

CHRISTINE MARTINS DE MATOS • ELLEN FELÍCIA FERREIRA MIRANDA
LÉRICA MARIA MENDES VELOSO • MIRELLE PEREIRA DA SILVA
ANDREY GUILHERME MENDES DE SOUZA • LUCAS GUIMARÃES PEREIRA
FÁBIA MAGALI SANTOS VIEIRA • ALCINO FRANCO DE MOURA JÚNIOR

LÓGICA DE PROGRAMAÇÃO

UMA PROPOSTA PARA CRIANÇAS
COM O USO DE SCRATCH



série
**POPULARIZAÇÃO
DO CONHECIMENTO**

LÓGICA DE PROGRAMAÇÃO

**UMA PROPOSTA PARA CRIANÇAS
COM O USO DE SCRATCH**

©UNIVERSIDADE ESTADUAL DE MONTES CLAROS - UNIMONTES

Wagner de Paulo Santiago
Reitor

Dalton Caldeira Rocha
Vice-Reitor

Ivana Ferrante Rebello
Pró-Reitora de Ensino

Marlon Cristian Toledo Pereira
Pró-Reitor de Pós-Graduação

Maria das Dores Magalhães Veloso
Pró-Reitora de Pesquisa

Pablo Peron de Paula
Pró-Reitor de Planejamento, Gestão e Finanças

Rogério Othon Teixeira Alves
Pró-Reitor de Extensão

©EDITORIA UNIMONTES

Maria Clara Maciel de Araújo Ribeiro
Editora Chefe

CONSELHO EDITORIAL

Maria Clara Maciel de Araújo Ribeiro
Gustavo Henrique Cepolini Ferreira
Ivana Ferrante Rebello
Leandro Luciano Silva Ravnjak
Luiz Henrique Carvalho Penido
Patrícia Takaki Neves
Tânia Marta Maia Fialho
Vanessa de Andrade Royo

APOIO:



LÓGICA DE PROGRAMAÇÃO

**UMA PROPOSTA PARA CRIANÇAS
COM O USO DE SCRATCH**

Christine Martins de Matos
Ellen Felícia Ferreira Miranda
Lérica Maria Mendes Veloso
Mirelle Pereira da Silva
Andrey Guilherme Mendes de Souza
Lucas Guimarães Pereira
Fábia Magali Santos Vieira
Alcino Franco de Moura Júnior

Jésus Ricardo de Faria Almeida
Projeto gráfico, capa e diagramação

Angela Heloíza Benedito Buxton
Revisão linguística

Maria Clara Maciel de Araújo Ribeiro
Editora geral

Este livro foi selecionado por edital e submetido a parecer duplo-cego

Parceria na produção do material:

Educar - Laboratório Multiusuário de Tecnologias Digitais na Educação - Edição 2024 2025, financiado pela Fundação de Amparo à Pesquisa do Estado de Minas Gerais (Fapemig). Edital 04/2023 - Centros de Tecnologia e Infraestrutura para a pesquisa na UEMG e UNIMONTES

Dados Internacionais de Catalogação na Publicação (CIP) (Câmara Brasileira do Livro, SP, Brasil)

Lógica de programação [livro eletrônico] : uma proposta para crianças com o uso de Scratch. — Montes Claros, MG : Editora Unimontes, 2025. — (Popularização do conhecimento)

PDF

Vários autores.

ISBN 978-85-7739-772-3

1. Scratch (Linguagem de programação para computador) I. Série.

25-312955.0

CDD-005.13

Índices para catálogo sistemático:

1. Scratch : Linguagem de programação : Computadores
: Processamento de dados 005.13

Livia Dias Vaz - Bibliotecária - CRB-8/9638

©EDITORA UNIMONTES

Campus Universitário Prof. Darcy Ribeiro - Montes Claros - Minas Gerais - Brasil
CEP 39401-089 - CAIXA POSTAL 126
www.editora.unimontes.br | editora@unimontes.br

Filiada à:



SUMÁRIO

7 APRESENTAÇÃO

10 MÓDULO 1 - INTRODUÇÃO À LÓGICA DE PROGRAMAÇÃO

21 MÓDULO 2 - INTRODUÇÃO AO PLANO CARTESIANO

29 MÓDULO 3 - INTRODUÇÃO AO SCRATCH

63 MÓDULO 4 - CRIANDO ANIMAÇÕES COM SCRATCH

84 MÓDULO 5 - CRIANDO JOGOS COM SCRATCH

99 MÓDULO 6 - CRIANDO HISTÓRIAS INTERATIVAS COM SCRATCH

109 MÓDULO 7 - BLOCOS DE COMANDO NO SCRATCH

130 AUTORES

APRESENTAÇÃO

Sejam bem-vindos à aventura da Lógica de Programação com o uso de Scratch!

Você já imaginou como os jogos, os desenhos animados e até os robôs funcionam? Tudo isso é possível graças à lógica de programação! E o melhor de tudo é que você também pode criar suas próprias histórias, animações e jogos com uma ferramenta super divertida chamada Scratch.

Este livro foi feito especialmente para você, que está começando a descobrir o incrível mundo da programação! Aqui, vamos aprender de forma fácil e divertida, passo a passo, como a lógica de programação funciona. O Scratch usa blocos coloridos que se encaixam como peças de um quebra-cabeça. Isso ajuda a transformar suas ideias em projetos reais, cheios de criatividade!



Por que aprender lógica de programação?

Aprender a programar é como aprender uma nova linguagem – a linguagem dos computadores! Isso é importante porque, cada vez mais, usamos a tecnologia em nosso dia a dia: no celular, no videogame, no tablet e até na escola. Entender como os computadores funcionam vai ajudá-lo a criar suas próprias soluções para desafios e, quem sabe, até inventar um jogo ou aplicativo incrível!

Objetivo do livro

O objetivo deste material é ajudá-lo a desvendar o mundo da lógica de programação, possibilitando-lhe a criação de suas primeiras animações e jogos no Scratch. Ao final, você entenderá os passos necessários para realizar as tarefas do computador, desenvolvendo seu raciocínio lógico e sua criatividade!

A quem se destina?

Este material foi pensado inicialmente para estudantes dos anos finais do Ensino Fundamental (a partir do 4º ou 5º ano), mas qualquer pessoa curiosa pode mergulhar nesse mundo, em seus primeiros passos na lógica de programação! Nosso objetivo é mostrar que a programação não é apenas para adultos ou profissionais de informática: todo mundo pode aprender. Saber programar desenvolve o raciocínio lógico, ajuda a resolver problemas do dia a dia e ensina a trabalhar em equipe, colaborando e compartilhando ideias.

Como usar este material?

Aqui vão algumas dicas de como aproveitar ao máximo:

- Siga o passo a passo: cada capítulo traz uma explicação clara e atividades práticas para você entender melhor a lógica de programação.
- Experimente sem medo: não tem problema errar! O Scratch é um espaço para testar e criar sem preocupações.

- Trabalhe com amigos e professores: aprender é mais legal quando compartilhamos nossas descobertas. Você pode trocar ideias com seus colegas e pedir ajuda ao professor sempre que precisar.
- Use a imaginação: aqui, não há limites para sua criatividade. Quanto mais você explorar, mais legal vai ser!

Uso de Ferramenta de Inteligência Artificial na produção do material

Um outro assunto, e que será apresentado posteriormente em uma nova aventura do mundo da computação, é a inteligência artificial (IA). Neste livro foi usado na revisão dos textos, na concepção de ideias para os jogos voltados aos iniciantes na programação e, em pontos específicos de introdução de unidades conceituais, adaptando o texto para o público a que se destina. As ferramentas de IA utilizadas foram o Gemini, da Google e o ChatGPT, da empresa OpenAI. Vale a pena pesquisar também!

Agora é com você!

Você está pronto para embarcar nessa aventura e se tornar um criador de histórias e jogos? Vamos lá! Prepare-se para programar e se divertir porque o universo da lógica de programação com o uso de Scratch está esperando por você!

MÓDULO 1 – INTRODUÇÃO À LÓGICA DE PROGRAMAÇÃO

Estudantes, antes de mergulharmos nesta aventura de aprendizagem, vamos conhecer algumas ideias básicas que vão fazer tudo ficar ainda mais divertido e fácil de entender! Para isso, precisamos saber o que é lógica de programação. Vamos lá!!!

O que é Lógica de Programação

A lógica de programação é uma maneira de organizar as ideias para escrever programas de computador de maneira clara. Podemos compará-la à resolução de um quebra-cabeça, em que antes de começarmos a montar, é necessário analisar como as peças se encaixam.

Pense da seguinte forma: na lógica de programação você deve organizar as ideias e montar as instruções (em forma de encaixes, como as peças de Lego) para que o computador possa entender o que deve ser feito e realizar uma atividade específica.

Ao escrever um programa, é necessário dividir o processo de criação em partes menores e fornecer instruções claras para resolver cada uma delas. É como se estivéssemos explicando, passo a passo, maneiras de solucionar um problema, por isso você precisa ser claro, exato e usar uma linguagem que o computador entenda. Assim, é preciso que se use a linguagem de programação. Você sabe o que é linguagem de programação?



Vamos aprender!

O que é Linguagem de Programação

Uma linguagem de programação é a maneira que você (programador) deve se comunicar/expressar para que o computador entenda e realize uma atividade, ou seja, a linguagem de programação se diferencia da linguagem que usamos habitualmente, sendo utilizados comandos, instruções de direcionamento para realização de tarefas e interação do programador com o computador. Assim como usamos palavras para nos comunicar, por exemplo, com amigos e familiares, devemos usar uma linguagem própria de programação para nos comunicarmos com os computadores.

Imagine a seguinte situação: ensinar um robô a fazer um sanduíche. Você não pode simplesmente dizer “Faça um sanduíche”, porque o robô não entenderia. Em vez disso, você precisa fornecer instruções detalhadas, seguindo um passo a passo com comandos específicos.

Primeiro comando:

- Separe os ingredientes, sendo duas fatias de pão, uma fatia de queijo e uma fatia de presunto.

Segundo comando:

- Monte o sanduíche: abra as fatias de pão, colocando-as lado a lado, adicione o recheio, colocando sobre uma das fatias de pão uma fatia de queijo e uma fatia de presunto; feche o sanduíche.

Terceiro comando:

- Sirva o sanduíche.

Perceba que a linguagem de programação é um recurso utilizado para ensinar aos computadores o que fazer, permitindo a criação de jogos, aplicativos, sites, histórias interativas, animações, entre outras atividades legais!

A comunicação com o computador é realizada em um espaço digital e neste espaço estão presentes diversos elementos organizados em um “menu”. No menu são realizados os comandos desejados para a execução das atividades. Por exemplo, imagine um carrinho de controle remoto; o controle remoto é a interface. Por meio dele é que você consegue dizer ao carrinho o que deve fazer e a direção que deve seguir (para a esquerda, para a direita, como ir para frente, para trás ou virar). Vamos entender um pouco mais sobre interface. Bora lá!

O que é uma interface?

Para fazer qualquer atividade no computador, precisamos seguir um passo a passo usando uma linguagem especial, chamada linguagem de programação, e um ambiente digital, chamado interface.

Uma interface é a tela ou o espaço digital onde você interage com um computador, tablet ou qualquer outro dispositivo eletrônico. É como o “rosto” do seu brinquedo ou videogame! Nesse “rosto” (interface), você encontra várias opções e botões. Quando você clica nessas opções, elas fazem coisas acontecerem como mover um personagem, ativar “poderes especiais”, voltar ao início do jogo ou avançar para a próxima fase, entre muitas coisas. É por meio da interface que você controle e se diverte com o que está na tela!

Além disso, é importante organizar bem as informações para que o computador consiga entender o que queremos que ele faça. Para isso, usamos os algoritmos. Eles nos ajudam a colocar todas as etapas em ordem para realizar as tarefas direitinho.

Sabia que, na Matemática, quando aprendemos a somar,

Com a interface
você conversa com
o computador e faz a
mágica acontecer!



subtrair, multiplicar ou dividir, estamos usando algoritmos? Isso mesmo! Cada operação tem um conjunto de regras e passos que precisamos seguir para chegar à resposta certa.

Agora, vamos entender melhor o que são algoritmos? Leia o texto a seguir e vamos nessa!

O que é um algoritmo?

Para entender o que é um algoritmo, vamos imaginar que você tem um quebra-cabeça para montar. Antes de começar, é importante olhar a imagem na caixa para saber como ele deve ficar quando estiver pronto.

Por exemplo, você pode começar pelas peças de canto, depois montar as peças das bordas, e seguir até que todas as peças se encaixem certinho.

Então, um algoritmo é justamente esse conjunto de instruções que você segue para montar o quebra-cabeça! Agora, você aprenderá mais sobre algoritmos resolvendo os exercícios a seguir.

ATIVIDADES

ATENÇÃO!
Vamos praticar.



1. Escreva aqui a lista de passos/atividades que você executa até o momento de ir à escola:

1 _____

2 _____

3 _____

4 _____

5 _____

6 _____

7 _____

8 _____

9 _____

10 _____

2. Enumere cada atividade para ordenar a lista de tarefas que você faz antes de dormir:

() Jantar

() Escovar os dentes

() Fazer a lição de casa

() Brincar

() Dormir

() Tomar banho

Observe que, nas duas atividades, você fez uma lista de comandos que devem ser seguidos em ordem para realizar uma tarefa. Isso é exatamente o que chamamos de algoritmo: uma sequência de passos para fazer alguma coisa.

Agora que você já conhece os conceitos básicos de Lógica de Programação, está pronto para mergulhar ainda mais nesse mundo

incrível! Como já conversamos, existem comandos especiais que ajudam a fazer tarefas. A partir de agora, vamos conhecer alguns desses comandos. Vamos aprender juntos!

Comandos de sequência

Os comandos de sequência são semelhantes ao que já aprendemos sobre algoritmos. Esses comandos funcionam como uma lista de instruções que o computador segue, uma de cada vez, para realizar uma tarefa. Assim, ele executa uma ação depois da outra, criando uma sequência de passos até completar a atividade.

Imagine que você vai fazer um bolo de chocolate e precisa seguir as etapas certas da receita. Os comandos de sequência funcionam como essa receita, na qual cada passo precisa ser feito em ordem para o bolo sair perfeito. Olha só:

1. Primeiro, quebre os ovos e coloque na tigela.
2. Depois, adicione o açúcar e misture bem.
3. Em seguida, acrescente o leite e mexa mais um pouquinho.
4. Agora é hora de colocar o chocolate em pó e misturar tudo.
5. Por fim, coloque a massa no forno e espere assar.

Cada passo precisa ser feito na sequência certa, assim como o computador faz quando recebe comandos. Se você mudar a ordem, como colocar a massa no forno antes de misturar os ingredientes, o bolo não vai dar certo, né?

É assim que os comandos de sequência funcionam no computador: ele segue uma lista de tarefas, uma de cada vez, até completar tudo.

Agora que entendemos como o computador pode seguir uma lista de ações em ordem, é hora de descobrir como ele pode tomar decisões nesse processo. Para isso, vamos aprender sobre os comandos de condição!

Comandos de condição

Com os comandos de condição, vamos aprender como um programa de computador pode tomar decisões e realizar ações com base nessas escolhas.

Para entender melhor, imagine que, antes de sair para a escola, você para na porta e olha para o céu. Se você perceber que o céu está cheio de nuvens e pode chover, então você volta e pega um guarda-chuva. Senão, você segue normalmente para a escola, sem se preocupar.

Viu só? Assim como você tomou uma decisão ao olhar para o céu, o computador também faz isso! Ele usa os comandos “Se”, “Então”, “Senão” para analisar condições e escolher a melhor sequência de ações para realizar uma tarefa.

Outra situação para entender melhor o que seja o comando de condição: vamos imaginar que você é o capitão de uma aventura pirata e está navegando pelas águas em busca de um tesouro. No meio do caminho, você encontra uma ilha misteriosa e precisa tomar uma decisão:

- Se a ilha estiver cheia de árvores com frutas, então você vai parar o navio para descansar e pegar as frutas.
- Senão, você vai continuar navegando em busca do tesouro.

Nesse caso, você está usando novamente os comandos de condição! Assim como no exemplo da ilha, o computador também toma decisões semelhantes. Ele analisa a situação (se tem frutas ou não) e decide o que fazer a seguir (parar ou seguir viagem).

Isso é o que o computador faz quando usamos os comandos “Se”, “Então”, “Senão” para ele tomar a melhor decisão e seguir o caminho certo!

Vimos que um computador pode tomar decisões com base em algumas condições. Mas, quando temos um comando que precisa ser repetido várias vezes, podemos usar os comandos de repetição. Vamos conhecê-los agora!

Comandos de repetição

Os comandos de repetição permitem fazer a mesma tarefa várias vezes sem precisar repetir os mesmos blocos de comando com as mesmas instruções. Em vez disso, você escolhe o bloco certo e diz quantas vezes a ação deve ser feita.

Por exemplo, imagine que uma criança decida fazer doação de alguns brinquedos que não usa mais e pede ajuda a um amigo para organizar os brinquedos em caixas. A criança avisa que é possível colocar apenas dez brinquedos em cada caixa. Então, o amigo vai repetir a ação de pegar um brinquedo e colocá-lo na caixa, até que os dez brinquedos estejam lá. Talvez ele faça isso dez vezes, até encher a caixa e precisar começar outra caixa, pois tem ainda mais brinquedos para doar.

Outra maneira de repetir uma tarefa é usando o comando Sempre. Quando você seleciona esse comando, uma sequência de blocos será repetida sem parar, infinitamente.

Para entender melhor, imagine que você está brincando com um autódromo de brinquedo, que simula uma corrida de carros. Se o botão do controle ficar preso, o carrinho continuará correndo pela pista sem parar, até que você conserte o controle.

Outro exemplo de repetição no cotidiano: vamos imaginar que você está brincando de pular corda com seus amigos. O desafio é pular a corda 5 vezes seguidas sem errar!

1. O amigo que está segurando a corda começa a gira
2. Você pula uma vez, duas vezes, três vezes... até chegar a 5 pulos.

Agora, pense que cada pulo é uma repetição. Em vez de falar para o seu amigo girar a corda toda vez, ele apenas repete a mesma ação várias vezes, uma atrás da outra, até atingir o número combinado. Assim funciona o comando de repetição no computador: ele faz uma tarefa várias vezes sem precisar de novas instruções a cada vez.

Se você decidir continuar pulando sem parar até alguém errar, isso seria um exemplo do comando Sempre: a ação se repete sem parar, até que uma condição (alguém errar) interrompa a brincadeira.

Embora os comandos de repetição permitam que uma tarefa seja feita várias vezes de forma automática, as variáveis mudam em cena para salvar informações que o programa pode precisar durante essas repetições. Vamos conhecer as variáveis agora!!!

Variáveis

As variáveis são usadas para salvar informações dentro de um código. É como se fosse uma caixa ou um baú onde você guarda seus brinquedos. Assim como você pode salvar coisas diferentes nesses lugares, as variáveis podem salvar qualquer tipo de informação, como palavras ou números.

Imagine que você está jogando um jogo de faz de conta em que você é o dono de uma sorveteria. Cada vez que um cliente chega e compra um sorvete, você precisa anotar quantos sorvetes foram vendidos para saber o total no final do dia.

Nesse caso, você usa uma variável chamada “sorvetes_vendidos” para salvar essa informação. Cada vez que um cliente compra um sorvete, você adiciona o número 1 à variável. Se no começo do dia a variável estava em 0 (nenhum sorvete vendido), ela vai aumentando a cada venda:

- Primeira venda: sorvetes_vendidos = 1
- Segunda venda: sorvetes_vendidos = 2
- Terceira venda: sorvetes_vendidos = 3

Assim, a variável “sorvetes_vendidos” guarda a quantidade total de sorvetes vendidos. Assim, você sabe direitinho quantos sorvetes foram comprados no final do dia!

Esse exemplo mostra como as variáveis guardam informações importantes que podem ser atualizadas e usadas sempre que necessário.

Vamos aprender juntos?



REVISÃO DO MÓDULO 1

Parabéns! Você chegou ao fim deste Módulo 1 - Introdução à Lógica de Programação.

Agora que você explorou os comandos de sequência, comandos de condição, comandos de repetição e variáveis, já sabe como o computador segue passos, toma decisões, repete tarefas e guarda informações importantes! Com os comandos de sequência, você aprendeu que o computador faz uma coisa de cada vez, em ordem certinha. Depois, viu que, com os comandos de condição, que é capaz de tomar decisões, escolhendo o melhor caminho com base em condições. Descobrimos também que, com os comandos de repetição, o computador pode repetir ações várias vezes sem precisar de novas instruções. E por último, vimos que as variáveis são como caixinhas onde o computador guarda palavras e números que ele pode precisar enquanto realiza tarefas.

Agora é hora de colocar tudo em prática! Com todas essas ferramentas, você pode criar jogos, animações e histórias incríveis no Scratch. Não tenha medo de experimentar, errar e tentar de novo – é assim que se aprende e se diverte ao mesmo tempo! Prepare-se para usar tudo o que aprendeu e soltar a imaginação. O mundo da programação está cheio de possibilidades e você está pronto para explorá-lo!

Nos vemos no próximo módulo!

Anotações

[illegible]

MÓDULO 2 – INTRODUÇÃO AO PLANO CARTESIANO



Imagine que você está jogando um jogo de tabuleiro. Nesse jogo, você precisa mover uma peça pelo tabuleiro para chegar até ao tesouro. O tabuleiro tem linhas e colunas para ajudá-lo a saber onde a peça está e para onde ela deve ir. No mundo da programação, temos algo parecido, chamado de plano cartesiano.

O plano cartesiano é como um mapa que ajuda o computador a saber onde colocar as coisas, como personagens, desenhos e objetos em uma tela. Ele tem dois lados: uma linha horizontal (que chamamos de eixo X) e uma linha vertical (que chamamos de eixo Y). Usamos números para dizer exatamente qual é seu lugar no mapa.

Se você quiser que um personagem se mova para a direita, precisa aumentar o número no eixo X. Se quiser que ele suba, deve aumentar o número no eixo Y. Assim como no jogo de tabuleiro, o plano cartesiano ajuda o computador a saber onde cada coisa está e para onde deve ir.

Então, entender o plano cartesiano é importante para você poder criar jogos e animações, sabendo exatamente onde posicionar e movimentar seus personagens e objetos no mundo da programação!

Vamos entender um pouco mais sobre o plano cartesiano, com resolução de algumas atividades!

Como você indicaria, para um amigo, que você mora na casa em destaque/sinalizada da Figura 1?

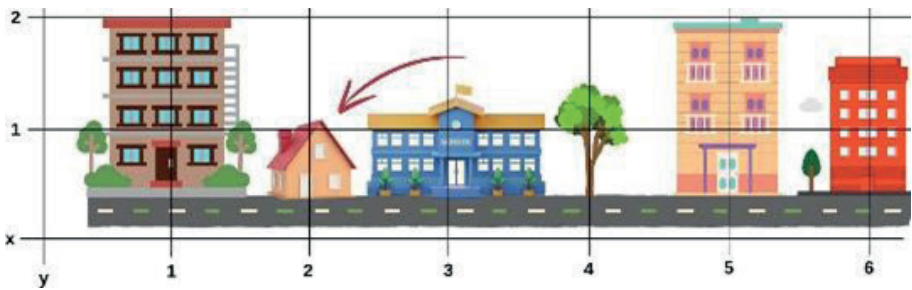
Figura 1 - Exemplo 1 de uma estrutura urbana



Fonte: Elaborado pelos autores, 2024, com uso da plataforma Canva.

Como você indicaria neste novo cenário da Figura 2?

Figura 2: Exemplo 2 de uma estrutura urbana



Fonte: Elaborado pelos autores, 2024, com uso da plataforma Canva.

Agora que você já fez as atividades de localização e seu amigo descobriu onde você mora, deu para perceber como é importante saber onde as coisas estão, certo? No mundo da programação, usamos o plano cartesiano exatamente para isso: para saber a posição exata de personagens, objetos e cenários que aparecem na tela. Se você quer que seu personagem corra até um ponto ou que um objeto

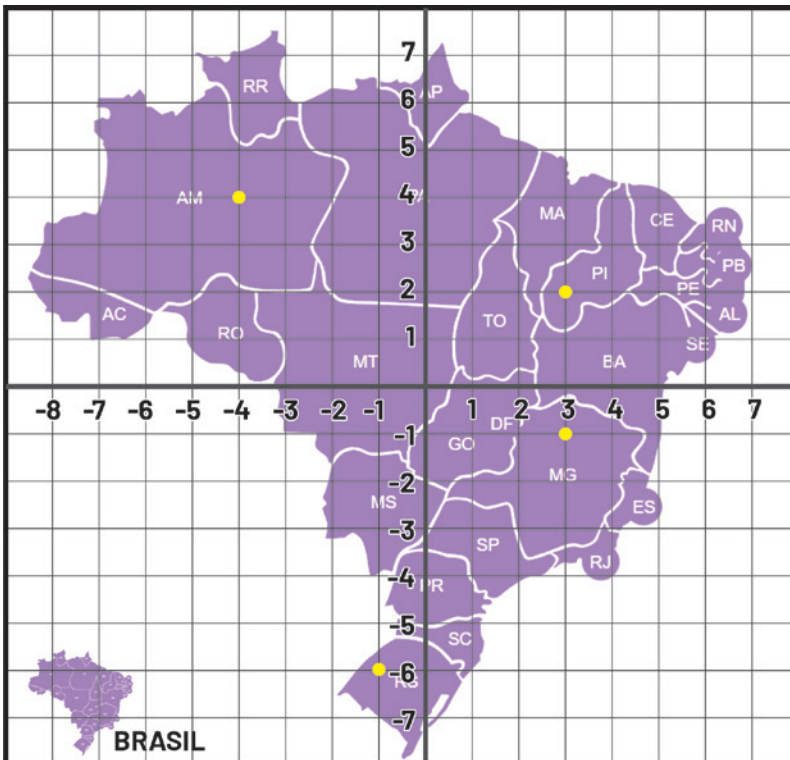
esteja em um certo lugar, precisa localizar essas coisas no plano cartesiano. Sem isso, os objetos poderiam ficar no lugar errado ou não se mexer como você imaginou.

Lembre-se de que o plano cartesiano é como um mapa, e nele você usa as posições nos eixos X e Y para orientar seus personagens, como faz um GPS. Assim, tudo ficará organizado e o jogo ou animação funcionará direitinho! Com essa habilidade, você poderá criar projetos cada vez mais incríveis e criativos!

Vamos reforçar a necessidade de que você conheça bem o plano cartesiano realizando novas atividades.

Com base na observação da Figura 3, responda as questões a seguir, tendo bastante atenção aos eixos X e Y, além dos números que estão presentes nos eixos.

Figura 3 – Representação do mapa do Brasil no Plano Cartesiano



Fonte: Elaborado pelos autores, 2025, com uso da plataforma Canva.

1. Onde está localizado o estado de Minas Gerais no mapa?

2. Qual estado está localizado na posição $(-4, 4)$?

3. Informe a coordenada (x, y) que indica a posição do estado do Rio Grande do Sul.

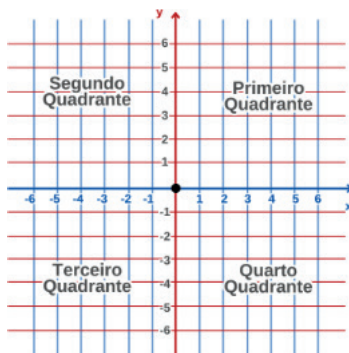
4. Qual estado está localizado na posição $(3, 2)$?

Quadrantes do Plano Cartesiano

O plano cartesiano é como um grande mapa dividido em quatro partes, que chamamos de quadrantes. Imagine que você tem uma folha com uma “cruz” no meio. A linha horizontal é o eixo X e a linha vertical é o eixo Y. Juntos, eles dividem o plano em quatro pedaços. Cada quadrante tem números diferentes e eles ajudam a encontrar a posição mais exata de algo na tela.

A Figura 4 traz uma representação dos quadrantes de um plano cartesiano.

Figura 4 - Plano Cartesiano em quadrantes



Fonte: Elaborado pelos autores, 2024, com uso da plataforma Canva.

Esses quadrantes dizem se o objeto está mais para cima, para baixo, para a direita ou para a esquerda, do lado positivo ou do lado negativo.

- **Primeiro Quadrante:** tudo para cima e para a direita e os dois números são positivos (X e Y).
- **Segundo Quadrante:** tudo para cima e para a esquerda, sendo que o número X é negativo e o Y é positivo.
- **Terceiro Quadrante:** tudo para baixo e para a esquerda e os dois números são negativos (X e Y).
- **Quarto Quadrante:** tudo para baixo e para a direita, sendo o X positivo e o Y negativo.

Agora pense que você é um personagem e está em um ponto do mapa. Se quiser andar para o quadrante 1, vai subir e andar para a direita. Assim, o plano cartesiano e seus quadrantes ajudam a se posicionar direitinho em jogos e animações!

Vamos praticar!

1. Determine o quadrante em que cada ponto está localizado:

a. Ponto (3, 4) _____

b. Ponto (-2, -5) _____

c. Ponto (0, -1) _____

d. Ponto (-6, 2) _____

2. No plano da Figura 5, marque os seguintes pontos:

a. (4, 2)

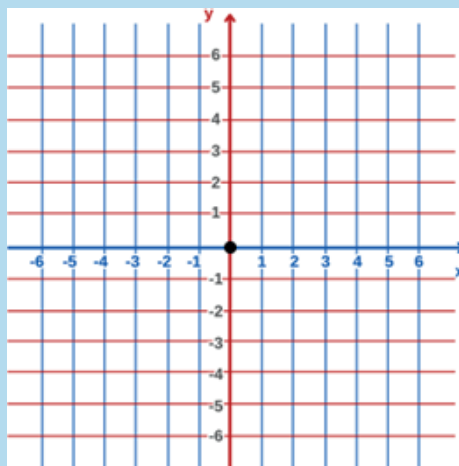
b. (-3, -1)

c. (0, 5)

d. (0, -5)

e. (-5, 4)

Figura 5 - Exercício de Plano Cartesiano



Fonte: Elaborado pelos autores, 2024, com uso da plataforma Canva.

REVISÃO DO MÓDULO 2

Parabéns! Você descobriu mais um módulo: Introdução ao Plano Cartesiano!

Agora que você aprendeu sobre o plano cartesiano e os quadrantes, já está preparado para criar e posicionar objetos e personagens em jogos, animações e outros projetos de programação!

Você descobriu que o plano cartesiano é como um mapa, com duas linhas, o eixo X (horizontal) e o eixo Y (vertical), que nos ajudam a encontrar posições exatas na tela. Também viu que esse plano é dividido em quatro quadrantes, cada um com números positivos e negativos, para indicar a direção onde as coisas estão.

Com essas habilidades, você já consegue:

- posicionar personagens e objetos no lugar certo;
- mover objetos para cima, para baixo, para a esquerda ou para a direita;
- criar cenários organizados e lógicos para seus jogos e projetos.

Lembre-se: entender como se localizar no plano cartesiano é muito importante para que seu código funcione corretamente. Cada coordenada que você usa é uma forma de dar instruções claras para o computador!

Agora é hora de colocar em prática tudo o que aprendeu. Continue explorando, brincando e criando coisas incríveis.

Nós nos encontramos no próximo módulo!

Anotações

This image shows a full page of white paper with horizontal dashed lines, typical of primary school handwriting practice paper. The lines are evenly spaced and run across the entire width of the page. There are no margins, text, or other markings present.

MÓDULO 3 – INTRODUÇÃO AO SCRATCH

O que é o Scratch?



O Scratch é a maior comunidade de programação para crianças no mundo! Ele é uma linguagem de programação super divertida que usa uma interface visual simples, com a qual você pode criar histórias, jogos e animações apenas arrastando blocos coloridos, como se estivesse montando um quebra-cabeça.

A grande importância do Scratch é que ele torna uma lógica de programação fácil de entender. Com ele, você aprende conceitos importantes, como sequências, repetições, condições e variáveis, sem precisar escrever códigos complicados. Isso ajuda a desenvolver o raciocínio lógico, a criatividade e até a resolução de problemas de uma forma interativa e lúdica.

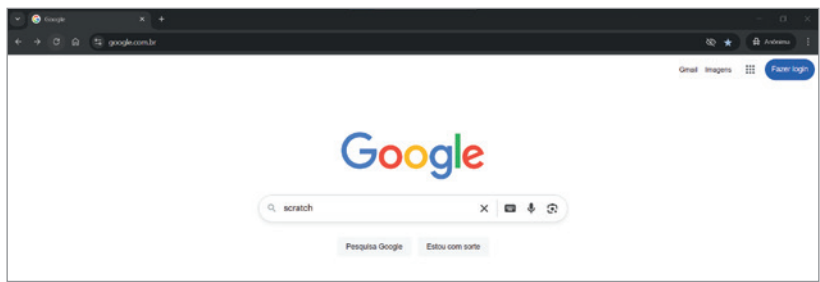
Com o Scratch, você pode transformar suas ideias em projetos digitais, criando coisas incríveis enquanto aprende a pensar como um verdadeiro programador!

Agora que já sabemos o que é o Scratch, vamos conhecê-lo!

Onde podemos encontrar o Scratch?

É simples! Ao abrir o Google (Figura 6), tudo o que você precisa fazer é digitar Scratch e apertar a tecla Enter.

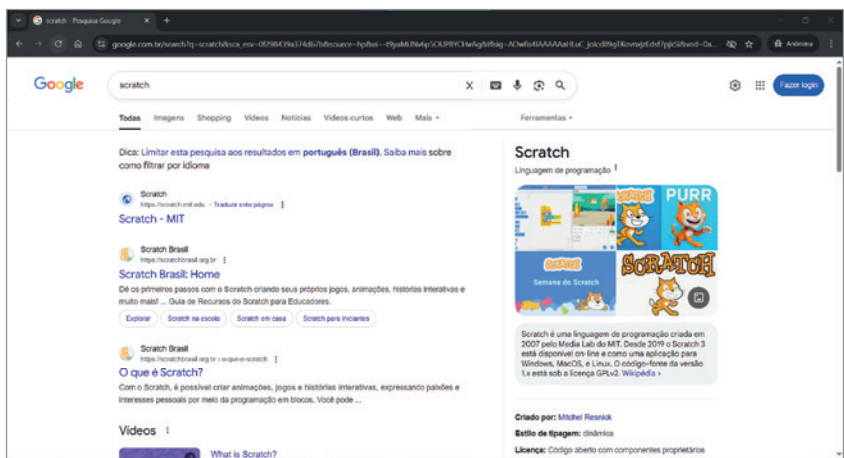
Figura 6 – Tela de navegação do Google



Fonte: Google, 2025.

Depois, basta clicar no primeiro link que aparece, como na Figura 7.

Figura 7 – Tela de Pesquisa de Scratch no Google



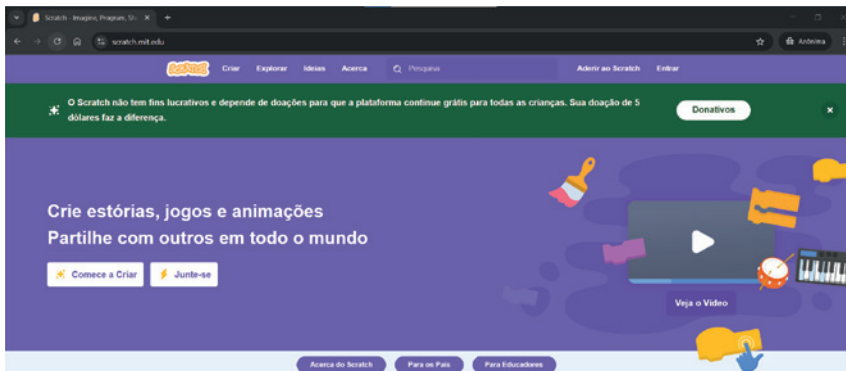
Fonte: Google, 2025.

Outra forma de acessar é digitar o link: <https://scratch.mit.edu> que já o direciona para a página inicial da plataforma.

Primeiros passos no Scratch

Ao entrar no site do Scratch, você será direcionado para a página inicial (Figura 8).

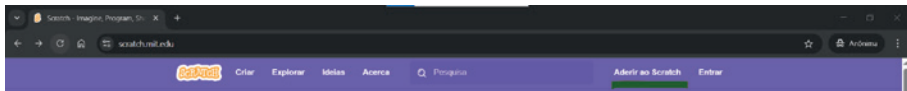
Figura 8 – Site do Scratch



Fonte: SCRATCH, 2025.

O primeiro passo será a criação de um usuário. Para isso, você precisa clicar onde está escrito Aderir ao Scratch, localizado na parte de cima do lado direito da tela, como na Figura 9.

Figura 9 – Parte de criação do usuário no Scratch



Fonte: SCRATCH, 2025.

Ao clicar em Aderir ao Scratch, você será levado para outra página (Figura 10), na qual precisará criar um nome de usuário (pode ser até o nome do seu personagem favorito de desenho, anime, filme ou livro).

Você também vai precisar criar uma palavra-passe, que funciona como uma senha. Ela é importante para que outras pessoas não consigam usar seu nome de usuário.



Figura 10 – Tela de “Aderir ao Scratch”

The registration screen has a blue background. At the top, it says "Aderir ao Scratch" in bold, followed by "Crie projectos, partilhe ideias, faça amigos. É grátis!". Below this, there are two sections: "Criar um nome de utilizador" with a text input field labeled "Nome de utilizador" and a blue button that says "Não use o seu nome real"; and "Criar uma palavra-passe" with two text input fields labeled "Palavra-passe" and "Introduza a sua palavra-passe de novo", and a checkbox labeled "Mostrar a palavra-passe" which is checked. At the bottom is a large blue button labeled "Próximo".

Fonte: SCRATCH, 2025.

Depois de Aderir ao Scratch, o próximo passo é informar o país onde estamos. Ao clicar no ícone da seta no meio da tela, uma lista com os nomes de vários países do mundo aparecerá. Aí, é só procurar pelo Brasil e seleccioná-lo (Figura 11).

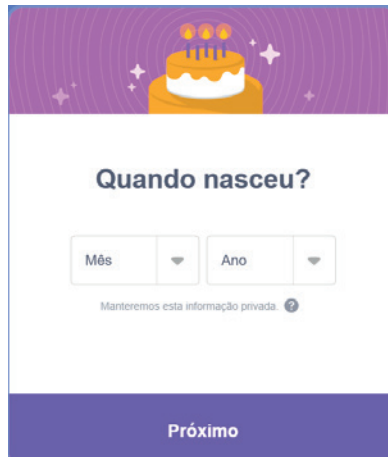
Figura 11 – Seleção do país no Scratch

The screen has a dark blue header with a space-themed illustration of the Earth, stars, and planets. Below the header, the text "Vive em que país?" is centered. Underneath is a dropdown menu with the placeholder text "Selecione o país" and a small downward arrow icon. At the bottom is a large blue button labeled "Próximo".

Fonte: SCRATCH, 2025.

Seguindo o mesmo processo de clicar no ícone da seta no meio da tela, você deverá informar o mês e o ano em que nasceu, conforme mostrado na Figura 12.

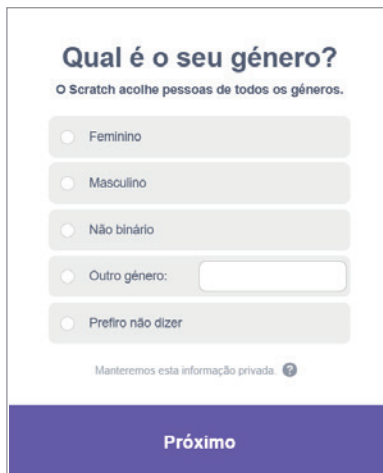
Figura 12 – Ano de nascimento do usuário do Scratch

A screenshot of the Scratch registration page for birth date. At the top, there is a purple header with a birthday cake icon and confetti. Below the header, the text "Quando nasceu?" is displayed in a bold, dark blue font. Underneath, there are two dropdown menus labeled "Mês" and "Ano". Below these menus, a small text says "Manteremos esta informação privada." with a question mark icon. At the bottom, there is a large purple button labeled "Próximo".

Fonte: SCRATCH, 2025.

Em seguida, é o momento de informar o seu gênero (Figura 13).

Figura 13 – Identificação de gênero do usuário no Scratch

A screenshot of the Scratch registration page for gender. At the top, there is a purple header with the text "Qual é o seu gênero?" in a bold, dark blue font. Below the header, a small text says "O Scratch acolhe pessoas de todos os gêneros." Underneath, there are five radio button options: "Feminino", "Masculino", "Não binário", "Outro gênero:" (with a text input field), and "Prefiro não dizer". Below these options, a small text says "Manteremos esta informação privada." with a question mark icon. At the bottom, there is a large purple button labeled "Próximo".

Fonte: SCRATCH, 2025.

E, como último passo, você deve informar o seu e-mail (Figura 14), aquele que seu papai ou sua mamãe, ou ainda o responsável disse que você poderia usar. Depois, é só clicar em Criar a Sua Conta para finalizar.

*Se você não souber qual é o seu e-mail, pergunte ao seu professor ou professora.

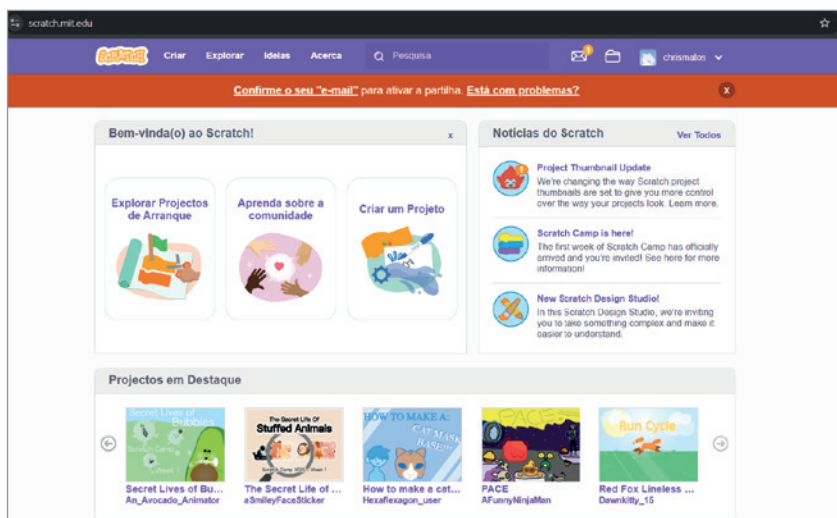
Figura 14 - E-mail de usuário do Scratch

A imagem mostra a interface de criação de uma conta no Scratch. No topo, há uma ilustração de um arco-íris, uma nuvem e um envelope branco. Abaixo, o texto pergunta "Qual é o seu endereço de correio electrónico?". Há um campo de entrada com o placeholder "Endereço de correio electrónico". Abaixo do campo, há um link "Manteremos esta informação privada." com um ícone de interrogação. Na base, há uma barra azul com o texto "Ao criar uma conta, aceita a Política de Privacidade e aceita e concorda com os Termos de Utilização." e um botão azul com o texto "Criar a Sua Conta".

Fonte: SCRATCH, 2025.

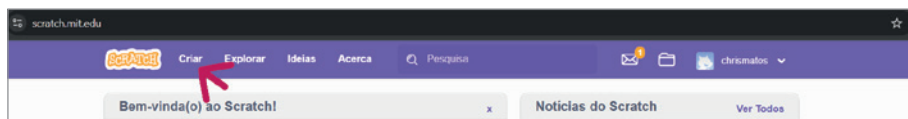
Apresentando a plataforma Scratch

Assim que sua conta de usuário for criada, você será direcionado para a página inicial do Scratch (Figura 15). Na barra roxa, localizada na parte de cima da tela, há algumas funções. Vamos conhecer cada uma delas agora.

Figura 15 – Tela do usuário no Scratch

Fonte: SCRATCH, 2025.

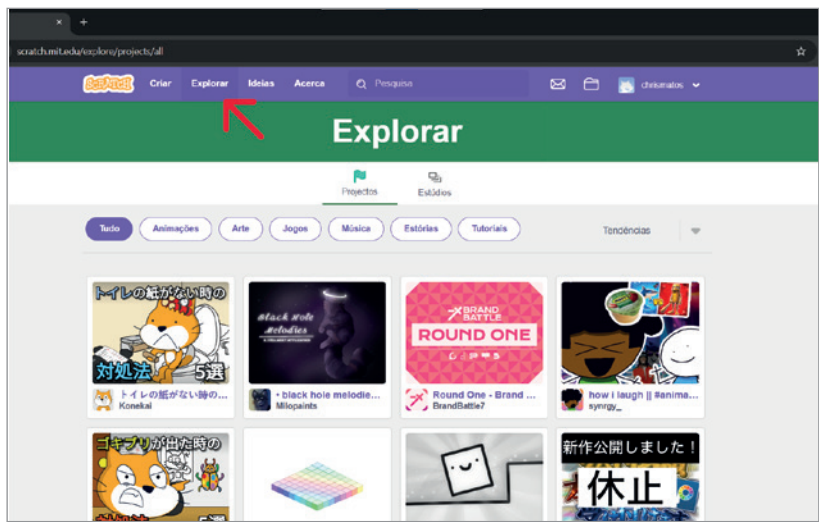
A primeira função na barra roxa é o “Criar”, e esse é o ambiente de programação em que faremos os nossos projetos (Figura 16).

Figura 16 – Função Criar no Scratch

Fonte: SCRATCH, 2025.

A segunda função é o “Explorar”, com a qual você pode ver os projetos feitos por outros usuários (Figura 17). Você pode pesquisar os projetos por categoria, como jogos, animações, arte e música.

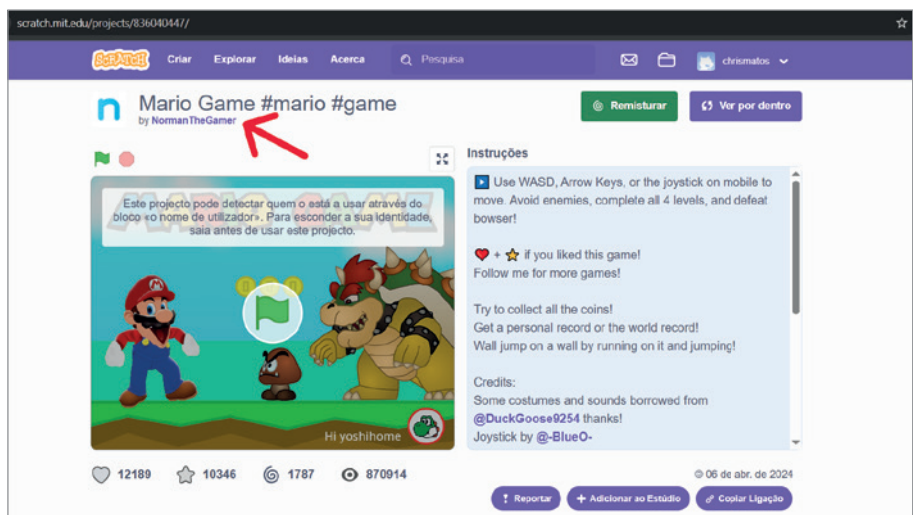
Figura 17 – Função Explorar no Scratch



Fonte: SCRATCH, 2025.

Se você gostar de algum projeto, pode clicar no nome do usuário e segui-lo para receber notificações sobre seus novos projetos, conforme mostrado no exemplo da Figura 18.

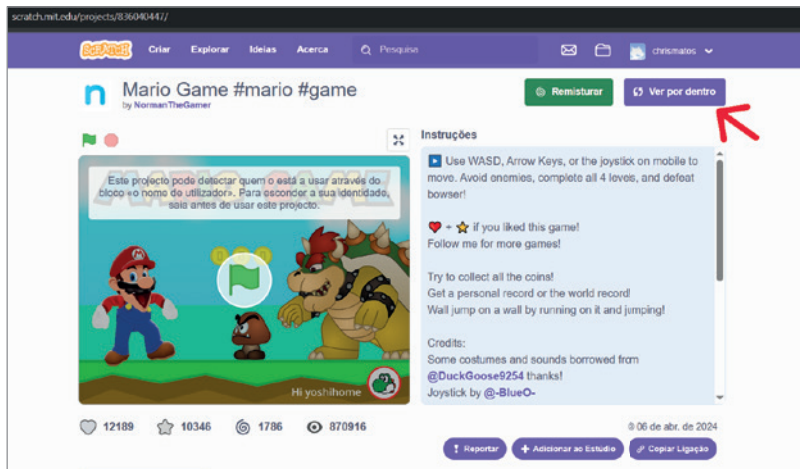
Figura 18 – Explorando um projeto no Scratch



Fonte: SCRATCH, 2025.

Também é possível ver como o usuário usou os blocos de comando para criar o projeto. Para isso, basta clicar em “Ver por dentro” (Figura 19).

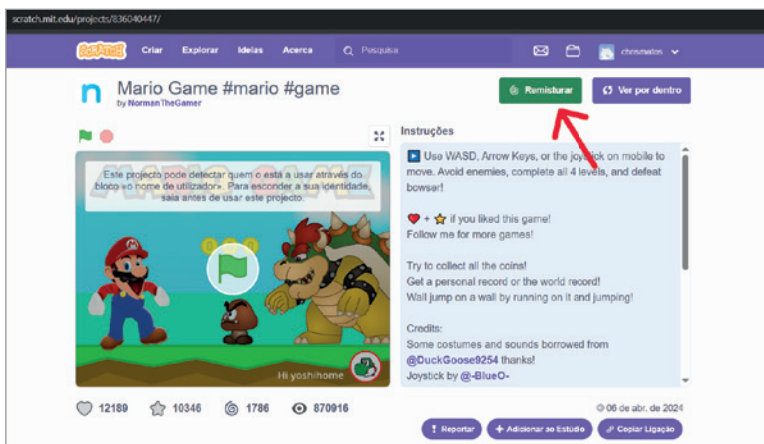
Figura 19 – Opção “Ver por Dentro” do Scratch



Fonte: SCRATCH, 2025.

E por fim, se você vir um projeto e pensar em algo nele que gostaria de adicionar, é só clicar em “Remisturar” (Figura 20).

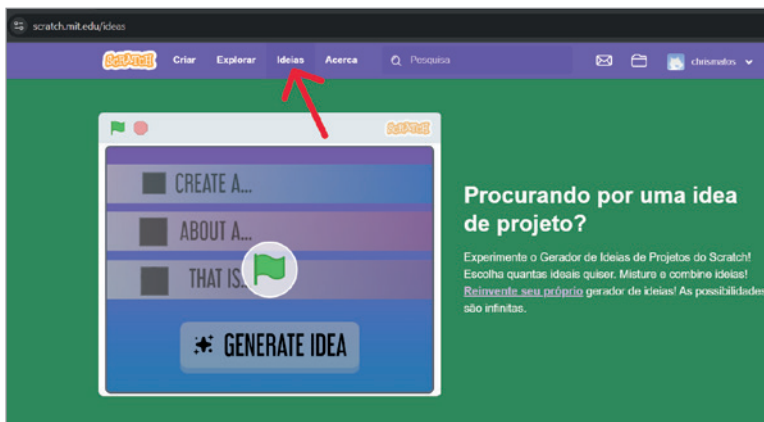
Figura 20 – Opção “Remisturar” no Scratch



Fonte: SCRATCH, 2025.

A terceira função do menu do Scratch é chamada de “Ideias”. Nela, você encontra tutoriais para projetos simples (Figura 21).

Figura 21 – Função “Ideias” no Scratch



Fonte: SCRATCH, 2025.

A quarta função é o “Acerca”, onde o próprio Scratch explica o que ele é e conta sobre a sua origem (Figura 22).

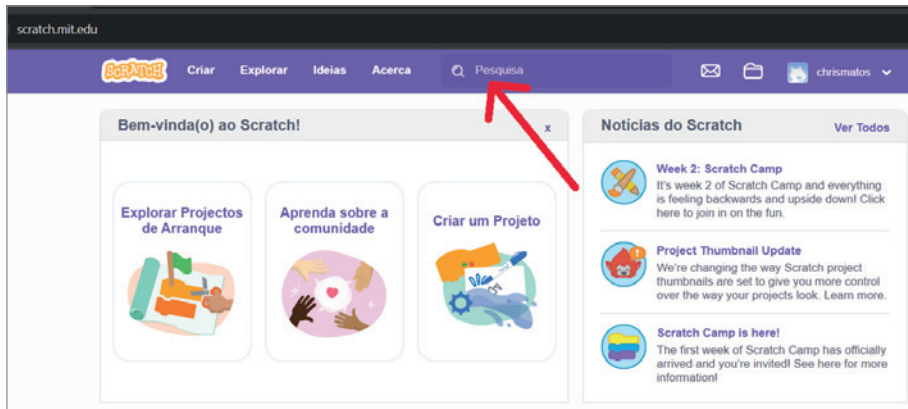
Figura 22 – Função “Acerca” no Scratch



Fonte: SCRATCH, 2025.

Ao lado da quarta função, encontramos o espaço Pesquisa (Figura 23). É nesse espaço que você vai procurar algo específico que queira ver.

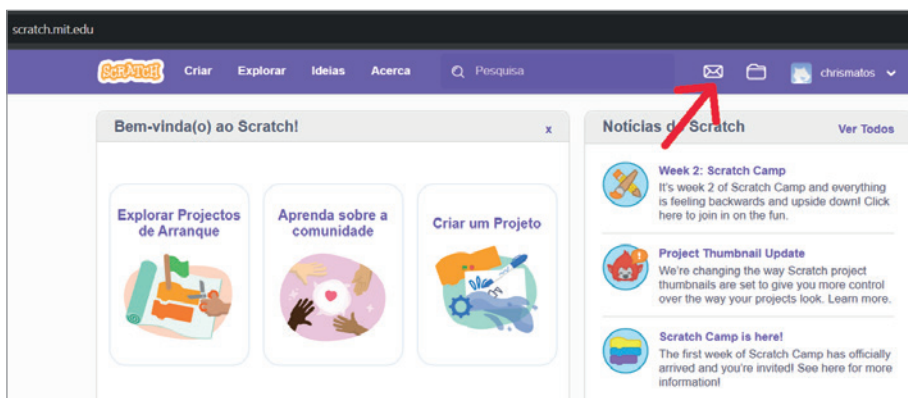
Figura 23 – Espaço de Pesquisa no Scratch



Fonte: SCRATCH, 2025.

Seguindo para o lado direito do menu, temos o símbolo de envelope, onde você receberá mensagens sobre seus projetos e comentários (Figura 24).

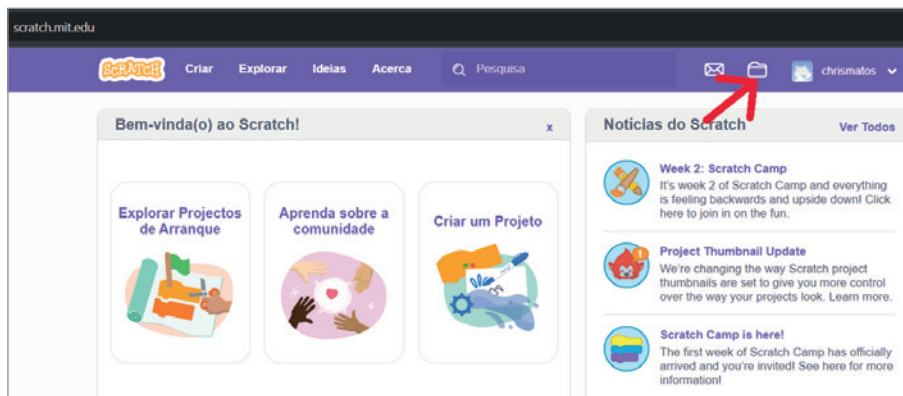
Figura 24 – Mensagens no Scratch



Fonte: SCRATCH, 2025.

Ao lado do símbolo do envelope, temos o símbolo da pasta, onde você pode encontrar seus projetos (Figura 25).

Figura 25 – Pasta de Projetos do usuário no Scratch



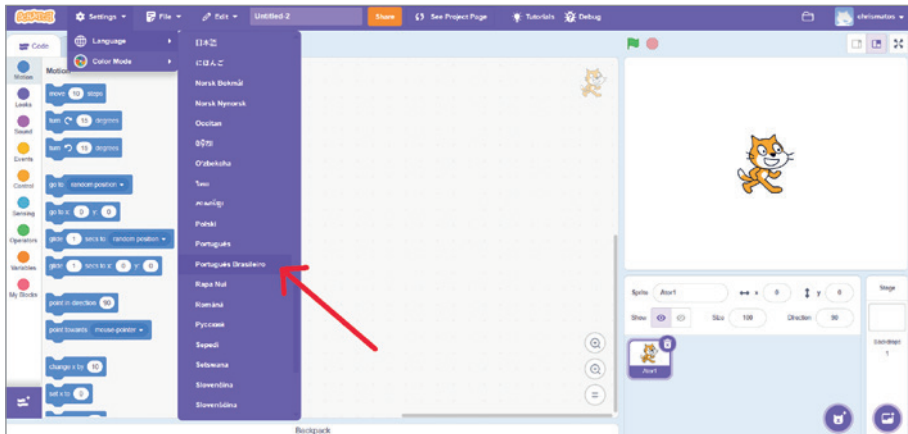
Fonte: SCRATCH, 2025.

Preste atenção no menu da página principal do Scratch! Ele é como um guia que o ajuda a encontrar tudo o que precisa. Nele, você pode explorar projetos, aprender com tutoriais, procurar coisas legais e ver suas mensagens. Também é onde você encontra seus próprios projetos guardados. Use o menu com cuidado para não se perder e aproveitar ao máximo tudo o que o Scratch oferece.

Ambiente de programação do Scratch

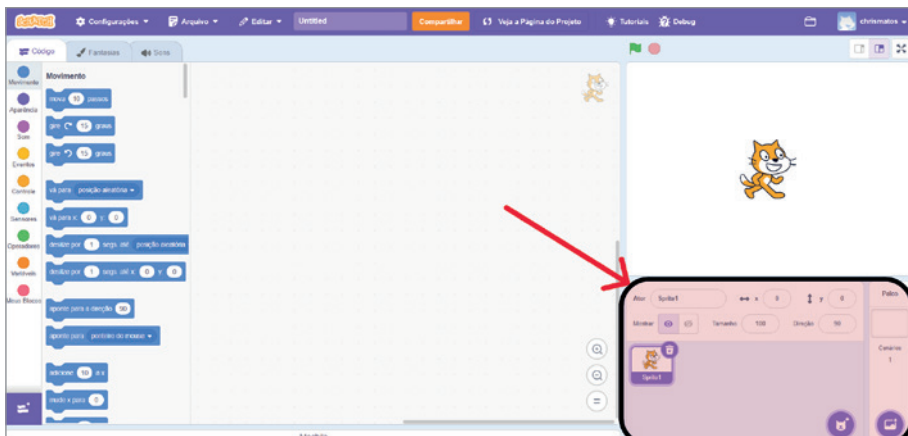
Para entrar no ambiente de programação do Scratch, você precisa clicar no menu principal na opção "Criar" e acessar a primeira função apresentada na seção Apresentando a Plataforma Scratch. Em seguida, é importante verificar se o idioma está correto.

Se o idioma não estiver em português, clique no menu superior em Configurações (caso esteja em inglês, a palavra é *settings*), depois em Idioma (caso esteja em inglês, a palavra é *language*) e escolha Português Brasileiro (Figura 26).

Figura 26 – Seleção de idioma no Scratch

Fonte: SCRATCH, 2025.

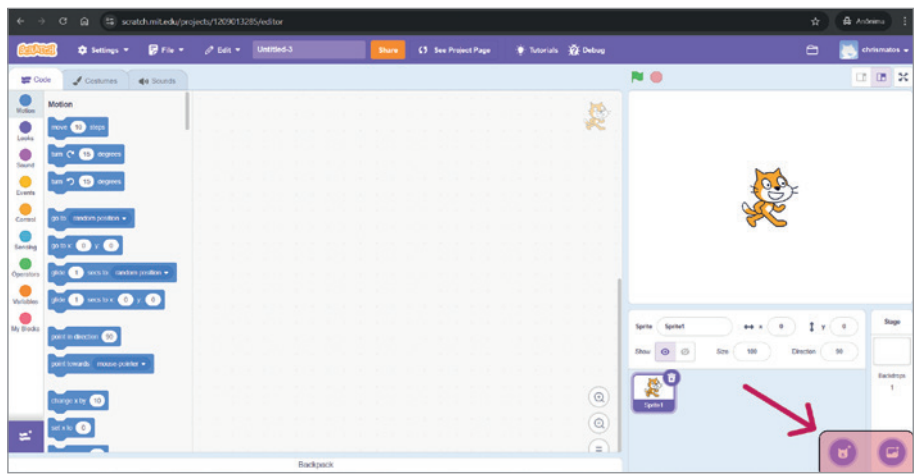
Do lado direito da tela, na parte debaixo, temos a área do palco. Nesse espaço, você pode adicionar fundos e personagens ao seu projeto (Figura 27).

Figura 27 – Área do palco no Scratch

Fonte: SCRATCH, 2025.

Os ícones no canto da tela (Figura 28) permitem importar, desenhar ou escolher entre vários personagens e cenários para o seu projeto.

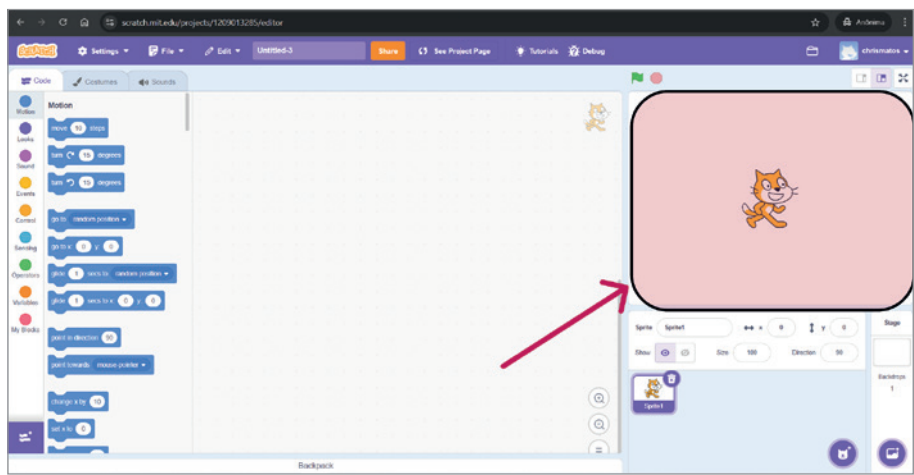
Figura 28 – Ícones do palco no Scratch



Fonte: SCRATCH, 2025.

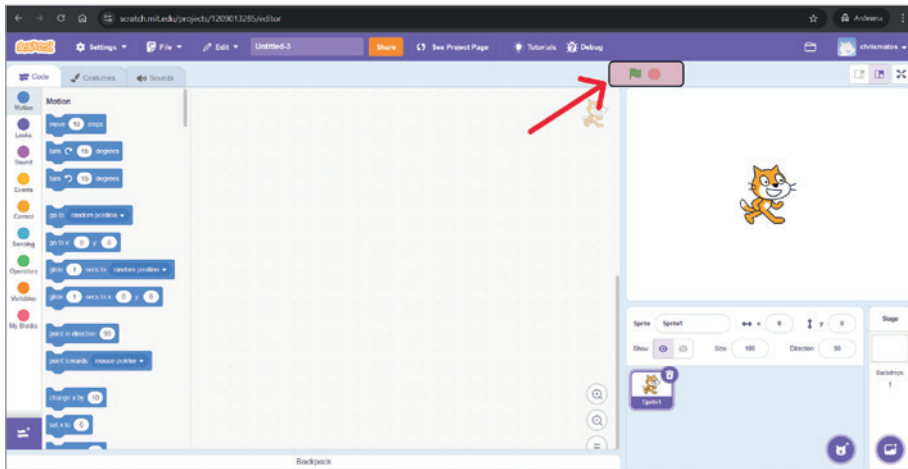
Do lado direito, na parte de cima da tela, onde vemos o gatinho, está o espaço no qual teremos a narrativa do projeto (Figura 29).

Figura 29 – Narrativa de projeto no Scratch



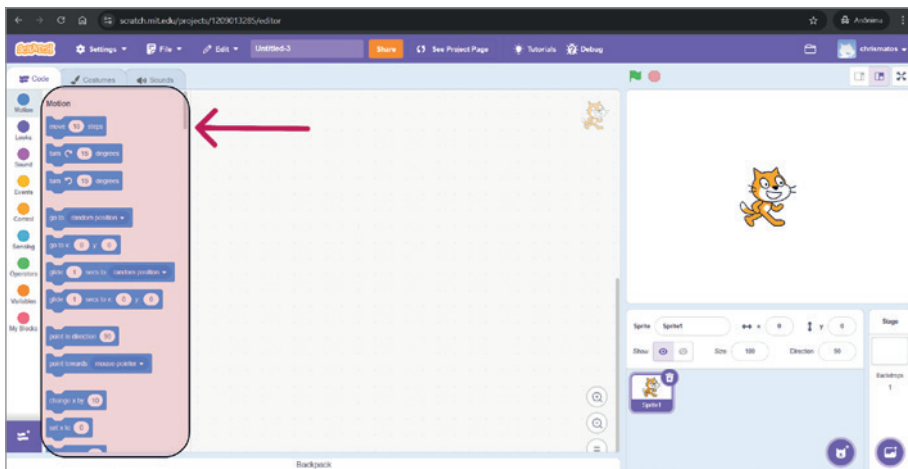
Fonte: SCRATCH, 2025.

Os ícones no canto superior da tela são os botões de controle de reprodução do Scratch (Figura 30). A bandeirinha verde é usada para iniciar e a bolinha vermelha para parar.

Figura 30 – Botões de controle de reprodução no Scratch

Fonte: SCRATCH, 2025.

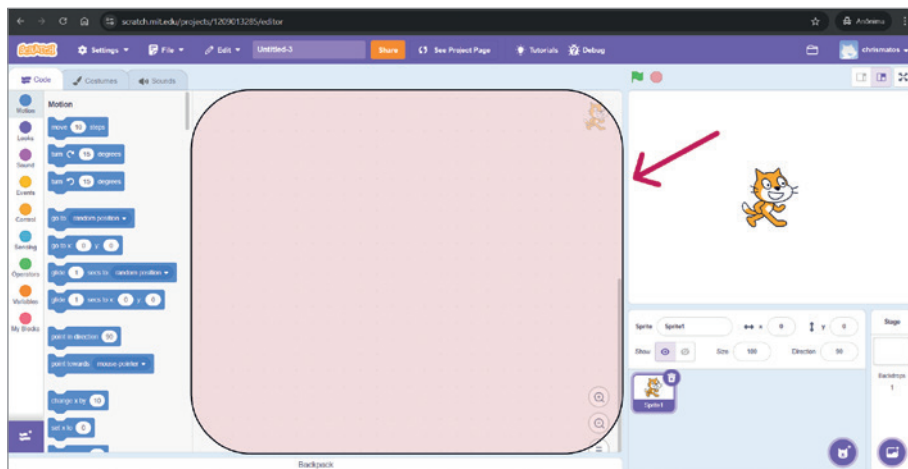
Na parte esquerda da tela, encontramos os blocos de comandos. Com esses blocos, podemos controlar o movimento, a aparência, o som, os eventos, o controle, os sensores, os operadores, as variáveis e até criar seus próprios blocos (Figura 31).

Figura 31 – Blocos de Comandos no Scratch

Fonte: SCRATCH, 2025.

No meio da tela, temos uma área de programação, para onde vamos clicar e arrastar os blocos de comando (Figura 32).

Figura 32 – Área de Programação no Scratch



Fonte: SCRATCH, 2025.

Agora que você já conhece a plataforma Scratch, que tal explorarmos juntos como aplicar o que aprendemos no Módulo 1? Vamos descobrir como usar os comandos de sequência, condição e repetição no Scratch para criar projetos incríveis!

Comandos de Sequência no Scratch

Agora que você já sabe o que é um algoritmo, entende o plano cartesiano e conhece os principais comandos no Scratch, vamos praticar a movimentação do personagem.

Você já brincou de seguir uma receita de bolo no Módulo 1. Para tudo dar certo, é preciso seguir os passos na ordem certa, não é? No mundo da programação, usamos o comando de sequência para fazer isso: ele organiza as ações para serem realizadas uma depois da outra, na sequência correta.

No Scratch, quando colocamos os blocos um embaixo do outro, estamos criando uma sequência de comandos. Por exem-

plo, se queremos que o gatinho diga “Oi!”, depois ande para a frente e por fim toque uma música, precisamos organizar os blocos nessa ordem.

Atenção! Sem a resposta correta, o computador não vai saber o que fazer primeiro. Com o comando de sequência, você garante que seu projeto siga passo a passo, como você planeja. Agora é a sua vez de experimentar e criar histórias e jogos que acontecem na ordem perfeita.

Para a criação de um código que permita mover o personagem para qualquer posição que quiser, apresentamos uma sequência de passos, como o que está na Figura 33.

É oportuno clicar e arrastar os blocos de comandos para a área de programação do Scratch para entender o que o código vai gerar.

Figura 33 – Código no Scratch para movimentação do personagem



Fonte: SCRATCH, 2025.

Movendo o Personagem no Scratch!

Para mover o personagem (como o gatinho do Scratch) para a esquerda, direita, ou para cima e para baixo, usamos comandos específicos. Veja como funciona:

- **Se você apertar a seta para a direita** no teclado, o gatinho vai andar 10 passos para a direita (ou seja, 10 posições no eixo X do plano cartesiano).
- **Se ele chegar à borda** do palco e continuar andando, ele automaticamente vira para a esquerda. E, se você apertar a seta para a esquerda, ele vai andar para o outro lado.

Agora, vamos ver como os passos funcionam:

- Da esquerda para a direita, os passos são positivos:
+1, +2, +3, +4, +... , +10.
- Da direita para a esquerda, os passos são negativos:
-1, -2, -3, -4, -... , -10.

Para andar para cima e para baixo, é só apertar as setas para cima e para baixo no teclado. Agora, em vez do eixo X (horizontal), usamos o eixo Y, que é o eixo vertical do plano cartesiano.

Com esses comandos, você pode controlar o personagem para andar por toda a tela de forma divertida. Agora é a sua vez de experimentar!

Nos “Blocos de comando” você encontrará uma breve descrição de funcionalidade de cada um desses blocos. Não deixe de explorar para conhecer mais!

Comandos de Condição no Scratch

Você já brincou de jogo da memória ou de “Sim ou Não”? Nessas brincadeiras, você toma decisões: Se acertar a carta certa, então ganha pontos. Senão, tente de novo. No Scratch, usamos o comando

de condição para fazer o computador tomar decisões da mesma forma.

O comando de condição funciona como uma pergunta. Se algo acontecer, o personagem faz uma ação. Se não acontecer, ele faz outra coisa ou não faz nada. Por exemplo:

- Se o personagem tocar na borda, então ele volta.
- Se o jogador ganhar 10 pontos, então o jogo termina.

Com as condições, o computador decide o que fazer em cada situação, deixando os projetos mais dinâmicos e inteligentes. Esse comando é muito usado em jogos quando um personagem só pode passar de fase se tiver uma chave ou quando perde uma vida ao tocar em um obstáculo.

Os comandos de condição no Scratch, frequentemente referidos como comandos de seleção, permitem que o programa tome decisões com base em condições específicas.

Para entender melhor como um computador pode tomar decisões com base em alguma condição, vamos fazer um código de adivinhação. Na figura 34 veremos a condição SE.

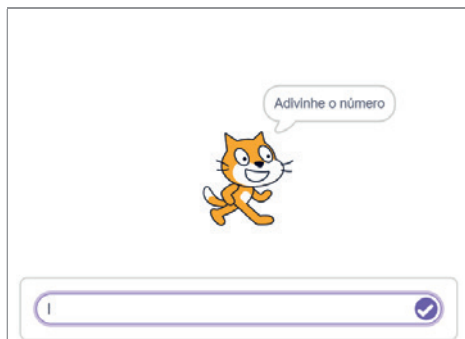
Figura 34 – Condição no Scratch com comando Se



Fonte: SCRATCH, 2025.

Primeiro será apresentada a mensagem “Adivinhe o número” e logo abaixo um espaço para receber a resposta. O quadrinho azul é o espaço destinado para guardar a resposta (Figura 35).

Figura 35 – Mensagem de digitar resposta no Scratch do Adivinhe o número



Fonte: SCRATCH, 2025.

Quando você digitar a resposta, o programa irá observar se ele é igual a 4. Então, se a resposta digitada for 4, ele realizará a ação de dizer "Parabéns!" (Figura 36).

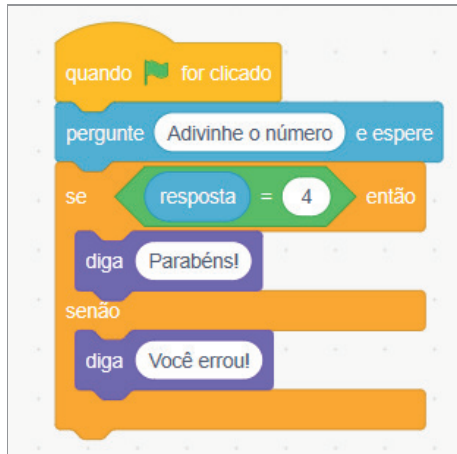
Figura 36 – Mensagem de Parabéns no Scratch



Fonte: SCRATCH, 2025.

Na Figura 37, foi adicionado o "senão" aos comandos do Scratch. Ele será acionado se uma resposta digitada não for 4, e então os comandos dentro dele serão realizados.

Figura 37 – Comando Senão no Scratch



Fonte: SCRATCH, 2025.

Se você digitar, por exemplo, o número 2, estará acionando o “senão”, e a mensagem que será exibida é: “Você errou!” (Figura 38).

Figura 38 – Ação de mensagem do comando Senão no Scratch



Fonte: SCRATCH, 2025.

Comandos de Repetição no Scratch

Você já viu que algumas tarefas precisam ser feitas várias vezes para dar certo? Como quando você joga bola e precisa chutar várias vezes até marcar um gol, ou quando precisa escovar os dentes todos os dias? No Scratch, usamos o comando de repetição para que o computador repita uma ação quantas vezes for necessário, sem precisar que você diga o mesmo comando toda hora.

Com o comando de repetição, podemos programar o personagem para fazer a mesma coisa várias vezes automaticamente. Por exemplo:

- repetir 10 vezes: o personagem pode dar 10 passos seguidos.
- repetir para sempre: ele pode continuar dançando até você mandar parar!

A repetição é muito útil porque economiza tempo e organiza o código, tornando tudo mais fácil de programar. Agora que você sabe como funciona, que tal experimentar no Scratch e ver seu personagem repetir ações várias vezes? Divirta-se criando e programando com repetições!

Para explorarmos inicialmente o conceito de repetição, vamos criar um simples contador de passos para o nosso personagem. O código apresentado na Figura 39 mostra como será feita a repetição.

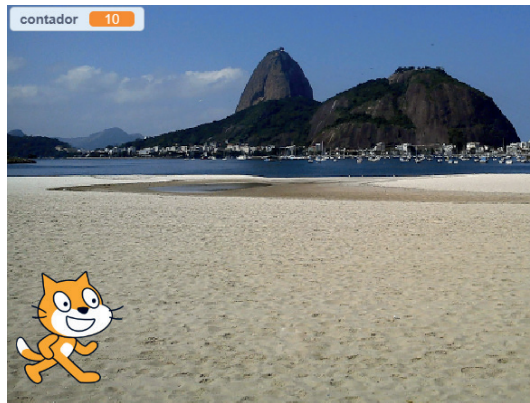
Figura 39 – Comando de repetição no Scratch



Fonte: SCRATCH, 2025.

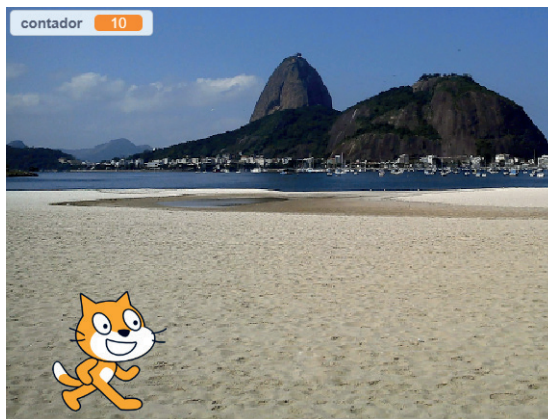
Primeiro, criamos uma variável chamada Contador e definimos seu valor inicial como zero. No restante do código, decidimos que o personagem se moveria 10 passos e, a cada passo dado, somaria 1 ao valor anterior da variável Contador, como mostrado nas Figuras 40 e 41. A posição do gatinho está inicialmente na borda da praia (Figura 40) e na Figura 41 já deslocou um pouco. A variável Contador mostra 10 em ambas as imagens, confirmando a execução do comando.

Figura 40 – Uso do contador no Scratch Inicial



Fonte: SCRATCH, 2025.

Figura 41 – Uso do contador no Scratch Final



Fonte: SCRATCH, 2025.

Para entendermos melhor como o comando de repetição SEMPRE funciona, vamos fazer um algoritmo para que o personagem Dinossauro dance. Para tanto utilizamos os mesmos comandos que fizemos o gatinho andar 10 passos na praia, substituindo o comando repita até pelo comando sempre. A Figura 42 traz o Dinossauro numa posição inicial da dança e a Figura 43 em outro passo da dança. É muito divertido ver o Dinossauro dançar!

Figura 42 – Uso do comando SEMPRE no Scratch para Dinossauro dançar –
Posição Inicial



Fonte: SCRATCH, 2025.

Figura 43 – Uso do comando SEMPRE no Scratch para Dinossauro dançar –
Posição Final



Fonte: SCRATCH, 2025.

Outra maneira de usar o comando SEMPRE é pedir que um personagem mude sua fantasia. A Figura 44 mostra como fazer isso. O próprio personagem tem várias fantasias, e podemos programá-lo para trocá-las em ciclo, dando a impressão de que ele está dançando.

Figura 44 – Comando repetição sempre



Fonte: SCRATCH, 2025.

Variáveis no Scratch

Você já teve uma caixinha onde guardar coisas especiais, como moedas, figurinhas ou brinquedos pequenos? No Scratch, usamos algo chamado variáveis, que são como essas caixinhas de itens especiais. Eles servem para guardar informações que o computador pode usar enquanto você programa.

Uma variável pode salvar números ou palavras e seu valor pode mudar conforme o que acontece no jogo ou projeto. Por exemplo:

- uma variável pode salvar o número de pontos que você ganha em um jogo.
- você também pode salvar o tempo que falta para encerrar uma corrida.

As variáveis são muito úteis porque permitem que o computador lembre de coisas importantes e úteis para tomar decisões ou fazer cálculos. Agora que você sabe o que é, vamos aprender a criar

e usar variáveis no Scratch para deixar seus projetos ainda mais incríveis!

Para usar esse comando no Scratch, você deve ir até a parte do menu onde está escrito “Variáveis” e clicar em “Criar uma Variável” (Figura 45).

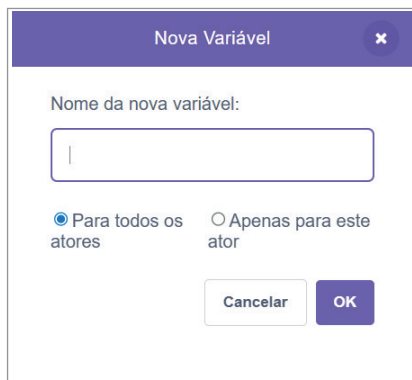
Figura 45 – Opção de Criar uma variável no Scratch



Fonte: SCRATCH, 2025.

Depois, é só escolher o nome que você quer dar para ela, como está demonstrado na Figura 46.

Figura 46 – Colocar nome em nova variável no Scratch



Fonte: SCRATCH, 2025.

Imagine que você está criando um jogo de aventura no Scratch. Cada vez que o personagem principal (um gatinho aventureiro) encontra uma estrela, ele ganha 1 ponto. Mas como o computador vai lembrar esses pontos? É aí que entra a variável!

Podemos criar uma variável chamada “Pontos” para salvar a quantidade de estrelas que o gatinho coletou. Cada vez que ele encontra uma estrela, o valor da variável aumenta em +1. Veja como funciona:

1. No começo do jogo, a variável Pontos vale 0 (porque ele ainda não pegou nenhuma estrela).
2. Quando o gatinho pega a primeira estrela, a variável Pontos muda para 1.
3. Na estrela seguinte, a variável Pontos passa a ser 2. E assim por diante.

Com essa variável, o computador consegue lembrar quantos pontos o gatinho ganhou durante o jogo. No final, o jogador pode ver o total de pontos e tentar bater o próprio recorde na próxima vez.

Agora é a sua vez de criar variáveis e inventar jogos incríveis no Scratch!

Praticando no Scratch

1. Vamos começar a praticar o uso da plataforma Scratch treinando com o que aprendemos sobre plano cartesiano.

1º PASSO

Nessa atividade a proposta é mudar o cenário atual para o do plano cartesiano (Xy-grid).

2º PASSO

Posicione o personagem nas posições (a, b, c e d) e em seguida informe em que quadrante essa posição se localiza.

- (-50, -50) _____
- (-150, 100) _____
- (-50, 100) _____
- (200, 50) _____

2. Nesta atividade, criaremos um cenário com 2 personagens diferentes, olhando um para o outro. Observe os passos descritos para produzir as ações e posições necessárias para o cenário.

1º PASSO

Adicione o cenário e os personagens Duck (pato) e Cat (gato).

2º PASSO

Na guia Fantasias, aperte CTRL + A em toda a imagem de um dos personagens e use o botão (Espelhar Horizontalmente) para fazer o personagem olhar para o outro lado.

3º PASSO

Mude a posição dos personagens utilizando o que foi aprendido sobre planos cartesianos.
A posição correta do Duck (pato): (X: 165 / Y: -120)=(165, -120).
posição correta do Cat (gato): (X: -165 / Y: -120)=(-165,-120).

3. Você deve elaborar comandos para o Mapa de Caça ao Tesouro. O código da Figura 33 apresenta uma das possibilidades de instruções que poderiam ser utilizadas para fazer com que o seu personagem realize o trajeto da espada até o baú do tesouro (Figura 47).

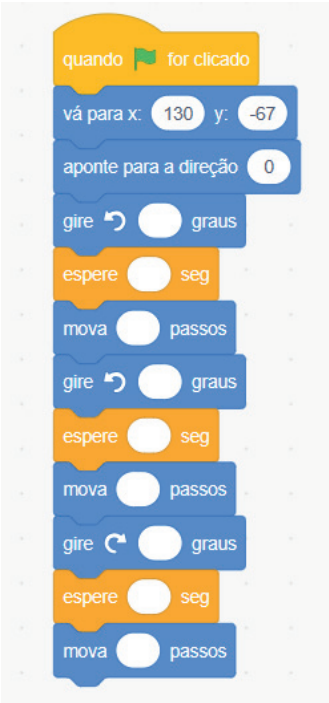
Figura 47 – Mapa de Caça ao Tesouro



Fonte: GEMINI, 2025.

Ops! Acho que houve algum bug, pois os valores do código estão todos apagados. Descubra os valores que permitam que o personagem consiga realizar o caminho em vermelho no mapa e os anote aqui na Figura 48.

Figura 48 – Código do Mapa de Caça ao Tesouro



Sabe o que é “Bug”?
Haha, eu sei! Um bug de software é um erro, falha ou falha no design, desenvolvimento ou operação de software de computador que faz com que ele produza um resultado incorreto ou se comporte de maneira não intencional.



Fonte: Autoria própria, 2025, utilizando SCRATCH, 2025.

4. Para esta atividade, vamos continuar utilizando o cenário criado na atividade 2 do módulo 2 para criar uma pequena história, em que dois personagens conversam.

1º PASSO	Indique a posição inicial dos nossos personagens.
2º PASSO	Selecione o personagem Cat (gato) e faça-o andar para perto do Duck (pato).
3º PASSO	Envie uma “mensagem” que mostre para o Duck (pato) que o Cat (gato) se aproximou.
4º PASSO	Escreva a mensagem que o Duck (pato) dirá ao Cat (gato).
5º PASSO	Envie uma “mensagem” que mostre para o Cat (gato) que o Duck (pato) acabou de falar algo.

6º PASSO

Escreva a mensagem que o Cat (gato) dirá ao Duck (pato).

**DESAFIO 1**

A movimentação de um personagem em direção ao outro não ficou muito legal. Selecione o personagem e adicione uma sequência de blocos para que essa movimentação se torne melhor.

5. Para praticar os comandos condicionais SE e SENÃO vamos criar um robô calculador, conforme os comandos a seguir:

1º PASSO

Adicione o personagem Robot e o cenário.

2º PASSO

Insira um bloco para o usuário informar ao robô que tipo de sinal (+ - * /) irá utilizar para a operação que quer fazer.

3º PASSO

Insira um bloco para que o robô saiba que tipo de operação irá realizar depois que o usuário informar qual sinal será usado: SE ... ENTÃO ... SENÃO.

4º PASSO

Insira um bloco para que você informe ao robô os valores para a operação.

5º PASSO

Utilize os blocos de aparência para que a resposta apareça na tela.

6º PASSO

Insira um comando de finalização para o caso de o sinal informado estiver errado.

DESAFIO 2

O nosso Robô Calculadora está bem incompleto, pois ele só faz um cálculo. O desafio será fazer com que o robô faça mais de uma operação diferente. Deixe também o nosso robô mais animado criando uma animação para ele não ficar parado.

6. Para essa atividade, nos basearemos em como funciona o nosso sistema solar. Temos que usar blocos de movimento para mostrar o movimento de rotação do planeta Terra e iremos também girar o Sol para parecer que ele está sendo animado a partir dos seguintes passos:

1º PASSO Adicione o cenário Stars ou Galaxy e os personagens Earth para representar o planeta Terra e Sun para representar o Sol.

2º PASSO Configure a Earth (Terra) para que ela sempre inicie apontada para a mesma direção.

3º PASSO Utilize o bloco REPITA ... VEZES para programar quantos giros que a Earth (Terra) irá fazer.

4º PASSO Teste a programação e veja se o planeta terra fez um giro completo. Se o planeta não fizer um giro completo, ajuste o bloco de repetição para descobrir quantas vezes o planeta precisa girar para realizar um giro completo.

DESAFIO 3

Agora que programamos o personagem Earth (Terra) para realizar uma quantidade definida de giros, faça com que o Sun (Sol) fique girando o tempo todo. Aproveite também e crie novos planetas utilizando a criação de atores deixando-os animados.



REVISÃO DO MÓDULO 3

Parabéns! Você concluiu o módulo 3 de Introdução ao Scratch!

Agora que você explorou o ambiente do Scratch e aprendeu sobre comandos de sequência, condição, repetição e variáveis, já está pronto para criar seus próprios projetos incríveis! Cada um desses conceitos é como uma peça de um grande quebra-cabeça que, quando juntamos, fazemos seus jogos e animações funcionarem corretamente.

- Comandos de sequência ajudam a organizar as ações na ordem certa.
- Comandos de condição ensinam o computador a tomar decisões com base em situações específicas.
- Comandos de repetição permitem que o computador repita tarefas automaticamente.
- Variáveis guardam informações importantes para que o programa possa usar durante a execução dos comandos.

Lembre-se de que praticar é essencial! Quanto mais você experimentar e brincar com os blocos no Scratch, mais fácil e divertido vai ser aprender.

As atividades propostas foram feitas para te ajudar a entender cada conceito e, ao mesmo tempo, soltar a imaginação. Não tenha medo de errar. Errar faz parte do aprendizado.

Agora é a sua vez! Continue explorando, praticando e criando seus próprios jogos e animações. Mal podemos esperar para ver tudo o que você vai inventar!

Nós nos encontramos no próximo módulo para aprender ainda mais!

Anotações

[illegible]

MÓDULO 4 – CRIANDO ANIMAÇÕES COM SCRATCH



Você sabia que pode criar suas próprias animações no Scratch? Assim como nos filmes e desenhos que você assiste, aqui no Scratch você pode fazer seus personagens se mexerem, falarem e até dançarem! Tudo isso usando os comandos que você já aprendeu.

Com as animações no Scratch, você pode contar histórias, criar aventuras e até inventar cenários incríveis. Para isso, vamos aprender a:

- mover os personagens de um lado para o outro;
- mudar a fantasia deles para que pareçam estar se movimentando;
- adicionar sons para deixar tudo mais divertido!

Criar animações é como brincar de dirigir um filme: você controla o que acontece, quem aparece e o que cada personagem faz. E a melhor parte? Tudo isso é muito fácil e divertido de programar no Scratch!

Antes de começarmos a criar nossas animações, é importante conhecer os blocos de comando que usaremos na plataforma e que permitem as animações. Eles são as ferramentas que darão vida aos nossos personagens e farão tudo funcionar do jeito que queremos. Vamos explorar juntos e descobrir como esses blocos podem nos ajudar a criar animações incríveis.

Blocos de Movimento no Scratch

Os blocos de movimento no Scratch permitem que você controle como os personagens se movem em seu projeto. Eles iniciam e controlam a mudança de movimento dos personagens em diferentes situações, dando vida às suas criações.

Imagine que você está criando uma corrida divertida no Scratch e o personagem principal é o gatinho. Ele precisa correr para a direita até chegar na linha de chegada!

Prontos
para
começar?
Vamos lá!



Usando os blocos de movimento, você pode programar o gatinho assim:

1. quando a bandeira verde é clicada, o gatinho começa a se mover;
2. ao acionar o Bloco de movimento para “Mover 10 passos” o gatinho anda para a frente;
3. configurando o Bloco de condição, se o gatinho tocar na borda do palco, ele vira para o lado contrário e continua correndo.

E pronto! O gatinho vai correr pelo palco e virar quando chegar nas bordas, como se estivesse em uma pista de corrida! Agora é a sua vez de experimentar e ver como os personagens se movem no Scratch.

Que tal experimentar? Você pode criar um cachorro que pega frisbee, um pássaro que voa de um lado para o outro ou qualquer outro personagem que você quiser!

Lembre-se: a diversão está garantida!

Vamos lá, solte sua imaginação e divirta-se criando animações incríveis! Veja que ideias legais para experimentar:

- Um coelho pulando em uma cenoura: Faça o coelho pular para pegar a cenoura e depois voltar para o chão.

- Um peixe nadando em um aquário: Faça o peixe se mover para frente e para trás, subindo e descendo.
- Uma bola quicando: Faça a bola cair e quicar no chão várias vezes.

Com os blocos de movimento, o céu é o limite!

Além de movimentarmos nossos personagens, é importante pensarmos em como eles parecerão enquanto se movem. Isso pode deixar seu projeto mais interessante e divertido. É aí que os blocos de aparência entram em ação. Vamos conhecer mais sobre eles!

Blocos de Aparência no Scratch

Os blocos de aparência no Scratch permitem que você controle e personalize a maneira como os personagens aparecem na tela do seu projeto. Eles funcionam como interruptores, mudando a aparência dos personagens em resposta a diferentes situações ou interações.

Imagine que o gatinho do Scratch vai participar de uma festa à fantasia e precisa trocar de roupa várias vezes para se preparar. Usando os blocos de aparência, você pode programar o gatinho para mudar de fantasia enquanto ele se apresenta na festa.

Aqui está um exemplo divertido:

1. Quando a bandeira verde é clicada, o gatinho aparece na tela e diz: “Estou pronto para a festa!”.
2. Em seguida, você usa o bloco “Próxima fantasia” para o gatinho trocar de roupa.
3. Bloco de repetição: Com o comando “Sempre”, ele vai continuar trocando de fantasia enquanto dança.
4. Use o bloco “Dizer” para que o gatinho fale: “Gostaram da minha fantasia?”.

Com esses blocos de aparência, o gatinho parece estar dançando

e se divertindo na festa, mudando de roupa sem parar! Agora é a sua vez de criar um projeto e deixar seus personagens ainda mais divertidos!

Imagine: você pode criar um peixe que muda de cor quando se aproxima de um coral ou um pássaro que muda de cor para se camuflar nas árvores! Ou ainda: Você pode criar um unicórnio que muda de cor quando usa sua magia ou um dragão que muda de cor quando está com raiva!

Vamos experimentar algumas ideias?

- Uma flor que abre e fecha: faça a flor abrir suas pétalas e mudar de cor.
- Um fantasma que fica transparente: faça o fantasma ficar cada vez mais transparente até sumir.
- Um carro que muda de cor: faça o carro mudar de cor quando acelerar.

Com os blocos de aparência, você pode criar personagens incríveis e histórias mágicas!

Além de decidirmos a aparência dos nossos personagens, você também pode adicionar sons que acompanhem as ações na tela, tornando assim o projeto mais divertido. Então vamos explorar os blocos de som???

Blocos de Som no Scratch

Os blocos de som no Scratch permitem que você controle e adicione áudio em seu projeto. Eles funcionam como botões que iniciam e param o som em resposta a diferentes situações.

Vamos criar uma banda divertida no Scratch, com um gatinho tocando bateria, um cachorrinho tocando guitarra e um pinguim cantando! Usando os blocos de som, você pode programar cada personagem para tocar seu instrumento ao receber o comando.

Aqui está como fazer:

1. Quando a bandeira verde é clicada, o gatinho começa a tocar bateria com o bloco “Iniciar som de bateria”.
2. Depois de 2 segundos, o cachorrinho entra tocando guitarra com o bloco “Iniciar som de guitarra”.
3. O pinguim pode começar a cantar usando o bloco “Iniciar som de canto” logo em seguida.
4. Para deixar mais divertido, você pode usar o bloco “Tocar som até terminar” para que o show continue até o fim da música.

Com esses blocos, você pode criar uma apresentação incrível e animada! Agora é a sua vez de experimentar e montar sua banda no Scratch. Será que você conseguirá deixar o show ainda mais legal?

Outra possibilidade: você pode criar uma história com vários animais, cada um fazendo um som diferente. Imagine uma floresta com vários animais fazendo barulhos! Qual será o resultado???

Lembre-se: com os blocos de som, você pode criar histórias ainda mais divertidas e interativas!

Embora os blocos de som e blocos de aparência ajudem a dar vida ao projeto, é necessário usar blocos que os acionem em momentos certos. É aí que entram os blocos de eventos, que determinam como e quando as ações, como os sons, vão começar a funcionar como gatilhos para essas ações. Vamos conhecê-los!!!

Blocos de Eventos no Scratch

Os blocos de eventos no Scratch são importantes para controlar o início e as ações iniciais de um programa. Eles funcionam como gatilhos que ativam o programa em resposta a diferentes estímulos. Assemelham-se a interruptores que dão início ao que acontece no seu projeto.

Imagine que você criou um jogo no Scratch chamado Corrida Maluca, onde o gatinho, o cachorrinho e o pinguim participam de uma corrida divertida.

Vamos aprender sobre eventos?



Para começar a corrida, usamos um bloco de evento que funciona como um sinal de largada. Veja como funciona:

1. Bloco de evento: Quando a bandeira verde for clicada, a corrida começa.
2. Cada personagem tem um bloco de evento separado que ativa sua ação. Por exemplo:
 - o gatinho começa a correr com o bloco "Mover 10 passos";.
 - o cachorrinho toca um som de latido antes de começar a correr;
 - O pinguim grita "Vamos lá!" e começa a se mover.
3. Bloco de repetição: Todos continuam correndo até alcançarem a linha de chegada.

Com o bloco de evento, você garante que todos os personagens comecem a correr ao mesmo tempo, como se um juiz tivesse dado o sinal de partida. Agora é sua vez de usar esses blocos para criar jogos e competições super divertidas no Scratch! Quem será o vencedor da sua Corrida Maluca?

Uma outra ideia interessante é produzir um jogo no qual você controla um personagem com o mouse. Com os blocos de eventos, podemos fazer com que nosso personagem faça coisas diferentes, dependendo do que você faz no computador!

Como vamos fazer isso?

1. Escolha um personagem: Escolha um personagem que você goste, como um alienígena, um super-herói ou até mesmo um animal fofo.
2. Crie um evento: use o bloco "quando o mouse é clicado" para fazer algo acontecer quando você clicar no seu personagem.

3. Crie uma ação: dentro do bloco de evento, você pode adicionar outros blocos para fazer o personagem se mover, mudar de cor, fazer um som ou qualquer outra coisa que você quiser!

Dica: Você pode usar outros blocos de evento, como “quando a tecla espaço é pressionada” ou “quando a bandeira verde é clicada” para criar diferentes interações.

Com os blocos de eventos você pode criar jogos incríveis e cada vez mais interativos!

Para entender as interações, algo importante para o desenvolvimento dos nossos programas é definir o ritmo das ações. Isso é feito com os blocos de controle. Essas funções são essenciais para garantir que o programa funcione como planejado, fornecendo instruções sobre como e quando as diferentes partes do programa devem acontecer. É este o assunto que vamos discutir agora.

Blocos de Controle no Scratch

Os blocos de controle no Scratch são usados para orientar como o programa vai funcionar. Eles permitem que você escolha como e quando os comandos do seu programa serão executados. Os blocos de controle são como “comandos mágicos” que dizem ao nosso programa o que fazer e quando fazer. Eles são como as rédeas de um cavalo, guiando o caminho do nosso projeto.

Imagine que você está criando um projeto no Scratch chamado “Dança do Robô”. O personagem principal é um robô que vai dançar ao som de uma música. Para fazer com que a dança tenha um ritmo legal e coordenado, você pode usar blocos de controle. Veja, a seguir, como isto funciona.

1. Bloco de evento: quando a bandeira verde for clicada, o robô começa a dançar.
2. Bloco de repetição: use o bloco “Repetir 10 vezes” para que o robô dê um passo de dança várias vezes. Por exemplo: “Mover 10 passos”, depois “Girar 15 graus”.

3. Bloco “Esperar 1 segundo”: esse bloco faz o robô pausar por um instante entre os passos para que a dança tenha ritmo
4. Bloco “Sempre”: Se você quiser que o robô dance sem parar, use o bloco “Sempre” e ele vai continuar dançando até você parar o programa.

Com esses blocos de controle, você pode definir quando o robô começa a dançar, quantas vezes ele repete cada passo e quando ele faz uma pausa. Agora é a sua vez! Experimente usar os blocos de controle e crie coreografias divertidas para seu robô dançar no Scratch!

E outras ideias podem ser experimentadas, tais como:

- Uma contagem regressiva: crie uma contagem regressiva para um foguete que vai decolar.
- Um esconde-esconde com um fantasma: crie um jogo para fazer um fantasma aparecer e desaparecer em lugares diferentes da tela.
- Um jogo de perguntas e respostas: crie um jogo no qual o computador faz perguntas e você precisa responder.

Viu como os blocos de controle podem te ajudar a criar histórias e jogos ainda mais interessantes?

Enquanto os blocos de controle determinam como o programa deve funcionar, os blocos de sensores deixam seu projeto ainda mais interativo. Vamos descobrir como esses blocos fazem isso!

Blocos de Sensores no Scratch

Os blocos de sensores no Scratch são usados para fazer o projeto interagir com o ambiente. Eles permitem detectar interações, como cliques do mouse, teclas pressionadas e colisões entre personagens e objetos.

Vamos criar um jogo chamado Pescaria Mágica no Scratch. O objetivo do jogo é usar um personagem pescador para pegar o maior número de peixes possíveis e, para isso, o Scratch vai usar blocos de sensores para detectar interações e tornar o jogo mais divertido!

Veja como você pode usar os blocos de sensores:

1. Sensor de clique do mouse: quando o jogador clica com o mouse na vara de pesca, o personagem vai lançar a linha na água.
 - Bloco de evento: “Quando este objeto for clicado”.
2. Sensor de profundidade: se a linha de pesca encostar em um peixe, o jogo detectará a profundidade e o peixe será capturado.
 - Bloco de sensor: “Se tocar no peixe”.
3. Bloco de operador: use um operador para contar quantos peixes foram capturados e classificar a pontuação em uma variável chamada “Pontos”.
4. Bloco de som: quando o peixe é capturado, um som de vitória toca para deixar o jogo mais animado.

Com esses blocos de sensores, o jogador pode interagir com o ambiente e se divertir, pegando o máximo de peixes possível. Agora é sua vez de criar um jogo incrível no Scratch e ver como as interações podem deixar tudo mais emocionante!

Você pode criar um jogo com o qual você controla um carro e precisa evitar os outros carros ou um jogo com o qual você ajuda um pássaro a encontrar seu ninho! Com os blocos de sensores, você pode criar jogos ainda mais interativos e divertidos!

É importante entender que os blocos de sensores ajudam o projeto a **interagir com o ambiente**, mas para que essas interações resultem em ações específicas, às vezes é necessário fazer **cálculos e comparações**. É aí que usamos os **blocos de operações** que são essenciais para **processar informações** e tomar **decisões** com base nas detecções dos sensores.

Blocos de Operadores no Scratch

Os blocos de operadores no Scratch são usados para fazer operações matemáticas, lógicas e de texto no seu programa. Eles permitem que você trabalhe com números e textos para realizar cálculos e tomar decisões.

Vamos desenvolver um jogo no Scratch chamado A Máquina de Pontos Mágicos. Nesse jogo, o jogador precisa somar pontos toda vez que o personagem coletar uma estrela e multiplicar a pontuação se ele encontrar um item especial!

Aqui está a sequência necessária para usar os blocos de operações e deixar seu jogo incrível:

1. Bloco de operador de soma:

- o personagem ganha 10 pontos sempre que ele pegar uma estrela;
- utilize o operador “+” para somar 10 ao valor atual da variável “Pontos”.

Exemplo:

- Se “Pontos” = 20, depois de pegar uma estrela: $\text{Pontos} = 20 + 10 = 30$.

2. Bloco de operador de multiplicação:

- os pontos são dobrados se o personagem pegar um item especial;
- utilize o operador “x” para multiplicar a pontuação por 2.

Exemplo:

- Se “Pontos” = 30 e o personagem pega o item especial: $\text{Pontos} = 30 \times 2 = 60$

Com os blocos de operadores, você pode criar jogos mais dinâmicos, calculando antecipadamente os movimentos e tomando decisões com base nos resultados. Agora é com você. Use esses blocos para inventar desafios divertidos!

Outras ideias divertidas para você testar:

- Uma calculadora mágica: faça cálculos simples como soma, subtração, multiplicação e divisão.
- Um jogo de dados: simule o lançamento de um dado e mostre o resultado.
- Um jogo de pedra, papel, tesoura: faça o computador escolher uma opção e compare com a sua.

Com os blocos de operadores, você pode criar jogos que envolvem lógica e raciocínio!

Você pode fazer cálculos e tomar decisões com base neles para seu projeto, mas, às vezes, é necessário guardar ou atualizar essas informações. Para isso, usamos os blocos de variáveis. Vamos conhecer um pouco mais sobre esses blocos!!!

Blocos de Variáveis no Scratch

Os blocos de variáveis no Scratch o ajudam a criar e controlar variáveis. Uma variável é como uma caixa onde você pode salvar e atualizar informações. Por exemplo, se você criar uma variável chamada PONTOS, sempre que alguém ganhar pontos, você adiciona 1 ponto a essa variável e guarda o novo total de pontos nela.

Imagine que você está criando um jogo no Scratch chamado Caça ao Tesouro. O personagem principal é um pirata, e o objetivo do jogo é encontrar as moedas de ouro espalhadas pelo mapa. Para salvar a pontuação do jogador, você pode usar blocos de variáveis.

Veja como funciona:

1. Criando uma variável: no início do jogo, crie uma variável chamada "Moedas" e defina seu valor inicial como zero (pois o pirata ainda não coletou nenhuma moeda).
2. Usando a variável para contar: cada vez que o pirata encontra uma moeda, o valor da variável "Moedas" aumenta em 1. Use o bloco "Mudar Moedas por 1" para fazer isso.

Exemplo:

- No começo, a variável "Moedas" = 0.
 - O pirata pega a primeira moeda: "Moedas" = 1.
 - Depois, ele encontra mais duas moedas: "Moedas" = 3.
3. Mostrando o total: no final do jogo, use o bloco "Dizer" para que o pirata fale: "Você coletou X moedas!", onde X será o valor salvo na variável.

Com esse uso de variáveis, o jogo se torna mais interessante, pois o jogador pode ver sua pontuação mudando em tempo real. Agora é a sua vez de criar um jogo e experimentar com variáveis no Scratch. Será que você consegue fazer o pirata encontrar todas as moedas?

Aqui tem outras ideias para você explorar:

- Um jogo de plataforma: use uma variável para contar o número de moedas coletadas, bem similar ao jogo Caça ao Tesouro, que está de exemplo.
- Um jogo de corrida: use uma variável para marcar o tempo de cada corrida.
- Um jogo de tiro: use uma variável para contar o número de alvos atingidos.

Com os blocos de variáveis, você pode criar jogos mais complexos e desafiadores!

E depois de aprender a usar os blocos de variáveis, você pode querer deixar o algoritmo do seu projeto mais fácil de entender. Para isso, você pode usar os Meus Blocos. Vamos entender o que são?!

Meus Blocos no Scratch

Em “Meus Blocos”, você pode criar e personalizar seus próprios blocos, quase como se estivesse criando suas próprias funções. Isso ajuda a reduzir a quantidade de blocos no desenvolvimento do seu projeto, tornando-o mais simples e fácil de entender.

Imagine que você esteja criando um novo jogo no Scratch chamado Aventura do Herói. Nesse jogo, o herói precisa fazer um super salto para atravessar obstáculos e continuar sua missão. Então, para facilitar o uso desse super salto várias vezes no jogo, você pode criar um bloco personalizado usando Meus Blocos. Assim, fica mais fácil e organizado programar o salto sempre que precisar.

Veja como usar Meus Blocos para isso:

1. Criando o bloco “Super Salto”

- Vá até Meus Blocos e crie um bloco chamado “Super Salto”.

2. Programando o super salto

- Dentro do bloco, adicione comandos de movimento, como “Mover 10 passos” para frente e “Mudar Y por 20”, para que o herói pule.
- Depois, use o comando “Esperar 0,5 segundos” para o herói ficar no ar e, em seguida, “Mudar Y por -20”, para que ele caia suavemente no chão.

3. Usando o bloco várias vezes

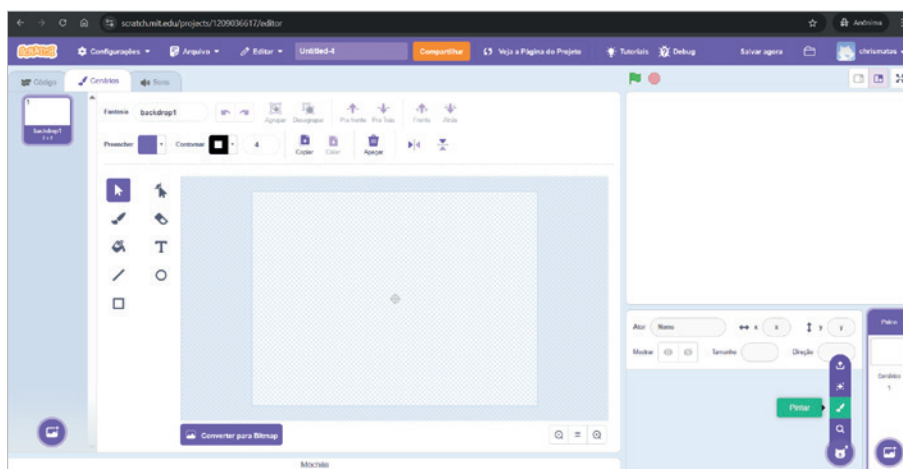
- Agora que você criou o bloco “Super Salto”, pode usá-lo sempre que quiser que o herói pule um obstáculo no jogo, sem precisar refazer todo o código.

Com o bloco personalizado “Super Salto”, você economiza tempo e deixa o código do seu jogo mais organizado. Que tal criar outros blocos personalizados e deixar seus jogos ainda mais legais? Vamos nessa!

Agora que você já conhece os vários blocos de comando do Scratch, que tal aprendermos a criar o nosso próprio cenário e nossos próprios personagens? Vamos lá!

Para criar um novo cenário, basta clicar no ícone do cenário e, em seguida, no símbolo do pincel (Pintar). Seguindo essas instruções, você será direcionado para a aba de edição, como mostrado na Figura 49.

Figura 49 – Tela da Minha Plataforma



Fonte: SCRATCH, 2025.

Nessa aba, podemos desenhar como quisermos, pintar com qualquer cor, adicionar textos e formas geométricas – a criatividade é o limite! Para criar um personagem, seguimos o mesmo processo que usamos para criar um novo cenário.

Agora que você já conhece os blocos de comando e aprendeu a criar cenários e personagens, está na hora de praticar esses conhecimentos. Vamos criar algumas animações!

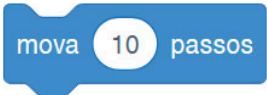
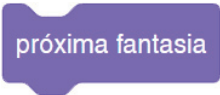
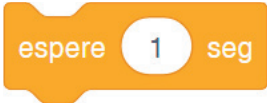
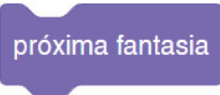
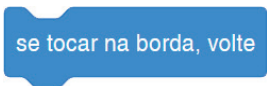
Praticando juntos

Primeiro, vamos fazer um pequeno projeto de animação juntos, e para isso vamos precisar seguir uma lista de passos para a montagem do cenário e a programação de comandos necessários.

Montar o cenário:

1º PASSO	Adicione o cenário Bedroom 1 (quarto)
2º PASSO	Posicione o personagem Cat (gato) no canto esquerdo na parte de baixo do cenário

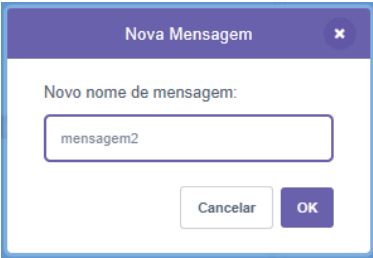
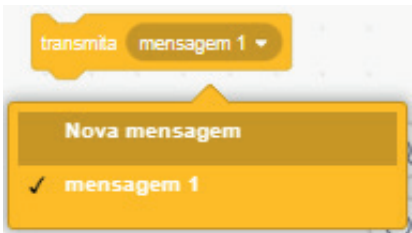
Programação:

3º PASSO	Adicione o Bloco de Evento	
4º PASSO	Junte o Bloco de Controle SEMPRE abaixo do bloco anterior	
5º PASSO	Dentro do bloco anterior, adicione o Bloco de Movimento	
6º PASSO	Abaixo do bloco de movimento, adicione o Bloco de Aparência	
7º PASSO	Abaixo do bloco de aparência, adicione o Bloco de Controle e altere o 1 por 0.1	
8º PASSO	Abaixo do bloco de controle, adicione novamente o Bloco de Aparência	
9º PASSO	Abaixo do bloco de aparência, adicione o Bloco de Movimento	

10º PASSO Teste

Para criarmos a interação entre os personagens Cat e Ballerina, vamos precisar adicionar novos blocos ao nosso algoritmo.

1º PASSO	Sem juntar aos demais blocos, adicione o Bloco de Eventos	
2º PASSO	Junte o Bloco de Aparência ao bloco anterior	
3º PASSO	Substitua "Olá" por "Estou caminhando, pois não tenho nada pra fazer!"	
4º PASSO	Adicione o bloco de Eventos ao bloco anterior e altere "mensagem1" para "mensagem 2", seguindo os passos da imagem a seguir:	



Agora que as alterações já foram feitas, vamos trabalhar com o outro personagem, a Ballerina!

Posicionamento:

1º PASSO	Adicione o personagem Ballerina
2º PASSO	Posicione o personagem em cima da cama do cenário

Programação do personagem Ballerina:

3° PASSO

Adicione o Bloco de Eventos



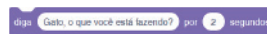
4° PASSO

Junte o Bloco de Aparência ao bloco anterior



5° PASSO

Substitua "Olá" por "Gato, o que você está fazendo?"



6° PASSO

Sem juntar aos outros blocos, adicione o Bloco de Eventos e altere "mensagem1" para "mensagem 2"



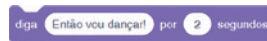
7° PASSO

Adicione ao bloco de eventos anterior, o Bloco de Aparência



8° PASSO

Substitua "Olá" por "Então vou dançar!"



9° PASSO

Teste

Agora é a sua vez de praticar o que aprendemos até agora, para isso resolva a atividade a seguir.

ATIVIDADES

ATENÇÃO!
Vamos praticar



1. Nesta atividade, nosso objetivo é criar um teclado musical, mas para isso precisamos desenhar as nossas teclas que irão reproduzir o som, com os passos a seguir:

1º PASSO	crie as teclas utilizando a guia Fantasia;
2º PASSO	configure a mudança de fantasia da tecla ao ser pressionada;
3º PASSO	programe cada tecla para que toque o som correspondente a ela;
4º PASSO	lembre-se de fazer com que a tecla volte para a primeira fantasia criada.

Desafio 4

Agora que várias teclas com sons diferentes foram adicionados ao nosso teclado musical, faça com que as teclas do nosso teclado do computador toquem sons no Scratch. Programa uma tecla para cada som diferente e adicione também as teclas pretas do teclado, faça com que elas diminuam e aumentem de tamanho.

Dica: Utilize o seguinte site para ajudar na criação e edição dos sons: <https://www.pianoeletronico.com.br>.

REVISÃO DO MÓDULO 4

Parabéns! Você concluiu o módulo 4 de Criação de Animações com Scratch!

Agora que você explorou e aprendeu sobre os blocos de movimento, aparência, som, eventos, controle, sensores, operadores, variáveis e Meus Blocos, está pronto para criar animações incríveis no Scratch! Cada bloco que você usou é uma ferramenta poderosa que torna seus projetos mais dinâmicos e divertidos.

Neste módulo, vimos como:

- Blocos de movimento fazem seus personagens se mexerem pela tela.
- Blocos de aparência mudam a fantasia e o visual dos personagens, deixando tudo mais criativo.
- Blocos de som adicionam música e efeitos, tornando o projeto mais animado.
- Blocos de eventos garantem que as ações comecem no momento certo, como ao clicar na bandeira verde.
- Blocos de controle definem o ritmo das ações, repetindo ou pausando movimentos.
- Blocos de sensores permitem que seus personagens interajam com o jogador e o ambiente.
- Blocos de operadores ajudam a fazer cálculos e comparações importantes para o jogo ou animação.
- Blocos de variáveis guardam informações essenciais, como pontos e tempos.
- Meus Blocos organizam e simplificam seu código, facilitando a criação de funções personalizadas.

Agora que você já conhece todas essas ferramentas, é hora de praticar! Quanto mais você explorar e testar, mais vai desenvolver sua criatividade e aprender novas maneiras de usar o Scratch. Cada atividade e exemplo desta unidade foi planejada para ajudá-lo a entender e aplicar os conceitos. Por isso, não tenha medo de experimentar, errar e tentar novamente – é assim que se aprende!

Com o Scratch, o limite é sua imaginação! Continue praticando e criando, e em breve você estará fazendo jogos, histórias e animações que impressionarão todos ao seu redor.

Pronto para o próximo desafio? A diversão e o aprendizado continuarão no próximo módulo!

Anotações

This image shows a full page of white paper with horizontal dashed lines, typical of primary-ruled notebook paper. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

MÓDULO 5 – CRIANDO JOGOS COM SCRATCH

No Scratch, você pode criar jogos incríveis e se divertir enquanto aprende a programar! O legal é que, com alguns blocos de comando e muita criatividade, você pode inventar desde jogos simples, como pega-pega, até aventuras mais desafiadoras, com vitórias e fases! Quer ver algumas ideias? Vamos explorar juntos!



1. Jogo de Pega-Pega

- **Como funciona:** o jogador controla um personagem (como o gatinho) e precisa pegar o máximo de moedas antes do tempo acabar.
- **Blocos usados:**
 - Movimento: para o personagem andar.
 - Sensores: para detectar quando o personagem toca nas moedas.
 - Variáveis: para contar a pontuação do jogador.

2. Corrida Maluca

- Como funciona: você cria uma pista e um programa com vários personagens para correr até a linha de chegada.
- Blocos usados:
 - Eventos: para iniciar a corrida ao clicar na bandeira verde.
 - Repetição: para os personagens correrem até o fim.
 - Aparência: para mudar a expressão dos personagens quando ganham ou perdem.

3. Jogo da Memória

- Como funciona: o jogador precisa encontrar pares de cartas iguais.
- Blocos usados:
 - Controle: para definir quando o jogador vira as cartas.
 - Sensores: para verificar se duas cartas são iguais.
 - Variáveis: para contar quantos pares foram encontrados.

4. Pula-Obstáculos

- Como funciona: o personagem precisa pular os obstáculos que aparecem na tela.
- Blocos usados:
 - Meus Blocos: para criar um bloco de “pular”.
 - Movimento: para o personagem ir para cima e para baixo.
 - Controle: para repetir a geração dos obstáculos.

Com essas ideias, você já pode começar a criar seus próprios jogos no Scratch! O mais divertido é que você pode personalizar tudo: mudar personagens, criar cenários diferentes e adicionar sons legais. Além disso, você pode experimentar o uso de fases e desafios avançados, deixando o jogo mais emocionante.

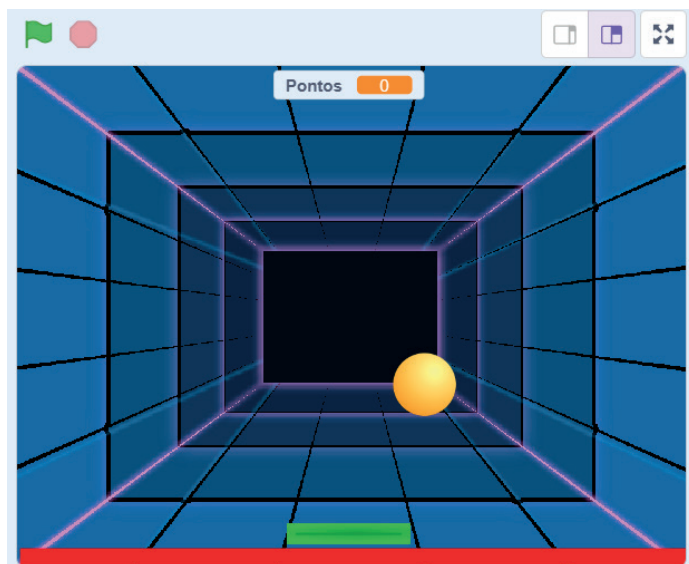
Agora é a sua vez! Pegue os blocos e comece a programar. Não tenha medo de errar — cada erro é uma chance de aprender algo novo!

Vamos praticar juntos?

Jogo do Pong

No jogo do Pong, você controla uma raquete para rebater uma bolinha e, assim, vai conseguir marcar pontos usando a aparência da Figura 50.

Figura 50 – Jogo do Pong no Scratch



Fonte: SCRATCH, 2025.

Esse nosso jogo é composto por três personagens: a Ball (bolinha), a Paddle (raquete) e a Line (linha vermelha).

Para iniciarmos a elaboração do jogo, vamos, inicialmente, fazer o cenário do jogo, conforme a Figura 50.

Construção de cenário:

1º PASSO

Adicione os nossos três personagens Ball (bolinha), Paddle (raquete) e Line (linha vermelha), clicando no ícone do gatinho no canto direito da área de palco do Scratch, conforme a Figura 51.

Figura 51 – Acréscimo de personagens ao Jogo de Pong



Fonte: SCRATCH, 2025.

Posicione de forma adequada os personagens:

2º PASSO

- Ball (bolinha): X: 70 | Y: 141
- Paddle (raquete): X: 0 | Y: -166
- Line (linha vermelha): X: 0 | Y: -179

Programação da Raquete:

3º PASSO

Faça com que a programação se inicie ao clicar na bandeirinha verde, adicionando o Bloco de Eventos



4º PASSO

Junte, ao bloco anterior, o Bloco de Movimento e altere o y:0 para y:-166. Isso indicará a posição em que o personagem estará quando o jogo for iniciado.



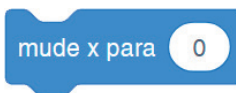
5º PASSO

Adicione o Bloco de Controle para que os blocos dentro dele se repitam.



6º PASSO

Dentro do bloco anterior, adicione o Bloco de Movimento e substitua o zero pelo Bloco de Sensores. Assim a posição da nossa raquete se movimenta, de acordo com o movimento do nosso mouse, apenas para os lados.



posição x do mouse

Programação da Bolinha:

7º PASSO

No Bloco de Variáveis, crie a variável PONTOS.

8º PASSO

Faça com que a programação se inicie ao clicar na bandeirinha verde, adicionando o Bloco de Eventos.



9º PASSO

Junte, ao bloco anterior, o Bloco de Variáveis.



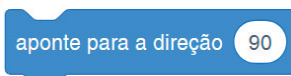
10º PASSO

Programe a posição em que a bolinha se iniciará, adicionando o Bloco de Movimento ao bloco anterior e alterando o y:0 para y: -136.



11º PASSO

Adicione o Bloco de Movimento e altere o 90 para 45. Isso indicará uma direção na qual ela irá se movimentar.



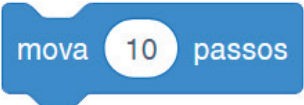
12º PASSO

Junte ao bloco anterior o Bloco de Controle para que os blocos dentro dele se repitam.

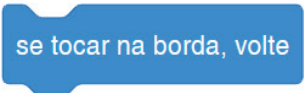


13º PASSO

Dentro do bloco anterior, adicione o Bloco de Movimento e altere o valor de 10 por 8. Este bloco controla a velocidade com que a bolinha vai se movimentar.


 move 10 passos
14º PASSO

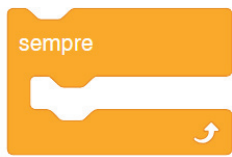
Adicione, abaixo do bloco anterior, o Bloco de Movimento para que, ao tocar nas bordas, a bolinha volte em direção à raquete.


 se tocar na borda, volte
15º PASSO

Adicione o Bloco de Eventos


 quando for clicado
16º PASSO

Abaixo do bloco anterior, adicione o Bloco de Controle para que os blocos dentro dele se repitam.


 sempre
17º PASSO

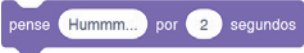
Dentro do bloco anterior, adicione o Bloco de Controle e, no espaço em branco deste bloco, adicione o bloco de condição


 se então
18º PASSO

Adicione o Bloco de Sensores. Esse comando fará com que o fim do jogo seja acionado.


 tocando em LINHA ?
19º PASSO

Dentro do bloco anterior, adicione o Bloco de Aparência e altere o Hummm... para FIM DE JOGO, para que uma mensagem apareça para o jogador.


 pense Hummm... por 2 segundos
20º PASSO

Junte, ao bloco anterior, o Bloco de Controle e todos os blocos irão parar de rodar, finalizando o jogo.


 pare todos

21º PASSO

Adicione o Bloco de Eventos.



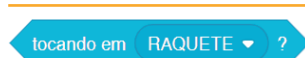
22º PASSO

Abaixo do bloco anterior, adicione o Bloco de Controle para que os blocos dentro dele se repitam.



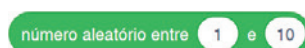
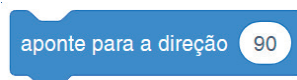
23º PASSO

Dentro do bloco anterior, adicione o Bloco de Controle e, no espaço em branco deste bloco, adicione o Bloco de Sensores. Esse comando fará com que a bolinha mude de direção.



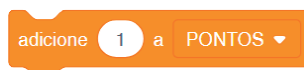
24º PASSO

Dentro do bloco anterior, adicione o Bloco de Movimento e altere o valor de 90 pelo Bloco de Operadores. Depois altere o 1 por -45 e o 10 por 45. Esses blocos combinados fazem com que a bolinha possa se mover em qualquer direção que a leve para a parte de cima do cenário.



25º PASSO

Adicione o bloco de pontuação ao Bloco de Variáveis para que, ao tocar na raquete, a pontuação aumente.



Desafio 5

Transforme esse jogo para uma versão a ser jogada por dois jogadores.

Jogo da cobrinha

No jogo da cobrinha, você controla o personagem da cobrinha para se mover pelo cenário e comer as frutinhas. A cada frutinha que ela come, ela cresce um pouquinho e ganha mais pontos. A primeira providência para o jogo é você dar asas à sua imaginação e desenhar os principais personagens da história, sendo uma cobrinha e uma frutinha.

Animação:

1º PASSO Desenhar o personagem SNAKE que será a cobrinha.

2º PASSO Desenhar o personagem da FRUTINHA.

Programação da Cobrinha:

3º PASSO Selecione o personagem SNAKE e programe a posição inicial.

4º PASSO Programe a movimentação do personagem, e lembre-se de relacioná-lo às setas do teclado.

5º PASSO Crie clones para possibilitar a função de que a cobrinha cresça ao comer a frutinha.

6º PASSO Crie uma automação dos passos 5 e 6 para que ele se repita de forma automática.

7º PASSO Crie a variável de pontuação do seu jogo.

8º PASSO Crie uma pequena sequência de blocos com a função de apagar os clones com o intervalo de segundos igual aos pontos.

9º PASSO Programe o personagem para perder quando tocar nas bordas da tela.

Programação da Frutinha:

10º PASSO Programe o algoritmo para iniciar quando a bandeirinha verde for clicada.

11º PASSO Inicie a pontuação em zero sempre que o jogo for reiniciado.

12º PASSO	Programe a posição da frutinha para aparecer no cenário de forma aleatória.
13º PASSO	Programe para que quando a frutinha tocar no personagem SNAKE (cobrinha), a pontuação aumente e a posição da frutinha mude para outra posição.
14º PASSO	Crie uma sequência de blocos para evitar que a frutinha vá para uma posição na borda da tela.

Jogo da cesta de maçãs

Para treinarmos os conhecimentos que aprendemos até agora, criaremos um pequeno jogo com o objetivo de coletar maçãs caindo, utilizando uma tigela, e toda vez que a maçã tocar a tigela, teremos uma variável chamada MAÇÃS para contar quantas maçãs foram coletadas.

Veja quais são as instruções que devemos executar para termos a programação do jogo.

Programação do jogo de cesta de maçãs

1º PASSO	Adicione os personagens Apple (maçã) e Bowl (tigela).
2º PASSO	Crie uma variável MAÇÃS.
3º PASSO	Programe a nossa Apple (maçã) para começar em uma posição e ficar caindo para ser pega com a tigela.
4º PASSO	Programe para que a maçã caia repetidas vezes.
5º PASSO	Programe uma condição para verificar se a maçã caiu dentro da cesta ou não.
6º PASSO	Selecione o personagem Bowl (tigela) e defina a sua posição inicial para quando o jogo iniciar.
7º PASSO	Iremos controlar a Bowl (tigela) do nosso jogo utilizando as setinhas do teclado. Para isso, programe para que cada bloco SE identifique se uma das teclas está sendo pressionada e, dependendo de qual esteja pressionada (seta esquerda ou direita), a Bowl (tigela) se moverá nessa mesma direção.

Jogo Criativo no Scratch: Aventuras na Floresta Mágica

Neste jogo super divertido chamado “Aventuras na Floresta Mágica”, o jogador controla um personagem que precisa coletar frutas mágicas e escapar de obstáculos enquanto explora uma floresta cheia de surpresas. Vamos usar blocos de controle, movimento, sensores, repetição, eventos e até criar blocos personalizados em Meus Blocos para tornar o jogo ainda mais legal. Preparado? Vamos lá!

Como funciona o jogo?

O objetivo do jogo é coletar 10 frutas mágicas e evitar obstáculos (como pedras e poças de lama) que aparecem no caminho. O jogador precisa mover o personagem pela floresta e pular para evitar os perigos.

Blocos Usados no Jogo

1. Blocos de Eventos

- Quando a bandeira verde for clicada, inicia-se o jogo e o personagem começa a se mover pela floresta.
- Quando o personagem tocar em uma fruta a pontuação aumenta.

2. Blocos de Movimento

- Mover 10 passos: o personagem se move para a frente enquanto explora a floresta.
- Girar 180 graus: quando o personagem bate em uma pedra, ele muda de direção.
- Mudar Y por 20: O personagem pula para evitar obstáculos.

3. Blocos de Controle

- Repetir para sempre: faz com que o jogo continue até que o jogador tenha coletado todas as frutas ou perdido todas as vidas.
- Esperar 1 segundo: controla o tempo entre os obstáculos que aparecem na tela.
- Se, então, senão: o jogo verifica se o personagem encostou em uma fruta ou em um obstáculo, decidindo o que acontece.

4. Blocos de Sensores

- Se tocar na fruta: detecta quando o personagem toca em uma fruta mágica e adiciona um ponto à pontuação.
- Se tocar em obstáculos: detecta quando o personagem bate em uma pedra ou cai na lama e, quando isso acontece, o jogador perde uma vida.

5. Blocos de Repetição

- Repita 10 vezes: o personagem precisa coletar 10 frutas para vencer o jogo.
- Sempre: fazem os obstáculos aparecerem de forma contínua enquanto o jogo está acontecendo.

6. Meus Blocos: Criando o Super Pulo

- Crie um bloco personalizado chamado "Super Pulo":
- Dentro desse bloco, adicione o comando "Mudar Y por 40", para que o personagem dê um pulo ainda mais alto.
- Use o bloco de repetição para fazer o personagem descer suavemente após o pulo.

Como Programar o Jogo?

1. Movimento do Personagem:

- No início do jogo (quando a bandeira verde for clicada), o personagem começa a se mover pela floresta usando o bloco “Mover 10 passos”.
- Se o jogador apertar a seta para cima, o personagem pula com o bloco “Mudar Y por 20”.

2. Coletando Frutas:

- Usando o bloco de sensores, o jogo verifica se o personagem tocou em uma fruta. Se sim, a pontuação aumenta com o bloco de variáveis.
- O jogador precisa coletar 10 frutas para vencer o jogo.

3. Evitando Obstáculos:

- Obstáculos, como pedras e lama, aparecem continuamente usando o bloco de repetição.
- Se o personagem tocar em um obstáculo, o bloco de controle “Se, então” é ativado, e o jogador perde uma vida.

4. Pontuação e Vidas:

- Crie uma variável chamada “Pontos” para contar quantas frutas o jogador coletou.
- Crie outra variável chamada “Vidas”. O jogador começa com 3 vidas e cada vez que o personagem bate em um obstáculo, perde uma vida.

Objetivo do Jogo

- Vencer o jogo: colete 10 frutas mágicas antes de perder todas as vidas.
- Desafio: evitar obstáculos que apareçam no caminho, pulando sobre eles com o Super Pulo que você criou em Meus Blocos!

Personalize o Jogo!

- Você pode criar personagens e cenários personalizados para deixar o jogo do seu jeito.
- Que tal trocar as frutas por moedas ou os obstáculos por monstros divertidos?
- Adicione sons de vitória ou derrota para deixar o jogo ainda mais interativo.

Agora é sua vez! Use tudo o que aprendeu sobre blocos de controle, movimento, sensores e repetição para criar Aventuras na Floresta Mágica ou inventar seu próprio jogo incrível. Solte a imaginação e divirta-se programando no Scratch!

REVISÃO DO MÓDULO 5

Parabéns! Você concluiu o módulo de Criação de Jogos com Scratch!

Agora que você explorou e aprendeu a criar vários tipos de jogos no Scratch, usando diferentes blocos e muitas ideias criativas, está pronto para se tornar um verdadeiro programador! Ao longo desta unidade, você experimentou como cada bloco pode dar vida às suas ideias e transformar comandos simples em jogos divertidos e interativos.

Continue praticando e criando jogos!

Cada exemplo e atividade foram pensados para ensinar-lhe como programar de forma divertida e fácil, mas a prática é fundamental! Quanto mais você brincar com os blocos e criar novos projetos, mais vai aprender e aperfeiçoar suas habilidades. Não tenha medo de errar — cada tentativa é uma chance de melhorar e descobrir novas soluções.

Agora que você já sabe programar seus próprios jogos, que tal criar algo totalmente novo? Pode ser uma corrida com personagens malucos, um jogo de aventura em um castelo ou até um labirinto cheio de desafios. O mais importante é soltar a imaginação e se divertir enquanto aprende.

Prepare-se para o próximo desafio! O Scratch oferece infinitas possibilidades para inventar e programar. Continue criando, praticando e se divertindo — o próximo grande jogo pode ser o seu!

Até o próximo módulo!

Anotações

Handwriting practice lines consisting of 20 horizontal dashed lines.

MÓDULO 6 – CRIANDO HISTÓRIAS INTERATIVAS COM SCRATCH

Agora que já conhecemos os conceitos básicos de lógica de programação, sabemos como fazer animações e criar jogos, estamos prontos para liberar nossa criatividade e criar nossas próprias histórias interativas.

Uma história interativa é como um jogo em que você decide o que vai acontecer. Enquanto você lê ou joga, pode fazer escolhas que mudam o rumo da história, como decidir o que o personagem principal fará em cada momento. Dependendo da sua escolha, a história pode ter um final diferente ou seguir por caminhos inesperados. É como se você fosse o autor e o jogador ao mesmo tempo, ajudando a criar a aventura e deixando tudo mais divertido e emocionante!

Para entender melhor, vamos seguir os próximos passos e criar uma história interativa, na qual nosso personagem vai escolher que tipo de passeio ele fará hoje. Para animar, espelhe-se no personagem que desenvolvemos na Figura 52. Vamos lá!

Figura 52 – Proposta de personagem para a história interativa



Fonte: SCRATCH, 2025.

Organizando a história:

Observe os cenários elaborados para a história interativa:

CENÁRIO 1	
PERSONAGENS:	Nano
DESCRIÇÃO:	Quarto: Nano acorda bem animado no seu quarto e decide que vai passear em algum lugar. É preciso decidir para onde ele irá: floresta ou praia?
FALAS:	Nano: Que belo dia! Acordei tão animado hoje! Nano: Eu poderia aproveitar este belo dia para passear Nano: Para onde eu devo ir passear? Praia ou Floresta?

CENÁRIO 2	
PERSONAGENS:	Nano, Giga, Borboletas
DESCRIÇÃO:	Floresta: Nano vai à floresta passear e acaba se encontrando com sua amiga Giga. Ela está brincando na floresta, contando o número de borboletas que ela acha. Ela pergunta ao Nano quantas borboletas tem no cenário e ele deve responder a quantidade correta. Após contar a quantidade correta de borboletas, Nano percebe que está ficando tarde e deve decidir se vai ficar mais tempo na floresta ou se voltará para casa.

FALAS:	Nano: Essa floresta é muito bonita! Espere, ali na frente não é a Giga? Nano pensando: Eu deveria dar um oi para ela. Nano: Olá, Giga! Você veio brincar na floresta? Giga: Olá, Nano! Hoje eu brinquei bastante na floresta. Giga: Estou tentando contar o número de borboletas aqui. Giga: Quantas borboletas tem aqui? Se responder certo, Giga fala: VERDADE! Tem 3 borboletas nesse lugar. Se responder errado, Giga fala: Tem certeza, amigo? Vamos contar novamente. Nano: Foi muito divertido contar borboletas com você, Giga. Nano: Vamos fazer isso mais vezes depois. Nano pensa: Devo decidir agora o que fazer. Nano: Para onde devo ir? Floresta ou Casa? Se Nano responder Floresta, ele fala: Vou continuar na floresta, talvez eu ache ainda mais borboletas. Se Nano responder Casa, ele fala: Vai escurecer logo, logo. Melhor eu voltar para casa.
--------	---

CENÁRIO 3	
PERSONAGENS:	Nano, Roberto
DESCRIÇÃO:	Praia: Nano chega à praia depois de uma longa caminhada. Ele logo percebe que seu amigo Roberto está passeando na praia também e o cumprimenta. Roberto cumprimenta Nano de volta, e eles colocam a conversa em dia. Depois de conversar, Roberto pergunta para onde o Nano vai depois da praia.

FALAS:	Nano: Finalmente cheguei na praia, tem muitas crianças brincando aqui. Nano pensa: Aquele não é o Roberto? Nano: Olá, Roberto! Quanto tempo não te vejo. Roberto: NANO! Você ficou muito tempo sem passear na praia. Roberto: Para onde você vai depois daqui? Mar ou Casa? Se Nano responder Mar, Roberto fala: Que legal! Então vou com você. Roberto: Eu adoro o mar. Se Nano responder Casa, Roberto fala: Tudo bem, foi bom te ver, amigo!
--------	---

CENÁRIO 4	
PERSONAGENS:	Nano, Roberto
DESCRIÇÃO:	Fundo do mar: Nano e Roberto nadaram para um lugar tão fundo do mar, que ele ficou preocupado com os perigos daquele lugar. Nano decide que o melhor a fazer é voltar para casa.
FALAS:	Nano: Acho que fui para fundo demais no mar. Nano: Pode ser perigoso aqui, Roberto. Melhor irmos embora. Nano: Vou procurar o meu caminho para casa.

CENÁRIO 5	
PERSONAGENS:	Nano
DESCRIÇÃO:	Floresta escura: Nano ficou até mais tarde na floresta e agora ela está muito escura. Pode ser muito perigoso andar em uma floresta à noite, então Nano decide que é melhor ele voltar para casa.
FALAS:	Nano: Acho que não foi uma boa ideia ficar até tão tarde. Nano: A floresta está tão escura e assustadora. Nano: Vou procurar o meu caminho para casa.

CENÁRIO 6	
PERSONAGENS:	Nano
DESCRIÇÃO:	Nano voltando para casa: Nano se aventurou muito nesse dia e ficou feliz de finalmente estar voltando para casa e descansar. Mas ele está empolgado para onde a próxima aventura o levará.
FALAS:	Nano: O dia foi bem divertido hoje. Nano: Muito obrigado por ter me acompanhado nessa aventura, amigo. Nano: Nos vemos na próxima!

2º passo - Adicione os personagens e os cenários em ordem de acordo com o roteiro criado no 1º passo.

Os personagens: Nano, Roberto, Giga e Butterfly (borboleta).

Os cenários: Bedroom 3, Forest, Beach Rio, Underwater 1, Woods e Night City With Street.

Estamos empolgados para desenvolver novas histórias. Vamos lá!!!

Nesta história interativa chamada A Grande Aventura no Parque, o jogador controla um personagem que visita um parque de diversões e precisa escolher quais atrações ele vai visitar. Dependendo das escolhas, a aventura pode ter resultados finais diferentes! Ao longo da história, usaremos vários blocos de controle, movimento, sensores, repetição, eventos e até blocos personalizados em Meus Blocos para tornar a experiência divertida e educativa. Vamos começar?

Como funciona a história?

O personagem principal chega ao parque e pode escolher entre três atrações:

- Montanha russa
- Roda-gigante
- Carrossel

Durante a visita, o personagem enfrentou alguns desafios, como escolher o brinquedo certo, comprar ingressos e tomar decisões que afetam o final da aventura.

Blocos Usados na História

1. Blocos de Eventos

- Quando a bandeira verde for clicada: inicia-se a história e o personagem chega ao parque.
- Quando uma tecla específica for pressionada: o jogador escolhe uma atração.

2. Blocos de Movimento

- Mover 10 passos: o personagem anda pelo parque para explorar as atrações.
- Apontar na direção de X: o personagem vira para onde o jogador escolher.

3. Blocos de Controle

- Se, então, senão: o personagem toma decisões com base na escolha do jogador (por exemplo, ir para a montanha-russa ou carrossel).
- Esperar 2 segundos: controle o tempo entre falas e ações do personagem.

4. Blocos de Sensores

- Se tocar no ingresso: o personagem compra ingressos para poder entrar em uma atração.
- Se tocar na borda: o personagem volta para o ponto inicial.

5. Blocos de Repetição

- Repetir 10 vezes: o carrossel gira 10 vezes antes de parar.

- Sempre: a roda-gigante continua girando até o jogador pressionar uma tecla.

6. Meus Blocos: Criando Ações Personalizadas

- Crie um bloco chamado “Super Passeio” que faz o personagem andar, pular e girar em uma combinação de ações.
- Use o bloco “Super Passeio” em várias partes da história, para tornar a experiência mais divertida.

Como Programar a História?

1. Início da História:

- Bloco de evento: “Quando a bandeira verde for clicada”, o personagem aparece na entrada do parque e diz: “Vamos escolher uma atração!”.

2. Escolhendo a Atração:

- O jogador pode iniciar, digitando 1 para ir na montanha-russa, 2 para a roda-gigante ou 3 para o carrossel.
- Bloco de controle: “Se a tecla 1 for pressionada, então vá para a montanha-russa”.

3. Desafio: Comprando Entradas:

- O personagem deve tocar em três entradas espalhadas pelo parque para entrar nas atrações.
- Bloco de sensor: “Se tocar no ingresso, aumente a variável ‘Ingressos’ por 1”.

4. Passeios Personalizados:

- Na montanha-russa, o personagem deve se mover para frente e para trás com blocos de movimento.
- No carrossel, o personagem usa blocos de repetição para girar

várias vezes.

- Na roda-gigante, o jogador pode controlar o ritmo do passeio com blocos de controle.

5. Final da História:

- Dependendo das atrações que o personagem visitou, o final da história muda.
- Bloco de controle: “Se a variável ‘Ingressos’ for igual a 3, então o personagem diz: “Você visitou todas as atrações! Parabéns!”.

Objetivo da História

- O jogador deve comprar 3 ingressos e visitar pelo menos duas atrações para completar a aventura.
- Cada escolha do jogador influencia o rumo da história, tornando-a única e divertida a cada vez que joga.

Personalize a História!

- Adicione novas atrações ou missões especiais para tornar a história ainda mais emocionante.
- Use blocos de som para adicionar músicas e efeitos a cada atração.
- Crie variações de pontuação para deixar o jogo mais desafiador.

Agora é sua vez! Experimente criar A Grande Aventura no Parque e solte a imaginação para inventar histórias únicas no Scratch. Quanto mais você praticar, melhor.

REVISÃO DO MÓDULO 6

Parabéns! Concluímos com sucesso o módulo de Criação de Histórias Interativas com Scratch!

Agora que você explorou vários exemplos de histórias interativas, já sabe como usar o Scratch para criar aventuras incríveis nas quais suas decisões mudam o rumo da história. Cada história que você cria é única, e, com o Scratch, você é ao mesmo tempo o autor e o jogador. Isso significa que sua imaginação não tem limites! A cada nova ideia, você pode inventar mundos, personagens e desafios que deixam suas histórias mais emocionantes e interativas.

Lembre-se: quanto mais você praticar, mais fácil e divertido será programar. Experimente usar todos os recursos que você aprendeu para inventar novas histórias e jogos interativos. Desafie-se a criar aventuras cada vez mais criativas, com múltiplos finais e caminhos inesperados.

Não tenha medo de errar e testar novas ideias. A programação é assim: tente, ajuste e melhore até que sua história fique exatamente como você imaginou. Use tudo o que aprendeu nesta unidade e explore ainda mais o Scratch. Quem sabe você cria a próxima grande história que todos vão adorar?

Agora é com você! Solte a imaginação e continue praticando para se tornar um excelente programador de jogos e histórias interativas.

Nos encontramos no próximo módulo para mais desafios e diversão!

Anotações

This image shows a full page of white paper with horizontal dashed lines, typical of primary school handwriting practice paper. The lines are evenly spaced and run across the entire width of the page. There are no margins, text, or other markings present.

MÓDULO 7 – BLOCOS DE COMANDO NO SCRATCH



Neste módulo, vamos revisar todos os blocos que exploramos ao longo do livro. Este material foi feito para fortalecer tudo o que você aprendeu e servir como apoio para continuar praticando. Lembre-se: a prática é essencial para se tornar um excelente programador e criar projetos cada vez mais incríveis no Scratch!

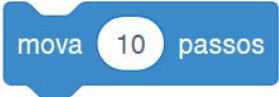
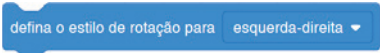
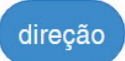
Com esta unidade, você tem tudo o que precisa para relembrar cada bloco e criar projetos incríveis! Use este conteúdo sempre que tiver dúvidas ou precisar de ajuda durante suas criações. Não se esqueça: a prática constante é essencial para se tornar cada vez melhor no Scratch.

Então, que tal começar agora? Continue explorando, testando e programando, pois a prática leva à perfeição! Quanto mais você criar e experimentar, mais habilidoso você se tornará.

Dica: Explore a comunidade do Scratch! Lá você encontrará muitos projetos incríveis e poderá aprender com outros programadores.

Blocos de Movimento no Scratch

Os blocos de eventos de movimento no Scratch permitem que você controle o movimento dos objetos em seu projeto. Eles funcionam como gatilhos que iniciam ou modificam o movimento dos objetos em respostas a diferentes eventos. Esses blocos são essenciais para criar animações, jogos e simulações interativas, dando vida aos objetos e proporcionando uma experiência dinâmica para o usuário. Aqui estão os comandos dos Blocos de Movimento:

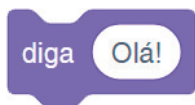
	Mover: move o sprite para frente ou para trás pelo número especificado de passos.
	Girar: gira o sprite no sentido horário ou anti-horário pelo número especificado de graus.
	Vá para: move o sprite para as coordenadas (x, y) especificadas na tela.
	Deslizar: move suavemente o sprite para as coordenadas (x, y) especificadas na tela durante um período de tempo determinado.
	Apontar para a direção: faz com que o sprite aponte na direção especificada em graus.
	Apontar para: faz com que o sprite aponte para onde o ponteiro do mouse está.
	Adicione a x: adiciona uma posição horizontal, que será somada à atual.
	Mude x para: muda a posição horizontal do sprite para o valor especificado.
	Definir estilo de rotação: define o estilo de rotação para o sprite. Você pode escolher entre "em todas as direções", "esquerda-direita" ou "não rotacionar".
☐ 	Posição X: retorna à posição horizontal atual do sprite.
☐ 	Direção: retorna à direção atual na qual o sprite está apontando, medida em graus.

Blocos de Aparência no Scratch

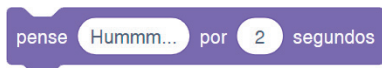
Os blocos de eventos de aparência no Scratch permitem que você controle a aparência visual dos objetos em seu projeto. Eles são como interruptores que ativam mudanças na aparência em resposta a diferentes eventos ou interações do usuário. Esses blocos são cruciais para criar animações, transições e feedback visual dinâmico em seus projetos. Vamos ver cada um dos comandos existentes nos seguintes Blocos de Aparência:



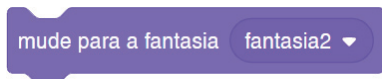
Diga por segundos: envia uma mensagem ou texto na tela por um determinado período de tempo especificado em segundos.



Diga: usado para exibir uma mensagem na tela ou reproduzir um som.



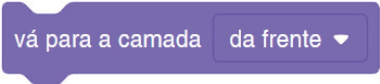
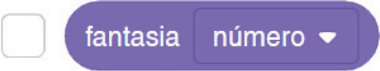
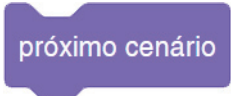
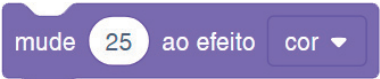
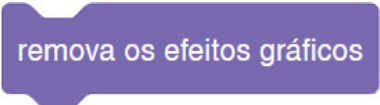
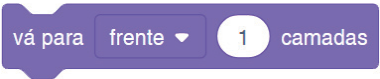
Pense por segundos: introduz um atraso ou pausa no programa por um número específico de segundos antes de executar a próxima ação.



Mude para a fantasia, próxima fantasia: pode estar relacionado à troca de sprites (personagens ou objetos) no projeto. Ele pode mudar de um sprite para outro ou avançar para o próximo na lista de sprites disponíveis.

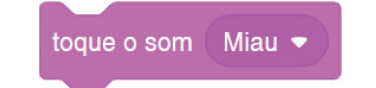
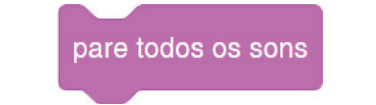
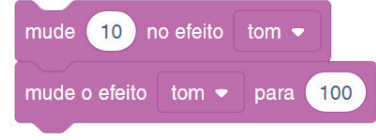
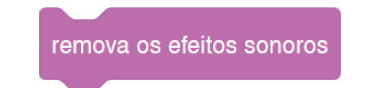
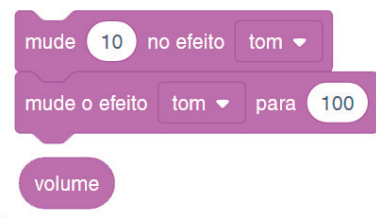


Mude para o cenário, mostre: pode ser usado para alterar o cenário atual do projeto para outro especificado e, em seguida, exibi-lo na tela.

  	Vá para a camada, fantasia, tamanho: controla a camada na qual um sprite está localizado (por exemplo, trazendo-o para frente ou enviando-o para trás) e pode também alterar o tamanho do sprite.
	Próximo cenário, mude: pode estar relacionado à navegação entre diferentes cenários ou fundos no projeto.
	Defina o tamanho como: é usado para definir o tamanho de um sprite, especificado em porcentagem ou pixels.
	Mude ao efeito, defina o efeito: controla os efeitos visuais aplicados a um sprite, como mudanças de cor, distorções ou animações.
	Remova os efeitos gráficos, esconda: pode ser usado para limpar todos os efeitos gráficos aplicados a um sprite e também para escondê-lo da tela.
	Vá para camadas, cenários: pode estar relacionado ao controle de visibilidade ou posicionamento de sprites em diferentes camadas ou cenários no projeto.
	Esconda: usado para tornar invisível um sprite na tela, fazendo com que ele não seja exibido, embora ainda esteja presente no projeto. Ele é frequentemente utilizado quando se deseja que um sprite permaneça oculto até que seja necessário exibi-lo novamente.

Blocos de Som no Scratch

Os blocos de eventos de som no Scratch permitem que você controle o áudio em seu projeto. Eles são como botões que iniciam e param o som em resposta a diferentes eventos. É oportuno conhecer cada um dos comandos existentes nos blocos a seguir:

	Toque o som até o fim: reproduz um som no projeto até que ele chegue ao final. Ele garante que o som seja reproduzido completamente antes de passar para a próxima ação.
	Toque o som: inicia a reprodução de um som no projeto.
	Pare todos os sons: interrompe a reprodução de todos os sons que estão atualmente em execução no projeto.
	Mude no efeito, muda o efeito para: pode controlar os efeitos sonoros aplicados a um som, como mudanças de tom, eco ou distorção.
	Remova os efeitos sonoros: usado para limpar todos os efeitos sonoros aplicados a um som, restaurando-o ao estado original.
	Mude o volume em, muda o volume para, volume: usados para ajustar o volume de um som. "Mude o volume em" pode aumentar ou diminuir o volume em uma certa quantidade, enquanto "mude o volume para" define o volume para um valor específico. O bloco "volume" pode ser usado para obter o valor atual do volume de um som.

Blocos de Eventos no Scratch

Os blocos de eventos desempenham um papel crucial no controle do início e das ações iniciais de um programa no Scratch. Eles são como gatilhos que ativam o funcionamento do programa em resposta a diferentes estímulos. São como interruptores que iniciam o fluxo de execução do programa, conforme o funcionamento dos blocos.



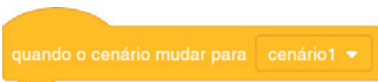
Acionado quando você clica na bandeira verde no topo da área de palco. É comumente usado para iniciar o projeto, redefinir variáveis ou configurar o estado inicial do programa.



Permite que você defina ações específicas para ocorrerem quando uma determinada tecla for pressionada.



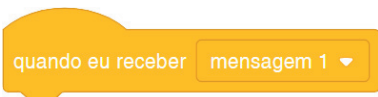
Possibilita que você defina ações específicas para ocorrerem quando o ator for clicado.



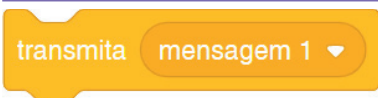
Proporciona que você defina ações específicas para ocorrerem quando o cenário for mudado.



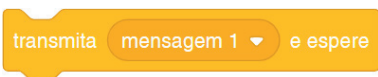
Acionado quando uma variável é maior que um valor especificado.



Permite que você defina ações para ocorrerem quando um sprite recebe uma mensagem específica de outro sprite. Ele é útil para criar interações entre diferentes partes do seu projeto.



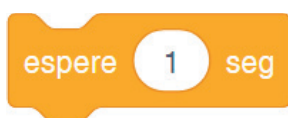
Estabelece a transmissão de uma determinada mensagem.



Define a transmissão de uma determinada mensagem e espera uma resposta do usuário.

Blocos de Controle no Scratch

Os blocos de controle são usados para controlar o fluxo de execução do programa. Eles permitem que você especifique como e quando os comandos em seu programa serão executados. Ou seja, blocos de controle são como “comandos mágicos” que dizem ao nosso programa o que fazer e quando fazer. Eles são como as rédeas de um cavalo, guiando o caminho do nosso programa. A seguir, apresentamos alguns deles.



Repita x vezes: é utilizado para repetir um conjunto de comandos várias vezes. Você pode especificar o número de repetições ou usar uma condição para determinar quando parar de repetir.



Sempre: é útil para manter comportamentos contínuos, como atualizações de tela em tempo real ou verificações de condições persistentes. É como dizer ao nosso programa para fazer algo o tempo todo, sem parar.



Se ... então: é usado para criar condições em um programa. Se uma determinada condição for verdadeira, então um conjunto específico de comandos será executado. É como se nosso programa pudesse pensar: “Se isso acontecer, então faça aquilo!”



Se... então, senão: é um bloco condicional que permite comportamentos alternativos com base em condições específicas. Ao usar esse bloco, os alunos aprendem a criar bifurcações no fluxo do programa, permitindo que ele tome decisões com base em diferentes cenários.

espere até que

Esperar até que ... : é utilizado para pausar a execução do programa até que uma determinada condição seja atendida. Pode ser útil para sincronizar eventos ou esperar pela entrada do usuário. Às vezes, nosso programa precisa fazer uma pausa e esperar por algo. Este bloco nos ajuda a fazer isso.

repita até que

Repita até que ... : é uma estrutura de controle de fluxo que permite a criação de loops baseados em condições de parada. Com o bloco, é possível instruir o programa a repetir um conjunto de comandos até que uma condição específica seja avaliada como verdadeira.

pare todos

Pare: é uma ferramenta de controle de fluxo que permite interromper a execução de um script ou programa em um ponto específico.

quando eu começar como um clone

Quando eu começar como um clone: é como dar instruções especiais a um personagem/objeto quando ele é clonado, ou seja, quando surgem cópias dele. É como se cada nova cópia recebesse suas próprias regras de como agir. Por exemplo, se temos um boneco e fazemos várias cópias dele, podemos dizer para cada cópia: "Quando você aparecer, comece a pular para a direita!".

crie clone de este ator

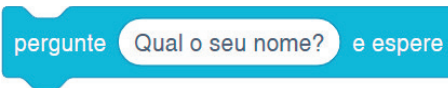
Crie clone de: é como fazer uma cópia de um personagem ou objeto no seu programa. Quando você usa o bloco, o personagem original continua fazendo o que está fazendo, mas também é criada uma cópia dele que pode ter suas próprias aventuras!

apague este clone

Apague esse clone: é como dizer adeus a uma cópia de um personagem que já fez o seu trabalho. Quando você usa esse bloco, você está dizendo ao programa para remover uma cópia do personagem que não é mais necessária.

Blocos de Sensores no Scratch

Os blocos de sensores são usados para interagir com o ambiente do projeto Scratch. Eles permitem que você detecte eventos, como cliques de mouse, pressionamentos de tecla e colisões entre personagens e/ou objetos. São eles:

	Tocando em: verifica se o personagem/objeto atual está colidindo com outro personagem/objeto na tela.
	Tocando na cor: observa se o personagem/objeto atual está colidindo com uma área de uma cor específica na tela.
	A cor ... está tocando na cor...?: identifica se uma cor específica está colidindo com outra cor na tela. Ele permite que os alunos programem comportamentos condicionais com base em interações de cores, o que pode ser útil para criar lógica complexa em projetos.
	Distância até: sensor que mede a distância entre o personagem/objeto atual e outro personagem/objeto (também podendo ser o ponteiro do mouse) na tela. Ele permite a programação de comportamentos com base na proximidade entre personagem/objeto, o que pode ser útil para criar interações dinâmicas em projetos.
	Pergunte e espere: possibilita que o programa solicite uma entrada do usuário e aguarde até que uma resposta seja fornecida antes de continuar.



resposta

Resposta: captura e armazena a resposta do usuário após uma pergunta feita pelo programa, permitindo que essa resposta seja utilizada em outras partes do código.

tecla

espaço ▼

pressionada?

Tecla pressionada: é usado para verificar se uma tecla específica do teclado foi pressionada pelo usuário. Ele permite que o programa responda a eventos de teclado, como movimentar um personagem ou acionar ações específicas em um jogo.

mouse pressionado?

Mouse pressionado: verifica se um dos botões do mouse foi pressionado pelo usuário, permitindo que o programa responda a eventos de mouse, como clicar em objetos ou interagir em um jogo.

posição x do mouse

Posição x do mouse: obtém a coordenada horizontal do cursor do mouse na tela, permitindo que o programa responda ao movimento do mouse, como mover objetos ou ativar eventos com base na posição do cursor.

posição y do mouse

Posição y do mouse: obtém a coordenada vertical do cursor do mouse na tela, permitindo que o programa responda ao movimento do mouse, como mover objetos ou ativar eventos com base na posição do cursor.

defina modo de arrasto para

arrastável ▼

Defina modo de arrasto para: determina se um objeto pode ser arrastado pelo usuário com o mouse, promovendo interações como mover objetos na tela.

☐	ruído	Ruído: gera um valor aleatório entre 0 e 100, útil para criar comportamentos imprevisíveis em jogos ou simulações.
☐	cronômetro	Cronômetro: mede o tempo decorrido desde o início do programa, sendo útil para controlar eventos com base no tempo, como cronometrar atividades ou criar temporizadores em jogos.
	zere o cronômetro	Zere o cronômetro: reinicia o tempo medido pelo cronômetro, definindo-o de volta para zero; útil para reiniciar a contagem do tempo desde o início do programa ou após um evento específico.
	n° do cenário ▼ de Palco ▼	N° do cenário de palco: é usado para obter o número do cenário atual do palco em um projeto. Isso pode ser útil para controlar a lógica do seu jogo ou animação, dependendo do cenário em que o personagem está.
☐	ano ▼ atual	Atual: detecta ano, mês, dia, hora e segundos.
	número de dias desde 2000	Número de dias desde 2000: calcula a quantidade de dias desde o ano 2000 até a data atual, útil para cálculos relacionados ao tempo ou sistema de contagem de dias.
☐	nome de usuário	Nome de usuário: é usado para obter o nome de usuário do jogador interagindo com o projeto, permitindo personalizar a experiência do usuário ou salvar seu progresso com seu usuário.

Blocos de Operadores no Scratch

Os blocos de operadores são usados para realizar operações matemáticas, lógicas e de texto dentro do programa. Eles permitem que você manipule valores numéricos e de texto para realizar cálculos e tomar decisões. Os blocos nesta categoria incluem os ícones apresentados do lado esquerdo da tabela a seguir:



É como uma calculadora que adiciona números. Por exemplo, se você tem o número 5 e o número 3, ao usar o operador de soma entre eles ($5 + 3$), você obtém o resultado 8.



Subtrai os números escolhidos. Por exemplo, se você tem o número 5 e o número 3, ao usar o operador de subtração entre eles ($5 - 3$), você obtém o resultado 2.



Multiplica os números que você escolher. Por exemplo, se você tem o número 5 e o número 3, ao usar o operador de multiplicação entre eles (5×3), você obtém o resultado 15.



Divide os números da sua escolha. Por exemplo, se você tem o número 15 e o número 3, ao usar o operador de divisão entre eles ($15 / 3$), você obtém o resultado 5.

número aleatório entre 1 e 10

É como jogar um dado de 10 lados, onde qualquer número de 1 a 10 pode aparecer aleatoriamente. Mas se você quiser escolher um número específico, basta digitar o número desejado.



Verifica se o valor de "x" é maior que 50. Se você quiser usar qualquer outro número no lugar de 50, basta editar o valor dentro do bloco para comparar "x" com qualquer número desejado.



Verifica se o valor de "x" é menor que 50. Se você quiser usar qualquer outro número no lugar de 50, basta editar o valor dentro do bloco para comparar "x" com qualquer número desejado.



Verifica se o valor de "x" é igual a 50. É como comparar se duas peças de quebra-cabeça se encaixam perfeitamente. Se você quiser usar qualquer outro número no lugar de 50, basta editar o valor dentro do bloco para comparar "x" com o número desejado.



Verifica se as duas condições dadas estão satisfeitas. Para a condição ser validada, ambas têm que estar satisfeitas.



Verifica se pelo menos uma das condições está satisfeita, não necessariamente as duas, mas apenas uma delas.



Essa condição só será validada quando o que pedir dentro dela não estiver acontecendo. Por exemplo, se a tecla "espaço" não estiver pressionada, o personagem vai andar, mas a partir do momento em que a tecla "espaço" estiver pressionada, o personagem para.



Junta dois valores... Como no exemplo, maçã com banana ficaria "maçã banana".



Encontra e mostra a letra em certa posição de uma palavra, por exemplo, letra 1 de maçã é "m", e a letra 4 de maçã é "ã".

	Mostra quantas letras tem a palavra digitada. Por exemplo, a palavra maçã tem 4 letras.
	Detecta se uma palavra tem uma letra específica. Nesse caso, irá procurar a letra “m” na palavra maçã.
	Devolve o resto da divisão do 1º número pelo 2º número. Por exemplo, na divisão 5: 2, temos o resultado 2 e o resto 1. Ele irá dividir o número 1 pelo número que você selecionar.
	Arredonda o número dado para o inteiro mais próximo. Por exemplo: 4,6 arredonda para 5.
	Existem várias opções de cálculo nesse bloco. Você pode escolher um número, um cálculo e ele faz pra você.
	Aqui estão algumas opções de cálculo que você pode fazer, basta clicar em “módulo” e escolher uma opção.

Bloco de Variáveis no Scratch

Os blocos de variáveis são blocos que vão permitir a criação e controle de variáveis. Uma variável é um elemento no qual é possível alterar o seu valor.

Por exemplo, em relação ao uso de uma variável chamada PONTOS, sempre que a pessoa pontuar, somará mais 1 ponto para essa variável e nela armazenaremos o novo número ou a nova pontuação.

Criar uma Variável

Cria uma variável com a qual é possível atribuir-lhe um nome. Evite colocar acentos ou espaço quando nomear uma variável.



minha variável

Utilizado para quando for necessário usar ou mostrar o valor da variável criada.

mude minha variável para 0

Utiliza-se para definir um valor para a variável. Neste exemplo, a “minha variável” terá o valor 0.

adicione 1 a minha variável

Incrementa o valor de “minha variável” de acordo com o valor digitado; neste caso, aumentará o valor em 1.

mostre a variável minha variável

Mostra o valor que está armazenado em “minha variável” em uma tela no palco do Scratch.

esconda a variável minha variável

Esconde o valor da variável que aparece no palco do Scratch.

Criar uma Lista

Cria uma lista de itens e/ou variáveis que serão mostrados no palco do Scratch. Deve ser inserido um nome à lista, assim como nas variáveis. É recomendado não utilizar acentos ou espaços.

adicione coisa a teste

Adiciona um item ou valor à lista. Neste caso, a palavra coisa está sendo adicionada na lista “teste”.

apague 1 de teste

Elimina o item que está na posição 1 da lista “teste”.

apague todos os itens de teste

Apaga todos os itens da lista “teste”.



Insere um item em uma certa posição.

Diferente do bloco de adicionar, neste bloco é possível digitar o item e em qual posição ele será inserido.



Troca um item que está na posição definida por outro item na lista.



Usado para mostrar ou utilizar o valor que está na posição definida da lista.



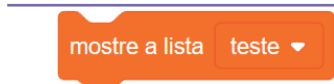
Utilizado para mostrar ou utilizar a posição que se encontra o valor na lista. Por exemplo, se "coisa" estiver na posição 3 da lista "teste", então o valor que será mostrado será 3.



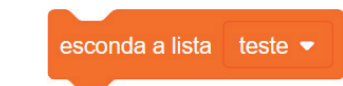
Destinado a mostrar ou utilizar o número de posições que existem na lista.



Verificar se existe um valor em uma lista (bloco condicional). Neste exemplo, o código irá procurar pela palavra "coisa" dentro da lista "teste".



Mostra a lista no palco do Scratch.



Esconde a lista do palco do Scratch.

Meus Blocos no Scratch

Em "Meus Blocos", é possível criar e customizar seus próprios blocos, como se fossem funções para serem utilizados de uma forma mais eficiente no Scratch. Com essa ferramenta, é possível diminuir a quantidade de blocos utilizados na hora de desenvolver.

Criar um bloco

Cria um bloco customizado, adicionando caixas de texto, caixa de valores ou caixa de operações.

Parabéns! Você concluiu o livro de Lógica de Programação: uma proposta para crianças com o uso do Scratch!



Chegamos ao fim da nossa jornada de aprendizado com o Scratch e agora você tem tudo o que precisa para se tornar um verdadeiro programador e criador de projetos incríveis! Ao longo desta apostila, você explorou um mundo de possibilidades e aprendeu como programar de forma divertida e fácil, utilizando cada recurso oferecido pela plataforma Scratch.

O que você aprendeu?

- Movimentos: como fazer os personagens se mexerem e explorarem o palco.
- Aparência: como mudar fantasias e criar efeitos visuais.
- Sons: como adicionar músicas e efeitos sonoros para animar seus projetos.
- Eventos: como dar início a ações e controlar interações.
- Controle: como organizar o fluxo do programa com repetições e condições.
- Sensores: como detectar interações e respostas do ambiente.
- Operadores: como realizar cálculos e comparações dentro do jogo.
- Variáveis: como guardar informações importantes, como pontos e tempo.
- Meus Blocos: como criar seus próprios blocos para deixar o projeto mais organizado.

ATIVIDADES E DESAFIOS

Além de aprender todos esses conteúdos, você participou de atividades práticas e desafios divertidos, como criar:

- Histórias interativas, nas quais você decidiu o rumo da aventura.
- Animações incríveis, dando vida aos personagens.
- Jogos desafiantes, com pontuação e múltiplas fases.

Cada prática foi pensada para ajudá-lo a entender a lógica da programação e mostrar que você é capaz de criar qualquer projeto que imaginar. E você sabe o que é mais legal? Com o Scratch, você é o protagonista. Não importa se você está criando uma história, um jogo ou uma animação: as ideias são suas e você pode inventar o que quiser.

Continue Criando e Explorando!

O Scratch é só o começo. A cada novo projeto, você aprende mais e se prepara para desafios maiores. A prática é essencial! Então, não tenha medo de experimentar, errar e tentar de novo. A programação é assim mesmo: quanto mais você pratica, mais fácil e divertido fica!

Agora que você tem tudo o que precisa, o que acha de continuar explorando o Scratch e criar seu próprio jogo ou história interativa? Quem sabe o próximo grande projeto seja o seu!

E agora, o que vem por aí?

O Scratch é apenas o começo de uma jornada incrível no mundo da programação. Compartilhe seus projetos com amigos e explore a comunidade do Scratch para se inspirar e colaborar com outros criadores. Quem sabe no futuro você queira aprender linguagens como Python ou JavaScript para criar jogos ainda mais com-

plexos ou aplicativos para celular? As possibilidades são infinitas! Lembre-se: a programação é uma ferramenta poderosa para resolver problemas e criar soluções inovadoras. Continue explorando, aprendendo e se divertindo!

Você está pronto para ser um programador! Lembre-se: a imaginação é o seu superpoder. Use-a para desenvolver projetos incríveis, compartilhar suas criações e se divertir programando.

ATÉ A PRÓXIMA AVENTURA!



REFERÊNCIAS

Gemini. *Imagem de mapa do tesouro para crianças*. Montes Claros, 2025. Disponível em: <https://gemini.google.com/>. Acesso em: 24 ago. 2025.

Scratch. *Scratch.Mit*. Disponível em: <https://scratch.mit.edu/>. Acesso em: 24 ago. 2025.

AUTORES



Christine Martins de Matos

<http://lattes.cnpq.br/3097685204253789>

Mestre em Administração pela Faculdade de Estudos Administrativos de Minas Gerais (FEAD/MG). Professora do Departamento de Ciências da Computação da Unimontes, atuando em cursos de graduação e pós-graduação, focando em disciplinas relacionadas a sistemas de informação e tecnologias educacionais, bem como participação em projetos de pesquisa financiados pela FAPEMIG, contribuindo para o avanço do conhecimento nas áreas de TI e educação. Atua como Diretora do CEAD-Unimontes desde 2023, liderando iniciativas inovadoras na área de educação a distância.



Ellen Felícia Ferreira Miranda

<http://lattes.cnpq.br/0905267831444298>

É graduanda em Engenharia de Sistemas pela Universidade Estadual de Montes Claros. Tem experiência na área de Ciência da Computação. Atualmente é bolsista FAPEMIG do Laboratório Multiusuário de Tecnologias Digitais na Educação (Educar) da Unimontes.



Lérica Maria Mendes Veloso

<http://lattes.cnpq.br/5623317033979181>

Mestre em Geografia pela Unimontes e Pós-Graduada em Meio Ambiente e Desenvolvimento Regional. Foi Bolsista PIBID/UNIMONTES/CAPES, ICV, UAB/Unimontes e UAB/IFNMG. Atualmente é bolsista FAPEMIG do Laboratório Multiusuário de Tecnologias Digitais na Educação (Educar) e do Laboratório de Educação Digital (LED) da Unimontes.



Mirelle Pereira da Silva

<http://lattes.cnpq.br/0601959249867123>

Estudante de mestrado em Educação no Programa de Pós-Graduação em Educação (PPGE/UNIMONTES). Possui Pós-Graduação Lato Sensu em Inspeção Escolar (Faculdade IMES) e Pós-Graduação Lato Sensu em Educação Matemática (UNIASSELVI). Licenciada em Matemática (IFNMG/Januária) e Licenciada em Pedagogia (UNIMONTES).



Andrey Guilherme Mendes de Souza

<http://lattes.cnpq.br/7265986380492823>

É graduando em Sistemas de Informação pela Universidade Estadual de Montes Claros. Tem experiência na área de Ciência da Computação, com ênfase em Educação Tecnológica. Atualmente é bolsista FAPEMIG do Laboratório Multiusuário de Tecnologias Digitais na Educação (Educar) da Unimontes.



Lucas Guimarães Pereira

<http://lattes.cnpq.br/7244131633932916>

Possui graduação em Sistemas de Informação pela Universidade Estadual de Montes Claros(2023). Tem experiência na área de Ciência da Computação. Foi bolsista FAPEMIG do Laboratório Multiusuário de Tecnologias Digitais na Educação (Educar) e do Laboratório de Educação Digital (LED) da Unimontes.



Fábila Magali Santos Vieira

<http://lattes.cnpq.br/5041706854491118>

Doutora em Educação pela Faculdade de Educação/Universidade de Brasília – UnB. Professora dos cursos de graduação e pós-graduação da Universidade Estadual de Montes Claros – Unimontes. Coordenadora dos projetos de pesquisas financiados pela Fapemig Educar: Laboratório Multiusuário de Tecnologias Digitais na Educação e Laboratório de Educação Digital – LED. Bolsista de produtividade da Fapemig.



Alcino Franco de Moura Júnior

<http://lattes.cnpq.br/7947733583131601>

Doutor em Administração pela Universidade de Brasília (UnB). Professor no Departamento de Ciências da Administração da Unimontes. Assessor de Comunicação/Projetos do Centro de Educação Profissional e Tecnológica (CEPT), Coordenador Adjunto do Laboratório de Educação Digital (LED), do projeto de extensão Hubware, e do Curso de Administração Pública do Instituto Federal do Norte de Minas Gerais (IFNMG), no âmbito da Universidade Aberta do Brasil (UAB).

O livro *Lógica de Programação: uma proposta para crianças com o uso do Scratch*, convida crianças a embarcar em uma jornada fascinante pela lógica de programação, utilizando a plataforma Scratch. Com linguagem acessível e atividades lúdicas, o livro explora conceitos fundamentais da programação de forma divertida. Através da criação de animações, jogos e histórias interativas, os pequenos desenvolvedores aprendem a pensar de forma lógica, a solucionar problemas e a dar vida às suas ideias. Uma experiência única para despertar a criatividade e o raciocínio computacional nas novas gerações, preparando-as para os desafios do mundo digital. E para quem está dando os primeiros passos na programação, este livro também é para você.