



Uso de Tecnologias Contemporâneas de Interface com o Usuário na Educação Digital: apoiando o ensino da Geometria

Submetido em: 20/02/2019

Aprovado em: 20/04/2019

João Pedro da Silva¹
Magali R. Gouvêa Meireles²

ABSTRACT

In the didactic segment, the use of the computer can promote an interactive environment. In the study of Geometry, many are the practical applications related and many are the applications used. Applications usually have satisfactory usability for users who have intermediate knowledge in Geometry. In order to improve the user's experience in learning Geometry, providing a pleasant and intuitive environment, we propose, in this work, the implementation of an application capable of performing the recognition of geometric figures drawn manually by means of a gesture recognition method and traits. In addition to recognition, the application performs calculations related to these figures. It is expected, with this work, to contribute with the introductory practices of the teaching of Geometry, increasing the students interest, through the use of computational technology, to develop studies related to geometric and algebraic representations.

Keywords: Digital Education; Geometrical Figures; Human-Computer Interface; Recognition of Traits; Technology in Teaching.

RESUMO

No segmento didático, o uso do computador pode promover um ambiente interativo. No estudo da Geometria, muitas são as aplicações práticas relacionadas e muitos são os aplicativos utilizados. Aplicativos geralmente têm usabilidade satisfatória para usuários que possuem

¹Graduado em Sistemas de Informação na Pontifícia Universidade Católica de Minas Gerais. E-mail: joao.silva.938586@sga.pucminas.br

²Professora Adjunta da Pontifícia Universidade Católica de Minas Gerais. Doutora em Ciência da Informação pela Universidade Federal de Minas Gerais (UFMG). E-mail: magali@pucminas.br

conhecimento intermediário em Geometria. A fim de melhorar a experiência do usuário no aprendizado de Geometria, proporcionando um ambiente agradável e intuitivo, é proposto, neste trabalho, a implementação de uma aplicação capaz de realizar o reconhecimento de figuras geométricas desenhadas manualmente por meio de um método de reconhecimento de gestos e traços. Além do reconhecimento, o aplicativo realiza cálculos relacionados a essas figuras. Espera-se, com este trabalho, contribuir com as práticas introdutórias do ensino de Geometria, aumentando o interesse dos alunos, por meio do uso de tecnologia computacional, para desenvolver estudos relacionados às representações geométricas e algébricas.

Palavras-chave: Educação Digital; Figuras Geométricas; Interface Homem-Computador; Reconhecimento de traços; Tecnologia em Ensino.

1. INTRODUÇÃO

Com o crescimento e a evolução das tecnologias computacionais, *mobile* e *touch*, cresceu, também, a utilização de ferramentas computacionais para auxiliar as atividades de aprendizagem nas mais diversas áreas do conhecimento, dentro e fora das salas de aula, como é destacado por Leite [12], que, paralelamente, demonstra a necessidade de se observar os dispositivos *mobile* como uma ferramenta com grande potencial de oportunidades para melhorar a aprendizagem. Dessa maneira, a utilização de ferramentas tecnológicas para o ensino tem se difundido rapidamente, como afirmam Gladcheff et al. [11], que ressaltam a Matemática como uma das áreas que mais apresenta possibilidades de utilização da tecnologia no ensino, com aplicativos de tutoriais, exercícios, simulações e jogos.

A Geometria é um ramo da Matemática que apresenta uma vasta gama de aplicações, que estão presentes na vida cotidiana e profissional dos indivíduos. Fonseca et al. [9] ressaltam que a Geometria tem importante papel no desenvolvimento de habilidades de percepção espacial, fornecendo maneiras de comparar, medir, generalizar e abstrair. Além do caráter utilitário que ela fornece para questões práticas do dia-a-dia, a Geometria é utilizada por profissionais das áreas de engenharia, arquitetura, astronomia e até mesmo por coreógrafos, costureiros e artistas plásticos. No ensino da Geometria, observa-se que, de um modo geral, os alunos possuem um baixo desempenho, como é ressaltado por Clements [3] em suas pesquisas. Esse baixo desempenho, segundo Andrade [1], está associado com a falta de interesse e com a dificuldade de interação dos alunos com o conteúdo demonstrado em uma aula tradicional. A tecnologia pode proporcionar um ambiente mais dinâmico, prático e próximo aos alunos como alternativa ao processo de ensino e aprendizagem convencional.

Nos aplicativos utilizados no ensino da Geometria, existem diferentes possibilidades para a inserção das figuras geométricas. A proposta deste trabalho é implementar uma aplicação capaz de realizar o reconhecimento de figuras geométricas desenhadas manualmente e, desse modo, proporcionar uma abordagem dinâmica e interativa, para o usuário, na manipulação de tais figuras, provendo uma ferramenta de apoio didático para resolução de cálculos envolvendo as figuras geométricas previamente reconhecidas. A inserção de figuras geométricas em algum *software* matemático, utilizando os métodos de entrada convencionais, como *mouse*, usualmente, exigem que o usuário tenha conhecimento das ferramentas disponíveis pelo aplicativo. Parte-se da hipótese, neste trabalho, que a experiência dos iniciantes no estudo da Geometria pode ser enriquecida caso possam, simplesmente, desenhar manualmente as figuras geométricas, utilizando-se de dispositivos sensíveis ao toque e de reconhecimento de figuras.

Este artigo está organizado em mais seis seções. A segunda seção descreve o referencial teórico, apresentando-se a base teórica utilizada e os trabalhos já desenvolvidos por outros autores. A terceira seção apresenta a metodologia proposta para o trabalho. A quarta seção apresenta o desenvolvimento do trabalho. As últimas seções apresentam os resultados, as discussões e as considerações finais.

2. REFERENCIAL TEÓRICO

Segundo Martins e Silva [13], as tecnologias, em especial, o computador, se tornaram parte do cotidiano atual da sociedade, seja para desempenho de suas atividades pessoais ou para enfrentar os desafios profissionais. Acompanhando a crescente utilização dos computadores, o segmento educacional também vem buscando ampliar a utilização da computação para oferecer novas maneiras de aprendizado. Porém, em muitos casos, existe a necessidade de se estabelecer uma conexão entre os elementos tipicamente manuscritos, especialmente os do domínio da Matemática, e os computadores.

Dentre os conteúdos matemáticos a serem estudados pelos alunos durante a sua formação, estão a Geometria plana e espacial. Santa Helena [18] destaca que, de modo geral, o baixo rendimento dos alunos nesses conteúdos é proveniente do distanciamento entre o modo como os professores e os alunos compreendem tais conteúdos. Assim, a proposta de uso de tecnologia pode reduzir esse distanciamento e permitir a criação de um ambiente de aprendizagem em que

os alunos pratiquem a experimentação.

Segundo Apte et al. [2], as *interfaces* computacionais evoluíram de textual para gráfica e, nessa evolução, a tecnologia de entrada de dados mudou de digitação para movimentação do *mouse* ou caneta *stylus* e utilização de gestos, que podem ser usados para a criação de desenhos ou esboços. A partir desses desenhos, podem ser identificadas diversas informações. Assim, Corey and Hammond [4] destacam a necessidade de se realizar o reconhecimento automatizado desses desenhos para traduzi-los em informações computacionais prontas para diversos usos, como, por exemplo, a digitalização de circuitos elétricos desenhados à mão. A busca por estabelecer uma relação natural e funcional entre figuras geométricas desenhadas manualmente, sua inserção em uma aplicação computacional e a sua posterior utilização didática inspirou este trabalho de implementação de uma aplicação capaz de realizar a digitalização e o reconhecimento de figuras geométricas, além de possibilitar a resolução de cálculos associados a tais figuras para o usuário final. A fundamentação teórica do trabalho é apresentada em três seções, quais sejam, "Utilização de tecnologia no ensino", "Reconhecimento de gestos" e "Algoritmos de reconhecimento de figuras geométricas".

2.1 Utilização de Tecnologia no Ensino

De acordo com Tahir e Arif [20], as tecnologias móveis têm um grande potencial para transformar a educação, mas, para isso, elas devem ser projetadas e implementadas observando o contexto cultural e social da aprendizagem. Assim, a tecnologia se apresenta como um canal de suporte à aprendizagem, oferecendo diversos formatos para a transmissão do conhecimento.

Segundo Couse [5], as crianças que utilizam a tecnologia para a sua aprendizagem apresentam melhores habilidades linguísticas, conhecimento estrutural e habilidades para resolução de problemas, em comparação com as que não utilizam. Embora esses benefícios do uso da tecnologia não sejam um consenso geral entre os pesquisadores do âmbito da pedagogia, a maioria vê um efeito positivo e com potencial de crescimento.

Franklin [10] destaca que os dispositivos móveis estão disponíveis para apoiar os educadores no desenvolvimento de uma nova comunidade de aprendizagem para os estudantes que utilizam *tablets*, *smarthphones* e outros dispositivos. Druin [7] afirma que o entusiasmo da criança na utilização de dispositivos tecnológicos pode ser convertido em um fator motivador do

processo de ensino, atraindo-as para as disciplinas que utilizam tais dispositivos. Para Andrade [1], a Matemática sob a ótica dos estudantes é uma disciplina complicada, fazendo com que os professores encontrem dificuldades em despertar o interesse dos alunos. Assim, a tecnologia pode facilitar o ensino da Matemática, à medida em que o estudante se depara com o conteúdo de uma maneira dinâmica, em uma plataforma familiar, como é o caso dos computadores e dispositivos móveis, bastante comuns na atualidade.

Especificamente no contexto da Geometria, Reis et al. [16] constata, em seu trabalho, a dificuldade existente na utilização de *softwares* de Geometria Interativa por conta de suas *interfaces* repletas de ferramentas, em formato de ícones, dispostas em extensas barras de ferramentas. Os autores destacam, ainda, que a vasta quantidade de ícones, muitos deles semelhantes, pode se tornar um complicador para aqueles usuários que estão utilizando o *software* como ferramenta de aprendizado e deixá-los ainda com mais dúvidas sobre o conteúdo.

2.2 Reconhecimento de Gestos

De acordo com Wobbrock et al. [22], a utilização de gestos como forma de interação com computadores, *tablets* e *smartphones* se tornou bastante comum e, com isso, cresceu, também, a necessidade de explorar esses gestos para proporcionar novas maneiras de interação. Nesse contexto, Vasiliev e Mardashov [21] destacam que a utilização de gestos para proporcionar a interação com uma aplicação em dispositivos sensíveis ao toque permite melhorar a usabilidade.

Ainda de acordo com Wobbrock et al. [22], os gestos são muito naturais para os seres humanos, sendo utilizados como base para criar a escrita e o desenho. Por esse motivo, existe uma grande variedade de gestos possíveis de serem feitos, sendo que muitos deles são semelhantes, mas possuem significados diferentes, gerando uma grande ambiguidade. Dessa maneira, o reconhecimento de gestos, em muitos casos, fica a cargo das disciplinas que utilizam técnicas de inteligência artificial e de correspondência de padrões para classificação.

Segura e Barbosa [19] afirmam que, quando um usuário começa a desenhar utilizando uma caneta *stylus* ou até mesmo com seu próprio dedo em uma tela sensível ao toque, ele pode expressar sua ideia utilizando um único traço contínuo, ou vários traços, de maneiras diferentes, dependendo de cada usuário. Essas especificidades de gestos e, por consequência, de desenhos, é que tornam o reconhecimento complexo. Para que o reconhecimento atinja a maior taxa de acerto

possível, uma etapa presente na maioria das técnicas de reconhecimento é o pré-processamento, que, segundo Oliveira [15], tem por objetivo minimizar as inconsistências no traçado, advindas de ruídos do *hardware* utilizado e das diferenças no traçado das pessoas, buscando uma uniformização. Nesse sentido, muitos trabalhos se propõem, exclusivamente, a desenvolver abordagens que aprimorem esse pré-processamento.

No contexto de usabilidade, Nacenta et al. [14] demonstram que os gestos são bastante importantes para que seja possível reduzir a quantidade de *menus*, barras de ferramentas e ícones nos aplicativos. Além de simplificar a *interface*, os gestos proporcionam uma melhor experiência de uso para aqueles que utilizam o *software* em dispositivos *touchscreen*.

2.3 Algoritmos de reconhecimento de figuras geométricas

De acordo com Fonseca et al. [9], os algoritmos que realizam o reconhecimento de figuras geométricas desenhadas manualmente seguem um fluxo baseado em três principais passos. O primeiro deles é a extração das propriedades geométricas das figuras desenhadas, que são importantes para caracterizar as figuras geométricas. O segundo passo é a aplicação de filtros para melhorar o desempenho do reconhecimento e, propriamente, para realizar o reconhecimento das formas geométricas. Por fim, o terceiro passo é a associação de graus de certeza às formas geométricas reconhecidas, como por exemplo, com a utilização de lógica *Fuzzy*.

Apte et al. [2] explicitam três dos algoritmos de filtros (*Area-Ratio*, *Triangle-Diamond* e *p/A Ratio*) para realizar o reconhecimento das formas geométricas. Esses filtros são utilizados de acordo com o sucesso ou a falha do anterior e o resultado final do reconhecimento é dado a partir da combinação de resultados obtidos com a aplicação de cada filtro.

Outro tipo de algoritmo utilizado para realizar o reconhecimento de figuras geométricas desenhadas tem sua estrutura detalhada por Corey and Hammond [4]. Nesse algoritmo, cada traço do desenho é submetido a um teste que determina se esse traço pode ser interpretado como uma forma primitiva e, então, cria uma forma primitiva ajustada. Ao término do desenho, essas primitivas ajustadas geradas são ordenadas, utilizando-se um classificador hierárquico que determina qual é a forma geométrica desenhada.

Segundo Das et al. [6], um importante algoritmo geométrico frequentemente utilizado como base para desenvolver outras soluções de Geometria Interativa, principalmente, em

processos de reconhecimento, é chamado de *Convex hull*. Ainda de acordo com Das et al. [6], o algoritmo fornece um conjunto de pontos com o menor espaço convexo. O resultado da execução desse algoritmo é denominado de envoltório externo, pois engloba, em um menor polígono convexo possível, todos os pontos inicialmente utilizados como entrada.

Muitos dos algoritmos utilizados nos tipos de reconhecimento citados estão disponíveis por meio de *Frameworks*, que segundo Riehle [17], são conjuntos de classes que incorporam soluções abstratas para determinados problemas já resolvidos por outros desenvolvedores. Portanto, seu uso proporciona maior produtividade por meio da reutilização de códigos que são disponibilizados, geralmente, em bibliotecas locais, importadas pelo desenvolvedor.

3. METODOLOGIA

A metodologia utilizada no desenvolvimento deste trabalho está dividida em três principais etapas: seleção dos *frameworks* utilizados no reconhecimento das figuras geométricas desenhadas manualmente e dos valores associados às figuras geométricas; avaliação do funcionamento do conjunto dos *frameworks* selecionados e implementação da aplicação para uso do usuário final.

3.1 Seleção dos *Frameworks*

A primeira etapa consistiu em identificar os possíveis algoritmos disponíveis para realizar o reconhecimento de elementos manuscritos e selecionar o *framework* que melhor se adequasse ao reconhecimento de figuras geométricas ou de elementos que as compõem, para, então, criar uma solução própria com base nos *frameworks* encontrados. Ao identificar essas ferramentas, foi necessário verificar os tipos de elementos reconhecidos por elas para estabelecer o escopo do reconhecimento.

Os *frameworks* impõem a quem os implementam um conjunto de pré-requisitos, principalmente, quanto à entrada dos dados. Portanto, foi preciso realizar uma análise das possíveis adaptações, melhorias e pré-processamentos necessários para que os *frameworks* pudessem ser utilizados e atingissem o resultado esperado.

3.1.1 Avaliação do Conjunto de *Frameworks*

Foi necessário avaliar o funcionamento do conjunto dos *frameworks* e, então, delimitar o escopo de formas geométricas que seriam suportadas pela aplicação a ser desenvolvida. Para avaliar o funcionamento desse conjunto, fez-se necessário o desenvolvimento de um protótipo.

3.2 Implementação da Aplicação para o Usuário Final

A implementação do *software* proposto se inicia com a especificação detalhada do funcionamento de suas duas grandes vertentes: o reconhecimento das figuras geométricas e a sua consequente utilização no contexto didático. A especificação contempla questões arquiteturais da construção do *software* e, também, de *layout* das telas. Assim, a implementação reflete essa divisão em duas etapas. Por tanto, a implementação dos *frameworks* escolhidos foi feita, primeiramente, constituindo o *back-end* da aplicação, responsável pelo processamento principal e, posteriormente, a implementação da *interface* gráfica foi realizada, constituindo o *front-end* da aplicação, responsável por permitir a interação com o usuário.

A especificação levou em consideração a avaliação de aplicativos já existentes, para que os pontos positivos de cada um deles fossem identificados e utilizados como referência para o aplicativo desenvolvido na aplicação proposta neste trabalho. A escolha da plataforma iOS se justifica pelo fato de que é, frequentemente, utilizada no contexto educacional, principalmente, em sala de aula, por apresentar ferramentas disponíveis para que alunos e professores possam interagir entre si de maneira dinâmica com os *tablets*, lousa interativa, entre outros dispositivos suportados pela plataforma.

4. DESENVOLVIMENTO

O desenvolvimento do trabalho está apresentado, nesta seção, de acordo com as etapas metodológicas da seção anterior.

4.1 Identificação dos *Frameworks*

O critério utilizado, inicialmente, para identificar os *frameworks* disponíveis para a aplicação foi bastante abrangente, com o objetivo de obter o maior número possível de ferramentas e, posteriormente, avaliar sua adequação. Em um segundo momento, esses *frameworks* foram submetidos a uma análise mais detalhada, que envolvia critérios mais específicos, como o algoritmo utilizado como base, suas limitações e as linguagens de programação possíveis de implementá-los.

Observando os dados obtidos por meio das pesquisas, os *frameworks* *\$P Point-Cloud Recognizer* e *MyScript Math*, inicialmente, são os mais adequados para o contexto deste trabalho. O *framework* *\$P Point-Cloud Recognizer* é uma evolução dos *frameworks* *\$1 Unistroke Recognizer* e *\$N Multistroke Recognizer*, apresentando melhor performance frente aos seus antecessores e está disponível gratuitamente no modelo *open-source* para implementações em Java e Objective-C. Portanto, oferece suporte para as plataformas *mobile* Android e iOS, representando considerável vantagem frente ao *framework* *Image-Based Recognizer Using Directional Bitmap Template*, que, para utilização neste trabalho, precisaria ser reescrito na linguagem *Objective-C*. Além disso, sua estratégia de reconhecimento não leva em consideração o fluxo utilizado pelo usuário para realizar o desenho, melhorando seu desempenho frente aos demais. O *MyScript Math* oferece uma vasta documentação, possui um desempenho de reconhecimento melhor frente aos demais e oferece suporte para as plataformas *mobile* Android, iOS e Windows Phone. Dessa maneira, eles foram escolhidos para a implementação de um protótipo em iOS, para verificação das necessidades de pré-processamento e para delimitação do escopo das figuras geométricas que seriam suportadas. É importante ressaltar que a implementação poderia ser realizada em Android pois os *frameworks* escolhidos também suportam essa plataforma.

4.2 Avaliação de Funcionamento dos *Frameworks*

A premissa do desenvolvimento de um protótipo com os *frameworks* selecionados advém da necessidade de perceber de maneira prática como funcionam. Diante disso, foi desenvolvido um protótipo em iOS com as funções essenciais e necessárias para realizar essa análise,

descartando questões estéticas da aplicação. Com a implementação do protótipo, foi possível definir o escopo das formas geométricas suportadas. O *framework \$P Point-Cloud Recognizer* utiliza *templates* pré-definidos de desenhos que devem ser reconhecidos. Portanto, no contexto do reconhecimento de figuras geométricas, é necessário que sejam criados *templates* contendo os vértices que compõem as figuras geométricas.

Para auxiliar a criação dos *templates*, foi utilizado o *software Shape Editor*, que fornece um arquivo no formato JSON com os vértices que compõem a figura geométrica desenhada previamente. Inicialmente, foram escolhidos, para o protótipo, o quadrado e o triângulo. Com as figuras geométricas criadas e exportadas pelo *Shape Editor*, os *templates* foram criados por meio da classe responsável por armazenar os modelos pré-definidos utilizados para realizar o reconhecimento.

Aliado ao reconhecimento das figuras geométricas, um importante componente a ser validado era o reconhecimento dos valores que seriam associados aos componentes da figura geométrica. Para isso, o *framework* selecionado (*MyScript Math*) também foi implementado, de modo que fosse possível grafar os valores de maneira manuscrita no protótipo e avaliar se o reconhecimento era eficaz. A primeira versão do protótipo foi submetida a testes para verificar a eficácia dos dois tipos de reconhecimentos descritos. Para isso, foram criados diversos desenhos de quadrados e triângulos de diferentes maneiras, tamanhos, com um ou múltiplos traços e, posteriormente, inseridos, de maneira manuscrita, diversos valores, com números inteiros, decimais, positivos e negativos.

O *framework MyScript Math* se mostrou bastante eficaz no reconhecimento dos valores, sendo necessário desenvolver, posteriormente, uma maneira de criar um vínculo entre o valor inserido e cada componente da figura geométrica. Já o reconhecimento das figuras geométricas não foi tão eficaz com relação ao reconhecimento dos quadrados desenhados, mas se mostrou eficaz com relação ao reconhecimento dos triângulos. Como tentativa de melhorar a taxa de reconhecimento, especificamente dos quadrados, um novo *template* foi criado para essa figura geométrica e os testes, refeitos, atingindo um melhor desempenho e validando a escolha do *framework*.

Durante a verificação da eficácia dos *frameworks*, foi constatado que as figuras geométricas inseridas pelo usuário deveriam ser redesenhadas para corrigir imperfeições no traçado e representar de maneira mais precisa as figuras, visando melhorar a usabilidade e

facilitar a associação dos valores a cada um dos elementos das figuras geométricas. Portanto, uma pesquisa por componentes que pudessem auxiliar esse processo de redesenho foi realizada, concluindo-se que o *framework MyScript* em sua versão *Graph* conseguia suprir essa necessidade, além de possuir suporte às plataformas iOS, Android e Windows Phone.

Outra percepção advinda da elaboração desse protótipo foi a necessidade de nomear os vértices das figuras reconhecidas, para compor integralmente a representação das figuras geométricas e, também, propiciar que fossem utilizados, posteriormente, como referência para a resolução dos cálculos desejados. Dessa maneira, uma pesquisa por componentes que pudessem auxiliar esse processo foi realizada. Como nenhuma biblioteca ou *framework* que resolvesse integralmente ou parcialmente essa demanda foi encontrada, uma solução própria com base em algoritmos geométricos foi implementada, criando uma heurística capaz de ser aplicada em outros trabalhos e contextos em que seja necessário percorrer pontos que compõem uma figura geométrica em busca de seus vértices.

4.3 Implementação da Aplicação

Com os *frameworks* selecionados e avaliados, as necessidades de pré-processamento, adaptações e melhorias listadas e o *wireframe* produzido, a arquitetura da aplicação pôde ser definida com o intuito de representar logicamente o seu desenvolvimento. O diagrama de componentes apresentado na Figura 1 representa a macro estrutura da aplicação.

Fica a cargo do componente de reconhecimento da figura geométrica o processo de receber um conjunto de pontos que compõe os traços inseridos pelo usuário e reconhecer a figura desenhada por meio do *framework \$P Point-Cloud Recognizer*, utilizando os modelos fornecidos. Diante da eficácia observada durante a validação do *framework* na fase de prototipação, o escopo das formas geométricas foi ampliado, agora, contemplando as seguintes figuras: círculo, losango, quadrado, retângulo, trapézio e triângulo.

A figura reconhecida é repassada ao componente de redesenho para que o traçado do usuário, contendo imperfeições, seja substituído por um traçado mais representativo e preciso da figura. O componente de redesenho da figura é responsável também por identificar os lados e adicionar os vértices da figura com o auxílio do componente de utilitários que fornece métodos de uso comum de outros componentes.

A figura já redesenhada e com seus elementos identificados e atribuídos é repassada aos componentes de resolução de cálculos ou de inserção/edição dos valores, de acordo com a opção escolhida pelo usuário. Os componentes interagem entre si, uma vez que são responsáveis por processos que podem acontecer concomitantemente. Já o componente de utilitários, por sua vez, fornece métodos de uso comum de outros componentes, como o redesenho da figura e a inserção/edição dos valores.

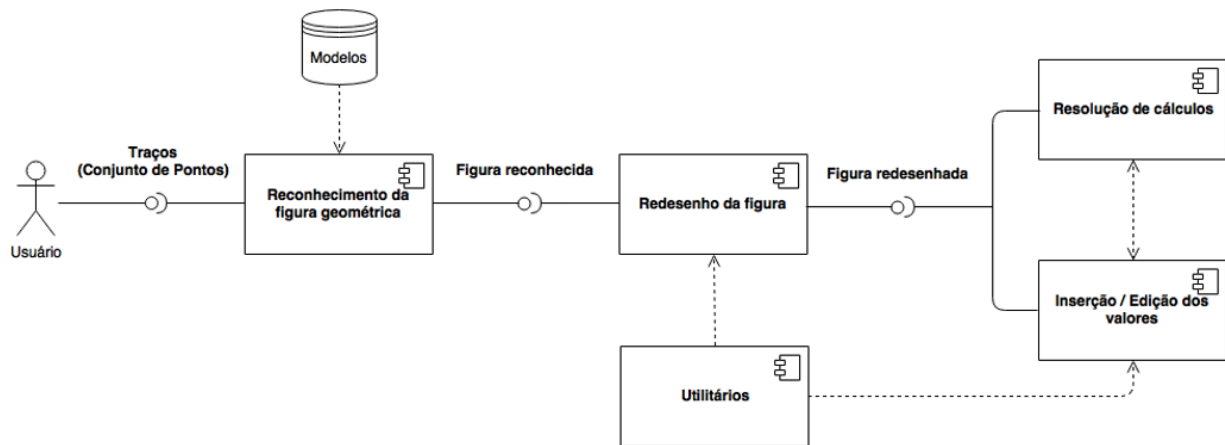


Figura 1. Diagrama de componentes.

O padrão arquitetural escolhido para o desenvolvimento da aplicação foi o *Model-View-Controller (MVC)*, padrão que já é adotado para a utilização de componentes do próprio iOS, plataforma em que o aplicativo foi desenvolvido. Os principais *models* utilizados no desenvolvimento da aplicação estão representados por meio de um diagrama de classes na Figura 2.

Todas as classes que representam as figuras geométricas suportadas são especializações da classe *BaseGeometric*, pois compartilham muitos de seus atributos e operações. O mesmo acontece com as classes *Vertex* e *Segment*, que são *models* herdados de *Element*, que possuem os atributos e operações comuns.

O processo de desenvolvimento da aplicação se iniciou pelo *back-end*, implementando os *frameworks* \$P Point-Cloud Recognizer, MyScript Math e MyScript Graph. Em seguida, foram criados os *templates* das figuras geométricas que não haviam sido criados ainda (círculo, losango, retângulo e trapézio), utilizando a ferramenta *Shape Editor*. Posteriormente, foram transpostos para a classe responsável por armazená-los no *framework* \$P Point-Cloud Recognizer.

Na sequência, o *front-end*, que diz respeito à tela de desenho da figura geométrica e inserção dos valores, foi implementado seguindo o *wireframe* produzido anteriormente. Em seguida, por meio de sua *controller*, foram associados os *frameworks* *\$P Point-Cloud Recognizer*, *MyScript Graph* e *MyScript Math*, já implementados anteriormente.

Neste momento, a figura geométrica já podia ser desenhada, reconhecida e sua representação otimizada por meio do seu redesenho, mas os vértices ainda não eram apresentados. Para isso, uma solução heurística foi desenvolvida para nomear e adicionar os vértices às figuras a partir do algoritmo *Convex hull*, que recebe um conjunto de pontos e fornece aqueles pontos que compõem um envoltório externo, ou seja, que juntos englobam todos os demais pontos. Dessa maneira, o algoritmo *Convex hull* foi utilizado com todos os pontos que compõem a figura desenhada e, a partir do seu resultado, esses pontos do envoltório externo foram ordenados e selecionados de acordo com as características de cada uma das figuras suportadas, como a provável posição de cada um dos vértices e quantidade de vértices.

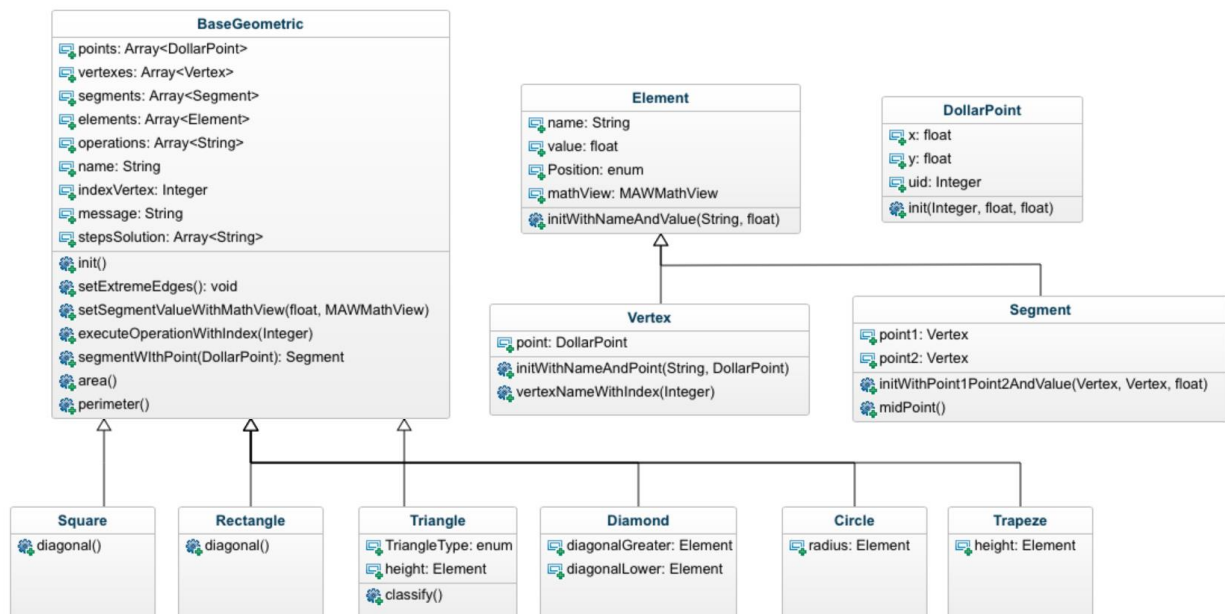


Figura 2. Diagrama de classes.

Para finalizar as implementações relativas à tela de desenho da figura geométrica, faltava, apenas, a associação dos valores reconhecidos pelo *framework* *MyScript Math* com cada elemento da figura geométrica. Para isso, outra solução heurística foi desenvolvida, com base no

cálculo da distância euclidiana. A interação proposta para a inserção dos valores previa o toque do usuário sobre os lados da figura geométrica e, então, o valor desejado seria inserido de maneira manuscrita. Portanto, para que essa interação fosse possível, o algoritmo desenvolvido capturava o toque do usuário na tela como um par ordenado de coordenadas (x,y) e, assim, era realizado o cálculo da distância euclidiana entre esse ponto capturado e todos os vértices da figura, encontrados anteriormente, em busca das duas menores distâncias. Por consequência, era descoberto o lado da figura que deveria receber o valor a ser inserido pelo usuário.

As implementações, agora, lidavam apenas com a segunda tela do aplicativo, composta pelo espaço para resolução dos cálculos desejados e a edição de valores, caso necessário. Essa etapa se iniciou com a implementação do *front-end*, seguindo o *layout* proposto no *wireframe*. Em seguida, por meio de sua *controller*, foram associadas as opções de cálculo disponíveis para cada figura geométrica, as resoluções passo-a-passo, além da opção de edição dos valores, reutilizando o *framework MyScript Math* já implementado anteriormente.

5. APRESENTAÇÃO DAS TELAS DO APLICATIVO

As telas da aplicação são apresentadas na Figura 3. À esquerda, é mostrada a primeira tela do aplicativo, referente à área de desenho, em que o usuário pode desenhar a figura geométrica e inserir os valores para os segmentos da figura. À direita, a segunda tela do aplicativo apresenta a resolução detalhada dos cálculos solicitados pelo usuário. Também é possível editar os valores associados, anteriormente, na primeira tela.

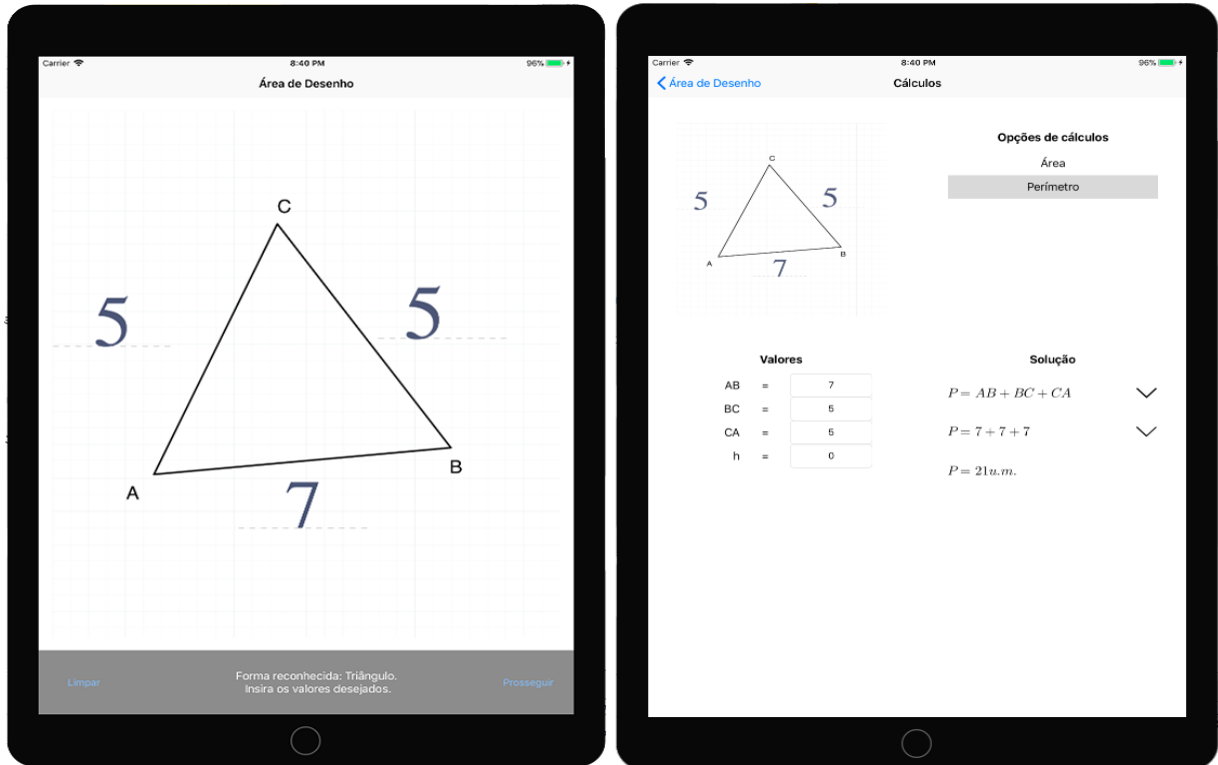


Figura 3. Telas do aplicativo.

6. CONSIDERAÇÕES FINAIS

Este trabalho propôs a implementação de um aplicativo capaz de reconhecer figuras geométricas desenhadas, assim como os valores informados, além de apresentar a solução detalhada dos cálculos associados a essas figuras para suprir a necessidade por *interfaces* mais simples e intuitivas para aplicações de Geometria Interativa no contexto didático. A partir da identificação dos *frameworks* disponíveis, foram avaliados os recursos necessários para o desenvolvimento desse aplicativo, definidas a estrutura e a arquitetura e desenvolvida a aplicação para *tablets* iOS. O objetivo era suprir a necessidade por *interfaces* mais simples e intuitivas, utilizando tecnologias contemporâneas de *interface* com o usuário. Espera-se, com a implementação da aplicação, tornar a experiência do usuário no aprendizado de geometria mais agradável, aproximando a maneira como aluno e professor compreendem o conteúdo em questão. Acompanhando a crescente utilização dos computadores, as tecnologias móveis têm um grande potencial para criar uma alternativa aos tradicionais métodos de ensino.

Existem, ainda, melhorias e aprimoramentos a serem implementados no aplicativo. Dessa maneira, são listados, como trabalhos futuros, a ampliação do escopo de figuras geométricas reconhecidas, o aprimoramento do processo de reconhecimento das figuras e a realização de testes de usabilidade com alunos iniciantes no estudo de Geometria.

REFERÊNCIAS

Raoni Aguiar Andrade. 2012. **Geogebra: Uma ferramenta computacional para o ensino de geometria no ensino fundamental 2** [Licenciatura em Matemática]. Vitória da Conquista.

Ajay Apte, Van Vo, e Takayuki Dan Kimura. 1993. Recognizing multistroke geometric shapes: an experimental evaluation. In *Proceedings of the 6th annual ACM symposium on user interface software and technology*, 121–128. DOI: <https://doi.org/10.1145/168642.168654>

Clements, Douglas H. 2003. Teaching and learning geometry. **A research companion to principles and standards for school mathematics**, 151–178.

Paul Corey e Tracy Hammond. 2008. Gladder: Combining gesture and geometric sketch recognition. In *Proceedings of the 23th AAAI Conference on Artificial Intelligence*, 1788–1789. Acesso em 22 de setembro de 2017. Disponível em: <http://www.aaai.org/Papers/AAAI/2008/AAAI08-284.pdf>

Leslie Couse e Dora W. Chen. 2010. A tablet computer for young children? exploring its viability for early childhood education. *Journal of Research on Technology in Education* 43, 1: 75–96. Acesso em: 30 de outubro de 2017. Disponível em: <https://files.eric.ed.gov/fulltext/EJ898529.pdf> Jofish Kaye and Paul Dourish. 2014. Special issue on science fiction and ubiquitous computing. *Personal Ubiquitous Comput.* 18, 4 (April 2014), 765–766. <http://dx.doi.org/10.1007/s00779-014-0773-4>

Nibaran Das, Sandip Pramanik, Subhadip Basu, Punam Kumar Saha, Ram Sarkar, Mahantapas Kundu, e Mita Nasipuri. 2014. Recognition of handwritten bangla basic characters and digits using convex hull based feature set. *arXiv preprint arXiv:1410.0478*. DOI: <https://arxiv.org/ct?url=http%3A%2F%2Fdx.doi.org%2F10%252E13140%2F2%252E1%252E3689%252E4089&v=bfee1e35>

Druin, Allisson. 2009. *Mobile technology for children: Designing for interaction and learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. Marilyn Schwartz. 1995. *Guidelines for Bias-Free Writing*. Indiana University Press.

Manuel J. Fonseca e Joaquim A. Jorge. 2000. Using fuzzy logic to recognize geometric shapes interactively. In *Ninth IEEE International Conference on Fuzzy Systems*, 291–296. DOI: 10.1109/FUZZY.2000.838674

Maria da Conceição F. R. Fonseca, Maria da Penha Lopes, Maria das Graças Gomes Barbosa, Maria Laura Magalhães Gomes e Mônica Maria Machado S. S. Dayrell. 2002. **O ensino de geometria na escola fundamental: três questões para a formação do professor dos ciclos iniciais**. Autêntica Editora.

Teresa Franklin. 2011. Mobile learning: At the tipping point. *The Turkish Online Journal of Educational Technology* 10, 4: 261-275. Acesso em: 30 de outubro de 2017. Disponível em: <https://files.eric.ed.gov/fulltext/EJ946635.pdf>

Ana Paula Gladcheff, Vera Barros de Oliveira, e Dilma Menezes da Silva. 2012. The Educational Software and the Psychopedagogy in Teaching Mathematics Geared to elementary school. *Brazilian Journal of Computers in Education* 8, 1: 63–70. DOI: <http://dx.doi.org/10.5753/rbie.2001.8.1.63-70>

Bruno Silva Leite. 2014. M-learning: the use of mobile devices as a didactic tool in the teaching of chemistry. *Brazilian Journal of Computers in Education* 22, 03: 55-68. DOI: <http://dx.doi.org/10.5753/rbie.2014.22.03.55>

Beatriz Souza Martins e Flávia Damacena Souza Silva. 2015. In *III Congresso de Educação*. Disponível em: <http://www.anais.ueg.br/index.php/congressoeducacaoipora/article/viewFile/4341/2520>. Acesso em 30 de outubro de 2017.

Miguel A. Nacenta, Yemliha Kamber, Yizhou Qiang, and Per Ola Kristensson. 2013. Memorability of pre-designed and user-defined gesture sets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* 1099–1108. DOI: <https://doi.org/10.1145/2470654.2466142>

Marcelo Valentim de Oliveira. 2014. **Um estudo empírico sobre classificação de símbolos matemáticos manuscritos**. Dissertação (Mestrado em Ciência da Computação). Universidade de São Paulo, Instituto de Matemática e Estatística, São Paulo.

Helena M. Reis, Seiji Isotani, Leônidas O. Brandao, Ramilio R. R. Filho, Wilmax M. Cruz, e Anarosa A. F. Brandão. 2015. Concepção de uma Família de Gestos para Construção de Objetos Geométricos e sua Utilização em um Sistema de Geometria Interativa para Dispositivos Móveis: GeoTouch. *Revista Brasileira de Informática na Educação*, 23, 2: 206–224. DOI: [10.5753/RBIE.2015.23.02.206](https://doi.org/10.5753/RBIE.2015.23.02.206)

Dirk Riehle. 2000. **Framework design: A role modeling approach**. Dissertação (Doutorado em Ciências Técnicas). Universität Hamburg, Swiss Federal Institute of Technology Zurich, Zurich.

Rainer Fischer Santa Helena, R. F. 2015. **Uma proposta para o ensino de geometria na educação de jovens e adultos com o uso de mídias digitais**. Acesso em: 22 de setembro de 2017. Disponível em: <http://www.lume.ufrgs.br/bitstream/handle/10183/134455/000985934.pdf?seque>

Vinícius C. V. Segura e Simone D. J. Barbosa. 2012. A combination of stroke manipulation and recognition strategies to support user interface construction and interactive behavior definition through sketching. In *2012 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 45-48. DOI: 10.1109/VLHCC.2012.6344479

Rabail Tahir e Fahim Arif. 2015. Mobile technology in children education: Analyzing parents' attitude towards mobile technology for children. In *2015 Science and Information Conference (SAI)*, 410-420. DOI: 10.1109/SAI.2015.7237175

Bogdan U. Vasiliev e Dmitrii V. Mardashov. 2017. Methods and tools for education and research in the information society. In *2017 International Conference Quality Management, Transport and Information Security, Information Technologies (IT&QM&IS)*, 699-703. DOI: 10.1109/ITMQIS.2017.8085920

Jacob O. Wobbrock, Andrew D. Wilson, e Yang Li. 2007. Gestures without libraries, toolkits or training: a \$1 recognizer for user interface prototypes. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*, 159-168. DOI: <https://doi.org/10.1145/1294211.1294238>