UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO

DEPARTAMENTO DE MATEMÁTICA E ESTATÍSTICA

GUILHERME CARVALHO RODRIGUES DA SILVEIRA

O Scratch como ferramenta no ensino de matemática

Rio de Janeiro 2014/2

GUILHERME CARVALHO RODRIGUES DA SILVEIRA

O Scratch como ferramenta no ensino de matemática

Trabalho apresentado junto ao curso de Licenciatura em Matemática da Universidade Federal do Estado do Rio de Janeiro, como requisito parcial para a obtenção do título de Licenciado em Matemática.

Orientador: Aline Caetano da Silva Bernardes Coorientador: Leonardo Tadeu Silvares Martins

Rio de Janeiro 2014/2 Nome: SILVEIRA, Guilherme Carvalho Rodrigues da Título: O Scratch como ferramenta no ensino de matemática

> Trabalho apresentado junto ao curso de Licenciatura em Matemática da Universidade Federal do Estado do Rio de Janeiro, como requisito parcial para a obtenção do título de Licenciado em Matemática.

Aprovado e	em:
------------	-----

Banca examinadora:

Prof. Dr.	Instituição:	
Julgamento:	Assinatura:	

Prof. Dr	Instituição:	
Julgamento:	Assinatura:	

Prof. Dr	_Instituição:
Julgamento:	Assinatura:

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

RESUMO

Existem diversas formas de ensinar conteúdos matemáticos, uma delas é a através da utilização de ferramentas tecnológicas, que com o passar dos anos se tornam cada vez mais acessíveis. O Scratch é um aplicativo de programação visual desenvolvido com o intuito de ensinar programação a jovens e adultos, mas também pode ser utilizado como ferramenta didática na disciplina de matemática. Este trabalho, por meio de atividades, mostra algumas formas possíveis de utilizá-lo de forma que aborde novos conteúdos ou mostre processos que não poderiam ser feitos sem alguma ferramenta, enquanto introduz os comandos do Scratch e a lógica por trás da programação.

Palavras chave: Scratch, programação visual, ensino de matemática, ferramenta,

Logo

ABSTRACT

There are several ways to teach math concepts, one of which is through the use of technological tools, which over the years become increasingly accessible. Scratch is a visual programming application developed with the purpose of teaching programming for youth and adults, but can also be used as a teaching tool in the discipline of mathematics. This work, through activities, shows some possible ways of using it in a way that covers new content or processes that could not be done without some tool, while introduces commands from Scratch and the logic behind the programming.

Keywords: Scratch, visual programming, math teaching tool, Logo

Sumário

1	Introdução e apresentação do trabalho	7
1.1	Introdução	7
1.2	Políticas públicas	8
2	Apresentação do Scratch	10
2.1	Origem e filosofia	10
2.2	Apresentando a interface e os principais comandos	12
2.3	O site e a comunidade do scratch	20
2.4	Estatísticas	24
3	Revisão bibliográfica	26
3.1	Logo: A proposta de aprender matemática utilizando uma	
lingu	agem de programação	26
3.2	Trabalhos com Scratch	29
4	Perspectiva de ensino e Proposta de atividades	31
4.1	A metodologia da resolução de problemas	31
4.2	Atividades	33
4.2.1	Atividade polígonos regulares	34
4.2.2	Atividade de construção da circunferência	39
4.2.3	Atividades Aproximação de π partes I, II e III	42
4.2.4	Atividade para construção de calculadoras	57
5 Co	nclusão	59
Refe	rências	60
Apêr	ndice	62

1 Introdução e apresentação do trabalho

1.1 Introdução

Com o advento e a popularização de novas tecnologias de forma cada vez mais acelerada, torna-se necessário que as escolas se adaptem a essas novas ferramentas e competências. A escolha dos conteúdos a serem ministrados em sala de aula, assim como a forma como eles devem ser passados, seguem algumas regras e objetivos, com o intuito de que todos os alunos possam desenvolver competências básicas, a fim de prepará-los para as demandas da sociedade contemporânea e para o mercado de trabalho.

No ensino de matemática, ferramentas como computadores, celulares e calculadoras, que podem auxiliar a abordagem de alguns conteúdos, não estão sendo explorados da melhor forma, parte pela crença de que seu uso, por facilitar algumas tarefas, desestimula o raciocínio mental e parte pela falta de conhecimento do aluno de como utilizar a ferramenta e do professor em como utilizar de maneira correta na sala de aula.

A programação é um exemplo de habilidade pouco ou nada explorada nas escolas e possui um enorme potencial no ensino de matemática. Com ela podemos entender e explorar diversos conceitos, sendo fundamental para uma melhor compreensão de como as ferramentas citadas anteriormente funcionam e são desenvolvidas.

O livro Recursos Computacionais no Ensino da Matemática [1] mostra o potencial, no ensino de matemática, de recursos como ambientes de geometria dinâmica, ambientes graficos, planilhas eletrônicas, calculadora, etc. O Scratch, como ferramenta de ensino de matemática, abrange diversos dos recursos e possibilidades, destacados no livro, de cada uma destas ferramentas, como a articulação de diversas formas de representação e o fato de permitir a construção de objetos geométricos com propriedades ou relações estabelecidas. O também pode ser utilizado para criar calculadoras normais ou gráficas.

Existe também, um crescente desinteresse por parte do aluno em aprender matemática de forma tradicional, o que vem sendo contornado com a utilização de jogos e atividades mais lúdicas que almejam obter mais atenção do aluno.

A escola é, ainda, espaço em que se abrigam desencontros de expectativas, mas também acordos solidários, norteados por princípios e valores educativos pactuados por meio do projeto político pedagógico concebido segundo as demandas sociais e aprovado pela comunidade educativa. Por outro lado, enquanto a escola se prende às características de metodologias tradicionais, com relação ao ensino e à aprendizagem como ações concebidas separadamente, as características de seus estudantes requerem outros processos e procedimentos, em que aprender, ensinar, pesquisar, investigar, avaliar ocorrem de modo indissociável. Os estudantes, entre outras características, aprendem a receber informação com rapidez, gostam do processo paralelo, de realizar várias tarefas ao mesmo tempo, preferem fazer seus gráficos antes de ler o texto, enquanto os docentes creem que acompanham a era digital apenas porque digitam e imprimem textos, têm *e-mail*, não percebendo que os estudantes nasceram na era digital.[2]

Este trabalho busca auxiliar a compreensão de alguns conteúdos matemáticos, através de atividades que utilizam um aplicativo de programação visual intitulado *Scratch*, pois possibilita um processo de descoberta, onde, com a devida orientação, o aluno consegue estabelecer conjecturas. Ao mesmo tempo, conceitos de programação são introduzidos, permitindo que o aluno possa utilizar o programa para criar jogos, ferramentas, histórias etc., facilitando a compreensão de qualquer outra linguagem de programação.

1.2 Políticas públicas

Um dos grandes questionamentos sobre o uso de ferramentas digitais nas escolas se dá pela falta de laboratórios de informática que, quando presentes, dificilmente possuem computadores suficientes para uma turma inteira. E os poucos laboratórios existentes apresentam problemas como "idade" das máquinas, manutenção, profissionais treinados para utilizá-las, etc.

Algumas ações têm sido realizadas com o intuito de viabilizar a inclusão digital. O Ministério da Educação (MEC), em 1997, criou o programa Nacional de Tecnologia Educacional (ProInfo) para promover o uso pedagógico de Tecnologias de Informática e Comunicações (TIC) na rede pública de ensino fundamental e médio [3].

O MEC compra, distribui e instala laboratórios de informática nas escolas públicas de educação básica. Em contrapartida, os governos locais (prefeituras e governos estaduais) devem providenciar a infraestrutura das escolas, o que é indispensável para que elas recebam

os computadores [3]. Na época que o programa foi criado, o computador não tinha a presença que tem hoje na vida do brasileiro, mas foi um passo importante.

O programa Um Computador por Aluno (UCA) [4] passou por duas fases, pré-piloto e projeto piloto. O pré-piloto ocorreu durante o ano de 2007, onde se iniciaram experimentos do UCA em cinco escolas, visando avaliar o uso de equipamentos portáteis pelos alunos em sala de aula. A Secretaria de Educação a Distância do Ministério da Educação (SEED/MEC) buscou estados e municípios com interesse em aderir ao programa. Foram pré-selecionadas dez escolas e, destas, cinco foram escolhidas. Para a fase dos experimentos três fabricantes de equipamentos doaram ao Governo Federal três modelos de *laptop*.

O projeto piloto iniciou em 2010. Esta etapa abrangeu cerca de 300 escolas públicas pertencentes às redes de ensino estaduais e municipais, distribuídas em todas as unidades da federação e selecionadas mediante critérios acordados com o Conselho Nacional de Secretários Estaduais de Educação (Consed), a União Nacional dos Dirigentes Municipais de Educação (Undime), a Secretaria de Educação a Distância do Ministério da Educação (SEED/MEC) e a Presidência da República.

Um grande exemplo de que é possível que políticas públicas, como UCA ou outros programas com objetivos semelhantes, viabilizem a utilização do computador como ferramenta diária no ensino de matemática é o município de Piraí, no Estado do Rio de Janeiro, que em 2004 implementou o programa de inclusão social "Piraí Digital". Neste projeto, cada aluno e professor da rede municipal receberam um laptop e a população de toda a cidade passou a ter acesso gratuito à internet por meio de *wi-fi*. Além disso, o programa informatizou bibliotecas, modernizou a administração pública e promoveu a participação popular nos processos decisórios da Prefeitura. Piraí fez da inclusão digital uma política para todos, criando condições para que a população possa se beneficiar das novas tecnologias em termos de formação profissional, acesso à informação, entretenimento e cidadania [5].

Além da presença do computador na sala de aula se tornar pouco a pouco uma realidade, seu uso para a realização de avaliações também poderá se tornar uma realidade no futuro. O Programa Internacional de Avaliação de Alunos (em inglês: *Programme for International Student Assessment* – PISA), pretende, em sua edição de 2015, realizar suas provas de forma informatizada.

O objetivo do Pisa é produzir indicadores que contribuam para a discussão da qualidade da educação nos países participantes, de modo a subsidiar políticas de melhoria do ensino básico. A avaliação procura verificar até que ponto as escolas de cada país participante estão preparando seus jovens para exercer o papel de cidadãos na sociedade contemporânea. [6]

De acordo com a estrutura do projeto de matemática para o PISA 2015 [7], os computadores são comumente usados nos locais de trabalho e na vida cotidiana, assim, um nível de competência em matemática no século 21 deve incluir o uso de computadores que ofereçam ferramentas para, entre outras coisas, realizar: representação, visualização, modificação, exploração e experimentação de uma grande variedade de objetos matemáticos, fenômenos e processos.

2 Apresentação do Scratch

2.1 Origem e filosofia

O Scratch é uma ferramenta livre desenvolvida em 2004 pelo grupo *Lifelong Kindergarten* no laboratório de mídia do Instituto de Tecnologia de Massachusetts (MIT) que é liderado por Mitchel Resnick.

Ele é a mais recente de uma longa linhagem a qual se iniciou com a criação da linguagem de programação LISP, desenvolvida nos anos sessenta por John McCarthy, também no MIT. LISP tornou-se uma linguagem comum para a inteligência artificial (IA) de programação, em parte devido aos trabalhos de IA no MIT. Ela tem evoluído através de numerosos dialetos, como Scheme e Common LIPS [8].

Seymour Papert, que trabalhou com Piaget em Genebra, foi para os Estados Unidos, onde co-fundou o Laboratório de Inteligência Artificial do MIT com Marvin Minsky. Papert trabalhou com a equipe da BBN Technologies¹, liderado por Wallace Feurzeig, que criou a primeira versão do LOGO em 1967. A linguagem LOGO é uma simplificação da LISP para uso em educação, sendo utilizada para ensinar programação para crianças. LOGO possui uma sintaxe mais convencional que seu antecessor e contou com "gráficos de tartarugas"² que é um modo simples para gerar gráficos, método que utiliza instruções como "*left90*" e

¹ Empresa americana de tecnologia criada por professores do MIT.

²Esse nome surgiu pois um robô com um lápis acoplado, programado com LOGO, era utilizado inicialmente para desenhar e seu formato lembrava uma tartaruga.

"forward", que em português significam "esquerda90" e "para frente", para modificar a orientação e posição de um objeto. O nome gráfico de tartarugas vem de um projeto também realizado no MIT onde o objetivo era programar um robô que parecia uma tartaruga [9].

Chamado inicialmente de *Logo Blocks*, o Scratch surgiu como um sucessor do LOGO. Sua proposta é ser simples e intuitivo, utilizando uma linguagem visual onde "blocos" são encaixados como em um quebra cabeça para formar o algoritmo, que, por sua vez, animará um personagem. Ele se tornou popular rapidamente e a programação em blocos vem sendo usada em diversos outros aplicativos também baseados em LOGO.

A versão abordada neste trabalho será a 2.0 beta. Para utilizá-la basta entrar no site oficial do Scratch³ [10] e clicar em experimentar. Uma vez que o programa foi aberto não é mais necessário que a internet esteja ligada, porém, algumas funções ficarão indisponíveis.

Para o uso completamente *off-line* é necessário fazer o download do Adobe AIR⁴ para executar o aplicativo.

³ https://scratch.mit.edu/

⁴ Adobe AIR é um sistema multi-plataforma desenvolvido pela Adobe Systems para a construção de aplicativos que permite ser executados como aplicativos de desktop ou em dispositivos móveis, no caso do Scratch ele é necessário para executar a aplicação disponível no site em seu desktop.

2.2 Apresentando a interface e os principais comandos

A interface do Scratch é dividida em quatro partes principais, como numerado na Figura 1.



Figura 1: Interface do Scratch.

A primeira parte é o palco, nela é possível ver o que cada bloco ou conjunto de blocos faz com o personagem. No caso de um projeto para fazer um jogo ou contar uma história, podemos expandir para tela cheia, clicando no botão que aparece na parte superior, à esquerda e visualizar apenas o personagem realizando os comandos, mas para isso todos os algoritmos devem poder ser ativados com auxilio do teclado e do mouse. Nela podemos ver também uma bandeira verde que pode ser utilizada como botão que começará o algoritmo e um octógono vermelho que interromperá qualquer processo que esteja sendo realizado.

A segunda parte são os blocos que formarão o algoritmo, os sons que podem ser adicionados e uma área de customização do personagem. Os blocos devem ser montado na terceira parte, que é a área de *script* e a quarta parte que é a lista de *sprite*⁵ organiza os personagens e planos de fundo utilizados de forma que, quando selecionada, o algoritmo criado atuará sobre aquele objeto. Os blocos são codificados com cores como na Quadro 1 e nas Figuras 2 e 3.

⁵ Em computação gráfica, um sprite (do latim spiritus, significando "duende", "fada") é um objeto gráfico bi ou tridimensional que se move numa tela sem deixar traços de sua passagem .

Azul para movimento	Blocos que, de alguma forma, mudam a posição do personagem ou informam a posição atual.
Lilás para aparência	Blocos que modificam cor, tamanho ou fantasia do personagem e ainda têm opções de falas, pensamentos e mudanças no plano de fundo.
Magenta para som	Blocos que incluiem não só os sons, mas modificam seu volume, sua intensidade e frequência.
Verde escuro para caneta	Blocos que utilizamos para, ao andar, o personagem desenhe uma linha no seu rastro. Também podemos modificar a cor, o tamanho e a intensidade da caneta.
Laranja para variáveis	Blocos que modificam o valor das variáveis criadas que podem aparecer próximas ao personagem, possuindo as formas: normal, grande e potenciômetro ⁶ .
Marrom para eventos	Blocos que devem ser inseridos no início do algoritmo determinando qual tecla ou botão da tela irá fazê-lo funcionar.
Bege para controle	Blocos onde estão os comandos de repetição que são responsáveis por reduzir o algoritmo e testar condições necessárias para que ele funcione.
Azul claro para sensores	Blocos que possuem comandos que avaliam se teclas ou botões estão sendo pressionados e também comandos que armazenam dados inseridos pelo usuário.
Verde claro para operadores	Blocos que são os operadores matemáticos como soma, subtração, igual, seno, tangente, etc.
Roxo para mais blocos	Ao se criar um algoritmo que precisará ser repetido várias vezes, podemos transformá-lo em um único bloco que fará tudo que o algoritmo faz.

Quadro 1: Codificação de cores.

⁶Potenciômetro é uma barra deslizante que permite alterar o valor da variável na janela de visualização, a variável pode ser modificada dentro de um limite estabelecido pelo usuário



Figura 2: Blocos.



Figura 3: Blocos.

Para começar a montar um algoritmo deve-se arrastar o bloco para a área de *script*, parte 3, como na Figura 4.



Figura 4: Montando algoritmo.

Para juntar dois blocos, basta repetir o processo arrastando um deles para perto do qual (o outro) se quer juntar como na Figura 5. Os blocos possuem certos formatos com pequenas abas onde outro bloco pode ser encaixado. Alguns deles também possuem espaços vazios em seu interior e ali se pode inserir um bloco. Para saber quais encaixam nesses espaços, basta prestar atenção no formato do mesmo e do espaço. Os que possuem um espaço oval, por exemplo, podem ser preenchidos com todos que possuem cantos arredondados. Também podemos inserir números, palavras ou escolher entre uma lista de opções preexistentes, que criam infinitas possibilidades de algoritmos.





Caso faça sentido que dois blocos fiquem juntos, assim que se aproximarem vai aparecer um contorno branco no primeiro bloco indicando aonde esse segundo irá encaixar, caso contrário, o novo bloco simplesmente ficará solto sem qualquer interação e, para excluir um bloco, basta arrastá-lo novamente para a parte 2 (veja Figura 1).

Todos esses fatores como formatos e cores evitam que erros aconteçam, pois se o aluno fizer um algoritmo muito grande e não testar corretamente ao longo do processo será muito difícil encontrar onde está o erro. Esses problemas relativos ao acúmulo de erros em algoritmos são muito comuns em outras linguagens de programação mais complexas, as quais somente depois que parte do código é escrito podemos compilar e testar se tudo funciona corretamente.

Os nomes dos blocos são simples, de modo que é fácil de saber qual bloco deve ser

usado para cada ação. Assim o aluno não precisa se preocupar com a sintaxe e pode focar na parte lógica. Por exemplo, quero um algoritmo que, ao clicar na barra de espaço, faça o gato andar para frente e miar, o algoritmo ficará como na Figura 6.

Quan	do alg	uém p	oressi	onar	a t	ecla	es	paço	•
anda	100 p	assos	1		÷.,		÷.,	÷.,	
toca	o som	meow	· •).						
1.0		· ·							

Figura 6: Exemplo de algoritmo que faz o personagem gato andar 100 passos para a frente e, em seguida, miar.

🔥 Scratch 2 Offlin	ne Editor					
SCRATCH (🗒 Arquivo 🔻 Editar	▼ Dicas	About			
	Kurdî					
v404	Latina					-
/	Latviešu					
	Lietuvių					
1000	Magyar					
	Nederlands					
	Norsk Bokmål	-				
	Polski	Contraction of the second seco				
	Português	K				
	Português (Brasil)					
	Română					
	Slovenčina					
	Slovenščina					
	suomi					
	svenska			X: -29	y: -5	0

Figura 7: Escolha de idioma..

Um passo importante a ser realizado quando se começa a utilizar o Scratch é observar se ele se encontra na língua de preferência do usuário. Quase todo o programa está traduzido. Caso não esteja na língua adequada, basta, a qualquer momento, clicar no globo indicado na figura 7 e escolher em uma lista o idioma desejado.

Pode-se observar ao lado do globo, uma aba de arquivos, onde se vê as opções abrir,

salvar, fechar, fazer *upload* de projetos e checar por atualizações; uma aba de editar com as opções de recuperar, que desfaz a última ação; disposição com palco pequeno, que muda o tamanho das partes citadas na Figura 1 e modo turbo, que acelera o processamento do algoritmo, permitindo que algoritmos longos possam ser executados rapidamente.

Por fim, temos as dicas (seção não traduzida), onde podemos encontrar um exemplo e uma breve explicação de cada bloco, além de tutoriais para iniciantes. Mesmo em inglês é possível entender muita coisa, pois o formato e a cor dos blocos se mantêm iguais em todas as traduções.

As mudanças nos personagens da Figura 8 são muito importantes, caso seja necessário criar um botão ou algo que, quando clicar com o mouse, resulte em uma ação. Se quisermos fazer um programa como uma calculadora, que se encontra nas atividades deste trabalho, devemos criar um personagem para cada botão e assim haverá uma área 3 para cada personagem, de acordo com a divisão apresentada na Figura 1.

Além das opções de desenhar seu próprio personagem, também podemos importar um já pronto da biblioteca do programa ou carregar um que já esteja no computador, o que é muito útil se o professor quiser criar os blocos para que os alunos não tenham que se preocupar e nem gastar muito tempo com a parte de design. Também é possível tirar uma foto com a *webcam*, a mesma será inserida no programa automaticamente, veja Figura 9.

O plano de fundo funciona da mesma forma que os personagens, porém, ele possui um número menor de blocos para serem utilizados em seu algoritmo. Por exemplo, como ele não se move, ele não possui nenhum bloco na parte de movimentação.



Figura 8: Edição do personagem.



Figura 9: Opções de inclusão de personagem.

A parte de mudança nos sons (veja Figura 10), também possui as opções de se importar sons da biblioteca, gravar um áudio utilizando o microfone ou utilizar um arquivo de áudio que exista no computador. As opções de mudanças nos efeitos do som já são mais complicadas de se mexer, pois o aluno necessitaria de um pouco mais de conhecimento na área, mas nada que não possa ser feito com um pouco de prática, tentativa e erro.



Figura 10: Opções de som.

Existem também cinco atalhos na parte superior do programa, como no Quadro 2, que facilitam o trabalho.

	Duplicar: Ele criará um outro personagem			
*	idêntico, com os mesmos algoritmos.			
	Apagar: Ao ser selecionado, transformará			
	a seta do mouse em uma tesoura e quando			
70	clicar em um bloco, personagem ou plano			
	de fundo, irá apagá-lo.			
X	Aumentar o tamanho do personagem.			
X	Reduzir o tamanho do personagem.			
	Ajuda do bloco: Transformará a seta em			
0	uma interrogação que, ao clicar em algo,			
	abrirá a barra de dicas respectiva ao objeto			
	clicado.			

Quadro 2: Atalhos.

2.3 O site e a comunidade do Scratch

O site do Scratch possui diversos ambientes para todos os tipos de público, ele é traduzido total ou parcialmente para mais de cinquenta línguas que podem ser modificadas numa caixa de seleção no final da página, como na Figura 11.

About Sobre Scratch Para os pais Para Educadores Trabalhos	Community Diretrizes da Comunidade Fóruns de Discussão Scratch Wiki Statistics	Support Help Page FAQ Offline Editor Fale conosco	Legal Terms of Use Politica de Privacidade DMCA	Donate If you enjoy using Scratch consider making a donation to the Code-to-Learn Foundation to support future development. Donate	
	O Scratch é	um projeto do Grupo Lifelong K	indergarten do MIT Media Lab		
E	daidiamaa				

Figura 11: Seleção de idioma.



Ao entrar, podemos visualizar três opções: experimente, ver exemplos e inscreva-se.

Na parte "Experimente", podemos abrir o aplicativo online do Scratch. É possível abrir mais de um aplicativo ao mesmo tempo e mexer simultaneamente.

Na parte "ver exemplos", podemos ver diversas atividades feitas no Scratch. Elas têm o papel de mostrar o potencial do aplicativo e inspirar novos usuários. Os exemplos são divididos em seis categorias: animações, jogos, arte interativa, música e dança, história e detecção de vídeo.

A parte "inscreva-se", permite que o usuário aproveite o máximo do site. É possível criar uma conta e fazer projetos no aplicativo online, podendo salvá-los e disponibilizá-los ao público em sua página. Também é possível receber mensagens de outros usuários na forma de comentários nos seus projetos publicados ou como mensagem privada. Sua conta possui um perfil onde é possível por informações sobre o que você está trabalhando.

Um outro site fornece um espaço para educadores denominado ScratchED [11]. A inscrição realizada no site oficial do Scratch não é a mesma do ScratchED, nele é feita uma verificação manual de cada usuário que se inscreve para evitar contas falsas. No ScratchED, existem fóruns de discussão que contam com profissionais de diversas áreas, tirando dúvidas e comentando sobre projetos e atividades. Um profissional que esteja pensando em utilizar o

Figura 12: Navegação no site.

Scratch de determinada forma, pode navegar entre diversos usuários e projetos divididos em quatro categorias: *Education Level* (Nível de Escolaridade), *Content type* (tipo de conteúdo), *Curricular Area* (Área curricular) e *Language* (Língua). Essas divisões possuem suas subdivisões como na Figura 13.

Search Resources



Figura 13: Divisões para busca.

Dessa forma é possível filtrar melhor o tipo de trabalho que deseja consultar.

Ao se registrar, os usuários devem dar informações de endereço. Caso tenha interesse de saber se existe algum educador que more perto de você e também esteja trabalhando com Scratch, o site possibilita que você busque usuários utilizando auxilio de um mapa do Google, como visto na Figura 14.



Figura 14: Busca de usuários.

Existe a possibilidade de filtrar por área de interesse do usuário, veja Figura 15.



Diversos eventos, minicursos ou pequenos encontros são compartilhados no site. A comunidade brasileira no site ainda é pequena, por isso ainda é difícil encontrar muitos trabalhos ou membros.

2.4 Estatísticas

Analisando os dados estatísticos nos gráficos fornecidos pelo site oficial do Scratch [10] observa-se não só quem são os usuários, mas também o quanto eles atuam no site, com mais de três milhões de projetos compartilhados, mais da metade dos usuários possuem idade entre dez e quinze anos.

Os dados em relação à idade ajudam a discernir qual público demonstra mais interesse na utilização do Scratch, facilitando a escolha quanto a usá-lo em determinada turma ou não.



O Gráfico 1, mostra a relação entre a quantidade de usuários e as suas idades.

Gráfico 1: Idade x Número de usuários.

Os usuários não são só numerosos, a atuação no site como quantidade de projetos que são postados ou quantidade de comentários vem aumentando desde 2007. Em abril do ano de 2013 foram criados mais de sessenta e um mil projetos, já em abril de 2014 foram mais de duzentos mil projetos, como mostra o Gráfico 2. Conforme o número de usuários e de projetos cresce, o site ganha mais conteúdo, tornando-o uma ferramenta cada vez melhor, pois muitos erros, dúvidas ou curiosidades podem ser sanadas nos fóruns e mensagens privadas para outros usuários.

O Gráfico 2 mostra em azul a quantidade de novos projetos ao longo dos meses e, em



laranja, a quantidade de projetos que foram alterados de alguma forma.



Um grande problema na análise dos números de usuários cadastrados é que eles podem ser apenas cadastrados, mas não ativos no site. No Gráfico 3, podemos observar os usuários ativos em cada mês. No primeiro quadrimestre de 2014, foram mais de trezentos e cinquenta mil usuários ativos, fazendo uma média de mais de oitenta e sete mil usuários ativos por mês.



Gráfico 3: Meses x Usuários ativos.

3 Revisão bibliográfica

Neste capítulo discutiremos sobre algumas pesquisas, veremos alguns trabalhos e artigos de professores, pesquisadores e estudantes que utilizam a linguagem de programação Logo ou Scratch para ensinar conteúdos matemáticos, mostrando como ela foi aplicada, com quais objetivos e alguns resultados encontrados.

3.1 Logo: A proposta de aprender matemática utilizando uma linguagem de programação.



Figura 16: Ambiente de trabalho de uma versão do LOGO.

Utilizar o LOGO na educação foi um desafio para muitas pessoas, principalmente pelo momento histórico em que ele foi desenvolvido.

Nos primeiros anos após sua criação, em 1967, era impossível utilizar o LOGO no ensino, pois não haviam computadores pessoais relativamente baratos que pudessem ser utilizados por escolas. Apenas no final dos anos setenta que isso começou a mudar e o LOGO foi se desenvolvendo cada vez mais.

Existem muitas dúvidas sobre a eficácia de se ensinar Logo para crianças. O trabalho com título *From Polygons to Functions to Orbits to Fractals: A Year of Logo Work With a Mathematically Gifted Student* (De Polígonos para Funções para Ortbitais para Fractais: Um Ano de trabalho utilizando Logo com um aluno que possui dons matemáticos)[12] mostra a

trajetória de um aluno que substituiu todas suas aulas de matemática da quarta série, em uma escola nos Estados Unidos, por aulas de Logo com seu professor Edie Adamson.

Os pais do aluno, ao perceberem seu excelente desempenho em matemática, queriam que ele se desenvolvesse ainda mais, assim, pediram para que Edie Adamson, que era seu professor de matemática na época, ministrasse aulas de Logo para seu filho por um ano, pois não seria um problema para ele perder um ano de matemática.

Pelos relatos de Adamson contidos no trabalho, ele não possuía um plano concreto sobre o que deveria ser explorado, sua única meta era que sempre tivesse algo para ser feito com o Logo e, ao longo do ano, ele abordou Polígonos, Funções, elipses e Fractais, o que ultrapassa em muito o conhecimento esperado para um aluno de nove anos.

Quando Adamson terminou o trabalho, ele sentiu vontade de repetir o processo com outros alunos, mesmo aqueles que não apresentassem um bom rendimento em matemática quanto o primeiro, mas o administrador da escola disse a ele que não havia mais ninguém que pudesse perder um ano de matemática. Quando Adamson contou para Seymour Papert sobre o ocorrido, ele o respondeu de uma forma que nos leva a refletir sobre como as aulas estão sendo ministradas, ele disse: "E o que eles perderiam?" se referindo ao que eles perderiam se não assistissem as aulas de matemática.

O Professor de informática Orlando Mihich, em seu trabalho intitulado *The Weaving Turtle, African Textiles* (A Tecelagem Tartaruga, Tecidos Africanos)[13], propôs um projeto para seus alunos em que eles eram solicitados a reproduzir padrões de tecidos africanos com o Logo, os quais seguem uma interessante simetria.

Esses padrões foram apresentados aos alunos devido a grande parte deles serem de descendência africana e, na época, tais padrões eram populares.

Com algumas alterações, essa atividade poderia trabalhar simetria, reflexão, rotação e muitas outras noções de geometria que estão implícitas nos desenhos, mas mesmo sem serem abordadas matematicamente, só a convivência do aluno com tais formas facilitaria a compreensão quando o assunto fosse abordado.

As atividades envolviam trabalhar transformações geométricas sobre figuras como simetrias, reflexões, rotações, para criar os padrões típicos. Mesmo sem serem abordadas matematicamente, o contato dos alunos com tais formas poderia facilitar a compreensão destes tópicos quando fossem abordados.

Um Artigo escrito por Michael Tempel com título The Turtle is Dead: Rethinking

Logo in the Age of Kid Pix (A Tartaruga está Morta: Repensando Logo em tempos de *Kid Pix*) [14] discute a questão sobre o Logo ser mais do que uma ferramenta para desenhar. Ao ser questionado, por um professor, porque ensinava Logo quando existia uma ferramenta para desenho muito melhor, o *Kid Pix*, que funciona de modo semelhante ao *Paint*, encontrado nos computadores com *Windows*, Tempel refletiu sobre o assunto e respondeu:

O Kid Pix pode ser usado para produzir um desenho, mas com o Logo você também aprende geometria, planejamento e depuração de erros enquanto cria sua imagem.

O professor que havia feito a pergunta ficou satisfeito, mas Tempel não, pois ao pedir que um aluno desenhasse um coelho no papel e depois tentasse reproduzir no computador utilizando Logo, ele se viu na difícil tarefa de mostrar porque o Logo era realmente melhor se seu desenho era rústico. Para ele o programa só faria sucesso se, não só os professores o considerassem importante, mas se a sociedade também o visse com bons olhos e só comparando o trabalho final nos dois programas, o *Kid Pix* seria melhor.

Ao pensar sobre outras figuras construídas no Logo ele chegou a conclusão que este aplicativo não é necessário para criar desenhos simples que podem ser feitos a mão livre, mas para criar padrões geométricos e figuras as quais os alunos não construiriam em outras ferramentas. Ele percebeu que o LOGO, diferente de outros aplicativos para desenhos, possibilitava colocar novas questões para ensinar matemática.

Em outro trabalho, cujo título é *Balance* [15], Tempel buscou estudar um pouco de probabilidade. Ele construiu um algoritmo que fazia a tartaruga ir para um ponto aleatório na tela branca desenhar um ponto preto, se ela parasse em um espaço branco, ou desenhar um ponto branco, caso contrário.

Como antes de iniciar o programa, a tela esta toda branca, o primeiro movimento cairá em um espaço branco e o segundo movimento tem chances muito pequenas de cair no mesmo lugar do primeiro. Utilizando o programa, os alunos puderam observar que, no início, diversos pontos pretos aparecem, até que haja quase tantos espaços pretos quanto brancos. Depois disso, nenhuma das duas cores irá ser visualmente predominante, pois as modificações nos próximos movimentos serão muito pequenas, fazendo com que o desenho tenda a um mesmo padrão.

Este padrão pode ser perturbado desenhando um grande círculo preto no meio, mas ao se executar o algoritmo mais vezes o círculo logo irá sumir, assim, o padrão encontrado é um balanceamento entre a quantidade de pontos pretos e brancos que ocorre quando a

probabilidade de a tartaruga se mover para um espaço preto seja quase a mesma do que se mover para um espaço branco.

Caso os alunos queiram um verificação mais precisa, do que a simples análise visual, de que as cores estão próximas de um balanceamento perfeito, poderiam fazer a tartaruga percorrer um enorme número de pontos, mas ao invés de desenhar um ponto branco ou preto, ela, apenas registraria a cor em uma variável, essa verificação não foi proposta por Tempel.

Em um trabalho de conclusão de curso, feito pela estudante de licenciatura em matemática Marília Luiza Matte, intitulado: A linguagem Logo como possibilidade de aprendizagem em matemática [16], são propostas atividades com o Logo para alunos do ensino médio. Em um minicurso, ela também o aplicou para estudantes de licenciatura em matemática, o que se mostrou muito útil uma vez que eles contribuíram para o aprimoramento do trabalho.

As atividades se assemelham às do presente trabalho, pois tem um grande foco na geometria, principalmente na construção de formas geométricas com auxílio do logo.

Com o grupo de estudantes de licenciatura, ela consegue não só ensiná-los a utilizar uma nova ferramenta, mas desperta também o interesse para que eles busquem formas de aplicar aquilo em sua vida profissional. Como muitos nunca haviam tido contato com uma ferramenta tecnológica para ensinar matemática, ela também expandiu as possibilidades desses alunos, o que pode levá-los a buscar outras formas de ensinar determinados conteúdos.

Quando aplicado para alunos do ensino médio, apenas uma aluna participou de todas as três aulas, isso dificultou uma melhor análise das atividades, mas nos dois grupos surgiram problemas relacionados a sintaxe, pois como no Logo os comandos devem se escritos, há uma necessidade que se aprenda diversos comandos e o não conhecimento de algum deles pode impossibilitar o trabalho.

3.2 Trabalhos com Scratch

Comparando com o Logo, o Scratch é muito mais atraente para crianças pois a sua interface é bastante intuitiva. Além disso, o Scratch possui muitas ferramentas que seu antecessor não tem.

A maioria dos trabalhos encontrados no site do Scratch [10], envolve a construção de jogos por parte dos alunos. Os projetos disponíveis no site do Scratch não mencionam como

os conceitos matemáticos envolvidos na construção dos jogos são explorados.

Na dissertação de mestrado de António Sorte Pinto, "Scratch na Aprendizagem da Matemática no Primeiro Ciclo do Ensino Básico" [17], temos um exemplo da utilização do Scratch para trabalhar conceitos matemáticos de uma maneira mais direta. Ele propõe atividades para serem resolvidas inicialmente com cálculo mental e, posteriormente, com o Scratch.

Ao fazer a transição do ambiente lápis e papel para o Scratch os alunos utilizam o Scratch apenas para registrar as contas feitas mentalmente e fazer o personagem dizer o resultado, assim o personagem "fala" todo o raciocínio e depois o resultado. Em nenhum momento o algoritmo criado pelo aluno resolve o problema, assim, a capacidade computacional do Scratch é posta de lado e ele passa a possuir a mesma função que uma folha de papel no que diz respeito a registrar o problema.

As atividades apresentadas no trabalho acima são muito úteis para uma introdução do programa, quando o usuário ainda não possui conhecimento de todas suas ferramentas.

Uma proposta interessante para explorar as operações aritméticas com os números inteiros e que, na nossa visão, aproveitaria mais o potencial do Scratch seria a construção de uma calculadora.

Em outros trabalhos, como ZCARTAS [18], o professor utiliza o Scratch como ferramenta para criar um jogo educativo com a função de explorar operações com números inteiros, onde uma operação entre dois inteiros é exibida na tela e o aluno deve dizer a resposta, recebendo assim uma confirmação de se acertou ou não. Uma atividade deste tipo é muito boa para ser utilizada fora de sala de aula, pois o jogo em si já oferece as perguntas, os meios para resolvê-las e as respostas, sendo possível que o aluno aprenda sem auxílio do professor. Se utilizada em sala, ela pode ser uma alternativa mais atrativa aos exercícios.

Acreditamos que um problema a ser contornado pelo professor seja o de estabelecer critérios para correção e avaliação do processo utilizado pelo aluno, pois em alguns jogos, por tentativa e erro, o aluno poderia obter "uma pontuação positiva" mesmo que não entenda nada do que esteja acontecendo.

Um artigo escrito por Fernando Celso Villar Marinho e Miriam Struchiner, com o título "Estudantes do ensino básico como desenvolvedores de jogos digitais" [19], aborda a criação de jogos digitais por alunos do ensino médio de um colégio da rede estadual do Estado do Rio de Janeiro, no distrito de Pedro do Rio, no município de Petrópolis.

O Scratch é utilizado como ferramenta inicial para a criação dos jogos, uma vez que sua interface intuitiva permite que os alunos sejam introduzidos no universo da programação e obtenham algum resultado em curto prazo, facilitando, posteriormente, a utilização de outros softwares mais complexos.

Os professores envolvidos no trabalho acima [19] relataram comportamentos muito interessantes nos alunos participantes, como, por exemplo: eles aprenderam a pedir e oferecer ajuda, a trabalhar em grupo, onde cada um tinha uma função fundamental no desenvolvimento do jogo, mostraram grande interesse e engajamento buscando e propondo soluções, entre outros.

Outro modo de utilizar o Scratch, que é visto em diversos projetos publicados no site[10], seria como ferramenta para mostrar processos dinâmicos ou simulações que não são fáceis de serem compreendidos quando são feitos no quadro ou em papel, por exemplo, a atividade de aproximação de π presente no anexo deste trabalho.

As atividades que serão apresentadas no próximo capítulo possuem uma estrutura semelhante a algumas atividades desenvolvidas pelo site *code.org* [20], porém, neste último, o foco é ensinar a programar e a matemática envolvida não é explorada. Neste trabalho, as atividades são propostas com o intuito de aproveitar a capacidade do Scratch e explorar problemas de uma forma diferente, focando no conteúdo matemático e deixando o aprendizado de programação como consequência.–Os alunos são direcionados a montar e modificar algoritmos a fim de desenhar figuras geométricas e investigar algumas propriedades.

4 Perspectiva de ensino e Proposta de atividades

4.1 A metodologia da resolução de problemas

A resolução de problemas é uma metodologia de ensino onde não se enfatiza a aplicação de fórmulas e a simples obtenção dos resultados, mas todo um processo de análise e compreensão dos métodos utilizados na resolução do mesmo. Nela, a construção do conhecimento não segue a regra de ensinar o conteúdo e resolver um problema aplicando este, pois, durante a resolução, novos conteúdos podem surgir.

O primeiro passo dessa metodologia é a concepção do problema, ele deve permitir uma

análise do processo de obtenção das respostas, um problema muito simples cuja solução é única como a resolução de uma equação de uma variável, por exemplo, não deixa muita margem a discussão e, por não permitir respostas diferentes, acaba enfatizando que o importante é o resultado.

Um problema ideal para este método pode abordar um conceito, princípio ou procedimento ainda não visto em sala de aula e deve permitir diferentes processos de resolução. Após resolvê-lo e discuti-lo, uma reformulação mais genérica pode ser apresentada.

Ao propor um problema em que o aluno desconheça um método específico para resolvê-lo, ele é incentivado a explorar e descobrir formas possíveis de fazê-lo, por vezes, encontrando métodos não tão práticos quanto as fórmulas conhecidas pelo professor, deixando a resposta como coadjuvante, focando assim no processo que levou o aluno a esta solução.

Por depender muito das descobertas feitas pelo aluno, algumas vezes, pode se tornar um processo lento, mas ele elimina a automatização na resolução com fórmulas aumentando as chances de que o aluno realmente compreenda o que está acontecendo.

No Scratch, não existe uma única maneira de montar um algoritmo que resolva um problema. Mesmo uma ação simples como "ande cem passos para frente" pode ser feita de diversas maneiras diferentes, exemplo:



Figura 17: Possiveis soluções da ação "andar cem passos para frente".

Assim, as resoluções de um problema proposto aos alunos podem gerar espaço para discussão e aprendizado sobre os diferentes caminhos para resolvê-lo.

Diferente de problemas feitos em sala de aula onde o aluno só terá um feedback do seu desenvolvimento e resultado depois que apresentar ao professor, no Scratch é exibido simultaneamente o resultado e o aluno pode observar se foi o esperado ou não. No exemplo da figura 17, o personagem claramente "andaria" para frente. A constatação de que foram cem passos poderia ser feita observando a abscissa no início e no fim da execução do algoritmo.

Podemos observar que, na resolução de um problema com o Scratch, existe um grande

foco no algoritmo que, no caso, é o método para resolvê-lo, pois mesmo que o personagem ande os cem passos para frente, o algoritmo pode estar errado, assim, ele não funcionaria em qualquer caso, exemplo:



Figura 18: Algoritmo errado para a ação de "andar cem passos".

O algoritmo da figura 18 modificaria as coordenadas x e y do personagem para x + 100 e zero respectivamente, assim, nos casos particulares onde a coordenada y já fosse zero, ele faria o personagem andar apenas os cem passos para frente e aparentaria estar correto. Neste exemplo, podemos observar que o método utilizado é mais importante do que o resultado.

O Scratch oferece um ambiente bastante propício para a utilização da metodologia da resolução de problemas. Veremos que a construção de algoritmos por blocos e, com isto, a possibilidade de fazer generalizações, possibilita colocar novos problemas aos alunos que não seriam viáveis apenas utilizando lápis e papel, como alguns presentes nas atividades deste trabalho.

4.2 Atividades

Nesta seção, apresentaremos cinco atividades que utilizam o Scratch para ensinar conteúdos matemáticos, formuladas por nós. Sempre que pertinente, discutiremos a diferença entre a atividade ser realizada no ambiente papel e lápis e com o uso do Scratch.

Para cada atividade, indicamos: os objetivos de aprendizagem, os pré-requisitos que consideramos necessários, o material necessário e os conceitos que serão explorados. A apresentação das atividades neste capítulo inclui os exercícios, respostas, textos voltados para os alunos, como definições, exemplos e textos voltados para o professor com explicações adicionais.

No Apêndice, as atividades são disponibilizadas na forma como entregaríamos aos alunos, sem as respostas ou anotações para os professores.

As atividades possuem como público-alvo, alunos do ensino médio.

4.2.1 Atividade polígonos regulares

O objetivo desta atividade é direcionar o aluno a construir um algoritmo que desenhe polígonos regulares.

Se feito de modo tradicional, e fosse pedido para que os alunos desenhassem, a mão livre, figuras que eles conhecem, como quadrado e triângulo, não haveria nenhuma dúvida no processo. Mesmo sem nenhum conhecimento das propriedades de cada polígono, até os mais complexos poderiam ser desenhados sem muita precisão, bastaria utilizar régua, compasso e esquadro. Porém, seria um trabalho demorado sendo difícil o aluno adotar uma postura exploradora e desenhar diversos polígonos diferentes por tentativa e erro.

No Scratch, para desenhar qualquer polígono, é preciso saber algumas das suas relações, assim, é possível mostrar, nos casos mais simples, relações como a soma dos ângulos externos ser igual a 360° e o ângulo interno e o externo serem suplementares.

Depois de montado o algoritmo, o aluno tem a possibilidade de brincar com ele, explorando as diversas possibilidades de polígonos regulares ou cometendo erros e observando o resultado desses erros.

Nesta atividade, o aluno é familiarizado com o comando de repetição, os blocos de movimento responsáveis por mover um número de passos e girar um determinado número de graus e também com o bloco "use a caneta", que desenhará o percurso realizado pelo personagem.

Atividade polígonos regulares

Resumo da atividade

Objetivos: explorar as propriedades dos polígonos regulares e chegar a uma fórmula que relacione o número de lados com os ângulos externos de um polígono regular e também uma fórmula que relacione o número de lados com os ângulos internos.

Pré-requisitos: noção de ângulo, noções básicas de informática.

Material: computador com o programa Scratch previamente instalado ou acesso a internet.

Conceitos explorados: ângulos internos e externos, polígonos regulares e suas propriedades.

Construção de polígonos regulares com Scratch

O quadrado é um exemplo de polígono regular, a palavra polígono vem do grego "poly" que significa muitos e "gon" que significa ângulo, ou seja, um polígono é uma figura fechada que possui muitos ângulos. Um polígono é chamado de regular se todos os seus ângulos internos forem congruentes e se todos os seus lados tiverem a mesma medida.

use a caneta	Desenhará uma linha no percurso que o personagem percorrer.
mova 50 passos	Fará o personagem andar, para frente, 50 passos.
gire (1 90 graus	Fará o personagem girar, para direita, 90 graus.

1) Comandos necessários para construir um quadrado:

Basta juntar estes três blocos para formar um algoritmo e, cada vez que se clicar neles, um lado do quadrado será desenhado. Para que não seja necessário clicar quatro vezes no algoritmo, podemos usar um bloco de repetição que fará com que, ao clicar, ele se repita um determinado número de vezes. No nosso caso, como o quadrado tem quatro lados, queremos quatro repetições.

O algoritmo para desenhar um quadrado será, então:



Figura 19: Algoritmo do quadrado.

Perceba que, no algoritmo, o personagem está andando sempre o mesmo número de passos antes de girar – desenhando, assim, cada lado - e girando a mesma quantidade de graus – produzindo, assim, os ângulos.

Para termos certeza de que a figura desenhada é um polígono, precisamos verificar se é fechada. Antes de clicar no algoritmo observe no canto direito superior da sua tela que aparece a posição do personagem em coordenadas, como na imagem:



Figura 20: Posição do personagem.

Se esses números permanecerem iguais antes e depois do algoritmo ser executado, quer dizer que o personagem voltou exatamente para a posição que ele estava, verifique algumas vezes para ter certeza. Caso seja a mesma posição, a figura é fechada.

2) Construção de um triângulo equilátero:

O triângulo regular é chamado de equilátero. Para construí-lo, nós utilizaremos o mesmo algoritmo do quadrado, porém precisamos saber qual ângulo de giro devemos usar e quantas repetições serão necessárias. Como ele possui três lados, serão necessárias três repetições. E o ângulo será de 120 graus, assim o algoritmo será:



Figura 21: Algoritmo do triângulo.
Os polígonos possuem ângulos internos e externos. O interno é o menor ângulo entre dois lados consecutivos (ângulo em vermelho na Figura 22). Prolongando um lado do polígono, o ângulo formado por esse prolongamento e pelo lado do polígono é chamado de externo (ângulo em verde na Figura 22).



Figura 22: Ângulos internos e externos

Na construção do quadrado e do triângulo, o personagem, ao final de cada lado, em vez de continuar em frente, ele vira para a direita (sentido horário), este movimento pode ser melhor percebido na Figura 23; assim, o ângulo que escolhemos para ele virar é o ângulo externo. Perceba também que juntando o ângulo interno e o externo obtemos um ângulo de 180 graus.



Figura 23: Rotação do personagem.

A partir das construções acima, responda às seguintes questões no seu caderno.

- 3) Qual a soma dos ângulos externos do quadrado que foi desenhado no Scratch?
- 4) Qual a soma dos ângulos externos do triângulo equilátero desenhado no Scratch?
- 5) Que conclusão você pode tirar ao comparar os resultados encontrados nos itens 3 e 4?
- 6) O que devemos alterar no algoritmo para construir um polígono de seis lados? sabendo que a soma de seus ângulos externos também será 360 graus.
- 7) E se quisermos construir um polígono de cinco lados?

	Lados	Ângulo externo	Ângulo interno	Soma dos ângulos internos
Triângulo	3			
Quadrado	4			
Pentágono	5			
hexágono	6			

8) Preencha a tabela com os dados os polígonos regulares construídos até agora:

9) Com as informações obtidas, você seria capaz de dizer a soma dos ângulos internos dos seguintes polígonos?

Heptágono (7 lados)	
Octógono (8 lados)	
Eneágono (9 lados)	
Decágono (10 lados)	

- 10) Qual a soma dos ângulos internos de um polígono regular de n lados?
- Com tudo que você descobriu, modifique o algoritmo para desenhar um icoságono (vinte lados).

4.2.2 Atividade de construção da circunferência

Imagens digitais são formadas por pixels quadrados, então, programas de computadores não constroem circunferências, mas sim aproximações das mesmas. O desenho que vemos na tela de um computador é, na verdade, uma série de pequenos quadrados que criam uma aproximação, existem diversos métodos para realizar essa construção, um deles é o de um polígono de n lados, com n suficientemente grande, como feito nesta atividade.

O Scratch não possui uma ferramenta específica para construir uma circunferência (ou melhor, uma aproximação de uma circunferência), assim esta atividade convida o aluno a ele próprio construir de um modo simples e, com isso, refletir sobre as limitações de um computador e sobre os processos de aproximação em matemática.

Desenhar uma circunferência em um papel é fácil, pois, dado o seu raio, basta marcar o centro, traçar um segmento de reta do tamanho deste raio e, com auxílio de um compasso cuja abertura seja do mesmo tamanho deste segmento de reta, traçar a circunferência. Vemos aqui o potencial do uso de um recurso como o Scratch para suscitar questões que não surgem quando a mesma atividade é realizada no ambiente lápis e papel.

Nesta atividade, partimos do algoritmo criado para a construção de polígonos regulares e introduzimos os blocos de variáveis e operadores.

Atividade construção da circunferência

Resumo da atividade

Objetivo: construir uma aproximação de uma circunferência dado seu raio.

Pré-requisitos: polígonos regulares, circunferência, algoritmo para construir polígonos regulares, conhecimento básico no uso do Scratch.

Material: computador com acesso à internet ou com o programa Scratch previamente instalado, papel e lápis.

Conceitos explorados: polígonos, raio da circunferência e aproximação.

Construindo uma circunferência a partir do raio

O Scratch não possui um bloco ou algoritmo específico para construir uma circunferência. Nesta atividade construiremos uma aproximação de uma circunferência.

1) Utilizando o algoritmo para construir polígonos regulares, da atividade anterior, experimente mudar as repetições para 360, o ângulo para 1 grau e o número de passos também para 1, como na figura:



2) A figura construída é um polígono ou uma circunferência? Por quê?

Para o professor:

Nesse momento será interessante pedir para que os alunos expliquem porque acham que a figura obtida é um polígono ou uma circunferência. Caso os alunos tenha dificuldade em entender que é um polígono, a seguinte figura pode ser mostrada para explicar:



Nesta figura, ampliamos um pedaço do polígono de 360 lados e, com essa ampliação, é fácil mostrar que se trata de uma forma poligonal..

Usaremos o polígono regular de 360 lados como uma aproximação da circunferência e para facilitar a compreensão dos próximos passos o chamaremos de circunferência. Lembre que como estamos trabalhando com uma aproximação de uma circunferência os resultados encontrados nas contas possuirão uma margem de erro.

3) Modifique o algoritmo para 180 repetições, observe e anote as coordenadas do personagem antes e depois de executá-lo.



Na Figura 26 podemos ver dois personagens, os quais representam as posições antes e depois da execução do algoritmo.

4) Com os dados anotados do exercício 3, encontre a medida da linha pontilhada entre os personagens da Figura 26. O que esta medida representa na circunferência? Qual a sua relação com o raio?

5) Qual o raio desta circunferência?

Para o professor:

Caso o personagem não esteja na posição de 90°, o aluno poderá ter alguma dificuldade em encontrar o raio, e, se encontrar, é provável que os valores encontrados pelos alunos difiram entre si. Nesse caso, aconselhe o aluno a utilizar o bloco:

aponte para a direção 907 graus

Figura 27: Bloco de direção.

que deverá ser inserido no inicio do algoritmo, assim, o raio será apenas a variação da coordenada y.

Como nosso objetivo é construir uma circunferência dado seu raio, temos que modificar este algoritmo para que ele funcione em função do raio. Nos exercícios anteriores, descobrimos o raio nas condições de girar um ângulo de 1 grau e mover 1 passo, assim temos a relação entre o raio e o lado.

6) Seguindo a proporção entre o lado dado e o raio encontrado no exercício anterior, se o raio for 1, quantos passos o personagem deverá dar? Escreva o resultado com quatro casas decimais, arredondando se necessário.

7) A partir do resultado do exercício anterior construa uma fórmula onde dado o raio, você obtenha quantos passos o personagem deve andar.
(dica: pense no problema 6, mas se o raio fosse X)

8) Crie uma variável chamada "raio" no Scratch e, no lugar do número de passos, utilize a fórmula encontrada acima, de forma que, modificando a variável raio, você obtenha uma circunferência com aproximadamente este raio.

Para o professor:

A medida encontrada no exercício 4, será de 114 se o personagem partir da posição inicial de 90° e próxima disto quando partir de outra posição. Este valor é referente ao diâmetro da circunferência, assim, o valor do raio encontrado deverá ser próximo de 57, neste caso teremos que para um raio de 57 devemos andar 1 passo e, com uma regra de três, podemos descobrir que para um raio de 1 devemos andar 0,0175 passos aproximadamente.

A fórmula pedida na questão 7 deverá ser o raio multiplicado por 0,0175 e o algoritmo deverá ficar da seguinte maneira:



Figura 28: Solução do exercício 8

4.2.3 Atividade para obter uma aproximação de π : partes I, II e III

Na primeira parte desta atividade, construiremos polígonos regulares inscritos em uma circunferência, através de uma junção dos algoritmos criados anteriormente.

Inscreveremos os polígonos da primeira na circunferência da segunda atividade, fazendo algumas alterações.

Será necessário utilizar o caderno para descobrir algumas relações que não seriam tão simples de serem mostradas no Scratch. O uso do aplicativo é interessante pois esta construção fará parte de um processo dinâmico que não poderia ser feito no papel.

Não será introduzido nenhum bloco novo, mas a utilização dos que já são conhecidos será aprofundada.

Atividade: Aproximação de π parte I.

Resumo da atividade

Objetivo: construir um polígono inscrito a circunferência a partir do raio da mesma.

Pré-requisitos: trigonometria; propriedades do triângulo retângulo, do triângulo isósceles; conhecimento básico no uso do Scratch como construção de polígonos regulares e variáveis; ter finalizado as atividades **polígonos regulares** e **construção da circunferência**.

Material: computador com acesso à internet ou com o programa Scratch previamente instalado, papel e lápis.

Conceitos explorados: polígonos regulares inscritos.

Polígono inscrito, circunscrito e aproximação de π Parte 1, polígono inscrito

Polígono inscrito é aquele cujos lados são cordas de uma circunferência de modo que todos os seus vértices são pontos da circunferência. Sabe-se que todo polígono regular pode ser inscrito. Exemplos:



Figura 29: Polígonos regulares inscritos.

Alguns polígonos não regulares também podem ser inscritos. Exemplos:



figura 30: Polígonos não regulares inscritos.

Nesta atividade trabalharemos somente com os polígonos regulares. Como todo polígono regular pode ser inscrito, basta desenhar o polígono e depois descobrir em qual circunferência ele está inscrito. Nosso objetivo será inscrever qualquer polígono regular em uma circunferência dada.

Partiremos do algoritmo obtido na atividade **construção da circunferência**, com um raio de tamanho 100.



figura 31: Algoritmo construção da aproximação de uma circunferência.

Com a circunferência desenhada, já podemos começar a fazer os polígonos. Se apenas juntarmos os blocos do algoritmo para desenhar polígonos regulares com os do algoritmo para construir a circunferência, ele não sairá nem do tamanho desejado e nem dentro da circunferência. Isto acontece porque no primeiro o movimento do algoritmo é andar um determinado número de passos para frente, assim ele já sairia da circunferência e o tamanho seria arbitrado por essa quantdade de passos e não pelo raio. Para que ele esteja dentro da circunferência vamos primeiro partir do desenho final já pronto e ver quais movimentos devemos fazer com o personagem. Vamos começar com o triângulo.



Figura 32: Triângulo inscrito

Observe que, na Figura 32, diferente da atividade **polígonos regulares**, onde o personagem girava sempre o mesmo ângulo, dividimos um ângulo em dois, no caso, a soma do ângulo girado no primeiro movimento com o do sétimo deve ser igual aos do terceiro e quinto movimento.

1) Em seu caderno, mostre que os ângulos que o personagem virará no primeiro e sétimo movimentos indicados na Figura 32 são iguais, sabendo que a reta R é tangente a circunferência no vértice A e paralela ao lado BC.

Feito o exercício 1, podemos concluir que em cada um dos movimentos citados, se girará um ângulo equivalente a metade do ângulo externo, sabemos também que o ângulo externo pode ser escrito em função do número de lados do polígono. Agora precisamos descobrir quantos passos devemos andar para desenhar o lado do triângulo.

2) Em seu caderno, encontre o lado, chamado de Li, do triângulo inscrito em função do raio, chamado de R, e do ângulo externo.



Figura 33: Lado em função do raio.

3) O algoritmo para construir o polígono inscrito ficará como na Figura 34, monte-o no Scratch completando as lacunas com as informações obtidas acima e escolha um raio e o número de lados do polígono



Figura 34: Algoritimo incompleto.

Depois de pronto, experimente modificar o raio e o número de lados para testar se está tudo funcionando corretamente.



Na segunda parte da atividade, construiremos polígonos regulares circunscritos à mesma circunferência da primeira parte. De modo muito semelhante, também utilizaremos o caderno, mas nesse caso o algoritmo fica mais complexo.

O uso do Scratch também se deve ao fato desta atividade fazer parte de um processo dinâmico que será concluído na terceira parte.

Utilizaremos o algoritmo construído nas etapas anteriores e não será necessário alterálo, apenas adicionaremos algumas variáveis que tornaram mais fácil a visualização dos resultados.

Atividade Aproximação de π parte II.

Resumo da atividade

Objetivo: construir um polígono circunscrito a circunferência a partir do raio da mesma.

Pré-requisitos: trigonometria; propriedades do triângulo retângulo, do triângulo isósceles; conhecimento básico no uso do Scratch como construção de polígonos regulares e variáveis; ter finalizado as atividades: **polígonos regulares e construção da circunferência**; parte I desta atividade.

Material: computador com acesso à internet ou com o programa Scratch previamente instalado, papel e lápis.

Conceitos explorados: polígonos regulares inscritos e circunscritos.

Polígono inscrito, circunscrito e aproximação de π Parte II, polígono circunscrito

Nesta parte da atividade, trabalharemos com os polígonos regulares circunscritos. Um polígono é circunscrito a uma circunferência se esta é tangente a todos os lados do polígono (Figura 36).



Figura 36 : Polígono regulares circunscrito.

Como nos polígonos inscritos, todo polígono regular poderá ser circunscrito, e alguns não regulares também poderão (Figura 37)



Figura 37 : polígonos não regulares circunscritos

Novamente, trabalharemos somente com os polígonos regulares. O desafio desta atividade será construir o polígono circunscrito começando do mesmo ponto em que o personagem parou na hora de fazer o inscrito, assim, quando executarmos o algoritmo completo, os dois polígonos estarão na mesma circunferência. Relembrando o algoritmo da atividade anterior (Figura 35).

Partindo do problema já resolvido, vamos ver quais movimentos o personagem deve fazer.



Figura 38: Mapa de movimentos.

Essa parte é muito parecida com a construção de um triângulo regular, a única diferença é que no primeiro e no último movimento andamos apenas metade do lado, sendo assim os ângulos permanecem iguais. Nesse momento encontramos o mesmo problema dos polígonos inscritos, devemos descobrir qual o tamanho do lado que chamaremos agora de Lc.

1) Em seu caderno, encontre Lc em função do raio e do ângulo externo.



Figura 39: Triângulo circunscrito.

Agora que sabemos o valor do lado em função do raio, só nos resta selecionar os comandos do Scratch para montar a parte que deverá ser encaixada no algoritmo da atividade passada.

2) Complete as lacunas do algoritmo com os valores encontrados.



circunscrito

Depois que terminar, junte este algoritmo ao criado na parte 1 desta atividade.





Na última parte desta atividade, partiremos do polígono inscrito e do circunscrito a uma mesma circunferência, ambos com mesmo número de lados. Escreveremos um algoritmo que calcule o perímetro de ambos, assim, aumentaremos a quantidade de lados de forma gradual para descobrir uma aproximação do perímetro da circunferência, com esses dados obteremos uma aproximação de π .

Com o Scratch é possível observar o processo dinâmico onde os polígonos gradualmente se aproximam da circunferência, até um ponto onde seja impossível diferenciálos, assim fica fácil observar que o perímetro das três figuras ficarão muito próximos uns dos outros.

Atividade Aproximação de π parte III

Resumo da atividade

Objetivo: obter uma aproximação de π através de um algoritmo que construa, dado uma circunferência e seu raio, um polígono inscrito e um circunscrito com o mesmo número de lados.

Pré-requisitos: perímetro de polígonos, comprimento de circunferências e conhecimento básico no uso do Scratch como construção de polígonos regulares e variáveis, ter finalizado as atividades de **polígonos regulares**, **construção da circunferência** e as partes um e dois desta atividade.

Material: computador com acesso à internet ou com o programa Scratch previamente instalado, papel e lápis.

Conceitos explorados: polígonos regulares inscritos e circunscritos, perímetro e método para descobrir o valor de π .

Polígono inscrito, circunscrito e aproximação de π Parte 3, aproximação de π

Nesta etapa da atividade, utilizaremos o algoritmo criado nas duas atividades anteriores para calcular uma aproximação de π . Sabemos que o comprimento da circunferência é $2\pi R$. Se explorarmos um pouco, desenhando polígonos com o número de lados cada vez maior, com o algoritmo construído nas duas primeiras partes desta atividade podemos perceber que eles se aproximam da circunferência. Como nós sabemos o raio da circunferência, podemos dizer que:

Perimetro do poligono circunscrito	>	π	>	Perimetro do poligono inscrito
2R				2R

Figura 43: Aproximação de π *.*

Agora precisamos calcular os perímetros.

1) Crie uma variável chamada "Perímetro do polígono circunscrito" e, utilizando apenas as outras variáveis já existentes, faça um algoritmo que calcule esse perímetro.

2) Crie uma outra variável chamada "Perímetro do polígono inscrito" e faça o mesmo do exercício 1.

3) Crie uma terceira variável chamada "Aproximação de π ". Utilizando os valores

encontrados nos exercícios anteriores, faça com que está variável seja o valor médio entre os extremos da desigualdade da Figura 43.

Para facilitar a visualização dos resultados na área de criação das variáveis, ao lado de cada variável, há uma caixa para ser marcada, deixe marcado somente as variáveis criadas nos exercícios 1, 2 e 3, conforme Figura 44. Assim os valores dessas variáveis aparecem próximos ao personagem, Figura 45.



Figura 44: Variáveis.



Figura 45: Janela de visualização.

4) preencha a tabela abaixo com algumas aproximações de π dizendo quantos lados possui o polígono utilizado nesse cálculo.

Número de lados dos polígonos	Valor aproximado de π		
3	3,897		
54			

Para o professor:

1)	
	mude Perimetro do polígono circunscrito 🗸 para 🛛 Lc * número de lados
2)	
_,	
	mude Perimetro do poligono inscrito V para
3)	
m	nude Aproximação do pi 🚽 para Perimetro do polígono circunscrito + Perimetro do polígono inscrito // raio * 4

4) Discuta os resultados obtidos pelos alunos, mostrando a precisão deste método, mostrando também que com 3 lados teremos a pior aproximação e a partir de 54 teremos a melhor

Número de lados dos polígonos	Valor aproximado do π	
3	3,897	
Livre para o aluno	Livre para o aluno	
Livre para o aluno	Livre para o aluno	
Livre para o aluno	Livre para o aluno	
54	3,142	
Livre para o aluno	Livre para o aluno	

Figura 46 : Soluções dos exercícios 1,2,3 e 4

Curiosidade: site com10.000 casas decimais de π ,

http://mathemathika.wordpress.com/2013/01/14/pi-com-10-000-casas-decimais/

4.2.4 Atividade para construção de calculadoras

Esta atividade tem como objetivo direcionar o aluno para que utilize o Scratch em algo que possa ser útil em sala de aula.

Diferente das propostas anteriores, neste caso não existe um roteiro, apenas uma ideia e um exemplo. Ao utilizar o Scratch para construir uma calculadora, o aluno aprenderá mais sobre o funcionamento da mesma e também terá um maior domínio do aplicativo e da estrutura lógica utilizada em algoritmos.

É possível criar calculadoras que resolvam os mais diversos cálculos, como uma rápida busca no site do Scratch [10] podemos encontrar vários exemplos já prontos, que podem auxiliar caso surja alguma dúvida de como fazer determinado algoritmo.

É interessante que a atividade acompanhe o conteúdo abordado no momento, como por exemplo, se os alunos estiverem estudando equação de segundo grau, eles devem construir uma calculadora que resolva essa equação.

Atividade construção de calculadoras

Resumo da atividade

Objetivo: montar um algoritmo para construir uma calculadora para auxiliar a resolução de problemas.

Material: computador com acesso à internet ou com o programa Scratch previamente instalado, papel e lápis.

Construção de calculadoras

Nesta atividade, vamos utilizar o Scratch para construir uma calculadora que poderá servir para resolver problemas que estejam sendo abordados com os alunos. Exemplo: caso o professor esteja ensinando fatorial de um número natural, pode-se construir uma calculadora para o cálculo do fatorial utilizando o algoritmo a seguir.



Figura 48: Algoritmo exponencial.

1) Construa uma calculadora que resolva um problema abordado em sala de aula.

5 Conclusão

Existem diversas formas de se utilizar a tecnologia no ensino, o Scratch, por ser programável, possibilita criar simulações, no sentido de imitar processos do mundo real e é visualmente atrativo e fácil de usar, sendo assim uma escolha de aplicativo para ser utilizado no ensino de matemática.

Seu uso vai além de apresentar conteúdos importantes do curso, visto que possibilita atividades mais lúdicas que, de forma indireta, ajudam na disciplina.

As atividades aqui propostas apresentam uma forma de ensinar o uso do aplicativo e dos conteúdos matemáticos simultaneamente, isto é importante devido ao já estipulado cronograma das escolas.

Como proposta de continuação deste trabalho, as atividades podem ser levadas para sala de aula, assim, eventuais problemas ou algumas melhorias podem surgir. Questões como o tempo necessário para realizar as atividades e o nível e ano ideal em que o aplicativo seja introduzido também são aspectos que podem ser respondidos uma vez que já tenha sido testado com algum grupo.

Seguindo o exemplo do TCC escrito por Marília Luiza Matte [16], seria interessante também aplicar para alunos de licenciatura em matemática e professores. Desta forma, há um aumento nas chances de tais ferramentas se popularizarem e realmente chegarem a fazer parte do dia a dia escolar, além das possíveis informações e sugestões que podem surgir, visando melhorar as atividades propostas.

O estudo de novas formas de se apresentar alguns conteúdos é fundamental para o bom desenvolvimento do ensino e aprendizado de matemática. O Scratch é uma ferramenta atual com potencial para se explorar esses novos métodos de exposição ao conhecimento e deve ser utilizado não só na matemática, mas sim ser explorado para que se encontre formas úteis de aplicá-lo nas outras disciplinas.

Referências

[1] GIRALDO, V.; MATTOS, F. R. P.; SILVANI P. A. C. Recursos Computacionais no Ensino da Matemática. SBM, 2014, (Coleção PROFMAT) Rio de janeiro.

[2] BRASIL. Ministério da Educação. Diretrizes Curriculares Nacionais para a Educação. Brasília: MEC, SEB, DICEI, 2013. 562p. ISBN: 978-857783-136-4 Disponível em: <u>http://portal.mec.gov.br/index.php?</u> <u>option=com_content&view=article&id=12663&Itemid=1152portal.mec.gov.br/index.php?</u> <u>option=com_docman&task=doc_download&gid=15548&Itemid</u>. Página 25. Acessado em: 14 Maio 2014.

[3] PROINFO. Disponível em: <u>http://www.fnde.gov.br/programas/programa-nacional-de-tecnologia-educacional-proinfo/proinfo-perguntas-frequentes</u> Acessado em: 14 Maio 2014.

[4] Projeto um computador por aluno. Disponível em: <u>http://www.uca.gov.br/institucional/projeto.jsp</u> Acessado em: 14 Maio 2014.

[5] PIRAÍ DIGITAL. Disponível em: <u>http://www.piraidigital.com.br/index.php/component/k2/item/235-inovacao-na-gestao</u> Acessado em: 14 Maio 2014.

[6] Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira. Disponível em: <u>http://portal.inep.gov.br/pisa-programa-internacional-de-avaliacao-de-alunos</u> Acessado em: 15 Maio 2014.

[7] PISA 2015: Draft Mathematics framework. <u>http://www.oecd.org/callsfortenders/Annex</u> <u>%20IC_PISA%202015%20Math%20Framework%20.pdf</u>. Acessado em 30 julho 2014.

[8] LIPS, em: Enciclopédia Britânica. Disponível em: <u>http://www.britannica.com/EBchecked/topic/343293/LISP</u> Acessado em: 14 Maio 2014.

[9] LOGO Foundation. Disponível em: <u>http://el.media.mit.edu/logo-foundation/logo/index.html</u> Acessado em: 30 Julho 2014.

[10] SCRATCH. Disponível em: http://scratch.mit.edu/ Acessado em: 30 Julho 2014.

[11] SCRATCHED Disponível em: <u>http://scratched.media.mit.edu/</u> Acessado em: 30 Julho 2014.

[12] ADAMSON, E. From Polygons to Functions to Orbits to Fractals: A Year of Logo Work With a Mathematically Gifted Student, Logo Foundation, 1993. Disponível em: <u>http://el.media.mit.edu/logo-foundation/pubs/papers/polygons_functions.html</u> Acessado em: 30 Julho 2014.

[13] MIHICH, O. The Weaving Turtle, African Textiles, Logo Foundation, 1993. Disponível em: <u>http://el.media.mit.edu/logo-foundation/pubs/papers/african_textiles.html</u> Acessado em: 30 Julho 2014.

[14] TEMPEL, M. The Turtle Is Dead, Rethinking Logo in the Age of Kid Pix, Logo foundation. Disponível em: <u>http://el.media.mit.edu/logo-Foundation/pubs/logoupdate/v4n1.html#dead</u> Acessado em: 30 Julho 2014 .

[15] TEMPEL, M. Balance, Logo foundation. Disponível em: <u>http://el.media.mit.edu/logo-foundation/pubs/titleindex.html</u> Acessado em: 29 Julho 2014.

[16] MATTE, M. L. Monografia (licenciatura em matemática). A linguagem Logo como possibilidade de aprendizagem em matemática, Porto alegre, Universidade Federal do Rio Grande do Sul, 2012.

[17] PINTO, A. S. Dissertação (Mestrado em estudos da criança), Scratch na aprendizagem da Matemática no1.º Ciclo do Ensino Básico: estudo de caso na resolução de problemas, Portugal, Universidade do Minho, 2010.

[18] BARRIO, M. J. C. Zcartas: operaciones com números enteros. Disponível em: <u>http://scratched.media.mit.edu/resources/zcartas-operaciones-con-n%C3%BAmeros-enteros</u> Acessado em: 10 Julho 2014

[19] MARINHO, S. C. V.; STRUCHINER, M. Estudantes do ensino básico como desenvolvedores de jogos digitais. **Em teia, Revista de Educação Matemática e Tecnológica Iberoamericana**, Pernambuco, vol. 4, número 3, 2013.

Apêndice: Neste Apêndice as atividades aparecem da forma como entregaríamos aos alunos, sem as respostas ou anotações para os professores.

Atividade polígonos regulares

Resumo da atividade

Objetivos: explorar as propriedades dos polígonos regulares e chegar a uma fórmula que relacione o número de lados com os ângulos externos de um polígono regular e também uma fórmula que relacione o número de lados com os ângulos internos.

Pré-requisitos: noção de ângulo, noções básicas de informática.

Material: computador com o programa Scratch previamente instalado ou acesso a internet.

Conceitos explorados: ângulos internos e externos, polígonos regulares e suas propriedades.

Construção de polígonos regulares com Scratch

O quadrado é um exemplo de polígono regular, a palavra polígono vem do grego "poly" que significa muitos e "gon" que significa ângulo, ou seja, um polígono é uma figura fechada que possui muitos ângulos. Um polígono é chamado de regular se todos os seus ângulos internos forem congruentes e se todos os seus lados tiverem a mesma medida.

use a caneta	Desenhará uma linha no percurso que o personagem percorrer.	
mova 50 passos	Fará o personagem andar, para frente, 50 passos.	
gire (1 90 graus	Fará o personagem girar, para direita, 90 graus.	

1) Comandos necessários para construir um quadrado:

Basta juntar estes três blocos para formar um algoritmo e, cada vez que se clicar neles, um lado do quadrado será desenhado. Para que não seja necessário clicar quatro vezes no algoritmo, podemos usar um bloco de repetição que fará com que, ao clicar, ele se repita um determinado número de vezes. No nosso caso, como o quadrado tem quatro lados, queremos quatro repetições. O algoritmo para desenhar um quadrado será, então:



quadrado.

Perceba que, no algoritmo que você criou, o personagem está andando sempre o mesmo número de passos antes de girar – desenhando, assim, cada lado - e girando a mesma quantidade de graus – produzindo, assim, os ângulos.

Para termos certeza de que a figura desenhada é um polígono, precisamos verificar se é fechada. Antes de clicar no algoritmo observe no canto direito superior da sua tela que aparece a posição do personagem em coordenadas, como na imagem:



Figura 2: Posição do personagem.

Se esses números permanecerem iguais antes e depois do algoritmo ser executado, quer dizer que o personagem voltou exatamente para a posição que ele estava, verifique algumas vezes para ter certeza. Caso seja a mesma posição, a figura é fechada. 2) Construção de um triângulo equilátero:

O triângulo regular é chamado de equilátero. Para construí-lo, nós utilizaremos o mesmo algoritmo do quadrado, porém precisamos saber qual ângulo de giro devemos usar e quantas repetições serão necessárias. Como ele possui três lados, serão necessárias três repetições. E o ângulo será de 120 graus, assim o algoritmo será:



Figura 3: Algoritmo do triângulo.

Os polígonos possuem ângulos internos e externos. O interno é o menor ângulo entre dois lados consecutivos (ângulo em vermelho na Figura 4). Prolongando um lado do polígono, o ângulo formado por esse prolongamento e pelo lado do polígono é chamado de externo (ângulo em verde na Figura 4).



Figura 4: Ângulos internos e externos.

Na construção do quadrado e do triângulo, o personagem, ao final de cada lado, em vez de continuar indo para frente, ele vira para a direita (sentido horário), este movimento pode ser melhor percebido na Figura 5; assim, o ângulo que escolhemos para ele virar é o ângulo externo. Perceba também que juntando o ângulo interno e o externo obtemos um ângulo de 180 graus.



Figura 5: Rotação do personagem. A partir das construções acima, responda às seguintes questões no seu caderno.

- 3) Qual a soma dos ângulos externos do quadrado que foi desenhado no Scratch?
- 4) Qual a soma dos ângulos externos do triângulo equilátero desenhado no Scratch?
- 5) Que conclusão você pode tirar ao comparar os resultados encontrados nos itens 3 e 4?
- 6) O que devemos alterar no algoritmo para construir um hexágono? Sabendo que ele possui seis lados e que a soma de seus ângulos externos também será 360 graus.
- 7) E se quisermos construir um polígono de cinco lados?

	Lados	Ângulo externo	Ângulo interno	Soma dos ângulos internos
Triângulo	3			
Quadrado	4			
Pentágono	5			
hexágono	6			

8) Preencha a tabela com os dados os polígonos regulares construídos até agora:

9) Com as informações obtidas, você seria capaz de dizer a soma dos ângulos internos dos seguintes polígonos?

Heptágono (7 lados)

Octógono (8 lados)_____Eneágono (9 lados)_____Decágono (10 lados)_____

- 10) Qual a soma dos ângulos internos de um polígono regular de n lados?
- Com tudo que você descobriu, modifique o algoritmo para desenhar um icoságono (vinte lados).

Atividade construção da circunferência

Resumo da atividade

Objetivo: construir uma aproximação de uma circunferência dado seu raio.

Pré-requisitos: polígonos regulares, circunferência, algoritmo para construir polígonos regulares, conhecimento básico no uso do Scratch.

Material: computador com acesso à internet ou com o programa Scratch previamente instalado, papel e lápis.

Conceitos explorados: polígonos, raio da circunferência e aproximação.

Construindo uma circunferência a partir do raio

O Scratch não possui um bloco ou algoritmo específico para construir uma circunferência. Nesta atividade construiremos uma aproximação de uma circunferência.

1) Utilizando o algoritmo para construir polígonos regulares, da atividade anterior, experimente mudar as repetições para 360, o ângulo para 1 grau e o número de passos também para 1, como na figura:



Figura 1: Algoritmo polígono regular.

2) A figura construída é um polígono ou uma circunferência? Por quê?

Usaremos o polígono regular de 360 lados como uma aproximação da circunferência e para facilitar a compreensão dos próximos passos o chamaremos de circunferência. Lembre que como estamos trabalhando com uma aproximação de uma circunferência os resultados encontrados nas contas possuirão uma margem de erro.

3) Modifique o algoritmo para 180 repetições, observe e anote as coordenadas do personagem antes e depois de executá-lo.



Figura 3: Descobrindo o raio

Na figura 3 podemos ver dois personagens, os quais representam as posições antes e depois da execução do algoritmo.

4) Com os dados anotados do exercício 3, encontre a medida da linha pontilhada entre os personagens da Figura 3. O que esta medida representa na circunferência? Qual a sua relação com o raio?

5) Qual o raio desta circunferência?

Como nosso objetivo é construir uma circunferência dado seu raio, temos que modificar este algoritmo para que ele funcione em função do raio. Nos exercícios anteriores, descobrimos o raio nas condições de girar um ângulo de 1 grau e mover 1 passo, assim temos a relação entre o raio e o lado.

6) Seguindo a proporção entre o lado dado e o raio encontrado no exercício anterior, se o raio for 1, quantos passos o personagem deverá dar? Escreva o resultado com quatro casas decimais, arredondando se necessário.

7) A partir do resultado do exercício anterior construa uma fórmula onde dado o raio, você obtenha quantos passos o personagem deve andar.
(dica: pense no problema 6, mas se o raio fosse X)

8) Crie uma variável chamada "raio" no Scratch e, no lugar do número de passos, utilize a fórmula encontrada acima, de forma que, modificando a variável raio, você obtenha uma circunferência com aproximadamente este raio.

Atividade: Aproximação de π parte I.

Resumo da atividade

Objetivo: construir um polígono inscrito a circunferência a partir do raio da mesma.

Pré-requisitos: trigonometria; propriedades do triângulo retângulo, do triângulo isósceles; conhecimento básico no uso do Scratch como construção de polígonos regulares e variáveis; ter finalizado as atividades: **polígonos regulares** e **construção da circunferência**.

Material: computador com acesso à internet ou com o programa Scratch previamente instalado, papel e lápis.

Conceitos explorados: polígonos regulares inscritos.

Polígono inscrito, circunscrito e aproximação de π Parte 1, polígono inscrito

Polígono inscrito é aquele cujos lados são cordas de uma circunferência de modo que todos os seus vértices são pontos da circunferência. Sabe-se que todo polígono regular pode ser inscrito. Exemplos:



Figura 1: Polígonos regulares inscritos.

Alguns polígonos não regulares também podem ser inscritos. Exemplos:



Figura 2: Polígonos não regulares inscritos.

Nesta atividade trabalharemos somente com os polígonos regulares. Como todo polígono regular pode ser inscrito, basta desenhar o polígono e depois descobrir em qual circunferência ele está inscrito. Nosso objetivo será inscrever qualquer polígono regular em uma circunferência dada.

Partiremos do algoritmo obtido na atividade **construção da circunferência**, com um raio de tamanho 100.



Figura 3 : Algoritmo construção da aproximação de uma circunferência.

Com a circunferência desenhada, já podemos começar a fazer os polígonos. Se apenas juntarmos os blocos do algoritmo para desenhar polígonos regulares com os do algoritmo para construir a circunferêncinão, ele não sairá nem do tamanho desejado e nem dentro da circunferência. Para que ele esteja dentro da circunferência vamos primeiro partir do desenho final já pronto e ver quais movimentos devemos fazer com o personagem. Vamos começar com o triângulo.



Figura 4: Triângulo inscrito.

Observe que, na figura 4, diferente da atividade **polígonos regulares** onde o personagem girava sempre o mesmo ângulo, dividimos um ângulo em dois, no caso, a soma do ângulo girado no primeiro movimento com o do sétimo deve ser igual aos do terceiro e quinto movimento.

 Em seu caderno, mostre que os ângulos que o personagem virará no primeiro e sétimo movimentos indicados na Figura 4 são iguais, sabendo que reta R é tangente a circunferência no vértice A e paralela a BC

Feito o exercício 1, podemos concluir que em cada um dos movimentos citados, se girará um ângulo equivalente a metade do ângulo externo, sabemos também que o ângulo externo pode ser escrito em função do número de lados do polígono. Agora precisamos descobrir quantos passos devemos andar para desenhar o lado do triângulo.

2) Em seu caderno, encontre o lado, chamado de Li, do triângulo inscrito em função do raio, chamado de R, e do ângulo externo.



Figura 5 : Lado em função do raio.

3) O algoritmo para construir o polígono inscrito ficará como na figura 6, monte-o no Scratch completando as lacunas com as informações obtidas acima e escolha um raio e o número de lados do polígono


Figura 6 : Algorítimo incompleto.

Depois de pronto, experimente modificar o raio e o número de lados para testar se está tudo funcionando corretamente.

Atividade Aproximação de π parte II.

Resumo da atividade

Objetivo: construir um polígono circunscrito a circunferência a partir do raio da mesma.

Pré-requisitos: Trigonometria; propriedades do triângulo retângulo, do triângulo isósceles; conhecimento básico no uso do Scratch como construção de polígonos regulares e variáveis; ter finalizado as atividades: **polígonos regulares** e **construção da circunferência**; parte I desta atividade.

Material: computador com acesso à internet ou com o programa Scratch previamente instalado, papel e lápis.

Conceitos explorados: Polígonos regulares inscritos e circunscritos.

Polígono inscrito, circunscrito e aproximação de π Parte II, polígono circunscrito

Nesta parte da atividade, trabalharemos com os polígonos regulares circunscritos. Um polígono é circunscrito a uma circunferência se esta é tangente a todos os lados do polígono (Figura 1).



Figura 1 : Polígono regulares circunscrito.

Como nos polígonos inscritos, todo polígono regular poderá ser circunscrito, e alguns não regulares também poderão (Figura 2).



Figura 2 : Polígonos não regulares circunscritos.

Novamente, trabalharemos somente com os polígonos regulares. O desafio desta atividade será construir o polígono circunscrito começando do mesmo ponto em que o personagem parou na hora de fazer o inscrito, assim, quando executarmos o algoritmo completo, os dois polígonos estarão na mesma circunferência. Relembrando o algoritmo da atividade anterior (Figura 3).



Figura 3: Algoritmo atividade anterior.

Partindo do problema já resolvido, vamos ver quais movimentos o personagem deve fazer.



Figura 4: Triângulo circunscrito passo a passo.

Essa parte é muito parecida com a construção de um triângulo regular, a única diferença é que no primeiro e no último movimento andamos apenas metade do lado, sendo assim os ângulos permanecem iguais. Nesse momento encontramos o mesmo problema dos polígonos inscritos, devemos descobrir qual o tamanho do lado que chamaremos agora de Lc.

1) Em seu caderno, encontre Lc em função do raio e do ângulo externo.



Figura 5: Triângulo circunscrito.

Agora que sabemos o valor do lado em função do raio, só nos resta selecionar os comandos do Scratch para montar a parte que deverá ser encaixada no algoritmo da atividade passada.

2) Complete as lacunas do algoritmo com os valores encontrados.



Figura 6: Algoritmo polígono circunscrito.

Depois que terminar, junte este algoritmo ao criado na parte 1 desta atividade.

Atividade Aproximação de π parte III

Resumo da atividade

Objetivo: obter uma aproximação de π através de um algoritmo que construa, dado uma circunferência e seu raio, um polígono inscrito e um circunscrito com o mesmo número de lados.

Pré-requisitos: perímetro de polígonos, comprimento de circunferências e conhecimento básico no uso do Scratch como construção de polígonos regulares e variáveis, ter finalizado as atividades de **polígonos regulares**, **construção da circunferência** e as partes um e dois desta atividade.

Material: Computador com acesso à internet ou com o programa Scratch previamente instalado, papel e lápis.

Conceitos explorados: Polígonos regulares inscritos e circunscritos, perímetro e método para descobrir o valor de π .

Polígono inscrito, circunscrito e aproximação de π Parte 3, aproximação de π

Nesta etapa da atividade, utilizaremos o algoritmo criado nas duas atividades anteriores para calcular uma aproximação de π . Sabemos que o comprimento da circunferência é $2\pi R$. Se explorarmos um pouco, desenhando polígonos com o número de lados cada vez maior, com o algoritmo construído nas duas primeiras partes desta atividade podemos perceber que eles se aproximam da circunferência. Como nós sabemos o raio da circunferência, podemos dizer que:

$\frac{\text{Perimetro do poligono circunscrito}}{2R} > \pi > \frac{\text{Perimetro do poligono inscrito}}{2R}$

Figura 1: Aproximação de π .

Agora precisamos calcular os perímetros.

1) Crie uma variável chamada "Perímetro do polígono circunscrito" e, utilizando apenas as outras variáveis já existentes, faça um algoritmo que calcule esse perímetro.

2) Crie uma outra variável chamada "Perímetro do polígono inscrito" e faça o mesmo do exercício 1.

3) Crie uma terceira variável chamada "Aproximação de π ". Utilizando os valores encontrados nos exercícios anteriores, faça com que esta variável seja o valor médio entre os extremos da desigualdade da figura 1

Para facilitar a visualização dos resultados na área de criação das variáveis, ao lado de cada variável, há uma caixa para ser marcada, deixe marcado somente as variáveis criadas nos exercícios 1, 2 e 3, conforme Figura 2. Assim os valores dessas variáveis aparecem próximos ao personagem, Figura 3.







Figura 4: Janela de visualização.

4) preencha a tabela abaixo com algumas aproximações de π dizendo quantos lados possui o polígono utilizado nesse cálculo.

Número de lados dos polígonos	Valor aproximado de π
3	3,897
54	

Atividade construção de calculadoras

Resumo da atividade

Objetivo: montar um algoritmo para construir uma calculadora para auxiliar a resolução de problemas.

Material: computador com acesso à internet ou com o programa Scratch previamente instalado, papel e lápis.

Construção de calculadoras

Nesta atividade, vamos utilizar o Scratch para construir uma calculadora que poderá servir para resolver problemas que estejam sendo abordados com os alunos. Exemplo: caso o professor esteja ensinando fatorial de um número natural, pode-se construir uma calculadora para o cálculo do fatorial utilizando o algoritmo a seguir.



Figura 1: Algoritmo de fatorial.

1) Construa uma calculadora que resolva um problema abordado em sala de aula.