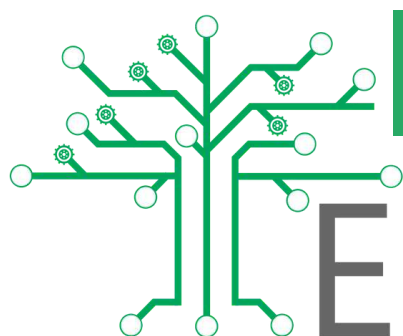


Escola Regional de Computação
Bahia - Alagoas - Sergipe



MINICURSOS

ERBASE 2020 26 a 30 de outubro

Inteligência Computacional na Indústria e no Agronegócio

Organização:
Felipe A. Lopes

Realização:



Patrocínio:



ORGANIZADOR
FELIPE ALENCAR LOPES

MINICURSOS DA XX ESCOLA REGIONAL DE COMPUTAÇÃO
BAHIA, ALAGOAS E SERGIPE
INTELIGÊNCIA COMPUTACIONAL NA INDÚSTRIA E NO AGRONEGÓCIO
<http://erbase2020.ifal.edu.br>

ISBN: 978-65-87003-62-7

Porto Alegre
Sociedade Brasileira de Computação – SBC
2021

Realização:



Patrocínio:





Dados Internacionais de Catalogação na Publicação
Instituto Federal de Alagoas
Pró-Reitoria de Ensino

E74

Escola Regional de Computação Bahia, Alagoas e Sergipe (20. : 2020 : Alagoas). Minicursos da XX Escola Regional de Computação Bahia, Alagoas e Sergipe : inteligência computacional na indústria e no agronegócio, 26 a 30 de outubro de 2020 [recurso eletrônico] / organizador Felipe Alencar Lopes. - Porto Alegre : Sociedade Brasileira de Computação - SBC, 2021.
Dados eletrônicos (pdf).

Inclui bibliografia.

ISBN: 978-65-87003-62-7

1. Tecnologia da Informação. 2. Inteligência Artificial. 3. Design Thinking. 4. Estudo e aprendizagem. I. Título. II. Lopes, Felipe Alencar.

CDD:006.3

Comitê de Programa

Coordenadores de Minicursos:

Felipe Alencar Lopes | Instituto Federal de Alagoas (IFAL)
Eduardo Cardoso Moraes | Instituto Federal de Alagoas (IFAL)

Comissão de Organização do Evento

Coordenação Geral:

Felipe Alencar Lopes | Instituto Federal de Alagoas (IFAL)
Cledja Karina Rolim da Silva | Instituto Federal de Alagoas (IFAL)

Membros da Comissão Organizadora:

Mônica Ximenes Carneiro da Cunha | Instituto Federal de Alagoas (IFAL)
Tarsis Marinho | Instituto Federal de Alagoas (IFAL)
Maurício Vieira | Instituto Federal de Alagoas (IFAL)
Eduardo Moraes | Instituto Federal de Alagoas (IFAL)
Fernando Tenório | Instituto Federal de Alagoas (IFAL)
Renata Imaculada | Instituto Federal de Alagoas (IFAL)
Wellington Barbosa | Instituto Federal de Alagoas (IFAL)
Augusto César | Instituto Federal de Alagoas (IFAL)
Ígor Rosa | Instituto Federal de Alagoas (IFAL)

RESUMO

O Livro de Minicursos da ERBASE 2020 aborda conteúdos relacionados à inteligência artificial, visão computacional e processo de ensino-aprendizagem. No primeiro capítulo, intitulado “Como as Máquinas Enxergam? Um apanhado teórico e prático sobre Visão Computacional aplicada a problemas cotidianos utilizando Aprendizado de Máquina e Inteligência Artificial”, os autores comentam as bases do processamento e manipulação de imagens, suas características relevantes e como essas informações podem ser utilizadas para soluções de Aprendizado de Máquina e Inteligência Artificial. No segundo capítulo, “Utilizando o Mapa de Empatia do Design Thinking no processo de ensino-aprendizagem”, os autores buscam capacitar pessoas para a utilização da ferramenta do *Design Thinking* (DT), o mapa de empatia. Serão apresentados os conceitos básicos do DT para a Educação e os procedimentos para o preenchimento do Mapa de Empatia sobre as percepções tanto dos Docentes quanto dos Discentes envolvidos no ensino-aprendizagem. Os dois capítulos deste livro possuem metodologias e ferramentas para a área de tecnologia da informação, sendo uma obra útil principalmente para pessoas que querem iniciar nas respectivas áreas abordadas.

ABSTRACT

The content of this book is related to artificial intelligence, computer vision and the teaching-learning process. In the first chapter, entitled “How Do Machines See? A theoretical and practical overview of Computational Vision applied to daily problems using Machine Learning and Artificial Intelligence”, the authors comment on the bases of image processing and manipulation, its relevant characteristics, and how this information can be used for Machine Learning and Artificial Intelligence solutions. In the second chapter, “Using the Design Thinking Empathy Map in the teaching-learning process”, the authors aim to train people to use the Design Thinking (DT) tool, the empathy map. The basic concepts of the DT for education and the procedures for completing the Empathy Map on the perceptions of both teachers and students involved in teaching-learning will be presented. The two chapters of this book have methodologies and tools for the area of information technology, being a useful work mainly for people who want to start in the respective areas covered.

Índice

Capítulo 1. Como as Máquinas Enxergam? Um apanhado teórico e prático sobre Visão Computacional aplicada a problemas cotidianos utilizando Aprendizado de Máquina e Inteligência Artificial..... 1-24

Lucas Amparo Barbosa e Leone da Silva de Jesus

Capítulo 2. Utilizando o Mapa de Empatia do Design Thinking no processo de ensino-aprendizagem..... 25-45

Gilton José Ferreira da Silva e Tássio José Gonçalves Gomes

1

Como as Máquinas Enxergam?

Um apanhado teórico e prático sobre Visão Computacional aplicada a problemas cotidianos utilizando Aprendizado de Máquina e Inteligência Artificial

Lucas Amparo Barbosa, MSc. e Leone da Silva de Jesus, MSc.

Brazilian Institute of Robotics (BIR), SENAI CIMATEC

{lucas.barbosa, leone.jesus}@fieb.org.br

Abstract

This chapter has the main objective to present the basis for image processing and manipulation, your relevant features, and how this information would be used for Machine Learning and Artificial Intelligence solutions. We explained and compared a few relevant and/or recent works in Computer Vision, consolidating the knowledge by simplified practical examples.

Resumo

Este capítulo tem como objetivos apresentar as bases do processamento e manipulação de imagens, suas características relevantes e como essas informações podem ser utilizadas para soluções de Aprendizado de Máquina e Inteligência Artificial. Apresentamos, explicamos e comparamos alguns dos trabalhos mais relevantes e/ou recentes da área de Visão Computacional, consolidando o conhecimento com atividades práticas simplificadas.

1.1. Imagens: Teoria e Aquisição

Visão Computacional (VC) é a área de estudos da Computação que investiga processos capazes de permitir que as máquinas tomem decisões a partir de elementos do seu ambiente. Essa linha de pesquisa recebe esse nome porque, de forma análoga, tenta imitar o processo de visão humana. Portanto, é através dessas técnicas que podemos fazer as máquinas enxergarem. O órgão humano que possibilita a visão humana são os olhos. De forma similar, as câmeras são utilizadas para capturar informações do mundo que serão processadas pelo computador posteriormente. Veja um comparativo na Figura 1.1.

Este modelo básico de câmera recebe um nome: *pinhole*. É constituída, basicamente, por uma pequena abertura por onde a luz vai passar e ser capturada por um

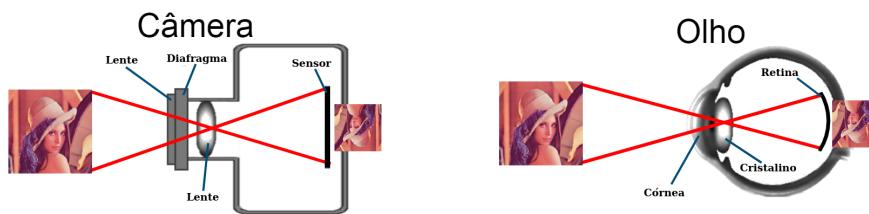


Figura 1.1. Comparativo entre o olho humano e a câmera

material fotossensível, podendo ser um filme fotográfico ou, nas câmeras mais modernas, um sensor eletrônico que gera uma imagem digital.

Outras configurações podem ser relevantes nas câmeras, como a lente e a angulação delas, o tempo que a abertura fica exposta a luz, etc. Tudo isso pode modificar significativamente a imagem gerada no final. Mas, por enquanto, não iremos abordar isso. Para simplificar, vamos adotar as câmeras como o modelo pinhole básico. Assim, poderemos evoluir nossos métodos de aquisição - ou seja, captura de dados - para outros sensores relevantes.

Os seres humanos possuem dois olhos. É graças a posição frontal desses olhos que podemos ter noção de profundidade. O nosso par de olhos nos permite saber se um objeto está próximo ou distante de nós. Em VC, também é possível simular essa característica: quando posicionamos duas câmeras em um único sistema calibrado, podemos ter informações de profundidade. Esse sistema recebe o nome de visão estéreo.

Quando falamos de um “sistema calibrado” (Figura 1.2), estamos deixando claro que as transformações entre uma câmera e outra são conhecidas. Assim, é possível criar uma correlação entre as duas imagens geradas. Dessa forma, é possível gerar informação útil para ser processada em dois domínios diferentes: A câmera simples gera imagens bidimensionais; O sistema estéreo gera informações tridimensionais.

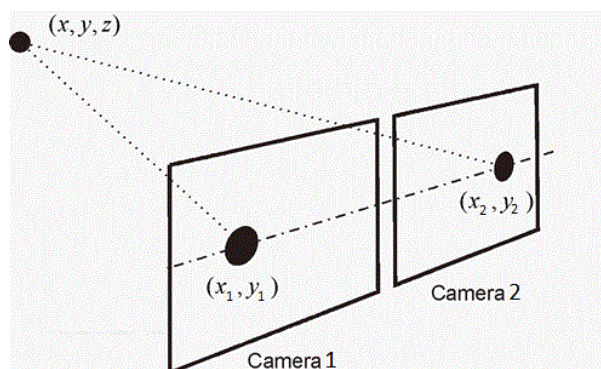


Figura 1.2. Imagens geradas em um sistema calibrado, análogo à visão estéreo.
Créditos da imagem: WikiMedia.

Outros tipos de sensores podem ser utilizados para capturar dados em duas dimensões: Sensores infravermelhos podem trazer informação de calor dos objetos e suas formas; Sinais diversos, como eletricidade ou pulsos magnéticos, podem ser processados e interpretados para recuperar informação e a interação de um objeto com o ambiente.

Tabela 1.1. Breve comparativo entre alguns tipos de imagem.

Aquisição	Vantagens	Desvantagens
Coloridas	Ampla variação de cores, similar ao olho humano	Apenas informações bidimensionais
Preto e Branca	Agilidade ao analisar escala de cinza	Único canal de cor; Apenas informações bidimensionais
Infravermelha	Informação de Calor	Apenas informações bidimensionais
Disparidade	Agilidade no processamento de formas geométricas	Baixa precisão da profundidade
Profundidade	Agilidade no processamento de formas geométricas	Fácil oclusão
Nuvens de Ponto	Melhor fonte de informações tridimensionais	Alto custo de processamento

Também existem outras formas de capturar informações tridimensionais: Ondas sonoras podem recuperar posicionamento de objetos (sonares); Projeções de luz podem ser interpretadas para recuperar informação de profundidade; etc.. A vasta gama de possibilidades para aquisição de dados permite um leque igualmente grande de opções a serem investigadas em um projeto que envolva VC. Cada tipo desses possui suas peculiaridades, vantagens e desvantagens. Na Tabela 1.1 podemos ver um breve comparativo entre algumas. A Figura 1.3 apresenta múltiplas capturas com sensores diferentes executadas numa posição próxima, utilizando o sensor RGB-D (cor + profundidade) *Kinect*¹.



Figura 1.3. Imagens geradas pelo *Kinect*. A primeira é uma aquisição por câmera comum. A segunda é um mapa de profundidade, capturando informações da forma do objeto. A última é gerada pelo sensor infravermelho.

Também é possível misturar informações de mais de uma aquisição, gerando dados mais completos. Por exemplo, podemos utilizar as informações de textura da captura de cor gerada por uma câmera em conjunto com a informação de geometria capturada pelo mapa de profundidade, constituindo, no final, uma nuvem de pontos colorida. A Figura 1.4 apresenta o resultado desse processo, ainda utilizando o *Kinect*.

1.2. Filtragem

Uma imagem é uma matriz onde cada campo possui um valor, que são os píxeis. No caso de imagens coloridas, cada píxel possui um valor para cada canal, como para vermelho, azul e verde para imagens RGB, ou tonalidade (*hue*), saturação (*saturation*) e o brilho (*lightness*) para imagens HLS. Um filtro é uma operação que será executada em cada píxel

¹<https://developer.microsoft.com/pt-br/windows/kinect/>



Figura 1.4. Imagens geradas pelo *Kinect*. Novamente, vemos o mapa de profundidade a esquerda e a captura da câmera a direita. Na segunda linha, temos dois pontos de vista diferentes da nuvem de pontos gerada com base nas duas imagens superiores.

da imagem. A relevância do filtro está nos cálculos feitos, que não levam em consideração apenas o píxel processado, mas também os píxeis próximos, num conceito de vizinhança. Para executar uma filtragem é necessário aplicar uma matriz (chamada de máscara) sobre a imagem original.

A definição da matriz vai depender do objetivo do filtro e da intensidade a ser aplicada. Por exemplo: Para suavizar uma imagem, ou seja, diminuir as divergências entre os píxeis, eu posso aplicar um filtro de média. Esse efeito é comumente chamado de desfoque (*blur*). Basta simplesmente somar todos os valores da vizinhança e dividir pela quantidade total de píxeis (uma média simples), produzindo uma imagem mais suavizada. Podemos ver um exemplo de borrão na Figura 1.5. Já na Figura 1.6, podemos ver o comparativo entre três métodos de borragem. Além do filtro de média, utilizamos um filtro gaussiano, onde o peso do píxel vai diminuindo conforme ele se afasta do centro da janela, e um filtro bilateral [20], que é ótimo para suavizar imagens sem que as informações de contraste sejam perdidas.

Outros efeitos para filtragem que também podem ser interessantes:

- *Motion Blur*: Simula o efeito de movimento na captura;
- *Sharpening*: Enfatiza divergências na imagem;



Figura 1.5. Imagem original à esquerda. À direita, imagem com aplicação do filtro de média com janela de tamanho 5.

- *Embossing*: Simula ou enfatiza iluminação;
- *Erosion*: Diminui regiões uniformes;
- *Dilation*: Aumenta regiões uniformes.



Figura 1.6. Comparativo de métodos de borragem. A primeira imagem é a original. A segunda, filtrada pela média. A terceira, utilizando um filtro gaussiano. A última, utilizando filtro bilateral.

Esses filtros podem ser utilizados com os mais diferentes objetivos. Por exemplo: observe na Figura 1.7 uma pequena planta. Esta planta possui detalhes em suas folhas que não são simples de serem vistas em imagens comuns. Para tentar enfatizar esses detalhes, você pode aplicar o filtro de *sharpening*, enfatizando as divergências entre píxeis vizinhos, fazendo assim os detalhes da imagem mais visíveis.



Figura 1.7. Aplicação do filtro de ênfase. A esquerda, imagem original. A direita, imagem filtrada.

Outro processo de filtragem muito comum e útil está no par *erosion/dilation*. Com eles é possível preencher buracos na imagem ou unir/separar objetos que estejam muito próximos. Um exemplo pode ser visto na Figura 1.8.

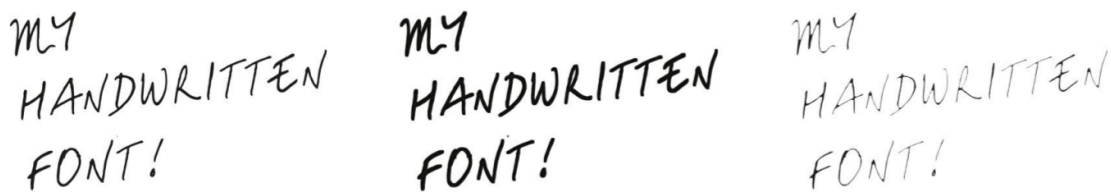


Figura 1.8. Aplicação do processo de dilatação e erosão. Imagem original, imagem dilatada e imagem erodida, respectivamente.

Os filtros não são úteis apenas para recuperar ou enfatizar o que já existe na imagem. Também é possível criar novas informações derivadas da imagem original, como é o caso dos algoritmos de borda. Utilizando filtros, podemos computar o gradiente da imagem, ou seja, a variação de intensidade em pixels vizinhos. A partir desse gradiente, as bordas da imagem ficam evidentes, podendo ser utilizadas como uma característica relevante da mesma. Várias técnicas podem ser utilizadas para a detecção das bordas. Uma das mais eficientes é a Diferença de Gaussianas, onde você irá filtrar a imagem usando dois filtros gaussianos de tamanho diferente e subtrair os resultados. Um comparativo breve desses métodos podem ser vistos na Figura 1.9.

De forma genérica, quanto mais informação relevante for enfatizada ou gerada na imagem, mais útil ela será para os processos de aprendizado de máquina. Os algoritmos utilizam essas características (em inglês *features*) para executar seus objetivos. Pense novamente no problema da folha, apresentada na Figura 1.7. Caso um especialista precise diferenciar duas plantas, a forma e os detalhes da folha são importantes para a atividade. O mesmo vale para os algoritmos inteligentes: precisamos passar para eles imagens que contenham *features* relevantes para executar a tarefa.



Figura 1.9. Vários métodos para detecção de bordas. No superior, temos a imagem original e o filtro de *Sobel*. No inferior, à esquerda temos o filtro de *Canny* e à direita a Diferença de Gaussianas.

1.3. Pontos Chave

O que identifica um objeto específico? Existem certos objetos que basta ver uma parte dele que já conseguimos dizer do que se trata. Algumas partes dos objetos são extremamente descritivas. Sabendo disso, seria possível levar essa nossa percepção para as máquinas e auxiliá-las nas atividades de reconhecimento de imagens? Sim. Atualmente, isso nem chega a ser um grande desafio. Assim como nós humanos reconhecemos determinados padrões e armazenamos essa informação em nossos cérebros, as máquinas também executam tarefas parecidas.

Esses padrões são compostos por pontos relevantes da imagem: os *keypoints* (em português, pontos-chave). Existem os mais diversos tipos deles: variação de contraste, bordas, repetição de padrão em uma região, etc. Lembra do capítulo anterior? Nele foi possível entender os processos de filtragem e como eles podem modificar as imagens. Muitos desses filtros são utilizados para criar ou enfatizar *keypoints*. Basta lembrar do filtro de *sharpening*, onde muitos detalhes da folha (Figura 1.7) foram evidenciados depois do processo de filtragem.

Entre os algoritmos de detecção de *keypoints* mais famosos, podemos destacar o SIFT, o BRISK e o ORB. O SIFT (*Scale-Invariant Feature Transform*) [14] é um dos mais eficientes processos de detecção de *keypoints*. Contudo, esse algoritmo é patenteado, impossibilitando seu uso em produtos de VC. Como potenciais soluções, podemos utilizar BRISK (*Binary Robust Invariant Scalable Keypoints*) [13], que possui uma qual-

idade semelhante ao SIFT, ou o ORB (*Oriented Fast and Rotated Brief*) [18], que é um dos algoritmos mais rápidos disponíveis. Na Figura 1.10 podemos ver o comparativo do resultado de cada um dos detectores citados.

Existem vários outros, com vantagens e desvantagens, podendo ser aplicados aos mais diversos tipos de imagens. A escolha do detector de pontos-chave vai estar muito ligado ao objetivo final do seu projeto: Vai rodar em tempo real? Tem tolerância a falhas na detecção? Tudo isso é pertinente no momento de escolher.



Figura 1.10. Vários métodos para detecção de *keypoints*. No superior, à esquerda temos a imagem original e à direita, a detecção usando SIFT. No inferior, à esquerda o resultado do BRISK e à direita, o ORB.

1.4. Descrição

No processo de utilizar imagens como dados em algoritmos de inteligência, nós já aprendemos a: melhorar o conteúdo que as imagens contém, através dos filtros; reconhecer pontos relevantes dessas imagens, os *keypoints*. Agora, precisamos padronizar as informações contidas em cada uma dessas figuras para que, assim, possamos comparar umas com as outras.

Esse processo é chamado de descrição. É bem complicado comparar duas imagens com uma numeração de *keypoints* divergente. Assim, os processos de descrição, executados pelos algoritmos descritores, conseguem receber como entrada uma coleção de tamanho irrestrito de pontos e devolver um vetor padronizado com os valores que representam aquele conjunto de informação.

O exemplo mais simples e útil para entender esse processo é feito utilizando o *His-*

togram of Oriented Gradients (HOG) [2]. Esse descritor utiliza a informação de gradiente e a orientação dos píxeis de uma imagem. Por se tratar de um histograma, é possível configurar quantos *bins* (espaços) o seu descritor vai ter. Por fim, basta distribuir os valores nesses espaços.

Outro descritor relevante é o *Local Binary Pattern Histogram* (LBPH). Semelhante ao HOG, ele também é um histograma. Contudo, dessa vez, o valor máximo vai estar relacionado ao tamanho da janela a ser utilizada no processo de binarização da imagem. Visualize exemplos de descrição nas Figuras 1.11 e 1.12.

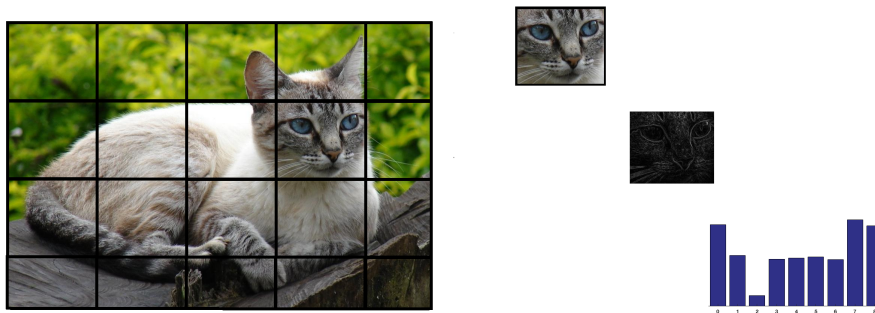


Figura 1.11. Um gatinho sendo descrito através do HOG. Cada janela irá gerar um vetor com base no gradiente de seus píxeis. Os vetores de todas as janelas serão concatenados, resultando no vetor de descrição.

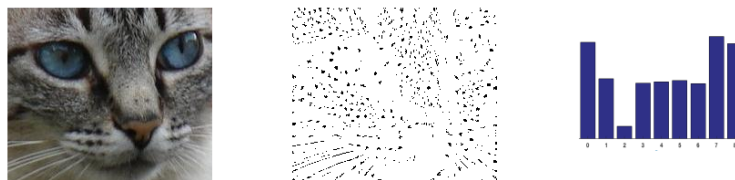


Figura 1.12. Uma das janelas da imagem do gatinho sendo descrita através do LBPH.

Outros descritores relevantes podem ser obtidos através das redes neurais. A técnica mais básica para criar um descritor desta forma é treinar um classificador básico e, durante o processo de descrição, utilizar os valores obtidos na penúltima camada da rede, aquela imediatamente anterior a camada de resultado da classificação. Um exemplo do uso desse tipo de redes será demonstrado mais adiante.

Agora, já que temos um processo padronizado para representar uma imagem, fica bem mais simples de encontrar as que são semelhantes, as que são diferentes e, até mesmo, o quanto diferente uma pode ser da outra. Distâncias bem conhecidas no ensino médio-como Euclidiana e Manhattan, ou mais refinadas, como Mahalanobis- são amplamente utilizadas para isso. No mais, descritores são ferramentas importantes para algoritmos de aprendizado de máquina, principalmente de duas formas:

- Para reduzir a dimensionalidade de um objeto durante o treinamento de um modelo de inferência, pois é bem menos custoso treinar um vetor de 100 valores do que uma imagem inteira de 200×200 píxeis;

- Em processos de validação, como uma classificação ou um agrupamento por similaridade.

1.5. Aprendizado de Máquina e Inteligência Artificial

De forma bem genérica, Inteligência Artificial (IA) são tecnologias que permitem às máquinas reproduzirem inteligência semelhante à humana. Dentro da IA, temos o Aprendizado de Máquina (*Machine Learning*, ML), que é a capacidade de criar modelos matemáticos capazes de detectar padrões em fontes de dados diversas.

Existem duas formas básicas de se treinar esses modelos de inferência: Ou passamos um conjunto de dados rotulados para o modelo aprender, num processo supervisionado de aprendizagem; ou passamos um conjunto desconhecido para que os algoritmos consigam construir os rótulos do dado apresentado, num processo não-supervisionado de aprendizado. Um modelo híbrido é chamado de semi-supervisionado, onde características dos dois podem ser utilizados. A Figura 1.13 ilustra os dois modelos básicos de treinamento.

No treinamento supervisionado, os rótulos informados são utilizados para corrigir o processo de aprendizagem. Assim, quando um novo dado for inserido, o modelo matemático construído anteriormente vai ser capaz de atribuir um rótulo para ele. Já no treinamento não-supervisionado, devido à ausência de rótulos, o modelo precisa extrair informação do que tem "em mãos". Por exemplo, imagine um modelo que seja capaz de classificar um veículo em carro, moto ou ônibus. Caso os rótulos do treinamento não sejam informados, precisamos criá-los. Motos tem 2 rodas, carros tem 4 e ônibus tem, pelo menos, 6. Isso já pode servir de base para criar as bases do classificador. Contudo, como o modelo não tem como deduzir os nomes "moto", "carro" e "ônibus", irá atribuir classe "1", "2" e "3", exigindo do programador que ele, explicitamente, faça a conversão das classes para algo compreensível para humanos (caso seja necessário).

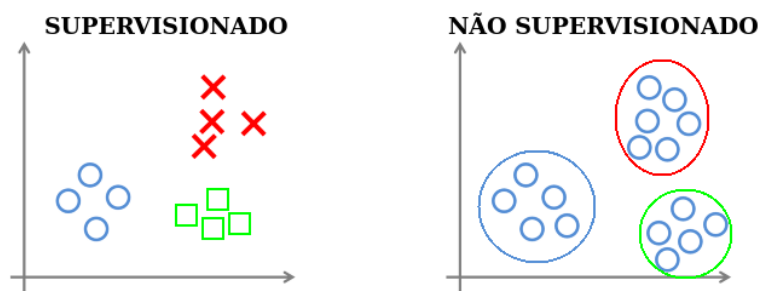


Figura 1.13. Imaginando um problema qualquer de aprendizado. Podemos ter dados já rotulados, como apresentado no gráfico à esquerda, ou podemos ter dados onde não se tem informação, como os apresentados na direita. Uma nova instância de dados tentará resolver o problema usando a informação derivada dos dados inseridos.

Problemas mais simples, que são facilmente detectáveis, como variações de cor ou formato dos objetos, podem ser resolvidos com técnicas menos refinadas, como SVM (*Support Vector Machine*) ou Árvores de Decisão. Quando encontramos problemas mais complexos, que envolvem análise de vizinhança ou coesão temporal, por exemplo, torna-se necessário utilizar técnicas mais refinadas, como as redes neurais e, até mesmo, as

redes profundas.

A dificuldade de um treinamento utilizando redes profundas é que, além de exigir um maior custo computacional para criar o modelo matemático, também é necessário que se tenha um conjunto relevante de dados a ser utilizado no treino. Relevância essa que varia de caso em caso.

Imagine o seguinte problema: um conjunto de imagens que contenham carros e barcos. São dois objetos muito diferentes, em seu formato principalmente. Não há necessidade de utilizar uma técnica muito avançada para resolver esse caso. Agora, imagine que o seu conjunto seja apenas de carros e seu desafio seja classificar o modelo de cada um. Bem mais complexo, pois carros do mesmo estilo tendem a ser parecidos. Por conta disso, precisamos ter um modelo matemático mais robusto e muito mais imagens para treinar. A rede irá aprender a extrair características relevantes de cada modelo e, assim, conseguir acertar a classe de cada um.

1.6. Problemas Comuns: Classificação

Um problema clássico da área de ML é a classificação. Este problema pode ser definido como "A capacidade de separar um conjunto de dados em N classes". Para entender melhor como esse problema é feito, iremos demonstrar um problema relativamente simples: Dado um conjunto de frutas, classificar corretamente cada uma delas. A Figura 1.14 apresenta como um classificador pode funcionar.

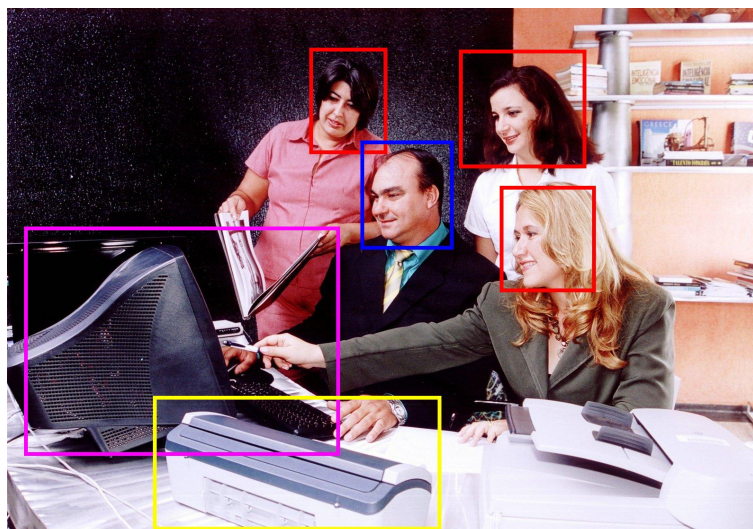


Figura 1.14. Ilustração de um processo de classificação. Em vermelho, mulher. Azul, homem. Rosa, computador. Amarelo, impressora.

O conjunto de imagens é formado por 28 frutas diferentes, como maçãs, abacaxis, peras, etc. (Veja Figura 1.15), e utilizamos 200 exemplos de cada classe. O objetivo da classificação é: dada uma nova imagem nunca antes vista pelo modelo matemático, atribuir uma classe correta para ele. Vale salientar que a classe da imagem deve existir no conjunto de imagens, porém aquele exemplo não pode ter sido utilizado no processo de treinamento. Uma boa forma para executar o treinamento parte do conjunto de imagens para treinamento e outra parte para a avaliação. Assim, podemos garantir que as classes de

treino e teste serão as mesmas, porém sem reutilização de membros. No nosso exemplo, utilizaremos 70% das imagens para treinamento e 30% para avaliação do modelo. Foram treinados três modelos básicos: um SVM, uma Árvore de Decisão e uma rede neural *Multilayer Perceptron* (MLP).

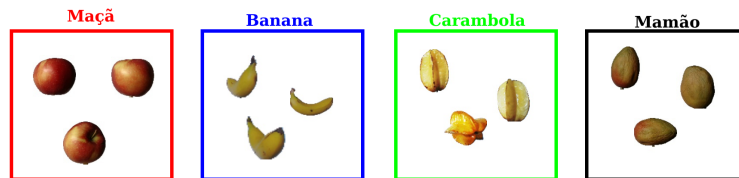


Figura 1.15. Exemplos de frutas presentes no conjunto de imagens.

Podemos então analisar mais a fundo como esse modelo pode ser treinado. Utilizamos a biblioteca *sklearn*, disponível em Python, para simplificar a implementação. Primeiro, carregamos a imagem em escala de cinza e utilizamos o HOG como descritor. Em seguida, informamos para os modelos um conjunto de dados e seus respectivos rótulos. Com esses dados, os modelos irão aprender a separar as classes. Para avaliar a precisão do mesmo, utilizamos outro conjunto de dados, que não foi usado durante o treinamento. Pegamos as predições do modelo para cada uma das imagens de teste e comparamos com a resposta correta. Assim, conseguiremos mensurar o quão bom é a técnica utilizada. A biblioteca *sklearn* já possui uma série de ferramentas para uso e validação de técnicas de aprendizado de máquina. Veja o exemplo no Algoritmo 1.1.

Algoritmo 1.1. Classificadores Simples

```

from sklearn import svm, tree
from sklearn.neural_network import MLPClassifier as mlp
from skimage.feature import hog
import numpy as np

def trainSVM(X_train, y_train):
    sig = svm.SVC(kernel='sigmoid', C=1).fit(X_train, y_train)
    return sig

def trainTree(X_train, y_train):
    sig = tree.DecisionTreeClassifier().fit(X_train, y_train)
    return sig

def trainMLP(X_train, y_train):
    sig = mlp(random_state=1).fit(X_train, y_train)
    return sig

def validModel(model, X_test, y_test):
    y_pred = model.predict(X_test)

    acc = 0.0
    for i in range(len(y_pred)):
        if y_test[i] == y_pred[i]:
            acc += 1

    return acc/len(y_test)

X_train, y_train, X_test, y_test = load_dataset("/path/to/dataset")

msvm = trainSVM(X_train, y_train)
mtree = trainTree(X_train, y_train)
mmlp = trainMLP(X_train, y_train)

```

```

acc_svm = validModel(msvm, X_test, y_test)
acc_tree = validModel(mtrees, X_test, y_test)
acc_mlp = validModel(mmlp, X_test, y_test)

```

Os resultados médios são apresentados na Tabela 1.2. Observe que a precisão do SVM é muito baixa, tornando o modelo pouco útil para esse problema. Quando utilizamos a Árvore de decisão, essa métrica já começa a se tornar mais relevante, acertando cerca de 6 a cada 10 tentativas. Porém, o resultado da MLP é, sem sombra de dúvidas, o melhor, praticamente não errando a classificação. Esse comportamento é comum na área e representa o porquê das redes neurais serem tão utilizadas na maioria dos problemas de ML.

Tabela 1.2. Comparativo entre alguns tipos de imagem.

Método	Tempo (s)	Precisão Média
Classificador SVM	~20	~40%
Árvore de Decisão	~4	~68%
MLP + HOG	~12	~99%

1.7. Problemas Comuns: Detecção

Uma especialização dos classificadores é utilizar os seus rótulos gerados para poder encontrar objetos relevantes em imagens complexas. Quando os modelos são treinados, geralmente são utilizadas imagens onde apenas um objeto está inserido. Mas problemas do mundo real não são assim. As imagens do mundo real, conhecidas como capturas *in the wild*, são mais complicadas: Inúmeros objetos, da mesma classe ou não; divergências de iluminação; posicionamento diferente; objetos escondidos ou apenas parcialmente visíveis; etc.. A Figura 1.14, apresentada para exemplificar a classificação, também é útil para entender a detecção.

A detecção é muito utilizada em conjunto com outros problemas pois diminui a quantidade de dados a ser processada por outros processos. Imagine o seguinte problema: Um sistema de reconhecimento facial identifica o rosto de um usuário qualquer. Sendo assim, qual a necessidade de enviar para o processo de identificação o corpo do indivíduo? Podemos utilizar um classificador de rosto humano para detectar uma ou mais faces na imagem, recortá-las e enviar apenas essas partes para o reconhecimento. A Figura 1.16 apresenta uma ilustração do uso da detecção.

1.8. Problemas Comuns: Segmentação

A segmentação é um dos desafios que surgiu na área de VC quando os algoritmos acadêmicos foram transportados do ambiente controlado de laboratório para o mundo real. O problema pode ser definido como a forma de identificar quais pixels na imagem pertencem a uma (ou várias) classe(s) determinada(s). Apesar de similar à ideia da detecção, a segmentação visa delimitar precisamente a área da imagem que compõe cada elemento, mesmo que seja somente a diferença entre um objeto e o plano de fundo (Figura 1.17).

Essa segmentação permite, por exemplo, uma melhor análise de elementos na imagem quando o plano de fundo ou o comportamento entre elementos da imagem pode



Figura 1.16. Exemplo da detecção aplicada a rostos humanos. A saída desse processo é muito útil em um sistema de reconhecimento facial.



Figura 1.17. Exemplo da segmentação de carros em uma imagem qualquer.

impactar nos resultados. Existem basicamente dois tipos de segmentação: a semântica (Figura 1.18), que separa pixels de uma mesma classe na imagem, e de instância (Figura 1.19), que define os pixels de cada instância conhecida da imagem, mesmo sendo da mesma classe. A disposição dos elementos na imagem ou a forma que eles se relacionam geram informações que não seriam possíveis de adquirir somente com a detecção e classificação (*i.e.* pessoa montada ou atrás da moto, visão de um carro obstruída por outro carro).

O Algoritmo 1.2 é um exemplo de como utilizar a segmentação semântica por meio da biblioteca *pixelLib*¹. Essa biblioteca permite o uso de forma simples de modelos prontos para segmentação semântica e de instâncias, que estão disponíveis para baixar. O treinamento desses modelos incluiu dezenas de classes distintas, então o uso destes só permite a segmentação dessas classes. Contudo, a biblioteca também oferece meios de adicionar classes personalizadas, também de forma simples, contanto que seja informado um conjunto de imagens e suas marcações seguindo uma estrutura de diretórios pré-estabelecida.

Algoritmo 1.2. Segmentador Simples

```
import cv2
import pixelLib
from pixelLib.semantic import semantic_segmentation
```

¹<https://github.com/ayoolaolafenwa/pixelLib>



Figura 1.18. A imagem possui diferentes classes, representadas por pixels de diferentes cores (segmentação semântica).

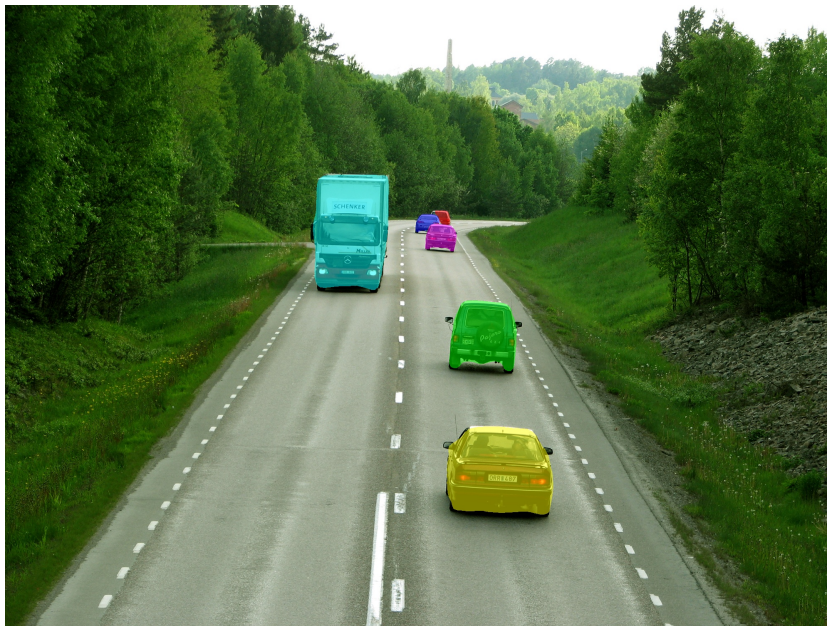


Figura 1.19. Objetos da mesma classe (veículo), mas cada um representado por pixels de uma cor distinta (segmentação de instâncias).

```
def segmentacao_semantica(caminho_imagem):
    segmentador = semantic_segmentation()
    segmentador.load_pascalvoc_model("deeplabv3_xception_tf_dim_ordering_tf_kernels.h5")
    _, area_segmentada = segmentador.segmentAsPascalvoc(caminho_imagem)
    cv2.imwrite("area_segmentada.png", area_segmentada)
    return area_segmentada

caminho_imagem = "imagem_com_fundo.png"
imagem_original = cv2.imread(caminho_imagem)
fundo = cv2.imread('fundo.png')
area_segmentada = segmentacao_semantica(caminho_imagem)
```

```

linhas,colunas,canais = area_segmentada.shape
roi = fundo[0:linhas, 0:colunas]
cinza = cv2.cvtColor(area_segmentada, cv2.COLOR_BGR2GRAY)

ret, mascara = cv2.threshold(cinza, 10, 255, cv2.THRESH_BINARY)
mascara_inversa = cv2.bitwise_not(mascara)

recorte_fundo = cv2.bitwise_and(roi, roi, mask = mascara_inversa)
recorte_mascara = cv2.bitwise_and(imagem_original, imagem_original, mask = mascara)

montagem = cv2.add(recorte_fundo, recorte_mascara)
fundo[0:linhas, 0:colunas ] = montagem
cv2.imwrite("resultado.png", fundo)

```

Além do Algoritmo 1.2 demonstrar o fácil uso da biblioteca de segmentação, ela também evidencia que o conhecimento sobre VC é essencial para o desenvolvimento da sua própria aplicação. Com poucos comandos do OpenCV ¹, conseguimos utilizar a área segmentada de um ser humano e aplicar um diferente plano de fundo. A Figura 1.20 mostra os resultados. Essa aplicação só era utilizada no passado com planos de fundos de cores únicas e específicas, mas utilizando modelos de redes neurais para segmentação, que já é um problema bem consolidado na literatura, é possível obter resultados com humanos em qualquer plano de fundo comum.

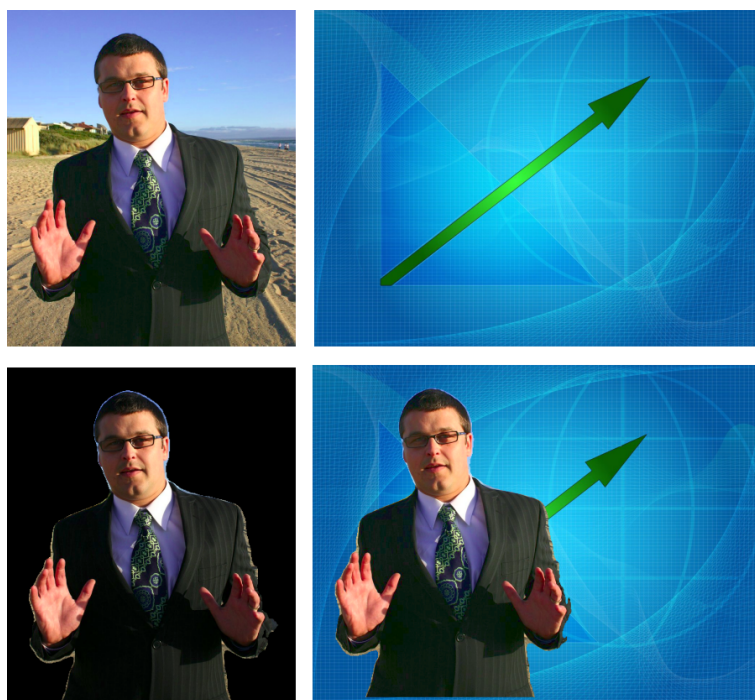


Figura 1.20. Em cima, imagens utilizadas como entrada do código. Em baixo, a imagem do humano segmentada e a montagem com um novo plano de fundo.

1.9. Um exemplo prático: Reconhecimento Biométrico

Há anos que o reconhecimento biométrico é estudado para diversos fins, especialmente para controle de acesso, seja de instalações militares ou dispositivos móveis, podendo proteger desde o acesso ao uso de recursos perigosos a informações pessoais. Diferentes

¹<https://opencv.org/>

características físicas e comportamentais tem sido experimentadas na literatura, mas as mais distintivas e de mais fácil coleta (*e.g* face, impressão digital e íris) [11] foram as mais exploradas, possuindo níveis de confiança aceitáveis até mesmo para modelos já publicamente disponíveis. O reconhecimento de faces, por exemplo, apesar de ainda ser alvo de pesquisas, já superou até mesmo a capacidade humana nessa tarefa em ambientes não controlados [23], e vários modelos podem ser encontrados para o fácil uso de quem tem um mínimo conhecimento de VC. O Algoritmo 1.3 é um exemplo simples de uma aplicação que faz verificação facial.

Algoritmo 1.3. Verificação Facial

```

import cv2
import numpy
from mtcnn.mtcnn import MTCNN
from keras.models import load_model

def le_imagem(nome_arquivo):
    img = cv2.imread(nome_arquivo)
    return img

def recorta_face(imagem, detector, margem):
    ponto_x, ponto_y, largura, altura = detector['box']
    ponto_x -= margem
    ponto_y -= margem
    largura += 2*margem
    altura += 2*margem
    if ponto_x < 0:
        largura += ponto_x
        ponto_x = 0
    if ponto_y < 0:
        altura += ponto_y
        ponto_y = 0
    return imagem[ponto_y:ponto_y+altura, ponto_x:ponto_x+largura]

def detecta_recorta(modelo_deteccao, imagem):
    faces_detectadas = modelo_deteccao.detect_faces(imagem)[0]
    margem = int(0.1 * imagem.shape[0])
    detecta_recorta = recorta_face(imagem, faces_detectadas, margem)
    return detecta_recorta

def pre_processamento(face, required_size=(160, 160)):
    ret = cv2.resize(face, required_size)
    ret = ret.astype('float32')
    mean, std = ret.mean(), ret.std()
    ret = (ret - mean) / std
    return ret

def extrai_caracteristicas(amostras):
    amostras = numpy.asarray(amostras, 'float32')
    model = load_model('facenet_keras.h5')
    yhat = model.predict(amostras)
    return yhat

def verificacao(nome_imagem, confianca, imagem):
    imagem = cv2.resize(imagem, (500, 500))
    local_texto = (50, 100)
    fonte = cv2.FONT_HERSHEY_SIMPLEX
    tamanho_fonte = 2
    largura = 5
    limiar = 10
    if confianca < limiar:
        imagem = cv2.putText(imagem, str(confianca),
            local_texto, fonte, tamanho_fonte, (0, 255, 0), largura)
    else:
        imagem = cv2.putText(imagem, str(confianca),
            local_texto, fonte, tamanho_fonte, (0, 0, 255), largura)
    cv2.imwrite(nome_imagem, imagem)

```

```

trump1 = le_imagem("Trump.jpg")
trump2 = le_imagem("Trump_2.jpg")
clinton = le_imagem("Clinton.jpg")
obama = le_imagem("Obama.jpg")

modelo_deteccao = MTCNN()
trump1 = detecta_recorta(modelo_deteccao, trump1)
trump2 = detecta_recorta(modelo_deteccao, trump2)
clinton = detecta_recorta(modelo_deteccao, clinton)
obama = detecta_recorta(modelo_deteccao, obama)

imagens = [trump1, trump2, clinton, obama]
amostras = [pre_processamento(i) for i in imagens]

caracteristicas = extrai_caracteristicas(amostras)

confianca_trump_trump = numpy.linalg.norm(caracteristicas[0] - caracteristicas[1])
confianca_trump_clinton = numpy.linalg.norm(caracteristicas[0] - caracteristicas[2])
confianca_trump_obama = numpy.linalg.norm(caracteristicas[2] - caracteristicas[3])

verificacao("TrumpxTrump.png", confianca_trump_trump, trump2)
verificacao("TrumpxClinton.png", confianca_trump_clinton, clinton)
verificacao("TrumpxObama.png", confianca_trump_obama, obama)

```

Todos os passos para esse reconhecimento de faces foram explicados previamente, mesmo que de forma diferente. Começando pela detecção fácil, aqui feita utilizando um modelo bem conhecido para o problema chamado MTCNN [22]. A imagem da face, extraída da original, passa por um processamento para ficar com as mesmas características das imagens que foram utilizadas no treino da rede de reconhecimento facial. Tal rede vem de um modelo também bem conhecido para transformar imagens em uma representação de seus principais componentes, como explicado na Seção 1.4. Esse processo é também chamado de extração de características. A comparação entre essas características, utilizando distâncias descritas previamente, retorna um valor que representa confiança do seu método de que as características pertencem à mesma pessoa. Então a aplicação só precisa decidir se aquele valor é suficiente para validar a verificação facial. Esse processo é a base para todos os reconhecimentos biométricos, seja para verificação ou identificação, e seu fluxo está presente na Figura 1.21.

Os resultados presentes na Figura 1.22 demonstram a eficácia dos modelos utilizado tanto para detecção da face quanto para o reconhecimento. Isso é mais uma demonstração de que para aplicações já pesquisadas há muito tempo, alguém na comunidade científica já teve o esforço de treinar um modelo eficaz, mas o conhecimento mínimo de VC é essencial para você saber como adaptá-los à sua aplicação.

1.10. O que há de bom na Literatura?

Todas as técnicas apresentadas e as que, por ventura, sejam conhecidas após a leitura desse material, estão publicadas nas mais diversas fontes: IEEEExplore¹, Nature², Elsevier³, etc.. Contudo, o objetivo aqui é entender como os processos de visão computacional podem ser aplicados, em conjunto com a ML e IA, para resolver problemas da área in-

¹<https://ieeexplore.ieee.org/>

²<https://www.nature.com/>

³<https://www.elsevier.com/pt-br>

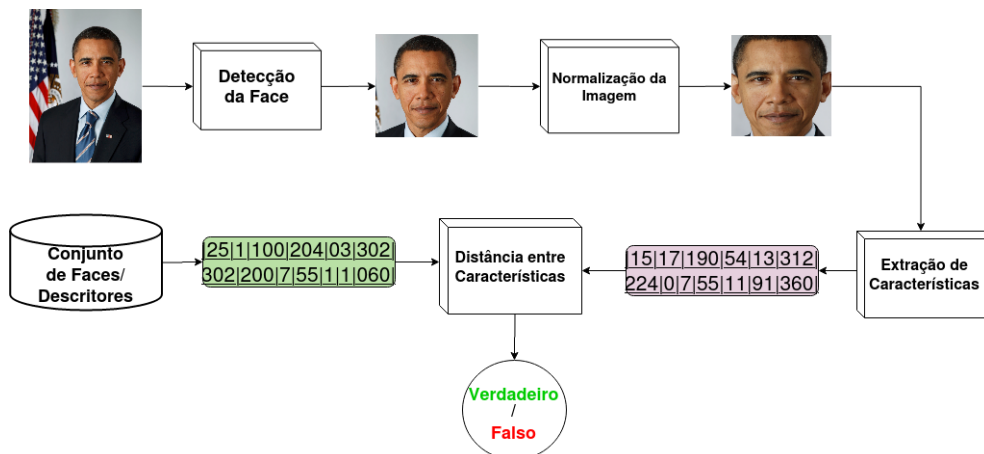


Figura 1.21. Diagrama dos passos realizados no Algoritmo 1.3. Esse é o processo básico de aplicações de biometria. Nesse caso, temos uma verificação facial. Para a identificação biométrica, o processo é similar, mas os descritores extraídos são comparados com os descritores de todas as pessoas conhecidas pela base, considerando a de maior similaridade como a pessoa identificada. Originais retiradas de Wikimedia.



Figura 1.22. No Algoritmo 1.3 realizamos uma comparação entre uma face e três outras. A verificação diz se são pessoas distintas, colocando o resultado de sua confiança de que são a mesma pessoa em verde (se positivo) e vermelho (se negativo). Numa aplicação real, o resultado dessa verificação poderia ser utilizada facilmente em diversas aplicações em que se tivesse uma câmera. Originais retiradas de Wikimedia.

dustrial. Por isso, selecionamos alguns trabalhos relevantes e interessantes para diversas áreas de atuação: agricultura, energia, alimentação, etc..

Um sistema de inspeção visual é composto, basicamente, por uma fonte de iluminação (natural ou não), um dispositivo de captura de imagem (independente de qual tipo) e poder computacional para processar os dados, transformando a imagem em informação

relevante. Quando bem parametrizados, esses sistemas permitem uma maior confiabilidade e repetibilidade nos processos de avaliação de qualquer objeto que seja necessário.

A revisão literária apresentada por Gomes & Leta (2012) [6] demonstra como as técnicas de visão computacional são utilizadas há anos para aprimorar os processos de inspeção nas mais diversas indústrias alimentares. Existem trabalhos para controle de pestes, diminuindo custos com pesticidas e permitindo uma medida preventiva antes que uma maior parcela da produção esteja comprometida. Também foi possível executar medições de quanto do campo estava sendo utilizado com plantações úteis ou ervas daninhas e, assim, mensurar a área útil e produtiva da propriedade. Os problemas encontrados nessas abordagens esbarram em iluminações divergentes, a depender da hora do dia ou do nível de maturação da plantação.

Ainda analisando a revisão de Gomes & Leta, também é possível perceber que os processos de visão computacional podem auxiliar na detecção e classificação de defeitos nos mais diversos tipos de vegetais, como batatas ou maçãs. Análises em formato, tamanho, cor e uniformidade são feitas para identificar produtos de alta ou baixa qualidade, permitindo uma melhor separação e precificação do que será entregue aos clientes.

Não foram somente as lavouras foram beneficiadas por essas evoluções tecnológicas. Os processos industriais também podem ser otimizados por meio da computação. A revisão da literatura trouxe uma pesquisa que utilizava informações espectrais em conjunto com imagens capturadas para avaliar a qualidade de vinhos produzidos, fazendo assim dois métodos de aquisição diferentes trabalharem em conjunto.

Assim como a otimização dos processos é relevante, também é a tentativa de transformar os produtos resultantes em algo mais agradável aos olhos humanos. É o caso de utilizar câmeras para avaliar a distribuição de ingredientes em pizzas ou treinar modelos matemáticos que auxiliem as máquinas a montarem pratos de refeições prontas que estejam mais parecidas com os pratos montados por humanos, também listados na publicação de Gomes & Leta.

Saindo um pouco da área alimentícia, o trabalho apresentado por Fathi, Dai & Lourakis (2015) [3] demonstra o uso da VC na área de Engenharia Civil. A publicação apresenta uma revisão das bases técnicas que possibilitam a reconstrução 3D de infraestruturas já existentes, possibilitando assim as mais diversas análises como a necessidade de reformas estruturais ou simular o efeito de forças adversas na construção. Também são citados no artigo os principais desafios encontrados nesse tipo de pesquisa, como o custo computacional de construção do objeto 3D ou a forte dependência que o resultado tem de características distintivas em toda a extensão do alvo da reconstrução. Esse trabalho é interessante por trazer a tona tanto as qualidades quanto os desafios do uso de tal tecnologia.

Algumas empresas de construção civil já fazem o uso de modelagem tridimensional durante a execução dos seus projetos (Modelo BIM [16], por exemplo), porém fazer a digitalização de estruturas já construídas, normalmente chamadas de *as-built*, ainda é um problema desafiador para a área. Outras abordagens úteis à área de Construção Civil está na detecção, classificação, avaliação e monitoramento de situações adversas no alvo de inspeção. Essas adversidades podem ser defeitos relevantes, como rachaduras em as-

falto [12] ou falhas estruturais em construções [4, 19].

Na área de geração e distribuição de energia elétrica, Mirallès, Pouliot & Montambault (2014) [15] apresentaram uma revisão da literatura com ênfase na gestão de linhas de transmissão: objetos muito grandes, caros e fundamentais para o funcionamento do negócio da empresa. Detecção e Inspeção desses objetos (presença de ferrugem, ausência de peças, etc.) são amplamente discutidos na publicação, demonstrando como a VC pode contribuir na manutenção de uma indústria vital para a sociedade moderna.

Trazendo nossa leitura para problemas mais "modernos", a Indústria 4.0 está intimamente ligada à automatização de processos com suporte de IA/ML e estas, intimamente ligadas com Visão Computacional. Villalba-Diez et al. (2019) [21] nos apresenta um trabalho utilizando técnicas de aprendizado profundo para monitorar e garantir a qualidade nos processos de Impressão 3D, agregando valor ao produto final produzido e reduzindo substancialmente os custos da produção.

Todos esses trabalhos foram selecionados apenas pela ênfase aplicada a problemas industriais. As linhas de pesquisa derivadas da VC são muito grandes, com vários problemas ainda em aberto e um potencial pouco explorado, principalmente pelo mercado brasileiro. Se fossemos analisar outras pesquisas desenvolvidas e relevantes, precisaríamos de uma especialização só para isso. O importante no momento é saber da existência dessa série de trabalhos e toda a sua aplicabilidade para problemas industriais. Na próxima seção, algumas ideias serão imaginadas. Um dia elas serão desenvolvidas? Só o tempo dirá.

1.11. O que o futuro nos reserva?

O futuro da Indústria está atrelado ao termo "4.0". Esse termo foi cunhado para descrever uma possível quarta revolução industrial, onde os processos serão desenvolvidos utilizando tecnologias que permitam a fusão entre o mundo físico, digital e biológico. Como? Através da Manufatura Aditiva (Impressão 3D), Inteligência Artificial, Internet das Coisas, Biologia Sintética e Sistemas Cyber-físicos. A grande sacada dessa nova revolução está na interação entre as partes. Sendo assim, é necessário que algumas informações do mundo real sejam inseridas no mundo digital. E como isso pode ser feito? Utilizando técnicas de Visão Computacional.

Vamos pensar na linha da Manufatura Aditiva. Para construir um novo objeto que será impresso, é necessária a edição dele utilizando um software. Porém, caso já exista algo similar no mundo real, esse objeto pode ser escaneado e carregado no sistema, agilizando o trabalho do designer responsável, que poderá focar apenas nas melhorias a serem desenvolvidas. Uma vez que o objeto esteja pronto, no mundo digital, ele será impresso de fato. Durante o processo, técnicas de monitoramento podem ser aplicadas para impedir que erros de grande escala aconteçam. Métodos de inspeção podem ser utilizados no produto final para validar se o objeto impresso é realmente aquilo que foi projetado.

No contexto de Internet das Coisas (IoT), que consiste em trazer o poder de tomar algumas decisões para as "pontas" das estruturas, descentralizando o processamento de informações, as coisas ficam um pouco mais interessantes. Podemos pensar em um prob-

lema relativamente simples para aplicar: Imagine uma caldeira química, que trabalhe sempre em uma temperatura qualquer. Uma câmera térmica, instalada próxima a essa caldeira, pode auxiliar nos processos de segurança, permitindo o monitoramento da temperatura. A tomada de decisão resultante da análise dessa imagem térmica não precisa ser centralizada, pois, uma vez informado que material está sendo "fervido" na caldeira, um processador simples é capaz de perceber se o aumento de temperatura, dada uma janela de tempo, é perigosa ou não.

Dos processos da Indústria 4.0, os Sistemas Cyber-físicos são, provavelmente, os mais desenvolvidos na atualidade. Já temos, por exemplo, braços robóticos que executam as mais diversas tarefas, como montagem de equipamentos ou coleta de materiais. Porém, ainda existem muitas outras aplicabilidades. Colheita automatizada, onde um robô é capaz de analisar se um vegetal está ou não em ponto de colheita, executando o processo com alta precisão e em um intervalo de tempo muito mais curto.

Todas essas linhas podem utilizar, como base, processos de Inteligência Artificial. Quanto mais automático o processo for, menor a interação humana. E, ao diminuir essa interação, há a necessidade de trazer dados do mundo real para o computador. Se a tendência industrial é a automação, não há dúvidas de como o conhecimento em Visão Computacional vai se tornar cada vez mais relevante e aplicável nas mais diversas atividades industriais. Que processo não precisa de inspeção e monitoramento? Todos os produtos não precisam ter sua qualidade analisada?

Referências

- [1] Baggio, D. L. et al. (2017) "Mastering OpenCV 3". Packt Publishing.
- [2] Dalal N. e Triggs B.. (2005) "Histograms of oriented gradients for human detection," In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 2005, pp. 886-893.
- [3] Fathi, H., Dai, F. e Lourakis, M.. (2015). "Automated as-built 3D reconstruction of civil infrastructure using computer vision: Achievements, opportunities, and challenges" In: Advanced Engineering Informatics, 29(2), p. 149–161.
- [4] Feng, D. e Feng, M. Q. (2018) "Computer vision for SHM of civil infrastructure: From dynamic response measurement to damage detection – A review" In: Engineering Structures, Volume 156, p. 105-117.
- [5] Garreta, R. et al. (2017) "Scikit-learn : Machine Learning Simplified: Learning Path". Packt Publishing.
- [6] Gomes, J. F. S., e Leta, F. R. (2012) "Applications of computer vision techniques in the agriculture and food industry: a review". In: European Food Research and Technology 235, p. 989–1000.
- [7] Gonzalez, R. C. e Woods, R. E. (2007) "Digital Image Processing". Pearson.
- [8] Goodfellow, I. et al. (2015) "Deep Learning". MIT Press.
- [9] Gulli, A. e Pal, S.. (2017) "Deep Learning with Keras". Packt Publishing.

- [10] Howse, J. et al. (2016) "OpenCV: Computer Vision Projects with Python: Learning Path". Packt Publishing.
- [11] Jain, A. K., Flynn, P. e Ross, A. A.. (2007) "Handbook of Biometrics". Springer-Verlag New York, Inc.
- [12] Koch, C. et al. (2015) "A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure" In: Advanced Engineering Informatics, Volume 29, Issue 2, p. 196-210.
- [13] Leutenegger, S., Chli M. e Siegwart R. Y.. (2011) "BRISK: Binary Robust invariant scalable keypoints," In: International Conference on Computer Vision, Barcelona, 2011, pp. 2548-2555.
- [14] Lowe, D. G.. (2004) "Distinctive Image Features from Scale-Invariant Keypoints" In: International Journal of Computer Vision 60, p. 91–110.
- [15] Miralles, F., Pouliot, N., e Montambault, S.. (2014). "State-of-the-art review of computer vision for the management of power transmission lines" In: Proceedings of the 2014 3rd International Conference on Applied Robotics for the Power Industry.
- [16] Mohandes, S. R. e Omrany, H.. (2013). "Building Information Modeling in Construction Industry: Review Paper" In: International Conference of Seminar Kebangsaan Aplikasi Sains & Mathematic (SKASM).
- [17] Rodrigues, V.. (2018) "Como Máquinas Aprendem: Fundamentos E Algoritmos de Machine Learning, Redes Neurais E Deep Learning". Independently Published.
- [18] Rublee, E. et al.. (2011). "ORB: an efficient alternative to SIFT or SURF" In: Proceedings of the IEEE International Conference on Computer Vision. 2564-2571.
- [19] Spencer, B. F., Hoskere, V. e Narazaki, Y.. (2019) "Advances in Computer Vision-Based Civil Infrastructure Inspection and Monitoring" In: Engineering, Volume 5, Issue 2, p. 199-222.
- [20] Tomasi, C. e Manduchi, R.. (1998) "Bilateral filtering for gray and color images," In: Sixth International Conference on Computer Vision. p. 839-846.
- [21] Villalba-Diez, J. et al.. (2019). "Deep Learning for Industrial Computer Vision Quality Control in the Printing Industry 4.0" In: Sensors, 19(18), 3987.
- [22] Zhang, K. et al.. (2016) "Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks". CoRR (ArXiv).
- [23] Zheng, T., Deng, W. e Hu, J.. (2018) "Cross-Pose LFW: A Database for Studying Cross-Pose Face Recognition in Unconstrained Environments". CoRR (ArXiv).

Capítulo

2

Utilizando o Mapa de Empatia do *Design Thinking* no processo de ensino-aprendizagem

Gilton José Ferreira da Silva, Tássio José Gonçalves Gomes

Abstract

The teaching-learning processes are changing due to the New Information and Communication Technologies (NTIC) and with this the role of the teacher has also been transformed. Among the various innovative processes, Design Thinking (DT) stands out, a methodology aimed at creating new solutions through a set of collaborative tools. Given this, the objective of this work is to train people to use the DT tool, Empathy Map. The basic concepts of DT for Education and the procedures for completing the Empathy Map on the perceptions of both Teachers and Students involved in teaching-learning will be presented.

Resumo

Os processos de ensino-aprendizagem estão modificando-se devido as Novas Tecnologias da Informação e Comunicação (NTIC) e com isso o papel do professor também tem se transformado. Dentre os diversos processos inovadores destaca-se o Design Thinking (DT), uma metodologia voltada para a criação de novas soluções por meio de um conjunto de ferramentas colaborativas. Diante disto, o objetivo deste trabalho é capacitar pessoas para a utilização da ferramenta do DT, Mapa de empatia. Serão apresentados os conceitos básicos do DT para a Educação e os procedimentos para o preenchimento do Mapa de Empatia sobre as percepções tanto dos Docentes quanto dos Discentes envolvidos no ensino-aprendizagem.

2.1. Introdução

Referente aos processos de ensino-aprendizagem, envolvendo a empatia, se destacam os estudos desenvolvidos por Vygotsky (1896-1934) [da Fonseca 2019]. Empatia, de maneira ampla, pode ser compreendida como o conjunto de manifestações humanas que envolve o conhecimento do outro, ideias e sentimentos sobre os acontecimentos do dia a dia [Brolezzi 2014].

As escolas tradicionais possuem em seus processos de ensino-aprendizagem a figura central do professor, sendo este tratado como o portador de todo o conhecimento e o discente, sendo este o que está buscando o conhecimento [Cruz 2008].

Os discentes esperam que seus professores demonstrem atitudes empáticas, como facilitador e motivador das aprendizagens [Silva 2015]. Sabe-se também que no Brasil temos nas escolas e universidades discentes com os mais variados perfis e com mais variados problemas emocionais e sociais. Sendo assim, a compreensão empática poderá ser um fator importante para melhorar a relação pedagógica entre professor-discente [Silva 2015].

De acordo com Vianna, [Vianna 2012], o Design Thinking (DT) veio da necessidade de novas formas para inovar que de forma multidisciplinar, a colaborativa e a tangível. Tim Brown, criador do DT, começou a utilizar o termo "thinking" junto ao "design" devido a influência de seu amigo David Kelly, para descrever princípios que podem ser aplicados em diversos tipos de problemas [Brown 2020].

Especificamente voltado para auxiliar educadores em seus desafios diários, [Educadigital 2018] reuniu um conjunto de materiais e os nomeou como *Design Thinking* para educação. Trata-se de um material cujo foco são as pessoas envolvidas no processo e como podem ser desenvolvidos valores que sejam úteis para a vida de todos.

O Mapa de Empatia é uma ferramenta do DT, comumente utilizada no *Marketing*. Essa ferramenta auxilia na representação do cliente buscando ter a percepção por meio do universo deste ao colocar-se em seu lugar estabelecendo conhecimentos sobre as necessidades, influências e comportamento [Osterwalder and Pigneur 2010, Cândido et al. 2017].

Objetiva-se nesse trabalho, utilizando a ferramenta do Design Thinking, Mapa de Empatia, capacitar professores e discentes na utilização da empatia como maneira de conhecer melhor todos envolvidos no processo de ensino-aprendizagem, sendo assim capaz de tornar a aprendizagem mais significativa.

2.2. Ensino-aprendizagem

Ensino-aprendizagem é o nome dado a um complexo sistema de interações comportamentais entre indivíduos, onde um assume um papel de professor e o outro de aluno [Kubo and Botomé 2001].

Vale ressaltar a importância da relação professor-aluno no processo ensino-aprendizagem, pois este é o ponto forte nas escolas e deve ser visto como ponto chave para o pleno desenvolvimento dos alunos [da Silva and Navarro 2012]. Ainda, pode-se afirmar que essa relação é o maior objetivo do processo pedagógico. Pois, esse contato estabelece os laços entre a escola e o mercado de trabalho. Além de formar a realidade vivenciada pelo aluno, tornando a escola um local de troca de experiências.

O professor ou educador deve ser considerado como o sujeito mais interativo e ativo no processo de construção e transferência de conhecimento. Muitas vezes se apresentando como a pessoa com mais experiência e conhecimento sistematizado sobre o tema. O professor ainda assume um papel fundamental na sociedade e com a cultura [da Silva and Navarro 2012].

No processo de ensino-aprendizagem, o professor é o agente central, realizando o planejamento e traçando estratégias para a utilização das Tecnologias Digitais da Informação e Comunicação (TDIC) apropriadas para cada cenário educacional [Generoso et al. 2013].

As TDIC são recursos de Tecnologias de Informação e Comunicação (TIC) [Pereira and Silva 2010] que utilizam como principal ferramenta o computador e a Internet para criar uma presença do digital [Crispim 2013]. Porém, as TDIC devem ser entendidas como recursos didáticos que auxiliam o professor no processo ensino-aprendizagem, mas não garantem o êxito por si só a este processo, os recursos podem tornar esse processo mais rico e menos complexo tanto em ambientes de aulas presenciais ou a distância [Soffa and Torres 2012].

Mesmo assim, é necessário que os professores saibam utilizar as TDIC no processo de aprendizagem, exigindo assim novas configurações e adaptações em seus processos metodológicos “tradicionais” usados na maioria das escolas [Mercado 1999]. Vale salientar que essas tecnologias exigem uma nova abordagem e postura por parte dos educadores, para assim saberem integrar na prática do dia-a-dia.

Um dos grandes desafios do século 21 é o de como a educação pode assumir uma perspectiva de desenvolver uma visão de todo o indivíduo e como o preparar para a vida em sociedade, tendo em vista a necessidade de uma abordagem multidisciplinar. Como consequência, existe uma crescente tendência em buscar métodos inovadores que auxiliem no processo pedagógico e na transformação da educação [de SOUZA et al. 2016].

Em suma, é necessário que o professor seja versátil ao nível de saber utilizar estrategicamente as tecnologias disponíveis e saber orquestrar o processo de ensino-aprendizagem com o auxílio de metodologias inovadoras que apoiem as tecnologias e a transferência de conhecimento entre os envolvidos.

2.3. Design Thinking

O *Design Thinking* (DT) pode ser compreendido como um conjunto de ideias que de maneira multidisciplinar e colaborativa auxiliam na criação de soluções tangíveis [Vianna 2012]. Seu surgimento veio das necessidades passadas pelos designers ao longo do tempo.

O termo *Design Thinking* ficou conhecido por Tim Brown, CEO da IDEO uma empresa de design, Brown utiliza o termo para descrever princípios do design que podem ser aplicados em diversos tipos de problemas [Brown 2018]. Ainda, entende-se que o Design tradicional evoluiu naturalmente para o DT que é utilizado não somente pelos designers.

O DT é baseado em três pilares: viabilidade (negócio), desejabilidade (pessoas) e praticabilidade (técnica), onde a interseção desses formará a solução proposta [SEBRAE 2017], conforme visualizado na Figura 2.1.

O DT busca transformar as necessidades humanas em possibilidades de negócio por meio da tecnologia existente [Brown 2018]. Ainda, a metodologia utiliza de processos inovadores para desenvolver produtos e serviços para solucionar problemas complexos [Lockwood 2010].

O DT possui três fases: Descoberta, Ideação e Experimentação [Vianna 2012]. Que por sua vez se subdividem em 5 etapas: Empatia, Definição, Ideação, Prototipação e

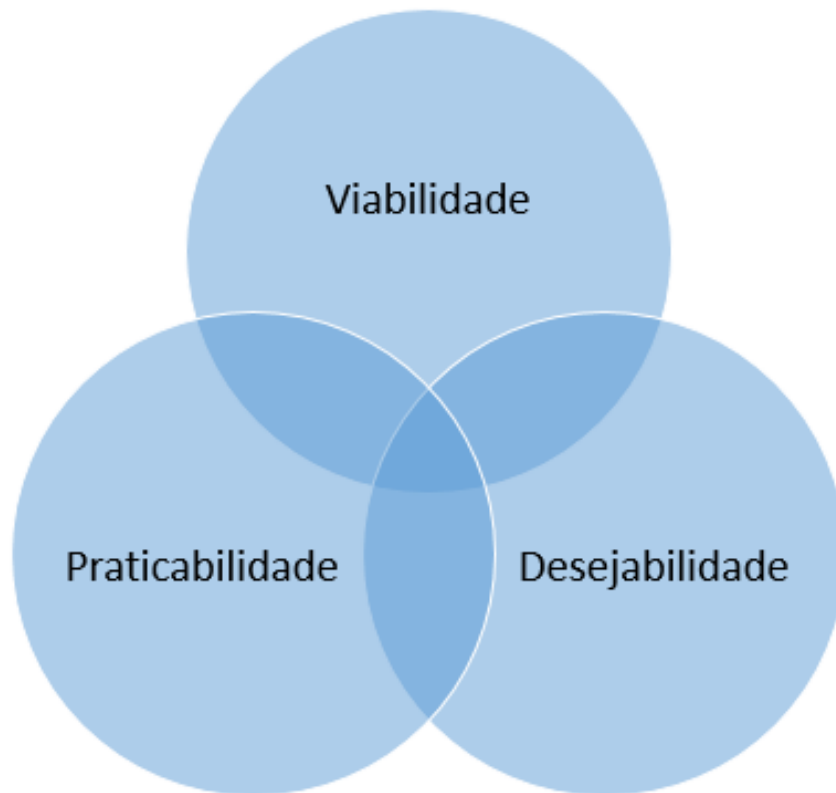


Figura 2.1. Pilares do Design Thinking

Validação [de Souza et al. 2017]. Conforme Figura 2.2.

Na fase de Descoberta, compreende-se os passos para compreender os desafios e problemas existentes. Além da empatia necessária com as pessoas que serão os usuários do produto ou serviço. A empatia trata de um profundo entendimento das necessidades e motivações das pessoas. Em uma sala de aula as pessoas envolvidas podem ser os estudantes, professores, pais, funcionários e gestores escolares que compõem seu cotidiano.

Já na fase de Ideação, compreende-se pelos passos da geração de ideias e oportunidades que podem existir dentre os envolvidos e os problemas relatados na fase anterior.

Por sua vez na fase de experimentação, são desenvolvidos protótipos e propostas de soluções de maneira colaborativa, rápida e de baixo custo.

2.3.1. Ferramentas do Design Thinking

A metodologia DT possui ferramentas compostas por artefatos e técnicas que podem ser utilizadas em cada uma das fases e etapas do processo de criação de uma nova solução [Brown 2018, Vianna 2012, Grotti 2016]. A seguir algumas são apresentadas algumas dessas ferramentas:



Figura 2.2. Fases do Design Thinking

- **Desafio:** Aplicar perguntas do tipo “Como podemos...?” para auxiliar na busca por soluções;
- **Pesquisa Desk:** Utilizar diferentes fontes, fora do grupo diretamente envolvido, para levantar informações contextuais sobre o problema;
- **Entrevistas:** Obter informações por meio de conversas e perguntas para compreender um contexto;
- **Registros de Histórias:** Aprender sobre o conhecimento adquirido durante as entrevistas e compartilhar as principais informações em formato de histórias;
- **Caderno de sensibilização:** Obter informações sobre um indivíduo com o mínimo de interferência possível, já que o próprio irá fazer o relato de suas atividades de acordo com seu contexto, rotina, opiniões e ideias;
- **Mapa de Processo:** Documentar a sequência de atividades ou eventos e suas relações;
- **Reenquadramento:** Observar os problemas não resolvidos sob diferentes perspectivas em ciclos de captura;
- **Pesquisa Exploratória:** Levantar informações que servirão para outras ferramentas, como também para a equipe conhecer o contexto com o qual deve trabalhar por meio de pesquisas de campo preliminares;
- **Persona:** Pessoalizar a representação das motivações, desejos, expectativas e necessidades, reunindo características significativas de um grupo de possíveis usuários;

- **Mapa de Empatia:** Sintetizar de maneira colaborativa as percepções observados expondo os comportamentos identificadas em um público entrevistado.

2.3.1.1. Mapa de Empatia

O Mapa de Empatia é uma ferramenta do DT, considerada como a mais importante, pois a sua utilização é voltada nos estágios iniciais do processo. Onde são identificadas as características do público alvo, também chamado de persona, sendo assim, a proposta deste trabalho é utilizar essa ferramenta para que possa ser descoberto o perfil, ou persona, dos envolvidos no processo de ensino-aprendizagem, a saber, professores e discentes.

Na Figura 2.3, é apresentado o mapa de empatia, composto por seis áreas onde os envolvidos no processo de ensino-aprendizagem irão descrever seus sentimentos e suas percepções diante dos seguintes questionamentos:

O que escuta?

O que geralmente escuta dos seus amigos, parentes e colegas de trabalho? O que o chefe ou superior costuma falar? O que os influenciadores costumam dizer? Quais produtos ou serviços de comunicação consome?

O que pensa e sente?

Quais são as ideias que um produto ou serviço despertaria nas mentes dos consumidores? Como se sente em relação ao mundo? Quais as suas preocupações? Quais são os seus sonhos?

O que vê?

Como é o mundo e o ambientes em que vive? O que costuma ver no seu cotidiano? Como são seus amigos, parentes e pessoas próximas? O que o mercado oferece de produto ou serviço sobre o problema? Quais estímulos visuais são percebidos?

O que fala e faz?

Quais as suas atitudes em público? Como é a sua aparência? Como é o seu comportamento em seu cotidiano? Como se relaciona com os outros? Qual discurso que faz constantemente? O que pratica? Como age? Quais são as suas ocupações?

Quais são as dores?

Quais são seus principais medos? Quais as suas frustrações? Que obstáculos precisa ultrapassar para conseguir o que deseja?

Quais são os ganhos?

Quais são os seus principais desejos? Quase são as suas necessidades cotidianas? Como é medido o sucesso? Quase são os obstáculos? Onde quer chegar? O que acabaria com seus problemas?

2.3.2. Design Thinking para educação

Devido as características do DT de permitir a liberdade de errar e aprender por meio da geração de novas ideias, *feedback* de outras pessoas, depois de repensa nessas ideias pode ser incorporado nos planos de aula de educadores no processo de ensino-aprendizagem [Educadigital 2018].

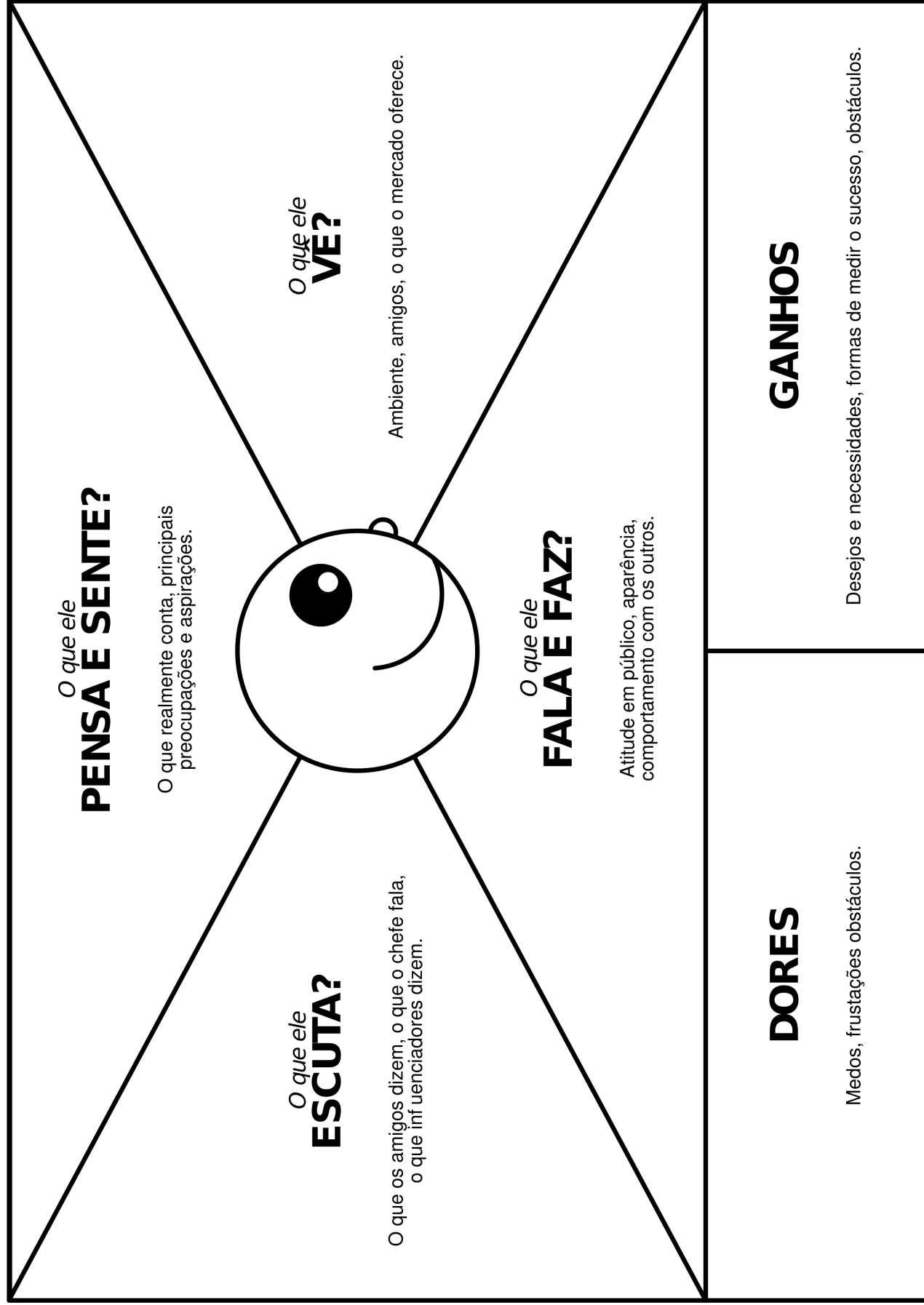


Figura 2.3. Mapa de Empatia

O processo do Design Thinking voltado para educadores trata-se também de abordagem estruturada para gerar e aprimorar ideias. Porém, são adaptadas para as necessidades dos professores. Esse processo é dividido em cinco fases que vão desde a geração de ideias, desenvolvimento, identificar desafios e construir soluções.

A seguir as cinco fases do Design Thinking para Educadores [Educadigital 2018]:

1. Descoberta, pode-se utilizar de um workshop de poucos dias com professores e alunos para com o auxílio da empatia e de ferramentas do DT possam ter uma percepção sobre os sonhos dos alunos, suas necessidades e sobre quais ferramentas os professores podem utilizar coletivamente para despertar as habilidades necessárias para os alunos desenvolverem para que tenham sucesso no futuro;
2. Interpretação, o grupo pode realizar pesquisas entre os envolvidos, criando perguntas genéricas, podendo utilizar a ferramenta "Como podemos...?";
3. Ideação, nessa fase pode-se compreender como o brainstorm com possíveis soluções para os problemas verificados nas fases anteriores;
4. Experimentação, realização da prototipação de algumas ideias levantadas no brainstorm e discutir colaborativamente entre alunos e professores;
5. Evolução, aqui serão realizados testes para possíveis melhorias ou formas de implementar as soluções propostas.

Pode-se utilizar os recursos do DT de maneira total por meio do DT para educadores ou também pode-se utilizar suas ferramentas individuais, como o mapa de empatia, tanto em ambiente empreendedor quanto em ambiente educacional.

2.4. Aplicação do Mapa de Empatia na Sala de Aula

A sua aplicação será dividindo em duas partes: A primeira será individual, onde cada discente fará o seu próprio mapa. Em seguida, na segunda parte, será composto um mapa geral para toda a turma, é interessante dividir a turma em grupos e solicitar que eles debatam cada questionamento, onde podem ter áreas em comum como problemas e até mesmo percepções.

É importante que o professor fale sobre a sua jornada, fale sobre a suas experiências, compartilhe com os seus discentes algo relacionado a sua vida, para que os discentes possam sentir mais afinidade com o docente, empatia é buscar pela necessidade do outro, e é uma via de mão dupla, o docente precisa mostrar para os discentes que está disposto a abrir-se contando sua história.

O professor ou professores também podem ao final apresentar o seu mapa afim de evidenciar quais as percepções do educador perante a turma, vejamos abaixo como conduzir o processo individual e em grupo.

2.4.1. Aplicação Individual

Para realizar a aplicação individual você deverá disponibilizar o mapa de empatia, conforme a Figura 2.4 e explicar ao discente que ele pode ou não se identificar, é muito importante enfatizar isso, pois muitos podem ficar com vergonha ou até mesmo medo de responder as perguntas do mapa.

Para realizar a aplicação de forma presencial o Mapa de Empatia poderá ser impresso em papel tamanho A4 com o desenho da Figura 2.3 e solicitado aos discentes que preencham, para a utilização de forma remota poderá ser utilizada a ferramenta Google Jamboard¹, é possível observar na Figura 2.5 uma possível representação do Mapa de Empatia utilizando a ferramenta Google Jamboard.

A disponibilização dos Mapas poderá ser no formato físico impresso, na sala de aula presencial, ou poderá ser entregue via Google Jamboard utilizando o Google Sala de Aula² para facilitar o processo, bastando para tal, criar uma atividade, anexar o modelo do Mapa de Empatia e escolher a opção “Criar uma cópia para cada discente”, ao fazer isso cada discente receberá uma cópia, aplicando de forma remota é mais difícil de deixar o mapa anônimo, pois o discente terá que lhe enviar através do Google Sala de Aula.

Uma outra opção para essa aplicação, caso a instituição não tenha a adesão ao Google Sala de Aula, é você compartilhar o link do Google Jamboard e adicionar “/copy” no final da URL, como no exemplo a seguir:

- **URL original:**
<https://jamboard.google.com/d/1M1sl...Sgg2vdECnCkA4owos/viewer>
- **URL para cópia:**
<https://jamboard.google.com/d/1M1sl...Sgg2vdECnCkA4owos/copy>

Ao adicionar “/copy” é gerada uma cópia do arquivo Jamboard para o discente, Figura 2.6, neste caso o professor pode compartilhar o link através de qualquer plataforma ou até mesmo por e-mail para seus discentes.

Para realizar a entrega dos mapas o docente pode solicitar que o aluno entregue o mapa impresso em cima de uma mesa ou até mesmo em uma pilha de papel embaralhada, se o discente desejar o anonimato, ou simplesmente recolher caso o discente tenha preenchido com o seu nome. Por parte do discente de forma remota, é possível compartilhar o link com o docente para que ele possa receber o mapa, nesse caso não tem como ser anônimo.

Como o objetivo é traçar um perfil da turma o preenchimento pode ser realizado de forma anônima, no entanto é interessante que o docente encoraje os discentes a preencher com o nome, pois assim é possível identificar algum problema individual ajudando na tomada de decisões específicas.

Para auxiliar os discentes, pois por muitas vezes podem ficar sem saber o que escrever, o docente pode escolher um ambiente para focar e fazer as seguintes perguntas:

¹<https://edu.google.com/intl/pt-BR/products/jamboard>

²<https://edu.google.com/intl/pt-BR/products/classroom/>

Nome: Andre Idade: 27

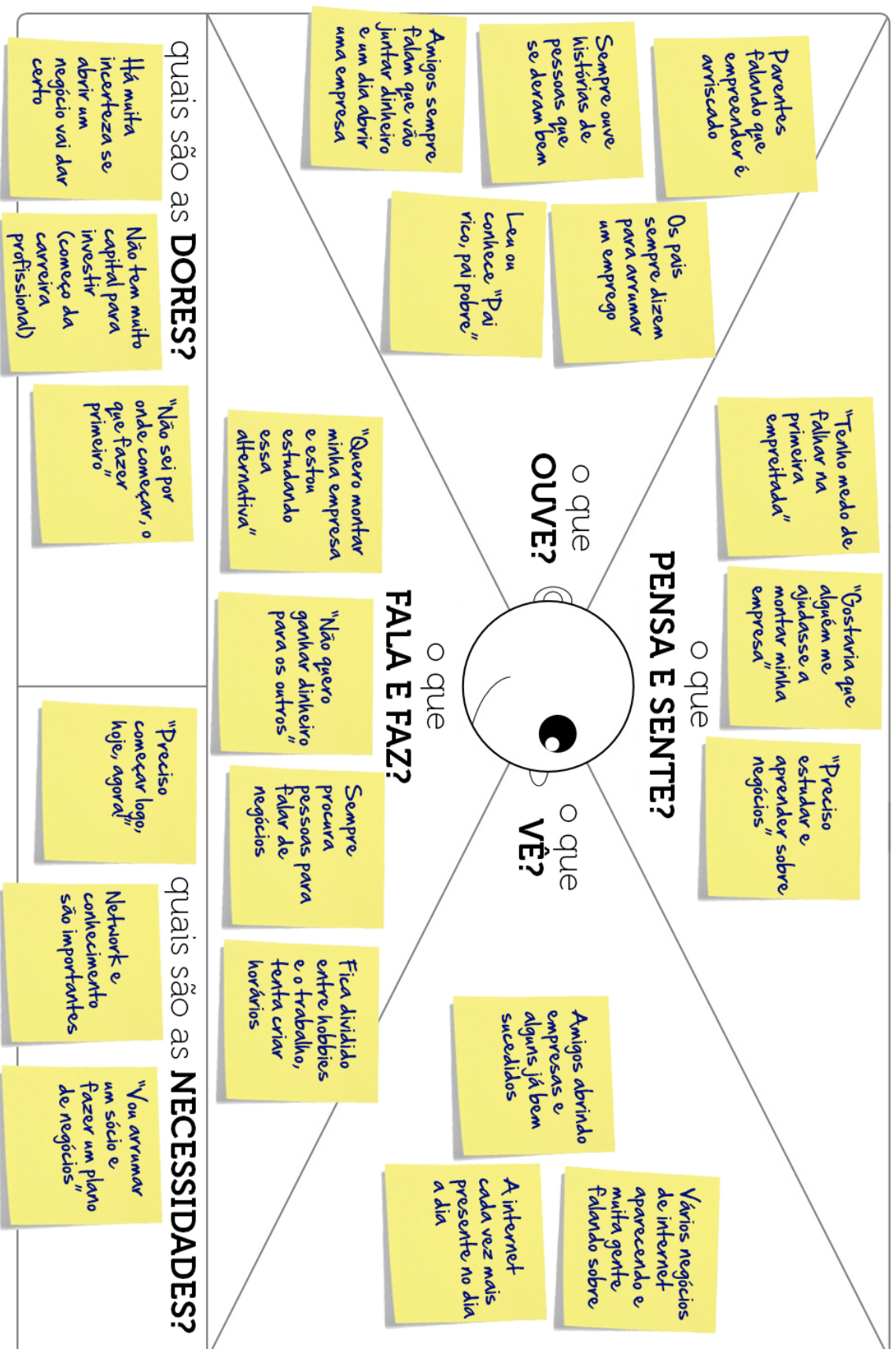


Figura 2.4. Mapa de Empatia individual preenchido

Fonte: [Blog Mastertech 2021]

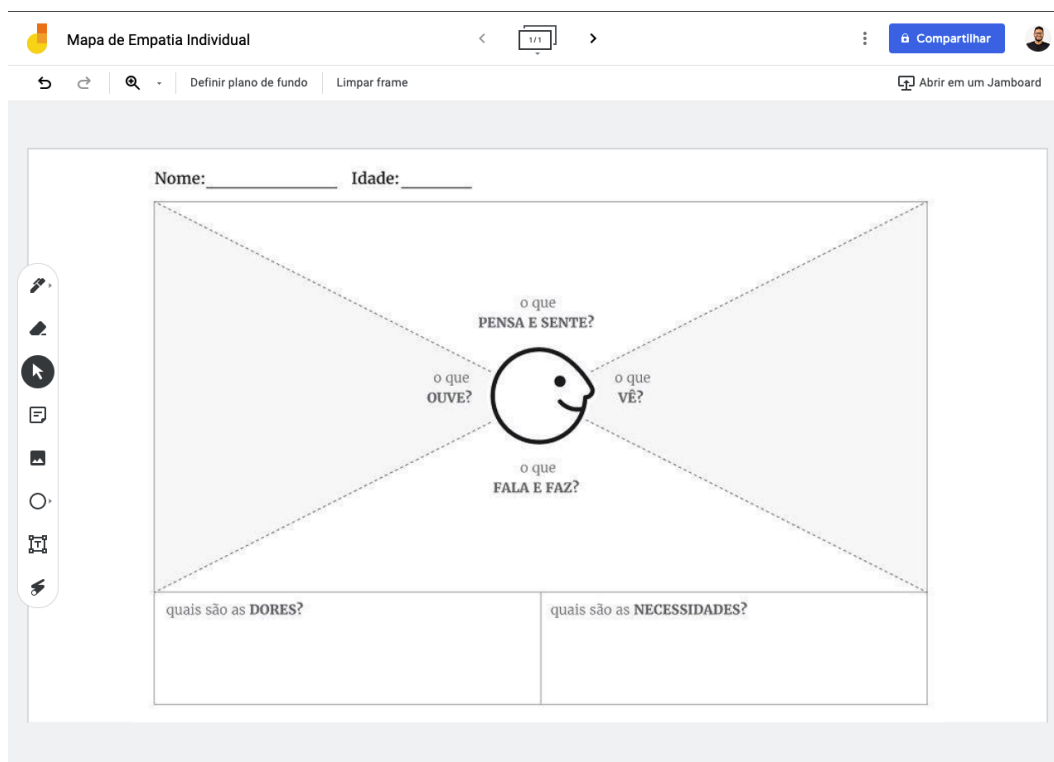


Figura 2.5. Mapa de Empatia individual Google Jamboard

Google Drive

Copiar documento

Você deseja fazer uma cópia de **Mapa de Empatia Individual**?

Fazer uma cópia



Figura 2.6. Criando cópia do Mapa de Empatia individual do Google Jamboard

- Qual a **melhor** percepção a respeito do ambiente acadêmico?
- Qual a **pior** percepção a respeito do ambiente acadêmico?
- Descreva **3 dores** e **3 necessidades** do ambiente acadêmico.

As questões apresentadas acima devem ser respondidas no Mapa de Empatia, ou

Atividade:

- Qual a **melhor** percepção a respeito do ambiente educacional (responda no mapa de empatia);
- Qual a **pior** percepção a respeito do ambiente educacional (responda no mapa de empatia);
- Descreva **3 dores** e **3 necessidades** do ambiente educacional no mapa de empatia.

Gênero: feminino Idade: 15

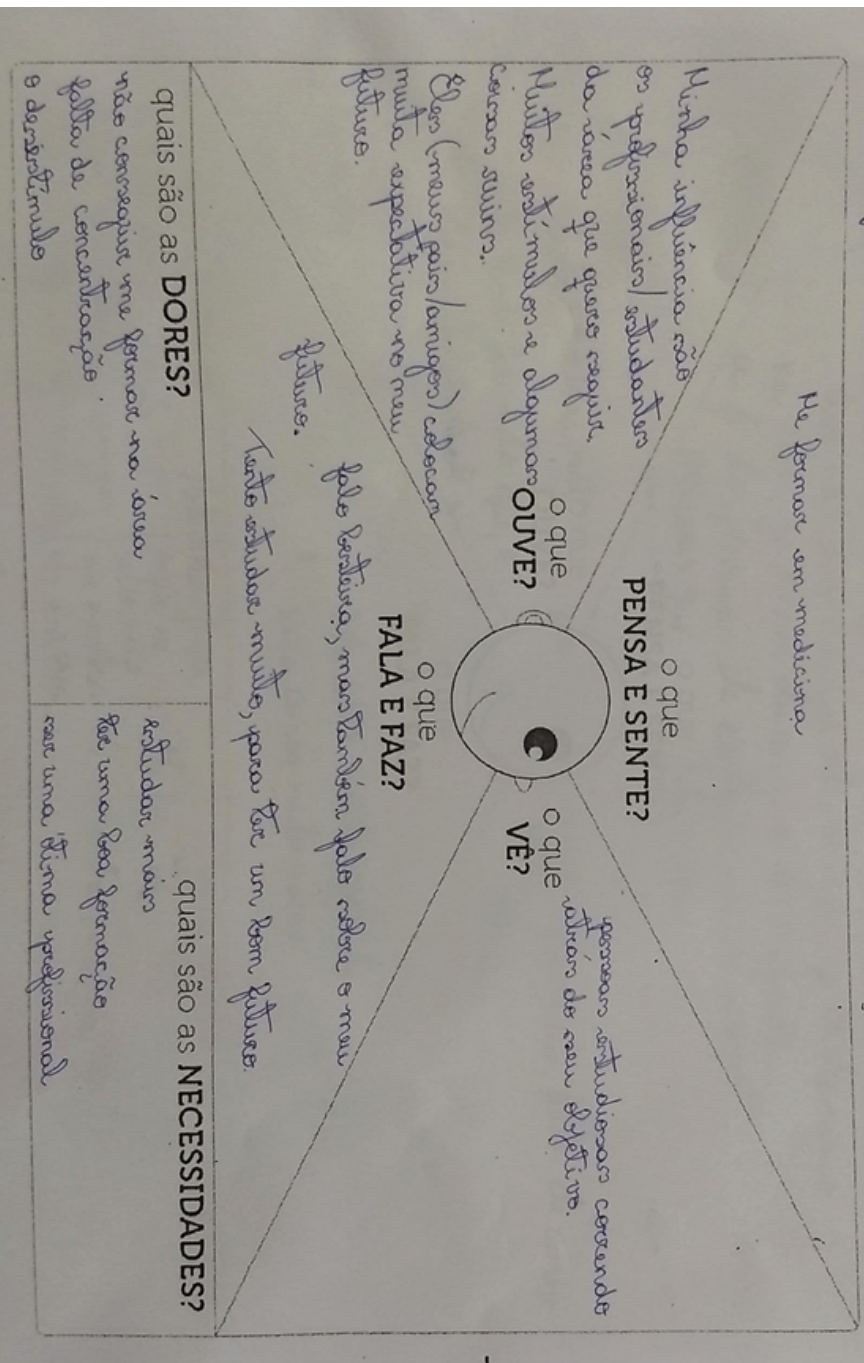


Figura 2.7. Mapa de Empatia preenchido por uma discente

seja, “o que pensa e sente?”, o discente elenca os melhores e os piores pensamentos e sentimentos relacionados ao ambiente acadêmico, ”o que vê?”, o discente elenca as suas melhores e piores visões relacionados ao ambiente acadêmico, e assim para as demais perguntas.

Quanto aos questionamentos das “dores” e “necessidades”, ao informar que são três itens de cada, o docente faz com que o discente pense em algumas situações, pois muitas vezes os mesmos tem muita dificuldade e acabam deixando apenas um item ou nem respondendo.

Caso o docente não consiga que os discentes adicionem seus nomes, ele pode adicionar os campos de gênero e idade, irão auxiliar à traçar o perfil da turma, como podemos observar na Figura 2.7 e na Figura 2.9 podemos observar um mapa preenchido através do Google Jamboard, lá o aluno pode utilizar a ferramenta “Nota autoadesiva”.



Figura 2.8. Mapa de Empatia preenchido de forma coletiva

Após a aplicação dos mapas individuais o docente pode digitalizar, se assim preferir, se for aplicado impresso e compilar todos os dados em um novo Mapa de Empatia, este que irá representar toda a turma, combinando as percepções de todos os discentes.

2.4.2. Aplicação em Grupo

Para a aplicação do Mapa de Empatia de forma presencial em grupo, utiliza-se um papel tamanho A0 com o desenho do mapa e solicita-se que os discentes preencham o mapa com notas autoadesivas, estilo *post-its*, para a utilização remota será utilizada a ferramenta Jamboard do Google.

Para realizar a aplicação em grupo é interessante que o docente divida a turma em pequenos grupos, dependendo do tamanho da turma, grupos com até 6 pessoas são ideais, mas é interessante o docente adaptar à sua realidade, dentro do grupo menor os discentes vão socializar as percepções, preencher uma nota autoadesiva e colar no Mapa

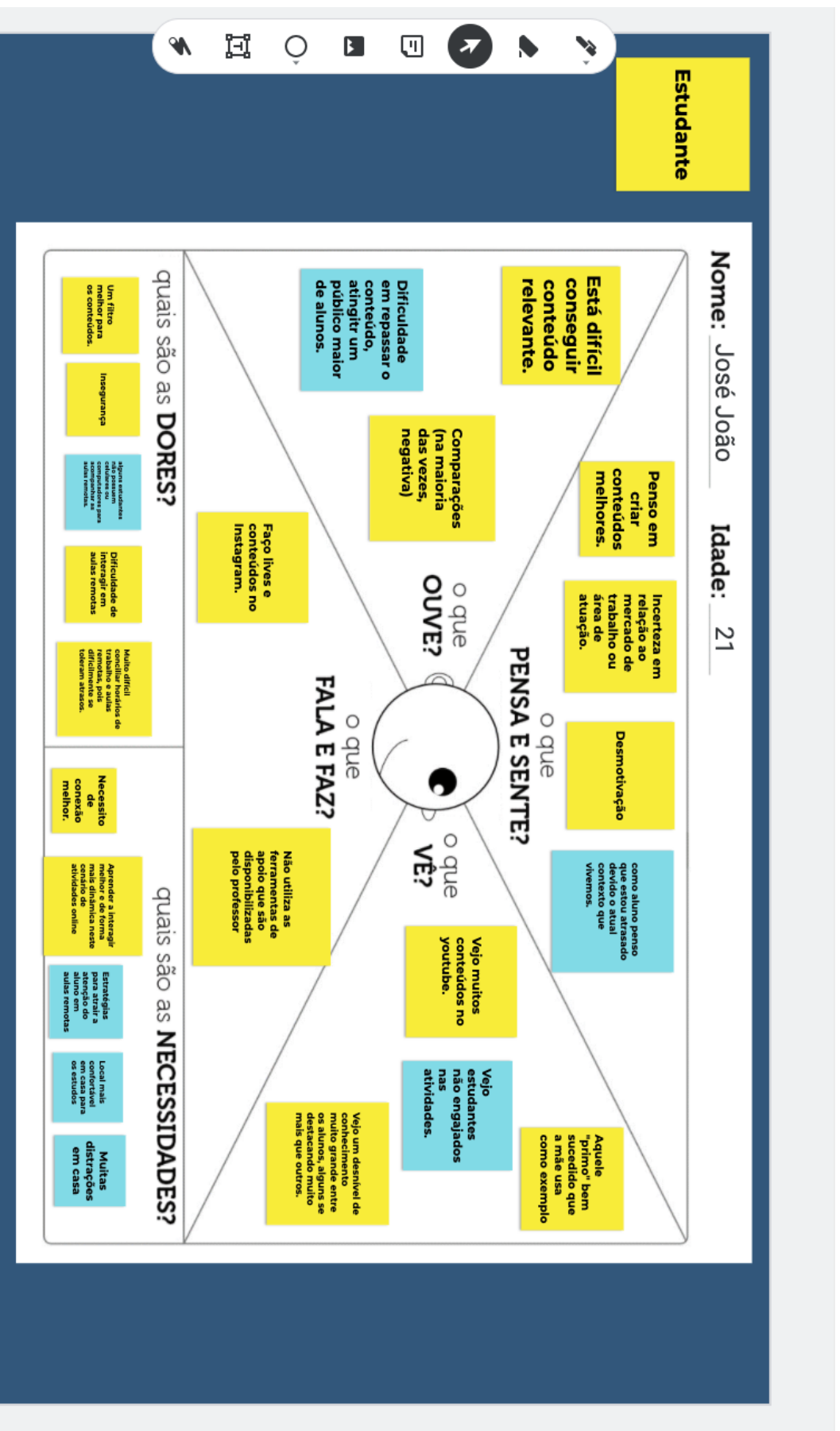


Figura 2.9. Mapa de Empatia preenchido por uma discente no Google Jamboard

de Empatia no papel A0, os demais grupos poderão ver toda a evolução e fazer o mesmo procedimento, é possível observar nas Figuras 2.8 e 2.10 um Mapa de Empatia preenchido de forma coletiva em uma aula presencial.



Figura 2.10. Mapa de Empatia preenchido de forma coletiva

Para aplicar o Mapa de Empatia de forma coletiva no ensino remoto é possível utilizando o Google Jamboard, neste caso é possível manter o anonimato dos discentes pois trata-se de um documento único para todos. Para realizar esta aplicação o docente executa o mesmo procedimento de criação do Google Jamboard, no entanto como são

muitas pessoas que irão responder é possível fazer uma adaptação e criar um “Quadro Jam”³ para cada pergunta, conforme podemos ver nas Figuras 2.11 e 2.12.

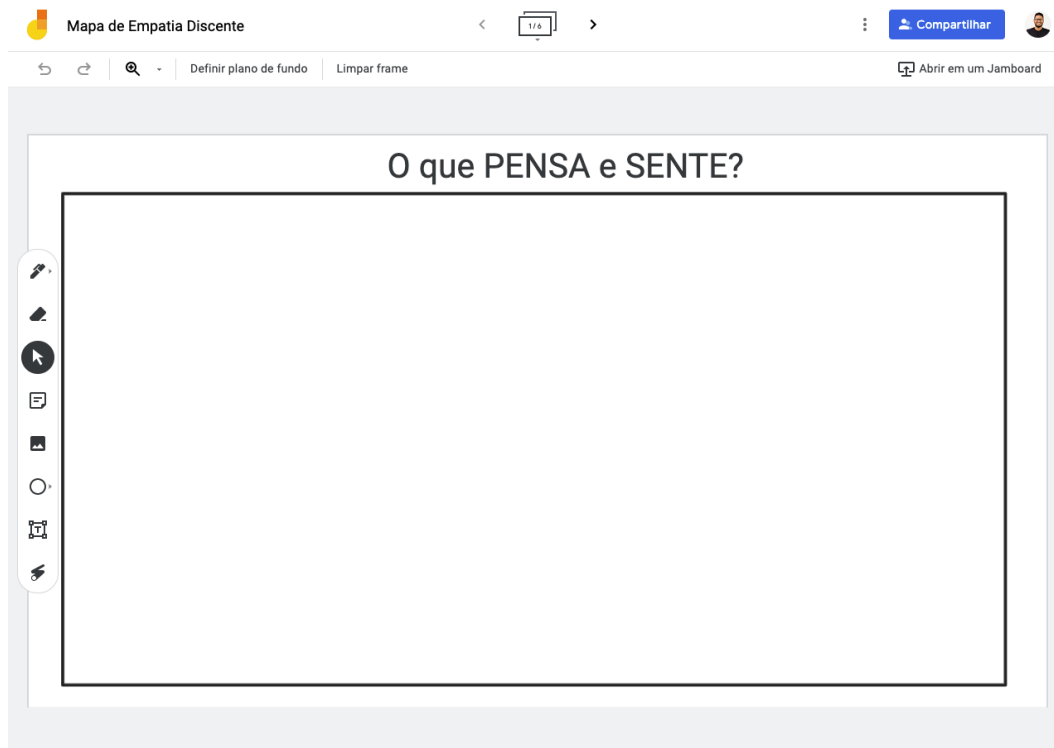


Figura 2.11. Exemplo de Mapa de Empatia para preenchimento remoto de forma coletiva

O preenchimento ocorre da mesma forma que o remoto individual, utilizando notas autoadesivas, e pode ser compartilhado através de um link para todos os alunos, para compartilhar basta clicar no botão “Compartilhar” e em seguida escolher a opção “Qualquer pessoa com o link” e selecionar que poderão “Editar”, conforme mostrado na Figura 2.13.

Após a aplicação é de suma importância que ocorra uma discussão acerca do que foi respondido no Mapa de Empatia coletivo, para que os alunos reflitam sobre as respostas e para que o docente possa já fazer alguma orientação inicial quanto aos tópicos abordados, para exemplificar traremos uma situação seguinte: se os discentes colocaram que uma das dores não conseguir concluir alguma atividade, TCC ou relatório, neste caso o docente pode aconselhar um caminho para a conclusão dessas atividades, motivando-os a concluir as atividades.

2.4.3. O que fazer após a aplicação

Após realizar as aplicações individuais e coletiva dos Mapas de Empatia o docente poderá tabular os dados dos individuais e gerar um novo coletivo com base nas respostas dos discentes, assim obtendo o perfil da sua turma, é possível também comparar com o coletivo que foi preenchido.

³ Assim são chamados os quadros criados no Google Jamboard, é possível criar vários, como se fossem slides.

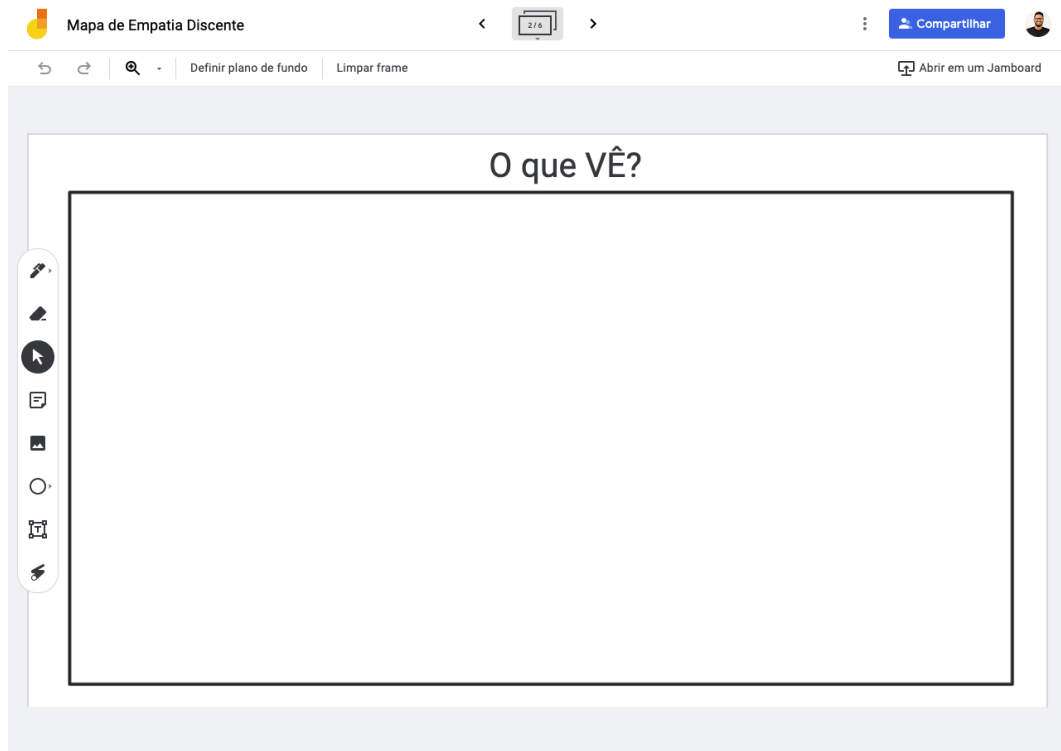


Figura 2.12. Exemplo de Mapa de Empatia para preenchimento remoto de forma coletiva

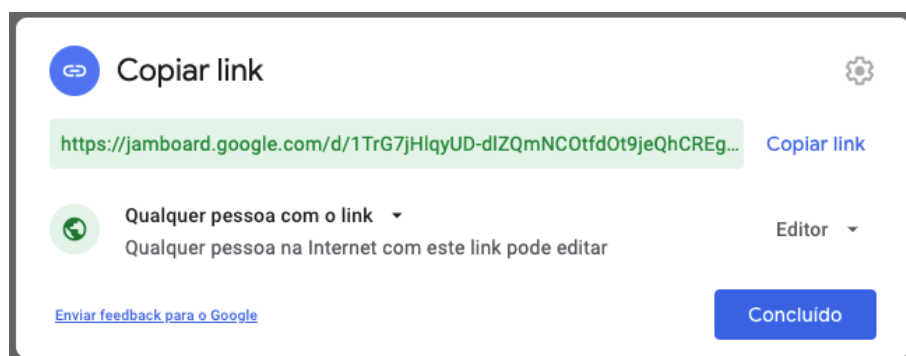


Figura 2.13. Compartilhando Jamboard com Mapa de Empatia para preenchimento remoto de forma coletiva

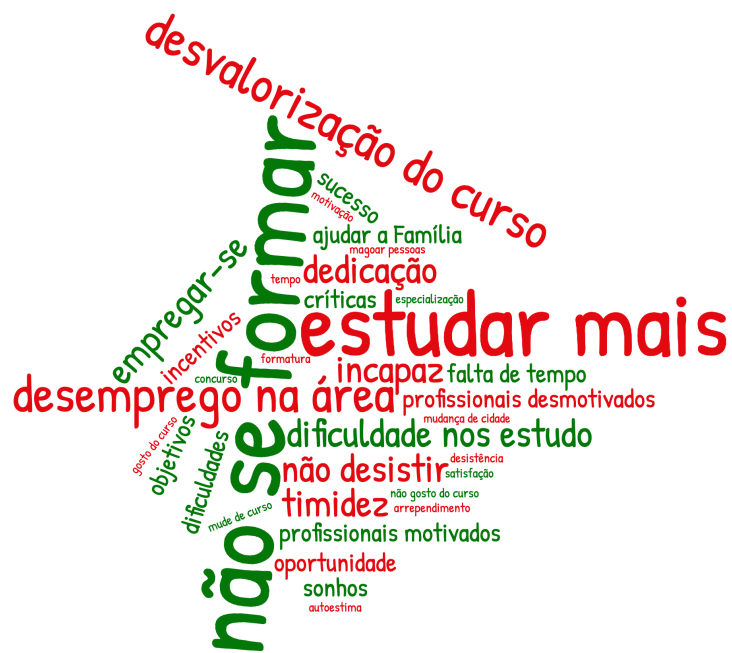


Figura 2.14. Nuvem de palavras gerada a partir de diversos mapas individuais - Turma 1

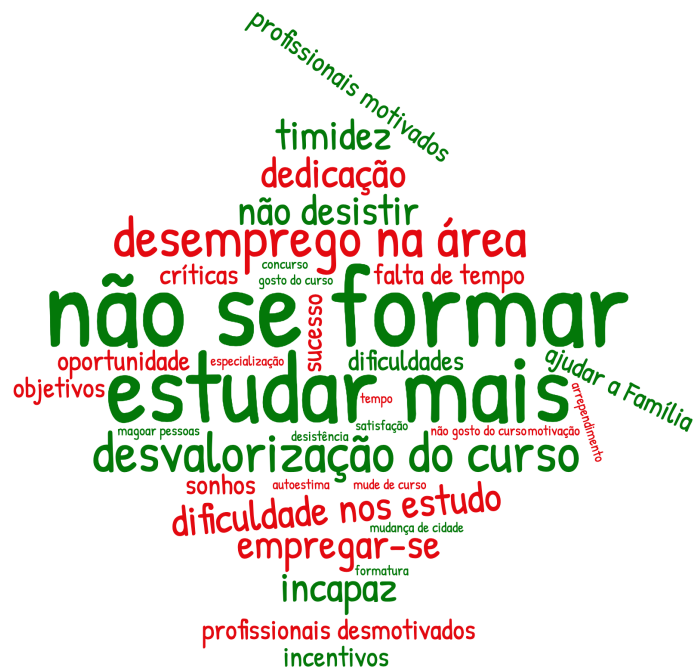


Figura 2.15. Nuvem de palavras gerada a partir de diversos mapas individuais - Turma 2

Em uma próxima etapa o docente pode compartilhar com a turma os resultados da combinação dos mapas, uma ferramenta interessante para tal é gerar uma nuvem de palavras de cada área do Mapa de Empatia ou juntar todas as áreas e gerar uma única nuvem, como podemos observar nas Figuras 2.14 e 2.15.

2.5. Considerações Finais

A partir da aplicação do mapa de empatia tem-se um persona de cada turma, com características únicas e suas dores e ganhos, ou seja, pode-se planejar aulas mais direcionadas à atingir objetivos específicos e mais significativos para cada turma, além de adaptar por meio da empatia e conseqüentemente do processo de Design Thinking.

Com os mapas individuais de cada discente, que podem ser anônimos, o professor pode tabular os dados e construir um novo mapa ou criar nuvens de palavras e socializar com a turma, afim de que eles se conheçam melhor, tornando assim um ambiente mais agradável e propício para o aprendizado.

Essas técnicas podem ser utilizadas em conjunto com a metodologia de ensino do próprio professor em sala de aula ou em modo remoto para podem auxiliar no processo de ensino-aprendizagem.

Além disso, o professor pode adaptar as perguntas do mapa de empatia para um contexto específico caso precise de respostas específicas sobre um problema já definido e conhecido por todos os envolvidos no processo de ensino-aprendizagem.

Referências

- Blog Mastertech (2021). Mapa de Empatia Andre.
- Brolezzi, A. C. (2014). Empatia em vygotsky. *Dialogia, São Paulo*, (20):153–166.
- Brown, T. (2018). *Design thinking: uma metodologia poderosa para decretar o fim das velhas ideias*. Alta Books Editora.
- Brown, T. (2020). *Design Thinking: uma metodologia poderosa para decretar o fim das velhas ideias*. Alta Books.
- Cândido, A. C., da Silva Bertotti, P. S., and Bedin, J. (2017). O potencial das ferramentas atuais de gestão & negócios aplicados às unidades de informação. *Revista Brasileira de Biblioteconomia e Documentação*, 13:1165–1182.
- Crispim, J. J. (2013). TIC vs NTIC.
- Cruz, J. M. d. O. (2008). Processo de ensino-aprendizagem na sociedade da informação. *Educação & Sociedade*, 29(105):1023–1042.
- da Fonseca, V. (2019). *Desenvolvimento cognitivo e processo de ensino aprendizagem: Abordagem psicopedagógica à luz de Vygotsky*. Editora Vozes Limitada.
- da Silva, O. G. and Navarro, E. C. (2012). A relação professor-aluno no processo ensino-aprendizagem. *Revista Eletrônica Interdisciplinar*, 2(8):95–100.

- de Souza, A. F. B., Ferreira, B. M., and Conte, T. (2017). Aplicando design thinking em engenharia de software: Um mapeamento sistemático.
- de SOUZA, C. D. F., Antonelli, B. A., and de OLIVEIRA, D. J. (2016). Metodologias ativas de ensino aprendizagem na formação de profissionais da saúde. *Revista da Universidade Vale do Rio Verde*, 14(2):659–677.
- Educadigital (2018). Design Thinking para Educadores.
- Generoso, A. A. P., Neto, J. C., Reinehr, S., and Malucelli, A. (2013). Abordagem qualitativa do uso das TIC na educação básica. In *Anais do Workshop de Informática na Escola*, volume 1, page 230.
- Grotti, M. e. a. (2016). A busca da eficiência por meio da aplicação do design thinking na integração de processos voltados á melhoria do produto na indústria. *XXXVI ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO*.
- Kubo, O. M. and Botomé, S. P. (2001). Ensino-aprendizagem: uma interação entre dois processos comportamentais. *Interação em Psicologia*, 5(1).
- Lockwood, T. (2010). *Design thinking: Integrating innovation, customer experience, and brand value*. Skyhorse Publishing, Inc.
- Mercado, L. (1999). *Formação Continuada de Professores e Novas Tecnologias*. EDUFAL.
- Osterwalder, A. and Pigneur, Y. (2010). *Business model generation: a handbook for visionaries, game changers, and challengers*. John Wiley & Sons.
- Pereira, D. M. and Silva, G. S. (2010). As Tecnologias de Informação e Comunicação (TICs) como aliadas para o desenvolvimento. *Caderno de Ciência Sociais Aplicadas*, (10):151–174.
- SEBRAE (2017). O design thinking como ferramenta estratégica para pequenos negócios. BigData Business Hekima.
- Silva, L. M. d. M. A. (2015). *A literacia emocional em sala de aula: A importância da empatia na relação pedagógica professor-aluno para o sucesso do ensino-aprendizagem na disciplina de Economia A*. PhD thesis.
- Soffa, M. and Torres, P. (2012). O processo ensino-aprendizagem mediado pelas tecnologias da informação e comunicação na formação de professores on-line. In *IX Congresso Nacional de Educação*.
- Vianna, M. (2012). *Design thinking: inovação em negócios*. Design Thinking.