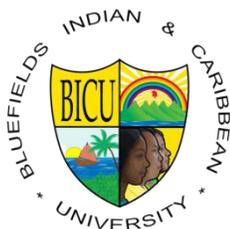


BLUEFIELDS INDIAN AND CARIBBEAN UNIVERSITY

BICU



Facultad de Ciencias de la Educación y Humanidades (FACEYH)

Escuela de Informática
Ingeniería de Sistema de Información

Proyecto para optar al título de Ingeniero de Sistemas

Título

Sistema de reconocimiento facial de control de entrada y salida de los empleados de la universidad BICU recinto Bluefields, primer semestre del año 2020

Autores:

Br. Yelsy Jiany Allen Ramirez

Br. Lorvick Kayton Tucker Knight

Tutor:

MSc. Dexon Mckensy Sambola

Bluefield's, RACC.S,
Nicaragua, Julio del 2021

“La Educación es la mejor opción para el desarrollo de los pueblos”

INDICE DE CONTENIDO

I. INTRODUCCIÓN	1
II. JUSTIFICACIÓN	2
III. OBJETIVOS	3
3.1. General	3
3.2. Objetivos de ejecución	3
IV. DIAGNÓSTICO DE LA SITUACIÓN	4
V. ESTADO DEL ARTE	5
5.1. Análisis de algoritmos y tecnologías basadas en reconocimiento facial	6
5.1.1. Reconocimiento Facial con el algoritmo de VIOLA-JONES	6
5.1.2. CARACTERÍSTICAS HAAR	7
5.1.3. ADAPTIVE BOOSTING	9
5.2. TÉCNICAS DE RECONOCIMIENTO FACIAL	10
5.2.1. PCA (PRINCIPAL COMPONENT ANALYSIS)	11
5.2.2. LDA (LINEAR DISCRIMINANT ANALYSIS)	12
5.2.3. EIGENFACES	14
5.2.4. FISHESFACES	16
5.2.5. APPLE FACEID	17
5.2.6. DEEPFACE	18
5.3. ANÁLISIS DE ESTUDIOS	19
5.3.1. Sistema informático de reconocimiento facial para el registro y control de asistencia de los socios de la cooperativa de taxis y camionetas puyo	20
5.3.2. Sistema de Notificación de Agenda a través de Reconocimiento Facial: Prototipo para Docentes a Tiempo Completo de la Carrera de Computación de la Universidad Católica de Santiago de Guayaquil	20
5.3.3. Análisis y Diseño de un Sistema de Reconocimiento Facial aplicando Machine Learning para detectar e identificar intrusos	21
5.4. SELECCIÓN DE TECNOLOGÍAS Y METODOLOGÍAS DE DESARROLLO	22
VI. HERRAMIENTAS Y TECNOLOGÍAS USADAS	23
6.1. Local binary patterns histograms (LBPH) Face Recognizer	23
6.2. Python versión 3.5	23
6.3. OpenCv 4.0	23

6.4.	Numpy	23
6.5.	Pandas.....	23
6.6.	Tkinter	24
6.7.	MySQL.....	24
6.8.	ReportLab.....	24
6.9.	WebCam V-U0026.....	24
VI.	MODELO DE DESARROLLO	25
VII.	DISEÑO METODOLÓGICO.....	26
7.1.	Planeación	26
7.2.	Modelado.....	27
7.3.	Elaboración del Prototipo.....	36
7.4.	Desarrollo, Entrega y Re-alimentación	37
7.4.1.	Módulo Almacenamiento.....	38
7.4.2.	Módulo Procesamiento	38
7.4.3.	Módulo Presentación	39
7.5.	Entrega del Producto Final	39
VIII.	RESOLUCIÓN	40
8.1.	Caso de estudio	40
8.2.	Experimento del caso de estudio.....	41
IX.	CONCLUSIONES Y VÍAS FUTURAS	46
9.1.	Conclusiones	46
9.2.	Vías Futuras.....	46
X.	RECOMENDACIONES	47
XI.	REFERENCIAS	48

LISTA DE ILUSTRACIONES

Ilustración 1. Etapas del método para detección de rostros de Viola-Jones (BARRIGA, 2017)	6
Ilustración 2. Tipo de descriptores utilizados por Haar. (Jones P. V., 2018)	7
Ilustración 3. Última etapa de la cascada de Haar, donde muchos más patrones de rectángulos en blanco y negro deben coincidir con la imagen del candidato. (Harvey, 2012)	8
Ilustración 4. Esquema Algoritmo Viola-Jones (Jones P. V., 2001.)	9
Ilustración 5. Técnicas de Reconocimiento Facial (Davisclick/Taller, 2018).....	9
Ilustración 6. Clasificadores Adaboost (Jain, 2016)	10
Ilustración 7. Comparación entre PCA, LDA (A.M. Martínez Robot Vision Lab., Purdue Univ., West Lafayette, IN, USA, A.C. Kak, 2001)	13
Ilustración 82. Entidad trabajador con su respectivo atributo (Elaborado por: Lorvick y Yelsy)	30
Ilustración 93. La relación entre trabajador, registro de login y registro logout (Elaborado por: Lorvick y Yelsy)	31
Ilustración 104. Entidad registro de login con su respectivo atributo (Elaborado por Lorvick y Yelsy)	32
Ilustración 115. Entidad Registro logout con su respectivo atributo (Elaborado por: Lorvick y Yelsy)	32
Ilustración 1612. Entidad de usuario (Elaborado por: Lorvick y Yelsy).....	33
Ilustración 131. Registrar características faciales (Elaborado por: Lorvick y Yelsy) ..	37
Ilustración 142. Registro de característico facial (Elaborado por: Lorvick y Yelsy) ...	41
Ilustración 153. Identifica y reconoce los rostros registrado (Elaborado por: Lorvick y Yelsy)	42
Ilustración 165. Proceso de registro de horas E/S (Elaborado por : Lorvick y Yelsy). 44	
Ilustración 17. Reporte de entrada y salida de un trabajador.....	45

LISTA DE TABLAS

Tabla 1. Planeación de Actividades 2019 (Elaborado por Lorvick y Yelsy)	27
---	----

RESUMEN

El reconocimiento facial se ha convertido en el método de identificación institucional más precisa en comparación con otros sistemas biométricos. El sistema implica un proceso de algoritmo automático, que verifica y reconoce la identidad de la persona en función de sus características faciales. La tecnología consiste en mejorar la seguridad, control de datos personales y mejorar la regulación institucional, Por lo tanto, el propósito de este proyecto es el desarrollo de un sistema de reconocimiento facial que sea capaz de controlar y registrar la entrada/salida de los empleados de la Bluefields Indian & Caribbean University (BICU) recinto Bluefields. Se incorporó la librería OpenCV para el reconocimiento facial con la función LBPH y Python como la fuente de lenguaje de programación, utilizando el modelo de prototipo, basada en prueba y error. Como resultado se obtuvo un sistema de reconocimiento facial para el buen control y manejo de las horas de las entrada y salida (E/S) de los trabajadores de la universidad BICU, donde cada empleado registrado en el sistema es identificado por los rasgos faciales.

PALABRAS CLAVE: Aprendizaje automático; Algoritmo de aprendizaje; Análisis inteligente de imágenes

ABSTRACT

Facial recognition has become the most accurate method of institutional identification compared to other biometric systems. The system implies an automatic algorithm process, which verifies and recognizes the identity of the person according to their facial characteristics. The technology consists of improving the security, control of personal data and improving the institutional regulation. Therefore, the purpose of this project is the development of a facial recognition system that is able to control and record the entry /exit of employees of the Bluefields Indian & Caribbean University (BICU). The OpenCV library for facial recognition was incorporated with the LBPH function and Python as the source of programming language, using the prototype model, based on trial and error. As a result, a facial recognition system was obtained for the good control and management of the hours of entry and exit (I/O) of the workers of the BICU university, where each employee registered in the system is identified by the facial features.

KEY WORDS: Machine learning; Learning algorithm; Intelligent image analysis

I. INTRODUCCIÓN

El reconocimiento facial es una tarea relativamente fácil por los humanos, fue comprobado en un estudio llamado "El hombre en contra de la maquina" realizado por Kochetkova (2016) . Un ejemplo claro de esto es el caso de los niños quienes rápidamente aprenden a reconocer el rostro de sus padres, es algo natural, pero puede haber dificultades ya que existen enfermedades que afectan el comportamiento del cerebro humano, uno de ello es la prosopagnosia, es un desorden cerebral que produce complicaciones en la identificación de rostros familiares (Sacks, 1985). Esto podría hacer pensar que enseñarle a una computadora como reconocer diferentes rostros es una tarea sencilla, pero desafortunadamente esto no es así.

En este proyecto hemos explorado nuevas tecnologías, en este caso particular lo que es el análisis inteligente de imágenes, al integrar el reconocimiento facial a través de la visión por computadoras, aplicando la misma en el proceso de Entrada y Salida (E/S) de los empleados de la Bluefields Indian and Caribbean University. Para el desarrollo del sistema se utilizó Python como el lenguaje de programación, incorporado con la librería de OpenCV, el cual tiene la función de entrenamiento para la visión por computadora. MySQL como administrador de la base de datos de nuestro sistema y también para guardar los datos de todos los usuarios registrados. Para la interfaz del sistema utilizamos a Tkinter, es multiplataforma, viene incluido con Python y se considera un estándar para la interfaz gráfica de usuario para el lenguaje de programación utilizado.

El sistema de reconocimiento facial (SDRF-BICU) lleva el control de las E/S de los empleados de la BICU, utilizando sus rasgos faciales como identificador para tener acceso a la misma, para registrar las características faciales de un empleador sus datos deberán estar registrado en el sistema (este proceso se da previo al registro facial) para poder llevar control de las E/S de sus horas laboradas y al final de cada día debe generar un reporte el cual estará destinado al área de administración para ser evaluada. También es capaz de generar un reporte en formato PDF para asegurar consistencia y transparencia de las horas laboradas por cada empleado.

II. JUSTIFICACIÓN

En los últimos años, la tecnología de visión por computadora se ha convertido en un desafío digital para todas las empresas, organizaciones, instituciones y especialmente para el gobierno, muchas entidades están empleando estas tecnologías de reconocimiento facial. Ejemplo de esto es Facebook una de las redes sociales más grandes a nivel mundial (Servicio de ayuda, 2021), por ser una herramienta ideal para el control de acceso, seguridad, vigilancia, aplicación de la ley y más. Es una tecnología inteligente, desafiante e interesante para el ser humano.

Actualmente BICU cuenta con dos sistemas de control de entrada y salida de los empleados, la huella digital (sistema biométrico), es para los trabajadores de contrato fijo y el control de asistencia para los de contrato temporal, que se realiza de forma manual creando una deficiencia en el proceso de control, ambos sistemas cumplen con el mismo objetivo. Por esta razón se propuso el desarrollo de SDRF-BICU, una nueva forma de administrar las horas de entrada y salida de los empleados de la BICU usando el rostro como identificación para registrar las horas de trabajo, lo cual es más fácil y flexible de usar, siendo un sistema que genera credibilidad, control y mayor seguridad.

Los beneficiarios directos son del área de recursos humanos de la universidad BICU/Bluefields, por lo que son los encargados de administrar de manera eficiente los procesos de entrada y salida de cada empleado de dicha institución. Y los beneficiarios indirectos son todos los empleados de la universidad BICU.

III. OBJETIVOS

3.1. General

Desarrollar un sistema de reconocimiento facial que sea capaz de administrar la entrada/salida de los empleados de la universidad BICU, recinto Bluefields, durante el primer semestre del año 2020.

3.2. Objetivos de ejecución

- ✓ Registrar las características faciales de los empleados e identificar y reconocer aquellos registrado dentro de la base de conocimiento.
- ✓ Identificar los rostros no registrados dentro de la base de conocimientos y recomendar su registro.
- ✓ Registrar la hora de entrada y salida de los usuarios.
- ✓ Llevar control de registro de datos de los empleadores.
- ✓ Generar reporte de entrada y salida de los empleados.

IV. DIAGNÓSTICO DE LA SITUACIÓN

En los últimos años las tecnologías de reconocimiento facial se han convertido en tema de gran interés para la comunidad científica, así mismo, las instituciones públicas y privadas han mostrado un gran interés en esa temática , principalmente en aspectos de control y seguridad (Yabiku, 2017). Sin embargo, la universidad BICU permanece ajena a la aplicabilidad de esta tecnología, de esta manera surge la posibilidad de realizar un proyecto sin antecedente en dicha institución.

BICU está situado en la ciudad de Bluefields localizado en la Costa Caribe Sur de Nicaragua, actualmente tiene operando un sistema biométrico de huella digital para llevar el control de horario de los trabajadores de contrato fijo que son 116 empleadores, pero presenta inconvenientes en el reconocimiento de las huellas, esto se da mayormente con los trabajadores de aseo por el uso de químicos. También cuenta con un control de asistencia manual para los empleados de contrato temporal que son 67, este control no tiene ningún tipo de seguridad en el manejo de los datos.

Con la SDRF-BICU, se registra las horas laborales mediante reconocimiento facial, promete mayor precisión en la identificación que los sistemas de controles que actualmente implementa la BICU, es capaz de manejar y controlar el registro de todos los trabajadores sea de contrato fijo o no, con solo una opción de desactivación los de contrato temporaria es deshabilitado, pero esta información queda guardada.

V. ESTADO DEL ARTE

La detección facial, consiste en localizar si existe la cara en una imagen. Para poder localizar la cara se pueden utilizar diversos métodos, Como métodos basados en las características geométricas de la cara, estas consisten en utilizar el cálculo de distancias geométricas y otras características que se combinan en un vector y se reduce quedándose con las más determinantes.

También existen los métodos de plantilla que mejoran respecto al anterior dado que guarda en el vector todas las características de una cara en lugar de las más relevantes por lo que a la hora de testear se tiene más información. Por otro lado, podemos encontrar diferentes algoritmos ya testeados previamente como puede ser el Algoritmo ‘top-Down’. Este algoritmo se basa en reglas que detectan caras. Por ejemplo, ubicando los ojos por simetría con respecto al centro de la cara o considerando que la cara ha de ser de color uniforme (Becerra Quispe, 2016).

En contraste con estos algoritmos, están los procedimientos ‘up-Down’. Estos buscan encontrar características que indiquen ciertas regiones de la cara como podrían ser colores, bordes o texturas. Sin embargo, estos algoritmos son sensibles a ruido u oclusión en las imágenes (Navio, 2017).

Los algoritmos denominados ‘Template-matching’ comienzan construyendo una plantilla predefinida a partir de una imagen de una cara establecida o el borde de una cara. Posteriormente aplica la correlación de esta plantilla a las imágenes de entrada y en las regiones que se obtenga una mayor respuesta se considerará que existe una cara, el problema de este tipo de algoritmos es que no es útil cuando nos encontramos ante variaciones de escala u orientaciones diferentes.

Uno de los algoritmos más nombrados en este ámbito es el algoritmo de **Viola-Jones**, es un algoritmo muy popular implementado en diversos lenguajes de programación con una alta tasa de acierto y con un coste computacional bajo. En la ilustración 1, que se muestra a continuación se podrá apreciar las etapas del método para detección de rostros de Viola-Jones.

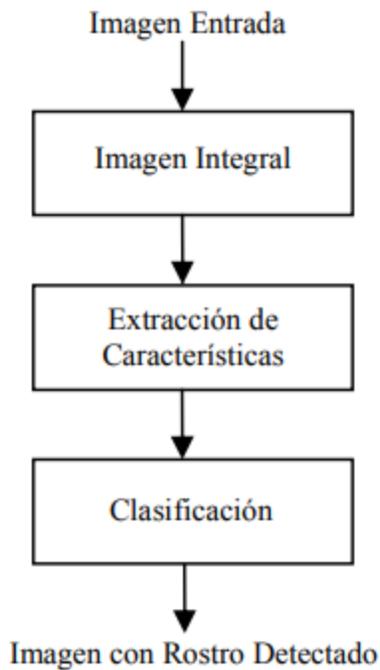


Ilustración 1. Etapas del método para detección de rostros de Viola-Jones (BARRIGA, 2017)

5.1. Análisis de algoritmos y tecnologías basadas en reconocimiento facial

En este apartado se abordará un conjunto de algoritmos y tecnología desarrollados para el reconocimiento de rostro, como se implementan y de qué manera.

5.1.1. Reconocimiento Facial con el algoritmo de VIOLA-JONES

Viola-Jones es un algoritmo capaz de detectar objetos y caras en tiempo real. Este algoritmo fue propuesto por Paul Viola y Michael Jones en 2001. La característica principal que consigue hacer que este algoritmo sea uno de los más usados son la robustez dado que tiene una tasa de detección alta, tasa de verdaderos positivos muy alta y tasa de falsos positivos muy baja.

Con el uso de este algoritmo se pueden realizar reconocimientos faciales en tiempo real para aplicaciones con un mínimo de 2 fotogramas por segundo. Aplica un clasificador en cascada, entrenado mediante un método de Machine Learning, con concreto Adaboost. Los descriptores empleados son las características Haar, muy adecuadas para la detección facial (Jones P. V., 2001.).

5.1.2. CARACTERÍSTICAS HAAR

Las características Haar-Like, son las características utilizadas en la visión artificial para la detección de objetos en imágenes digitales. Estas características son utilizadas por el método de la Viola y Jones (Jones P. V., 2018). Una simple característica rectangular de tipo Haar puede definirse como la diferencia de la suma de píxeles de las áreas dentro del rectángulo, que puede estar en cualquier posición y escala dentro de la imagen original (Barriga, 2017).

Cada una de las características se superponen sobre la imagen en todas las posiciones y tamaños posibles, el valor de la característica se obtiene restando a la suma de todos los píxeles la zona negra de la característica.

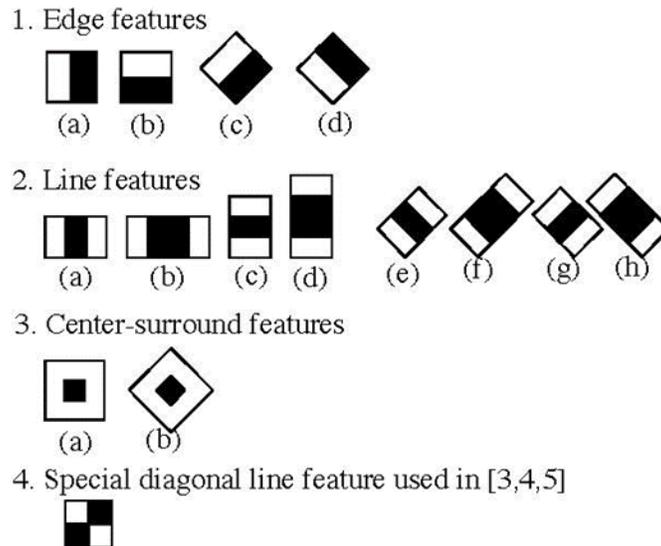


Ilustración 2. Tipo de descriptores utilizados por Haar. (Jones P. V., 2018)

En la Ilustración 2, **(a)** se puede ver un descriptor de dos rectángulos. Con este descriptor se calcula la diferencia entre la suma de los píxeles que se encuentran en el lado negro y el lado blanco de la imagen, estos dos rectángulos tienen el mismo tamaño y forma. En **(b)** se encuentra un descriptor de tres rectángulos, en este caso se calcula la suma de los píxeles entre los dos rectángulos exteriores y se restan la suma de los píxeles que se encuentran en el rectángulo central.

En **(c)** hay un descriptor de cuatro rectángulos, haciendo uso de este descriptor se calcula la diferencia

de los píxeles que se encuentran en los pares diagonales de los rectángulos. Con estas características se obtiene una imagen integral para la posición (x, y) de la imagen original, haciendo uso de este algoritmo se crea la imagen integral con un solo barrido de la imagen original.

La utilidad de esta imagen integral es calcular las características Haar de forma muy eficiente. Una vez se ha obtenido esta imagen integral se hace uso del algoritmo Adaboost de Freund y Schapire, algoritmo utilizado para el aprendizaje de Vila-Jones. Las muestras obtenidas con la parte oscura se tratan de forma independientes a las obtenidas con la parte blanca del descriptor, con lo que se podría utilizar un número diferente de muestras positivas y negativas en el set de entrenamiento.

Como se ha hecho referencia al inicio, en el último paso en el algoritmo de Viola-Jones se hace uso de varios clasificadores en cascada. Esto se realiza para aumentar la velocidad del algoritmo lo que nos proporciona una buena característica para ser usado en tiempo real. Estos clasificadores en cascada van siendo cada vez más estrictos conforme avanza la imagen sobre ellos, es decir, el clasificador dos será más estrictos que el clasificador uno y así sucesivamente. En cada clasificador se segmenta la imagen y se va identificando si hay cara o no. Lo que podemos observar en la siguiente ilustración.



Ilustración 3. Última etapa de la cascada de Haar, donde muchos más patrones de rectángulos en blanco y negro deben coincidir con la imagen del candidato. (Harvey, 2012)

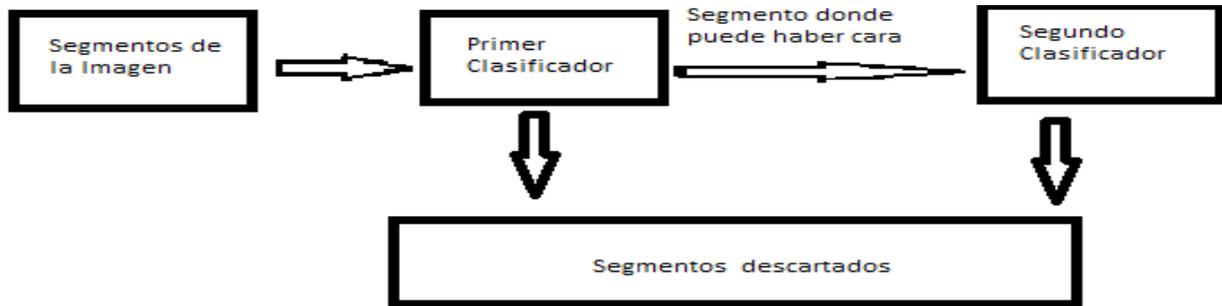


Ilustración 4. Esquema Algoritmo Viola-Jones (Jones P. V., 2001.)

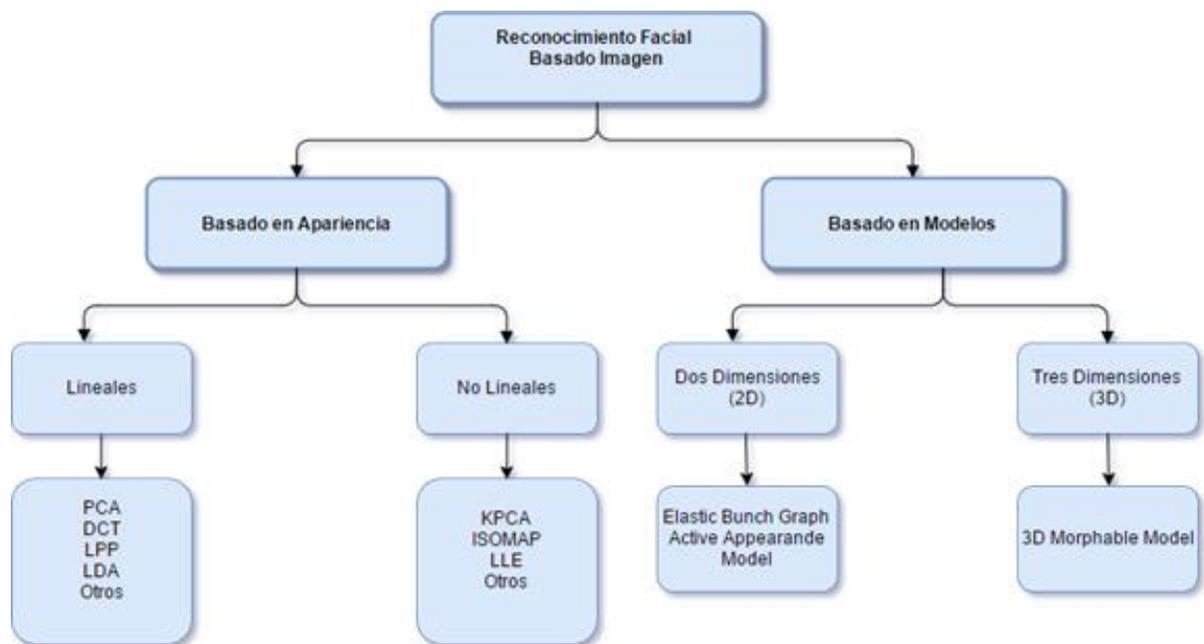


Ilustración 5. Técnicas de Reconocimiento Facial (Davis/Taller, 2018)

5.1.3. ADAPTIVE BOOSTING

Adaptive Boosting (AdaBoost) es un algoritmo de aprendizaje que se utiliza para mejorar el rendimiento de un clasificador simple. clasificador se denomina clasificador débil si no se puede usar solo para predecir la clase a la que pertenece un ejemplo. En general, es computacionalmente económico. Por el contrario, un clasificador se denomina clasificador fuerte si se puede usar solo para clasificar el ejemplo. Es un cálculo intensivo.

Los datos de entrenamiento (una colección de muestras positivas y negativas, es decir, las imágenes con y sin rostro humano) se introducen en un clasificador débil. Después de la primera ronda de aprendizaje, los pesos se normalizan para los ejemplos (imágenes de datos de entrenamiento) para enfatizar en aquellos que fueron clasificados incorrectamente por el clasificador anterior. Este proceso se repite hasta que se obtiene la precisión requerida del clasificador. El resultado final es un clasificador fuerte que es una combinación lineal de un número de clasificadores débiles ponderados seguidos de un umbral (Jain, 2016). En la ilustración 6, se aprecia los clasificadores AdaBoost.

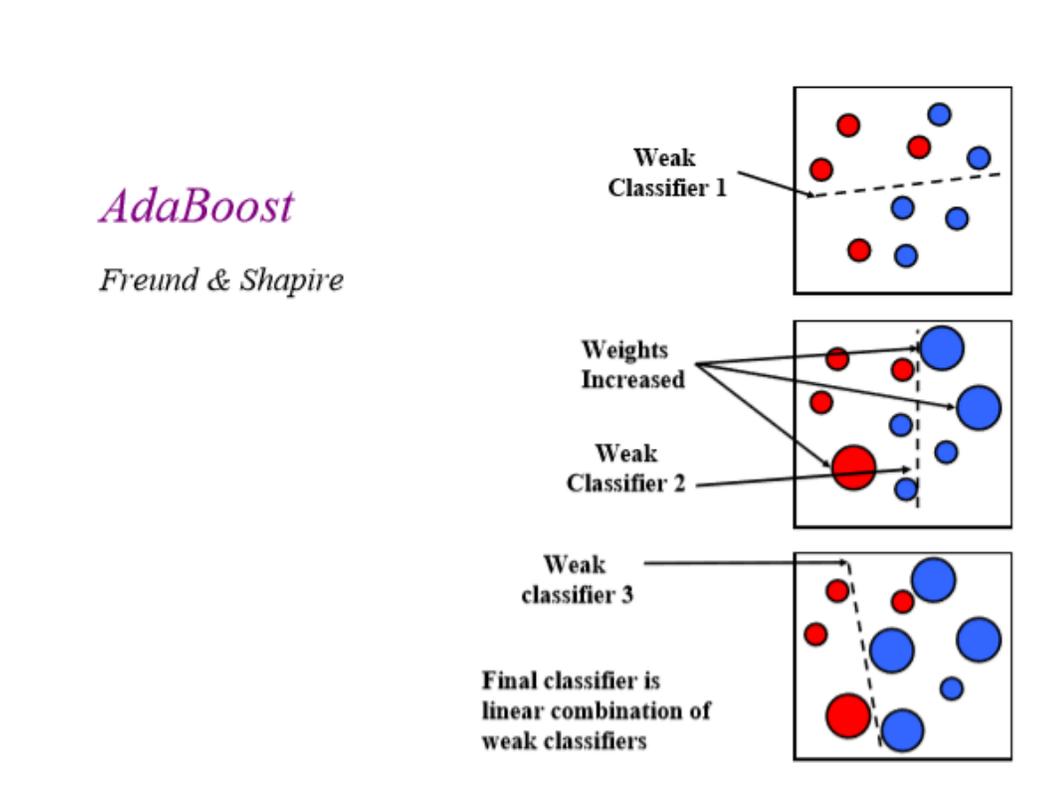


Ilustración 6. Clasificadores Adaboost (Jain, 2016)

5.2. TÉCNICAS DE RECONOCIMIENTO FACIAL

Existen diversas técnicas sobre reconocimiento que se pueden aplicar a esta área específica. Algunas de estas técnicas están basadas en la apariencia que se centran en analizar las características de la imagen como la textura, técnicas basadas en modelos que consisten en extraer las características de forma de las imágenes (Flores L & Cadena Moreano, 2018).

En este proyecto los autores usaron técnicas basadas en la apariencia dado que aportan información más característica que las técnicas basadas en apariencia que suelen utilizarse para realizar modelos en 2D y 3D a partir de imágenes. Las técnicas basadas en la apariencia se clasifican en las técnicas lineales y las técnicas no lineales. Las técnicas lineales consisten en realizar una transformación de las imágenes de entrada a un nuevo subespacio a través de una matriz de proyección. Cada una de las técnicas lineales utiliza una representación propia en un espacio vectorial de alta dimensionalidad. (Flores L & Cadena Moreano, 2018)

Para obtener la representación se utilizan los coeficientes como la representación de cada una de las imágenes. Las transformaciones lineales conseguidas del vector imagen original cumplen con la siguiente característica:

$$y = WT * X$$

Donde “y” es el vector de características de dimensión, “WT” es la matriz de transformación que se utiliza para cada uno de las técnicas y “X” es la imagen de entrada. (Brems, 2017).

5.2.1. PCA (PRINCIPAL COMPONENT ANALYSIS)

PCA es una técnica de detección de tipo lineal que permite reducir la dimensionalidad de un conjunto de datos. Esta técnica consiste en generar un pequeño número de variables encorraladas a partir de un número de variables corraladas (Brems, 2017). Esta técnica consiste en representar una imagen en términos de un sistema de coordenadas óptimo reduciendo el número final de componentes. Con PCA se consigue encontrar los vectores que mejor representan la distribución que existe en un grupo de imágenes.

Para el uso de esta técnica previamente se crea una matriz de proyección. La creación de esta matriz viene a partir de entrenar PCA con N imágenes, estas imágenes deben ser de diferentes personas y a su vez con diferentes ángulos, iluminación y gestos (components, s.f.).

Para calcular la matriz de proyección se realizan los siguientes pasos:

1. Cálculo de la media del vector:

$$\mu = \sum Xi$$

N

$i=1$

2. Estimación de la matriz de covarianza:

$$ST = \sum (Xi - \mu) \cdot (Xi - \mu) r N i=1$$

3. Cálculo de los autos vectores y auto valores de ST y generar W . Donde las columnas de W son los autos vectores correspondientes a los autos valores más significativos.

$$ST \rightarrow W = (e1, e2 \dots eM)$$

En cuanto al reconocimiento, para poder aplicar esta técnica se necesitan realizar las siguientes fases cuando se obtiene una nueva imagen de prueba en el entrenamiento del sistema.

1. La nueva imagen de prueba se redimensiona al mismo tamaño que se ha utilizado para la creación de la matriz de proyección.
2. Esta imagen se normaliza en media 0 y norma 1.
3. Eliminación de la media de la clase PCA por filas.

5.2.2. LDA (LINEAR DISCRIMINANT ANALYSIS)

Es una técnica de aprendizaje supervisado para clasificar datos, la idea central de LDA es obtener una proyección de los datos en un espacio de menor o incluso igual dimensión que los datos ententes con el fin de que la separabilidad de las clases sea la mayor posible. (Flores L & Cadena Moreano, 2018)

LDA pretende convertir un problema de alta dimensionalidad en uno de baja dimensionalidad. Se basa en aumentar la distancia existente entre clases para conseguir una mayor discriminación entre ellas. La gran diferencia entre LDA y PCA es que esta técnica requiere de una supervisión durante el entrenamiento mientras que PCA utiliza un entrenamiento no supervisado.

Para conseguir la maximización entre clases se hace uso de la siguiente expresión:

$$J(w) =$$

$$W^T \cdot S_B \cdot W \quad W^T \cdot S_w \cdot W$$

Donde “ S_B ” es la matriz de dispersión dentro de clases, “ S_w ” es la matriz de dispersión de clase que puede calcularse con las expresiones siguientes:

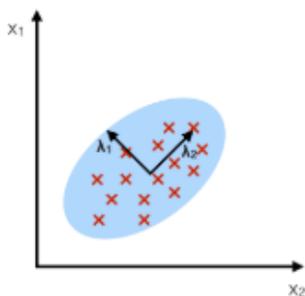
$$S_B = \sum N_C \cdot (\mu_C - x) \cdot (\mu_C - x)^T$$

$$S_w = \sum \sum (X_i - \mu_C) \cdot (X_i - \mu_C)^T \quad i \in C$$

$$\mu_C = \frac{1}{N_C} \cdot \sum X_i \quad i \in C \quad x = \frac{1}{N} \cdot \sum X_i \quad i = 1 \dots N \quad \sum N_C \cdot \mu_C$$

Donde, “ C ” es el número de clases, “ N_C ” el número de casos dentro de la clase “ C ”, “ X_i ” el conjunto de muestras y “ μ_C ” el vector medio de cada clase C . A través de estas ecuaciones se obtiene la matriz de proyección con la que se podrá pasar del problema de alta dimensionalidad al de baja dimensionalidad. Cada columna de esta matriz son las bases del subespacio creado. Con estas bases se consigue una máxima discriminación entre las clases (Raschka, 2014).

PCA:
component axes that maximize the variance



LDA:
maximizing the component axes for class-separation

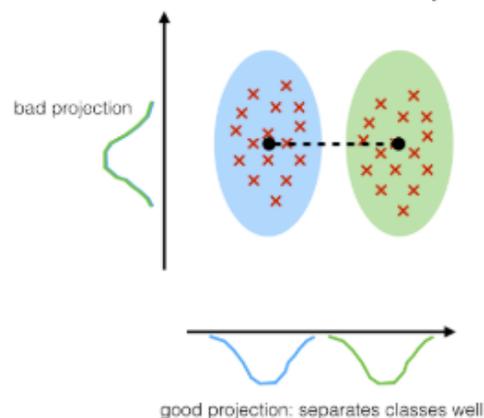


Ilustración 7. Comparación entre PCA, LDA (A.M. Martínez Robot Vision Lab., Purdue Univ., West Lafayette, IN, USA, A.C. Kak, 2001)

5.2.3. EIGENFACES

Eigenfaces es un algoritmo desarrollado por Turk y Pentland. Este método de reconocimiento facial se basa en el análisis por componentes principales. Identifica patrones en las imágenes y clasifica a cada individuo en función de las coordenadas Proyección LDA obtenidas en el sub-espacio que se forma a través de estos componentes (Pentland, 1991) .

Para conseguir esta clasificación previamente se necesita de un set de entrenamiento, es decir, se realiza una colección de diferentes imágenes de caras de personas que harán papel de vectores de observación. Estos vectores vendrán determinados por cada píxel de la imagen.

Para poder utilizar este algoritmo, todas las imágenes han de tener el mismo tamaño. Todas las imágenes deben tener $n = W \times H$ píxeles, donde W es el ancho de la imagen y H la altura. Por lo tanto, cada imagen es representada como un punto en un subespacio vectorial de n dimensiones. Los desarrolladores de este algoritmo utilizaron imágenes de 256×256 píxeles, por lo que las imágenes estaban contenidas en un subespacio vectorial de 65536 dimensiones.

El primer paso para realizar es el cálculo de la matriz de covariancia S_X :

$$S_X = \frac{1}{M} \sum_{k=1}^M (x_k - \mu)(x_k - \mu)^T$$

$$\sum_{k=1}^M (x_k - \mu)(x_k - \mu)^T = \frac{1}{M} A^T A$$

Donde, “ M ” es el número de imágenes de entrenamiento a utilizar y “ X ” son los vectores formados con los valores de los píxeles de cada imagen. El principal problema de esta formulación es que resulta una matriz $n \times n$ por lo que se obtienen demasiados valores para realizar el cálculo de los auto vectores. Por ello se utiliza el cálculo de la matriz $A^T A$ que tiene unas dimensiones $M \times M$. Por lo tanto, es necesario tener una relación de estos autos vectores y autovalores con la matriz S_X . Se hace uso de la siguiente definición donde los autos vectores de $A^T A$, μ_i , cumplen la siguiente ecuación:

$$\frac{1}{M} A^T A \mu_i = \sigma_i \mu_i$$

Donde σ_i es el autovalor correspondiente al auto vector μ_i . A partir de esta ecuación se multiplica a la ecuación por la matriz A :

$$\frac{1}{M} A^T A \mu_i = \sigma_i \mu_i \Rightarrow A \mu_i = \sqrt{\sigma_i} \nu_i$$

Con esto, llegamos a que el conjunto de auto vectores utilizados para Eigenfaces viene por la matriz P:

$$P_{n \times M} = A_{n \times M} [\mu_1 \mu_2 \dots \mu_n]$$

El conjunto de estos auto vectores resultante se denominan Eigenfaces. Una vez se han calculado los Eigenfaces, se procede a reducir la dimensionalidad de los datos. Para poder realizar este paso, se desechan los Eigenfaces que tienen autovalores más bajos haciendo uso sólo de los que corresponden a los valores más altos.

Durante el entrenamiento con Eigenfaces para cada imagen patrón de cada persona se guardan sus componentes en el subespacio formado por Eigenfaces. Una vez terminado con la parte de entrenamiento, cuando se quiere realizar un reconocimiento, la imagen que entra se proyecta en el subespacio descrito anteriormente y se usan las coordenadas de esta proyección para tomar una decisión con los patrones que existen guardados. Para poder realizar este reconocimiento se realizan los siguientes pasos:

1. Normalización de la imagen de entrada y proyección del resultado en el subespacio de Eigenfaces obtenido en el entrenamiento.

$$y' = p^T * (x' - \mu) = [y_1' \dots y_m']$$

Donde m es el número de Eigenfaces utilizadas e y' son las coordenadas o descriptores de la imagen de entrada.

2. Cálculo de la distancia Euclídea d entre la imagen de entrada y cada patrón almacenado

$$d = \|x' - y'\|$$

3. Por último, si esta distancia Euclídea es inferior a un umbral establecido previamente, se decidirá que la imagen de entrada es el individuo correspondiente a ese patrón. Por el contrario, si no cumple esta restricción, se decidirá que la imagen de entrada no pertenece a la persona existente en el sistema.

$$h(x') = \begin{cases} 1 & \text{Si } d < \text{Umbral} \\ 0 & \text{Si } d > \text{Umbral} \end{cases}$$

5.2.4. FISHESFACES

Como se ha explicado anteriormente, existe otro método alternativo de reconocimiento de personas llamado Fisherfaces. Este sistema en diferencia, al basado en Eigenfaces, utiliza LDA como técnica para el reconocimiento facial. Con este sistema se pretende maximizar la distancia entre clases y minimizar la distancia dentro de la misma clase (Belhumeur & Hespanha, 1997).

Este algoritmo fue propuesto por Belhumeur, que consiste en usar LDA para reducir la dimensión de los datos a $N - C$, donde N es el número de imágenes y C es el número de individuos que se encuentran en el set de entrenamiento, también llamado clases. La técnica LDA reduce la dimensionalidad hasta $C - 1$, que será el número de auto vectores obtenidos distintos de cero.

Para aplicar Fisherfaces se hace uso de la matriz P formado por los vectores columna:

$$P = PPCA * PLDA \rightarrow PT = PLDA T * PPCA T$$

Igual que en Eigenfaces se realiza una etapa de entrenamiento donde se almacenan los patrones de las personas que se deseen clasificar. El procedimiento es el mismo que en caso anterior donde se proyecta la imagen de entrada al subespacio formado por las Fisherfaces. Hace uso de la siguiente ecuación para calcular las coordenadas x e y de la imagen de entrada en el subespacio de Fisherfaces:

$$y = PT * x = PLDA T * PPCA T * x$$

Necesita conocer previamente el etiquetado de cada imagen del set de entrenamiento. Esa es la diferencia fundamental respecto al método de Eigenfaces. Como se ha mencionado anteriormente, una vez se le pasa esta información al algoritmo procede a la extracción de las características Fisherfaces de cada clase. Como se observa en la ecuación siguiente, Fisherfaces utiliza tanto la técnica PCA para reducir la dimensión de espacio de características y después se aplica LDA. Es decir, se proyecta la imagen de forma óptima a partir de:

$$Wopt T = Wlda T * Wpca T$$

Cuando se han obtenido los pesos de Fisherfaces de cada clase y estos son proyectados en el subespacio se emplean para concluir si la imagen de entrada pertenece a una de las clases existentes en la base de datos. Se puede hacer uso de varios sistemas de decisión que calculen la distancia entre la imagen de

entrada y a cada uno de los patrones de la base de datos. Este método de decisión se tratará en el siguiente apartado.

De forma sintética un algoritmo propio de Fisherfaces seguirá las siguientes características:

1. Construcción de la matriz de proyección con los vectores propios SW y SB correspondientes a los valores no nulos.
2. Reducción de dimensionalidad de los vectores de características.
3. Construcción de un vector de características proyectando la matriz anterior por cada una de las clases.
4. Clasificación por distancia a los vectores de características.

(Face Detection and Recognition Theory and Practice eBookslib, 2016)

5.2.5. APPLE FACEID

Podemos observar varios métodos de implementar lo que es la visión por computadora y también hemos visto varias formas de generar una imagen y al mismo tiempo poder detectar los elementos de este mediante varios tipos de algoritmos. Además de los avances que se han dado en la última década con respecto a los algoritmos también tenemos otras formas de tecnologías que están facilitando el desarrollo de lo que es el reconocimiento facial utilizando la visión por computadoras.

Apple presentó FaceID con el iPhone X, una nueva forma de proteger el teléfono. Esto es un claro ejemplo del avance en hardware ya que esta tecnología es diferente y permite reducir en 100% el margen de error, utiliza reconocimiento en 3D además de utilizar los rasgos faciales tiene acceso a sensores infrarrojos que le permite reducir el margen de error al efectuar lo que es el reconocimiento facial. (support.apple.com, 2020)

El Face ID es una forma de autenticación biométrica es usado para desbloquear el teléfono mediante el uso de escáneres de luz infrarroja y visible para identificar de forma única su rostro. Funciona en una variedad de condiciones y es extremadamente seguro. Lo cual marca un récord y favorece la implementación del reconocimiento facial. (Fleishman, 2017)

En lugar de una contraseña (algo que ya sabes) o un dongle de seguridad o una aplicación de autenticación (algo que tienes), la biométrica es algo que eres. El reconocimiento de las huellas dactilares también es biométrico.

En lugar de una o más huellas dactilares, como con Touch ID, Face ID se basa en las características únicas de su rostro. Apple está apostando a que su tecnología puede superar seis obstáculos:

1. Inicialmente escanea tu cara con la precisión suficiente para reconocerla más tarde.
2. Compare un nuevo escaneo con el almacenado con suficiente flexibilidad para reconocerlo casi todo el tiempo.
3. Escanea tu rostro en una amplia variedad de condiciones de iluminación.
4. Actualice sus detalles faciales a medida que envejece, cambie los peinados, haga crecer un bigote, cambie las cejas, realice una cirugía plástica y demás para que aún lo reconozca.
5. Te permite usar sombreros, bufandas, guantes, lentes de contacto y lentes de sol, y aun así ser reconocido.
6. No permita que una persona de aspecto similar, una fotografía, una máscara u otras técnicas desbloqueen su teléfono.

5.2.6. DEEPFACE

Facebook tiene un proyecto de investigación de reconocimiento facial llamado DeepFace. DeepFace, ahora es casi tan preciso como el cerebro humano. DeepFace puede mirar dos fotos, e independientemente de la iluminación o el ángulo, puede decir con una precisión del 97.35% si las fotos contienen la misma cara. El software DeepFace, desarrollado por el grupo de investigación de IA de Facebook en Menlo Park, California, está respaldado por una avanzada red neuronal de aprendizaje profundo. Una red neuronal es un software que simula una aproximación (muy básica) de cómo funcionan las neuronas reales. (Desai, 2016)

El aprendizaje profundo es uno de los muchos métodos para realizar el aprendizaje automático; Básicamente, observa un gran cuerpo de datos (por ejemplo, rostros humanos) e intenta desarrollar una abstracción de alto nivel (de un rostro humano) buscando patrones recurrentes (mejillas, cejas, etc.).

En este caso, DeepFace consta de un grupo de neuronas de nueve capas de profundidad, y luego un proceso de aprendizaje que ve la creación de 120 millones de conexiones (sinapsis) entre esas neuronas, en base a un corpus de cuatro millones de fotos de caras. Una vez que se completa el proceso de aprendizaje, cada imagen que se alimenta al sistema pasa a través de las sinapsis de una manera diferente, produciendo una huella digital única en la parte inferior de las nueve capas de neuronas. Por ejemplo, una neurona podría simplemente preguntar "¿tiene la cara una frente gruesa?": Si es así, se sigue una sinapsis, si no, se toma otra ruta (Vincent, 2015).

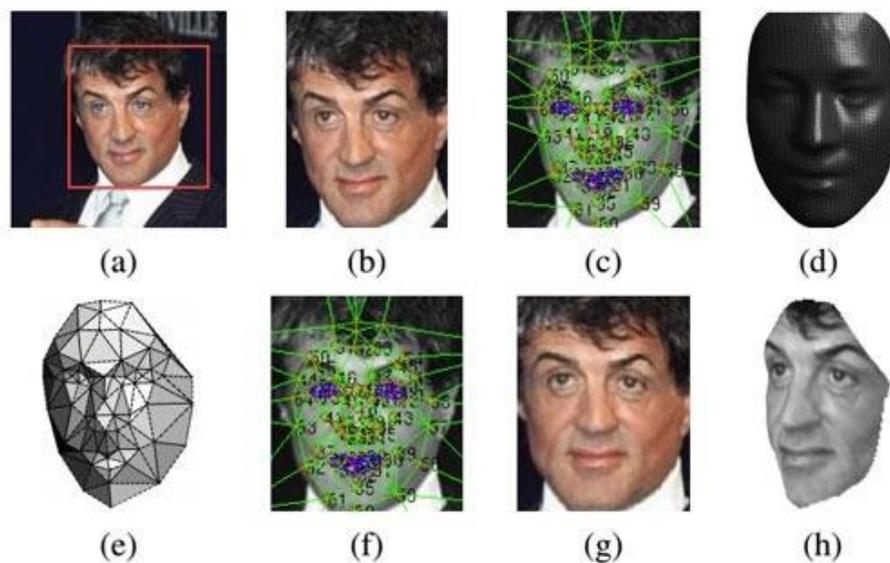


Ilustración 8. DeepFace Recognition (Rigg, 2014)

en la ilustración 8 se ve el rostro de Sylvester Stallone que está siendo procesada con algoritmo de Forward-Face, observe los cambios ligeros en el ángulo de la imagen (a) están corregidos en la imagen (g). (d) es el estándar que el algoritmo está ocupando para posicionar las imágenes, Forward-Looking usa los rostros para la transformación, ignore h) ya que esta no tiene relación.

5.3. ANÁLISIS DE ESTUDIOS

En este acápite se presenta el análisis de tres estudios relacionados al proyecto de investigación (*SDRF-BICU*), en donde se aborda las características principales y los resultados de cada uno de los estudios en mención.

5.3.1. Sistema informático de reconocimiento facial para el registro y control de asistencia de los socios de la cooperativa de taxis y camionetas puyo.

Este proyecto fue desarrollado en el año 2019 en la Universidad Regional Autónoma De Los Andes (UNIANDES), Puyo, Ecuador. Con la finalidad de desarrollar un sistema informático de reconocimiento facial para el registro y control de asistencia de los socios de la cooperativa de Taxis y Camionetas Puyo, que permita llevar un control adecuado del registro de las asistencias de los socios a reuniones y convocatorias dentro de la sede de la Cooperativa (Fernando, 2019).

La metodología utilizada en este proyecto es el modelo en cascada, consiste en un proceso de desarrollo secuencial, en donde se concibe como un conjunto de etapas que se ejecutan una tras otra. Por otro lado, los autores adoptaron el algoritmo de Viola-Jones, por la alta capacidad de detección de objetos y rapidez de procesamiento de imagen, la librería OpenCV, una biblioteca de visión artificial, análisis de imagen y aprendizaje automático, integrado con Python facilitaron la codificación del sistema de reconocimiento facial ya que ambas tecnologías se integran de manera sencilla y SQL server para el gestor de base de datos.

Como resultado obtuvieron un sistema de reconocimiento facial para el registro y control de asistencia de socios de la cooperativa de Taxis y Camionetas Puyo, logrando convertirse en una herramienta para obtener datos confiables de los socios en las reuniones y facilitando el trabajo manual al personal administrativo de la institución.

5.3.2. Sistema de Notificación de Agenda a través de Reconocimiento Facial: Prototipo para Docentes a Tiempo Completo de la Carrera de Computación de la Universidad Católica de Santiago de Guayaquil.

El segundo trabajo para tomar en cuenta fue desarrollado en la Universidad Católica De Santiago De Guayaquil, Guayaquil, Ecuador. Tuvo la finalidad de diseñar e implementar un prototipo de plataforma informática basada en visión por computador y para notificaciones de agenda a los docentes de tiempo completo de la Facultad de Ingeniería, Carrera de Computación de la UCSG (Suquinagua León, 2019).

Se implementó la metodología basada en entregables o componentes, fundamentada en la reutilización y se compone de una gran base de componentes de software que son reutilizable, aplicando un proceso de experimento hacia la teoría de prueba y error.

Por otro lado, utilizaron la librería OpenCv para la detección de rostro con la versión escrito en JavaScript para navegación dado que el proyecto es un sistema Web, el modelo de red neuronal convolucional llamado “Redes Siamesas” para la verificación de identidad de dicho rostro, Laravle, framework de código abierto para desarrollar aplicaciones y servicios web con PHP, asimismo Tenserflow, biblioteca de código abierto dirigida al aprendizaje automático creado por Google escrito en Python y Keras, una API de redes neuronales de alto nivel, escrita en Python es fácil de usar, modular y extensible. Como resultado los autores lograron mejorar el control de actividades agendadas para el desarrollo de sus clases.

5.3.3. Análisis y Diseño de un Sistema de Reconocimiento Facial aplicando Machine Learning para detectar e identificar intrusos

Este estudio fue desarrollado en la Universidad De Guayaquil, Guayaquil, Ecuador. La finalidad del proyecto es mantener un mejor control en el acceso de las personas al interior del edificio de las Carrera de Ingeniería en Sistemas Computacionales e Ingeniería en Networking y Telecomunicaciones de la Universidad de Guayaquil (Javier, 2018).

Adoptando como metodología, la metodología de Prototipo, basada en lo re-alimentable o evolutivo y permite al desarrollador regresar a la versión anterior del programa y realizar modificaciones para mejorarlo y no utilizar muchos recursos. Esta metodología contiene las siguientes fases: Planeación, Modelado, Elaboración del Prototipo, Desarrollo, Entrega y Re-alimentación, y por último la entrega del producto final.

Para el desarrollo del sistema usaron el algoritmo método comparativo, PhpMyAdmin para gestor de base de datos (XAMPP), Cmake el cual es necesario para que pueda funcionar la biblioteca face_recognition, Yii Framework basado en PHP, Python y la librería OpenCV.

5.4. SELECCIÓN DE TECNOLOGÍAS Y METODOLOGÍAS DE DESARROLLO

Después de haber realizado un análisis exhaustivo de las tecnologías, algoritmos y metodologías de desarrollo propuestos por otros autores para el desarrollo de proyectos del dominio en cuestión, se decidió que la metodología de prototipo es lo más adecuado para implementar en nuestra investigación, debido a la naturaleza de este proyecto que acoge el uso de una tecnología emergente, misma que se encuentra en desarrollo constante, en cuanto a la precisión y compatibilidad con otros sistemas operativos y lenguajes de programación.

Por otro lado, también se considera el uso de Python debido a sus características de compatibilidad con múltiples librerías de reconocimiento facial, además es fácil y simple de utilizar, minimiza tiempo de desarrollo e implementación. OpenCV es una Librería que puede ser utilizada con Python la cual permite poder hacer uso de los algoritmo de Viola-Jones, el cual fue el primer algoritmo funcional de reconocimiento facial, y es la base de la mayoría de los sistemas modernos de reconocimiento facial.

Para el Gestor de la Base de datos se decidió por MySQL y el desarrollo de la interfaz de usuario Tkinter. Se uso el algoritmo de Viola-Jones y Keras (Tensorflow) como API (Aplicación Programming Interface). Que será utilizado para entrenar el modelo de Deep learning, ya que es rápido para ejecutar búsquedas y crear modelos de prototipos y búsquedas avanzadas, es modular y permite configurar y conectar bloques múltiples de códigos, con pocas restricciones, cuenta con la capacidad de extender o expresar ideas nuevas de investigación. Al implementar la red neuronal siamesa con Keras esto minimizará drásticamente el tiempo de entrenamiento y ejecución de nuestro sistema.

VI. HERRAMIENTAS Y TECNOLOGÍAS USADAS

En esta sección se presenta la conceptualización de las herramientas y tecnologías usadas en el desarrollo de *SDRF-BICU*.

6.1. Local binary patterns histograms (LBPH) Face Recognizer

El algoritmo de histograma de patrones binarios locales se propuso en 2006. Se basa en el operador binario local. Es ampliamente utilizado en el reconocimiento facial debido a su simplicidad computacional y poder discriminativo. El algoritmo LBPH es parte de OpenCv. (Open Genus, n.d.)

6.2. Python versión 3.5

Python es un lenguaje de programación de alto nivel diseñado para ser fácil de leer y sencillo de implementar. Es de código abierto, lo que significa que es de uso gratuito, incluso para aplicaciones comerciales. Python puede ejecutarse en sistemas Mac, Windows y Unix y también ha sido portado a máquinas virtuales Java y .NET. (Techterms, 2010)

6.3. OpenCv 4.0

OpenCV (Open Source Computer Vision) es una biblioteca de funciones de programación dirigida principalmente a la visión por computadora en tiempo real. En lenguaje simple, es una biblioteca utilizada para el procesamiento de imágenes. Se utiliza principalmente para hacer todas las operaciones relacionadas con las imágenes. (Galgalin, 2016)

6.4. Numpy

Numpy es el paquete fundamental para la computación científica en Python. Es una biblioteca de Python que proporciona un objeto de matriz multidimensional, varios objetos derivados (como matrices y matrices enmascaradas) y una variedad de rutinas para operaciones rápidas en matrices, incluidas las matemáticas, lógica, manipulación de formas, clasificación, selección, E / S, transformadas discretas de Fourier, álgebra lineal básica, operaciones estadísticas básicas, simulación aleatoria y mucho más. (Numpy.org, 2020)

6.5. Pandas

Pandas es una librería de Python destinada al análisis de datos, que proporciona unas estructuras de datos flexibles y que permiten trabajar con ellos de forma muy eficiente. (Jarroba.com, 2015)

6.6. Tkinter

Es una librería que proporciona a las aplicaciones de Python una interfaz de usuario fácil de programar. Además, es un conjunto de herramientas GUI de Tcl/Tk (Tcl: Tool Command Language), proporcionando una amplia gama de usos, incluyendo aplicaciones web, de escritorio, redes, administración, pruebas y muchos más. (Python, n.d.)

6.7. MySQL

MySQL es un sistema de gestión de bases de datos relacionales de código abierto (RDBMS, por sus siglas en inglés, Sistema de Gestión de Bases de Datos Relacionales) con un modelo cliente-servidor. RDBMS es un software o servicio utilizado para crear y administrar bases de datos basadas en un modelo relacional. (B., 2019)

6.8. ReportLab

ReportLab es el motor de código abierto, ultra robusto y de gran duración para crear documentos PDF complejos y basados en datos y gráficos vectoriales personalizados. Es gratis, de código abierto y escrito en Python. (Salcedo, 2017)

6.9. WebCam V-U0026

Una cámara web o cámara de red (en inglés: webcam) es una pequeña cámara digital conectada a una computadora la cual puede capturar imágenes y transmitir las a través de Internet, ya sea a una página web u otras computadoras de forma privada. (Wikipedia, 2012)

VI. MODELO DE DESARROLLO

Para el desarrollo de este proyecto se empleó el modelo de desarrollo conocido como "Desarrollo de Prototipo", por lo cual se llevó a cabo paso a paso las fases que comprende este modelo. Se escogió este modelo porque es re-alimentable y permite al desarrollador regresar a la versión anterior del programa y realizar modificaciones para mejorarlo. Con esto se puede añadir nuevas características a la implementación que se está llevando a cabo. A continuación, se describe cada una de las etapas de desarrollo del modelo:

Fase 1: Planeación, En esta fase se hace la planeación de actividades que se debe realizar para empezar el proyecto.

Fase 2: Modelado, Es donde se realiza el diseño preliminar del sistema, aunque no es el diseño final, también se define el funcionamiento del sistema, que se debe hacer y cómo debe hacerlo.

Fase 3: Elaboración del Prototipo, Se comienza la elaboración del prototipo del sistemas, el base de conocimiento, los principales usuarios y su permiso o accesibilidad, además la experimentación con los lenguajes de programación para definir lo adecuado dependiendo a la compatibilidad.

Fase 4: Desarrollo, Entrega y Re-alimentación, Se realiza el diseño final del sistema, mejorado el sistema por retroalimentación, el seguimiento se da en tres partes que son:

- **Módulo de Almacenamiento:** Es donde se hacen el desarrollo del proceso de almacenamiento de datos. en este contexto la base de conocimientos del sistema.
- **Módulo de Procesamiento:** Aquí se realizó la conexión entre el sistema con el proceso de almacenamiento, también es donde se realizó toda la operación de reconocimiento facial con machine learning.
- **Módulo de Presentación:** Este módulo es la interfaz que utilizará el usuario final.

Fase 5: Entrega del producto final, En esta fase es la presentación del sistema tomando en cuenta que el cliente está satisfecho y cumple con todos los objetivos y también se entrega un manual.

Cabe mencionar que en este acápite únicamente se presenta una descripción general de modelo de desarrollo. A continuación, se presentara la climatización del modelo, como se realizó en cada una de las etapas del desarrollo *SDRF-BICU*.

VII. DISEÑO METODOLÓGICO

Como se mencionó anteriormente, el modelo adoptado fue el de prototipo. El cual consta de cinco fases descrita en el acápite anterior, aquí se presentará detalladamente lo realizado en cada una de ellas en base a los requerimientos exigidos en este proyecto.

7.1. Planeación

En esta etapa se realizó la planeación de actividades, como su nombre indica, se representa una tabla de actividades de cada fase que se realizó en el procesamiento de este proyecto en base al modelo prototipo. En la tabla 1. se muestra la tabla de actividades y el periodo de ejecución

PLANEACION DE ACTIVIDADES													
DEL MODELO PROTOTIPO DEL AÑOS 2019													
<u>Etapas</u>		<u>Descripción</u>	<u>Meses del año 2019</u>										
			<u>fe</u> <u>b</u>	<u>mar</u>	<u>apr</u>	<u>m</u> <u>av</u>	<u>jun</u>	<u>jul</u>	<u>agt</u>	<u>sep</u>	<u>oct</u>	<u>no</u> <u>v</u>	<u>de</u> <u>c</u>
1	Planeación	Plantear actividad, Observación y entrevista											
2	Modelado	Diseño preliminar y funcionamiento del sistema											
3	Elaboración del Prototipo	Elaboración del sistema, definir permiso, accesibilidad de usuarios y experimento con lenguaje de programación											
4	Desarrollo, Entrega y Re-	Modelo de almacenamiento Diseño general del base de datos											

	alimentación	Modelo de procesamiento	Conexión entre el modelo de almacenamiento con la presentación mediante código											
		Modelo de presentación	Interfaz del diseño final											
5	Entrega del producto final		Defensa											

Tabla 1. Planeación de Actividades 2019 (Elaborado por Lorvick y Yelsy)

Una vez realizado el plan de trabajo, se procedió a determinar los requerimientos necesarios para la construcción del sistema, luego se pasa a la fase dos del modelo de desarrollo.

7.2. Modelado

En esta fase se realizó el diseño preliminar del sistema, para sentar una base sobre la cual se realizará el diseño final del sistema.

Dentro de los diseños preliminares que se realizaron durante esta fase se puede mencionar el modelo relacional y modelo de entidad – relación. Estos diagramas fueron indispensable para la creación de la base de datos que se utilizará para guardar los registros de cada persona. También se decidió almacenar las fotos de los trabajadores en un directorio en el equipo, debido a que registrará 150 fotos de cada personal, grabar tantas imágenes en MySQL se harán más pesado y lento el proceso de reconocimiento facial. Por otro lado, cada imagen se convertirá en array (números) lo cual es escáner del rostro que se procesa y guarda en un archivo tipo “YML”.

El modelo relacional presenta la relación entre las tablas que contiene la base de datos, muestra la interacción, conexión y el flujo de datos de cada tabla, brinda una visión unificada de los datos y el entendimiento de la función del sistema. A continuación, se muestra en la ilustración 9 el modelo relacional.

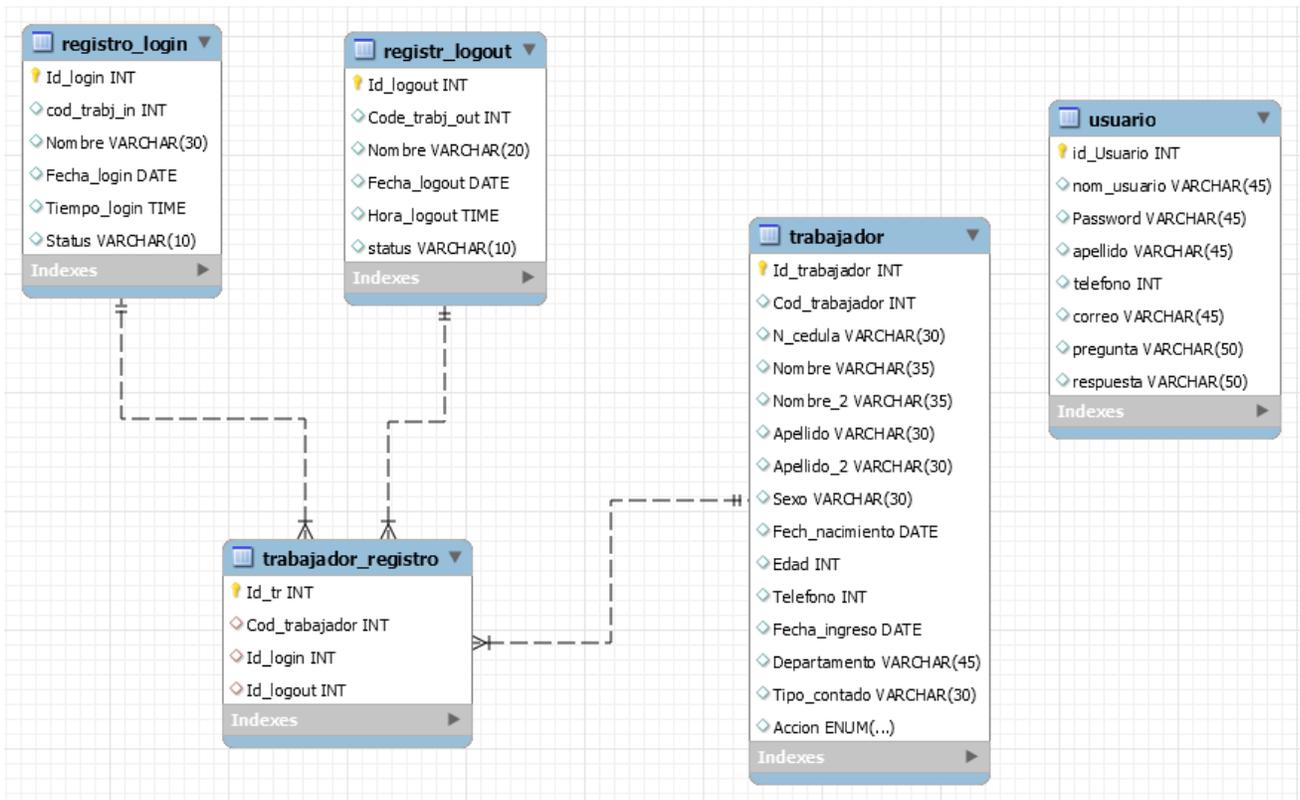


Ilustración 9. Diagrama de modelo entidad (Elaborado por: Lorvick y Yelsy)

El diagrama entidad-relación, representa la relación entre entidades con sus respectivos atributos, se determina cual será los datos que se manejan en el sistema. Las entidades de contenido en el diagrama tienen el siguiente nombre: trabajador, registro_trabajador, registro_login, registro_logout, usuario,

cada una de ellas guarda información importante para la correcta funcionalidad del sistema. Lo puede observar en la figura 10.

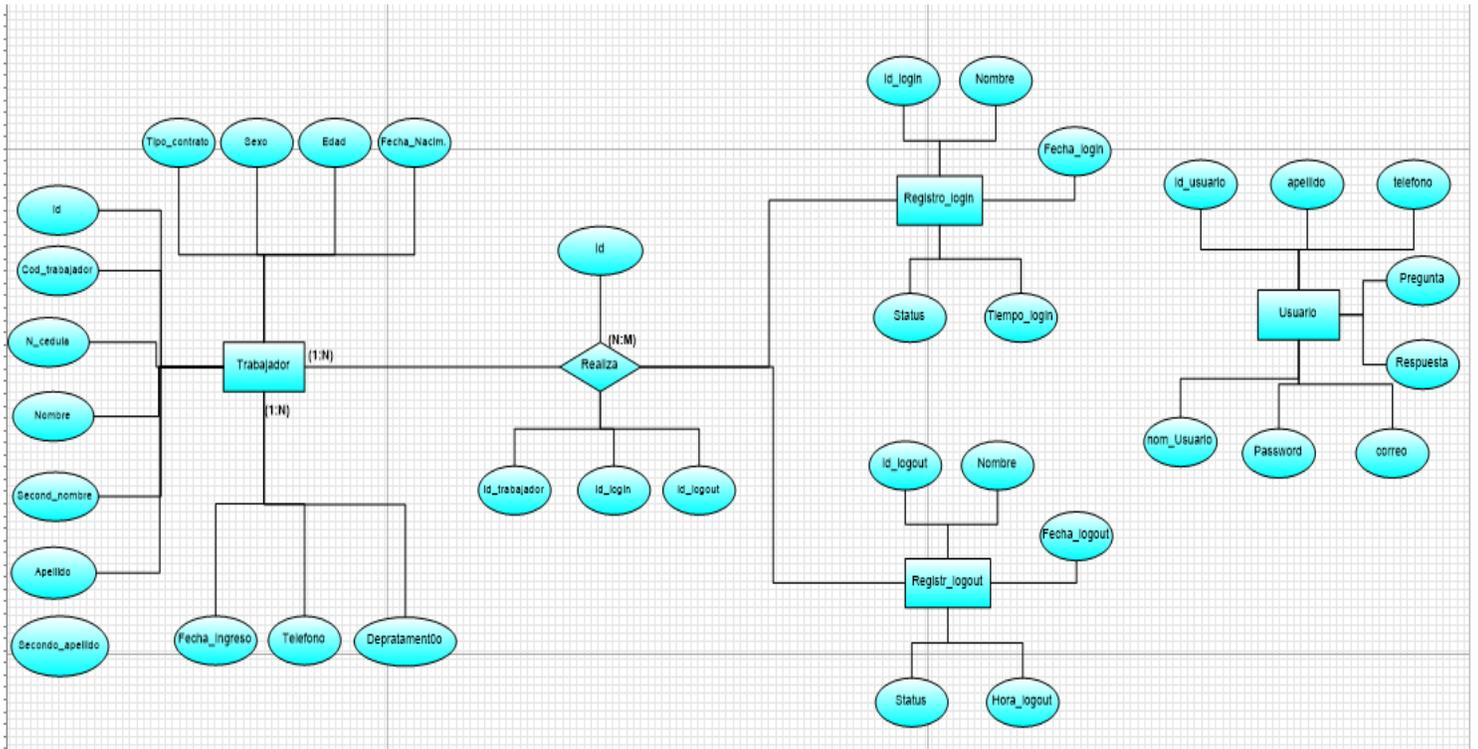


Ilustración 10. Modelo de Entidad Relacional (Elaborado por: Lorvick y Yelsy)

La entidad trabajadora está compuesta de 14 atributos que son: *Id_trabajador*, Código de trabajador, Nombre, Secundo nombre, Apellido, Secundo apellido, *N_cedula*, , tipo de contrato, Sexo, edad, Fecha de ingreso ,Fecha de nacimiento Teléfono, acción y departamento. Cuenta con la llave primaria *Id_trabajador* y también una llave única que es el código de trabajado. Así como se muestra en la figura 12.

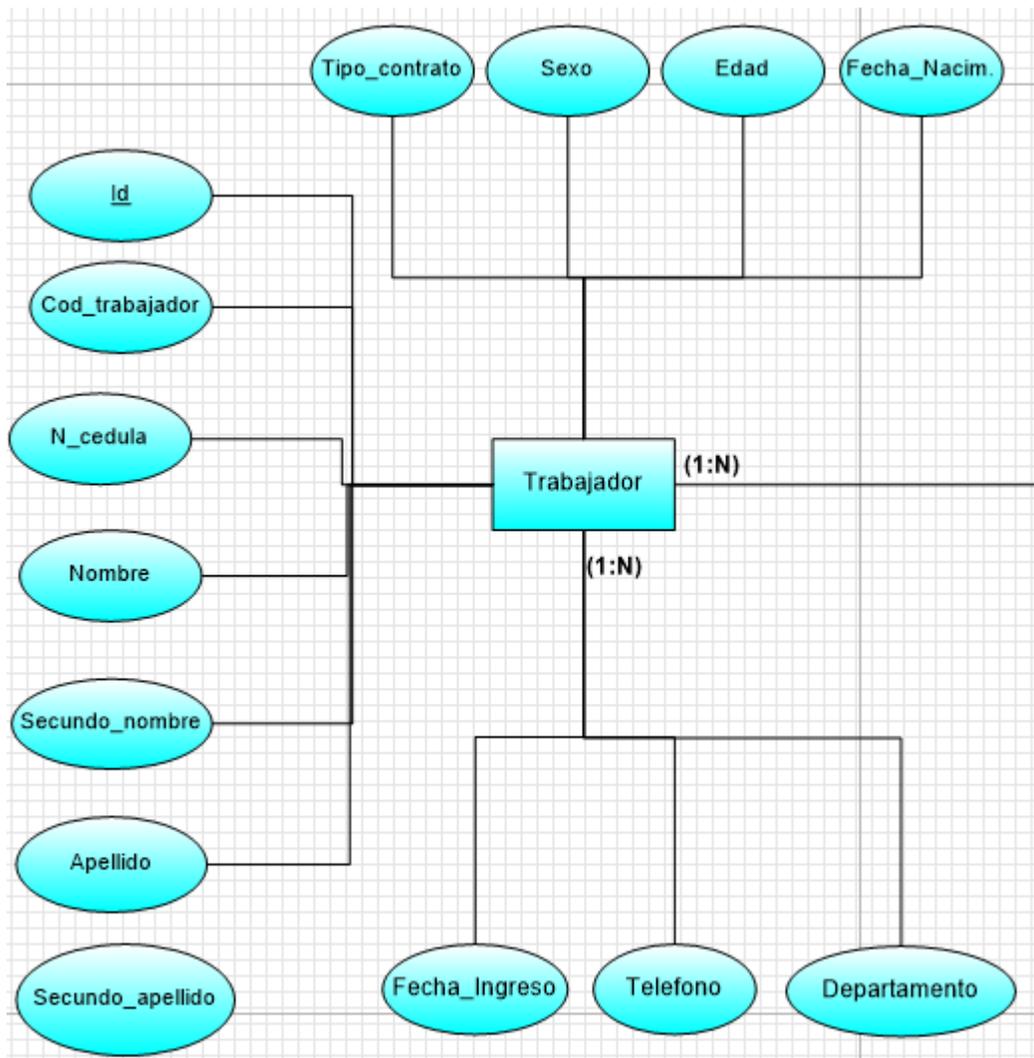


Ilustración 82. Entidad trabajador con su respectivo atributo (Elaborado por: Lorvick y Yelsy)

En la ilustración 12, se muestra la relación que es entre la entidad trabajador, registro de login y registro de logout, contiene los siguientes atributos: tres llaves foránea, Id_trabajador del trabajador, id registro de login e id registro de logout e Id de la relación.

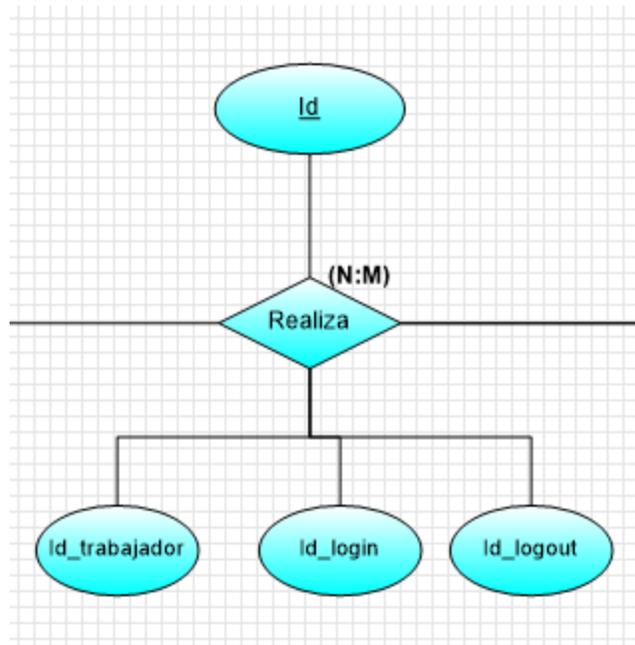


Ilustración 93. La relación entre trabajador, registro de login y registro logout (Elaborado por: Lorvick y Yelsy)

La entidad registro login, contiene 5 atributos que son: Id login , nombre, fecha de login, tiempo de login , status. Cuenta con una llave primaria que es Id login, nota que esta entidad gradara o maneja solo las datas de registro de entrada, que sería dos entrada por persona al día . Así como se visualiza en la figura 13.

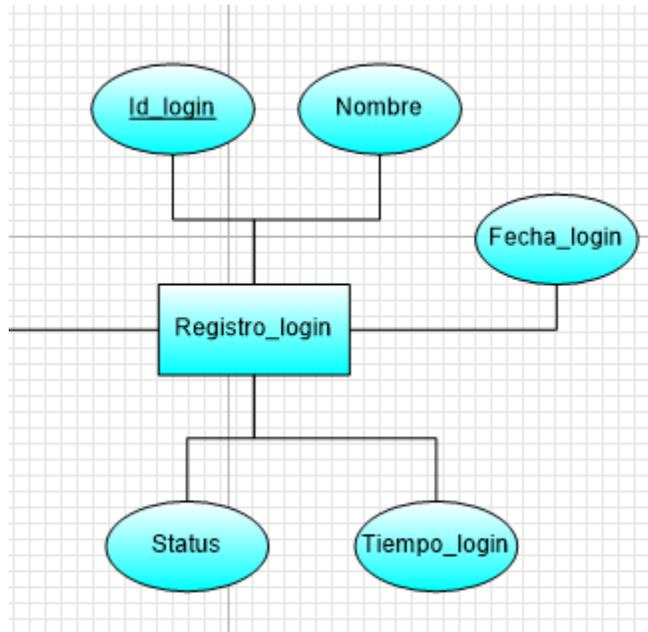


Ilustración 104. Entidad registro de login con su respectivo atributo (Elaborado por Lorvick y Yelsy)

La entidad registro logout, se contiene 5 atributos que son: Id logout, nombre, fecha de logout, hora de login, status. Cuenta con una llave primaria que es Id logout, nota que esta entidad gradara solo las datas de registro de salida, que sería dos salidas por persona al día. Así como se visualiza en la figura 14.

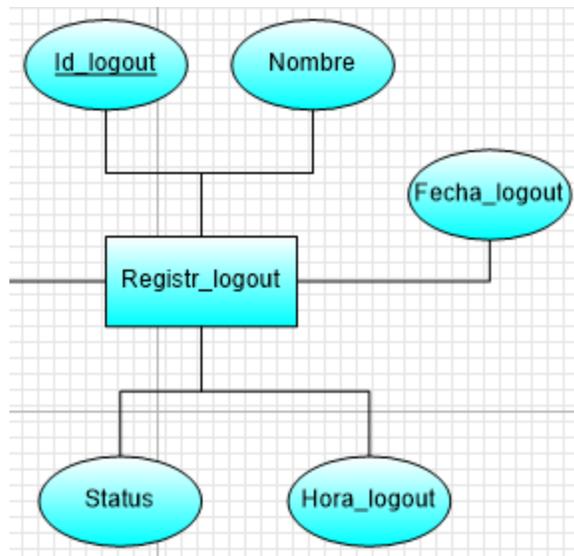


Ilustración 115. Entidad Registro logout con su respectivo atributo (Elaborado por: Lorvick y Yelsy)

La entidad usuario se almacena los datos de acceso para el parte administrativo del sistema, contiene los siguientes atributos, id usuario, nombre de usuario, password, apellido, teléfono, correo, pregunta, respuesta. Contiene una llave primaria id usuario. Como se visualiza en la ilustración 16.

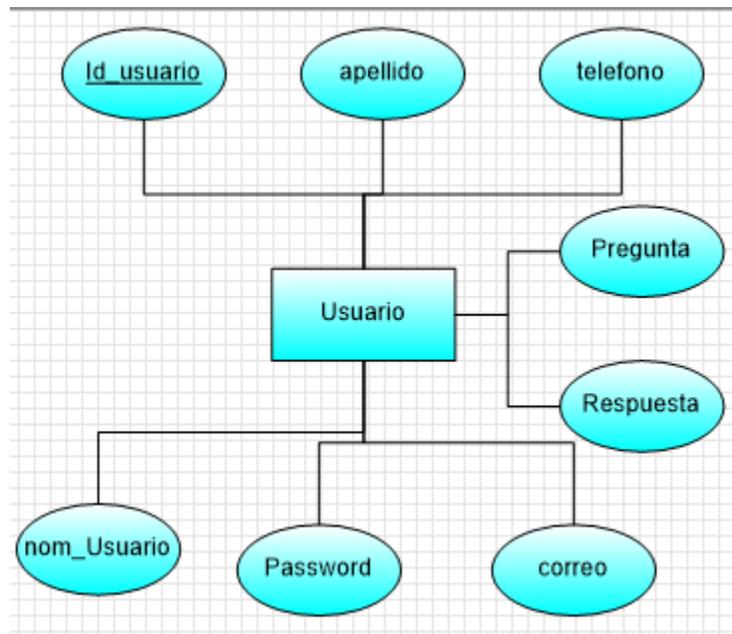


Ilustración 1612. Entidad de usuario (Elaborado por: Lorvick y Yelsy)

Las fotos de los trabajadores se almacenaron en un directorio en el equipo, debido a que se registrará 150 fotos de cada empleado, cada imagen se convertirá en array y se guarda en un archivo tipo “YML”. En la figura 17 se aprecia el directorio.

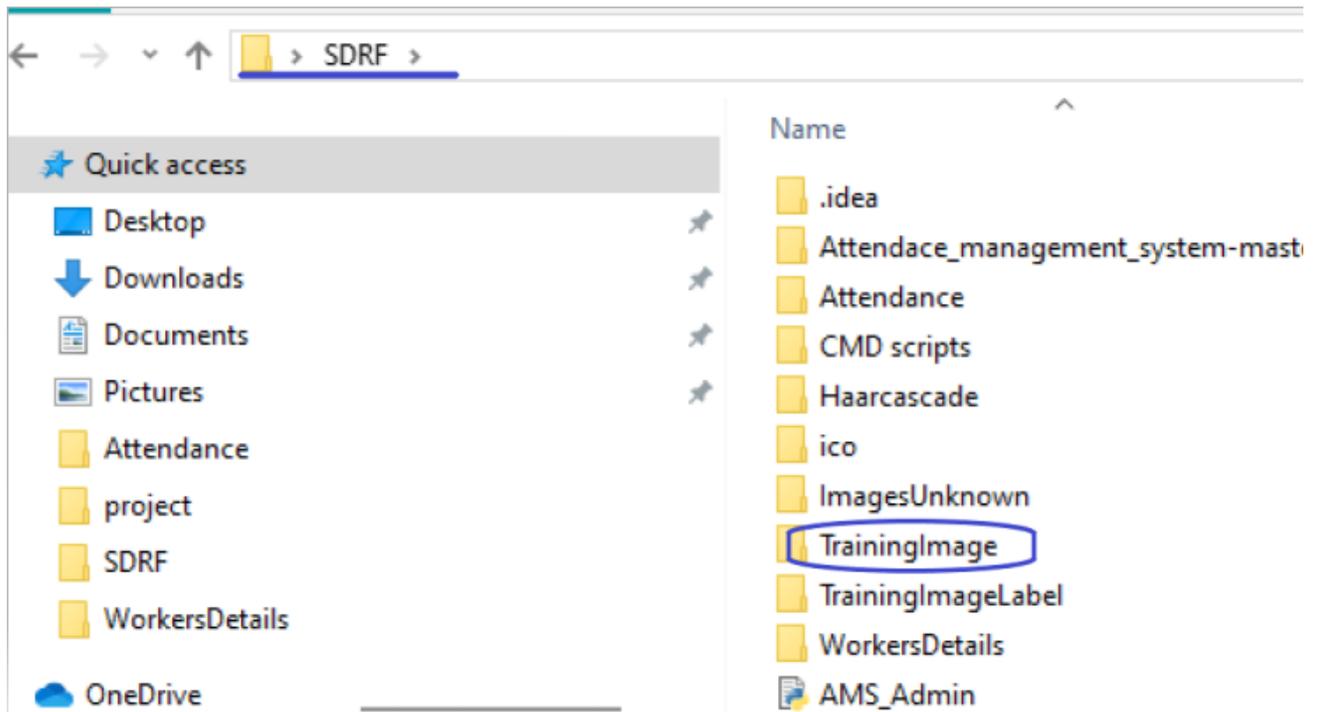


Ilustración 17. Directorio de las imágenes (Elaborado por : Lorvick y Yelsy)

En esta fase también se diseñó el interfaz gráfico del sistema, diseño que destaca por su sencillez. Además de un algoritmo básico que mediante técnicas de comparación permite determinar el reconocimiento de las personas. El algoritmo se mostrará en la Ilustración 20, el interfaz será abarcado más adelante.

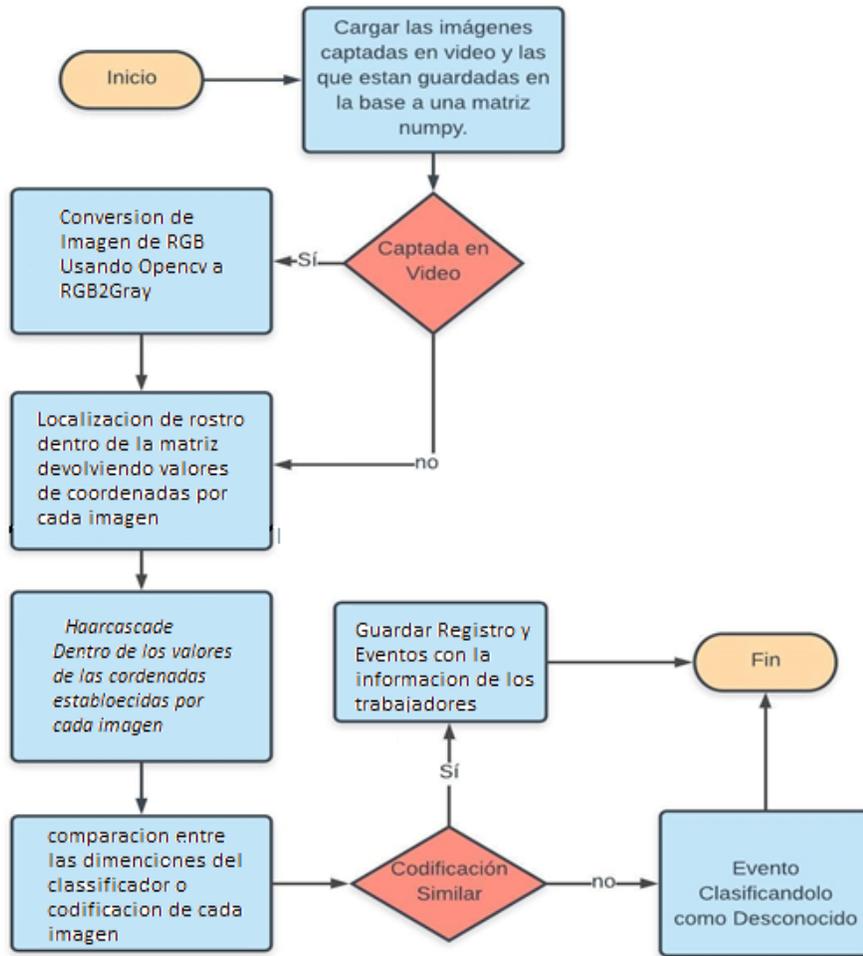


Ilustración 18. Algoritmo del sistema prototipo (Elaborado por: Lorvick y Yelsy)

7.3. Elaboración del Prototipo

El prototipo está compuesto de un data set de imágenes o directorio, que contiene 151 imagen por persona que se presenta en la figura 19.

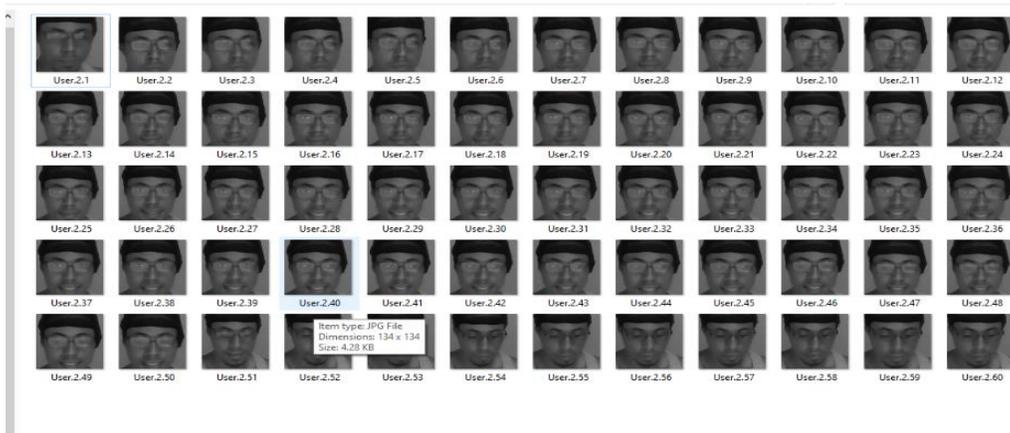


Ilustración 19.. Dataset de 151 imágenes (Elaborado por Lorvick y yelsy)

El interfaz del sistema está estructurado de la siguiente forma; opción de insertar código de trabajador y nombre y cuatro botones, el cual se observa en la figura 20.

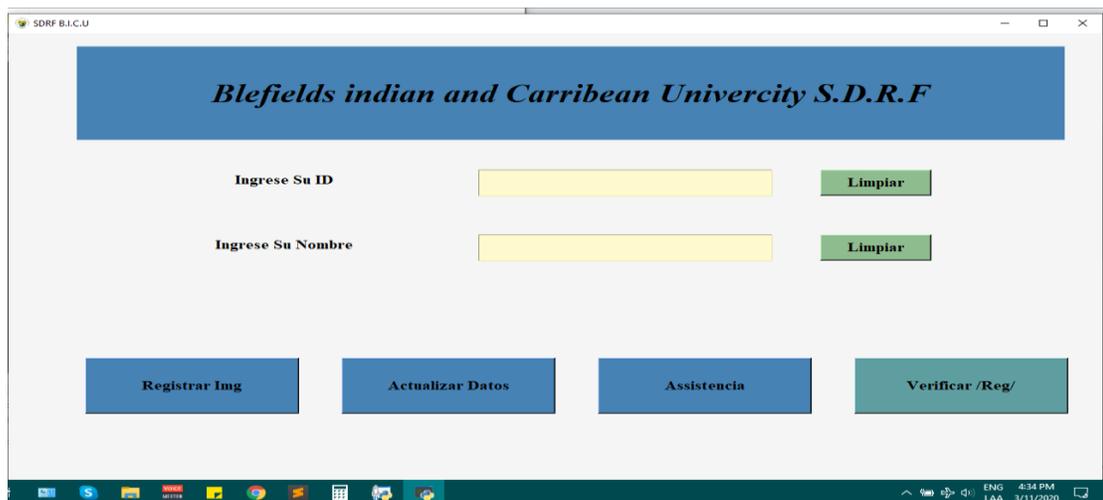


Ilustración 20. Interfaz del prototipo (Elaborado por: Lorvick y Yelsy)

Para realizar el proceso de registro de las características del trabajador, se insertó el código de trabajador y su nombre después se presiona el botón de “registro img.” que registra las características faciales del individuo, como se muestra en la figura 21.



Ilustración 131. Registrar características faciales (Elaborado por: Lorvick y Yelsy)

La opción de “actualizar datos”, es para entrenar las imágenes, el botón de “asistencia” es para registrar las horas de trabajo del empleado por medio del rostro insertando solo su código, y el ultimo el botón “verificar/reg.” para confirmar que el registro fue exitoso. Los problemas que presentó el prototipo fueron el bajo nivel de credibilidad y poca exactitud de precisión en cuanto a la detección de rostros

7.4. Desarrollo, Entrega y Re-alimentación

En esta fase se permite crear más de una versión del sistema hasta que cumple con el propósito. Dentro de las librerías que se utilizaron en el desarrollo de este proyecto se menciona que la librería OPenCV se mantuvo por lo que resulta de vital importancia para el desarrollo del proyecto. El programa pudo llevarse a cabo empleando solo esta librería para lograr el reconocimiento facial.

El sistema en general esta segmentado en tres partes de desarrollo que son:

- Módulo Almacenamiento (MySQL)
- Módulo Procesamiento (Opencv)
- Módulo Presentación (Python-Tkinter)

7.4.1. Módulo Almacenamiento

En este módulo es donde se hizo el total desarrollo de la base de datos para el proyecto. Hay tres datos principales en el proceso de almacenamiento las cuales fueron las tablas de trabajador, registro_h y el directorio de imágenes, estos datos cumplieron con la demanda de la información que se pretendía almacenar. Una vez terminado con el módulo de Almacenamiento se procede a continuar con el desarrollo del siguiente módulo el cual corresponde al módulo de procesamiento.

7.4.2. Módulo Procesamiento

Se puede llamar a este módulo también como el módulo de enlace o módulo central y esto se debe a que dentro de este módulo se realizó la conexión tanto del Módulo de Almacenamiento como del módulo de presentación, sirviendo de esta manera como nexo entre todo el sistema.

En esta parte del desarrollo es donde se evidenciaron muchos más cambios, dado que al ser el módulo de nexo se debieron cumplir con varias pautas que permitan realizar el enlace correcto del sistema con la base de datos. Como punto de partida se realizó toda la operación de reconocimiento facial con machine learning.

En esta ocasión se instalaron las librerías necesarias como OpenCV, Numpy, datetime, mysql connector para Python, pandas. Dentro de este apartado de librerías se determinó que contiene funciones para poder operar en base al algoritmo que se planteó inicialmente, por lo que es imprescindible el uso de esta librería.

Esta versión no contó con una prueba final debido a que no se pudo llevar a cabo porque a pesar de tener las librerías no se podía ejecutar por problemas de compatibilidad de versiones. Estas fallas fueron totalmente investigadas por lo que se tuvo que recurrir a la instalación de nuevas librerías más actualizadas.

Librerías instalado por medio de command line(CMD):

- Pip install OpenCV
- Pip install numpy
- Pip install pandas
- Pip install pillow
- Pip install datetime

- Pip install time
- Pip install reportlab
- Pip install mysql-connector-python

Al realizar estas modificaciones a la versión anterior del programa se pudo realizar la utilización de la librería y se procedió a llevar a cabo el algoritmo presentado anteriormente. Los resultados en esta versión fueron satisfactorios ya que se logró completar la fase de reconocimiento y conexión con la base de conocimiento, en donde se guardaban los datos con total normalidad.

7.4.3. Módulo Presentación

Esta fase es la que visualizará plenamente el usuario final dado que aquí es donde se crea la interfaz que utiliza el usuario, cabe destacar que si bien esta parte no es el tema central del proyecto no deja de ser menos importante. Por medio de este módulo se da a conocer la manera en cómo se muestran los resultados previamente canalizados por los otros módulos.

7.5. Entrega del Producto Final

En este acápite es la fase se alcanzaron todas las sus metas, ya realizado todas las etapas del modelo prototipo, así tener un producto final, donde el usuario está satisfecho y el proyecto esté lista para entregar. En este caso particular el propósito de este proyecto fue para optar al título de Ingeniería de sistema, la entrega de proyecto final será entregado y juzgado en el día de defensa realizado en la universidad BICU en el recinto Bluefields

Los entregables representan la base fundamental de todo el trabajo realizado. Por cada una de las fases de la metodología que abarco el proyecto se determinaron entregables que ayudarán a constatar todo el trabajo realizado.

A continuación, se presentan cada uno de los entregables realizados a lo largo del proyecto:

- Proyecto final
- Manual de usuario
- Prototipo Final (software *SDRF-BICU*)

VIII. RESOLUCIÓN

8.1. Caso de estudio

Los casos que se presentan en esta sesión se plantean con la finalidad de demostrar la funcionalidad del sistema, por ello están relacionados directamente con los objetivos de ejecución planteados en este proyecto.

1. **Etapa uno:** Registrar las características faciales de un trabajador.

El trabajador 1 realizó los siguientes pasos para registrarse en el SDRF-BICU, primero ir a la oficina de recursos humanos, segundo proveer los requisitos necesarios, como datos personales al administrador, después tomar una foto del rostro frontal del trabajador la duración de la sesión de fotos es de aproximadamente 1 minuto, las fotos fueron guardadas, después se procesan los datos conocido como entrenamiento del sistema para procesar las características faciales del trabajador 1 los cuales son actualizados al base de conocimientos.

2. **Etapa dos:** Identificar y reconocer a un trabajador ya registrado en el sistema.

Primero el trabajador 2 se presentó al área de registro y coloca su rostro frente al equipo, luego en menos de minutos el sistema automáticamente verifica, identifica y reconoce si dicha persona está registrado en la basa de datos, al final mostrar un mensaje de afirmación indicando que está registrado.

3. **Caso tres:** Identificar los rostros no registrados dentro de la base de conocimientos y recomendar el registro de un trabajador.

Primero el trabajador 3 se presentó al área de registro y coloca su rostro frente al equipo, luego en menos de minutos el sistema automáticamente verifica, identifica y reconoce si aquella persona está registrado en la base de datos, al final proporciona un mensaje de si o no está registrado, si no está registrado se recomienda ir al área de recursos humanos para registrarse.

4. **Etapa cuatro:** Registrar la hora de entrada y salida de un trabajador

El trabajador 4, se presentó al área de registro, luego se insertó su código de trabajador en el sistema, que verifica si el código es correcto, después se abrió la ventana de registro en donde se captura solo su cara y verifica, reconoce si el trabajador 4 está en la base de conocimientos, al finalizar se guardó su hora de entrada o salida y fecha de registro, este proceso antes mencionado se repite cuatro veces al día para registrar las horas de E/S del trabajador 4.

5. Etapa cinco: Llevar control de registro de datos de los empleadores.

El SDRF-BICU controla los horarios de cada trabajador y lo guarda el código, nombre y apellido, fecha de registro y las horas de trabajo E/S.

6. Etapa seis: Generar reporte de entrada y salida de los empleados.

El administrador ingresa a la cuenta de administrador del sistema, después selecciona la opción de reportes por individual o grupal, y lo imprime.

8.2. Experimento del caso de estudio

Aquí se presenta los experimentos realizados de los casos expuestos anteriormente, en la misma secuencia.

Etapa 1: Registrar las características faciales de un trabajador.

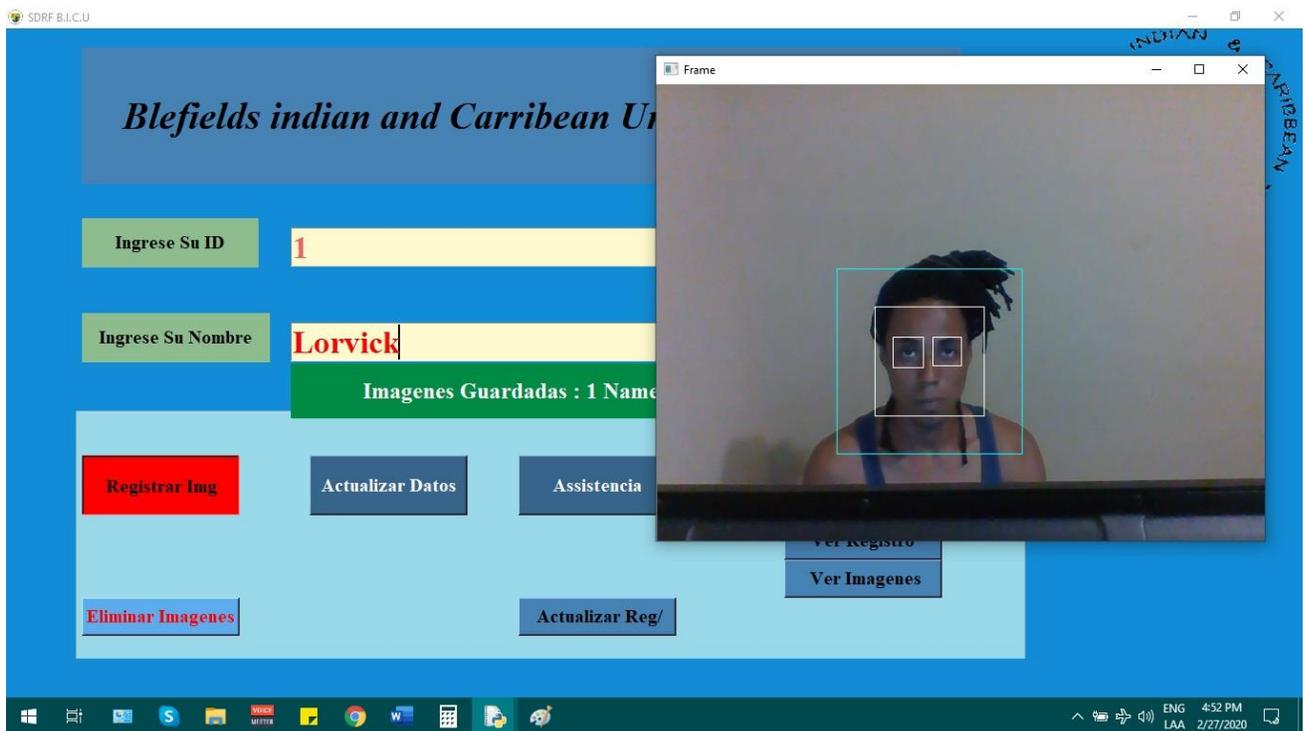


Ilustración 142. Registro de característico facial (Elaborado por: Lorvick y Yelsy)

En la ilustración 22, es donde se registra la característica facial del rostro de un trabajador, primero se ingresa el código de trabajador y su nombre, luego se presiona el botón llamada Registro Img, después

muestra una ventana de forma de una cámara como se puede observar a la esquina de la ilustración, se captura solo el rostro del trabajador por un minuto, solo la selección del cuadro, se guarda los imágenes de escala gris, al final se entrena el sistema para que puede identificar y reconocer el rostro del trabajador.

Etapas 2: Identificar y reconocer a un trabajador ya registrado en el sistema.

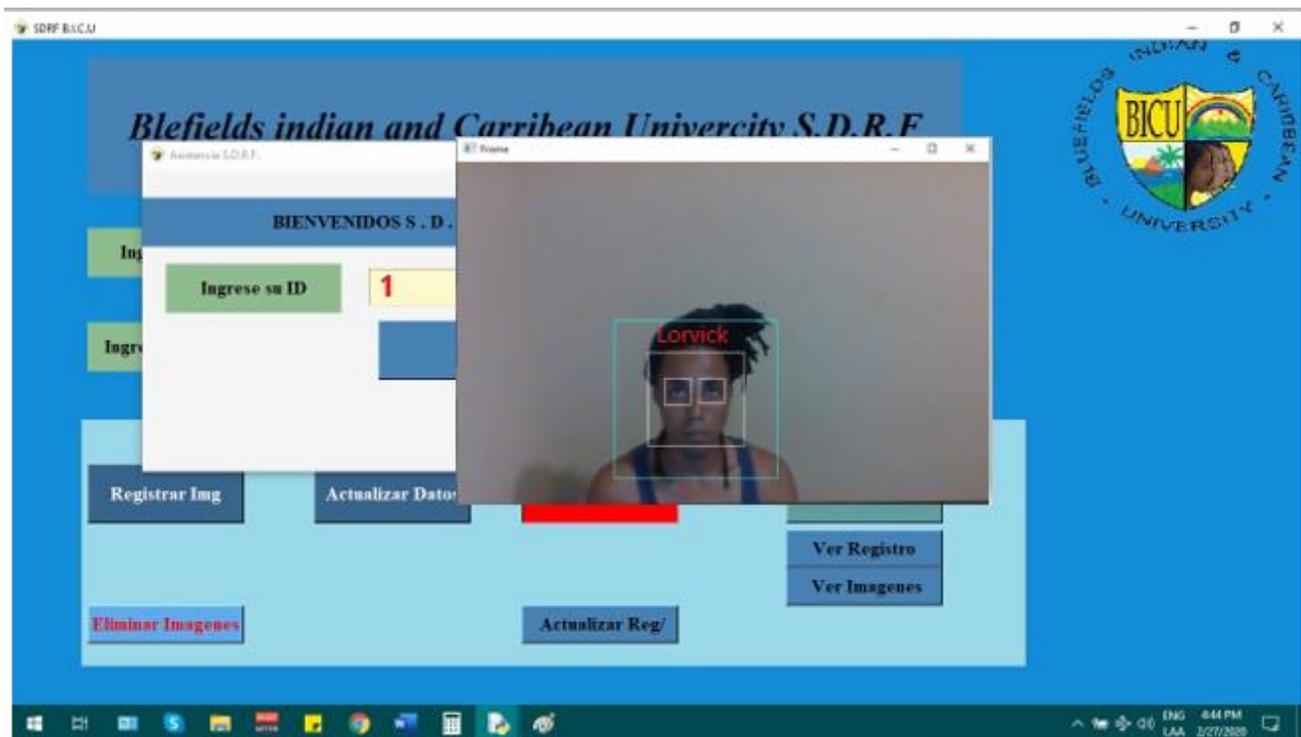


Ilustración 153. Identifica y reconoce los rostros registrado (Elaborado por: Lorvick y Yelsy)

Se observa que en la ilustración 23, Que el sistema identificó un rostro ya registrado y guardado en el sistema, primero se inserta el código de trabajador, después se abre una ventana de forma de cámara, se verifica y compara el rostro presente con las imágenes ya guardado y luego se muestra el nombre del trabajador, al final se cierra la ventana en menos de minutos.

Etapa 3: Identificar los rostros no registrados dentro de la base de conocimientos



Ilustración 24. Identifica rostros desconocido (Elaborado por: Lorvick y Yelsy)

La ilustración 24, muestra que el sistema reconoce un rostro desconocido, primero se inserta el código de trabajador, después se abre una ventana de forma de cámara, se verifica y compara el rostro presente con las imágenes ya guardado y como no está registrado se muestra desconocido al trabajador, al final se cierra la ventana en menos de minutos.

Etapa 4: Registrar la hora de entrada y salida de un trabajador

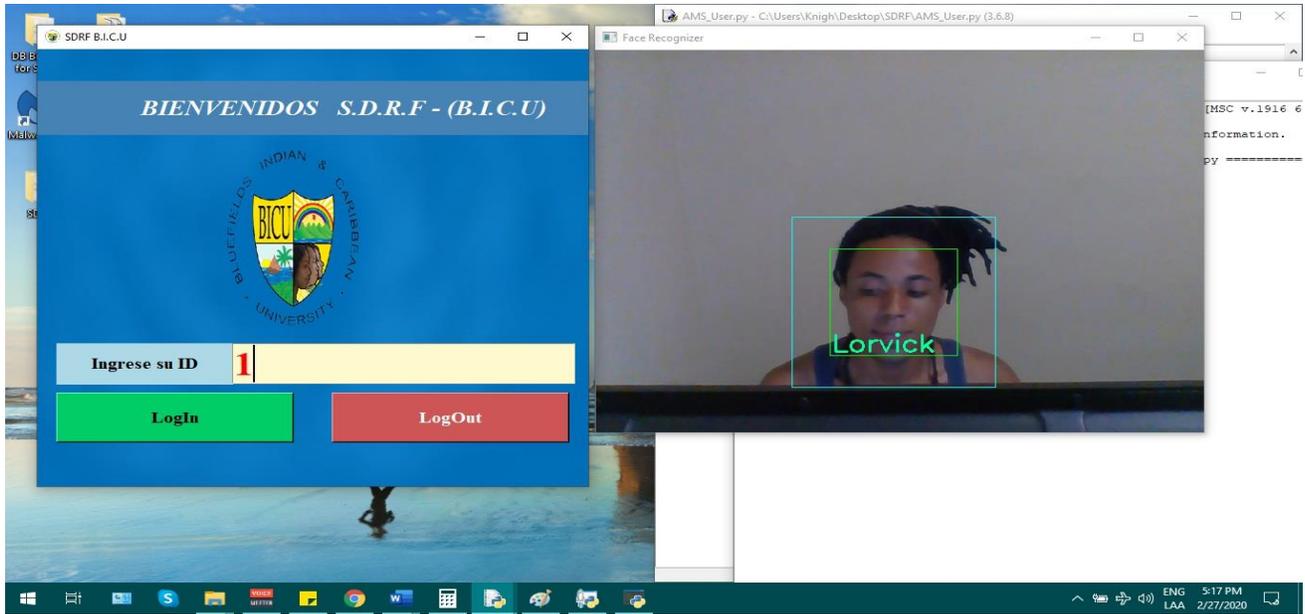


Ilustración 165. Proceso de registro de horas E/S (Elaborado por : Lorvick y Yelsy)

Se observa que en la ilustración 25, se realizó el proceso de registro de las horas de entrada y salida, primero se inserta el código de trabajador y se selecciona que hora se registre sea entrada “Login” o salida “Logout”, después se abre una ventana de forma de cámara, el sistema automáticamente verifica y compara el rostro presente con las imágenes ya guardado y se muestra el nombre del trabajador, al final se cierra la ventana y guarda el código, nombre, fecha de registro y la hora en el sistema.

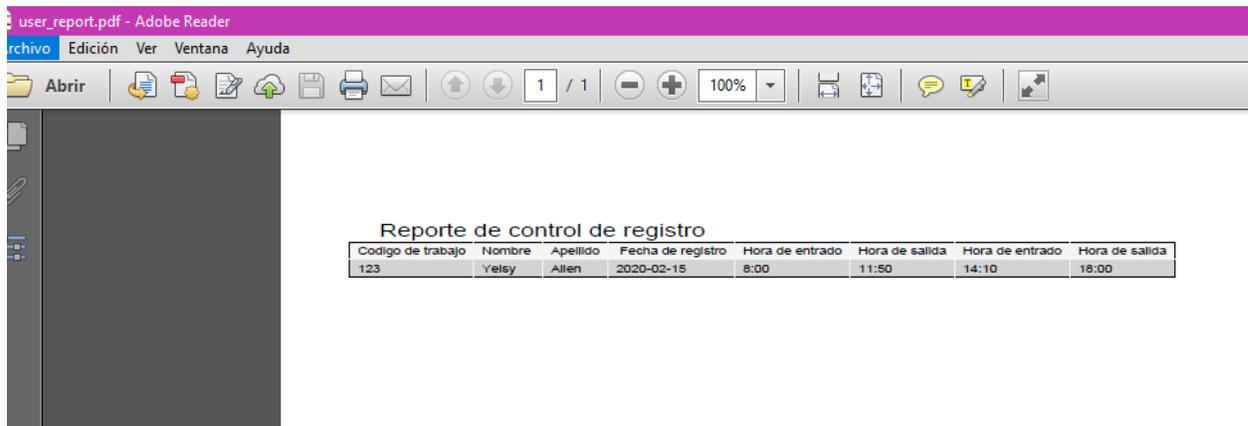
Etapa 5: Llevar control de registro de datos de un trabajador

	A	B	C	D	E	F	G	H
1	Codigo de trabajo	Nombre	Apellido	Fecha de registro	Hora de entrada	Hora de salida	Hora de entrada	Hora de salida
2	123	Yelsy	Allen	2/17/2020	8:04:10	12:23:02	14:10:04	18:05:02
3								
4								
5								
6								

Ilustración 26. Control de registro (Elaborado por: Lorvick y Yelsy)

Como se observa en la ilustración 26 el sistema maneja los datos del registro en un archivo Excel, guarda los datos esenciales de registro, este archivo no se puede modificar o eliminar, dicha información también se guarda en la base de datos.

Etapa 6: Generar reporte de entrada y salida de un trabajador.



The screenshot shows the Adobe Reader interface with a PDF document titled 'user_report.pdf'. The document content is a report titled 'Reporte de control de registro' which contains a table with the following data:

Codigo de trabajo	Nombre	Apellido	Fecha de registro	Hora de entrada	Hora de salida	Hora de entrada	Hora de salida
123	Yelsy	Allen	2020-02-15	8:00	11:50	14:10	18:00

Ilustración 17. Reporte de entrada y salida de un trabajador.

La ilustración 27 muestra el reporte de los datos de registro de entrada y salida de un trabajador.

IX. CONCLUSIONES Y VÍAS FUTURAS

9.1. Conclusiones

1. Se concluye que el sistema de reconocimiento facial es una tecnología relativamente nueva, una solución biométrica que emplea un algoritmo automático, identifica auténticamente a un individuo y brinda muchos beneficios a las instituciones públicas o privados quienes emplea esta tecnología. Por lo que esta biométrica es ideal para implementar en el recinto BICU para administrar las entradas y salidas de los empleados.
2. El SDRF-BICU cumplió con los objetivos planteados que registra las características de una persona, mediante eso registra las horas de entrada y salida de los empleados por medio de su rostro, reconoce aquellos no registrados y recomienda su registro.
3. El sistema brinda seguridad y eficaz en base a al manejo de datos, ya que no da autoridad de eliminar o modificar información a cualquier usuario y el reporte dado mensualmente se imprima en pdf.

9.2. Vías Futuras

- Para desarrollos futuros del sistema se recomienda la utilización de las librerías de tensor Flow con Keras para mejorar la precisión, y también se podría implementar la detección de movimientos los cuales son funciones que están presentes en estas librerías y de esta forma se podría automatizar el proceso de registro que no tenga más necesidad de interacción con el usuario más que la utilización de su rostro como su único id.
- También implementar una librería de machine learning con la cual se podrá actualizar constantemente los datos o características guardadas del usuario y de esta forma se tendrá una imagen más reciente del individuo, ya que el rostro humano cambia con el tiempo y si el sistema no tiene forma de actualizar estos cambios a la base de conocimiento este no podría seguir siendo tan eficiente.

X. RECOMENDACIONES

Para el correcto funcionamiento del sistema, se recomienda lo siguiente:

- El área donde el sistema estará instalado debe de tener iluminación apropiada para que la cámara pueda reconocer los rasgos faciales correctamente.
- Se recomienda una cámara digital de alta resolución con un HD 1280*960 a tiempo real, el sistema depende de la calidad de las imágenes principalmente, entre menos estática y ruido presente en las imágenes más incrementa la precisión del sistema.
- Al registrar las horas trabajadas, el trabajador tendrá que remover cualquier objeto que pueda causar interferencia del rostro, por ejemplo, unas gafas o lentes los cuales modifican los rasgos faciales del individuo causando que sus características no sean detectadas correctamente en comparación a sus rasgos faciales.
- Para su adecuada instalación del sistema, en cuanto a su ubicación se requiere de una cabina o un lugar cerrado o apartado de ruidos externos y otros factores que pueden interferir en la calidad de la imagen, es preferible tener un fondo blanco en la parte trasera para remover cualquier factor u objeto que pueda causar interferencia al momento de efectuar el registro de sus entradas y salidas.

XI. REFERENCIAS

- A.M. Martinez Robot Vision Lab., Purdue Univ., West Lafayette, IN, USA, A.C. Kak. (2001). PCA versus LDA . *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume: 23 , Issue: 2.
- Apple. (11 de 2017). *business/site/docs/FaceID_Security_Guide.pdf*. Obtenido de apple.com: https://www.apple.com/business/site/docs/FaceID_Security_Guide.pdf
- B., G. (13 de Mato de 2019). *HOSTINGER TUTORIAL*. Obtenido de <https://www.hostinger.es/tutoriales/que-es-mysql/>
- Barriga, R. C. (2017). APLICACIÓN PRÁCTICA DE LA VISIÓN ARTIFICIAL PARA EL. *RENE CABALLERO BARRIGA*, 15.
- BARRIGA, R. C. (2017). *repository*. Obtenido de repository.udistrital.edu.co: <http://repository.udistrital.edu.co/bitstream/11349/6104/1/CaballeroBarrigaEdisonRene2017.pdf>
- Becerra Quispe, D. (2016). *Dirección de Servicios de Información y Bibliotecas (SISIB) - Universidad de Chile*. Obtenido de repositorio.uchile.cl: <http://repositorio.uchile.cl/handle/2250/146204>
- Brems, M. (17 de april de 2017). *A One-Stop Shop for Principal Component Analysis*. Obtenido de towardsdatascience.com: <https://towardsdatascience.com/a-one-stop-shop-for-principal-component-analysis-5582fb7e0a9c>
- components, I. t. (s.f.). *Atchley*. Obtenido de statgen.ncsu.edu: <ftp://statgen.ncsu.edu/pub/thorne/molevclass/AtchleyOct19.pdf>
- Conger, k. (12 de 09 de 17). Obtenido de <https://gizmodo.com/face-id-raises-some-scary-questions-here-are-some-answe-1804313075>
- Davisclick/Taller. (3 de 2018). *commons.wikimedia.org*. Obtenido de https://commons.wikimedia.org/wiki/File:Clasificaci%C3%B3n_de_los_m%C3%A9todos_de_reconocimiento_de_rostros.png?uselang=es
- deAgonia, M. (01 de 11 de 2017). *article/3235140/apples-face-id-the-iphone-xs-facial-recognition-tech-explained.html*. Obtenido de www.computerworld.com: <https://www.computerworld.com/article/3235140/apples-face-id-the-iphone-xs-facial-recognition-tech-explained.html>
- Desai, K. (11 de 03 de 2016). Obtenido de <https://www.quora.com/Which-face-detection-algorithm-is-used-by-Facebook#aSoON>
- espana, k. (4 de may de 2016). *www.kaspersky.es*. Obtenido de www.kaspersky.es: <https://www.kaspersky.es/blog/how-facial-recognition-works/8564/>

Face Detection and Recognition Theory and Practice eBookslib. (2016). Florida: /www.academia.edu.

Fernando, C. C. (Marzo de 2019). *DSpace en UNIANDES*. Obtenido de <http://dspace.uniandes.edu.ec/handle/123456789/9765>

Fleishman, M. S. (25 de 12 de 2017). *article/3225406/face-id-iphone-x-faq*. Obtenido de www.macworld.com: <https://www.macworld.com/article/3225406/face-id-iphone-x-faq.html>

Galgalin, R. (27 de Febrero de 2016). *Quora*. Obtenido de <https://www.quora.com/What-is-openCV>

Harvey, A. (8 de april de 2012). *web.archive.org*. Obtenido de <https://web.archive.org/web/20171204220159/http://www.makemantics.com/research/viola-jones/>

Jain, A. (28 de 06 de 2016). *vinsol.com*. Obtenido de <https://vinsol.com/blog/2016/06/28/computer-vision-face-detection/>

Jarroba.com. (30 de Octubre de 2015). Obtenido de <https://jarroba.com/pandas-python-ejemplos-parte-i-introduccion/>

Javier, P. J. (2018). Obtenido de Repositorio Institucional de la Universidad de Guayaquil: <http://repositorio.ug.edu.ec/handle/redug/32954>

Jones, P. V. (21 de 3 de 2001.). *The Viola/Jones Face Detector*. Obtenido de cs.ubc.ca: <https://www.cs.ubc.ca/~lowe/425/slides/13-ViolaJones.pdf>

Jones, P. V. (04 de july de 2018). *OpenCV*. Obtenido de https://docs.opencv.org/3.4.2/d7/d8b/tutorial_py_face_detection.html

José Augusto Cadena Moreano, G. A. (Ene-Jun. 2018). Reconocimiento facial en sub-espacios: lineales y no-lineales, bases de datos de. *ARJÉ. Revista de Postgrado FaCE-UC., vol. 12-22*. Obtenido de <http://arje.bc.uc.edu.ve/arje22e/art41.pdf>

Navio, N. C. (2017). *handle*. Obtenido de riunet.upv.es: <https://riunet.upv.es/handle/10251/86962>

Numpy.org. (29 de Junio de 2020). Obtenido de <https://numpy.org/doc/stable/user/whatisnumpy.html>

Open Genus. (s.f.). Obtenido de <https://iq.opengenus.org/lbph-algorithm-for-face-recognition/>

Pentland, M. T. (1991). Obtenido de <https://www.face-rec.org/algorithms/PCA/jcn.pdf>

Peter N. Belhumeur, J. P. (Julio de 7 de 1997). Obtenido de <https://www.cse.unr.edu/~bebis/MathMethods/LDA/belhumeur.pdf>

Python, M. D. (s.f.). Obtenido de <https://unipython.com/tkinter-introduccion/>

Raschka, e. (3 de August de 2014). *Articles/2014_python_lda*. Obtenido de sebastianraschka.com: https://sebastianraschka.com/Articles/2014_python_lda.html

Rigg, J. (18 de 03 de 2014). Obtenido de <https://www.engadget.com/2014/03/18/facebook-deepface/>

Sacks, O. (1985). Obtenido de <https://neurorhb.com/blog-dano-cerebral/prosopagnosia-quien-es-quien/#:~:text=La%20prosopagnosia%20es%20un%20tipo,espejo%20o%20en%20una%20fotograf%C3%ADa..>

Salcedo, L. (16 de Octubre de 2017). *Python Diario*. Obtenido de <http://www.pythondiario.com/2017/10/creacion-de-archivos-pdf-con-python-y.html>

support.apple.com. (26 de February de 2020). Obtenido de <https://support.apple.com/en-us/HT208108>

Suquinagua León, G. E. (19 de marzo de 2019). *UNIVERSIDAD CATOLICA DE SANTIAGO DE GUAYAQUIL*. Obtenido de <http://192.188.52.94:8080/handle/3317/12595>

team, F. (s.f.). Obtenido de <https://www.facebook.com/help/122175507864081>

Techterms. (15 de Junio de 2010). *Techterms.com*. Obtenido de <https://techterms.com/definition/python>

Varman, M. (02 de 03 de 2017).

Vincent, V. (18 de 01 de 2015). Obtenido de quora.com: <https://www.quora.com/Which-face-detection-algorithm-is-used-by-Facebook#aSoON>

Viola, P. (2001). Ada Boost .

Wikipedia. (26 de Octubre de 2012). Obtenido de [https://es.wikipedia.org/wiki/C%C3%A1mara_web\(WIKIPEDIA\)](https://es.wikipedia.org/wiki/C%C3%A1mara_web(WIKIPEDIA))

Yabiku, O. (Agosto de 2017). Obtenido de <https://www.vexsoluciones.com/biometria/ventajas-beneficios-reconocimiento-facial/>