O GeoGebra como estratégia para ensino remoto

Criando atividades com *feedback* automático

Celina Aparecida Almeida Pereira Abar
José Manuel Dos Santos Dos Santos
Marcio Vieira de Almeida

Organização:

IGP

INSTITUTO SÃO PAULO GeoGebra

Parceiros:

OEI

P.PORTO

ESCOLA SUPERIOR DE EDUCAÇÃO POLITÉCNICO DO PORTO

1946 • 2021 75 anos PUC-SP

Organizado pelo:

Instituto GeoGebra Internacional de São Paulo
Instituto GeoGebra Portugal

Publicado pela

Organização de Estados Ibero-Americanos para a Educação a Ciência e a Cultura (OEI),
a partir do Escritório de Lisboa

Com as parcerias

Pontifícia Universidade Católica de São Paulo
Instituto Politécnico do Porto - Escola Superior de Educação
InED - Centro de Investigação e Inovação em Educação

Apoios

PIPRINT-PG EDITAL PIPRINT-PG 9302/ 2020
Fundação para a Ciência e a Tecnologia no âmbito do projeto UIDB/05198/2020

Dezembro de 2022

ISBN: 978-989-54789-1-0

O GeoGebra como estratégia para ensino remoto

Criando atividades com feedback automático

O GeoGebra como estratégia para ensino remoto: criando atividades com feedback automático

Editores:. Abar, C., Almeida, M & Dos Santos, J.

Título:

O GeoGebra como estratégia para ensino remoto: Criando atividades com feedback automático

Organização de Estados Ibero-Americanos para a Educação a Ciência e a Cultura (OEI)—Bravo Murillo, 38 28015 Madrid, Espanha https://oei.int

Escritório de Lisboa, Palácio das Laranjeiras, Estrada das laranjeiras 205, 1649-018 Lisboa, Portugal https://oei.int/pt/escritorios/portugal

Editores:

Celina Aparecida Almeida Pereira Abar

José Manuel Dos Santos Dos Santos

Marcio Vieira de Almeida

Autores:

Astrigilda Pires Rocha Silveira

Arlindo Tavares Semedo da Veiga

Alexandre Emanuel Batista Trocado

Celina Aparecida Almeida Pereira Abar

Cristina da Silva Ferreira Alves

Diogo Meurer de Souza Castro

Idalinda Pereira da Cunha

Ilda Marisa de Sá Reis

José Manuel Dos Santos Dos Santos

Marcio Vieira de Almeida

Rui Josué Boucinha Torres Eusébio

ISBN: 978-989-54789-1-0

O GeoGebra como estratégia para ensino remoto: criando atividades com feedback automático

Editores:. Abar, C., Almeida, M & Dos Santos, J.

Organização:

Parceiros:

Apoios:

Índice

Índice de Figuras

Índice de Tabelas

O GeoGebra como estratégia para ensino remoto: criando atividades com feedback automático

Editores:. Abar, C., Almeida, M & Dos Santos, J.

Prefácio

José Manuel Dos Santos Dos Santos, Márcio Vieira de Almeida, Celina Abar

Neste livro apresentamos os resultados do trabalho do projeto de investigação "O GEOGEBRA COMO ESTRATÉGIA PARA ENSINO REMOTO: CRIANDO ATIVIDADES COM FEEDBACK AUTOMÁTICO", aprovado no âmbito da Pontifícia Universidade Católica de São Paulo, Brasil atendendo ao edital PIPRINT de 2021. No desenvolvimento dessa pesquisa foi estudado o uso do GeoGebra na criação de recursos, por professores de diferentes países como Brasil, Portugal, Cabo Verde e Angola, para o ensino da matemática em diferentes vertentes: adaptando ou criando materiais, ajustados aos interesses, às necessidades e aos problemas que enfrentam os professores nas escolas, investigando o uso destes materiais nos contextos escolares e o seu efeito na melhoria dos resultados dos estudantes e criando possibilidades de feedback imediato em suas propostas de avaliação.

O projeto responde à solicitação de uma das linhas de pesquisa do Programa de Estudos Pós-Graduados em Educação Matemática, Tecnologias da Informação e Educação Matemática no contexto do grupo de pesquisa da autora e proponente: Tecnologias e Meios de Expressão em Matemática-TECMEM e tem como proposta oferecer a professores da escola básica, preferencialmente de escolas públicas, uma formação para sua atuação no contexto do ensino remoto com a utilização do software GeoGebra não apenas como mais um recurso tecnológico, mas como um recurso que colabore no desenvolvimento da prática docente, envolvendo conceitos matemáticos e métodos de avaliação automática.

A investigação foi desenvolvida em parceria com o Instituto GeoGebra de Portugal - IGP, nomeadamente no apoio à formação que é dada a vários níveis e com as parcerias que o IGP já estabeleceu com Países Africanos de língua oficial portuguesa como Cabo Verde e Angola, e será mais uma valia para o aprofundamento da investigação conduzida sobre os programas de formação em desenvolvimento.

Deste modo, este livro, cujos autores são coordenadores de Institutos GeoGebra no Brasil e em Portugal, tem como objetivo apresentar os resultados obtidos no desenvolvimento do projeto, disponibilizando os recursos criados pelos professores participantes de modo a serem inspiradores para que outros docentes os utilizem na sua prática e sejam autores de suas próprias propostas.

Este e-book contém onze capítulos. No Capítulo I são apresentados os contributos teóricos para a conceção de tarefas com *feedback* automático no GeoGebra que permitem refletir sobre a importância em se considerar os "erros" dos alunos como indicação de oportunidades para sua aprendizagem. Além disso, são apresentados os procedimentos realizados para a execução do projeto com professores participantes do Brasil, Portugal, Cabo Verde e Angola. O Capítulo II, desenvolvido pelos autores, está dedicado a exemplos de construções de atividades ressaltando comandos no GeoGebra, muitas vezes desconhecidos pelos participantes, e que permitiram aprimoramentos nas atividades que foram criadas. A partir do Capítulo III até o Capítulo X, são dedicados, cada um deles, às propostas dos participantes com diferentes conteúdos da matemática, trazendo uma breve explicação da respetiva aplicação, seu enquadramento e descrição além de um guião da construção realizada no GeoGebra. Finalizamos com o Capítulo XI das conclusões, as referências e notas bibliográficas dos autores.

Os autores esperam que este e-book possa inspirar outros professores a refletirem sobre a importância de um *feedback* automático que possa orientar os alunos na construção do seu conhecimento e construam suas próprias tarefas.

Bom trabalho!!

Capítulo I – Introdução e contributos teóricos para a concepção de tarefas com *feedback* automático no GeoGebra

José Manuel Dos Santos Dos Santos, Márcio Vieira de Almeida, Celina Abar

Os passos iniciais no desenvolvimento do projeto "O GEOGEBRA COMO ESTRATÉGIA PARA ENSINO REMOTO: CRIANDO ATIVIDADES COM FEEDBACK AUTOMÁTICO" procurou colaborar para a inserção da tecnologia na prática docente, aprimorando os estudos e as análises no que diz respeito à tecnologia no contexto da educação matemática.

Com as mudanças tecnológicas no ensino, presentes na prática docente em tempos de covid19 e, ainda, constatando o ensino híbrido em algumas escolas, interessa, também, estudar de que modo os recursos para a atuação do professor de matemática podem adaptar-se à sua prática docente nestes contextos.

O recurso tecnológico principal utilizado no desenvolvimento da pesquisa foi o GeoGebra, um instrumento importante para a prática do professor, pois na interação com o *software* ocorrem a reorganização e a modificação dos esquemas de utilização desse recurso, transformando-o de artefato em instrumento (Rabardel, 1995) e possibilitando sua inclusão na propostas de aula do professor, configurando fatos que permitem a estruturação da suas ações, colaborando para sua formação e aprimoramento de conceitos matemáticos.

O desenvolvimento do projeto passou por momentos síncronos e assíncronos, trabalhados ao longo de um ano, considerando inicialmente a apresentação do *software* de matemática dinâmica GeoGebra, no qual foram revistas as múltiplas aplicações que possui nos diversos temas da Matemática.

As aplicações do GeoGebra foram sempre discutidas do ponto de vista do ensino e aprendizagem da Matemática e, sempre que possível, estas aplicações foram enquadradas a diferentes níveis de escolaridade. Desta forma os trabalhos realizados nos encontros com os professores são uma importante oportunidade de interlocução, por parte das ações que foram ou serão implementadas.

Todos os momentos realizados seguiram um modelo de formação tanto com aspetos teóricos da educação matemática e *feedback* automático, para suporte nas ações desenvolvidas, como no contexto do GeoGebra considerando diferentes tópicos da Matemática. Tais contributos teóricos e técnicos serviram de reflexão para a construção das atividades com *feedback* automático, pretendendo dar resposta à indagação: nos respetivos contextos, no qual os participantes desenvolvem a sua prática docente, que tipo de atividades com recurso ao *feedback* automático poderão ser desenvolvidas com o GeoGebra?

Podemos considerar outras questões como: Que interpretação faz o professor diante dos possíveis erros dos alunos num contexto matemático? Como intervém, o que pedirá aos alunos e que *feedbacks* irá considerar? Para que serve o *feedback* automático, em que momentos intervém no processo de aprendizagem, sob que formas?

Do ponto de vista do suporte teórico para a construção das atividades, sete princípios de boas práticas de *feedback* procuraram nortear os trabalhos dos participantes indicando que nesse contexto é importante considerar que um *feedback* deve:

> 1. ajudar a esclarecer o bom desempenho (metas, critérios, padrões esperados);
>
> 2. facilitar o desenvolvimento da autoavaliação (reflexão) na aprendizagem;
>
> 3. fornecer informações de alta qualidade aos alunos sobre a sua aprendizagem;
>
> 4. incentivar o diálogo entre professores e pares em torno da aprendizagem;
>
> 5. incentivar crenças motivacionais positivas e autoestima;
>
> 6. oferecer oportunidades para diminuir a distância entre o desempenho atual e o desejado;
>
> 7. fornecer informações aos professores que podem ser usadas para ajudar a moldar o ensino (retroalimentação).
>
> (Nicol & Macfarlane-Dick, 2006).

Compreendendo *feedback* como toda informação pós-resposta que é fornecida a um aluno para informá-lo sobre seu estado real de aprendizagem ou desempenho e considerando que a exploração da matemática com o GeoGebra, entre outras possibilidades, permite a criação de atividades com *feedback* automático, estas propostas, considerando os princípios acima expostos, podem ter como subjacente conceções da Teoria das Situações Didáticas-TSD criada e desenvolvida por Guy Brousseau, educador matemático francês que, em 2003, recebeu a medalha Felix Klein pelo desenvolvimento da sua teoria.

Podemos considerar um *feedback* automático, construído no GeoGebra, como uma situação adidática, de acordo com Brousseau, e planeada visando uma aprendizagem pois, ao realizar atividades com *feedback* automático, o aluno vivencia momentos adidáticos, favorecendo o processo de construção de conhecimento.

Há que se considerar possíveis erros apresentados pelos alunos e identificados para um *feedback* construtivo. Segundo Brousseau (1983), o erro é a expressão, ou a manifestação explícita, de um conjunto de perceções espontâneas, ou reconstruídas, que, integradas numa rede coerente de representações cognitivas, tornam-se obstáculo à aquisição e ao domínio de novos conceitos.

Deste modo, o professor precisa intervir efetivamente na aprendizagem dos alunos, visando a superação desses obstáculos, como proposto no projeto, por meio da tecnologia, com a utilização dos recursos do GeoGebra, procurando garantir as condições e meios pedagógico-didáticos para que os alunos sejam estimulados nos seus estudos e compreendam os erros cometidos.

Brousseau (2008) apresenta uma situação como:

> O modelo de interação de um sujeito com um meio determinado. O recurso de que esse sujeito dispõe para alcançar ou conservar, nesse meio, um estado favorável é um leque de decisões que dependem do emprego de um conhecimento específico. (Brousseau,2008, p. 21).

Brousseau apresenta uma situação matemática, como todas aquelas situações que levam o aluno a uma atividade matemática sem a intervenção do professor. (Brousseau, 2008, p. 21).

Uma situação adidatica caracteriza-se essencialmente pelo fato de representar determinados momentos do processo de aprendizagem nos quais o aluno trabalha de maneira independente, não sofrendo nenhum tipo de controle direto do professor relativamente ao conteúdo matemático em jogo. (Freitas, 2008, p. 84).

Na situação adidática, o professor deve proceder de forma a não dar a resposta ao aluno, que aprende adaptando-se a um meio, no qual o professor provoque as adaptações desejadas.

Freitas (2008, p. 86) afirma que "*as situações adidáticas representam os momentos mais importantes da aprendizagem, pois o sucesso do aluno nelas significa que ele, por seu próprio mérito, conseguiu sintetizar algum conhecimento*".

Podemos considerar um *feedback* automático, construído no GeoGebra, e pela luz da teoria TSD de Brousseau como uma situação adidática, planeada, visando uma aprendizagem. Ao realizar tais atividades, o aluno vivencia momentos que podem favorecer o processo de construção de conhecimento.

Deste modo, o professor precisa intervir efetivamente na aprendizagem dos alunos visando a superação desses obstáculos. De acordo com esta teoria o papel do professor não se limita a simples comunicação de um conhecimento, mas à devolução de um bom problema.

A devolução aqui tem o significado de transferência de responsabilidade, uma atividade na qual o professor, além de comunicar o enunciado, procura agir de tal forma que o aluno aceite o desafio de resolvê-lo como se o problema fosse seu, e não somente porque o professor quer. Se o aluno toma para si a convicção de sua necessidade de resolução do problema, ou seja, se ele aceita participar desse desafio intelectual e se ele consegue sucesso nesse seu empreendimento, então inicia-se o processo da aprendizagem. (Freitas, 2008, p.83).

Numa perspetiva epistemológica há a necessidade de identificação de obstáculos pelos professores, pois, a partir dessa constatação, poderão organizar propostas específicas que possibilitem a superação desses. Tal fato foi discutido na elaboração das propostas dos participantes que serão disponibilizadas neste ebook.

Igliori (2008) considera que os mecanismos produtores de obstáculos são também produtores de conhecimentos novos e fatores de progresso, fazendo uso do conceito de desequilíbrio desenvolvido por Piaget.

Brousseau (2008) considera que:

> As conceções atuais do ensino exigirão do professor que provoque no aluno – por meio da seleção sensata dos 'problemas' que propõe – as adaptações desejadas. Tais problemas, escolhidos de modo que o estudante os possa aceitar, devem fazer, pela própria dinâmica, que o aluno atue, fale, reflita e evolua. (Brousseau, 2008, p. 34-35).

Numa situação em que o aluno trabalha de maneira independente, ele toma o problema como se fosse seu e essa atitude foi chamada por Brousseau (2008) de devolução. A partir do momento em que ocorre a devolução, pode-se dizer que fica caracterizada uma situação adidática (Freitas, 2008). Cabe ao professor criar meios e desafiar os alunos de tal forma que estes aceitem o problema como seu.

A devolução é o ato pelo qual o professor faz com que o aluno aceite a responsabilidade de uma situação adidática de aprendizagem. Nesse caso, devolver as soluções depende da motivação do aluno, o que se relaciona com as questões psicoafetivas.

Brousseau associa a sua teoria a quatro vertentes norteadoras: ação, formulação, validação e institucionalização.

Ação: *momento da tomada de decisões por parte do aluno, os saberes são colocados em prática com o objetivo de resolver os problemas propostos.* Gera uma interação entre os alunos e o meio físico. Os alunos devem tomar as decisões que faltam para organizar sua atividade de resolução do problema formulado.

Formulação: *o conhecimento implícito é transformado em explícito, as estratégias usadas são explicadas.* O objetivo é a comunicação de informações entre alunos. Para isto, devem modificar a linguagem que utilizam habitualmente, precisando-a e adequando-a às informações que devem comunicar.

Nas ações acima é o que se espera numa proposta de *feedback* automático.

Validação: *a estratégia apresentada precisa ser provada dentro de um determinado contexto, n*a qual se tenta convencer um ou vários interlocutores da validade das afirmações que são feitas. Neste caso, os alunos devem elaborar provas para demonstrá-las. Num *feedback* automático é o momento no qual o aluno ultrapassou e venceu os seus obstáculos.

Institucionalização: *ocorre a validação da atitude matemática.* É um resumo de todo o processo que foi construído durante o trabalho e o professor faz uma recuperação de tudo que foi realizado e sistematiza esse saber. Nestas situações procura-se que o conjunto de alunos de uma aula assuma o significado socialmente estabelecido de um saber que foi elaborado por eles mesmos, em situações de ação, de formulação e de validação.

Brousseau (2008) relata nos seus estudos a existência de um sistema didático composto pelo professor, aluno e pelo objeto de conhecimento em questão: *"Em síntese, trata-se de colocar os alunos diante de uma situação que evolua de forma tal, que o conhecimento que se quer que aprendam seja o único meio eficaz para controlar tal situação"* (p. 33).

Compreendemos que a TSD pode criar uma visão diferenciada sobre o erro no sentido de considerá-lo apenas como um obstáculo e caminhar para a obtenção do saber, apresentando condições que devem ser consideradas numa proposta de *feedback* automático.

A dinâmica do trabalho caracterizou-se por uma participação ativa dos professores, alicerçada e estimulada pela associação entre prática e teoria e a manipulação e análise de situações-problema. Nas sessões de trabalho os participantes tiveram a oportunidade de aprender e aprofundar as suas competências com a utilização do GeoGebra, também com apoio dos autores em situações que envolviam comandos necessários e específicos para o desenvolvimento dos recursos.

Numa primeira fase foram abordadas as ferramentas, os comandos e as interfaces necessárias e algumas tarefas apresentadas pelos coordenadores do projeto. Seguidamente, houve um trabalho tutelado pelos coordenadores do projeto capacitando os professores para a criação de aplicações do GeoGebra em diversas áreas, que paulatinamente se converteu num trabalho de partilha em que os participantes trouxeram novas questões e abordagens técnicas relevantes para o estudo que aqui se apresenta.

Numa segunda fase houve a discussão, sobre as atividades propostas, focando-se nas implicações conceptuais, teóricas e metodológicas destas tarefas do ponto de vista do ensino e da aprendizagem da matemática e as implicações do *feedback* imediato.

A discussão das implicações de algumas atividades no campo da investigação educacional, bem como, na pesquisa matemática, não foi negligenciada em virtude do trabalho empírico que se sucedeu à aplicação de algumas tarefas no contexto de sala de aula, as quais colocaram em evidência as potencialidades do GeoGebra na criação de novo conhecimento científico didático para o professor e a educação matemática em geral.

Assim, neste e-book, diferentes propostas são apresentadas para inspirar outros professores a construírem as suas próprias propostas. Em cada uma delas está, em detalhes, o autor ou autores, o conteúdo matemático envolvido, objetivo, um fluxograma que guiou os passos da construção no GeoGebra e a indicação da própria atividade.

As propostas estão direcionadas para diferentes escolaridades e algumas delas foram aplicadas pelos autores durante a docência na escola. Cada um dos participantes avaliou também a relevância de projetos dessa natureza, a respetiva participação e a contribuição dada à sua formação em depoimentos que serão disponibilizados num número temático da Revista Internacional do Instituto GeoGebra de São Paulo em 2023.

O GeoGebra como estratégia para ensino remoto: criando atividades com feedback automático

Editores:. Abar, C., Almeida, M & Dos Santos, J.

Capítulo II – As tarefas iniciais do projeto

José Manuel Dos Santos Dos Santos, Márcio Vieira de Almeida, Celina Abar

As tarefas produzidas no projeto partiram de propostas apresentadas pela equipa de coordenação do projeto. Com elas pretendeu-se ilustrar um conjunto de técnicas que podiam ser usadas para o desenvolvimento das propostas pensadas pelos participantes.

Ao pretender-se utilizar o GeoGebra para desenvolver tarefas com feedback automático torna-se necessário dominar uma série de procedimentos no uso do *software*. Para o desenvolvimento destas destrezas pensou-se em trabalhar com os participantes do projeto a partir de exemplos preparados pela equipa de formação.

A primeira tarefa proposta*, Construção de uma atividade com autoavaliação com funções reais de variável real*, partiu de um exemplo com famílias de funções, onde se pretendia desenvolver o reconhecimento de representações algébricas e gráficas de famílias funções reais de variáveis reais. Em relação ao uso do GeoGebra para fornecer *feedback* automático, usou-se um *feedback* certificativo, neste sentido são utilizadas caixas de texto ou imagens que se mostram ou escondem em função de determinadas condições, que dependem de parâmetros relacionados com elementos da família. Também foi criada uma variável pontuação que se altera ao longo do desenvolvimento da interação do utilizador com a aplicação. Finalmente esta tarefa também contribui com a indicação de como se pode introduzir *feedback* usando mensagens previamente gravadas em áudio. Apesar desta tarefa ter uma trajetória hipotética de aprendizagem associada, bem como de um diagrama de processo associado, foi opção da equipa de coordenação de não explicitar estes detalhes na primeira fase de trabalho no projeto onde se procurava um entendimento comum dos participantes sobre os passos de programação para a construção deste tipo de aplicações com o GeoGebra.

Depois da apresentação e discussão da primeira tarefa os participantes foram convidados a realizá-la e a criarem as suas próprias propostas, que de restam constam dos capítulos que se seguem. Foi a partir das propostas dos participantes que foi introduzida a necessidade de elaborar esquemas que evidenciassem o conhecimento matemático e pedagógico inerente às propostas, bem como os diagramas de processo que suportam a programação necessária na construção da aplicação em GeoGebra. Em resumo, cada proposta englobava uma série de decisões do autor sobre estratégias pedagógicas e curriculares inerentes à proposta e, também, sobre as estratégias de programação em GeoGebra que iriam utilizar.

Na tarefa apresentada aos participantes do projeto, *Condição de existência de um triângulo*, seguem os passos da primeira tarefa, agora incluindo o diagrama de processo e a possibilidade de existir a modelação e visualização geométrica do assunto em estudo. Também foi introduzida a utilização de protocolo da construção para facilitar e apresentar uma forma de evidenciar os passos a seguir, no sentido de poder contribuir para a adaptação ou o desenvolvimento de tarefas semelhantes a outros interessados.

Finalmente a terceira tarefa disponibilizada, destinou-se a apresentar os rudimentos de HTML e JavaScript que poderia ser interessante para o desenvolvimento de propostas mais complexas, elaboradas em páginas de Web autónomas ou que carecessem de intercomunicação de mais do que uma aplicação de GeoGebra numa mesma página de Web.

De seguida se apresentam nas seções deste capítulo, as três tarefas acima mencionadas, descrevendo os detalhes necessários à construção das aplicações de GeoGebra. Contudo interessará referir que o impacto destas tarefas nas propostas apresentadas pelos participantes do projeto foi diverso. Em relação às duas primeiras tarefas propostas neste capítulo podemos dizer que elas tiveram impacto nas propostas apresentadas pelos participantes no projeto. Em relação à terceira tarefa, ela teve interesse no caso de propostas que utilizaram programação em JavaScript nas aplicações de GeoGebra.

1ª Tarefa: Construção de uma atividade de autoavaliação com funções reais de variável real.

A primeira tarefa do projeto de formação foi a apresentação de uma aplicação de GeoGebra que permitisse dar *feedback* ao utilizador sobre as suas respostas a três questões colocadas.

Para além do *feedback* esta tarefa proporcionava a apresentação de pontuação das respostas.

Breve explicação da aplicação - Funções e suas transformações.

Nesta atividade pretendemos que o aluno seja capaz de identificar qual a família de funções que pertence uma determinada função, observando apenas o seu gráfico cartesiano, bem como as transformações que sofrem. A aplicação deverá ficar com um aspeto semelhante ao da Figura 1.

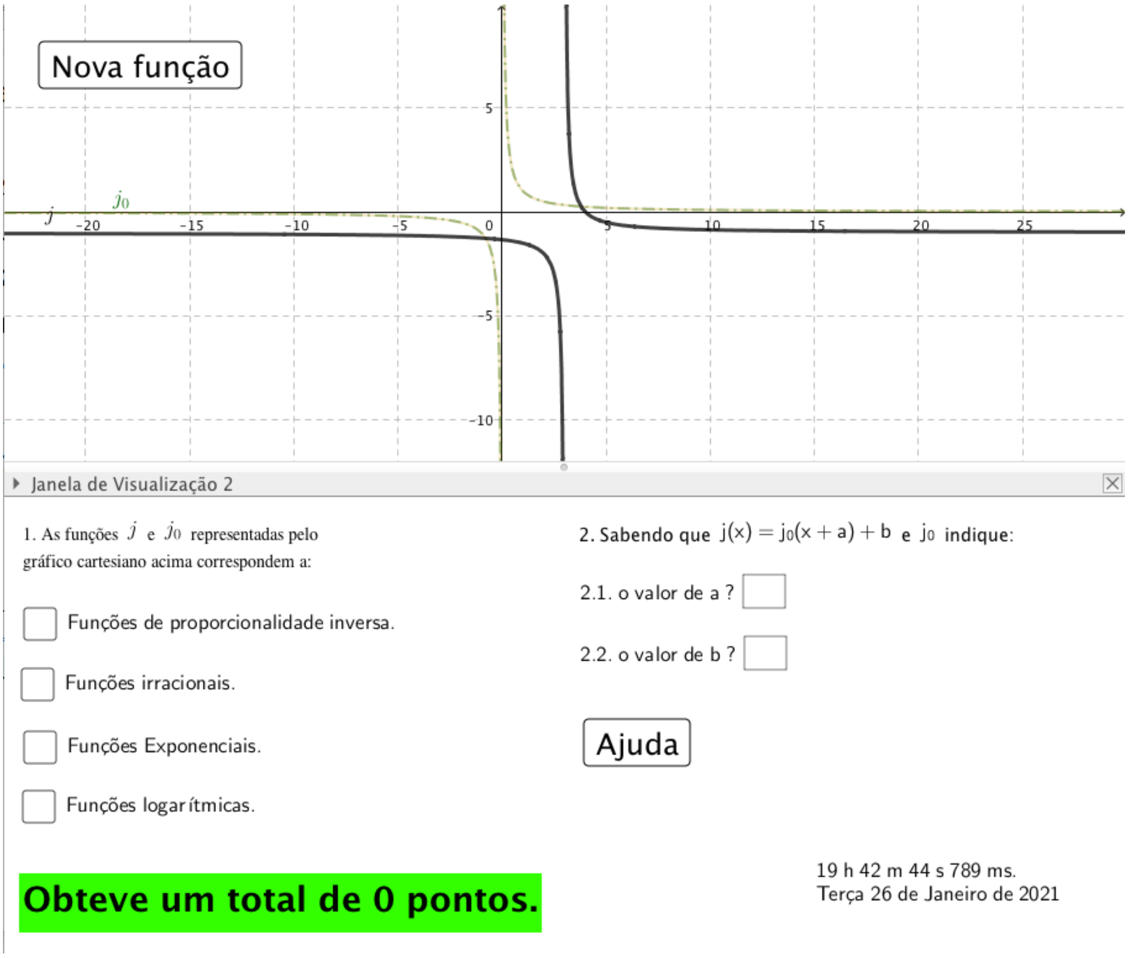


Figura 1 - Imagem final da aplicação "Transformações de funções reais de variável real.".

Para a realização desta aplicação procedemos usando os passos apresentados na seção seguinte.

Guião de construção da aplicação

1) Abrimos um novo arquivo de GeoGebra, onde na Janela de Visualização teremos a malha e os eixos ativados. Também ativamos a Janela de Visualização 2, que incorporamos na parte inferior da malha, mas com os eixos ocultos.

2) Na Entrada criamos três listas e um parâmetro que escolhe um valor da respectiva lista:

a) **la= {-4, -3, -2, -1, 1, 2, 3, 4}**

a= EscolherElementoAleatoriamente(la), que transforma o gráfico cartesiano da função no sentido horizontal;

b) **lb= {-4, -3, -2, -1, 1, 2, 3, 4}**

b= EscolherElementoAleatoriamente(lb), que transforma o gráfico cartesiano da função no sentido vertical.

c) **lc = {1, 2, 3, 4}**

c= EscolherElementoAleatoriamente(lc), valor aleatório que irá escolher entre as funções.

3) Vamos criar 8 funções na Janela de Visualização 1. Quatro delas ficarão sempre invisíveis, tendo uma outra associada que só será visível em função do valor aleatoriamente escolhido. A forma de apresentação dos gráficos cartesianos da função deve ser ajustada com cores diferentes e uma certa espessura. Damos a cada um dos quatro pares de funções, uma cor diferente e uma espessura de **5** na aba ***Estilo***. As funções serão:

i) de Proporcionalidade Inversa, na Entrada vamos escrever

f_i(x)=b+(1/(x-a)), com legenda **f_i** que pode ser inserida na aba 'Básico' encontrada nas propriedades desse objeto;

f=Simplificar(Se(c ≟ 1, f_i(x + a) - b)), com legenda **j_0**;

ii) Irracional, na Entrada vamos escrever

g_i(x)=b+sqrt(x-a), com legenda **g_i**;

g=Simplificar(Se(c ≟ 2, g_i(x + a) - b)), com legenda **j_0;**

iii) Exponencial de base 2, na Entrada vamos escrever

h_i(x)=b+2^(x-a), com legenda **h_i**;

h=Simplificar(Se(c ≟ 3, h_i(x + a) - b)), com legenda **j_0;**

iv) Logarítmica de base decimal, na Entrada vamos escrever

i_i(x)=b+lg(x-a) , com legenda **i_i;**

i=Simplificar(Se(c ≟ 4, i_i(x + a) - b)), com legenda **j_0**.

Escondemos todas as 4 funções.

4) Definimos agora a função **j(x)** que será a que o aluno terá de estudar. Esta poderá ser qualquer uma das quatro anteriores que definimos. Para definir esta função escreveremos na entrada:

j(x)= Se(c ≟ 1, f_i, Se(c ≟ 2, g_i, Se(c ≟ 3, h_i, Se(c ≟ 4, i_i)))) , com legenda **j**.

Observação: o sinal ≟ corresponde ao == e que corresponde a testar se o elemento da direita coincide com o elemento da esquerda.

Também damos a esta função uma cor e espessura distinta.

Recordamos que pode utilizar o control+c para copiar e control+v para colar, as partes de código sugeridas neste protocolo.

5) Na Janela de Visualização 1, vamos criar um botão, com auxílio da ferramenta respetiva, OK, que será o início da tarefa para o utilizador. O seu conteúdo é o seguinte:

Legenda:

"Nova Função".

Aceda às propriedades do botão e na aba Programação, inserir o seguinte código de GeoGebra na aba "Ao Clicar":

a=EscolherElementoAleatoriamente(la)

b=EscolherElementoAleatoriamente(lb)

c=EscolherElementoAleatoriamente(lc)

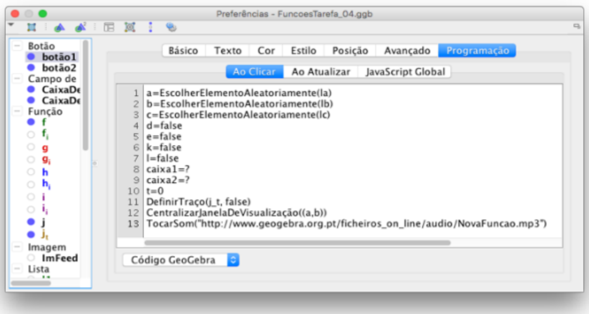


Figura 2 - Menu preferências do botão.

Observação: Na imagem da Figura 2, para além de observar as três linhas de código, acima apresentadas, podemos ver outras linhas de código da aplicação final. Em devido tempo, vamos voltar a este menu no decorrer da construção desta aplicação, mas acautele a transcrição de todo *script* como se refere no ponto 15).

6) Vamos ativar a Janela de Visualização 2. Na nossa aplicação vamos ter três textos designados por *texto*1; *texto2* e *texto3*. Neste momento vamos introduzir o *texto1*. Para isso utilize a ferramenta **Texto**, ABC, para escrever, na *Janela de Visualização 2*, a primeira tarefa, que será de escolha múltipla, associada à aplicação:

“1. As funções j e $ j_0$ representadas pelo gráfico cartesiano acima correspondem a:”

Pode usar a ferramenta de introdução de texto para este efeito, proceda selecionando a opção ***Fórmula LaTex***, como pode ver na Figura 3.

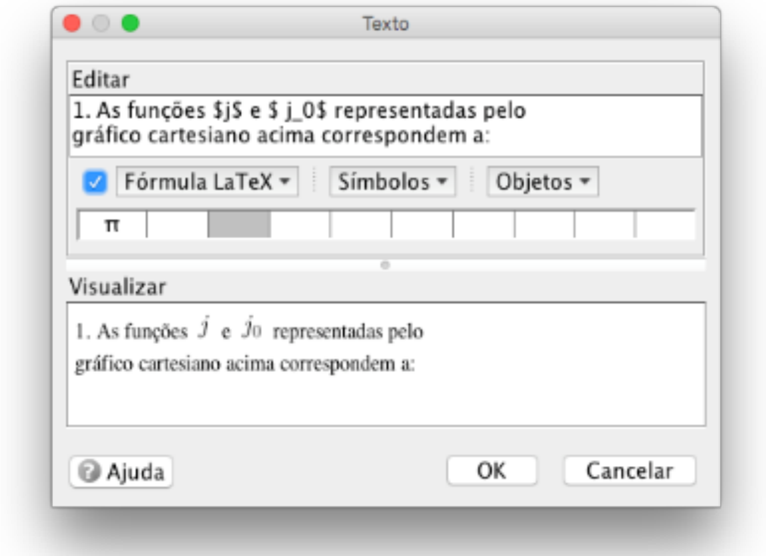


Figura 3 - Menu para entrada do texto.

7) Vamos criar 4 ***Caixas para Exibir / Esconder Objetos***, *d*, *e*, *k* e *l*, por exemplo, usando a ferramenta . As 4 ficarão inativas, mas contendo os textos seguintes:

a) **$Funções\;de\;proporcionalidade\;inversa.$** , para a caixa *d* ;

b) **$Funções\;irracionais.$** , para a caixa *e*;

c) **$Funções\;Exponenciais.$** , para a caixa *k*;

d) **$Funções\;logarítmicas.$** , para a caixa *l*.

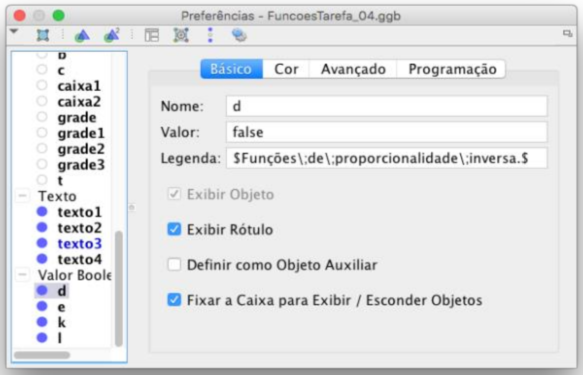


Figura 4 - Exemplo de como fica a janela de preferências da Caixas para Exibir.

No botão 'Nova Função' na Ka, que atualiza a construção e exibe uma nova função, as variáveis booleanas *d*, *e*, *k* e *l* devem ser atualizadas como falsas; inserindo o **Código de GeoGebra** no evento **Ao Clicar**, por exemplo "d=false", nas propriedades do botão, na aba Programação. Veja passo 5) e Figura 2.

8) Vamos construir as variáveis relacionadas com a classificação do trabalho do aluno. Como vamos ter três questões usaremos as variáveis *grade1*, *grade2* e *grade3*, para registo dos resultados parcelares e a variável *grade* para ter a soma das classificações. Como até agora só temos a primeira questão, cuja resposta correta depende da escolha acertada de uma das quatro variáveis *d*, *e*, *k* e *l*, construídas no passo anterior, a variável *grade1*, valerá 0 ou 4, obtendo-se digitando na Entrada o seguinte comando:

$grade1=Se(c \stackrel{?}{=} 1 \land d, 4, Se(c \stackrel{?}{=} 2 \land e, 4, Se(c \stackrel{?}{=} 3 \land k, 4, Se(c \stackrel{?}{=} 4 \land l, 4, 0))))$

$grade2=Se(caixa1 \stackrel{?}{=} -a, 3, 0)$

$grade3=Se(caixa2 \stackrel{?}{=} b, 3, 0)$

$grade=grade1 + grade2 + grade3$

Recorde que a variável c é um número inteiro entre 1 e 4, e que as variáveis d, e, k e *l são variáveis booleanas assumindo o valor "true" ou "false".*

O GeoGebra permite reproduzir um som de um arquivo mp3 alojado num servidor, usando o comando *TocarSom(<url do ficheiro>)*. Assim, no caso de o utilizador obter a primeira resposta certa vamos dar um feedback auditivo de que obteve 4 pontos, pelo que na janela de **Preferências**, na aba de **Programação** no evento **Ao Atualizar**, vamos introduzir o **Código GeoGebra**:

Se(grade1==4 ∧ grade<7, TocarSom("http://www.geogebra.org.pt/ficheiros_on_line/audio/Obteve4.mp3"))

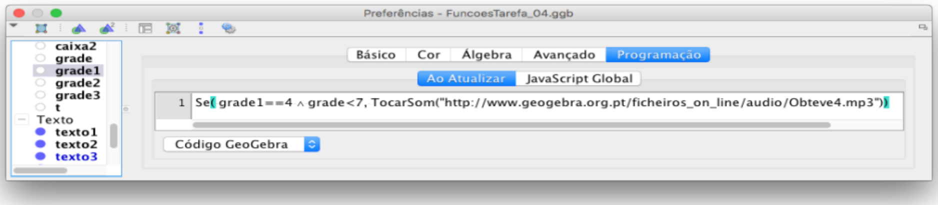


Figura 5 - Introdução de código para feedback auditivo.

9) Utilize a ferramenta **Texto**, ABC,para escrever, na Janela de Visualização 2, a segunda tarefa associada a aplicação:

"2. Sabendo que $j(x)= j_0(x+a)+b$ e $ j_0$ indique:"para o texto2.

Observe que este procedimento é semelhante ao realizado no passo 6).

10) Para questionar o utilizador dos parâmetros associados entre a função **j** e **j_0** que lhe será exibida vamos necessitar de duas caixas de entrada de texto e duas variáveis associadas.

a) As variáveis serão *caixa1* e *caixa2*. Para obtê-las basta digitar na ***Entrada*** cada um dos comandos **caixa1=0** e **caixa2=0**. Em seguida altere o valor destas variáveis, nas propriedades, para **?**

Neste ponto vamos usar, mais uma vez, a programação para dar um feedback auditivo ao utilizador colocando uma mensagem sonora correspondente aos três pontos em cada uma das alíneas da segunda questão.

Na variável caixa1:

Se(caixa1 ≟ -a, TocarSom("http://www.geogebra.org.pt/ficheiros_on_line/audio/Obteve3.mp3"))

Na variável caixa2:

Se(caixa2 ≟ b, TocarSom("http://www.geogebra.org.pt/ficheiros_on_line/audio/Obteve3.mp3"))

b) Agora vamos criar dois **Campos de entrada** de texto usando a ferramenta a=1. Cada um destes campos está vinculado a uma das variáveis definidas no passo anterior. As propriedades destes **Campos de entrada** devem ser semelhantes às que se indicam na Figura 6.

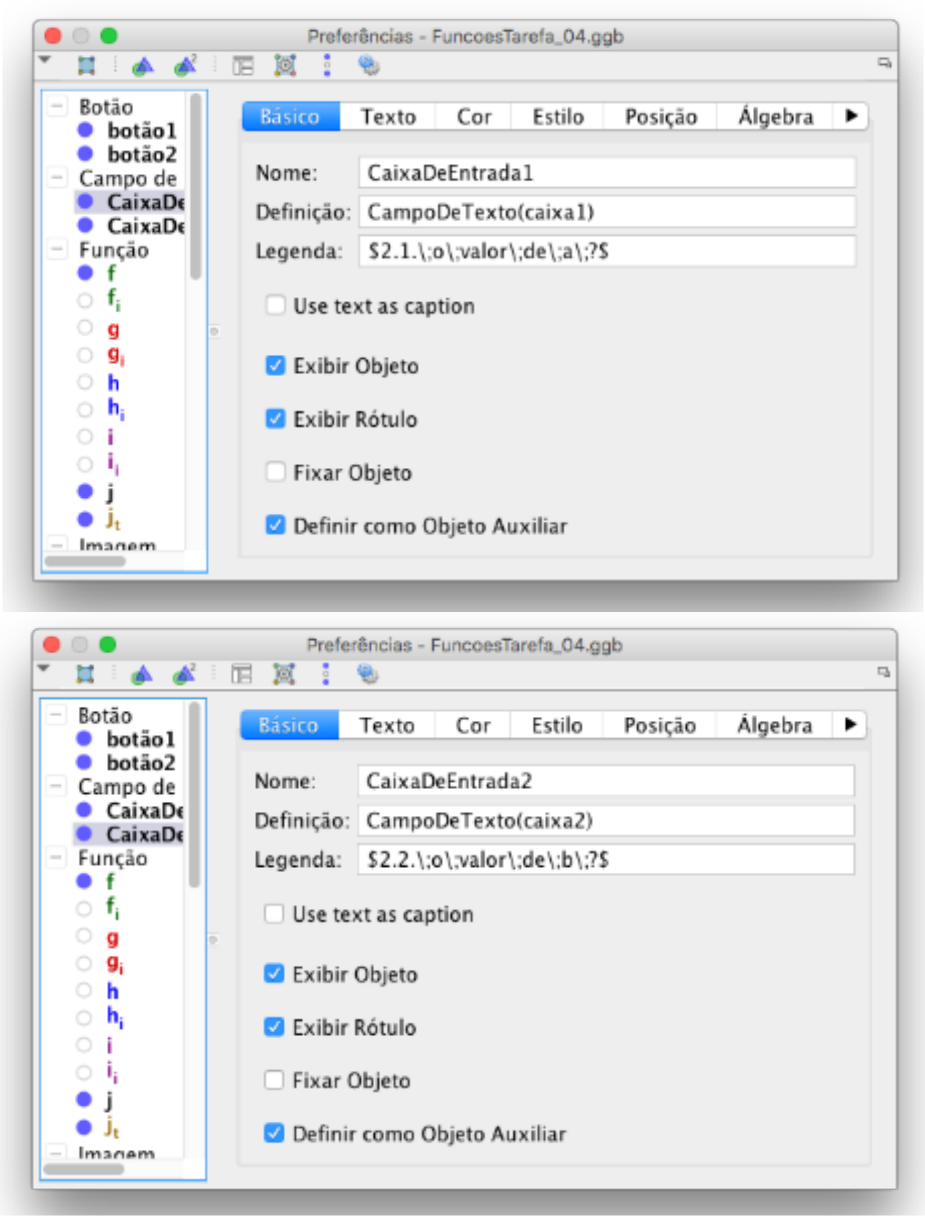


Figura 6– Exemplo de como fica a janela de preferências dos Campos de Entrada.

11) Em seguida, vamos construir uma ajuda visual em relação à segunda questão, na Janela de Visualização 1. Para isso, primeiro ative a Janela clicando sobre ela. Uma vez que visualmente, a resposta a esta questão está associada a uma transformação geométrica de translação, vamos construir:

a) o seletor o **Controle Deslizante** *t*, na **Entrada** com o comando

t= ControleDeslizante(0, 1) e ocultá-lo.

b) a função j_t , na **Entrada** com o comando

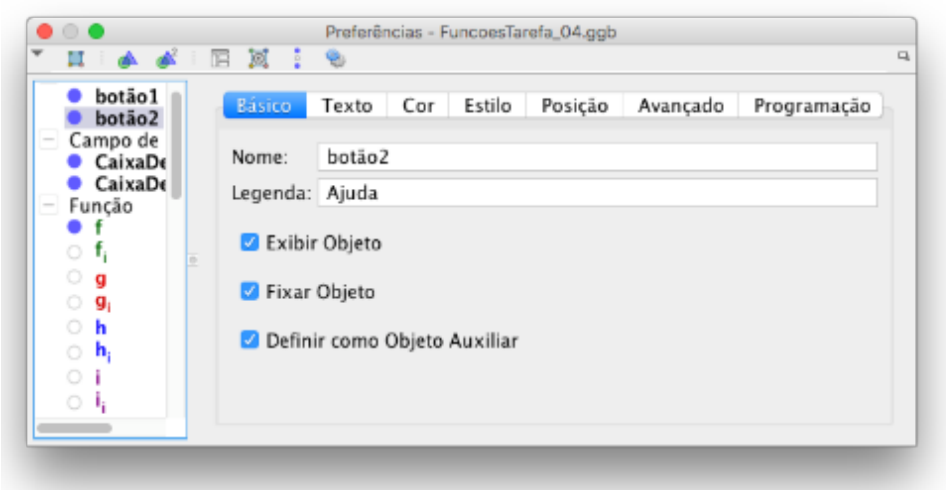
j_t=Transladar(j(x + a) - b, Vetor(t Vetor((a, b))))

c) um segundo botão, na Janela de Visualização 2, com auxílio da ferramenta respetiva, OK, designado por botão2, com a legenda **Ajuda**, e contendo no evento **Ao Clicar,** na aba de **Programação**, na janela das **Propriedades**, o **Código do GeoGebra** seguinte:

IniciarAnimação(t,true)

DefinirTraço(j_t, true)

como pode observar na Figura 7. O código faz com que se inicie a animação do controle deslizante ***t*** e se ative o traço da função ***j_t*** .



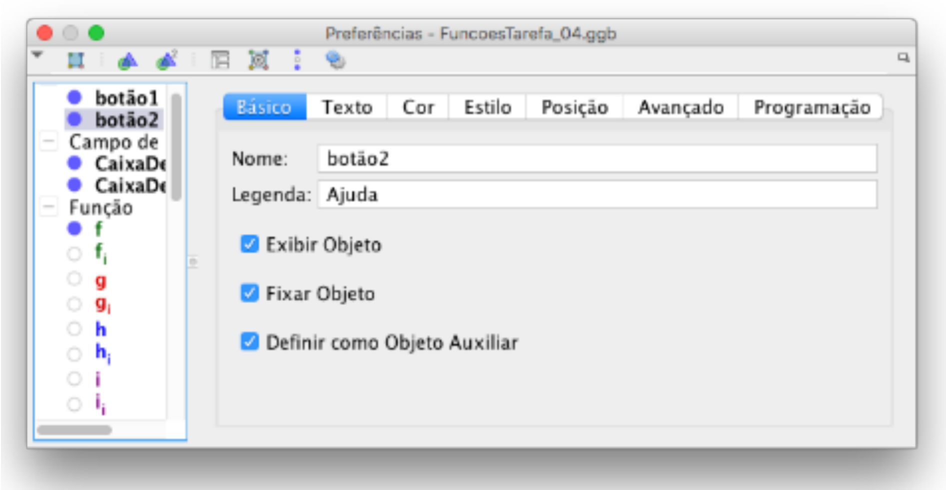


*Figura 7 – Janela de preferências do botão de **Ajuda**.*

12) Na Janela de Visualização 2, à medida que a tarefa vai sendo atualizada, o feedback da pontuação no *texto3* pode ser obtido usando a **Entrada** e o comando

texto4="Obteve um total de " + (LaTeX(grade)) + " pontos."

Repare na sintaxe do comando anterior, entre:

i) as aspas se encontram o texto propriamente dito;

ii) entre os sinais **+(** e **)+** se encontra escrito o comando LaTeX(...), que força a usar o formato em LaTeX

iii) nos parênteses do comando LaTeX consta a variável *grade*, pelo que o texto mudará de forma dinâmica em função apresentando o conteúdo da variável *grade*.

13) A nossa aplicação terá também um relógio. Para a sua construção começamos por:

a) definir a lista inicial do relógio com o comando

LR=Relógio() ;

b) criar o contador vt com o comando

vt=ControleDeslizante(0, 60, 1, .5, 10, false, true, true, false)

c) criar o texto designado por Relógio editando-o como se ilustra na Figura 8

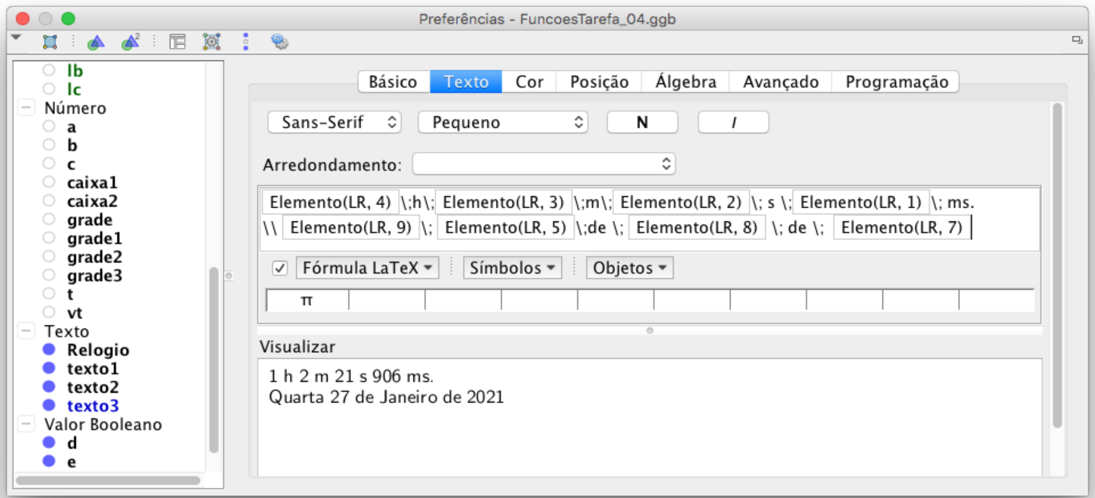


Figura 8 – Formatação do texto que gera o relógio.

14) Tendo sido realizada a tarefa com pontuação máxima (Figura 9) será mostrada uma imagem[1]. Para este efeito é necessário:

a) inserir a imagem com a ferramenta , ajustando o seu tamanho;

b) abrir a janela das suas **Preferências,** a aba **Avançado** , e no campo *Condição para Exibir Objeto(s)* usar o comando **grade ≟ 10.**

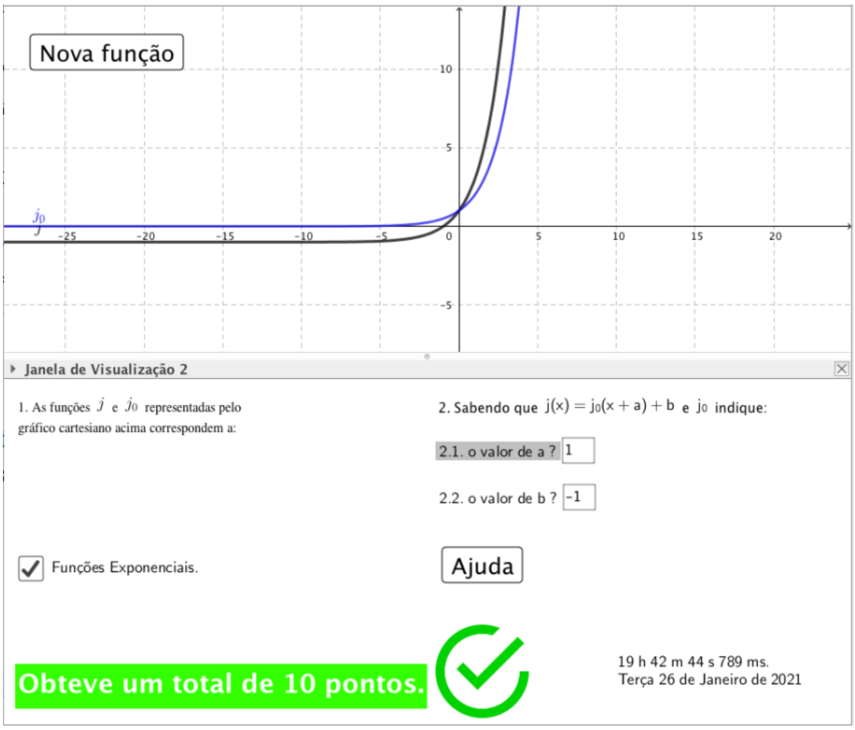


Figura 9 – Imagem com feedback da aplicação "funções reais de variável real.".

15) Finalmente reveja as **Preferências** do botão1 de modo que na aba de **Programação** no evento **Ao Clicar** encontre o Código de GeoGebra que a seguir se apresenta:

a=EscolherElementoAleatoriamente(la)

b=EscolherElementoAleatoriamente(lb)

c=EscolherElementoAleatoriamente(lc)

d=false

---

[1] por exemplo, a disponível em

http://www.geogebra.org.pt/ficheiros_on_line/imagem/feedbackcheck.png

e=false

k=false

l=false

caixa1=?

caixa2=?

t=0

DefinirTraço(j_t, false)

CentralizarJanelaDeVisualização((a,b))

TocarSom("http://www.geogebra.org.pt/ficheiros_on_line/audio/NovaFuncao.mp3")

Desafio: Construa uma aplicação de GeoGebra, usando algumas das estratégias aqui apresentadas, relativamente a outro tópico da matemática à sua escolha.

2ª. Tarefa: Condição de existência de um triângulo

Breve explicação da aplicação

Objetivo da tarefa:

Desenvolver a perceção sobre quais seriam as condições que devem ser satisfeitas entre os lados de um triângulo para a construção dele.

Desafio ao utilizador:

Verificar qual terna de números *a*, *b* e *c* que constrói um triângulo.

No caso de resposta acertada os valores *a*, *b* e *c* são alterados, apresentando ao utilizador um reforço positivo, caso contrário o *feedback* deve permitir ao aluno aproximar-se da resposta correta, apresentando dentre todas as possibilidades da condição de existência de um triângulo, qual a que não foi satisfeita, ver Figura 10.

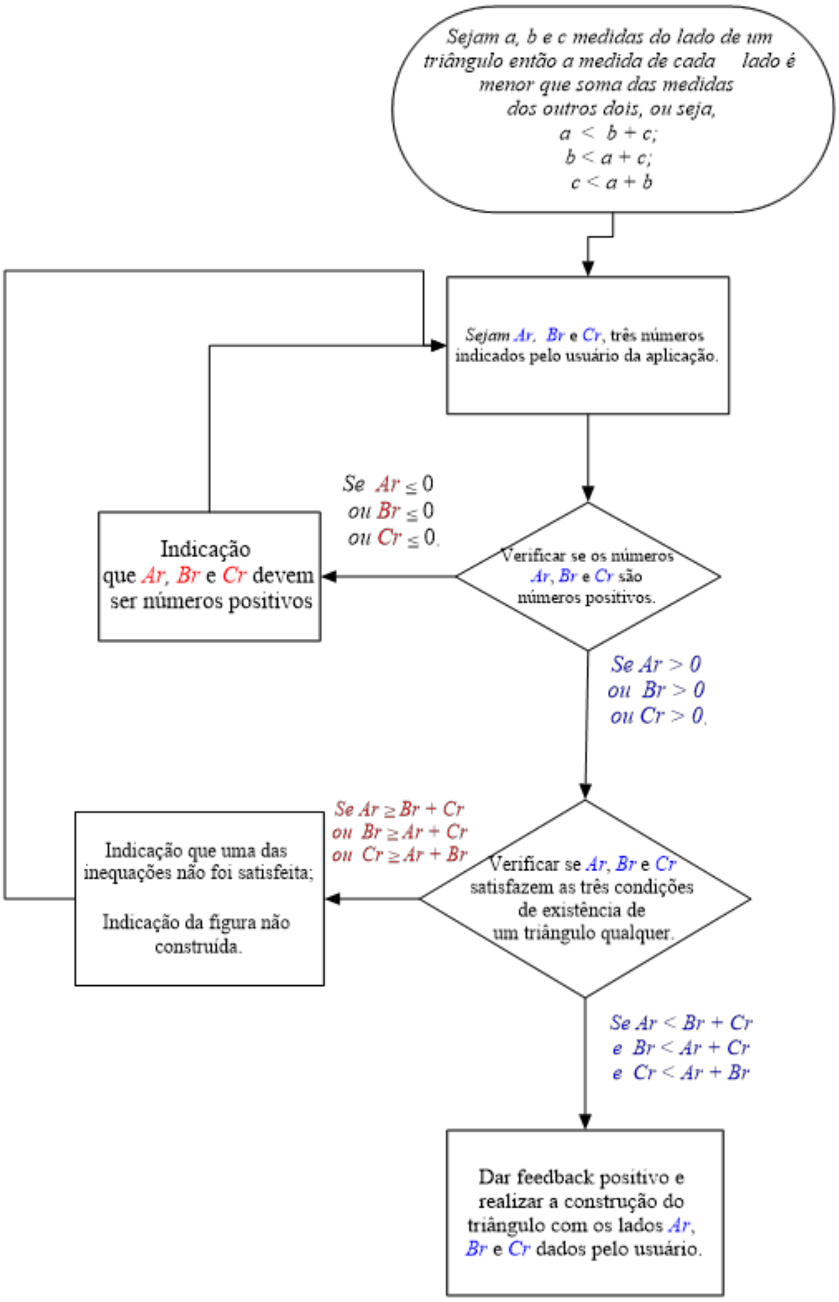


Figura 10 – Diagrama de processo da aplicação

Tipos de Feedback:

a) visual - uma imagem que aparece se a resposta é correta (RC);

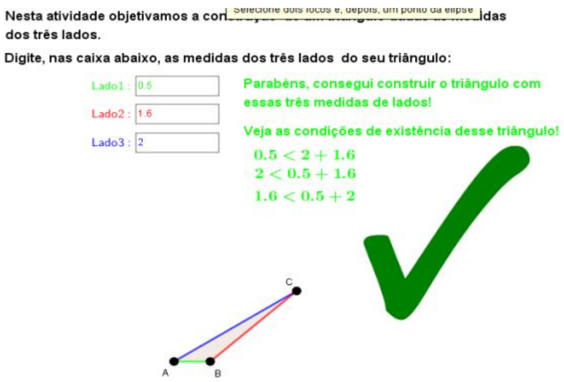
Uma figura que apresenta a impossibilidade da construção do triângulo, no caso do acerto e a impossibilidade de construção, no caso do erro;

b) textual – com a apresentação de um texto dadas determinadas condições:

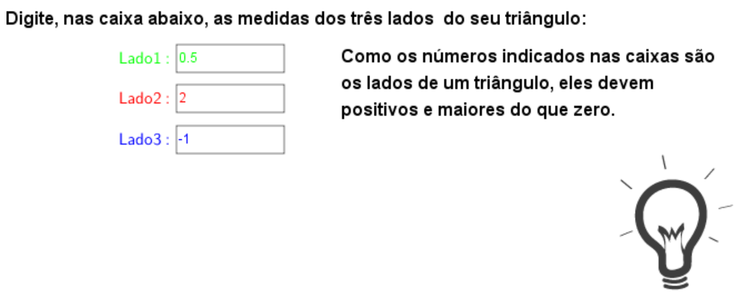
i) fRC - "Parabéns, conseguiu construir o triângulo com essas três medidas de lados!"

Veja as condições de existência desse triângulo!

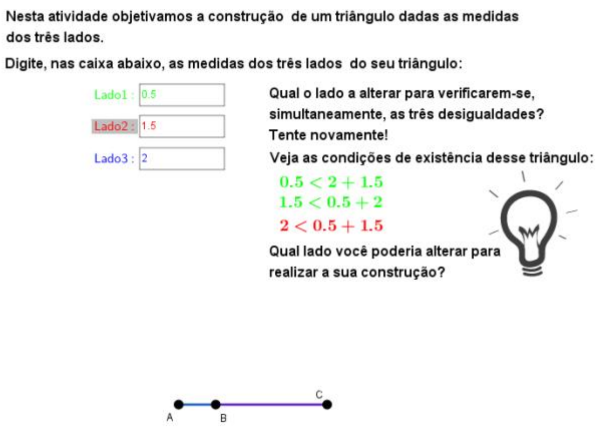
E as condições de existência satisfeita com os três lados do triângulo,



ii) fRI1 - No caso de o usuário inserir um número negativo na caixa de texto:

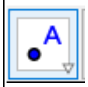
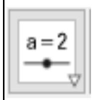
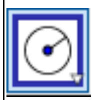
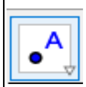
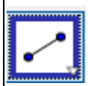
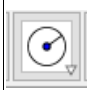
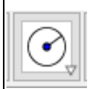


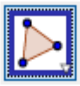
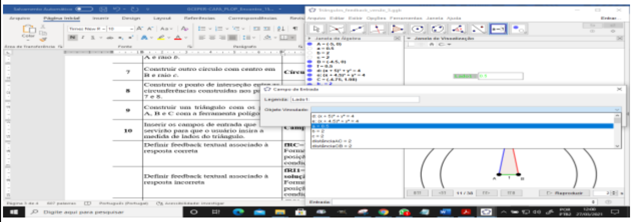
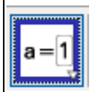
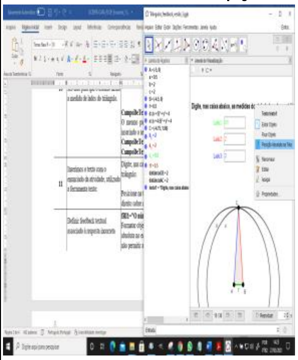
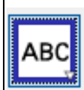
iii) fRI2 - no caso de resposta incorreta apresentação de texto com pistas para o utilizador, sugerindo a mudança de um dos lados para a realização da construção.

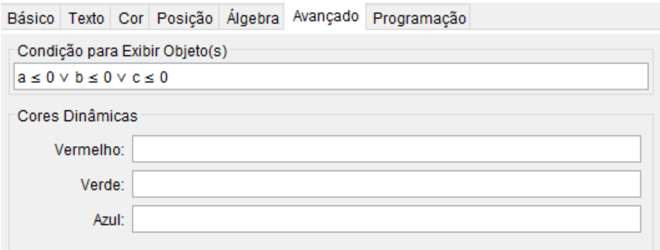
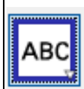
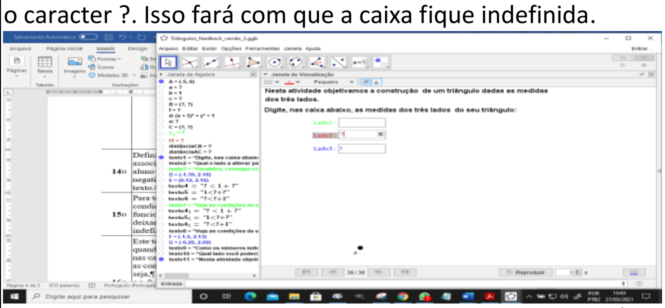
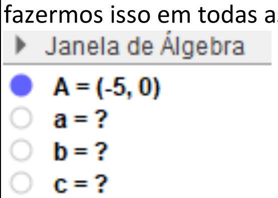


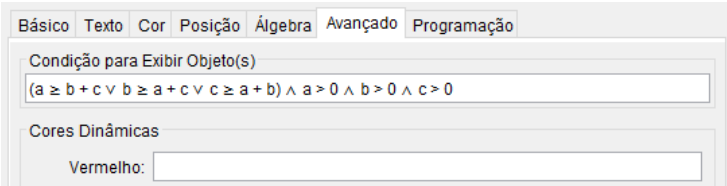
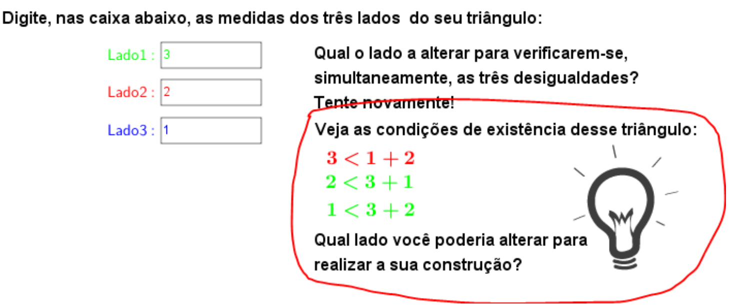
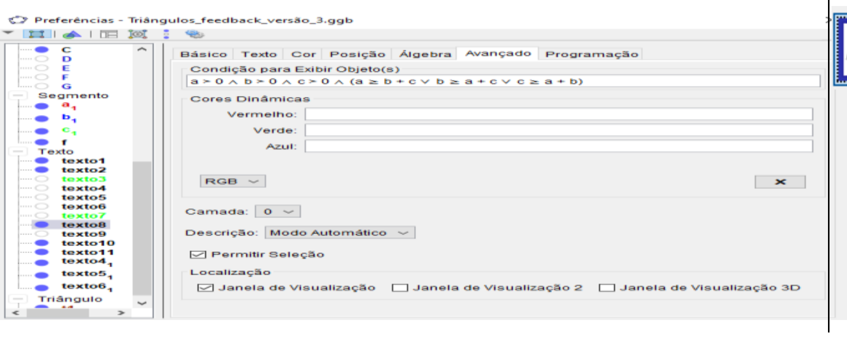
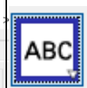
Guião de construção da aplicação

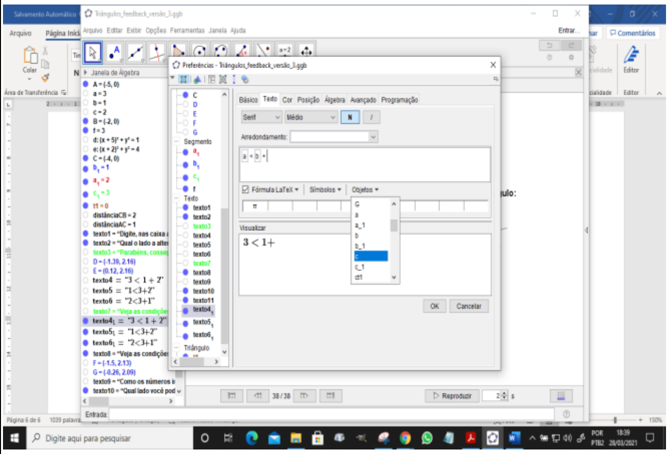
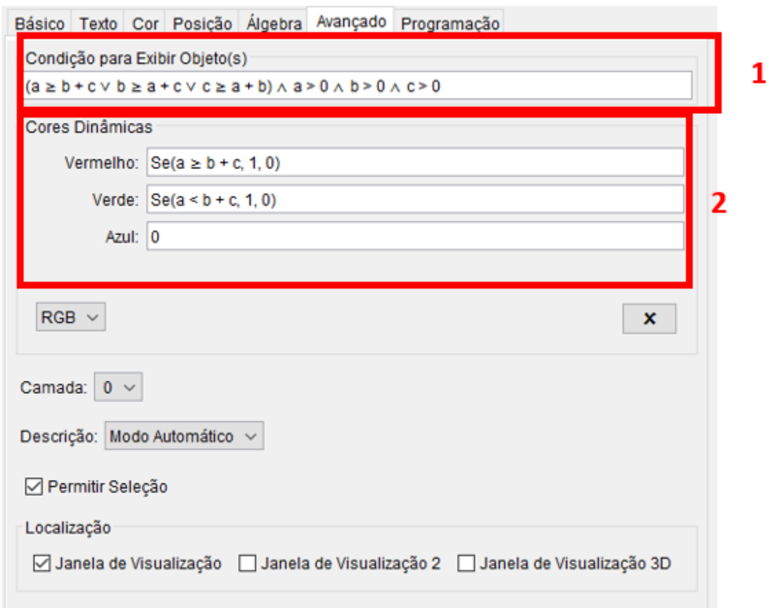
Tabela 1 - Protocolo de construção da aplicação "Condição de existência de um triângulo"

Passo	Objetivo	Comando a escrever na barra de Entrada	Ferramenta
1	Construa o ponto A	(-5, 0) A escolha das coordenadas deste ponto foi arbitrária.	
2	Depois construir três números (*a*, *b* e *c*) que serão os lados do triângulo que será construído.	Digite *a*, tecle enter e selecione a opção criar um controle deslizante; Repita o mesmo processo digitando *b*, tecle enter e selecione a opção criar um controle deslizante; A mesma sequência de comando para o número *c*.	
3	Construir um círculo com centro em A e raio *a*.	Pode ser digitado o seguinte comando no campo de Entrada: Círculo(A, a)	
4	Com a ferramenta ponto, crie o ponto B sobre a circunferência criada.		
5	Construir o segmento AB, que terá a mesma medida do número *a*.	Segmento(A, B)	
6	Construir outro círculo com centro em A e raio *b*.	Círculo(A, b)	
7	Construir outro círculo com centro em B e raio *c*.	Círculo(B, c)	

8	Construir o ponto de interseção entre as circunferências construídas nos passos 7 e 8.	C	
9	Construir um triângulo com os pontos A, B e C com a ferramenta polígono.	Polígono(A,B,C)	
10	Inserir os campos de entrada que servirão para que o usuário insira a medida de lados do triângulo.	 CampoDeTexto(a) O mesmo para os números b e c. Isso pode ser feito, inserindo o seguinte comando no campo de Entrada: CampoDeTexto(b) CampoDeTexto(c)	
11	Inserir o texto com o enunciado da atividade, utilizando a ferramenta texto:	Digite, nas caixas abaixo, as medidas dos três lados do seu triângulo: Posicione na tela no local que preferir, clique com o botão direito sobre o texto e selecione a opção 	
12	Definir *feedback* textual associado ao evento: inserir número(s)	fRI1="Como os números indicados nas caixas são os lados de um triângulo, eles devem ser positivos e maiores do que zero."	

	negativo(s) na(s) caixa(s) de texto.		
13	Este texto só vai aparecer quando os valores inseridos nas caixas não satisfizerem a condição do fRI1, ou seja, a ≤ 0 ou b ≤ 0 ou c ≤ 0.	Para fazer isso, clicamos com o botão direito sobre o texto relacionado ao fRI1, selecionamos a aba ‘Avançado’ e inserimos as condições que gostaríamos que fossem satisfeitas para que o texto apareça. Neste caso, como os lados tem que ser positivos, então: a ≤ 0 V b ≤ 0 V c ≤ 0 Abaixo segue uma imagem de como isso pode ser feito no GeoGebra: 	
14	Definir *feedback* textual associado à resposta ao aluno inserir número(s) negativo(s) na(s) caixa(s) de texto.	fRI2a="Qual o lado a alterar para verificarem-se, simultaneamente, as três desigualdades? Tente novamente!”	
15	Para testarmos se as condições impostas estão a funcionar, é melhor deixar os textos da caixa indefinidos.	Para fazer isso, clique com o botão esquerdo sobre a caixa de texto, e digite o caracter ?. Isso fará com que a caixa fique indefinida.  E veja a denominação das variáveis na Janela de Álgebra, depois de fazermos isso em todas as caixas: 	
16	Este texto só	Para fazer isso, clicamos com o botão direito sobre o texto relacionado ao	

	vai aparecer quando os valores inseridos nas caixas não satisfizerem as condições da fRC, ou seja, a > 0 e b > 0 e c > 0 (os valores dos lados são positivos) e a ≥ b + c ou b ≥ a + c ou c ≥ a + b	fRI2, selecionamos a aba ‘Avançado’ e inserimos as condições que gostaríamos que fossem satisfeitas para que o texto apareça. Neste caso, como os lados tem que ser positivos, então: a > 0 ∧ b > 0 ∧ c > 0 e Os números *a*, *b* e *c* não satisfazem pelo menos uma das condições de existência de um triângulo: a ≥ b + c V(ou) b ≥ a + c V(ou) c ≥ a + b Abaixo segue uma imagem de como isso pode ser feito no GeoGebra: 
	Outros elementos que inserimos em fRI2 são textos dinâmicos que se alteram de acordo com os valores fornecidos pelos usuários.	
17	Para fazer isso, criamos com a ferramenta Textos	fRI2b1: Os trechos “Vejas as condições de existência desse triângulo:” e “Qual lado você poderia alterar para realizar a sua construção” são feitos utilizando a ferramenta Texto duas vezes e possui a seguinte condição de existência. a > 0 ∧ b > 0 ∧ c > 0 ∧ (a ≥ b + c V b ≥ a + c V c ≥ a + b)  

<table>
<tr>
<td>18</td>
<td>Para fazermos os trechos das desigualdades faremos da seguinte forma.</td>
<td>Clicamos com a ferramenta ‘Texto’, na janela de edição do texto, selecionamos a caixa de seleção Fórmula Latex e selecionamos a opção ‘Objetos’ e selecionamos os números a, b e c criados no passo 1 que são as medidas dos lados dos triângulos.
Para a primeira desigualdade selecionamos o objeto “a”, digitamos “<” selecionamos o objeto “b”, digitamos “+”, e selecionamos o objeto “c”.

Repetimos o mesmo processo para os textos das desigualdades: “b < a + c” e “c < a + b”</td>
<td>
</td>
</tr>
<tr>
<td>19</td>
<td>Cada uma das desigualdades construídas no passo anterior possuem uma condição de existência e outra linha de programação que fará com que a cor seja alterada, para que o aluno identifique qual condição não foi satisfeita.</td>
<td>

Na figura acima, estamos mostrando como a desigualdade a < b + c, deve ser programada.

A condição 1 possibilita ao texto da desigualdade apareça se o usuário indicar valores de *a*, *b* e *c* positivos (a > 0 ∧ b > 0 ∧ c > 0) e não satisfaça pelo menos uma das condições de existência do triângulo (a ≥ b + c ∨ b ≥ a + c ∨ c ≥ a + b)).</td>
<td></td>
</tr>
</table>

		A condição 2, indica que o texto da desigualdade vai ser apresentado na cor vermelha se ele não satisfizer a condição de existência, e na cor verde se satisfizer a condição de existência. Isso é feito da seguinte forma: Na caixa Vermelho é digitado o seguinte: Se(a ≥ b + c, 1, 0); Na caixa Verde, é digitado o seguinte: Se(a < b + c, 1, 0); Na caixa Azul, é digitado zero; Para os outros textos o processo é repetido, com as devidas alterações nas condições.	

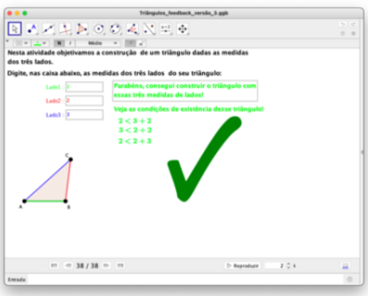
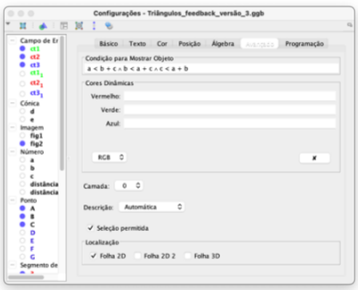
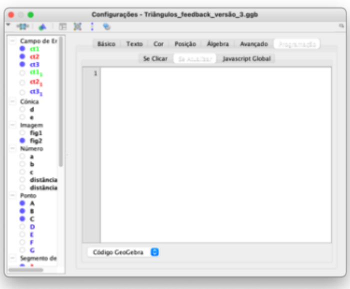
Feedback através de som.

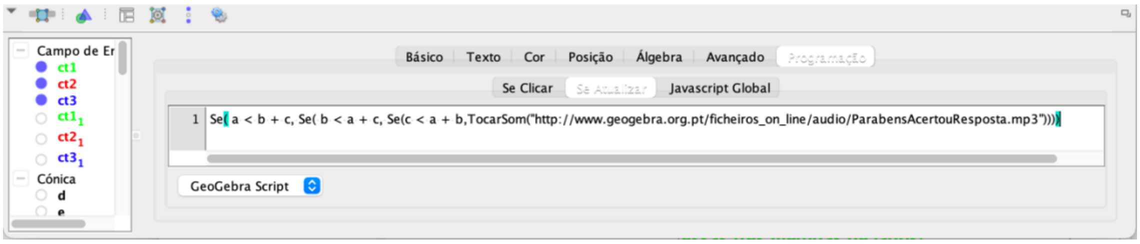
Imaginemos que se o *feedback* fRC pretendia enviar uma mensagem de som referindo *"Parabéns acertou na resposta*!" , tem então a possibilidade de no GeoGebra reproduzir um som de um arquivo mp3 alojado num servidor, usando o comando TocarSom(<url do ficheiro>). Em alternativa poderá transformar esse som em código

TocarSom("`http://www.geogebra.org.pt/ficheiros_on_line/audio/ParabensAcertouResposta.mp3`")

No caso da aplicação anterior, em relação ao *texto 3*, no menu *Propriedades do Objetos,* separador *Programação,* e evento Se Atualizar, poderá colocar o comando:

Se(a < b + c, Se(b < a + c, Se(c < a + b,TocarSom("`http://www.geogebra.org.pt/ficheiros_on_line/audio/ParabensAcertouResposta.mp3`"))))



Em alternativa ao link entre as aspas poderá escrever `data:audio/mp3;base64`, ou seja

TocarSom("`data:audio/mp3;base64`")

onde `base64` terá de ser substituído pela codificação do seu áudio, previamente gravado em formato `mp4`. Na página https://test.geogebra.org/~mike/utils/base64Encode.html terá o texto exato, desde que converta o comando PlaySound , versão inglesa, para TocarSom, versão portuguesa. Observe-se que ira obter uma lista de código como a que se segue:

PlaySound
("data:audio/mp3;base64,//uQBAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAASW5mbwAAAA8AAAA/AABofAAECAgMEBAUGBgcHCAkJCgsLDAwNDg4PEFBRUlJTU1RVVVZXV1hYWVpaW1xcXV1eX19goaGio6OkpKWmpqe
[illegible]

Que corresponde a conversão do áudio disponível em:

[http://www.geogebra.org.pt/ficheiros on line/audio/ParabensAcertouResposta.mp3](http://www.geogebra.org.pt/ficheiros_on_line/audio/ParabensAcertouResposta.mp3)

Para:

PlaySound("data:audio/mp3;base64")

A vantagem deste segundo método é que o som estará disponível numa aplicação sem conexão à rede e a sincronização do som é mais eficaz. Contudo tem o inconveniente de poder ter de ser necessário colocar uma grande quantidade de texto.

O feedback de áudio pode ser uma alternativa eficaz pois considera uma maior diversidade de estilos de aprendizagem, nomeadamente atendendo a alunos com necessidades educativas especiais.

3ª. Tarefa- Como criar código de HTML que permite interagir com os métodos da applet do GeoGebra

Breve explicação da aplicação

Uma das vantagens do uso do GeoGebra é a possibilidade de criação de páginas de HTML onde o usuário pode introduzir dados que interagem com as applets do GeoGebra. Estas páginas podem conter mais do que uma aplicação de GeoGebra que interagem entre elas. As alterações nas aplicações de GeoGebra podem conter parâmetros que podem ser mostrados ao utilizador. Esta interação é útil quando se pensa em construir situações adidáticas (Brosseau, 2008) para promover a aprendizagem autónoma em contextos síncronos e assíncronos, nomeadamente no ensino e aprendizagem da matemática. O dispositivo tecnológico necessário para a construção destas aplicações passa pelo uso de propriedades elementares de HTML e de JavaScript, para além do conhecimento dos comandos e métodos do GeoGebra associados.

Assim a tarefa usa apenas comandos, botões e caixas de texto para construir uma aplicação simples para demonstrar como poderemos alterar uma página em HTML que se obtém com o GeoGebra e poder criar alguma interação com o utilizador da página. De seguida apresentamos um guião de uma aplicação que permite que o utilizador introduza o valor numérico do raio de uma circunferência num formulário de HTML alterando-se a representação dessa circunferência em função do valor introduzido.

Guião de construção da aplicação

Vamos criar uma página de HTML que permitirá alterar o raio de uma circunferência pelo utilizador, através de um botão, inserido numa ***form***, usando os comandos ***evalCommand()*** e ***reset()*** .

1º Passo

Criar o ficheiro **circunferencia.ggb**, por exemplo, utilizando a linha de comandos, para isso basta dar entrada:

r=2

A=(0,0)

Circunferência[A,r]

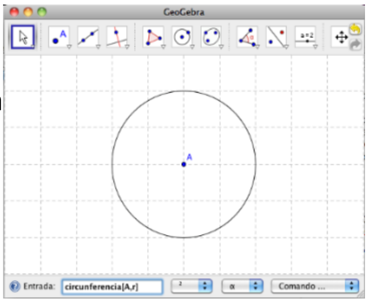


Figura 11 –Vista do ficheiro circunferencia.ggb

2º Passo

Exportar o código do ficheiro ***circunferencia.ggb*** via o parâmetro ggbBase64, para isso deve usar sobre a janela do ficheiro aberta a sequência de teclas: Ctrl+Shift+B, em PC e Cmd+Shift+B, no OS. No caso do arquivo proposto no 1ª passo terá o código da Figura 12.

UEsDBBQACAgIAHSimlQAAAAAAAAAAAAAAAAAWAAAAZ2VvZ2VicmFfamF2YXNjcmlwdC5qc0srzUsuyczPU0hPT/LP88zLLNHQVKiuBQBQSwcI1je9uRkAAAAXAAAAUEsDBBQACAgIAHSimlQ
AAAAAAAAAAAAAAAAXAAAAZ2VvZ2VicmFfZGVmYXVsdHMyZC54bWztWltz4jYUfu7+Co2f2oeAZTCQTMhOdmc6zUw222kynb4K+2DUGMm1RDD59StLvhFjkkAIbhsekl6s6/cdHellOv+czEP0
ALGgnl0t3LEtBMzjPmXB2Frl6cnI+nzx6TwAHsAkJmjK4zmRY8tNcxbllNQZ2naaRqJobHkhEYJ6FopCItMiY8u3EEoEPWP8hsxBRMSDW28Gc3LNPSJ1LTMpo7Nud7lcdvL2OjwOuqpK0U2E3w0C2VG
hhVSnmRhbWeRM1btWetnT5Rzbxt2/vl2bdk4oE5lwDyykBuTDlCxCKVQUQpgDk0iuIhhb0wXz0u6wBxJbKCQTCNOGsvxja+BaF59+OhczvkR88jd4Kk3GCyjyaqGb5lGfv/KQxygeW47dt5CCVC
E00f8kjGZExTpD1+QOyQpi9EDC9LNOIQvJPV2BTp2SUECeVzX1jftgvvSz/IzONZRISFAsYAuJCMDXMTM6W1Oy0uwW9Z13MwhqYIRUyGJg11oogMA9u46EqXM7FLYGAttPoThJte/IUKj6KINbuQo
ByRn17hkIpWhupVAa+Y36PqTzxZSJOGXylj5mfXCrqbqqtaGwlISnyc0URDxcBZwVmP6eywURQ8PDKzuOuJqCVK5UfDgyPWvW6Qo+VSax29Ncurim1Ype/cP9UxvjAXaOzex2hFPw1iA2CSXG+
OAYb5ssdatx9JnSjGduQYuRfV3ED1Uwe317g+I4BzDKWo9sNWqKsU07lRWpqKaS0M/TGOCXqi3eDc/hUAOaBgpRx3XfQ8GUbajDijfD+irD6nEe+wlIY+uG3FholYWPJnw9yD5EwBRhcg1pvBPSg
5FGOg0mJmg10P13BPp7dS1T68Qu8GLHNZuKNPxQ5ScIX7E/IKBrewbc+8D5wJrc33uJayG6GjCDo0j/lavH51ElydHBb7LXg/+xvcb7lx1ybyFKF9BIBbqj5/YdDY5Da3e1aUsJDSmJV/WWDrbHW/
M+rtc9D+c4W+X2eNgbnCj4h63ZIKpMEFV59tZ2AUEqFdje5nJJx277v3enY6MxORghfCHDtNorJiEWoN0/Uev5PUB0p6r6zu5iwkR6LvjULxOSxFULluGe0c78pk97cP7cgZ+z21apeSF3WnzgV/Pdq0
Dsv6P5d1uircjFZLXNbOx2avzfwe6QRmOPyf+gmuXltP8zE8ujqo/Vt8bbBh9AGW0QlLC9+ZCQIKvvnRYqh1ytJKN5MB5n6R1YfgpqpGI4/VaO5tg7ie1bO6ABMDNnlX7Z2YXlyjYAoMc8JcFZygpnK
Y9ZRFejxhfTBF3m5S7z7JdOHunlkX4ecQuUn2E8UrOhshQ8sSj93TaQbbhqaZMmvNni0XaVUhQEpXJcGalyHG2syJQqoBiZqwKmP5R9Id59EPMF82uwvI3NacF1bTNsbDGHuGJ9b3K5gM41yKluLP
JzqXxAIqS+GtGcKt08UUo7J4lWXjIRPFxIuPViAFY+XDAQLqkvZ6nDrsmd0iTtf3YhPuMxfeRMFrqHUg4uQ/3IYe1iatO8cbZfP79keWgwEs1EVSjZjwl9313wcGmkkgVz4L/xUvx5cuyMm0HHGfXwyO3Z
Qzw8dUeDF3KFR1WuzKf9qGoycc4mE0dirzzD69lNbNafSuzHZ2bMbTzsuz3n1HHx6WlfRdy9Z92E8xBI6T1+yeXKBU9t2jUZppe/JTngbsibgXc/4cmajrzOhf61SCgfEbXz/luPsZb1DY89u5VnV938jdfFD
1BLBwjgJDGpHgUAAIUmAABQSwMEFAAICAgAdKKaVAAAAAAAAAAAAAAABcAAABnZW9nZWJyYV9kZWZhdWx0czNkLnhtbO1YzXLbNhA+N0+BwT0iQJGU6TGd0SSHdCbJNJNLrzC5ktC
SAANAouhX6zv0mQqCgEw1duKoqpNJq4Ow+Nld4Pt2FySvXuybGu1AaS5FgemMYASilBUX6wJvzer5BX5x/exqDXINN4qhlVQNMwVOh5UHPdubLQgZxljbFrismda8xKitmRlUClxhhPaaXwr5jjW
gW1bCh3IDDXsjS2acIY0x7WUUdV03C/5mUq0ja1JHe11F67WZ2RYju2mhC+yFS2v3SLubO72YEBr9+vbN6Oc5F9owUQJG9kAVrNi2NtqKUEMDwiDTt1Bgu2MBc+ujZjdQF/gX13+Fkdco8NzaxdfPf
rrSG9khefMblHbUqC0clFas1oCjYZGdfylrqZAqcJ5jZGGNqW1vfMvqdsMKTGbpuL5mPSi0Y9YKGUfY1sjSmXCjE9vO21tZwTiTjKMrNrDnTcyoX8oFfDB9DchsePm7AK0HB/7QXnjNqwqGMBh1SiIVp
dF+MGNX9L699W03tnbpVeQx/ATNUgpeTtD8WRgbMxYwSzgqt2oHU2BpOgLrHcvgWQbXMviWnQ/WXVizC2t2h/2FqS5MdbeHLZ/AXkwde5RQxx7N4wI7llw/muSE0ozG52KTCd64/EDaQDsyi
nQLUA083fFn47Z3STu1eB/p6edJh49iVNHDv80Hpmy+GmWzeZznaxA7C5pUNjSIp6EnITjCyJ6GsKF+5NYLzow9k+J7tAx6y7B8GQdhHoQkCOkjIo7tuR7S1XO59N1JkJH5adILHP3EkU8m1JN/i+pv
RjTyMthT//nHFzJ8SOKSKQOaMzFJ9ZfDxN+hz/6H/jNYtrLuN1ApKe6un8nQHZBzfwOdwvtXg0/TuYM/pZ/gn/iql+YZSbLkbFfY6UXvYXA/blnlgtuf9n3oT2Glp8VnTMe7IVk4iIbmgFGW5MNvkdH0giYx
PRdG58h53rQ1L7l5dGknD5f2cSpU7z4lt/GZqj5aZkFYBOEiCPkjbga9VSv71HIfrfJTx7GQnBgLVvH+arV4bH44N09Xrehp1UqAOWDxbpCn4KX/qfr0z6vTfaW/t5cwr45wpU9Y993rid19Pha1+McBVn
HdHMNKnxLWbHwLHGHNsx8F1qUqN7yBCtjxs4p9F3pCcB++iL/vh5Vo8kUiCp8/rv8CUEsHCDzdVE5rAwAAoBEAAFBLAwQUAAgICAB0oppUAAAAAAAAAAAAAAAADAAAAGdlb2dlYnJhLnh
tbL1YW4/bthJ+Tn/FQE8t4AspUZld2CmcoGgLpG3QPacozhstcW12ZUmV6F076I/vDC+yvJu02eag3pV5G87Im+EM5dXXp0MF96rrdVOvIz5jEai6aEpd79bR0dxOF9HXr75Y7VSzU9tOwm3THaRZ
RyIRDvtwNMsZoznZtuuoqGTf6yKCtpKGtqyjMgJdIuWCpcltLKfJQuRToZZyKvNFPFVJmcRlkRRyi0zg1OuXdfOjPKi+IYW6KfbqIN82hTRW3t6Y9uV8/vDwMAuazZpuN0fh/fzUI/PdbjvDNgl0r+7Xke+8
RL5Xux8Suy9mjM9//eGtkzPVdW9kXagIyPSjfvXFi9WDrsvmAR50afbrSKRZBHuld3vEImNJBHMiahGQVhVG36set46G1nhzaCNLJmtaf+F6UA12RVDqe12qbh2xWZqKRYyCuGBJyhaLCJpOq9p4
Wu5lzgO31b1WD44t9RzcPI7ANE21IcQT/gAOKcMH+BImkOU4EwNPQeDMAmdySGgu5QISIBKegBDYCprmGa7QMn6nKQPOcQViBnEMMYc4wWGaQopkOe2NkTZbWn4MH6JGjfBJaC5J8L
FzicAnph4ySh0b1CNNMttL7feC9qCUNCY77BLOiSWKo4k055CgJjjOGSDfhIRwa41gQP8cBAmJc4gXYLla/gwxute93lZqHd3KqkfH6/q2w+gbxr05V8qCCcuTuMT/EMK/R7JU4bh62IFVxib0JPhI
2hhfu0dce0bdAVD2ybUcNeQuInmIpibY4IrYtcI16SORrjtwpE6a5lwNCJ5jpmmOz62MtiYPMfGxchGTjagT0h52yRAanOrPjXCDzM3tCHHOPOzC/pa0gAhyRa285meS/6RTYIt878Q7KD7NCj5ciS
Wp2ySxMtJHqNsvngil4+EOqbPMTbIFCL9dFM/72gM8MYpdo3crqPN22+/ef3z5tM14J/p46ACgnsRikDbf/s8EZn8ldV/691BILEZrMRMPsmTeLLMkgImtCcys6tk8K+bnLMP5iLXct8+Kxg+jsvfaLWah0
K28hpBvydaH/JGHXrSMVlCnkAWD0Uto5rjK1seQ55Cno3q24QqXJZeihyVuMVVkUsX15Uuo8nclk2UR0XKlbxYhKo38XXvjyd1DwuUuNQoVJBYcQCsrDZ5hWKFWsRDuYpTqlgxpjeslzFklC4/Urnwc
tX0esB2r6p28IKFUdft0VxBVxzK0DUNUsvKXp08fdkUd68fga1kb0IfifCycbnSuMvH1Y3nxaqSW1Xh5fGG4gDgXlYUzZb/bVMbCPlAuLmiqft3XWPeNNXxUPcARVOxQd2m4qN+POono74YzMZBOlrl
xgt5GFzJbXAFjr1C+U3Xh72yLL8niktUlyQ/1dX5dafkXdvo2vQjdqu5vSOu1LGodKll/QvGb7iQ/Xg8bFUHttuQp6x8QgGGyyQl5HCZFIIFFZuuvDn3GO5w+p/qcHPM4xkek+HDsQqdwxKbLccfOm
6FpJMqltcrWFfOfinFK/v4k2ROtrq/UcagG3uQJ3XBYdfpIYKo/33/uqnKYdki80a25tjZ9wfM+R1Ztal3lbIRYUMVL9nF3bY53bhQyByv/5xbKohOge3OegQ6Kh2lzs63W9daGtJsoGKWhlkKDyAxHdb5
MrYUtt261IJhsDrVvKU8WMmCFN2DG/uTEfltRTpd2Y+1Nm/DwOji7mlpbXABEGLrmif/f/FczR8F38qf7RCKh6ZU7jQmjv5qfXWnulpV/mSg44/NsXfko0OD5+SdNPtNXf6sdph33klK/QYVcaQXA0tV6
ANudPMeaUlR8F80zM2WatepAlhTxvnB5wzoWzxtZb9XygzecldiTGaNCeqv8lpRKVvUDhrz4jSmV9KDPKES1MPD1Lq3XNrWF51uKcphi2XqTl0CudQ9MSrHh5wSAlpYUMZFlxhyB6aKo9k3nX1zk4Z
mKA+cUPOe3oyDQzvMpKfWI7FSlTrgexwYG/H18aA6elceaG3qkNXR2xjPvPHkeWi2v2G6fRwxF+hx/SOHAmTV7iW9WfKQP8SScZ7xcBTkmTLVKGVb7j8MoTM4Ep1rYcDUr1wYO1uYfeM/28M/So
/e4A9Cs/HQfMkmwL4KjGwm+QBYbv6y9wks/tbh1kexO4Ylzy0u1Ph8coGGfSIU7AkUNrQQ2+gJKvxDqOC7P9lCOdBxTMezjyLcB34PJ6cjnH37/hLNI5SL5nCQdQm1vZm90V1RqehyJZDMwi45B
ZtD52jCQuGYeRZPHICnYBSrxT9zwMfi8pngX1K32WOKrDGwCMZLLFLnO12Wyl2RmlYW2iB0PF/4Ovd77Rj0Lr3SVUbf6sKv6p2q79EqvBrAifmfx84sYB9mTgjk1LmF+6n3fHAM5I5pOn2CTaDfBKoNX
mmmwvYSz3QjAq9N6nuPnDsfJyJ7+fC/gL36E1BLBwjrZdu5lwcAAM0TAABQSwECFAAUAAgICAB0oppU1je9uRkAAAAXAAAAFgAAAAAAAAAAAAAAAAAAAAAAZ2VvZ2VicmFfamF2YXNjcmlwdC5qc1BLAQIUABQACAgIAHSimlTgJDGpHgUAAIUmAAAXAAAAAAAAAAAAAAAAAAF0AAABnZW9nZWJyYV9kZWZhdWx0czJkLnhtbFBLAQIUABQACAgIAHSimlQ83VROawMAAKARAAAXAAAAAAAAAAAAAAAAMAFAABnZW9nZWJyYV9kZWZhdWx0czNkLnhtbFBLAQIUABQACAgIAHSimlTrZdu5lwcAAM0TAAAMAAAAAAAAAAAAAAAAAHAJAABnZW9nZWJyYS54bWxQSwUGAAAAAAQABAAIAQAAzRAAAAAA

Figura 12 –Código a inserir no parâmetro ggBBase64 do código de HTML.

3º Passo

Ao exportar o ficheiro *circunferencia.ggb* será criado outro com o nome *circunferencia.html* e será neste que irão ser introduzidas as marcas, *bem como algum código* de HTML ou JavaScript. Para alterar o ficheiro HTML deve-se utilizar um processador de texto semelhante ao *notepad* ou *textedit*.

Ao abrir o ficheiro *circunferencia.html* observará o texto seguinte:

Informação sobre o tipo de documento para as aplicações de acesso a Web. Início do documento com a marca ***HTML***.	<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"> <html xmlns="http://www.w3.org/1999/xhtml">
Cabeçalho entre as marcas ***head***. Na marca ***title*** consta o título da página cujos caracteres especiais devem ser escritos em código[2]. Inclui marcas de formatação.	<head> <title>Circunferência dado o seu raio e centro. - GeoGebra Folha de trabalho dinâmica</title> <meta http-equiv="Content-Type" content="text/html; charset=utf-8" /> <meta name="generator" content="GeoGebra" /> <style type="text/css"><!--body { font-family:Arial,Helvetica,sans-serif; margin-left:40px }--></style> </head>
A biblioteca javascript deployggb.js precisa ser incluída com a tag à direita. Neste caso a biblioteca está no servidor do IGP. Para obter ficheiros mais recentes consulte: https://download.geogebra.org/package/geogebra-math-apps-bundle	<script src="http://geogebra.ese.ipp.pt/ficheiros_on_line/app/rebootggb.js"> </script>
Início do corpo do HTML: ***<body>...</body>*** Marca de tabela[3]: ***<table>...</table>*** Marcas de formatos de paragrafo: Cabeçalho *** ... *** Normal *** ... ***	<body> <table border="0" width="600"> <tr><td> Circunferência dado o seu raio e centro. Altere o valor del raio da circunferência e pressione o botão.
Marca da applet: ***<applet>...</applet>*** Parâmetros da applet: Gerais O nome importante para interagir com a applet: name="ggbApplet"	<applet name="ggbApplet" code="geogebra.GeoGebraApplet" archive="geogebra.jar" codebase="http://geogebra.ese.ipp.pt/ficheiros_on_line/app/" width="547" height="329" MAYSCRIPT> <param name="UEsDBBQACAgIAHSimlQAAAAAAAAAAAAAAAAAWAAAAZ2VvZ2VicmFfamF2YXNjcmlwdC5qc0srzUsuyczPU0hPT/LP88zLLNHQVKiuBQBQSwcI1je9uRkAAAAXAAAAUEsDBBQACAgIAHSimlQAAAA

[2] O código ê corresponde ao carácter **ê** e o código â corresponde ao carácter **â**. Para obter o código pode consultar: http://www-atm.physics.ox.ac.uk/user/iwi/charmap.html

[3] As marcas **<td>...</td>** referem-se a célula e **<tr>...</tr>** a linhas. Pode ainda usar-se **<th>...</th>** para cabeçalhos da tabela. Para saber mais sobre marcas de html consulte: http://www.w3schools.com/tags/ .

O local onde pode ser encontrada a aplicação, on-line:
codebase="..."
Neste caso a aplicação está no servidor do IGP.

Específicos
O parâmetro arquivo, ***filename***, ou de codificação imerso no HTML, ***ggbBase64***.

Parâmetros para controlar a memoria das aplicações de acesso a Web.

Vários outros parâmetros booleanos de formatação.

AAAAAAAAAAAAXAAAAZ2VvZ2VicmFfZGVmYXVsdHMyZC54bWzt
Wltz4jYUfu7+Co2f2oeAZTCQTMhOdmc6zUw222kynb4K+2DUGM
m1RDD59StLvhFjkkAIbhsekI6s6/cdHelIOv+czEP0ALGgnI0t3LEtBM
zjPmXB2FrI6cnI+nzx6TwAHsAkJmjK4zmRY8tNcxbllNQZ2naaRqJob
HkhEYJ6FopCItMiY8u3EEoEPWP8hsxBRMSDW28Gc3LNPSJ1LTMp
o7Nud7lcdvL2OjwOuqpK0U2E3w0C2VGhhVSnmRhbWeRM1btWe
tnT5Rzbxt2/vl2bdk4oE5IwDyykBuTDlCxCKVQUQpgDk0iuIhhb0wX
z0u6wBxJbKCQTCNOGsvxja+BaF59+OhczvkR88jd4Kk3GCyjyaqGb5
lGfv/KQxygeW47dt5CCVCE00f8kjGZExTpD1+QOyQpi9EDC9LNOIQ
vJPV2BTp2SUECeVzX1jftgvvSz/IzONZRISFAsYAuJCMDXMTM6W1O
y0uwW9Z13MwhqYIRUyGJg11oogMA9u46EqXM7FLYGAttPoThJte
/IUKj6KINbuQoByRn17hkIpWhupVAa+Y36PqTzxZSJOGXylj5mfXCr
qbqqtaGwIISnyc0URDxcBZwVmP6eywURQ8PDKzuOuJqCVK5UfDg
yPWvW6Qo+VSax29Ncurim1Ype/cP9UxvjAXaOzex2hFPw1iA2CSX
G+OAYb5ssdatx9JnSjGduQYuRfV3ED1Uwe317g+l4BzDKWo9sNW
qKsU07lRWpqKaS0M/TGOCXqi3eDc/hUAOaBgpRx3XfQ8GUbajDijf
D+irD6nEe+wIlY+uG3FholYWPJnw9yD5EwBRhcg1pvBPSg5FGOg0
mJmg10P13BPp7dS1T68Qu8GLHNZuKNPxQ5ScIX7E/IKBrewbc+8D
5wJrc33uJayG6GjCDo0j/lavH51EIydHBb7LXg/+xvcb7Ix1ybyFKF9BI
Bbqj5/YdDY5Da3e1aUsJDSmJV/WWDrbHW/M+rtc9D+c4W+X2eN
gbnCj4h63ZIKpMEFV59tZ2AUEqFdje5nJJx277v3enY6MxORghfCH
DtNorJiEWoN0/Uev5PUB0p6r6zu5iwkR6LvjULxOSxFULluGe0c78p
k97cP7cgZ+z21apeSF3WnzgV/Pdq0Dsv6P5d1uircjFZLXNbOx2avzf
we6QRmOPyf+gmuXltP8zE8ujqo/Vt8bbBh9AGW0QlLC9+ZCQlKvv
nRYqh1ytJKN5MB5n6R1YfgpqpGI4/VaO5tg7ie1bO6ABMDNnlX7Z
2YXlyjYAoMc8JcFZygpnKY9ZRFejxhfTBF3m5S7z7JdOHunlkX4ecQu
Un2E8UrOhshQ8sSj93TaQbbhqaZMmvNni0XaVUhQEpXJcGalyHG
2syJQqoBiZqwKmP5R9Id59EPMF82uwvI3NacF1bTNsbDGHuGJ9b3
K5gM41yKluLPJzqXxAIqS+GtGcKt08UUo7J4lWXjIRPFxIuPViAFY+X
DAQLqkvZ6nDrsmd0iTtf3YhPuMxfeRMFrqHUg4uQ/3IYe1iatO8cbZ
fP79keWgwEs1EVSjZjwl9313wcGmkkgVz4L/xUvx5cuyMm0HHGfX
wyO3ZQzw8dUeDF3KFR1WuzKf9qGoycc4mE0dirzzD69lNbNafSuz
HZ2bMbTzsuz3n1HHx6WlfRdy9Z92E8xBI6T1+yeXKBU9t2jUZppe/J
TngbsibgXc/4cmajrzOhf61SCgfEbXz/luPsZb1DY89u5VnV938jdfFD
1BLBwjgJDGpHgUAAIUmAABQSwMEFAAICAgAdKKaVAAAAAAAAA
AAAAAAABcAAABnZW9nZWJyYV9kZWZhdWx0czNkLnhtbO1YzXLb
NhA+N0+BwT0iQJGU6TGd0SSHdCbJNJNLrzC5ktCSAANAouhX6zv0
mQqCgEw1duKoqpNJq4Ow+Nld4Pt2FySvXuybGu1aAS5FgemMYA
SilBUX6wJvzer5BX5x/exqDXINN4qhlVQNMwVOh5UHPdublLQgZxlj
bFrismda8xKitmRlUClxhhPaaXwr5jjWgW1bCh3IDDXsjS2aclY0x7W
UUdV03C/5mUq0ja1JHe11F67WZ2RYju2mhC+yFS2v3SLubO72YE
Br9+vbN6Oc5F9owUQJG9kAVrNi2NtqKUEMDwiDTt1Bgu2MBc+uj
ZjdQF/gX13+Fkdco8NzaxdfPfrrSG9khefMblHbUqC0clFas1oCjYZGd
fylrqZAqcJ5jZGGNqW1vfMvqdsMKTGbpuL5mPSi0Y9YKGUfY1sjSm
XCjE9vO21tZwTiTjKMrNrDnTcyoX8oFfDB9DchsePm7AK0HB/7QXn
jNqwqGMBh1SilVpdF+MGNX9L699W03tnbpVeQx/ATNUgpeTtD8
WRgbMxYwSzgqt2oHU2BpOgLrHcvgWQbXMviWnQ/WXVizC2t2h/
2FqS5MdbeHLZ/AXkwde5RQxx7N4wl7lIw/muSE0ozG52KTCd64/E
DaQDsyinQLUA083fFn47Z3STu1eB/p6edJh49iVNHDv80Hpmy+G
mWzeZznaxA7C5pUNjSIp6EnITjCyJ6GsKF+5NYLzow9k+J7tAx6y7B

8GQdhHoQkCOkjIo7tuR7S1XO59N1JkJH5adlLHP3EkU8m1JN/i+pv
RjTyMthT//nHFzJ8SOKSKQOaMzFJ9ZfDxN+hz/6H/jNYtrLuN1ApKe
6un8nQHZBzfwOdwvtXg0/TuYM/pZ/gn/iql+YZSbLkbFfY6UXvYXA/
blnlgtuf9n3oT2Glp8VnTMe7IVk4iIbmgFGW5MNvkdH0giYxPRdG5
8h53rQ1L7l5dGknD5f2cSpU7z4It/GZqj5aZkFYBOEiCPkjbga9VSv71
HlfrfJTx7GQnBgLVvH+arV4bH44N09Xrehp1UqAOWDxbpCn4KX/qf
r0z6vTfaW/t5cwr45wpU9Y993rid19Pha1+McBVnHdHMNKnxLWb
HwLHGHNsx8F1qUqN7yBCtjxs4p9F3pCcB++iL/vh5Vo8kUiCp8/rv8
CUEsHCDzdVE5rAwAAoBEAAFBLAwQUAAgICAB0oppUAAAAAAAA
AAAAAAAADAAAAGdlb2dlYnJhLnhtbL1YW4/bthJ+Tn/FQE8t4AspU
ZId2CmcoGgLpG3QPacozhstcW12ZUmV6F076I/vDC+yvJu02eag3p
V5G87lm+EM5dXXp0MF96rrdVOvIz5jEai6aEpd79bR0dxOF9HXr75
Y7VSzU9tOwm3THaRZRylRDvtwNMsZoznZtuuoqGTf6yKCtpKGtqyj
MgJdIuWCpcltLKfJQuRToZZyKvNFPFVJmcRlkRRyi0zg1OuXdfOjPKi+
lYW6KfbqIN82hTRW3t6Y9uV8/vDwMAuazZpuN0fh/fzUl/PdbjvDN
gI0r+7Xke+8RL5Xux8Suy9mjM9//eGtkzPVdW9kXagIyPSjfvXFi9WD
rsvmAR50afbrSKRZBHuld3vEImNJBHMiahGQVhVG36set46G1nhza
CNLJmtaf+F6UA12RVDqe12qbh2xWZqKRYyCuGBJyhaLCJpOq9p4
Wu5lzgO31b1WD44t9RzcPI7ANE21lcQT/gAOKcMH+BImkOU4Ew
NPQeDMAmdySGgu5QISIBKegBDYCprmGa7QMn6nKQPOcQViBnE
MMYc4wWGaQopkOe2NkTZbWn4MH6JGjfBJaC5J8LFzicAnph4yS
h0b1CNNMttL7feC9qCUNCY77BLOiSWKo4k055CgJjjOGSDfhIRwa4
1gQP8cBAmJc4gXYLla/gwxute93lZqHd3KqkfH6/q2w+gbxr05V8qC
6CcuTuMT/EMK/R7JU4bh62IFVxib0JPhI2hhfu0dce0bdAVD2ybUc
NeQulnmlpibY4lrYtcI16SORrjtwpE6a5lwNCJ5jpmmOz62MtiYPMf
GxchGTjagT0h52yRAanOrPjXCDzM3tCHHOPOzC/pa0gAhyRa285m
eS/6RTYIt878Q7KD7NCj5ciSWp2ySxMtJHqNsvngil4+EOqbPMTbIF
CL9dFM/72gM8MYpdo3crqPN22+/ef3z5tM14J/p46ACgnsRikDbf/
s8EZn8ldV/691BILEZrMRMPsmTeLLMkglmtCcys6tk8K+bnLMP5iLX
ct8+Kxg+jsvfaLWah0K28hpBvydaH/JGHXrSMVlCnkAWD0Uto5rjK1
seQ55Cno3q24QqXJZeihyVuMVVkUsX15Uuo8nclk2UR0XKlbxYhKo
38XXvjyd1DwuUuNQoVJBYcQCsrDZ5hWKFWsRDuYpTqlgxpjeslzFk
lC4/UrnwctX0esB2r6p28IKFUdft0VxBVxzK0DUNUsvKXp08fdkUd6
8fga1kb0IfifCycbnSuMvH1Y3nxaqSW1Xh5fGG4gDgXlYUzZb/bVMb
CPlAuLmiqft3XWPeNNXxUPcARVOxQd2m4qN+POono74YzMZBOl
rIxgt5GFzJbXAFjr1C+U3Xh72yLL8niktUIyQ/1dX5dafkXdvo2vQjdqu
5vSOu1LGodKll/QvGb7iQ/Xg8bFUHttuQp6x8QgGGyyQl5HCZFIIFF
ZuuvDn3GO5w+p/qcHPM4xkek+HDsQqdwxKbLccfOm6FpJMqltcr
WFfOfinFK/v4k2ROtrq/UcagG3uQJ3XBYdfpIYKo/33/uqnKYdki80a
25tjZ9wfM+R1Ztal3lbIRYUMVL9nF3bY53bhQyByv/5xbKohOge3O
egQ6Kh2Izs63W9daGtJsoGKWhlkKDyAxHdb5MrYUtt261lJhsDrVv
KU8WMmCFN2DG/uTEfItRTpd2Y+1Nm/DwOji7mIpbXABEGLrmif/
f/FczR8F38qf7RCKh6ZU7jQmjv5qfXWnulpV/mSg44/NsXfko0OD5+
SdNPtNXf6sdph33klK/QYVcaQXA0tV6ANudPMeaUlR8F80zM2Wat
epAIhTxvnB5wzoWzxtZb9XygzecIdiTGaNCeqv8IpRKVvUDhrz4jSm
V9KDPKES1MPD1Lq3XNrWF51uKcphi2XqTl0CudQ9MSrHh5wSAlp
YUMZFlxhyB6aKo9k3nX1zk4ZmKA+cUPOe3oyDQzvMpKfWI7FSlTr
gexwYG/H18aA6elceaG3qkNXR2xjPvPHkeWi2v2G6fRwxF+hx/SOH
AmTV7iW9WfKQP8SScZ7xcBTkmTLVKGVb7j8MoTM4Ep1rYcDUr1
wYO1uYfeM/28M/So/e4A9Cs/HQfMkmwL4KjGwm+QBYbv6y9wk

	s/tbh1kexO4Ylzy0u1Ph8coGGfSIU7AkUNrQQ2+gJKvxDqOC7P9lCO dBxTMezjyLcB34PJ6cjnH37/hLNI5SL5nCQdQm1vZm90V1RqehyJZ DMwi45BZtD52jCQuGYeRZPHICnYBSrxT9zwMfi8pngX1K32WOKrD GwCMZLLFLnO12Wyl2RmlYW2iB0PF/4Ovd77Rj0Lr3SVUbf6sKv6p 2q79EqvBrAifmfx84sYB9mTgjk1LmF+6n3fHAM5l5pOn2CTaDfBKo NXmmmwvYSz3QjAq9N6nuPnDsfJyJ7+fC/gL36E1BLBwjrZdu5IwcA AM0TAABQSwECFAAUAAgICAB0oppU1je9uRkAAAAXAAAAFgAAA AAAAAAAAAAAAAAAAAAZ2VvZ2VicmFfamF2YXNjcmlwdC5qc1B LAQIUABQACAgIAHSimlTgJDGpHgUAAIUmAAAXAAAAAAAAAAAAA AAAAAF0AAABnZW9nZWJyYV9kZWZhdWx0czJkLnhtbFBLAQIUAB QACAgIAHSimlQ83VROawMAAKARAAAXAAAAAAAAAAAAAAAAAAAA MAFAABnZW9nZWJyYV9kZWZhdWx0czNkLnhtbFBLAQIUABQACA gIAHSimlTrZdu5IwcAAM0TAAAMAAAAAAAAAAAAAAAAAAHAJAAB nZW9nZWJyYS54bWxQSwUGAAAAAAQABAAIAQAAzRAAAAAA "/> <param name="image" value="http://www.geogebra.org/webstart/loading.gif" /> <param name="boxborder" value="false" /> <param name="centerimage" value="true" /> <param name="java_arguments" value="-Xmx512m" /> <param name="cache_archive" value="geogebra.jar, geogebra_main.jar, geogebra_gui.jar, geogebra_cas.jar, geogebra_export.jar, geogebra_properties.jar" /> <param name="cache_version" value="3.2.36.0, 3.2.36.0, 3.2.36.0, 3.2.36.0, 3.2.36.0, 3.2.36.0" /> <param name="framePossible" value="false" /> <param name="showResetIcon" value="false" /> <param name="showAnimationButton" value="true" /> <param name="enableRightClick" value="false" /> <param name="errorDialogsActive" value="true" /> <param name="enableLabelDrags" value="false" /> <param name="showMenuBar" value="false" /> <param name="showToolBar" value="false" /> <param name="showToolBarHelp" value="false" /> <param name="showAlgebraInput" value="false" /> Sorry, the GeoGebra Applet could not be started. Please make sure that Java 1.4.2 (or later) is installed and active in your browser (Click here to install Java now) </applet>
Para proceder a um link usam-se as marcas: ***<a>...*** A marca ***span*** permite alterar a formatação de uma parte do documento. Fecho do documento.	 José Manuel dos Santos dos Santos, Criado com GeoGebra </td></tr> </table></body> </html>

As marcas que delimitam a chamada da *applet* de GeoGebra, assim como os diferentes parâmetros aplicados, iniciam-se com:

```
<applet name="ggbApplet" code="geogebra.GeoGebraApplet" archive="geogebra.jar"
codebase=" http://geogebra.ese.ipp.pt/ficheiros_on_line/app/"
width="547" height="329" MAYSCRIPT>
```

e terminam com

```
</applet>
```

Antes e depois da marca relacionada com a *applet*, devemos colocar os textos que são ajustados para a exploração da nossa atividade. Este texto poderá ser editado com as *marcas* adequadas ou, em alternativa, com um editor de texto elementar.

Para inserir um botão que permita alterar o valor do raio da circunferência numa página Web vamos usar o ***JavaScript*****.** Para tal teremos de introduzir um formulário **<form>** com um botão de entrada, imediatamente antes da *marca* da *applet,* como por exemplo:

```
<form>
<p align=center> Raio: <b>r=</b>
<input type="text" name="T1" size="20" value="2">
<input type="button" value="Altera" name="B1"
onclick="document.ggbApplet.evalCommand('r='+T1.value);">
<input type="button" value="Reinicia" name="B2" onclick="T1.value='2';document.ggbApplet.reset();">
</p>
</form>
```

Em rigor, no texto anterior, o JavaScript limita-se às ações atribuídas aos botões, ou seja:

```
document.ggbApplet.evalCommand('r='+T1.value);
T1.value='2';document.ggbApplet.reset();
```

O JavaScript é uma linguagem de manipulação de objetos. A sintaxe de um objeto de Javascript para comunicar com a applet do GeoGebra será algo do tipo:

document.nomedaapplet.metodo

A primeira parte, ***document***, refere-se ao próprio documento HTML; segue-se o nome da applet, ***ggbApplet*** no exemplo anterior e, por último, o valor correspondendo a um comando ou método. Para executar a ação esta deve terminar com ponto e vírgula.

No exemplo anterior foi usado o comando ***reset()*** que é um dos métodos essenciais para a intercomunicação com a linha de comandos do GeoGebra o ***evalCommand()***.

document.ggbApplet.evalCommand('r='+T1.value);

document.ggbApplet.reset();

evalCommand() – é um método que permite dar entrada do texto <<**'r='+T1.value**>> na *applet* do GeoGebra. Assim atribui-se ao valor de ***r***, na *applet* que foi construída, o valor correspondente ao conteúdo da caixa de texto ***T1*** , para nos referirmos a este valor teremos de usar ***T1.value***.

reset() - é um comando que permite reiniciar a *applet.*

Finalmente ao abrir o ficheiro HTML observará uma caixa de texto e um botão. Quando na caixa de texto for alterado o valor do raio e for pressionado o botão *Altere* será alterada a circunferência. Ao pressionar o botão *Reinicia* o valor da applet volta ao início. Para mais detalhes para embutir applets do GeoGebra consulte: https://wiki.geogebra.org/en/Reference:GeoGebra_Apps_Embedding .

Capítulo III – Adição de números inteiros até 10, tarefas para o início da adição de inteiros

Cristina da Silva Ferreira Alves

Resumo:

Apresentamos uma aplicação de uma tarefa com feedback automático, visando o conteúdo da adição com números inteiros até 10 a partir de um problema de palavras, para uma aula de matemática de alunos do 1° ano de escolaridade, e usando tablets. Apresentamos as dimensões didáticas e tecnológicas que estiveram na base da sua criação, nomeadamente se apresentam os procedimentos necessários para a sua construção.

ENQUADRAMENTO

A tarefa criada visou o conteúdo da Adição com números inteiros até 10. Do ponto de vista curricular e de acordo com as aprendizagens essenciais, esta escolha enquadra-se nos temas: números e operações, onde se pretende que o aluno desenvolva o sentido do número e compreenda os números/ as operações e a fluência do cálculo mental e escrito. Desenvolve também a capacidade de resolução de problemas onde se pretende que o aluno desenvolva a capacidade de resolver problemas e raciocinar matematicamente.

DESCRIÇÃO DA TAREFA

A tarefa *Resolução do Problema dos Berlindes*, com a utilização do recurso digital GeoGebra, pretende que o aluno encontre o número que transforma numa proposição verdadeira, uma expressão do tipo **"a + b= __"**. Tendo em conta as considerações anteriores a tarefa foi elaborada com base na matriz de análise da Figura 13.

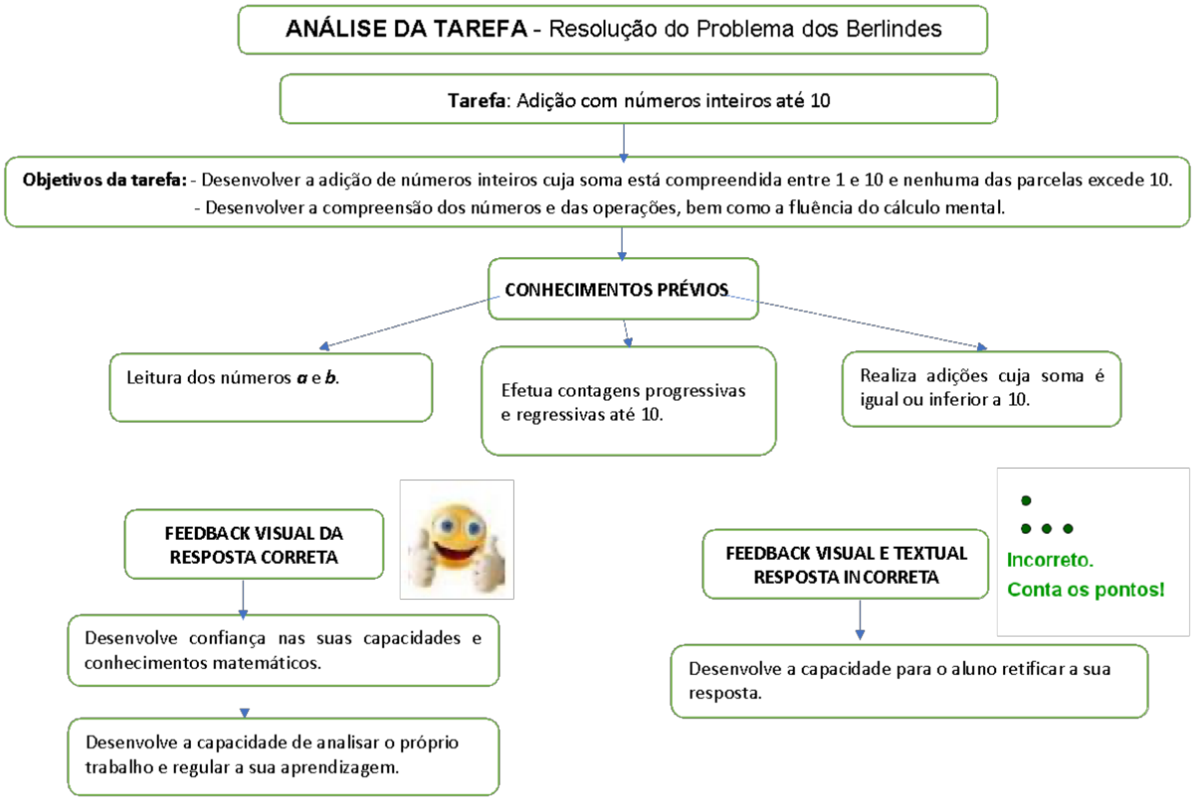


Figura 13 – Matriz de análise da tarefa "Resolução do Problema dos Berlindes"

No GeoGebra, a tarefa foi realizada com base no diagrama de processo da Figura 14, onde se ilustram as tomadas de decisão em relação à programação necessária.

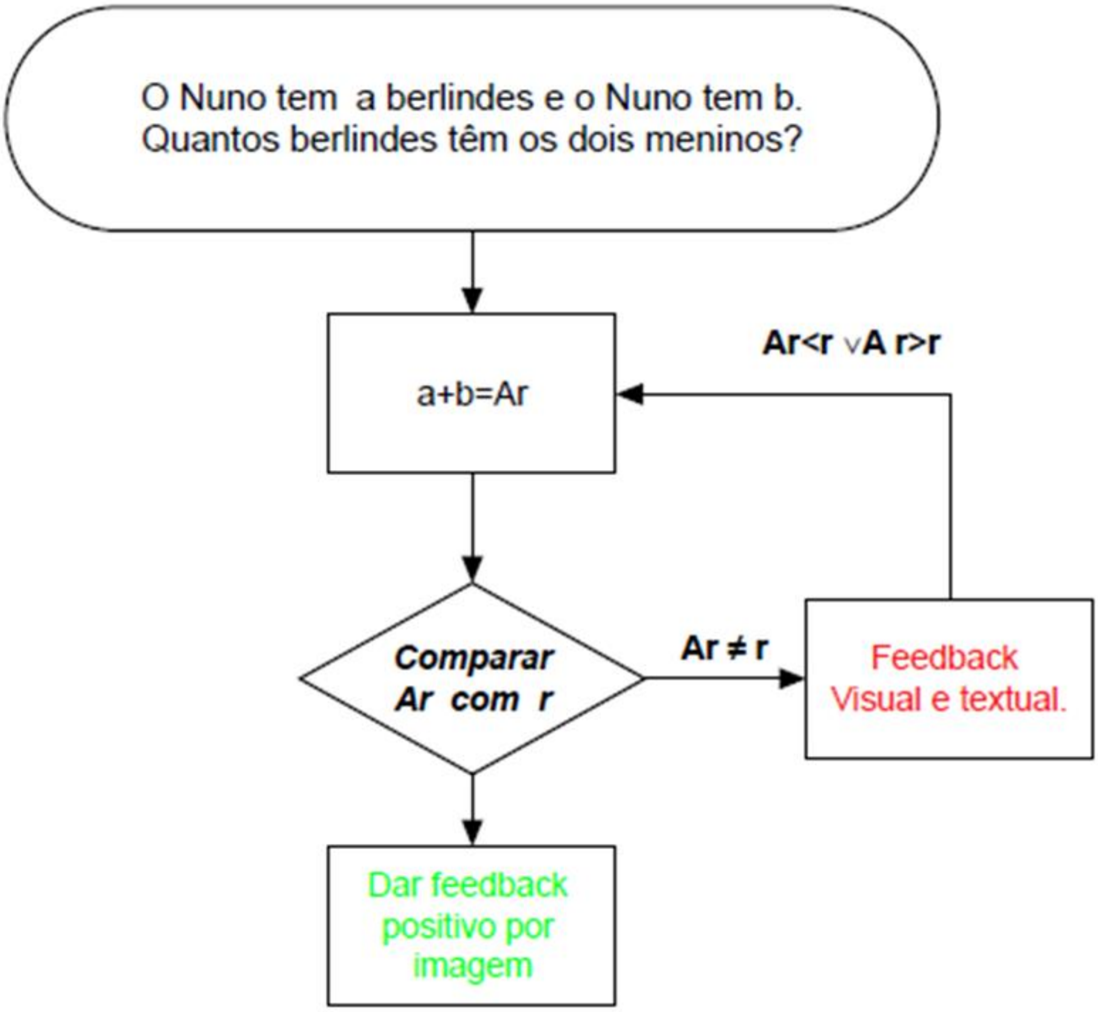


Figura 14 – Diagrama de processo da tarefa "Resolução do Problema dos Berlindes"

Guião de construção da aplicação

A aplicação anteriormente apresentada foi realizada com base no guião de construção da Tabela 2. Observe-se que seguido o guião facilmente pode adaptar-se o esquema utilizado para outros problemas de palavras simples, envolvendo porventura outras operações. Apesar desta tarefa não prever a introdução de feedback auditivo, este pode facilmente ser introduzido utilizando um processo semelhante ao referido na secção X.

Tabela 2 - Protocolo de construção da aplicação "Adição de números inteiros até 10"

Passo	Objetivo	Comando a escrever na barra de Entrada ou em separadores
1.º	Escolha de a .	a=AleatórioInteiroEntre (1,5)
2.º	Escolha de b .	b=AleatórioInteiroEntre (1,5)
3.º	Definir a variável de controlo da resposta r.	r=a+b
4.º	Definir alvo da resposta.	Ar=undefined
5º	Definir texto do enunciado.	aT= "O Nuno tem a berlindes e o Pedro tem b berlindes. Quantos berlindes têm os dois meninos?" Formatar objeto: fixar objeto; mostrar objetos; texto grande; cor azul, alinhado à direita; fixar objeto; posição absoluta no ecrã.
6.º	Definir caixa de resposta.	CAr=CaixadeEntrada(Ar) Formatar objeto: sem mostrar rótulo; fixar objeto; mostrar objetos objeto auxiliar texto muito grande; comprimento da caixa de entrada 2; alinhado à direita; posição absoluta no ecrã; no avançado não permitir a seleção.
7.º	Definir o ponto C.	C =valor (-3396.46373,5.8645) Formatar objeto: mostrar rótulo; cor verde; tamanho do ponto 5; coordenadas cartesianas.
8.º	Definir o ponto D.	D: valor (-3404.27, - 1320.38) Formatar objeto: mostrar rótulo; cor verde; tamanho do ponto 5; coordenadas cartesianas.
9.º	Definir os berlindes do Nuno.	Bn= Sequência(Translação(C, Vetor(Vetor((0, 0), (1000, 0)) i)), i, 1, a) Formatar objeto: mostrar rótulo; cor verde; tamanho do ponto 5; no avançado escrever a condição Ar > r V Ar < r
12.º	Definir os berlindes do Pedro.	Bp= Sequência(Translação(D, Vetor(Vetor((0, 0), (1000, 0)) i)), i, 1, b) Formatar objeto: mostrar rótulo; cor verde; tamanho do ponto 5; no avançado escrever a condição Ar > r V Ar < r
13º	Definir feedback textual associado à resposta incorreta	fRI= "Incorreto. Conta os pontos!" Formatar objeto: fixar o objeto; texto médio; cor

		verde; negrito; posição absoluta no ecrã; no avançado escrever a condição Ar>r ∨ Ar < r; não permitir a seleção.
14.º		iRC 
	Definir feedback visual por imagem associado à resposta correta	Formatar objeto: posição absoluta no ecrã; no avançado escrever a condição Ar $\stackrel{?}{=}$ r e não permitir a seleção.
15.º	Definir botão para novo desafio	B1=Botão("Novo desafio") Formatar objeto: mostrar objetos; mostrar rótulo; fixar objeto; texto médio; cor azul; posição absoluta no ecrã; na programação em Se clicar escrever as duas linhas do seguinte texto: AtualizarConstrução() DefinirValor(Ar,undefined)

Capítulo IV - Construção de tarefas para trabalhar aritmética elementar e pré álgebra o caso da adição com lacunas de números inteiros até 20

Idalinda Pereira da Cunha

Resumo:

Neste capítulo ilustra-se o design de uma tarefa no GeoGebra para desenvolver a adição de números inteiros positivos cuja soma está compreendida entre 1 e 20 e nenhuma das parcelas excede 20. Esta tarefa apresentará feedback automático aos alunos no sentido que eles possam progredir e fazer aprendizagens de modo autónomo.

ENQUADRAMENTO

A tarefa que a seguir se apresenta, com recurso ao GeoGebra*,* enquadra-se no tema Números e Operações para o 1º Ano de escolaridade. Para a realização desta tarefa pressupõe-se alguns conhecimentos prévios, neste sentido o aluno deverá ser capaz de: i) ler e representar números no sistema de numeração decimal até 20; ii) comparar números inteiros até 20; iii) já ter realizado adições, com lacunas, cuja soma é igual ou inferior a 5. De facto, previamente à realização da tarefa com o GeoGebra há o desenho de uma THA, que origina uma matriz de análise que se apresenta na Figura 15.

Nesta matriz de análise consta o objetivo de modo claro, uma inventariação dos objetivos prévios em função da tarefa a construir, e o desenho dos feedbacks que serão dados em função das respostas dos alunos.

DESCRIÇÃO DA TAREFA

A tarefa *Adição com lacunas de números inteiros até 20*, com a utilização do recurso digital GeoGebra, pretende que o aluno encontre o número que transforma numa proposição verdadeira, uma expressão do tipo **“a +___= b”**.

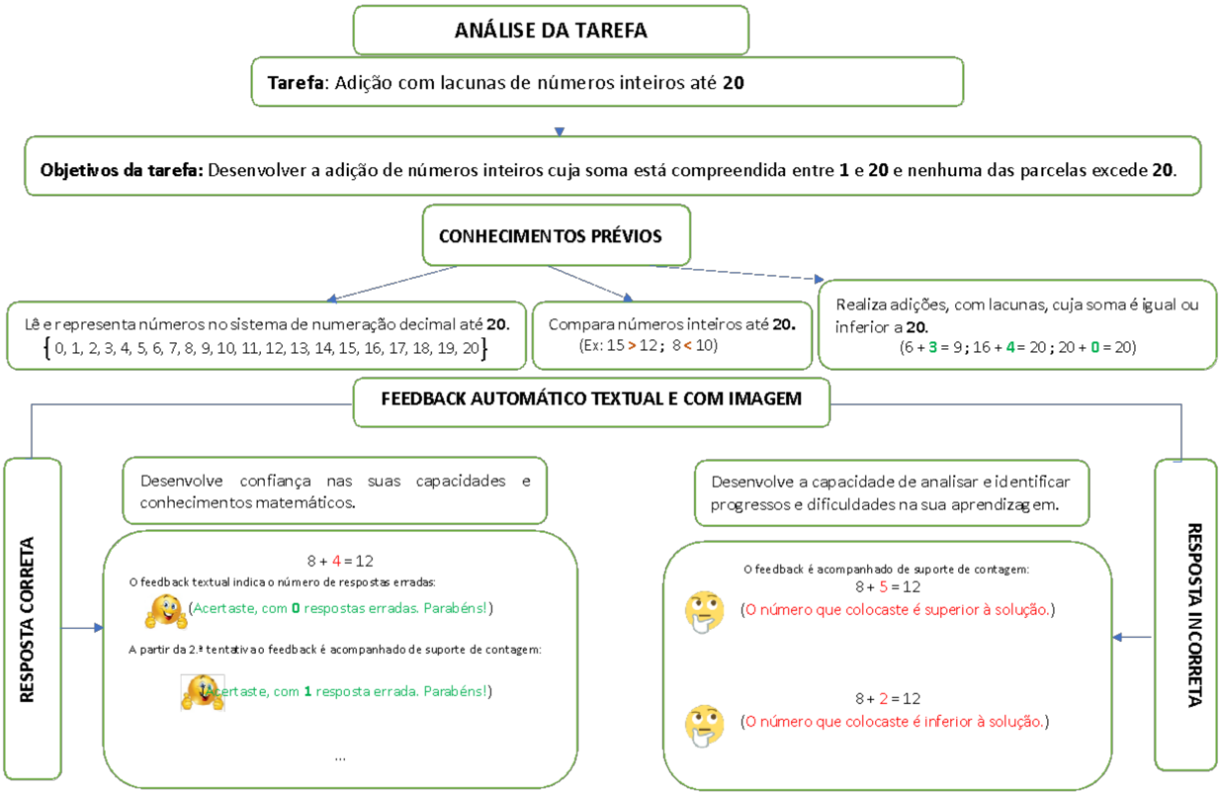


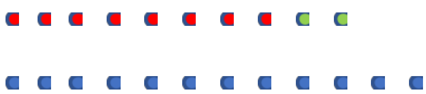
Figura 15 – Matriz de análise da tarefa "Adição com lacunas de números inteiros até 20"

O feedback foi pensado de modo que em cada desafio, sempre que o aluno escrever uma resposta correta recebe um reforço positivo, através de feedback automático por imagem e texto:

 "Acertaste, com 0 respostas erradas. Parabéns!

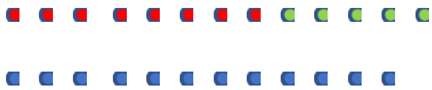
Se o aluno apresentar uma resposta inferior à solução, para o auxiliar na descoberta do valor correto de forma autónoma, recebe um feedback automático por imagem e texto acompanhado de suporte de contagem:

 "O número que colocaste é inferior à solução".



Se o aluno apresentar uma resposta superior à solução recebe um feedback automático por imagem e texto acompanhado de suporte de contagem:

"O número que colocaste é superior à solução".



À medida que o aluno interage com a aplicação é contado o número de erros. Quando o aluno descobre a solução, após respostas incorretas, recebe um feedback automático de imagem e texto com a indicação do número de tentativas falhadas. A tarefa foi realizada com base no diagrama de processo da Figura 16, onde se ilustram as tomadas de decisão em relação às decisões de programação que vão ser tomadas.

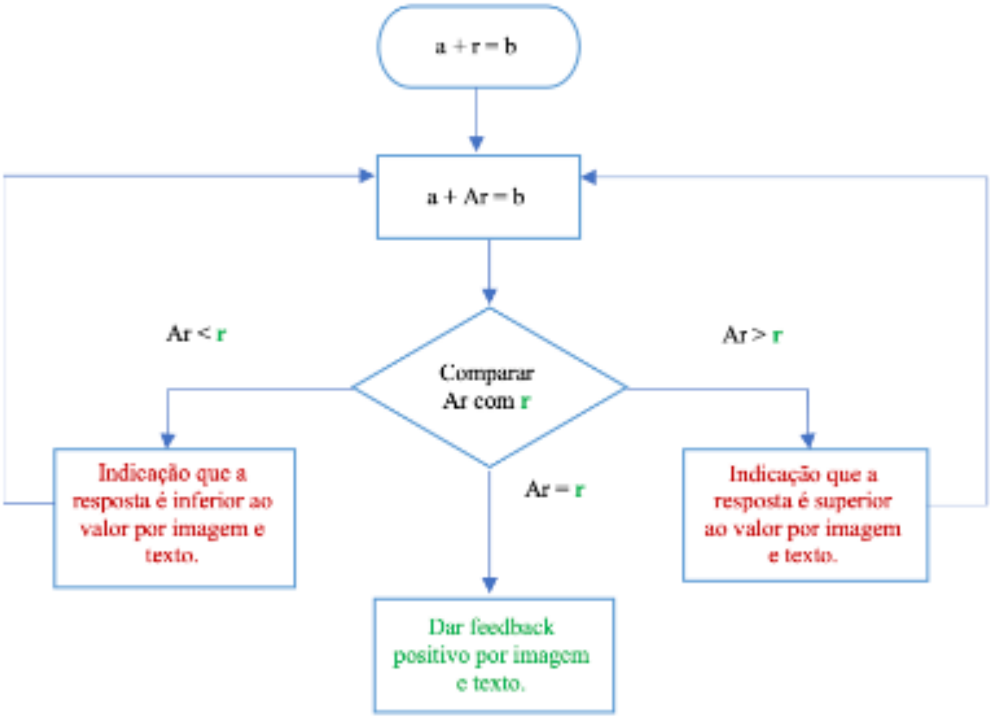


Figura 16 – Diagrama de Processo da aplicação "Adição com lacunas de números inteiros até 20"

A resposta ao problema é dada pelo valor da variável interna r, calculado pelos valores dos parâmetros *a* e *b*, gerados aleatoriamente, tomando valores entre 0 e 10. Assim, para certificação da resposta do utilizador, guardada na variável *Ar*, a aplicação compara Ar com r e em função desta comparação apresenta um feedback.

Guião de construção da aplicação

A tarefa pode ser construída no GeoGebra com base no protocolo que consta da Tabela 3, onde se referem os 20 passos necessários para a sua construção, apresentando o objetivo e o código para cada um dos passos.

Tabela 3 - Protocolo de construção da aplicação "Adição com lacunas de números inteiros até 20"

Passo	**Objetivo**	**Comando a escrever na barra de Entrada ou em separadores**
1.º	Escrever enunciado da tarefa	Escrever o texto do enunciado da tarefa **Formatar objeto**: texto grande; cor azul, alinhado à direita; fixar objeto; posição absoluta no ecrã.
2.º	Escolha de **a**	**a=AleatórioInteiroEntre(0,20)**
3.º	Escolha de **b > a**	**b=AleatórioInteiroEntre(a,20)**
4.º	Definir a variável de controlo da resposta **r**	**r=b-a**
5.º	Definir alvo da resposta	**Ar=undefined**
6.º	Definir caixa de resposta	**Car=CaixadeEntrada(Ar)** **Formatar objeto**: sem mostrar rótulo; texto muito grande; comprimento da caixa de entrada 2; alinhado à direita; posição absoluta no ecrã.
7.º	Definir texto para mostrar a expressão antes da caixa de resposta	**aT="a+"+"** **Formatar objeto**: texto muito grande; alinhado à direita; posição absoluta no ecrã; no avançado não permitir a seleção.
8.º	Definir texto para mostrar a expressão	**bT="="+b**

	antes da caixa de resposta	**Formatar objeto**: texto muito grande; alinhado à direita; posição absoluta no ecrã; no avançado não permitir a seleção.
9.º	Definir feedback visual por imagem associado à resposta correta	Inserir imagem **Formatar objeto**: nome **iRC**; posição absoluta no ecrã; no avançado escrever a condição **Ar==r** e não permitir a seleção.
10.º	Definir feedback textual associado à resposta correta	**fRC="Acertaste, parabéns!"** **Formatar objeto**: texto grande; cor verde; posição absoluta no ecrã; no avançado escrever a condição **Ar==r** e não permitir a seleção.
11.º	Definir feedback visual por imagem associado à resposta superior à solução	**Inserir imagem** **Formatar objeto**: nome **iR11**; posição absoluta no ecrã; no avançado escrever a condição **Ar>r** e não permitir a seleção.
12.º	Definir feedback textual associado à resposta superior à solução	**fR11="O número que colocaste é superior à solução."** **Formatar objeto**: texto grande; cor vermelha; posição absoluta no ecrã; no avançado escrever a condição **Ar>r** e não permitir a seleção.
13.º	Definir feedback visual por imagem associado à resposta inferior à solução	**Inserir imagem** **Formatar objeto**: nome **iR12**; posição absoluta no ecrã; no avançado escrever a condição **Ar<r** e não permitir a seleção.
14.º	Definir feedback textual associado à resposta inferior à solução	**fR12="O número que colocaste é inferior à solução."** **Formatar objeto**: texto grande; cor vermelha; posição absoluta no ecrã; no avançado escrever a condição **Ar<r** e não permitir a seleção.

15	Número de erros	**nre = 0**
16.º	Definir botão para novo desafio	**B1=Botão("Novo desafio")** **Formatar objeto**: texto médio; cor castanho; posição absoluta no ecrã; na programação em ***Se clicar*** escrever as duas linhas do seguinte texto: **AtualizarConstrução()** **DefinirValor(Ar,undefined)** **DefinirValor(nre,0)**
17	Feedback visual do número **a**	Sequência(Translação((3, 2), Vetor(k Vetor((0, 0), (0.3, 0)))), k, 0, a - 1)
18	Feedback visual do número **b**	Sequência(Translação((3, 2 - 0.3), Vetor(k Vetor((0, 0), (0.3, 0)))), k, 0, b - 1)
19	Feedback visual do número **r**	Sequência(Translação((3, 2), Vetor(k Vetor((0, 0), (0.3, 0)))), k, a, a + Ar - 1)
20	Feedback textual da resposta correta e do número de tentativas falhadas	**fRC= Se(Ar==r, "Acertaste, com "+nre+" respostas erradas. Parabéns!")**

Capítulo V– Divisores de um número natural de 2 a 50

Arlindo Tavares Semedo da Veiga e Astrigilda Pires Rocha Silveira

Resumo:

Apresentamos uma aplicação de uma tarefa com feedback automático, que visa o conteúdo dos divisores de um número natural entre 2 e 50, a partir da resposta do utilizador, no caso de erro são dados uma série de feedbacks no sentido que o utilizador encontre a resposta correta fazendo apelo a conceitos relacionados com a divisão inteira. São apresentadas diversas considerações que levaram à conceção desta tarefa, nomeadamente as formas de feedback usadas e os passos necessários para a sua construção da aplicação no GeoGebra.

ENQUADRAMENTO

As duas tarefas de feedback automático, suportadas por aplicações no GeoGebra e representadas por fluxogramas, com enquadramento no domínio de Números e Operações, visam potenciar o reforço das capacidades transversais em matemática, atribuição de significados aos conceitos matemáticos em estudo e promoção do envolvimento, entusiasmo, reflexão e evolução dos alunos em prol de uma aprendizagem significativa da matemática.

DESCRIÇÃO DA TAREFA

Os divisores de um número natural é um dos temas do 4º ano do Ensino Básico Obrigatório (EBO) de Cabo Verde. O objetivo da tarefa para os alunos é reconhecer um número natural como divisor de um outro número natural (2 a 50). A tarefa foi pensada com base no diagrama de processo da Figura 17.

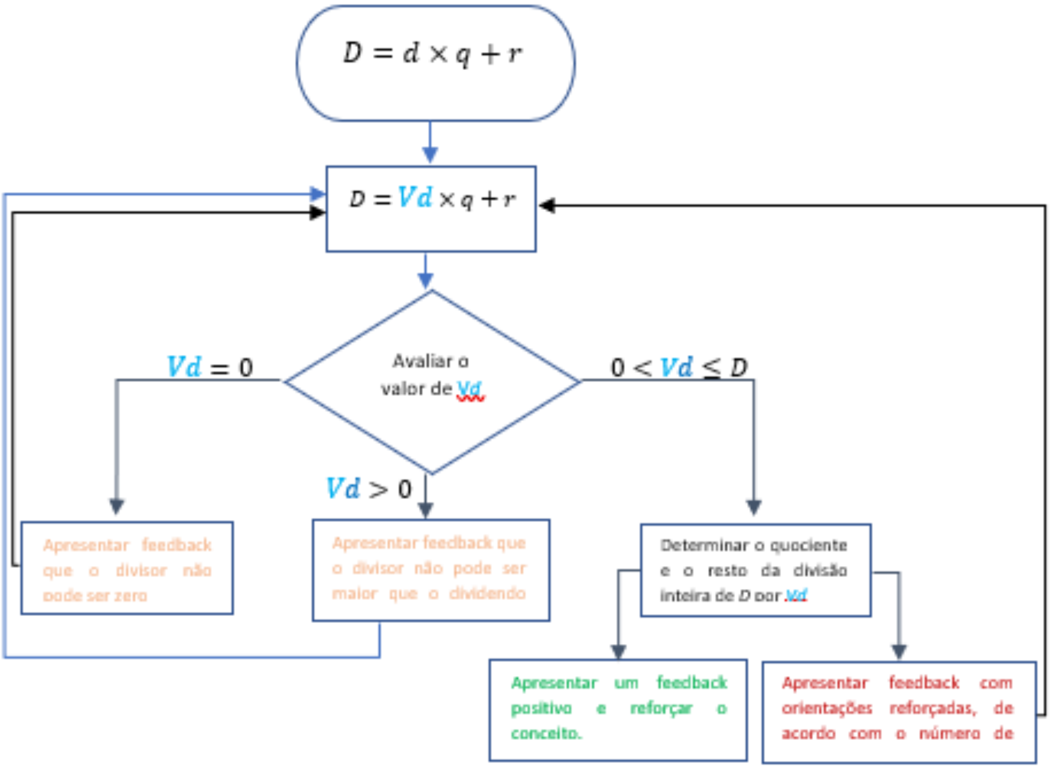


Figura 17 – Diagrama de Processo da aplicação "Divisores de um número natural de 2 a 50"

Tipos de feedback

A tarefa apresenta vários tipos de feedback, visual e textual em função da resposta do utilizador:

a) Feedback visual (imagens)

- Para resposta corneta, surge a imagem

- Para resposta incorreta, surge a imagem

- Para resposta incorreta pela 3ª vez, aparece um conjunto com ***D*** pontos distribuídos em ***q*** grupos com ***p*** pontos, seguidos de um grupo com ***r*** pontos de cores diferentes, conforme a Imagem que se segue:

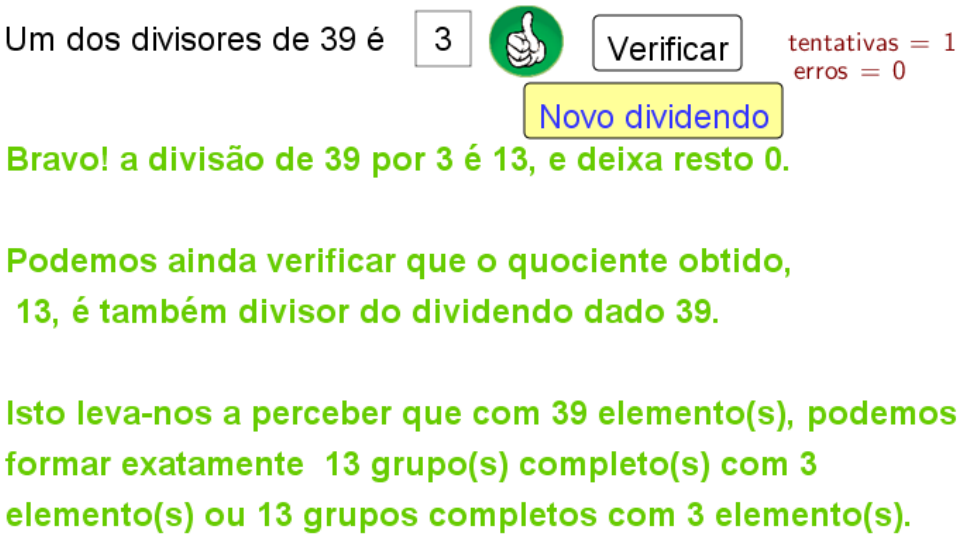
b) Feedback textual – apresenta textos conforme as determinadas condições:

i. Resposta correta:

"Bravo! A divisão de D por d é q , e deixa resto 0.

Podemos ainda verificar que o quociente obtido, q , é também divisor do dividendo dado D .

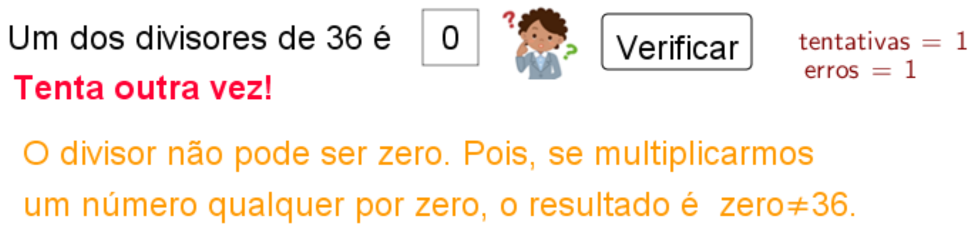
Isto leva-nos a perceber que com D elemento(s), podemos formar exatamente q grupo(s) completo(s) com d elemento(s) ou d grupos completos com q elemento(s)."



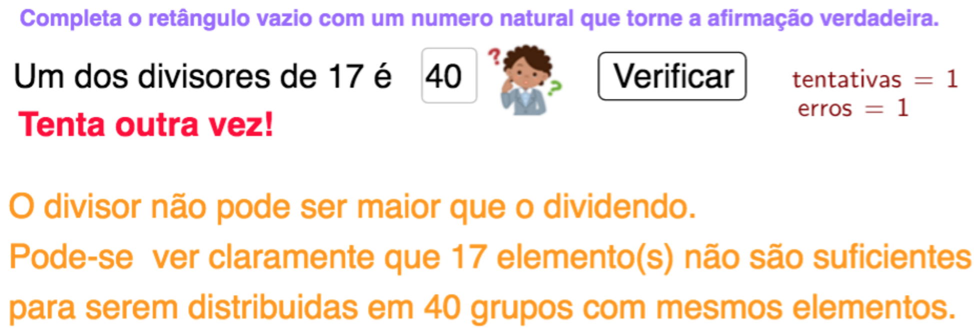
ii. Resposta incorreta:

1. Se o número introduzido for 0 (zero), apresenta o texto:

"o divisor não pode ser zero. Pois se multiplicarmos um número qualquer por zero o resultado é 0 ≠D "



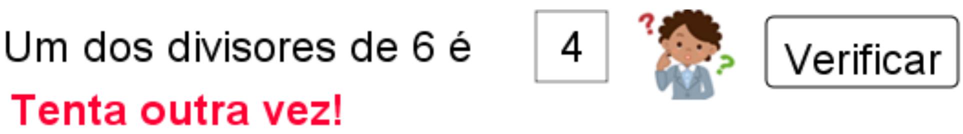
2. Se o número introduzido, *d*, for superior ao dividendo, *D*, apresenta o texto: "o divisor não pode ser superior ao dividendo. Pode-se ver claramente que *D* elementos não são suficientes para serem distribuídos em *d* grupos com mesmos elementos"



3. Se o número introduzido for diferente zero, menor ou igual ao dividendo e não for divisor deste, apresenta textos diferentes, conforme o número de erros:

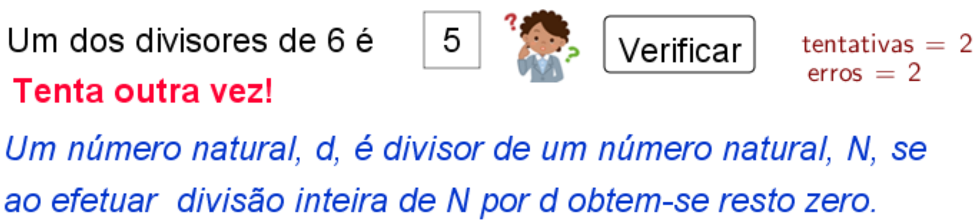
Feedback para 1º até o 3º erro

"Tenta outra vez!"



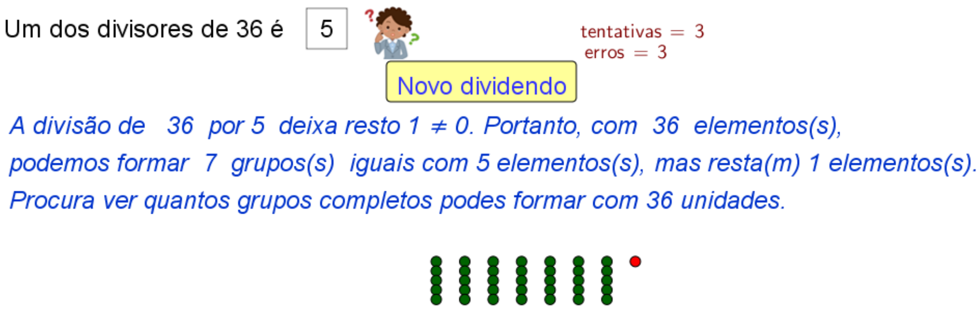
Feedback para 2º erro:

"Um número natural, *d*, é divisor de um número natural, *n*, se ao efetuar divisão inteira de *n* por *d* obtém-se resto zero."



Feedback para 3º erro:

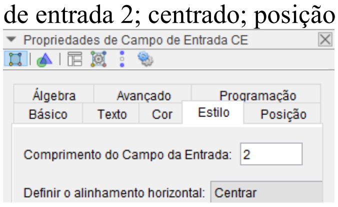
"A divisão de D por d deixa resto $r \neq 0$. Portanto, com D elementos(s), podemos formar q grupos(s) iguais com d elementos(s), mas resta(m) r elementos(s). Procura ver quantos grupos completos podes formar com D elementos."

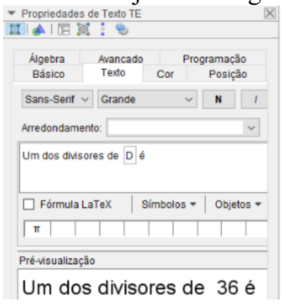
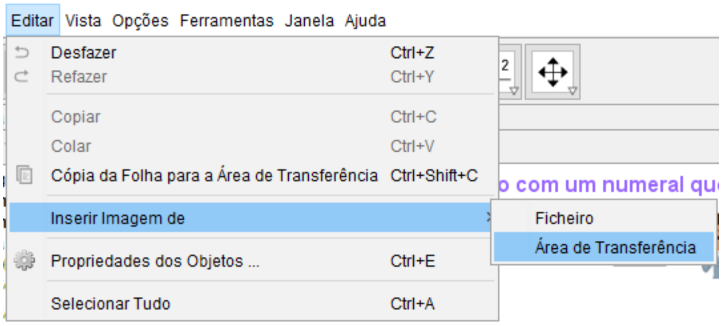
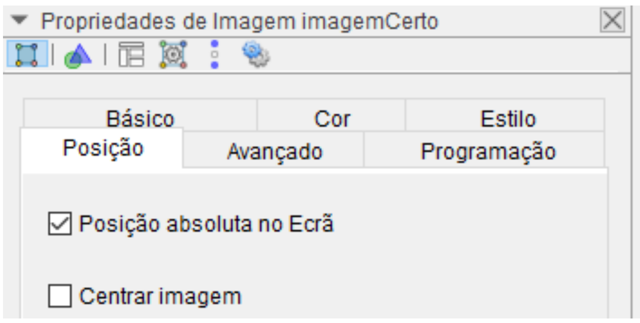


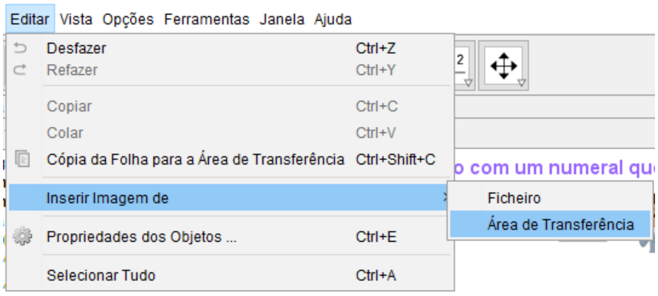
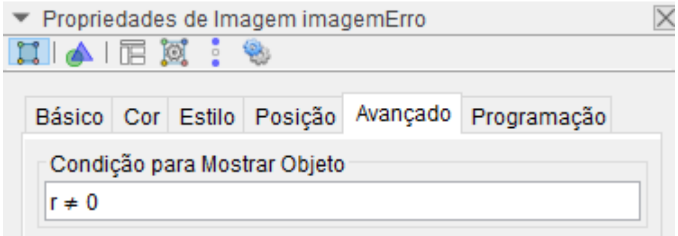
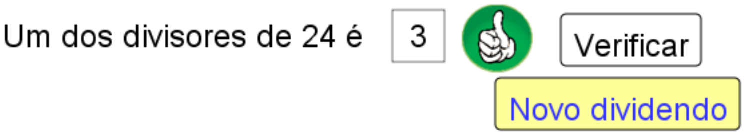
Guião de construção da aplicação

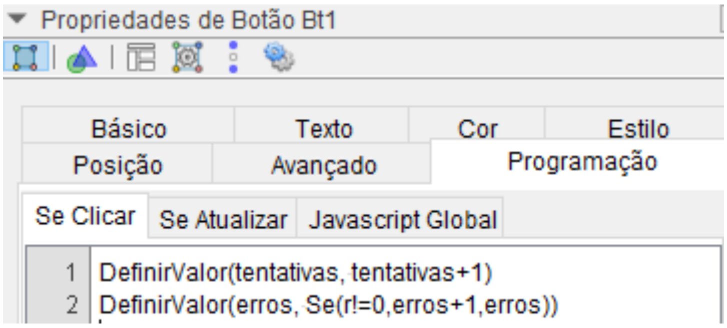
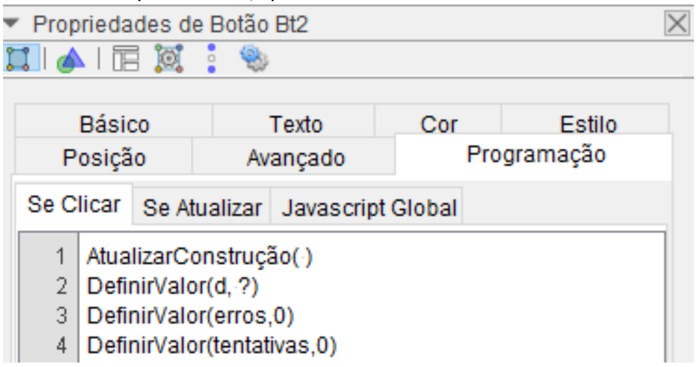
A tarefa pode ser construída no GeoGebra com base no protocolo que consta da Tabela 4, onde se referem os 19 passos necessários para a sua construção, apresentando o objetivo e o código para cada um dos passos.

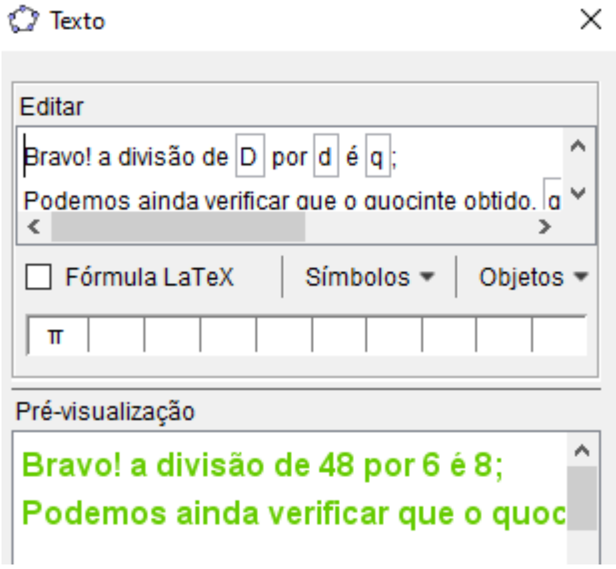
Tabela 4 - Protocolo de construção da aplicação "Divisores de um número natural de 2 a 50"

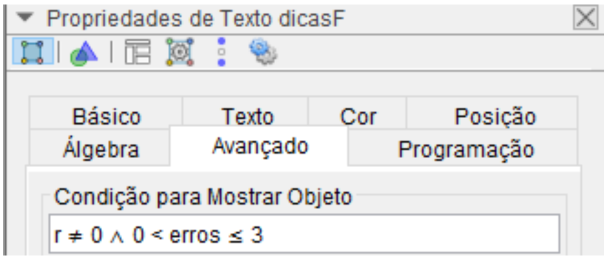
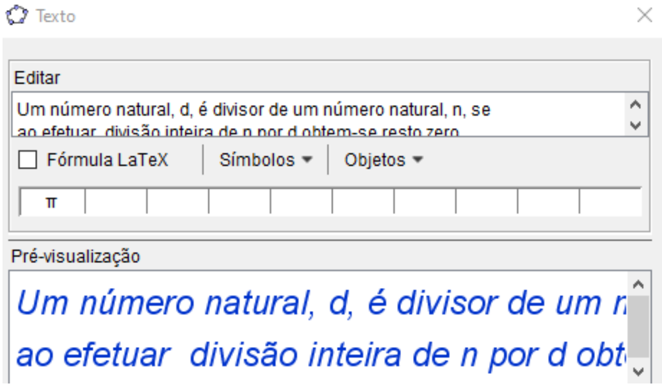
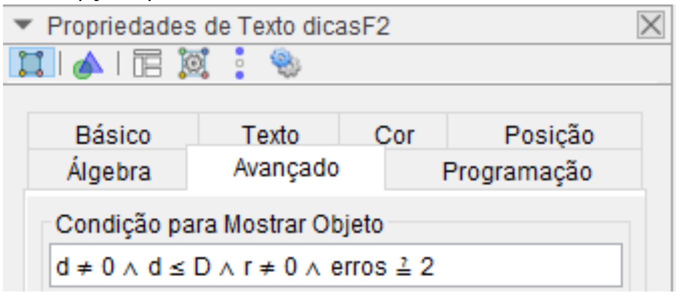
Passo	Objetivo	Comando a escrever na barra de Entrada ou em Separadores
1	Escolher divisor (D)	Digitar: D=AleatórioInteiroEntre(2, 50)
2	Definir o elemento da resposta (d)	Digitar: d=?
3	Definir a caixa da resposta associada ao elemento d	Digitar: resposta=CaixadeEntrada(d) Formatar objeto: sem mostrar rótulo; texto grande; comprimento da caixa de entrada 2; centrado; posição absoluta no ecrã. 
4	Definir as variáveis de controlo (q e r)	Digitar: q=Quociente(D,d) r=Resto(D,d)
5	Definir a expressão para	Digitar: TE="Um dos divisores de" D "é"

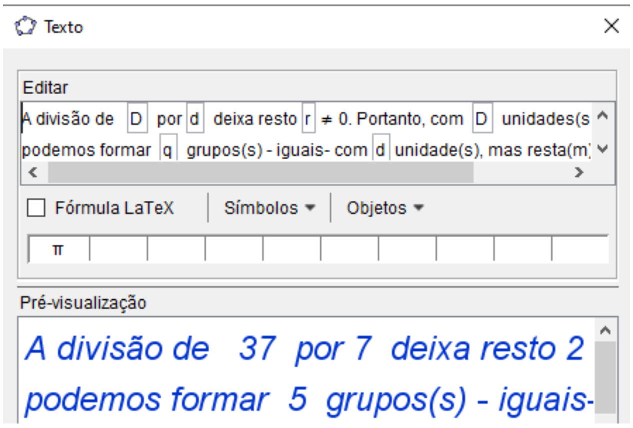
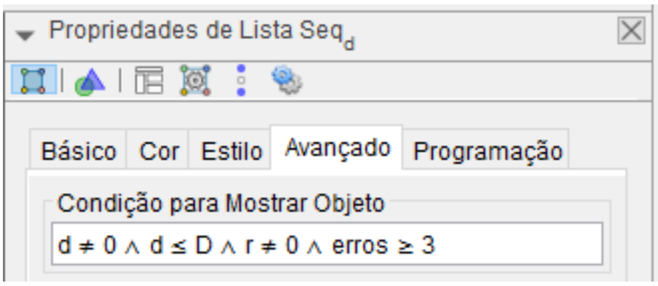
	mostrar o texto antes da caixa de entrada	Formatar objeto: texto grande; alinhado.  Alinhar o texto com a caixa de entrada, como se segue: 
6	Definir feedback visual, por imagem, associado à resposta correta.	Procurar a imagem  e copia; No menu Editar, inserir imagem de área de transferência;  Localizar a imagem à frente da caixa de entrada; Definir as propriedades: nome: “imagemCerto”; Posição absoluta no ecrã; ativar a opção “avançado” e escrever a condição: r==0. 
7	Definir feedback visual, por imagem, associado às respostas incorretas	Procurar a imagem  e copia; No menu Editar, inserir imagem de área de transferência;

		 Localizar a imagem à frente da caixa de entrada; Definir as propriedades: nome: “imagemCerto”; Posição absoluta no ecrã; ativar a opção “avançado” e escrever a condição r<>0 ($r \neq 0$). 
8	Construir dois botões: O botão “*Verificar*”, com nome verificador e botão “*Novo dividendo*”, com nome recomeçar.	Clicando na ferramenta “Botão”, criar os dois botões e posicioná-los adequadamente, como indicado abaixo: 
9	Criar duas variáveis: (tentativas e erros)	Digitar na entrada: tentativas = 0 □para contar o número de tentativas erros = 0; □ para contar erros cometidos
10	Programação do botão “Verificar”	Clicando com o botão direito do rato sobre o botão “Verificar”, selecionar a opção propriedades, escolher “Programação”, selecionar “se clicar” e digitar: *DefinirValor(tentativas, tentativas+1);* *DefinirValor(erros, Se(r!=0,erros+1,erros));*

		 *Esse comando faz com que ao clicarmos no botão, a variável 'tentativas' é acrescida de mais uma unidade. No caso da variável, 'erros' é feito um teste condicional. Se o valor digitado pelo usuário for diferente (indicada pelo símbolo !=) da resposta correta, esse valor é somado uma unidade ao valor da variável (indicado pela sentença erros+1). Caso contrário, ou seja se o valor da resposta dada pelo usuário for igual a resposta correta (r=0), o valor da variável "erros" permanece o mesmo.* No "avançado" escrever a condição para mostrar: erros < 3 *Este comando faz com que o botão* "Verificar" *fique oculto depois do 3º erro.*
11	Programação do botão "Novo dividendo"	Clicando com o botão direito do rato sobre o botão "Novo dividendo" selecionamos a opção propriedades, escolher "Programação", selecionar "se clicar" e digitar: *AtualizarConstrução( )* ⍰isto gera outro dividendo; *DefinirValor(d, ?)* ⍰isto limpa a caixa de texto; DefinirValor(erros,0) ⍰isto atribui valor 0 à variável "erros"; DefinirValor(tentativas,0) ⍰isto atribui valor 0 à varável "tentativa"  No "avançado" escrever a condição para mostrar: r == 0 V erros==3 *Este comando faz com que o botão "Novo dividendo" aparece quando a resposta é correta ou já forram cometidos 3 erros.*
12	Definir feedback textual quando a resposta é correta	Através da ferramenta ABC, inserir o texto: Bravo! A divisão de D por d é q é deixa resto 0. Podemos ainda verificar que o quociente obtido, q , é também divisor do

		dividendo D. Isto leva-nos a perceber que que com D elemento(s), podemos formar exatamente d grupos completos com q elemento (s) ou q grupos completos com d elemento(s).  Propriedade do texto: Grande, Fixar Objeto, cor verde, nome: Tverd, condição para mostrar: *r==0.*
13	Definir feedback de texto quando o valor introduzido para divisor é zero	Escrever na entrada: *divZero = "O divisor não pode ser zero. Pois, se multiplicarmos um número qualquer por zero, o resultado é 0 ≠ D"* Propriedade do texto: Grande, Fixar Objeto, cor laranja, condição para mostrar: *d==0*,
14	Definir feedback de textual quando o valor introduzido para divisor é superior ao dividendo	dSupD = "O divisor não pode ser maior que o dividendo. Pode-se ver claramente que não é possível distribuir " D " elemento(s) em" d " grupos de " q " elemento(s)" Propriedade do texto: texto Grande, Fixar Objeto, cor laranja, condição para mostrar: d>D
15	Definir feedback textual quando a resposta introduzida é incorreta.	Através da ferramenta "Inserir texto" digitar o texto: Tenta outra vez! Localizar o texto no lugar conveniente; Definir propriedades: Nome: dicasF; texto: Grande; cor: vermelha; Fixar objeto; posição absoluta no ecrã; opções para mostrar: r ≠ 0 ∧ 0< erros < 3

		 Este texto só aparece quando a resposta está incorreta e o número de erro cometido for inferior ou igual a 3.
16	Definir feedback textual quando a resposta introduzida é incorreta pela segunda vez, mas diferente de zero e não superior ao divisor.	Através da ferramenta texto, digitar: Um número natural , d, é divisor de um número natural, N, se ao efetuar divisão inteira de N por d obtém-se resto zero.  Localizar o texto no lugar conveniente; Definir propriedades: nome dicasF2; texto Grande; cor azul; Fixar objeto; posição absoluta no ecrã; opções para mostrar: d ≠ 0 ∧ d ≤ D ∧ r ≠ 0 ∧ erros ==2 
17	Definir feedback textual quando a resposta introduzida é incorreta por 3 ou mais vezes, mas diferente de zero e não superior ao divisor.	Através da ferramenta texto, digitar: A divisão de D por d deixa resto r ≠ 0. Pode -se ainda verificar que com D elemento(s), pode-se formar q grupos(s) de d elemento (s), mas resta(m) r elemento(s). Procura ver quantos grupos completos podes formar com D elemento(s).

		 Localizar o texto no lugar conveniente; Definir propriedades: nome: “dicasF3”; texto Grande; cor azul; Fixar objeto; posição absoluta no ecrã; opções para mostrar: d ≠ 0 ∧ d ≤ D ∧ r ≠ 0 ∧ erros ≥ 3
18	Definir um feedback visual que permite observar a distribuição de *D* pontos por *d* grupos de *q* elementos, apresentando o resto $r \neq 0$ $r \neq 0$·	E=(-1,-1) ⍰ *ocultar* v=(0.6,0) ⍰ *ocultar* u=(0, -0.2) ⍰ *ocultar* w=((q + 1) x(v), 0) ⍰ *ocultar* Seq_q=Sequência(Translação(E, Vetor(i v)), i, 1, q, 1) Seq_d= Sequência(Translação(Sequência(Translação(E, Vetor(i v)), i, 1, q, 1), Vetor(j u)), j, 1, d - 1) Seq_r= Sequência(Translação(Translação(E, w), Vetor(j u)), j, 0, r - 1) Propriedades de objeto: cor verde para “Seq_q” e “Seq_d” e cor vermelha para “Seq_r” Condição para mostrar imagem gerada pelas sequências (Seq_d, Seq_q e Seq_r): d ≠ 0 ∧ d ≤ D ∧ r ≠ 0 ∧ erros ≥ 3  Exemplo de uma figura em que foi introduzido o número 7 para divisor de 37, errando pela terceira vez:  Obs.: Esta imagem aparece a partir do terceiro erro.

19	Finalizar as construções, de modo que os objetos criados fiquem fixos, numa posição absoluta e sem poderem ser selecionados.	Organizar todos os objetos para obter uma boa apresentação. Clicando com o botão direito do rato sobre um dos objetos na janela gráfica, na opção “propriedade dos objetos” certifique ou altere as propriedades de todos os objetos, com a exceção da caixa de entrada onde se introduz a resposta: ☑ Fixar Objeto , na opção “Básico” ☑ Posição absoluta no Ecrã , na opção “Posição” ☐ Seleção permitida , na opção “Avançado” Estes comandos servem para impedir que os objetos criados sejam deslocados ou excluídos dos lugares onde se encontram. Os mesmos não podem ser selecionados.

Capítulo VI – Adição de frações próprias, denominadores variando de 2 a 9

Arlindo Tavares Semedo da Veiga e Astrigilda Pires Rocha Silveira

Resumo:

Apresentamos uma aplicação de uma tarefa com feedback automático, que visa a adição de frações próprias, a partir da resposta do utilizador, no caso de erro, são dados uma série de feedbacks no sentido que o utilizador encontre a resposta correta fazendo apelo a conceitos relacionados com a adição de números racionais na forma de fração. São apresentadas diversas considerações que levaram à conceção desta tarefa, nomeadamente as formas de feedback usadas e os passos necessários para a sua construção da aplicação no GeoGebra.

ENQUADRAMENTO

As duas tarefas de feedback automático, suportadas por aplicações no GeoGebra e representadas por fluxogramas, com enquadramento no domínio de Números e Operações, visam potenciar o reforço das capacidades transversais em matemática, atribuição de significados aos conceitos matemáticos em estudo e promoção do envolvimento, entusiasmo, reflexão e evolução dos alunos em prol de uma aprendizagem significativa da matemática.

DESCRIÇÃO DA TAREFA

Considerando o tópico da adição de frações próprias, com denominadores variando de 2 a 9, desenhou-se a tarefa e aplicação com feedback em GeoGebra. O objetivo da tarefa para o aluno é efetuar a adição de duas frações próprias e simplificar o resultado. O Diagrama de processo da aplicação, ver Figura 18, contempla as situações em que os alunos tendem a falhar.

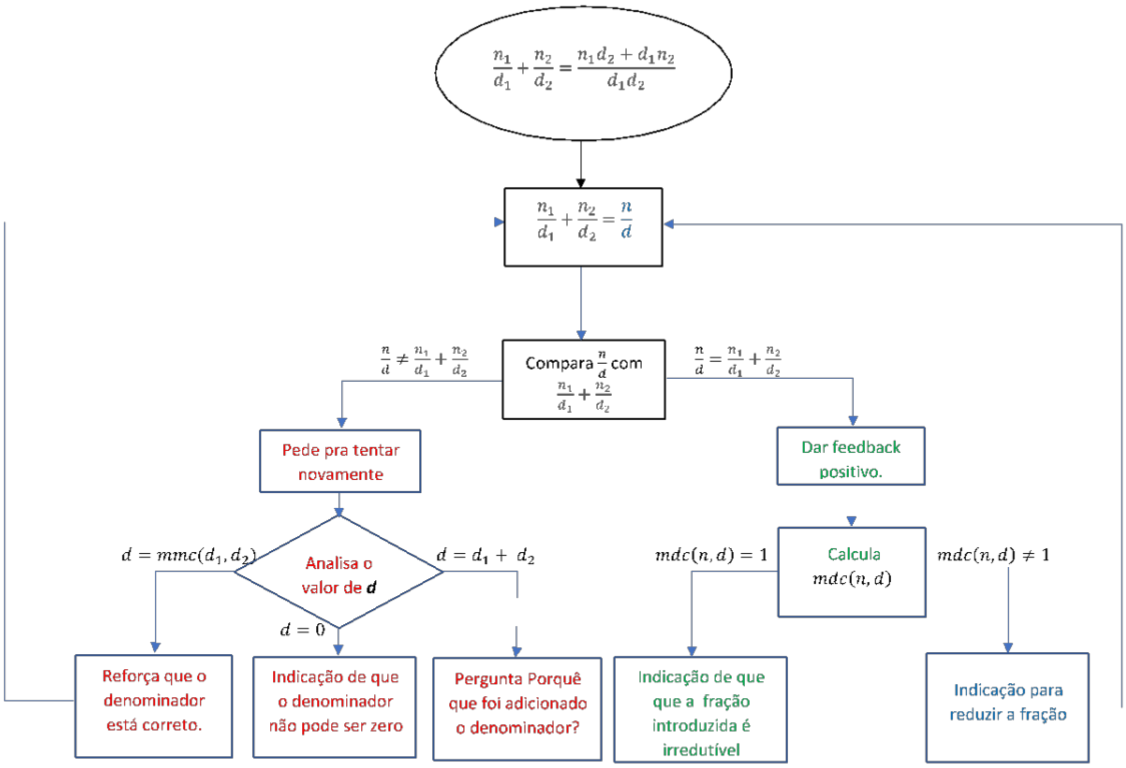


Figura 18 – Diagrama de Processo da aplicação "Adição de frações próprias, denominadores variando de 2 a 9"

Tipos de feedback

Em relação aos feedbacks a tarefa opta por dois tipos de feedback visual e escrito.

a) Visual

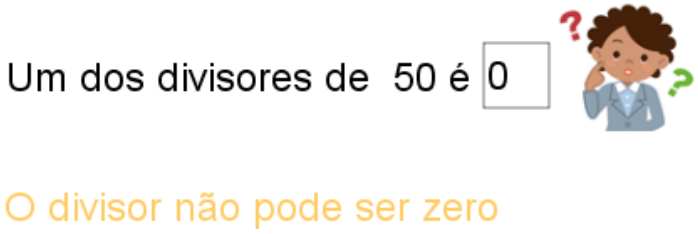
- Para resposta correta aparece esta imagem 

- Para resposta incorreta aparece outra imagem 

b) Textual – apresenta um texto dadas determinadas condições:

i. Resposta correta: "Bravo! Acertou, ..."

ii. Se a resposta for zero apresenta o texto: "o divisor não pode ser zero"

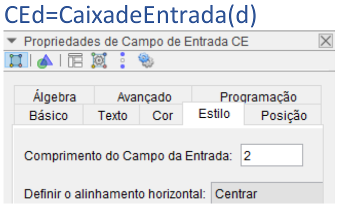


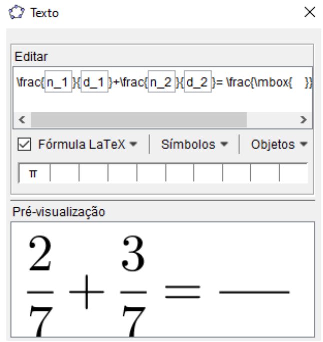
iii. Se a resposta for maior que o dividendo apresenta um texto dizendo que o divisor não pode ser zero, com um breve esclarecimento
iv. Se a resposta for errada, mas diferente de zero e não maior que o dividendo

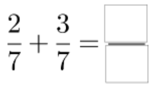
Guião de construção da aplicação

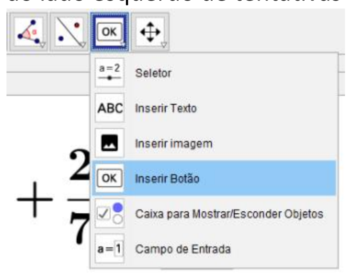
Para a construção da aplicação foram utilizados os passos que constam da Tabela 15.

Tabela 5 - Protocolo de construção da aplicação "Adição de frações próprias, denominadores variando de 2 a 9"

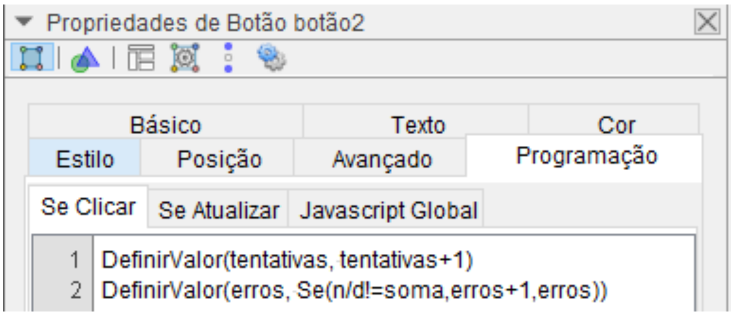
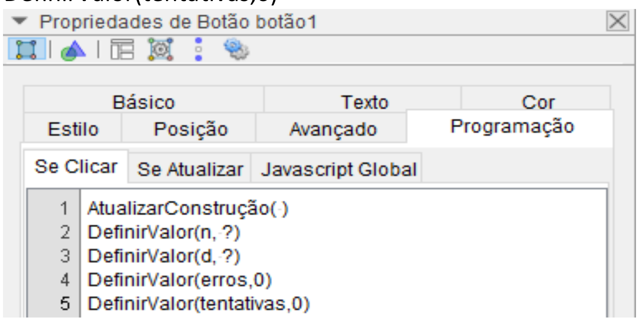
Passo	Objetivo	Comandos a escrever na barra de Entrada ou em separadores
1	Escolher os termos da 1ª fração (numerador n_1 e denominador d_1)	d_1= AleatórioInteiroEntre(2, 9) n_1=AleatórioInteiroEntre(1, d_1-1)
2	Escolher os termos da 2ª fração fração (numerador n_2 e denominador d_2)	d_2= AleatórioInteiroEntre(2, 9) n_2=AleatórioInteiroEntre(1, d_1-1)
3	Definir o alvo da resposta (d)	d=? n=?
4	Definir a caixas de respostas para numerador (n) e denominador (d)	CEn=CaixadeEntrada(n) CEd=CaixadeEntrada(d)  Propriedades objeto: não mostrar Rótulo; texto muito grande; comprimento da caixa de entrada 2; centrado; posição absoluta no ecrã, fixar objeto.
5	Definir a variável de controlo (soma)	soma=n_1 / d_1 + n_2 / d_2
6	Definir a expressão para a ser visualizada na tela	Inserir na caixa de texto, através de barras de ferramentas, a expressão: \frac{n_1 }{ d_1}+\frac{ n_2}{ d_2}= \frac{\mbox{ }}{\mbox{ }}

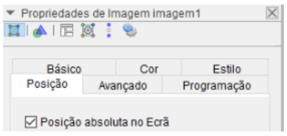
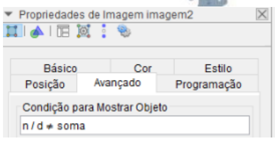
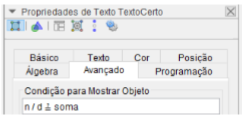
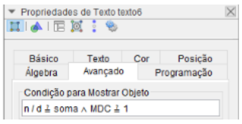
<table>
<tr><td></td><td></td><td>


Posicionar as caixas de entrada CEd e CEn, sobre o traço de fração depois do sinal de igual, nos respetivos lugares como numerador e denominador respectivamente

Propriedades objeto: texto muito grande; negrito; posição absoluta no ecrã, fixar objetos.
</td></tr>
<tr><td>7</td><td>Determinar o Máximo divisor comum entre n e d</td><td>mdc=MDC(n,d)</td></tr>
<tr><td>8</td><td>Determinar o Mínimo múltiplo comum entre os denominadores (d_1 e d_2)</td><td>mmc=MMC(d_1,d_2)</td></tr>
<tr><td></td><td>Criar dois seletores: “erros” para contar erros cometidos e “tentativas” para contar tentativas feitas</td><td>Digitar na entrada:
erros=0
tentativas=0

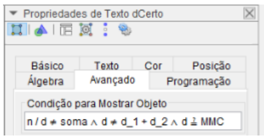
Arrastar os dois seletores para Folha Gráfica, posicionando-os num local visível (canto superior direito, por exemplo).</td></tr>
<tr><td></td><td>Criar dois botões: “Verificar”, para verificar o resultado inserido e “Novas parcelas”, para gerar números aleatórias nas parcelas.</td><td>Clicando na ferramenta “Botão”, inserir os respetivos e posicioná-los ao lado esquerdo de tentativas e erros.


</td></tr>
<tr><td></td><td>Programar o botão “Verificar”</td><td>Clicando com o botão direito do rato sobre o botão “Verificar”, selecionamos a opção propriedades, escolher “Programação”, selecionar “se clicar” e digitar:
DefinirValor(tentativas, tentativas+1);
DefinirValor(erros, Se(n/d!=soma,erros+1,erros));</td></tr>
</table>

		 *Esse comando faz com que ao clicarmos no botão, a variável 'tentativas' é acrescida de mais uma unidade. No caso da variável, 'erros' é feito um teste condicional. Se o valor digitado pelo usuário for diferente (indicada pelo símbolo !=) da resposta correta, esse valor é somado uma unidade ao valor da variável (indicado pela sentença erros+1). Caso contrário, ou seja, se o valor da resposta dada pelo usuário for igual a resposta correta (n/d=soma), o valor da variável "erros" permanece o mesmo.* No "avançado" escrever a condição para mostrar: erros < 3 *Este comando faz com que o botão* "Verificar" *fique oculto depois dos 3º erros.*
	Programação do botão "Novas parcelas"	Clicando com o botão direito do rato sobre o botão "Novas parcelas" selecionamos a opção propriedades, escolher "Programação", selecionar "se clicar" e digitar: *AtualizarConstrução( )* ⍰isto gera outro exercício; *DefinirValor(n, ?)* ⍰isto limpa a caixa de texto para numerador; *DefinirValor(d, ?)* ⍰isto limpa a caixa de texto para denominador; DefinirValor(erros,0) ⍰isto atribui valor 0 à variável "erros"; DefinirValor(tentativas,0) ⍰isto atribui valor 0 à variável "tentativa"  No "avançado" escrever a condição para mostrar: erros = 3 V n / d == soma *Este comando faz com que o botão* "Novas parcelas" *aparece quando a resposta é correta ou já foram cometidos 3 erros.*
11	Definir feedback visual por imagem associado à resposta correta	Inserir imagem 

		 Propriedades do objeto: nome: imagemCerto; posição absoluta no ecrã; fixar objeto, no avançado escrever a condição: n/d==soma
12	Definir feedback visual por imagem associado às respostas incorretas	Inserir imagem   Propriedades do objeto: nome: imagemErro; posição absoluta no ecrã; fixar objeto, condição para mostrar: $n/d \neq \ soman/d \neq \ soma$
13	Definir feedback textual quando a resposta é correta e com fração irredutível	TextoCerto = "Ótimo! acertaste no resultado." Irredutível = "Obtiveste uma fração irredutível." Propriedades do texto "*TextoCerto*": verde, muito grande, Condição para Mostrar objeto: n/d==soma Propriedades do texto "Irredutível": verde, muito grande, condição para aparecer: n / d== soma ∧ mdc == 1  
14	Definir feedback textual quando a resposta é correta e com fração redutível	Redutível= "Contudo, a fração ainda é redutível. Pode-se dividir ambos os termos da fração pelo MDC(" d_1 "," d_2 ")" Propriedades do texto "*TextoCerto*": verde, grande, fiar objeto, Condição para Mostrar objeto: n / d == soma ∧ mdc > 1
15	Definir feedback de texto quando o valor introduzido no denominador é zero	TextoErro = "Tenta outra vez!" dZero = "O denominador não pode ser zero." Propriedades do texto "TextoErro":Grande, Fixar Objeto, azul, Condição para mostrar objeto: n / d ≠ soma Propriedades do texto "dZero":Grande, Fixar Objeto, vermelha, Condição para mostrar objeto: d==0
16	Definir feedback	somad1ed2 = "Porquê que adicionaste os denominadores?"

<table>
<tr><td></td><td>textual quando na resposta errada o denominador corresponde à soma entre os denominadores das parcelas</td><td>Propriedades do texto:Grande, azul(com destaque), Condição para mostrar objeto: n / d ≠ soma ∧ d == d_1 + d_2</td></tr>
<tr><td>17</td><td>Definir feedback textual quando a resposta é errada, mas o denominador está certo</td><td>dCerto = "O denominador está correto. Verifica os cálculos para obter o numerador."



Propriedades do texto “dZero”:Grande, Fixar Objeto, vermelha, Condição para mostrar objeto: n / d ≠ soma ∧ d ≠ d_1 + d_2 ∧ d == MMC</td></tr>
</table>

Capítulo VII - Representação da reta que contém o gráfico de uma função linear de variável racional

Ilda Marisa de Sá Reis

Resumo:

Neste capítulo apresenta-se as diversas fases da construção de uma tarefa, implementada em GeoGebra, do domínio da álgebra e enquadrada no 7º ano de escolaridade do sistema de ensino português. Pretende-se representar graficamente uma função linear, de domínio racional, e relacionar a expressão algébrica com a sua representação gráfica. Com base nos possíveis erros cometidos pelo estudante são apresentados feedbacks automáticos e progressivos de modo que possa alcançar os objetivos delineados de forma autónoma.

ENQUADRAMENTO

Consta do currículo da disciplina de Matemática do sétimo ano português o estudo de funções lineares de domínio racional (o conceito de número irracional é apenas introduzido no oitavo ano de escolaridade), nomeadamente a sua representação gráfica num referencial cartesiano. De acordo com o documento “Aprendizagens Essenciais|Articulação com o perfil dos alunos”, de julho 2018, um objetivo essencial de aprendizagem, é “Representar e interpretar graficamente uma função linear e relacionar a sua representação gráfica com a algébrica e reciprocamente.” (MEC, 2018, p. 11). Para alcançar este objetivo propõe-se, no mesmo documento, a utilização da prática essencial de aprendizagem “Utilizar tecnologia digital, nomeadamente aplicações interativas, programas computacionais específicos e calculadora”. É neste contexto que se pretende recorrer à tecnologia digital, no caso concreto, à utilização do software GeoGebra.

DESCRIÇÃO DA TAREFA

A tarefa aqui proposta tem também por objetivo desenvolver outras capacidades matemáticas tais como “… confiança [do estudante] nas suas capacidades e conhecimentos matemáticos, e a capacidade de analisar o próprio trabalho e regular a sua aprendizagem." (MEC, 2018, p. 12) uma vez que foi desenvolvida para sistematizar a representação gráfica de uma função linear, mas também para orientar o estudante autonomamente, dado que, ao longo da mesma, são fornecidos feedbacks que o ajudam a analisar o seu trabalho, identificando progressos e dificuldades, como se ilustra no fluxo de análise da tarefa (Figura 19).

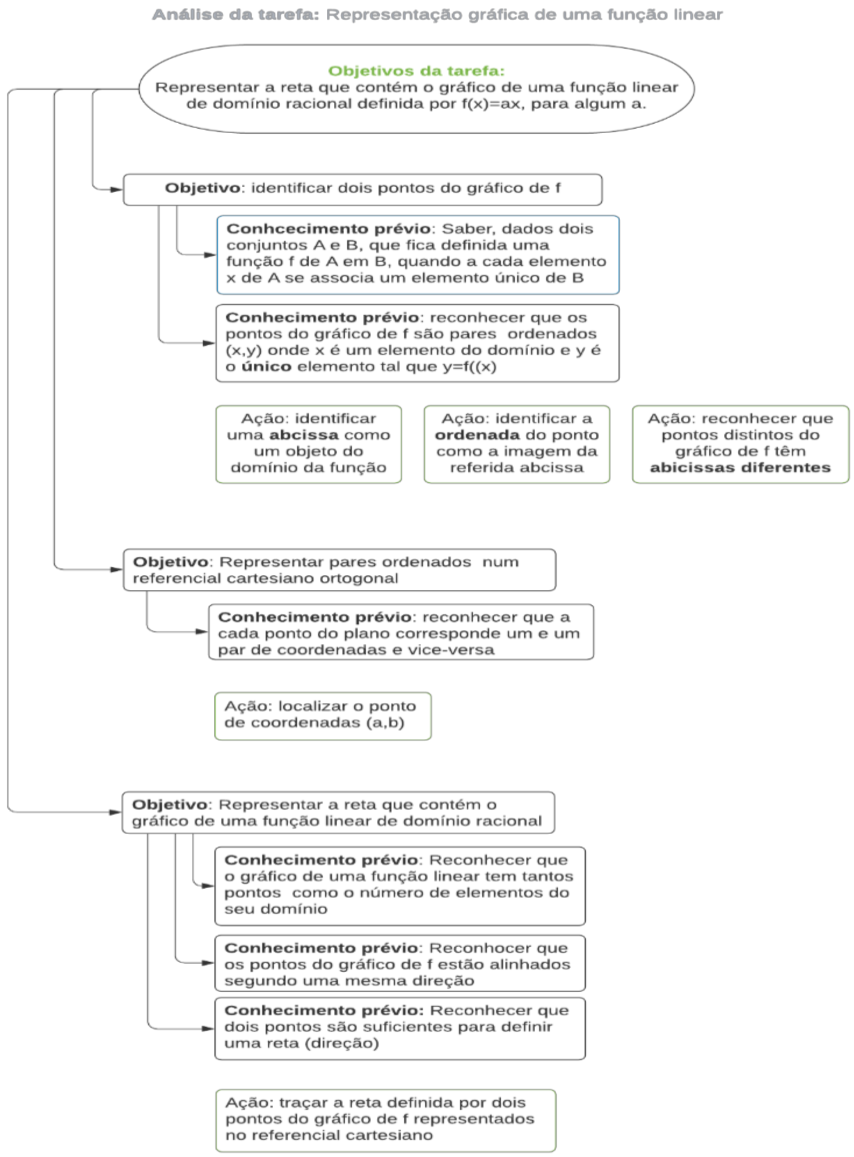


Figura 19 – Fluxo de análise da tarefa "Representação da reta que contém o gráfico de uma função linear de variável racional"

A tarefa está dividida em três partes: identificação de dois pontos do gráfico de uma função linear de variável racional, representação desses pontos num referencial cartesiano e representação da reta que contém o gráfico da função linear. Estas partes integram-se no diagrama de processo apresentado na Figura 20.

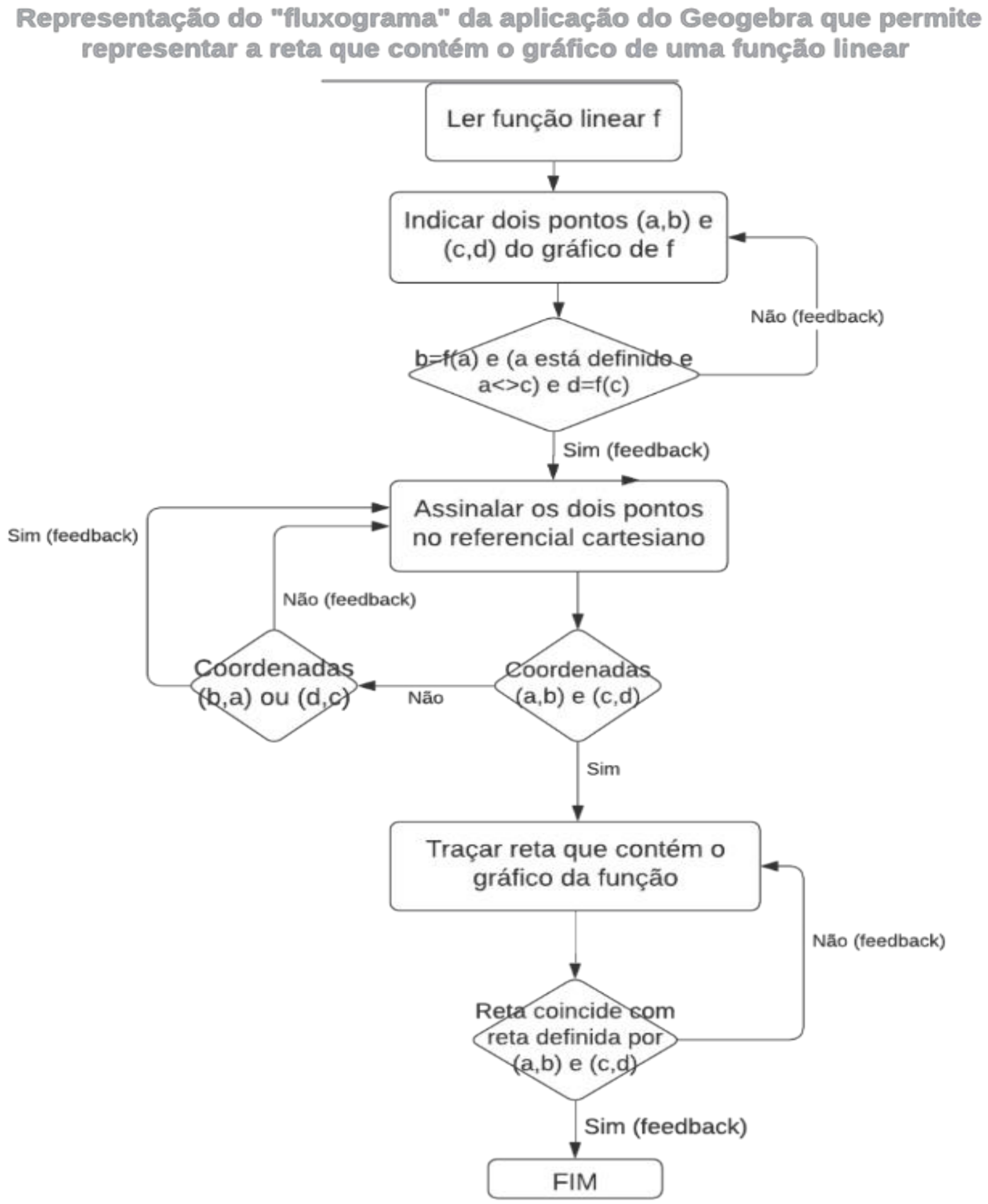


Figura 20 – Diagrama de processo da tarefa "Representação da reta que contém o gráfico de uma função linear de variável racional"

Parte I

Na primeira parte da tarefa é apresentada uma função linear $f(x)=ax$ onde o coeficiente da função, a, é gerado aleatoriamente de entre os elementos de uma lista de números inteiros. Na aplicação do GeoGebra optou-se por considerar $a \in \{-10,-9,-8,\ldots,8,9,10\}$ para facilitar a representação dos pontos do gráfico no referencial cartesiano. Solicita-se ao estudante que comece por identificar dois pontos do gráfico da função, designados internamente por $A_1(x_A, fx_A)$ e $B_1(x_B, fx_B)$, onde $fx_A = f(x_A)$ e $fx_B = f(x_B)$.

As primeiras duas caixas dizem respeito às coordenadas do ponto A_1 e as últimas duas às coordenadas do ponto B_1. Será exigido que se comece por indicar a abcissa do ponto seguida da respetiva ordenada. Importa salientar que a aplicação não permite alterar a abcissa do ponto sem antes indicar corretamente a sua ordenada. Sempre que o par inserido corresponde a um ponto do gráfico da função linear, as caixas ficam com fundo verde (feedback visual) e o contador dos "Acertos" (correspondente aos pontos do gráfico corretamente identificados) é incrementado de uma unidade (feedback numérico). Uma vez validado o ponto do gráfico, o estudante pode optar por o alterar bastando para isso selecionar a abcissa do ponto em causa. Note-se que se for selecionada a ordenada não acontecerá nada, pois a ordenada é única para cada abcissa. Caso os valores inseridos no par de caixas não correspondam às coordenadas de um ponto do gráfico da função linear, será fornecido feedback progressivo para que o estudante possa dar continuidade à tarefa. Da primeira vez a caixa de entrada com o valor incorreto da ordenada fica com fundo vermelho, da segunda vez aparecerá um texto a explicar o significado de um ponto pertencer ao gráfico de uma função, da terceira será indicada simbolicamente a imagem a inserir na caixa e, finalmente, na quarta vez aparecerá o valor da imagem a inserir com o respetivo cálculo. De cada vez que o aluno recebe um feedback o contador das "Falhas" (correspondente aos pontos do gráfico incorretamente identificados) é incrementado de uma unidade.

Validadas as coordenadas de um dos pontos do gráfico da função, o estudante pode passar ao preenchimento das outras duas caixas de entrada tendo em atenção que pontos distintos do gráfico têm abcissas distintas. Caso o estudante não tenha o conceito de função presente verá um texto (feedback escrito) alertando-o para esse facto.

Neste processo de identificação de dois dos pontos do gráfico da função é registado internamente o número de falhas cometidas na identificação de cada ponto, bem como, as imagens das abcissas indicadas. Deste modo o "supervisor" da tarefa, acedendo a esses dados poderá analisar o progresso no cumprimento da tarefa e melhor direcionar o estudante para tarefas futuras.

Parte II

Validadas as quatro caixas de entrada com as coordenadas de dois pontos do gráfico da função linear, a próxima etapa é a sua representação num referencial cartesiano ortogonal. O estudante deve recorrer à ferramenta "Novo Ponto" para assinalar na janela gráfica 2D os dois pontos que indicou pertencerem ao gráfico da função linear. Nessa janela está representado um referencial cartesiano e um reticulado para auxiliar na marcação dos pontos. A localização dos pontos marcados no referencial cartesiano é validada no botão "Verifique a localização dos pontos". Esta aplicação só testa a posição dos dois primeiros pontos assinalados no referencial. Se o botão for acionado e os dois pontos não estiverem assinalados é emitido um feedback sonoro (mensagem de áudio) a alertar para esta situação. Se existirem outros pontos, além do feedback escrito relativos aos dois primeiros pontos com etiquetas "A" e "B", a aplicação emite um feedback sonoro (mensagem de áudio) a alertar que os restantes pontos não foram tidos em consideração. No feedback escrito é dada a indicação para mover o(s) ponto(s) para a localização(ões) correta(s), no entanto, em certas situações pode não ser possível fazê-lo (exemplo em que o ponto se encontra sobre os eixos coordenados). Nesse caso, o estudante deve usar a ferramenta "Apagar" e voltar a marcar o ponto. Esta situação ocorre quando é criado um ponto dependente de um outro objeto.

De cada vez que o botão é acionado, o estudante também visualiza um feedback numérico com o número de pontos corretamente assinalados e o número de pontos mal localizados (só os dois primeiros pontos são considerados). Um ponto bem assinalado passa à cor verde (feedback visual) e as suas coordenadas são exibidas, deixando de ser possível alterar a sua posição no referencial cartesiano, bem como as caixas de entrada que contêm as coordenadas. Porém, se o ponto for incorretamente assinalado passa à cor vermelha e o estudante pode alterar a sua posição ou até alterar os valores das caixas de entrada que lhe deram origem para poder representar um outro ponto.

Sempre que um ponto é mal localizado é incrementada uma unidade a um contador que ao atingir o valor dois torna acessível um botão de ajuda para visualizar um vídeo explicativo da marcação de pontos num referencial cartesiano. O código, em JavaScript, para abrir uma página com o link de um vídeo do Youtube foi adaptado do site https://www.geogebra.org/m/bqDMD6sC (acedido em abril de 2022).

Na verificação da localização dos pontos do gráfico da função linear, o algoritmo, apenas, testa se as coordenadas de algum dos pontos foram trocadas, dado este ser o erro mais frequente neste tipo de tarefa.

Um constrangimento nesta validação surge quando o estudante renomeia pontos, pois pode interferir com os pontos A_1, A_2, B_1 e B_2, usados internamente para efetuar as comparações e fornecer feedback. Uma forma de contornar este problema é não tornar acessível o menu de contexto do ponto (bastando para tal desabilitar a opção “Permitir o uso do botão direito do rato, do zoom e da edição pelo teclado” nas configurações avançadas da atividade) pois o estudante tem sempre a possibilidade de usar a ferramenta “Apagar”.

Parte III

Validada a localização no referencial cartesiano dos pontos do gráfico da função linear indicados na parte I, o passo seguinte consiste na representação da reta que contém o gráfico da função linear dada. Nesta etapa podem ser aproveitados os pontos anteriormente assinalados ou escolhidos novos pontos para definir essa reta que será posteriormente comparada com a definida pelos pontos A_1 e B_1 (mencionados atrás). A reta é traçada na folha gráfica 2D recorrendo à ferramenta “Reta (Dois Pontos)”. Caso seja coincidente com a reta A_1 B_1 passa para a cor verde e é alterada a sua espessura. Caso contrário, fica a cor vermelha e devem ser deslocados os pontos que definem a reta para novas posições para posteriormente voltar a ser testada.

Se o botão de verificação da reta for acionado antes do traçado da reta é emitido um feedback sonoro (mensagem de áudio) como alerta. Caso sejam traçadas mais do que uma reta, apenas a primeira será testada (feedback escrito) e é emitido um feedback sonoro (mensagem de áudio) alertando para esse facto.

De cada vez que o botão é acionado estando pelo menos uma reta traçada, o estudante visualiza um feedback numérico com o número de acertos e falhas cometidos até ao momento.

Depois de bem conseguida a localização da reta, aparece um botão “Novo Exercício” permitindo que o estudante realize um novo exercício com outro valor do coeficiente da função linear. Sempre que o botão é pressionado são eliminados os pontos construídos e inicializadas todas as variáveis para o estado inicial.

GUIÃO DE CONSTRUÇÃO DA APLICAÇÃO

Ao iniciar a aplicação GeoGebra devem estar visíveis a Folha Gráfica 2D e a Folha Gráfica 2D 2. Nesta última deve estar visível um referencial cartesiano monométrico (com rótulos nos eixos e as linhas principais da grelha) para posteriormente facilitar a marcação dos pontos solicitados.

A seguir é apresentado o protocolo completo da construção da aplicação em GeoGebra.

Definir função linear:

Tabela 6 - Protocolo de construção da aplicação "Representação da reta que contém o gráfico de uma função linear de variável racional", 1ª parte

1º	a = RandomBetween(-10,10)
2º	f(x) = ax

Escrever a primeira parte do enunciado do exercício:

Tabela 7 - Protocolo de construção da aplicação "Representação da reta que contém o gráfico de uma função linear de variável racional", 2ª parte

3º	texto1 = "\text{Considere a função } f:\mathbb{Q}\longrightarrow\mathbb{Q} \text{definida por }f(x)= f.\\\\ \text{Indique dois pontos do gráfico de }f." f é dinâmico. Formatação: posição absoluta no ecrã; não permitir seleção.
4º	texto2 = " (,) \text{e} (,)" Formatação: posição absoluta no ecrã; não permitir seleção.

Criar caixas de entrada para escrever a resposta (indicar dois pontos do gráfico da função linear) e programá-las para validarem os inputs.

Tabela 8 - Protocolo de construção da aplicação "Representação da reta que contém o gráfico de uma função linear de variável racional", 3ª parte

5º	x_{A} = ?
6º	y_{A} = ? Programação (On update) If(y_A==fx_A, {RunClickScript(Ponto1Validar)}, {SetValue(Falhas, Falhas+1), SetValue(Falhas1,1+Falhas1), If(Falhas1==1, {SetBackgroundColor(InputBox2,"Red")}, {If(Falhas1==2, {SetVisibleInView(texto6,1,true)},{If(Falhas1==3, {SetVisibleInView(texto7,1,true)},{SetVisibleInView(texto8,1,true)})})})}) Comentários: - Valida o ponto se a imagem for bem calculada ou dá feedback de acordo com o número de falhas cometidas; - O botão Ponto1Validar está definido mais abaixo.
7º	x_{B} = ?
8º	y_{B} = ? Programação (On update) If(y_{B}==fx_{B}, {RunClickScript(Ponto2Validar)}, {SetValue(Falhas, Falhas+1), SetValue(Falhas1,1+Falhas1), If(Falhas1==1, {SetBackgroundColor(InputBox4,"Red")}, {If(Falhas1==2, {SetVisibleInView(texto6,1,true)},{If(Falhas1==3, {SetVisibleInView(texto9,1,true)},{SetVisibleInView(texto10,1,true)})})})}) Comentário: - Valida o ponto se a imagem for bem calculada ou dá feedback de acordo com o número de

	falhas cometidas; - O botão Ponto2Validar está definido mais abaixo.
9º	fx_{A} = f(x_{A})
10º	fx_{B} = f(x_{B})
11º	texto3 = "\red{\text{O gráfico de }f \text{ não pode ter dois pontos}\\\text{com a mesma abcissa!}}" Formatação: posição absoluta no ecrã; não permitir seleção
12º	Acertos = 0 Variável usada para contabilizar o número de pontos corretamente identificados do gráfico da função linear
13º	Falhas = 0 Variável usada global para contabilizar o número de falhas cometidas na identificação de pontos do gráfico da função linear
14º	Falhas1 = 0 Variável usada para contar o número de falhas cometidas na identificação de cada ponto
15º	PtsGfErros = {} Esta lista guarda o número de erros cometidos na identificação de cada ponto do gráfico da função linear.
16º	ImgPtsGf = {} Esta lista guarda as imagens dos pontos do gráfico da função que o estudante encontra corretamente.
17º	comp = 0 Variável auxiliar que servirá para validar os pontos do gráfico da função linear (usada nos botões Ponto1Validar e Ponto2Validar criados mais à frente) e para validar a localização dos pontos no referencial cartesiano..
18º	texto4 = "Acertos: "+ Acertos + "" Formatação: posição absoluta no ecrã; não permitir seleção, cor verde.
19º	texto5 = "Falhas: "+ Falhas + "" Formatação: posição absoluta no ecrã; não permitir seleção, cor vermelha.
20º	texto6 = "\red{(x,y)\in G_f \text{ quando }y=f(x).}" Formatação: posição absoluta no ecrã; não permitir seleção.
21º	texto7 = "\red{\text{Insira o valor de }f(x_A).}" Formatação: posição absoluta no ecrã; não permitir seleção. x_A é texto dinâmico.
22º	texto8 = "f(x_A) = a × x_A = fx_A Formatação: posição absoluta no ecrã; não permitir seleção, cor vermelha. x_A, a e fx_A é texto dinâmico.
23º	texto9 = "\red{\text{Insira o valor de }f(x_B).}" Formatação: posição absoluta no ecrã; não permitir seleção. x_B é texto dinâmico.
24º	Texto10 = "f(x_B) = a × x_B = fx_B Formatação: posição absoluta no ecrã; não permitir seleção, cor vermelha.

	x_B, a e fx_B é texto dinâmico.
25º	flagPt1 = false Variável auxiliar que servirá para mostrar a segunda parte do exercício.
26º	flagPt2 = false Variável auxiliar que servirá para mostrar a segunda parte do exercício.
27º	Button("LimpaPonto1Grafico") Formatação: não mostrar objeto e alterar nome para LimpaPonto1Grafico Programação (On Click): SetValue(x_A,?) SetValue(y_A, ?) SetValue(flagPt1, false) SetVisibleInView(texto6,1,false) SetVisibleInView(texto7,1,false) SetVisibleInView(texto8,1,false) SetVisibleInView(texto9,1,false) SetVisibleInView(texto10,1,false) SetVisibleInView(texto14, 1, false) SetVisibleInView(texto15, 1, false) SetVisibleInView(texto16, 1, false) SetVisibleInView(texto17, 1, false) SetVisibleInView(texto18, 1, false) SetBackgroundColor(InputBox1,"#FFEACD") SetBackgroundColor(InputBox2,"#FFEACD") SetFixed(InputBox2, true, true) SelectObjects(InputBox1)
28º	Button("LimpaPonto2Grafico") Formatação: não mostrar objeto e alterar nome para LimpaPonto2Grafico Programação (On Click) SetValue(x_B,?) SetValue(y_B, ?) SetValue(flagPt2, false) SetVisibleInView(texto6,1,false) SetVisibleInView(texto7,1,false) SetVisibleInView(texto8,1,false) SetVisibleInView(texto9,1,false) SetVisibleInView(texto10,1,false) SetVisibleInView(texto14, 1, false) SetVisibleInView(texto15, 1, false) SetVisibleInView(texto16, 1, false) SetVisibleInView(texto17, 1, false) SetVisibleInView(texto18, 1, false) SetBackgroundColor(InputBox3,"#FFEACD") SetBackgroundColor(InputBox4,"#FFEACD") SetFixed(InputBox4, true, true) SelectObjects(InputBox3)
29º	Button("Ponto1Validar") Formatação: não mostrar objeto e alterar nome para Ponto1Validar Programação (On Click)

	SetValue(comp, Length(ImgPtsGf)) If(comp==0, {SetValue(Acertos,Acertos+1), SetValue(ImgPtsGf, comp+1, y_A), SetValue(PtsGfErros, comp+1, Falhas1)},{If((Element(ImgPtsGf, comp)!=y_A||Falhas1!=0), {SetValue(Acertos,Acertos+1), SetValue(ImgPtsGf, comp+1, y_A), SetValue(PtsGfErros, comp+1, Falhas1)})}) SetValue(Falhas1,0) SetValue(flagPt1, true) SetVisibleInView(texto6,1,false) SetVisibleInView(texto7,1,false) SetVisibleInView(texto8,1,false) SetBackgroundColor(InputBox1,"Light Green") SetBackgroundColor(InputBox2,"Light Green") SetFixed(InputBox2, true, false)
30º	Button("Ponto2Validar") Formatação: não mostrar objeto e alterar nome para Ponto2Validar Programação (On Click) SetValue(comp, Length(ImgPtsGf)) If(comp==0, {SetValue(Acertos,Acertos+1), SetValue(ImgPtsGf, comp+1, y_B), SetValue(PtsGfErros, comp+1, Falhas1)},{If((Element(ImgPtsGf, comp)!=y_B||Falhas1!=0), {SetValue(Acertos,Acertos+1), SetValue(ImgPtsGf, comp+1, y_B), SetValue(PtsGfErros, comp+1, Falhas1)})}) SetValue(Falhas1,0) SetValue(flagPt2, true) SetVisibleInView(texto6,1,false) SetVisibleInView(texto9,1,false) SetVisibleInView(texto10,1,false) SetBackgroundColor(InputBox3,"Light Green") SetBackgroundColor(InputBox4,"Light Green") SetFixed(InputBox4, true, false)

31º	InputBox1=InputBox(x_A)
	Formatação: comprimento da caixa de entrada: 4; alinhamento: central; posição absoluta no ecrã; álgebra: simbólico;
	Programação (On update)
	If(!IsDefined(x_B), If(IsDefined(x_A), If(!IsDefined(y_A), SelectObjects(InputBox2), If(y_A==fx_A, RunClickScript(LimpaPonto1Grafico), SelectObjects(InputBox2)))))
	If(IsDefined(x_B), If(IsDefined(y_B), If(y_B==fx_B, If(IsDefined(x_A), If(x_A==x_B, {SetVisibleInView(texto3,1,true), SelectObjects(InputBox1)}, {SetVisibleInView(texto3,1,false), If(!IsDefined(y_A), SelectObjects(InputBox2), If(y_A==fx_A, RunClickScript(LimpaPonto1Grafico), SelectObjects(InputBox2)))})))))
	If(IsDefined(x_B), If(IsDefined(y_B), If(y_B!=fx_B, SelectObjects(InputBox4))))

32º	InputBox2=InputBox(y_A)
	Formatação: comprimento da caixa de entrada: 4; alinhamento: central; posição absoluta no ecrã; álgebra: simbólico;
	Programação (On update)
	If(!IsDefined(x_B), If(!IsDefined(x_A), SelectObjects(InputBox1)))
	If(IsDefined(x_B), If(IsDefined(y_B), If(y_B==fx_B, If(!IsDefined(x_A), SelectObjects(InputBox1)))))
	If(IsDefined(x_B), If(IsDefined(y_B), If(y_B==fx_B, If(IsDefined(x_A), If(x_B==x_A, SelectObjects(InputBox1))))))

	If(IsDefined(x_B), If(IsDefined(y_B), If(y_B!=fx_B, SelectObjects(InputBox4)), SelectObjects(InputBox4)))

33º	InputBox3=InputBox(x_B)
	Formatação: comprimento da caixa de entrada: 4; alinhamento: central; posição absoluta no ecrã; álgebra: simbólico;
	Programação (On update)
	If(!IsDefined(x_A), If(IsDefined(x_B), If(!IsDefined(y_B), SelectObjects(InputBox4), If(y_B==fx_B, RunClickScript(LimpaPonto2Grafico), SelectObjects(InputBox4)))))
	If(IsDefined(x_A), If(IsDefined(y_A), If(y_A==fx_A, If(IsDefined(x_B), If(x_A==x_B, {SetVisibleInView(texto3,1,true), SelectObjects(InputBox3)}, {SetVisibleInView(texto3,1,false), If(!IsDefined(y_B), SelectObjects(InputBox4), If(y_B==fx_B, RunClickScript(LimpaPonto2Grafico), SelectObjects(InputBox4)))})))))
	If(IsDefined(x_A), If(IsDefined(y_A), If(y_A!=fx_A, SelectObjects(InputBox2)), SelectObjects(InputBox2)))

34º	InputBox4=InputBox(y_B)
	Formatação: comprimento da caixa de entrada: 4; alinhamento: central; posição absoluta no ecrã; álgebra: simbólico;
	Programação (On update)
	If(!IsDefined(x_A), If(!IsDefined(x_B), SelectObjects(InputBox3)))
	If(IsDefined(x_A), If(IsDefined(y_A), If(y_A==fx_A, If(!IsDefined(x_B), SelectObjects(InputBox3)))))
	If(IsDefined(x_A), If(IsDefined(y_A), If(y_A==fx_A, If(IsDefined(x_B), If(x_B==x_A, SelectObjects(InputBox3))))))
	If(IsDefined(x_A), If(IsDefined(y_A), If(y_A!=fx_A, SelectObjects(InputBox2)), SelectObjects(InputBox2)))

Escrever a segunda parte do enunciado do exercício e validar a localização dos pontos do gráfico anteriormente indicados nas caixas de entrada:

Tabela 9 - Protocolo de construção da aplicação "Representação da reta que contém o gráfico de uma função linear de variável racional", 4ª parte

35º	texto11 = "\text{Selecione a ferramenta para assinalar os pontos indicados\\ no referencial cartesiano ao lado.}" **Formatação**: posição absoluta no ecrã; não permitir seleção. **Condição de visualização**: flagPt1 ∧ flagPt2
36º	A_3 = (105,347)
37º	B_3 = (127,346)
38º	PicPonto =ToolImage(1, A_3, B_3) **Condição de visualização**: flagPt1 ∧ flagPt2 A_3 e B_3 são os pontos onde a imagem do ícone fica ancorada.
39º	A_1 = (x_A, fx_A) **Formatação**: objeto fixo.
40º	A_2 = (fx_A, x_A) **Formatação**: objeto fixo. Este ponto servirá para testar se o aluno troca as coordenadas na representação de um ponto

41º	B_1 = (x_B, fx_B) **Formatação**: objeto fixo.
42º	B_2 = (fx_B, x_B) **Formatação**: objeto fixo. Este ponto servirá para testar se o aluno troca as coordenadas na representação de um ponto

43º	Button("ListaPontosReferencial")
	Formatação: não mostrar objeto e alterar nome para ListaPontosReferencial
	Programação (On Click)(Javascript)
	ggbApplet.evalCommand('PtsRef={}')
	for (ind = ggbApplet.getObjectNumber()-1; ind>=0; ind=ind-1)
	{nam = ggbApplet.getObjectName(ind)
	typ = ggbApplet.getObjectType(nam)
	if (typ == "point") {
	    if ((nam != "A_1") && (nam != "A_2") && (nam != "A_3") && (nam != "B_1") && (nam != "B_2") && (nam != "B_3")){
	        ggbApplet.evalCommand("SetValue[PtsRef, Length[PtsRef] + 1, " + nam + "]")}}
	}

44º	PtsAcertos = 0 Variável usada para contabilizar quantos pontos indicados da função linear estão corretamente assinalados no referencial cartesiano.
45º	PtsFalhas = 0 Variável usada para contabilizar quantos pontos indicados da função linear estão corretamente assinalados no referencial cartesiano.
46º	PtsFalhasTotal = 0 Variável auxiliar usada para contabilizar quantos falhas foram cometidas até que os dois pontos do gráfico da função linear estejam bem localizados no referencial.
47º	texto12 = "Acertos: "+ PtsAcertos + "" **Formatação**: posição absoluta no ecrã; não permitir seleção. **Condição de visualização**: flagPt1 ∧ flagPt2
48º	texto13 = "Falhas: "+ PtsFalhas + "" **Formatação**: posição absoluta no ecrã; não permitir seleção. **Condição de visualização**: flagPt1 ∧ flagPt2
49º	texto14 = "\red{\text{Trocou as coordenadas de um dos pontos.}\\\text{Selecione a ferramenta para o mover}\\\text{para a posição correta.}}" **Formatação**: posição absoluta no ecrã; não permitir seleção.
50º	texto15 = "\red{\text{Reveja a localização de um dos pontos.}\\\text{Selecione a ferramenta para o mover}\\\text{para a posição correta.}}" **Formatação**: posição absoluta no ecrã; não permitir seleção.
51º	texto16 = "\red{\text{Trocou as coordenadas nos dois pontos.}\\ \text{Selecione a ferramenta para os mover}\\ \text{para as posições corretas.}}" **Formatação**: posição absoluta no ecrã; não permitir seleção.
52º	texto17 = "\red{\text{Reveja localização de pontos em referencial\\ Selecione a ferramenta para os mover\\

<table>
<tr><td></td><td>para as posições corretas.}}"
Formatação: posição absoluta no ecrã; não permitir seleção.</td></tr>
<tr><td>53º</td><td>texto18 = "\red{\text{Um ponto com coordenadas trocadas\\
e outro mal assinalado.\\
Selecione a ferramenta para os mover\\
para as posições corretas.}}"
Formatação: posição absoluta no ecrã; não permitir seleção.</td></tr>
<tr><td>54º</td><td>flagPts = false
Variável auxiliar que servirá para mostrar a terceira parte do exercício.</td></tr>
<tr><td>55º</td><td>Button(“TestePts”)
Formatação: não mostrar objeto e alterar nome para TestePts
Programação (On Click)
# Os pontos A e B coincidem com os pontos identificados do gráfico da função linear
If((A_1 == A && B_1 == B) || (A_1 == B && B_1== A), {SetFixed(A, true, True), SetFixed(B, true, True), ShowLabel(A, true), SetLabelMode(A, 2), ShowLabel(B, true), SetLabelMode(B, 2), SetPointSize(A, 5), SetPointSize(B, 5), SetColour(A, "Dark Green"), SetColour(B, "Dark Green"), SetFixed(InputBox1, true, false), SetFixed(InputBox3, true, false), SetValue(PtsAcertos, 2), SetValue(PtsFalhas, 0), SetValue(PtsFalhasTotal, 0), SetValue(flagPts, true)})
# --
# O ponto A coincide com algum dos pontos identificados do gráfico da função linear
If((A_1 == A && B_1 != B) || (A_1 != B && B_1 == A), {SetFixed(A, true, True), ShowLabel(A, true), SetLabelMode(A, 2), SetPointSize(A, 5), SetColour(A, "Dark Green"), SetColour(B, "Crimson"), SetValue(PtsAcertos, 1), SetValue(PtsFalhas, 1), SetValue(PtsFalhasTotal, 1+PtsFalhasTotal)})
# O ponto A é ponto do gráfico mas B tem coordenadas trocadas
# O ponto A é o primeiro ponto indicado do gráfico
If(B_1 != B_2, If((A_1 == A && B_2 == B), {SetFixed(InputBox1, true, false), SetVisibleInView(texto14, 1, true)}))
# O ponto A é o segundo ponto indicado do gráfico
If(A_1 != A_2, If((A_2 == B && B_1 == A), {SetFixed(InputBox3, true, false), SetVisibleInView(texto14, 1, true)}))
# O ponto A é ponto do gráfico mas as coordenadas de B estão erradas e não estão trocadas
# O ponto A é o primeiro ponto indicado do gráfico
If((A_1 == A && B_1 != B && B_2 != B), {SetFixed(InputBox1, true, false), SetVisibleInView(texto15, 1, true)})
# O ponto A é o segundo ponto indicado do gráfico
If((A_1 != B && A_2 != B && B_1 == A), {SetFixed(InputBox3, true, false), SetVisibleInView(texto15, 1, true)})
#---
# O ponto B coincide com algum dos pontos identificados do gráfico da função linear
If((A_1 != A && B_1 == B) || (A_1 == B && B_1 != A), {SetFixed(B, true, True), ShowLabel(B, true), SetLabelMode(B, 2), SetPointSize(B, 5), SetColour(B, "Dark Green"), SetColour(A, "Crimson"), SetValue(PtsAcertos, 1), SetValue(PtsFalhas, 1), SetValue(PtsFalhasTotal, 1+PtsFalhasTotal)})
# O ponto B é ponto do gráfico mas A tem coordenadas trocadas
# O ponto B é o primeiro ponto indicado do gráfico
If(B_1 != B_2, If((A_1 == B && B_2 == A), {SetFixed(InputBox1, true, false), SetVisibleInView(texto14, 1, true)}))</td></tr>
</table>

	# O ponto B é o segundo ponto indicado do gráfico
	If(A_1 != A_2, If((A_2 == A && B_1 == B), {SetFixed(InputBox3, true, false), SetVisibleInView(texto14, 1, true)}))
	# O ponto B é ponto do gráfico mas as coordenadas de A estão erradas e não estão trocadas
	# O ponto B é o primeiro ponto indicado do gráfico
	If((A_1 == B && B_1 != A && B_2 != A), {SetFixed(InputBox1, true, false), SetVisibleInView(texto15, 1, true)})
	# O ponto B é o segundo ponto indicado do gráfico
	If((A_1 != A && A_2 != A && B_1 == B), {SetFixed(InputBox3, true, false), SetVisibleInView(texto15, 1, true)})
	# ---
	# Ambos os pontos estão incorretos
	If((A_1 != A && B_1 != B) && (A_1 != B && B_1!= A), {SetColour(A, "Crimson"), SetColour(B, "Crimson"), SetValue(PtsAcertos, 0), SetValue(PtsFalhas, 2), SetValue(PtsFalhasTotal, 2+PtsFalhasTotal)})
	# Ambos os pontos estão incorretos com coordenadas trocadas
	If(A_1 != A_2 && B_1 != B_2, If((A_2 == A && B_2 == B) || (A_2 == B && B_2== A),SetVisibleInView(texto16, 1, true)))
	# Ambos os pontos estão incorretos, mas um deles tem coordenadas trocadas
	If(B_1 != B_2, If((A_1 != A && A_2 != A && B_2 == B) || (A_1 != B && A_2 != B && B_2 == A), SetVisibleInView(texto18,1,true))
	If (A_1 != A_2), If((A_2 == A && B_1 != B && B_2 != B) || (A_2 == A && B_1 != B && B_2 != B), SetVisibleInView(texto18,1,true))
	# Ambos os pontos estão incorretos, mas as coordenadas não estão trocadas
	If((A_1 != A && A_2 != A && B_1 != B && B_2 != B) && (A_1 != B && A_2 != B && B_1 != A && B_2 != A), SetVisibleInView(texto17, 1, true))

56º	Button("Ajuda marcar ponto")
	Formatação: posição absoluta no ecrã, tamanho da letra "muito pequeno" e alterar nome para "Ajuda".
	Condição de visualização: PtsFalhasTotal >=2
	Programação (On Click)(javascript)
	function openInNewTab(url) {
	  var win = window.open(url, '_blank');
	try { win.focus(); } catch (e) {}
	}
	openInNewTab("https://www.youtube.com/watch?v=Il9J1Nyzj4U&t=27s ")

57º	Button("Verifique a localização dos pontos") **Formatação**: posição absoluta no ecrã e alterar nome para "VerifiquePontosReferencial" **Condição de visualização**: flagPt1 ∧ flagPt2 **Programação** (On Click) SetVisibleInView(texto14, 1, false) SetVisibleInView(texto15, 1, false) SetVisibleInView(texto16, 1, false) SetVisibleInView(texto17, 1, false) SetVisibleInView(texto18, 1, false) SetValue(PtsAcertos,0) SetValue(PtsFalhas,0) RunClickScript(ListaPontosReferencial) SetValue(comp, Length(PtsRef))

	If(comp <2, PlaySound("#my27aahm"))
	If(comp == 2, RunClickScript(TestePts))
	If(comp >2, {RunClickScript(TestePts), PlaySound("#ve32w7nd")})

Escrever a terceira parte do enunciado do exercício e validar a localização dos pontos do gráfico anteriormente indicados nas caixas de entrada:

Tabela 10 - Protocolo de construção da aplicação "Representação da reta que contém o gráfico de uma função linear de variável racional", 5ª parte

58º	texto19 = "\text{Selecione a ferramenta para traçar a reta que}\\ \text{contém o gráfico (cartesiano) da função } f." **Formatação**: posição absoluta no ecrã; não permitir seleção. **Condição de visualização**: flagPts
59º	RetaAcertos = 0 Variável usada para contabilizar o número de acertos na localização da reta que contém o gráfico da função linear.
60º	RetaFalhas = 0 Variável usada para contabilizar o número de vezes que é indevidamente identificada a reta que contém o gráfico da função linear.
61º	texto20 = "Acertos: "+RetaAcertos+ "" **Formatação**: posição absoluta no ecrã; não permitir seleção. **Condição de visualização**: flagPts
62º	texto21 = "Falhas: "+RetaFalhas+""" **Formatação**: posição absoluta no ecrã; não permitir seleção. **Condição de visualização**: flagPts
63º	texto22 = "\red{\text{Reta incorreta.}\\ \text{Selecione a ferramente para deslocar os pontos}\\ \text{que definem a reta para as posições corretas.}}" **Formatação**: posição absoluta no ecrã; não permitir seleção.
64º	g = Line(A_1, B_1)
65º	flagReta = false Variável auxiliar que permitirá repetir o exercício para outro parâmetro da função linear.

66º	Button("ListaRetas")
	Formatação: não mostrar objeto e alterar nome para ListaRetas
	Programação (On Click)(Javascript)
	ggbApplet.evalCommand('RetasRef={}')
	for (ind = 0; ind < ggbApplet.getObjectNumber(); ind=ind+1)
	{
	Var num = 0:
	nome = ggbApplet.getObjectName(ind)
	tipo = ggbApplet.getObjectType(nome)
	if (tipo == "line")
	{
	  If (nome != "g")
	  {
	    If (num == 0)

<table>
<tr><td></td><td><pre>
      {
         ggbApplet.renameObject(nome, “h”);
         ggbApplet.evalCommand("SetValue[RetasRef, Length[RetasRef] + 1, h]");
      }
      else
      {
     ggbApplet.evalCommand("SetValue[RetasRef, Length[RetasRef] + 1, " + nome+ "]")
      }
      num = num+1;
   }
}
}
</pre></td></tr>
<tr><td>64º</td><td>Button(“TestaReta”)

Formatação: não mostrar objeto e alterar nome para TestaReta

Programação (On Click)
If(g == h ,{SetLineThickness(h, 5), SetColour(h,"Dark Green"), ShowLabel(h, false), SetValue(RetaAcertos, 1+RetaAcertos), SetValue(flagReta, true)}, {SetColour(h,"Crimson"), SetValue(RetaFalhas, 1+RetaFalhas), SetValue(flagReta, false), SetVisibleInView(texto22, 1, true)})</td></tr>
<tr><td>67º</td><td>Button(“Verifique a localização da reta”)

Formatação: posição absoluta no ecrã e alterar nome para “VerifiqueReta”

Condição de visualização: flagPts

Programação (On Click)
SetVisibleInView(texto22,1,false)

RunClickScript(ListaRetas)
SetValue(ListaRetas, Unique(ListaRetas))

SetValue(comp, Length(RetasRef))

If(comp == 0, PlaySound("#q6ns744h"))
If(comp == 1, RunClickScript(TestaReta))
If(comp >1, {RunClickScript(TestaReta), PlaySound("#q7ybueeq")})</td></tr>
</table>

Terminada a validação da reta que contém o gráfico da função linear é dada a possibilidade de repetir o exercício para outro coeficiente da função linear.

Tabela 11 - Protocolo de construção da aplicação “Representação da reta que contém o gráfico de uma função linear de variável racional”, 6ª parte

68º	texto23 = ““\text{Realizou o exercício com sucesso!}”" **Formatação**: posição absoluta no ecrã; não permitir seleção e alterar a cor para verde. **Condição de visualização**: flagReta

69º	Button(“EliminaPontos”)
	 Formatação: alterar nome para “EliminaPontos”
	 Programação (On Click)(Javascript)
	 for (i = ggbApplet.getObjectNumber()-1; I >= 0; I = i-1)
	 {
	 nome= ggbApplet.getObjectName(i);
	 tipo= ggbApplet.getObjectType(nome);
	 if (tipo == "point")
	 {
	 if ((nome != "A_1") && (nome != "A_2") && (nome != "A_3") && (nome != "B_1") && (nome != "B_2") && (nome != "B_3"))
	 {
	 ggbApplet.deleteObject(nome)
	 }
	 }
	 }

70º	Button(“Novo Exercício”)
	 Formatação: posição absoluta no ecrã e alterar nome para “NovoExercicio”
	 Condição de visualização: flagReta
	 Programação (On Click)
	 #Torna as caixas de entrada editáveis
	 SetFixed(InputBox1, true, true)
	 SetFixed(InputBox2, true, true)
	 SetFixed(InputBox3, true, true)
	 SetFixed(InputBox4, true, true)
	
	 #Inicializa as variáveis das caixas de entrada
	 SetValue(x_A,?)
	 SetValue(y_A,?)
	 SetValue(x_B,?)
	 SetValue(y_B,?)
	
	 # Alterar cor do fundo das caixas
	 SetBackgroundColour(InputBox1, "#FFEACD")
	 SetBackgroundColour(InputBox2, "#FFEACD")
	 SetBackgroundColour(InputBox3, "#FFEACD")
	 SetBackgroundColour(InputBox4, "#FFEACD")
	
	 SetVisibleInView(texto6,1, false)
	
	 SetValue(Acertos, 0)
	 SetValue(PtsAcertos, 0)
	 SetValue(RetaAcertos, 0)

```
SetValue(Falhas, 0)
SetValue(Falhas1, 0)
SetValue(PtsFalhas, 0)
SetValue(RetaFalhas, 0)

SetValue(flagPt1, false)
SetValue(flagPt2, false)
SetValue(flagPts, false)
SetValue(flagReta, false)

RunClickScript(EliminaPontos)
Delete(PtsRef)
Delete(RetasRef)

SetValue(a, RandomBetween(-10,10))

SelectObjects(InputBox1)
```

No final deixar apenas as ferramentas  visíveis.

Quando da publicação online da tarefa, esta deve ser editada para ativar a opção “Exibir barra de ferramenta” nas configurações avançadas da aplicação.

Capítulo VIII – Funções polinomiais do primeiro grau

Diogo Meurer de Souza Castro

Resumo:

Apoiados na importância que vemos no ensino e aprendizagem das características da função afim e no uso de tecnologias digitais em sala de aula servindo como suporte para tal, apresentamos uma aplicação criada no GeoGebra cujo objetivo é fazer com que os alunos investiguem o papel que o coeficiente angular realiza na função afim. Para auxiliar o usuário na atividade, foram inseridos feedbacks automáticos tanto para acertos quanto para erros. Além da explicação dos detalhes da atividade, será fornecido um passo a passo da construção.

ENQUADRAMENTO

O conceito de função é iniciado no 9º Ano do Ensino Fundamental e está presente em todo o Ensino Médio, onde o aluno estuda as mais variadas funções: afim, quadrática, exponencial, logarítmica, modular e trigonométrica. Como o primeiro tipo de função que eles estudam é a função afim, acreditamos que ela se torna uma das mais importantes e que deve-se ter um maior cuidado ao apresentar as suas características. Além do mais, através das nossas experiências em sala de aula, vemos a dificuldade que eles têm em utilizar casos de proporcionalidade nesse conteúdo.

DESCRIÇÃO DA TAREFA

Criamos uma atividade no GeoGebra cujo objetivo é fazer com que os alunos investiguem o papel que o coeficiente angular realiza na função afim. Dentro da atividade foram construídos feedbacks automáticos que visam auxiliar os alunos durante a atividade. Na Figura 21vemos a tela inicial da atividade:

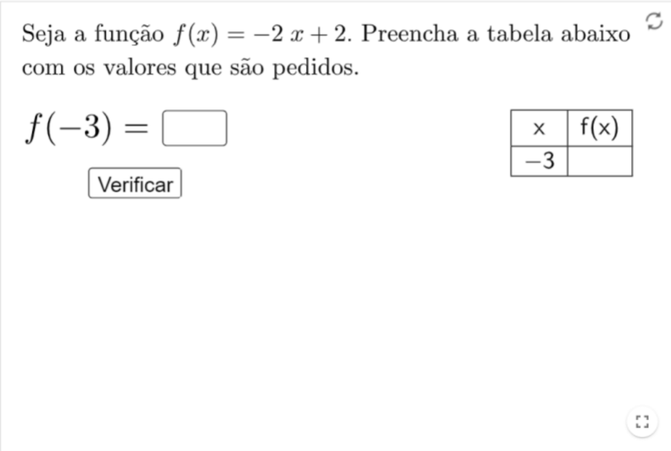


Figura 21 – Figura 9: Tela inicial da tarefa "Funções polinomiais do primeiro grau"

É gerado aleatoriamente uma função afim e pede ao usuário que responda o valor da função dado um valor de x, escolhido também aleatoriamente. A partir daí, se o usuário acertar, o valor da função é inserido na tabela ao lado. Ao preencher toda a tabela, aparecem na tela um feedback textual e o botão REINICIAR para refazer a atividade, ver Figura 22.

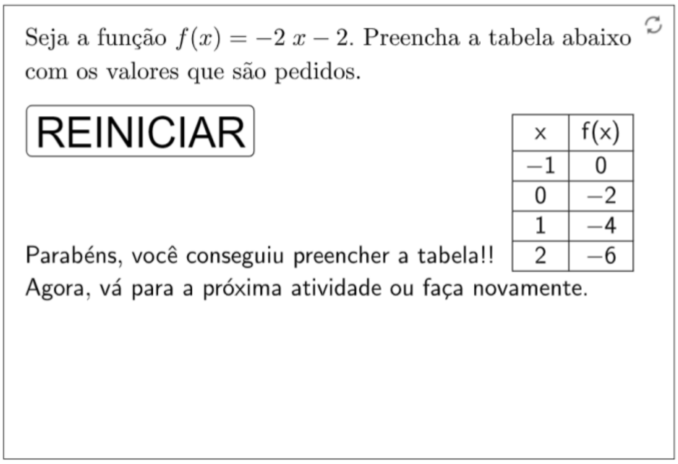


Figura 22 – Tela da tarefa "Funções polinomiais do primeiro grau" com o feedback após uma primeira ação correta.

Quando o usuário erra, a construção tem dois textos previamente criados onde um deles é escolhido aleatoriamente para aparecer na tela:

- Tente novamente.
- Lembre que se ***f(x)=5x-7***, ***f(1)=5.1-7=5-7=-2*** .

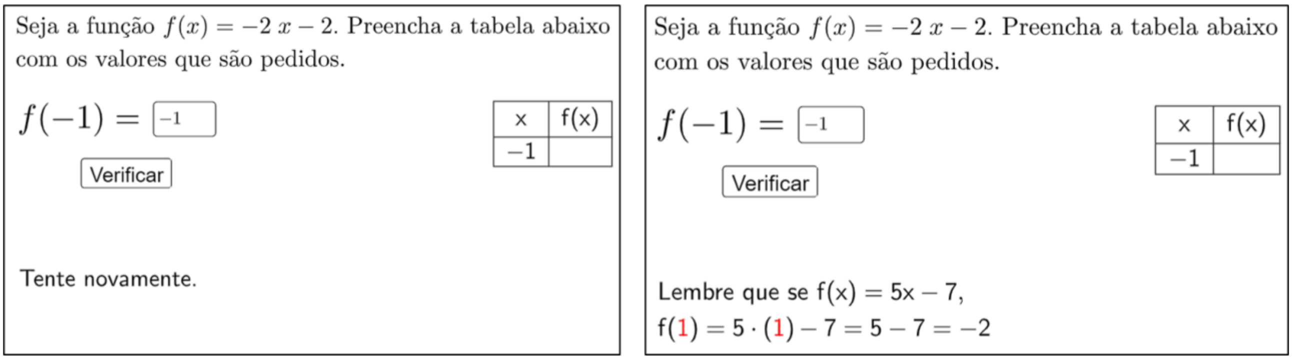


Figura 23 – Tela da tarefa "Funções polinomiais do primeiro grau" com o feedback após uma ação incorreta e respetiva ajuda.

Um segundo feedback foi construído especificamente para o caso em que o(a) usuário(a) erra a multiplicação a.x fazendo −(a,x). Por exemplo, na imagem abaixo, f(-1)=0, mas o(a) usuário(a) fez f(-1)=-2.(-1)-2=-2-2=-4. O feedback que aparece na atividade indica que deve-se ter cuidado com a multiplicação a.x (Figura 24).

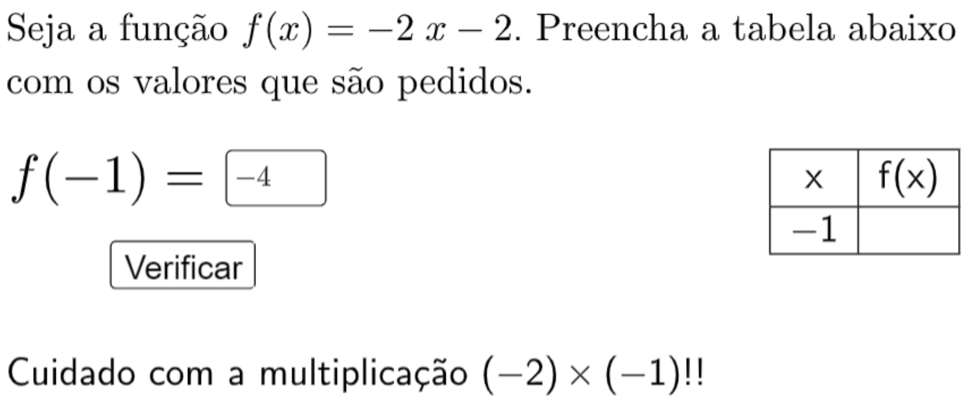


Figura 24 – Tela da tarefa "Funções polinomiais do primeiro grau" com o alerta para introdução do sinal de multiplicação.

A análise geral da aplicação onde constam a análise de possíveis respostas e os feedbacks desenhados podem ser visualizados na Figura 25.

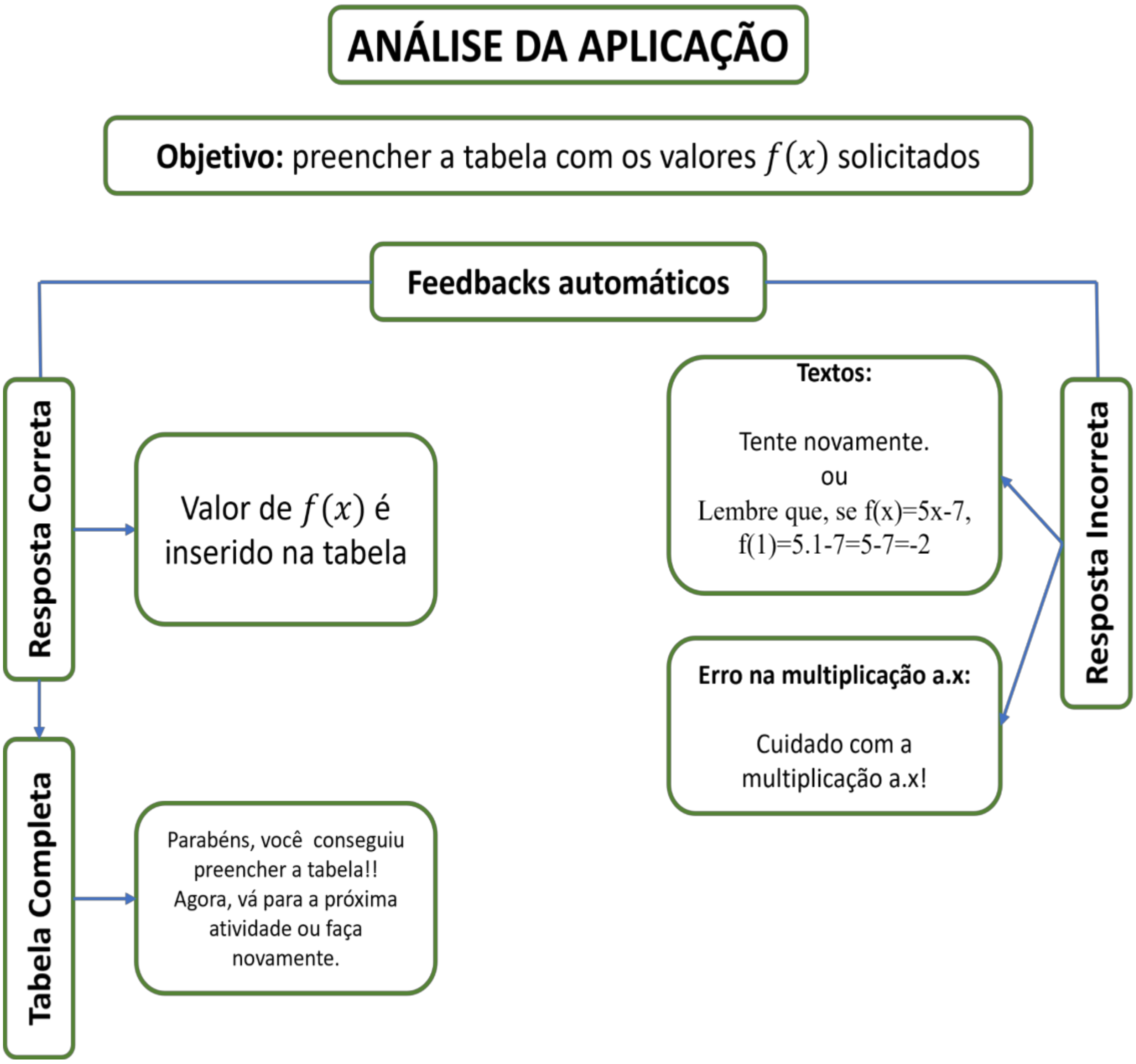


Figura 25 – Matriz de análise da tarefa "Funções polinomiais do primeiro grau"

O diagrama de processo que orientou a construção da aplicação consta da Figura 26.

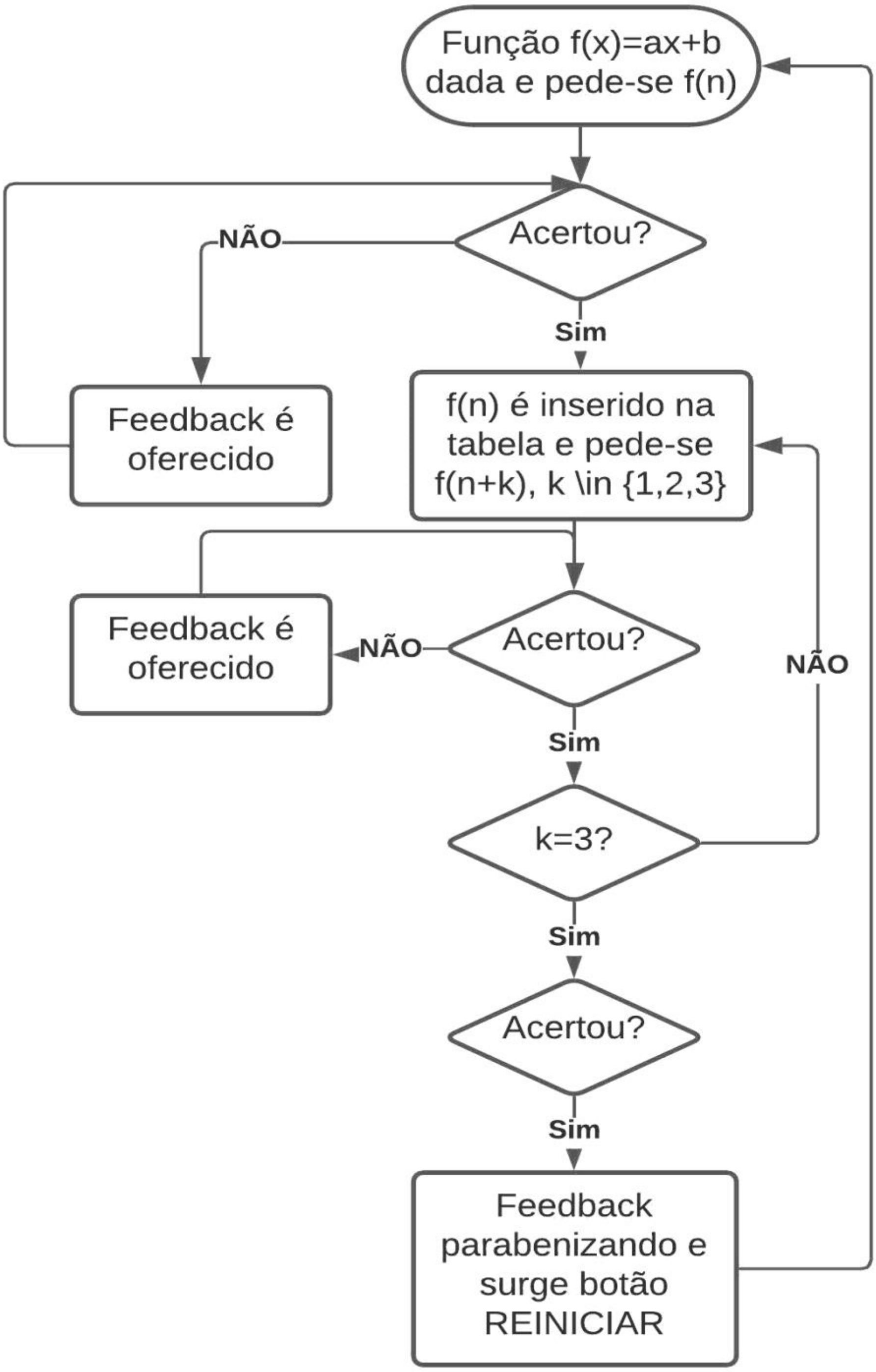
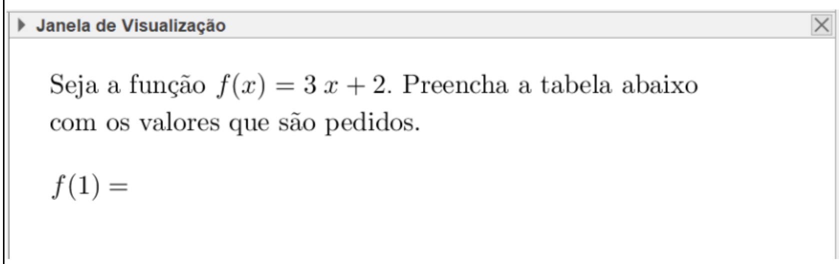
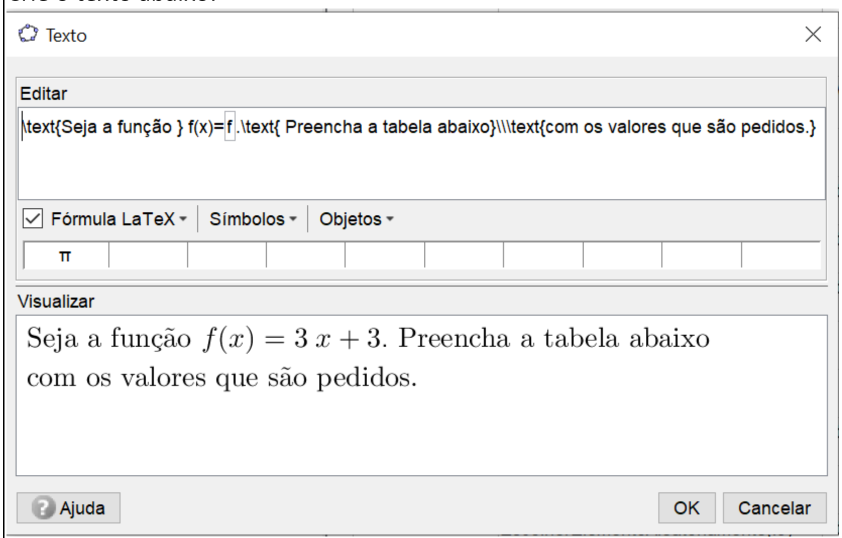


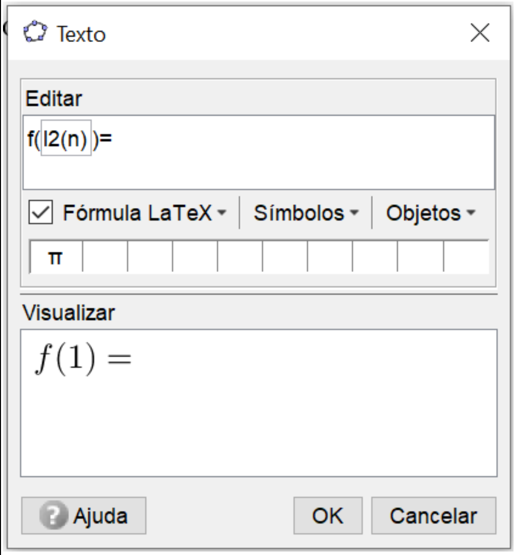
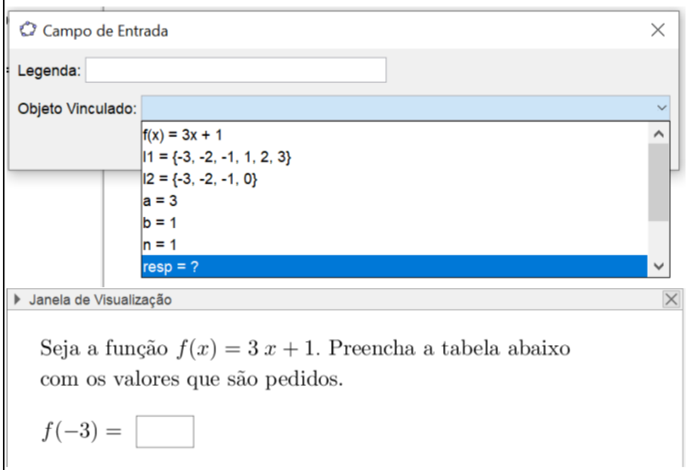
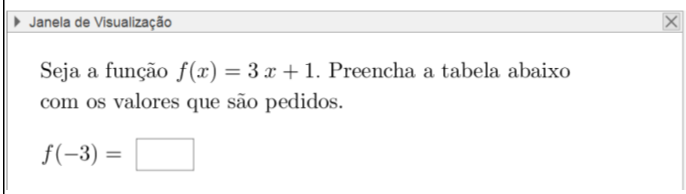
Figura 26 – Diagrama de Processo da aplicação de GeoGebra para a tarefa "Funções polinomiais do primeiro grau".

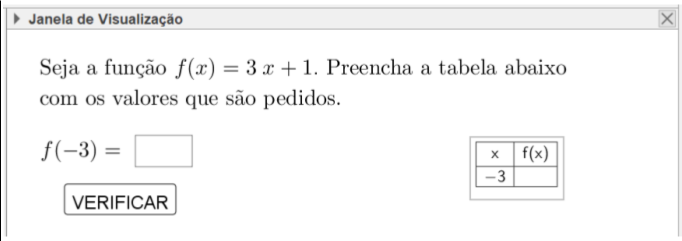
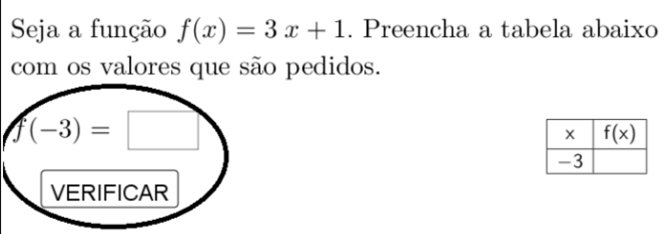
Guião de construção da aplicação

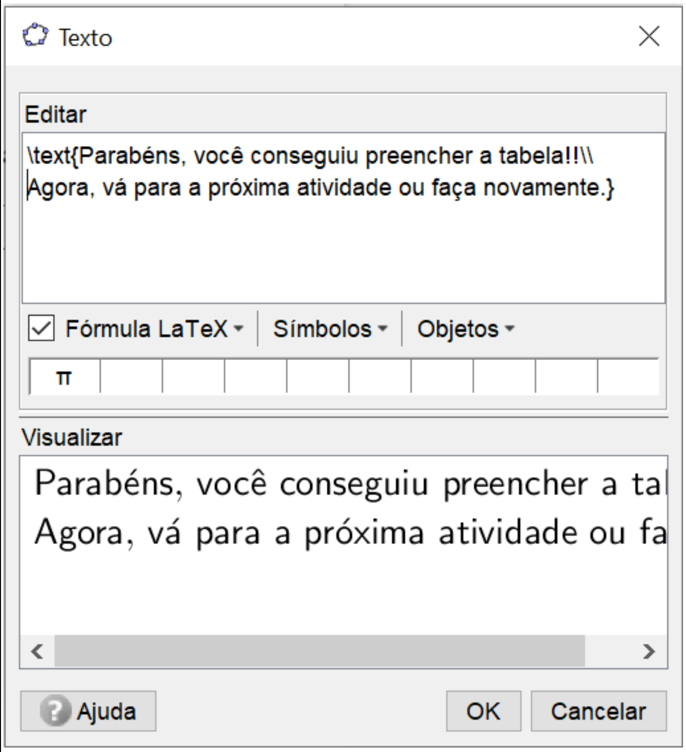
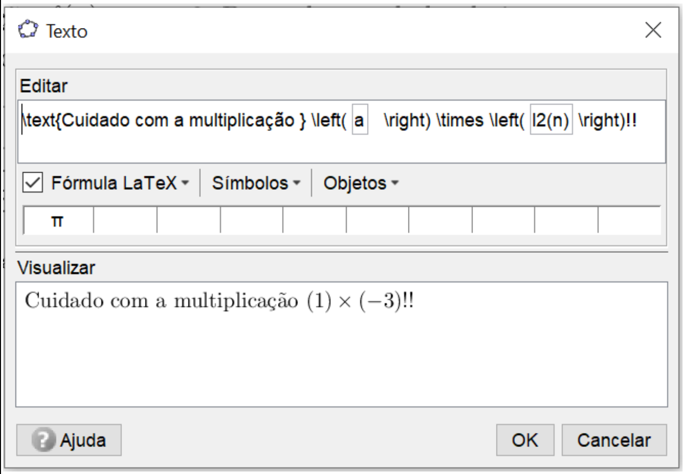
De seguida, iremos apresentar, na Tabela 12 - Protocolo de construção da aplicação "Funções polinomiais do primeiro grau", o passo-a-passo de como construir a primeira aplicação da tarefa.

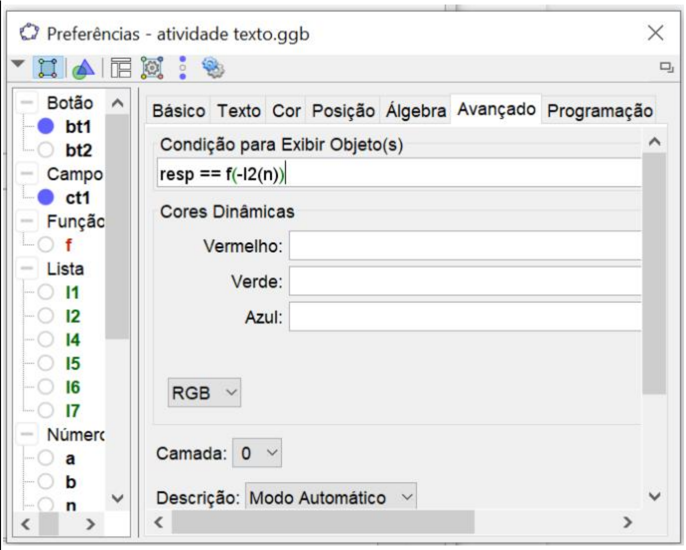
Tabela 12 - Protocolo de construção da aplicação "Funções polinomiais do primeiro grau"

Passo	Objetivo	Comando a escrever na barra de Entrada
1	Criar a função afim:	Crie a lista l1={-3,-2,-1,1,2,3} Defina duas variáveis a e b utilizando o comando EscolherElementoAleatoriamente(l1) Crie a função f(x) através do comando Polinômio(a*x+b) e oculte o seu gráfico.
2	Escolher o primeiro valor de x da tabela e os próximos 3 valores:	O comando x_1 = EscolherElementoAleatoriamente(l1) irá gerar o primeiro valor de x Já o comando Sequência(x_1+k, k, 0, 3) irá nos fornecer os próximos três valores
3	Criar o layout da atividade como a imagem.	 Crie o texto abaixo:  Defina a variável n=1 Criar um outro texto que vá mostrar qual valor de f(x) deverá ser calculado:

		
4	Criar a variável resp=? e o campo de entrada vinculado a essa variável;	
5	Acrescentar o botão VERIFICAR (Sua configuração será feita posteriormente)	
6	Construção da tabela	Crie 4 listas: l4={"f(x)"} l5={"x"} l6=SSequência(x_1 + i, i, 0, mínimo(n-1,3)) l7={} Crie a tabela com o código TabelaDeTexto(Concatenar({l5, l6}), Concatenar({l4, l7}), " |_vc ")

		
7	Criar a caixa de Exibir/Esconder Objetos para os objetos marcados abaixo. Ele será nomeado como c.	
8	Criar botão REINICIAR	Com a programação AtualizarConstrução() DefinirValor(c,true) DefinirValor(d,false) DefinirValor(n,1) DefinirValor(l7,{}) Criar uma caixa de Exibir/Esconder Objetos para ele (será chamada de d). Este botão aparecerá quando o(a) usuário(a) completar a tabela.
9	Primeira configuração do botão Verificar	Se(resp==f(l2(n)),DefinirValor(l7,Anexar(l7,f(l2(n))))) Se(resp==f(l2(n)),DefinirValor(n,n+1)) Se(resp==f(l2(n-1)),DefinirValor(ct1,"")) Se(n==5,DefinirValor(c,false)) Se(n==5,DefinirValor(ct1,"")) Se(n==5,DefinirValor(d,true)) Observe que, a partir de agora, o(a) usuário(a) já consegue responder à questão. Quando a tabela é preenchida, o botão REINICIAR surge para que uma nova função apareça. O que iremos fazer agora é inserir os feedbacks automáticos. Primeiro, iremos fazer o feedback para o caso em que o(a) usuário(a) preencha a tabela e, depois, faremos os outros feedbacks.

10	Crie um texto	 Depois de criar o texto, vá nas propriedades do texto e, na aba Avançado, coloque a condição para que ele seja exibido somente quando n for igual a 5 (observe que, no GeoGebra, utilizamos o símbolo == para o igual).
11	O próximo texto irá aparecer quando o(a) usuário(a) erra a multiplicação.	Para isso, crie o texto abaixo:  Observe que são inseridos dois objetos: a variável a e l2(n). Dessa forma, o texto ficará dinâmico para cada função que aparecerá aleatoriamente na atividade. Agora, nas propriedades do texto, insira a condição abaixo para que ele só apareça quando a resposta inserida for igual a f(-x):

		
12	O terceiro texto será dando uma pista de como aplicamos um valor de x na função.	Para isso, escreva: \text{Lembre que se}f(x)=5x-7,\\f\left(\color{red}{1}\right)= 5\cdot\left(\color{red}{1}\right)-7=5-7=-2

Capítulo IX – Função quadrática e transformações de funções

Alexandre Emanuel Batista Trocado

Resumo:

Tendo como base as Aprendizagens Essenciais da Matemática A do 10º Ano de Portugal e o tema funções, é proposta a construção de uma tarefa em GeoGebra, com feedback automático, que relaciona a representação gráfica de uma função quadrática e as suas transformações simples.

ENQUADRAMENTO

O tema das funções é um dos principais temas da Matemática A do ensino secundário de Portugal tendo em conta as Aprendizagens Essenciais do Ensino Secundário do 10ª Ano de Matemática A (AE 2018). Em particular, as transformações simples dos gráficos de funções onde o gráfico de uma função f é relacionado com os gráficos das funções definidas por $af(x)$, $f(bx)$, $f(x+c)$, $f(x)+d$, $a,b \in R\backslash\{0\}, c,d \in R$. É esperado que o estudante, após a realização da tarefa, perceba melhor qual o contributo que cada alteração da expressão analítica da função f exerce sobre o seu gráfico e escolha quais as alterações a efetuar para que o gráfico seja o pretendido.

DESCRIÇÃO DA TAREFA

A tarefa de feedback automático proposta (Figura 27) tem como principal objetivo colocar à prova os conhecimentos do estudante em conteúdos lecionados previamente, levando-o a refletir e também tentando contribuir para sua autonomia na resolução de problemas onde estes conteúdos são aplicados ao caso particular de uma função quadrática.

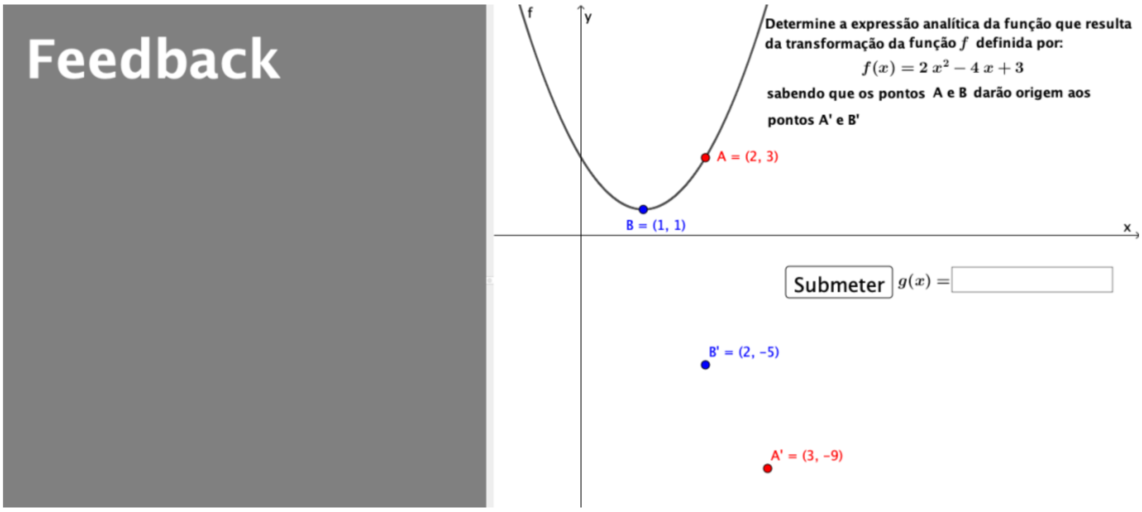


Figura 27 – Imagem da tarefa "Função quadrática e transformações de funções"

Para que chegue ao resultado correto o estudante deverá saber qual a influência de cada parâmetro a, b, c, d na representação gráfica de g a partir de f. Tendo em conta os erros que previsivelmente possam ser cometidos na definição da expressão analítica de g foram estabelecidas cinco mensagem que permitem ao aluno repensar a estratégia implementada na definição da sua resposta, como se pode ver na Figura 28.

Numa primeira versão desta tarefa as mensagens encontram-se definidas através de caixas de texto distintas que se encontram ocultas e que vão sendo mostradas à medida que é submetida uma resposta, tendo como limite cinco tentativas a partir das quais será mostrada a solução.

Após a implementação desta tarefa numa aula do 10º ano, deverão ser registadas as reações dos estudantes à realização da tarefa, tentando entender quais os maiores obstáculos sentidos e quais as mensagens de feedback que poderão ser melhoradas. Em particular, se o feedback fornecido pela atividade deverá ainda ser melhorado quando o estudante se aproxima da resposta correta ou se, por outro lado, há erros cometidos que não foram previstos, mas que possam ser incluídos. A recolha das reações dos estudantes deverá ser efetuada através do preenchimento de um pequeno inquérito no final da tarefa e, caso seja possível, também através da filmagem da aula.

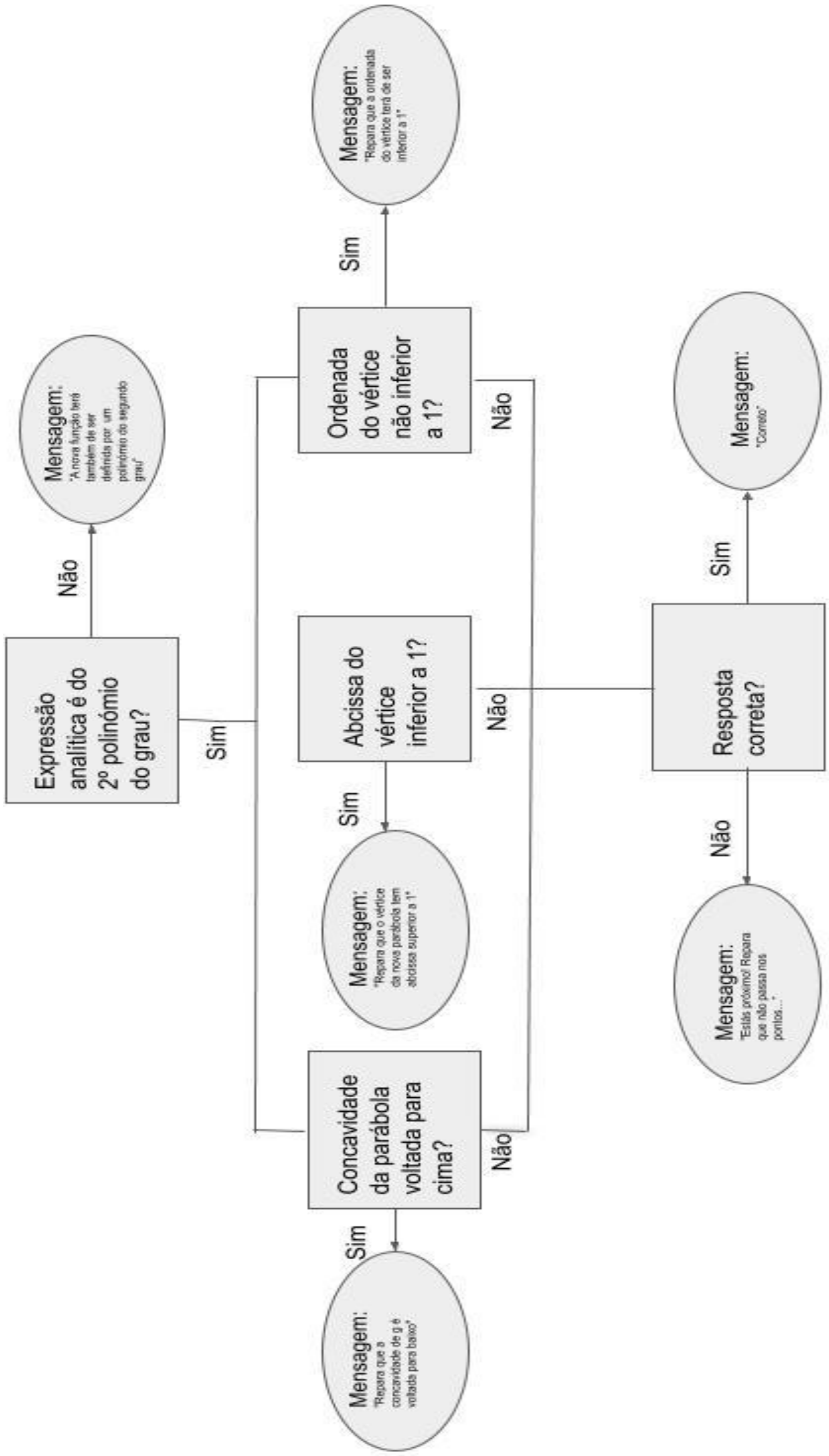


Figura 28 – Matriz de análise e diagrama de processo da tarefa "Função quadrática e transformações de funções"

Guião de construção da aplicação

Para a construção desta tarefa deverão estar visíveis a Folha Gráfica 2D e a Folha Gráfica 2D 2 onde esta última deverá ser utilizada como janela dedicada ao Feedback fornecido pela aplicação.

Definição da função f e dos pontos A e B na Folha Gráfica 2D

Tabela 13 - Protocolo de construção da aplicação "Função quadrática e transformações de funções", 1ª parte

1º	f(x) = 2*(x - 1)^2+1 **(Localização):** Folha 2D
2º	A=(2,f(2)) **(Localização):** Folha 2D
3º	B=(1,f(1)) **(Localização):** Folha 2D

Definição da resposta ao exercício (a função h) e os pontos correspondentes a A' e B':

Tabela 14 - Protocolo de construção da aplicação "Função quadrática e transformações de funções", 2ª parte

4º	h(x)=-2*f(x - 1) – 3 **Formatação:** deverá estar escondido
5º	A1=(3,h(3))
6º	B1=(2,h(2))

Criação das ferramentas responsáveis pela interação com o utilizador:

Tabela 15 - Protocolo de construção da aplicação "Função quadrática e transformações de funções", 1º parte

7º	ct=0 **Comentário**: definição do contador para o número de tentativas.
8º	a=false **Comentário:** variável boleana que permitirá esconder e mostrar objetos.
9º	Botão("Submeter") **Programação: (Se clicar)** Executar({"ct=ct+1"}) Executar({"a=true"}) **Comentário:** O contador das tentativas será adicionado 1 e deverão ser mostrados elementos.
10º	g(x)=? **Comentário:** função que irá armazenar a resposta submetida. **Avançado: (Condição para Mostrar Objeto)** a $\stackrel{?}{=}$ true ∧ ct ≥ 1 ∧ ct ≤ 5
11º	CaixadeEntrada(g) **Básico:** Sem mostrar rótulo. **Programação: (Se clicar)** a=false **Comentário:** espaço onde será submetida a resposta.
12º	Texto("g(x)=")
13º	coef=Coeficientes(g(x)) **Comentário:** criação da lista de coeficientes do polinómio de g .
14º	lider=Elemento(coef,1) **Comentário:** definição do coeficiente de maior grau de g .
15º	a1=Elemento(coef,1) b1=Elemento(coef,2) c1=Elemento(coef,3) **Comentário:** definição dos coeficientes da resposta
16º	h1=(-b1) / (2a1) k1=(-(b1² - 4a1 c1)) / (4a1) **Comentário:** definição coordenadas do vértice da parábola definida por g .
17º	respondido=Se(a1 $\stackrel{?}{=}$ -4 ∧ b1 $\stackrel{?}{=}$ 16 ∧ c1 $\stackrel{?}{=}$ -21 ∧ ct ≤ 5, true, false) **Comentário:** variável boleana que avalia se a resposta está correta e o número de tentativas é inferior ou igual a 5.
18º	fs=Se(Comprimento(coef) $\stackrel{?}{=}$ 3, true, false) **Comentário:** variável boleana que avalia se a resposta é um polinímio de grau 2

	(logo 3 coeficientes).
19º	text1=Texto("A nova função terá também de ser definida por um polinómio do segundo grau") **Avançado: (Condição para Mostrar Objeto)** Comprimento(coef) ≠ 3 ∧ a ≟ true ∧ respondido ≟ false∧ ct ≤ 5 **(Localização): Folha 2D 2**
20º	text2=Texto(”Repara que a concavidade é voltada para baixo”) **Avançado: (Condição para Mostrar Objeto)** ct > 0 ∧ fs ≟ true ∧ lider > 0 ∧ a ≟ true ∧ respondido ≟ false∧ ct ≤ 5 **(Localização):** Folha 2D 2
21º	text3=Texto(”Repara que o vértice da nova parábola tem abcissa superior a 1") **Avançado: (Condição para Mostrar Objeto)** a ≟ true ∧ fs ≟ true ∧ h1 ≤ 1 ∧ respondido ≟ false∧ ct ≤ 5 **(Localização):** Folha 2D 2
22º	text4=Texto(”Repara que a ordenada do vértice terá de ser inferior a 1") **Avançado: (Condição para Mostrar Objeto)** a ≟ true ∧ fs ≟ true ∧ k1 ≥ 1 ∧ respondido ≟ false∧ ct ≤ 5 **(Localização):** Folha 2D 2
23º	prox=a ≟ true ∧ fs ≟ true ∧ (g(2) ≠ -5 ∨ g(3) ≠ -9) **Comentário:** valor boleano que avalia se a resposta falha algum dos pontos.
24º	text5=Texto("Estás próximo! Repara que não passa nos pontos...") **Avançado: (Condição para Mostrar Objeto)** a ≟ true ∧ fs ≟ true ∧ a1 < 0 ∧ prox ≟ true ∧ respondido ≟ false ∧ ct ≤ 5 **(Localização):** Folha 2D 2
25º	Vector(B,B1) **Comentário:** Seta que ajudará a identificar a transformação de um vértice de uma parábola noutro à 5ª tentativa falhada. **Avançado: (Condição para Mostrar Objeto)** ct ≟ 5 **(Localização):** Folha 2D
26º	text6=Texto(”Repara!”) **Comentário:** Mensagem de ajuda quando se chega à 5º tentativa falhada. **Avançado: (Condição para Mostrar Objeto)** ct ≟ 5 **(Localização):** Folha 2D
27º	text7=Texto(”Excesso de tentativas! Esta é a solução:”) **Avançado: (Condição para Mostrar Objeto)** ct > 5 **(Localização):** Folha 2D 2
28º	text8=”g(x)=-2\times f(x-1)-3” **Texto:** Fórmula LaTeX **Avançado: (Condição para Mostrar Objeto)** ct>5 **(Localização):** Folha 2D 2
29º	text9=Texto(”Correto!!!”) **Comentário:** Mensagem a ser exibida quando é respondido corretamente

	dentro das 5 tentativas. **Avançado: (Condição para Mostrar Objeto)** a1 $\stackrel{?}{=}$ -4 ∧ b1 $\stackrel{?}{=}$ 16 ∧ c1 $\stackrel{?}{=}$ -21 ∧ ct ≤ 5 **(Localização):** Folha 2D 2
30º	text10=Texto(”Determine a expressão analítica da função que resulta da transformação da função f definida por:”) **(Localização):** Folha 2D
31º	text11=”f(x)=2 x^2-4 x+3” **Texto:** Fórmula LaTeX **(Localização):** Folha 2D
32º	text12=Texto(”sabendo que os pontos A e B darão origem aos pontos A1 e B1”) **(Localização):** Folha 2D
33º	text13=”Feedback” **Texto:** Muito grande **(Localização):** Folha 2D 2

Todos os elementos criados anteriormente devem ser posicionados de forma a ocuparem uma posição absoluta no ecrã e fixa, como se pode ver na Figura 27 apresentada no início da secção anterior.

Capítulo X – Factorização de polinómios: “Regra de Ruffini”

Rui Josué Boucinha Torres Eusébio

Resumo:

A criação desta tarefa visa ilustrar como um tópico matemático, que se caracteriza pela utilização de procedimentos e algoritmos bem definidos, pode ser auxiliado pelas potencialidades do GeoGebra. É criada e descrita uma aplicação para a regra de Ruffini, um algoritmo que auxilia a fatorização de polinómios reais e de variáveis reais, nomeadamente no contexto do Ensino Secundário, importante para o desenvolvimento de divisibilidade de polinómios. Assim apresenta-se de modo breve a ideia que orientou a conceção da aplicação, bem como a programação que lhe está associada.

ENQUADRAMENTO

Os tipos de modelos de raciocínio desenvolvidos ao trabalhar com equações polinomiais podem ser generalizados para outras situações. Através dos polinómios, podem-se introduzir noções de nível superior sobre funções. A solução de problemas que, à primeira vista, parecem não ter qualquer ligação com polinômios, acaba dependendo muito deles. Os polinómios são onipresentes em matemática, e é importante que os alunos os dominem com segurança e é extremamente importante no Ensino Básico, pois, através deles, podemos entender muitos aspetos do pensamento matemático através de seu estudo.

DESCRIÇÃO DA TAREFA

Pensando numa forma automática, os alunos vão aferindo os passos da regra de Ruffini e a aplicação do GeoGebra da tarefa “Regra de Ruffini” oferece o feedback instantâneo sobre as ações do utilizador.

Apesar de no início a aplicação apresentar dois polinómios o utilizador pode alterá-los, inserindo um polinómio dividendo, e um outro do divisor. A primeira ação solicitada ao utilizador é que escreva o número de termos que tem o polinómio dividendo na regra de

Ruffini. A inserção de um valor incorreto fá-lo representar a vermelho, apesar do feedback imediato, o utilizador pode recorrer ao botão ajuda sendo apresentado feedback escrito à verde, ver Figura 29.

Figura 29 - Imagem da tarefa "Regra de Ruffini", 1º feedback em ação incorreta e ajuda.

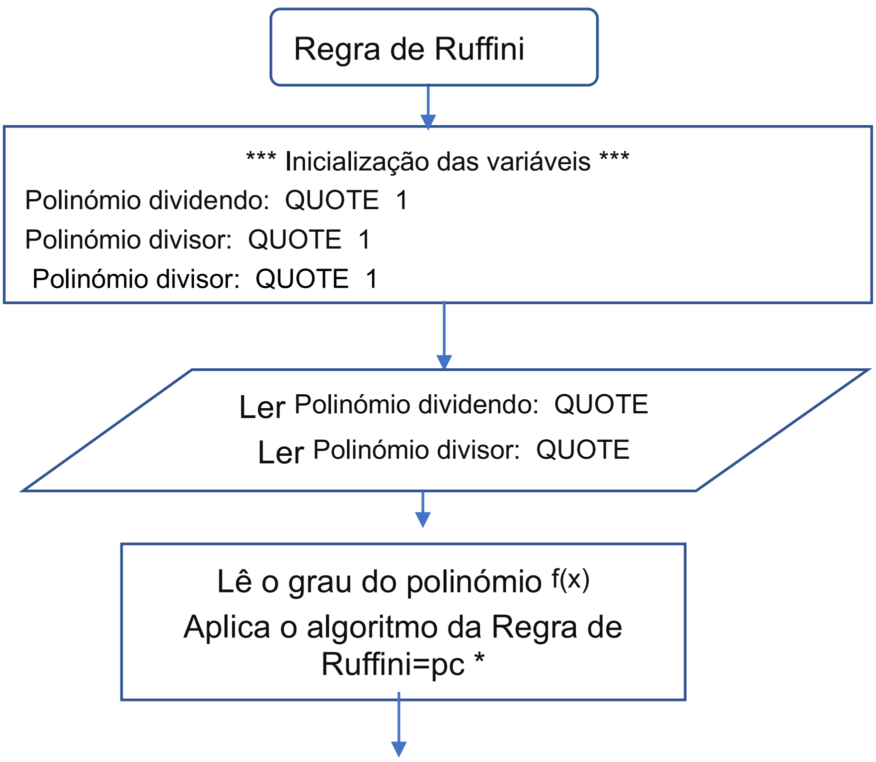
Logo que a primeira ação decorre de forma certa, é apresentado o esquema que orienta o algoritmo de Ruffini, sendo apresentado feedback para a ação, ver Figura 30 , ajudando o aluno a prosseguir, podendo ou não voltar a exibir a ajuda que se apresenta com mais itens.

Figura 30 - Imagem da tarefa "Regra de Ruffini", 2º feedback em ação incorreta.

A interação durante a aplicação é norteada pelo completar do algoritmo. Sempre que os valores introduzidos permanecerem à azul é a indicação que se está a desenvolver ou se concluiu trabalho de modo adequado (Figura 31), caso contrário o texto aparece a vermelho, sendo a indicação que o utilizador deve rever a sua ação .

Figura 31 - Imagem da tarefa “Regra de Ruffini”, 2º feedback em ação incorreta.

A construção desta aplicação orienta-se pela matriz de análise e diagrama de processo apresentado na Figura 32.



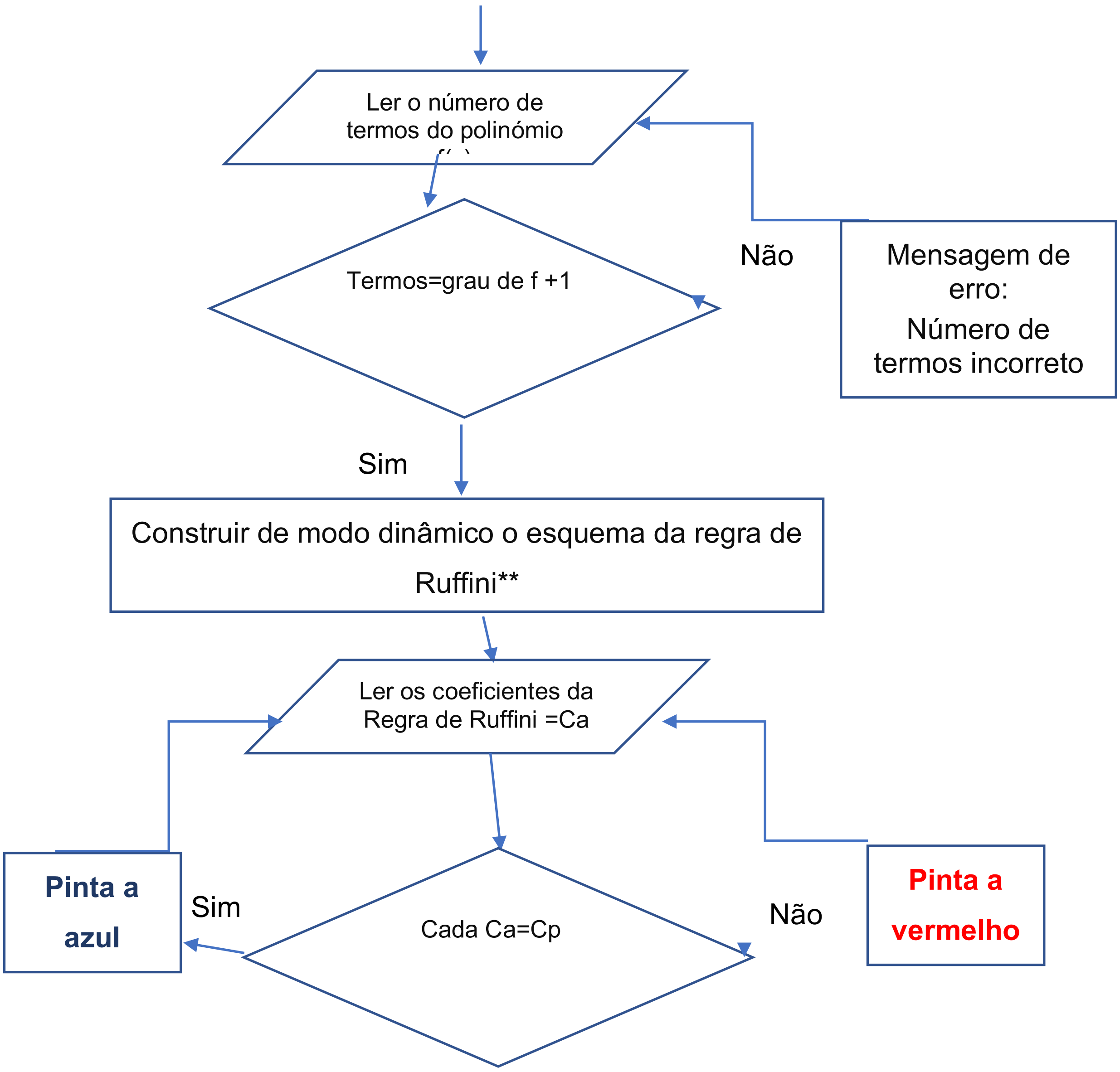


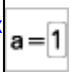
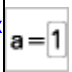
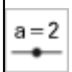
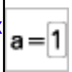
Figura 32 – Diagrama de processo da tarefa "Regra de Ruffini"

Guião de construção da aplicação

A construção desta aplicação exige uma programação bastante complexa, orientando-se pelo diagrama de processo apresentado na secção anterior. De facto, há um conjunto de 102 passos para a construção da aplicação, ilustrados na Tabela 16: Protocolo de construção da aplicação "Regra de Ruffini", que se obteve com auxílio do protocolo de construção no GeoGebra.

Tabela 16 - Protocolo de construção da aplicação "Regra de Ruffini"

Passo		Tipo Nome do objeto	Ícone da Ferramenta	Descrição	Valor
1		Text texto1			"Regra de Ruffini"
2		Function f			$f(x) = x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x + 1$
3		List lista1		Coefficients(f)	lista1 = {1, 1, 1, 1, 1, 1, 1, 1}
4		Number comp1		Length(lista1)	comp1 = 8
5		Function g			g(x) = x + 1
6		List lista2		Coefficients(g)	lista2 = {1, 1}
7		Number div		-lista2(2)	div = -1
8		Number pa7		If(comp1 ≥ 8, Element(lista1, 1), 0)	pa7 = 1
9		Number pa6		If(comp1 ≥ 7, Element(lista1, comp1 - 6), 0)	pa6 = 1
10		Number pa5		If(comp1 ≥ 6, Element(lista1, comp1 - 5), 0)	pa5 = 1
11		Number pa4		If(comp1 ≥ 5, Element(lista1, comp1 - 4), 0)	pa4 = 1
12		Number pa3		If(comp1 ≥ 4, Element(lista1, comp1 - 3), 0)	pa3 = 1
13		Number pa2		If(comp1 ≥ 3, Element(lista1, comp1 - 2), 0)	pa2 = 1
14		Number pa1		If(comp1 ≥ 2, Element(lista1, comp1 - 1), 0)	pa1 = 1
15		Number pa0		If(comp1 ≥ 1, Element(lista1, comp1), 0)	pa0 = 1
16		Number pc7		pa5	pc7 = 1
17		Number pb6		pc7 div	pb6 = -1

18		Number pc6		pa6 + pb6	pc6 = 0
19		Number pb5		pc6 div	pb5 = 0
20		Number pc5		pa5 + pb5	pc5 = 1
21		Number pb4		pc5 div	pb4 = -1
22		Number pc4		pa4 + pb4	pc4 = 0
23		Number pb3		pc4 div	pb3 = 0
24		Number pc3		pa3 + pb3	pc3 = 1
25		Number pb2		pc3 div	pb2 = -1
26		Number pc2		pa2 + pb2	pc2 = 0
27		Number pb1		pc2 div	pb1 = 0
28		Number pc1		pa1 + pb1	pc1 = 1
29		Number pb0		pc1 div	pb0 = -1
30		Number pc0		pa0 + pb0	pc0 = 0
31		Input Box cxpoldividendo		InputBox(f)	cxpoldividendo
32		Input Box cx$_{p}$oldivisor		InputBox(g)	cx$_{p}$oldivisor
33		Number termos			termos = ?
34		Input Box cxnumtermos		InputBox(termos)	cxnumtermos
35		Text texto2			"Insere o polinómio dividendo"
36		Text texto3			"Insere o polinómio divisor da forma x+α"
37		Text texto4			"Polinómio divisor da forma x+α tem zero -α"

38		Text texto5			"Coeficientes do polinómio divisor"
39		Point A			A = (-3.44, -1.1)
40		Point B		(x(A) - 2, y(A))	B = (-5.44, -1.1)
41		Point C		(x(A), y(A) + 2)	C = (-3.44, 0.9)
42		Point D		(x(A), y(A) - 1)	D = (-3.44, -2.1)
43		Point E		(x(A) + termos + 1, y(A))	E = (?, ?)
44		Segment h		Segment C, D	h = 3
45		Segment i		Segment B, E	i = ?
46		Number d			d = ?
47		Number a0			a0 = ?
48		Number a1			a1 = ?
49		Number a2			a2 = ?
50		Number a3			a3 = ?
51		Number a4			a4 = ?
52		Number a5			a5 = ?
53		Number a6			a6 = ?
54		Number a7			a7 = ?
55		Number b0			b0 = ?
56		Number b1			b1 = ?
57		Number b2			b2 = ?

58		Number b3			b3 = ?
59		Number b4			b4 = ?
60		Number b5			b5 = ?
61		Number b6			b6 = ?
62		Number c0			c0 = ?
63		Number c1			c1 = ?
64		Number c2			c2 = ?
65		Number c3			c3 = ?
66		Number c4			c4 = ?
67		Number c5			c5 = ?
68		Number c6			c6 = ?
69		Number c7			c7 = ?
70		Input Box cdiv		InputBox(d)	cdiv
71		Text texto6			"Correto"
72		Text texto7			"Errado"
73		Boolean Value b			b = false
74		Input Box Ca7		InputBox(a7)	Ca7
75		Input Box Ca6		InputBox(a6)	Ca6
76		Input Box Ca5		InputBox(a5)	Ca5
77		Input Box Ca4		InputBox(a4)	Ca4

78		Input Box Ca3	a=1	InputBox(a3)	Ca3
79		Input Box Ca2	a=1	InputBox(a2)	Ca2
80		Input Box Ca1	a=1	InputBox(a1)	Ca1
81		Input Box Ca0	a=1	InputBox(a0)	Ca0
82		Input Box Cb6	a=1	InputBox(b6)	Cb6
83		Input Box Cb5	a=1	InputBox(b5)	Cb5
84		Input Box Cb4	a=1	InputBox(b4)	Cb4
85		Input Box Cb3	a=1	InputBox(b3)	Cb3
86		Input Box Cb2	a=1	InputBox(b2)	Cb2
87		Input Box Cb1	a=1	InputBox(b1)	Cb1
88		Input Box Cb0	a=1	InputBox(b0)	Cb0
89		Input Box Cc7	a=1	InputBox(c7)	Cc7
90		Input Box Cc6	a=1	InputBox(c6)	Cc6
91		Input Box Cc5	a=1	InputBox(c5)	Cc5
92		Input Box Cc4	a=1	InputBox(c4)	Cc4
93		Input Box Cc3	a=1	InputBox(c3)	Cc3
94		Input Box Cc2	a=1	InputBox(c2)	Cc2
95		Input Box Cc1	a=1	InputBox(c1)	Cc1
96		Input Box Cc0	a=1	InputBox(c0)	Cc0
97		Button botão1	OK		botão1

98		Text texto8			"Rui Eusébio ^{ \copyright}"
99		Number gr		Degree(f)	gr = 7
100		Text texto9		"Grau do polinómio = " + gr + " O número de termos é sempre " + gr + "+1=" + comp1 + ""	"Grau do polinómio = 7 O número de termos é sempre 7+1=8"
101		Text texto10			"Um polinómio de grau n tem n+1 termos."
102		Text texto11		"" + d + " não é um zero do divisor."	"? não é um zero do divisor."

Capítulo XI– Conclusões

Neste Capítulo serão apresentados excertos dos depoimentos dos professores sobre as respetivas participações no desenvolvimento do projeto de investigação “O GEOGEBRA COMO ESTRATÉGIA PARA ENSINO REMOTO: CRIANDO ATIVIDADES COM FEEDBACK AUTOMÁTICO”, aprovado nas instâncias da Pontifícia Universidade Católica de São Paulo, Brasil atendendo ao edital PIPRINT de 2021.

No desenvolvimento deste estudo o uso do GeoGebra foi o recurso principal na criação de tarefas para o ensino da matemática, ajustadas aos interesses, às necessidades e aos problemas que enfrentam os professores nas escolas, alguns deles investigando o uso destes materiais nos contextos escolares e o seu efeito na melhoria dos resultados dos estudantes e criando possibilidades de feedback imediato em suas propostas de avaliação.

Uma professora participante argumenta que:

> A realização de tarefas com *feedback* automático ajuda o estudante a alcançar os objetivos delineados de forma mais autónoma e dá-lhe a possibilidade de evoluir ao seu ritmo, potenciando a aprendizagem dos conteúdos trabalhados. Por outro lado, o professor consegue melhorar a monitorização do trabalho dos estudantes e dar um apoio mais individualizado aos estudantes que dele carecem. A realização de tarefas desta natureza enriquece a atividade docente e melhora a didática da Matemática, no sentido em que o professor antes de propor uma tarefa, para além dos objetivos que se propõe a alcançar com a sua realização, já refletiu nos erros que o estudante poderá cometer e nos feedbacks progressivos que lhe pode facultar para desbloquear situações que impeçam a progressão/consolidação das aprendizagens.

E ainda salienta que:

> A criação de tarefas com feedback automático com recurso ao GeoGebra requer

> algum domínio deste software, pois é natural que a sua implementação imponha alguns desafios.

Tal afirmação vem ao encontro sobre o conteúdo proposto no Capítulo II pelos coordenadores do projeto, com o objetivo de minimizar os desafios que se apresentariam.

Consideramos que tarefas com *feedback* automático permitem a criação de um contexto tecnológico que propicia o envolvimento do utilizador da aplicação, uma vez que este recebe *feedbacks* que lhe permitem obter a confirmação da resposta correta, ou pistas para melhorar a sua resposta até poder responder de modo acertado.

Num outro depoimento, dois participantes afirmam que o projeto contribuiu para o desenvolvimento de competências matemáticas, didáticas, curriculares e tecnológicas, bem como para o desenvolvimento profissional. Consideram importante que outros professores que venham a integrar-se neste projeto possam perceber a dinâmica do feedback automático, num cenário hipotético de aprendizagem e que se inspirem para a implementação nas suas aulas, em prol da aprendizagem significativa da matemática dos seus alunos.

É significativo constatar que embora finalizando os trabalhos, houve o entendimento da importância de um *feedback* formativo que possa contribuir para aprendizagem dos alunos e para aprimoramento pedagógico do professor como declara outro participante.

> Pretendemos continuar com essa pesquisa pois acreditamos que a aplicação que foi construída pode ser aperfeiçoada após realizações desta situação didática em sala de aula. Ao perceber os erros que os alunos vão cometendo, novos feedbacks automáticos podem ser inseridos e outros ajustes também podem ser feitos.

Noutro depoimento a participante afirma que:

> As expectativas iniciais face aos objetivos do projeto foram atingidas, permitindo-me aperfeiçoar o planeamento e a execução da minha prática letiva. Os conhecimentos adquiridos orientarão a elaboração de tarefas desafiantes e

> motivadoras para ajudar o aluno na construção e autorregulação do seu conhecimento matemático.

Para o aprimoramento profissional dos participantes foi gratificante verificar que temos consciência da necessidade de inovarmos sempre como afirma um professor:

> A participação neste projeto ajudou-nos a perceber que havia muita coisa a ser aprendida e forneceu-nos mais ferramentas para o nosso trabalho. O projeto contribuiu para aperfeiçoamos em diversos domínios e inteirarmos de vários aspetos de que não conhecíamos, concretamente os comandos de programação e a introdução de som nas aplicações.

Importa, também, referir que a utilização de tarefas com *feedback* automático é importante no contexto de ensino remoto, mas tem também muito potencial no ensino presencial, precisamente pelas razões elencadas acima, embora ainda subsistam alguns impasses na sua utilização.

Apoios e questões éticas

Este trabalho foi apoiado pelo Plano de Incentivo ao Programa de Internacionalização da Pós-Graduação de Pontifícia Universidade Católica de São Paulo (PIPRINT-PG EDITAL PIPRINT-PG 9302/ 2020 BRASIL), em parte, por fundos do Estado Português através da Fundação para a Ciência e a Tecnologia (FCT), I.P., no âmbito do projeto UIDB/05198/2020 (Centro de Investigação e Inovação em Educação, inED), e pela Organização de Estados Ibero-americanos para a Educação, a Ciência e a Cultura, através do escritório de Lisboa.

Todos os participantes no projeto salvaguardaram as considerações éticas inerentes ao trabalho e investigação desenvolvida. Refira-se ainda que o projeto foi submetido a avaliação do Comitê de Ética em Pesquisa da PUC/SP campus Monte Alegre (CEP-PUC/SP), tendo recebido o parecer favorável com o número 4.747.583 .

Referências

Abar, C. A. A. P. & Cotic, N. (2014) GeoGebra: na produção do conhecimento matemático. São Paulo: Iglu.

Ministério da Educação. (2018) Base Nacional Comum Curricular. Brasília: Ministério da Educação.

Breda, A., Trocado, A., & Santos, J. (2013). O GeoGebra para além da segunda dimensão. *Indagatio Didactica*, *5*(1), 60-84. ISSN: 1647-3582. https://doi.org/10.34624/id.v5i1.4304

Brosseau, G. (2008) Introdução ao estudo das situações didáticas: conteúdos e métodos de ensino. São Paulo: Ática.

Brousseau, G. (2013) Introduction à l´ingénierie didactique. Laboratoire, Cultures, Education, Sociétés (LACES). Université Bordeaux 2. http://guy-brousseau.com/wp-content/uploads/2013/12/Introduction-%C3%A0-ling%C3%A9nierie-didactique3.pdf

Brousseau, G. (1983) Les obstacles épistemologiques et les problémes en mathématiques. http://guy-brousseau.com/2760/introduction-a-l%E2%80%99ingenierie-didactique-2013/

Cabo, M., Dos Santos, J.,Fernandes, N. & Trocado, A. (2012) GeoGebra para a Sala de Aula. Curso 11, ProfMat 2012. Coimbra

Cury, H. N. (2008) Análise de erros: o que podemos aprender com os erros dos alunos. Belo Horizonte: Autêntica.

Dos Santos, José Manuel. "Quais as soluções da equação (x^ 2-5x+ 5)^(x^ 2-9x+ 20)." *UNIÓN. Revista Iberoamericana de Educación Matemática* 29 (2012): 161-172. http://funes.uniandes.edu.co/15794/1/DosSantos2012Quais.pdf

Dos Santos, J. M. D. S. (2012b) Introducción a GeoGebra 3D - Taller. XIV Congreso de Enseñanza y Aprendizaje delas Matemáticas. Sociedad Andaluza de Educación Matemática THALES. 4 a 6 de Julho, Málaga.

Dos Santos, J. M. D. S., Geraldes, J. Ribeiro, A. & Trocado, A. (2010). Como usar o GeoGebra para ensinar e aprender Matemática. Curso n.3, ProfMat 2010. Aveiro. http://www.geogebra.org.pt/images/arquivos/profmat_2010/profmat_2010_c3.pdf

Dos Santos, J. M. D. S & Trocado, A. (2008) Estudo de isometrias com o GeoGebra como abordagem possível para o 1o e 2o Ciclo do Ensino Básico no contexto do reajustamento do Programa de Matemática do Ensino Básico. Sessão Pratica, MinhoMat 2008. Vila Verde. http://www.geogebra.org.pt/images/arquivos/minhomat_2008/MinhoMat_2008.pdf

Dos Santos, J. M. D. S & Trocado, A. (2013) GeoGebra 3D. Curso 1 do MinhoMat2013 – Escola EB 3/S de Arcos de Valdevez. http://www.geogebra.org.pt/ficheiros_on_line/pdf/geogebra3d-MinhoMat2013_pub.pdf

Flores, J. & Dos Santos, J. M. D. S (2013) GeoGebra no Ensino Básico. Sessão Prática 1 do MinhoMat2013 – Escola EB 3/S de Arcos de Valdevez.

Freitas, J. L. M. (2008). Teoria das Situações Didáticas. In: Machado, S.D.A.(Org.). *Educação Matemática: uma (nova) introdução. 3ª ed*. (pp. 77-111). São Paulo(SP): EDUC.

Igliori, S. B. C. (2008) A noção de obstáculo epistemológico e a educação matemática. In: Machado, S.D.A.(Org.). *Educação Matemática: uma (nova) introdução. 2ª ed.* (pp. 99-113). São Paulo(SP): EDUC.

Ministério da Educação (2018a). Aprendizagens Essenciais, Articulação com o Perfil dos Alunos - Matemática - Ensino Básico, 3.º ciclo, Matemática, 7° ano. Direção-Geral da Educação, Lisboa: Ministério da Educação.

Ministério da Educação (2018b). Aprendizagens essenciais - Articulação com o perfil dos alunos – 10° ano - Ensino secundário - Matemática A. Direção-Geral da Educação, Lisboa: Ministério da Educação.

Nicol, D. J., & Macfarlane-Dick, D. (2006). Formative assessment and self-regulated learning: A model and seven principles of good feedback practice. *Studies in higher education*, 31(2), 199-218.

Rabardel, P. (1995) Les hommes et les technologies: une approche cognitive des instruments contemporains. Paris: Armand Colin.

Zöchbauer, J., & Hohenwarter, M. (2020). Developing a collaboration tool to give every student a voice in a classroom discussion. In. *Seventh ERME Topic Conference on Language in the Mathematics Classroom*. https://hal.archives-ouvertes.fr/hal-02970629/document

Dados dos autores

Astrigilda Pires Rocha Silveira

astrigilda.silveira@docente.unicv.edu.cv
Universidade de Cabo Verde
https://orcid.org/0000-0003-1725-3090

Arlindo Tavares Semedo da Veiga

arlindo.semedo@docente.unicv.edu.cv
Universidade de Cabo Verde
https://orcid.org/0000-0002-6265-9658

Alexandre Emanuel Batista Trocado

mail@alexandretrocado.com
Instituto GeoGebra Portugal
Externato Camões
https://orcid.org/0000-0001-5589-8100

Celina Aparecida Almeida Pereira Abar

abarcaap@pucsp.br
Programa de Estudos Pós-graduados em Educação Matemática
Pontifícia Universidade Católica de São Paulo, Brasil
https://orcid.org/0000-0002-6685-9956

Cristina da Silva Ferreira Alves

crismaiata@gmail.com
Agrupamento de Escolas do Castêlo da Maia
https://orcid.org/0000-0003-4806-3820

Diogo Meurer de Souza Castro

diogo.castro@ifal.edu.br
Instituto Federal de Alagoas
https://orcid.org/0000-0001-5725-2274

Idalinda Pereira da Cunha

idalindacunha@gmail.com
Agrupamento Clara de Resende
https://orcid.org/0000-0002-8913-2741

Ilda Marisa de Sá Reis

ildareis.escola@gmail.comi
Agrupamento de Escolas do Castêlo da Maia
https://orcid.org/0000-0001-6761-3264

José Manuel Dos Santos Dos Santos

santosdossantos@ese.ipp.pt
Instituto GeoGebra de Portugal
Escola Superior de Educação Instituto Politécnico do Porto
Agrupamento de Escolas do Castêlo da Maia
https://orcid.org/0000-0002-6830-6503

Marcio Vieira de Almeida

marcioalmeidasp@gmail.com
Instituto GeoGebra de São Paulo
http://lattes.cnpq.br/4591847517909670
https://orcid.org/0000-0001-8326-202X

Rui Josué Boucinha Torres Eusébio

ruieusebio1@gmail.com
Agrupamento de Escolas do Castêlo da Maia
https://orcid.org/0000-0001-6298-0436

OEI

9 789895 478910