



**CENTRO UNIVERSITÁRIO INSTITUTO DE EDUCAÇÃO  
SUPERIOR DE BRASÍLIA – IESB  
BACHAREL EM CIÊNCIA DA COMPUTAÇÃO**

**ISMAEL RAMOS DE OLIVEIRA**

**ORGANIZAÇÃO E BUSCA DE DADOS EM IMAGENS USANDO  
RECONHECIMENTO DE CALIGRAFIA E MINERAÇÃO DE DADOS**

**BRASÍLIA - DF**

**2019**

**ISMAEL RAMOS DE OLIVEIRA**

**TRABALHO DE CONCLUSÃO DE CURSO**

**ORGANIZAÇÃO E BUSCA DE DADOS EM IMAGENS USANDO  
RECONHECIMENTO DE CALIGRAFIA E MINERAÇÃO DE DADOS**

Estudo apresentado como pré-requisito para obtenção do título de Bacharel em Ciência da Computação pelo Centro Universitário Instituto de Educação Superior de Brasília, como requisito parcial para aprovação.

Orientador: MsC Cristiano Soares de Aguiar

Brasília – DF

2019

**ISMAEL RAMOS DE OLIVEIRA**

**ORGANIZAÇÃO E BUSCA DE DADOS EM IMAGENS USANDO  
RECONHECIMENTO DE CALIGRAFIA E MINERAÇÃO DE DADOS**

Estudo apresentado como pré-requisito para  
obtenção do título de Bacharel em Ciência  
da Computação pelo Centro Universitário  
Instituto de Educação Superior de Brasília,  
como requisito parcial para aprovação.

Brasília, maio de 2019.

Banca Examinadora

---

Professor MsC Cristiano Soares de Aguiar (orientador)

---

Professor Convidado

---

Professor Convidado

## DEDICATÓRIA

Dedico este trabalho à minha esposa e aos meus amigos, pela compreensão das noites escrevendo e pelo apoio e acalento, também agradeço ao meu professor pela paciência na orientação acerca da construção deste trabalho.

## **RESUMO**

O trabalho consiste na criação de um sistema que permita a estudantes tirar fotos de conteúdos relevantes utilizando dispositivos móveis. Esse sistema irá organizar as fotos baseado em matérias previamente cadastradas (manualmente ou automaticamente quando disponibilizadas pela instituição de ensino do estudante) para que seja agilizado o processo de busca visual. Outro ponto do sistema será a “tradução” das imagens em texto, automatizando a classificação e extração de pontos principais sem a intervenção do usuário. Além disso, irá permitir também o compartilhamento, entre pessoas que possuem vínculo de amizade pelo aplicativo ou para todos os usuários cadastrados no sistema, de imagens, fazendo com que o aluno tenha acesso facilitado ao conteúdo de outros usuários sem a necessidade da solicitação de envio de arquivos.

Palavras-chave: Sistema. Fotos. Dispositivos móveis. Educação.

## **ABSTRACT**

The work consists of creating a system that allows students to take pictures of relevant content using mobile devices. This system will organize the photos based on previously registered subjects (manually or automatically when made available by the student's educational institution) so that the visual search process is streamlined. Another point of the system will be the "translation" of the images into text, automating the classification and extraction of main points without the intervention of the user. In addition, it will also allow the sharing, between people who have a friendship bond for the application or for all the users enrolled in the system, of images, making the student have easy access to the content of other users without the need of the request of sending of files.

Keywords: System. Pictures. Mobile devices. Education.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Neurônio Artificial .....	18
Figura 2 – Recorrência aplicada a um neurônio .....	20
Figura 3 – Passos DM .....	24
Figura 4 – Diagrama de arquitetura .....	26
Figura 5 – Diagrama de caso de uso mantém servidor .....	28
Figura 6 – Diagrama de caso de uso mantém usuário .....	29
Figura 7 – Diagrama de caso de uso mantém aplicativo móvel .....	30
Figura 8 – Diagrama de classes .....	31
Figura 9 – Diagrama entidade relacionamento .....	33
Figura 10 – Imagens protótipo .....	43
Figura 11 – Inicialização serviço standalone Java .....	44
Figura 12 – Página inicial .....	45
Figura 13 – Tela principal .....	45
Figura 14 – Tela de cadastro de matéria .....	46
Figura 15 – Menu lateral .....	47
Figura 16 – Dados matéria .....	47
Figura 17 – Galeria de imagens .....	48
Figura 18 – Método de salvar mídia .....	49
Figura 19 – Método cadastrar mídia .....	49
Figura 20 – Método processar mídia .....	50
Figura 21 – Folha branca em letra cursiva .....	51
Figura 22 – Resultado processamento figura 21 .....	51
Figura 23 – Folha com linhas e letra de forma .....	52
Figura 24 – Resultado processamento figura 24 .....	52
Figura 25 – Imagem letra cursiva a partir de quadro branco .....	52
Figura 26 – Resultado processamento figura 25 .....	53
Figura 27 – Imagem letra de forma em quadro branco .....	53
Figura 28 – Resultado processamento figura 27 .....	54

## LISTA DE ABREVIATURAS E SIGLAS

IA	Inteligência Artificial
RNA	Rede Neural Artificial
SDK	<i>Software Development Kit</i>
DM	<i>Data Mining</i>
IDE	<i>Integrated Development Environment</i>
OCR	<i>Optical Character Recognition</i>
ICR	<i>Intelligent Character Recognition</i>
UML	<i>Unified Modeling Language</i>
HD	<i>Hard Drive</i>
API	<i>Application Programming Interface</i>



## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>11</b>
1.1	JUSTIFICATIVA .....	12
1.2	PROBLEMA .....	12
1.3	OBJETIVOS GERAIS .....	14
1.4	OBJETIVOS ESPECÍFICOS .....	14
1.5	ORGANIZAÇÃO DO TRABALHO.....	14
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>15</b>
2.1	DISPOSITIVOS MÓVEIS .....	15
2.2	USO DA TECNOLOGIA EM SALAS DE AULA.....	15
2.3	INTELIGÊNCIA ARTIFICIAL .....	16
2.3.1	<i>Redes neurais artificiais</i> .....	18
2.3.1.1	Redes neurais artificiais recorrentes .....	19
2.4	RECONHECIMENTO ÓTICO DE CARACTERES.....	20
2.5	MINERAÇÃO DE DADOS.....	22
<b>3</b>	<b>DESENVOLVIMENTO.....</b>	<b>25</b>
3.1	PROJETO .....	25
3.1.1	<i>Arquitetura</i> .....	25
3.2	ANÁLISE E MODELAGEM DA APLICAÇÃO .....	27
3.2.1	<i>Unified modeling language</i> .....	27
3.2.2	<i>Banco de dados</i> .....	32
3.2.2.1	Diagrama entidade relacionamento .....	32
3.3	TECNOLOGIAS UTILIZADAS .....	34
3.3.1	<i>Ferramentas de desenvolvimento</i> .....	34
3.3.1.1	Eclipse.....	34
3.3.1.2	Astah UML.....	34
3.3.1.3	DBeaver .....	34
3.3.1.4	Android Studio.....	35
3.3.2	<i>Linguagens de programação e SDKs</i> .....	35
3.3.2.1	Java.....	35
3.3.2.2	Firestore auth .....	36
3.3.2.3	Google Vision .....	36
3.3.2.4	Facebook Android SDK.....	36
3.3.2.5	Retrofit.....	36
3.3.2.6	Scotkyab Secure Preferences.....	37
3.3.2.7	Picasso.....	37
3.3.2.8	Spring Boot.....	37
3.3.2.9	Weka .....	38
3.3.3	<i>Banco de dados</i> .....	38
3.4	FUNCIONALIDADES DA APLICAÇÃO .....	39
3.4.1	<i>Front-end</i> .....	39
3.4.2	<i>Back-end</i> .....	39
3.5	IMPLEMENTAÇÃO DOS SERVIÇOS .....	39
<b>4</b>	<b>PROTÓTIPO E RESULTADOS.....</b>	<b>42</b>

4.1	PROTÓTIPO .....	42
4.2	RESULTADO .....	44
<b>5</b>	<b>CONCLUSÃO.....</b>	<b>55</b>
	<b>TRABALHOS FUTUROS.....</b>	<b>56</b>
	<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>57</b>

## 1 INTRODUÇÃO

O mundo virtual é um grande aliado do desenvolvimento acadêmico. A difusão de computadores pessoais portáteis – *tablets*, *smartphones*, *notebooks* – bem como a grande quantidade de conteúdos acadêmicos disponibilizados na internet são de grande utilidade para a formação profissional. O uso de recursos multimídia tornou-se uma constante no processo de oferta e procura de informações acadêmicas. Imagens, sejam estáticas (fotos, tabelas, gráficos, etc.), sejam dinâmicas (vídeo aulas, palestras, entrevistas, etc.), são formas recorrentes de facilitar o acesso a conteúdos de diversas áreas do conhecimento.

Além dos conteúdos disponibilizados por meio da internet, os estudantes também produzem outros materiais que, durante o processo de formação, serão utilizados como fonte de pesquisa e referência: aulas são capturadas em áudios, vídeos e fotos; textos são produzidos tanto em papel quanto em arquivos digitais, dentre outros recursos. Tendo em vista estas possibilidades, é perceptível que há a necessidade de armazenamento e organização para os inúmeros registros feitos, principalmente em sala de aula, os quais, muitas vezes, correm o risco de serem deletados por falta de atenção ou, até mesmo, esquecidos por não terem sido arquivados de maneira correta.

Devido ao grande volume de informação que permeia a atividade de aprendizado, é cada vez mais difícil e complexa a concatenação dos dados acumulados na trajetória acadêmica. Frequentemente, os estudantes, embora guardem muitos apontamentos, não são capazes de localizá-los com rapidez e perdem muito tempo procurando os documentos que precisam. Por isso, o motivo deste trabalho é justamente tentar solucionar este problema tão corriqueiro da vida estudantil: reunir informações a respeito dos conteúdos de modo ordenado a fim de agilizar o processo de procura e facilitar o acesso aos conteúdos, sem a necessidade de mergulhar em um “mar de papéis avulsos” ou passar horas remexendo em inúmeros arquivos de um HD.

## 1.1 Justificativa

É perceptível que os *smartphones* e *tablets* fazem parte do dia a dia da população, não só com a finalidade de lazer e entretenimento, mas, são empregados, especialmente, em estudos e trabalho. Isso deve-se, sobretudo, à agilidade nas trocas de informações possibilitadas pelos aplicativos instalados nos dispositivos móveis. Desse modo, praticamente em todos os ambientes sociais, tais aparelhos estão presentes; e as salas de aula – da educação básica ao ensino superior – não poderiam ficar de fora desse novo padrão que se instalou na sociedade brasileira.

A quantidade de estudantes e professores que usam internet pelo celular cresceu nos últimos anos. Isso é evidente tanto em escolas públicas quanto particulares, como demonstra os dados publicados pelo Comitê Gestor da Internet no Brasil, em que foi constatado que 79% dos colégios públicos e 84% das escolas particulares têm acesso à internet pelo celular (MACIEL, 2015).

Porém, os aparelhos portáteis, que estão presentes nos colégios, têm tomado outras proporções: os professores tendem a substituir o quadro e o giz por aulas preparadas em apresentações de *Power Point* e, em alguns casos, por lousas interativas em que as informações a serem transmitidas podem estar gravadas em *smartphones* ou *tablets*, por exemplo. Mas, não só os docentes, como também os alunos utilizam os recursos possibilitados por seus dispositivos, tal como fotografar os conteúdos registrados no quadro, armazenar documentos referentes às disciplinas, entre outras possibilidades.

## 1.2 Problema

A dupla jornada faz parte da vida dos brasileiros, principalmente dos jovens que se encontram divididos entre trabalho e estudo. É evidente que tal aspecto se tornou necessário para esta sociedade, tendo em vista que muitos querem se especializar, mas também precisam contribuir com as despesas domiciliares.

Ademais, o mercado de trabalho está cada vez mais exigente, com demanda não só de mão de obra qualificada, como também de pessoas com experiência.

Tal problemática é ressaltada em uma reportagem da CBN, com o título “Mão de obra qualificada perde espaço no mercado de trabalho”, publicada em fevereiro de 2016 – a qual aponta os dados do IBGE, referentes a janeiro de 2016, em que 7,3% de indivíduos com ensino superior não possuíam emprego –, trazendo a opinião de Leandro Reis, diretor de *marketing* do Salão do Estudante, sobre esse cenário: ele destaca a importância da formação acadêmica, mas deixa claro que “a pessoa tem que saber equilibrar os estudos com o trabalho”.

Consequentemente, a falta de tempo e cansaço físico daqueles que se desdobram em dupla jornada podem não permitir um bom rendimento nas aulas diárias, as quais estão repletas de informações acerca dos conteúdos necessários para a conclusão dos estudos. Estes são problemas que influenciam na retenção de conhecimentos pelos alunos, os quais, em certos momentos, deveriam registrar as informações anotadas no quadro, mas não o fazem, seja pela falta de tempo hábil na sala de aula, seja pela exaustão causada por uma rotina acelerada.

Em decorrência disso, os materiais convencionais de estudo, em especial cadernos, fichários e, até mesmo, *notebooks*, usados para anotações, não se mostram tão funcionais na realidade contemporânea. A escrita e a digitação estão caindo em desuso devido à economia de tempo almejada pelas pessoas, sendo substituídas pelo registro de informações e de conteúdos por meio de fotografias tiradas pelos *smartphones* ou *tablets*. Contudo, a falta de ferramentas que auxiliem na organização dessas imagens de modo rápido e seguro pode dificultar o processo de aprendizagem do aluno, pois a desordem dos documentos impossibilita o reconhecimento de apontamentos feitos anteriormente.

Considerando os dados expostos nesse capítulo, infere-se que a falta de uma ferramenta que proporcione a organização e o armazenamento específicos para os registros feitos em imagens de conteúdos ministrados em sala de aula representa uma questão problemática que pode ser facilitada por meio do sistema a ser desenvolvido neste trabalho. Posto isto, almeja-se colocar em prática tal projeto, visando melhorias na qualidade de vida dos universitários.

### 1.3 Objetivos gerais

Dada a dificuldade de organizar o grande volume de informações acumuladas, pretende-se desenvolver um sistema capaz de indexar as informações, possibilitando a pesquisa objetiva, facilitando, assim, a consulta dos dados acumulados, além de oferecer maior organização, praticidade e propiciar mais agilidade aos usuários em suas consultas.

### 1.4 Objetivos específicos

O sistema desenvolvido neste trabalho possui cinco objetivos específicos, sendo eles:

- Organizar o conteúdo acumulado, agrupando-os por disciplina e indexando as datas de inclusão;
- Agilizar as atividades acadêmicas tornando as aulas mais dinâmicas e objetivas;
- Facilitar a busca das informações acumuladas;
- Centralizar o material de estudo em um único banco de dados de informações.

### 1.5 Organização do trabalho

Esse trabalho está organizado em 5 capítulos, além deste, nos quais serão abordados, respectivamente, os seguintes aspectos:

- Capítulo 2 – Fundamentação Teórica: Apresentação da base teórica para a compreensão geral do objeto de pesquisa do projeto.
- Capítulo 3 – Desenvolvimento: Detalhamento do processo de elaboração do sistema e exposição de aplicação das teorias relevantes para tal trabalho.
- Capítulo 4 – Protótipo e Resultados: Exposição do protótipo do sistema e respostas esperadas/obtidas.
- Capítulo 5 – Conclusão: Apresentação final dos resultados obtidos.

## 2 FUNDAMENTAÇÃO TEÓRICA

A fim de respaldar as ideias que deram suporte para a criação do sistema desenvolvido, serão apresentados, neste capítulo, a fundamentação teórica e os conceitos necessários para a compreensão do trabalho desenvolvido.

### 2.1 Dispositivos móveis

Os populares *smartphones* e *tablets* são comumente denominados dispositivos móveis, pois caracterizam-se como pequenos computadores que podem ser carregados com facilidade para qualquer lugar. Esses dispositivos possuem telas pequenas combinadas com minis teclados e, muitas vezes, os *displays* são sensíveis ao toque, o que ficou consagrado como *touchscreen*. Além disso, semelhante aos computadores convencionais, tais aparelhos móveis estão munidos de sistemas operacionais, (os mais utilizados, no Brasil, são Android e IOS) os quais executam aplicativos que podem ser programados para realizarem inúmeras tarefas – dentre elas, tirar e editar fotos, acessar redes sociais, baixar e ouvir músicas, etc.

A facilidade de realizar diversas atividades ao mesmo tempo é proporcionada por tais dispositivos móveis, por exemplo, é possível ouvir música, conversar com pessoas que estão em outros lugares e ainda usar o GPS, tudo ao mesmo tempo. Essa simplicidade em executar multitarefas conquistou todo o mundo, inclusive os brasileiros. Os dados do Pnad (Pesquisa Nacional por Amstras de Domicílios), divulgados pelo IBGE em abril de 2016, mostram que 80% das casas possuem celulares e que estes são mais utilizados para acessarem à internet do que os computadores.

### 2.2 Uso da tecnologia em salas de aula

Os pedagogos Tatiana Mousquer e Carlos Oberdan Rolim, no artigo *A Utilização de Dispositivos Móveis como Ferramenta Pedagógica Colaborativa na*

*Educação Infantil*, publicado em 2013, afirmam que “hoje a maioria das crianças crescem manuseando tecnologia, habilidade que lhes confere acesso a um universo ilimitado de saberes e informações” (p. 2). Sendo assim, o uso de diversos dispositivos móveis faz, cada dia mais, parte da rotina escolar e tais aparelhos, aos poucos, configuram-se como materiais não só de apoio pedagógico, como também, utensílios necessários para o efetivo trabalho em sala de aula realizado pelo professor junto aos alunos.

Em uma matéria publicada pelo jornal Folha de São Paulo, em outubro de 2015, um novo comportamento dos estudantes brasileiros é retratado: as cópias no caderno de anotações feitas pelos professores em lousas estão sendo substituídas por fotos tiradas em celulares. Algumas das justificativas dadas pelos alunos para tal substituição são: nem sempre dá tempo de copiar tudo, às vezes “dá preguiça de copiar” ou fazem isso quando a tarefa é passada no final das aulas, entre outras (GRAGNANI, 2015).

Assim, os registros feitos em celulares pelos alunos são aglomerados nas pastas de seus *smartphones* ou *tablets* sem nenhum critério relevante para o ordenamento de tais informações. A simples organização das “galerias” dos dispositivos em ordem cronológica não é suficiente para o armazenamento de tantas informações que vão se acumulando no decorrer do ano/semestre letivo.

### 2.3 Inteligência artificial

As correntes de pensamento que se cristalizaram em torno da IA já estavam em gestação desde os anos 40, devido à Segunda Guerra Mundial, em que surgiu a necessidade de desenvolver tecnologia para impulsionar a indústria bélica. No entanto, oficialmente, a primeira vez que o termo inteligência artificial foi utilizado foi em 1956 por John McCarthy, Marvin Minsky, Nathaniel Rochester e Claude Elwood Shannon em uma conferência na Faculdade de Dartmouth, localizada em Hanover, no estado americano de New Hampshire, como uma proposta de pesquisa de verão sobre o tema Inteligência Artificial (MCCARTHY; MINSKY; ROCHESTER; SHANNON, 1955).



Desde seus primórdios, a IA gerou polêmica, a começar pelo seu próprio nome, considerado presunçoso por alguns, até a definição de seus objetivos e metodologias. O desconhecimento dos princípios que fundamentam a inteligência, por um lado, e dos limites práticos da capacidade de processamento dos computadores, por outro, levou periodicamente a promessas exageradas e às correspondentes decepções.

No final dos anos 50 e início dos anos 60, os cientistas Newell, Simon, e J. C. Shaw introduziram o processamento simbólico. Ao invés de construir sistemas baseados em números, eles tentaram construir sistemas que manipulassem símbolos. A abordagem era poderosa e foi fundamental para muitos trabalhos posteriores.

Desde então, as diferentes correntes de pensamento em IA têm estudado formas de estabelecer comportamentos “inteligentes” nas máquinas. Portanto, o grande desafio das pesquisas em IA, desde a sua criação, pode ser sintetizado com a indagação feita por Minsky em seu livro *Semantic Information Processing*, há mais de cinquenta anos: “Como fazer as máquinas compreenderem as coisas?” (MINSKY, 1968).

Ademais, com a finalidade de criação de sistemas inteligentes, é preciso que sejam levadas em consideração duas linhas de pesquisa, a linha simbólica e a linha conexionista.

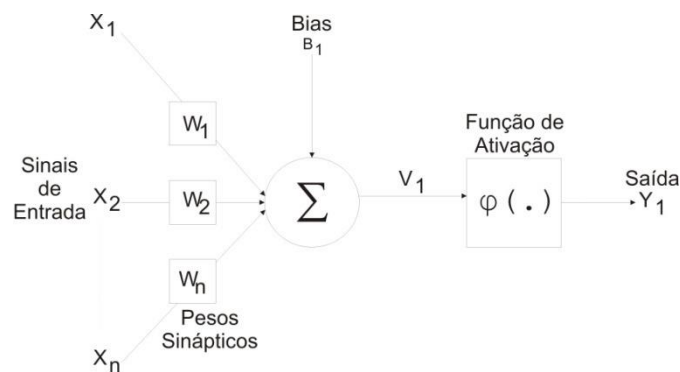
Na concepção simbólica, idealizada por McCarthy e Newell, a manipulação de símbolos de fatos especializados acerca de um domínio restringido tornou-se modelo vigente para a construção de sistemas inteligentes, consolidada pelo Expert System (SE) a partir da década de 70.

Já na visão conexionista, apresentada por McCulloch e Pitts, em 1943, o objetivo era modelar a inteligência humana por meio da demonstração de neurônios e suas interligações, isto é, composições do cérebro humano. Porém, por muito tempo, essa linha de pesquisa não teve grande notoriedade. Apenas com o surgimento de microprocessadores acessíveis, essa concepção tornou-se relevante para possibilitar a criação de máquinas conectadas por meio de muitos

microprocessadores. Tais empreendimentos deram origem ao campo de redes neurais artificiais.

### 2.3.1 Redes neurais artificiais

As Redes Neurais Artificiais (RNA), também conhecidas como métodos conexionistas, são inspiradas nos estudos acerca da maneira como se organiza e como funciona o cérebro humano. Dessa forma, foi criado o neurônio artificial, que, assim como o neurônio biológico, possui um ou mais sinais de entrada, que pode receber informações de sensores ou de outros neurônios por meio de uma conexão chamada sinapse. Após o recebimento da informação, o neurônio efetua uma soma ponderada das informações auferidas juntamente com os respectivos ganhos como fatores excitatórios, inibitórios ou de ponderação e envia-os para uma única saída, chamada de axônio (Figura 1). A junção de vários neurônios artificiais é capaz de gerar uma grande rede, chamada rede neural (RAUBER, 2005).



*Figura 1 - Neurônio Artificial*

Uma RNA tem a habilidade de aprender com seu ambiente por meio da experiência, visando à melhoria de desempenho. Tal habilidade é gerada a partir do treinamento, isto é, um processo iterativo em que são ajustados valores salvos na sinapse, que indicam a influência na saída, também chamados de pesos sinápticos (LUGER; A STUBBLEFIELD, 1998). Caso o resultado da soma dos sinais de entrada e seus respectivos pesos sinápticos atinjam um valor predeterminado, calculado na função de ativação, também conhecido como limiar lógico, o sinal será transferido para o neurônio seguinte.

Existem diversos tipos de treinamento para aprendizado das RNAs, os quais foram divididos em dois padrões: supervisionado e não supervisionado.

No treinamento supervisionado, os pesos sinápticos terão seus valores iniciais atribuídos de forma aleatória e são ajustados pela rede neural na próxima interação ou ciclo. Nesse padrão de aprendizado, as redes neurais artificiais devem ser treinadas anteriormente e o treinamento é considerado completo quando a RNA atinge uma performance satisfatória.

Ao contrário do aprendizado supervisionado, o não supervisionado não aceita interferência externa no ajuste dos pesos sinápticos, pois possui particularidades de auto nivelção, analisando as tendências e/ou regularidades dos sinais de entrada agrupando-os de maneira a atender as necessidades da RNA.

#### 2.3.1.1 Redes neurais artificiais recorrentes

As redes recorrentes são um poderoso conjunto de algoritmos de redes neurais artificiais especialmente úteis para o processamento de dados sequenciais, como som, dados de séries temporais ou linguagem natural. Embora algumas formas de dados, como imagens, não pareçam ser sequenciais, elas podem ser entendidas como sequências quando alimentadas em uma rede recorrente.

Assim como as redes recorrentes processam a escrita manual, convertendo cada imagem em uma letra e usando o início de uma palavra para adivinhar como essa palavra terminará, então as redes podem tratar parte de qualquer imagem como letras em uma sequência. Uma rede neural que percorre uma imagem grande pode aprender a partir de cada região, o que as regiões vizinhas têm a probabilidade de ser.

Uma estrutura utilizada na inserção de memória na RNA é a recorrência. Esta opção é mais indicada para sistema com maiores complexidades, uma vez que modifica toda a estrutura da rede neural, exigindo a presença de mais ligações e pesos, aumentando, conseqüentemente, o tempo de processamento computacional. Uma rede que possui recorrência é chamada de rede neural recorrente, sendo

caracterizada pela realimentação dos neurônios de uma determinada camada pelas suas saídas de um estado ou época anterior, como ilustrado na figura 2.

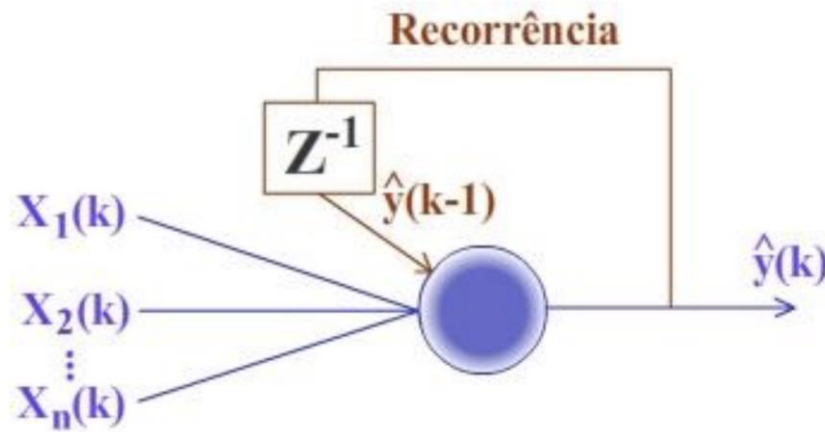


Figura 2 - Recorrência aplicada a um neurônio

#### 2.4 Reconhecimento ótico de caracteres

Reconhecimento ótico de caracteres (OCR) é uma tecnologia que permite reconhecer caracteres de texto em imagens, transformando-os em texto editável (CAESAR; GLOGER; MANDLER, 1993).

O reconhecimento de caracteres é um subconjunto da área de reconhecimento de padrões e foi ele que estabeleceu as bases e a motivação para tornar o reconhecimento de padrões e a análise de imagens campos individuais de interesse da ciência (SENIOR; ROBINSON, 1998). A importância das pesquisas nesta área não se baseia apenas em aplicações comerciais, mas também no reconhecimento de padrões complexos em 2D e 3D para sistemas de visão robótica e computacional.

Os sistemas de OCR possuem algumas características e funcionalidades interessantes como:

- A utilização de um sistema OCR para modificação de apenas uma parte de um texto que está em formato de papel, modificando-o para um texto editável;

- O reconhecimento de caracteres padronizados de escritas de outros idiomas tais como, japonês, chinês e árabes, transcrevendo-as para língua portuguesa;
- Na segmentação de caracteres, e dos métodos usados para a extração de características e o reconhecimento dos caracteres das placas de veículos;
- No reconhecimento de escritas manuscritas da língua portuguesa ou demais línguas.

Dessa forma, verifica-se que um sistema de OCR pode ser utilizado para diversas funcionalidades e ramos de atividade, dependendo da forma que ele seja implementado (PAVLIDIS, 1982). Sendo assim, é possível identificar e classificar um OCR através do algoritmo que seja implementado em determinado contexto.

A Tecnologia de Reconhecimento de Caracteres pode ser dividida em dois tipos:

- OCR (*Optical Character Recognition*) - Reconhecimento de Caracteres Ópticos, melhor entendido como reconhecimentos de caracteres impressos.
- ICR (*Intelligent Character Recognition*) - Reconhecimento de Caracteres Inteligentes, melhor entendido como reconhecimento de caracteres impressos, manuscritos, código de barra e marcações.

Desse modo, os sistemas de ICR são, de certa forma, um tipo de OCR avançado capaz de reconhecer diferentes formas de escritas manuscritas. Os sistemas de ICR são aplicados na captura de dados de formulários, reconhecendo caracteres em formulários já impressos anteriormente (formulários pré-existent), ou desenhados através da ferramenta de desenho.

Até agora os sistemas de OCR para escritas manuscritas cursivas têm sido usados para ambientes em que as frases são curtas, o conteúdo controlado, e há possibilidade de validar o resultado com outra informação do mesmo documento. Exemplo, o valor por extenso nos cheques, formulários de concursos, segmentação de caracteres numéricos para endereçamento postal.

O reconhecimento de caracteres cursivos não é algo tão simples, pois trata de um tipo de caractere sem restrição, que, na maioria das vezes, estão conectados. Definitivamente é um dos mais complexos e mais difíceis de ser implementado. Até hoje é muito difícil de conseguir qualquer técnica de desempenho seguramente satisfatório, uma vez que não se podem limitar os inúmeros parâmetros, formas e características individuais da escrita cursiva.

O desempenho de um sistema automático de reconhecimento depende da qualidade dos documentos nas suas formas originais e digitais. Usam-se processos para compensar uma qualidade pobre nos originais e nas imagens após a captura e digitalização.

Estes processos incluem realces, remoção de linhas, de sublinhados e de ruídos, entre outros. Os problemas geralmente relacionados com a qualidade e a dificuldade de tratamento da imagem e do texto segundo Berthold são: ruído, distorção, variação de estilo, translação, escala, rotação, texturas e traços (SUEN; BERTHOD; MORI, 1980).

## 2.5 Mineração de dados

Mineração de dados (*Data Mining* ou DM) faz parte de um processo chamado busca de conhecimento em banco de dados (Knowledge Discovery in Database), que possui uma metodologia específica para preparação, exploração, interpretação e assimilação de conhecimento. Entretanto, se tornou mais conhecida do que o processo KDD, pois a *data mining* é a etapa na qual são aplicados métodos de busca de conhecimento.

De acordo com Berry e Linoff (2004), o conceito de mineração de dados é “the exploration and analysis of large quantities of data in order to discover meaningful patterns and rules”, ou seja, é um processo que utiliza várias consultas e análise de dados, em grande quantidade, para extrair informações úteis, tendências, regras ou padrões interessantes.

Ademais, o chamado processo não-trivial de identificação de padrões, isto é, o processo de descoberta de conhecimento em bancos de dados pode ser dividido nas seguintes etapas, de acordo com Fayyad, Piatetsky-shapiro e Smyth (1996):

- I. Entendimento do domínio da aplicação, das competências prévias importantes e das intenções dos usuários finais;
- II. Escolha de dados e atributos relevantes para a elaboração de um grupo de dados com a finalidade de serem utilizados no processo de descoberta;
- III. Organização e pré-processamento de dados, retirada de ruídos e inadequações, definição de procedimentos com dados incompletos, normalização ou indexação de sequências temporais, etc.;
- IV. Diminuição e nova representação dos dados em outra organização de coordenadas, caso seja preciso;
- V. Seleção da atividade de mineração de dados, levando em consideração a finalidade geral do processo;
- VI. Definição dos algoritmos de mineração, fundamentado no propósito global e na estrutura exigida dos dados;
- VII. Mineração dos dados propriamente dita: investigação de modelos de interesse, empregando os dados escolhidos e algoritmos;
- VIII. Análise dos efeitos gerados pela mineração de dados, além da verificação de paradigmas, regras, entre outras informações descobertas por meio da mineração;
- IX. Estabelecimento e apreciação dos referências obtidas, documentação e criação de relatório, além da solução de problemas com informações já conhecidas.

A seguir, apresenta-se as categorias a que pertencem algumas técnicas empregadas para criar os paradigmas utilizados em mineração de dados (Figura 3), segundo Kantardzic (2011).

- Classificação: reconhecimento de função preditiva, a qual categoriza os dados em classes diretas, já definidas anteriormente.

- Regressão: reconhecimento de função preditiva de modo semelhante ao da classificação, porém, visando o cálculo de valor numérico real.
- Agrupamento ou *Clustering*: reconhecimento de grupos naturais de dados semelhantes entre si. Ao contrário das categorias de classificação e regressão, no *Clustering* os algoritmos constituem grupos naturais de acordo com alguma medida, a fim de serem processados como elementos de uma mesma divisão.
- Sumarização: reconhecimento de uma representação sólida e acessível para os dados, mesmo que estes sejam imprecisos.
- Modelagem de dependência: reconhecimento de paradigma que detalha a dependência de valores de atributos de um grupo de dados com outros valores já conhecidos. Também é considerada modelagem de dependência as técnicas de procura de regras de associação.
- Detecção de mudança ou inadequações (*outliers*): reconhecimento e revelação de dados que não procedem consoante a um padrão aceitável. Usada para identificar mudanças e inadequações em um grupo específico de dados ou em todos eles.



Figura 3 - Passos DM



### **3 DESENVOLVIMENTO**

O projeto em questão buscar desenvolver um sistema que permita a divisão de conteúdos em categorias, proporcionando o armazenamento dessas informações em nuvem, para posteriormente filtrar e indexar esses dados visando compartilhamento de conteúdo similar entre os usuários.

#### **3.1 Projeto**

O sistema desenvolvido consiste em um aplicativo, desenvolvido para celulares com sistema operacional Android, no qual o usuário poderá se cadastrar, adicionar matérias com seus respectivos períodos, tirar fotos que serão automaticamente salvas nas matérias previamente cadastradas, além de compartilhar esse conteúdo com outros usuários.

Para completo funcionamento, também é necessária uma parte do sistema funcionando em um servidor para armazenamento e processamento dos dados do aplicativo móvel.

##### **3.1.1 Arquitetura**

Os serviços e componentes que compõem o sistema estão estruturados de acordo com o diagrama de arquitetura mostrado na figura 4.

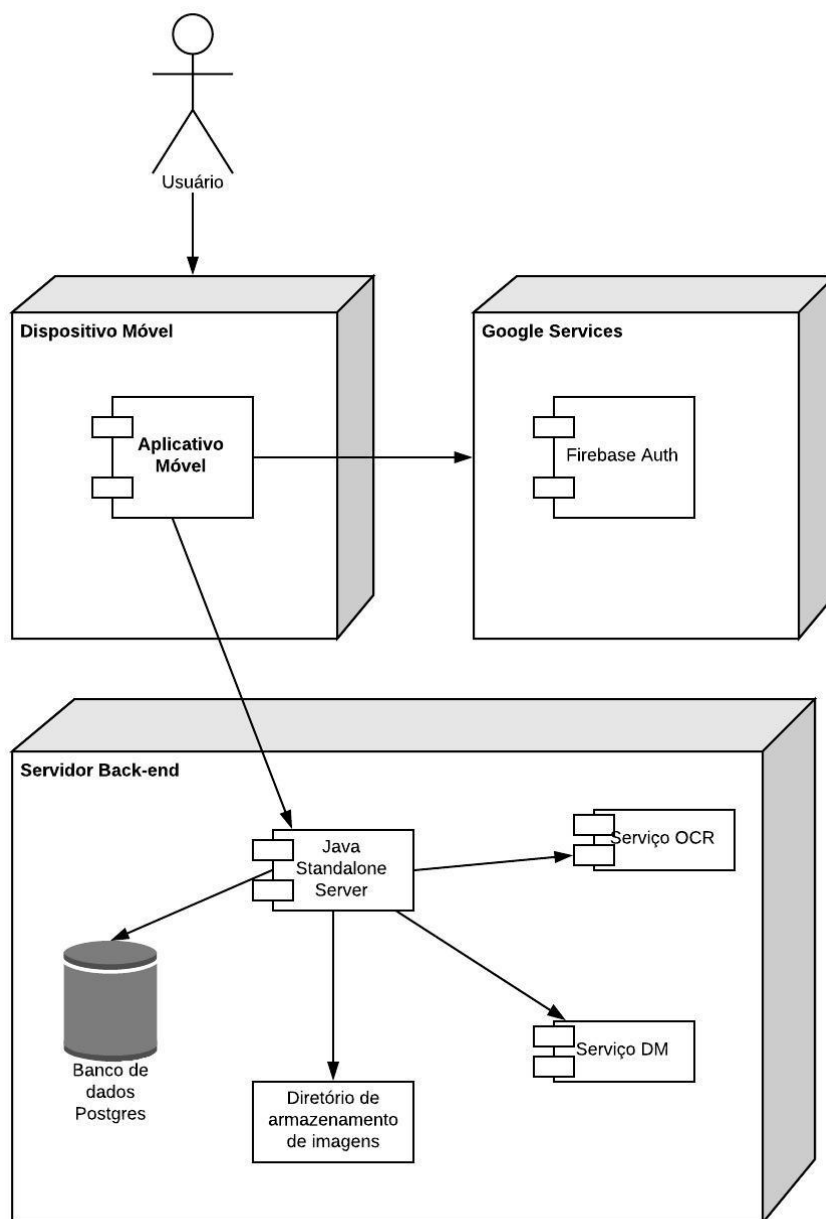


Figura 4 - Diagrama de arquitetura

Nessa arquitetura proposta, o dispositivo móvel pode comunicar somente com o servidor de autenticação disponibilizado pelo Google chamado Firebase Auth e com um serviço standalone em Java situado em um servidor *back-end* na nuvem.

Esse serviço *standalone* é responsável por delegar as responsabilidades entre outros serviços e processar os dados enviados pelo aplicativo.

## 3.2 Análise e modelagem da aplicação

Este capítulo irá tratar sobre os diagramas que vão orientar o desenvolvimento da aplicação, explicando as classes e suas relações, e a descrição das entidades no modelo do banco.

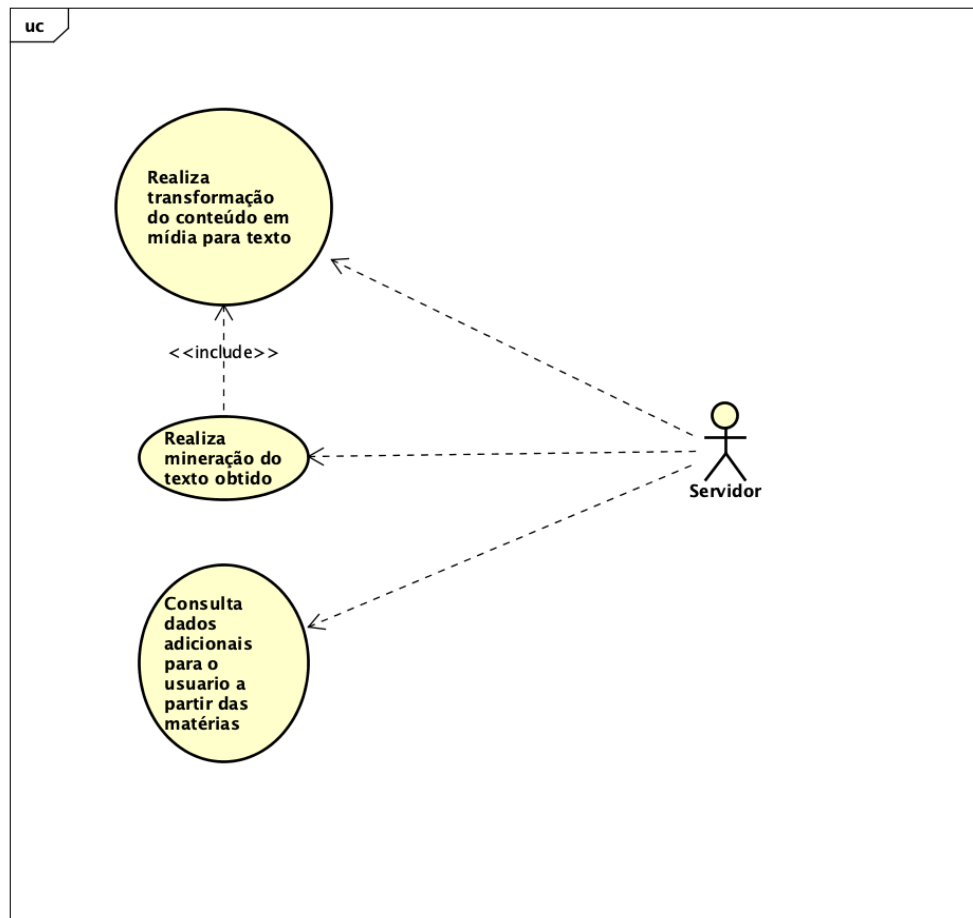
### 3.2.1 *Unified modeling language*

*Unified modeling language* (UML) é uma linguagem que foi criada para auxiliar a visualização dos produtos desenvolvidos utilizando técnicas de notação gráfica para criação de modelos visuais. Ela possui vários diagramas cujos são divididos em estruturais e comportamentais, os quais visam obter todos os aspectos e visões do sistema.

#### 3.2.1.1. Diagrama de caso de uso

Diagrama de caso de uso está incluído como um dos diagramas comportamentais da UML e é utilizado para descrever as principais funcionalidades do sistema e a integração dos atores, que podem ser usuários ou outros sistemas, com as funcionalidades. Ele é muito importante para as fases de levantamento e análise de requisitos.

Na Figura 5 é possível verificar os casos de uso do agente servidor.



*Figura 5 - Diagrama de caso de uso manter servidor*

A figura 6 mostra o maior diagrama de caso de uso do sistema, os casos de uso do agente usuário, que irá efetuar a maior parte das ações do sistema.

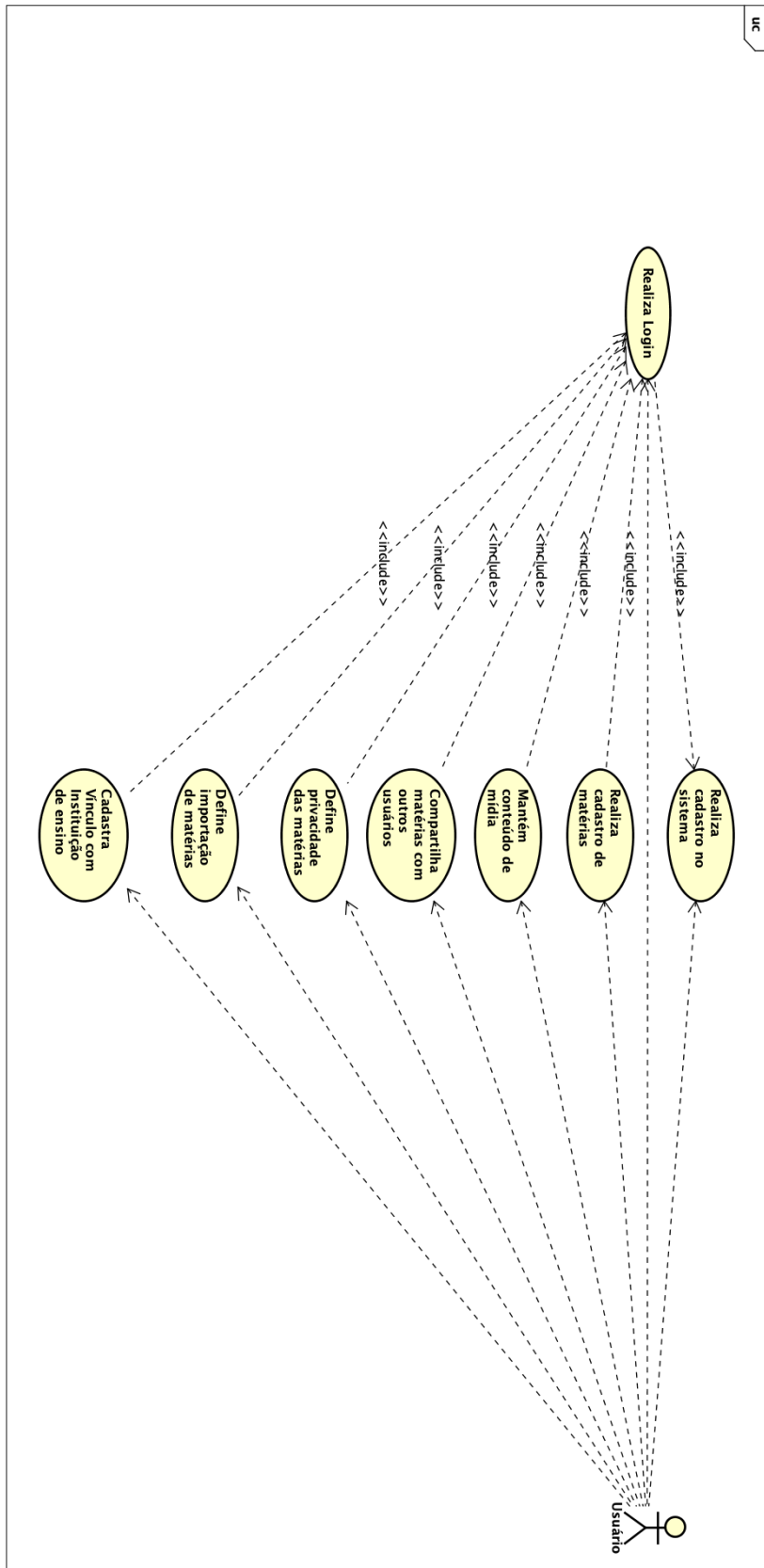
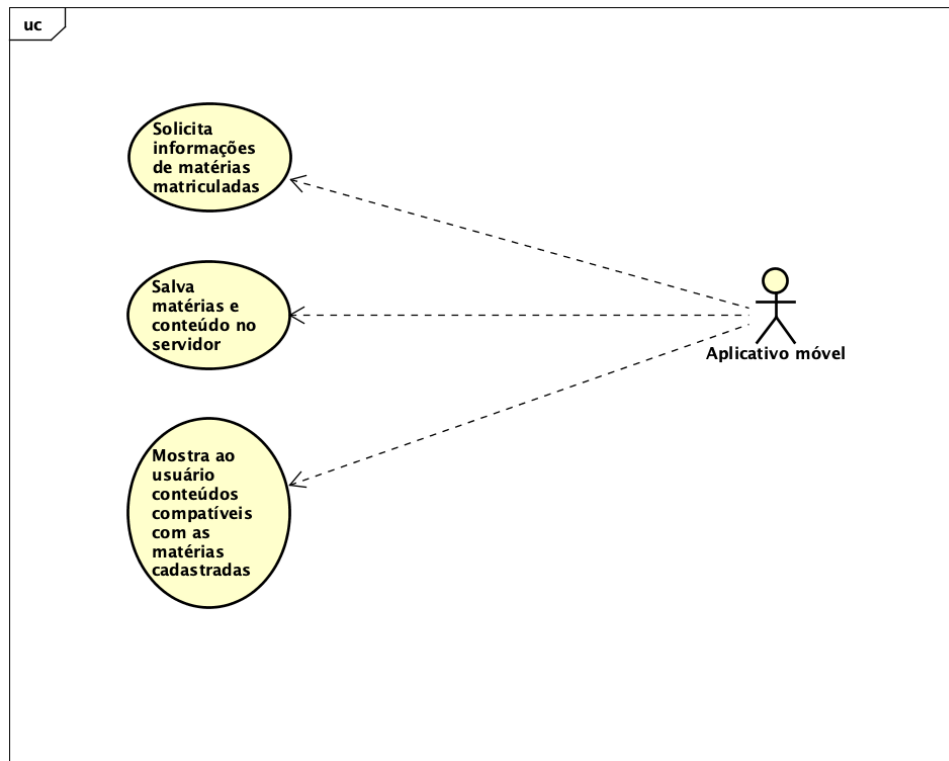


Figura 6 - Diagrama de caso de uso mantém usuário

Na imagem abaixo (Figura 7) é possível verificar os casos de uso possíveis para o agente aplicativo móvel:



*Figura 7 - Diagrama de caso de uso mantém aplicativo móvel*

#### 3.2.1.2. Diagrama de classes

É o diagrama mais utilizado na UML, pois mostra os relacionamentos das classes juntamente com seus atributos e métodos e por isso, serve de auxílio aos outros diagramas.

Para desenvolvimento do sistema proposto, foi criado o diagrama possuindo 8 classes (Figura 8).

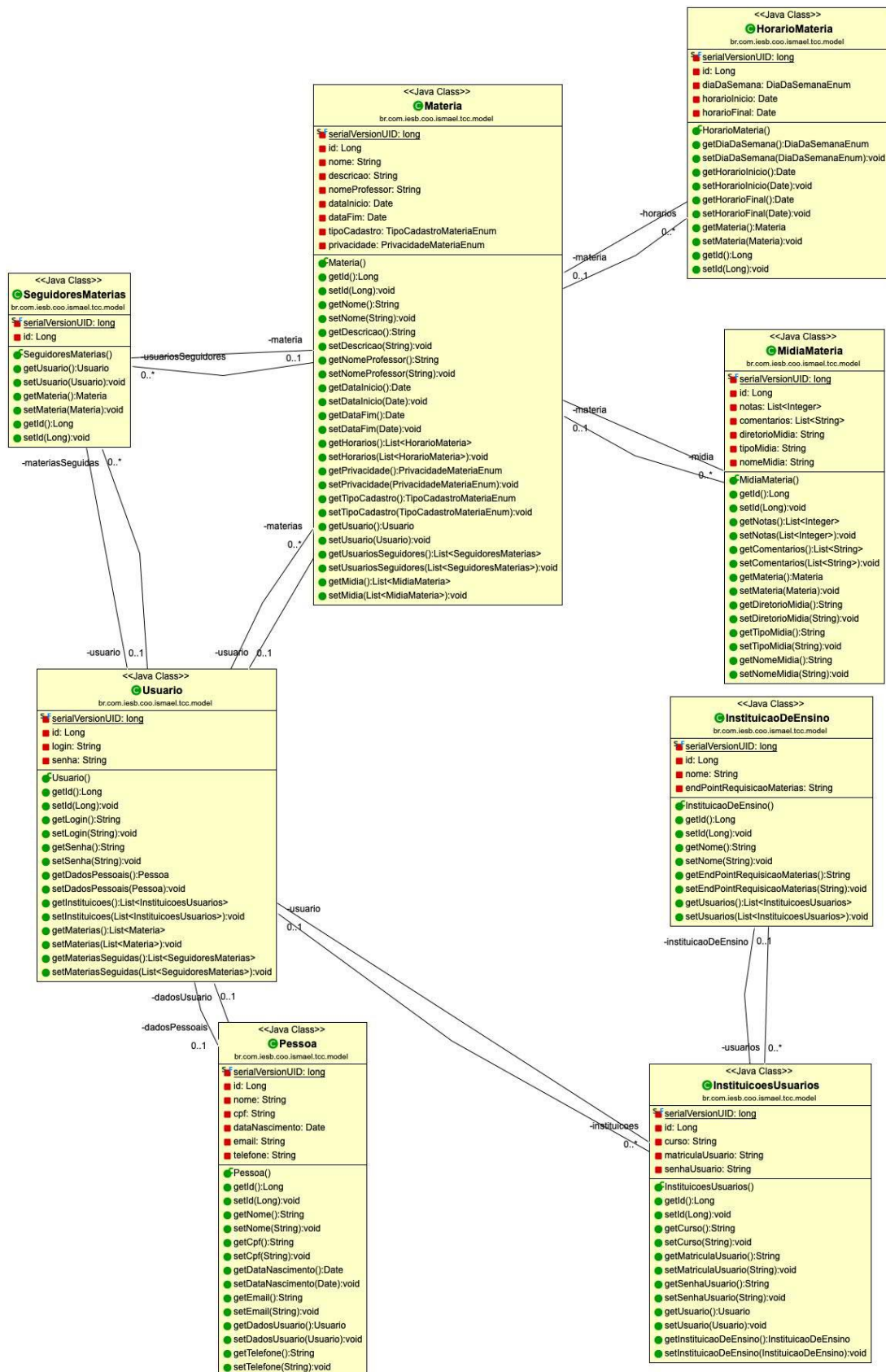


Figura 8 - Diagrama de classes

### 3.2.2 Banco de dados

Um banco de dados é um sistema cujo principal objetivo é armazenar informações e permitir que os usuários, quando quiserem, busquem e atualizem essas informações. A maioria dos *softwares* que necessitam guardar algo em algum lugar utiliza banco de dados (DATE, 1989).

O banco de dados representa uma parte muito importante, se não a mais relevante, do *software*/empresas. A perda de dados, em alguns casos, paralisa completamente o programa, por exemplo, um *software* de consulta de saldo depende cem por cento do banco de dados para cumprir com o seu objetivo principal e, se os dados se perderem, ele para de cumprir esse objetivo.

Banco de dados pode parecer simples, porém, quando os dados são de extrema importância, essa simplicidade não existe mais, é tudo muito complexo, necessitando de profissionais de qualidade para controle e manutenção da base de dados (FEITOSA, 2013).

#### 3.2.2.1 Diagrama entidade relacionamento

É uma representação gráfica de um modelo de banco de dados relacional, que mostra como objetos ou conceitos se relacionam entre si dentro de um sistema.

Segue abaixo diagrama entidade relacionamento:



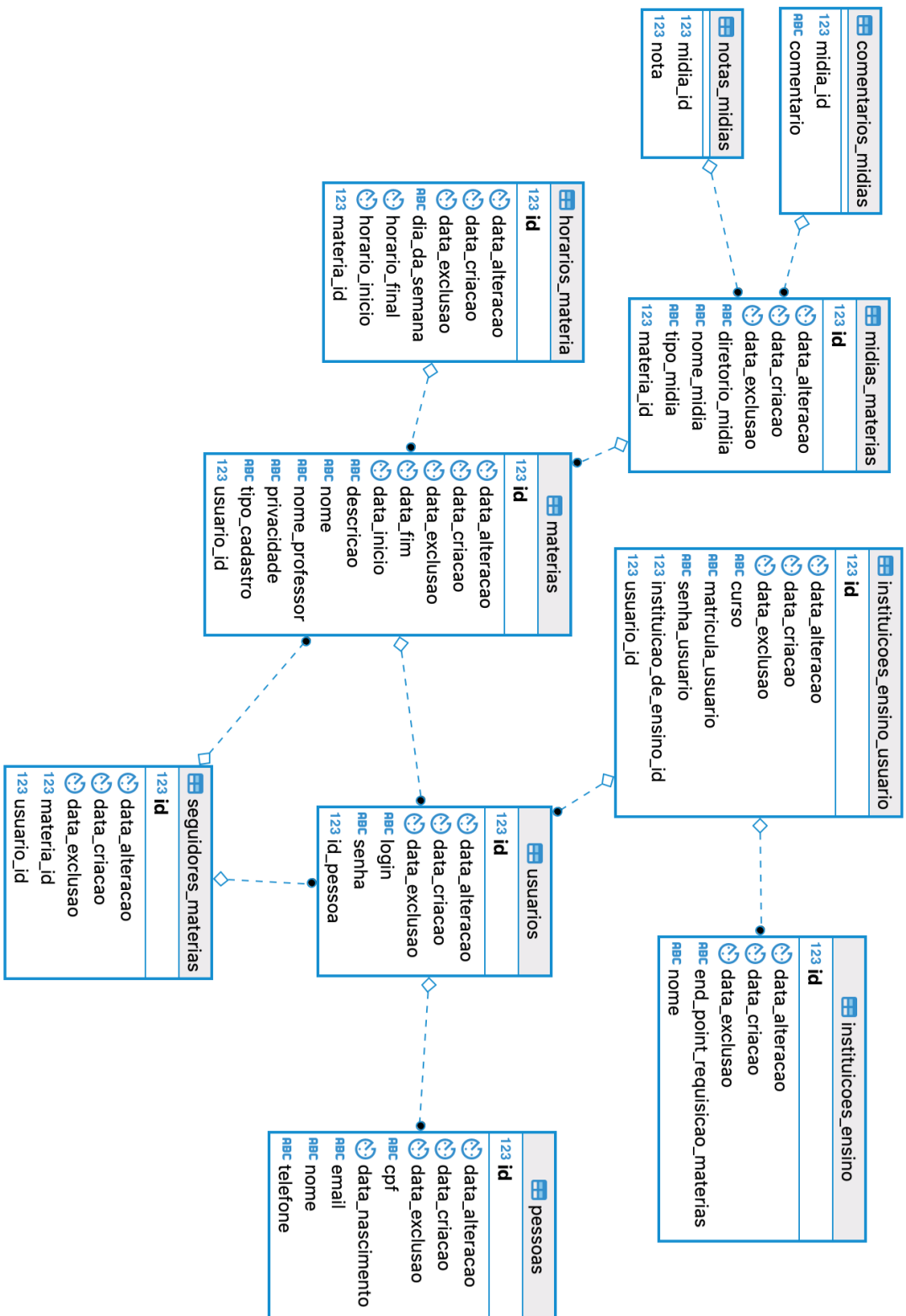


Figura 9- Diagrama entidade relacionamento

### 3.3 Tecnologias utilizadas

Durante a criação do sistema, em algumas situações, foi necessária a aplicação de diversas ferramentas em áreas específicas, para a comunicação entre o usuário e o aplicativo e, entre o aplicativo e o servidor do sistema.

#### 3.3.1 Ferramentas de desenvolvimento

A seguir serão descritas as ferramentas que foram necessárias para o desenvolvimento do projeto proposto.

##### 3.3.1.1 Eclipse

Eclipse é um ambiente de desenvolvimento integrado (IDE) voltado para desenvolvimento utilizando a linguagem Java. O projeto eclipse foi iniciado na IBM e posteriormente doado para comunidade como *software* livre (ECLIPSE, 2018).

Sua utilização nesse projeto foi importantíssima durante o desenvolvimento do serviço *standalone* (*back-end*) que foi desenvolvido em Java.

##### 3.3.1.2 Astah UML

Astah é um programa para modelagem UML desenvolvido pela empresa Change Vision (UML, 2018). Foi utilizado na criação dos diagramas de caso de uso.

##### 3.3.1.3 DBeaver

Durante o desenvolvimento do sistema, foi necessária a verificação dos dados que estavam sendo gravados e atualizados no banco de dados. Devido essa necessidade, foi utilizado o software DBeaver, que é uma ferramenta multiplataforma de administração de banco de dados (DBEAVER, 2018).

#### 3.3.1.4 Android Studio

Para desenvolvimento da aplicação móvel, a IDE primária do Google para desenvolvimento nativo na linguagem Android foi necessária.

#### 3.3.2 Linguagens de programação e SDKs

Dentre as várias linguagens de programação e SDKs existentes no mercado, neste capítulo serão descritas as escolhidas para compor as funcionalidades do sistema.

##### 3.3.2.1 Java

A proposta será desenvolvida em uma linguagem orientada a objeto, neste caso JAVA. Flanagan (2006) discorre observando que todos os programas Java utilizam objetos e o tipo de um objeto é definido pela sua classe ou interface. Cada programa Java é definido como uma classe, e programas não triviais normalmente incluem algumas definições de classes e de interfaces.

Flanagan (2006) também explica o conceito de classe e objeto. Ele disserta que classe é uma coleção de campos que contêm valores e métodos que operam sobre esses valores. As classes são o elemento estrutural mais fundamental de todos os programas Java. Você não pode escrever código Java sem definir uma classe. Já o objeto é uma instância de uma classe.

A linguagem de programação Java é uma linguagem orientada a objetos de última geração que tem uma sintaxe semelhante à de C. Os projetistas da linguagem se esforçaram ao máximo para tornar a linguagem Java eficiente, mas, ao mesmo tempo, tentaram evitar os recursos extremamente complexos que arruinaram outras linguagens orientadas a objetos, como c++.

Mantendo a linguagem simples, os projetistas também tornaram mais fácil para o programador escrever um código robusto e isento de falhas. Como resultado do seu projeto elegante e de recursos de última geração, a linguagem Java tornou-

se popular entre os programadores que, em geral, descobrem o prazer de trabalhar com Java depois de se debaterem com linguagens mais difíceis e menos eficientes. (FLANAGAN, 2006).

#### 3.3.2.2 Firebase auth

Para ter a identidade do usuário reconhecida e permitir que o sistema salve os dados do usuário com segurança, foi necessária a integração do SDK do Firebase Authentication, tirando a responsabilidade de verificação dos dados de *login* e segurança do aplicativo móvel e enviando para um componente que implementa as práticas recomendadas de autenticação em dispositivos móveis além de lidar com provedores de identidades conhecidos, como login do Google e Facebook (FIREBASE, 2018).

#### 3.3.2.3 Google Vision

O Google Cloud Vision oferece modelos pré-treinados por meio de uma API que permite aos desenvolvedores entender o conteúdo de imagens. Essa API rapidamente classifica as imagens em milhares de categorias e extrai palavras impressas contidas nas imagens (GOOGLE, 2019).

#### 3.3.2.4 Facebook Android SDK

Foi utilizado a API do Facebook para realização do *login* no aplicativo móvel e obter perfil do usuário da rede social (FACEBOOK, 2019).

#### 3.3.2.5 Retrofit

Retrofit é uma API de transmissão de dados, que foi criada pela Square e atualmente mantida pela comunidade de *software* livre. Bem aceita pela comunidade Android, pois sua simplicidade de criar e tratar uma conexão entre cliente e servidor

configurações de conexões, além de, ter opção de interceptar requisições e inserir *token* de segurança (SQUARE, 2018).

#### 3.3.2.6 Scottyab Secure Preferences

O Android permite salvar informações em preferencias compartilhadas (*shared preferences*), contudo, esta biblioteca pode ser facilmente alvo de acessos indevidos (via linha de comando que a plataforma Android possui) e assim seu uso é recomendado para informações não-sensíveis do aplicativo

Visando tornar o aplicativo mais seguro, foi utilizada a biblioteca *secure-preferences*, que criptografa (ofusca) estas informações a serem salvas e descriptografa (obtem) quando o aplicativo necessita dessas informações (SCOTTYAB, 2019).

#### 3.3.2.7 Picasso

Conforme o *retrofit*, a biblioteca Picasso foi utilizada para ser responsável por todas as requisições de imagens, de forma assíncrona, do aplicativo. Esta biblioteca é desenvolvida pela Square (SQUARE, 2019).

#### 3.3.2.8 Spring Boot

Spring Boot é um *framework* desenvolvido em Java pela empresa Pivotal para criação de aplicativos independentes. Esse *framework* facilita o processo de configuração e publicação gerenciando dependências de maneira automática (pelo site do *framework* é possível marcar quais módulos deseja utilizar, como Web, segurança e persistência de dados e fazer o *download* do projeto inicial já configurado), simplificando a execução do projeto e diminuindo tempo de desenvolvimento. Ele também possui empacotamento automático da aplicação em um arquivo executável (JAR) com todas as dependências necessárias para iniciar a aplicação, sem necessidade de configurar servidor de aplicação (SPRING, 2019).

### 3.3.2.9 Weka

Weka é um *software* de código aberto que foi implementado pela Universidade de Waikato na Nova Zelândia. Possui uma coleção de algoritmos (preparação, classificação, regressão, visualização de dados) de aprendizado de máquina para mineração e exploração de dados. Nesse projeto, foi utilizada a API do Weka incorporada ao projeto para realização de mineração de dados de maneira automatizada (WAIKATO, 2019).

### 3.3.3 Banco de dados

Para inserção dos dados será utilizado o Banco de Dados PostgreSQL. Segundo a documentação oficial, PostgreSQL é um sistema de gerenciamento de banco de dados objeto-relacional baseado no desenvolvido pelo Departamento de Ciência da Computação da Universidade da Califórnia em Berkeley. O Postgre foi pioneiro em vários conceitos que somente se tornaram disponíveis mais tarde em alguns sistemas de bancos de dados comerciais (POSTGRESQL, 2018).

O PostgreSQL é um descendente de código fonte aberto deste código original de Berkeley, que suporta grande parte do padrão SQL e oferece muitas funcionalidades modernas. Devido à sua licença liberal, o PostgreSQL pode ser utilizado, modificado e distribuído por qualquer pessoa para qualquer finalidade, seja particular, comercial ou acadêmica, livre de encargos (POSTGRESQL, 2018).

As principais características do PostgreSQL são as seguintes:

- Banco de dados do tipo objeto-relacional;
- É de livre distribuição e de código aberto;
- Capacidade ilimitada para armazenamento de dados;
- Cada tabela pode armazenar até 16TB (cada campo até 1GB);
- Pode ser usado em diversas plataformas;
- Implementa *triggers*, herança, sequências, *stored procedures*, *functions*, etc;
- Compatíveis com linguagens *open source*.

### 3.4 Funcionalidades da aplicação

As funcionalidades foram divididas em 2 partes, as funcionalidades que são executadas no lado cliente (*front-end*), ou seja, no aplicativo para dispositivos móveis e, para as funcionalidades que serão executadas no servidor (*back-end*).

#### 3.4.1 Front-end

- Realizar cadastro no sistema;
- Realizar login;
- Realizar cadastro de matérias;
- Tirar fotos;
- Adicionar comentários a fotos;
- Definir privacidade das matérias cadastradas;
- Importar matérias;

#### 3.4.2 Back-end

- Realizar transformação do conteúdo em mídia para texto;
- Realizar mineração de dados;
- Consultar dados adicionais de acordo com dados minerados.

### 3.5 Implementação dos serviços

A implementação do sistema de acordo com a arquitetura proposta foi efetuada nas seguintes etapas de desenvolvimento:

#### 1. Implementação do aplicativo móvel

O desenvolvimento foi iniciado pelas classes de segurança, responsáveis pela comunicação com o serviço de autenticação com o Firebase Auth, garantindo a

segurança e integridade dos usuários que utilizarão o aplicativo. Foi gerado e disponibilizado, a partir do console de configurações do Firebase Auth, o arquivo “google-services.json” que contém informações necessárias para que o aplicativo possa se comunicar com o serviço de autenticação.

A comunicação a partir desse ponto passou a ser somente com o servidor *back-end*, conectando com Java *standalone server*. Para garantir a autenticidade dos dados, a cada requisição ou envio de dados é encaminhado também um *token* criado pelo *back-end*.

Após a autenticação ter sido concluída, iniciou-se a implementação do cadastro de matérias e seus devidos horários. Tal cadastro é necessário para que as fotos tiradas sejam automaticamente organizadas em suas devidas disciplinas. Foi implementado também a importação das matérias e seus devidos horários diretamente da instituição de ensino do usuário do aplicativo. Para que a importação automática ocorra, é necessário que a instituição de ensino implemente um serviço para disponibilizar as informações, a partir da matrícula e senha passada pelo aluno. Após esse procedimento, as matérias serão apresentadas em uma lista na tela inicial do aplicativo.

O final da implementação deu-se com o desenvolvimento da operação de tirar fotos, na qual é solicitada permissão do usuário para acessar a câmera do aparelho e gravar dados na memória. Após a foto ser tirada, ela é armazenada no dispositivo e não é mostrada na galeria de fotos. Para encaminhar as fotos para o servidor *back-end*, foi criada uma *Thread* que verifica se tem fotos novas e encaminha as fotos no segundo plano, sem bloquear a utilização do aplicativo.

## 2. Implementação do Servidor *Back-end*

O *back-end* é composto pelo serviço Java *standalone*, que é a base do sistema e é responsável pelo controle de todos os serviços; banco de dados Postgresql, um diretório reservado para armazenamento das imagens encaminhadas pelo aplicativo móvel e a chamada dos serviços de ocr e mineração de dados.



A implementação começou pelo serviço Java, pois as tabelas e colunas do banco de dados são criadas automaticamente quando o serviço é executado devido a anotações nas classes Java, não necessitando a criação de *scripts* para tal ação.

Foi criada então a verificação do *token* encaminhado pelo aplicativo móvel e cadastro do usuário, caso o cadastro ainda não tenha sido efetuado, com as informações passadas pelo serviço *firebase auth*.

Foram criadas classes para controle e cadastro das informações das matérias passadas pelo aplicativo móvel, assim como as mídias passadas. Após o *upload* das fotos ser concluído, é criada uma *thread* para encaminhar a foto para o serviço de processamento de imagens Google Vision no qual retorna o texto retirado da imagem, que é salvo no banco de dados.

Também foi implementada a análise de similaridade entre as matérias cadastradas dos usuários para encaminhar notificação para o aplicativo móvel informando uma possível matéria em comum com outro usuário.

## 4 PROTÓTIPO E RESULTADOS

### 4.1 Protótipo

Integrando o detalhamento dos requisitos de interface aos casos de uso, cenário, regras de dados e de negócio, a prototipagem frequentemente é definida pelas partes interessadas como um meio concreto de análise da interface, em que se observar as necessidades do projeto para validação como para a identificação de novas funcionalidades (IIBA, 2011). O protótipo é visto e revisto por um ou mais usuários, no qual são feitas análises críticas acerca de suas características e, com base nessas observações, o protótipo é corrigido ou melhorado. Esse processo pode ocorrer diversas vezes até o usuário aceitar (BEZERRA, 2007).

A utilização do protótipo pode ser realizada de forma descartável ou evolucionária. Na forma descartável, o protótipo visa esclarecer, de forma ágil, os requisitos de interface utilizando ferramentas simples, papel e caneta, por exemplo, e é descartado quando o sistema finaliza. Já o protótipo evolucionário produz um aplicativo de *software* funcional (IIBA, 2011).

Uma das vantagens da prototipagem é a questão do *feedback* antecipado. Através de um meio barato, o protótipo descartável é utilizado para descobrir e confirmar uma série de requisitos que vai além da interface como, por exemplo, os processos e regras de dado e negócio. O usuário se sente mais confortável, pois consegue ver a interface futura do sistema (IIBA, 2011).

Abaixo é possível verificar algumas telas do protótipo inicial criado (Figura 10).

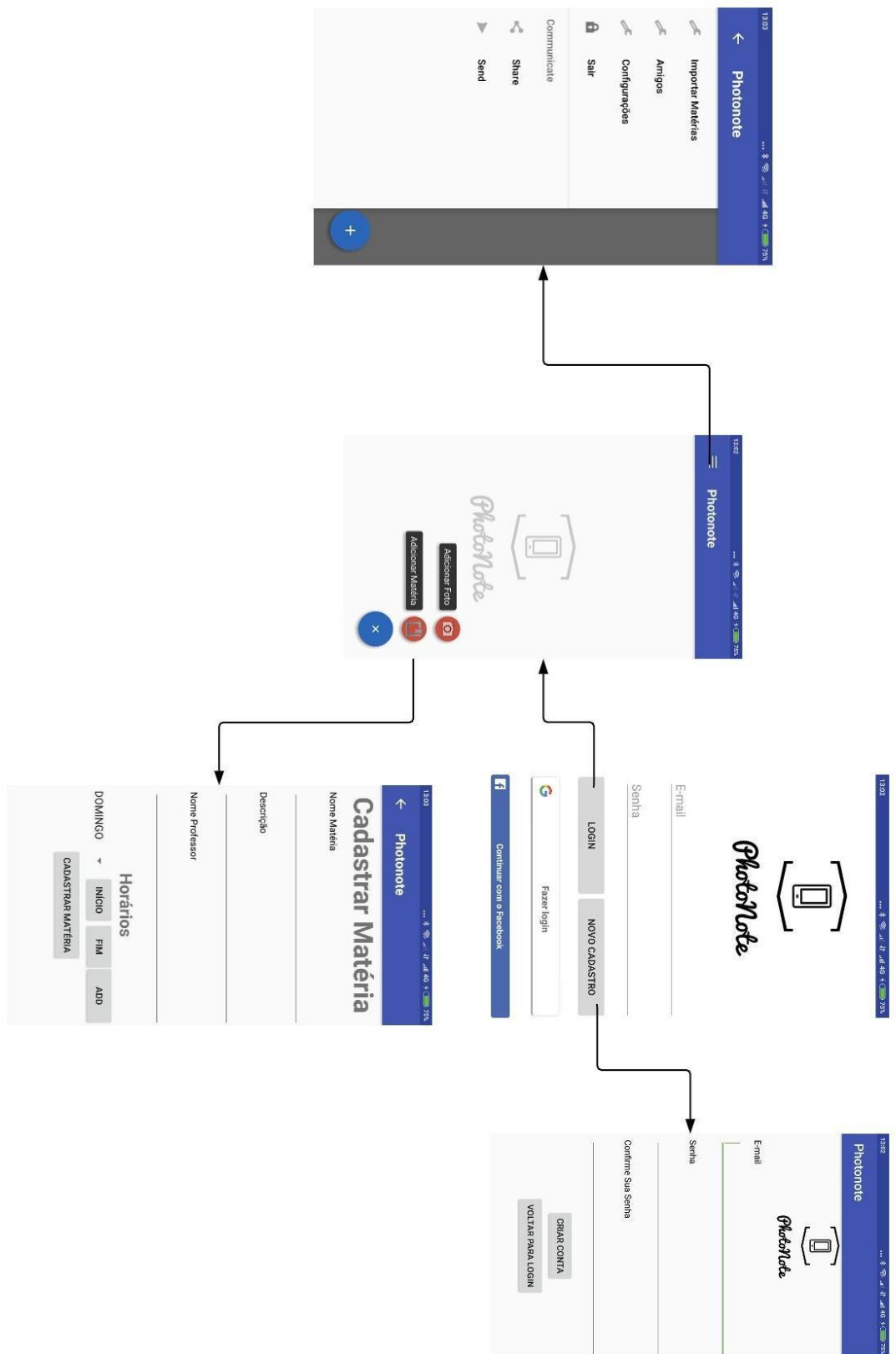


Figura 10- Imagens protótipo

## 4.2 Resultado

A primeira operação necessária para o completo funcionamento do sistema, é a inicialização do serviço *standalone* Java, para que o aplicativo móvel consiga se comunicar. Após o serviço ser inicializado conforme figura 11, o aplicativo móvel já pode ser instalado e ser utilizado pelo usuário.

```

Spring Boot :: (v2.1.3.RELEASE)

2019-04-06 21:44:32.885 INFO 12788 --- [main] b.c.i.p.PhotonoteserverApplication : Starting PhotonoteserverApplication on DESKTOP-SUM4BV5 with PID 12788
2019-04-06 21:44:32.887 INFO 12788 --- [main] b.c.i.p.PhotonoteserverApplication : No active profile set, falling back to default profiles: default
2019-04-06 21:44:33.898 INFO 12788 --- [main] s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data repositories in DEFAULT mode.
2019-04-06 21:44:34.078 INFO 12788 --- [main] s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 85ms. Found 5 repository interfaces.
2019-04-06 21:44:34.422 INFO 12788 --- [main] trationDelegate$BeanPostProcessorChecker : Bean 'org.springframework.transaction.annotation.ProxyTransactionManagementConfiguration' of type [org.springframework.transaction.annotation.ProxyTransactionManagementConfiguration$$EnhancerBySpringCGIJB$$3f32652] is not eligible for getting processed by all BeanPostProcessors (for example: not eligible for auto-proxying)
2019-04-06 21:44:34.848 INFO 12788 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 9080 (http)
2019-04-06 21:44:34.875 INFO 12788 --- [main] org.apache.catalina.core.StandardService : Starting service [Tomcat]
2019-04-06 21:44:34.875 INFO 12788 --- [main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.16]
2019-04-06 21:44:35.010 INFO 12788 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2019-04-06 21:44:35.010 INFO 12788 --- [main] o.s.web.context.ContextLoader : Root WebApplicationContext: initialization completed in 2090 ms
2019-04-06 21:44:35.182 INFO 12788 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting
2019-04-06 21:44:35.302 INFO 12788 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2019-04-06 21:44:35.347 INFO 12788 --- [main] o.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitInfo [ name: default ... ]
2019-04-06 21:44:35.396 INFO 12788 --- [main] org.hibernate.Version : HHH000412: Hibernate Core (5.3.7.Final)
2019-04-06 21:44:35.397 INFO 12788 --- [main] org.hibernate.cfg.Environment : HHH000208: hibernate.properties not found
2019-04-06 21:44:35.598 INFO 12788 --- [main] o.hibernate.annotations.common.Version : HCANN000001: Hibernate Commons Annotations (5.0.4.Final)
2019-04-06 21:44:36.659 INFO 12788 --- [main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2019-04-06 21:44:37.009 INFO 12788 --- [main] o.h.h.i.QueryTranslatorFactoryInitiator : HHH000397: Using ASTQueryTranslatorFactory
2019-04-06 21:44:37.424 WARN 12788 --- [main] aWebConfiguration$JpaWebMvcConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may be performed during view rendering. Explicitly configure spring.jpa.open-in-view to disable this warning
2019-04-06 21:44:37.717 INFO 12788 --- [main] s.s.UserDetailsServiceAutoConfiguration :
2019-04-06 21:44:37.833 INFO 12788 --- [main] o.s.s.concurrent.ThreadPoolTaskExecutor : Initializing ExecutorService
2019-04-06 21:44:37.834 INFO 12788 --- [main] o.s.s.concurrent.ThreadPoolTaskExecutor : Initializing ExecutorService 'taskExecutor'
2019-04-06 21:44:38.271 INFO 12788 --- [main] d.s.w.p.DocumentationPluginsBootstrapper : Context refreshed
2019-04-06 21:44:38.285 INFO 12788 --- [main] d.s.w.p.DocumentationPluginsBootstrapper : Found 1 custom documentation plugin(s)
2019-04-06 21:44:38.353 INFO 12788 --- [main] d.s.w.s.ApiListingReferenceScanner : Scanning for api listing references
2019-04-06 21:44:38.563 INFO 12788 --- [main] d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: atualizar/UsingPUT_1
2019-04-06 21:44:38.565 INFO 12788 --- [main] d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: cadastrar/UsingPOST_1
2019-04-06 21:44:38.568 INFO 12788 --- [main] d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: excluir/UsingDELETE_1
2019-04-06 21:44:38.571 INFO 12788 --- [main] d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: listar/UsingGET_1
2019-04-06 21:44:38.572 INFO 12788 --- [main] d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: obter/UsingGET_1
2019-04-06 21:44:38.622 INFO 12788 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 9080 (http) with context path '/'
2019-04-06 21:44:38.626 INFO 12788 --- [main] b.c.i.p.PhotonoteserverApplication : Started PhotonoteserverApplication in 6.045 seconds (JVM running for 6.49)
  
```

Figura 11- Inicialização serviço standalone Java

Na tela inicial do aplicativo móvel, nomeado como Photonote, é possível realizar o *login* informando usuário e senha, uma conta válida do Google ou com uma conta do Facebook (Figura 12). Caso o usuário deseje realizar o login informando usuário e senha, é necessário clicar no botão novo cadastro e preencher os campos de e-mail, senha e confirmação de senha. Ao clicar para criar conta, o *login* é efetuado de maneira automática.



Figura 12- Página inicial

Já na tela inicial do aplicativo são listadas as matérias cadastradas ou importadas pelo usuário, juntamente com um botão para adicionar mais matérias ou abrir a câmera para capturar o conteúdo desejado conforme pode ser visto na figura 13.

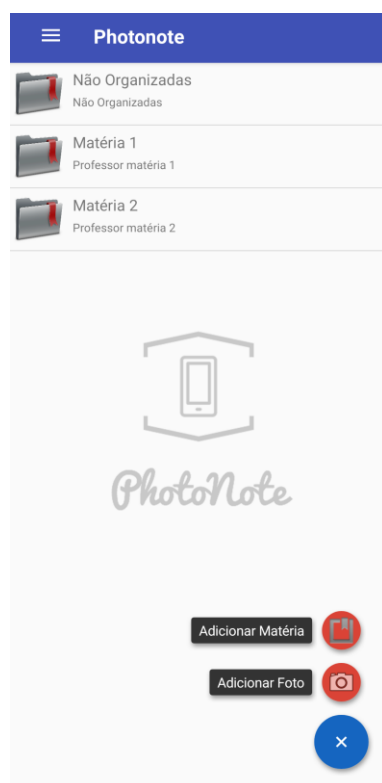


Figura 13- Tela principal

Ao clicar em adicionar matéria, o aplicativo mostra a tela (Figura 14) com todos os campos de preenchimento necessários para o correto funcionamento do sistema, pois, é a partir desses dados inseridos que a organização automática das mídias será efetuada, concluindo um dos objetivos do trabalho proposto.

← Photonote

## Cadastrar Matéria

Nome Matéria  
Matéria 1

Descrição  
matéria de segunda

Nome Professor  
Professor matéria 1

**Privacidade**  
PRIVADA

**Duração**  
01/04/2019 30/06/2019

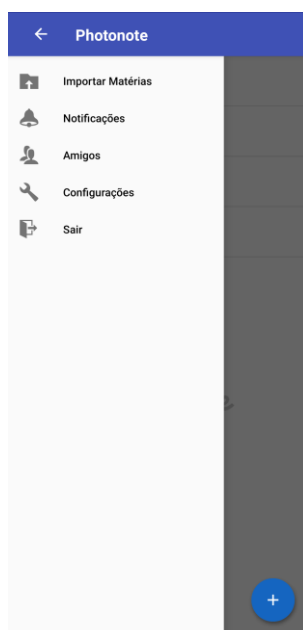
**Horários**

SEGUNDA	19:15	20:15	ADD
SEGUNDA	20:30	22:00	REMOVE

CADASTRAR MATÉRIA

Figura 14- Tela de cadastro de matéria

Na tela principal (figura 13) também é possível clicar na parte superior esquerda no menu, onde são mostradas opções (figura 15) de importação de matérias, notificações de possíveis matérias em comum com outros usuários, configurações do aplicativo e os demais usuários que podem ser adicionados como amigo.



*Figura 15- Menu lateral*

Na tela principal, ao clicar em qualquer matéria, é aberta uma caixa de diálogo no qual são mostradas informações referentes à matéria e botões para visualização das imagens, edição dos dados da matéria e opção de compartilhamento da matéria com demais usuários (figura 16).



*Figura 16- Dados matéria*

Clicando no botão ver mídia é mostrada a galeria de imagens vinculadas a matéria (figura 17).

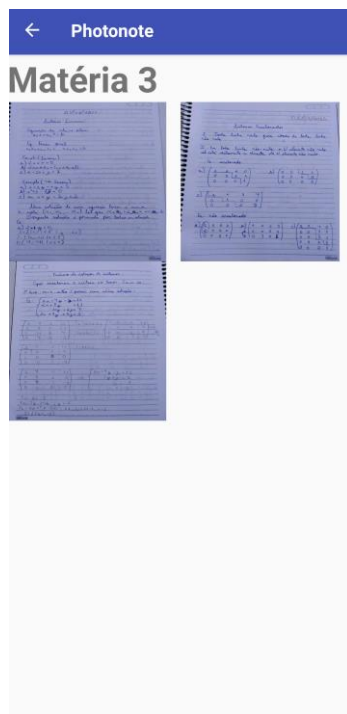


Figura 17- Galeria de imagens

Esses são procedimentos visíveis ao usuário. Em segundo plano, caso o usuário esteja conectado em alguma rede *wifi*, o aplicativo ativa uma execução a cada 15 minutos, verificando se existe alguma foto nova adicionada para encaminhá-la ao servidor para ser salva e processada.

No servidor, foi criado um método para salvar a foto no diretório de armazenamento de arquivos e, gravar somente o diretório de armazenamento com o nome da imagem no banco de dados (figura 18).



```

public MidiaMateria saveMidia(MidiaMateria midia, MultipartFile file) {
    String diretorio = path + midia.getMateria().getId() + "/";
    if (file.isEmpty()) {
        throw new StorageException("Arquivo vazio");
    }
    if (!Files.exists(Paths.get(diretorio), LinkOption.NOFOLLOW_LINKS)) {
        try {
            Files.createDirectories(Paths.get(diretorio));
        } catch (IOException e) {
            throw new StorageException(String.format("Erro ao gravar arquivo", file.getName()), e);
        }
    }
    try {
        Files.copy(file.getInputStream(), Paths.get(diretorio + file.getOriginalFilename()), StandardCopyOption.REPLACE_EXISTING);
    } catch (IOException e) {
        throw new StorageException(String.format("Erro ao gravar arquivo", file.getName()), e);
    }
    midia.setDiretorioMidia(diretorio);
    midia.setTipoMidia(file.getContentType());
    midia.setNomeMidia(file.getOriginalFilename());
    midia.setProcessada(Boolean.FALSE);
    midia = midiaMateriaRepository.save(midia);
    return midia;
}

```

*Figura 18- Método de salvar mídia*

No término desse processo, é retornado para o aplicativo a mensagem informando se ocorreu erro ou se a imagem foi gravada com sucesso (figura 19).

```

@PostMapping("/cadastrarMidia")
public ResponseEntity<?> cadastrarMidia(@RequestParam("arquivoMidia") MultipartFile arquivoMidia, Long materiaId) {
    MidiaMateria midia = new MidiaMateria();
    Materia materia = null;
    try {
        materia = materiaService.findById(materiaId);
    } catch (ResourceNotFoundException e) {
        return new ResponseEntity(new ApiResponseVo(false, e.getMessage()), HttpStatus.INTERNAL_SERVER_ERROR);
    }

    midia.setMateria(materia);
    midia = materiaService.saveMidia(midia, arquivoMidia);
    if (NullUtil.isNullOrEmpty(midia) || NullUtil.isNullOrEmpty(midia.getId())) {
        return new ResponseEntity(new ApiResponseVo(false, "Ocorreu um erro ao cadastrar!"),
            HttpStatus.INTERNAL_SERVER_ERROR);
    }

    materiaService.processarMidia(midia.getId());
    return ResponseEntity.ok(new ApiResponseVo(true, "Midia cadastrada com sucesso"));
}

```

*Figura 19-Método cadastrar mídia*

Assim que o processo de salvar a mídia é concluído, é iniciado um procedimento assíncrono para capturar o texto das imagens obtidas (figura 19). Nesse procedimento, é utilizado o SDK do Google Vision, onde é encaminhada a imagem e o servidor retorna o texto extraído. Caso tenha êxito na extração, é salvo o retorno no banco de dados (figura 20).

```

@Async
public void processarMidia(Long id) {
    MidiaMateria midiaMateria = midiaMateriaRepository.findById(id).orElseThrow(() -> new ResourceNotFoundException("MidiaMateria", "id", id));
    List<AnnotateImageRequest> requests = new ArrayList<>();
    Boolean possuiErro = Boolean.FALSE;
    StringBuilder mensagemErro = new StringBuilder();
    StringBuilder textoImagem = new StringBuilder();
    try {
        ByteString imgBytes = ByteString.readFrom(new FileInputStream(midiaMateria.getDiretorioMidia() + midiaMateria.getNomeMidia()));
        Image img = Image.newBuilder().setContent(imgBytes).build();
        Feature feat = Feature.newBuilder().setType(Type.DOCUMENT_TEXT_DETECTION).build();
        AnnotateImageRequest request = AnnotateImageRequest.newBuilder().addFeatures(feat).setImage(img).build();
        requests.add(request);

        try (ImageAnnotatorClient client = ImageAnnotatorClient.create()) {
            BatchAnnotateImagesResponse response = client.batchAnnotateImages(requests);
            List<AnnotateImageResponse> responses = response.getResponsesList();
            client.close();
            for (AnnotateImageResponse res : responses) {
                if (res.hasError()) {
                    possuiErro = true;
                    mensagemErro.append(res.getError().getMessage());
                } else {
                    textoImagem.append(res.getFullTextAnnotation().getText());
                }
            }
            if (!possuiErro) {
                midiaMateria.setTextoMidia(textoImagem.toString());
            } else {
                throw new ProcessMidiaException("Erro ao processar mídia: " + mensagemErro.toString());
            }
        }
    } catch (FileNotFoundException e) {
        e.printStackTrace();
        throw new ProcessMidiaException("Erro ao processar mídia", e);
    } catch (IOException e) {
        e.printStackTrace();
        throw new ProcessMidiaException("Erro ao processar mídia", e);
    }

    midiaMateria.setProcessada(Boolean.TRUE);
    midiaMateriaRepository.save(midiaMateria);
}

```

*Figura 20 - Método processar mídia*

Foi iniciado então envio de diversas imagens para o servidor para verificar as dificuldades no processamento correto do texto encaminhado. A primeira foi uma foto de escrita cursiva em folha branca (figura 21).

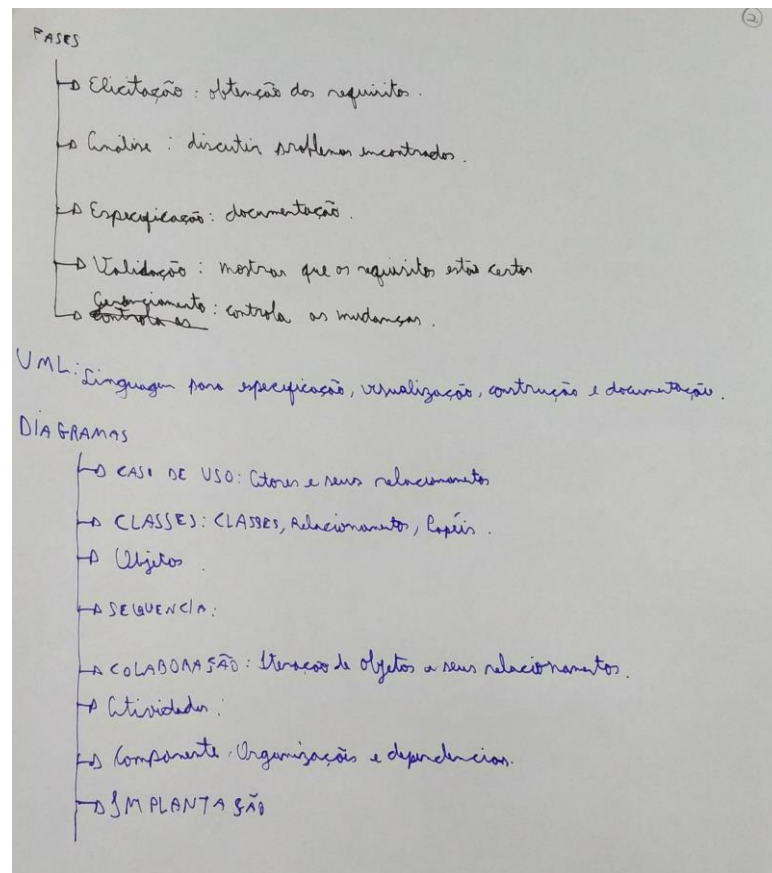


Figura 21- Folha branca em letra cursiva

E foi obtido o resultado abaixo conforme figura 22:

```

FASES
to Elicitação: obtenção dos requisitos
to Análise discutir problemas encontrados
po Especificação: documentação.
M Validação: mostrar que os requisitos estes certos
La casa in to controla as mudanças
UML Singungen
Linguagen para especificação, visualização, construção e doamentarai.
DIAGRAMAS
LO CASO DE USO: Citores e seus relacionamentos
LA CLASSES: CLASSES, Relacionamentos, Papéis.
to Objetos.
KASEGUENCIA:
LA COLABORAÇÃO: Steração de objetos a seus relacionamentos.
A atividades:
hos componente Organizações e dependencion.
TOIMPLANTAÇÃO
  
```

Figura 22- Resultado processamento figura 21

Foi utilizada uma imagem tirada a partir de uma folha com linhas e escrita com letra de forma (figura 23):

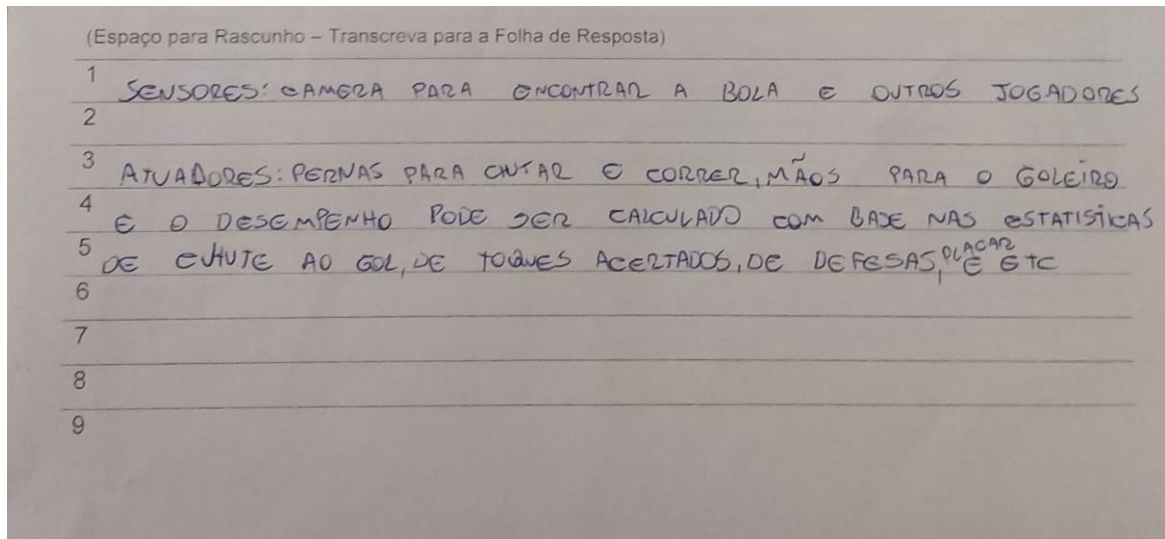


Figura 23- Folha com linhas e letra de forma

Após processamento da figura 23 foi gerado o texto da imagem abaixo:

**SENSORES:** CAMERA PARA  
ENCONTRAR  
A BOLA  
E  
OUTROS JOGADORES  
**3 ATUADORES:** PERNAS PARA CHUTAR E correr MÃOS PARA O GOLEIRO  
E O DESEMPENHO PODE SER CALCULADO com BASE NAS ESTATÍSTICAS  
DE CHUTE AO GOL, DE to Ques ACERTADOS, DE DEFESAS QUE STC

Figura 24- Resultado processamento figura 24

Para verificar se o objetivo do trabalho pode ser obtido, foi processada imagem de letra cursiva a partir de quadro branco (figura 25):

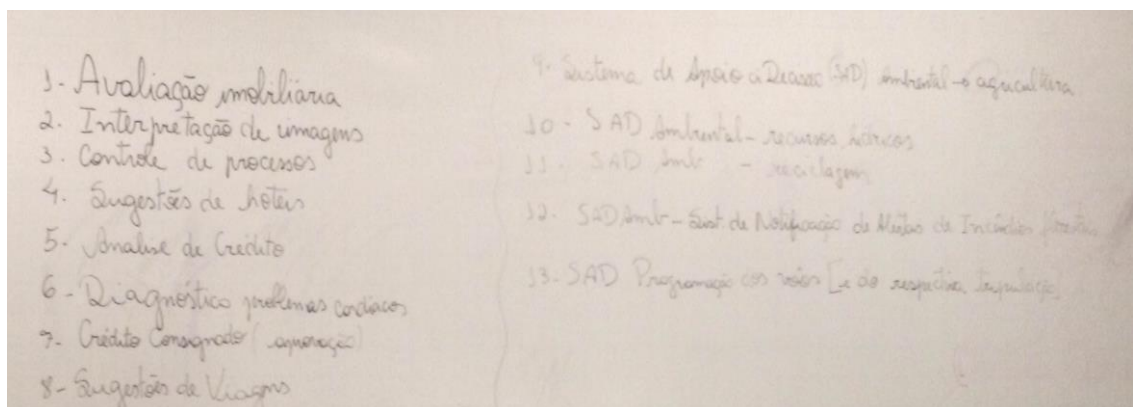


Figura 25- Imagem letra cursiva a partir de quadro branco

Foi obtido resultado abaixo conforme figura 26:

Lastema de spaie a Deanac SD emberta  
 agucultura  
 5. Avaliação imobiliária  
 2. Interpretação de imagens  
 3. Controle de processos  
 4. Sugestões de hotes.  
 5. nause de Crechte  
 6- Diagnostico poreklenas canciacos  
 - Credito Consonas quereços  
 5- Sugestas de Viagnos  
 33. SAD, ami - Sist die Notwarcia dla Wasas de Incine  
 13. SAD Presseomaca s os e do sen t e

Figura 26- Resultado processamento figura 25

Por último, foi utilizada imagem com letra de forma tirada a partir do quadro branco (figura 27).

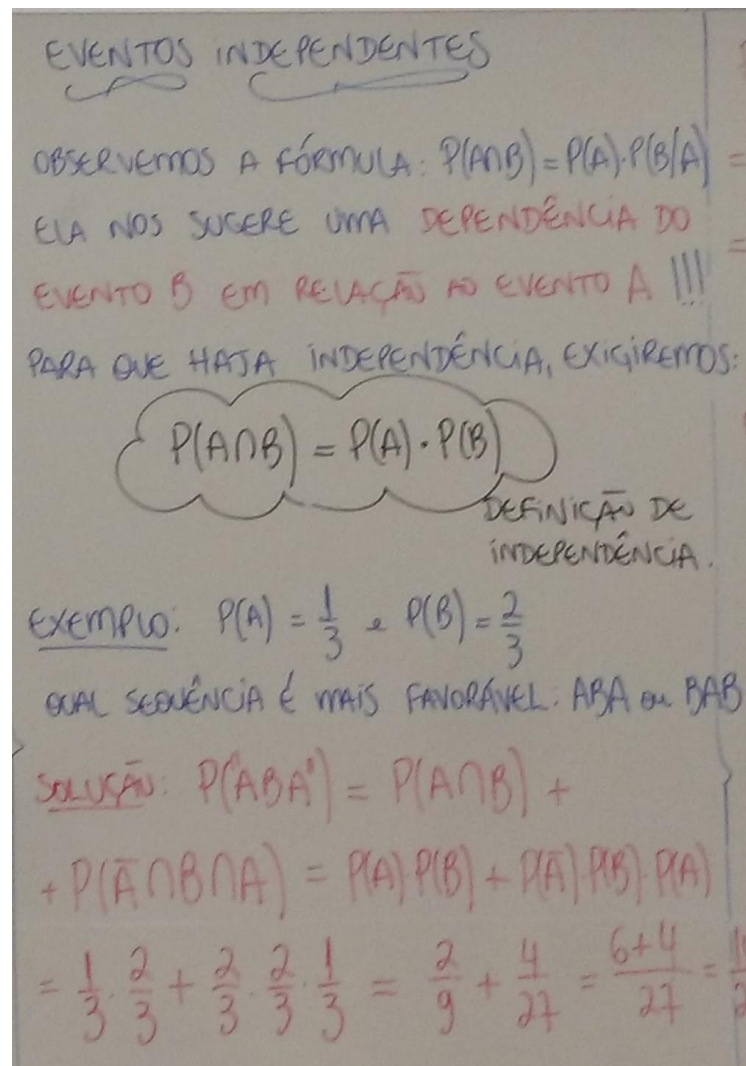


Figura 27- Imagem letra de forma em quadro branco

O resultado do processamento da foto tirada do quadro branco (figura 27) pode ser verificado abaixo (figura 28):

```
EVENTOS INDEPENDENTES
OBSERVemos A Fórmula:  $P(A \cap B) = P(A) \cdot P(B)$ 
ELA NOS SUGERE UMA DEPENDENCIA DO
EVENTO B EM RELAÇÃO AO EVENTO A III
PARA QUE HAJA INDEPENDENCIA, EXIGIREMOS:
 $P(A \cap B) = P(A) \cdot P(B)$ 
TÉCNICA DE
INDEPENDENCIA
Exemplo:  $P(A) = P(B) = 2$ 
GAL RESUMÊNCIA É mais FAVORÁVEL. ABA B BAB
SAUSAS PRABA) =  $P(A \cap B) + P(A \cap C) + P(A \cap D) + P(A \cap E)$ 
```

Figura 28- Resultado processamento figura 27

Após o processamento da imagem, o texto obtido através da foto pode ser obtido no aplicativo, sendo necessário o usuário selecionar a foto que ele deseja verificar.

Com as mídias processadas e os dados armazenados no banco de dados, é necessária a realização de tratamento dos dados, pois, na maneira que foram inseridos existem vários caracteres que não são reconhecidos na tradução da imagem e acabam gerando ruídos.

Para tratativa desses ruídos, foi usada a API Weka, que realiza utilização de técnicas de mineração de dados.

A partir dos dados gerados pela Weka, o servidor encaminha sugestões de matérias entre usuários do sistema, gerando uma rede de amigos e compartilhamento de matérias.

## 5 CONCLUSÃO

Embora, a internet ajude nos estudos em alguns casos com disponibilidade de artigos científicos e, até mesmo, *sites* especializados em conteúdos técnicos que são fontes fundamentadas de pesquisa, as informações e o conhecimento transmitidos pelos professores em sala de aula nem sempre podem ser substituídos por outras fontes. Assim, como apresentado no decorrer deste trabalho, existe certa carência de uma ferramenta que centralize as imagens feitas pelos estudantes em seus *smartphones* e *tablets* acerca dos conteúdos das disciplinas que foram registrados na lousa.

Tendo em vista este fator, foi desenvolvido o sistema referente a este trabalho, o qual permite a subdivisão de conteúdos em categorias (matérias) de forma que estas informações estejam disponíveis na interface do usuário.

Proporcionando o armazenamento em nuvem, o estudante terá toda a liberdade para poder usar seu aparelho celular sem tomar espaço, de seus demais aplicativos ou arquivos, dando uma segurança e robustez maior ao sistema.

No desenvolvimento do trabalho, foi verificado que fotos de baixa qualidade, cálculos e fórmulas, além da maneira de escrever pode influenciar diretamente no funcionamento da tradução das fotos em texto, assim com a identificação de similaridade entre as matérias dos usuários.

Dessa forma faça-se necessário um estudo mais aprofundado sobre novas técnicas de leitura de letra cursiva, assim como verificação de espaçamento entre palavras e fórmulas, visando um resultado mais limpo e compreensível.

O objetivo deste projeto será realmente alcançado quando estudantes puderem se beneficiar do sistema desenvolvido neste trabalho para oferecer maior organização e praticidade, proporcionando aulas mais dinâmicas e produtivas.

## TRABALHOS FUTUROS

A partir do trabalho desenvolvido, pode-se destacar possíveis trabalhos futuros:

- Criação de um serviço Web para visualização do conteúdo obtido pelo celular.
- Melhoria das técnicas de tradução de imagens em texto.
- Implementação da aplicação móvel para dispositivos com outros sistemas operacionais, visto que o projeto desenvolvido foi feito para funcionar somente em celulares com sistema operacional Android.



## REFERÊNCIAS BIBLIOGRÁFICAS

BERRY, Michael J. A.; LINOFF, Gordon S.. **Data Mining Techniques For Marketing, Sales, and Customer Relationship Management**. 2. ed. Indianapolis: Wiley Publishing Inc, 2004.

BEZERRA, Eduardo. **Princípios de análise e projeto de sistemas com UML**. 2.ed. Rio de Janeiro: Campus, 2007.

BRASIL, Univesia. **Aplicativo organiza fotos tiradas da lousa e auxilia nos estudos**. 2016. Disponível em: <<http://noticias.universia.com.br/destaque/noticia/2016/02/03/1136057/aplicativo-organiza-fotos-tiradas-lousa-auxilia-estudos.html>>. Acesso em: 12 nov. 2018.

CAESAR, T.; GLOGER, J.m.; MANDLER, E. **Preprocessing and feature extraction for a handwriting recognition system**. 1993. Disponível em: <<https://ieeexplore.ieee.org/document/395706>>. Acesso em 12 nov. 2018.

DATE, C. J.. **Introdução a Sistemas de Banco de Dados**. 7. ed. Rio de Janeiro: Campus, 1989.

DBEAVER. **DBeaver Community**. Disponível em: <<https://dbeaver.io/>>. Acesso em: 12 nov. 2018.

DURAN, Pedro. **Mão de obra qualificada perde espaço no mercado de trabalho**. Disponível em: <<http://cbn.globoradio.globo.com/editorias/economia/2016/02/27/MAO-DE-OBRA-QUALIFICADA-PERDE-ESPACO-NO-MERCADO-DE-TRABALHO.htm>>. Acesso em 12 nov. 2018.

ECLIPSE. **Eclipse desktop & web IDE**. Disponível em: <<https://www.eclipse.org/ide/>>. Acesso em: 12 nov. 2018.

FACEBOOK. **Adroid - Login do Facebook**. Disponível em: <<https://developers.facebook.com/docs/facebook-login/android>>. Acesso em: 30 mar. 2019.

FIREBASE. **Firestore Authentication**. Disponível em: <<https://firebase.google.com/docs/auth>>. Acesso em: 12 nov. 2018.

FAYYAD, Usama M.; PIATETSKY-SHAPIO, Gregory; SMYTH, Padhraic. **Advances in Knowledge Discovery and Data Mining**. Ca,usa: American Association For Artificial Intelligence Menlo Park, 1996. Disponível em: <[https://openlibrary.org/books/OL808581M/Advances\\_in\\_knowledge\\_discovery\\_and\\_data\\_mining](https://openlibrary.org/books/OL808581M/Advances_in_knowledge_discovery_and_data_mining)>. Acesso em: 12 nov. 2018.

FEITOSA, Marcio Porto. **Fundamentos de Banco de Dados: Uma abordagem prático-didática**. São Paulo: Edição do Autor, 2013.

FLANAGAN, David. **Java. O Guia Essencial**. 5. ed. Porto Alegre: Bookman, 2006.

GOOGLE. **Cloud Vision**. Disponível em: <<https://cloud.google.com/vision/>>. Acesso em: 30 mar. 2019.

GRAGNANI, Juliana. **Estudantes trocam anotações em cadernos por foto da lousa no celular**. 2015. Disponível em: <<https://www1.folha.uol.com.br/educacao/2015/10/1698273-estudantes-trocam-anotacoes-em-cadernos-por-foto-da-lousa-no-celular.shtml>>. Acesso em: 12 nov. 2018.

IIBA. **Um guia para o Corpo de Conhecimento de Análise de Negócios™ (Guia BABOK®) (Portuguese Edition)**. Toronto: International Institute Of Business Analysis, 2011.

KANTARDZIC, Mehmed. **DATA MINING Concepts, Models, Methods, and Algorithms**. 2. ed. New Jersey: John Wiley & Sons, Inc., 2011.

LUGER, George F; A STUBBLEFIELD, William. **Artificial intelligence: structures and strategies for complex problem solving**. 3. ed. Harlow: Addison-wesley Longman, 1998.

MACIEL, Camila. **Uso de internet pelo celular cresce entre estudantes e professores, diz pesquisa**. 2015. Disponível em: <<http://agenciabrasil.ebc.com.br/educacao/noticia/2015-09/internet-pelo-celular-cresce-entre-estudantes-e-professores-mostra-pesquisa>>. Acesso em 12 nov. 2018.

MCCARTHY, John; MINSKY, Marvin; ROCHESTER, Nathaniel; SHANNON, Claude. **A PROPOSAL FOR THE DARTMOUTH SUMMER RESEARCH PROJECT ON ARTIFICIAL INTELLIGENCE**. 1955. Disponível em: <<http://www-formal.stanford.edu/jmc/history/dartmouth/dartmouth.html>>. Acesso em 12 nov. 2018.

MINSKY, Marvin. **Semantic information processing**. Cambridge: Mit Press, 1968.

MINSKY, Marvin; PAPERT, Seymour. **Perceptrons: An Introduction to Computational Geometry**. Cambridge: M.i.t. Press, 1969.

MOUSQUER, Tatiana; ROLIM, Carlos Oberdan. **A UTILIZAÇÃO DE DISPOSITIVOS MÓVEIS COMO FERRAMENTA PEDAGÓGICA COLABORATIVA NA EDUCAÇÃO INFANTIL**. Disponível em: <<http://www.santoangelo.uri.br/stin/Stin/artigos.html>>. Acesso em 12 nov. 2018.

PAVLIDIS, Theo. **Algorithms for Graphics and Image Processing**. New York: Springer-verlag Berlin Heidelberg, 1982.

POSTGRESQL. **What is PostgreSQL?** Disponível em: <<https://www.postgresql.org/docs/10/intro-what-is.html>>. Acesso em 12 nov. 2018.

RAUBER, Thomas Walter. **Redes neurais artificiais**. Vitória: Universidade Federal do Espírito Santo, 2005.

SCOTTYAB. **Secure Preferences**. Disponível em: <<https://github.com/scottyab/secure-preferences>>. Acesso em: 30 mar. 2019.

SPRING. **Spring Boot**. Disponível em: <<https://spring.io/projects/spring-boot>>. Acesso em: 30 mar. 2019.

SQUARE. **Retrofit**. Disponível em: <<https://square.github.io/retrofit/>>. Acesso em: 12 nov. 2018.

SQUARE. **Picasso**. Disponível em: <<https://square.github.io/picasso/>>. Acesso em: 30 mar. 2019.

QUINLAN, J Ross. **Machine Learning**. Kluwer Academic Publishers Hingham. MA, USA, p. 81-106. 25 mar. 1986.

SENIOR, A.w.; ROBINSON, A.j. **An off-line cursive handwriting recognition system**. 1998. Disponível em: <<https://ieeexplore.ieee.org/document/667887>>. Acesso em 12 nov. 2018.

SUEN, C.y.; BERTHOD, M.; MORI, S. **Automatic recognition of handprinted characters—The state of the art**. 1980. Disponível em: <<https://ieeexplore.ieee.org/document/1455944>>. Acesso em 12 nov. 2018.

UML, Astha. **Astah UML**. Disponível em: <<http://astah.net/editions/uml-new>>. Acesso em: 12 nov. 2018.

WAIKATO. **Weka**. Disponível em: <<https://www.cs.waikato.ac.nz/ml/weka/>>. Acesso em: 30 mar. 2019.