UNIVERSIDAD NACIONAL DE SAN AGUSTÍN

FACULTAD DE INGENIERÍA PRODUCCIÓN Y SERVICIOS

ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACION

ARQUITECTURA DE COMPUTADORES

Proyecto

Sistema de Verificación de reconocimiento de Estudiantes Universitarios utilizando Webcam y OpenCV y Java-Eclipse

Elaborado por:

Ccasa Quispe Sharmely (grupo A)

Montoya Pinto Sneyder (grupo A)

Índice

1. Introducción (descripción del sistema informatico)
2. Problemática (que se quiere solucionar)
3. Trabajos relacionados (resumen de artículos que deben referenciarse en bibliografía)
4. Herramientas: Software,Librerías y Periféricos Utilizados
   - 4.1 Eclipse Java
   - 4.2 OpenCv 3.1
   - 4.3 Algoritmo Cascade Haar Classifier
   - 4.4 WebCam: especificaciones tecnicas
5. Instalación de Hardware y Software
6. Propuesta de solución general
7. Desarrollo de Software : Código y documentacion
8. Resultados: Interfaces o Pantallas
   - 8.1 Interface 1: Contador de rostros
   - 8.2 Interface 2: Comparación de Imágenes
9. Viabilidad del Proyecto
10. Trabajos Futuros
11. Conclusiones
12. Referencias

Sistema de Verificacion de reconocimiento de Estudiantes Universitarios utilizando Webcam y OpenCV y Java-Eclipse

1. Introducción

   En la actualidad el examen de admisión a las distintas universidades se ha vuelto una meta que todos los estudiantes que terminaron nivel secundario quieren lograr, pero para ello se necesita un esfuerzo constante y su consecuente dedicación, en mérito a ello muchos logran cumplir el sueño de ingresar a la carrera de su preferencia mientras que otros son asaltados por el temor y algunas inseguridades.

2. Problemática

   Debido a aspectos personales o subjetivos u otros factores, algunos estudiantes que deciden postular a la universidad optan por contratar a personas con cierta similitud en el rostro para realizar el examen de admisión. Esto se denomina una suplantación que se está convirtiendo en una práctica común en nuestra ciudad.

   En el Perú acciones como estas ocurren constantemente, por ello se han tomado medidas para evitar que actos como estos sucedan, uno de los más importantes y al que menos importancia se toma , es a la corroboración de persona , siendo este punto solucionado , puramente por el criterio que los encargados en los exámenes tienes, acuñando al hecho de parcial similitud una igualdad completa, pero este erroren precisión no es culpa de los asistentes , puesto que el ojo humano puede ser fácilmente engañado.

3. Trabajos relacionados

   Reconocimiento facial para la autentificación de usuarios (link…)

(

   El desarrollo de este proyecto consiste en crear una herramienta que, mediante el reconocimiento facial, permite la autentificación del usuario a través de imágenes tomadas en tiempo real.

   Con ese fin, se realizarán unas tomas de imágenes del usuario mediante la webcam y del documento de identificación del usuario que serán las muestras a analizar en el proceso de validación. Por una parte, la toma de imagen del usuario se realizará a través del algoritmo Cascade Haar Classifier para detectar el rostro. Por otra parte, en las tomas de imágenes del documento será necesario recurrir a diversas tecnologías como detectores y descriptores de imágenes y el reconocimiento óptico de caracteres con el fin de extraer información del documento de identificación. Para la realización de esta primera parte comentada se utilizará la librería OpenCV.

4. Herramientas: Software,Librerías y Periféricos Utilizados

- Programa de Eclipse IDE java
- Librería openCV v3.1
- Camara o WebCam (especificaciones técnicas de la WebCam)

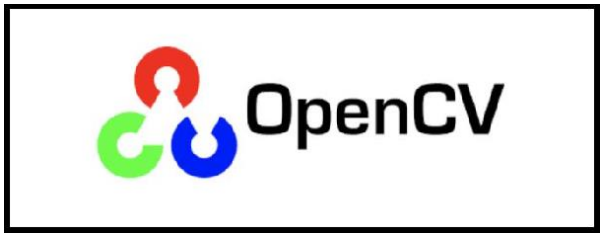
4.1 Eclipse Java

Es un entorno de desarrollo integrado, de Código abierto y Multiplataforma. Es una potente y completa plataforma de programación, desarrollo y compilación de elementos tan variados como sitios web, programas en C++ o aplicaciones Java.

4.2 OpenCv 3.1

OpenCV es una librería software de visión artificial y sistemas de aprendizaje automático siendo diseñado para conseguir una eficiencia computacional elevada y enfocada en aplicaciones de tiempo real desarrollada nativamente en C/C++ y bajo una licencia BSD. Esta licencia permite el uso y modificación del código tanto para fines comerciales como académicos. Actualmente dispone de interfaces de desarrollo o API para C++, C, Python Java y MATLAB siendo compatible con Linux, Mac OS X y Windows, así como también con dispositivos móviles Android y iOS.

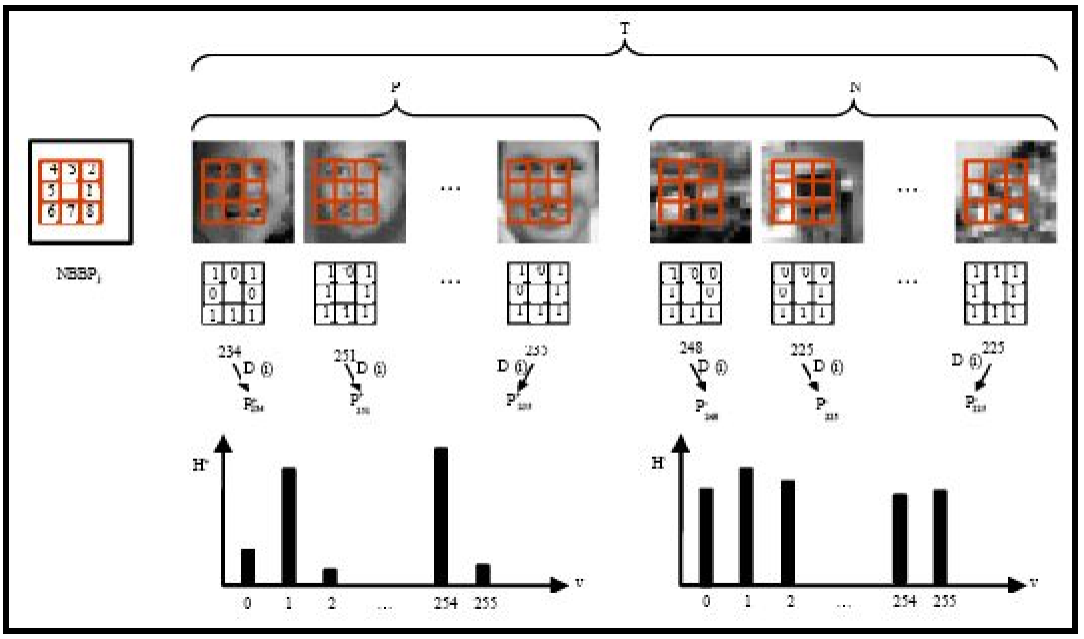
La librería ofrece más de 2500 algoritmos que han sido optimizados llegando a abarcar un conjunto completo de algoritmos en el campo de la visión artificial y del aprendizaje automático. Estos algoritmos que nos proporciona pueden ser usados para detectar y reconocer rostros, identificación de objetos, detectar y clasificar las acciones humanas en videos, el seguimiento de objetos en movimiento, crear composiciones de imágenes de 360 grados de alta revolución, eliminación de los rojos ojos en imágenes o seguir los movimientos oculares, entre un largo etc.



4.3 Algoritmo Cascade Haar Classifier

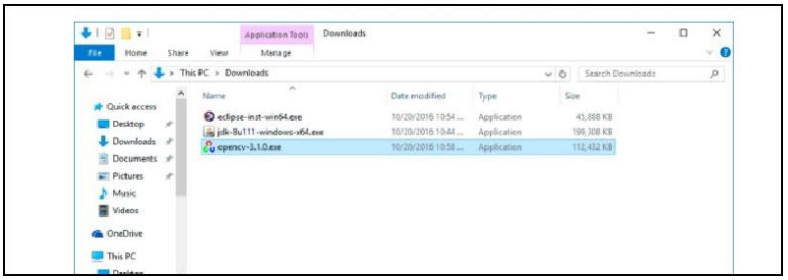
El algoritmo Cascade Haar Classifier, también conocido como algoritmo de Viola Jones, es una aportación de Paul Viola y Michel Jones al mundo computacional de la detección de objetos.

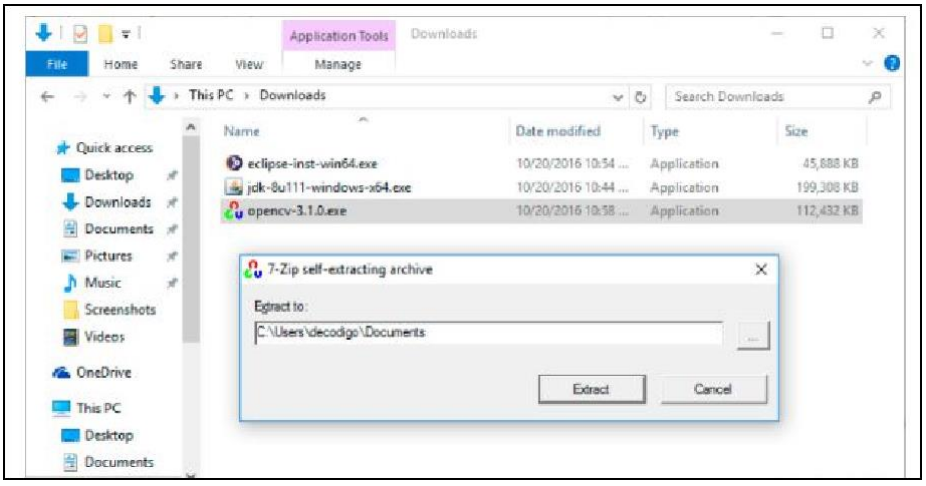
El algoritmo se basa en una serie de clasificadores débiles denominados Haar-Like features que se calculan de forma eficiente a partir de una imagen integral. Estos clasificadores se agrupan en cascada empleando un algoritmo de aprendizaje basado en AdaBoost para conseguir una capacidad discriminativa en las primeras etapas logrando así un alto rendimiento.



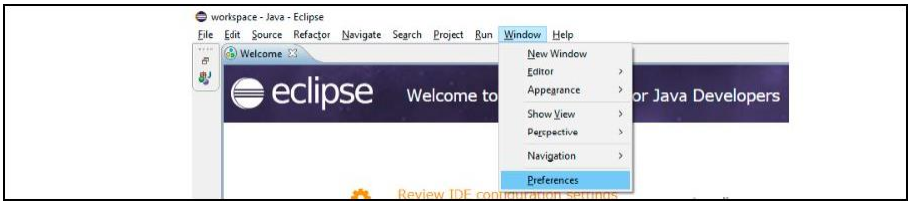
5. Instalación de Hardware y Software

Después de descargar y dar click en el archivo opencv-3.1.0.exe, una ventana de diálogo te preguntará donde deseas extraer OpenCV,elige la ruta que desees y recuerdala,más tarde se utilizara.

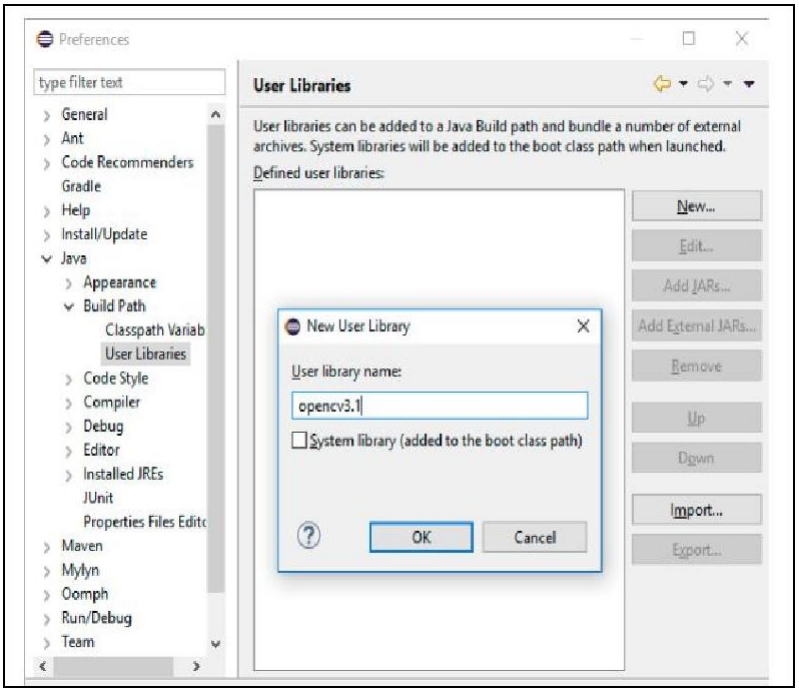




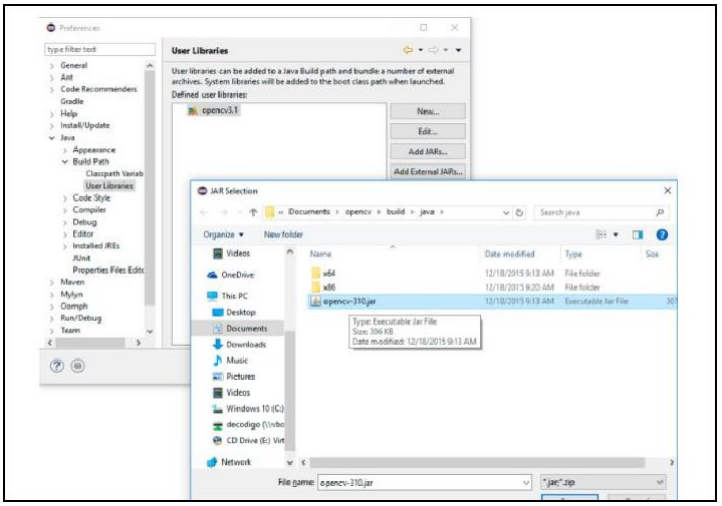
Con Eclipse ya instalado, se debe ir al menú principal y seleccionar la opción **Window >Preferences**



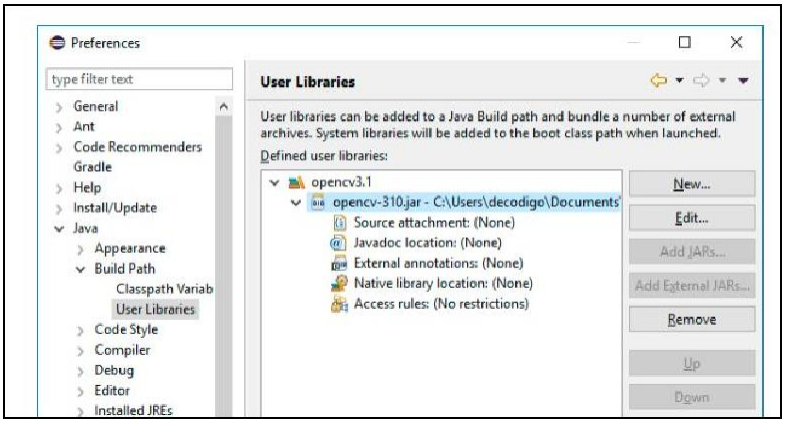
Se creara una librería de usuario nueva, Se busca la opción **Java>Buil Path>User Libraries>New**



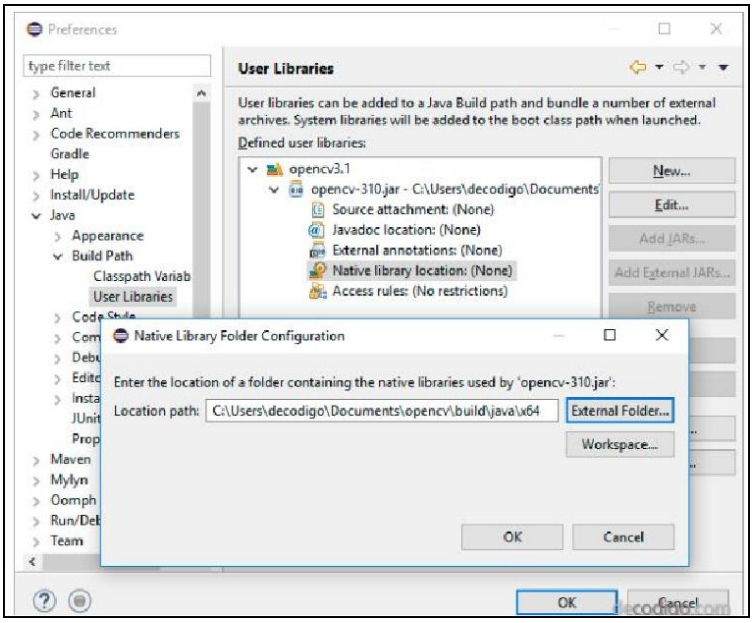
Se crear una nueva librería y se le nombra como opencv.3.1 y se presiona enter despues se seleccionara la nueva libreria y seleccionamos **Add External JARs**



En la carpeta **OpenCV** buscar el **archivo opencv\build\java\opencv-3.1.0.jar** y se selecciona.

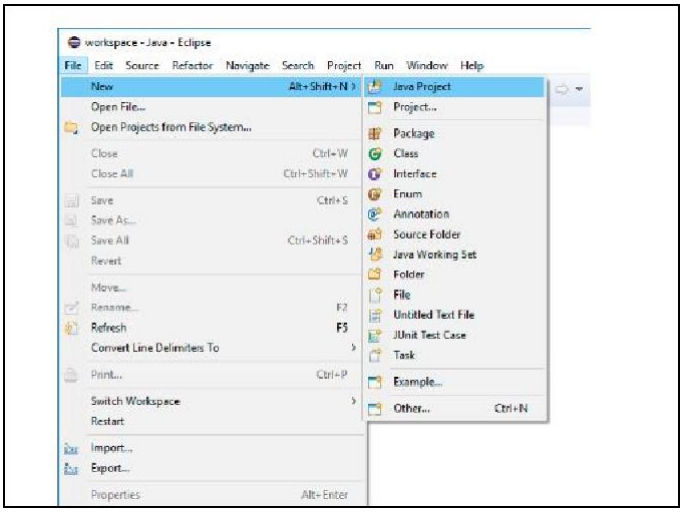


Una vez seleccionado el archivo correcto se podrá visualizar qué opciones nuevas aparecen. Se procederá a seleccionar Native library location

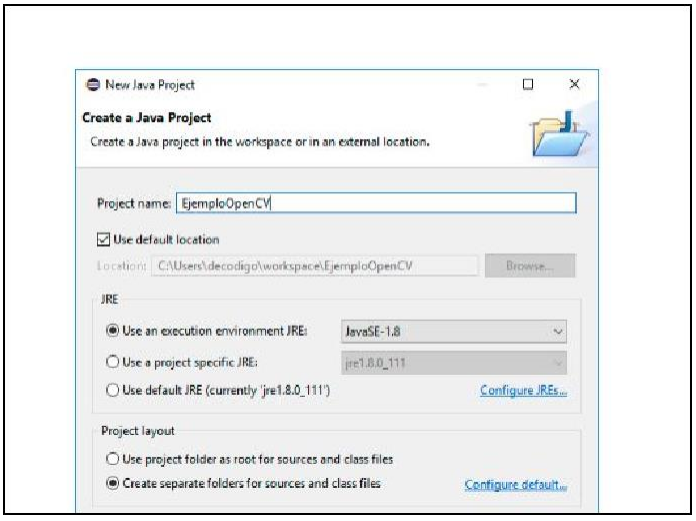


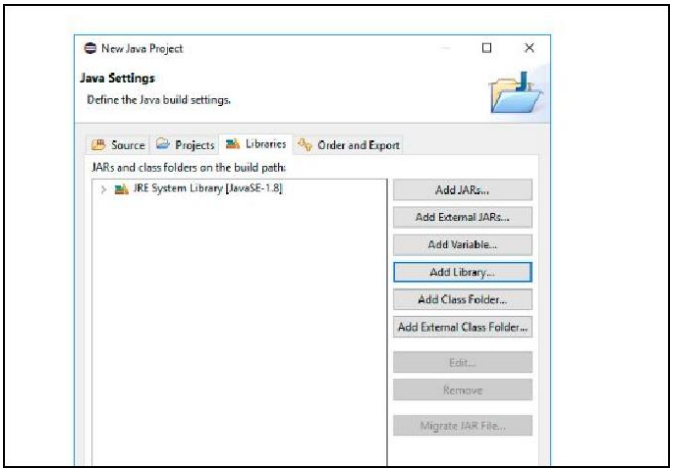
Se da click en **Edit** y se procede a seleccionar la carpeta **opencv/build/java/x64** Presionas OK y termina el proceso.

Después se procede a crear un nuevo proyecto y se utiliza la librería creada.

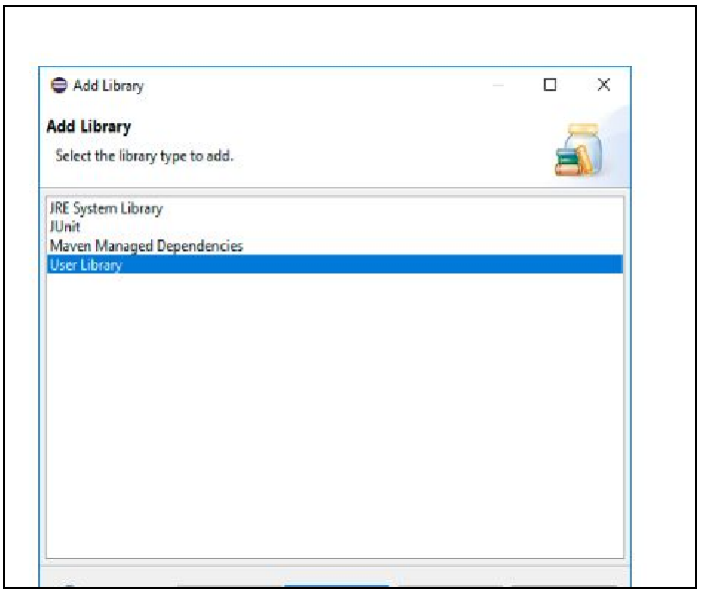


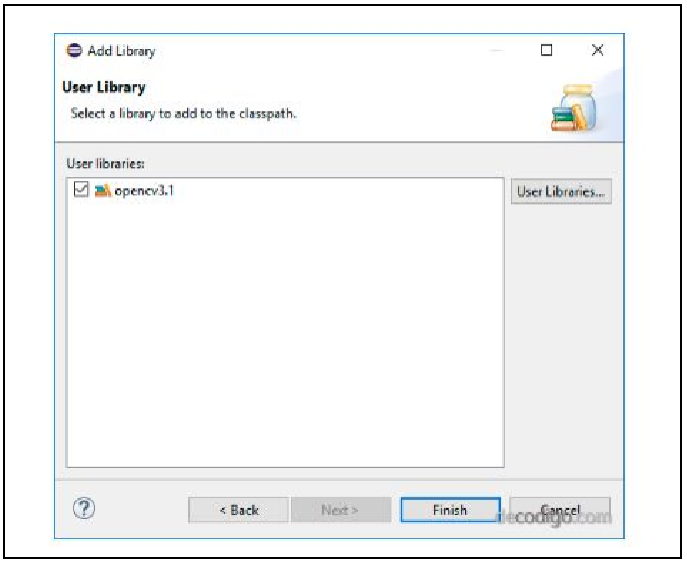
Se le da nombre al proyecto.

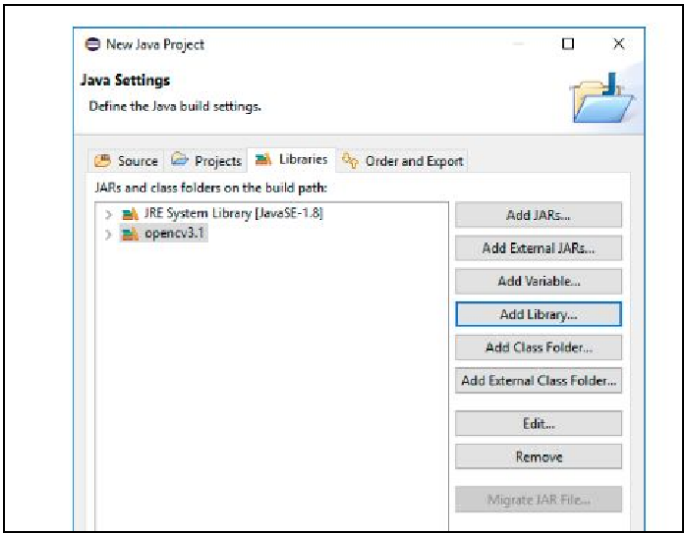




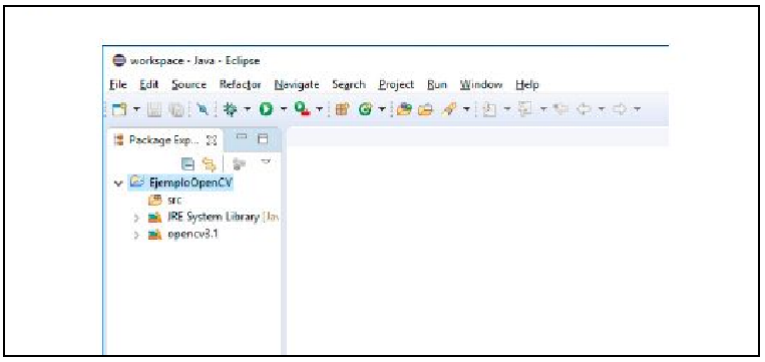
Seleccionamos la librería creada.







Al finalizar correctamente los pasos , la librería podrá visualizarse.



6. Propuesta de Solución a la Problematica

   Para solucionar la problemática, se propone un programa capaz de detectar el rostro de una persona como punto inicial, después de ser detectado el rostro se procederá a realizarse una captura que se almacenará y se comparará con una foto en el banco de imágenes ya establecido.
   De esa manera se podrá verificar correctamente si el estudiante es en efecto la persona que dice ser y no una persona contratada.
   (Incluir un diagrama de bloques o esquema que grafique la solución propuesta)

7. Desarrollo de Software: Código y Documentacion

```
İmport java.awt.BorderLayout;
import java.awt.Color;
import  java.awt.Graphics;
import java.awt.image.BufferedImage;
import java.io.ByteArrayInputStream;
import java.io.IOException;
import javax.imageio.ImageIO;
import javax.swing.JFrame;
import javax.swing.JPanel;
import org.opencv.core.Core;
import org.opencv.core.Mat;
import org.opencv.core.MatOfByte;
import org.opencv.core.MatOfRect;
import org.opencv.core.Point;
import org.opencv.core.Rect;
import org.opencv.core.Scalar;
import org.opencv.imgcodecs.Imgcodecs;
import org.opencv.imgproc.Imgproc;
import   org.opencv.objdetect.CascadeClassifier;
import   org.opencv.videoio.VideoCapture;

class PanelDeRostros extends JPanel {
	private static final long serialVersionUID = 1L;
	private BufferedImage imagen;

	public PanelDeRostros() {
		super();
	}

	/*
	 * Convierte y escribe una Matriz en un objeto BufferedImage
```

```
	 */
	public boolean convierteMatABufferedImage(Mat matriz) {
		MatOfByte mb = new MatOfByte();
		Imgcodecs.imencode("ima.jpg", matriz, mb);try
		{
			 this.imagen = ImageIO.read(new ByteArrayInputStream(mb.toArray()));
		} catch (IOException e) {
			e.printStackTrace();
			return false; // error
		}
		return true; // éxito
	}

	public void paintComponent(Graphics g) {
		super.paintComponent(g);
		if (this.imagen == null)
			return;
		g.drawImage(this.imagen, 10, 10, this.imagen.getWidth(), this.imagen.getHeight(),
null);

	}
}

class DetectorRostros {
	private CascadeClassifier clasificador;

	public DetectorRostros() {
		//Se lee el archivo Haar que le permite a OpenCV detectar rostros frontales en una
imagen
		clasificador = new CascadeClassifier("D:/workspace/Librerias
java/opencv/sources/data/haarcascades/haarcascade_frontalface_alt.xml");
		if (clasificador.empty()) {
			System.out.println("Error de lectura.");
			return;
		} else {
			System.out.println("Detector de rostros leido.");
		}
	}


	public Mat detecta(Mat frameDeEntrada) {
		Mat mRgba = new Mat();
		Mat mGrey = new Mat();
		MatOfRect rostros = new MatOfRect();
		frameDeEntrada.copyTo(mRgba);
		frameDeEntrada.copyTo(mGrey);
		Imgproc.cvtColor(mRgba, mGrey, Imgproc.COLOR_BGR2GRAY);
		Imgproc.equalizeHist(mGrey, mGrey);
		clasificador.detectMultiScale(mGrey, rostros);
		System.out.println(String.format("Detectando %s rostros",
rostros.toArray().length));
		for (Rect rect : rostros.toArray()) {
			//Se dibuja un rectángulo donde se ha encontrado el rostro
			Imgproc.rectangle(mRgba, new Point(rect.x, rect.y), new Point(rect.x +
rect.width, rect.y + rect.height), new Scalar(255, 0, 0));

		}
		return mRgba;
	}
}

public class Principal {

	public static void main(String arg[]) throws InterruptedException {
		// Leyendo librería nativa
```

```
                System.loadLibrary(Core.NATIVE_LIBRARY_NAME);

                // Se crea el JFrame
                JFrame frame = new JFrame("Detección de rostros");
                frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

                DetectorRostros detectorRostros = new DetectorRostros();
                PanelDeRostros panel = new PanelDeRostros();
                frame.setSize(400, 400);
                frame.setBackground(Color.BLUE);
                frame.add(panel, BorderLayout.CENTER);
                frame.setVisible(true);

                // Se crea una matriz que contendrá la imagen
                Mat imagenDeWebCam = new Mat();
                VideoCapture webCam = new VideoCapture(0);

                if (webCam.isOpened()) {
                        Thread.sleep(500);
                    // Se interrumpe el thread para permitir que la webcam se inicialice
                        while (true) {
                                webCam.read(imagenDeWebCam);
                                if (!imagenDeWebCam.empty()) {
                                        Thread.sleep(200);
                                    // Permite que la lectura se
frame.setSize(imagenDeWebCam.width() + 40, imagenDeWebCam.height() + 60);

        // Invocamos la rutina de opencv que detecta rostros sobre la imagen obtenida por la webcam
          imagenDeWebCam = detectorRostros.detecta(imagenDeWebCam);
                                        // Muestra la imagen

panel.convierteMatABufferedImage(imagenDeWebCam);
                                        panel.repaint();
                                } else {
                                        System.out.println("No se capturó nada");
                                        break;
                                }
                        }
                }
                webCam.release(); // Se libera el recurso de la webcam
        }
}
```

```
 package trabajo;

import  trabajo.Control2;
import java.awt.Component;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.FileReader;
import javax.imageio.ImageIO;
import javax.swing.JDesktopPane;
import javax.swing.JOptionPane;

public class Control implements ActionListener {
  Control2 vista;
```

```
File  fileImagen;
String  imagen1;
String imagen2;

public Control(Control2 v) {
  this.vista = v;
}
/*
 * Banco de Postulantes
 */

public void actionPerformed(ActionEvent e) {
  int  returnVal  =  this.vista.jfcExaminarEntrada.showOpenDialog((Component)this.vista);if
  (e.getActionCommand().equals("Banco de Postulantes") && returnVal == 0) {
    this.fileImagen = this.vista.jfcExaminarEntrada.getSelectedFile();
    this.vista.txt_url_left.setText(this.fileImagen.toString());
    this.imagen1  =  this.fileImagen.toString();
  }
  if (e.getActionCommand().equals("Cargar") && this.fileImagen != null)
    cargarImagen(this.vista.jDesktopPane1, this.fileImagen);
  returnVal  =  this.vista.jfcExaminarEntrada.showOpenDialog((Component)this.vista);

  /*
   * Postulante
   */
  if (e.getActionCommand().equals("Postulante") && returnVal == 0) {
            this.fileImagen = this.vista.jfcExaminarEntrada.getSelectedFile();
    this.vista.txt_url_right.setText(this.fileImagen.toString());
    this.imagen2  =  this.fileImagen.toString();
  }
  if (e.getActionCommand().equals("Cargar Foto") &&
    this.fileImagen != null)
    cargarImagen(this.vista.jDesktopPane2,   this.fileImagen);
  if (e.getActionCommand().equals("COMPARAR IMAGENES"))
    try {
      boolean found = false;
      FileReader fr1 = new FileReader(getImagen1());
      FileReader fr2 = new FileReader(getImagen2());
      while (true) {
        int pix1 = fr1.read();
        int pix2 = fr2.read();
        if (pix1 != pix2)
          break;
        if (pix1 == -1) {
          found = true;
          break;
        }
      } //respuestas
      if (found) {
        JOptionPane.showMessageDialog((Component)this.vista,  "Rostro  de  estudiante  confirmado");
      } else {
        JOptionPane.showMessageDialog((Component)this.vista,  "Rostros  diferentes  _  ¡REPORTE
USURPACION DE IDENTIDAD!");
      }
    } catch (Exception ex) {
      ex.printStackTrace();
    }
 }
 //Banco de Preguntas
 public String getImagen1() {
  return this.imagen1;
 }
 //Postulante
```

```
  public String getImagen2() {
    return this.imagen2;
  }

  public void cargarImagen(JDesktopPane jDeskp, File fileImagen) {
    try {
      BufferedImage image = ImageIO.read(fileImagen);
      jDeskp.setBorder(new Imagen(image));
    } catch (Exception e) {
      JOptionPane.showMessageDialog(jDeskp, "error");
    }
  }
}
```

```
package trabajo;
import  java.util.*;
import  trabajo.Control2;
import  java.awt.Component;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.FileReader;
import javax.imageio.ImageIO;
import javax.swing.JDesktopPane;
import  javax.swing.JOptionPane;

public class Control implements ActionListener {
  Control2 vista;
  File  fileImagen;
  String  imagen1;
  String  imagen2;

  public Control(Control2 v) {
    this.vista = v;
  }
  /*
   * Banco de Postulantes
   */

  public void actionPerformed(ActionEvent e) {
    int  returnVal  =  this.vista.jfcExaminarEntrada.showOpenDialog((Component)this.vista);if
    (e.getActionCommand().equals("Banco de Postulantes") && returnVal == 0) {
      this.fileImagen = this.vista.jfcExaminarEntrada.getSelectedFile();
      this.vista.txt_url_left.setText(this.fileImagen.toString());
      this.imagen1  =  this.fileImagen.toString();
    }
    if (e.getActionCommand().equals("Cargar") && this.fileImagen != null)
      cargarImagen(this.vista.jDesktopPane1, this.fileImagen);
    returnVal  =  this.vista.jfcExaminarEntrada.showOpenDialog((Component)this.vista);

    /*
     * Postulante
     */
    if (e.getActionCommand().equals("Postulante") && returnVal == 0) {
            this.fileImagen = this.vista.jfcExaminarEntrada.getSelectedFile();
      this.vista.txt_url_right.setText(this.fileImagen.toString());
      this.imagen2  =  this.fileImagen.toString();
    }
    if (e.getActionCommand().equals("Cargar Foto") &&
      this.fileImagen != null)
```

```
    cargarImagen(this.vista.jDesktopPane2,  this.fileImagen);
   if (e.getActionCommand().equals("COMPARAR IMAGENES"))
    try {
     boolean found = false;
     FileReader fr1 = new FileReader(getImagen1());
     FileReader fr2 = new FileReader(getImagen2());
     while (true) {
      int pix1 = fr1.read();
      int pix2 = fr2.read();
      if (pix1 != pix2)
       break;
      if (pix1 == -1) {
       found = true;
       break;
      }
     } //respuestas
     if (found) {
      JOptionPane.showMessageDialog((Component)this.vista, "Rostro  de  estudiante  confirmado");
     } else {
      JOptionPane.showMessageDialog((Component)this.vista, "Rostros diferentes _ ¡REPORTE
USURPACION DE IDENTIDAD!");
     }
    } catch (Exception ex) {
     ex.printStackTrace();
    }
  }
  //Banco de Preguntas
  public String getImagen1() {
   return this.imagen1;
  }
  //Postulante
  public String getImagen2() {
   return this.imagen2;
  }

  public void cargarImagen(JDesktopPane jDeskp, File fileImagen) {
   try {
    BufferedImage image = ImageIO.read(fileImagen);
    jDeskp.setBorder(new  Imagen(image));
   } catch (Exception e) {
    JOptionPane.showMessageDialog(jDeskp, "error");
   }
  }
}
```

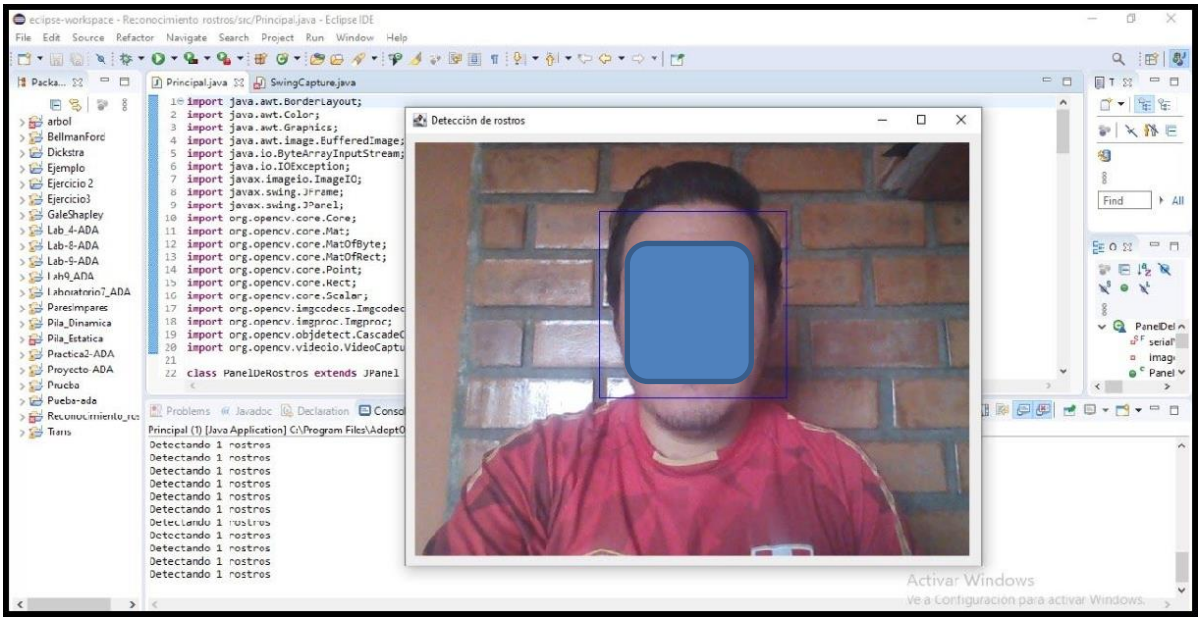
```
package trabajo;

import java.awt.Component;
import java.awt.Graphics;
import java.awt.Insets;
import java.awt.image.BufferedImage;
import javax.swing.border.Border;

public class Imagen implements Border {
  private BufferedImage image;

  public Imagen(BufferedImage image) {
   this.image = image;
  }

  public void paintBorder(Component c, Graphics g, int x, int y, int width, int height) {
```

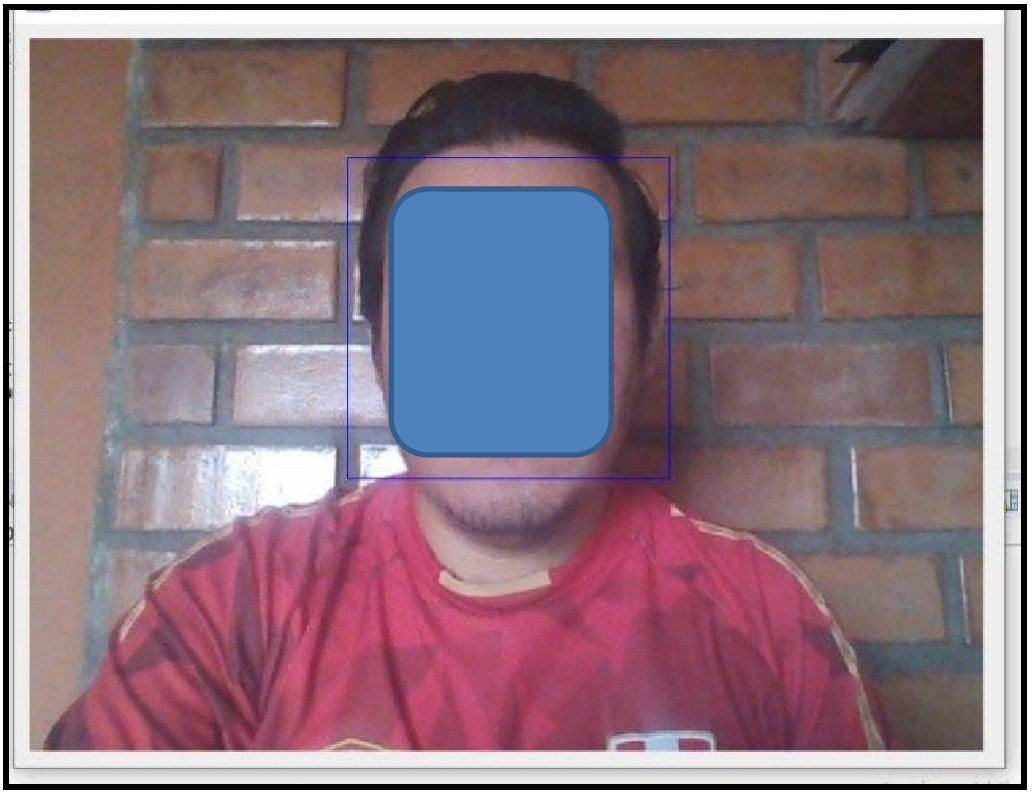
```
    int x0 = x + (width - this.image.getWidth()) / 2;
    int y0 = y + (height - this.image.getHeight()) / 2;
    g.drawImage(this.image, x0, y0, null);
  }

  public Insets getBorderInsets(Component c) {
    return new Insets(0, 0, 0, 0);
  }

  public boolean isBorderOpaque() {
    return true;
  }
}
```

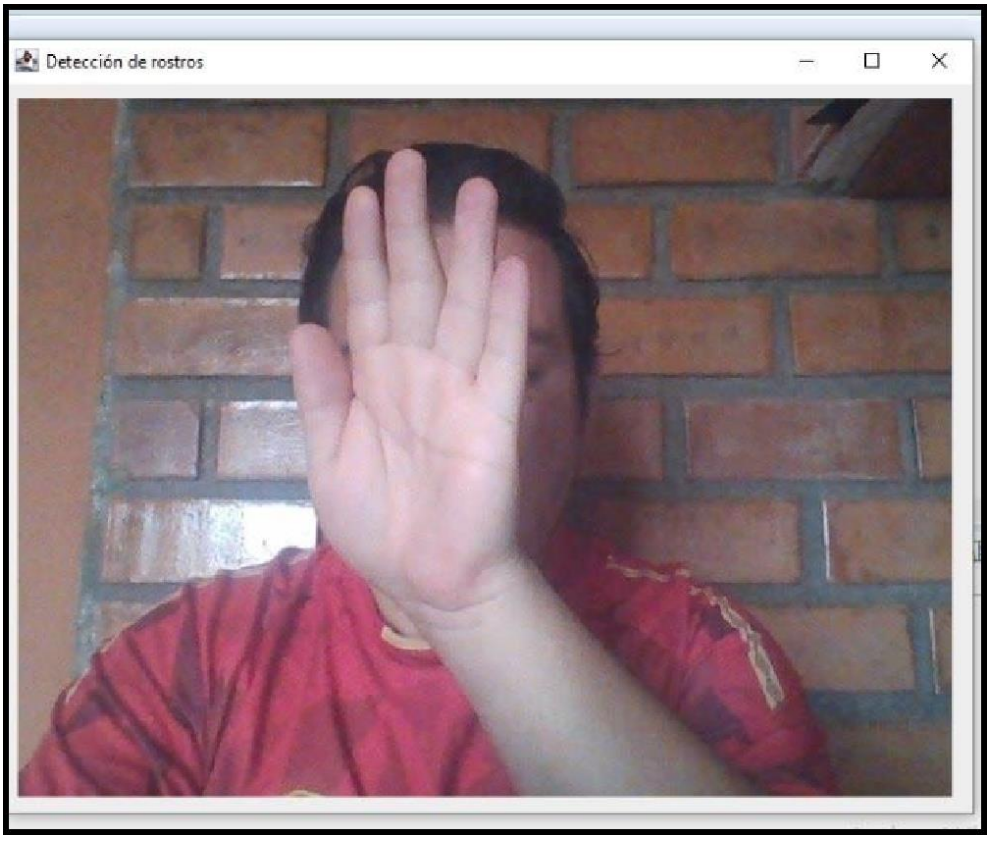
8. Interfaces

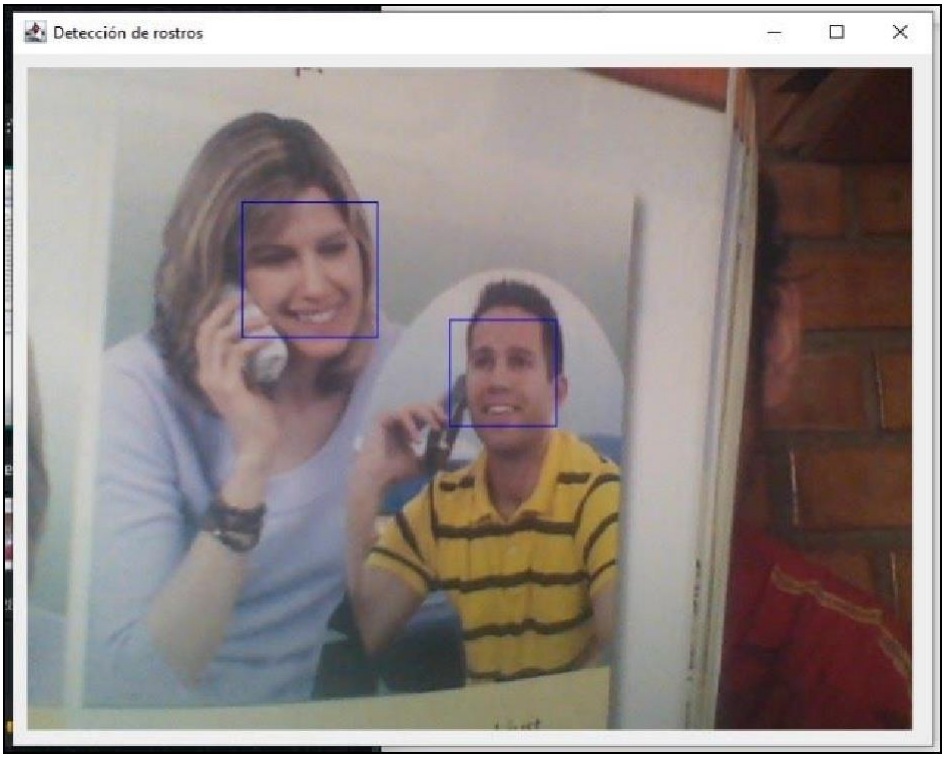
Base de datos: Banco de imágenes de estudiantes

8.1 Reconocimiento de Rostros

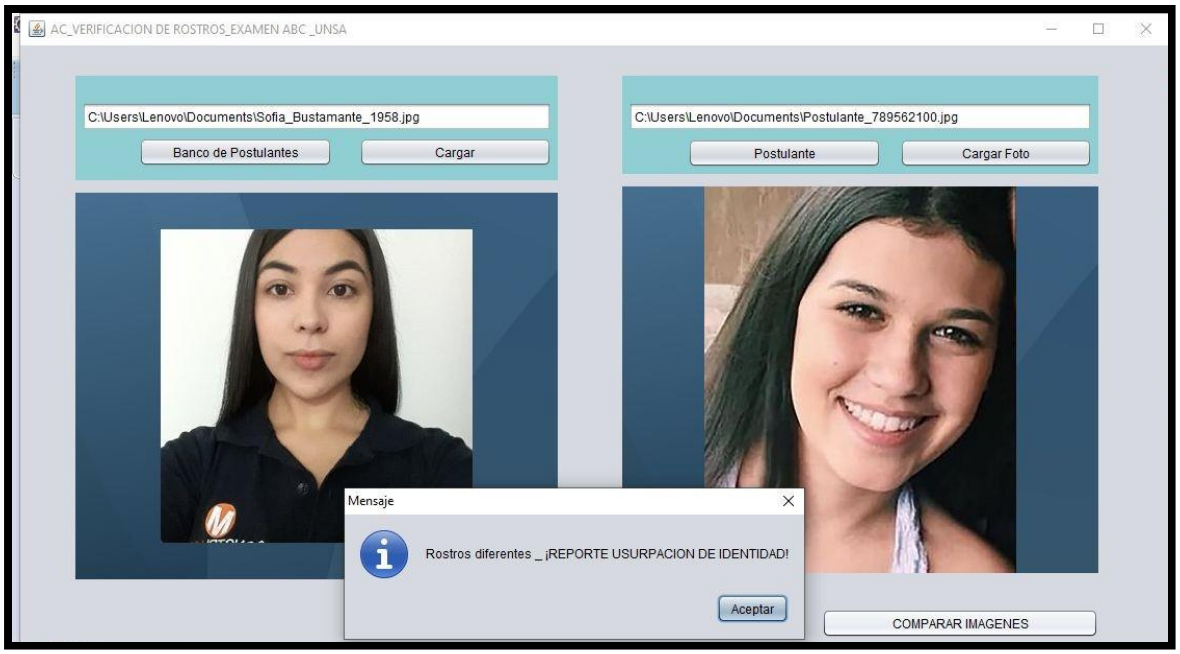


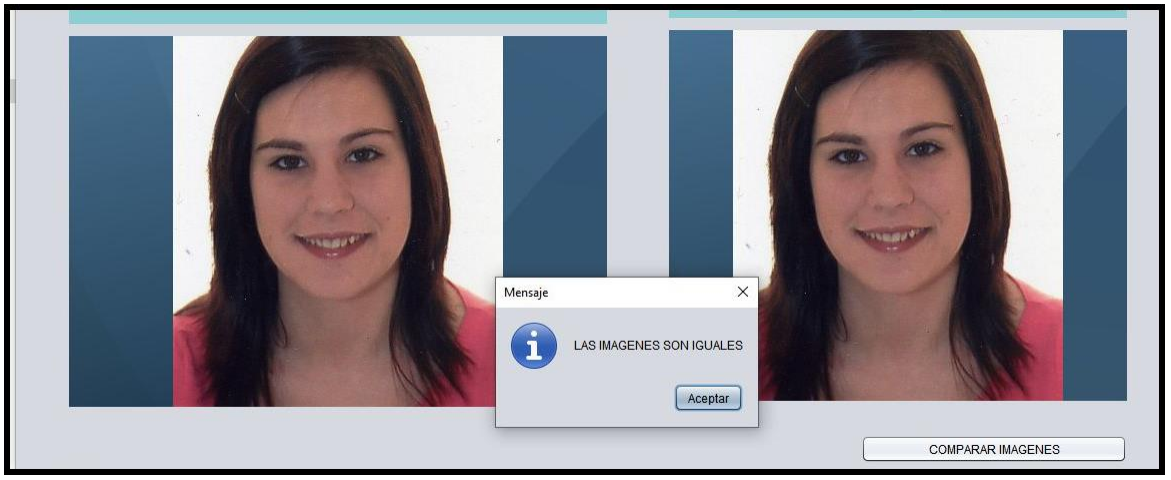


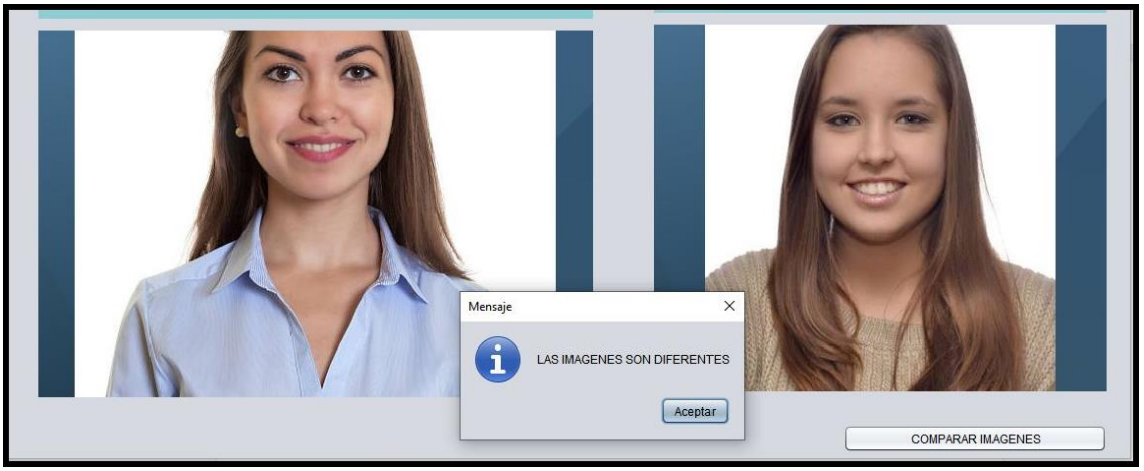

Detección de rostros

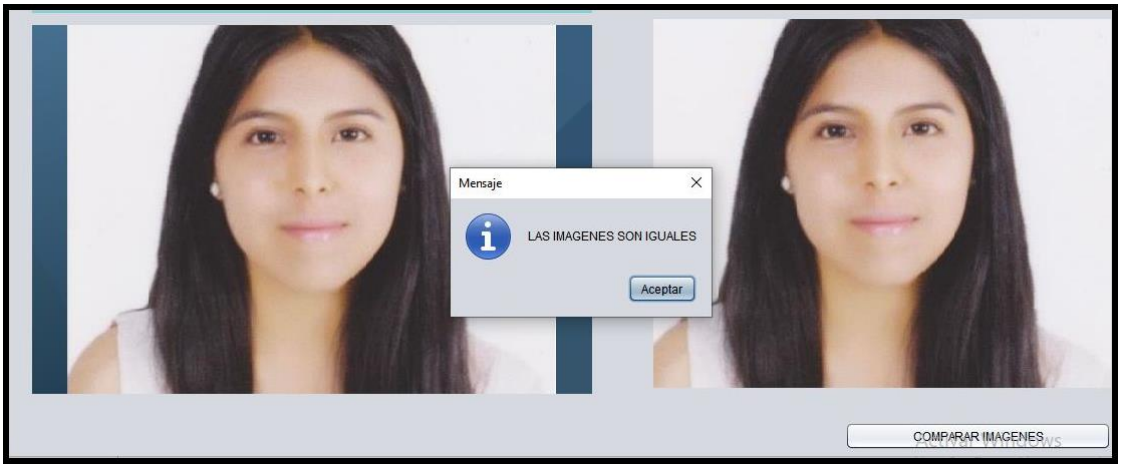


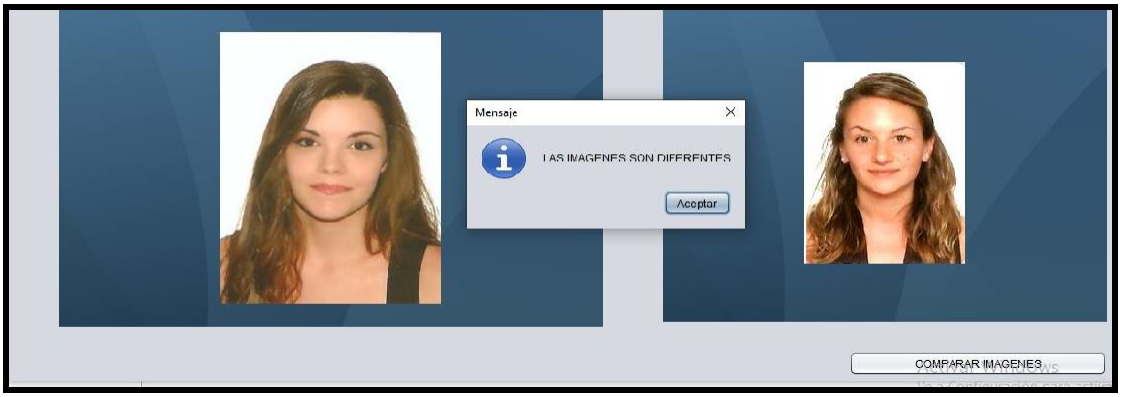
8.2 Comparación de Imágenes

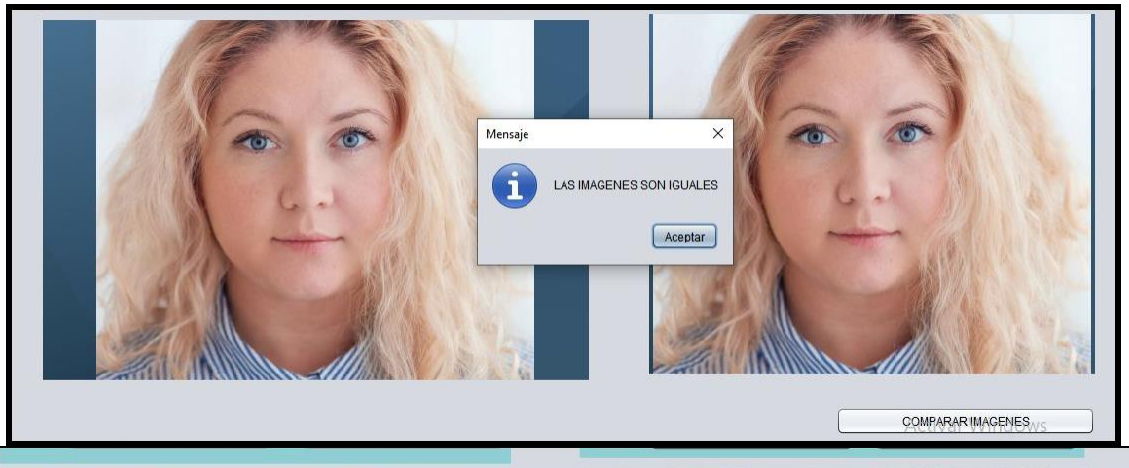

AC_VERIFICACION DE ROSTROS_EXAMEN ABC _UNSA
C:\Users\Lenovo\Documents\Sofia_Bustamante_1958.jpg
Banco de Postulantes
Cargar
C:\Users\Lenovo\Documents\Postulante_789562100.jpg
Postulante
Cargar Foto
Mensaje
Rostros diferentes _ ¡REPORTE USURPACION DE IDENTIDAD!
Aceptar
COMPARAR IMAGENES


Mensaje
LAS IMAGENES SON IGUALES
Aceptar
COMPARAR IMAGENES


Mensaje
LAS IMAGENES SON DIFERENTES
Aceptar
COMPARAR IMAGENES


Mensaje
LAS IMAGENES SON IGUALES
Aceptar
COMPARAR IMAGENES


Mensaje
LAS IMAGENES SON DIFERENTES
Aceptar
COMPARAR IMAGENES


Mensaje
LAS IMAGENES SON IGUALES
Aceptar
COMPARAR IMAGENES


Mensaje
LAS IMAGENES SON IGUALES
Aceptar

9. Viabilidad del Proyecto

   De acuerdo con lo desarrollado hasta el momento y la investigación realizada es posible concluir con satisfacción el proyecto en su totalidad logrando realizar el prototipo de detección y comparación de rostro

10. Trabajos Futuros

    El proyecto puede ser mejorado ,se le pueden aplicar otros tipos de verificación, apoyados de hardware externo, el cual permite aseverar la respuesta dada por el programa de esta manera se tendrá una respuesta mucho más óptima.

11. Conclusiones

    Las ventajas del presente proyecto junto con un desarrollo más profundo beneficiaria al correcto procedimiento en distintas actividades que verifiquen a las personas y control de una gran cantidad de estudiantes.

12. Referencias

https://opencv.org(Libreria)
https://www.eclipse.org/downloads/packages/release/kepler/sr1/eclipse-ide-java-developers
Martinez Guerrero, M. (2018, julio). “Reconocimiento facial para la autentificación de usuarios”. Escola Tècnica Superior d’Enginyeria Informàtica.
https://docs.opencv.org/master/d9/df8/tutorial_root.html
Links de los artículos de investigación relacionados al proyecto
.
.
.
.
.