UNIP
UNIVERSIDADE PAULISTA

6º Semestre Ciência da Computação (CC)

Atividade Prática Supervisionada

Tema: "DESENVOLVIMENTO DE UM SISTEMA DE IDENTIFICAÇÃO E AUTENTICAÇÃO BIOMÉTRICA."

Alunos participantes:

Filipi Yukio Iwakami Itoyama, RA – N4453J-1

William Rossi do Carmo Ruiz, RA - N473GF-8

Luiz Gabriel Zeferino Duarte, RA – N454CD-8

Lucas de Oliveira Brandolezi, RA – D9380G-9

Índice

1. Objetivo do Trabalho

Têm-se como objetivo deste trabalho, o desenvolvimento de uma ferramenta para efetuar uma identificação e autenticação biométrica. Para isso, utilizamos a linguagem computacional do GNU Octave e algumas de suas bibliotecas famosas para fazer as alterações, identificações e autenticações das biometrias.

2. Introdução

A biometria vem se destacando recentemente pela sua grande participação e importância nos sistemas de segurança, sendo utilizado em diversos sistemas e programas para proteger, identificar e autenticar os usuários. O método antigo, porém ainda usado bastante, é o famoso “Login e Senha”, ou mais especificamente, o método da digitação, onde o usuário deve digitar o seu código de usuário (pode ser o e-mail, nome do usuário, até mesmo algum número significando um ID) e a sua senha para se identificar e autenticar.

Com o desenvolvimento dessa nova técnica de segurança utilizando a biometria, os sistemas ganharam um nível elevado de segurança para os sistemas, e não somente a segurança aumentou, mas nível de conforto também aumentou, sendo assim, um método de segurança mais seguro e confortável em relação ao método da digitação.

3. Fundamentos das principais técnicas biométricas

Existem diversas técnicas biométricas para a construção de um sistema seguro, como o reconhecimento da impressão digital, reconhecimento facial, reconhecimento da íris, reconhecimento da voz, e até mesmo existem sistemas que fazem o reconhecimento da retina. Neste trabalho, iremos focar principalmente o sistema biométrico no qual faz a leitura e reconhecimento de uma impressão digital.

O sistema de reconhecimento da impressão digital foi bastante usado no começo de seus usos, no setor da segurança pública, principalmente aqui no Brasil.

Hoje, o sistema de reconhecimento de impressões digitais é usado em diversas outras situações e aplicações, como na autenticação móvel, onde o usuário utiliza da sua impressão digital para confirmar sua identidade e ter acesso aos seus dispositivos móveis; sistemas de identificação civil para que o governo identifique e autentique os civis para fins de votação, pagamentos de benefícios e até mesmo segurança; controle de acesso físico para que empresas tenha controle sobre o acesso de entrada e saída dos funcionários; e até mesmo para a prevenção de fraudes em alguns sistemas, evitando identidades falsas ou duplicadas.

O algoritmo para o reconhecimento de impressões digitais consiste em coletar as impressões digitais de um indivíduo e gerar um “modelo” a partir das características dessa imagem. Após isso, o algoritmo de correspondência compara as características do modelo original gerado no começo, com o modelo da amostra recém coletada, ou recém inserida. Nos algoritmos mais complexos e mais estruturados, essas imagens recém coletadas ou recém inseridas, são usadas para gerarem novos modelos caso o modelo recém coletado corresponda e seja autenticado.

4. Plano de desenvolvimento da aplicação

O software escolhido para esta aplicação foi o GNU Octave, já que todos os membros do grupo aprenderam essa linguagem no sexto semestre do curso de Ciência da Computação, além de ser gratuita e de fácil acesso. O Octave utiliza uma linguagem de alto nível, usada principalmente para cálculos numéricos.

Para ajudar importamos o pacote “image”, que fornece funções para o processamento de imagens, extração de recursos, estatísticas de imagem, transformações espaciais e geométricas, operações morfológicas, filtragem linear e outros.

Para carregar as imagens no octave usaremos a função “imread”, também será usada a função “size()”, usado para visualizar o número de elementos atualmente presentes na imagem. Outra função que será utilizada é a “imsmooth()”, que suaviza a imagem fornecida usando vários algoritmos diferentes, outro função será o “find”, que retorna um vetor de índices de elementos diferentes de zero de uma matriz e pôr

fim a função “subplot”, que dividi a área do gráfico em uma série de janelas indexadas por um inteiro.

5. Plano de Desenvolvimento, Projeto e Relatório com as linhas de Código da Aplicação

Inicialmente começamos importando o pacote “image” para utilizar algumas funções importantes, depois inserimos as imagem que serão comparadas (I e i2) através função imread(). Na sequência definimos uma variável getAnomalia como false (essa função será usada mais a frente) depois pegamos as dimensões da imagem com a função size() e aguardamos na matriz [l c].

Por fim submetemos nossa imagem I e I2 a função preprocessamento() que será analisada na sequência.

```
pkg load image
I = imread('D:\_NaoApagarNunca\Desktop\unip_4semestre\processamento de imagem\t1.jpg');

I2 = imread('D:\_NaoApagarNunca\Desktop\unip_4semestre\processamento de imagem\t2.jpg');

getAnomalia = false;

[l c] = size(I);

I = preprocessamento(I);
I2 = preprocessamento(I2);
```

Dentro da função preprocessamento começando definido o valor de retorno como imResult, na sequencia guardamos as dimensões da imagem im dentro de uma matriz [l c]. em sequência submetemos nossa imagem im a função imsmooth(“Average”) que aplicará uma suavização por média. Por fim percorremos transformando todos os nossos pixels a baixo de 128 em 0 e os outros em 250 a fim de binarizar nossa imagem e retornamos o resultado.

```
function imResult = preprocessamento(im)
  [l c] = size(im);
  #suavisar imagem pelo metodo da media
  im = imsmooth(im,"Average");
  #tranformar em imagem binária
    for x = 1:l
      for y = 1:c
        if(im(x, y)< 128)
        im(x, y) = 0;
        else
          im(x, y) = 250;
        endif
      endfor
    endfor
  imResult = im;
  return;
endfunction
```

Voltando ao código principal, fazemos dois for para percorrer a imagem e subtrair a I da I2 e guardá-la dentro da ImSubtract a fim de utilizara-la como parâmetro para analisar as diferenças entre as duas.

```
#imagem subtraída
for x = 1:l
  for y = 1:c
    ImSubtract(x,y) = (abs(I(x,y)-I2(x,y)));
  endfor
endfor
```

Agora usamos um for duplo para percorrer a ImSubtract e agrupa-la em vetores 3x3 a fim de submete-los a função findAnomalia e retornar um valor de verdadeiro ou falso para nossa variável getAnomalia, além disso, foi colocado um if em cada for a fim de quebrar o loop assim que a primeira anomalia for encontrada.

```
#verificar anomalia imagem 1
for x = 1:l-2
   if(getAnomalia)
      break;
    endif
  for y = 1:c-2
    if(!getAnomalia)
    vetorComparacao =[ImSubtract(x,y),  ImSubtract(x+1,y),  ImSubtract(x+2,y),
                      ImSubtract(x,y+1),ImSubtract(x+1,y+1),ImSubtract(x+2,y+1),
                      ImSubtract(x,y+2),ImSubtract(x+1,y+2),ImSubtract(x+2,y+2)];
    getAnomalia = findAnomalia(vetorComparacao);
  endif

    if(getAnomalia)
      break;
    endif
  endfor
```

Dentro da função findAnomalia nós começamos definido o valor de retorno sendo anomDetect, depois fazemos um for para percorrer nosso vetor 3x3 (9 posições) e, caso algum deles seja diferente de branco(250), consideramos assim que nesse ponto a ImSubtract não apresenta uma grande diferença para considerar i e I2 como diferentes e já retorna nosso anomDetect como falso para continuar o loop da código principal. Agora, caso todo o vetor seja branco(250), foi encontrada uma diferença relevante na ImSubtract , o que significa que a I é diferente da I2, então retornamos o valor da anomDetect como true.

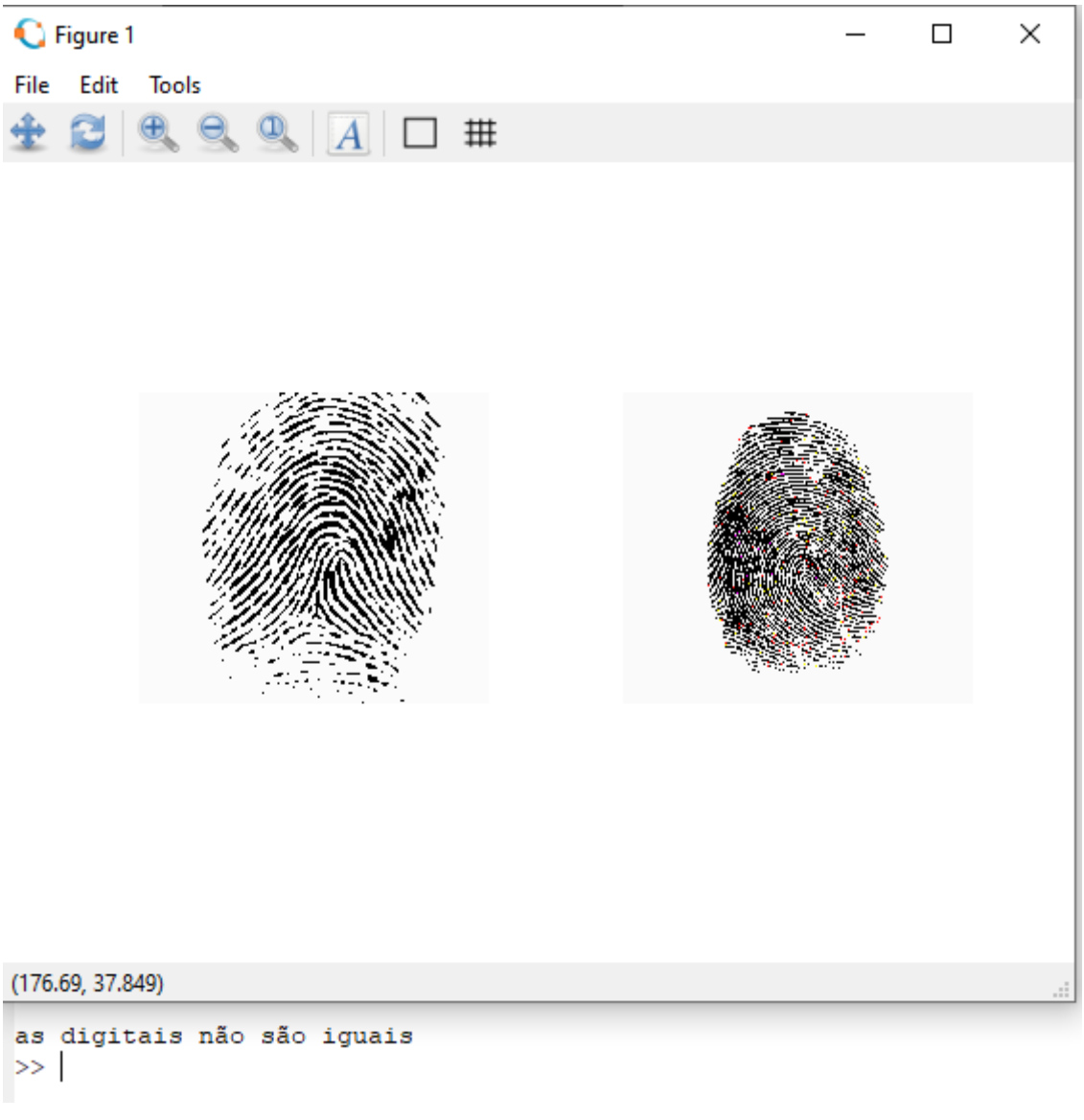
```
#fazer isso l x c vezes
function anomDetect = findAnomalia(vetImg,anomalia)
    for i=1:9
      if(vetImg(i) != 250)
        anomDetect = false;
        return;
      endif
    endfor
    anomDetect = true;
    return;

endfunction
```

Por fim, de volta ao código principal, definimos os outputs quando o getAnomalia for falso, colocando as duas imagens lado a lado com a função subplot() com a mensagem de que as duas imagens são iguais, e, caso getAnomalia seja verdadeiro, fazemos a mesma coisa só mudando a mensagem de que as duas imagens são diferentes.

```
#outputs
if(!getAnomalia)
  subplot(1,2,1);
  imshow(I);
  subplot(1,2,2);
  title({"digitais são iguais"});
  imshow(I2);
  printf("as digitais são iguais\n");
else
   subplot(1,2,1);
  imshow(I);
  subplot(1,2,2);
  title({"digitais não são iguais"});
  imshow(I2);
  printf("as digitais não são iguais\n");
endif
```

Prints do programa:



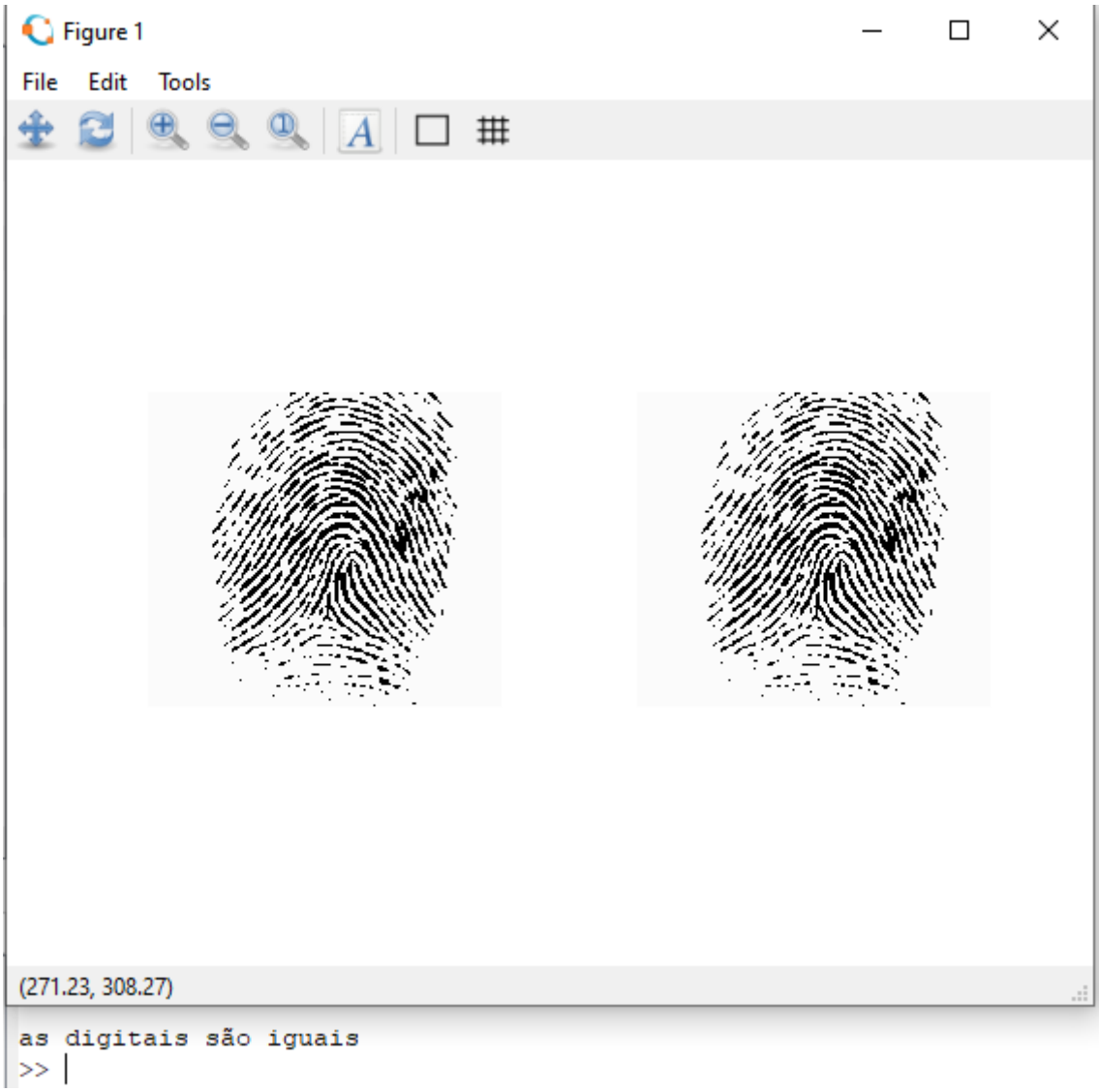
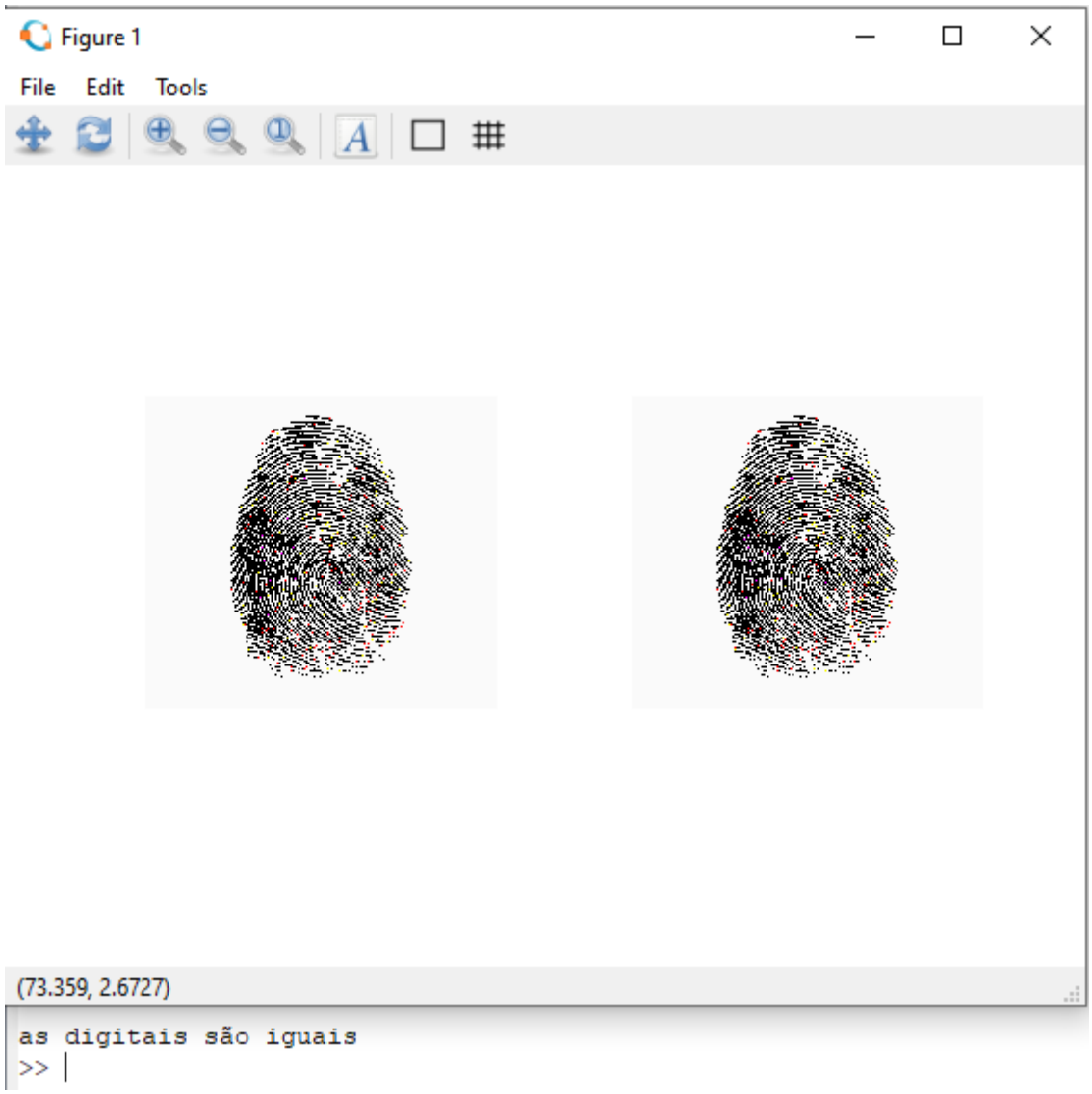


Figure 1
File Edit Tools
(271.23, 308.27)
as digitais são iguais
>>



6. Bibliografia

https://tecnoblog.net/273655/o-que-e-biometria-tecnologia/

https://www.aware.com/pt/reconhecimento-de-impressao-digital/

https://octave.sourceforge.io/

https://octave.org/

7. Fichas de Atividades Práticas Supervisionadas

UNIVERSIDADE PAULISTA

FICHA ATIVIDADES PRÁTICAS SUPERVISIONADAS - APS

NOME:____ Filipi Yukio Iwakami Itoyama | Turma: CC6P28 | RA: N4453J1

CURSO:Ciência da Computação | CAMPUS: São Jos JK - São José do Rio Preto

CÓDIGO DA ATIVIDADE:_____ SEMESTRE:___ 6º | Ano Grade: 3º

DATA DA ATIVIDADE	DESCRIÇÃO DA ATIVIDADE	TOTAL DE HORAS	ASSINATURA DO ALUNO	HORAS ATRIBUÍDAS (1)	ASSINATURA DO PROFESSOR
25/05/2019	Apresentação da disciplina	4	Filipi Yukio Iwakami Itoyama	4	
25/05/2019	Orientação da atividade	4	Filipi Yukio Iwakami Itoyama	4	
25/05/2019	Pesquisa Bibliográfica	10	Filipi Yukio Iwakami Itoyama	10	
25/05/2019	Elaboração de texto	10	Filipi Yukio Iwakami Itoyama	10	
25/05/2019	Correções e orientações	4	Filipi Yukio Iwakami Itoyama	4	
25/05/2019	Montagem do grupo no sistema	2	Filipi Yukio Iwakami Itoyama	2	
25/05/2019	Desenvolvimento do protótipo	10	Filipi Yukio Iwakami Itoyama	10	
25/05/2019	Postagem do trabalho	2	Filipi Yukio Iwakami Itoyama	2	
25/05/2019	Apresentação do trabalho	4	Filipi Yukio Iwakami Itoyama	4	

TOTAL DE HORAS ATRIBUÍDA 50

AVALIAÇÃO:_____

Aprovado ou Reprovado

NOTA:_____

DATA:____/_____/_____

_____

CARIMBO E ASSINATURA DO COORDENADOR DO CURSO

UNIVERSIDADE PAULISTA

FICHA ATIVIDADES PRÁTICAS SUPERVISIONADAS - APS

NOME:____ Luiz Gabriel Zeferino Duarte | Turma: CC6P28 | RA:N454CD8

CURSO:Ciência da Computação | CAMPUS: São Jos JK - São José do Rio Preto

CÓDIGO DA ATIVIDADE:_____ SEMESTRE:___ 6º | Ano Grade: 3º

DATA DA ATIVIDADE	DESCRIÇÃO DA ATIVIDADE	TOTAL DE HORAS	ASSINATURA DO ALUNO	HORAS ATRIBUÍDAS (1)	ASSINATURA DO PROFESSOR
25/05/2019	Apresentação da disciplina	4	Luiz Gabriel Zeferino Duarte	4	
25/05/2019	Orientação da atividade	4	Luiz Gabriel Zeferino Duarte	4	
25/05/2019	Pesquisa Bibliográfica	10	Luiz Gabriel Zeferino Duarte	10	
25/05/2019	Elaboração de texto	10	Luiz Gabriel Zeferino Duarte	10	
25/05/2019	Correções e orientações	4	Luiz Gabriel Zeferino Duarte	4	
25/05/2019	Montagem do grupo no sistema	2	Luiz Gabriel Zeferino Duarte	2	
25/05/2019	Desenvolvimento do protótipo	10	Luiz Gabriel Zeferino Duarte	10	
25/05/2019	Postagem do trabalho	2	Luiz Gabriel Zeferino Duarte	2	
25/05/2019	Apresentação do trabalho	4	Luiz Gabriel Zeferino Duarte	4	

TOTAL DE HORAS ATRIBUÍDA 50

AVALIAÇÃO:_____

Aprovado ou Reprovado

NOTA:_____

DATA:____/_____/_____

_____

CARIMBO E ASSINATURA DO COORDENADOR DO CURSO

UNIVERSIDADE PAULISTA

FICHA ATIVIDADES PRÁTICAS SUPERVISIONADAS - APS

NOME:____ William Rossi do Carmo Ruiz | Turma: CC6P28 | RA:n473gf8

CURSO:Ciência da Computação | CAMPUS: São Jos JK - São José do Rio Preto

CÓDIGO DA ATIVIDADE:_____ SEMESTRE:____ 6º | Ano Grade: 3º

DATA DA ATIVIDADE	DESCRIÇÃO DA ATIVIDADE	TOTAL DE HORAS	ASSINATURA DO ALUNO	HORAS ATRIBUÍDAS (1)	ASSINATURA DO PROFESSOR
25/05/2019	Apresentação da disciplina	4	William Rossi do Carmo Ruiz	4	
25/05/2019	Orientação da atividade	4	William Rossi do Carmo Ruiz	4	
25/05/2019	Pesquisa Bibliográfica	10	William Rossi do Carmo Ruiz	10	
25/05/2019	Elaboração de texto	10	William Rossi do Carmo Ruiz	10	
25/05/2019	Correções e orientações	4	William Rossi do Carmo Ruiz	4	
25/05/2019	Montagem do grupo no sistema	2	William Rossi do Carmo Ruiz	2	
25/05/2019	Desenvolvimento do protótipo	10	William Rossi do Carmo Ruiz	10	
25/05/2019	Postagem do trabalho	2	William Rossi do Carmo Ruiz	2	
25/05/2019	Apresentação do trabalho	4	William Rossi do Carmo Ruiz	4	

TOTAL DE HORAS ATRIBUÍDA 50

AVALIAÇÃO:_____

Aprovado ou Reprovado

NOTA:_____

DATA:____/_____/_____

_____

CARIMBO E ASSINATURA DO COORDENADOR DO CURSO

UNIVERSIDADE PAULISTA

FICHA ATIVIDADES PRÁTICAS SUPERVISIONADAS - APS

NOME:____ Lucas de Oliveira Brandolezi | Turma: CC6P28 | RA:D9380G9

CURSO:Ciência da Computação | CAMPUS: São Jos JK - São José do Rio Preto

CÓDIGO DA ATIVIDADE:_____ SEMESTRE:____ 6º | Ano Grade: 3º

DATA DA ATIVIDADE	DESCRIÇÃO DA ATIVIDADE	TOTAL DE HORAS	ASSINATURA DO ALUNO	HORAS ATRIBUÍDAS (1)	ASSINATURA DO PROFESSOR
25/05/2019	Apresentação da disciplina	4	Lucas de Oliveira Brandolezi	4	
25/05/2019	Orientação da atividade	4	Lucas de Oliveira Brandolezi	4	
25/05/2019	Pesquisa Bibliográfica	10	Lucas de Oliveira Brandolezi	10	
25/05/2019	Elaboração de texto	10	Lucas de Oliveira Brandolezi	10	
25/05/2019	Correções e orientações	4	Lucas de Oliveira Brandolezi	4	
25/05/2019	Montagem do grupo no sistema	2	Lucas de Oliveira Brandolezi	2	
25/05/2019	Desenvolvimento do protótipo	10	Lucas de Oliveira Brandolezi	10	
25/05/2019	Postagem do trabalho	2	Lucas de Oliveira Brandolezi	2	
25/05/2019	Apresentação do trabalho	4	Lucas de Oliveira Brandolezi	4	

TOTAL DE HORAS ATRIBUÍDA 50

AVALIAÇÃO:_____

Aprovado ou Reprovado

NOTA:_____

DATA:____/_____/_____

_____

CARIMBO E ASSINATURA DO COORDENADOR DO CURSO