



# ESCUELA DE INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN

Tesis previa a la obtención del título de  
Ingeniero en Tecnologías de la Información

**AUTOR: Eduardo J. Araujo A.**

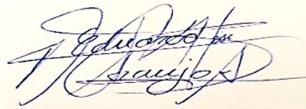
**DIRECTOR: Mgtr. Milton Ricardo Palacios Morocho**

**Diseño de un prototipo para el control de asistencia de los docentes de la  
carrera de Tecnologías de la información en la Universidad Internacional del Ecuador  
sede Loja mediante el uso de visión artificial**

## ACUERDO DE CONFIDENCIALIDAD

Yo, Eduardo José Araujo Abad, declaro bajo juramento, que el trabajo aquí descrito es de mi autoría; que no ha sido presentado anteriormente para ningún grado académico o título profesional y que se ha consultado la bibliografía necesaria para su elaboración.

Cedo mis derechos de propiedad intelectual a la Universidad Internacional del Ecuador, para que sea publicado y divulgado en internet, según lo establecido en la Ley de Propiedad Intelectual, reglamento y leyes.



---

Eduardo José Araujo Abad

Yo, Mgtr. Milton Ricardo Palacios Morocho, certifico que conozco al autor del presente trabajo, siendo el responsable exclusivo tanto de su originalidad y autenticidad, como de su contenido.

---

Mgtr. Milton Ricardo Palacios Morocho, Mg. Sc.  
DIRECTOR DE TESIS

## **DEDICATORIA**

El presente proyecto de tesis se lo dedico a: mis padres; Lourdes y Eduardo, a mi hermana; Salome, a mi abue; Ena. Por su incondicional apoyo que me han brindado durante toda mi vida académica y deportiva. Gracias a ustedes por ser la parte fundamental de mis logros.

Eduardo J. Araujo A.

## **AGRADECIMIENTO**

La Universidad Internacional del Ecuador con sede en Loja, ha sido parte fundamental en mi educación superior, por lo que agradezco profundamente a todos los docentes, plantel administrativo y compañeros, que han sido parte del diario vivir en las aulas de tan prestigiosa y querida institución.

Mi agradecimiento especial a Magister Lorena Conde, directora de la Carrera de informática y Multimedia, por ser mentora, consejera y maestra, en la transmisión de sus conocimientos para la oportuna realización del presente proyecto de titulación.

Agradezco a Magister Milton Palacios, Director de Tesis, por su sabiduría impartida en las aulas y en el asesoramiento eficaz para la consecución del presente proyecto de titulación.

Eduardo J Araujo A.

## ÍNDICE DE CONTENIDOS

PORTADA .....	i
ACUERDO DE CONFIDENCIALIDAD .....	ii
DEDICATORIA .....	iii
AGRADECIMIENTO .....	iv
ÍNDICE DE CONTENIDOS .....	v
Resumen .....	ix
Abstract .....	x
Introducción .....	1
Problematización .....	3
Objetivos.....	4
CAPITULO I: ESTADO DEL ARTE .....	5
1. Conceptos. ....	6
1.1. Lenguaje de Programación utilizado para la realización del Proyecto de titulación: Python. ....	6
1.1.1. ¿Qué es Python?.....	6
1.2.1. ¿Qué es reconocimiento facial?.....	6
1.2.2. ¿En qué consiste la biometría facial? .....	7
1.3. Inteligencia artificial. ....	7
1.3.1. ¿Qué es la inteligencia artificial? .....	7
1.3.2. Tipos de inteligencia artificial. ....	7
1.3.3. Tipos de aprendizaje en inteligencia artificial .....	10
1.3.4. HAAR Features.....	12
1.4. Herramientas. ....	12
1.4.1. Open CV.....	12
1.4.2. Librerías, modelos y bibliotecas usadas para la construcción del sistema. ....	12
1.5. Metodología.....	14
1.5.1. Métodos.....	14
1.5.2. Metodología de desarrollo. ....	14
CAPITULO II: ANÁLISIS Y DISEÑO .....	18
2. Investigación preliminar. ....	19
2.1. Sistema de registro biométrico-dactilar UIDE.....	19
2.2. Actualización del Sistema de control de asistencia de docentes del Área de Tecnología de la Información. ....	19
2.3. Requerimientos .....	20
2.3.1. Requerimientos funcionales.....	20

2.3.2. Requerimientos no funcionales .....	20
2.3.3. Requerimientos de hardware para el uso del ambiente de desarrollo Anaconda Navigator .....	21
2.4. Modelado .....	21
2.4.1. Diagrama sistema simplificado .....	21
2.4.2. Esquema de funcionamiento .....	21
2.5. Metodología para el desarrollo del proyecto .....	22
2.6. Elementos Scrum .....	22
2.6.1. Lista de tareas o Product Backlog .....	23
2.6.2. Backlog Sprint .....	23
CAPITULO III: DESARROLLO .....	25
3. Desarrollo del Sistema .....	26
3.1. Antecedentes de la Marcación de asistencia docente del Área de Tecnologías de la información de la Universidad Internacional del Ecuador, sede Loja .....	26
3.2. Experimentación .....	26
3.2.1. Experimentación 1 .....	26
3.3. Sprints .....	45
CAPITULO IV: CONCLUSIONES Y RECOMENDACIONES .....	66
4.1. Conclusiones .....	67
4.2. Recomendaciones .....	68
ANEXOS .....	71
5. Anexos .....	72
5.1. Anexo 1. Anteproyecto .....	72
5.2. Cuestionario .....	84

## ÍNDICE DE TABLAS

<b>Tabla 1</b> Backlog Sprint.....	23
<b>Tabla 2.</b> Especificaciones .....	67
<b>Tabla 3.</b> Cronograma de actividades .....	81

## ÍNDICE DE FIGURAS

<b>Figura 1.</b> Funciones de activación para una red neuronal .....	8
<b>Figura 2.</b> Red Neuronal – Perceptrón Simple con cálculo de activación.....	8
<b>Figura 3.</b> Red neuronal – perceptrón multicapa.....	9
<b>Figura 4.</b> Filtrado de red convolucional mediante los métodos maxpooling, minpooling y avrgpooling .....	9
<b>Figura 5.</b> Representación funcionamiento del aprendizaje supervisado .....	10
<b>Figura 6.</b> Representación del aprendizaje no supervisado .....	11
<b>Figura 7.</b> Representación del aprendizaje semi-supervisado .....	11
<b>Figura 8.</b> Representación del Aprendizaje por refuerzo.....	12
<b>Figura 9.</b> Proceso de Scrum.....	15
<b>Figura 10.</b> Desinfección de elementos de trabajo.....	20
<b>Figura 11.</b> Diagrama del prototipo simplificado.....	21
<b>Figura 12.</b> Esquema de funcionamiento completo del prototipo .....	22
<b>Figura 13.</b> Interfaz de registro de un nuevo usuario. ....	27
<b>Figura 14.</b> Segmento del código perteneciente a la vista del formulario de registro. ....	28
<b>Figura 15.</b> El sistema realiza la toma de datos, imágenes de los usuarios. ....	29
<b>Figura 16.</b> Con diferente usuario se detalla la creación satisfactoria del dataset.....	29
<b>Figura 17.</b> Sección de código de la creación del dataset.....	30
<b>Figura 18.</b> Continuación de la sección de Código para la creación de dataset.....	32
<b>Figura 19.</b> Entrenamiento de datos .....	33
<b>Figura 20.</b> Segmento del modelo de entrenamiento de datos .....	34
<b>Figura 21.</b> Segmento final del modelo de entrenamiento. ....	35
<b>Figura 22.</b> Marcación de asistencia, entrada, de la Directora de carrera, Lorena Conde, con el nombre de usuario “lorena”. ....	37
<b>Figura 23.</b> Marcación de asistencia, entrada, del director de Tesis Milton Palacios, con el nombre de usuario “milton” .....	37
<b>Figura 24.</b> Sección de código marcación de asistencia .....	38
<b>Figura 25.</b> Sección de Código marcación de asistencia .....	40
<b>Figura 26.</b> Sección de Código marcación de asistencia .....	41
<b>Figura 27.</b> Sección final de Código marcación de asistencia.....	42
<b>Figura 28.</b> Marcaje de salida del usuario “milton” .....	42
<b>Figura 29.</b> Marcaje de salida del usuario “lorena”.....	43

<b>Figura 30.</b> Representación del marcaje de asistencia valido para el usuario “milton”.....	43
<b>Figura 31.</b> Representación del marcaje de asistencia valido para el usuario “lorena”.....	44
<b>Figura 32.</b> Gráfico de barras realizado con matplotlib .....	44
<b>Figura 33.</b> Diagrama de clases (elaborado en <a href="http://online.visual-paradigm.com">online.visual-paradigm.com</a> ).....	45
<b>Figura 37.</b> Datos pertenecientes al modelo de la Figura,34.....	47
<b>Figura 38.</b> Diseño vista preliminar vista principal del sistema.....	48
<b>Figura 39.</b> Vista principal del sistema.....	49
<b>Figura 40.</b> Construcción de la vista principal en html5. (segmento).....	50
<b>Figura 41.</b> Inserción de estilos CSS para la vista principal. (segmento) .....	50
<b>Figura 42.</b> Inserción de estilos CSS para la vista principal y (segmento), Construcción de la vista principal en html5. (segmento).....	51
<b>Figura 43.</b> Construcción de la vista principal en html5. (segmento).....	51
<b>Figura 44.</b> Importación de módulos de Java Script.....	52
<b>Figura 45.</b> Diseño preliminar de la vista de reporte de asistencia.....	53
<b>Figura 46.</b> Definición de la función y codificación del algoritmo para visualizar los datos requeridos en la vista principal. (segmento).....	54
<b>Figura 47.</b> Codificación del algoritmo para visualizar los datos requeridos en la vista principal. (segmento) .....	54
<b>Figura 48.</b> Codificación del algoritmo para visualizar los datos requeridos en la vista principal. (segmento) .....	55
<b>Figura 49.</b> Etiquetas HTML para la creación de la vista mostrada en la figura27. ....	56
<b>Figura 50.</b> Etiquetas HTML para la creación de la vista mostrada en la figura27. ....	56
<b>Figura 51.</b> Etiquetas HTML para la creación de la vista mostrada en la figura27 .....	57
<b>Figura 52.</b> Etiquetas HTML para la creación de la vista mostrada en la figura27. ....	57
<b>Figura 53.</b> Diseño preliminar de la sección horarios por asignar. ....	58
<b>Figura 54.</b> Asignar horarios a los usuarios .....	59
<b>Figura 55.</b> Etiquetas HTML para la creación de la Fig.52.....	60
<b>Figura 56.</b> Etiquetas HTML para la creación de la Fig.52.....	60
<b>Figura 57.</b> Diseño preliminar de la vista de horarios.....	61
<b>Figura 58.</b> Tabla de horario vigentes por asignatura. ....	62
<b>Figura 59.</b> Clases del modelo pertenecientes a la asignación de horarios, .....	62
<b>Figura 60.</b> Código de la función que valida la entrada del usuario.....	63
<b>Figura 61.</b> Código de la función que valida la salida del usuario. ....	63
<b>Figura 62.</b> Agregación de horario al usuario “eduardo” .....	64
<b>Figura 63.</b> Marcación dentro del rango del horario. ....	64
<b>Figura 64.</b> Demostración del almacenamiento de los registros. ....	65

## Resumen

Este proyecto de titulación tiene como objetivo agilizar y controlar el proceso de marcación de asistencia del plantel docente del área de Tecnología de la Información Perteneciente a la Universidad Internacional del Ecuador sede Loja, mediante el registro de asistencia ingreso y salida, con el uso de técnicas de reconocimiento facial. El prototipo se implementa mediante el uso del framework Django basado en el lenguaje de programación Python en su versión final hasta la fecha 3.10, para las funcionalidades transaccionales y para reconocimiento facial se utilizan diferentes funciones, métodos, objetos y modelos que, en conjunto, con las librerías apropiadas de inteligencia artificial, (visión artificial): OpenCV, Dlib, imutils, Time, Os, Face\_recognition, Picle, Sklearn, Numpy, Matplotlib, Datetime, Seaborn, Pandas, Math, dan funcionamiento al sistema. Permitiendo realizar la marcación de horarios mediante el reconocimiento facial.

**Palabras clave:** Python, Inteligencia Artificial, Reconocimiento Facial, Marcación de asistencia.

## **Abstract**

This titling Project aims to speed up and control the attendance marking process of the teaching staff of the Information Technology area belonging to the Universidad Internacional del Ecuador sede Loja, through the registration of incoming and outgoing attendance, with the use of facial recognition techniques. The prototype is implemented using the Django framework based in its last version up to date 3.10, for the transactional functionalities and for facial recognition different functions, methods, objects, and models are used that, together with the appropriate libraries of artificial intelligence (artificial vision): OpenCV, Dlib, imutils, Time, Os, Face\_recognition, Picle, Sklearn, Numpy, Matplotlib, Datetime, Seaborn, Pandas, Math, give operation to the system. Allowing to make the marking of schedules with facial recognition.

**Keywords:** Python, Artificial Intelligence, Facial Recognition, Attendance Marking

## Introducción

Desde el año 2020 debido a la situación global con el virus Covid-19, declarada como pandemia por la OMS, el COE Nacional mediante resolución del 16 de marzo del 2020, “recomienda al señor Presidente de la Republica del Ecuador, la declaratoria de estado de excepción en todo el territorio nacional”. Ante este requerimiento urgente el señor Presidente Constitucional de la Republica Lic. Lenin Moreno Garces con decreto No. 1017 de fecha 16 de marzo del 2020, “declara el estado de excepción por calamidad pública en todo el territorio nacional, a fin de adoptar algunas medidas, como la suspensión total de la jornada laboral presencial en el sector público y privado a partir del martes 17 de marzo. En razón que “la enfermedad puede propagarse de persona a persona a través de las goticulas procedentes de la nariz o la boca que salen despedidas cuando una persona infectada estornuda, tose o exhala. Estas goticulas caen sobre objetos y superficies que rodean a la persona, de modo que otras personas pueden contraer COVID-19 si tocan estos objetos o superficies y luego se tocan los ojos, la nariz o la boca””. El Ministerio da Salud Publica, publica el Protocolo Lineamientos para la prevención COVID-19 e Inocuidad de los Alimentos MTT6-LIA (versión 1.2) Año 2020, con el propósito de “franquear esta pandemia y proteger la salud y seguridad de todas las personas, recomendando que se deben intensificar las medidas de higiene personal como el lavado y desinfección de manos, así como el uso en todo momento de los elementos de protección personal como mascarillas, guantes, etc.” Ante estos ineludibles hechos, varios de los dispositivos tecnológicos que el ser humano tenía que usar para la captura de datos biométricos dactilares, se dejaron de emplear, por cuanto se consideran un foco de infección inminente, teniendo que ser desinfectados constantemente para que puedan ser usados, acortando su vida útil.

Por los antecedentes expuestos, la Universidad Internacional del Ecuador sede Loja, decidió suprimir el servicio del equipo biométrico dactilar de registro de asistencia tanto para docentes como personal administrativo, y retornar al registro de asistencia manual.

Dada la situación, el presente proyecto se basa en un método factible para evitar el contacto físico con un dispositivo, el mismo que haciendo el reconocimiento facial con el uso de una cámara, automáticamente realizara la marcación de entrada-salida según sea el caso, de la persona en cuestión.

Para registrar cada individuo el administrador, admin o superusuario deberá realizar el proceso adecuado de registro seguro, desde su cuenta, ya que solo este tendrá el acceso a todas las herramientas: creación de usuario, toma de fotografías, entrenamiento de la

inteligencia artificial en base a las fotografías capturadas, ver reportes de asistencias, ver horarios de clases y asignarlos. Internamente en la dirección /admin, este podrá tener un control más detallado de las distintas funciones del proyecto.

El individuo no administrador solo podrá en su cuenta realizar la acción de ver sus reportes de asistencia.

## **Problematización**

Por motivos de contagios de Síndrome Agudo Respiratorio Severo- Covid-19, el método biométrico dactilar fue dado de baja en el alma mater UIDE-Loja. Llevando al plantel docente del Área de Tecnologías de la Información a usar una libreta de asistencia para la marcación.

### **Causas**

Contagios de personas.

Método de marcación obsoleto en la actualidad.

### **Efectos**

Riesgo de vida por morbilidad.

Dificultad de las personas al acceder a la marcación.

## **Objetivos**

### **Objetivo general**

Desarrollar el Prototipo de Sistema de asistencia para los docentes del Departamento de Tecnologías de la Información en UIDE (Universidad Internacional del Ecuador) sede Loja, mediante el uso de visión artificial.

### **Objetivos específicos**

- Consultar el Estado del arte para la elaboración de prototipo //tecnologías a usarse.
- Determinar las metodología y herramientas que se adapten al proyecto
- Analizar la situación actual y levantar de requerimientos del control de asistencia dentro de la UIDE sede Loja
- Diseñar el prototipo para la asistencia docente
- Realizar fase de pruebas del prototipo de asistencia

# **CAPITULO I: ESTADO DEL ARTE**

## **1. Conceptos.**

### **1.1. Lenguaje de Programación utilizado para la realización del Proyecto de titulación: Python.**

#### **1.1.1. ¿Qué es Python?**

Es un lenguaje de programación de alto nivel, muy popular en la actualidad, ya que permite al programador realizar diferentes tareas complejas en escasas líneas de código, dirigido hacia la sistematización de objetos y discrecionalmente para la construcción de agentes inteligentes (Challenger, Díaz y Becerra, 2014).

Este lenguaje de programación fue creado con la finalidad de enfocar el esfuerzo del programador por encima del computacional. Le da más relevancia al entendimiento del código sobre la rapidez o elocuencia.

### **1.2. Visión artificial.**

Siempre ha sido, un par de cámaras que juegan el rol de los ojos, y una computadora que realizara la función de procesar las imágenes (Brahath y Petrov, 2008).

Este apartado está diseñado para tener una visión tipo perspectiva mejor a la humana, donde nosotros podremos encontrar patrones en cuerpos minúsculos. Aquí es donde necesitamos sensores más o igual de precisos que nuestros ojos, aunque la realidad es distinta, incluso se pueden ejecutar operaciones con imágenes insuficientemente detalladas, obteniéndose resultados no deseados.

#### **1.2.1. ¿Qué es reconocimiento facial?**

Es un proceso de sistemas que puede reconocer o autenticar a una persona a través de canales audiovisuales del rostro de una persona. En la mayoría de los casos, este método de identificación se utiliza para acceder a aplicaciones, sistemas o servicios (Osorio y Garcia, 2022).

### 1.2.2. ¿En qué consiste la biometría facial?

Mediante patrones matemáticos únicos y dinámicos del individuo, transforman a este método de acceso uno de los procedimientos más seguros (Osorio y Garcia, 2022).

Teniendo una imagen de entrada, se debe encontrar un conjunto de datos del mismo rostro en un almacén de imágenes de entrenamiento dentro de una base de datos o de manera local.

### 1.3. Inteligencia artificial.

#### 1.3.1. ¿Qué es la inteligencia artificial?

Son máquinas o sistemas que intentan simular la inteligencia del ser humano, para poder llevar a cabo tareas y que puedan evolucionar constantemente gracias a la repetición de trabajos (Rouhiainen, 2018).

#### 1.3.2. Tipos de inteligencia artificial.

**Aprendizaje automático.** Es la capacidad que tiene un programa o dispositivo de instruirse por su cuenta. Este tiene tres elementos principales: aprendizaje, entrenamiento y resultados (Sandoval, 2018).

**Aprendizaje Profundo (Deep Learning).** Abarca y gestiona una mayor cantidad de datos e información que el aprendizaje automático, al mismo tiempo. Este tipo de inteligencia artificial está muy relacionado con Big Data. Usos: traducir textos, reconocimiento facial, etc. (LeCun, Bengio y Hinton, 2021)

**Redes neuronales.** Las redes neuronales nacieron para imitar el funcionamiento de las neuronas de nuestro cerebro. Dado el caso, las redes neuronales siguen una tendencia diferente al enfoque clásico de inteligencia artificial, que tratan de adaptar la inteligencia en forma de razonamiento humano a un grupo de procesos (Lawrence, 1993).

**Funciones de activación.** Es una función que realiza el trabajo semejante a un filtro o un limitador, umbral, que cambia el valor del resultado o aplica un límite que se debe exceder para así poder continuar a una siguiente neurona la de salida.

Función escalón.

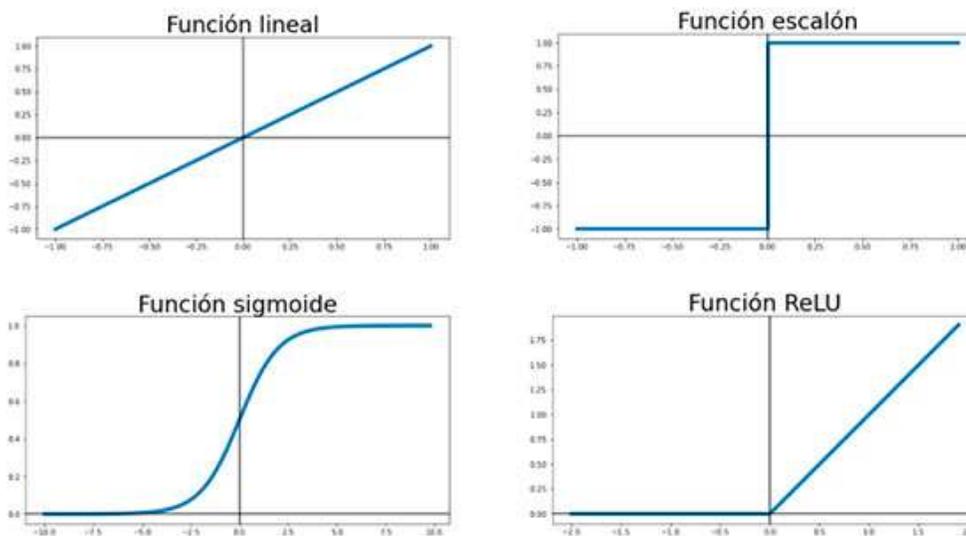
Función sigmoideal.

Función rectificadora (ReLU).

Función tangente hiperbólica.

**Figura 1.**

Funciones de activación para una red neuronal

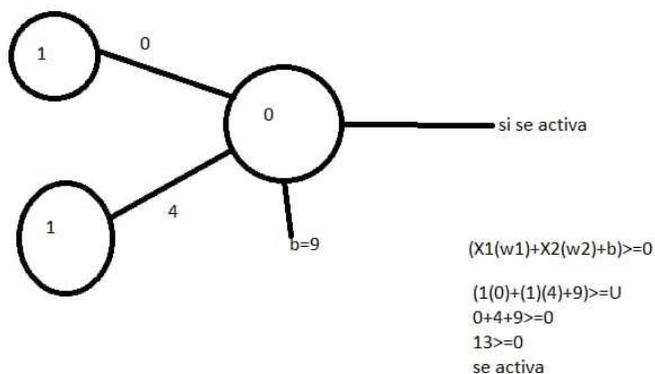


**Nota.** Adaptado de (Futurespace, 2018)

**Perceptrón simple.** Es el tipo de red neuronal más simple, está constituida por una capa de entrada que contiene cierta cantidad de neuronas, que arroja la entrada a la capa de salida en donde se ejecutan los cálculos pertinentes.

**Figura 2.**

Red Neuronal – Perceptrón Simple con cálculo de activación



**Nota.** Adaptado de (Futurespace, 2018)



**Redes neuronales recurrentes.** La categoría no tiene una conformación por capas, esta a su vez posibilita conexiones arbitrarias entre cada una de las neuronas, estas puedes realizar ciclos, para que la Red Neuronal pueda tener memoria (Tsoi, 2006).

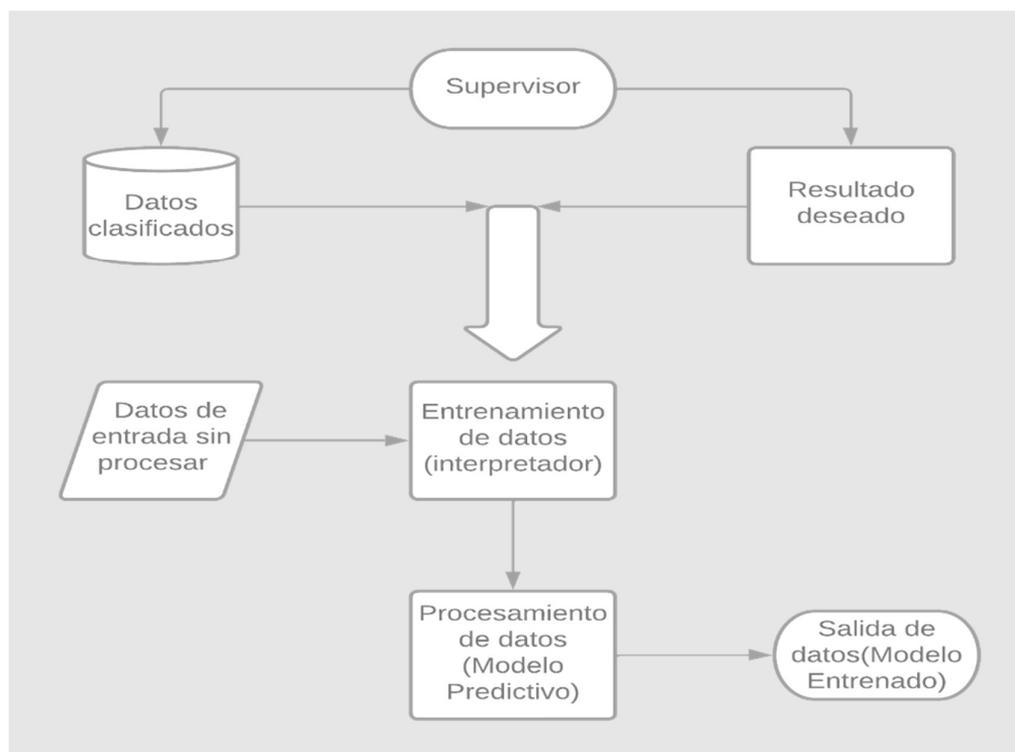
**Sistemas expertos.** Su funcionamiento se lleva a cabo mediante la lógica racional que busca simular a un ser humano que domina una materia en concreto.

### 1.3.3. Tipos de aprendizaje en inteligencia artificial

**Aprendizaje supervisado.** Este tipo de aprendizaje requiere una serie de datos clasificados, nosotros le damos al agente inteligente que es lo que tiene que aprender.

**Figura 5.**

Representación funcionamiento del aprendizaje supervisado

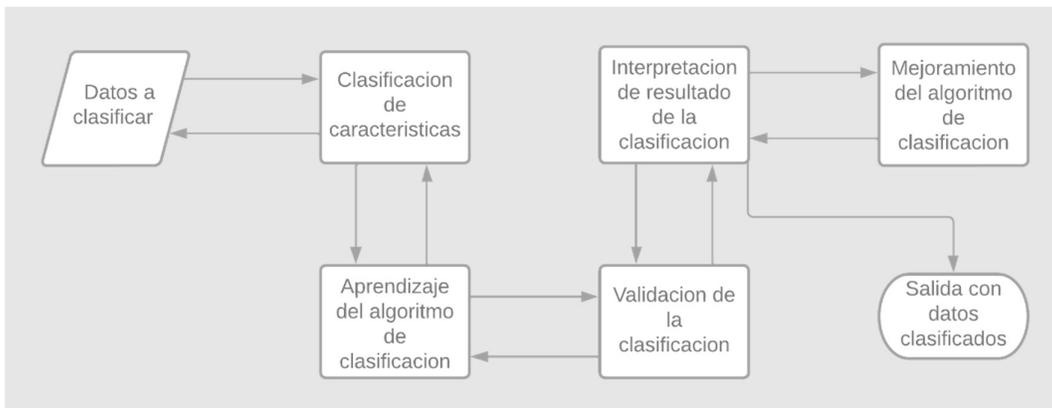


**Nota.** Adaptado de (Futurespace, 2018)

**Aprendizaje no supervisado.** Este modelo de aprendizaje se desempeña con datos que no han sido clasificados. Se utilizan en actividades donde es imprescindible analizar los datos para obtener nuevo conocimiento o hacer un conjunto de las entidades por afinidad.

**Figura 6.**

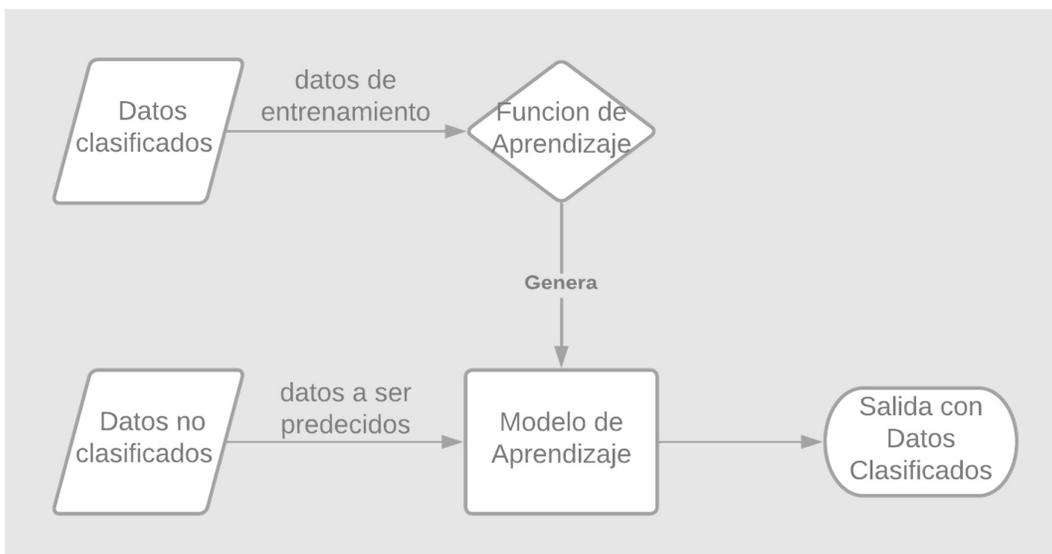
Representación del aprendizaje no supervisado

**Nota.** Adaptado de (Futurespace, 2018)

**Aprendizaje semi-supervisado.** Este tipo de modelo de aprendizaje contiene parte del aprendizaje supervisado ya que, con pocos atributos entrenados, se puede dejar al agente inteligente utilizar modelos derivados del entrenamiento para poder distinguir nuevos atributos.

**Figura 7.**

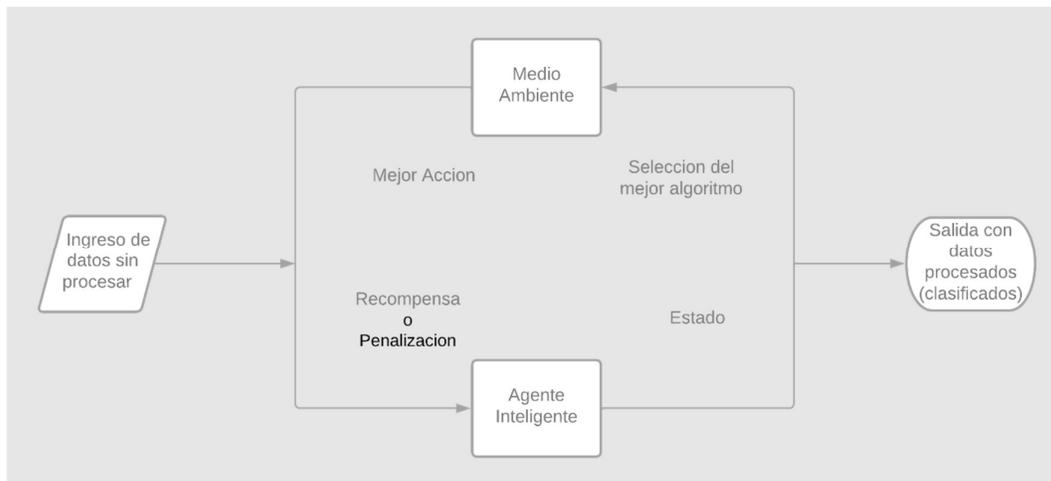
Representación del aprendizaje semi-supervisado

**Nota.** Adaptado de (Futurespace, 2018)

**Aprendizaje por refuerzo.** Este método de aprendizaje se basa en incentivar los comportamientos requeridos y castigar los no aprobados o no requeridos.

**Figura 8.**

Representación del Aprendizaje por refuerzo

**Nota.** Adaptado de (Futurespace, 2018)

### 1.3.4. HAAR Features.

Los HAAR features, son filtros o también conocidos como kernels que se obtiene de la sustracción entre la suma de los píxeles entre rectángulos de diverso color, por lo general monocromático, se usa más en el campo de detección de rostros (Martinez, 2020).

## 1.4. Herramientas.

### 1.4.1. Open CV.

Open Source Computer Vision es un casillero de programación de código libre hecha para permitir aplicaciones que usan visión artificial. Esta librería tiene muchas funciones tales como: captura, análisis y manipulación de datos visuales y también puede eliminar algunos de los grandes problemas que los programadores nos enfrentamos al momento de desarrollar aplicaciones que son dependientes de visión por computadoras (Open CV, 2019).

### 1.4.2. Librerías, modelos y bibliotecas usadas para la construcción del sistema.

**Dlib:** Es una librería/biblioteca hecha con el lenguaje de programación C++ que tiene algoritmos de machine learning y de visión por computadora que puede ser usado para desarrollo con el lenguaje de programación Python, en diferentes sistemas operativos. Detecta 68 puntos clave en el rostro.

**imutils:** Es un paquete que tiene como origen OpenCV, su función es traer a la interfaz de OpenCV de una forma más precisa, siendo capaz de ejecutar más fácilmente un conjunto de operaciones como: acercamiento, traducción, rotación de imágenes.

**Time:** Es una biblioteca que alberga un conjunto de funciones asociado con la medición temporal.

**Os:** Es un módulo que adapta con facilidad las funciones anexas al sistema operativo.

**Face\_recognition:** Es una librería de lenguaje de programación Python, que se usa para distinguir y alterar los rostros de una imagen, en esta librería tenemos inmersa la red neuronal convolucional de tipo ResNet (la normalización se realiza por lotes) haciendo uso de 27 capas, además del descriptor HOG (Histograma de Gradientes Orientados).

**Pickle:** Es un módulo que ejecuta protocolos binarios para serializar y deserializar una estructura de objetos tipo Python y la transforma a una cadena de bytes desde una jerarquía de objetos tipo Python (Pyspanish, 2017).

**Sklearn:** Es una librería del lenguaje de programación de Python que da apertura a versiones más eficientes de la mayoría de los algoritmos frecuentes (Data Scientist, 2019).

**Numpy:** Es una biblioteca de programación Python que se dirige expresamente al cálculo numérico y análisis de datos, por lo general se lo usa para grandes cantidades de datos.

**Matplotlib:** Es una librería que nos permite crear gráficos en dos dimensiones tales como:

Histogramas

Diagramas de barras

Diagramas de sectores

Diagramas de caja y bigote

Diagramas de dispersión o puntos

Diagramas de líneas

Mapas de color

Así mismo como la combinación entre todos estos diagramas (Aprende con ALF, 2020).

**Datetime:** Esta librería une los tipos de datos date, time y datetime para mostrar fechas y funciones para manipularlas.

**Seaborn:** Es una librería de Python que nos posibilita realizar de manera sencilla gráficos. Es dependiente de matplotlib y nos da una interfaz de alto nivel que es de fácil aprendizaje.

**Pandas:** Esta librería de Python se especializa en la utilización y análisis de estructuras de datos

**Math:** Es un módulo de Python que nos da paso a las funciones matemáticas dadas en el modelo de C.

## 1.5. Metodología.

### 1.5.1. Métodos.

El desarrollo del documento necesita seguir directrices metodológicas, así como técnicas y herramientas que permitan la recolección y análisis de la información necesaria para la presentación del proyecto de tesis, tales como:

**Método inductivo:** En relación con el análisis de las dificultades encontradas y sostenido en el método inductivo se pueden identificar preguntas generales de investigación.

**Método deductivo:** Una vez identificada la pregunta general de investigación. Este enfoque nos ayudó a definir los objetivos específicos a alcanzar para resolver los problemas encontrados en el desarrollo del prototipo.

**Método analítico:** Al obtener información sobre la situación actual de los docentes que se abstienen de calificar registro/asistencia en biometría, este método nos ayuda a determinar cuáles son las posibles soluciones al problema y elegir la mejor.

### 1.5.2. Metodología de desarrollo.

**Metodología Ágil Scrum.** El control sobre el estado de calidad del software se considera el objetivo principal y se da prioridad a las partes interesadas (Pérez A., 2011).

La figura 1 muestra la línea de procesos que debemos seguir para realizar un proyecto de la mejor manera utilizando la metodología ágil Scrum.

### Características:

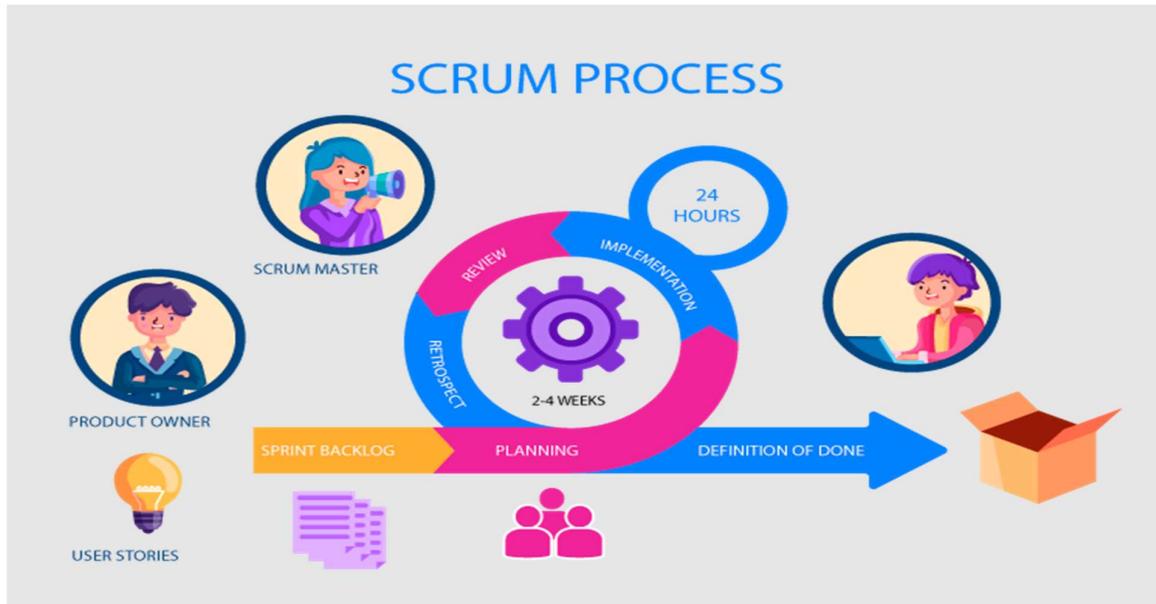
El enfoque principal de esta metodología ágil es generar software de la mejor calidad posible sin extenderse de gran manera en la documentación.

Se realizan y presentan soluciones aceptables hacia la parte interesada, no tan solo reportes, dado el caso la parte interesada podrá tener la última palabra sobre si continuar o dar por terminado el proyecto de acuerdo con el enfoque establecido.

Lo que la metodología ágil Scrum pretende, que el desarrollador y la parte interesada tengan cercanía, y dar oportunidad al cliente de tener seguimiento sobre el proyecto, de tal manera que este pueda aportar con su opinión, así mismo debemos tener la capacidad de generar cambios.

### Figura 9.

Proceso de Scrum



**Nota.** Adaptado de (Digite, 2021)

**Metodología Ágil Kanban.** Utiliza como método base la aplicación de JIT (Just Intime, Justo a Tiempo) (just intime, Justo a tiempo). La cual utiliza tarjetas post-it para poder seguir el proceso de producción.

Esta metodología tiene como objetivo en que el trabajo en proceso debe ser limitado hasta que se termine de manera correcta el bloque antecesor, o en su defecto ha sido dado a la función siguiente de la cadena (Maida & Pacienza, 2015).

**Características.**

- Calidad Garantizada.
- Reducción de Desperdicios.
- Mejora Continua.
- Flexibilidad.

Las tres reglas de Kanban.

- Mostrar el proceso.
- Limitar el trabajo en curso.
- Optimizar el flujo de trabajo.

Tablero Kanban.

- Cola de Entradas.
- Análisis.
- Desarrollo.
- Test.
- Entrega.

**Adaptative Software Development (ASD).** ASD tiene como particularidad que los clientes siempre suelen cambiar sus necesidades con respecto al modelo inicial de los proyectos e incluso ya en la fase de desarrollo, para luego ser implementados y entregados. Esta metodología en si no es una metodología, es una técnica la cual fomenta la adaptación de la empresa hacia los requerimientos de los clientes (Maida & Pacienza, 2015).

**Características.**

- Es adaptable.
- Guiado en base a riesgos.
- Iterativo.

**Ciclo de vida ASD.**

- Especular.

Se determina el enfoque del proyecto.

Se determinan los tiempos de entrega.

Se separa el proyecto por iteraciones y cuánto tiempo tomara realizar cada una.

Se da a conocer los objetivos de cada iteración.

Se da funcionalidad a cada iteración.

- Colaborar.

En esta fase el desarrollo se vuelve cíclico, ya que es de suma importancia la información que transmite el equipo de trabajo en base a lo aprendido.

- Aprender.

Control de calidad dada por el cliente.

Control de calidad dada por los desarrolladores.

Funcionalidad del producto, software en proceso desarrollada.

# **CAPITULO II: ANÁLISIS Y DISEÑO**

## **2. Investigación preliminar.**

### **2.1. Sistema de registro biométrico-dactilar UIDE.**

Desde la apertura de nuestra alma mater Universidad Internacional del Ecuador con sede en Loja, se ha utilizado el sistema tradicional de registro de asistencia docente y administrativo, el cual consiste en la biometría-dactilar, el usuario, docentes y personal administrativo, en nuestro caso, posan su mano sobre el lector dactilar del dispositivo, permitiéndose así la marcación y registro correcto de su ingreso y salida del establecimiento.

A inicios del año 2020, en la región ecuatoriana fuimos partícipes de una de las peores crisis sanitarias vividas, no solo en nuestro País, fue a nivel mundial, causada por el virus COVID-19, denominado estado pandémico. En dicho escenario la mayoría de los artefactos dactilares dejaron de usarse, por varios motivos, uno de los más importantes era que el virus podía ser adquirido mediante la manipulación de objetos sin la debida precaución.

La Universidad Internacional del Ecuador con sede en Loja no fue la excepción, debido a que durante 3 (tres) periodos académicos, las instalaciones universitarias estaban desoladas debido al confinamiento ordenado por los altos mandos del país, con el propósito de precautelar la salud de los ciudadanos, desde ese periodo hasta el día de hoy el sistema sigue sin uso.

### **2.2. Actualización del Sistema de control de asistencia de docentes del Área de Tecnología de la Información.**

Debido al distanciamiento social y al no poder realizar labores de desinfección del dispositivo biométrico.

La OMS dio a conocer un listado todos los artefactos que pueden ser ocupados, a continuación, en la figura 2 se detalla con exactitud la disposición de cómo mantener el área de trabajo desinfectado.

**Figura 10.**

Desinfección de elementos de trabajo



**Nota.** Adaptado de (WHO, 2017)

La sensibilidad hacia elementos líquidos por parte de los artefactos tecnológicos, se propuso la baja de aquel sistema de registro (dactilar). Ya que estos al momento de ser expuestos muchas veces a líquidos o humedad tienden a acelerar el fin de su vida útil.

## 2.3. Requerimientos

### 2.3.1. Requerimientos funcionales

Visualización por video de cada persona. Marcar asistencia/Ingreso y salida de los miembros de la Universidad Internacional sede Loja del Área de Tecnologías de la información, mediante el video. La interfaz del prototipo tiene que contener los colores pertenecientes a la Universidad Internacional sede Loja o un distintivo haciendo alusión a la misma. Los datos de ingreso y salida deben almacenarse en una base de datos, para luego ser exportada como emisión de reportes.

### 2.3.2. Requerimientos no funcionales

Base de datos que contenga las horas de ingreso y salida. Uso de inteligencia artificial para la construcción del modelo de aprendizaje, de la captura de imágenes por video. Uso de herramientas para la construcción (HTML, CSS...) de la interfaz.

### 2.3.3. Requerimientos de hardware para el uso del ambiente de desarrollo Anaconda Navigator

En este caso para la programación de la visión artificial usaremos el entorno de programación en Python, Anaconda Navigator, el cual nos permite usar cualquier herramienta y librerías de Python.

Versión de Python 2.7 o 3.5+

Procesador Corei5 o superior de 7ma generación o más reciente

Tarjeta de video NVidia o AMD 4Gb dedicados o superior

Estos requerimientos de hardware para la ejecución del programa se han dictaminado en cuanto a la experiencia obtenida en el transcurso de la realización del proyecto y que constituyen el mínimo de requisitos de un computador para la correcta manifestación del entorno de programación con sus diferentes herramientas.

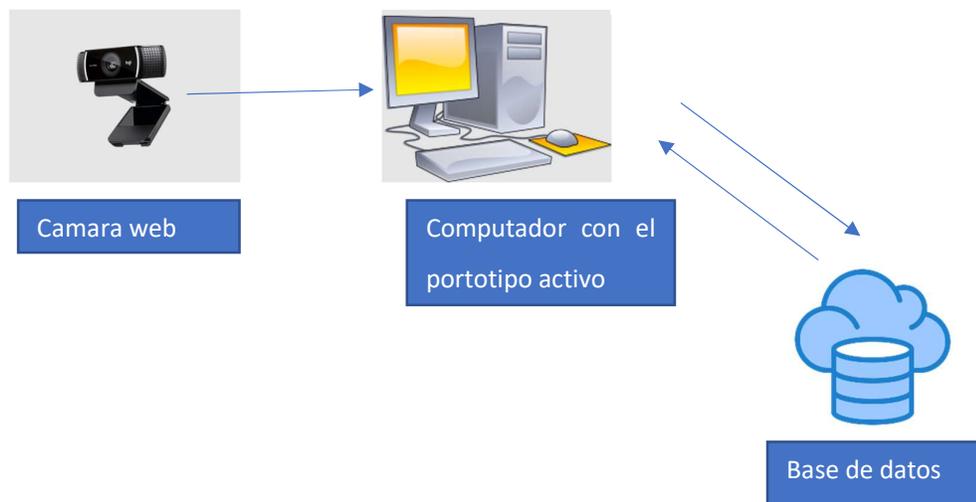
## 2.4. Modelado

### 2.4.1. Diagrama sistema simplificado

Se muestra el planteamiento simplificado del prototipo.

**Figura 11.**

Diagrama del prototipo simplificado



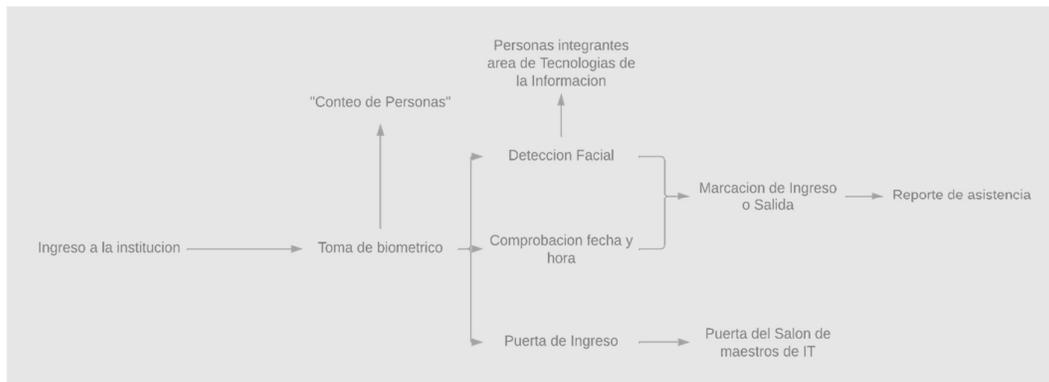
### 2.4.2. Esquema de funcionamiento

El presente esquema, figura 11, detalla el funcionamiento del prototipo de asistencia de

personal docente del área de Tecnologías de la información, planteado por la parte interesada, siguiendo los parámetros establecidos.

### Figura 12.

Esquema de funcionamiento completo del prototipo



## 2.5. Metodología para el desarrollo del proyecto

La metodología seleccionada que es más fiel a la construcción del presente proyecto de titulación es: Scrum, ya que es una metodología ágil, que a los programadores nos ayuda a visualizar de mejor manera, el cómo desarrollar un programa, ya que nos permite reducir el trabajo en partes entregables o iteraciones, llamadas Sprints.

Scrum es una metodología que, facilita al grupo de desarrollo aprender y organizarse de mejor manera, gracias a las experiencias adquiridas por cada uno de los integrantes dando la mano así a la reflexión inequívoca entre aciertos y fracasos. Todo esto bajo una serie de recursos y herramientas que permiten al grupo de desarrollo organizarse de manera mucho más eficiente, rápida y ágil.

El valor agregado de la metodología Scrum, permite el trabajo compartido entre el usuario y el desarrollador, con fines de mejorar al máximo posible la entrega del producto final.

## 2.6. Elementos Scrum

El sistema desarrollado es de pertenencia de la dirección de la carrera de Tecnologías de la Información, misma que dio a conocer los requerimientos funcionales y no funcionales para su implementación.

### 2.6.1. Lista de tareas o Product Backlog

Como usuario siento la necesidad que en el sistema debería tener un botón distintivo para el ingreso y la salida.

El sistema debería mostrar el reporte de asistencia diario.

El administrador debería poder ver el historial de reportes de entrada y salida.

El administrador deberá estar en la capacidad de agregar horarios de clase al usuario.

El administrador debería ser capaz de poder visualizar los horarios por materia.

El sistema deberá poder controlar los horarios de entrada y salida dando un plazo máximo de 10 minutos adicionales para cada operación.

### 2.6.2. Backlog Sprint

**Tabla 1**

Backlog Sprint.

Para hacerse	En proceso	Terminado
Definición de modelos de acuerdo con los requerimientos planteados. Y en dependencia a los usos del presente sistema.	Diseño de tablas de base de base de datos.	Construcción de los diferentes modelos requeridos.
Construcción en la vista principal botones capaces de ejecutar la verificación facial tanto para marcar entrada, como para marcar salida	Creación de las funciones que permitan la verificación facial y los modelos pertinentes para la misma.	Creación del Código que permite el reconocimiento facial
Construcción del Código que me permita visualizar las horas de entrada y salida	Definición del Modelo capaz de enviar los tipos de datos requeridos	Vista de reporte de en asistencias
Realizar la construcción de una vista donde se permita tanto al administrador como al usuario poder ver los reportes de asistencias	Realizar de manera adecuada, mediante la utilización de CSS y las diferentes etiquetas HTML, la vista requerida.	Codificación del algoritmo de marcación de asistencias

Crear una vista en donde el administrador pueda asignar horarios a las personas registradas	Realizar de manera adecuada, mediante la utilización de CSS y las diferentes etiquetas HTML, la vista requerida.	Vista creada para que el administrador pueda asignar horario a los usuarios
Desarrollar una vista en donde el administrador pueda visualizar los horarios por asignar	Realizar de manera adecuada, mediante la utilización de CSS y las diferentes etiquetas HTML, la vista requerida.	Vista creada que muestra los horarios registrados en la base de datos por materia
Creación de una función que permita el control de horarios asignados	Desarrollo del proceso de validación de horario	Codificación de una función capaz de controlar la asistencia en los horarios asignados
Desarrollo de la inteligencia artificial capaz de poder realizar la operación de reconocimiento facial	Mediante el uso del librerías y módulos de face_recognition, dlib y sklearn. Codificar los modelos de entrenamiento y predictivos.	Codificación del algoritmo capaz de realizar mediante el modelo de entrenamiento y predictivo la correcta validación de los datos.
Desarrollar el mecanismo por el cual se realizará el entrenamiento de los datos ingresados y los datos de aprendizaje	Mediante el uso del librerías y módulos de face_recognition, dlib y sklearn. Codificar los modelos de entrenamiento	Codificación del algoritmo capaz de realizar el modelo de entrenamiento de datos.

# **CAPITULO III: DESARROLLO**

### **3. Desarrollo del Sistema.**

#### **3.1. Antecedentes de la Marcación de asistencia docente del Área de Tecnologías de la información de la Universidad Internacional del Ecuador, sede Loja.**

La Organización Mundial de la Salud (OMS (Organización Mundial de la Salud)), declaro a la epidemia de Covid-19 originada en China- Wuhan, como emergencia de salud pública de preocupación internacional el 30 de enero del 2020. Ante este hecho mediante decreto ejecutivo Nro. 1017 el Estado declara estado de excepción por calamidad pública en el territorio ecuatoriano, por casos de coronavirus confirmados.

Por este acontecimiento varios de los dispositivos tecnológicos que el ser humano usaba cotidianamente se consideraban como un foco de infección inminente, teniendo que ser desinfectados constantemente para ser reutilizados, acortando su vida útil.

A causa de las restricciones planteadas por el COE cantonal de Loja, la Universidad Internacional del Ecuador, sede Loja, adoptó como medida preventiva el inhabilitar el uso del método biométrico de registro de asistencia de todo el personal del Alma Mater. Por lo antes mencionado se realiza el presente trabajo de titulación.

#### **3.2. Experimentación.**

Para el desarrollo de este sistema, se realizó un trabajo de investigación sobre diferentes métodos, herramientas y procesos, mediante los cuales se procedió a obtener el proceso de marcación de asistencia por medio del reconocimiento facial, a través de la aplicación de los diferentes módulos, librerías, bibliotecas, algoritmos en el Framework Django, que hace uso del lenguaje de programación Python.

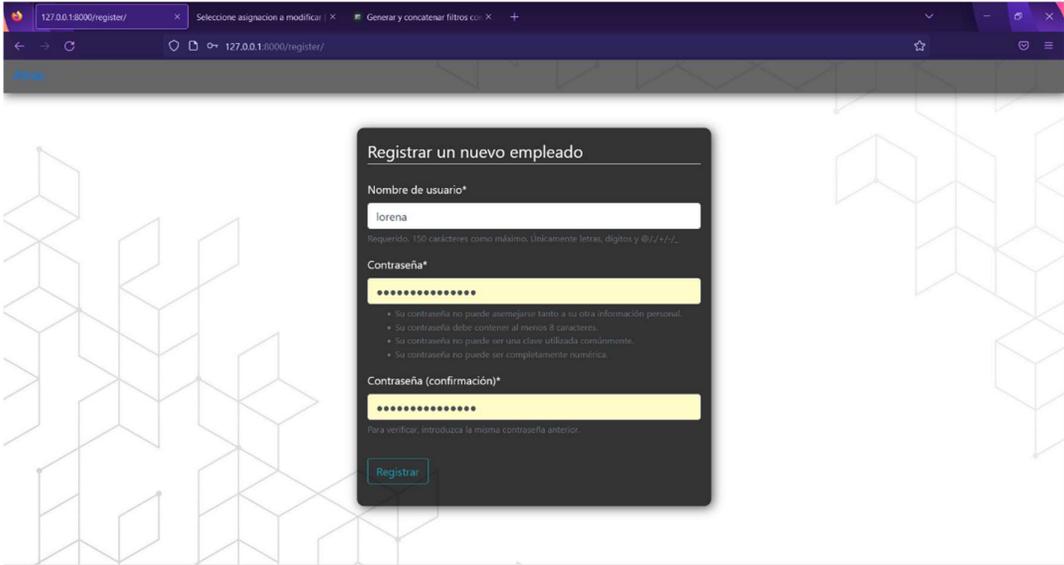
##### **3.2.1. Experimentación 1**

Para la realización de la prueba global del sistema de reconocimiento facial con la función de marcación de horario de entrada y salida, se contó con la participación de la Ing. Lorena Conde Ms. Directora de Carrera del Área de Tecnologías de la información y del Mgtr. Milton Palacios Director de Tesis del presente proyecto de titulación. Para lo cual tuvo que registrarse a cada usuario con un nombre de usuario en este caso "lorena" y "milton", además con una contraseña segura generada por el algoritmo automático que nos provee el navegador, para este caso se procedió a usar dos navegadores web, uno para la vista de la

interfaz home, el cual es, Microsoft Edge; y, para ver el apartado del administrador se usó el navegador Firefox, para así poder agilizar el proceso de captura, marcaje de entrada y salida, y revisión del reporte de asistencia.

**Figura 13.**

Interfaz de registro de un nuevo usuario.



The image shows a web browser window with a registration form. The browser's address bar shows the URL '127.0.0.1:8000/register/'. The form is titled 'Registrar un nuevo empleado' and contains the following fields and text:

- Nombre de usuario\***: A text input field containing 'loreana'. Below it, a note reads: 'Requerido, 150 caracteres como máximo. Únicamente letras, dígitos y @, /, +, -, \_'.
- Contraseña\***: A password input field with masked characters. Below it, a list of requirements is shown:
  - Su contraseña no puede asemejarse tanto a su otra información personal.
  - Su contraseña debe contener al menos 8 caracteres.
  - Su contraseña no puede ser una clave utilizada comúnmente.
  - Su contraseña no puede ser completamente numérica.
- Contraseña (confirmación)\***: A second password input field for confirmation, also masked.
- Below the confirmation field, a note reads: 'Para verificar, introduzca la misma contraseña anterior.'
- A blue 'Registrar' button is located at the bottom of the form.

En la figura 13, se exhibe el formulario de registro en donde los docentes tienen que insertar su nombre de usuario en un campo de texto y de contraseñas, una para insertar la contraseña deseada y la otra para realizar la confirmación de esta, creados por defecto llamando a la clase **UserCreationForm()**.

**Figura 14.**

Segmento del código perteneciente a la vista del formulario de registro.

```
Reconocimiento_Facial > users > views.py > ...
1  from django.shortcuts import render, redirect
2  from django.contrib.auth.forms import UserCreationForm
3  from django.contrib import messages
4  from django.contrib.auth.decorators import login_required
5  from django.contrib.auth.models import User
6
7
8  @login_required
9  def register(request):
10     if request.user.username != 'admin':
11         return redirect('not-authorized')
12     if request.method == 'POST':
13         form = UserCreationForm(request.POST)
14         if form.is_valid():
15             form.save() # add user to database
16             messages.success(
17                 request, f'Empleado registrado satisfactoriamente')
18             return redirect('dashboard')
19     else:
20         form = UserCreationForm()
21         return render(request, 'users/register.html', {'form': form})
22
23
```

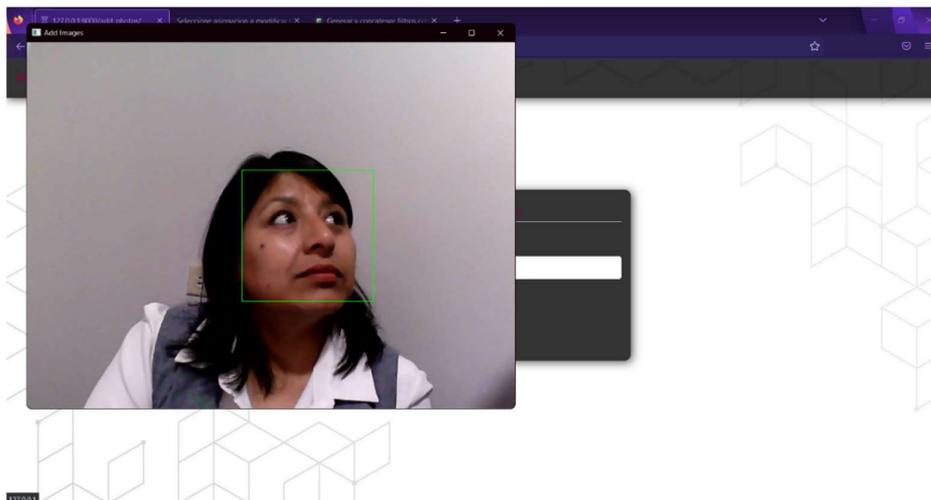
Para poder llamar a la clase **UserCreationForm()**, primero debemos importar el sistema de autenticación de **Django** al cual lo importamos de la siguiente manera:

```
from django.contrib.auth.forms import UserCreationForm
from django.contrib.auth.decorators import login_required
```

Con la importación del decorador **login\_required**, se puede restringir el acceso a la vista, ya que el usuario tiene que estar logeado para que pueda presentarse la vista.

**Figura 15.**

El sistema realiza la toma de datos, imágenes de los usuarios.



El administrador para crear el **dataset** tiene que desde la interfaz principal ir al botón de añadir fotos y posteriormente ingresar el nombre de usuario de la persona a registrar sus fotografías en el **dataset**. Para dar paso al procedimiento, se debe clicar en el botón etiquetado “**Enviar**”, esto desplegará una pantalla auxiliar para que el administrador pueda confirmar que efectivamente se está capturando el rostro de la persona, nos podemos percatar que existe un marco que siempre se va a centrar alrededor del rostro de la persona.

Una vez concluida la iteración de creación del **dataset**, la ventana se cerrará automáticamente y nos devolverá hacia la interfaz de añadir una foto, con un mensaje que nos dirá “Dataset creado”.

**Figura 16.**

Con diferente usuario se detalla la creación satisfactoria del dataset.

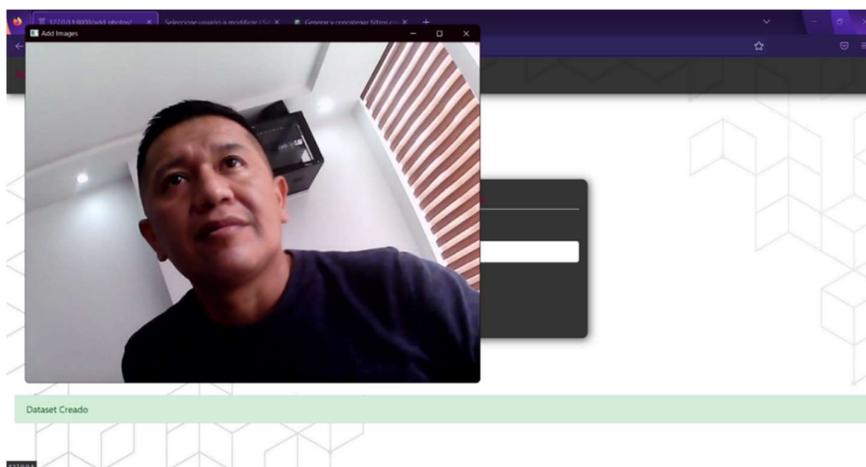


Figura 17.

Sección de código de la creación del dataset.

```

52 def create_dataset(username):
53     id = username
54     if (os.path.exists('face_recognition_data/training_dataset/{}/'.format(id)) == False):
55         os.makedirs('face_recognition_data/training_dataset/{}/'.format(id))
56         directory = 'face_recognition_data/training_dataset/{}/'.format(id)
57
58         # Detección facial
59         # Carga del HOG de la detección facial y del modelo predictor
60
61         print("[INFO] Cargando el detector Facial")
62         detector = dlib.get_frontal_face_detector()
63         # Se añade el predictor
64         predictor = dlib.shape_predictor(
65             'face_recognition_data/shape_predictor_68_face_landmarks.dat')
66         fa = FaceAligner(predictor, desiredFaceWidth=96)
67         # captura imagenes desde la camara, las procesa y detecta los rostros
68         # Inicia la captura por video
69         print("[INFO] Inicializando Video Captura")
70         vs = VideoStream(src=1).start()
71         # Con src podemos cambiar el dispositivo de entrada 0 para una camara unica y
72         # n valores dependiendo del numero de la camara a usarse
73
74         # El identificador
75         #Aqui se pondra el id y se guardara el rostro de acuerdo al id
76
77         # Contador del etiquetado del dataset
78         sampleNum = 0
79         # Capturando los rostros uno por uno y la detección de rostros mientras se muestra en una ventana
80         while (True):
81             # Capturando imagen
82             # vs para cada frame
83             frame = vs.read()
84             # redimensionado de la imagen
85             frame = imutils.resize(frame, width=800)
86             #la imagen que se retorna por camara es a color, para el clasificador se necesita una imagen a escala a grises
87             # para convertir de color a escala a grises
88             gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
89             # para guardar los rostros
90             # Se detectaran todas las imagenes en el frame y estas retornaran las coordenadas de los rostros.
91             # Se toma una imagen y un parametro para medir el resultado
92             faces = detector(gray_frame, 0)

```

La función **create\_dataset** con el parámetro **username**, va a crear el **dataset** en base al nombre de usuario registrado, realizamos la acción de asignar el nombre de usuario a una variable **id**. Según la lógica, si en el ordenador no se encuentra el directorio con el **id**, el programa mediante la llamada a las funcionalidades del sistema operativo procederá a crear una carpeta con el **id** del usuario en cuestión, en la ruta o directorio deseado.

Luego asignaremos la dirección de la carpeta creada a una variable llamada **"directory"**. Después procederemos a asignar a la variable "detector" la función de **dlib.get\_frontal\_face\_detector()**. Seguidamente a la variable "predictor" le asignamos la función **dlib.shape\_predictor()**, pero tenemos que agregar como parámetro el módulo de predicción con 68 puntos de referencia faciales el nombre del modelo es

“shape\_predictor\_68\_face\_landmarks.dat”. -Es indispensable para la correcta utilización de las librerías mencionadas en el capítulo primero-.

La función que nos permitirá identificar estructuras geométricas sobre los rostros en imágenes digitales es **FaceAligner(predictor, desiredFaceWidth=96)**, el cual lleva como parámetros el predictor y el ancho de cara ideal para la correcta alineación del rostro basada en la traslación, escala y rotación, esta función se albergara en la variable “fa”.

Enseguida se asignará la video captura con la función **VideoStream(src=1).start()**, teniendo como parámetro “src” que dentro de esta función significa la cámara que estaremos usando, para su ejecución se realizará la llamada al método **start()**, en la variable “vs”.

Procedemos a inicializar una variable llamada “sampleNum” con un valor de cero (0). Iniciamos un bucle repetitivo **while** que se va a ejecutar mientras la condición sea verdadera, en este caso la condición está declarada como **True**, que nos servirá para ir capturando los rostros uno por uno, mientras la detección de rostros se visualiza. Dentro del bucle repetitivo **while**, capturaremos las imágenes con el método **vs.read()**, este se albergara en una variable llamada “frame”, a continuación a la imagen capturada la redimensionaremos con el siguiente método, **imutils.resize(frame, width=800)**, para poder redimensionar la imagen capturada debemos especificar la imagen capturada y un ancho, y a esta la volveremos asignar a la variable “frame”, para después convertir esta imagen a escala de grises con la ayuda de la función **cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)**, usando como parámetros la imagen ya redimensionada y la configuración de color deseada, para este caso utilizaremos el módulo **cv2** y la constante **COLOR\_BGR2GRAY**.

Al detector le asignamos como parámetros la imagen convertida a escala de grises en la cual nosotros necesitamos realizar el reconocimiento facial y, el sobre muestreo que es opcional y es el responsable de sobre muestrear la imagen entrante antes de ejecutar el reconocimiento facial dado este caso este parámetro llevara el valor de cero (0).

Figura 18.

Continuación de la sección de Código para la creación de dataset.

```

93 # dentro de la variable 'faces' puede haber multiples rostros entonces tenemos que capturar cada uno de ellos
94 # y dibujar un rectangulo alrededor
95
96 for face in faces:
97     print("iniciando ")
98     (x, y, w, h) = face_utils.rect_to_bb(face)
99
100     face_aligned = fa.align(frame, gray_frame, face)
101     # Siempre que el programa capture rostros, los escribira como si fueran una carpeta
102     # Antes de capturar los rostros, necesitamos hacerle saber al script de quien es cada rostro
103     # Para eo necesitamos un identificador
104     # So now we captured a face, we need to write it in a file
105     #ahora que se ha capturado un rostro, necesitamos transformalo en un archivo
106     sampleNum = sampleNum+1
107     # Se guarda la imagen en el dataset, pero solo el rostro, cortando el resto de la captura de imagen
108
109     if face is None:
110         print("no se identifican rostros")
111         continue
112
113     cv2.imwrite(directory+'/' +str(sampleNum)+'.jpg', face_aligned)
114
115     face_aligned = imutils.resize(face_aligned, width=400)
116     # @params el punto inicial del rectangulo sera x e y
117     # @params el punto final sera x+width e y+height
118     # @params conjunto va el color del marco
119     # @params espesor del marco
120     cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 1)
121     cv2.waitKey(50)
122
123     # Se muestra la imagen en otra ventana
124     # Se crea una ventana con el nombre Rostro y con una imagen img
125     cv2.imshow("Anadir Imagenes", frame)
126     # Antes de cerrar la ventana necesitamos dar un comando de espera, por que si no open cv no funcionara
127     # @params una millonesima de segundo en un segundo
128     cv2.waitKey(1)
129     # Para salir del bucle
130     if (sampleNum > 150):
131         break
132
133 # Interrupcion del video
134 vs.stop()
135 # eliminacion de todas las ventanas
136 cv2.destroyAllWindows()

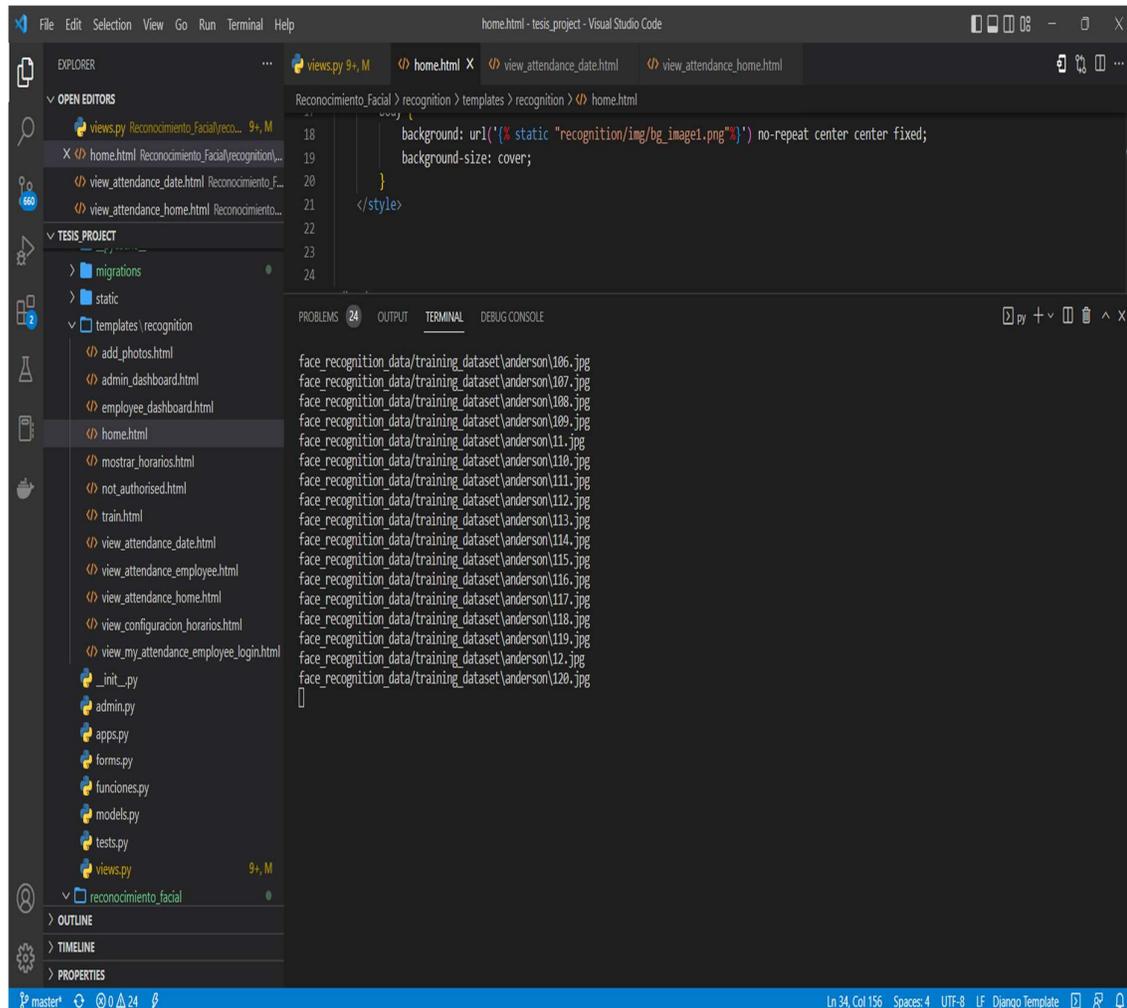
```

Mediante un ciclo repetitivo **“for”** inmerso en el ciclo repetitivo **“while”**, con la condición de que **“face”** dentro de **“faces”**, se procederá a identificar cada rostro con un marco y este será asignado a la variable **“face\_aligned”** que tiene como parámetros nuestra imagen reescalada, la imagen convertida en blanco y negro y los vectores pertenecientes a cada rostro. Seguidamente se hace una operación de adición en donde el contador de etiquetado del **dataset** se incrementará en uno (+1), en caso de que **“face”** no exista se imprimirá en el terminal un mensaje de advertencia, con la instrucción `continue` nos da la opción de omitir un segmento del bucle cuando se ingresa en una condición externa. Se procederá a guardar cada rostro en una ruta específica de nuestra elección.

El bucle terminará cuando la variable **“sampleNum”** supere las 150 iteraciones, ejecutándose la instrucción **break**, se procederá a la interrupción del video con la función **vs.stop()**, y también se cerrarán todas las ventanas creadas con **cv2**, con la función **cv2.destroyAllWindows()**.

**Figura 19.**

Entrenamiento de datos



En la figura18 podemos observar que el algoritmo toma todos los datos de tipo imagen en nuestro directorio, para así mediante el modelo de entrenamiento poder aprender que rostro lleva cada nombre de usuario.

**Figura 20.**

Segmento del modelo de entrenamiento de datos

```
786  
787 @login_required  
788 def train(request):  
789     if request.user.username != 'admin':  
790         return redirect('not-authorized')  
791  
792     training_dir = 'face_recognition_data/training_dataset'  
793  
794     count = 0  
795     for person_name in os.listdir(training_dir):  
796         curr_directory = os.path.join(training_dir, person_name)  
797         if not os.path.isdir(curr_directory):  
798             continue  
799         for imagefile in image_files_in_folder(curr_directory):  
800             count += 1  
801  
802     X = []  
803     y = []  
804     i = 0  
805  
806     for person_name in os.listdir(training_dir):  
807         print(str(person_name))  
808         curr_directory = os.path.join(training_dir, person_name)  
809         if not os.path.isdir(curr_directory):  
810             continue  
811         for imagefile in image_files_in_folder(curr_directory):  
812             print(str(imagefile))  
813             image = cv2.imread(imagefile)  
814             try:  
815                 X.append((face_recognition.face_encodings(image)[0]).tolist())  
816
```

**Figura 21.**

Segmento final del modelo de entrenamiento.

```

817         y.append(person_name)
818         i += 1
819     except:
820         print("removido")
821         os.remove(imagefile)
822
823     targets = np.array(y)
824     encoder = LabelEncoder()
825     encoder.fit(y)
826     y = encoder.transform(y)
827     X1 = np.array(X)
828     print("shape: " + str(X1.shape))
829     np.save('face_recognition_data/classes.npy', encoder.classes_)
830     svc = SVC(kernel='linear', probability=True)
831     svc.fit(X1, y)
832     svc_save_path = "face_recognition_data/svc.sav"
833     with open(svc_save_path, 'wb') as f:
834         pickle.dump(svc, f)
835
836     vizualize_Data(X1, targets)
837
838     messages.success(request, f'Entrenamiento completado')
839
840     return render(request, "recognition/train.html")
841

```

Para poder realizar el entrenamiento de los datos el usuario activo tiene que ser administrador o superusuario, vamos a albergar la dirección de los datos de entrenamiento a la variable **training\_dir**. Inicializaremos una variable **"count"** en cero (0),

Realizaremos un bucle repetitivo principal **"for"** con la condición de que cada **person\_name** exista en la dirección donde yacen los datos de entrenamiento, mediante el método **os.path.join()**, uniremos los siguientes componentes: La dirección de datos de entrenamiento y el nombre del usuario a entrenar, y este será asignado a la variable **curr\_directory**.

Crearemos otro ciclo repetitivo secundario **"for"** perteneciente al anterior, con la condición de que exista una imagen dentro de la variable **curr\_directory**, entonces el contador se incrementara en uno (+1).

Después de esto inicializaremos dos **vectores vacíos (X Y)** y una variable **"i"** en cero.

Generaremos otro ciclo repetitivo **“for”** que tiene como condición que, por cada nombre de usuario en la dirección de datos de entrenamiento, se realizará lo explicado en el ciclo repetitivo **“for”** principal anterior, a diferencia que en el ciclo repetitivo secundario se cambiará la operación de incremento por una lectura de imagen dada por la función **cv2.imread(imgfile)** que será asignado a una variable **image**. Definiremos un **bloque try**: que ejecutara en el vector **X** la codificación de cada imagen transformándola en una lista, y en el vector **Y** se agregaran los nombres de usuario y nuestra variable **“i”** se incrementara en uno (**+1**); **except**: que realizara la impresión por terminar del mensaje removido y mediante la función **os.remove(imgfile)** eliminara la imagen no etiquetada en el directorio.

Con la ayuda de la función **np.array()** podemos transformar listas y vectores en un array, como es el caso del vector **Y**, a su vez asignándolo en la variable **targets**. Luego destinaremos la función **LabelEncoder()**, a la variable **“encoder”**, para luego hacer que esta variable **“encoder”** se entrene mediante la función **fit** para extraer los parámetros que se usaran sobre los datos de prueba con la función **predict()**, mediante el método **encoder.transform()**, transformara a las etiquetas del arreglo **Y**, a nuevas etiquetas numéricas, para que estas puedan ser interpretadas.

Transformaremos el vector **X** en **array**, siguiendo el proceso antes realizado para **Y**, y se asignara la variable **X1**, con la función **np.save** guardaremos una matriz en un archivo tipo **NumPy** de carácter binario, y como parámetros tenemos el archivo **.npy** y como arreglo, todas las etiquetas que contiene la variable **“encoder”** para cada clase.

Con SVC clasificaremos los datos con el uso de un **kernel='linear'**, **probability=True**, y se albergara en la variable **svc**. Con la clasificación dispuesta procedemos a entrenar los datos con la función **svc.fit(X1, Y)**. A continuación, asignaremos la ruta de un archivo tipo **sav** (son archivos genéricos de almacenamiento), a la variable **svc\_save\_path**

Mediante la sentencia **with** con la expresión **open(svc\_save\_path, 'wb')** as **f**, se realizará con la función **pickle.dump(svc, f)** la **serialización** de los datos que contiene el objeto **svc** y el archivo donde se escribirá **“f”**.



Figura 24.

Sección de código marcación de asistencia

```

587 def mark_your_attendance(request):
588
589     detector = dlib.get_frontal_face_detector()
590
591     # Anade el predictor con los 68 puntos principales del rostro
592     predictor = dlib.shape_predictor(
593         'face_recognition_data/shape_predictor_68_face_landmarks.dat')
594     svc_save_path = "face_recognition_data/svc.sav"
595
596     with open(svc_save_path, 'rb') as f:
597         svc = pickle.load(f)
598     fa = FaceAligner(predictor, desiredFaceWidth=96)
599     encoder = LabelEncoder()
600     encoder.classes_ = np.load('face_recognition_data/classes.npy')
601
602     faces_encodings = np.zeros((1, 128))
603     no_of_faces = len(svc.predict_proba(faces_encodings)[0])
604     count = dict()
605     present = dict()
606     log_time = dict()
607     start = dict()
608     for i in range(no_of_faces):
609         count[encoder.inverse_transform([i])[0]] = 0
610         present[encoder.inverse_transform([i])[0]] = False
611
612     vs = VideoStream(src=1).start()
613

```

Definimos la función **mark\_your\_attendance(request)**, que contendrá el código capaz de realizar el reconocimiento facial conjunto a la marcación de asistencia de entrada para los docentes.

Primero, mediante la función **get\_frontal\_face\_detector()**, de la clase **dlib**, se asignara a la variable “**detector**”, a continuación mediante la función **shape\_predictor()**, cabe resaltar que debemos tener en un directorio deseado el módulo predictor **shape\_predictor\_68\_face\_landmarks.dat**, que nos ayudara a marcar la imagen de entrada con 68 puntos de referencia en el rostro, para luego este ser llamado gracias a la función antes mencionada, esta también perteneciente a la clase **dlib**, será asignada a la variable “**predictor**”.

Tendremos almacenado la ubicación del archivo tipo **sav**, en la variable **svc\_save\_path**. A través de la sentencia **with** realizaremos la acción de abrir el archivo anteriormente mencionado albergado en la variable **svc\_save\_path**, en modo solo lectura en formato binario, y este podrá ser llamado como “**f**”.

Seguidamente, con la función **pickle.load(f)**, asignándole como parámetro el archivo de tipo **sav** en modo de lectura en formato binario antes señalado como **"f"**, en la variable **"svc"**.

A continuación, en la variable **"fa"** dispondremos de la función **FaceAligner()**, dándole como parámetro el **"predictor"**, y el ancho de rostro deseado como numero entero asignado a una variable, **"desiredFaceWidth=96"**.

Luego se realizará la asignación de la función **LabelEncoder()**, en la variable **"encoder"**, esta variable añadiéndole el atributo **"encoder.classes\_ "**, recuperará todas las clases o etiquetas, desde la función **np.load()**, que contendrá el archivo con las clases y es de tipo **numpy**.

Con la función **np.zeros((1, 128))**, esta nos ayudara con la creación y devolución de una referencia a un arreglo con especificadas dimensiones en la tupla dimensiones que sus elementos son cero (0), esta se albergara en la variable **faces\_encodings**.

Por medio de la variable **"no\_of\_faces"**, que tiene asignado la longitud de la variable **svc** que con el método **predict\_proba(faces\_encodings)**, asignándole como parámetro la variable **face encodings**.

Definiremos cuatro diccionarios: **count**, **present**, **log\_time** y **start**. Proseguimos con la creación de un bucle repetitivo **for** que realizara, por medio de la siguiente condición, por cada **"i"** dentro del rango de la variable **"no\_of\_faces"**, **count[encoder.inverse\_transform([i])[0]]=0**; **present[encoder.inverse\_transform([i])[0]] = False**

**Figura 25.**

Sección de Código marcación de asistencia

```

614     sampleNum = 0
615
616     while (True):
617
618         frame = vs.read()
619
620         frame = imutils.resize(frame, width=800)
621
622         gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
623
624         faces = detector(gray_frame, 0)
625
626         for face in faces:
627             print("INFO : dentro del bucle ")
628             (x, y, w, h) = face_utils.rect_to_bb(face)
629
630             face_aligned = fa.align(frame, gray_frame, face)
631             cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 1)
632
633             (pred, prob) = predict(face_aligned, svc)
634

```

Realizaremos el procedimiento de captura antes mencionado. En el bucle repetitivo de la línea 625 añadiremos la predicción y probabilidad que tendrán la predicción dada por la función **predict()**, que tendrá como parámetros **face\_aligned, svc**.

Entonces generaremos una sentencia **if** la cual tendrá como condición que la predicción no tendrá que contener un índice negativo **if (pred != [-1])**.

Se realizará la transformación inversa de la matriz aplanada de la variable a predecir **“pred”**, mediante la utilización del codificador **“encoder”**, que se asignará como la variable **person\_name**, después se reasignará a **“pred”**, **person\_name**.

Figura 26.

Sección de Código marcación de asistencia

```

635     if (pred != [-1]):
636         person_name = encoder.inverse_transform(np.ravel([pred]))[0]
637         pred = person_name
638         if count[pred] == 0:
639             start[pred] = time.time()
640
641             count[pred] = count.get(pred, 0) + 1
642
643
644         if count[pred] == 4 and (time.time()-start[pred]) > 1.2:
645             count[pred] = 0
646         else:
647             present[pred] = True
648             log_time[pred] = datetime.datetime.now()
649             count[pred] = count.get(pred, 0) + 1
650             print(pred, present[pred], count[pred])
651             cv2.putText(frame, str(person_name) + str(prob), (x+6,
652                 y+h-6), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 1)
653
654         else:
655             person_name = "desconocido"
656             cv2.putText(frame, str(person_name), (x+6, y+h-6),
657                 cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 1)
658

```

Generará una condición **if** , que realizara que el contador de predicciones tiene que ser igual a cero(0), para asignarle tiempo a la variable **start** de predicciones con la biblioteca **time** que llamara a su clase **time()**, y el contador de predicciones aumentara en uno (+1), caso contrario si el diccionario booleano de las predicciones es verdadero asignaremos a **“log\_time”** de las predicciones tiempo actual y el contador de predicciones aumentara en uno (+1), y en la ventana se procederá a mostrar la etiqueta con el nombre de usuario y la probabilidad que este tiene con el rostro de entrada., caso contrario este desplegara como etiqueta que es desconocido.

Figura 27.

Sección final de Código marcación de asistencia

```

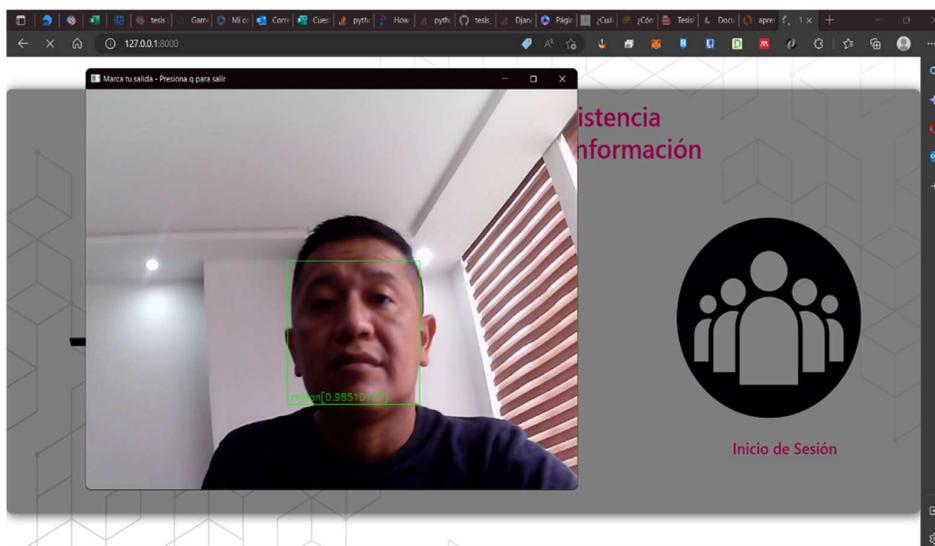
658
659     # Mostrando la imagen en una ventana
660     # Se crea una ventana con el nombre de la accion y la accion necesaria para salir
661     cv2.imshow("Marca tu entrada - Presiona q para salir", frame)
662     # Antes de cerrar se necesita dar un comando de espera, caso contrario opencv no funcionara
663     # @params with the millisecond of delay 1
664     # @params con los millisegundos en un delay de 1
665     # para salir del bucle
666     key = cv2.waitKey(50) & 0xFF
667     if (key == ord("q")):
668         break
669
670 # Para el stream
671 vs.stop()
672
673 # Se sale de todas las ventanas
674 cv2.destroyAllWindows()
675 print("#####")
676 print(present)
677 update_attendance_in_db_in(present)
678
679 return redirect('home')
680
681

```

Para salir de la ventana que devuelve y nos muestra el proceso de captura de imagen tenemos que presionar la tecla **q**, una vez acabado las sentencias de condición se para la video captura y se destruirán las ventanas, y se procederá a cargar esos datos a la siguiente función con parámetro **“present”** de tipo booleano, **update\_attendance\_in\_db\_in(present)**. Y nos retornara hacia la vista de home.

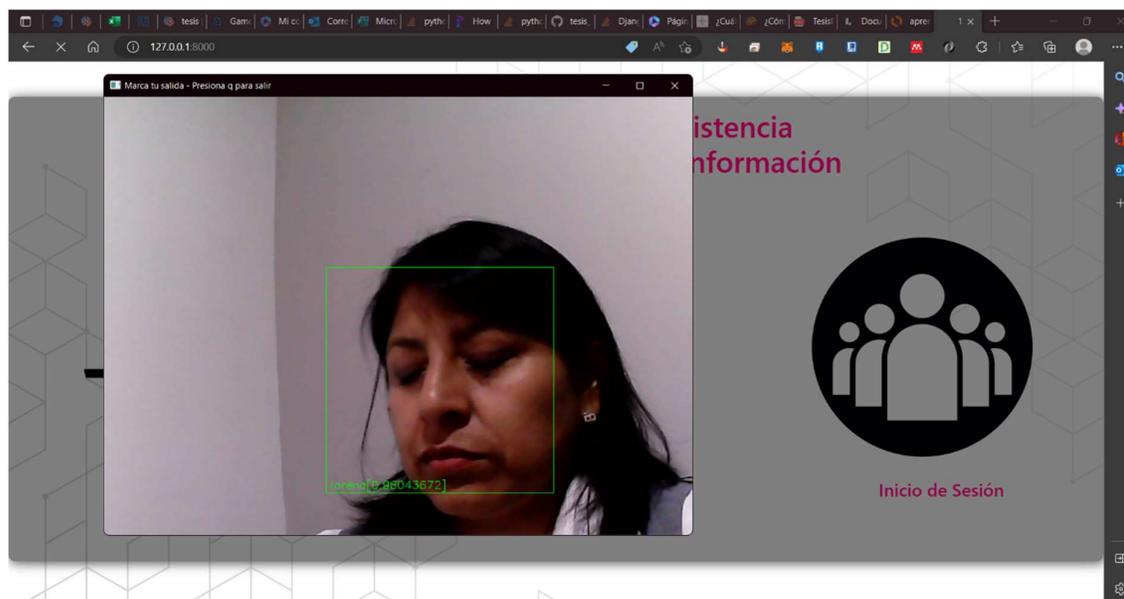
Figura 28.

Marcaje de salida del usuario “milton”.



**Figura 29.**

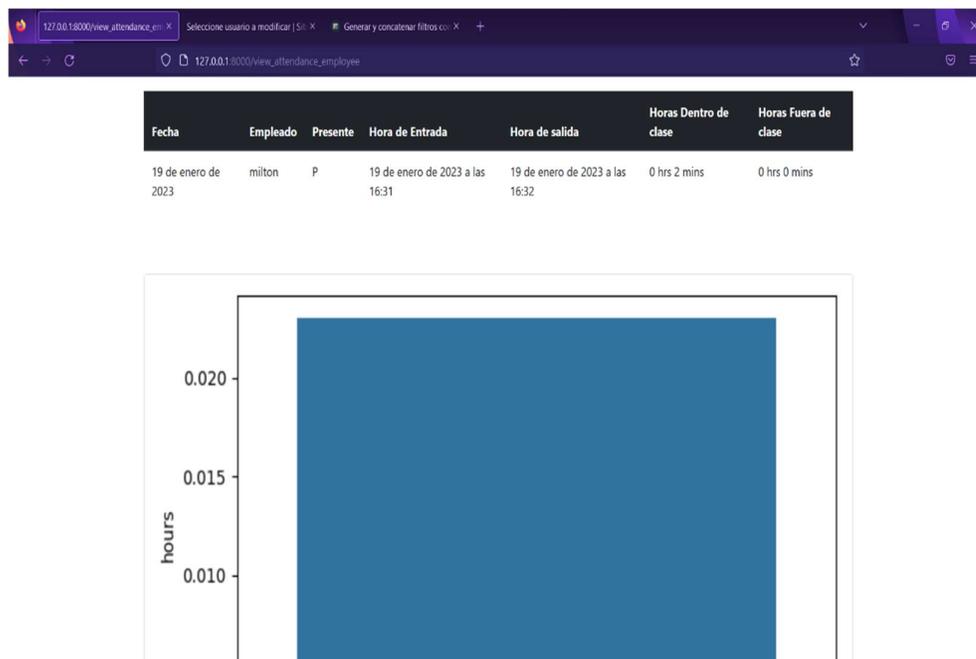
Marcaje de salida del usuario “lorena”.



Para el proceso de marcación de salida se realiza el esquema de código como si fuera una marcación de entrada porque el “predictor” no puede devolver valores negativos de una persona que si existe. La única distinción es que nos llevara a una función diferente que validara la salida de la persona y esta se define **update\_attendance\_in\_db\_out(present)**.

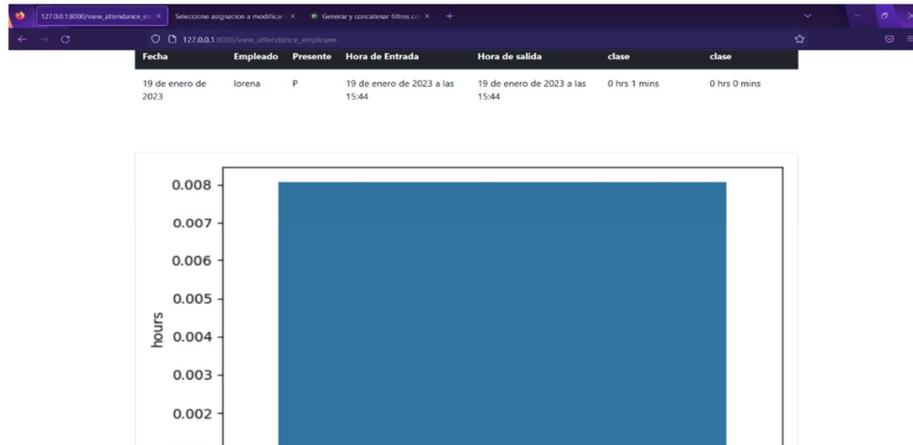
**Figura 30.**

Representación del marcaje de asistencia valido para el usuario “milton”.



**Figura 31.**

Representación del marcate de asistencia valido para el usuario "lorena".



Para realizar las vistas del gráfico y la tabla utilizamos librerías como: **matplotlib**, **datetime**, **seaborn**.

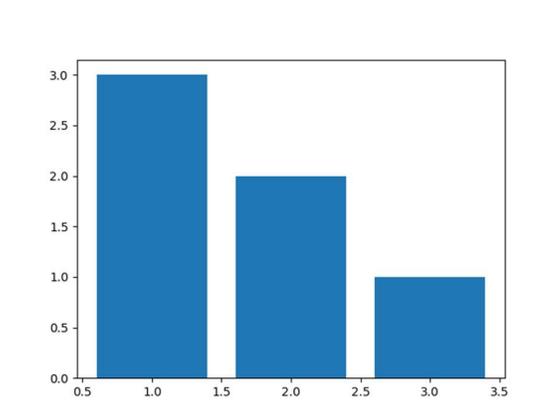
Con la librería **matplotlib** podemos realizar diagrama de barras como se mencionó anteriormente en el primer capítulo y su correcta utilización para esta clase de diagramas está definida por las siguientes líneas de código:

```
import matplotlib.pyplot as plt
fig, ax = plt.subplots()
ax.bar([1, 2, 3], [3, 2, 1])
plt.show()
```

Estas a su vez nos dan como resultado la figura que se mostrara a continuación

**Figura 32.**

Gráfico de barras realizado con matplotlib



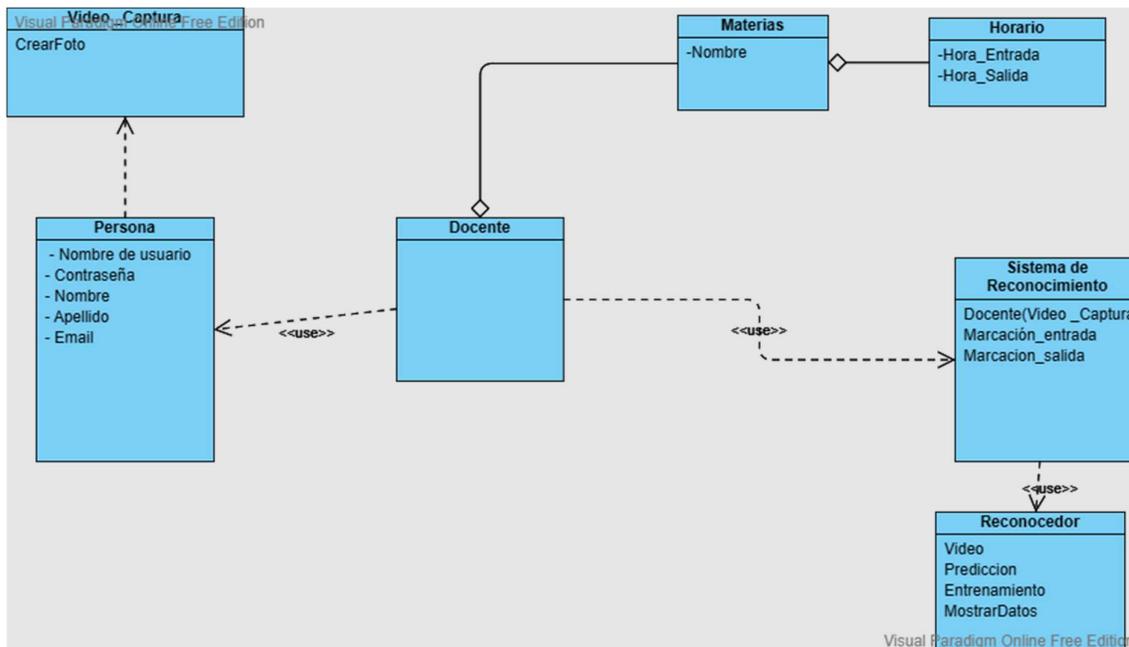
**Nota.** Adaptado de (Aprende con ALF, 2020)

### 3.3. Sprints

**Sprint 1: Definición de clases de acuerdo con los requerimientos planteados. Y en dependencia a los usos del presente sistema.**

**Figura 33.**

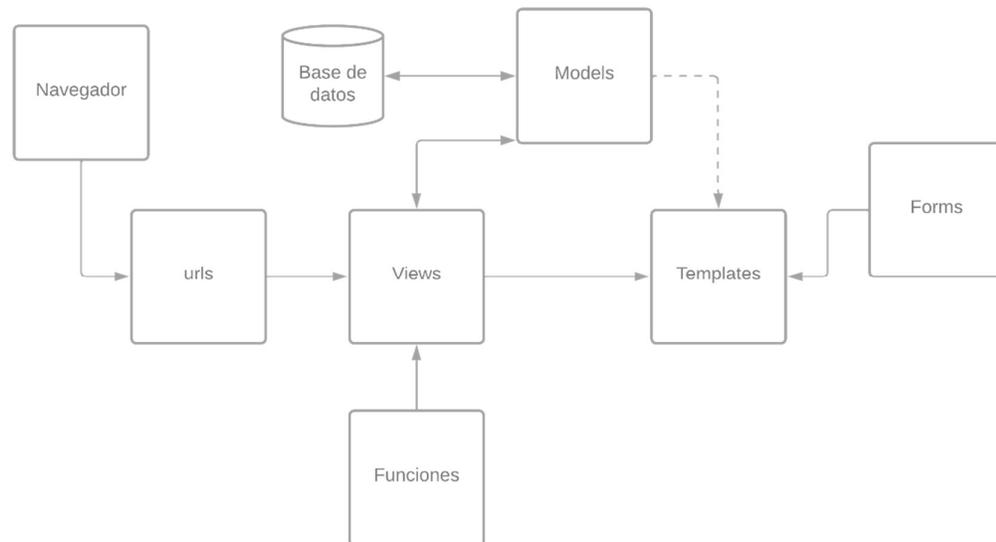
Diagrama de clases (elaborado en [online.visual-paradigm.com](http://online.visual-paradigm.com))



El presente diagrama de clases nos muestra el funcionamiento de manera que cada clase tenga una acción de pertenencia, uso o agregación con las otras.

**Figura34.**

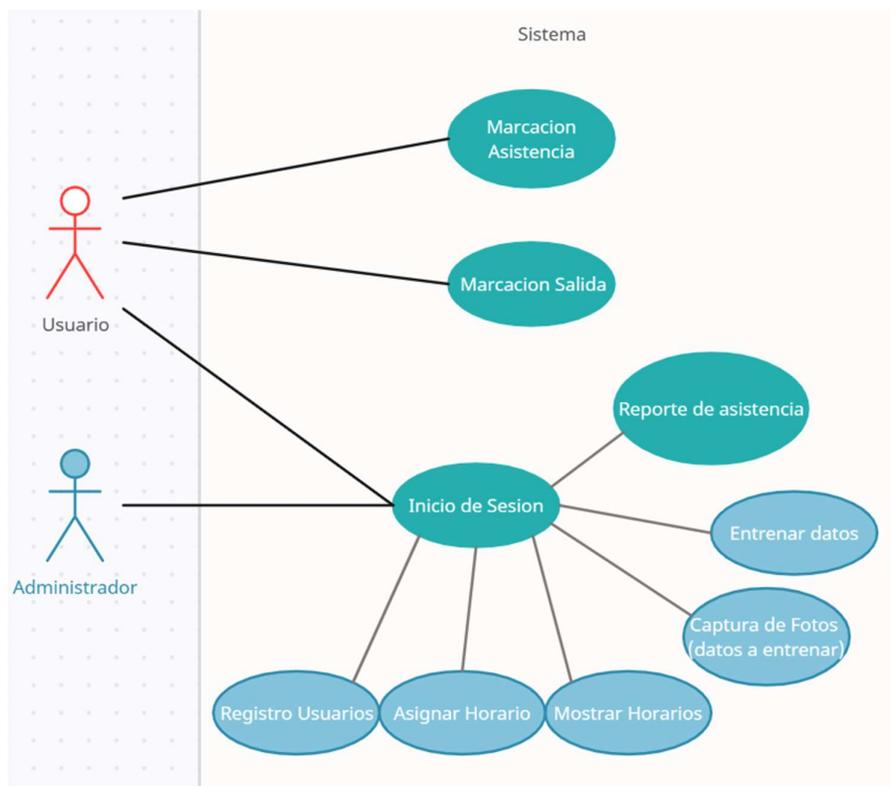
Diagrama de Arquitectura del Sistema



En la presente figura, se detalla la arquitectura del sistema que se desarrolló en el framework Django, mostrando la relación entre las diferentes clases.

**Figura35.**

Diagrama de casos de uso.



En la Figura35, se demuestra el proceso que pueden realizar los diferentes actores, en este caso los usuarios (docentes) y el administrador del sistema, los casos de uso de color verde son acciones que pueden realizar todos los usuarios, en cambio, los celestes son casos de uso que pueden ser realizados únicamente por el administrador.

**Figura 36.**

Clase perteneciente a la base datos.

```

18 class Time(models.Model):
19     user = models.ForeignKey(User, on_delete=models.CASCADE)
20     date = models.DateField(default=datetime.date.today)
21     time = models.DateTimeField(null=True, blank=True)
22     out = models.BooleanField(default=False)
23
24     def __str__(self):
25         return "%s: %s %s %s %s" % (self.id, self.user, self.date, self.time, self.out)
26

```

Esta clase representa el modelo que contiene al usuario, la fecha, el tiempo y el valor booleano de la marcación, para **salida = true y para entrada = false**.

**Figura 34.**

Datos pertenecientes al modelo de la Figura,34.

Administración de Django BIENVENIDOS, ADMIN VER EL SITIO / CAMBIAR CONTRASEÑA / CERRAR SESIÓN

Inicio > Users > Times

Emplee a escribir para filtrar...

AUTENTICACIÓN Y AUTORIZACIÓN

- Grupos + Añadir
- Usuarios + Añadir

USERS

- Asignacions + Añadir
- Horarios + Añadir
- Presents + Añadir
- Times + Añadir

Seleccione time a modificar AÑADIR TIME +

Acción:  Ir seleccionados 0 de 4

USER	DATE	TIME	OUT
<input type="checkbox"/> eduardo	20 de enero de 2023	20 de enero de 2023 a las 21:53	<input checked="" type="checkbox"/>
<input type="checkbox"/> eduardo	20 de enero de 2023	20 de enero de 2023 a las 21:52	<input type="checkbox"/>
<input type="checkbox"/> eduardo	20 de enero de 2023	20 de enero de 2023 a las 21:51	<input checked="" type="checkbox"/>
<input type="checkbox"/> eduardo	20 de enero de 2023	20 de enero de 2023 a las 21:51	<input type="checkbox"/>

4 times

Esta prueba se realiza en la dirección **/view\_configuracion\_horarios**, en donde el superusuario puede asignarle a cada usuario su materia independiente si la materia se repite para diferentes usuarios (Many to One).

**Sprint 2: Construcción en la vista principal de botones capaces de ejecutar la verificación facial tanto para marcar entrada, como para marcar salida.**

La interfaz gráfica de usuario (GUI), busca la manera más óptima en la cual el usuario final puede interactuar con el programa de manera más sencilla, mediante los diferentes elementos visuales que facilitaran su comprensión, tales como: palabras de acción, imágenes, entre otros.

Para el diseño de la interfaz gráfica de usuario se tomó en cuenta determinados parámetros tales como: consistencia, familiaridad del usuario, mínima sorpresa, diversidad de usuarios (Diseño de Interfaz de usuario (UI)).

*Diseño Sprint 2*

**Figura 358.**

Diseño vista preliminar vista principal del sistema



Este diseño preliminar nos muestra la primera etapa de la interfaz principal la cual se planteó para que fuese una vista sencilla de entender para el usuario en cuestión.

## Implementación sprint 2

**Figura 36.**

Vista principal del sistema.



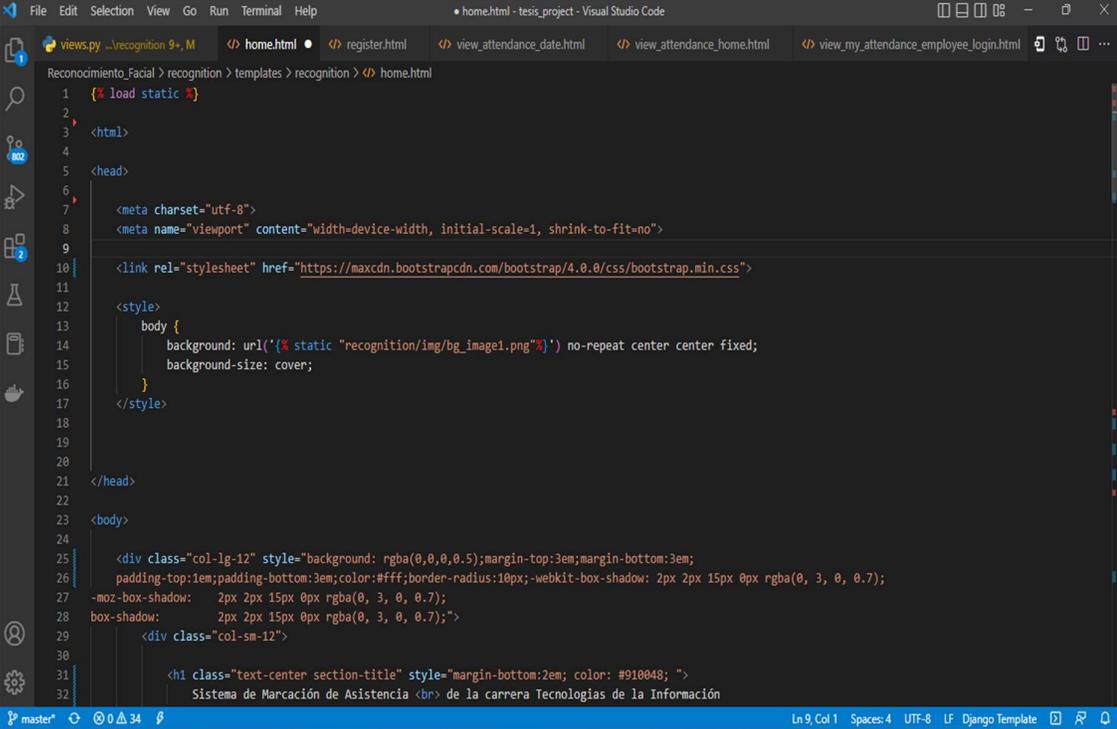
La presente figura nos muestra los botones requeridos y además se insertó un botón para que los usuarios puedan iniciar sesión.

El botón “Marca tu asistencia” efectúa la acción de marcación de entrada mediante el reconocimiento facial y despliega una ventana emergente para poder visualizar la correcta utilización y que las imágenes estén siendo interpretada, esta ventana al momento de ser cerradas redirige a **/home**.

El botón “Marca tu salida” efectúa la acción de marcación de salida mediante el reconocimiento facial y despliega una ventana emergente para poder visualizar la correcta utilización y que las imágenes estén siendo interpretada, esta ventana al momento de ser cerradas redirige a **/home**.

Figura 37.

Construcción de la vista principal en html5. (segmento)



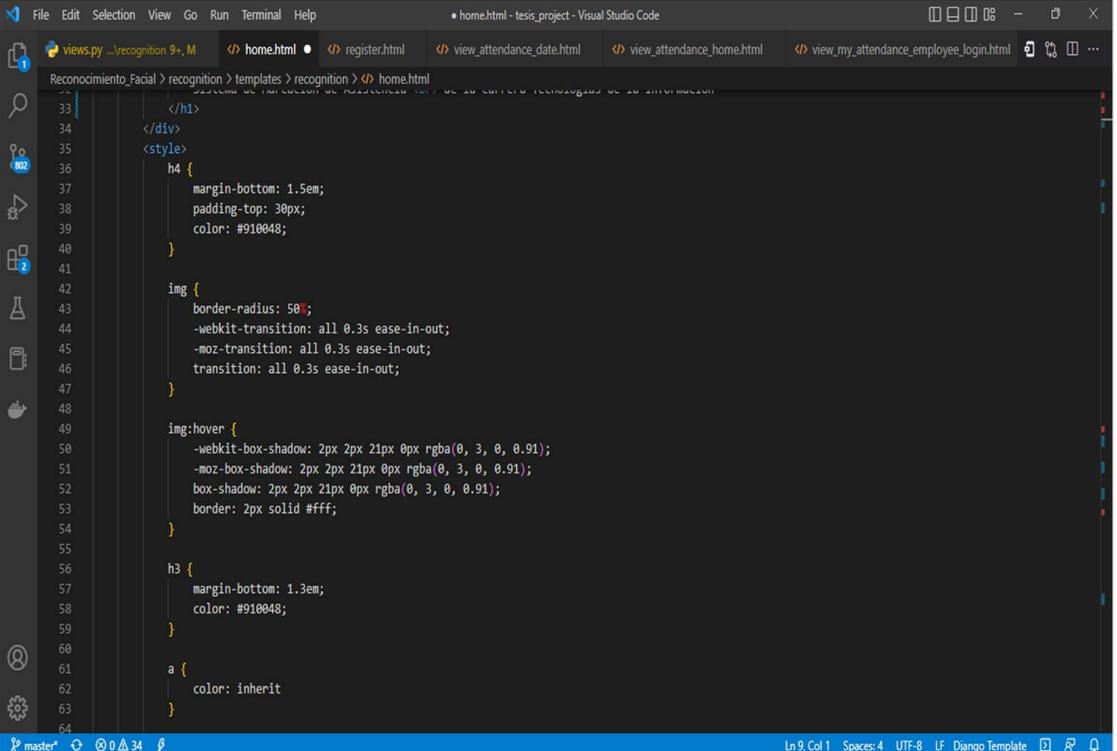
```

1  {% load static %}
2
3  <html>
4
5  <head>
6
7  <meta charset="utf-8">
8  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
9
10 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
11
12 <style>
13     body {
14         background: url('{% static "recognition/img/bg_image1.png"%}') no-repeat center center fixed;
15         background-size: cover;
16     }
17 </style>
18
19
20 </head>
21
22 <body>
23
24
25 <div class="col-lg-12" style="background: rgba(0,0,0,0.5);margin-top:3em;margin-bottom:3em;
26     padding-top:1em;padding-bottom:3em;color:#fff;border-radius:10px;-webkit-box-shadow: 2px 2px 15px 0px rgba(0, 3, 0, 0.7);
27     -moz-box-shadow: 2px 2px 15px 0px rgba(0, 3, 0, 0.7);
28     box-shadow: 2px 2px 15px 0px rgba(0, 3, 0, 0.7);">
29     <div class="col-sm-12">
30
31     <h1 class="text-center section-title" style="margin-bottom:2em; color: #910048; ">
32         Sistema de Marcación de Asistencia <br> de la carrera Tecnologías de la Información

```

Figura 38.

Inserción de estilos CSS para la vista principal. (segmento)



```

33
34 </h1>
35 </div>
36 <style>
37     h4 {
38         margin-bottom: 1.5em;
39         padding-top: 30px;
40         color: #910048;
41     }
42
43     img {
44         border-radius: 50%;
45         -webkit-transition: all 0.3s ease-in-out;
46         -moz-transition: all 0.3s ease-in-out;
47         transition: all 0.3s ease-in-out;
48     }
49
50     img:hover {
51         -webkit-box-shadow: 2px 2px 21px 0px rgba(0, 3, 0, 0.91);
52         -moz-box-shadow: 2px 2px 21px 0px rgba(0, 3, 0, 0.91);
53         box-shadow: 2px 2px 21px 0px rgba(0, 3, 0, 0.91);
54         border: 2px solid #fff;
55     }
56
57     h3 {
58         margin-bottom: 1.3em;
59         color: #910048;
60     }
61
62     a {
63         color: inherit
64     }

```

Figura 39.

Inserción de estilos CSS para la vista principal y (segmento), Construcción de la vista principal en html5. (segmento)

```

64
65     a:hover {
66         color: inherit;
67         text-decoration: none;
68     }
69     /*
70     .section-title:after {
71         content: '';display:block;margin:0 auto;width:100px;margin-top: 6px;border:2px solid #d0d0d0;border-radius:4px;
72         -webkit-border-radius:4px;
73         -moz-border-radius:4px;
74         box-shadow:inset 0 1px 1px rgba(0, 0, 0, .05);
75         -webkit-box-shadow:inset 0 1px 1px rgba(0, 0, 0, .05);
76         -moz-box-shadow:inset 0 1px 1px rgba(0, 0, 0, .05);
77         margin-bottom:1em;
78     }
79     */
80     </style>
81
82     <div class="col-lg-12" >
83         {% if messages %} {% for message in messages%}
84         <div class="alert alert-{{message.tags}}"> {{message}}
85         </div>
86         {%endfor %} {%endif%}
87     </div>
88
89
90     <div class="row">
91
92         <div class="col-md-4" style="padding-left: 100px">
93             <a href="{%url 'mark-your-attendance' %}">
94                 
95             </a>

```

Figura 40.

Construcción de la vista principal en html5. (segmento)

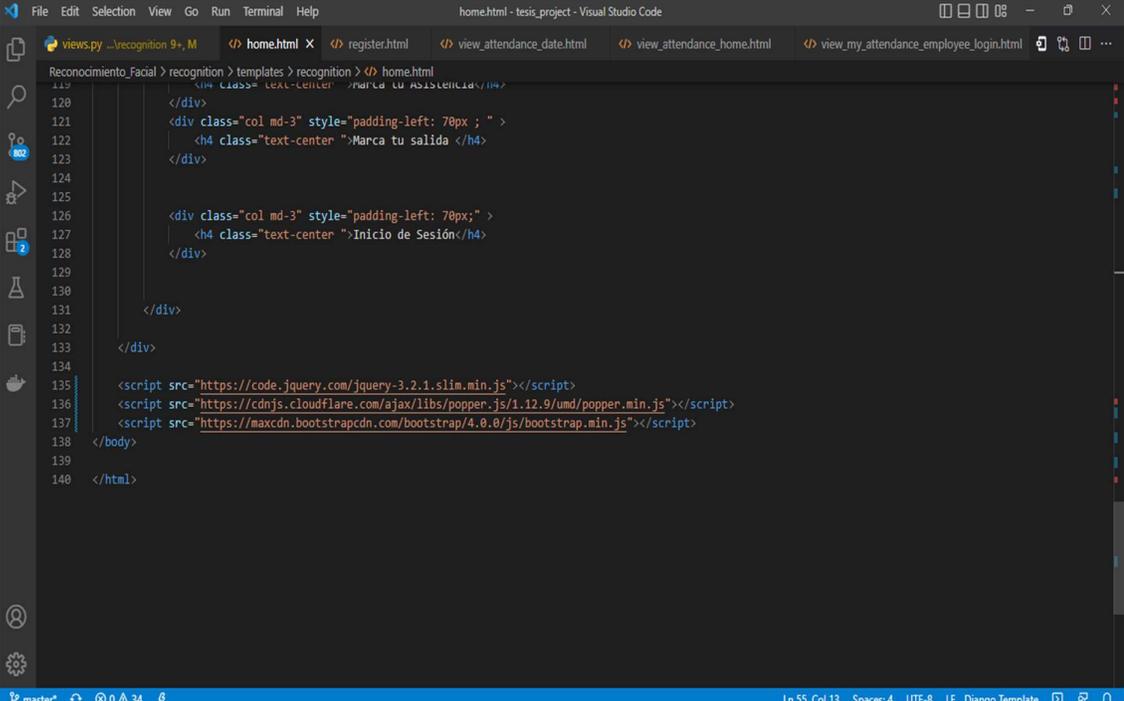
```

96
97     </div>
98
99     <div class="col-md-4" style="padding-left: 100px">
100         <a href="{%url 'mark-your-attendance-out' %}">
101             
102         </a>
103     </div>
104
105
106     <div class="col-md-4" style="padding-left: 100px">
107         <a href="{%url 'login' %}">
108             
109         </a>
110     </div>
111
112 </div>
113 <div class="row">
114
115     <div class="col md-3" style="padding-left: 70px;" >
116         <h4 class="text-center">Marca tu Asistencia</h4>
117     </div>
118     <div class="col md-3" style="padding-left: 70px ;" >
119         <h4 class="text-center">Marca tu salida </h4>
120     </div>
121
122     <div class="col md-3" style="padding-left: 70px;" >
123         <h4 class="text-center">Inicio de Sesión</h4>
124     </div>
125
126
127

```

Figura 41.

## Importación de módulos de Java Script



```
119 <h4 class="text-center">Marca tu asistencia</h4>
120 </div>
121 <div class="col md-3" style="padding-left: 70px;" >
122 <h4 class="text-center">Marca tu salida</h4>
123 </div>
124
125
126 <div class="col md-3" style="padding-left: 70px;" >
127 <h4 class="text-center">Inicio de Sesión</h4>
128 </div>
129
130
131 </div>
132
133 </div>
134
135 <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"></script>
136 <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"></script>
137 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
138 </body>
139
140 </html>
```

Desde la Figura40 hasta la Figura44 se realizó el procedimiento de etiquetado con **HTML** para la creación de la interfaz principal, a estas etiquetas les aplicamos estilo mediante el lenguaje de diseño gráfico **CSS**.

### Sprint 3: Construcción del código que me permita visualizar las horas de entrada y salida.

#### Diseño sprint 3

#### Figura 42.

Diseño preliminar de la vista de reporte de asistencia

Atras		Asistencia Marcada	
Usuario	Día de Marcación	Hora Entrada	Hora Salida
Usuario 1	Lunes 23 de enero de 2021	9h00	12h00
Usuario 2	Lunes 23 de enero de 2021	11h00	13h00
Usuario 3	Lunes 23 de enero de 2021	12h00	14h00
Usuario 4	Lunes 23 de enero de 2021	15h00	17h00
Usuario 5	Lunes 23 de enero de 2021	17h00	20h00

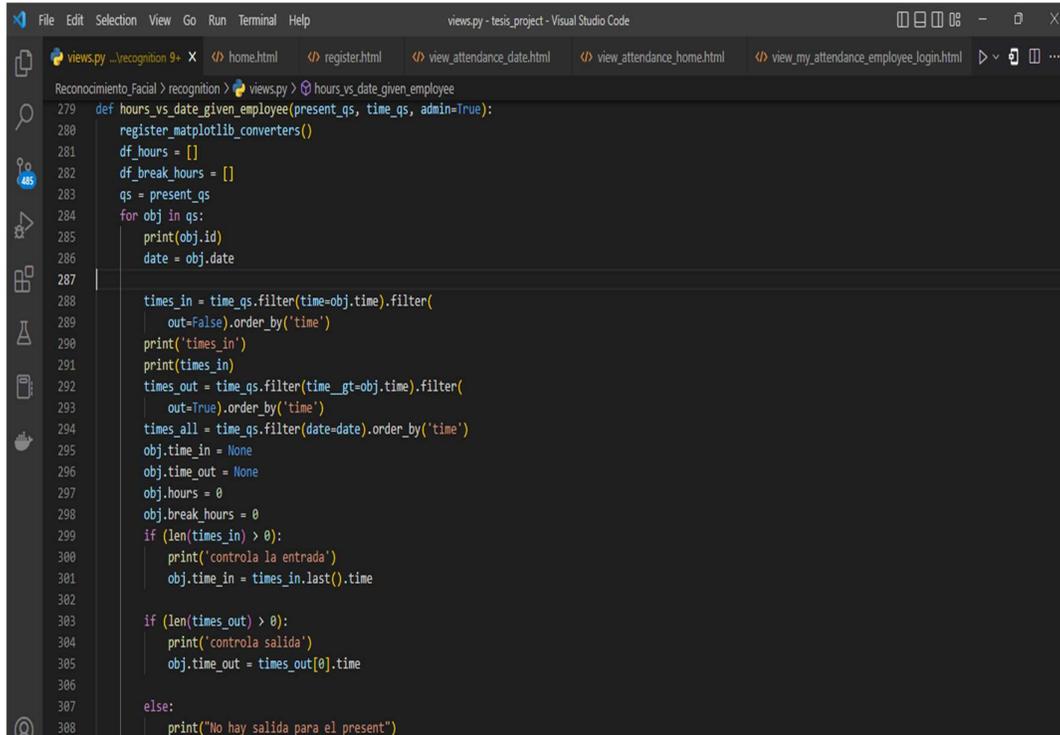
Este diseño preliminar construido en proto.io. Nos refleja un preámbulo de lo que será el apartado de la tabla de registro, de horas de entrada y salida conjunto, al nombre de usuario y la fecha de marcación

#### Implementación sprint 3.

En la Figura 30. Se visualiza mediante el uso de una tabla, las horas de entrada y salida, además del nombre de usuario, fecha de ingreso en formato: día, mes y año.

Figura 43.

Definición de la función y codificación del algoritmo para visualizar los datos requeridos en la vista principal. (segmento)



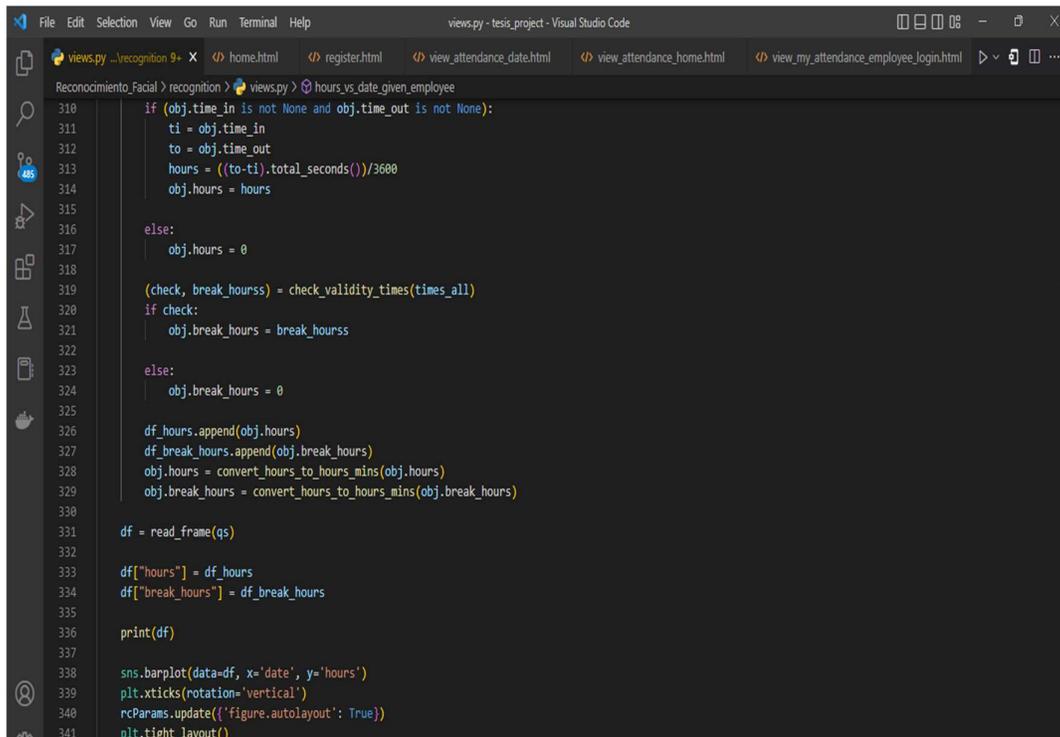
```

279 def hours_vs_date_given_employee(present_qs, time_qs, admin=True):
280     register_matplotlib_converters()
281     df_hours = []
282     df_break_hours = []
283     qs = present_qs
284     for obj in qs:
285         print(obj.id)
286         date = obj.date
287
288         times_in = time_qs.filter(time=obj.time).filter(
289             out=False).order_by('time')
290         print('times_in')
291         print(times_in)
292         times_out = time_qs.filter(time_gt=obj.time).filter(
293             out=True).order_by('time')
294         times_all = time_qs.filter(date=date).order_by('time')
295         obj.time_in = None
296         obj.time_out = None
297         obj.hours = 0
298         obj.break_hours = 0
299         if (len(times_in) > 0):
300             print('controla la entrada')
301             obj.time_in = times_in.last().time
302
303         if (len(times_out) > 0):
304             print('controla salida')
305             obj.time_out = times_out[0].time
306
307         else:
308             print("No hay salida para el present")

```

Figura 44.

Codificación del algoritmo para visualizar los datos requeridos en la vista principal. (segmento)



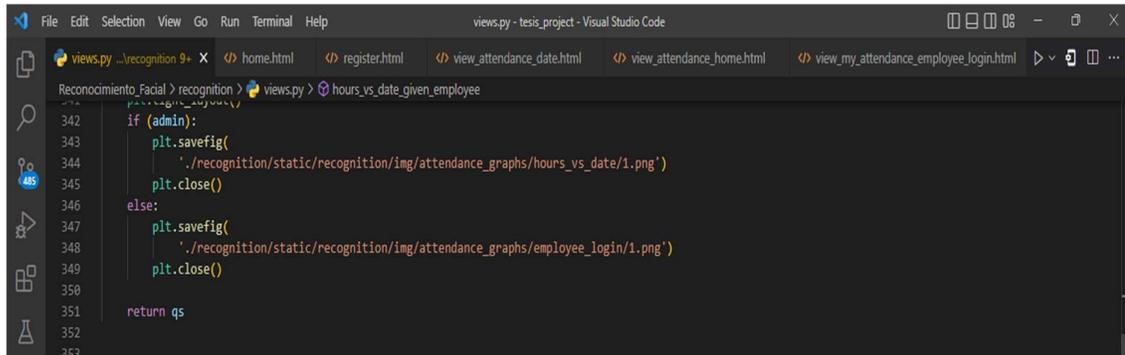
```

310     if (obj.time_in is not None and obj.time_out is not None):
311         ti = obj.time_in
312         to = obj.time_out
313         hours = ((to-ti).total_seconds())/3600
314         obj.hours = hours
315
316     else:
317         obj.hours = 0
318
319     (check, break_hourss) = check_validity_times(times_all)
320     if check:
321         obj.break_hours = break_hourss
322
323     else:
324         obj.break_hours = 0
325
326     df_hours.append(obj.hours)
327     df_break_hours.append(obj.break_hours)
328     obj.hours = convert_hours_to_hours_mins(obj.hours)
329     obj.break_hours = convert_hours_to_hours_mins(obj.break_hours)
330
331     df = read_frame(qs)
332
333     df["hours"] = df_hours
334     df["break_hours"] = df_break_hours
335
336     print(df)
337
338     sns.barplot(data=df, x='date', y='hours')
339     plt.xticks(rotation='vertical')
340     rcParams.update({'figure.autolayout': True})
341     plt.tight_layout()

```

**Figura 45.**

Codificación del algoritmo para visualizar los datos requeridos en la vista principal. (segmento)

A screenshot of the Visual Studio Code editor showing a Python file named 'views.py' in a project called 'Reconocimiento\_Facial'. The code is located in the 'views.py' file under the 'recognition' directory. The code snippet shows an 'if' statement that checks if the user is an 'admin'. If true, it saves a plot of attendance data for a specific date to a file named 'hours\_vs\_date/1.png'. If false, it saves a plot of attendance data for an employee login to a file named 'employee\_login/1.png'. The code uses 'plt.savefig()' to save the plot and 'plt.close()' to close the plot window. The code is as follows:

```
342     if (admin):
343         plt.savefig(
344             './recognition/static/recognition/img/attendance_graphs/hours_vs_date/1.png')
345         plt.close()
346     else:
347         plt.savefig(
348             './recognition/static/recognition/img/attendance_graphs/employee_login/1.png')
349         plt.close()
350
351     return qs
352
353
```

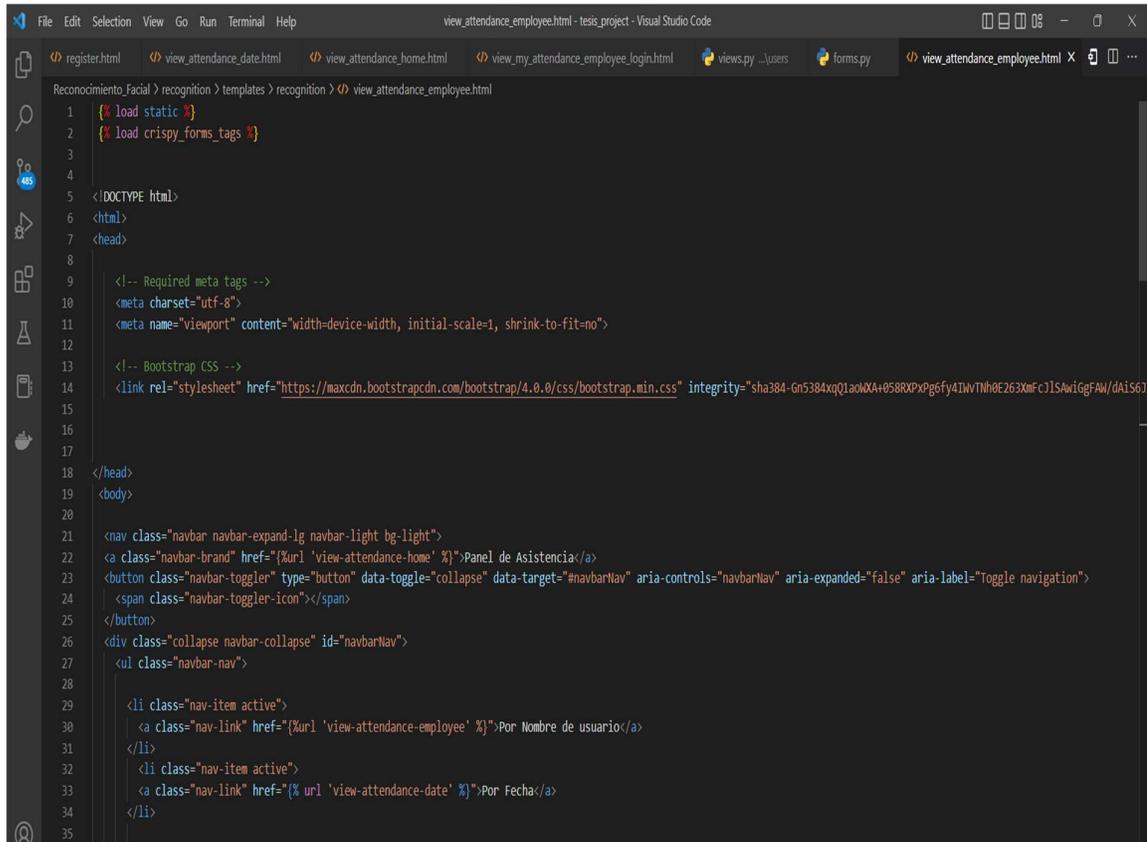
Desde la Figura 46 hasta la Figura48, realizamos mediante el uso de librerías y módulos como: **matplotlib, datetime, date y seaborn**, y sus respectivas funciones, la traslación de los datos requeridos hacia la tabla y añadiendo el grafico correspondiente a la tabla.

**Sprint 4: Realizar la construcción de una vista donde se permita tanto al administrador como al usuario poder ver los reportes de asistencias.**

De acuerdo con la Figura27, se visualiza el espacio donde se muestran la tabla estilizada, que nos permite enterarnos de los reportes de cada usuario.

Figura 46.

Etiquetas HTML para la creación de la vista mostrada en la figura27.



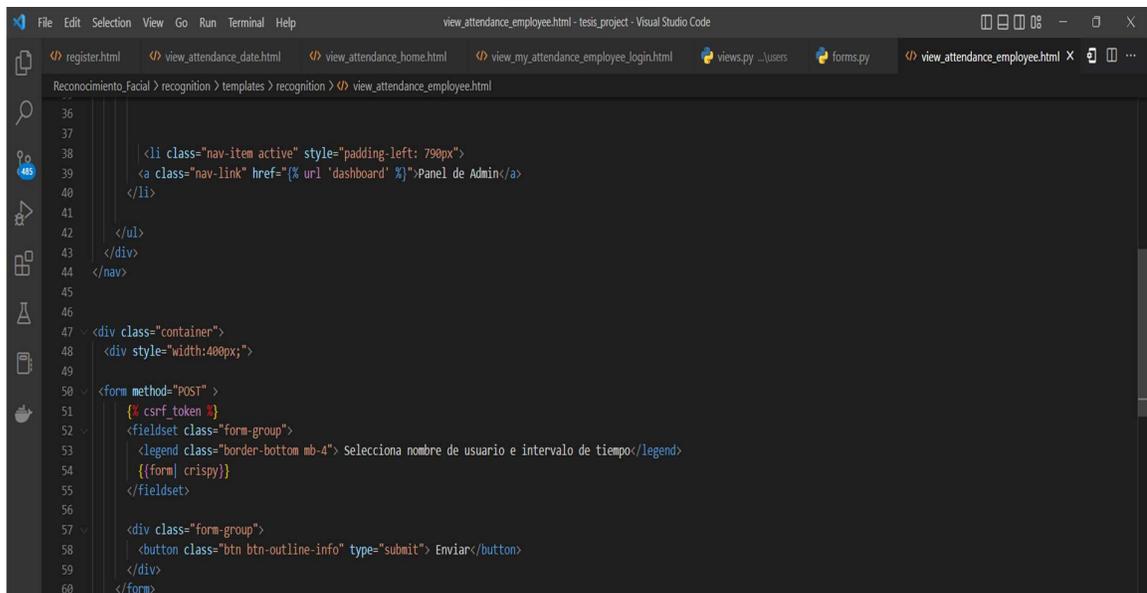
```

1  {% load static %}
2  {% load crispy_forms_tags %}
3
4
5  <!DOCTYPE html>
6  <html>
7  <head>
8
9      <!-- Required meta tags -->
10     <meta charset="utf-8">
11     <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
12
13     <!-- Bootstrap CSS -->
14     <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1ao7XA+058RXPxPg6fy4IWVTNh0E263XmfcJ1S1AiGgFaw/dA1S6J"
15
16
17
18 </head>
19 <body>
20
21 <nav class="navbar navbar-expand-lg navbar-light bg-light">
22 <a class="navbar-brand" href="{%url 'view-attendance-home' %}">Panel de Asistencia</a>
23 <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
24 <span class="navbar-toggler-icon"></span>
25 </button>
26 <div class="collapse navbar-collapse" id="navbarNav">
27 <ul class="navbar-nav">
28
29 <li class="nav-item active">
30 <a class="nav-link" href="{%url 'view-attendance-employee' %}">Por Nombre de usuario</a>
31 </li>
32 <li class="nav-item active">
33 <a class="nav-link" href="{% url 'view-attendance-date' %}">Por Fecha</a>
34 </li>
35

```

Figura 47.

Etiquetas HTML para la creación de la vista mostrada en la figura27.



```

36
37
38 <li class="nav-item active" style="padding-left: 790px">
39 <a class="nav-link" href="{% url 'dashboard' %}">Panel de Admin</a>
40 </li>
41
42 </ul>
43 </div>
44 </nav>
45
46
47 <div class="container">
48 <div style="width:400px;">
49
50 <form method="POST" >
51 <{% csrf_token %}>
52 <fieldset class="form-group">
53 <legend class="border-bottom mb-4"> Selecciona nombre de usuario e intervalo de tiempo</legend>
54 <{{form| crispy}}>
55 </fieldset>
56
57 <div class="form-group">
58 <button class="btn btn-outline-info" type="submit"> Enviar</button>
59 </div>
60 </form>

```

Figura 48.

Etiquetas HTML para la creación de la vista mostrada en la figura27.

```

view_attendance_employee.html - tesis_project - Visual Studio Code
Reconocimiento_Facial > recognition > templates > recognition > view_attendance_employee.html
59     </div>
60     </form>
61
62 </div>
63
64 {% if qs %}
65 <table class="table" style="margin-top: 5em;">
66   <thead class="thead-dark">
67     <tr>
68       <th scope="col">Fecha</th>
69
70       <th scope="col">Nombre de usuario</th>
71       <th scope="col">Presente</th>
72       <th scope="col">Hora de Entrada</th>
73       <th scope="col">Hora de salida </th>
74       <th scope="col">Horas Dentro de clase </th>
75       <th scope="col">Horas Fuera de clase</th>
76     </tr>
77   </thead>
78   <tbody>
79     {% for item in qs %}
80     <tr>
81       <td>{{ item.date }}</td>
82       <td>{{ item.user.username }}</td>
83
84       {% if item.present %}
85       <td>P </td>
86       {% else %}
87       <td>A </td>
88       {% endif %}
89       {% if item.time_in %}
90       <td>{{ item.time_in }}</td>
91       {% else %}
92       <td>- </td>
93       {% endif %}
94       {% if item.time_out %}
95       <td>{{ item.time_out }}</td>
96       {% else %}

```

Figura 49.

Etiquetas HTML para la creación de la vista mostrada en la figura27.

```

view_attendance_employee.html - tesis_project - Visual Studio Code
Reconocimiento_Facial > recognition > templates > recognition > view_attendance_employee.html
96     {% else %}
97     <td>- </td>
98     {% endif %}
99     <td>{{ item.hours }}</td>
100    <td>{{ item.break_hours }}</td>
101  </tr>
102  {% endfor %}
103 </tbody>
104 </table>
105
106 <div class="card" style="margin-top: 5em; margin-bottom: 10em;">
107   
108   <div class="card-body">
109     <p class="card-text" style="text-align: center;">Numero de horas trabajadas al dia:</p>
110   </div>
111 </div>
112
113 {% endif %}
114
115
116 {% if messages %}
117   {% for message in messages%}
118   <div class="alert alert-{{message.tags}}"> {{message}}
119   </div>
120   {%endfor %}
121 {%endif%}
122
123 </div>
124
125
126 <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" ></script>
127 <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"></script>
128 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
129
130
131
132 </body>
133 </html>

```

Mediante los requerimientos establecidos se procedió a realizar la construcción de la vista con el uso del lenguaje de etiquetas **HTML**, mientras que con el uso del lenguaje de diseño **CSS** se realizó el embellecimiento de la interfaz

**Sprint 5: Crear una vista en donde el administrador pueda asignar horarios a las personas registradas.**

*Diseño sprint 5*

**Figura 50.**

Diseño preliminar de la sección horarios por asignar.



Atras

**Asignar Horarios**

User

User 1

Materias

Materia 1 hh:mm:ss, dd

Materia 2 hh:mm:ss, dd

Materia 3 hh:mm:ss, dd

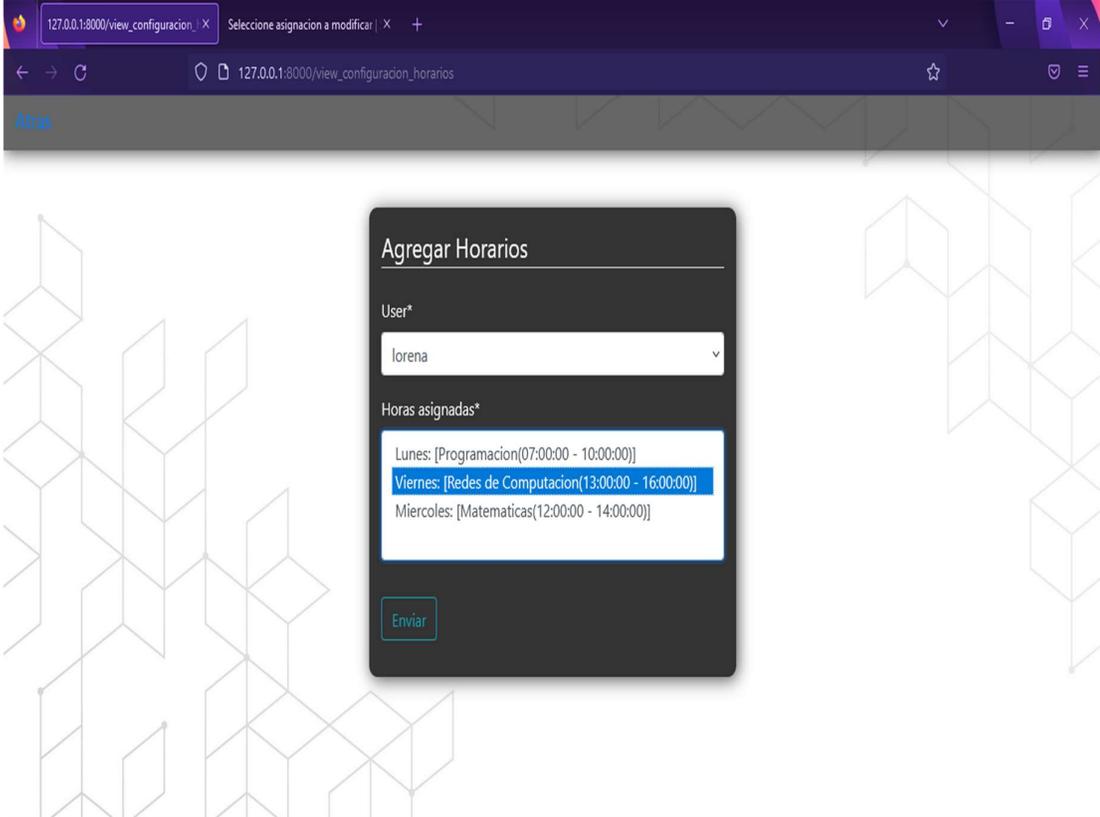
Enviar

El presente diseño realizado en proto.io. Nos refleja de manera preliminar la vista de horarios por asignar que cuenta con un **box de selección** en el cual se podrán seleccionar los usuarios a lo que les añadiremos un horario y un **check box** que detallara el horario a asignarse.

## Implementación sprint 5.

### Figura 51.

#### Asignar horarios a los usuarios



The image shows a web browser window with a dark theme. The address bar displays the URL `127.0.0.1:8000/view_configuracion_horarios`. The page content features a modal form titled "Agregar Horarios". The form includes a "User\*" dropdown menu with "lorena" selected. Below it, the "Horas asignadas\*" section contains a list of schedule entries: "Lunes: [Programacion(07:00:00 - 10:00:00)]", "Viernes: [Redes de Computacion(13:00:00 - 16:00:00)]", and "Miercoles: [Matematicas(12:00:00 - 14:00:00)]". The "Viernes" entry is highlighted with a blue background. At the bottom of the form is a blue "Enviar" button. The browser's back button is labeled "Atrás".

La presente vista nos permite enviar un formulario a la base de datos con el nombre de usuario conjunto al horario asignado.

El botón enviar, envía el formulario sin redireccionarnos a ninguna otra ruta, mientras tanto el botón atrás nos redirige a **/dashboard**

Figura 52.

Etiquetas HTML para la creación de la Fig.54.

```

1  {% load static %}
2  {% load crispy_forms_tags %}
3
4  <html>
5  <head>
6
7      <!-- Required meta tags -->
8      <meta charset="utf-8">
9      <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
10
11     <!-- Bootstrap CSS -->
12     <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
13
14     <style>
15     body{
16         background: url('{% static "recognition/img/bg_image1.png"%}') no-repeat center center fixed;
17         background-size: cover;
18     }
19     </style>
20
21 </head>
22
23 <body>
24
25 <div class="col-lg-12" style="background: rgba(0,0,0,0.6);max-height: 20px;
26 padding-top:0.5em;padding-bottom:2.5em;color:#fff;webkit-box-shadow: 2px 2px 15px 0px rgba(0, 3, 0, 0.7);
27 -moz-box-shadow: 2px 2px 15px 0px rgba(0, 3, 0, 0.7);
28 box-shadow: 2px 2px 15px 0px rgba(0, 3, 0, 0.7); margin-left:auto; margin-right: auto;">
29
30 <a href="{% url 'dashboard' %}"><h5 class="text-left">Atras</h5></a>
31 </div>
32
33 <div class="col-lg-4" style="background: rgba(0,0,0,0.8);margin-top:50px ;margin-bottom:50px;
34 padding-top:1em;padding-bottom:1em;color:#fff;border-radius:10px;-webkit-box-shadow: 2px 2px 15px 0px rgba(0, 3, 0, 0.7);
35 -moz-box-shadow: 2px 2px 15px 0px rgba(0, 3, 0, 0.7);
36 box-shadow: 2px 2px 15px 0px rgba(0, 3, 0, 0.7); margin-left:auto; margin-right: auto;">
37
38 <form method="POST" >

```

Figura 53.

Etiquetas HTML para la creación de la Fig.54.

```

39     {% csrf_token %}
40     <fieldset class="form-group">
41         <legend class="border-bottom mb-4">Agregar Horarios </legend>
42         {{form| crispy}}
43     </fieldset>
44
45     <div class="form-group">
46         <button class="btn btn-outline-info" type="submit"> Enviar</button>
47     </div>
48 </form>
49
50 </div>
51
52 <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"></script>
53 <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"></script>
54 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
55 </body>
56 </html>

```

Las etiquetas **HTML** presentadas en las Figuras 55 y 56, mediante su correcta utilización y conjunto a los estilos asignados con el uso de **CSS**, se ha podido realizar satisfactoriamente el Sprint correspondiente.

**Sprint 6: Desarrollar una vista en donde el administrador pueda visualizar los horarios por asignar.**

*Diseño sprint 6.*

**Figura 54.**

Diseño preliminar de la vista de horarios

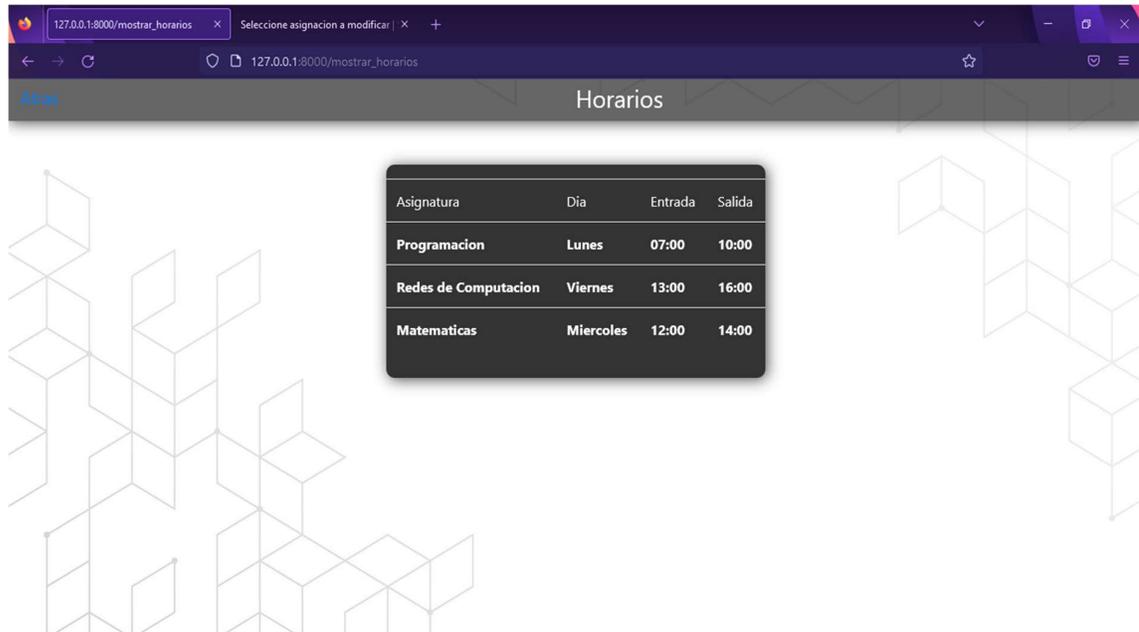


La presente Figura muestra de manera preliminar el diseño del apartado de horarios, fue construido con la herramienta proto.io. Que nos muestra de manera sencilla y explícita como se reflejaría la vista de los horarios disponibles, conjunto de un botón atrás, para poder regresar a la vista anterior.

## Implementación sprint 6.

**Figura 55.**

Tabla de horario vigentes por asignatura.



Con el uso de etiquetas **HTML** que nos facilitan la construcción de tablas, conjunto al recorrido de ítems dentro de la clase horarios de la base de datos, se pueden integrar a la interfaz de mostrar horarios, y con el uso de estilos se mejora la presentación final de la misma.

La única ruta accesible de este diseño es el botón Atrás que nos redirige hacia **/dashboard**

Sprint 7: Creación de una función que permita el control de horarios asignados.

**Figura 56.**

Clases del modelo pertenecientes a la asignación de horarios,

```

28 class Asignacion(models.Model):
29     dia_semana = (('lunes', 'Lunes'), ('martes', 'Martes'), ('miercoles',
30                                                         'Miercoles'), ('jueves', 'Jueves'), ('viernes', 'Viernes'))
31     dia = models.CharField(max_length=30, choices=dia_semana)
32     hora_inicio = models.TimeField()
33     hora_fin = models.TimeField()
34     asignatura = models.CharField(max_length=255)
35
36     def __str__(self):
37         return self.dia.capitalize()+':'+ " ["+str(self.asignatura)+"(" +str(self.hora_inicio)+" - "+str(self.hora_fin)+")"+"]"
38
39
40 class Horarios(models.Model):
41     user = models.ForeignKey(User, on_delete=models.CASCADE)
42     horas_asignadas = models.ManyToManyField(Asignacion, related_name="horas")
43
44     def __str__(self):
45         return "%s: %s %s " % (self.id, self.user, self.horas_asignadas)
46
47

```

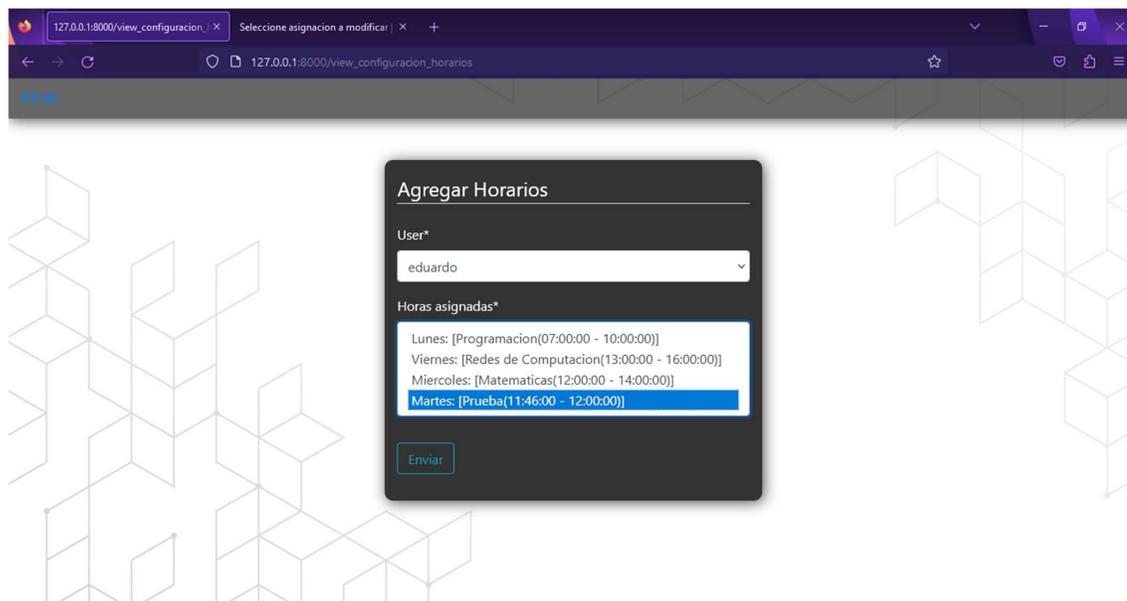


Esta función creada nos permite mediante el **user.id**, utilizar la hora de entrada del usuario para validar que se ha efectuado la marcación de la hora correspondiente.

En la Figura61. Se realiza el mismo proceso, sin embargo, esta validara la marcación de la hora de salida.

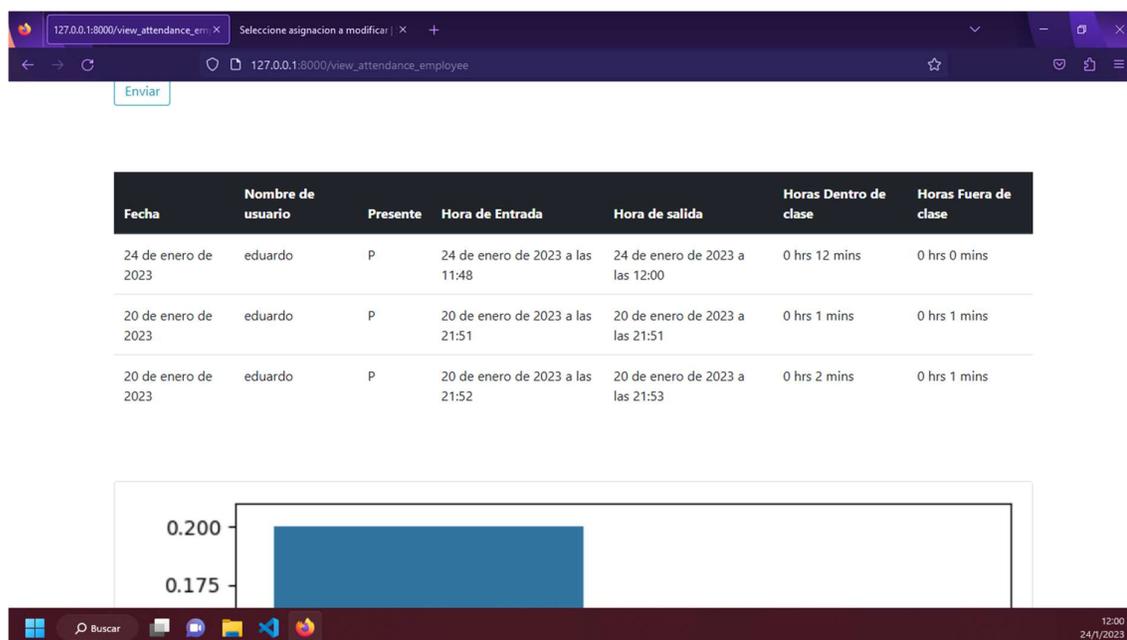
**Figura 59.**

Agregación de horario al usuario “eduardo”



**Figura 60.**

Marcación dentro del rango del horario.

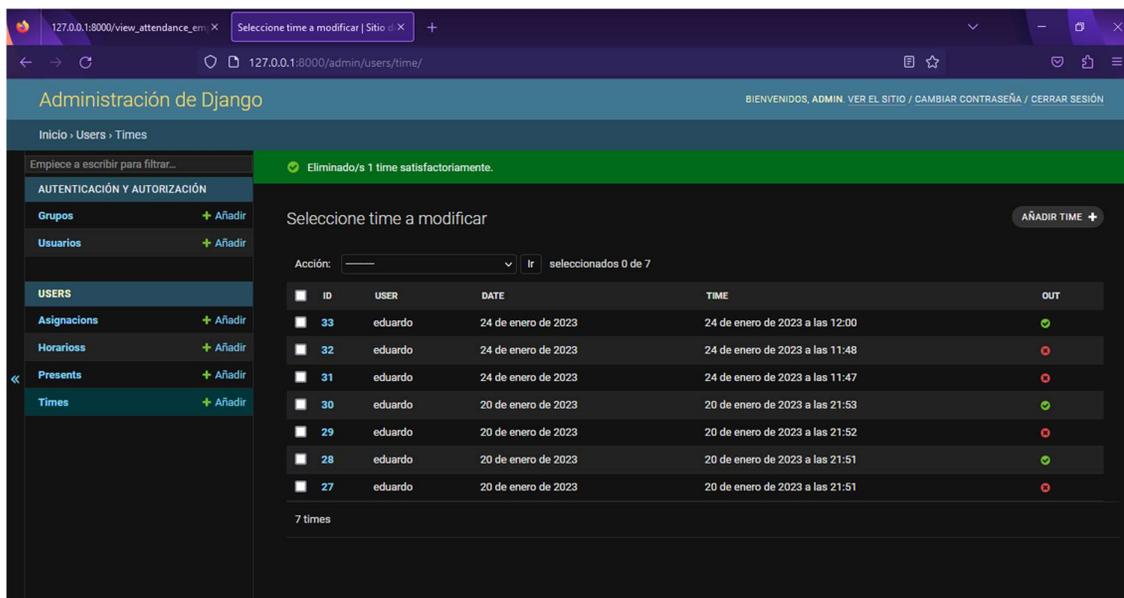


En la Figura 62, se efectúa la asignación del horario del horario al usuario en cuestión, el usuario, solo puede marcar dentro de esas horas.

En la Figura 63. Se ve el reporte cuando la marcación es dentro del horario establecido, caso contrario la marcación no se realizará y en la vista de los reportes, no aparecerá.

### Figura 61.

Demostración del almacenamiento de los registros.



*Nota: Cabe mencionar que el registro con id 32 y 33 son los registros de marcación entrada y salida pertenecientes a la asignación dada en la Figura 62, de acuerdo con el horario.*

## Sprint 8: Desarrollo de la inteligencia artificial capaz de poder realizar la operación de reconocimiento facial.

En la explicación correspondiente en las Figuras 19 y 20, se detalla el proceso.

**CAPITULO IV:  
CONCLUSIONES Y  
RECOMENDACIONES**

#### 4.1. Conclusiones.

En el transcurso del desarrollo del proyecto de Titulación, según el estado del arte en cuestión, se ha logrado encontrar diferentes maneras para implementar la solución del control de asistencia con la técnica de visión artificial, existe la disponibilidad de varias librerías para la plataforma Python, como **Tensorflow y Keras**, estas brindan prestaciones similares **sklearn** y las dependencias de **dlib**.

En el caso del presente proyecto se utilizó una cámara de marca havit, con un sensor CMOS, la cámara según especificaciones es full HD en resolución de 720p, también se utilizó la cámara integrada al ordenador Lenovo Y520, la única información existente en el manual del dispositivo indica que la cámara es HD de 720p, obteniendo que la cámara externa funciona de mejor manera en diferentes ambientes, con la cual se puede reconocer a la persona en una distancia aproximada de un metro 1m ( $\pm 50$ cm).

El framework Django permite desarrollar soluciones en poco tiempo, fáciles de usar (UX enriquecida) en base a la posibilidad de insertar HTML, CSS y Bootstrap. Este conjunto de herramientas permite presentar una experiencia de usuario fácil de entender y usar, con esto se reduce la curva de aprendizaje de un nuevo sistema y se allana el camino a la adopción de la innovación.

De acuerdo con la prueba realizada en una máquina Core i5 de 4ta generación con tarjeta gráfica integrada al procesador Intel, con 4 Gb de memoria RAM, y una máquina Core i7 de Intel de 7ma generación con tarjeta gráfica dedicada Nvidia 1060 with Max-q designs y 16 Gb de memoria RAM, se pudo comprobar que la maquina con mejores prestaciones obtiene una mejora en cuanto al procesamiento de imágenes.

**Tabla 2.**

#### Especificaciones

Hardware	Lenovo Y520	Dell Inspiron 3442
Procesador	Intel Core i7-7700HQ 2.80GHz(8CPU)	Intel Core i5-4210 1.70Gh (4CPU)
RAM	16384MB	4096MB
Tarjeta de video	Nvidia 1060 with Max Q designs 6GB	Intel HD Graphics Family 2112 MB
Cámara	Cámara Integrada 720p	Cámara Integrada 720p

## 4.2. Recomendaciones

Utilizar el sistema de reconocimiento facial y el proceso de añadir fotos de los usuarios, en lugares con la claridad suficiente en caso de usar cámaras de baja calidad de resolución. Se recomienda que para mejor exactitud se utilice dispositivos con una resolución suficiente, de 720p en adelante.

Se recomienda para la aplicación de las diferentes técnicas relacionadas a la inteligencia artificial, se use el lenguaje de programación Python, debido a que presenta un número significativo de librerías como: tensorflow, keras, HAAR features, Face\_recognition, entre otras, que facilitan la implementación de este tipo de soluciones.

Se recomienda usar un procesador AMD threadripper 2990wx, con 32Gb de Memoria RAM y un una de video de procesamiento grafico dedicado Nvidia 2070 en adelante.

Se puede complementar el presente proyecto, manteniendo las diferentes técnicas de programación utilizadas durante el desarrollo, además, añadir más funcionalidades de acuerdo con nuevos requerimientos, para versiones posteriores.

## Referencias

- Aprende con ALF. (2020). *Python*. Obtenido de <https://aprendeconalf.es/docencia/python/manual/matplotlib/>
- Braharath, A., & Petrov, M. (2008). *Next Generation Artificial Vision Systems*. Norwood: Imperial College London.
- Challenger, I., Díaz, Y., & Becerra, R. A. (2014). *El lenguaje de programación Python*. Holguin.
- Chiarely, A. (2018). *Beginning React*. Birmingham B3 2PB, Uk: Packt Publishing.
- Data Scientist. (2019). *Scikit-Learn : Descubre la biblioteca de Python dedicada al Machine Learning*. Obtenido de <https://datascientest.com/es/scikit-learn-descubre-la-biblioteca-python>
- Digite. (2021). *Scrum methodology*. Obtenido de <https://www.digite.com/wp-content/uploads/2021/08/scrum-methodology.png>
- Dongwei, X., Shuyu, Z., Yun, M., Deyu, T., & Xuanzhe, L. (2019). Moving Deep Learning into Web Browser: How Far can we go? *Key Lab of High-Confidence Software Technology, MoE*, 3-4.
- Futurespace. (2018). *Redes neuronales*. Obtenido de <https://www.futurespace.es/redes-neuronales-y-deep-learning-capitulo-2-la-neurona/>
- Lawrence, J. (1993). Introduction to neural networks. *California Scientific Software*.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning for AI. *ACM*, 64(7), 58-65.
- Mohedano, J., Saiz, M., & Salazar Roman, P. (2013). *Iniccion a JavaScript*. Madrid: Ministerio de Educacion y Formacion Profesional de España.
- Open CV. (2019). Obtenido de <https://spiegato.com/es/que-es-opencv>
- O'Shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. *arXiv preprint*, 3(2), 1-11.
- Osorio, D., & Garcia, S. (2022). *Sistema biométrico de reconocimiento facial basado en inteligencia artificial*. Santiago de Cali: Universidad Autonoma de Occidente.
- Pavón Puertas, J., & Llarena Borges, E. (2015). *Cracion de un sitio web con PHP y MySQL E. (2015)*. Paracuellos de Jarama, Madrir: RA-MA, S.A. Editorial y Publicaciones.
- Pyspanish. (2017). *Serialización de objetos Python*. Obtenido de <https://pyspanishdoc.sourceforge.net/lib/module-pickle.html>

Rouhiainen, L. (2018). *Inteligencia artificial*. Madrid: Alienta.

Sandoval, L. (2018). Algoritmos de aprendizaje automático para análisis y predicción de datos.

*REDICCES*, 11(2), 36-41. Obtenido de

[http://redicces.org.sv/jspui/bitstream/10972/3626/1/Art6\\_RT2018.pdf](http://redicces.org.sv/jspui/bitstream/10972/3626/1/Art6_RT2018.pdf)

Tsoi, A. C. (2006). Recurrent neural network architectures: an overview. Adaptive processing of sequences and data structures: international summer school on neural networks “er caianiello” vietri sul mare. *Salerno*, 5(2), 1-26.

WHO. (2017). *MN68*. Obtenido de

<https://who.canto.global/v/MN68KSP9SD/album/NNBOS?display=curatedView&viewIndex=0>

# ANEXOS

## 5. Anexos

### 5.1. Anexo 1. Anteproyecto

**UNIVERSIDAD INTERNACIONAL DEL ECUADOR**

**ESCUELA DE INFORMATICA Y MULTIMEDIA**



**PRESENTACION DE ANTE - PROYECTO DE TESIS DE GRADO**

#### **TEMA**

“Diseño de un prototipo para el control de asistencia de los docentes de la carrera de Tecnologías de la información en la Universidad Internacional del Ecuador sede Loja mediante el uso de visión artificial “.

PRESENTADA POR

Eduardo J. Araujo A.

Loja 17/06/2022

## **A. PROBLEMÁTICA**

### **1. SITUACIÓN PROBLEMÁTICA**

#### **PROBLEMA DE INVESTIGACIÓN**

Desde el año 2020 debido a la situación global con el virus Covid-19 y sus posteriores mutaciones, se ha entrado a un periodo de pandemia, crisis en salud y recesión económica. Ante estos hechos, varios de los dispositivos tecnológicos que el ser humano tenía que usar para la captura de datos biométricos dactilares, se dejaron de emplear, ya que dichos instrumentos se consideran un foco de infección inminente, teniendo que ser desinfectados constantemente para que puedan ser usados, acortando su vida útil.

En el caso de la Universidad Internacional del Ecuador sede Loja los docentes y el personal administrativo, han dejado de usar el método biométrico de registro de asistencia, por lo antes mencionado, se realiza la propuesta del desarrollo de: "Diseño de un prototipo para el control de asistencia de los docentes de la carrera de Tecnologías de la información en la Universidad Internacional del Ecuador sede Loja mediante el uso de visión artificial".

## **B. Justificación**

- **ACADÉMICA**

Este proyecto está enfocado a utilizar conocimientos impartidos durante la carrera como es el uso de inteligencia artificial, reconocimiento de patrones, base de datos y programación.

- **ECONÓMICA**

El presente prototipo será realizado con herramientas open source, ya que de esta manera poder ayudar a la comunidad informática y a la UIDE (Universidad Internacional del Ecuador) sede Loja.

- **TÉCNICA**

Para la realización del presente proyecto se cuenta con la información recibida en clases tales como la programación, uso de bases de datos, uso de la lógica y herramientas basadas en la incorporación de inteligencia artificial, que se usará a lo largo del desarrollo del proyecto, para el óptimo desarrollo del propósito de esta investigación.

// justificación técnica del proyecto

- **SOCIAL**

Este proyecto tiene un aporte social debido a que está dirigido a empresas y personal que no cuentan con un sistema de asistencias touchless, para el presente plan hemos decido diseñar un prototipo para el área de TI de la Universidad Internacional Del Ecuador sede Loja.

Que brindara una mejor seguridad industrial y salud ocupacional ya que con esta propuesta de proyecto se busca eludir en lo posible el contacto físico con un artilugio para el registro de asistencia.

## **C. Objetivos**

### **1. Objetivo General**

Desarrollar el Prototipo de Sistema de asistencia para los docentes del Departamento de Tecnologías de la Información en UIDE sede Loja, mediante el uso de visión artificial.

### **2. Objetivos Específicos**

- 2.1. Consultar el Estado del arte para la elaboración de prototipo //tecnologías a usarse.
- 2.2. Determinar las metodología y herramientas que se adapten al proyecto
- 2.3. Analizar la situación actual y levantar de requerimientos del control de asistencia dentro de la UIDE sede Loja
- 2.4. Diseñar el prototipo para la asistencia docente
- 2.5. Realizar fase de pruebas del prototipo de asistencia

## **D. Alcance**

Los rangos de la visión artificial son extensos en este caso usaremos el reconocimiento facial, en el momento que una cámara detecte el rostro de una persona la/lo reconocerá y mediante el uso de una base de datos comparara y verificara a los docentes marcando la hora de ingreso y salida.

Con los conocimientos adquiridos hemos decidido hacer un prototipo de visión artificial para la asistencia en el que podrán realizar la marcación al momento que ingresen al establecimiento, El alma mater Universidad Internacional del Ecuador sede Loja, consta de varios departamentos, siendo de uso practico para los docentes del área de Tecnologías de la Información.

El sistema de visión artificial permitirá el ingreso de nuevos docentes, horarios, un reporte de marcación por persona y por fecha.

El sistema no controlara registro de fechas espaciales, permisos, vacaciones, fechas recuperables, de acuerdo con la Ley de Trabajo de la Republica del Ecuador.

## **E. Marco Teórico.**

### **CAPÍTULO 2**

#### **2.1 Herramientas Usadas**

##### **2.1.1. OpenCV.**

Open Source Computer Vision es una librería de programación de código libre hecha para permitir aplicaciones que usan visión artificial. Esta librería tiene muchas funciones tales como: captura, análisis y manipulación de datos visuales y también puede eliminar algunos de los grandes problemas que los programadores nos enfrentamos al momento de desarrollar aplicaciones que son dependientes de visión por computadora S. (2021, 25 julio). *¿Qué es OpenCV?* Spiegato. <https://spiegato.com/es/que-es-opencv>.

#### **2.2 Redes Neuronales.**

Las redes neuronales nacieron para imitar el funcionamiento de las neuronas de nuestro cerebro. Dado el caso, las redes neuronales siguen una tendencia diferente al enfoque clásico de inteligencia artificial, que tratan de adaptar la inteligencia en forma de razonamiento humano a un grupo de procesos.

#### **2.3 CNN**

Las redes neuronales convolucionales son un algoritmo utilizado en el campo del aprendizaje automático el cual le da a un ordenador la capacidad de poder ver. Se puede clasificar imágenes, detectar cualquier tipo de cosas, enseñarle a un coche a conducir, etc. Las CNN son una NN de aprendizaje supervisado que intenta replicar al córtex visual humano mediante capas, para así poder identificar características únicas de las entradas que tenga.

#### **2.4 HAAR Features**

Los HAAR features, son filtros o también conocidos como kernels que se obtiene de la sustracción entre la suma de los píxeles entre rectángulos de diverso color, por lo general monocromático, se usa más en el campo de detección de rostros (Martinez, 2020).

#### **2.5 Visión Artificial.**

La visión Artificial la mayoría del tiempo ha caído en caminos similares. Siempre ha sido que un par de cámaras juegan el rol de los ojos, una computadora para que analice las imágenes (Brahath & Petrov, 2008).

Este apartado está diseñado para tener una visión tipo perspectiva mejor a la humana, donde nosotros podremos encontrar patrones en cuerpos minúsculos. Aquí es donde necesitamos sensores más o igual de precisos que nuestros ojos, aunque la realidad es distinta incluso se pueden ejecutar operación con imágenes poco detalladas pero el resultado no será del deseado.

## **2.6 Librerías de Machine Learning**

Tensorflow es una librería de machine learning y computación grafica lanzada por Google. Basada en la interfaz de Python. Es el mecanismo de aprendizaje de inteligencia artificial.

Keras es una librería de código abierto desarrollada en Python, el principal objetivo de esta librería es mejorar el ritmo de creación de las redes neuronales.

Pythorch es otra de las grandes librerías basadas en Python, que se enfoca en la realización de cálculos numéricos mediante la programación de tensores, lo que ayuda mucho en la aplicación para el desarrollo de aplicaciones con Deep learning.

Theano es una biblioteca propia de Python y un compilador que nos ayuda en la optimización para manipular y poder evaluar expresiones matemáticas de una manera más sencilla, especialmente las que son matriciales.

## F. METODOLOGÍA

### 1. Métodos

El desarrollo del proyecto requiere seguir los lineamientos de ciertos métodos, así como de técnicas e instrumentos que permitan la recopilación y análisis de la información necesaria para la presentación del proyecto de tesis, tales como:

- **Método Inductivo.** -Con base al análisis de los problemas encontrados y con la ayuda de este método se pudo determinar cuál es el problema general de investigación.

- **Método Deductivo.** -Una vez determinado el problema general de investigación. Este método nos ayudó a definir los objetivos específicos a obtenerse para dar solución a los problemas encontrados en el desarrollo del prototipo

- **Método Analítico.** -Al momento de obtener información de la situación actual de los docentes que se abstienen al momento de marcar su registro/asistencia en el biométrico, este método nos ayudó a determinar cuáles son las posibles soluciones para dicho problema y decantarnos por la más óptima.

### 2. Metodología de Desarrollo

Para el desarrollo del proyecto se utilizará:

#### La Metodología ICONIX

Iconix deriva directamente del RUP (Proceso Unificado de Desarrollo) y su fundamento es el hecho de que un 80% de los casos pueden ser resueltos tan solo con un uso del 20% del UML (lenguaje de modelado unificado), con lo cual se simplifica muchísimo el proceso sin perder documentación al dejar solo aquello que es necesario. Esto implica un uso dinámico del UML de tal forma que siempre se pueden utilizar otros diagramas además de los ya estipulados si se cree conveniente. Iconix se guía a través de casos de uso y sigue un ciclo de vida iterativo e incremental. El objetivo es que a partir de los casos de uso se obtenga el sistema final.

#### Características:

**Iterativo e incremental:** Suceden iteraciones entre el desarrollo de modelo del dominio y la identificación de los casos de uso. El modelo estático es incrementalmente refinado por los modelos dinámicos.

**Trazabilidad:** Cada paso está referenciado por algún requisito. Se debe considerar a la trazabilidad como la capacidad de seguir una relación entre los diferentes artefactos producidos.

**Dinámica del UML:** Uso dinámico de UML en los diagramas de caso de uso, diagramas de secuencia y de colaboración.

**Tareas:**

- **Análisis de requisitos**
  - 1) Modelo de dominio
  - 2) Prototipación rápida
  - 3) Modelo de casos de uso
  
- **Análisis y diseño preliminar**
  - 1) Descripción de casos de uso
  - 2) Diagrama de robustez
  
- **Diseño**
  - 1) Diagrama de secuencia
  - 2) Completar el modelo estático
  
  
- **Implementación**
  - 1) Utilizar un diagrama de componentes
  - 2) Escribir / Generar código
  - 3) Realización de pruebas

En conclusión, ICONIX es un proceso intermedio entre XP y RUP, siendo el primero muy útil para softwares pequeños y, el segundo, muy útil para softwares industriales; por tanto, ICONIX es una mezcla entre la agilidad de XP y la robustez de RUP.

## G. Cronograma

**Tabla 3.** Cronograma de actividades

Actividades/Semana	Agosto				Septiembre				octubre				Noviembre				Responsable
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	
1 Cap. 1: Análisis de la propuesta • Levantamiento de información sobre el sistema de registro actual • Análisis de la información provista post al levantamiento																	
2 Cap. 2: Estado del arte sobre la solución propuesta • Lectura sobre los mecanismos usados para la creación de la propuesta • Análisis de los mecanismos a usarse																	
3 Cap. 3: Diseñar el modelo del prototipo en cuanto a los requerimientos. • Diseño del modelo • Maquetación del prototipo																	
4 Cap. 4: Realización del prototipo • Construcción del prototipo • Etapa de pruebas • Discusión acerca del sistema antes empleado y el prototipo																	
5 Cap. 5: Conclusiones • Conclusión y Recomendaciones																	
Documentación del proyecto de tesis.																	

## H. Presupuesto

El presupuesto es totalmente cubierto por el estudiante a cargo del proyecto a presentarse, para lo cual se procede a detallar el presupuesto aproximado de los implementos a usarse para la elaboración del presente proyecto

Cámara.....	\$300
Computadora.....	\$1600
Internet.....	\$120
Software.....	\$0
Celular.....	\$400
Copias.....	\$100
Transporte.....	\$100
Otros.....	\$100

Total, aproximado al culmine del presente proyecto: \$2720 USD dólares norteamericanos.

## Bibliografía

- Aprende con ALF. (2020). *Python*. Obtenido de <https://aprendeconalf.es/docencia/python/manual/matplotlib/>
- Braharath, A., & Petrov, M. (2008). *Next Generation Artificial Vision Systems*. Norwood: Imperial College London.
- Challenger, I., Díaz, Y., & Becerra, R. A. (2014). *El lenguaje de programación Python*. Holguin.
- Chiarely, A. (2018). *Beginning React*. Birmingham B3 2PB, Uk: Packt Publishing.
- Data Scientist. (2019). *Scikit-Learn : Descubre la biblioteca de Python dedicada al Machine Learning*. Obtenido de <https://datascientest.com/es/scikit-learn-descubre-la-biblioteca-python>
- Digite. (2021). *Scrum methodology*. Obtenido de <https://www.digite.com/wp-content/uploads/2021/08/scrum-methodology.png>
- Dongwei, X., Shuyu, Z., Yun, M., Deyu, T., & Xuanzhe, L. (2019). Moving Deep Learning into Web Browser: How Far can we go? *Key Lab of High-Confidence Software Technology, MoE*, 3-4.
- Futurespace. (2018). *Redes neuronales*. Obtenido de <https://www.futurespace.es/redes-neuronales-y-deep-learning-capitulo-2-la-neurona/>
- Lawrence, J. (1993). Introduction to neural networks. *California Scientific Software*.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning for AI. *ACM*, 64(7), 58-65.
- Mohedano, J., Saiz, M., & Salazar Roman, P. (2013). *Iniccion a JavaScript*. Madrid: Ministerio de Educacion y Formacion Profesional de España.
- Open CV. (2019). Obtenido de <https://spiegato.com/es/que-es-opencv>
- O'Shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. *arXiv preprint*, 3(2), 1-11.
- Osorio, D., & Garcia, S. (2022). *Sistema biométrico de reconocimiento facial basado en inteligencia artificial*. Santiago de Cali: Universidad Autonoma de Occidente.
- Pavón Puertas, J., & Llarena Borges, E. (2015). *Cracion de un sitio web con PHP y MySQL E. (2015)*. Paracuellos de Jarama, Madrir: RA-MA, S.A. Editorial y Publicaciones.
- Pyspanish. (2017). *Serialización de objetos Python*. Obtenido de <https://pyspanishdoc.sourceforge.net/lib/module-pickle.html>

- Rouhiainen, L. (2018). *Inteligencia artificial*. Madrid: Alienta.
- Sandoval, L. (2018). Algoritmos de aprendizaje automático para análisis y predicción de datos. *REDICCES*, 11(2), 36-41. Obtenido de [http://redicces.org.sv/jspui/bitstream/10972/3626/1/Art6\\_RT2018.pdf](http://redicces.org.sv/jspui/bitstream/10972/3626/1/Art6_RT2018.pdf)
- Tsoi, A. C. (2006). Recurrent neural network architectures: an overview. Adaptive processing of sequences and data structures: international summer school on neural networks “er caianiello” vietri sul mare. *Salerno*, 5(2), 1-26.
- WHO. (2017). *MN68*. Obtenido de <https://who.canto.global/v/MN68KSP9SD/album/NNBOS?display=curatedView&viewIndex=0>
- Dalton, la extensión de Chrome que ayuda a los daltónicos*. (2019). <https://www.trespm.mx/curiosidades/tecnologia/dalton-la-extension-de-chrome-que-ayuda-a-los-daltonicos>
- Osipyan, H., Edwards, B. I., & Cheok, A. D. (2022). Deep Neural Network Applications. In *Deep Neural Network Applications*. CRC Press. <https://doi.org/10.1201/9780429265686>
- Pelaez Chavez Norma. (2012). *Aprendizaje no supervisado y el algoritmo wake-sleep en redes neuronales*. Universidad Tecnológica de la Mixieca.
- Martinez, J. (2020, 2 julio). *Detección de Rostros a la Vieja Usanza Con Haar Cascades*. DataSmarts Español. <https://datasmarts.net/es/deteccion-de-rostros-a-la-vieja-usanza-con-haar-cascades/>

## 5.2. Cuestionario

## Cuestionario Tesis de Grado Eduardo Araujo

3  
Respuestas

04:14  
Tiempo medio para finalizar

Activo  
Estado

1. ¿A qué departamento pertenece? (0 punto)

<span style="color: blue;">●</span> Directores de Camera	1
<span style="color: orange;">●</span> Docencia	2
<span style="color: green;">●</span> Administrativo	0



2. ¿Cómo realiza el proceso el registro de asistencia del personal que está a su cargo? (0 punto)

3  
Respuestas

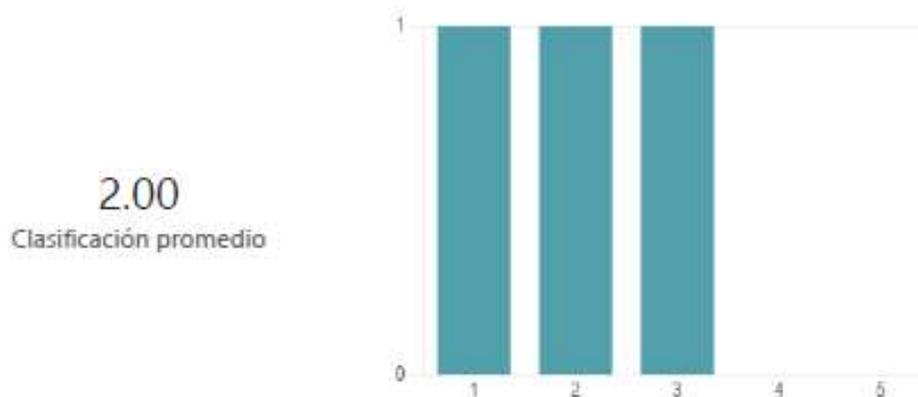
Respuestas más recientes

*"Manual, por registro de marcaciones de clases en el sistema..."*

*"Mediante el Portal Docente (Seguimiento de Syllabus)"*

*"No registro asistencia"*

3. ¿Cómo calificaría el proceso antes mencionado? 1: es malo; 5: Muy bueno (0 punto)



4. ¿Qué personas están involucradas en el proceso de asistencia? (0 punto)



5. ¿Qué Tecnologías están disponibles en la Universidad? (0 punto)



6. ¿Cuál es la apertura de implementación de nuevas herramientas tecnológicas (0 punto) dentro de la Institución?

