

Julho 2021

Mestrado em Ensino do 1.º Ciclo do Ensino Básico e de Matemática e de Ciências Naturais no 2.º Ciclo do Ensino Básico

Scratch no Ensino da Álgebra e Geometria no 2.º Ciclo do Ensino Básico

RELATÓRIO DE ESTÁGIO APRESENTADO À
ESCOLA SUPERIOR DE EDUCAÇÃO DE PAULA FRASSINETTI
PARA A OBTENÇÃO DE
GRAU DE MESTRE EM ENSINO DO 1.º CICLO DO ENSINO BÁSICO E DE MATEMÁTICA E DE CIÊNCIAS NATURAIS NO
2.º CICLO DO ENSINO BÁSICO

DE

Mónica Pinto Correia

ORIENTAÇÃO

Doutor Rui João Teles da Silva Ramalho



PAULA
FRASSINETTI

Scratch no Ensino da Álgebra e Geometria no 2.º Ciclo do Ensino Básico

Orientanda: Mónica Pinto Correia, 2016159, mestrado em ensino do 1.º ciclo do ensino básico e de matemática e de ciências naturais no 2.º ciclo do ensino básico

Orientador: Rui João Teles da Silva Ramalho

Data: 12 de julho de 2021

Resumo

O foco de investigação do presente relatório passa por estudar se a utilização do *Scratch* influencia positivamente o processo de ensino-aprendizagem, centrando-se no ensino de conteúdos na área da matemática do 2.º Ciclo do Ensino Básico (CEB) através da programação em *Scratch*, em específico no domínio Álgebra e Geometria e Medida no 6.º ano.

Primeiramente foi necessário investigar este recurso, assim como, as competências que a exploração do mesmo poderá promover junto dos estudantes, no sentido de perceber se o *Scratch* funciona como um instrumento facilitador das aprendizagens. Desta forma, investigou-se sobre o pensamento computacional, sobre o conceito de programação, sobre a linguagem de programação *Scratch* e o seu ambiente digital e por fim sobre os benefícios do ensino da matemática através da programação em *Scratch*.

Para tal foi envolvida uma turma de 20 estudantes de uma escola pública localizada na Foz do Douro, no Porto, na qual decorreram três aulas de matemática com programação para lecionar os conteúdos: expressão geradora de uma sequência numérica; reflexão central e reflexão axial.

A fim de compreender se de facto o *Scratch* influencia positivamente o processo de ensino-aprendizagem optou-se por usar tanto o método qualitativo, como o quantitativo, durante esta investigação-ação. Para tal, aplicaram-se questionários para avaliar o grau de satisfação e aquisição de conhecimentos aos estudantes no final de cada aula e realizou-se uma entrevista à professora cooperante, que conjuntamente com as observações realizadas, criaram dados suficientes para responder à questão de partida.

Palavras-chaves: Matemática; *Scratch*; Programação; Educação; Tecnologias

Abstract

The research focus of this report is to study whether the use of Scratch positively influences the teaching-learning process, focusing on the teaching of contents in the area of Mathematics of the 2nd Cycle of Basic Education (CEB) through programming in Scratch, specifically in Algebra and Geometry and Measure in the 6th grade.

First, it was necessary to investigate this resource, as well as the skills that its exploration can promote among students, to understand if Scratch works as a facilitator of learning. Thus, it was investigated about computational thinking, the concept of programming, the Scratch programming language, and its digital environment, and finally about the benefits of teaching mathematics through Scratch programming.

To this end, a group of 20 students from a public school located in Foz do Douro, in Porto, was involved, in which three mathematics classes took place with programming to teach the contents: expression generating a numerical sequence; central reflection, and axial reflection.

To understand if Scratch positively influences the teaching-learning process, it was decided to use both the qualitative and the quantitative methods during this action research. To this end, questionnaires were applied to assess the degree of satisfaction and acquisition of knowledge of students at the end of each class, and an interview was carried out with the cooperating teacher, who, together with the observations made, created enough data to answer the starting question.

Keywords: Mathematics; Scratch; Schedule; Education; Technologies

Índice

Introdução	1
Capítulo 1 – Revisão bibliográfica	2
1.1. O pensamento computacional.....	2
1.2. Programação	5
1.3. Scratch	9
1.4. Ensino da matemática através da programação.....	13
Capítulo 2 – Metodologia	16
2.1. Investigação-ação qualitativa/quantitativa	16
2.2. Caracterização do contexto de investigação.....	21
Capítulo 3 – Análise dos dados	24
Capítulo 5 – Conclusão.....	27
Referências bibliográficas	29
Apêndice	34
Atividades de programação com Scratch	34
Atividade 1	34
Atividade 2	45
Atividade 3	52
Entrevista.....	59

Lista de figuras

<i>Figura 0-1 – Ambiente de trabalho do Scratch</i>	10
<i>Figura 0-2 Exemplo de um projeto da 1.^a aula</i>	26

Lista de gráficos

Gráfico 1 – Avaliação da satisfação e aquisição de conhecimento da 1.ª aula..... 24

Gráfico 2 – Avaliação da satisfação e aquisição de conhecimento da 2.ª aula..... 25

Gráfico 3 – Avaliação da satisfação e aquisição de conhecimento da 3.ª aula..... 25

Lista de abreviaturas, siglas e acrónimos

TIC – Tecnologias da Informação e Comunicação

MIT – *Massachusetts Institute of Technology*

CEB – Ciclo do Ensino Básico

PE – Professora Estagiária

PC – Professora Cooperante

Introdução

O presente relatório de investigação, intitulado “*Scratch* no Ensino da Álgebra e Geometria no 2.º Ciclo do Ensino Básico”, surge no âmbito do Mestrado em Ensino do 1.º Ciclo do Ensino Básico e de Matemática e Ciências Naturais no 2.º Ciclo do Ensino Básico da Escola Superior de Educação de Paula Frassinetti, sob orientação do Doutor Rui Ramalho.

Este estudo surge na medida em que o ensino está em constante expansão de abordagens pedagógicas inovadoras e, atualmente, avança com o uso das tecnologias da informação e comunicação (TIC) para o ensino e a aprendizagem (Alves et al., 2015). Neste sentido, emergiu a pergunta de partida para este estudo: Será que a utilização do *Scratch* influencia positivamente o processo de ensino-aprendizagem? Esta questão surge na medida em que o *Scratch* se enquadra num dos novos *media* e artefactos de ludicidade e artefactos lúdicos que fazem parte das vidas das crianças e estão a alterar a forma como estes pensam, interagem e aprendem (Oliveira & Lopes, n.d.).

Assim sendo, e de modo a compreender melhor as potencialidades do *Scratch*, no primeiro capítulo, que é referente ao enquadramento teórico, é dada a conhecer a revisão bibliográfica sobre o pensamento computacional, a programação, o *Scratch* e por fim o ensino da matemática através da programação.

No capítulo seguinte, é exposta a metodologia escolhida (investigação-ação) e os métodos utilizados na mesma qualitativos e quantitativos. Ademais é feita, de forma pormenorizada, a contextualização da investigação. Esta, como já foi referido, decorreu ao longo do 3.º período do ano letivo 2020/2021, numa escola situada na Foz do Douro, no Porto, numa turma de 20 estudantes do 6.º ano do ensino básico.

Sendo que no último capítulo, é apresentada a análise dos dados recolhidos ao longo da investigação, as observações da investigadora, em conjunto com os questionários de avaliação da satisfação e aquisição de conhecimentos, realizados pelos estudantes, ao longo das aulas de matemática com o *Scratch* e com a entrevista realizada à professora cooperante. Nas considerações finais, reflete-se sobre os principais resultados obtidos através da investigação.

Capítulo 1 – Revisão bibliográfica

1.1. O pensamento computacional

“Quando uma pessoa usa um computador com o objetivo de resolver um dado problema, o tipo de pensamento envolvido no mapeamento do problema com vista à sua implementação computacional é designado de pensamento computacional.” (Pedro et al., 2017).

A história do pensamento computacional remonta pelo menos aos anos 50, mas a maioria das ideias são muito mais antigas. Ideias essas como a abstração, a representação de dados e a organização lógica dos mesmos. Nem o conceito nem o termo anteriormente referido são recentes. Este é precedido por termos como algoritmo, pensamento processual, pensamento algorítmico e alfabetização computacional por pioneiros da computação como Alan Perlis e Donald Knuth. O termo pensamento computacional surgiu pela primeira vez nas palavras de Seymour Papert em 1980 e novamente em 1996.

O pensamento computacional envolve a resolução de problemas, de modo a transformar um mais complexo em outro mais simples. Para isso, utiliza uma abordagem direcionada para a solução, envolvendo mecanismos de abstração, tais como a redução ou a divisão. Na resolução de problemas do mundo real, este tipo de pensamento é aplicado sistematicamente (Wing citado por Jesus et al., 2016).

Segundo este conceito, existe um processo de seleção e preparação onde, através de um mecanismo de abstração, designado por *prefetching*, se seleciona o que é essencial. Posteriormente, efetua-se a tarefa de *caching*, ao armazenar a seleção efetuada (Jesus et al., 2016). Este mecanismo é espelhado no ato de ir ao supermercado, onde se escolhe quais os produtos necessários (*prefetching*), colocando-os no carrinho de compras (*caching*).

Outro mecanismo de abstração é a tarefa de *backtracking*, um exercício de memória, onde se relembra passo a passo o que se fez ao longo do tempo até ao momento. Esta tarefa é realizada, por exemplo, quando se perde um objeto e se tenta reviver todos os passos dados até ao momento (Jesus et al., 2016).

Pode haver também um processo de escolha por melhor desempenho (rapidez e eficiência). Neste caso, pensa-se no modelo de desempenho de um “sistema multisservidor”, comparando o “multisservidor” ao conjunto existente (Jesus et al., 2016). Ou seja, é preciso levar o carro ao mecânico, há vários mecânicos a trabalhar que

se encontram disponíveis, então comparam-se as várias características, dentro das opções existentes, e seleciona-se a que melhor se adequa.

É possível, pelos motivos anteriormente referidos, afirmar que o conceito do pensamento computacional relaciona a capacidade do pensamento e raciocínio (humano) com as capacidades das máquinas e computadores (Philips, 2009). A essência deste pensamento é, de facto, pensar em dados, ideias, modelos e soluções para a resolução de problemas. A fluência em uma ou mais linguagens de programação e o pensamento computacional colaboram não apenas para o desenvolvimento cognitivo, como também para o desenvolvimento pleno, ao promover o raciocínio lógico, entre outras habilidades. (Martins & Eloy, 2019)

De acordo com Ribeiro et al., (2017) há neste caso, um conjunto de competências relevantes associadas ao desenvolvimento do pensamento computacional, como a capacidade de:

- Abstração – compreender as abstrações necessárias para estruturar dados e processos, e as técnicas de construção de soluções (algoritmos);
- Análise – consiste de técnicas de análise de processos e algoritmos, quanto à sua correção e eficiência, sob diferentes aspetos;
- Automação – envolve a mecanização de processos (ou subprocessos), reduzindo a necessidade de interação manual com os mesmos.

Ou seja, a capacidade de abstração permite a representação de uma situação (domínio) do mundo real, através de um modelo para a resolução de problemas.

Na área das ciências computacionais é necessário saber lidar com algoritmos, nomeadamente ao nível da sua análise, compreensão, desenvolvimento e manipulação. Este conhecimento é essencial no desenvolvimento de programas eficazes e eficientes para a resolução de problemas nos diversos domínios do conhecimento (Jesus et al., 2016). A programação estruturada permite fasear o processo de construção de um programa descrevendo o processo computacional de um modo não ambíguo (algoritmo), um programa pode ser definido pela forma seguinte: Programa = Estrutura de Dados + Algoritmo (Running, 2006).

Henderson et al., (2007) sugeriram que pensar computacionalmente era uma habilidade fundamental para todos, não apenas para os cientistas de computação, e defendia a importância de integrar ideias computacionais na escola.

Segundo Papert (citado por DGE, 2016) os estudantes não aprendem melhor pelo facto do professor ter encontrado melhores maneiras de os instruir, mas por lhes ter proporcionado melhores oportunidades de construírem. Surge assim, o construcionismo, teoria proposta por Papert em 1980, referindo-se a um provérbio africano para a explicar “Se um homem tem fome, poderás dar-lhe um peixe, mas no dia seguinte ele terá fome novamente. Se lhe deres uma cana de pesca e lhe ensinares a pescar, ele nunca mais terá fome.” (citado por Camacho & Fino, 2010). Nesta teoria, a aprendizagem acontece quando os estudantes se envolvem no desenvolvimento de algo que lhes é significativo, quer isso seja um castelo de areia, uma máquina, um poema, uma história, uma canção ou um programa de computador (DGE, 2016).

Deste modo, o construcionismo envolve dois tipos de construção: construção de objetos ou artefactos que o estudante realiza a partir de materiais cognitivos, recolhidos do mundo que o rodeia, e a construção do conhecimento que está relacionado com os objetos criados (DGE, 2016).

Assim sendo, na área educativa, o conceito de pensamento computacional implica que os estudantes se tornem “pensantes computacionais” de forma a entenderem como as tecnologias digitais podem ajudar a resolver problemas. Nesse processo de resolução estão incluídas características como a formulação de problemas computáveis; organização; análise e representação de dados, através de modelos e simulações; implementação de soluções, visando a otimização de passos e recursos, bem como a generalização das soluções para uma ampla gama de problemas (CTSA & ISTE, 2011).

Logo, o pensamento computacional, passa a ser entendido, para efeito desse trabalho, como um “conjunto de habilidades relacionadas à ciência de computação que deveriam ser desenvolvidas pelos estudantes de educação básica” com vista a propiciar-lhes o desenvolvimento de competências de resolução de problemas (Barcelos et al., 2015).

1.2. Programação

Estudos sobre desenvolvimento de *software* enquanto possibilidade de qualificação da atividade pedagógica, não são novidade. Seymour Papert, desde os anos de 1960, tem vindo a desenvolver pesquisas no *Massachusetts Institute of Technology* (MIT), através das quais defende que a programação promove o desenvolvimento da criatividade, de conhecimentos matemáticos, físicos e de outras áreas, na qual os estudantes se tornam sujeitos ativos da sua aprendizagem (Escola Superior de Educação do Instituto Politécnico de Setúbal, 2015).

De acordo com a professora Alexandra Costa (citada por Cardoso, 2013) deve promover-se no estudante esta capacidade, de modo a que seja ele o ator da sua própria aprendizagem, levando-o a reconhecer não só o interesse e importância da aprendizagem em geral, como também a importância do que deve aprender em cada uma das disciplinas.

Segundo Papert, a aprendizagem está relacionada com a ação de inventar e, neste processo, a criança concretiza na linguagem do *software* conhecimentos que muitas vezes são abstratos e de difícil compreensão (Escola Superior de Educação do Instituto Politécnico de Setúbal, 2015). Assim sendo os estudantes não aprendem só a programar, também exercitam a criatividade e o raciocínio científico, lógico e matemático.

Isso acontece porque é necessário compreender e aplicar os princípios e conceitos fundamentais da programação (lógica, tipo de dados, variáveis, estruturas condicionais e repetitivas, entre outros). Assim como analisar programas, identificando o seu resultado, erros e respetivas correções, de modo a otimizar-se a solução encontrada para determinado problema (Pedro et al., 2017). Isto levará ao desenho de programas com diversos níveis de complexidade na resolução de problemas específicos.

De acordo com Sobral (citada por Ressurreição, 2012) a programação divide-se em quatro fases:

1. Conceptualização e análise, em que os problemas são divididos em tarefas mais pequenas;
2. Resolução e criação de algoritmos usando pseudocódigo que traduza os problemas e as tarefas da fase anterior;
3. Definição das variáveis e estruturas de dados para caracterizar o modelo de dados a utilizar;
4. Tradução do algoritmo numa linguagem de programação e a sua execução.

Para além destas fases, a autora menciona outras, como a verificação de erros e a comparação dos resultados obtidos com os esperados.

Na perspetiva de Running (2006) a criação de um programa de computador passa por cinco passos principais, que são coletivamente conhecidos como produção de *software*, sendo que a programação coloca ênfase nos passos 2, 3 e 4, já a codificação no passo 3:

1. Reconhecer a necessidade de um programa para resolver um problema ou fazer alguma coisa;
2. Planificar o programa e selecionar as ferramentas necessárias para resolver o problema;
3. Escrever o programa na linguagem de programação escolhida;
4. Compilar: traduzir o código fonte legível pelo homem em código executável pela máquina, o que é feito através de compiladores e outras ferramentas;
5. Testar o programa, para ter a certeza de que funciona; se não, regressar ao passo três.

Pode-se então afirmar que no que concerne ao conceito de programação, este é o processo de escrita, teste e manutenção de um programa de computador.

Um programa é escrito numa linguagem de programação, embora seja possível, com alguma dificuldade, escrevê-lo diretamente em linguagem máquina. Diferentes partes de um programa podem ser escritas em diferentes linguagens, diferentes linguagens de programação funcionam de diferentes modos.

A linguagem de programação escolhida, para este relatório, foi o *Scratch*. Este é um *software* de programação desenvolvido pelo grupo de investigadores *Lifelong Kindergarten* do MIT, que é uma combinação perfeita entre duas linguagens de programação LOGO e LEGO (Ramalho & Ventura, 2017).

O *software* é o conjunto de instruções, dados ou programas usados para operar computadores e executar tarefas específicas, geralmente dividido em *software* de aplicação e *software* de sistema, que inclui sistemas operacionais e qualquer programa que suporte *software* de aplicação (Ressurreição, 2012).

Contrariamente ao *hardware*, as componentes físicas de um computador, o primeiro conceito é um termo genérico usado para se referir a aplicações, *scripts* e programas executados num dispositivo. Esta designação pode ser considerada a parte

variável de um computador enquanto o *hardware* é considerado a parte invariável (Running, 2006).

Um programa de computador não existe sem um algoritmo associado, pois este é uma sequência finita ou infinita de ações executáveis que visam obter uma solução para um determinado tipo de problema, os algoritmos são procedimentos precisos, não ambíguos, mecânicos, eficientes e corretos (Dasgupta et al., 2006).

A Algoritmia centra-se na forma como se representam entidades e objetos reais através de estruturas de dados e não na forma como são armazenados fisicamente na memória do computador. Uma estrutura de dados primitiva são a base para estruturas de dados mais complexas, ambas as estruturas (primitivas e complexas) definem a forma como os dados de um algoritmo são relacionados e agrupados de modo coerente. Estruturas de dados primitivos são, por exemplo, valores *booleanos*, numéricos e alfanuméricos (Jesus et al., 2016):

- Tipo de dados *booleano* – este tipo é aplicado em situações reais que unicamente denotam dois estados possíveis: 0 (falso) ou 1 (verdadeiro);
- Tipo de dados numérico – este tipo é representativo de valores numéricos no domínio dos números inteiros e reais.
- Tipo de dados alfanumérico – o tipo de dados alfanuméricos é constituído por uma sequência de caracteres contendo letras; dígitos e símbolos especiais.

A definição das estruturas de dados a utilizar por um algoritmo também deverá ter em atenção as operações a executar sobre os dados, que podem ser para: criar, alterar e eliminar estruturas de dados; inserir, alterar e eliminar dados desses objetos; aceder a estruturas de dados (Jesus et al., 2016).

Uma eficiente manipulação das estruturas de dados, associados à resolução de um problema e conseqüente construção de um algoritmo, envolve uma análise dos seguintes aspetos (Vasconcelos, 2015):

- Compreender a relação entre os dados;
- Compreender a gestão das estruturas dos dados em memória;
- Decidir as operações a executar nos dados logicamente relacionados;
- Representar os elementos dos dados;
- Manter as relações lógicas entre os dados;
- Executar de forma eficiente as operações sobre os dados;

- Especificar genericamente os algoritmos e as estruturas de dados e implementá-los numa linguagem de programação.

Tendo em conta estes fatores, compreende-se que um algoritmo é uma sequência de passos que realizam uma tarefa ou solucionam um problema. No quotidiano são utilizados algoritmos para realizar atividades, define-se a sequência de atividades que devem ser realizadas para atingir um determinado objetivo, como por exemplo uma receita, onde temos que seguir os passos descritos com precisão pela ordem, para que se consiga cozinhar uma receita perfeita (Running, 2006).

Dessa forma, um algoritmo é uma descrição passo a passo de como o computador irá executar uma operação específica, como, por exemplo, uma ordenação. Um programa, por outro lado, é uma entidade que na verdade implementa uma ou mais operações de forma que seja útil para as pessoas que o utilizam.

É possível então concluir que o conceito abordado é, num certo sentido, um programa abstrato, ou seja, um programa é um algoritmo concretizado. Os programas são, neste caso, visualizados mais facilmente como uma coleção de algoritmos menores combinados de um modo único, da mesma forma, por exemplo, que uma casa é construída a partir de componentes.

1.3. *Scratch*

O *Scratch* é uma linguagem de programação que permite programar e partilhar multimédia interativa, como histórias, jogos e animações interativas, na comunidade em linha, com pessoas de todo o mundo. À medida que as crianças criam estão a aprender a pensar de forma criativa, a raciocinar de modo sistémico e a trabalhar colaborativamente, estas competências de informação e comunicação, de raciocínio e resolução de problemas, interpessoais e de colaboração, são essenciais à vida no séc. XXI (Gordinho, 2009).

O grupo *Lifelong Kindergarten do MIT Media Lab*. concebeu esta linguagem de programação, sendo que a organização que, atualmente, trabalha no *design*, desenvolvimento e moderação do *Scratch* é a *Scratch Foundation*, uma organização sem fins lucrativos, motivo pelo qual a plataforma é de uso gratuito, sendo o seu desenvolvimento e manutenção pago por subsídios e doações.

Inicialmente foi desenvolvido para crianças com idades compreendidas entre os 8 e os 16 anos, porém já criaram uma versão simplificada, o *Scratch Jr.*, idealizada para crianças dos 5 aos 7 anos. É utilizado por pessoas de todas as origens, em todos os países à volta do mundo, em todos os tipos de contextos: em casa, nas escolas, em bibliotecas, nos museus e muito mais, estando traduzido, por voluntários, em mais de 40 línguas, incluindo a língua portuguesa (The Scratch Foundation, n.d.).

Para trabalhar no *Scratch* é apenas necessário ter um computador de secretária ou portátil ou até mesmo um tablete, pois o *Scratch* corre na maioria dos navegadores *web*, sendo igualmente possível transferir o programa para um dispositivo eletrónico, o *Scratch* 3.0, e criar projetos estando *offline*, sem acesso à internet.

Centenas de milhares de escolas em todo o mundo usam o *Scratch*, em diversas áreas disciplinares, incluindo línguas, artes, ciências, história, matemática e ciências da computação, por esse motivo existe uma página oficial específica para educadores, local onde é possível descobrir mais sobre estratégias e recursos para usar o *Scratch* em ambiente educativos.

Apesar de não ser necessário criar conta para programar com o *Scratch*, para partilhar na comunidade em linha, é obrigatória a criação da mesma. Esta comunidade é moderada e regulada pelas Linhas de Orientação da Comunidade *Scratch*, tendo também acesso a outros projetos podendo até comentá-los, algo que é incentivado pela

comunidade. Quando se cria uma conta é se automaticamente considerado um Novo *Scratchador*, só se consegue evoluir para *Scratchador* quando se criar, partilha e comenta, de forma construtiva, projetos da comunidade, pois dessa forma cresce enquanto programador, e não só (The Scratch Foundation, n.d.).

Resnick destaca que se estabelece um “ciclo de retornos positivos de aprendizagem (...) quando os estudantes concebem projetos, têm ideias novas que os levam a conceber novas coisas, a partir das quais voltam a surgir novas ideias e assim sucessivamente” (Gomes & Freitas, 2017). Este encorajamento no que diz respeito ao pensamento criativo é, sem dúvida, importante para o mundo em que vivemos, pois este está sempre em crescente evolução e sempre em mudança. O *Scratch* envolve os jovens na procura de soluções inovadoras para problemas inesperados – prepara-os para encontrar novas soluções à medida que vão surgindo novos desafios e não apenas para saber como resolver um problema pré-definido (Rusk et al., 2006).

O principal objetivo deste software é programar e possibilitar aos iniciantes a criação de programas no computador sem a aprendizagem prévia de uma sintaxe de linguagem de programação (Maloney et al., 2010), isto porque o *Scratch* surgiu da combinação perfeita entre a linguagem Logo e a Lego, tendo sido “criado a partir da ideia do artefacto de ludicidade Lego, substituindo a ideia de código da programação, por pega-encaixe-larga, característica da construção Lego” (Gordinho, 2009).

Este ambiente de programação é composto por cinco áreas principais: o palco; o painel de cenários; a listagem de atores; a paleta de código em blocos e a área de guiões, havendo também uma área de trajes e outra de sons, como se pode observar na seguinte figura:

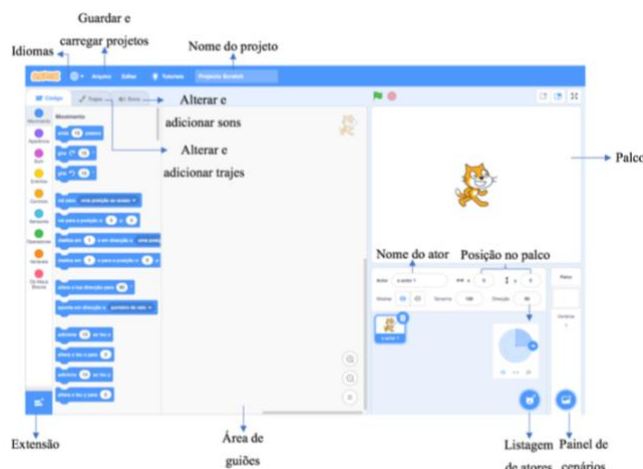


Figura 0-1 – Ambiente de trabalho do Scratch

Como se pôde observar na paleta de blocos há nove categorias de instruções mais uma extensão, cada categoria tem um nome característico e uma cor associada, ou seja, todos os blocos de código tem uma cor, de acordo com a categoria a qual estão associados:

- A categoria de movimento, designada com a cor azul, têm quinze blocos que permitem o ator deslocar-se pela área do palco.
- A categoria de aparência, designada com a cor roxo, têm dezassete blocos que possibilitam ao ator alterar o seu aspeto, atribuir-lhe falas e fazê-lo aparecer e desaparecer.
- A categoria de som, designada com a cor púrpura, têm oito blocos que proporcionam a atribuição de sons ao ator, assim como o controlo do volume e adição de efeitos.
- A categoria de eventos, designada com a cor amarelo, têm oito blocos que geralmente são colocados no início do código, pois programam o projeto para reagir a ações específicas.
- A categoria de controlo, designada com a cor laranja, têm onze blocos que criam condições e repetições na execução do programa.
- A categoria de sensores, designada com a cor azul turquesa, têm dezoito blocos que são utilizados em conjunto com os blocos laranja, de controlo, ou os verdes, de operadores.
- A categoria de operadores, designada com a cor verde, têm dezoito blocos que são utilizados em união com outros blocos, nomeadamente os azul turquesa, de sensores, e os laranjas, de controlo, contendo também várias funções matemáticas.
- A categoria de variáveis, designada com a cor laranja escura, têm quatro blocos que só podem ser usados quando é criada uma variável ou uma lista.
- A categoria de os meus blocos, designada com a cor rosa, não têm blocos, pois são criados pelo utilizador.
- Há ainda a extensão que contém vários blocos diferentes, alguns criados em colaboração com outras entidades como a Lego, a Google e a Amazon.

O *Scratch* foi desenhado de modo a que o utilizador dedique tempo apenas à construção da lógica necessária para o desenvolvimento de uma tarefa, os blocos de programação são feitos de forma a só encaixarem numa forma sintática correta. Assim

sendo, os estudantes não necessitam de se preocuparem com erros de sintaxe (Andrade et al., 2013). A codificação é realizada por blocos (simples pré-constituídos), deste modo a construção é orientada por sequências e iterações de ações a serem implementadas, permitindo que as crianças criem instruções complexas (Cruz et al., 2020).

É assim perceptível que criando projetos *Scratch* os utilizadores aprendem a seleccionar, criar e gerir múltiplas formas de *media*, incluindo texto, imagens, animações e registos áudio, adotando também formas de raciocínio crítico e de pensamento sistémico, pois para construir projetos necessitam de coordenar o tempo e interação entre múltiplos atores (Gordinho, 2009).

1.4. Ensino da matemática através da programação

O Ministério da Educação (2017) diz que “Educar no século XXI exige a percepção de que é fundamental conseguir adaptar-se a novos contextos e novas estruturas, mobilizando as competências, mas também estando preparado para atualizar conhecimento e desempenhar novas funções”, deste modo “Todas as crianças e jovens devem ser encorajados, nas atividades escolares, a desenvolver e a pôr em prática os valores por que se deve pautar a cultura de escola, como querer aprender mais; desenvolver o pensamento reflexivo, crítico e criativo; procurar novas soluções e aplicações.”(Ministério da Educação, 2017).

Estes valores estipulados no Perfil dos Alunos a Saída da Escolaridade Obrigatória, são espelhados durante a programação de um projeto, sendo também estipulados, como objetivos específicos, pela Direção-geral da Educação (2016) nas Linhas Orientadoras para a Iniciação à Programação no 1.º CEB, a fomentação de estratégias de resolução de problemas a partir de necessidades identificadas nos projetos, a estimulação da curiosidade pela investigação, assim como da criatividade no âmbito do cruzamento de saberes de diferentes áreas.

Silva, Zampieri & Javaroni (Martins & Eloy, 2019) dão vários exemplos de como ensinar através da programação na disciplina de matemática e não só. Um dos exemplos remete para um projeto sobre a pirâmide alimentar, onde trabalham a quantidade de calorias ingeridas pelo ser humano ao consumir determinados alimentos, projeto este que foi desenvolvido no *Scratch*, sendo que dos cinco professores envolvidos, duas eram de ciências, dois de matemática e um de educação física.

Desta forma, os estudantes conjugam o pensamento matemático com o pensamento computacional, interligando com conteúdos das ciências naturais e da educação física, para resolverem os problemas que lhes são propostos pelos professores ou que lhes surgem durante a sua programação, em simultâneo estão a estimular e desenvolver a sua criatividade.

Mitchel Resnick, diretor do grupo *Lifelong Kindergarten* do MIT, numa conferência TED (Resnick & TED, 2012), contou a história de uma visita que fez a uma escola que utilizavam o *Scratch*. Um dos estudante tinha criado um jogo onde um peixe grande, controlado pelo utilizador, comia peixes mais pequenos, que se moviam no ecrã. O estudante em questão, queria saber quantos peixes tinham sido comidos e pediu ajuda

ao investigador. Resnick explicou-lhe que o poderia fazer criando uma variável que seria incrementada cada vez que um peixe era comido. Com alguma ajuda, passados poucos minutos, o estudante criou uma variável e a contagem começou a aparecer no ecrã. No final, agradeceu ao investigador dizendo-lhe: "Obrigado, obrigado, obrigado! Obrigado por me ensinar o que são variáveis!". Este estudante tinha a necessidade de aprender variáveis, conteúdo lecionado no 6.º ano, em organização e tratamento de dados, para resolver o seu problema. Este acontecimento demonstra que é possível lecionar conteúdos matemáticos através da programação.

No ensino da matemática há três grandes finalidades, de acordo com o Programa e Metas Curriculares da Matemática no Ensino Básico, sendo uma delas a estruturação do pensamento, ou seja, os estudantes devem aprender e hierarquizar conceitos matemáticos, estudar de forma sistemática as suas propriedades e argumentar de forma clara e precisa, estes passos descritos contribuem para a criação de uma gramática basilar do raciocínio hipotético-dedutivo (Ministério da Educação, 2013).

O trabalho desta gramática contribui para a fomentação da capacidade de elaboração de análises objetivas, coerentes e comunicáveis, assim como para uma melhor capacidade de argumentação, de justificação de uma dada posição e de detetar falácias e raciocínios falsos (Ministério da Educação, 2013). Capacidades essas que são desenvolvidas através da programação, pois é necessário selecionar os elementos que melhor se adequem à resolução dos projetos, identificar e lidar com o erro/ falha, redesenhar os projetos corrigindo as falhas identificadas e melhorar e aperfeiçoar os trabalhos desenvolvidos (Direção Geral da Educação, 2016).

O professor João Vítor Torres (Conselho Nacional de Educação, 2017) dá o seu testemunho sobre como ensinar as propriedades do quadrado, conteúdo lecionado no 5.º ano, em geometria e medida, através da programação. O objetivo era desenhar um quadrado no *Scratch*, para tal teriam de programar o ator a "repetir quatro vezes" o comando de "andar 60 e girar 90º". Só neste código o estudante utiliza o conhecimento matemática das propriedades do quadrado como da multiplicação.

Portugal, Nunes & Goetz (Amélia & Carvalho, 2020) exemplificam como os estudantes podem aprender os ângulos e as figuras geométricas através da programação em *Scratch*, assim como Ventura & Ramalho (2018) demonstram que é possível trabalhar

a localização e orientação no espaço, através do registo de percursos, e as figuras geométricas.

No 6.º ano do 2.º CEB em geometria e medida no domínio das isometrias no plano é espectável que os estudantes aprendam a construir o transformado de uma dada figura através de reflexões centrais e axiais, prevendo e descrevendo os resultados obtidos, já em álgebra no domínio das sequências e regularidades é pressuposto que saibam determinar termos de uma sequência definida por uma expressão geradora (Ministério da Educação, 2013, 2018). Estas são as áreas de domínio onde incidiu a intervenção deste projeto.

Capítulo 2 – Metodologia

2.1. Investigação-ação qualitativa/quantitativa

A educação é como que uma caminhada de aperfeiçoamento que os membros de uma comunidade realizam com a ajuda e o apoio de outros, logo investigar em educação não é o mesmo que investigar numa outra área social. Devido à especificidade do fenómeno educativo, é às ciências da educação que compete procurar explicar e compreender a complexidade e multireferencialidade dos fenómenos educativos e, ao mesmo tempo, construir de forma progressiva, inter e transdisciplinarmente, um património de saberes ao dispor das práticas (Amado, 2014).

A investigação em ciências sociais é, geralmente, qualificada como um trabalho que pretende compreender melhor os significados de um acontecimento ou de uma conduta, captando com maior perspicácia as lógicas de funcionamento de uma organização, refletindo acertadamente sobre as implicações de uma decisão, ou ainda melhor compreender com mais nitidez como determinadas pessoas aprendem um problema e tornam visíveis alguns dos fundamentos das suas representações (Campenhoudt et al., 2019).

Numa investigação o primeiro problema que se põe ao investigador é saber qual o seu objetivo, ou seja, qual é a questão fundamental a que quer dar resposta, pois este é um ponto de partida, como um acampamento-base que os alpinistas constroem para se prepararem para a escalada de um cume. Neste processo de investigação a pergunta que se colocou foi “Se a utilização do *Scratch* influencia positivamente o processo de ensino-aprendizagem”, com os seguintes objetivos: – motivar os estudantes para a aprendizagem dos conteúdos na área da matemática; – proporcionar uma abordagem diferente no ensino de conteúdos matemáticos.

De forma a se conseguir obter uma resposta a esta pergunta escolheu-se utilizar uma metodologia de investigação-ação, que consiste na recolha de informações sistemáticas com o objetivo de promover mudanças sociais, a investigação a ser aplicada procura resultados que possam ser utilizados pelas pessoas para tomarem decisões práticas relativas a determinados aspetos da sua vida (Bogdan & Biklen, 1994).

A expressão investigação-ação, que ficou a dever-se a Kurt Lewin, exprime alguma ambiguidade de sentidos que a torna polissémica, porém Simões (1990) considera que:

(...) evitar-se-iam muitas confusões, se a ação fosse entendida no sentido instrumental – uma intervenção sobre a situação real – e se a investigação fosse encarada como a busca de uma resposta a um problema, através de dados empíricos, recolhidos de uma forma sistemática e controlado, portanto, com caráter público e objetivo (...) [e acrescenta que] tal perspectiva implicaria, porventura, negar o caráter científico a algumas daquelas atividades, sem que isso necessariamente significasse negar a sua utilidade.

Guerra (2007) diz que a investigação-ação não constitui tanto uma técnica de recolha de informação quanto uma nova aproximação da investigação, sendo uma modalidade que torna o ator investigador, e vice-versa, e que conduz a ação para considerações de investigação. O seu ponto de partida é particular, já que fundamenta a sua dinâmica sobre a ação, e considera os atores não como objetos passivos de investigação, mas como sujeitos participantes.

Pode-se considerar que há duas modalidades de investigação-ação: Investigação-para-a-ação e Investigação-na/pela-ação. A primeira é desencadeada por alguém “que tem necessidade de informações/conhecimentos de uma situação/problema a fim de agir sobre ela e dar-lhe soluções” (Esteves, 1986), a segunda constitui-se como um procedimento de grande complexidade, devido à multidirecionalidade e coexistência dos seus objetivos, Esteves aponta para a produção de conhecimentos (objetivos de investigação), para a introdução de mudanças (objetivos de inovação) e de formação de competência nos participantes (objetivos de formação), tratando-se por isso de um processo coletivo que envolve investigadores e a sociedade em estudo (Amado, 2014), motivo pelo qual a Investigação-na/pela-ação foi a selecionada para este estudo.

O conceito de investigação-ação consiste numa investigação aplicada, na qual o investigador se envolve ativamente na causa da mesma, podendo tanto usar os métodos qualitativos como os quantitativos. Na investigação qualitativa a fonte direta de dados é o ambiente natural, construindo o investigador o instrumento principal, sendo que os dados recolhidos são descritivos, em forma de palavras ou imagem, e não números. Por esse motivo os investigadores interessam-se mais pelo processo do que simplesmente os resultados ou produtos, e quando analisam os dados fazem-no de forma indutiva (Bogdan & Biklen, 1994). Para estes o significado é de extrema importância, motivo pelo qual estão continuamente a questionar os sujeitos de investigação, com o objetivo de perceber “aquilo que eles experimentam, o modo como eles interpretam as suas experiências e o modo como eles próprios estruturam o mundo social em que vivem” (Psathas, 1973).

Já o segundo método, o de investigação quantitativa é um processo sistemático de recolha de dados observáveis e quantificáveis. É por isso baseado na observação de factos objetivos, de acontecimentos e de fenómenos que existem independentemente do investigador (Fortin, 2003). Nos estudos quantitativos é admitido tudo o que pode ser quantificável, isto é, que é possível traduzir em números, as opiniões e as informações para, de seguida, poderem ser classificadas e analisadas, visando a apresentação e a manipulação numérica de observações com vista à descrição e à explicação de fenómenos sobre o qual recaem as observações (Vilelas, 2009). Para tal usam técnicas como inquéritos, entrevistas estruturadas, experimentos ou quase experimentos, observações estruturadas e conjuntos de dados (Bogdan & Biklen, 1994).

Existem estudos que integram componentes qualitativos e quantitativos, essa prática é comum quando se constroem questionários, frequentemente, a estatística descritiva e os resultados qualitativos são apresentados conjuntamente (Bogdan & Biklen, 1994). Nesta investigação, usou-se ambos os métodos, sendo que o qualitativo se baseou na observação de três aulas e na entrevista aberta à professora cooperante, enquanto o quantitativo se baseou nos questionários de satisfação e aquisição de conhecimentos das aulas, preenchidos pelos estudantes.

Como foi referido estes métodos, qualitativo e quantitativo, usam técnicas de recolha e análise de dados distintas, sendo uma das técnicas, comum a ambos os métodos a observação, sendo que há diferentes formas de observar um grupo de forma direta e participante ou não participante.

A observação direta capta os comportamentos no momento em que eles se produzem, sem a mediação de um documento ou de um testemunho. O campo de observação do investigador é, *a priori*, infinitamente amplo e só depende, em definitivo, dos objetivos do seu trabalho e das duas hipóteses de partida. Esta observação é aquela em que o próprio investigador procede diretamente à recolha das informações, sem se dirigir aos sujeitos interessados, apelando diretamente ao sentido de observação. O ato de observar poderá ser através de uma grelha de observações e/ou recolha de imagens. (Campenhoudt et al., 2019).

Na observação participante há um envolvimento direto do investigador de campo com o grupo social que estuda, dentro de parâmetros das próprias normas do grupo. É este envolvimento que despoja o investigador do seu conhecimento cultural próprio,

enquanto veste o do grupo investigado (Silva & Pinto, 1986). A observação é realizada de forma continuada, de modo a conseguir confrontar as observações e as hipóteses interpretativas, num movimento iterativo.

As observações não participantes apresentam perfis muito diferentes, sendo o seu único ponto comum o facto de o investigador não participar na vida do grupo, ou seja, observa do “exterior”. Pode ser de longa ou curta duração, com ou sem o consentimento do grupo (Campenhoudt et al., 2019).

Por estes motivos o tipo de observação considerada mais adequada para este estudo foi a direta e participativa, visto que o investigador é a professora estagiária da turma, estando por isso não só presente nas três aulas de implementação do *Scratch*, como nas restantes aulas de matemática e ciências naturais da turma, durante o 1.º e 3.º período do ano letivo 2020/2021.

A entrevista é um dos mais poderosos meios para se chegar ao entendimento dos seres humanos e para a obtenção de informações nos mais diversos campos, podendo classificar-se de diferentes modos: estruturada, não estruturada e semiestruturada (Amado, 2014).

No que concerne à primeira, geralmente, centra-se num tema determinado e restrito, daí o nome *focused interview*, é útil para a obtenção de informação quantificável, de um grande número de entrevistados (Amado, 2014; Campenhoudt et al., 2019). Ademais, deve ser preparada antecipadamente, o que significa que haverá um guião por onde o entrevistador se orientará, onde as mesmas perguntas são aplicadas a todos os entrevistados, não havendo espaço para o entrevistado contar a sua história em termos pessoais, pelas suas próprias palavras (Bogdan & Biklen, 1994). Em síntese, “nas entrevistas estruturadas, cada entrevistado responde a uma série de perguntas pré-estabelecidas dentro de um conjunto limitado de categorias de respostas.” (Afonso, 2014).

Ao contrário do tipo anterior, a entrevista não estruturada, é considerada por Bogdan & Biklen (1994) uma entrevista muito aberta, pois parte de uma noção de grande complexidade do comportamento humano, motivo pelo qual o entrevistador tem mais liberdade, já que este modelo é mais flexível. “(...) A interação verbal entre entrevistador e entrevistado desenvolve-se à volta de termos ou grandes questões organizadoras de discurso, sem perguntas específicas e respostas codificadas.” (Afonso, 2014). Significa

isto que as perguntas derivam da interação, não existindo, portanto, qualquer tipo de grelha prévia de questões (Amado, 2014).

Por fim, surge a entrevista semiestruturada, “(...) não é inteiramente aberta nem encaminhada por um grande número de perguntas precisas” (Campenhoudt et al., 2019). Ou seja, as questões derivam de um plano prévio, um guião onde se define e regista, numa ordem lógica para o entrevistador, o essencial do que se pretende obter, embora na interação se venha a dar uma grande liberdade de resposta ao entrevistado (Amado, 2014). De acordo com Bogdan & Biklen (1994) com este tipo de entrevista fica-se com a certeza que se irá obter dados comparáveis

Neste estudo optou-se por entrevistar de forma semiestruturada a professora cooperante, sendo que os dados obtidos foram gravados e posteriormente transcritos, tendo sido sujeitos a análise do conteúdo.

Relativamente aos questionários, na maior parte das vezes intervêm num processo de dados numéricos destinados a medir e/ou a compreender um determinado fenómeno social. Deste modo, é uma ferramenta de objetivação dos fenómenos sociais observados na sala de aula. Traduzindo assim uma visão simplificada da realidade social assumida pelo investigador e inscrita no seu modelo de análise (Hill & Hill, 2009). Assim sendo, o questionário apenas poderá desempenhar este papel com um certo grau de padronização, que neste caso poderá ser extremo, pois apenas há perguntas de resposta fechada, porque o estudante é obrigado a escolher entre as respostas predefinidas. Os questionários podem ser de administração indireta, quando o próprio inquiridor o completa a partir das respostas que lhe foram fornecidas ou de administração direta, quando é o próprio inquirido que o preenche (Campenhoudt et al., 2019).

Escolhemos aplicar questionários de aplicação direta a todos os estudantes no final de cada aula com o *Scratch* de modo a podermos avaliar o nível de satisfação e aquisição de conhecimentos dos mesmos.

2.2. Caraterização do contexto de investigação

Numa investigação, além de se ter em conta os dados recolhidos “(...) é também preciso circunscrever o campo das análises empíricas no espaço, geográfico e social, e no tempo” (Campenhoudt et al., 2019). Posto isto é imprescindível uma caraterização do contexto do estudo.

A escola onde decorreu a investigação é a Escola Básica Francisco Torrinha (2.º e 3.º CEB), que pertence ao Agrupamento de Escolas Garcia de Orta, que é constituído por mais quatro unidade orgânicas, desde o Pré-escolar até ao Secundário, sendo que esta escola é a única do agrupamento que abrange o 2.º CEB.

Esta escola foi oficialmente criada em outubro de 1973, contudo só dois anos depois é que estava em funcionamento, sendo que à data se era o antigo Colégio Brotero, porém em junho de 2003 passou a integrar o Agrupamento até à data de hoje, passando o nome para Francisco Torrinha, um professor e escritor, que dedicou a sua vida à língua e a cultura portuguesa.

O Agrupamento abrange prioritariamente a população das freguesias de Aldoar, Nevogilde, Foz do Douro, Lordelo do Ouro e Ramalde, neste sentido e do ponto de vista habitacional, em todas as freguesias há zonas com residências de propriedade privada/cooperativa de grande qualidade, que coexistem, lado a lado, com zonas de empreendimentos camarário de baixo custo (Agrupamento de escolas Garcia e Orta, n.d.).

No Projeto Educativo é possível verificarmos que o Agrupamento se rege pelo lema “Escola Singular num Mundo Plural” o que significa que pretendem contribuir para a formação de cidadãos críticos e conscientes dos seus deveres e direitos, potenciando as competências do Perfil dos Alunos à Saída da Escolaridade Obrigatória.

O grupo social investigado foi uma turma de 20 estudantes do 6.º ano do ensino básico, esta turma é heterogénea, tendo estudantes de vários contextos sociais, havendo 10 estudantes do sexo masculino e 10 do sexo feminino, sendo que dois deles se enquadram dentro de medidas seletivas.

São na sua globalidade participativos e bastante curiosos. Interessam-se pelas diferentes áreas curriculares, incluindo as TIC e a matemática. Apresentam uma postura pautada pela educação e pelo respeito, quer entre os pares, quer para com as docentes. Não há registos de casos de absentismo escolar e, por norma, cumprem as tarefas que lhes são solicitadas, tanto na escola, como em casa.

O projeto colocado em prática decorreu no 3.º período do ano letivo 2020/2021, em três aulas de matemática, que ocorreram na sala de informática, tendo sido os conteúdos programáticos álgebra, a expressão geradora de uma sequência numérica, e isometrias, reflexão central e axial, sendo que a aula de álgebra foi uma aula de introdução à temática, enquanto que as aulas de isometria já foram de consolidação.

A primeira aula iniciou com uma exploração do ambiente de trabalho do *Scratch*, sendo que tinha sido entregue um guia de iniciação (Atividade no Apêndice) com as explicações essenciais, estando assim os estudantes a acompanhar a exploração, cada um num computador. De seguida, a professora demonstrou o resultado final do projeto que iriam realizar, sobre expressão geradora de uma sequência numérica, uma pequena história onde as personagens contam qual é o seu número preferido, e fez uma breve explicação dos conceitos matemáticos.

Posteriormente foi entregue os estudantes uma folha de exploração com o código de cada personagem (Atividade no Apêndice), a partir desse momento os estudantes, de forma autónoma criaram a expressão geradora da sequência numérica que iriam criar no *Scratch* e prosseguiram com o código, foi lhes dada a ressalva que tinham a liberdade de escolherem eles o cenário e as personagens.

Quer a professora estagiária, quer a professora cooperante foram auxiliando e retirando dúvidas à medida que estas surgiam, os estudantes que criaram o projeto mais rapidamente também foram ajudar os colegas na criação dos seus. No final, quando todos os estudantes tinham terminado o seu projeto e enviado o ficheiro por e-mail à professora estagiária, preencheram o questionário de satisfação e aquisição de conhecimentos da aula (Atividade no Apêndice).

A segunda aula, consistiu na reflexão central, um conteúdo previamente ensinado pelo método habitual da professora cooperante, pelo que o objetivo desta aula era consolidar os conhecimentos aprendidos, ao criarem uma rosácea através da reflexão central. A aula iniciou com a apresentação, da professora estagiária, do resultado final do projeto a ser criado pelos estudantes, assim como do código feito, sendo que os estudantes estavam a acompanhar com as suas folhas de exploração (Atividade 2 no Apêndice).

Após a apresentação os estudantes começaram a criar os seus projetos, foi lhes dada a liberdade de usarem as cores que preferissem e o número de reflexões também. Assim como na aula anterior a professora estagiária e a professora cooperante foram

esclarecendo as dúvidas que emergiam, e os estudantes que concluíram o projeto mais rapidamente também ajudaram os colegas. Quando todos tinham concluído com sucesso preencheram o questionário de satisfação e aquisição de conhecimentos da aula (Atividade 2 no Apêndice) e enviaram os seus projetos à professora estagiária.

A terceira e última aula abordou o conteúdo reflexão axial, que já tinha sido ensinado aos estudantes pela professora estagiária, pelo que agora iriam consolidar o aprendido, ao construírem um papagaio por meio de reflexão axial. Esta aula iniciou com a visionamento do resultado final do projeto, tal como o respetivo código, os estudantes acompanhar a explicação do código com as folhas de exploração (Atividade 3 no Apêndice), foi lhes também demonstrado onde estava o eixo de simetria na área de desenho nos trajes.

Assim como nas aulas anteriores, os estudantes tiveram a liberdade de escolher o ator, o cenário e as cores do papagaio que iriam desenhar. Mais uma vez ambas as professoras estiveram disponíveis para quaisquer dúvidas e os estudantes que terminaram primeiro foram ajudar os colegas. A aula terminou com o preenchimento do questionário de satisfação e aquisição de conhecimentos da aula (Atividade 3 no Apêndice).

Capítulo 3 – Análise dos dados

Campenhoudt et al., (2019) caracterizaram a investigação em ciências sociais da seguinte forma:

A investigação em ciências sociais segue um procedimento análogo ao do pesquisador de petróleo. Não é perfurando ao acaso que se encontrará o que procura. Pelo contrário, o êxito de um programa (...) depende do procedimento seguido. (...) Importa, acima de tudo, que o investigador seja capaz de conceber e de pôr em prática um dispositivo para a elucidação do real (...).

Os gráficos que se seguem são referentes aos questionários de avaliação da satisfação e aquisição de conhecimento por parte dos estudantes nas aulas com *Scratch*, pois importa perceber se para eles a utilização do *Scratch* influencia positivamente os seus processos de ensino-aprendizagem.

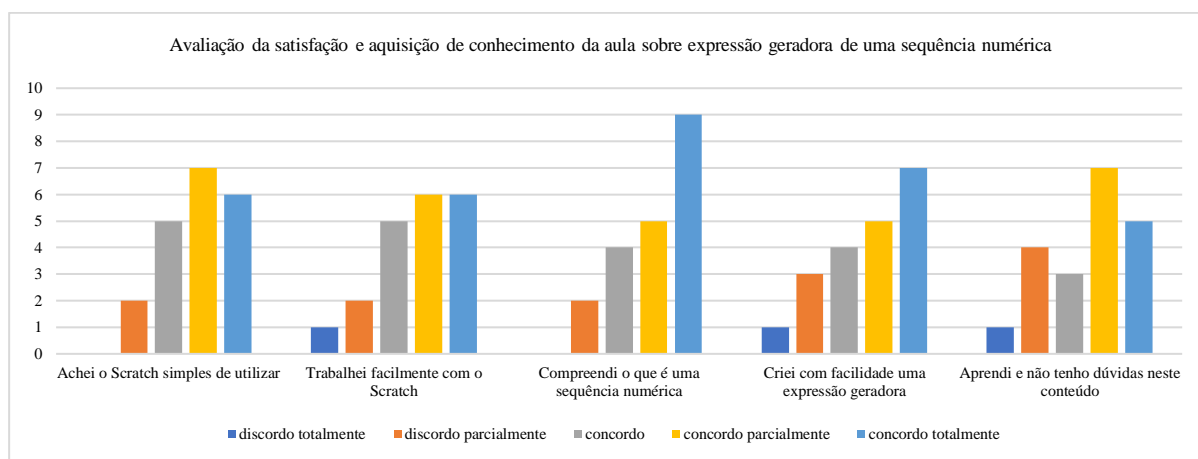


Gráfico 1 – Avaliação da satisfação e aquisição de conhecimento da 1.ª aula

Na primeira aula é perceptível que houve estudantes que não estiveram à vontade a trabalhar com o *Scratch*, apesar de terem tido previamente uma explicação do ambiente de trabalho do mesmo, mas foram uma minoria, é relevante ressaltar também que um quinto da turma já tinha, em casa, criado projetos no *Scratch*. De acordo com a professora cooperante (Entrevista no Apêndice), os estudantes têm mais facilidade a trabalhar com este *software* do que a própria, que sente que precisaria de mais treino.

Relativamente ao conteúdo lecionado na aula, expressão geradora de uma sequência numérica, é notório que não foi simples para os estudantes terem de ser os próprios a criarem a sua expressão, contudo a grande maioria conseguiu compreender o papel de um expressão geradora. Por esse motivo a professora cooperante é da opinião que este tipo de metodologia é de aplicar apenas como consolidação (Entrevista no Apêndice). Tendo sido esta uma primeira

aula de abordagem ao conteúdo é compreensível que os estudantes tenham sentido que não estavam seguros o suficiente e que necessitassem de o estudar melhor.

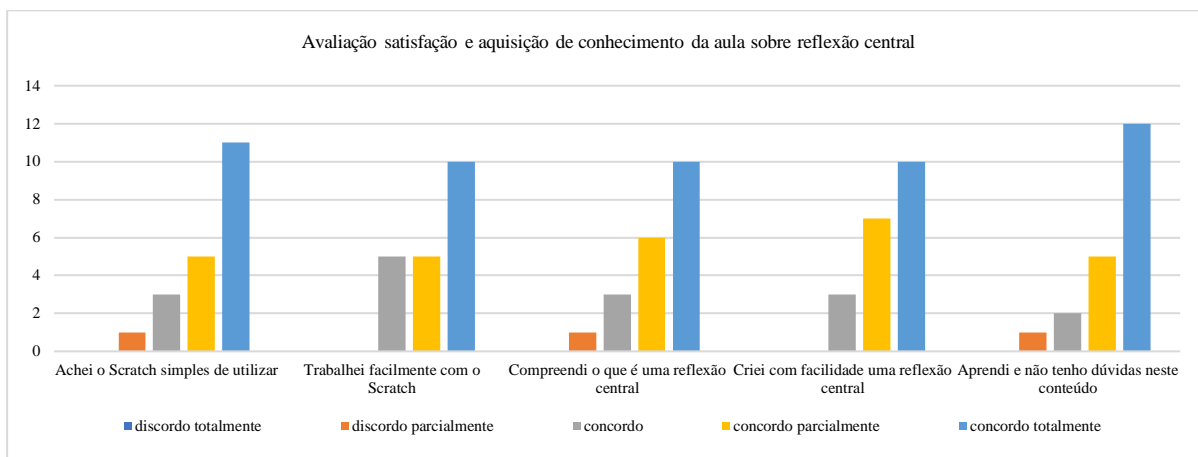


Gráfico 2 – Avaliação da satisfação e aquisição de conhecimento da 2.ª aula

Na segunda aula é notória a evolução dos estudantes a trabalharem com o *Scratch*, comparativamente com a primeira aula. Esta aula, contrariamente à anterior, era de consolidação e não de introdução ao conteúdo, pelo que também é visível uma melhor compreensão sobre o mesmo. A maior dificuldade sentida pelos estudantes foi no desenho da rosácea com o rato, mas mesmo assim obtiveram o resultado pretendido.

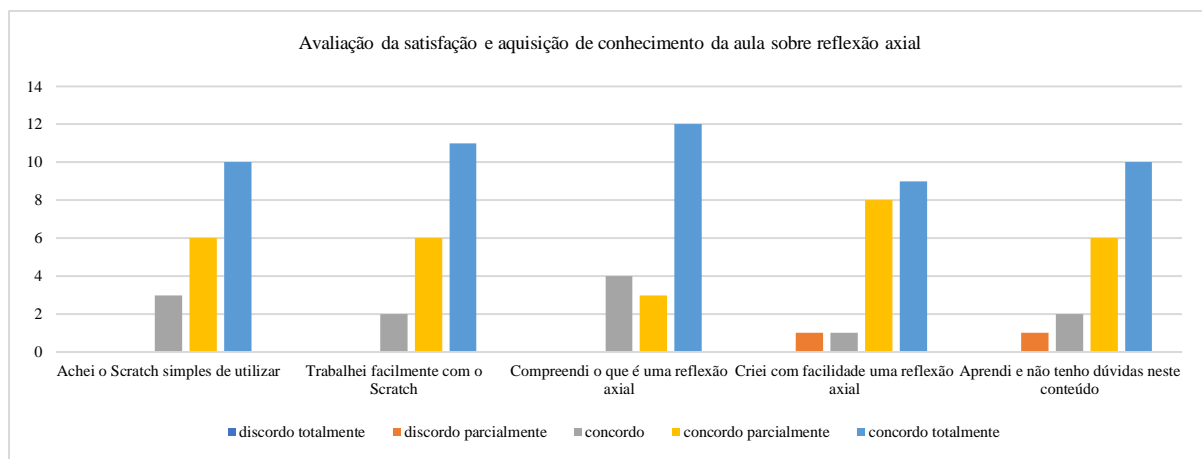


Gráfico 3 – Avaliação da satisfação e aquisição de conhecimento da 3.ª aula

Na última e terceira aula é mais uma vez perceptível a evolução da turma ao trabalharem cada vez mais facilmente com a plataforma. Relativamente ao conteúdo, e apesar de aula ter sido igualmente de consolidação, houve mais dificuldades, pois alguns estudantes não perceberam onde estava o eixo de simetria, pelo que no final o papagaio não surgiu, após lhes ter sido explicado como viam o eixo conseguiram corrigir e obter o papagaio.

Na opinião da professora cooperante (Entrevista no Apêndice) esta foi a aula onde os estudantes tiveram melhor desempenho, pois já sabiam a teoria e já a tinham colocado em prática, este era mais um exercício, mas que lhes dava uma motivação extra.



Figura 0-1 Exemplo de um projeto da 1.ª aula

No decorrer das aulas foi possível observar o empenho dos estudantes e a criatividade deles nos seus projetos, como este exemplo:

Os estudantes colaboravam entre si algo notório para ambas as professoras (Entrevista no Apêndice), pediam ajuda primeiro ao colega no lado e só se a dúvida persistisse é que chamavam a professora, os estudantes que terminavam mais rapidamente iam ajudar os outros, o que demonstra o espírito de cooperação e autonomia fomentada através deste projeto.

O enorme desejo de quererem continuar a ter mais aulas com o *Scratch* era perceptível mesmo nas outras aulas porque passavam a semana a colocar a mesma questão: “Quando vamos outra vez para a sala de TIC trabalhar com o *Scratch*?” apesar de já saberem a resposta, tinham esperança de que esse dia viesse mais cedo e quando o dia finalmente chegava eles vinham sempre mais motivados.

Capítulo 5 – Conclusão

“A capacidade de programar oferece benefícios importantes. Por exemplo, expande o alcance do que é possível criar (de como nos podemos expressar) com o computador. Também expande o alcance do que podemos aprender. Em particular, a programação oferece suporte ao pensamento computacional, ajudando a aprender estratégias importantes de resolução de problemas e *design* que são transferidas para domínios de não programação. Uma vez que a programação envolve a criação de representações externas de processos de solução de problemas, a programação oferece oportunidades para refletir sobre o nosso pensamento, até mesmo pensar sobre o próprio pensamento.” (Resnick et al., 2009)

A profissão de professor está sujeita a muitos desafios e responsabilidades, no entanto é uma profissão aliciante, pois possibilita contribuir para o crescimento e desenvolvimento de seres humanos. Para que o professor possa ser um bom profissional precisa, entre outras coisas, de estar permanentemente atualizado, de modo a estar preparado para enfrentar os desafios que são colocados pela escola e pela sociedade, que se encontra em constante mutação.

No séc. XXI os dispositivos tecnológicos de comunicação e informação, bem como os *media* de comunicação de massas habitam os mundos de vida e de experiência das crianças, estes dispositivos devem por isso fazer também parte da escola e do estudo.

De modo a unirmos a tecnologia à matemática, decidimos utilizar o *Scratch*, para tal planificamos três aulas em que interligássemos os conteúdos matemáticos a ser lecionados no 3.º período do 6.º ano com o pensamento computacional e a programação em *Scratch*.

Como o processo de pensamento computacional está relacionado com formulação de problemas e com as suas várias resoluções existentes, possíveis de ser solucionadas por um humano com eficácia e de forma otimizada, é este o pensamento associado ao desenvolvimento e utilização de qualquer tecnologia de computação – pensada por um ser humano, conhecendo muito bem o poder da automação (Wing, 2012).

De acordo com a revisão bibliográfica realizada, constatámos que todos os elementos do *Scratch* foram pensados de forma estratégica para que se atribuam tarefas orientadas com vista à resolução de problemas, esta abordagem construcionista procurou, e encontrou, dar bases para que os estudantes possam aprender a aprender, porque o tipo de conhecimento que as crianças necessitam é aquele que as ajudará a adquirir mais conhecimento (Sousa & Lencastre, 2014).

O projeto aplicado foi centrado numa turma, envolvendo os estudantes na aplicação de conceitos matemáticos e na resolução de problemas enquanto programam em *Scratch*. As bases

que foram dadas aos estudantes, para que estes pudessem aprender a aprender, foi uma breve exploração acompanhada do ambiente do *Scratch* e para cada atividade foi entregue uma folha com o código base que deveriam realizar, tendo-lhes sido dada a liberdade de o poderem alterar e de se expressarem enquanto seres criativos, através da programação.

Com a análise dos dados, percebemos que o desafio da criação de jogos no *Scratch* entusiasmou, motivou e envolveu todos os sujeitos de forma ativa no trabalho de conceitos matemáticos, com diferentes graus de envolvimento. Os estudantes demonstraram-se mais autónomos, responsáveis e auto motivados para solucionar os problemas que iam surgindo. Tal como aconteceu no exemplo dado por Resnick numa palestra TED (2012), apesar de os estudantes sentirem dificuldades, continuam motivados e querem aprender a ultrapassar a dúvida que surgiu.

Através dos dados que foram recolhidos ao longo da investigação, as observações, os questionários e a entrevista, foi possível concluir que o *Scratch* promove o pensamento computacional, a criatividade e cooperação e que de facto influencia positivamente o processo de ensino-aprendizagem, motivando os estudantes para o mesmo.

Posto isto fica uma satisfação por ter sido possível impactar estes estudantes com as aulas de matemática no *Scratch*, mas também a questão: Será que o mesmo acontecerá com outras disciplinas? Uma pergunta que fica presente, talvez para outra investigação.

Referências bibliográficas

- Afonso, N. (2014). *Investigação Naturalista em Educação*. Fundação Manuel Leitão.
- Agrupamento de escolas Garcia e Orta. (n.d.). *Projeto Educativo*.
- Alves, A. P., Ferreira, C., Ribeiro, R., Machado, S., & Barbosa, S. (2015). *LABORATÓRIOS DE APRENDIZAGEM: cenários e histórias de aprendizagem*.
https://erte.dge.mec.pt/sites/default/files/Projetos/Laboratorios_aprendizagem/magazine_la_final.pdf
- Amado, J. (2014). *Manual de investigação qualitativa em educação* (2.^a Edição). Imprensa da Universidade de Coimbra. <https://doi.org/http://dx.doi.org/10.14195/978-989-26-0879-2>
- Amélia, A., & Carvalho, A. (2020). *Aplicações para dispositivos móveis e estratégias inovadoras na educação* (M. da Educação & D.-G. da Educação (eds.)). Ministério da Educação | DGE.
http://hdl.handle.net/10316/90484%0Ahttps://drive.google.com/open?id=1_86dn_Vh-t3pQG5El8hYJfTgnfreZ5-P&authuser=figueiredorodrigues%40arquidiocese-braga.pt&usp=drive_fs
- Andrade, M., Silva, C., & Oliveira, T. (2013). Desenvolvendo games e aprendendo matemática utilizando o Scratch. *SBGames'13*, 260–263.
http://www.sbgames.org/sbgames2013/proceedings/cultura/Culture-5_short.pdf
- Barcelos, T., Muñoz, R., Villarroel, R., & Silveira, I. (2015). Relações entre o Pensamento Computacional e a Matemática: uma Revisão Sistemática da Literatura. *Anais Dos Workshops Do IV Congresso Brasileiro de Informática Na Educação (CBIE 2015)*, 1369–1378. <https://doi.org/10.5753/cbie.wcbie.2015.1369>
- Bogdan, R. C., & Biklen, S. K. (1994). *Investigação qualitativa em educação* (1.^a Edição). Porto Editora.

- Camacho, R., & Fino, C. (2010). *Síntese Crítica ao livro de Seymour Papert “A Máquina das Crianças - Repensando a escola na era da informática.”*
- Campenhoudt, L. Van, Marquet, J., & Quivy, R. (2019). *Manual de investigação em ciências sociais* (5.^a Edição). Gradiva.
- Cardoso, J. (2013). *O Professor do Futuro* (1.^a Edição). Guerra e Paz, Editores, S.A.
- Computer Science Teachers Association, & International Society for Technology in Education. (2011). *Computational Thinking - Teacher resources. 2.*
- Conselho Nacional de Educação. (2017). *Aprendizagem, TIC e Redes Digitais* (Conselho Nacional de Educação (ed.)). Conselho Nacional de Educação. www.cnedu.pt
- Cruz, S., Bento, M., & Lencastre, J. A. (2020). Construção de jogos em Scratch sobre conceitos da Matemática. *Atas Do 5.º Encontro Sobre Jogos e Mobile Learning*, 600–607. <https://estudogeral.sib.uc.pt/jspui/handle/10316/25740>
- Dasgupta, S., Vazirani, U., & Papadimitriou, C. (2006). Algorithms. In *SpringerBriefs in Computer Science* (1.^a Edição). Mc Graw Hill.
- Direção Geral da Educação. (2016). *Iniciação à Programação no 1.º Ciclo do Ensino Básico - Linhas Orientadoras para a Robótica.*
- Escola Superior de Educação do Instituto Politécnico de Setúbal. (2015). Atas do XVII Simpósio Internacional de Informática Educativa. In M. Rodrigues, M. Nistal, & M. Figueiredo (Eds.), *Atas do XVII Simpósio Internacional de Informática Educativa.* <http://siie15.ese.ips.pt/>
- Esteves, A. J. (1986). A investigação-ação. In *Metodologia das Ciências Sociais* (210.^a Edic, pp. 251–278). Edições Afrontamento.
- Fortin, M.-F. (2003). *O processo de investigação: da concepção à realização* (3.^a Edição). Lusociência.

- Gomes, A., & Freitas, N. (2017, December). *Jornal das primeiras matemáticas. Potencialidades Da Utilização Do Scratch Na Aprendizagem Da Matemática: Uma Experiência No 4.º Ano de Escolaridade*, 5–13. https://repositorium.sdum.uminho.pt/bitstream/1822/52627/1/Freitas_Scratch_5_13%289_2017%29_low.pdf
- Gordinho, S. S. V. (2009). *Interfaces de comunicação e ludicidade na infância: brincadeiras na programação Scratch* [Universidade de Aveiro]. <http://hdl.handle.net/10773/1225>
- Guerra, I. (2007). *Fundamentos e Processos de Uma Sociologia de Ação - O planeamento em Ciências Sociais* (2.ª Edição). Príncipia Editora.
- Hill, M., & Hill, A. (2009). *Investigação por questionário* (2.ª Edição). Edições Sílado.
- Jesus, C., Vasconcelos, J. B. de, & Lima, R. (2016). *Scratch e Kodu Iniciação à programação no ensino básico* (1ª Edição). FCA - Editora de Informática, Lda.
- Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The scratch programming language and environment. *ACM Transactions on Computing Education*, 10(4). <https://doi.org/10.1145/1868358.1868363>
- Martins, A., & Eloy, A. (2019). *Educação Integral Por Meio Do Pensamento Computacional - Letramento em Programação: Relatos de Experiências e Artigos Científicos* (1.ª Edição). Editora Appris Ltda.
- Ministério da Educação. (2013). Programa e Metas Curriculares. Matemática. Ensino Básico. In *Ministério da Educação e da Ciência*. https://www.dge.mec.pt/sites/default/files/Basico/Metas/Matematica/programa_matematica_basico.pdf
- Ministério da Educação. (2017). Perfil Dos Alunos À Saída Perfil Dos Alunos. In *Editorial do Ministério da Educação e Ciência*.

http://www.dge.mec.pt/sites/default/files/Curriculo/Projeto_Autonomia_e_Flexibilidade/perfil_dos_alunos.pdf

Ministério da Educação. (2018). *Aprendizagens Essenciais: Matemática (6.º ano)*.

Oliveira, P., & Lopes, C. (n.d.). Scratch na infância: experiencição comunicacional, lúdica e criativa. *Tecnologias e Novas Mídias: Da Educação Às Práticas Culturais e de Consumo*, 162–178.

Pedro, A., Matos, J. F., Piedade, J., & Dorotea, N. (2017). *Probótica - Programação e Robótica no Ensino Básico - Linhas Orientadoras*.

Philips, P. (2009). Computational Thinking - a Problem-Solving Tool for every classroom. *Computer Science Teachers Association (CSTA) and Microsoft*.
<http://education.sdsc.edu/resources/CompThinking.pdf>

Psathas, G. (1973). *Phenomenological Sociology: Issues and Applications*.

Ramalho, R., & Ventura, A. (2017). O potencial do scratch no ensino – aprendizagem da geometria. *Revista de Estudios e Investigación En Psicología y Educación*, 13, 172–175.
<https://doi.org/https://doi.org/10.17979/reipe.2017.0.13.2666>

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009, November). Scratch: Programming for all. *Communications of the ACM*, 52(11), 60–67.
<https://doi.org/10.1145/1592761.1592779>

Resnick, M., & TED. (2012). *TEDxBeaconStreet*. Technology Entertainment Design.
https://www.ted.com/talks/mitch_resnick_let_s_teach_kids_to_code?language=pt

Ressurreição, R. P. L. da. (2012). A Consolidação De Conceitos De Programação Utilizando a Robótica Educativa. In *Universidade de Lisboa*. Universidade de Lisboa.

Ribeiro, L., Foss, L., & da Costa Cavalheiro, S. A. (2017). *Entendendo o Pensamento*

Computacional.

- Running, D. (2006). *Algoritmia e Programação. Fevereiro.*
<https://www.researchgate.net/publication/266220066>
- Rusk, N., Resnick, M., Maloney, J., & MIT Media Laboratory. (2006). *21st Century Learning Skills.* <https://ilk.media.mit.edu/papers/scratch-21st-century.pdf>
- Silva, A., & Pinto, J. M. (1986). *Metodologia das ciências sociais* (270.^a Edic). Edições Afrontamento.
- Simões, A. (1990). A Inestigação-Ação: Natureza e Validade. *Revista Portuguesa e Pedagogia*, pp.39-51.
- Sousa, R. M., & Lencastre, J. A. (2014). Scratch : uma opção válida para desenvolver o pensamento computacional e a competência de resolução de problemas. *2º Encontro Sobre Jogos e Mobile Learning*, 256–267.
<http://repositorium.sdum.uminho.pt/handle/1822/29944>
- The Scratch Foundation. (n.d.). *Scratch.* <https://scratch.mit.edu/>
- Vasconcelos, J. B. de. (2015). *Python - Algoritmia e Programação WEB.* FCA - Editora de Informática, Lda.
- Ventura, A., & Ramalho, R. (2018). O Scratch promotor do pensamento computacional na geometria do ensino básico. *Currículo e Formação de Educadores e Professores*, 232–242. <http://hdl.handle.net/20.500.11796/2728%0A>
- Vilelas, J. (2009). *Investigação: o processo de construção de conhecimento.* Edições Sílabo.
- Wing, J. M. (2012). Microsoft Research Asia Faculty Summit 2012. *Computational Thinking.*
<https://doi.org/10.1145/1227504.1227378>

Apêndice

Atividades de programação com *Scratch*

Atividade 1

Os números preferidos gerados por uma expressão

Enquadramento programático

Programa e Metas Curriculares do Ensino Básico	
Domínio:	Álgebra
Subdomínio:	Sequências
Descritores de desempenho:	Determinação de expressões geradoras de sequências definidas por uma lei de formação recorrente
Finalidades do Ensino da Matemática:	A estruturação do pensamento: a organização do pensamento, constituindo-se como uma gramática basilar do raciocínio hipotético-dedutivo

Aprendizagens Essenciais		Descritores do Perfil dos alunos
Tema:	Álgebra	Conhecedor/ sabedor/ culto/ informado (A, B, G, I, J)
Conteúdos de aprendizagem:	Sequências	Respeitador da diferença/ do outro (A, B, E, F, H)
Objetivos essenciais de aprendizagem-	Determinar uma lei de formação de uma sequência numérica e uma expressão algébrica que represente uma sequência numérica em que	Sistematizador/ organizador (A, B, C, I, J) Auto avaliador (transversal às áreas) Participativo/ colaborador (B, C, D, E, F)

conhecimentos, capacidades e atitudes:	a diferença entre termos consecutivos é constante	Responsável/ autónomo (C, D, E, F, G, I, J) Cuidador de si e do outro (B, E, F, G)
--	---	---

Perfil dos Alunos à Saída da Escolaridade Obrigatória










Linguagens e textos: utilizar de modo proficiente diferentes linguagens e símbolos associados à matemática

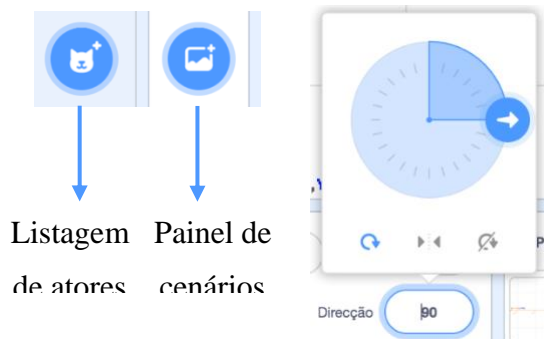
Desenvolvimento pessoal e autonomia: consolidar e aprofundar as competências que já possuem, numa perspetiva de aprendizagem ao longo da vida

Tempo	Percurso de aprendizagem	Recursos
50 minutos	Apresentação do software <i>Scratch</i> Visualização de um jogo sobre sequência numérica no <i>Scratch</i> Exploração e criação de um projeto no <i>Scratch</i> , usando uma expressão geradora de uma sequência numérica	Computador, projetor, <i>Scratch</i> , pen, 20 guias de iniciação e 20 folhas de exploração

Guia de iniciação

Instruções que permitem:

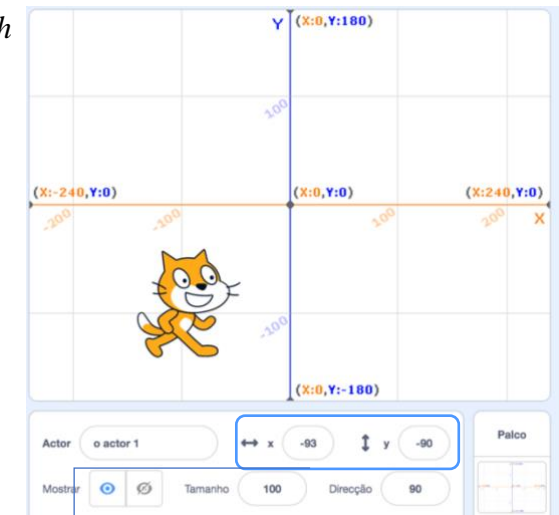
-  → Descolar os atores pela área Palco
-  → Alterar o aspeto dos atores, atribuir-lhes balões de fala, fazê-los aparecer e desaparecer
-  → Atribuir sons aos atores, assim como controlar o seu volume
-  → Determinar quando e como inicia o projeto
-  → Criar condições e repetições na execução do programa
-  → Em conjunto com os blocos Controlo controlar a execução do programa
-  → Em conjunto com os blocos Sensores, Controlo e Variáveis controlar a execução do programa
-  → Criar variáveis e listas
-  → Criar blocos personalizados, que funcionam como subprogramas dos projetos *Scratch*



Por predefinição os atores deslocam-se horizontalmente da esquerda para a direita, mas isto pode ser alterado

- 0** – direcionado para cima
- 90** – direcionado para a direita

Posicionam os atores no Palco através de um sistema de coordenadas que formam pares ordenados (x,y)



Folha de exploração

Quando alguém clicar em

- esconde-te
- espera 1 s
- mostra-te
- altera o teu tamanho para 60 %
- vai para a posição x: -207 y: 36
- diz O meu número preferido é o 3 durante 3 s

Quando alguém clicar em

- esconde-te
- espera 5 s
- mostra-te
- altera o teu tamanho para 60 %
- vai para a posição x: -66 y: 28
- diz O meu número preferido é o 9 durante 3 s

Quando alguém clicar em

- esconde-te
- espera 8 s
- mostra-te
- altera o teu tamanho para 60 %
- vai para a posição x: 69 y: 25
- diz O meu número preferido é o 27 durante 3 s

Quando alguém clicar em

- esconde-te
- espera 11 s
- mostra-te
- altera o teu tamanho para 45 %
- vai para a posição x: 121 y: 44
- diz O meu número preferido é o 81 durante 3 s

Quando alguém clicar em

- esconde-te
- espera 14 s
- mostra-te
- altera o teu tamanho para 100 %
- vai para a posição x: -126 y: -45
- diz A lei de formação deles é 3^n durante 3 s

Questionário de satisfação e aquisição de conhecimentos da aula

Nome:	N.º:	Data:			
	1	2	3	4	5
Achei o <i>Scratch</i> simples de utilizar					
Trabalhei facilmente com o <i>Scratch</i>					
Compreendi o que é um sequência numérica					
Criei com facilidade uma lei de formação					
Aprendi e não tenho dúvidas neste conteúdo					
<p>1 – discordo totalmente</p> <p>2 – discordo parcialmente</p> <p>3 – concordo</p> <p>4 – concordo parcialmente</p> <p>5 – concordo totalmente</p>					

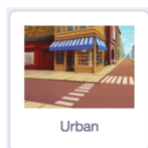
Passo a passo para o professor

Desafio – Criar um projeto em que várias personagens surjam a dizer o seu número preferido

Variante – Criar diferentes expressões geradoras para a sequência de números

Definir o cenário:

1. No painel de cenários clicar no botão “Escolher um cenário” e seleccionar “Exterior” e, de seguida, seleccionar o cenário “Urban”.



Remover ator:

1. Na área “Listagem de atores” clicar no ator “Sprite1” (gato) e, de seguida, remover clicando no caixote do lixo.



Acrescentar atores:

1. Na área “Listagem de atores” clicar no botão “Escolher atores”.
2. Na janela “Biblioteca de atores” seleccionar “Pessoas” e, de seguida, “Characters 1 ou 2”, repetir o processo cinco vezes, escolhendo o traje pretendido para cada personagem.
3. Alterar o nome de cada personagem para a sua expressão



Posicionar os atores no palco:

1. Seleccionar o ator “3^1”.
2. Alterar o valor de x para -207 e de y para 36.
3. Alterar o valor do tamanho para 60.



4. Seleccionar o ator “3^2”.
5. Alterar o valor de x para -66 e de y para 28.

- Alterar o valor do tamanho para 60.



- Selecionar o ator “3^3”.
- Alterar o valor de x para 69 e de y para 25.
- Alterar o valor do tamanho para 60.




- Selecionar o ator “3^4”.
- Alterar o valor de x para 121 e de y para 44.
- Alterar o valor do tamanho para 45.



- Selecionar o ator “3^n”.
- Alterar o valor de x para -126 e de y para -45.



Programar ator “3^1”

- Selecionar o ator “3^1”.
- Na “Paleta de códigos” selecionar “Eventos” e, de seguida, arrastar a instrução “Quando alguém clicar em ” para a área “Guiões”.
- Na “Paleta de códigos” selecionar “Aparência” e, de seguida, arrastar a instrução “esconde-te” para a área “Guiões”.
- Na “Paleta de códigos” selecionar “Controlo” e, de seguida, arrastar a instrução “espera 1 s” para a área “Guiões”, verifica que o valor 1 se mantém.
- Na “Paleta de códigos” selecionar “Aparência” e, de seguida, arrastar a instrução “mostra-te” para a área “Guiões”.
- Na “Paleta de códigos” selecionar “Aparência” e, de seguida, arrastar a instrução “altera o teu tamanho para 100%” para a área “Guiões”, altera o valor para 60%.

7. Na “Paleta de códigos” selecionar “Movimentos” e, de seguida, arrastar a instrução “vai para a posição x:-207 y:36” para a área “Guiões”, verifica que os valores se mantêm.
8. Na “Paleta de códigos” selecionar “Aparência” e, de seguida, arrastar a instrução “diz Olá! durante 2s” para a área “Guiões”, altera o texto para “O meu número preferido é o 3” e o valor para 3.

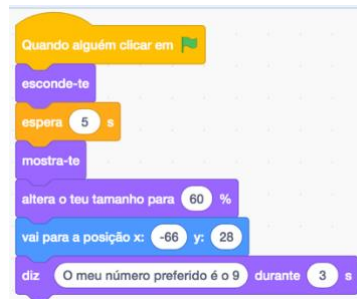


Programar ator “3^2”

1. Selecionar o ator “3^2”.
2. Copiar os blocos de código do ator “3^1”, arrastando até ao ícone do ator “3^2”



3. Alterar o valor de 1s para 5s.
4. Altera o valor da posição para x: -66 e y: 28.
5. Altera o texto para “O meu número preferido é o 9”.



Programar ator “3^3”

1. Selecionar o ator “3^3”.
2. Copiar os blocos de código do ator “3^1”, arrastando até ao ícone do ator “3^3”



3. Alterar o valor de 1s para 8s.
4. Altera o valor da posição para x: 69 e y: 25.
5. Altera o texto para “O meu número preferido é o 27”.



Programar ator “3^4”

1. Selecionar o ator “3^4”.
2. Copiar os blocos de código do ator “3^1”, arrastando até ao ícone do ator “3^4”



3. Alterar o valor de 1s para 11s.
4. Altera o valor do tamanho para 45%.
5. Altera o valor da posição para x: 121 e y: 44.
6. Altera o texto para “O meu número preferido é o 81”.



Programar ator “3^n”

1. Seleccionar o ator “3^n”.
2. Copiar os blocos de código do ator “3^1”, arrastando até ao ícone do ator “3^n”



3. Alterar o valor de 1s para 14s.
4. Altera o valor do tamanho para 100%.
5. Altera o valor da posição para x: -126 e y: -45.
6. Altera o texto para “A expressão geradora deles é 3^n”.



Verifica:

1. Clica em  e verifica se a sequência corre, se houver erros corrige.



Atividade 2

Rosácea por meio de reflexão central

Enquadramento programático

Programa e Metas Curriculares do Ensino Básico	
Domínio:	Geometria
Subdomínio:	Isometrias do plano
Descritores de desempenho:	Construção de imagens de figuras planas por reflexão central
Finalidades do Ensino da Matemática:	A estruturação do pensamento: a organização do pensamento, constituindo-se como uma gramática basilar do raciocínio hipotético-dedutivo

Aprendizagens Essenciais		Descritores do Perfil dos alunos
Tema:	Geometria	Conhecedor/ sabedor/ culto/ informado (A, B, G, I, J) Respeitador da diferença/ do outro (A, B, E, F, H) Sistematizador/ organizador (A, B, C, I, J) Auto avaliador (transversal às áreas) Participativo/ colaborador (B, C, D, E, F) Responsável/ autónomo (C, D, E, F, G, I, J)
Conteúdos de aprendizagem:	Figuras planas	
Objetivos essenciais de aprendizagem- conhecimentos, capacidades e atitudes:	Construir o transformado de uma dada figura através de isometrias	

		Cuidador de si e do outro (B, E, F, G)
--	--	--

Perfil dos Alunos à Saída da Escolaridade Obrigatória

Linguagens e textos: utilizar de modo proficiente diferentes linguagens e símbolos associados à matemática

Desenvolvimento pessoal e autonomia: consolidar e aprofundar as competências que já possuem, numa perspectiva de aprendizagem ao longo da vida

Tempo	Percurso de aprendizagem	Recursos
50 minutos	<p>Visualização de um projeto sobre reflexão central no <i>Scratch</i></p> <p>Criação de um projeto no <i>Scratch</i>, usando a reflexão central</p>	Computador, projetor, <i>Scratch</i> , pen, 20 folhas de exploração

Folha de exploração

The image displays the Scratch 3.18.1 interface. On the left, a character named 'Traje 2' is shown with a red and green wavy body. The right side shows the code editor with a script for a click event:

```
Quando alguém clicar em [bandeira]
  vai para a posição x: 0 y: 0
  apaga tudo do palco
  repete 36 vezes
    carimba-te
    gira 10 graus
  espera 0.5 segundos
```

Below the code editor, there is a search bar and a category filter for 'Escolher um Cenário' (Choose a Scenario). The categories are: Tudo, Fantasia, Música, Desporto, Exterior, Interior, Espaço, Debaxo de água, and Padrões. The 'Padrões' category is selected, showing five pattern options: Circles, Hearts, Light, Rays, and Stripes.

Questionário de satisfação e aquisição de conhecimentos da aula

Nome:	N.º:	Data:			
	1	2	3	4	5
Achei o <i>Scratch</i> simples de utilizar					
Trabalhei facilmente com o <i>Scratch</i>					
Compreendi o que é um reflexão central					
Criei com facilidade uma reflexão central					
Aprendi e não tenho dúvidas neste conteúdo					
<p>1 – discordo totalmente</p> <p>2 – discordo parcialmente</p> <p>3 – concordo</p> <p>4 – concordo parcialmente</p> <p>5 – concordo totalmente</p>					

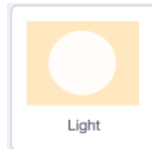
Passo a passo para o professor

Desafio – Criar um projeto em que um desenho se transforme numa rosácea, por meio de reflexão central

Variante – Cria desenhos diferentes para obteres diferentes rosáceas, utilizando cores e espessuras distintas

Definir o cenário:

2. No painel de cenários clicar no botão “Escolher um cenário” e seleccionar “Padrões” e, de seguida, seleccionar o cenário “Light”.



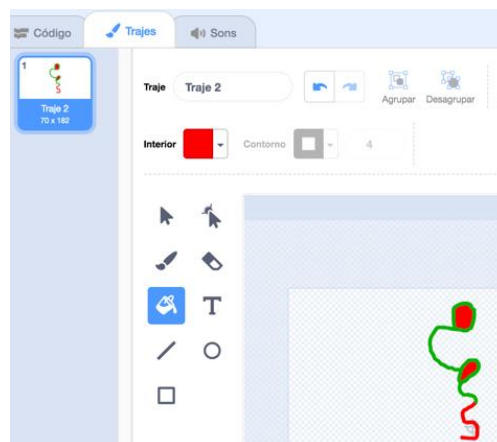
Remover ator:

2. Na área “Listagem de atores” clicar no ator “Sprite1” (gato) e, de seguida, remover clicando no caixote do lixo.



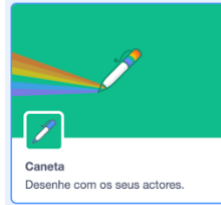
Acrescentar atores:

1. Na área “Listagem de atores” clicar no botão “Pintar” e, de seguida seleccionar “Pincel”, alterar a cor para matriz 30, saturação 100 e brilho 80 (cor verde)/ matriz 100, saturação 100 e brilho 100 (cor vermelha) e a espessura para 10, desenha algo semelhante ao exemplo, para preencheres clicas no botão “Preencher”.
2. Nomeia o ator de “Rosácea”.



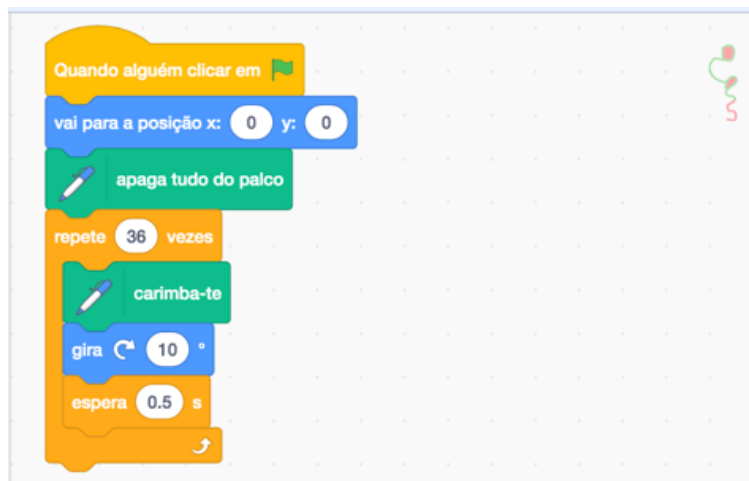
Acrescentar extensão:

1. Na “Paleta de códigos” clicar no botão “Extensões”.
2. Na janela “Extensões” selecionar caneta.




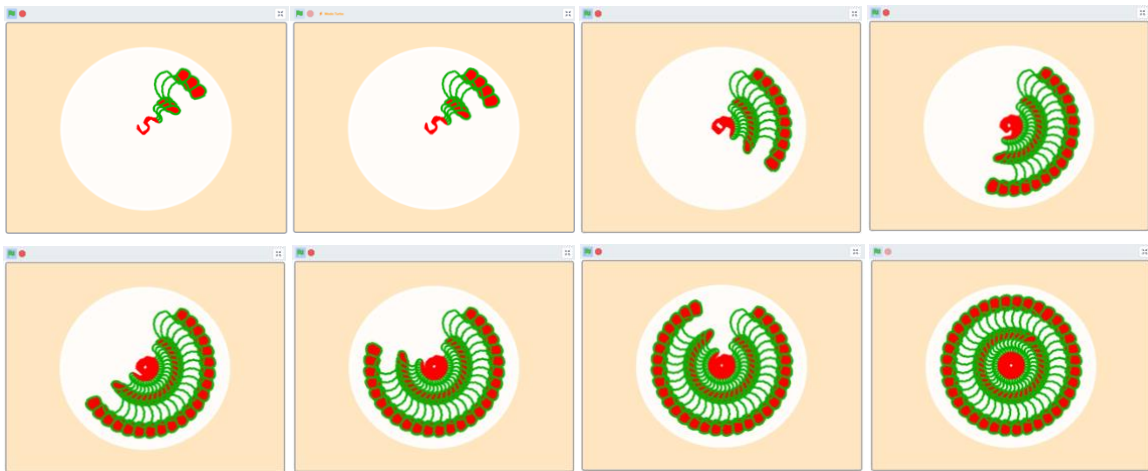
Programar ator “Rosácea”:

1. Selecionar o ator “Rosácea”.
2. Na “Paleta de códigos” selecionar “Eventos” e, de seguida, arrastar a instrução “Quando alguém clicar em []” para a área “Guiões”.
3. Na “Paleta de códigos” selecionar “Movimentos” e, de seguida, arrastar a instrução “Vai para a posição x: 0 y: 0 para a área “Guiões”, verifica que os valor de x e y são estes.
4. Na “Paleta de códigos” selecionar “Caneta” e, de seguida, arrastar a instrução “Apaga tudo do palco” para a área “Guiões”.
5. Na “Paleta de códigos” selecionar “Controlo” e, de seguida, arrastar a instrução “Repete 10 vezes” para a área “Guiões”, altera o valor para 36.
6. Na “Paleta de códigos” selecionar “Caneta” e, de seguida, arrastar a instrução “Carimba-te” para a área “Repete 36 vezes”.
7. Na “Paleta de códigos” selecionar “Movimento” e, de seguida, arrastar a instrução “Gira 15º” para a área “Repete 36 vezes”, altera o valor para 10º.
8. Na “Paleta de códigos” selecionar “Controlo” e, de seguida, arrastar a instrução “Espera 1 s” para a área “Repete 36 vezes”, altera o valor para 0,5.



Verifica:

2. Clica em  e verifica se a sequência corre, se houver erros corrige.



Atividade 3

Papagaio de papel por meio de reflexão axial

Enquadramento programático

Programa e Metas Curriculares do Ensino Básico	
Domínio:	Geometria
Subdomínio:	Isometrias do plano
Descritores de desempenho:	Construção de imagens de figuras planas por reflexão axial
Finalidades do Ensino da Matemática:	A estruturação do pensamento: a organização do pensamento, constituindo-se como uma gramática basilar do raciocínio hipotético-dedutivo

Aprendizagens Essenciais		Descritores do Perfil dos alunos
Tema:	Geometria	Conhecedor/ sabedor/ culto/ informado (A, B, G, I, J)
Conteúdos de aprendizagem:	Figuras planas	Respeitador da diferença/ do outro (A, B, E, F, H)
Objetivos essenciais de aprendizagem- conhecimentos, capacidades e atitudes:	Construir o transformado de uma dada figura através de isometrias	Sistematizador/ organizador (A, B, C, I, J) Auto avaliador (transversal às áreas) Participativo/ colaborador (B, C, D, E, F) Responsável/ autónomo (C, D, E, F, G, I, J) Cuidador de si e do outro (B, E, F, G)

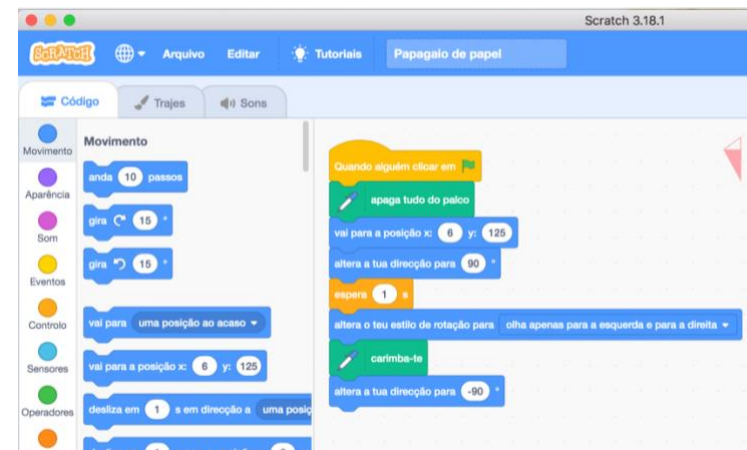
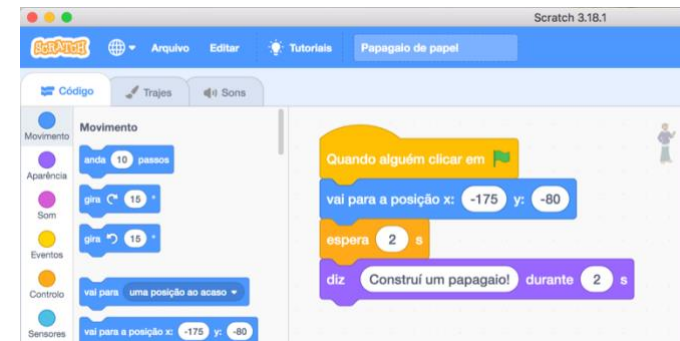
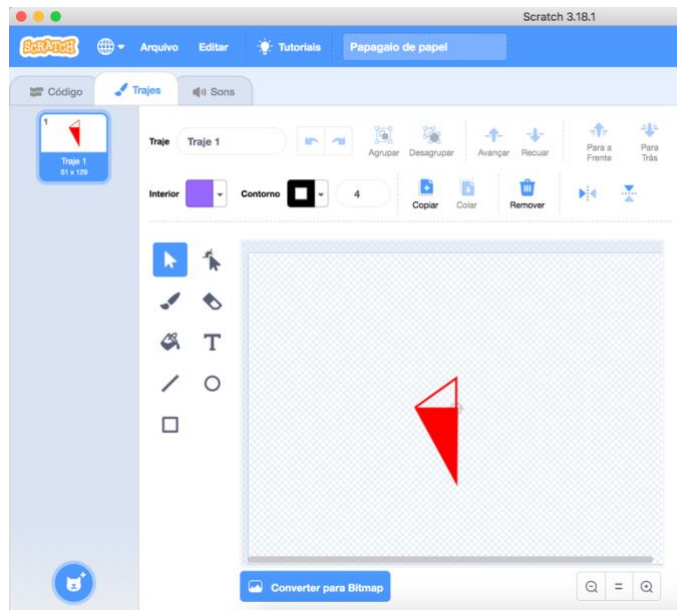
Perfil dos Alunos à Saída da Escolaridade Obrigatória

Linguagens e textos: utilizar de modo proficiente diferentes linguagens e símbolos associados à matemática

Desenvolvimento pessoal e autonomia: consolidar e aprofundar as competências que já possuem, numa perspectiva de aprendizagem ao longo da vida

Tempo	Percurso de aprendizagem	Recursos
50 minutos	Visualização de um projeto sobre reflexão axial no <i>Scratch</i> Criação de um projeto no <i>Scratch</i> , usando a reflexão axial	Computador, projetor, <i>Scratch</i> , pen, 10 folhas de exploração

Folha de exploração



Questionário de satisfação e aquisição de conhecimentos da aula

Nome:	N.º:	Data:			
	1	2	3	4	5
Achei o <i>Scratch</i> simples de utilizar					
Trabalhei facilmente com o <i>Scratch</i>					
Compreendi o que é um reflexão axial					
Criei com facilidade uma reflexão axial					
Aprendi e não tenho dúvidas neste conteúdo					
1 – discordo totalmente 2 – discordo parcialmente 3 – concordo 4 – concordo parcialmente 5 – concordo totalmente					

Passo a passo para o professor

Desafio – Criar um projeto em que uma personagem tenha um papagaio desenhado por meio de reflexão axial

Variante: Altera a figura, para obteres imagens diferentes

Definir o cenário:

1. No painel de cenários clicar no botão “Escolher um cenário” e selecionar “Exterior” e, de seguida, selecionar o cenário “Blue Sky”.



Remover ator:

1. Na área “Listagem de atores” clicar no ator “Sprite1” (gato) e, de seguida, remover clicando no caixote do lixo.

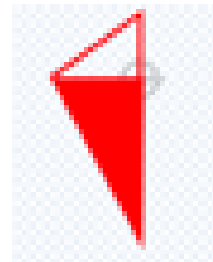
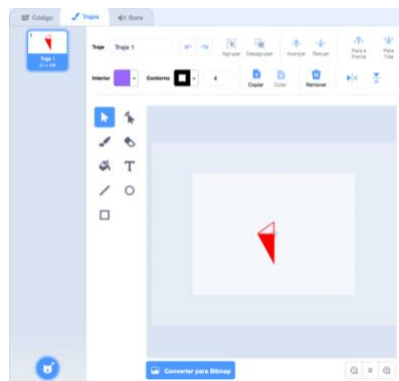


Acrescentar atores:

1. Na área “Listagem de atores” clicar no botão “Escolher atores”.
2. Na janela “Biblioteca de atores” selecionar “Pessoas” e, de seguida, “Devin”.



3. Na área “Listagem de atores” clicar no botão “Pintar” e, de seguida selecionar “Linha”, alterar a cor para 100 na matriz, saturação e brilho e a espessura para 4, desenhar metade do papagaio, junto ao eixo de simetria, clicar no botão “Preencher” para preencher o desenho, nomear o desenho de “Papagaio”.



Posicionar os atores no palco:

1. Selecionar o ator “Devin”.
2. Alterar o valor de x para -175 e de y para -80.

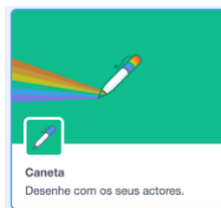


3. Selecionar o ator “Papagaio”.
4. Alterar o valor de x para 6 e de y para 125.
5. Alterar a direção para 90°.



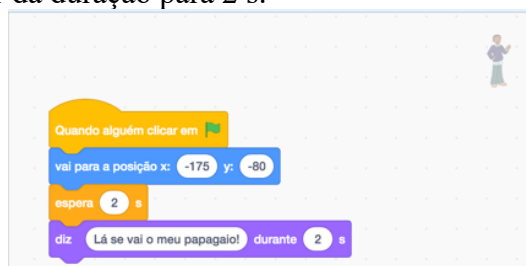
Acrescentar extensão:

1. Na “Paleta de códigos” clicar no botão “Extensões”.
2. Na janela “Extensões” selecionar caneta.




Programar ator “Devin”:

1. Selecionar o ator “Devin”.
2. Na “Paleta de códigos” selecionar “Eventos” e, de seguida, arrastar a instrução “Quando alguém clicar em []” para a área “Guiões”.
3. Na “Paleta de códigos” selecionar “Movimentos” e, de seguida, arrastar a instrução “Vai para a posição x: -175 y: -80” para a área “Guiões”, verifica que os valor de x e y são estes.
4. Na “Paleta de códigos” selecionar “Aparência” e, de seguida, arrastar a instrução “Diz Olá durante 1 s” para a área “Guiões”, altera a frase para “Lá se vai o meu papagaio” e o valor da duração para 2 s.




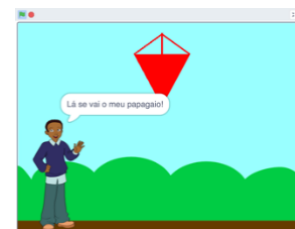
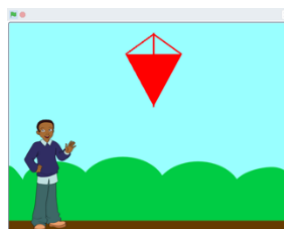
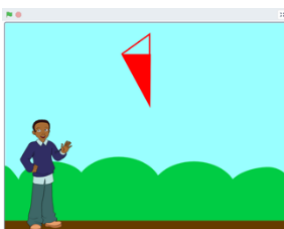
Programar ator “Papagaio”:

1. Na “Paleta de códigos” selecionar “Eventos” e, de seguida, arrastar a instrução “Quando alguém clicar em ” para a área “Guiões”.
2. Na “Paleta de códigos” selecionar “Caneta” e, de seguida, arrastar a instrução “Apagar tudo do palco” para a área “Guiões”.
3. Na “Paleta de códigos” selecionar “Movimento” e, de seguida, arrastar a instrução “Vai para a posição x: 6 y: 125” para a área “Guiões”, verifica que os valor de x e y são estes.
4. Na “Paleta de códigos” selecionar “Movimento” e, de seguida, arrastar a instrução “Altera a tua direção para 90°” para a área “Guiões”, verifica que é este o valor da direção.
5. Na “Paleta de códigos” selecionar “Controlo” e, de seguida, arrastar a instrução “Espera 1 s” para a área “Guiões”.
6. Na “Paleta de códigos” selecionar “Movimento” e, de seguida, arrastar a instrução “Altera o teu estilo de rotação para olha apenas para a esquerda e para a direita” para a área “Guiões”.
7. Na “Paleta de códigos” selecionar “Caneta” e, de seguida, arrastar a instrução “Carimba-te” para a área “Guiões”.
8. Na “Paleta de códigos” selecionar “Movimento” e, de seguida, arrastar a instrução “Altera a tua direção para -90°” para a área “Guiões”, verifica que é este o valor da direção.



Verifica:

1. Clica em  e verifica se a sequência corre, se houver erros corrige.



Entrevista

Entrevista conduzida pela professora estagiária (PE) à professora cooperante (PC).

PE: Considera o *Scratch* um *software* simples de utilizar?

PC: Relativamente simples, não é direto, tem que se saber explicar. Os estudantes tem que ter uma explicação primeiro de como é que funciona, após essa explicação eles conseguem trabalhar facilmente, já eu precisava de mais treino, de mais tempo.

PE: Acha que o uso do *Scratch* para iniciar o conteúdo das sequências numéricas foi enriquecedor?

PC: No princípio de um conteúdo acho que não se deve aplicar, deve se aplicar mais no fim, quando eles já aprenderam o conteúdo e aplicaram-no em exercícios.

PE: Acredita que os dois últimos projetos criados ajudaram a consolidar os conhecimentos sobre reflexão axial e central?

PC: Resultou melhor, pois já sabiam o que era na teoria e na prática e depois foi só aplicar mais uma vez. Então o último que se fez eles já perceberam muito melhor.

PE: Crê que estes três projetos motivaram mais os estudantes para a aprendizagem?

PC: Sim, sim, isso sim, estes trabalhos motivam sempre muito mais os estudantes.

PE: Julga que com este recurso se ensinou competências além dos conteúdos da matemática?

PC: Sim, acho que sim, então se se fizer pares eles tem que interagir entre eles e pedir ajuda um ao outro. Até alguns, mesmo não estando a trabalhar a pares, pediam ajuda e iam ajudar os outros, os que acabaram primeiro foram ajudar os restantes.

PE: É da opinião que no ensino da matemática se deve utilizar mais recursos inovadores e tecnologias?

PC: Sim, sou a favor disso.

PE: Pondera utilizar este *software*?

PC: Este não sei, mas outros como o *Quizizz*, o *Kahoot*, o *Plickers*, utilizaria, por serem mais simples, de eu utilizar, e os estudantes vem logo o resultado se está a dar certo ou se está a dar errado, e dá para fazer perguntas mais rápidas, enquanto que este demora mais um bocadinho a obter-se o resultado e não é para qualquer conteúdo, enquanto que os outros podem ser.