

STEMKIT

4SCHOOLS

Introdução ao Scratch 2.0

Plano de Aula 1



Cofinanciado pelo
Programa Erasmus+
da União Europeia

Este projeto é cofinanciado pelo Programa Erasmus + da União Europeia.

O apoio da Comissão Europeia à produção desta publicação não constitui um aval do seu conteúdo, que reflete unicamente o ponto de vista dos autores, e a Comissão não pode ser considerada responsável por eventuais utilizações que possam ser feitas com as informações nela contidas.



Índice

1.	Introdução ao Scratch 2.0	2
1.1	Informação geral.....	2
1.1.1	Breve descrição	2
1.1.2	Objetivos de aprendizagem.....	2
1.1.3	Ligação ao currículo.....	3
1.1.4	Materiais necessários	4
1.1.5	Duração	4
1.2	Plano de aula.....	4
1.2.1	Introdução ao Scratch	5
1.2.2	Preparação	5
1.2.3	Investigação.....	7
1.2.4	Conclusão.....	11
1.2.5	Exercício de follow-up (opcional).....	12
1.3	Referências ou Recursos.....	12



1. Introdução ao Scratch 2.0

1.1 Informação geral

1.1.1 Breve descrição

Neste plano de aula, o professor/a professora apresentará a aplicação Scratch, utilizada para criar projetos, contendo suportes, guiões e conceção de linguagem de programação para os/as jovens explorarem, para se expressarem e aprenderem. As atividades encorajam a exploração de conceitos e práticas chave de pensamento computacional.

Envolve três dimensões chave: (1) conceitos computacionais, (2) práticas computacionais, e (3) perspetivas computacionais.

Ao participar nas atividades da comunidade online e nos workshops do Scratch, os/as jovens criarão as suas próprias histórias interativas, jogos, e simulações, partilhando essas criações na comunidade online com outros/outras jovens programadores/as de todo o mundo.

A computação criativa apoia o desenvolvimento de ligações pessoais à computação, recorrendo à criatividade, imaginação e interesses. Os/as estudantes estarão mais preparados para carreiras profissionais como cientistas, informáticos e/ou programadores.

1.1.2 Objetivos de aprendizagem

O Scratch é uma linguagem de programação, criada pelo MIT Media Lab (Instituto de Tecnologia de Massachusetts), num ambiente de desenvolvimento de código aberto que facilita a criação de arte interativa, histórias, simulações e jogos. O seu objetivo é educar pessoas com pouca ou nenhuma experiência de programação, principalmente crianças entre os 8 e 16 anos de idade.

Esta aula introduz os alunos/as alunas aos conceitos centrais de programação informática e às capacidades de pensamento computacional, explorando aspetos do ambiente de programação Scratch.

É uma ótima forma de apresentar a programação a quem nunca teve qualquer experiência anterior. Os alunos/as alunas aprenderão a importar imagens e sons criados em Scratch, utilizando arte interativa, histórias, simulações, e jogos, uma ferramenta de pintura de edifícios e um gravador de som como atividade experimental.



Os principais objetivos de aprendizagem deste plano de aula são:

- Compreensão do conceito e conteúdo do Scratch 2.0., para inspirar os/as estudantes a aprender programação de computadores enquanto trabalham em projetos pessoalmente significativos, tais como histórias animadas e jogos.
- Conceber e realizar uma experiência ou investigação científica com recolha de dados, análise e apresentação de resultados, fornecendo ferramentas para resolver os desafios tecnológicos de amanhã.
- Familiarização com o Scratch, utilizado pelas escolas em múltiplas disciplinas (matemática, informática, artes linguísticas, estudos sociais).
- Compreender as estruturas básicas de programação, utilizando a linguagem de programação.

1.1.3 Ligação ao currículo

- ✓ O Scratch visa utilizadores mais jovens, centra-se na aprendizagem autodirigida, inclui ferramentas para desenhar imagens e gravar sons.
- ✓ O Scratch baseia-se nas ideias construcionistas (Construcionismo), para ajudar os utilizadores/as utilizadoras a tornar os seus projetos pessoalmente envolventes, motivadores e significativos.
- ✓ O Scratch facilita a importação ou criação de muitos tipos de suportes (imagens, sons, música); foi concebido para convidar à elaboração de guiões, fornecer feedback imediato na execução de guiões, tornar a execução e os dados visíveis.

Os/as estudantes são capazes de ver visualmente o progresso da sua aprendizagem no seu mundo, como uma série de diferentes projetos e construções físicas.

Os domínios, subdomínios, assuntos/tópicos a que este plano de aula pode ser ligado são:

- ✓ Ciência (Física/Química/Biologia/Geologia): método científico, investigação, experimentação, análise e interpretação dos resultados
- ✓ Ciências da computação /Informática: unidade de processamento e periféricos, interfaces, linguagem de programação e estruturas principais, codificação
- ✓ Tecnologia: eletrónica, hardware e software de fonte aberta, sensores, sinal digital, circuitos, computadores de placa única
- ✓ Matemática/Estatística: folhas de cálculo e estatísticas básicas



1.1.4 Materiais necessários

Para este plano de aula (e para cada grupo de estudantes) para além da consola STEMKIT, vamos precisar:

- Hardware e dispositivos para o educador/a educadora e para cada aluno/aluna
- Recomenda-se um PC, portátil ou tablet com um rato externo (a maioria dos estudantes acham mais fácil navegar no jogo com um rato em vez do touchpad)
- Os auscultadores são úteis durante o jogo (em alternativa, o áudio do jogo pode ser reduzido ou desligado)
- O acesso à Internet é necessário para login e multijogadores
- O projetor ligado a um computador com o Scratch aberto para mostrar quais os blocos e scripts que serão executados, e os blocos físicos Scratch (opcional)

1.1.5 Duração

A duração deste plano de aula é estimada em cerca de 45-60 minutos, ou seja, uma hora de aula.

1.2 Plano de aula

A atividade do aluno/da aluna irá centrar-se no domínio dos novos conceitos apresentados. A prática com o Scratch e a exploração de todas as possibilidades dos novos conceitos aprendidos é essencial para um processo de aprendizagem robusto. Os exercícios propostos no plano de aula são concebidos para reforçar a abordagem aprender fazendo.

O plano de aula ajuda os/as estudantes a desenvolver e reforçar os conhecimentos e as técnicas aprendidos nos tutoriais, para a futura profissão, desenvolvendo competências como colaboração, comunicação, pensamento crítico e pensamento sistémico.

O ambiente de aprendizagem aberto, dá aos/às estudantes a liberdade de experimentar, encorajando a autoexpressão criativa e a resolução de problemas.



1.2.1 Introdução ao Scratch

O Kit Informático inclui tudo o que é necessário para introduzir os/as estudantes à informática, eletrónica e codificação. Construa o seu próprio computador totalmente funcional e explore as STEAM (Ciência, Tecnologia, Engenharia, Artes e Matemática).

Muitas crianças não têm qualquer conceito sobre os componentes nos seus dispositivos ou como funcionam coisas básicas como ligações de rede e sistemas de ficheiros. O objetivo final da construção de computadores para crianças, torna-se a aprendizagem de como todas as partes físicas interagem e se relacionam com o que veem no ecrã.

Depois de estudarem os tutoriais, os alunos/as alunas replicarão no seu próprio ambiente Scratch as atividades apresentadas durante a aula. São encorajados/as a explorar, a começar pelo ambiente apresentado nos tutoriais, todas as possibilidades abertas pelos conceitos aprendidos.

Os estudantes serão introduzidos aos conceitos de pensamento computacional de loops, eventos, e paralelismo, familiarizando-se mais com os conceitos de sequência, blocos nas categorias Eventos, Controlo, Som, e Aparência + explorar vários programas de Scratch com tema artístico, criar um projeto de vídeo musical animado.

Ao completar esta atividade, os/as estudantes serão introduzidos aos conceitos de eventos (uma coisa que faz acontecer outra coisa) e de paralelismo (coisas que acontecem ao mesmo tempo) através da performance, serão capazes de explicar o que são eventos e como funcionam no Scratch, serão capazes de explicar o que é o paralelismo e como funciona no Scratch.

1.2.2 Preparação

Introduzindo o design (desenho) do Scratch

Os principais motores de mudança, têm sido o desejo de alargar as características do editor e de o tornar disponível numa gama mais vasta de dispositivos. A versão 2.0 é baseada na tecnologia Flash; a maioria das escolas e clubes de codificação desenvolveram os seus planos de aulas de codificação Scratch com base na versão 2.0.

Os alunos/as alunas criarão e executarão projetos em tablet, bem como no seu computador portátil e computador fixo (de secretária). Também estarão disponíveis em smartphones - embora o seu pequeno tamanho de ecrã possa ser um desafio. Começarão a explorar esta diversidade criativa com um mergulho profundo na animação, arte e música.

Existem duas versões do Scratch: Scratch 1.4 e Scratch 2.0. O editor offline Scratch 2.0 será utilizado para esta aula. Mas primeiro vamos ver o Scratch 2.0 online. Recomendamos trabalhar com FireFox ou Google Chrome quando se trabalha com Scratch online. O Internet Explorer pode ser problemático durante o Login. Navegue para Scratch 2.0 em scratch.mit.edu.

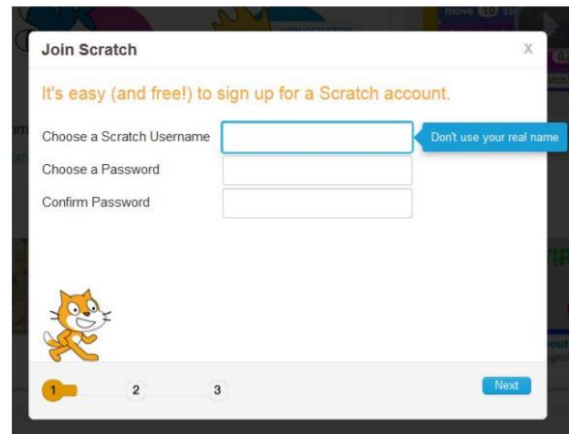


Fig.1. Navegar para o Scratch 2.0 em scratch.mit.edu

Clique em "Criar" para navegar para o ambiente de programação Scratch. Isto é como aceder ao ambiente de programação online Scratch 2.0. Demore algum tempo a explorar o website [Scratch.mit.edu](https://scratch.mit.edu) [Clique nas opções do menu Scratch, Create, Explore, Discuss e Help (Criar, Explorar, Discutir e Ajudar)].

O Editor do Projeto Scratch (Project Editor) está descrito em Dicas, Começar, Mapa do Editor do Projecto (Tips, Getting Started, Map of Project Editor.). Clique nisto para ver os detalhes do Editor do Projecto. A seguir, iniciar o Passo-a-Passo, começar com o Scratch.

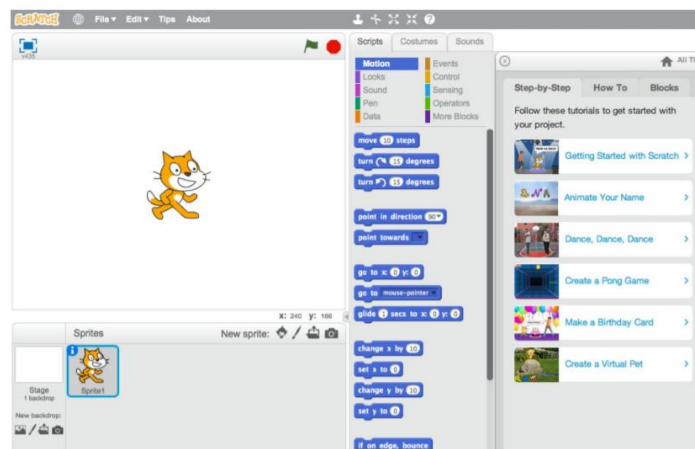


Fig.2. Navegar Passo-a-Passo, Introdução



Percorra cada uma das 13 etapas da introdução passo-a-passo. Este guia passo a passo introduz conceitos fundamentais de programação e permite aos/às participantes familiarizarem-se com o ambiente de programação Scratch.



Fig.3. Clique no botão HELP no topo para começar. (Fonte: <https://resources.scratch.mit.edu/www/guides/en/EducatorGuidesAll.pdf>)

1.2.3 Investigação

Atividade de Scratch: Animar o seu nome

Esta atividade foi concebida para ajudar os/as estudantes a explorar os conceitos computacionais de loops, eventos e paralelismo, culminando com a conceção de vídeos musicais personalizados. Os/as participantes ganharão experiência com a codificação à medida que animam as letras no seu nome.

Em primeiro lugar, reúnem-se em grupo para introduzir o tema e suscitar ideias. Peça a cada participante para dizer o seu nome, e depois peça a todos no grupo que façam a forma da primeira letra.

Para pré-visualizar o tutorial: scratch.mit.edu/nome ou vimeo.com/llk/nome.

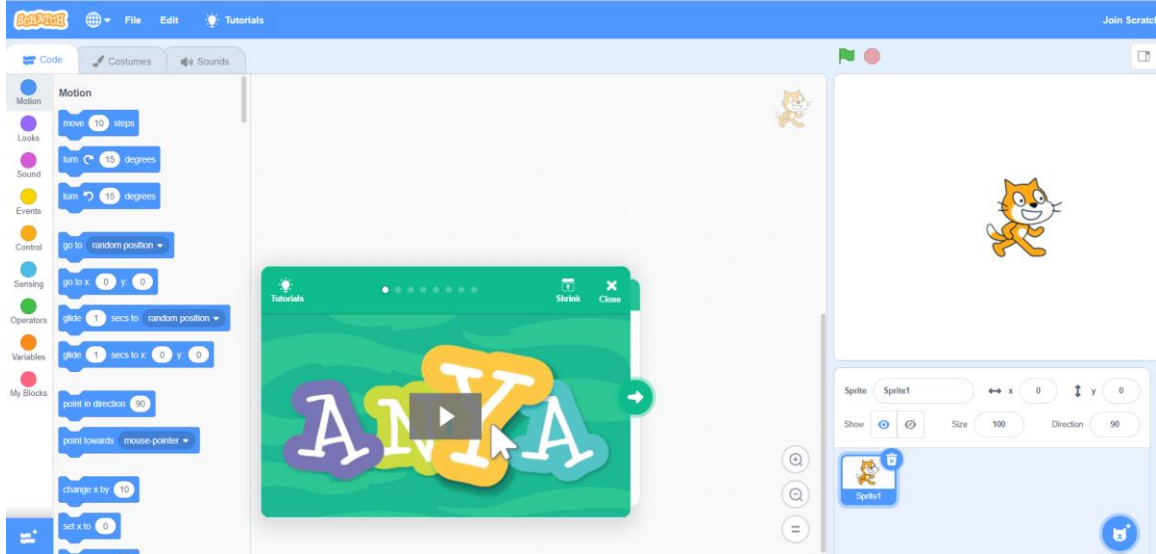


Fig. 4. Tutorial: scratch.mit.edu/nome

O Professor/a Professora apresenta o vídeo introdutório para o tutorial “Animar o Seu Nome”. O vídeo mostra uma variedade de projetos para ideias e inspiração.

O professor/a professora irá demonstrar os primeiros passos do tutorial e os alunos/as alunas verão como começar.

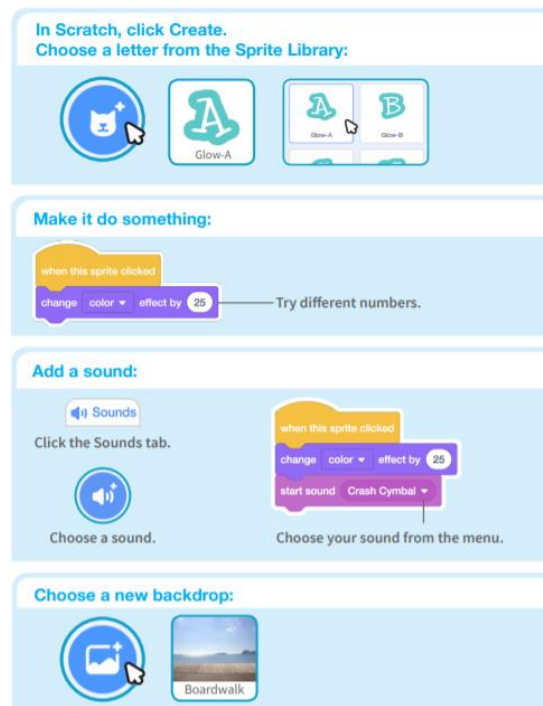


Fig. 5. Passos do tutorial: [scratch.mit.edu/nome](https://resources.scratch.mit.edu/www/guides/en/EducatorGuidesAll.pdf) (Fonte: <https://resources.scratch.mit.edu/www/guides/en/EducatorGuidesAll.pdf>)



O professor/a professora imprimirá os Cartões de Atividades - conjuntos de Cartões de Animar o Seu Nome para ter à disposição dos/das participantes durante a aula. (scratch.mit.edu/ideas).

Os professores/as professoras podem usar um projetor para mostrar exemplos e demonstrar como começar.

O tutorial “Animar o Seu Nome” mostra aos/às participantes como criar os seus próprios projetos. O professor/a professora irá apoiar os/as participantes na criação de projetos com nomes interativos.



Provide Resources
Offer options for getting started



Some participants may want to follow the online tutorial:
scratch.mit.edu/name

Others may want to explore using the activity cards:
scratch.mit.edu/ideas

Suggest Ideas for Starting

- Choose a letter
- Make it change color
- Add a sound
- Add a backdrop

Fig. 6. Criar projetos de nomes interativos (Fonte: <https://resources.scratch.mit.edu/www/guides/en/EducatorGuidesAll.pdf>)

Quando alguém fica preso/a, o professor/a professora irá conectá-los/las a outro/a outra participante que possa ajudar.

O professor/a professora ajudará os/as participantes a sentirem-se à vontade a experimentar diferentes combinações de blocos e a ver o que acontece.

Os/as estudantes podem utilizar as ideias e conceitos desta aula para criar uma grande variedade de projetos.

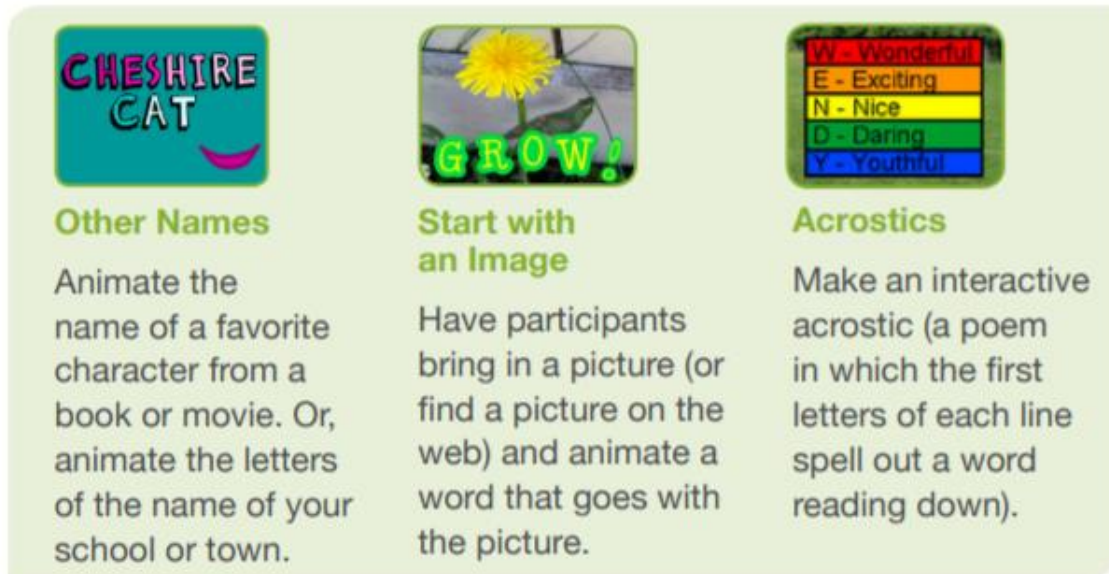


Fig. 7. Projeto de nome animado (Fonte: <https://resources.scratch.mit.edu/www/guides/en/EducatorGuidesAll.pdf>)

1.2.4 Conclusão

O **Scratch** é uma das ferramentas de codificação mais utilizadas nas escolas. O Scratch tem impacto em quase tudo o que fazemos na escola, por diversão, na nossa vida pessoal e profissional.

O ambiente e a linguagem de programação Scratch trabalham em conjunto para criar um sistema que é excepcionalmente rápido de aprender. Os educadores/as educadoras estão a integrar Scratch em muitas áreas disciplinares e grupos etários diferentes.

A atividade é concebida para apoiar a familiaridade e aumentar a fluência com a criatividade e o pensamento computacional. Os/as participantes estão agora prontos/as para completar algum trabalho de projeto utilizando recursos no Scratch.

Os/as estudantes devem guardar a sua própria cópia numa área apropriada na rede escolar com um nome de ficheiro apropriado, por exemplo, acrescentando as suas iniciais ao nome do ficheiro. Assim que um projeto for partilhado, outro utilizador/outra utilizadora pode acrescentar comentários.



Fig. 8. Guardar uma cópia do ficheiro Scratch no Scratch 1.4, 2.0 e 3.0 respetivamente
(Fonte: <https://resources.scratch.mit.edu/www/guides/en/EducatorGuidesAll.pdf>)

Na fase final, recapitulamos o que fizemos e como, quais foram os principais passos, discutimos quaisquer dificuldades experimentadas.

Encerramento: O que aprendi

Três perguntas-padrão de reflexão para avaliar a aprendizagem dos alunos/das alunas e dar resposta aos seus desafios e questões através da reflexão dos alunos/das alunas e da discussão em sala de aula durante o encerramento de cada aula.

- - O que aprenderam hoje que não sabiam antes?
- - O que foi mais desafiante?
- - Que perguntas têm?

1.2.5 Exercício de follow-up (opcional)

Como seguimento deste plano de aula, podemos proceder ao seguinte exercício:

Atividade no Scratch: Reproduzir as ações no tutorial. Experimentar o tamanho e a direção dos passos e das voltas.

1.3 Referências ou Recursos

Lista de referências úteis e recursos adicionais.

Aqui estão algumas referências úteis e recursos adicionais relacionados com este plano de aula:

- KAY, A. 2010. Squeak etoys, children, and learning. <http://www.squeakland.org/resources/articles>
- Resnick, M., Maloney, j., Monroy-Hernandez, 2009. Scratch: Programming for all. Comm. ACM 52, 11, 60–67.



- ComputerProgrammingInTheEnglishClassroom.pdf
- Maloney, J., Resnick, M., Rusk, N., Silverman, B., and Eastmond, E. 2010. *The scratch programming language and environment*. ACM Trans. Comput. Educ. 10, 4, Article 16 (November 2010), 15 pages. DOI = 10.1145/1868358.1868363. <http://doi.acm.org/10.1145/1868358.1868363>
- <https://education.abc.net.au/home#!/media/1214681/intro-to-scratch-20>
- <https://scratch.mit.edu>
- <http://web.media.mit.edu/~jmaloney/papers/ScratchLangAndEnvironment.pdf>
- <https://www.thomasbuxton.towerhamlets.sch.uk/blogs/year3/2017/11/17/year-3-computing-scratch-projects/>
- <http://scratched.gse.harvard.edu/guide/>
- scratch.mit.edu/name
- ScratchManualTermTime20152016.pdf
- <https://resources.scratch.mit.edu/www/guides/en/EducatorGuidesAll.pdf>

Glossário de termos-chave:

- Algoritmo: conjunto de passos para a realização de uma tarefa, procurando resolver um problema.
- Código ou programa de computador: conjunto de instruções que um computador pode seguir. Por exemplo, uma aplicação ou um jogo, como o Minecraft, é um programa de computador. Os termos podem ser utilizados indiferentemente.
- Sequência: identificação de uma série de passos para uma tarefa
- Loops: executar a mesma sequência várias vezes
- Paralelismo: fazer as coisas acontecerem ao mesmo tempo
- Acontecimentos: uma coisa faz com que outra coisa aconteça
- Condicionais: tomada de decisões com base em condições
- Operadores: apoio a expressões matemáticas e lógicas
- Dados: armazenamento, recuperação e atualização de valores