

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE SISTEMAS DE INFORMAÇÃO – BACHARELADO

APLICATIVO ANDROID DE CADERNO ORTOGRÁFICO
UTILIZANDO RECONHECIMENTO DE ESCRITA

GABRIELA BONIFÁCIO DE SENA

BLUMENAU
2019

GABRIELA BONIFÁCIO DE SENA

APLICATIVO ANDROID DE CADERNO ORTOGRÁFICO

UTILIZANDO RECONHECIMENTO DE ESCRITA

Trabalho de Conclusão de Curso apresentado ao curso de graduação em Sistemas de Informação do Centro de Ciências Exatas e Naturais da Universidade Regional de Blumenau como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informação.

Prof(a). Dalton Solano dos Reis, Mestre - Orientador

**BLUMENAU
2019**

**APLICATIVO ANDROID DE CADERNO ORTOGRÁFICO
UTILIZANDO RECONHECIMENTO DE ESCRITA**

Por


GABRIELA BONIFÁCIO DE SENA

Trabalho de Conclusão de Curso aprovado
para obtenção dos créditos na disciplina de
Trabalho de Conclusão de Curso II pela banca
examinadora formada por:

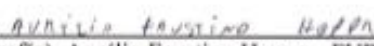
Presidente:


Prof(a). Dalton Sotero dos Reis – Orientador, FURB

Membro:


Prof(a). Andreza Sartori – FURB

Membro:


Prof(a). Aurélio Faustino Hoppe – FURB

Blumenau, 12 de julho de 2019

Dedico este trabalho para minha mãe Janice
Lourdes Peters.

AGRADECIMENTOS

Ao meu orientador, Mestre Dalton Solano dos Reis pela paciência, auxílio e empenho dedicado a colaboração e conclusão deste trabalho.

À minha família e noivo, que sempre acreditaram em mim e me incentivaram todas as vezes que precisei.

A todos os professores do curso de Sistemas de Informação da FURB, com os quais aprendi ao longo da minha jornada acadêmica.

Seu trabalho vai preencher uma parte grande da sua vida, e a única maneira de ficar realmente satisfeito é fazer o que você acredita ser um ótimo trabalho. E a única maneira de fazer um excelente trabalho é amar o que você faz.

Steve Jobs

RESUMO

Este trabalho apresenta a especificação e o desenvolvimento de um aplicativo para a plataforma Android que auxilia no processo de alfabetização com o reconhecimento de caracteres a partir de desenhos realizados na tela do dispositivo móvel. O usuário pode selecionar qual caractere quer treinar e em seguida o aplicativo disponibiliza uma área de treinamento com o caractere, uma imagem e uma palavra referente ao caractere e um espaço para desenho. Após o desenho finalizado o aplicativo faz o reconhecimento do caractere com o Tesseract e apresenta o resultado em tela. Houve bons resultados, vários caracteres foram reconhecidos, porém também houve pontos que devem ser melhorados no reconhecimento do caractere como identificar caracteres com pequenas diferenças em rotações e tamanhos. Por fim foram sugeridas algumas extensões, como utilizar o processamento de imagem, que busca melhorar e expandir as funcionalidades de reconhecimento de caractere no aplicativo.

Palavras-chave: Tesseract. Alfabetização. Android, OCR.

ABSTRACT

This work presents the specification and development of an application for the Android platform that assists in the literacy process with the recognition of characters from drawings made on the screen of the mobile device. The user can select which character to train and then the application provides a training area with the character, an image and a word referring to the character and a space for drawing. After the finished drawing, the application recognizes the character with the Tesseract and displays the result on screen. There were good results, several characters were recognized, but there were also points that should be improved in character recognition as identifying characters with small differences in rotations and sizes. Finally, some additions have been suggested, such as using image processing, which seeks to improve and expand character recognition capabilities in the application.

Key-words: Tesseract. Literacy. Android, OCR.

LISTA DE FIGURAS

Figura 1 - Interface do software	17
Figura 2 - Telas de leitura do cartão	18
Figura 3 - a) Tela de sons e letras b) Tela de imagens e palavras	19
Figura 4 - a) Tela de quantificação de letras b) Tela de quantificação de sílabas	19
Figura 5 - Diagrama de caso de uso	21
Figura 6 - Diagrama de classes.....	22
Figura 7 - Tela inicial	26
Figura 8 - Opção Assertividade	26
Figura 9 - a) Letras maiúsculas b) Letras minúsculas c) Números	27
Figura 10 - Tela de treinamento	28
Figura 11 - Feedback de sucesso	29
Figura 12 - Feedback de erro	30
Figura 13 - Imagem ajustada	30
Figura 14 - Resultado A - J das letras maiúsculas	38
Figura 15 - Resultado K - Z das letras maiúsculas	39
Figura 16 - Resultado A - P das letras minúsculas	40
Figura 17 - Resultado Q - Z das letras minúsculas.....	41
Figura 18 - Resultado dos números	42

LISTA DE QUADROS

Quadro 1 - Método que processa o reconhecimento de caractere com Tesseract	23
Quadro 2 - Método que processa o desenho.....	25
Quadro 3 - Comparativo entre os trabalhos correlatos	32
Quadro 4 - UC01 Ajustar assertividade.....	36
Quadro 5 - UC02 – Selecionar treinamento	36
Quadro 6 - UC03 – Executar treinamento	37
Quadro 7 - UC04 – Consultar resultado do desenho	37

LISTA DE TABELAS

Tabela 1 - Consolidação dos resultados	31
Tabela 2 - Resultados letras maiúsculas	31
Tabela 3 - Resultados letras minúsculas.....	31
Tabela 4 - Resultados números.....	31

LISTA DE ABREVIATURAS E SIGLAS

RF – Requisito Funcional

RNF – Requisito Não-Funcional

OCR – Optical Character Recognition

SUMÁRIO

1 INTRODUÇÃO.....	13
1.1 OBJETIVOS.....	13
1.2 ESTRUTURA.....	14
2 FUNDAMENTAÇÃO TEÓRICA	15
2.1 ALFABETIZAÇÃO	15
2.2 RECONHECIMENTO DE CARACTERES.....	15
2.3 TESSERACT.....	16
2.4 TRABALHOS CORRELATOS	16
2.4.1 Software de auxílio à pré-alfabetização infantil (Aprendendo as letras)	16
2.4.2 CardTess.....	17
2.4.3 Alfabetize	18
3 DESENVOLVIMENTO	20
3.1 LEVANTAMENTO DE INFORMAÇÕES	20
3.2 ESPECIFICAÇÃO	20
3.2.1 Diagrama de casos de uso	20
3.2.2 Diagrama de classes	21
3.3 IMPLEMENTAÇÃO	22
3.3.1 Técnicas e ferramentas utilizadas.....	23
3.3.2 Operacionalidade da implementação	26
3.4 RESULTADOS E DISCUSSÕES.....	30
3.4.1 Análise dos resultados.....	31
3.4.2 Comparativo entre o aplicativo desenvolvido e seus correlatos	32
4 CONCLUSÕES	33
4.1 EXTENSÕES	33
REFERÊNCIAS	34
APÊNDICE A – DESCRIÇÃO DOS CASOS DE USO	36
APÊNDICE B – IMAGENS DA BATERIA DE TESTES DO APLICATIVO	38

1 INTRODUÇÃO

Com a tecnologia é possível investir em estudos para a aplicação de tecnologias de informação em ambientes educacionais, assim como em estudos de como introduzir a tecnologia e seus benefícios na fase de alfabetização das crianças. Conforme Andrade (2011, p.7) “O desenvolvimento cognitivo do ser humano está sendo mediado por dispositivos tecnológicos, onde as novas tecnologias da informação e comunicação estão ampliando o potencial humano.”.

Alfabetizar tem por objetivo o domínio do sistema alfabético e ortográfico (KLEIMAN, 2005), e conforme Cruz e Albuquerque (2012, p.13), “é uma tarefa complexa, mas é possível ensinar e aprender a ler e a escrever por meio de brincadeiras que estimulem a reflexão sobre o Sistema de Escrita Alfabética (SEA).”. Para alfabetizar é possível utilizar inúmeros recursos como exercícios repetitivos de codificação e decodificação da língua escrita (BARBOSA et al., 2017), brincadeiras, quadro-de-giz, ilustrações, livros didáticos e quaisquer outros recursos pedagógicos (KLEIMAN, 2005).

Uma pesquisa realizada pela Avazu Inc (2016) revela que mais de sessenta e quatro milhões de brasileiros possuem celular e que o sistema operacional Android é utilizado em noventa por cento dos dispositivos móveis no Brasil. Em maio de 2017 a empresa Google anunciou o marco de mais de dois bilhões de dispositivos Android ativos no mundo (BURKE, 2017). O sistema operacional não é utilizado somente em celular, mas também em relógios, carros, televisões, chromebooks e na internet das coisas. Com esses números esse sistema operacional é um dos recursos tecnológicos de dispositivos móveis com maior alcance no Brasil e no mundo.

Diante do exposto, esse trabalho apresenta o desenvolvimento de um aplicativo Android que visa auxiliar crianças no período de alfabetização a treinarem a grafia das letras. Essa ferramenta permite que a criança visualize uma letra e submeta a uma avaliação de reconhecimento de caractere um desenho realizado na tela do dispositivo móvel.

1.1 OBJETIVOS

O objetivo principal deste trabalho é disponibilizar um aplicativo Android que simule um caderno ortográfico.

Os objetivos específicos são:

- a) disponibilizar uma área para desenhar uma letra ou número;
- b) reconhecer um caractere a partir do desenho realizado em um dispositivo móvel;
- c) disponibilizar um percentual de assertividade comparando o desenho e o caractere.

1.2 ESTRUTURA

Este trabalho está dividido em quatro capítulos: introdução, fundamentação teórica, desenvolvimento e conclusões. O segundo capítulo apresenta a fundamentação teórica, onde será abordada a alfabetização e o reconhecimento de caracteres. No terceiro capítulo é demonstrada a especificação do aplicativo e o detalhamento da implementação. Por fim, no quarto capítulo são descritas as conclusões, os resultados obtidos e algumas extensões.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo está organizado em cinco seções. A seção 2.1 detalha que é a Alfabetização. A seção 2.2 apresenta conceitos de reconhecimento de caracteres. Na seção 2.3 é descrita a biblioteca Tesseract. Na seção 2.4 são listados três trabalhos correlatos.

2.1 ALFABETIZAÇÃO

Alfabetizar, segundo Kleiman (2005, p.12), é uma prática. Kleiman (2005, p.13) explica também que uma prática consiste em atividades com um objetivo e essas atividades podem precisar de tecnologias (lápiz e papel, as diferentes mídias), habilidades especiais e saberes.

Mendonça e Mendonça (2011 apud GROSSI, 1985, p.15) argumentam a importância do reconhecimento das letras e a divide em dois grandes trilhos paralelos. Um é o reconhecimento das letras como um dos papéis principais na escrita e o outro trilho é a compreensão ampla da vinculação do discurso oral com o texto escrito. A didática do nível pré-silábico visa que a criança distinga imagem de texto e letras de número (MENDONÇA; MENDONÇA, 2011 apud GROSSI, 1985, p.15).

A prática de alfabetização se concretiza em eventos que se situam dentro da sala de aula, liderados por um especialista (o professor) que se encarrega de ensinar sistemicamente as regras de funcionamento e uso do código alfabético aos iniciantes no assunto. (KLEIMAN, 2005, p. 13).

Cruz e Albuquerque (2012, p.13) confirmam que o processo de alfabetização pode ser lúdico e discutem que na alfabetização também se pode envolver as crianças em situações prazerosas, contextualizadas e significativas que explorem a compreensão e a produção de textos de variados gêneros orais e escritos.

2.2 RECONHECIMENTO DE CARACTERES

O reconhecimento de caracteres (ou OCR, abreviado do inglês Optical Character Recognition) é o reconhecimento de texto em documentos impressos ou escritos à mão transformados em imagens (SHIGEHARU JUNIOR, 2014). Facon (1993, p.5) argumenta que o processamento e a análise de imagens são uma ciência e que por meio do computador é possível modificar, analisar e manipular imagens.

Conforme Shigeharu Junior (2014, p.18) comenta, existem softwares para reconhecimento de caracteres que realizam um pré-processamento de imagem para aprimorar as chances de reconhecimento por meio da binarização e segmentação. Um dos algoritmos de OCR é a matriz de correspondência que compara uma imagem com um glifo (desenho ou

carácter simbólico) armazenado em uma base de pixel-a-pixel, esse algoritmo também é conhecido como reconhecimento de padrão (SHIGEHARU JUNIOR, 2014). Outro algoritmo é o de extração de características, esse examina as características de um glifo em linhas, ciclos fechados, direção e cruzamento de linha e compara com uma representação vetorial de um carácter (SHIGEHARU JUNIOR, 2014).

Para reconhecer um carácter é possível utilizar um classificador chamado Bayes, ele utiliza uma classificação estatística que utiliza uma abordagem probabilística. Esse classificador calcula uma distância mínima entre vetores (EIKVIL, 1993, p.19).

2.3 TESSERACT

O Tesseract, conforme descrito pela Google (2019), é um mecanismo de OCR Open Source no qual possui capacidade para reconhecer mais de cem idiomas, contém suporte para Unicode e pode ser treinada para reconhecer outras línguas. O Google (2019) também cita que a empresa utiliza a ferramenta para detecção de texto em dispositivos móveis, em vídeos e na detecção de spam de imagens no Gmail, o e-mail do Google.

Para Android existe um módulo chamado `tess-two` que reúne várias ferramentas para compilar as bibliotecas do Tesseract, esse módulo fornece uma biblioteca Java para acessar as bibliotecas Tesseract. O módulo `tess-two` disponibiliza também funções nativas para detecção de texto, detecção de borrões, detecção de fluxo óptico e limite (THEIS, 2011).

2.4 TRABALHOS CORRELATOS

Neste capítulo são apresentados três trabalhos correlatos que apresentem características semelhantes ao objetivo proposto. A seção 2.4.1 descreve um software para alfabetização com técnica de Redes Neurais Artificiais (RNA). Já seção 2.4.2 detalha um aplicativo que utiliza reconhecimento de caracteres para ler o número de um cartão de crédito. E a seção 2.4.3 apresenta um aplicativo que auxilia o processo de alfabetização.

2.4.1 Software de auxílio à pré-alfabetização infantil (Aprendendo as letras)

O software de auxílio à pré-alfabetização infantil baseado no reconhecimento de caracteres manuscritos por meio de Redes Neurais Artificiais (RNA) supervisionada voltada ao público infantil e que auxilie no período de alfabetização. O software pretende incentivar a criança utilizar o computador como ferramenta de aprendizado (BELAN; NERY; ARAÚJO, 2006). O software apresentou uma média de erro de 6,15% ao reconhecer os caracteres desenhados.

Para obter o desenho a ser reconhecido como caractere, o software exibe em uma tela (Figura 1) palavras incompletas e solicita a criança que preencha nas lacunas das palavras as letras faltantes, com o auxílio de uma caneta óptica. A interface ainda possui um recurso multimídia para emitir o som correspondente a letra faltante. Após a criança desenhar a letra e submeter ao reconhecimento é fornecido um *feedback* sinalizando se a criança acertou (BELAN; NERY; ARAÚJO, 2006).

Figura 1 - Interface do software



Fonte: Belan, Nery, Araújo (2006).

2.4.2 CardTess

O aplicativo CardTess possui o objetivo de reconhecer o número de um cartão de crédito. Desenvolvido para a plataforma Android, o aplicativo reconhece o modelo de cartão de crédito e utiliza o reconhecimento de caracteres para identificar os dígitos presentes no número do cartão. Para o reconhecimento é utilizado Template Matching que é um método de filtragem com a finalidade de detectar pequenas partes de uma imagem (OHTA et al., 2017).

O aplicativo apresentou um resultado de 75% de eficiência no reconhecimento dos dígitos do cartão (KOMATI; RODRIGUES JUNIOR, 2017). O aplicativo acessa a câmera do dispositivo móvel já na tela inicial e apresenta ao usuário um formato de cartão. Assim é possível encaixar o cartão de crédito pela câmera no modelo da tela e o aplicativo escaneia os dados do cartão (Figura 2), após o escaneamento é utilizado a biblioteca OpenCV com interface para Android para realizar o reconhecimento de caracteres.

Figura 2 - Telas de leitura do cartão



Fonte: Komati; Rodrigues Junior (2017).

2.4.3 Alfabetize

O Alfabetize é um aplicativo para dispositivos móveis que visa auxiliar os professores no processo de alfabetização. O aplicativo é desenvolvido para a plataforma Android cujo objetivo é auxiliar os professores na alfabetização de alunos que se encontram na hipótese pré-silábica e estão em processo de evolução para a hipótese silábica (BARBOSA et al., 2017).

O aplicativo oferece quatro tipos de atividades para trabalhar as seguintes áreas: sílabas, letras, sons das letras e associação entre palavras e imagens. Através do aplicativo é possível cadastrar palavras, em tempo real, para serem trabalhadas nos exercícios, proporcionando uma maior dinâmica durante a aula. Essas palavras cadastradas são compartilhadas entre os usuários, evitando o retrabalho de que outro professor cadastre essa mesma palavra posteriormente (BARBOSA et al., 2017).

É disponibilizado no aplicativo a opção “Sons e letras” que permite ao usuário pressionar um botão para escutar o som de uma letra e sempre três opções para marcar a alternativa correta (Figura 3.a). Também é disponibilizado a opção “Imagens e palavras” (Figura 3.b), no qual é possível visualizar uma imagem e três opções para marcar a palavra correta relacionada a imagem (BARBOSA et al., 2017). A atividade “Quantificação de letras” (Figura 4.a) e a “Quantificação de sílabas” (Figura 4.b) possibilitam o usuário quantificar as letras e as sílabas de uma palavra.

Figura 3 - a) Tela de sons e letras b) Tela de imagens e palavras



Fonte: Barbosa et al (2017).

Figura 4 - a) Tela de quantificação de letras b) Tela de quantificação de sílabas



Fonte: Barbosa et al (2017).

3 DESENVOLVIMENTO

Neste capítulo são apresentadas as etapas do desenvolvimento do protótipo. A seção 3.1 consta os requisitos principais do trabalho. Na seção 3.2 é apresentada a especificação, com modelos e diagramas. A implementação está na seção 3.3. Por fim, a seção 3.4 detalha os resultados obtidos.

3.1 LEVANTAMENTO DE INFORMAÇÕES

O aplicativo deverá atender os seguintes requisitos:

- a) disponibilizar uma interface com imagens do alfabeto para seleção e uma área para desenho correspondente a imagem selecionada (Requisito Funcional - RF);
- b) disponibilizar uma interface com a opção “Letra Aleatória” para seleção e uma área para desenho correspondente a imagem escolhida aleatoriamente (RF);
- c) reconhecer o desenho feito na tela do dispositivo móvel (RF);
- d) disponibilizar para o usuário um indicador de correspondência entre a imagem selecionada e a imagem desenhada (RF);
- e) disponibilizar um aplicativo Android (Requisito Não-Funcional - RNF);
- f) reconhecer a imagem com a biblioteca Tesseract (RNF).

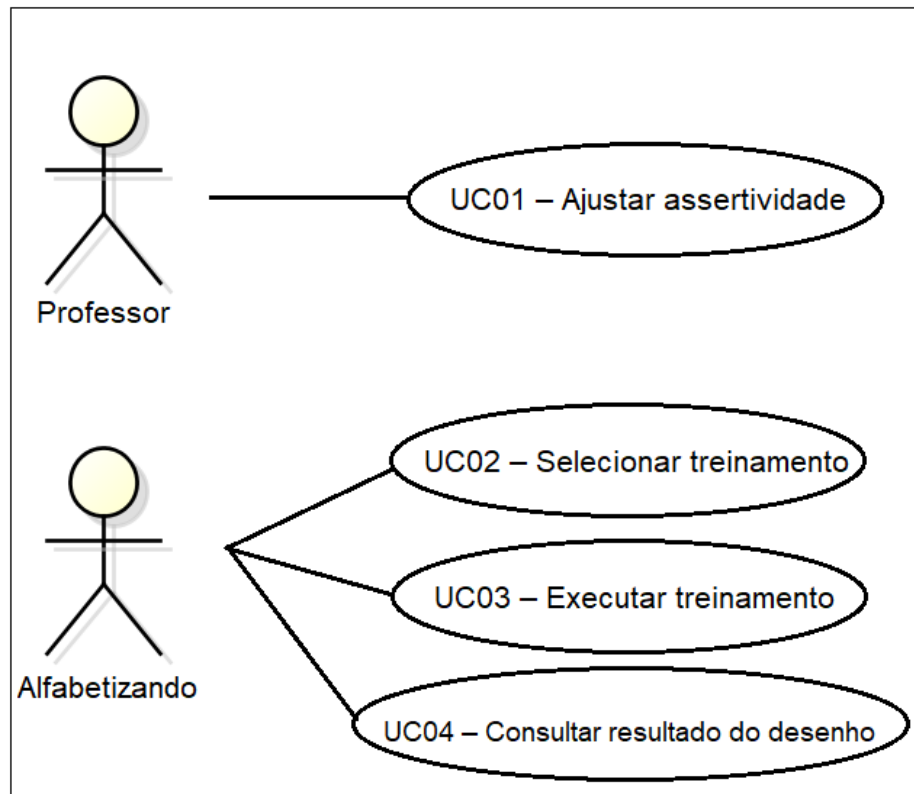
3.2 ESPECIFICAÇÃO

A especificação do aplicativo foi representada através de diagramas casos de uso e diagrama de classes.

3.2.1 Diagrama de casos de uso

Nesta seção são apresentados os casos de uso para o desenvolvimento do aplicativo conforme ilustra a Figura 5. Identificou-se os atores denominados professor e alfabetizando, na qual realizam todas as funcionalidades do aplicativo.

Figura 5 - Diagrama de caso de uso



Fonte: elaborado pela autora.

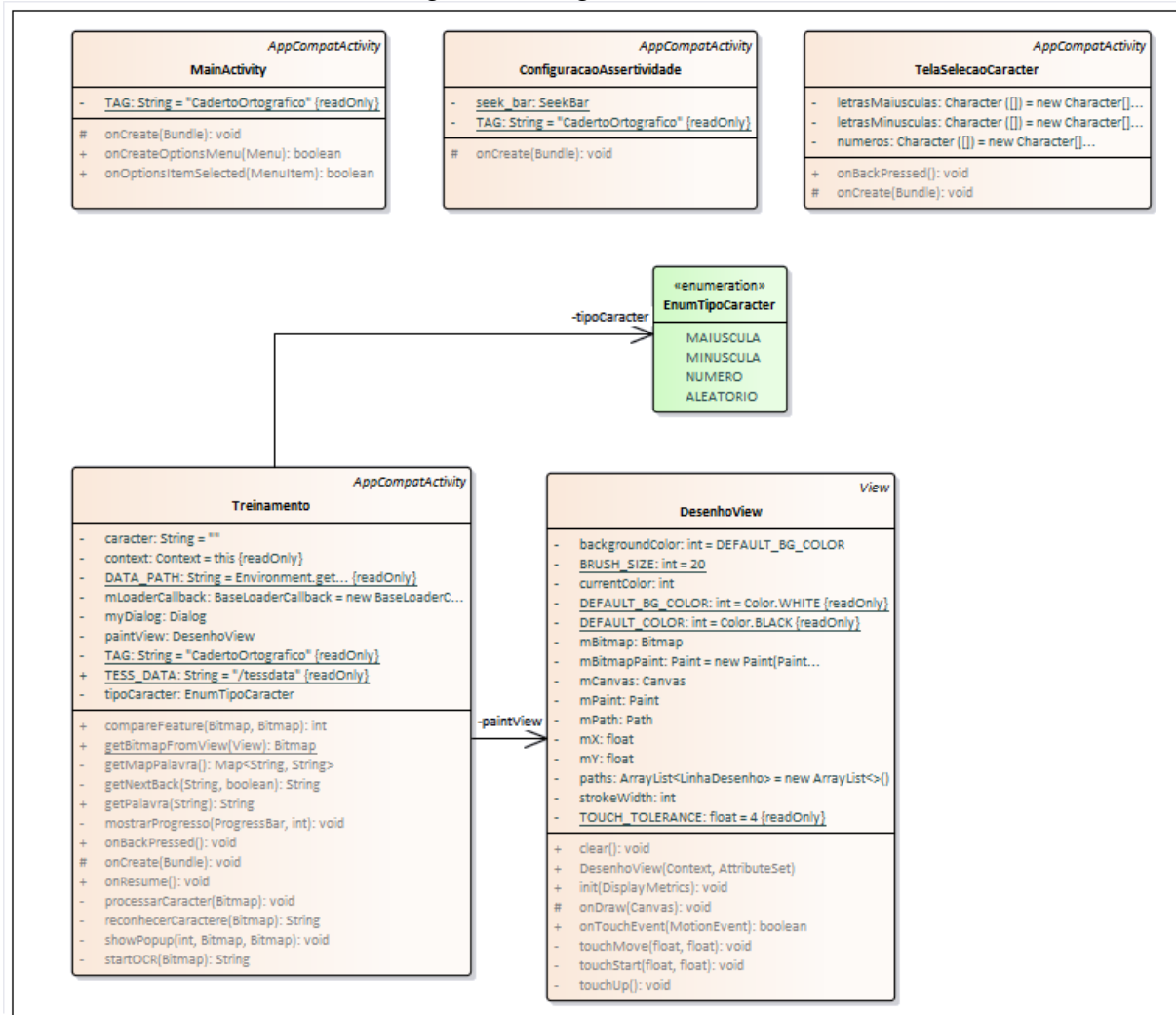
O ator `professor` tem como objetivo realizar a edição da assertividade, na qual poderá definir se o aplicativo deve ajustar a rotação da imagem para realizar o reconhecimento do caractere, através do caso de uso UC01 - Ajustar assertividade.

O ator denominado `alfabetizando` deve escolher qual o tipo de treinamento que deseja por meio do caso de uso UC02 - Selecionar treinamento e deve executar o treinamento do caractere a partir do caso de uso UC03 - Executar treinamento. O alfabetizando poderá ver o resultado através do caso de uso UC04 - Consultar resultado do desenho. Os casos de usos estão descritos no Apêndice A.

3.2.2 Diagrama de classes

Nesta seção constam o diagrama de classes do aplicativo, explicando como funciona o relacionamento e os objetos. A Figura 6 apresenta o diagrama de classes.

Figura 6 - Diagrama de classes



Fonte: elaborado pela autora

A classe `MainActivity` fornece as informações da primeira tela, onde desta serão chamadas as telas de seleção de caractere ou treinamento. A classe `ConfiguracaoAssertividade` possui a funcionalidade de ajustar a assertividade. Já a classe `TelaSelecaoCaracter` disponibiliza ao usuário os caracteres conforme o tipo de treinamento escolhido. A classe `Treinamento` é o núcleo do aplicativo, onde possui as funcionalidades de validação de caractere e utiliza a classe `DesenhoView` como componente de desenho.

3.3 IMPLEMENTAÇÃO

A seguir são mostradas as técnicas e ferramentas utilizadas na implementação do aplicativo, a descrição do desenvolvimento do trabalho e a sua operacionalidade.

3.3.1 Técnicas e ferramentas utilizadas

O desenvolvimento do aplicativo foi realizado com a linguagem Java e auxílio do ambiente de desenvolvimento Android Studio, versão 3.4. Como ferramenta de *build* foi utilizado o Gradle. A versão mínima do Android configurada para que o aplicativo execute é 23 e a versão de destino é a 28.

Para identificar o desenho realizado na tela do dispositivo móvel foi criado um componente chamado `DesenhoView` que estende da classe `View` e pelo toque realizado na tela cria um mapa dos pixels, esse representado pela `Bitmap` do Android.

A identificação do caractere ficou sobre responsabilidade da biblioteca Tesseract, versão 7.0 para Android, para isso foi inserido uma dependência no arquivo `build.gradle` chamada “com.rmtheis:tess-two:7.0.0”. Para utilizar a biblioteca somente é necessário criar um objeto do tipo `TessBaseAPI` e invocar o método `init`, no qual é necessário passar o diretório que se encontra o arquivo `eng.traineddata`. Então é possível utilizar o método `setImage`, passando como parâmetro o *Bitmap* do desenho criado pelo usuário e após invocar o método `getUTF8Text` que retorna os caracteres reconhecidos na imagem enviada anteriormente. A função que realiza o reconhecimento é apresentada no Quadro 1.

Quadro 1 - Método que processa o reconhecimento de caractere com Tesseract

```

1 private String reconhecerCaractere(Bitmap bitmap){
2     try{
3         TessBaseAPI tessBaseAPI = new TessBaseAPI();
4         String dataPath = getExternalFilesDir("/").getPath() + "/";
5         tessBaseAPI.init(dataPath, "eng");
6         tessBaseAPI.setImage(bitmap);
7         String retStr = tessBaseAPI.getUTF8Text();
8         tessBaseAPI.end();
9         return retStr;
10    }catch (Exception e){
11        Log.e(TAG, e.getMessage());
12    }
13    return null;
14 }
```

Fonte: elaborado pela autora.

Para processar o caractere foi criado um método chamado `processarCaracter` que a partir de um `Bitmap` gerado fosse possível validar se a biblioteca Tesseract conseguiria interpretar. Nesse mesmo método foi escrita a regra que valida a rotação da imagem, a partir de uma configuração que o usuário poderia alterar. Caso não fosse possível interpretar os caracteres no desenho e a configuração de rotação estivesse ativa o método utilizava a função

`postRotate` da classe `Matrix` para realizar a rotação da imagem com os graus configurados pelo usuário até atingir 360 graus. No Quadro 2 é apresentada a função `processarCaracter`, é possível verificar que no início a função carrega a biblioteca `tess-two` através do método `prepareTessData` e a configuração de rotação para então realizar a primeira validação do desenho, caso o resultado seja o caractere esperado chama a função `showPopup` enviando o percentual de acerto, que no caso é 100, a letra e o desenho para que seja exibido em tela. Caso o resultado não seja o caractere esperado é verificado a variável `validarRotacao`, se o valor da mesma for falso, irá apresentar também pela função `showPopup` um percentual de erro que é 5%. O valor de 5% foi considerado para que o usuário veja um pouco de resultado e continue tentando. Caso a variável `validarRotacao` possuir um valor verdadeiro a mesma rotina acima se repetirá mudando o desenho, que sofrerá ajustes na rotação.

Quadro 2 - Método que processa o desenho

```

1 private void processarCaracter(Bitmap desenhoFeito){
2     prepareTessData();
3     SharedPreferences settings =
4     getSharedPreferences("CadernoOrtograficoPreferences",
5     MODE_PRIVATE);
6     int grau = settings.getInt("grau_rotacao", 10);
7     boolean validarRotacao =
8     settings.getBoolean("validar_grau_rotacao", false);
9     String resultado = startOCR(desenhoFeito);
10    TextView letraTv = findViewById(R.id.apresentacaoLetra);
11    letraTv.buildDrawingCache();
12    Bitmap letraBit = letraTv.getDrawingCache();
13
14    if(resultado.toUpperCase().equals(caracter.trim().toUpperCase())){
15        showPopup(100, letraBit, desenhoFeito);
16    } else {
17        Bitmap rotated = null;
18        boolean reconheceu = false;
19        if (validarRotacao) {
20            int rotacao = grau;
21            while (rotacao <= 360) {
22                Matrix matrix = new Matrix();
23                matrix.postRotate(rotacao);
24                rotated = Bitmap.createBitmap(desenhoFeito, 0, 0,
25                desenhoFeito.getWidth(), desenhoFeito.getHeight(), matrix, false);
26                resultado = startOCR(rotated);
27
28                if(resultado.toUpperCase().equals(caracter.trim().toUpperCase())){
29                    reconheceu = true;
30                    break;
31                }
32                rotacao += grau;
33            }
34            if(reconheceu) {
35                showPopup(100, letraBit, rotated);
36                paintView.setBackground(new
37                BitmapDrawable(getApplication().getResources(), rotated));
38            } else {
39                showPopup(5, letraBit, desenhoFeito);
40            }
41        }
42    }
43 }

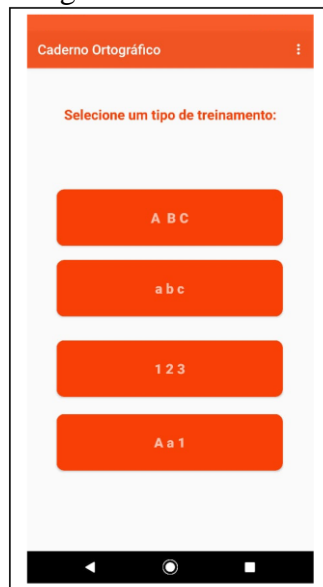
```

Fonte: elaborado pela autora

3.3.2 Operacionalidade da implementação

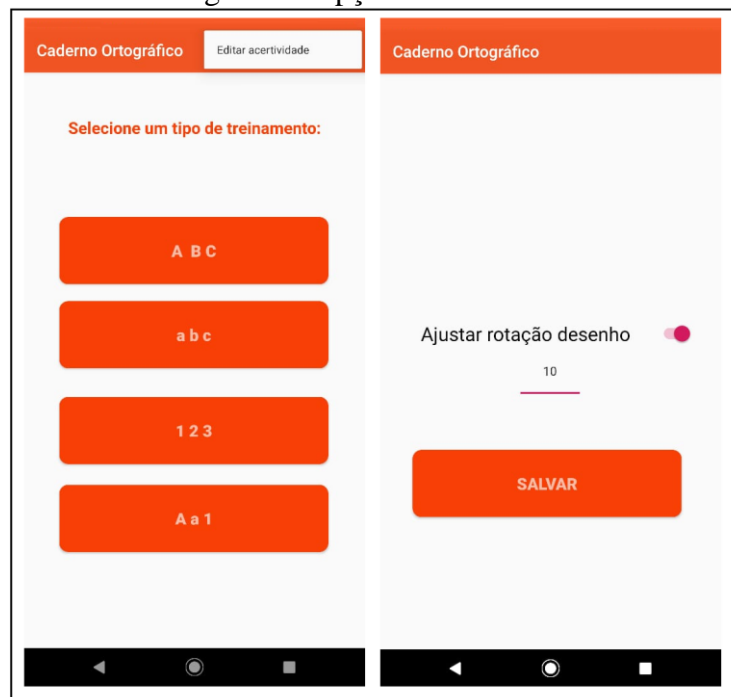
Nessa subseção será apresentado o funcionamento do aplicativo a nível de usuário. O aplicativo *Caderno Ortográfico*, ao ser iniciado (Figura 7), apresenta uma tela de seleção com quatro tipos de treinamento, sendo eles: letras maiúsculas (A, B, C), letras minúsculas (a, b, c), números (1, 2, 3) e opção aleatória (A, a, 1). Nessa tela também é apresentado um menu com a opção de editar assertividade (Figura 8).

Figura 7 - Tela inicial



Fonte: elaborado pela autora.

Figura 8 - Opção Assertividade

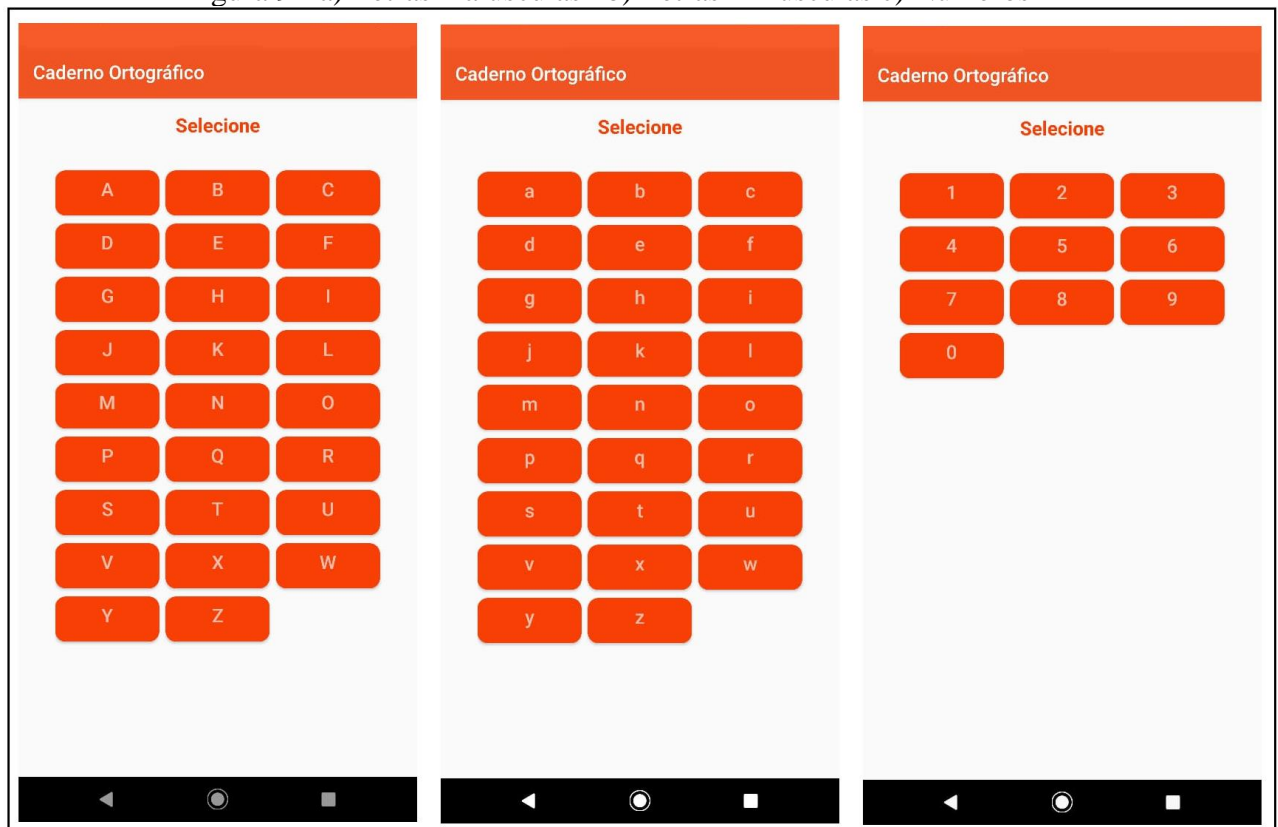


Fonte: elaborado pela autora.

Caso o usuário escolha a opção de editar assertividade (Figura 8), será apresentada uma nova tela, na qual poderá ser selecionado se o sistema deve ajustar o grau no desenho de treinamento e de quantos em quantos graus o sistema deve ajustar o desenho para posterior validação da escrita. Após o usuário alterar na tela essas duas opções, ajustar rotação e quantidade de graus, ele poderá salvar através do botão salvar suas alterações.

O usuário, ainda na tela inicial do aplicativo, pode escolher o tipo de treinamento de letras maiúsculas ou minúsculas, a partir da sua escolha o usuário é transferido para uma segunda tela que apresenta todas as letras presentes no alfabeto maiúsculas (Figura 9.a) ou minúsculas (Figura 9.b). Com isso é possível selecionar a letra e ser transferido para a área de treinamento. Se o usuário escolher a opção de treinamento de números, ele também será transferido para uma nova tela onde é apresentado os números (Figura 9.c) de 0 a 9, após a escolha de um desses números ele será transferido para a área de treinamento.

Figura 9 - a) Letras maiúsculas b) Letras minúsculas c) Números

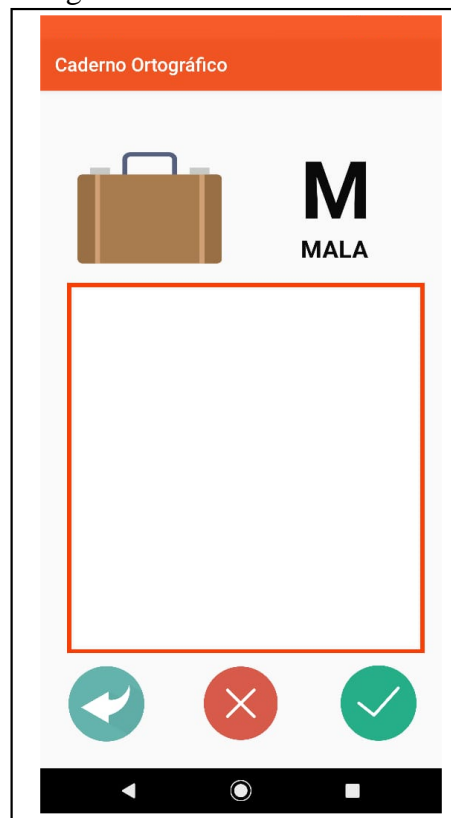


Fonte: elaborado pela autora.

Na tela inicial ainda há uma quarta e última opção que é a aleatória, em seguida o sistema utiliza uma rotina de escolha aleatório do caractere, sendo eles letras maiúscula, minúsculas ou números e direciona o usuário para a tela de treinamento com o caractere aleatório escolhido pelo sistema (Figura 10).

A tela de treinamento (Figura 10) apresenta uma imagem e uma palavra relacionada com o caractere escolhido, juntamente com uma área para o desenho. É apresentado também três botões, o de voltar, retratado por um ícone de seta virada a esquerda, o botão de cancelar, retratado por um ícone de x, e o botão de validar, retratado por um ícone verde com uma seta de certo. Caso o usuário selecione o de voltar o aplicativo redireciona para a tela anterior. O botão de cancelar limpa o campo de desenho, permitindo que o usuário inicie novamente. Já o botão de validar avalia se o desenho feito é o caractere escolhido anteriormente. A rotina de validação verifica se foi possível extrair do desenho o caractere que está sendo treinado. Caso não consiga verifica se o usuário configurou para que seja validado a partir de rotações no desenho. Com a opção de rotação desativada é apresentado o resultado, caso a opção de rotação esteja ativada é iniciada uma rotina que executa a rotação da imagem conforme os graus inseridos na configuração e tenta novamente reconhecer o caractere. Essa rotina de rotação é executada até que a biblioteca consiga reconhecer o caractere ou então até ter rodado o desenho em 360 graus.

Figura 10 - Tela de treinamento

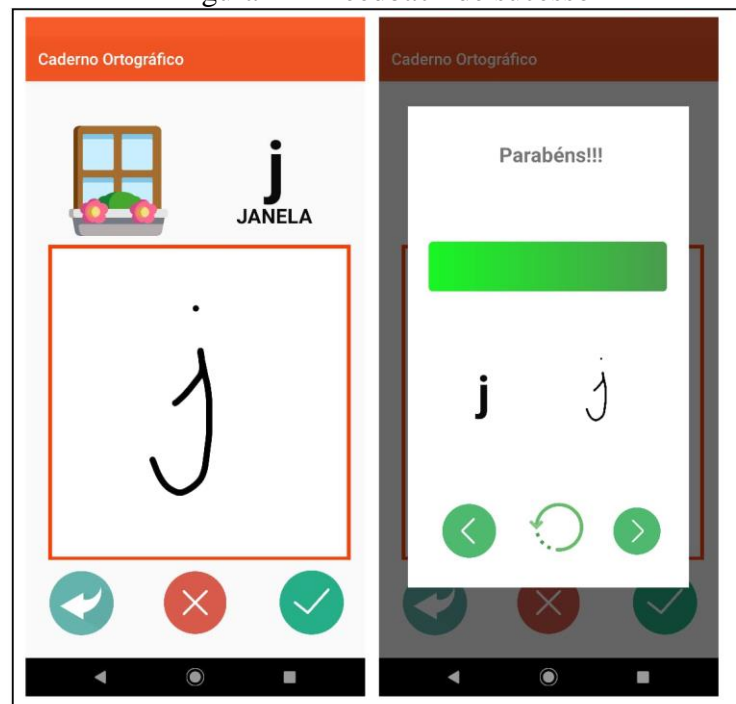


Fonte: elaborado pela autora.

A apresentação do resultado da validação do desenho se dá por meio de um `popup` na tela do dispositivo móvel. É apresentado uma frase para demonstrar o sucesso ou a falha ao reconhecer o caractere. Caso a rotina reconheça o caractere a frase apresentada é “Parabéns”

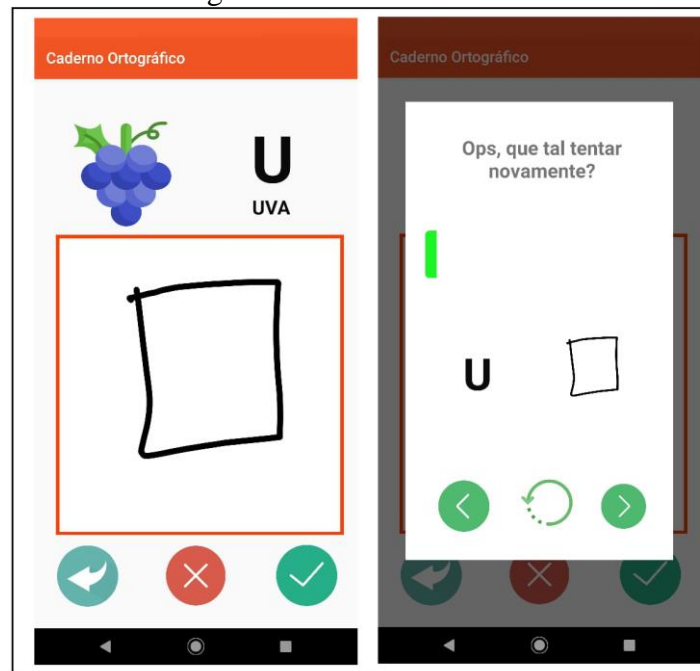
(Figura 11), caso não reconheça é apresentado “Ops, que tal tentar novamente?” (Figura 12). É possível visualizar também uma barra com a cor verde que apresenta a barra cheia quando for sucesso ou a barra preenchida apenas cinco por cento, para não dar a entender ao usuário que ele falhou, mas sim que ele precisa melhorar. No `popup` é possível visualizar a letra e o desenho para que seja realizado uma comparação visual, caso a configuração de rotação estiver ligada é apresentado o desenho ajustado (Figura 13). Além disso são disponibilizados três botões, o de anterior, representado por uma seta para a esquerda que leva o usuário a voltar um caractere. O botão de tentar novamente, representado por uma seta em círculo que leva ao usuário ao treinamento da letra atual. E também há o botão próximo, que é um ícone de seta apontando para a direita, que leva a um próximo caractere para treinamento.

Figura 11 - Feedback de sucesso



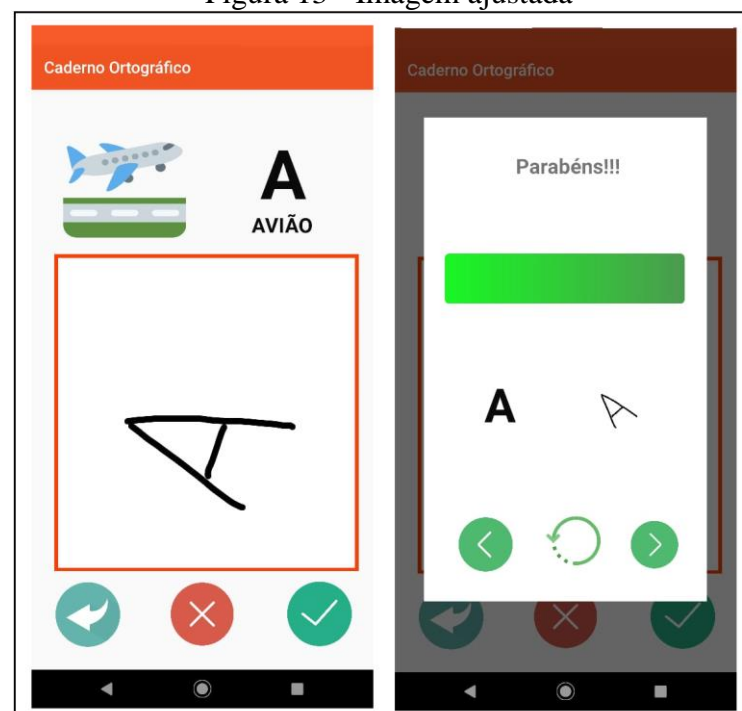
Fonte: elaborado pela autora.

Figura 12 - Feedback de erro



Fonte: elaborado pela autora.

Figura 13 - Imagem ajustada



Fonte: elaborado pela autora.

3.4 RESULTADOS E DISCUSSÕES

Nesta seção são apresentados os resultados obtidos com o aplicativo. Na seção 3.4.1 é feita a análise do resultado do reconhecimento dos caracteres. Na seção 3.4.2 é apresentado uma comparação entre os trabalhos correlatos e este trabalho.

3.4.1 Análise dos resultados

Foram analisados os 62 caracteres disponíveis no aplicativo para identificar se todos eram reconhecidos pelo Tesseract. Essa análise foi realizada com a configuração de rotação desligada. Para a análise foi realizado três desenhos para cada caractere que seguiam três cenários: desenho realizado com o caractere no formato padrão, desenho realizado com uma leve inclinação e desenho realizado com os traços curvilíneos. Do total de 62 caracteres, 57 (91,94%) foram reconhecidos no cenário de formato padrão, 5 (8,06%) caracteres foram reconhecidos no cenário com uma leve inclinação e 29 (46,77%) reconhecidos com traços curvilíneos. O resumo dos resultados é apresentado em quatro tabelas, a Tabela 1 que demonstra consolidação dos resultados, a Tabela 2 que apresenta os resultados das letras maiúsculas, a Tabela 3 apresenta as letras minúsculas e a Tabela 4 na qual é possível validar o resultado dos números. As evidências dos resultados estão no Apêndice B.

A configuração de rotação ligada não influencia diretamente nos resultados do reconhecimento de caractere pela biblioteca Tesseract, a opção de rotação fornece mais chances de acerto para o reconhecimento dependendo de quantos graus foram configurados.

Tabela 1 - Consolidação dos resultados

cenários	maiúscula	minúscula	números
padrão	26	23	8
inclinação	3	2	0
curvilíneo	18	6	5

Fonte: elaborado pela autora.

Tabela 2 - Resultados letras maiúsculas

cenários	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
reconhece	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
inclinação			•		•														•							
curvilíneo			•	•	•		•	•	•			•	•	•	•	•		•	•	•	•		•		•	•

Fonte: elaborado pela autora.

Tabela 3 - Resultados letras minúsculas

característica	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
reconhece	•	•	•	•	•	•		•	•	•	•	•	•		•	•	•		•	•	•	•	•	•	•	•
inclinação			•								•															
curvilíneo	•								•	•									•				•		•	•

Fonte: elaborado pela autora.

Tabela 4 - Resultados números

característica	0	1	2	3	4	5	6	7	8	9
reconhece			•	•	•	•	•	•	•	•
inclinação										
curvilíneo			•			•	•	•	•	

Fonte: elaborado pela autora.

3.4.2 Comparativo entre o aplicativo desenvolvido e seus correlatos

A partir das informações obtidas com os trabalhos correlatos e com o trabalho acima descrito, foi montado o quadro abaixo, com comparação com os trabalhos correlatos.

Quadro 3 - Comparativo entre os trabalhos correlatos

Características	Aprendendo as letras Belan, Nery, Araújo (2006)	CardTess Komati; Rodrigues Junior (2017)	Alfabetize Barbosa et al (2017)	Caderno Ortográfico Sena (2019)
Plataforma	Windows	Android	Android	Android
layout interativo	X	X	X	X
reconhece caracteres	X	X		X
propõe auxílio na alfabetização	X		X	X
determina percentual de igualdade entre imagens	X			

Fonte: elaborado pela autora.

Analisando o quadro é possível verificar que o trabalho Caderno Ortográfico não possui todas as características apresentadas nos trabalhos correlatos, porém o comparando com os trabalhos CardTess (2017) e Alfabetize (2017), que também são trabalhos desenvolvidos para a plataforma Android, é possível avaliar que este atingiu mais resultado, focando no auxílio a alfabetização e no reconhecimento dos caracteres. Para que o Caderno Ortográfico (2019) determinasse o percentual de igualdade e entregasse um feedback ao usuário mais assertivo seria necessário investir mais horas em estudos no processamento de imagens combinado com uma base de comparação de letras.

4 CONCLUSÕES

O objetivo principal do trabalho, que era interpretar o desenho realizado na tela de um dispositivo móvel identificando se era compatível com um caractere e apresentar um indicador de correspondência foi parcialmente alcançado, pois não foi possível determinar um percentual de assertividade para o usuário. A partir da escolha da letra a ser treinada e da captura do desenho na tela do dispositivo móvel é possível identificar com a biblioteca *Tesseract* se o desenho é o caractere esperado, porém caso o usuário informe algo minimamente diferente do esperado, conforme apresentado na seção 4.1.1, a biblioteca não consegue reconhecer.

O protótipo permite ao usuário realizar a seleção de qual caractere quer treinar, podendo escolher por letras maiúsculas, minúsculas, números ou uma opção aleatória. Também permite ao usuário configurar se caso o sistema não reconhecer o caractere, o desenho feito pelo usuário possa ser ajustado em até trezentos e sessenta graus para avaliar se é possível ainda reconhecer a letra. É apresentado no protótipo uma área para desenho, componentes que ajudam no reconhecimento do caractere como imagens, palavras e o próprio caractere, porém na hora de validar o desenho somente é possível indicar se foi sucesso ou falha e não exatamente um indicador de assertividade.

O uso do *Tesseract* não garante a eficácia na interpretação, para isso seria necessário utilizar outros métodos como aprendizado de máquina, bibliotecas com base de dados de caracteres escrito a mão, como o *MNIST*, e processamento de imagens.

A ideia é válida, porém, é necessário realizar mais estudos afim de aprimorar o reconhecimento dos dígitos.

4.1 EXTENSÕES

Algumas das possíveis extensões para este trabalho são:

- a) aprimorar o reconhecimento do desenho com processamento de imagens;
- b) apresentar um indicador de semelhança validando ponto a ponto na imagem;
- c) colocar um sistema de pontuação;
- d) criar uma funcionalidade de cadastro para guardar o desenvolvimento do usuário;
- e) criar novos perfis como professor e aluno para as configurações de rotação.

REFERÊNCIAS

- ANDRADE, Ana P. R. **O uso das tecnologias na educação: computador e internet**. 2011. 22 f. Monografia (Licenciatura em Biologia) – Consórcio Setentrional de Educação a Distância, Universidade de Brasília / Universidade Estadual de Goiás, Brasília.
- AVAZU INC. **Avazu – Global Internet Industry Research**. 2016. Disponível em: <<http://avazuinc.com/resources/whitepapers/>>. Acesso em: 14 abr. 2018.
- BARBOSA, Maria Eduarda Q.; SOUZA, Naijda; SILVA, Adilza; LINS, Robson. **Alfabetize: um aplicativo móvel de apoio à alfabetização**. In: Congresso Brasileiro de Informática na Educação, VI., 2017, Recife - PE. Local: Anais dos Workshops do VI Congresso Brasileiro de Informática na Educação, 2017. p.308 - 314.
- BELAN, Peterson Ariano.; NERY, Edilaine Petinari.; ARAÚJO, Sidnei Alves. **Software para auxílio à pré-alfabetização infantil baseado em reconhecimento inteligente de caracteres manuscritos**. In: Exacta, São Paulo, v. 4, n. especial, p. 85-86, 25 nov. 2006.
- BURKE, Dave. **Android: celebrating a big milestone together with you**. 2017. Disponível em: <<https://blog.google/products/android/2bn-milestone/>>. Acesso em: 10 abr. 2018.
- CRUZ, Magna do Carmo Silva; ALBUQUERQUE, Eliana Borges Correia de. A complexidade da aprendizagem do sistema de escrita alfabética: ampliação do tempo para a consolidação da leitura e da escrita pela criança. In: Ministério da Educação. Secretaria da Educação Básica. **Pacto Nacional pela Alfabetização na Idade Certa: Currículo no ciclo de alfabetização: consolidação e monitoramento do processo de ensino e de aprendizagem: ano 2: unidade 1**. Brasília, 2012.
- EIKVIL, L. Oslo, 1993. **OCR Optical Character Recognition**. [S.l.], 1993. Disponível em: <<http://www.nr.no/~eikvil/OCR.pdf>>. Acesso em: 22 abr. 2018.
- FACON, Jacques; ESCOLA BRASILEIRO-ARGENTINA DE INFORMATICA. 6, 1993, Embalse. **Processamento e análise de imagens**. Embalse : EBAI, 1993. xv, 198p, il.
- GROSSI, E. P. **Alfabetização em classes populares: didática do nível pré-silábico**. São Paulo: SE/CENP, 1985.
- GOOGLE. **Google Open Source – Tesseract OCR**. [2019?]. Disponível em: <<https://opensource.google.com/projects/tesseract/>>. Acesso em: 20 jun. 2019.
- KLEIMAN, A. B. **Preciso ensinar o letramento? Não basta ensinar a ler e a escrever?** São Paulo: Produção Editorial, 2005.
- KOMATI, Karin Satie; RODRIGUES JUNIOR, Lincoln Soares. **Um Sistema Android para Reconhecimento Automático do Número de Cartão de Crédito usando Template Matching**. Revista Eletrônica Argentina-Brasil de Tecnologias da Informação e da Comunicação, [S.l.], v. 1, n. 7, ago. 2017. ISSN 2446-7634. Disponível em: <<https://revistas.setrem.com.br/index.php/reabtic/article/view/232>>. Acesso em: 19 abr. 2018.
- MENDONÇA, Onaide Schwartz; MENDONÇA, Olympio Correa de. **Psicogênese da Língua Escrita: contribuições, equívocos e consequências para a alfabetização**. In: UNIVERSIDADE ESTADUAL PAULISTA. Pró-Reitoria de Graduação. Caderno de formação: formação de professores: Bloco 02: Didática dos conteúdos. São Paulo: Cultura Acadêmica, 2011. v. 2. p. 36-57. (D16 - Conteúdo e Didática de Alfabetização). Disponível em: <<http://acervodigital.unesp.br/handle/123456789/40138>>. Acesso em: 10 abr. 2018.

OHTA, Daniel L. A.; BAUER, Elisa; MACHADO, Will R. S.; BONFIM, Thatia R.. **Um protótipo de uma ferramenta de detecção automática de alterações posturais baseada em imagens.** In: WVC 2006 - II Workshop de Visão Computacional, São Carlos, SP 2006, p. 370-373.

SHIGE HARU JUNIOR, I. A. **Estudo da viabilidade do uso de um OCR na placa Beagleboard e sua integração no xLupa embarcado.** 2014. 67f. Monografia para conclusão de curso (Ciência da Computação). Centro e Ciências Exatas e Tecnológicas, Universidade Estadual do Oeste do Paraná, Cascavel.

THEIS, Robert. **Fork of Tesseract Tools for Android.** 2011. Disponível em: <<https://github.com/rmtheis/tess-two/>>. Acesso em: 01 abr. 2019.

APÊNDICE A – Descrição dos Casos de Uso

Este Apêndice apresenta a descrição dos principais casos de uso conforme previstos no diagrama apresentado na seção 3.2.1.

Quadro 4 - UC01 Ajustar assertividade

<p>UC01 Ajustar assertividade Caso de uso onde pode-se ajustar qual a assertividade que o usuário alfabetizando deve executar o treinamento do caractere.</p> <p>Ator: Professor</p> <p>Pré-condição – Não há Pós-condição – Configuração de validar grau e quantidade de grau atualizada.</p> <p>Cenário: O ator acessa o aplicativo no dispositivo móvel e aciona o menu da tela inicial chamado “Ajustar assertividade”. O aplicativo apresenta uma nova tela com a informação de validar grau e quantidade. O professor indica os dados e aciona o processo de salvar. O aplicativo valida os dados e armazenas as informações. Fim do caso de uso.</p>

Fonte: elaborado pela autora.

Quadro 5 - UC02 – Selecionar treinamento

<p>UC02 Selecionar treinamento Caso de uso no qual ocorre a escolha do tipo de treinamento.</p> <p>Ator: Alfabetizando</p> <p>Pré-condição – Não há Pós-condição – Usuário estar na tela de treinamento com o tipo escolhido correto.</p> <p>Cenário: O ator acessa o aplicativo no dispositivo móvel. O aplicativo apresenta uma nova tela com quatro opções: “A B C” (representa o treinamento de letras maiúsculas), “a b c” (representa o treinamento de letras minúsculas), “1 2 3” (representa o treinamento de números) e por último “A a 1” (representa a opção caractere aleatório). O usuário seleciona uma das quatro opções. O aplicativo direciona para a próxima tela a partir da opção escolhida: Opção “A B C”: Direciona para a tela de seleção de caractere com todas as letras do alfabeto em maiúscula. Opção “a b c”: Direciona para a tela de seleção de caractere com todas as letras do alfabeto em minúscula. Opção “1 2 3”: Direciona para a tela de seleção e caractere com os números de 0 a 9. Opção “A a 1”: Execução do caso de uso “UC03 – Executar treinamento” O usuário seleciona o caractere escolhido e inicia a execução do caso de uso “UC03 – Executar treinamento” Fim do caso de uso.</p>
--

Fonte: elaborado pela autora.

Quadro 6 - UC03 – Executar treinamento

UC03 Executar treinamento

Caso de uso no qual é possível executar o treinamento do caractere.

Ator: Alfabetizando

Pré-condição – Execução do caso de uso “UC02 – Selecionar treinamento”

Pós-condição – Usuário validar o resultado: “UC04 – Consultar resultado do desenho”

Cenário:

O aplicativo apresenta o caractere escolhido, uma imagem e uma palavra relacionada ao caractere e uma área para desenho.

O ator desenha no campo de desenho traços iguais ao do caractere escolhidos

O aplicativo executa o caso de uso “UC04 – Consultar resultado do desenho”

Fim do caso de uso.

Fonte: elaborado pela autora.

Quadro 7 - UC04 – Consultar resultado do desenho

UC04 Consultar resultado do desenho

Caso de uso no qual é possível validar o resultado da comparação do caractere com o desenho realizado no caso de uso UC03 – Executar treinamento.

Ator: Alfabetizando

Pré-condição – Execução do caso de uso “UC03 – Executar treinamento”

Pós-condição – Usuário verificar o resultado da comparação entre o desenho e o caractere

Cenário:

O aplicativo apresenta o desenho do caso de uso “UC03 – Executar treinamento” e apresenta ao usuário uma informação de sucesso ou falha, uma barra de validação e duas imagens lado a lado, sendo uma do desenho e a outra do caractere.

O ator seleciona se quer treinar outro caractere podendo escolher entre anterior e próximo ou se quer executar o mesmo caractere.

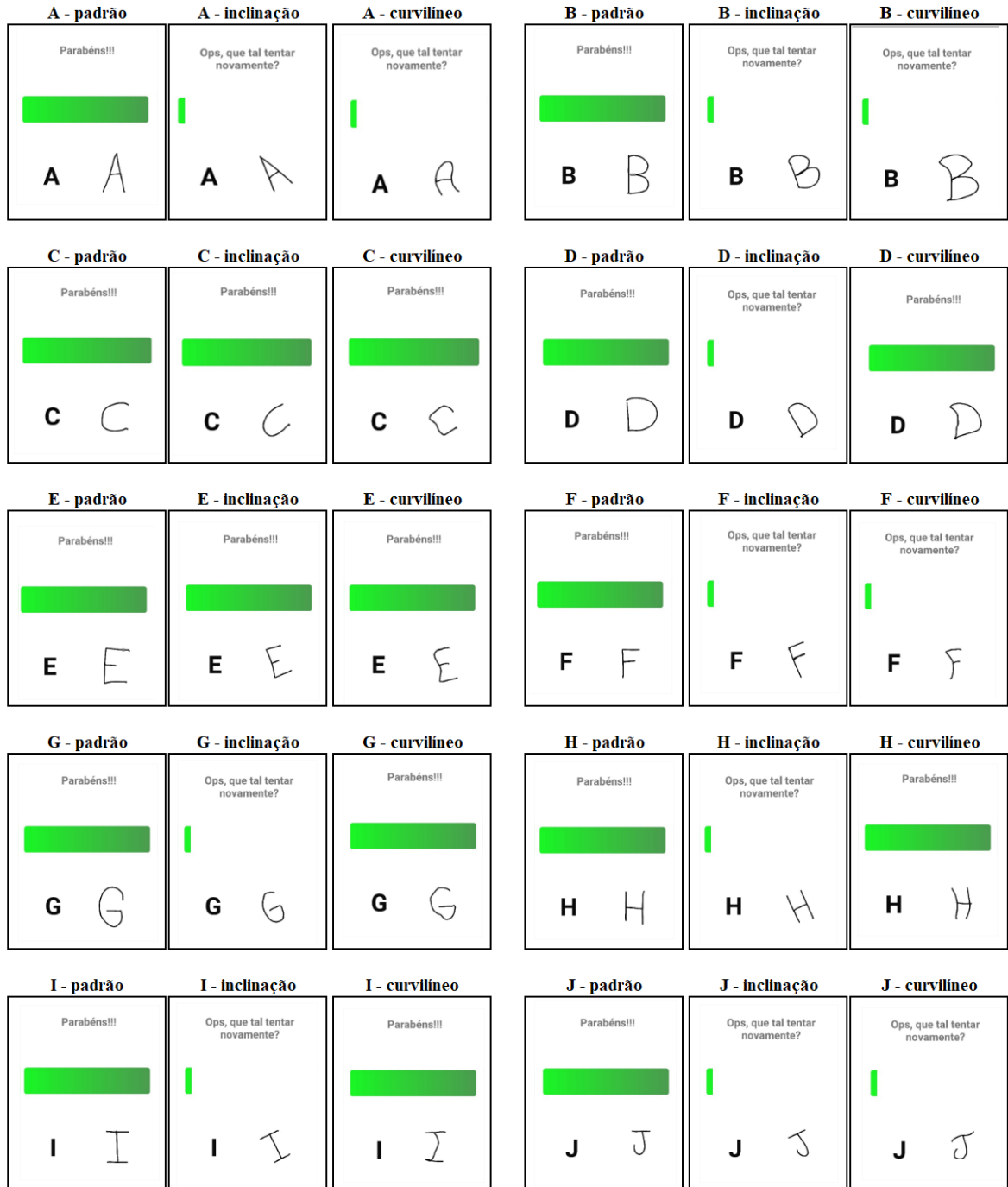
Fim do caso de uso.

Fonte: elaborado pela autora.

APÊNDICE B – Imagens da bateria de testes do aplicativo

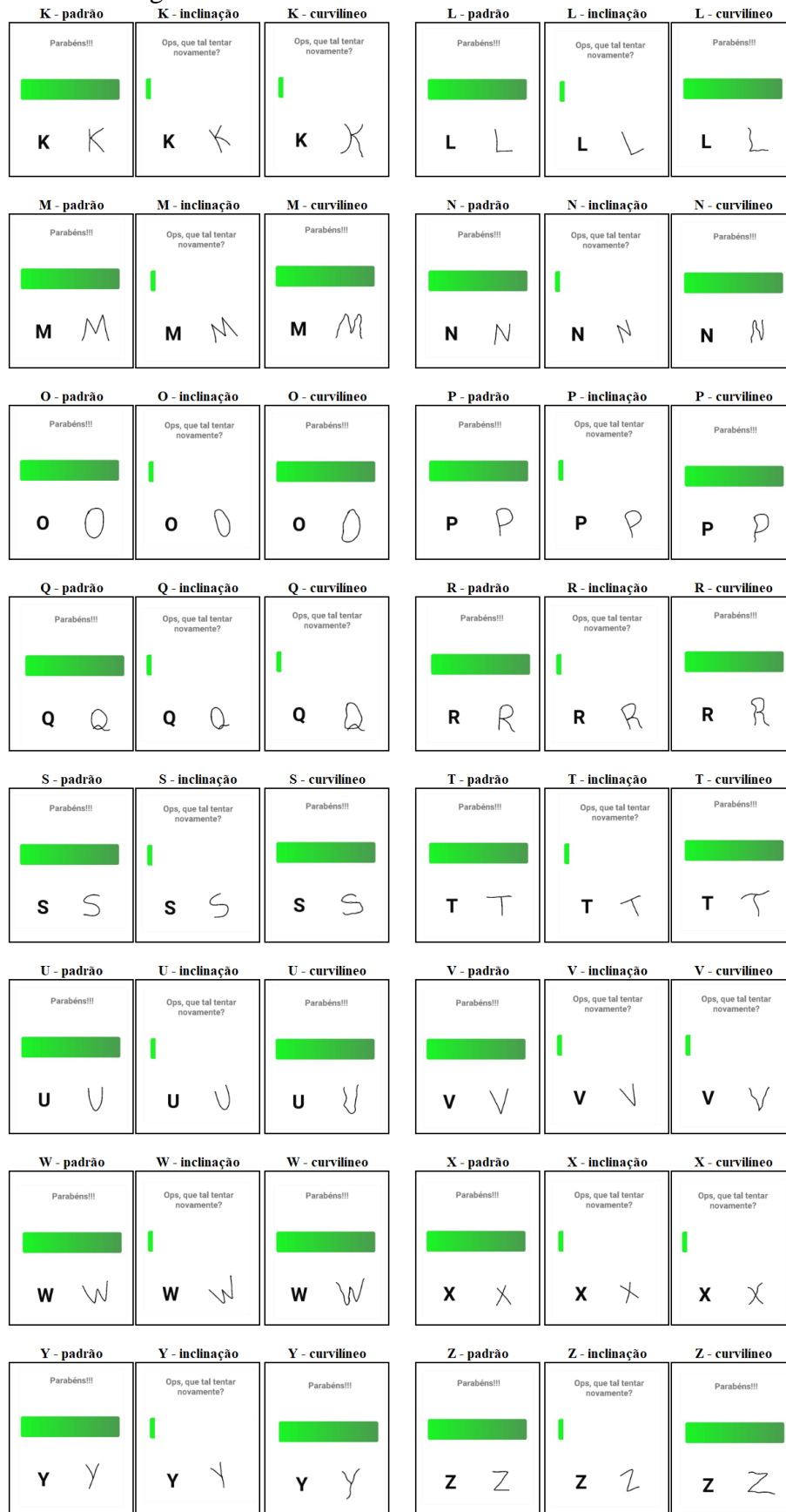
A seguir serão exibidas algumas imagens demonstrando o reconhecimento dos caracteres avaliados na seção 3.4.1.

Figura 14 - Resultado A - J das letras maiúsculas



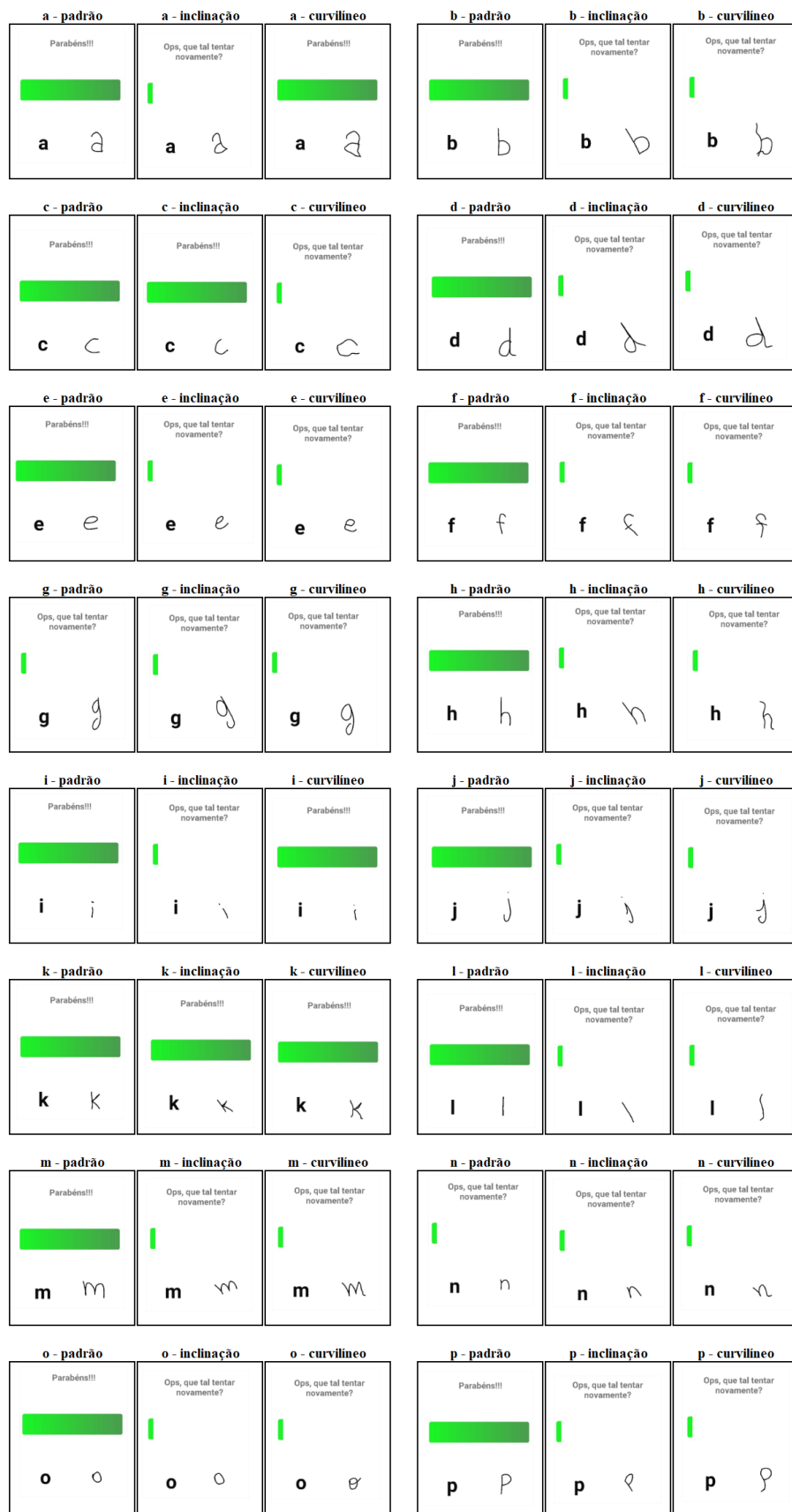
Fonte: elaborado pela autora.

Figura 15 - Resultado K - Z das letras maiúsculas



Fonte: elaborado pela autora.

Figura 16 - Resultado A - P das letras minúsculas



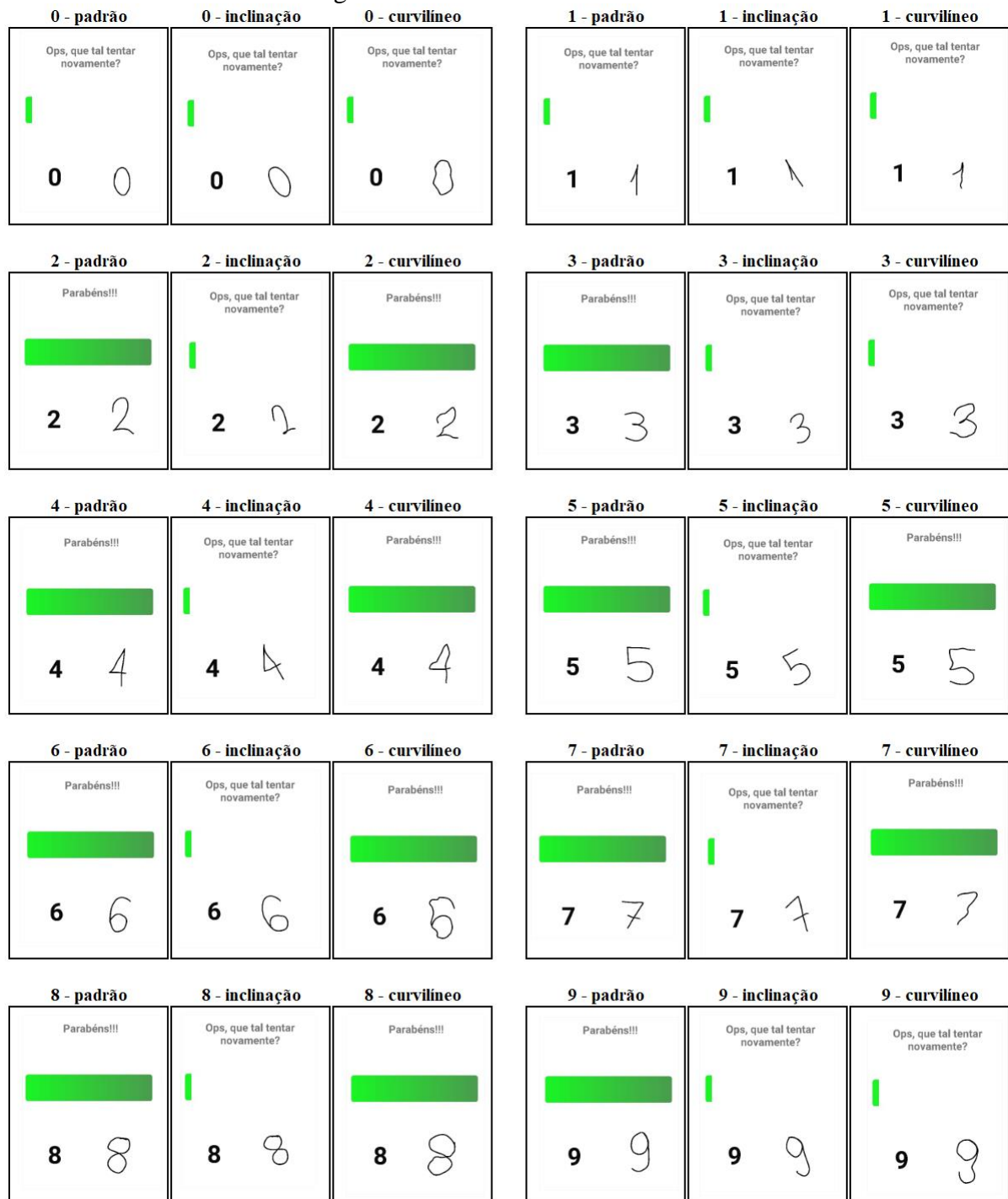
Fonte: elaborado pela autora.

Figura 17 - Resultado Q - Z das letras minúsculas



Fonte: elaborado pela autora.

Figura 18 - Resultado dos números



Fonte: elaborado pela autora.