbolo, para expressar relação entre duas grandezas, diferenciando-a da ideia de incógnita
- **EF07MA18:** Resolver e elaborar problemas que possam ser representados por equações polinomiais de primeiro grau, redutíveis à forma $a x + b = c$, fazendo uso das propriedades de igualdade
- **EF08MA06:** Resolver e elaborar problemas que envolvam cálculo de valor numérico de expressões algébricas, utilizando as propriedades das operações
- **EF08MA19:** Resolver e elaborar problemas que envolvam medidas de áreas de figuras geométricas, utilizando expressões de cálculo de área (quadriláteros, triângulos e círculos), em situações como determinar a medida de terrenos
- **EF09MA04:** Resolver e elaborar problemas com números reais, inclusive em notação científica, envolvendo diferentes operações
- **EF09MA06:** Compreender as funções como relações de dependência unívoca entre duas variáveis e suas representações numérica, algébrica e gráfica e utilizar esse conceito para analisar situações que envolvam relações funcionais entre duas variáveis
- **EM13MAT302:** Construir modelos empregando as funções polinomiais de primeiro e segundo graus, para resolver problemas em contextos diversos, com ou sem apoio de tecnologias digitais

- **EM13MAT315:** Investigar e registrar, por meio de um fluxograma, quando possível, um algoritmo que resolve um problema
- **EM13MAT405:** Utilizar conceitos iniciais de uma linguagem de programação na implementação de algoritmos escritos em linguagem corrente e/ou matemática

4. RECURSOS GERAIS

Recursos envolvidos na elaboração destes projetos-exemplo de Objetos de Aprendizagem (OAs) digitais:

- Programação por Blocos Visuais
- Criação e Manipulação de Script
- Noção sobre Fluxo de Programa
- Manipulação de Atores
- Manipulação de blocos de Eventos
- Manipulação de blocos de Sensores
- Manipulação de blocos de Controle
- Manipulação de blocos de Aparência
- Manipulação de blocos de Som
- Manipulação de blocos de Operadores
- Manipulação de blocos de Variáveis
- Manipulação de blocos Meus Blocos
- Manipulação de blocos Extensão
- Manipulação de Cenários
- Utilização de Comandos Básicos: Iniciar Script, Movimento (Andar), Posição (Plano Cartesiano), Interação (Falar), Rotação, Ângulos
- Criação e Manipulação Digitalmente de Imagens

5. CARACTERÍSTICAS PEDAGÓGICAS

Principais características pedagógicas desejáveis na elaboração destes projetos-exemplo de Objetos de Aprendizagem (OAs) digitais:

- **Interatividade:** indica se há suporte às consolidações e ações mentais, requerendo que o aluno interaja com o conteúdo do Objeto de Aprendizagem (OA) de alguma forma, podendo ver, escutar ou responder algo;
- **Autonomia:** indica se os Objetos de Aprendizagem (OAs) apoiam a iniciativa e tomada de decisão;
- **Cooperação:** indica se há suporte para aos alunos trocarem opiniões e trabalhar coletivamente sobre o conceito apresentado;
- **Cognição:** refere-se às sobrecargas cognitivas alocadas na memória do aluno durante o processo de ensino-aprendizagem;
- **Afetividade:** refere-se aos sentimentos e motivações do aluno com sua aprendizagem e durante a interação com o Objeto de Aprendizagem.

6. CARACTERÍSTICAS TÉCNICAS

Principais características técnicas desejáveis na elaboração destes projetos-exemplo de Objetos de Aprendizagem (OAs) digitais:

- **Acessibilidade:** indica se o Objeto de Aprendizagem pode ser acessado por diferentes tipos de usuários (ex.: deficientes visuais e auditivos etc.), em diferentes lugares (ex. lugares com acesso à internet e sem acesso à internet, etc.), e por diferentes tipos de dispositivos (ex. computadores, notebooks, dispositivos móveis, etc.);
- **Agregação:** indica se os componentes do Objeto de Aprendizagem podem ser agrupados em conjuntos maiores de conteúdos (ex. estruturas tradicionais de um curso);

- **Confiabilidade:** indica que o Objeto de Aprendizagem não apresenta defeitos técnicos ou problemas no conteúdo pedagógico;
- **Disponibilidade:** indica se o Objeto de Aprendizagem está disponível para ser utilizado;
- **Durabilidade:** indica se o Objeto de Aprendizagem se mantém intacto quando o repositório em que ele está armazenado muda ou sofre problemas técnicos;
- **Facilidade de instalação:** indica se o Objeto de Aprendizagem pode ser facilmente instalado caso ele exija esse recurso;
- **Granularidade:** é a extensão à qual um Objeto de Aprendizagem é composto por componentes menores e reutilizáveis.
- **Interoperabilidade:** medida do esforço necessário para que os dados dos Objetos de Aprendizagem (OAs) possam ser integrados a vários sistemas;
- **Manutenibilidade:** é a medida de esforço necessária para alterações do Objeto de Aprendizagem (OA);
- **Reusabilidade:** indica as possibilidades de reutilizar os Objetos de Aprendizagem (OAs) em diferentes contextos ou aplicações. Essa é a principal característica de um Objeto de Aprendizagem e pode ser influenciada por todas as demais.

7. CULTURA DIGITAL BNCC (2017)

Características da competência Cultura Digital BNCC (2017) encontradas na elaboração destes projetos-exemplo de Objetos de Aprendizagem (OAs) digitais:

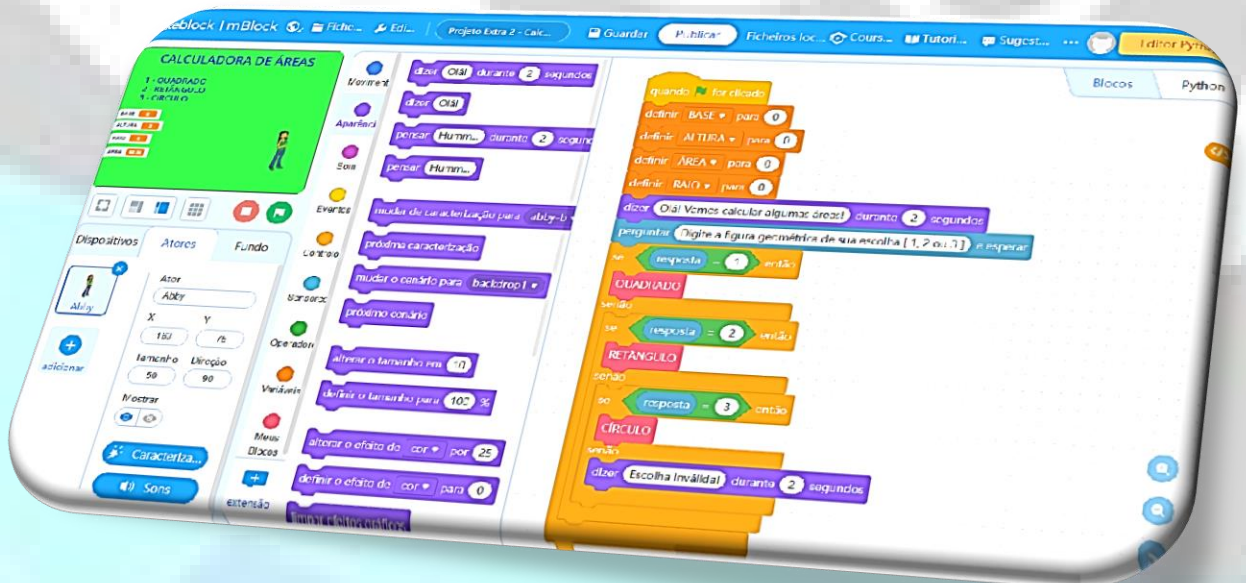
- **Computação e Programação** – Utilização de Ferramentas Digitais, Produção de Multimídia, Linguagens de Programação;
- **Pensamento Computacional** – Domínio de Algoritmos e Resolução de Problemas;
- **Cultura e Mundo Digital** – Mundo Digital, Uso Ético.

8. PROJETOS-EXTRA

8.1 PROJETO EXTRA 1 – O PANDA NO PARQUE DE DIVERSÕES



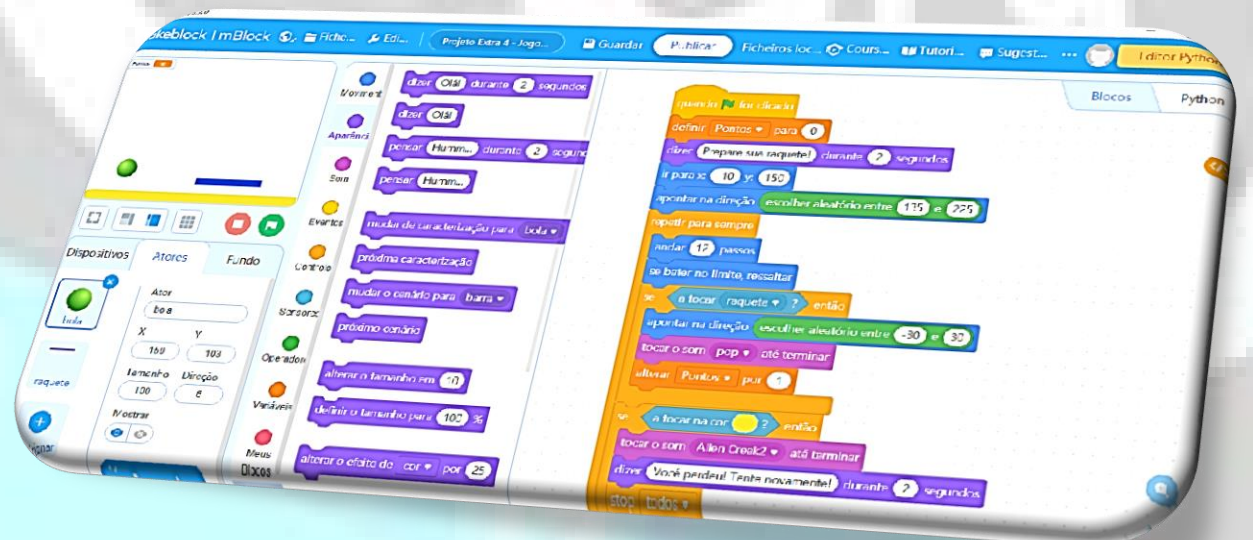
8.2 PROJETO EXTRA 2 – CALCULADORA DE ÁREAS



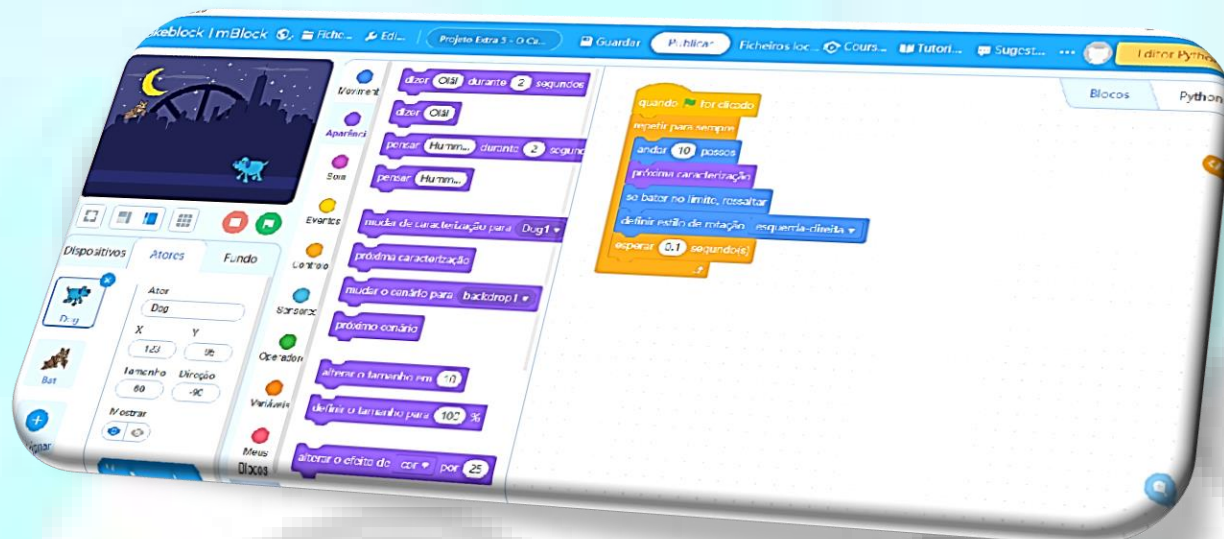
8.3 PROJETO EXTRA 3 – QUIZ



8.4 PROJETO EXTRA 4 – O JOGO PONG



8.5 PROJETO EXTRA 5 – O CACHORRO AZUL E O MORCEGUINHO



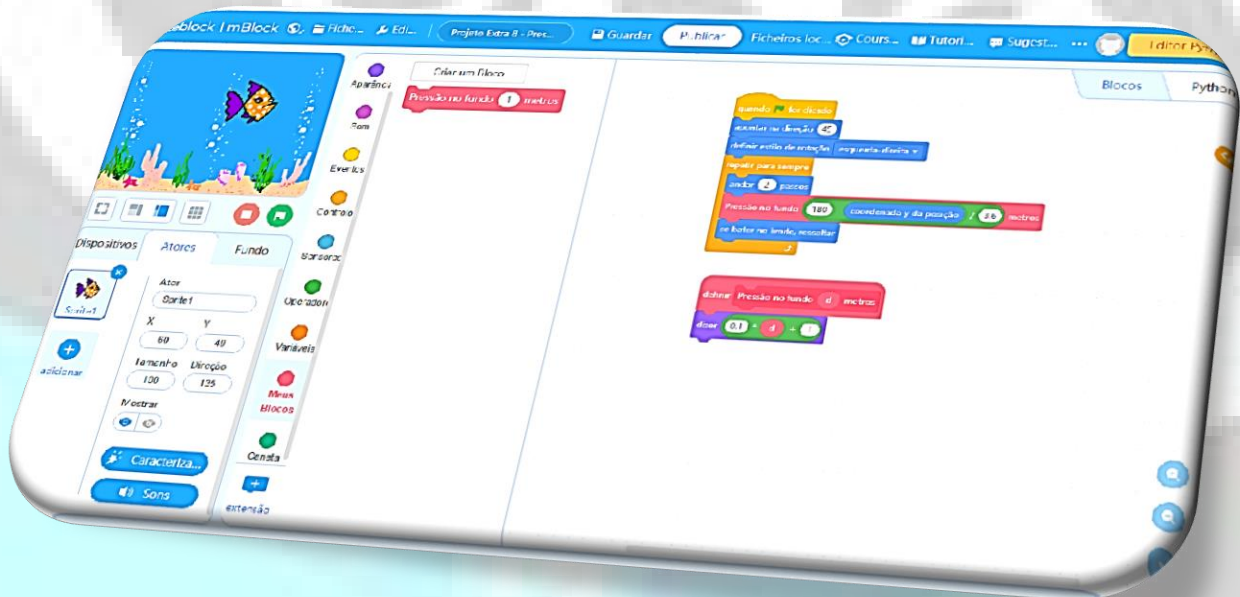
8.6 PROJETO EXTRA 6 – SIMULADOR DE QUEDA LIVRE



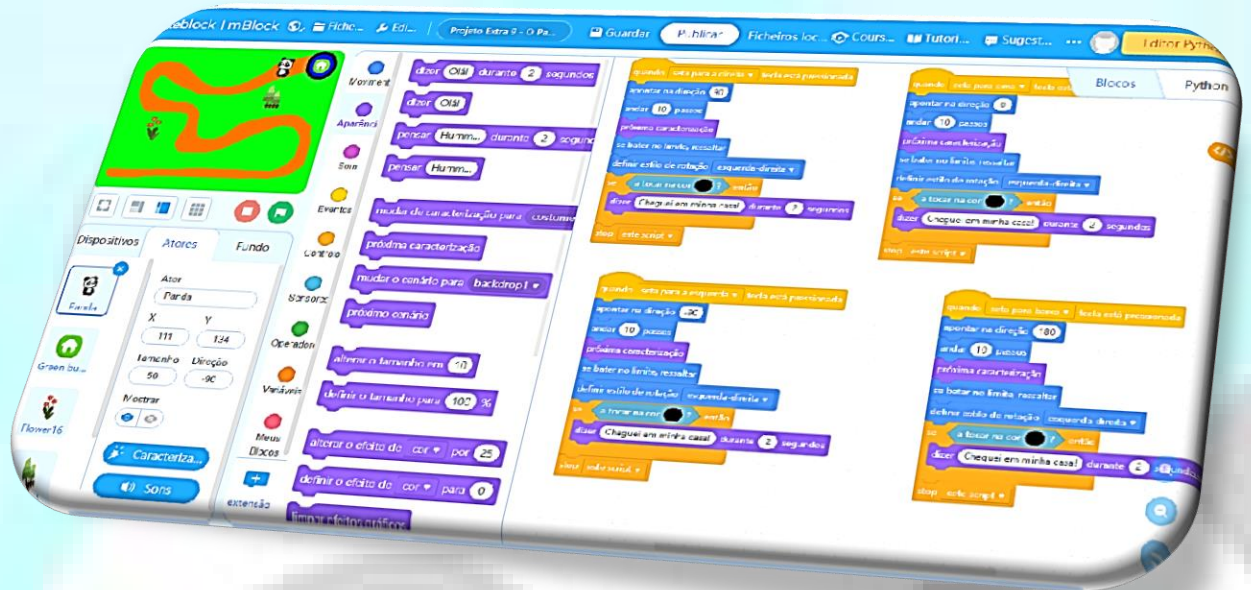
8.7 PROJETO EXTRA 7 – GATO MIANDO NO QUARTO



8.8 PROJETO EXTRA 8 – PRESSÃO NO FUNDO DO MAR



8.9 PROJETO-EXTRA – O PANDA NA TRILHA DE CASA



BONS ESTUDOS!

