

Instituto Tecnológico de las Américas ITLA



Prof. Willis Ezequiel Polanco Caraballo

Programación III

Proyecto Visión Computacional – Análisis y Diseño

Grupo #1

Integrantes:

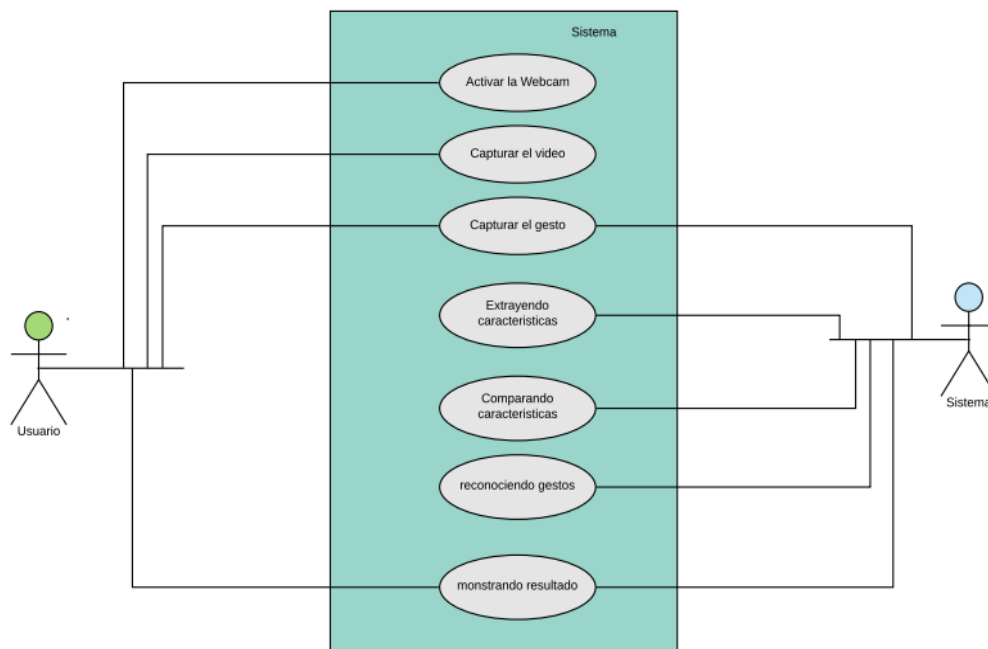
Ángel Martin Mongrut Tejeda	2018-7151
Carlos Daniel Soriano Rosa	2018-6756
Juan Manuel Pérez Ramírez	2018-6671
Karen Onaly Bautista Terrero	2018-6564
Saul Johan Alonzo Placencia	2018-6764
Starling Daniel Rosario Franco	2018-6931

❖ Propósito del Proyecto (Casos de uso)

El propósito de este proyecto es esencialmente aplicar conocimientos de visión computacional en el desarrollo de un sistema capaz de reconocer los gestos de auxilio hechos por una persona a la cámara con una de sus manos.

Esto puede ser utilizado en diversos casos de la vida real, uno de ellos puede ser: Implementar el sistema a uno que monitoree una zona considerada peligrosa o de riesgo y se mantenga analizando si logra detectar algún gesto de auxilio desde dicha zona; También puede ser usado en cámaras de vigilancia, que al hacer un gesto específico a la cámara se realiza una llamada automática a la policía local con un mensaje previamente preparado, para pedir auxilio de manera automática y rápida en caso de alguna emergencia como robos, incendios, etc.

Con esto llegamos a los casos de uso del sistema; como este debe ser lo más general posible para lograr adaptarse a las diferentes aplicaciones que pueda llegar a tener terminamos con esto:

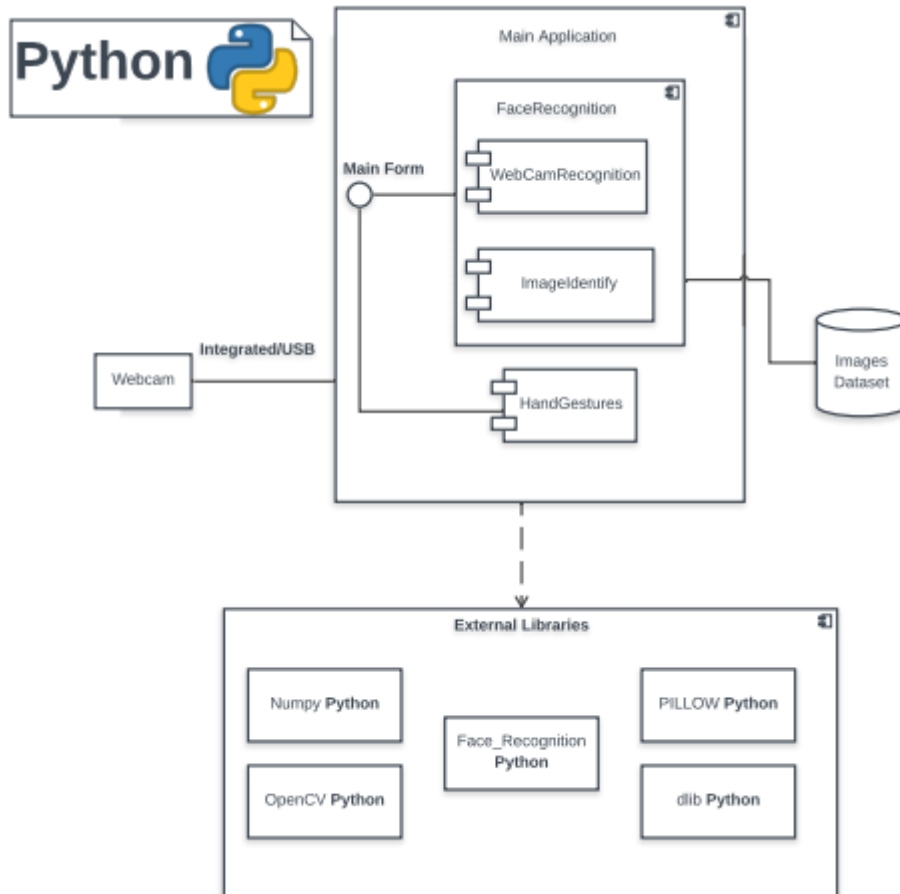


Contando con un caso de uso con tal nivel de generalidad logramos dar total libertad a la imaginación, permitiendo con esto dar una mayor gama de aplicaciones posible al sistema, y no simplemente cerrándolo a un único uso.

❖ Composición del Sistema (Diagrama de Arquitectura)

La tecnología elegida para el desarrollo de esta aplicación es Python, haciendo uso de varias de las librerías que este tiene a disposición.

Demos un vistazo al diagrama de componentes, en high level design (para una mejor resolución referirse al archivo adjunto, “Diagrama de arquitectura.pdf”):

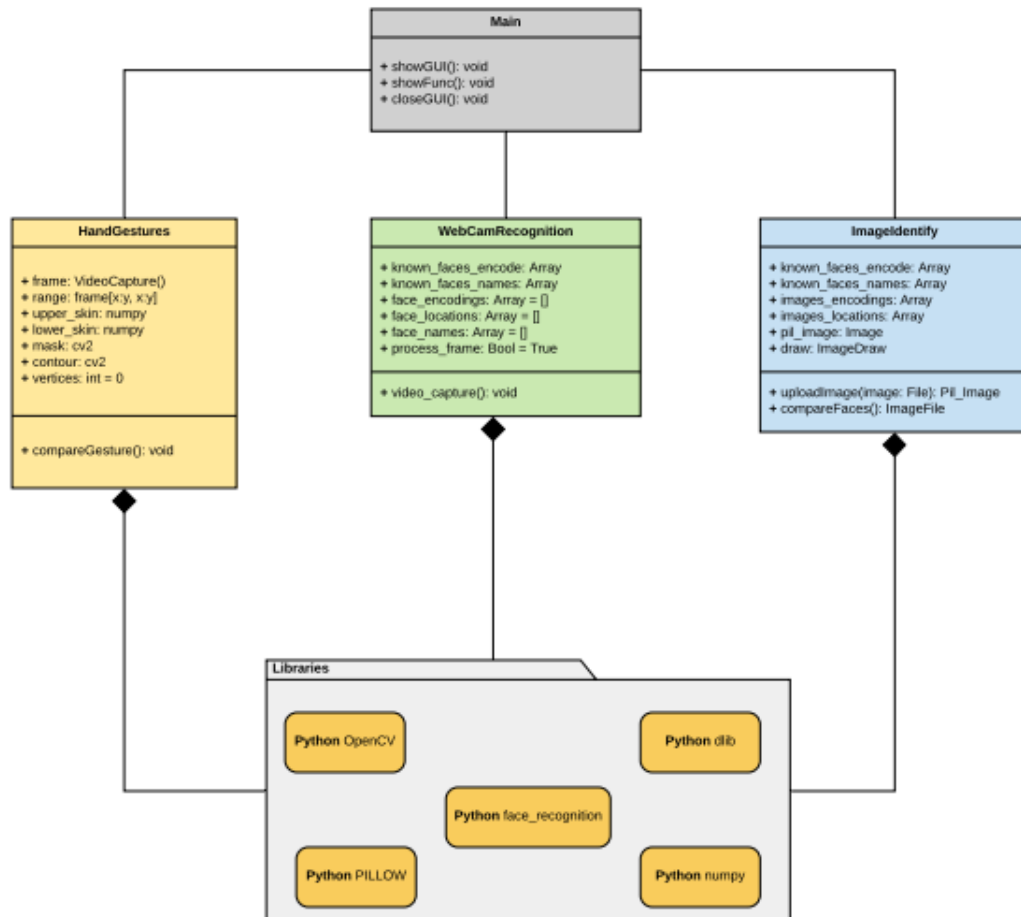


El sistema toma como un punto de inicio un formulario principal, el “Main Form” desde el cual se tiene acceso a dos funciones, reconocer gestos de una mano en la cámara del dispositivo o identificar un rostro conocido en la cámara o a partir de una imagen. Para esto último hacemos uso de varias librerías externas las cuales son requeridas para el correcto funcionamiento del sistema; además de esto, los paquetes de Visual C++ Redistributable son requeridos para cualquier acción de OpenCV, por lo tanto es obligatorio para la correcta ejecución del sistema.

❖ Lógica Interna (Diagrama de Clases)

Continuando con la explicación fundamental del sistema llegamos a la sección de la lógica interna de este, que al contrario de lo que aparenta es más lógica matemática que lógica de programación en sí.

Empecemos revisando el diagrama de clases (para una mejor resolución referirse al archivo adjunto, “Diagrama de clases.pdf”):



Comencemos con la clase encargada del reconocimiento de los gestos manuales, **HandGestures**, esta clase utiliza las librerías *OpenCV* y *numpy* para lograr reconocer a tiempo real los gestos efectuados con una mano frente a la cámara y esto se logra de la siguiente manera:

Primero, utilizando *opencv* se accede a la cámara del dispositivo y en esta se extrae un “*frame*”. Luego, se especifica el área de interés, que es la sección donde evaluaremos la mano que entre en esta. A partir de esta área de interés es que definirá la lógica de acción para encontrar lo que matemáticamente se podría llamar “gesto”. En el área de interés se define el tono de piel requerido, este es un color en formato RGB, y se forma una máscara binaria en el área de interés. El tono de piel será considerado los ‘1s’ de esta máscara, por lo que todo lo demás serán ‘0s’, con esto obtenemos una figura geométrica que podemos evaluar matemáticamente,

cuya forma será la que tenga nuestra mano en ese instante. La figura será evaluada extrayendo los vértices de un ángulo formado entre los dedos de la mano, al calcular la cantidad de estos vértices y sus valores, como los grados entre estos se puede predecir el gesto que la mano está formando.

Pasemos a los reconocimientos faciales, las clases **WebCamRecognition** (encargada de reconocer rostros en tiempo real mediante la cámara) e **ImageIdentify** (encargada de reconocer rostros en imágenes) se ocupan de esto haciendo uso de las librerías Python *face_recognition* y *PILLOW*.

Gracias a estas librerías el trabajo es mucho más sencillo, al entrenar el algoritmo de reconocimiento propiciado por *face_recognition* con distintas imágenes de personas, especificando sus nombres, obtenemos una potente herramienta de detección, que haciendo uso de los rasgos faciales de la persona, encuentra la similitudes entre las caras que conoce y las caras que está viendo. A partir de aquí es simplemente implementar esa herramienta, usando opencv la conectamos a la cámara y dibujamos los rectángulos para seccionar las caras en pantalla, estas se evalúan y se escribe su nombre en el recuadro inferior. Para reconocer rostros a partir de imágenes es muy similar, usando las clases de PILLOW Image e ImageDraw, creamos una imagen temporal idéntica a la original y dibujamos sobre esta los rectángulos sobre las caras y sus nombres.