



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

Diseño, implementación y pruebas de desempeño de un sistema bajo costo para emular las funciones del ratón de un computador a través de gestos faciales utilizando algoritmos de visión artificial y reconocimiento de voz

Valencia Aragón, Kevin Gabriel

Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Ingeniería en Electrónica, Automatización y Control

Trabajo de titulación, previo a la obtención del título de Ingeniero en Electrónica, Automatización y Control

Ing. Ramos Vargas, Pablo Francisco PhD.

21 de julio del 2022



Tesis_Kevin_Valencia.pdf

Scanned on: 16:25 July 6, 2022 UTC



Overall Similarity Score



Results Found



Total Words in Text

Identical Words	465
Words with Minor Changes	82
Paraphrased Words	89
Omitted Words	0



Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Ingeniería en Electrónica, Automatización y Control

Certificación

Certifico que el trabajo de titulación: **“Diseño, implementación y pruebas de desempeño de un sistema bajo costo para emular las funciones del ratón de un computador a través de gestos faciales utilizando algoritmos de visión artificial y reconocimiento de voz”** fue realizado por el señor **Valencia Aragón, Kevin Gabriel**; el mismo que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, además fue revisado y analizado en su totalidad por la herramienta de prevención y/o verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que se lo sustente públicamente.

Sangolquí, 21 de julio del 2022

Firma



Firmado electrónicamente por:
**PABLO FRANCISCO
RAMOS VARGAS**

Ing. Ramos Vargas, Pablo Francisco PhD

C. C.: 1712447976



Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Ingeniería en Electrónica, Automatización y Control

Responsabilidad de Autoría

Yo, **Valencia Aragón, Kevin Gabriel**, con cédula de ciudadanía n°1724301583, declaro que el contenido, ideas y criterios del trabajo de titulación: **“Diseño, implementación y pruebas de desempeño de un sistema bajo costo para emular las funciones del ratón de un computador a través de gestos faciales utilizando algoritmos de visión artificial y reconocimiento de voz”** es de mi autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Sangolquí, 21 de julio del 2022

Firma

.....*gabriel Valencia*.....

Valencia Aragón Kevin Gabriel

C.C.: 1724301583



Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Ingeniería en Electrónica, Automatización y Control

Autorización de Publicación

Yo **Valencia Aragón, Kevin Gabriel**, con cédula de ciudadanía n°1724301583, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **"Diseño, implementación y pruebas de desempeño de un sistema bajo costo para emular las funciones del ratón de un computador a través de gestos faciales utilizando algoritmos de visión artificial y reconocimiento de voz"** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Sangolquí, 21 de julio del 2022

Firma

.....*Kevin Gabriel Valencia Aragón*.....

Valencia Aragón, Kevin Gabriel

C.C.: 1724301583

Dedicatoria

Este trabajo está dedicado a mis padres, César y Dora quienes a lo largo de mi vida me han educado para ser una persona de bien y siempre han estado apoyándome incondicionalmente en todo lo que me he propuesto.

Agradecimiento

En primer lugar, agradezco a mis padres quienes a lo largo de toda mi etapa estudiantil han estado presentes, dándome todo el apoyo que he necesitado y la motivación para seguir adelante con mis sueños. Agradezco a mis hermanos Dennis y Christian quienes han estado a mi lado toda mi vida y me han ayudado a llegar a ser la persona que soy.

A mi abuelita Blanca y a mi tía Xime, por siempre haberme dado su apoyo y motivación, por siempre estar pendiente de mí, cuando más lo necesitaba y haberme dado tantas lecciones.

A Lesly, quien durante toda esta etapa estuvo siempre presente y dispuesta a apoyarme de forma incondicional. Agradezco cada ayuda y consejo que me dio para lograr cumplir esta meta.

Agradezco a todos mis profesores y a mi tutor, Ingeniero Pablo Ramos por su paciencia y consejos, los cuales que me ayudaron a finalizar este trabajo.

Finalmente quiero agradecer a todos mis amigos, con quienes compartí tiempo en las aulas y fuera de ellas, ya que sin ellos esta etapa no hubiera sido lo que es.

Índice de contenidos

Carátula	1
Análisis de similitud de contenidos.....	2
Certificación	3
Responsabilidad de autoría.....	4
Autorización de publicación.....	5
Dedicatoria.....	6
Agradecimiento.....	7
Índice de contenidos.....	8
Índice de tablas	14
Índice de figuras.....	15
Resumen	18
Abstract.....	19
Capítulo I. Generalidades.....	20
Justificación e importancia	20
Objetivos	21
Objetivo general.....	21
Objetivos específicos.....	21
Alcance	22
Capítulo II. Fundamentos teóricos y estudio del estado del arte	23

Fundamentos teóricos	23
Visión artificial.....	23
Reconocimiento de voz.....	25
Estudio del estado del arte	27
Técnicas para lograr la emulación del mouse.....	29
Eye tracking.....	29
Face tracking	33
Otras técnicas.....	37
Capítulo III. Desarrollo del software	41
Lenguaje de programación	41
Python	41
Librerías aplicadas.....	42
Pyautogui	42
Time	42
Threading	43
Queue.....	43
Pyaudio.....	44
Vosk.....	44
Json	44
Pygame.....	45

	10
OpenCV	45
Dlib	45
Numpy.....	46
Imutils.....	46
Desarrollo y características	47
Hilo “video” y modulo “utils”	48
Hilo “audio”	61
Hilo “comandos”	65
Hilo “hablarcompu” y Función “Compu_Habla2”	69
Capítulo IV. Características del Hardware y plataformas de prueba.....	72
Características generales de hardware.....	72
Procesador y características generales.....	72
Cámara	73
Micrófono.....	73
Otras plataformas de prueba.....	73
Toshiba Satellite L 15W-B	73
Biostar H61MHB.....	74
Lenovo 81D6	74
Dell Inspiron 15 3515	74
ASUS K53E	75

	11
Dell G3 3579.....	75
Toshiba Satellite P55W-C.....	75
HP - HP Laptop 15-db1xxx.....	75
Toshiba Satellite P75-A7200	76
Capítulo V. Pruebas de desempeño y análisis de resultados.....	77
Pruebas generales.....	77
Variación de iluminación.....	77
Prueba con varios rostros en el frame	81
Prueba con voces de más de una persona.....	82
Pruebas de inicio automático	83
Pruebas de desempeño en distintas plataformas	84
Pruebas en computador Biostar H61MHB.....	85
Pruebas en computador Lenovo 81D6	87
Pruebas en computador Dell Inspiron 15 3515	88
Pruebas en computador ASUS Vivobook 15	90
Pruebas en computador Dell G3 3579	91
Pruebas en computador Toshiba Satellite P55W-C	93
Pruebas en computador Toshiba Satellite L 15W-B.....	95
Pruebas en computador ASUS K53E	96
Pruebas en computador HP – HP Laptop 15-db1xxx	97

Pruebas en computador Toshiba Satellite P75-A7200	98
Pruebas en Linux.....	102
Pruebas en tarjeta Raspberry Pi 3 B.....	105
Pruebas en usuarios.....	108
Pregunta 1: ¿Qué tan funcional considera que es el emulador para personas con discapacidades motrices?.....	108
Pregunta 2: ¿Qué tan difícil fue la adaptación al uso del emulador?	109
Pregunta 3: ¿Qué tan avanzados son sus conocimientos en cuanto al manejo del computador?	111
Pregunta 4: ¿Qué tan difícil fue la realización del ejercicio?	112
Pregunta 5: ¿Considera que la velocidad de movimiento del cursor es adecuada para facilitar la selección de objetos pequeños?	114
Pregunta 6: ¿Qué tan útiles le resultaron los comandos de segmentación de pantalla?	115
Pregunta 7: ¿Cuántas veces tuvo que reiniciar el movimiento del cursor (manteniendo abierto la boca) para poder realizar el ejercicio?	116
Pregunta 8: ¿Qué tan preciso considera que es la detección de los comandos de voz en su computador?.....	117
Pregunta 9: ¿Utilizó un micrófono externo para realizar la prueba?	117
Pregunta 10: ¿Qué tan preciso considera que fue la detección de su rostro?.....	118
Pregunta 11: ¿Utilizó una cámara externa para realizar la prueba?	118
Pregunta 12: Cronometrar el tiempo que tarda en realizar el ejercicio.....	119

Pregunta 13: ¿Tuvo dificultades a la hora de realizar la prueba?.....	120
Limitaciones del emulador.....	120
Capítulo VI. Conclusiones y recomendaciones	122
Conclusiones	122
Recomendaciones.....	125
Trabajos futuros.....	126
Bibliografía	127
Apéndices.....	134

Índice de tablas

Tabla 1 <i>Comandos generales para realizar acciones con el emulador</i>	64
Tabla 2 <i>Comandos exclusivos para modo dictado</i>	65
Tabla 3 <i>Comandos de segmentación y posicionamiento</i>	67
Tabla 4 <i>Resultados de pruebas en diferentes computadores</i>	100

Índice de figuras

Figura 1 <i>Puntos de la cara que proporciona el predictor de Dlib</i>	47
Figura 2 <i>Esquema general del funcionamiento del emulador</i>	48
Figura 3 <i>Preprocesamiento, generación de espejo en el video</i>	50
Figura 4 <i>Preprocesamiento, conversión a escala de grises</i>	51
Figura 5 <i>Representación de contorno y coordenadas de la boca</i>	53
Figura 6 <i>Relación de aspecto de la boca (MAR)</i>	54
Figura 7 <i>Diagrama de flujo de la función "mouth_aspect_ratio" del módulo "utils"</i>	55
Figura 8 <i>Rectángulo y línea de referencia</i>	57
Figura 9 <i>Diagrama de flujo de la función "direction" del módulo "utils"</i>	59
Figura 10 <i>Diagrama de flujo del hilo "video"</i>	60
Figura 11 <i>Diagrama de flujo de la función "Reconocer_Voz" del hilo "audio"</i>	63
Figura 12 <i>Posición de los segmentos en que se divide la pantalla</i>	66
Figura 13 <i>Diagrama de flujo de la función "Ejecutar_Comandos" del hilo "comandos"</i>	68
Figura 14 <i>Diagrama de flujo de la función "Compu_Habla" del hilo "hablarcompu"</i>	70
Figura 15 <i>Diagrama de flujo de la función "Compu_Habla2"</i>	71
Figura 16 <i>Micrófono Genius MIC-01A</i>	73
Figura 17 <i>Webcam Adesso CyberTrack H4</i>	74
Figura 18 <i>Rostro con iluminación de 469 luxes</i>	78
Figura 19 <i>Rostro con iluminación de 107 luxes</i>	78
Figura 20 <i>Rostro con iluminación de 19 luxes</i>	79
Figura 21 <i>Rostro con iluminación de 5 luxes</i>	80
Figura 22 <i>Frame con más de un rostro (Funcionamiento normal)</i>	82
Figura 23 <i>Frames con más de un rostro (Funcionamiento erróneo)</i>	82

Figura 24 <i>Detalles básicos del computador Biostar H61MHB consultados usando el emulador</i>	85
Figura 25 <i>Costo computacional del emulador en Biostar H61MHB</i>	86
Figura 26 <i>Detalles básicos del computador Lenovo 81D6 consultados usando el emulador</i>	87
Figura 27 <i>Costo computacional del emulador en Lenovo 81D6</i>	88
Figura 28 <i>Detalles básicos del computador Dell Inspiron 15 3515 consultados usando el emulador</i>	89
Figura 29 <i>Costo computacional del emulador en Dell Inspiron 15 3515</i>	89
Figura 30 <i>Detalles básicos del computador ASUS Vivobook 15 consultados usando el emulador</i>	90
Figura 31 <i>Costo computacional del emulador en ASUS Vivobook 15</i>	91
Figura 32 <i>Detalles básicos del computador Dell G3 3579 consultados usando el emulador</i>	92
Figura 33 <i>Costo computacional del emulador en Dell G3 3579</i>	93
Figura 34 <i>Detalles básicos del computador Toshiba Satellite P55W-C consultados usando el emulador</i> ..	94
Figura 35 <i>Costo computacional del emulador en Toshiba Satellite P55W-C</i>	94
Figura 36 <i>Error generado al intentar usar el emulador en computador Toshiba Satellite L 15W-B</i>	95
Figura 37 <i>Detalles básicos del computador Toshiba Satellite L 15W-B</i>	95
Figura 38 <i>Error generado al intentar usar el emulador en computador ASUS K53E</i>	96
Figura 39 <i>Detalles básicos del computador ASUS K53E</i>	96
Figura 40 <i>Detalles básicos del computador HP – HP Laptop 15-db1xxx consultados usando el emulador</i>	97
Figura 41 <i>Costo computacional del emulador en HP – HP Laptop 15-db1xxx</i>	98
Figura 42 <i>Detalles básicos del computador Toshiba Satellite P75-A7200 consultados usando el emulador</i>	99
Figura 43 <i>Costo computacional del emulador en Toshiba Satellite P75-A7200</i>	99
Figura 44 <i>Características de la máquina virtual</i>	102
Figura 45 <i>Comandos para ejecutar el Emulador en Linux Mint 20.3</i>	103
Figura 46 <i>Funcionamiento del emulador en Linux Mint 20.3</i>	104

Figura 47 Costo computacional en Máquina Virtual con Linux Mint 20.3	104
Figura 48 Raspberry Pi 3 B conectada a dispositivos periféricos de entrada	105
Figura 49 Comandos utilizados para ejecutar el emulador en Raspberry Pi 3 B.....	106
Figura 50 Emulador funcionando en Raspberry Pi 3 B	107
Figura 51 Costo computacional del emulador en Raspberry Pi 3 B.....	107
Figura 52 Resultados Pregunta 1	109
Figura 53 Resultados Pregunta 2	109
Figura 54 Resultados por edades, Pregunta 2.....	110
Figura 55 Resultados Pregunta 3	111
Figura 56 Resultados por edades, Pregunta 3.....	112
Figura 57 Resultados Pregunta 4	113
Figura 58 Resultados por edades, Pregunta 4.....	114
Figura 59 Resultados Pregunta 5	115
Figura 60 Resultados Pregunta 6	115
Figura 61 Resultados pregunta 7	116
Figura 62 Resultados Pregunta 8	117
Figura 63 Resultados Pregunta 9	118
Figura 64 Resultados Pregunta 10	119
Figura 65 Resultados Pregunta 11	119

Resumen

Hoy en día, saber usar una computadora es casi tan importante como saber leer y escribir. Para operar eficientemente estas máquinas, se requiere el uso de las manos para interactuar con el mouse y el teclado. Sin embargo, hay personas que no tienen la capacidad de utilizar sus miembros superiores al 100% por una u otra razón. Debido a esto, el presente trabajo tiene como objetivo desarrollar un emulador de mouse basado en visión artificial y reconocimiento de voz, para reemplazar la necesidad de usar las manos para controlar el mouse de una computadora. Se utilizan herramientas de visión artificial y reconocimiento de voz para operar el ordenador mediante movimientos de cabeza y comandos de voz específicos. Se han aplicado diferentes técnicas para hacer más cómodo y útil el uso de este emulador. Por ejemplo, se utilizan comandos para la segmentación de pantalla y el modo de dictado. Este trabajo explica en detalle el desarrollo del emulador, sus características y funcionalidades. Para definir el rango de utilidad del emulador se realizaron pruebas de desempeño en diferentes ambientes de trabajo para determinar cuáles son las condiciones óptimas de funcionamiento. Además, se probó el emulador en varias computadoras con diferentes características de hardware y software, para observar los recursos computacionales utilizados por el programa, así como determinar las características mínimas para su correcto funcionamiento. Finalmente, se realizó una encuesta a usuarios de diferentes edades que han probado el emulador, para comprobar su utilidad. Los resultados de la encuesta muestran que el emulador es fácil de usar, práctico, bastante preciso, funciona en una amplia gama de computadoras y no sobrecarga el sistema.

Palabras Clave: detección de rostros, reconocimiento de voz, emulador de mouse.

Abstract

Nowadays, knowing how to use a computer is almost as important as knowing how to read and write. To efficiently operate these machines, the use of the hands is required to interact with the mouse and keyboard. However, there are people who do not have the ability to use their upper limbs at 100% for one reason or another. Due to this, the present work aims at developing a mouse emulator based on artificial vision and voice recognition, for replacing the need of using the hands to control a computer mouse. Artificial vision and voice recognition tools are used to operate the computer by means of head movements and specific voice commands. Different techniques have been applied to make the use of this emulator more comfortable and useful. For example, commands for screen segmentation and dictation mode are used. This work explains in detail the development of the emulator, its characteristics and functionalities. In order to define the range of utility of the emulator, performance tests were carried out in different work environments to determine what the optimal operating conditions are. In addition, the emulator was tested on several computers having different hardware and software features, to observe the computational resources used by the program, as well as to determine the minimum characteristics for its correct operation. Finally, a survey was carried out on different age users who have tested the emulator, to verify its utility. Results of the survey show that the emulator is easy to use, practical, fairly accurate, works on a wide range of computers and does not overload the system.

Key words: face tracking, speech recognition, mouse emulator.

Capítulo I. Generalidades

Justificación e importancia

El uso de la computadora cada día se vuelve más importante debido a que es una herramienta indispensable para poder estar conectado a través de la red mundial del internet. En los tiempos de la pandemia, su uso ha sido mucho más frecuente, siendo fundamental en los procesos de aprendizaje y representando una de las mejores alternativas para lograr la comunicación y mantener el distanciamiento social. Estos dispositivos están diseñados para ser controlados a través del uso de las extremidades superiores del cuerpo, siendo esto un grave problema para las personas con discapacidades motrices, quienes no tienen la capacidad de usar sus manos, por lo que se genera una gran barrera en los procesos para adquirir conocimientos, donde la herramienta principal es el computador. Por esta razón es indispensable buscar alternativas que sean accesibles y faciliten el manejo del computador por parte de este grupo de personas, sin la necesidad de adquirir dispositivos externos que puedan resultar incómodos y costosos para los usuarios. Entonces, se puede considerar que la opción más adecuada para lograr estos objetivos, es la implementación de algoritmos de visión artificial que permitan detectar cualquier parte de la cara, y combinarlo con reconocimiento de voz para utilizar comandos que faciliten la emulación de las acciones que comúnmente se realizan con el mouse. Para esto, solo se necesitaría que el computador tenga implementada una cámara y un micrófono, los cuales son componentes que hoy en día la mayoría poseen. De forma general, es indispensable que se busquen e implementen métodos y tecnologías de bajo costo que se enfoquen en la utilización de otras partes del cuerpo como la cabeza; con la finalidad de que puedan servir como herramientas de control y poder simular el funcionamiento del mouse, haciendo de esta manera el uso de la computadora más inclusivo y fácil para las personas con discapacidades motrices. Debido a esto, se ha optado por aplicar algoritmos de visión artificial, específicamente de *face tracking*, para detectar distintos gestos faciales, en conjunto con comandos de voz, para lograr obtener un sistema preciso, fácil de usar y que genere libertad de movimiento al usuario.

Además, se considera que su implementación requiere únicamente de los periféricos de uso común en el computador.

Objetivos

Objetivo general

- Diseñar, implementar y validar un sistema que permita emular las funcionalidades de un ratón, utilizando algoritmos de visión artificial y de reconocimiento de voz, enfocado para personas con discapacidades motrices.

Objetivos específicos

- Realizar el estudio del estado del arte de los métodos existentes, para emular el funcionamiento del periférico ratón de un computador.
- Implementar algoritmos de visión artificial que permitan detectar diferentes gestos faciales, que puedan funcionar como comandos para emular las funcionalidades de un ratón.
- Utilizar algoritmos de reconocimiento de voz en conjunto a segmentación de pantalla, con la finalidad de mejorar la velocidad y la precisión al momento de posicionar el cursor en zonas específicas.
- Desarrollar un software que combine las funcionalidades de los algoritmos de visión artificial y reconocimiento de voz para emular el funcionamiento del *mouse*
- Realizar diferentes pruebas para medir su funcionalidad y validar su desempeño bajo distintas situaciones.

Alcance

En el presente proyecto de titulación se plantea el desarrollo de un sistema que permitirá emular las funciones de un *mouse*, teniendo como población objetivo a las personas con discapacidades motrices. Como base fundamental está la aplicación de algoritmos de visión artificial, en combinación con reconocimiento de voz, para poder generar un sistema que pueda ser preciso y fácil de usar. Se plantea que el uso del *mouse* sea general, lo que quiere decir que se permitirá utilizar para todas las funciones del computador, y no esté limitada a aplicaciones específicas. La mayor contribución de este trabajo es la unión de visión artificial, reconocimiento de voz y segmentación de pantalla, para lograr un sistema que permita emular la funcionalidad del *mouse*, a través de los movimientos de la cabeza y realizar las acciones comunes como el *click*, *doble click*, *click derecho* o *scroll*; mediante el uso de comandos de voz que también permitirán realizar la segmentación de pantalla, la cual consiste en dividirla en diferentes zonas de interés que responden a un comando específico, esto con la finalidad de que se pueda aumentar la velocidad y la precisión del posicionamiento del cursor. Adicional a las funciones del mouse se pretende implementar un modo de dictado simple, que permita escribir texto. A través de la cámara del computador se detectarán gestos faciales, que serán vinculados a diferentes acciones relacionadas al movimiento del cursor y con diferentes comandos de voz, se pretende realizar las acciones propias del *mouse* y mejorar el posicionamiento del cursor. Usando el mismo módulo de reconocimiento de voz, se implementará el modo dictado. Debido a que se busca realizar un sistema de bajo costo, el desarrollo del mismo estará enfocado para ser utilizado en computadoras personales de gama media.

Capítulo II. Fundamentos teóricos y estudio del estado del arte

En este capítulo se explicará los conceptos y fundamentos teóricos que están involucrados en el desarrollo del sistema propuesto, además se realizará el estudio del estado del arte, en donde se presentarán varios trabajos significativos disponibles en la literatura, que están relacionados con la temática de este trabajo de titulación.

Fundamentos teóricos

En el presente proyecto se plantea el desarrollo de un sistema que permita controlar el cursor del computador a través de la implementación de técnicas, basadas en visión artificial y reconocimiento de voz, por lo que se considera que el estudio de dichos campos, es de gran importancia para comprender las bases con las que se desarrolló el proyecto.

Visión artificial

El concepto de visión artificial o visión por computador, surge a partir del interés de la comunidad científica en dotar a las computadoras de la capacidad para realizar actividades relacionadas con la inteligencia humana, como el análisis de información visual. Está enfocada en la programación de las máquinas para que puedan procesar y entender imágenes o videos. De la misma forma que el ser humano logra captar imágenes a través de los ojos, la herramienta principal para que los computadores cumplan la misma función, vienen a ser las cámaras (Alvear et al., 2017).

La visión artificial son todos los procesos que se involucran en la adquisición, el tratamiento y la interpretación de una imagen por parte de un computador, iniciando por la captura de la imagen a través de algún dispositivo similar a una cámara. Después de obtener la imagen, se la trata en una etapa de preprocesamiento, en donde se aplican técnicas para mejorar el contraste, la nitidez o realzar características y detalles. Posteriormente la imagen es segmentada en distintos objetos o zonas,

compuestas por grupos de píxeles que sean objetivo de análisis, en una siguiente etapa en donde se describen las características más relevantes, como por ejemplo color, perímetro, forma, entre otros; todo esto con la finalidad de poder diferenciarlos entre ellos. A partir de las descripciones obtenidas en el anterior proceso, se generan distintas categorías y se llega a una etapa de reconocimiento, que es en donde se clasifica los diferentes objetos de la imagen. Por último, se llega al proceso de interpretación, que es en donde se da un significado a las diferentes categorías de objetos que se encontraron en la imagen (García & Caranqui, 2015).

El campo de la visión artificial es amplio y se encuentra en constante estudio debido a las grandes facilidades que su aplicación puede brindar, por lo que actualmente se la puede encontrar implementada en sistemas de seguridad, detección y análisis de movimientos, interacción hombre-máquina, inspección y control de calidad, también está presente en otros campos como por ejemplo en la astronomía, para poder observar diferentes objetos en el cosmos, en el campo de la robótica, ya que facilita la navegación automática de las máquinas, en la medicina para poder diagnosticar diferentes patologías provenientes de resonancias, mamografías, entre otras (García & Caranqui, 2015; Sanabria S & Archila D, 2011).

Utilizando visión artificial se han desarrollado varios sistemas que están basados en movimientos oculares, los cuales utilizan como referencias principales la dirección en la que mira la persona y el ángulo de la mirada. Gran parte de los dispositivos que tienen esta finalidad se basan en el principio de la videooculografía (VOG), que consiste en registrar la imagen del ojo mediante una cámara para luego ser procesada por un software que determina la dirección de la mirada. Otra forma de realizar esta técnica es a través de la electrooculografía (EOG) que, por su parte, logra captar el movimiento ocular a través de la medición de la diferencia de potencial entre la retina y la córnea para lo cual se utilizan electrodos. La oculografía infrarroja (IROG) es otro de los principios con los que se logran estos resultados, y consiste en detectar el reflejo de una luz infrarroja sobre el ojo, permitiendo detectar la posición del iris, para lograr este cometido se utilizan LED's como emisores de infrarrojos y cámaras como receptores. Estas

metodologías, dependiendo de cómo se la implemente, registran algunos inconvenientes, como puede ser su baja resolución, la dificultad de operar con niveles de luminosidad poco adecuados, las dificultades presentadas por los parpadeos; los cuales hacen que se modifique la cantidad de luz que se refleja en los ojos por tiempos cortos dificultando la detección, y por último, el hecho de requerir de potentes equipos de reconocimiento de imagen, donde se incluye el hardware y el software, los cuales pueden llegar a ser muy costosos (Freire et al., 2013).

Otra gran rama del campo de la visión artificial es el que se enfoca en el reconocimiento de rostros en imágenes o videos. Existen diferentes maneras de realizar el seguimiento facial, las cuales pueden ser a través de técnicas basadas en apariencias, y otras que están basadas en modelos. En el primer caso, los algoritmos no requieren que haya un conocimiento previo del rostro para lograr el seguimiento, se sigue cada una de las características del rostro utilizando modelos obtenidos a través de entrenamiento de imágenes, lo que convierte a cada una en un vector de características. Por otro lado, las técnicas basadas en modelos, requieren el conocimiento previo del rostro para poder realizar el seguimiento, se busca generar un modelo que sea lo más descriptivo posible para poder detectar los diferentes movimientos faciales; esta metodología presenta algunas desventajas a comparación de las que se basan en la apariencia, debido a que requieren mayor resolución de la imagen, mayor costo computacional, tienden a fallar cuando hay cambios faciales bruscos en aplicaciones de tiempo real, y son más complejos y lentos (Espinoza & Jorquera, 2015; Gimeno, 2010; Zuluaga et al., 2009).

Reconocimiento de voz

El reconocimiento de voz hace referencia a la capacidad que tienen los computadores para recibir y convertir la voz humana a código binario, para que pueda ser comprendida por la máquina (Barrios et al., 2018). Para lograr el objetivo de que la computadora entienda lo que se le ha dicho, las señales de voz tendrán que pasar por una etapa de preprocesamiento, la cual estará constituida por un micrófono para

captar las ondas sonoras y convertirlas en señales eléctricas, las cuales deberán pasar por un amplificador que las llevará a un nivel adecuado para su manejo, por último, se utiliza un filtro pasa bajos, que eliminará las frecuencias no deseadas. Los computadores modernos tienen implementado internamente la mayoría de estos componentes, por lo que después de tener la señal preprocesada, se las aplica técnicas de muestreo y cuantificación para obtener una señal digitalizada, la cual pasará a un proceso en donde se realizará una segmentación de las palabras; se obtendrán sus características y se las comparará con otras señales precaracterizadas, por último, el computador logrará la identificación de las palabras. Para lograr estos resultados por lo general se utilizan dos modelos preentrenados a través de *machine learning*, el primero es conocido como modelo acústico, el cual utiliza como entrada, las formas de onda del audio sin procesar que se obtiene del habla humana, y genera una predicción probabilística de qué fonema se pronuncia cada determinado tiempo. Para que un modelo acústico sea eficiente y funcione de la mejor manera, es necesario entrenarlo con una amplia cantidad de audios y transcripciones que permitan lograr la detección de diferentes voces y dialectos de un idioma, además de reducir la probabilidad de error en entornos ruidosos. El segundo modelo es conocido como modelo del lenguaje, y es el que permite que la detección de varias palabras tenga el sentido y la coherencia de una oración bien estructurada, según un contexto específico, es decir, se encarga de predecir y discernir la palabra que más se adapte al contexto propuesto por la serie de palabras detectadas anteriormente (Chávez, 2009; Esparza & Avalos, 2003; Rev, 2021).

En el reconocimiento de voz, al igual que en la visión artificial, se busca dotar a la máquina de cualidades relacionadas a la inteligencia humana, en este caso se busca replicar la acción de escuchar e interpretar un mensaje, por lo que a continuación se presentan ciertos problemas que se relacionan con las capacidades limitadas que los computadores tienen en la actualidad (Chávez, 2009).

- Los computadores no tienen la misma capacidad sensorial de los seres humanos por lo que no es sencillo lograr buenas interpretaciones de las ondas sonoras.

- En una conversación, el oído humano no se limita a escuchar solamente palabras, sino que también percibe componentes externos que no necesariamente tienen que ver con el mensaje que se desea transmitir, debido a esto es que se tiene que clasificar los componentes de las señales que serán útiles de los que no.
- Cuando una persona escucha algo lo que procesa son sonidos ininterrumpidos y no existen separación entre ellos. Para que la máquina comprenda el mensaje proveniente de la voz humana es necesario que esta tenga entendimiento de componentes del lenguaje como pausas, letras, silabas y palabras para lograr una correcta interpretación.
- Cada persona tiene una voz única y hablará con diferentes tonalidades por lo que se necesita extraer las características de la señal, compararla con señales caracterizadas con anterioridad y encontrar similitudes.

A pesar de la presencia de todas estas dificultades, gracias a la investigación y avances tecnológicos se ha logrado desarrollar sistemas con niveles de exactitud elevado como por ejemplo en los que están enfocados en la detección de dígitos, sin presencia de ruido y trabajando solamente con la voz de una persona, se han obtenido resultados del 99% de exactitud. Otros sistemas más versátiles, con vocabularios extensos y para diferentes usuarios y diferentes canales, han logrado una exactitud de 87% lo que viene a ser bastante aceptable, teniendo en cuenta la complejidad de todas las características nombradas (Martinez et al., 2003).

Estudio del estado del arte

El uso de la computadora cada día se ha ido incrementando debido a sus amplias funcionalidades, las cuales en su mayoría han permitido facilitar distintos aspectos de la vida cotidiana del ser humano. Con el desarrollo del internet, los aspectos relacionados a educación cambiaron drásticamente debido a que el principal medio de búsqueda de información cambió de ser las bibliotecas, a las computadoras

conectadas a internet, en donde se encuentra todo tipo de información en diferentes formatos, como pueden ser libros digitales, blogs, videos tutoriales, entre otros. Las comunicaciones también cambiaron, ya que se empezó a utilizar los correos electrónicos como la principal forma de comunicarse a la distancia, además se empezó a usar chats y redes sociales para interactuar con otras personas. Estas y otras muchas funcionalidades del computador, lo han convertido en una herramienta fundamental en la vida cotidiana del ser humano por lo que, en la actualidad, el saber manejar esta máquina es tan importante como saber leer y escribir (Arango, 2010).

En este contexto, las personas con discapacidades motrices han sido un grupo al que, debido a sus circunstancias se lo ha excluido del uso de ciertos avances tecnológicos, debido a que en su mayoría se basan en la interacción de las extremidades del cuerpo con dispositivos que captan de alguna manera los movimientos. El *mouse* es un claro ejemplo, ya que permite a los usuarios interactuar con el computador al existir una relación entre el movimiento de la mano y el cursor visible en la pantalla, esta y otras herramientas como el teclado o controles, que, si bien a la gente promedio le facilita la interacción con las máquinas, para las personas con limitaciones motrices representa un gran problema ya que forman parte de un grupo que debido a su condición, tienen dificultades para adaptarse al uso de estas tecnologías. Por esta razón se ha buscado dar solución a este problema mediante el desarrollo de una variedad de proyectos e investigaciones que tienen como finalidad brindar nuevas herramientas que permitan el manejo de estos dispositivos, sin la necesidad de usar los miembros superiores (Azevedo & Nunes da Ponte, 2013). Un ejemplo de la implementación de estas nuevas tecnologías se lo puede apreciar en el trabajo realizado por Espada et al. (2020), donde implementaron sistemas de barrido ocular para una niña con parálisis cerebral grave en España, esto con la finalidad de que su proceso de aprendizaje pueda ser más efectivo e inclusivo mejorando sus habilidades sociales y comunicativas. En este estudio de caso se obtuvieron resultados positivos debido a que, con la implementación de este sistema, la niña pudo mejorar su desempeño académico y social, dándole de esta manera un mejor estilo de vida.

Similares a este ejemplo, existen varios trabajos que han sido realizados con la finalidad de facilitar la interacción hombre-máquina enfocada para personas con discapacidades motrices y para esto, algunas de las técnicas que se han utilizado son el *eye tracking* y el *face tracking*.

Técnicas para lograr la emulación del mouse

Con la finalidad de dar solución a los problemas que implica el uso del computador por parte de personas con discapacidades motrices, se ha implementado diversas tecnologías que permiten simular las funcionalidades del cursor a través de movimientos corporales, como por ejemplo de ojos o de la cabeza.

Eye tracking. En el trabajo realizado por Lupu et al.(2013) implementan un sistema para controlar el *mouse* de un computador mediante la técnica del *eye tracking* a través de oculografía infrarroja, para esto proponen la utilización de dos dispositivos, unas gafas de video y una *webcam*. Los dos componentes son acoplados entre sí, a través de una barra de aluminio que permite que la cámara detecte la mirada a una distancia inferior a diez centímetros, las gafas poseen seis *LED's* infrarrojos que proveen de iluminación constante al ojo, lo que permite que luz externa no influya en la detección de la pupila. Las gafas proyectan en sus *displays* una copia de la pantalla del computador y el software se encarga de detectar la dirección de la pupila y posicionar el cursor en el lugar a donde está enfocada la mirada. Para simular la acción de un *click* izquierdo, el usuario tiene que permanecer observando un mismo punto por un tiempo de entre uno y dos segundos. El uso de las gafas que proyectan la pantalla es justificado debido a que cuando se usa el monitor del computador, la posición de la cabeza afecta al algoritmo de *eye tracking*, por lo que con el uso de las gafas se descarta este problema. Para el desarrollo del software de este trabajo se utilizó como herramienta principal Visual Studio 2010 y la librería OpenCV, que facilita el procesamiento de imágenes. El programa consta de una capa de entrada desarrollada en C++, la cual consta de tres módulos, los cuales se encargan de la adquisición y procesamiento de la imagen, encontrar

la región de interés y la detección de la pupila. Por otro lado, hay una capa de salida desarrollada en C# que tiene la funcionalidad de procesar lo que se recibe de la capa de entrada, de igual manera, consiste en tres módulos, los cuales se encargan de la calibración, encontrar las coordenadas del cursor basándose en la dirección de la pupila, y de generar la interfaz de usuario que permite hacer *click* y moverse a través de la pantalla. Para obtener resultados de la usabilidad, precisión y fiabilidad, se acudió a veinte personas para que hagan uso del dispositivo y posteriormente llenen una encuesta que evaluaba aspectos de ergonomía, tiempo de adaptación para usar el sistema, facilidad de manejo, entre otros, obteniendo una calificación de 603 sobre 800 puntos posibles. Los autores justifican el rendimiento debido a que algunos usuarios eran mujeres las cuales estaban maquilladas con delineadores oscuros lo que influenciaba en la detección de la pupila y afectaba al algoritmo.

Si bien los métodos de *eye tracking* basados en videooculografía (VOG) son más precisos que los que implementan electrooculografía (EOG), los primeros vienen a ser más costosos, intrusivos y requieren de mayores recursos computacionales (Majaranta & Bulling, 2014). Por esta razón, ingenieros de Brasil y Alemania propusieron el diseño y la evaluación de un sistema que emule las funcionalidades del mouse, utilizando electrooculografía. Para lograr este cometido, utilizaron dos electrodos encargados de captar la señal de los movimientos de un solo ojo, y otro que se conecta al cuello del usuario para eliminar interferencias. La señal que se obtiene de los electrodos es amplificada y filtrada por un circuito conformado por varios amplificadores operacionales, para luego ser procesada por una tarjeta embebida especial denominada TIVA, que tiene las cualidades computacionales necesarias para correr el algoritmo que controla el cursor, y además permite la comunicación USB con el computador. Utilizando una librería propia del fabricante del microcontrolador, este puede ser configurado para que funcione como una interfaz hombre-máquina, permitiendo que funcione como un mouse común sin la necesidad de instalar otras aplicaciones en la computadora. Para poder emular la acción del *click*, el usuario debe realizar un parpadeo voluntario, ya que la señal EOG en estos casos llega a un pico más alto y su duración es más

corta, por lo que a partir de estos parámetros se logra discernir entre un parpadeo natural y un voluntario. En los resultados de este trabajo se evidencia que hay limitaciones a la hora de usar el dispositivo, siendo algunas de estas su baja resolución, y la necesidad de mantener la cabeza quieta, debido a que genera señales no deseadas. Para dar solución a esto, implementaron otro método que se basó en la idea de un teclado, para lo que a diferentes movimientos de los ojos se les asignó un comando predefinido, que es asociado al movimiento del ratón. De esta manera se puede solventar los problemas de baja resolución y las señales no deseadas que interfieren por los diferentes movimientos corporales (de Lucena & Conzelmann, 2019).

En el trabajo realizado por Freire et al. (2013) se presentan dos estudios llevados a cabo por la Universidad Miguel Hernández de Elche (UMH) en España y en la Universidad Federal do Espírito Santo (UFES) en Brasil, donde se desarrollaron dos interfaces que permiten aplicar la videooculografía con fines de ayudar a la gente discapacitada; en la primera se utilizó una cámara con un soporte fijo que enfoca al rostro de la persona y permite detectar la dirección de la mirada, sin embargo presenta un inconveniente en el cual para que la detección de la mirada sea precisa, el movimiento de la cabeza debe ser muy limitado. Para la implementación de este sistema se usó una cámara, una tarjeta de adquisición de datos, el computador y todo el equipo necesario para una buena iluminación. Debido a que la cámara se encuentra sobre un soporte fijo, no resultó molesto para los usuarios y en varias pruebas experimentales se obtuvieron resultados positivos en cuanto a la precisión de la detección de la mirada. Por otra parte, el prototipo desarrollado en Brasil, consistió en un sistema de *eye-tracking* para comandar una silla de ruedas robotizada, y también para interactuar con una interfaz de comunicación en donde se pueden redactar textos o seleccionar íconos representativos de necesidades o sentimientos. Para la implementación de este sistema se usó unas gafas a las cuales se le acopló una pequeña cámara, que permitió detectar la dirección de la mirada con altos niveles de resolución y precisión, logrando cumplir con toda la funcionalidad a la que estaba destinada. En estos dos aplicativos de la videooculografía se

puede notar las diferencias entre usar hardware intrusivo y no intrusivo, en donde se puede ver que al usar dispositivos como gafas con cámaras se gana precisión, pero puede resultar ser más costoso e incómodo de usar. En el caso de utilizar solamente las cámaras de los computadores, si bien pueden ser precisos a la hora de detectar la dirección de la mirada, tienen la desventaja de que se limita el movimiento natural de la cabeza.

Estudiantes de la Universidad Peruana de Ciencias Aplicadas (UPC) desarrollaron un sistema que permite controlar un computador utilizando *eye-tracking*, basándose en la electrooculografía. El funcionamiento del dispositivo parte de la adquisición de la señal EOG, utilizando dos electrodos colocados en la parte superior e inferior del ojo y se coloca un electrodo de referencia adicional en la frente del usuario. La señal adquirida pasa por varios filtros y amplificadores, para posteriormente ser digitalizada por un microcontrolador que se encargará de reconocer la forma de onda de la señal, y a partir de esto, enviará a la computadora a través de *bluetooth* un carácter diferente dependiendo del movimiento que se realizó, como puede ser mirar arriba o abajo más parpadeo, parpadeo prolongado o doble parpadeo. Por último, un software desarrollado en *Python*, se encargará de ejecutar las diferentes funciones dependiendo del parámetro que reciban. Para facilitar el manejo total de la computadora implementaron varios menús que se activan al parpadear dos veces, en estos se puede seleccionar las diferentes funcionalidades del *mouse* ya sea *click*, doble *click* o *click* derecho o también se puede seleccionar una opción que desplegará un teclado virtual que permite predecir palabras para facilitar la escritura del usuario. Para aumentar la resolución y la precisión de la herramienta, cuando se selecciona la opción del *mouse*, automáticamente se divide la pantalla en cuatro segmentos que pueden ser seleccionados dependiendo del cuadrante en donde se quiera mover el cursor, para seleccionar una zona se utiliza un indicador que se desplaza cada cierto tiempo por cada uno de los segmentos en donde el usuario tiene que parpadear dos veces cuando el indicador esté en la posición deseada, a partir de esto el cuadrante que se escogió se presentará en toda la pantalla simulando la opción de zoom y facilitando

el posicionamiento del cursor. Entre las limitaciones que presenta el dispositivo propuesto está la susceptibilidad a generar interferencias cuando hay movimientos bruscos de la cabeza o el cuerpo, la necesidad de utilizar un computador con conexión bluetooth y por último el uso de baterías para la alimentación del controlador. Los desarrolladores realizaron pruebas de desempeño con 16 usuarios para validar la capacidad del sistema de reconocer los diferentes gestos obteniendo una eficiencia mayor a 88.44%. También se realizó una prueba para verificar el funcionamiento del *mouse*, la cual consiste en ejecutar diferentes acciones, como por ejemplo colocar el cursor en un ícono específico, abrir la aplicación, ubicar el cursor en una zona específica y hacer click. Los resultados de esta evaluación se midieron a partir del número de intentos requeridos por cada usuario para realizar dichas acciones, obteniendo resultados buenos y coherentes, teniendo en cuenta la falta de experiencia de las personas (Díaz & Morales, 2019).

Face tracking. Otra alternativa que se ha utilizado para simular las funcionalidades del *mouse* es el *face tracking*, que tiene como elemento principal una cámara que permite detectar los diferentes movimientos de la cabeza o gestos faciales. En el trabajo realizado por Sinha et al. (2015), proponen combinar la detección facial en conjunto a reconocimiento de voz, por lo que el sistema consta de dos módulos. El primero es el encargado de detectar la cara de la persona, con la finalidad de poder cambiar la posición del cursor de manera proporcional a la que se mueve la cabeza. Para lograr este cometido se utiliza el algoritmo de reconocimiento facial, que proporciona la librería OpenCV, una vez que el rostro se ha detectado, en relación a este se calcula la posición de los ojos. A través de la línea de la cara y el contorno de los ojos, se detectan otras características de la cara, como puede ser la posición de las orejas, la quijada, el contorno de los labios, entre otros. El software se encarga de continuamente comparar las características anteriores con las nuevas, y si detecta algún movimiento lo traduce a un movimiento del cursor. Por otro lado, el segundo módulo consiste en el reconocimiento de voz, para lo que utilizan como herramienta principal el kit de desarrollo *Microsoft Speech SDK 5.1*, que entre sus funciones permite

reconocer la voz del usuario y transformarla a texto. A través de diferentes comandos de voz, se puede emular las acciones del *mouse*, por ejemplo, un *click*, doble *click*, *click* derecho y adicionalmente, para facilitar el manejo del computador también se implementan comandos para abrir aplicaciones específicas como *Word*, *Paint*, *Excel*, entre otras.

Loewenich & Maire (2007), presentan el prototipo de un sistema similar al anteriormente descrito, en donde de igual manera, se busca emular el funcionamiento del mouse usando *face tracking* y reconocimiento de voz. La metodología implementada para mover el cursor se basa en la detección del rostro, y la ubicación de distintos puntos característicos que cuando se registra movimiento, son comparados con una posición inicial y de acuerdo a su diferencia el cursor se mueve. Este trabajo presenta la novedad de que tiene dos modos de funcionamiento, el primero es llamado “modo relativo” y consiste en que, si el sistema detecta una desviación de los puntos característicos respecto a su posición inicial, el cursor se moverá en la misma dirección continua, pero dependiendo del tiempo que los puntos no estén en su posición normal, la velocidad de movimiento aumentará, reseteándose cuando no se registre movimiento. Por otro lado, el “modo absoluto” permite el movimiento del cursor utilizando como posición inicial el centro de la pantalla y la distancia que se moverá depende del equivalente a la distancia de los puntos característicos hacia su posición original. En cuanto al sistema de reconocimiento de voz, este es implementado utilizando el mismo kit de desarrollo *Microsoft Speech SDK 5.1*, sin embargo, presenta otras funcionalidades aparte de las que se nombraron anteriormente las cuales son “*click drag*” y “*click drop*”, la primera cumple la función de mantener presionado el click izquierdo y la segunda lo deshace, esta función es necesaria para situaciones en donde se quiere por ejemplo arrastrar un archivo a una carpeta. Para validar el funcionamiento del emulador, se puso a prueba a diez voluntarios con experiencia previa en uso de computadores, por lo que la adaptación resulto más sencilla. La prueba consistió en jugar un videojuego, y posterior a esto los voluntarios dieron una retroalimentación en donde se nombró la fácil

calibración y configuración del programa, además de la ventaja de no requerir de ningún implemento para usar en la cabeza o en los ojos.

También se han desarrollado sistemas que están enfocados en la interacción con aplicaciones específicas, como es el caso del trabajo desarrollado por Ferrin et al. (2021), donde se implementa un interfaz hombre-máquina, que funciona para facilitar el manejo de la aplicación Whatsapp Web. Para lograr esto se aprovecha los dos segmentos que existen en dicha aplicación, los cuales son una zona de chat y otra de contactos, por lo que, tanto la cara del usuario como la pantalla son procesadas como imágenes. Los comandos son enfocados solo a la utilización de la aplicación, por lo que movimientos laterales de la cabeza permitirán cambiar de zona, es decir entre contactos y chat, los movimientos verticales permitirán hacer scroll arriba/abajo en cualquiera de las zonas, diferenciándose en que para contactos se seleccionará uno a la vez. Para lograr la ejecución de funciones como la activación, *scroll*, click derecho o click izquierdo, se estima la posición del rostro, utilizan un modelo que se encuentra en la biblioteca DLIB, el cual brinda 68 puntos característicos del rostro, con los cuales se puede definir relaciones de aspecto que facilitan la detección de eventos como la apertura de la boca o el cierre de un ojo, de esta manera se puede asociar una función con un evento.

Mosquera et al. (2017) desarrolla e implementa una interfaz dotada de varios menús y opciones que permiten interactuar de una manera más fluida con la página de *Facebook*, logrando asociar el movimiento del cursor con el movimiento de la cara, y los *clicks* con parpadeos. Para lograr esto, el sistema implementado consiste en cinco procesos, el primero es el encargado de lograr la captura de la imagen, para lo que se usa una cámara web con resolución estándar, después la imagen pasa por una etapa de preprocesamiento, donde se la transforma a escala de grises, y se hace un efecto espejo horizontalmente para lograr el correcto movimiento del cursor. Después de esta etapa, se realiza una segmentación de la imagen, en donde se localiza la posición de la cara y los ojos para poder interpretar dos gestos, el movimiento de la cara que se traduce en el movimiento del cursor y los parpadeos que permiten hacer

click. Cabe recalcar que el sistema descarta los movimientos leves de la cabeza y el parpadeo natural. La funcionalidad del *click* puede también ser realizada manteniendo el cursor en un solo punto durante un tiempo determinado. En cuanto a los menús de control, con la finalidad de facilitar el acceso a las diferentes características de la aplicación, al mover el cursor ya sea a la parte superior o inferior de la pantalla, se activan dos menús de desplazamiento que permiten emular la función de *scroll*, por otro lado, si el usuario mueve el cursor hacia los límites izquierdo y derecho de la pantalla, se activan dos menús donde se puede encontrar opciones para realizar acciones de *zoom*, ir al *Home* de Facebook, mirar notificaciones, ir al perfil de usuario, ver contactos, escribir mensajes y salir de la aplicación. Para realizar las pruebas de desempeño, se trabajó con dos grupos de diez personas cada uno, el primero consistió en personas jóvenes con experiencia en el uso de Facebook y el segundo estuvo conformado por personas adultas sin experiencia previa en la red social. En las pruebas realizadas se midió la facilidad para desplazar del cursor, el uso de los menús y de manera general, la interacción con la aplicación. Se obtuvieron resultados de eficiencia mayores a 80% en la mayoría de aspectos donde se resaltó la conformidad de los usuarios con las dos técnicas para hacer *click*, el movimiento del cursor y la funcionalidad de los menús desplegables.

Como una continuación del trabajo anteriormente descrito, se presenta el diseño e implementación de una interfaz multimodal, que permite navegar en internet de una forma más general. La novedad de este trabajo, es la implementación de un módulo de reconocimiento de voz, que permite la detección de 39 comandos en lengua española con los cuales se puede navegar en *Google Chrome*, *Gmail* y *Facebook*, estos se dividen en los de uso general y los específicos para cada una de las tres aplicaciones. Entre las funcionalidades más destacadas está la de permitir dictados con el comando "Activar dictado", con lo que se puede escribir textos en correos, documentos, entre otros. Para el control de este sistema se utiliza una unidad multimodal que sincroniza las funcionalidades del módulo de reconocimiento de gestos con las de reconocimiento de voz, facilitando de esta manera el manejo y la

interacción con las aplicaciones especificadas. Para determinar la funcionalidad del dispositivo se puso a prueba a tres grupos de personales, el primero consistió en gente joven, el segundo en gente adulta y el tercero, gente con discapacidades motrices. Como resultados generales se obtuvo un 75.7% en el desempeño del reconocedor de comandos y un 81.1% para dictados, en el caso del reconocimiento de gestos y aplicación en los menús se obtuvo un 100% de aceptación, sin embargo, el desempeño a la hora de hacer *click* fue de 70.2% pero subió a 84.2% al utilizar la opción de zoom. En cuanto a la percepción del dispositivo, se determinó que los usuarios adultos se sintieron atraídos por la interfaz ya que les facilita el uso del computador y les permite realizar otras actividades con sus manos, en el caso de los jóvenes se determinó que prefieren seguir utilizando las interfaces tradicionales, y por último, las personas con discapacidades motrices evidenciaron su aceptación debido a que les da autonomía a la hora de tener que utilizar esas aplicaciones sin requerir de usar un *mouse* o un teclado (Mosquera et al., 2020).

Otras técnicas. En la literatura referente a este tema se ha encontrado trabajos en los que se enfocan en el uso de algoritmos de visión artificial que permiten detecta movimientos de la cabeza, sin la necesidad de reconocer la cara o el ojo como es el caso de *Tracker Pro*, el cual es un sistema básico que permite a personas con dificultades motrices manejar el *mouse* del computador únicamente con el movimiento de su cabeza, y utilizando un pequeño reflector adhesivo en la frente o en las gafas de la persona con la finalidad de que la cámara siga ese punto específico para convertirlo en el movimiento del cursor. La desventaja de este dispositivo es que, para realizar la función de *click* y doble *click* se requiere de softwares externos o de otros dispositivos que puedan ser operados por el usuario (Azevedo & Nunes da Ponte, 2013).

En el trabajo realizado por Gür et al. (2020) se presenta una manera de emular el funcionamiento del mouse enfocada para personas con limitaciones motrices en los miembros superiores en donde utilizando sensores de movimiento y de presión se realiza acciones como el movimiento del cursor y los *clicks*. Adicional a esto, implementa un *pad* de diez teclas ultrasensibles que permite teclear a personas

con poca movilidad en sus manos. El sistema enfocado en replicar al *mouse* consiste en unas gafas, a las cuales se les ha montado una unidad de medida inercial (IMU), que consiste en un giroscopio de tres ejes, un acelerómetro de tres ejes y un procesador de 16 bits que se encarga de fusionar los datos de los otros componentes, procesarlos y detectar el movimiento de la cabeza, ya sea de derecha a izquierda, o arriba y abajo. El segundo componente es un sensor de presión que se coloca en los marcos laterales de las gafas, son fabricados por un polímero semiconductor que reduce su resistencia en respuesta al incremento de la presión. Este sensor cumple la función de detectar cuando los músculos de la mandíbula se contraen, por lo que para realizar un *click*, el usuario debe realizar una leve mordida, y si desea hacer un *click* derecho deberá realizar una mordida más prolongada. El mismo sensor es el que se utiliza para realizar el teclado que funciona de manera similar a lo que poseen los celulares no táctiles, es decir cada botón activa un número y tres o cuatro letras del abecedario. A partir de pruebas técnicas de desempeño, se llegó a la conclusión de que el sistema en general viene a ser una buena opción de bajo costo, sin embargo, presenta algunos problemas con la estructura de las gafas, las cuales se fabricaron en una impresora 3D, por lo que su ajuste es complicado y no está enfocado a todos los usuarios.

El cursor del computador también puede ser emulado a partir del uso de la lengua, lo que resulta muy útil para personas con discapacidades motrices, Lund et al. (2009), muestran el prototipo de un sistema para controlar el mouse y el teclado a través de un dispositivo intra oral encapsulado, que a través de sensores capta el movimiento de un piercing ferromagnético. Las señales de los sensores son transmitidas a un controlador embebido por radio a una frecuencia de 2.4GHz, que es una banda orientada a usos científicos y médicos. El controlador se encarga de procesar la información y detectar cuál de los dispositivos externos es el que se está utilizando. El dispositivo intra oral consiste en 18 sensores, de los cuales, ocho están destinados al movimiento del cursor y diez para funciones de teclado y activar funciones del *mouse* como *click*, *doble click*, *scroll up* y *scroll down*. El teclado funciona como los teléfonos antiguos, es decir, para cada tecla se pueden escribir tres o cuatro letras, y la última sirve para

generar un espacio. Como otro elemento de este sistema se tiene un dispositivo USB, en donde se encuentra el microcontrolador y los módulos de radio frecuencia, también es el responsable del envío de los comandos que realizarán el mouse y el teclado. A partir de una prueba se pudo verificar que el sistema funciona para controlar tanto el teclado como el *mouse* de un computador con sistemas operativos Windows XP, Ubuntu y OS X, utilizando solo los drivers propios de la máquina. También se realizaron pruebas manuales, es decir con el dispositivo fuera de la boca, simulando el movimiento del piercing y se logró escribir pequeñas oraciones y borrar caracteres, además de mover el cursor de manera correcta.

Otra manera de emular el funcionamiento de un mouse, es la que se presenta en el trabajo realizado por McFarland et al. (2008), donde a través de patrones de ondas cerebrales captadas a partir de un electroencefalograma, se logra realizar diferentes acciones con el cursor como puede ser mover para arriba, abajo, derecha e izquierda, y además opciones para seleccionar y soltar un elemento. A partir de la necesidad de mover un objeto y tener que seleccionar algo se generan tres tipos de señales, de las cuales dos se relacionan con los movimientos horizontales y verticales y la otra es para seleccionar o soltar un objetivo. A partir del análisis de estas señales se logra el objetivo de emular las funciones del *mouse*.

Tomando en cuenta los trabajos expuestos anteriormente, para la ejecución de este proyecto se ha descartado el uso de técnicas que requieran de dispositivos externos que sean de difícil adquisición o fabricación. En caso del eye tracking, si bien el captar la posición de la mirada puede dar buenos resultados a la hora de emular el funcionamiento del mouse, para lograr que uno de estos sistemas sea preciso y confiable, se necesita de dispositivos externos como gafas o electrodos que pueden llegar a ser costosos e intrusivos, también se podría lograr eye tracking únicamente con las webcams de los computadores, pero estos algoritmos podrían ser más propensos a generar fallas a bajos niveles de luz o movimientos bruscos de la cabeza.

Con la intención de lograr un sistema que sea funcional y accesible para cualquier persona que tenga un computador con sus componentes básicos como es el micrófono y su cámara, se ha escogido como base fundamental los conceptos aplicados en el último artículo descrito en la sección de *face tracking* (Mosquera et al., 2020), en el cual combinan las funciones del reconocimiento facial con el reconocimiento de voz para navegar por aplicaciones específicas, además, se busca reinterpretar y combinar los conceptos de segmentación presentados por Díaz & Morales (2019), ya que se considera que al utilizar todas estas técnicas juntas se puede llegar a desarrollar un sistema que cumpla con las características de ser funcional, preciso, de uso general en el computador y de bajo costo, además de brindar facilidad y libertad de movimiento a los usuarios, sin la necesidad de dispositivos intrusivos para su funcionamiento. A continuación, en el siguiente capítulo se presentarán las metodologías que se utilizaron para desarrollar el software del emulador.

Capítulo III. Desarrollo del software

En este capítulo se nombrará y justificará el uso de las herramientas de software que se usaron para desarrollar el emulador, es decir, el lenguaje de programación en conjunto con las librerías aplicadas. Además, se explicará de forma detallada como fue el desarrollo y las características del programa.

Lenguaje de programación

Es necesario conocer los motivos que influyeron a la hora de decidir que lenguaje de programación utilizar para desarrollar el sistema, por esta razón a continuación se describirá las características principales de *Python*.

Python

Python es un lenguaje de programación de muy alto nivel que se ha popularizado en estos últimos años debido a la simplicidad que brinda a la hora de programar, su sintaxis es muy simple y códigos básicos pueden ser entendibles únicamente teniendo conocimientos del idioma inglés. Es un lenguaje de propósito general, y su uso no se limita a un solo paradigma, es decir que se puede trabajar tanto con programación imperativa (usando bucles), funcional (usando funciones y módulos) o programación orientada a objetos. Otra ventaja que ofrece este lenguaje es que permite generar programas para diferentes sistemas operativos como pueden ser Linux, Windows o Mac, adoptando así la característica de ser multiplataforma. Existe una gran comunidad que se encarga de dar soporte y a la vez se han generado una gran cantidad de librerías y funciones que se pueden usar para facilitar el desarrollo de cualquier programa. Para el desarrollo del sistema propuesto en este trabajo, se tiene como uno de sus pilares principales el reconocimiento facial, por lo que es importante destacar que Python es compatible con OpenCV, que es una librería para visión artificial muy robusta y eficiente. Por el lado del reconocimiento de voz, de igual manera existen varias librerías que permiten y facilitan tareas como pasar de voz a texto, con o sin conexión a internet. Para el segundo caso se necesitan modelos de idiomas y

dialecto en el que se quiera trabajar, los cuales ya han sido desarrollados previamente por la comunidad (del Valle, 2016).

Librerías aplicadas

A continuación, se presentarán las librerías que se implementaron para el desarrollo del software, nombrando brevemente sus características y justificando su utilización.

Pyautogui

Es un módulo de Python que permite de una forma sencilla programar las funcionalidades del *mouse* y del teclado para cualquier aplicación en el computador, tiene soporte para Windows, macOS y Linux. Para la implementación del emulador se consideró como un módulo de gran utilidad debido a que, a través de este se ejecutan acciones propias del mouse como el click, doble click, mantener presionado y soltar el click para realizar la acción de *scroll*. Además, permite realizar el movimiento del cursor a través de la pantalla de diferentes formas, como puede ser realizando un movimiento relativo, o dando como instrucción una coordenada específica de la pantalla, lo que es de gran utilidad para lograr la segmentación. Adicionalmente, permite manipular el teclado, lo que facilita el ingreso de texto para el modo dictado, y la ejecución de comandos que pueden llegar a ser de utilidad como por ejemplo la combinación de teclas “alt+F4” que permite cerrar una aplicación (PyAutoGUI, 2021).

Time

Es un módulo que permite utilizar varias funciones que hacen relación al tiempo. Entre sus utilidades está la espera de tiempos definidos durante la ejecución del código, lo que se conoce como *delays*, permite también representar y manejar el tiempo en sus diferentes unidades a través de varias funciones que facilitan su cometido. La principal razón por la que se usa este módulo en este trabajo es

debido a que en distintas funciones se requiere generar retardos, para que el sistema funcione correctamente (Ronquillo, 2019).

Threading

El módulo *Threading* permite que una sola aplicación ejecute varias tareas a la vez utilizando el concepto de hilo, que viene a ser cada uno de los flujos que son programados para que se ejecuten simultáneamente. En Python un hilo se define como un objeto *Thread*, el cual estará asociado a elementos que se puedan invocar, como es el caso de una función, la cual se ejecutará de forma paralela a otras funciones de ser el caso. Al momento de crear el objeto *Thread* existe la posibilidad de definirlos como “demonios”, este atributo permite dar jerarquía a los hilos que existan, y es de utilidad cuando se requiere dar por finalizado un programa, ya que este finalizará su ejecución cuando solo haya hilos “demonios” activados. En la implementación del emulador del *mouse*, este módulo cumple un papel fundamental, ya que a través de este se pueden ejecutar de forma paralela las funciones que permiten realizar el reconocimiento de voz, la ejecución de comandos, el reconocimiento de rasgos faciales y una función que reproduce un audio en formato MP3, para avisar al usuario que el mouse esta activado (Anderson, 2019).

Queue

Este módulo permite implementar estructuras de datos lineales, las cuales almacenan y distribuyen elementos de forma ordenada. Se puede implementar tres clases de colas: la primera puede ser de tipo FIFO (first-in-first-out) donde los datos que ingresan primero son los primeros en ser retirados, la segunda es de tipo LIFO (last-in-first-out) en donde el último dato en ser ingresado es el primero en ser retirado, y por último las colas de prioridad en donde los datos que ingresan se mantienen ordenados, dependiendo de su prioridad y los primeros en salir son los que menor valor tengan. Estas estructuras, específicamente las de tipo FIFO, fueron utilizadas en la implementación del emulador, ya que son de gran utilidad a la hora de compartir datos entre diferentes hilos (Aggarwal, 2020).

Pyaudio

Este módulo permite de una manera simple realizar acciones como reproducir o grabar audio a través de Python accediendo al micrófono del computador, tiene soporte para distintas plataformas como Windows, Linux, MacOS, GNU, entre otros. En la implementación del emulador del mouse se usa para captar la voz del usuario (Pypi, 2017).

Vosk

Es un módulo que contiene un conjunto de herramientas que permite realizar reconocimiento de voz, sin necesidad de tener una conexión a internet. Permite trabajar en más de veinte idiomas y dialectos incluido el español. Para su funcionamiento requiere del modelo de las señales acústicas y del modelo del lenguaje correspondiente al idioma con el que se quiere trabajar, el cual puede ser desarrollado desde cero o se puede adquirir los que se encuentran en el repositorio propio de la librería Vosk. Una de las ventajas principales es que existe una gran variedad de modelos que soportan diferentes idiomas, y entre estos existen algunos que están enfocados para ser utilizados en plataformas pequeñas, como dispositivos móviles o tarjetas Raspberry debido a que pesan menos de 50Mb, y requieren aproximadamente 300Mb de memoria a la hora de ejecución. Se consideró que esta es la herramienta adecuada para desarrollar el sistema de reconocimiento de voz del emulador del *mouse*, debido a que en las pruebas de funcionamiento dio buenos resultados y se consideró que para lograr que el sistema sea de uso general, sería necesario que funcionara sin la necesidad de tener una conexión a internet (Alpha Cephei, 2020).

Json

Es un paquete que permite trabajar en Python con un formato de datos basado en la sintaxis de objetos en JavaScript. Es usado principalmente para intercambiar datos usando un formato estandarizado que es sencillo de entender y crear. Esta herramienta permite codificar y decodificar datos JSON. En este trabajo se utilizó esta herramienta para decodificar los resultados del reconocimiento de vos

proporcionado por la librería Vosk, los cuales se transmiten en formato JSON, y al decodificarlos Python entiende al resultado como un diccionario (Lofaro, 2018).

Pygame

Pygame es un módulo que permite un acceso simple a la librería SDL (Simple DirectMedia Layer), a través de Python. Está enfocado en la creación de videojuegos, para lo que permite de manera sencilla acceder a los diferentes componentes hardware del computador que permiten trabajar con video, sonido, mouse, teclado, entre otros. Una ventaja de usar este módulo es que es multiplataforma, lo que permite crear programas para una gran variedad de sistemas operativos. En la implementación del emulador se lo utilizó únicamente para reproducir audios en formato MP3, que señalen la activación del mouse y del modo dictado (Fincher, 2019).

OpenCV

Es una biblioteca de visión artificial de código abierto, la cual posee más de 2500 algoritmos de visión por computador y *machine learning*, que pueden ser utilizados en distintas plataformas sin ningún problema. Debido a su gran popularidad, el acceso a su documentación es amplio, y está siendo actualizada con frecuencia lo que significa que hay mucha facilidad para su aprendizaje. Entre algunas de sus funcionalidades esta la identificación de rostros, objetos, captura de video, procesamiento y captura de imágenes, entre otras. Para el desarrollo del emulador propuesto en este trabajo, esta librería es utilizada para realizar la captura del video, su preprocesamiento y la representación de las zonas de interés del rostro que fueron detectadas utilizando las funciones de la librería Dlib (Marín, 2020).

Dlib

Es una biblioteca multiplataforma de propósito general y código abierto creada en C++ que puede ser utilizada en Python a través de su propia API (Interfaz de programación de aplicaciones). Su utilidad

es amplia, si bien su fuerte son sus algoritmos de *machine learning*, también implementa gran cantidad de funciones que permiten trabajar con procesamiento de imagen, programación por hilos, redes, creación de interfaces gráficas, entre otras. Su utilidad en este proyecto se centra en los algoritmos de detección de rostros y extracción de características, los cuales permiten detectar el rostro y sus rasgos en tiempo real con una muy buena precisión. El detector de rostros que esta librería proporciona fue entrenado utilizando los conceptos expuestos en el paper desarrollado por Dalal & Triggs (2005), donde aplican un método conocido como histograma de gradientes, orientados en conjunto a una máquina de vectores para entrenar un clasificador de objetos con una muy buena precisión. Por otro lado, Dlib posee un modelo preentrenado, que permite estimar la posición de 68 coordenadas (ver Figura 1), que son asignadas a las diferentes zonas de interés de una cara en una imagen, dicho modelo se creó a partir de la implementación del paper desarrollado por Kazemi & Sullivan (2014), utilizando Dlib y entrenándolo usando una base de datos de zonas características de rostros. Esto se aplica con la finalidad de poder generar las acciones de activación y los movimientos del cursor a partir de la detección de la cara y de los rasgos de la boca y la nariz (King, 2014; Rosebrock, 2014, 2017).

Numpy

Es una biblioteca que permite trabajar con arreglos en Python de diferentes formas y además posee una gran cantidad de funciones que facilitan las operaciones entre matrices las cuales pueden ser matemáticas, lógicas, de formas, algebra lineal, estadísticas, transformadas de Fourier, entre otras. En este trabajo se utiliza para realizar diferentes operaciones con las matrices de datos que se proporcionan con cada *frame* del video capturado (Sánchez, 2020).

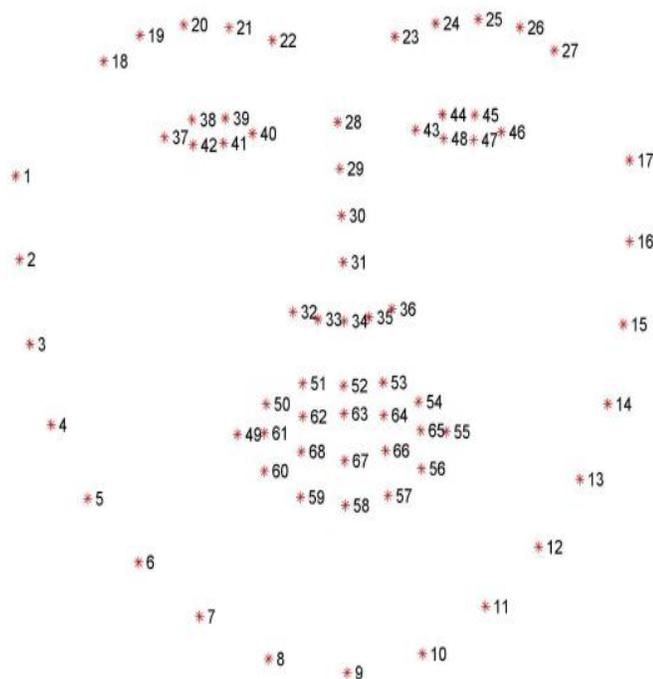
Imutils

Es un paquete de funciones convenientes para realizar funciones de procesamiento de imagen básico como translación, rotación, escalar, entre otras. También tiene funciones que permiten facilitar el

trabajo con los modelos de detección facial de la librería Dlib. Debido a su aplicación directa en la implementación de este proyecto ha sido usado tanto para el preprocesamiento de los *frames* como para manejar de manera más sencilla las herramientas de Dlib (Rosebrock, 2015).

Figura 1

Puntos de la cara que proporciona el predictor de Dlib



Nota. Tomado de *300 Faces in-the-Wild Challenge: The First Facial Landmark Localization Challenge*, por Sagonas et al., 2013, *IEEE International Conference on Computer Vision Workshops*.

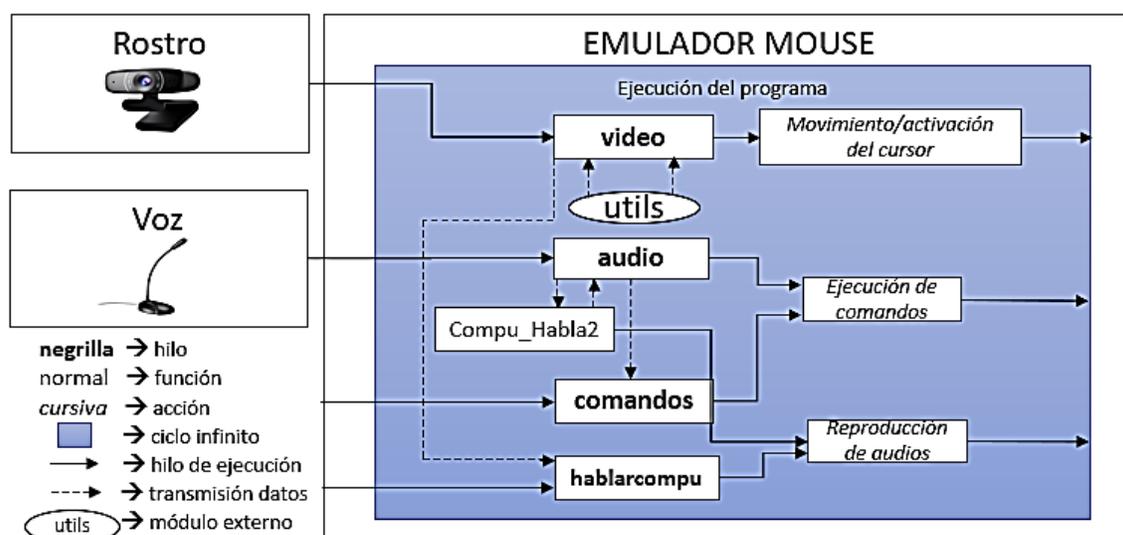
Desarrollo y características

El sistema desarrollado consiste en la implementación de diferentes funciones que se ejecutan de forma paralela a través del uso de hilos. Se programaron siete funciones para cumplir con los propósitos del sistema, cuatro de estas se ejecutan como subprocesos de forma simultánea, una es llamada de forma ocasional en programa principal y las otras dos conforman un módulo que se lo utiliza de manera frecuente en uno de los hilos. En la Figura 2 se presenta un esquema que describe de forma general como

es el funcionamiento del emulador en donde se muestran los hilos, módulos y funciones con las líneas de transmisión de datos entre los mismos y se representan en cuadros distintos las acciones que realizan teniendo en cuenta que estas se ejecutan dentro de cada subproceso y no aparte. A continuación, se procederá a explicar de forma detallada como ha sido la programación, el funcionamiento y utilidad de cada hilo y función representada en el gráfico.

Figura 2

Esquema general del funcionamiento del emulador



Hilo “video” y modulo “utils”

El hilo “video”, es el encargado de generar los movimientos y modificar el estado de activación del cursor. Para la implementación de esta función se utilizó como base el trabajo desarrollado por Chandra (2018) en el cual se desarrolló un algoritmo para mover el cursor del mouse a través de movimientos faciales. En este algoritmo inicialmente se tiene que definir todos los parámetros con los que se trabajará y además instanciar el detector de rostros y el predictor de características de la cara que provee la librería Dlib para lo cual se utiliza el siguiente algoritmo.

```

shape_predictor = "paths/model/shape_predictor_68_face_landmarks.dat"
detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor(shape_predictor)

```

Utilizando funciones de la librería *imutils* se puede acceder a los índices de los puntos iniciales y finales de cada parte de la cara que provee el predictor, sin embargo, para el caso de este proyecto no es necesario utilizar todos los puntos ya que las partes de la cara que se utilizan son la nariz (para dirigir los movimientos del cursor) y la boca (para activar el movimiento) por lo que se definen variables que contengan los índices para acceder a puntos correspondientes a cada una de estas partes de la cara, siendo del 48 al 68 los de la boca y del 27 al 35 los de la nariz.

```

(nStart, nEnd) = face_utils.FACIAL_LANDMARKS_IDXS["nose"]
(mStart, mEnd) = face_utils.FACIAL_LANDMARKS_IDXS["mouth"]

```

Después de esto a través del uso de la librería *OpenCV* se accede a la cámara del computador usando un bucle *while* permitiendo así trabajar con los *frames* que capturan los movimientos del rostro del usuario. Sin embargo, la imagen que se proyecta no es apta para lograr los resultados esperados por lo que se procede a realizar una etapa de preprocesamiento donde inicialmente se debe hacer un espejo de la pantalla ya que se busca que los movimientos horizontales que el computador capte tengan coherencia con lo esperado, es decir, si hay un movimiento a la derecha, la imagen proyectará un movimiento a la derecha y no al contrario como inicialmente se capta el video. Esto se logra realizando una rotación sobre el eje “y” utilizando la función “flip” de OpenCV. A continuación, se muestra el código utilizado para realizar esta acción y en la Figura 3 se puede observar el resultado.

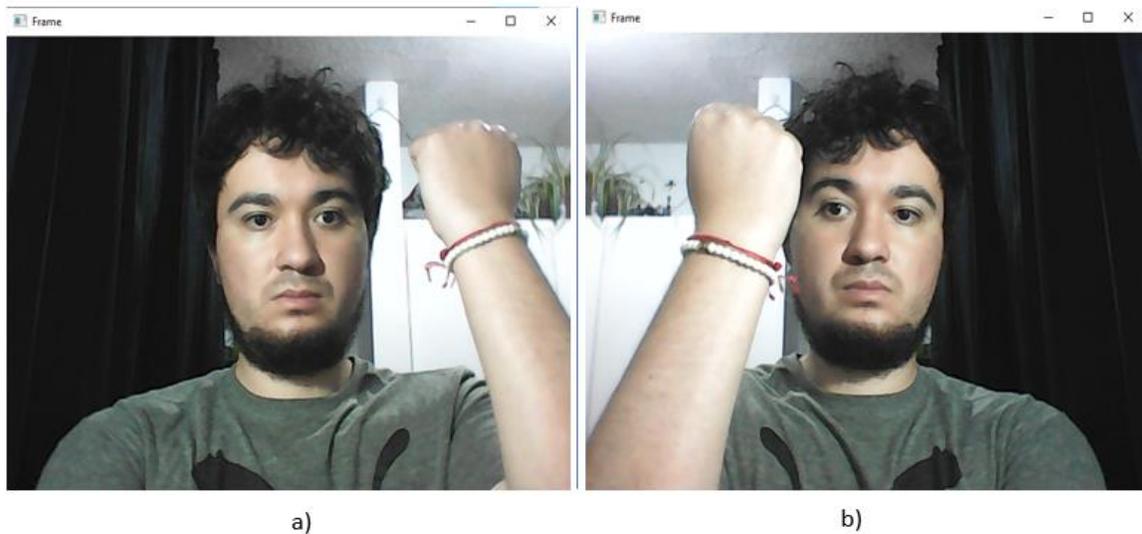
```

while True:
    _, frame = vid.read()
    frame = cv2.flip(frame, 1)
    ...

```

Figura 3

Preprocesamiento, generación de espejo en el video



Nota.

El

usuario tiene colocado la banda roja en su brazo izquierdo. a) Captura de pantalla tomada antes de aplicar la función “flip”, la banda roja se ve al lado derecho. b) Captura de pantalla tomada después de aplicar la función “flip”, la banda roja se va al lado izquierdo.

Posteriormente, se realiza un cambio de tamaño de la imagen a 640x480 píxeles utilizando la función “resize”. Con la intención de aplicar el detector de rostros, y que el costo computacional sea inferior, se procede a transformar los *frames* a escala de grises utilizando la función “cvtColor” de OpenCV, el resultado se puede ver a continuación en la Figura 4.

```
frame = imutils.resize(frame, width=680, height=480)
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```

Figura 4

Preprocesamiento, conversión a escala de grises



Una vez que el *frame* pasó por el preprocesamiento y se encuentra en escala de grises, se le aplica el detector de rostros inicializado anteriormente el cual devuelve una lista de listas que contienen dos tuplas que representan la esquina superior izquierda y la posterior derecha de una zona rectangular que se posiciona sobre la región de cada rostro detectado. Si el número de rostros detectados es mayor a cero, entonces se trabajará únicamente con la primera zona rectangular de la lista, es decir la primera cara detectada.

```
caras = detector(gray, 0)

if len(caras) > 0:
    cara = caras[0]
else:
    cv2.imshow("Frame", frame)
    key = cv2.waitKey(1) & 0xFF
    continue
```

A la zona detectada se le aplica el predictor de características faciales el cual se encarga de predecir las coordenadas de los 68 puntos característicos del rostro que se encuentra en la zona generada por el detector. Para esto, se genera un objeto “*shape*” que contiene las 68 coordenadas de las regiones faciales de interés. Para facilitar el manejo de estos valores se las transforma a un arreglo *NumPy* de 68*2 utilizando la función “*shape_to_np*” de la librería *Imutils*. De este nuevo arreglo se extrae las coordenadas correspondientes a la boca y la nariz utilizando los índices obtenidos previamente para posteriormente poder trabajar con ellos.

```
shape = predictor(gray, cara)
shape = face_utils.shape_to_np(shape)

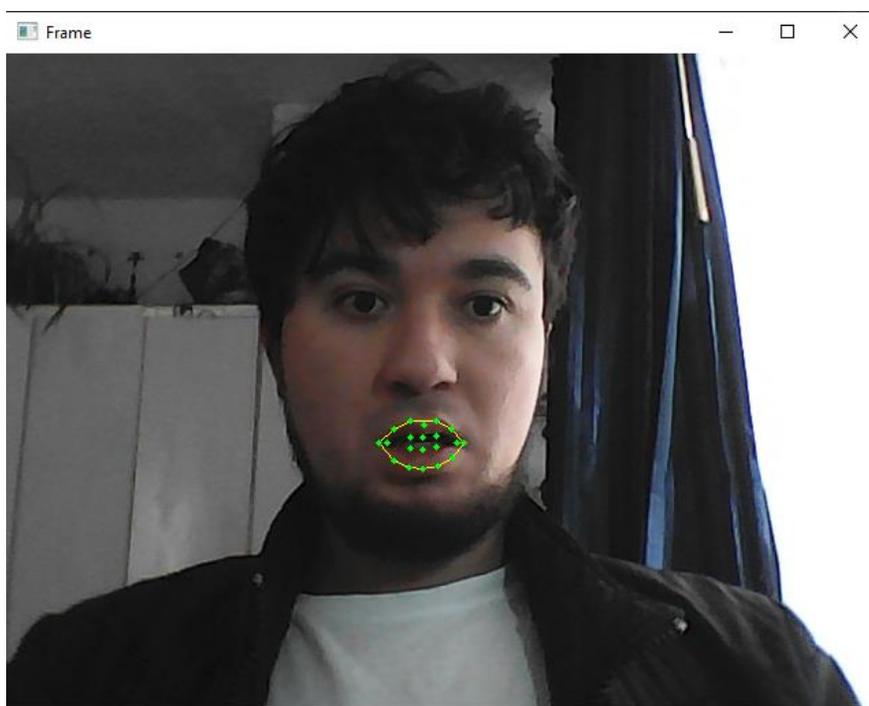
mouth = shape[mStart:mEnd]
nose = shape[nStart:nEnd]
```

A continuación, se agrega elementos a los *frames* de la imagen que no ha sido transformada a escala de grises y que será utilizada para presentar al usuario. Estos corresponden al dibujo de la boca con puntos y líneas utilizando las coordenadas correspondientes y las funciones de la librería OpenCV y Numpy que se muestran en el siguiente código. El resultado se lo puede ver en la Figura 5.

```
# Generación de contorno de la boca
mouthHull = cv2.convexHull(mouth)
cv2.drawContours(frame, [mouthHull], -1, Amarillo, 1)
#Generación de puntos en la boca
for (x, y) in np.concatenate((mouth, ), axis=0):
    cv2.circle(frame, (x, y), 2, Verde, -1)
```

Figura 5

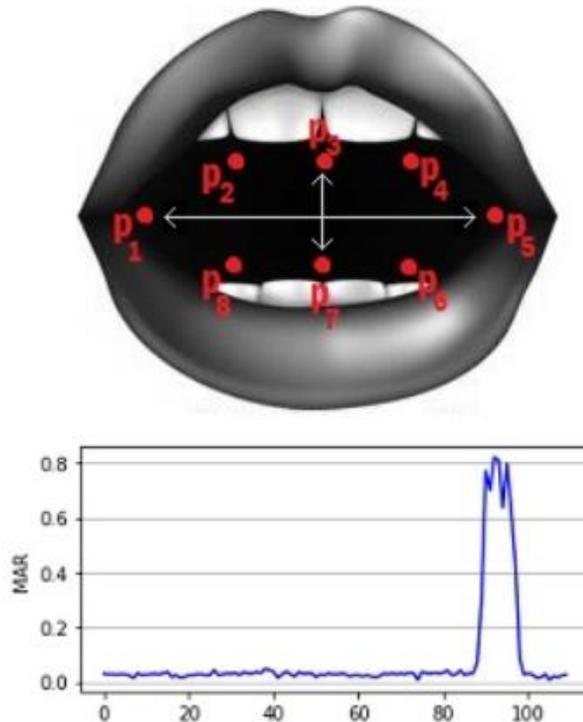
Representación de contorno y coordenadas de la boca



Se busca lograr que al abrir la boca el emulador del mouse se active, por lo que se utiliza las coordenadas previamente adquiridas para lograr cumplir con esta función. En la publicación de Soukupova & Cech (2016) se presenta una metodología para detectar y contar el número de parpadeos que hace una persona y para esto utiliza el concepto de “relación de aspecto del ojo” (EAR) en donde se establece una relación entre el ancho y el alto de las seis coordenadas que definen al ojo. Basado en el anterior trabajo, Chandra (2018) adapta los conceptos para poder detectar cuando la boca está abierta o cerrada para lo que presenta el concepto de “Relación de aspecto de la boca” o por sus siglas en inglés “MAR”, en donde de igual manera se relacionan las coordenadas de la boca con el ancho y largo que hay entre estas, en la Figura 6 se presenta los puntos que se consideran y como varía el “MAR” cuando el usuario abre la boca.

Figura 6

Relación de aspecto de la boca (MAR)



Nota. Tomado de *Mouse Cursor Control Using Facial Movements*, por Chandra, 2018, <https://towardsdatascience.com/mouse-control-facial-movements-hci-app-c16b0494a971>

Basándose en la notación de los puntos presentados en la Figura 6, la relación establecida entre los mismos está definida por la siguiente fórmula:

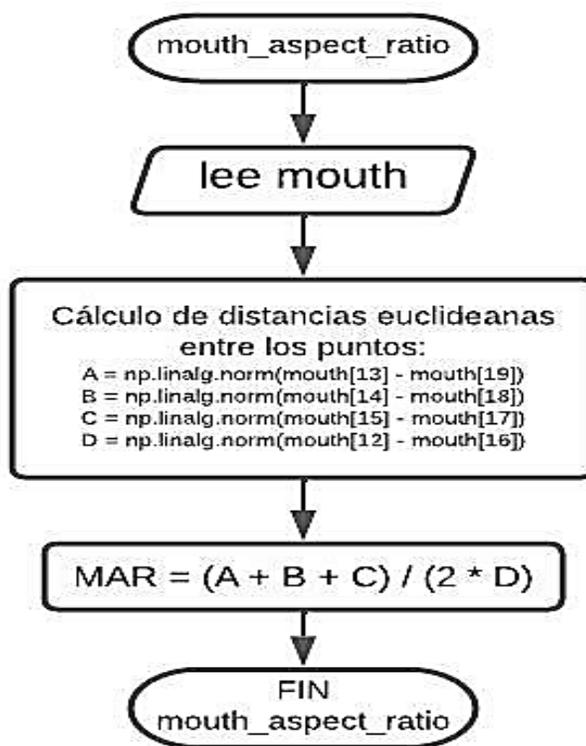
$$MAR = \frac{\|p_2 - p_8\| + \|p_3 - p_7\| + \|p_4 - p_6\|}{2\|p_1 - p_5\|}$$

Lo que se traduce en la suma de las distancias euclidianas entre los puntos que definen la altura sobre el doble de la distancia entre los puntos que definen la anchura de la boca. Para la implementación de esta aplicación se accede al módulo “utils” desarrollado por el mismo autor el cual contiene dos funciones siendo una de ellas la que permite realizar el cálculo del “MAR” ingresando el arreglo que

contiene las coordenadas de las zonas características de la boca. En esta función se utiliza la librería *Numpy* que permite realizar operaciones de arreglos relacionadas con algebra lineal entre las cuales está el cálculo de la norma de vectores, que en este caso es el resultado de la resta entre cada coordenada que genera las alturas y el par que representa el ancho. A continuación, en la Figura 7 se puede observar el diagrama de flujo de esta función.

Figura 7

Diagrama de flujo de la función "mouth_aspect_ratio" del módulo "utils"



Nota. El dato "mouth" es el arreglo de las coordenadas que corresponden a la boca.

Adicional al uso de la anterior función descrita para el cálculo del "MAR", también se define una coordenada de la nariz que será la que permita realizar los movimientos del cursor. Este punto es el número 31 de la Figura 1.

```
# Cálculo del MAR y definición del punto móvil de la nariz
mar = mouth_aspect_ratio(mouth)
P_NARIZ = (nose[3, 0], nose[3, 1])
```

Una vez que se ha obtenido el valor del “MAR” se lo compara con un valor límite de 0.6 el cual fue definido en los parámetros iniciales. Entonces, cuando supere dicho valor un contador se activará y por cada *frame* en donde el límite sea superado, se aumentará en uno. Si el contador llega a 15, entonces activará una bandera que significa que el movimiento del mouse se ha activado; si no sucede esto, el contador se reseteará. Esto significa que el usuario cuando mantenga abierta la boca durante el transcurso de 15 *frames* activará el movimiento del *mouse*, reseteará el contador y a su vez definirá la coordenada actual del punto de la nariz como una referencia. A continuación, se muestra el código que permite realizar lo anteriormente dicho.

```
if mar > MAR_LIM:
    CONT_BOCA += 1

    if CONT_BOCA >= MAR_NUM_FRAMES:

        BANDERA_MOV_MOUSE = not BANDERA_MOV_MOUSE
        CONT_BOCA = 0
        REF_NARIZ = P_NARIZ
        Im.put(BANDERA_MOV_MOUSE)

else:
    CONT_BOCA = 0
```

Previo al llamado del subprocesso se definió el objeto de la clase “Queue” denominado “Im” el cual permiten almacenar datos y compartirlos a través de los hilos. Se usa para enviar el valor de la bandera hacía el hilo “hablarcompu” para que este se ejecute.

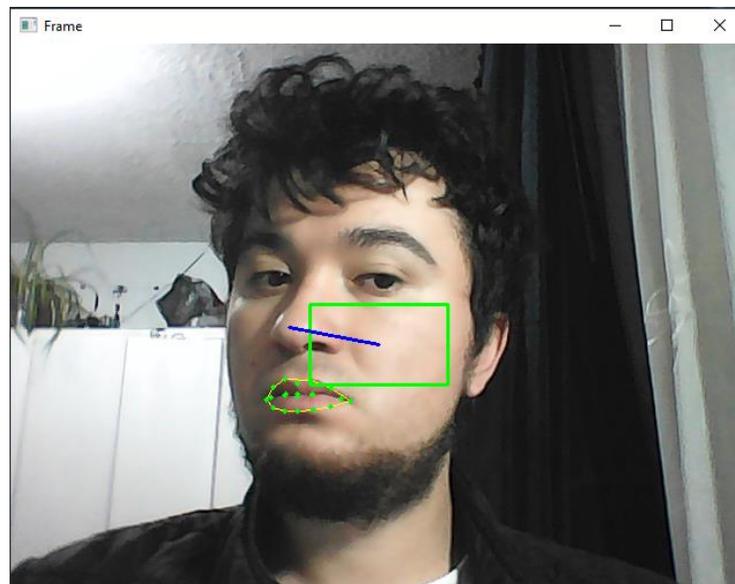
Si la bandera de activación de movimiento está seteada, entonces se generará un rectángulo de dimensiones 60*35 pixeles sobre el punto de referencia de la nariz que se capturó anteriormente.

Adicional se representa con una línea el movimiento de la nariz, teniendo como inicio el punto de referencia y como final la coordenada actual. El resultado se puede observar en la Figura 8.

```
if BANDERA_MOV_MOUSE:
    x, y = REF_NARIZ
    nx, ny = P_NARIZ
    w, h = 60, 35
    multiple = 1
    cv2.rectangle(frame, (x - w, y - h), (x + w, y + h), Verde, 2)
    cv2.line(frame, REF_NARIZ, P_NARIZ, Azul, 2)
    ...
```

Figura 8

Rectángulo y línea de referencia



Para generar el movimiento del cursor, inicialmente se utiliza la segunda función del módulo `utils` denominada “`direction`” (ver Figura 9), la cual recibe como parámetros el punto actual de la nariz, el punto de referencia previamente capturado y las dimensiones del rectángulo. A partir de esto, la función se encarga de comparar las coordenadas en “`x,y`” actuales del punto de la nariz con los segmentos del rectángulo. Por ejemplo, si el punto de la nariz salió por el segmento izquierdo del rectángulo, entonces

la función devolverá un dato tipo *string* con valor *“left”* y así sucesivamente para los casos en donde salga por arriba, abajo o la derecha. En el caso de que no salga del rectángulo devolverá el *string “none”*. Aplicando esta función se comparará lo que haya devuelto, y dependiendo de esto, se utilizará la librería *“pyautogui”* para realizar movimientos relativos del cursor en *“x”* o *“y”*, a una velocidad de 5 pixeles por cada frame. Por último, en el bucle infinito se presenta el *frame* en la pantalla.

```
dir = direction(P_NARIZ, REF_NARIZ, w, h)

drag=5
if dir == 'right':
    pyag.moveRel(drag, 0)
elif dir == 'left':
    pyag.moveRel(-drag, 0)
elif dir == 'up':
    pyag.moveRel(0, -drag)
elif dir == 'down':
    pyag.moveRel(0, drag)
```

Una de las funciones de este hilo, es la de resetear el punto de referencia generado por la nariz cuando el usuario haya realizado movimientos bruscos, y perciba que el cursor se está moviendo involuntariamente. Para realizar esto se tendrá que mantener abierta la boca hasta que se escuche el audio de *“Mouse desactivado”* y nuevamente el de *“Mouse activado”*. Esto cambiará el punto de referencia del cual se crea el rectángulo de color verde, y la línea azul que permiten determinar la dirección del movimiento del cursor.

Cabe mencionar que todo este subproceso es definido como un hilo secundario o *“demonio”*, por lo que cuando el subproceso principal termine, este también lo hará. El diagrama de flujo que representan al hilo *“video”* se muestran en la Figura 10.

Figura 9

Diagrama de flujo de la función "direction" del módulo "utils"

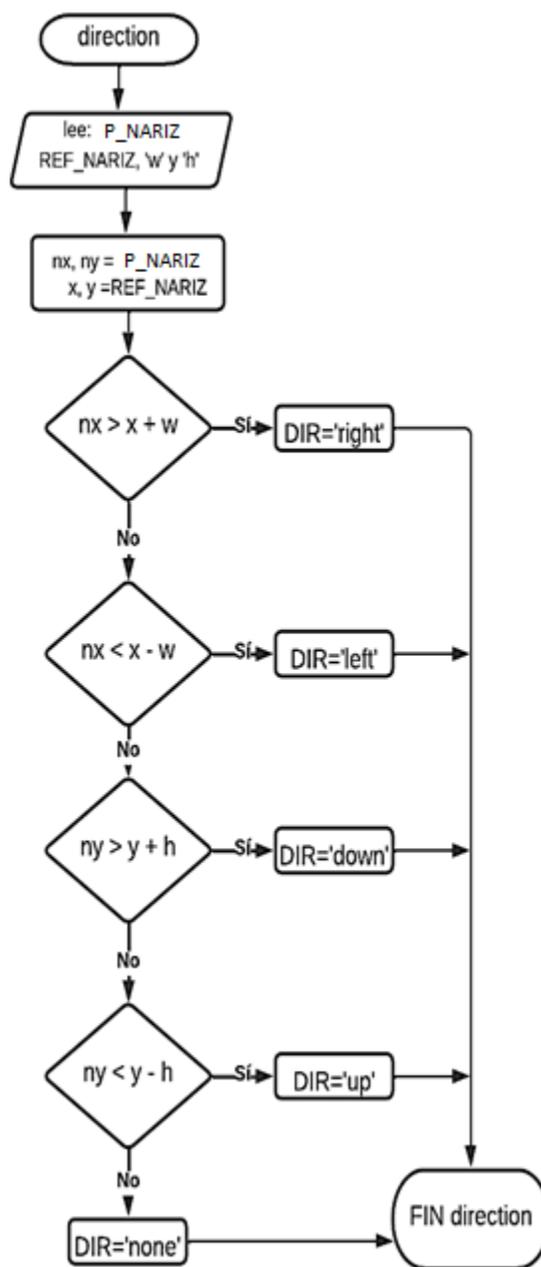
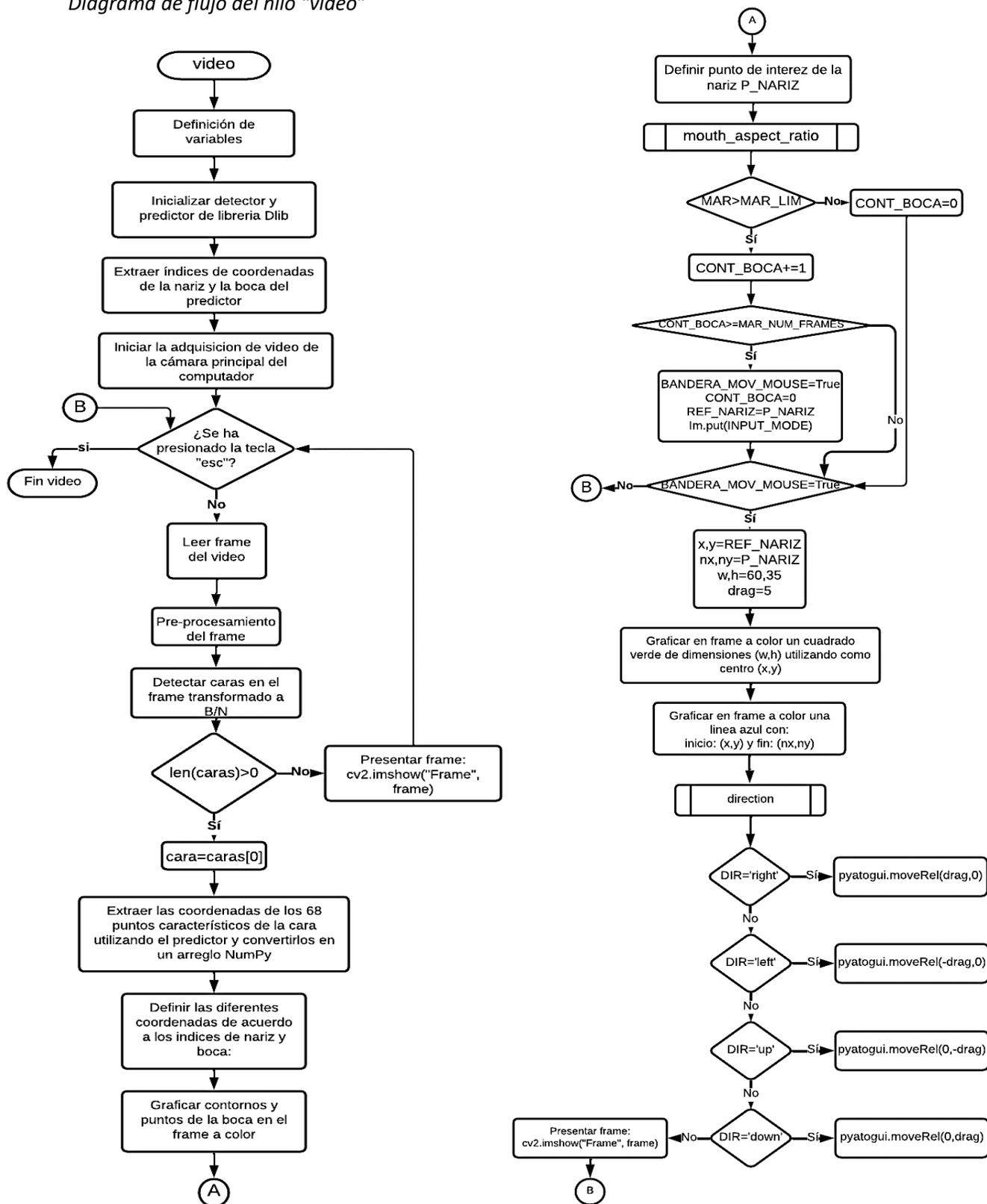


Figura 10

Diagrama de flujo del hilo "video"



Hilo “audio”

El hilo “audio” está constituido por la función “Reconocer_Voz”, la cual se encarga de acceder al micrófono del computador, y a través del reconocedor de voz, detectar las palabras que el usuario ha dicho para poder asociarlos a diferentes comandos que facilitan el manejo del computador. Inicialmente, antes de la ejecución del hilo se instancia un objeto de la clase “*Model*” del paquete “*Vosk*”, el cual permite acceder a los modelos acústicos y de lenguaje que fueron descargados previamente. También se instancia un reconocedor de voz “*recognizer*” de la clase “*KaldiRecognizer*” del mismo paquete, el cual recibe como parámetros el objeto “*model*” y la frecuencia en la que trabaja. A través de este objeto, en la función se podrá acceder a diferentes métodos que facilitarán todo el proceso de reconocimiento de voz.

```
model=Model('paths/es')
recognizer=KaldiRecognizer(model,16000)
```

En la función “Reconocer_Voz”, se instancia un objeto “*cap*” de la clase *PyAudio* que permite acceder al método “*open*”, que facilita el acceso al puerto de audio del computador para inicializar una transmisión, lo cual se logra con el método *start_stream*.

```
cap=py.PyAudio()
stream=cap.open(format=py.paInt16, channels=1,rate=16000,...
...input=True, frames_per_buffer=8192)
stream.start_stream()
```

Una vez que la transmisión de audio se ha inicializado, dentro de un bucle infinito se realizará la lectura de la señal de audio, y si el reconocedor ha detectado una señal y la ha comparado con los modelos, entonces se analizará el resultado decodificado como un *string* en minúsculas, almacenado en la variable “*texto*”. Dependiendo de la palabra que la variable contenga se comparará y se realizará una acción específica (ver Tabla 1). Con la finalidad de mejorar la precisión del reconocedor de voz, en ciertos

comandos se agregaron otras posibles interpretaciones que permitan realizar una misma acción, esto se puede observar por ejemplo en la sentencia “if”, que permite realizar un clic.

```
while (True):
    data=stream.read(4096,exception_on_overflow = False)
    if len(data)==0:
        break
    if recognizer.AcceptWaveform(data) :
        texto=recognizer.Result()
        texto=json.loads(texto)
        texto=texto["text"].lower()
        print(texto)
        if texto == "clic" or texto=="clip" or texto=="lic" or...
        ...texto=="lip clic" :
            pyag.click(button='left')
        elif texto == "uno":
            Op.put(1)
        ...
```

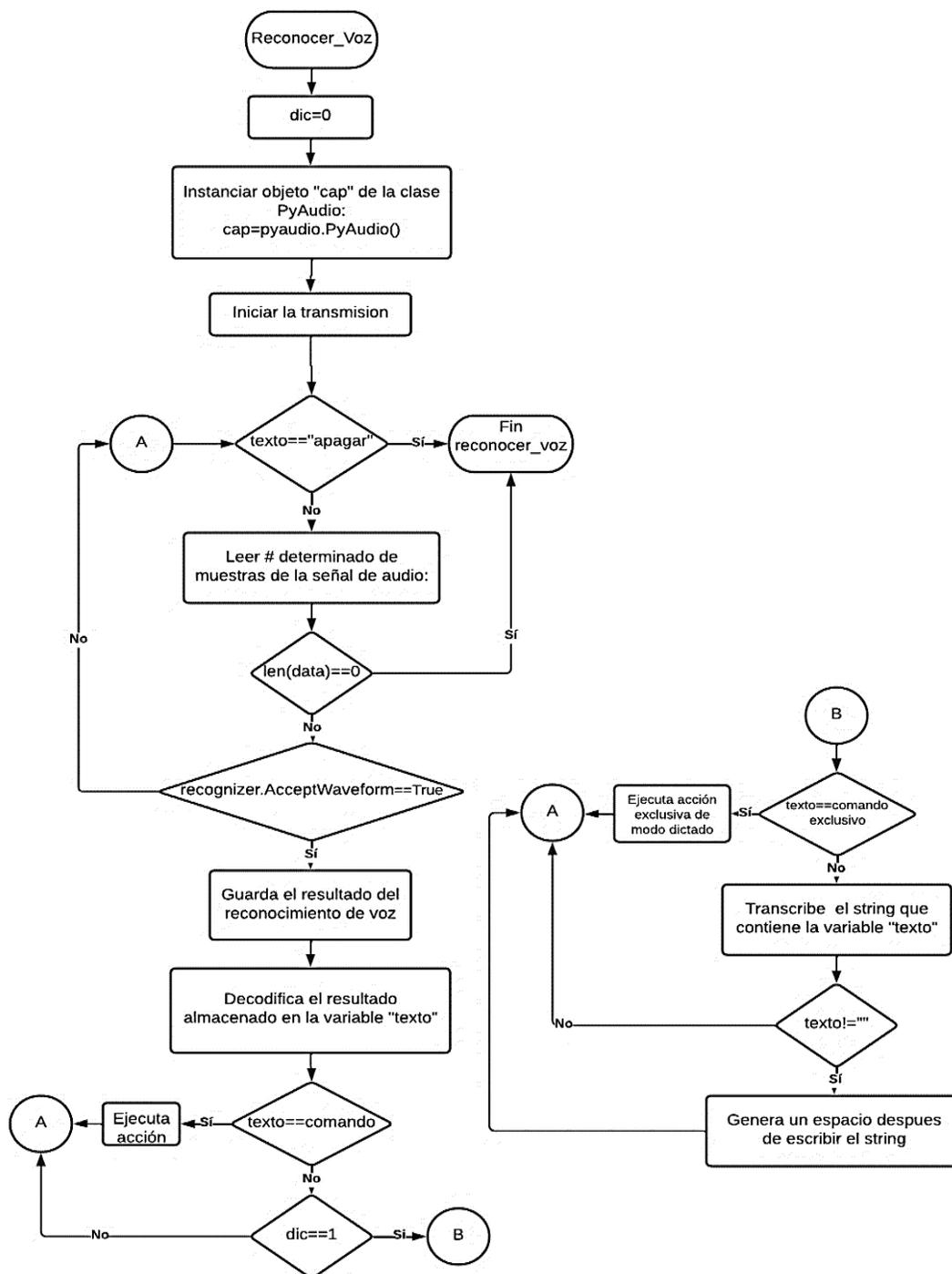
En el caso donde el usuario diga la palabra “dictado”, se cambia el valor de la variable “dic” que funciona como bandera para activar el modo dictado, el cual permite que todo lo que el usuario diga sea escrito en forma de texto en el computador, si el usuario dice la frase “salir dictado” o “apagar dictado” esta función se deshabilitará. También existen otros comandos exclusivos que solo funcionan cuando este modo está activado (ver Tabla 2), los cuales en su mayoría sirven para eliminar una cantidad específica de caracteres, según se requiera.

```
...
elif dic==1:
    if texto=="borrar":
        for x in range(1):
            pyag.hotkey('left', 'delete')
    elif texto=="borrar dos":
        for x in range(2):
            pyag.hotkey('left', 'delete')
    ...
```

Previo al llamado de los subprocesos se definió el objeto de la clase “Queue” denominado “Op”, que facilita la transmisión de datos a través de los hilos, es usado para almacenar valores enteros en el hilo “audio”, para después ser compartidos al hilo “comandos”, en donde se ejecutarán acciones dependiendo del valor que contenga el objeto. El diagrama de flujo que representa a la función “Reconocer_Voz”, del hilo “audio” se lo puede ver a continuación en la Figura 11.

Figura 11

Diagrama de flujo de la función "Reconocer_Voz" del hilo "audio"



Nota. Este diagrama de flujo resume la ejecución de las acciones. Asociar "comando" y "comando exclusivo" con los comandos y funciones presentadas en las Tablas 1 y 2.

Tabla 1*Comandos generales para realizar acciones con el emulador*

Comando	Función
uno	Coloca el valor de "1" en el objeto Op
dos	Coloca el valor de "2" en el objeto Op
tres	Coloca el valor de "3" en el objeto Op
cuatro	Coloca el valor de "4" en el objeto Op
cinco	Coloca el valor de "5" en el objeto Op
seis	Coloca el valor de "6" en el objeto Op
siete	Coloca el valor de "7" en el objeto Op
ocho	Coloca el valor de "8" en el objeto Op
nueve	Coloca el valor de "9" en el objeto Op
diez	Coloca el valor de "10" en el objeto Op
once	Coloca el valor de "11" en el objeto Op
doce	Coloca el valor de "12" en el objeto Op
inicio	Coloca el valor de "17" en el objeto Op
escritorio	Coloca el valor de "18" en el objeto Op
centro	Coloca el valor de "19" en el objeto Op
clic	Realiza un clic
"doble" o "doble clic"	Realiza doble clic
derecho	Realiza un clic derecho
sujetar	Mantiene presionado clic izquierdo
soltar	Deshace acción "sujetar"
cerrar	Cierra la ventana actual
minimizar	Minimiza la ventana actual
ventanas	En Windows, abre el menú de ventanas abiertas
enter	Realiza la función de la tecla "ENTER"
dictado	Coloca el valor de 1 en la variable "dic", llama a la función compuHabla2 con parámetro: estado=1
salir dictado	Coloca el valor de 0 en la variable "dic", llama a la función compuHabla2 con parámetro: estado=0
apagar	Cierra el emulador

Tabla 2*Comandos exclusivos para modo dictado*

Comando	Función
borrar	Borra 1 caracter
borrar dos	Borra 2 caracteres
borrar tres	Borra 3 caracteres
borrar cuatro	Borra 4 caracteres
borrar cinco	Borra 5 caracteres
borrar diez	Borra 10 caracteres
borrar mucho	Borra 40 caracteres
espacio	Genera un espacio en blanco en el texto

Es importante resaltar que, en cuanto a la jerarquía de los hilos del programa, el hilo “audio” es el principal, por lo que es el único al que no se lo define como “demonio”, entonces cuando termina su ejecución al decir la palabra “apagar”, el programa con todos sus hilos secundarios finalizará.

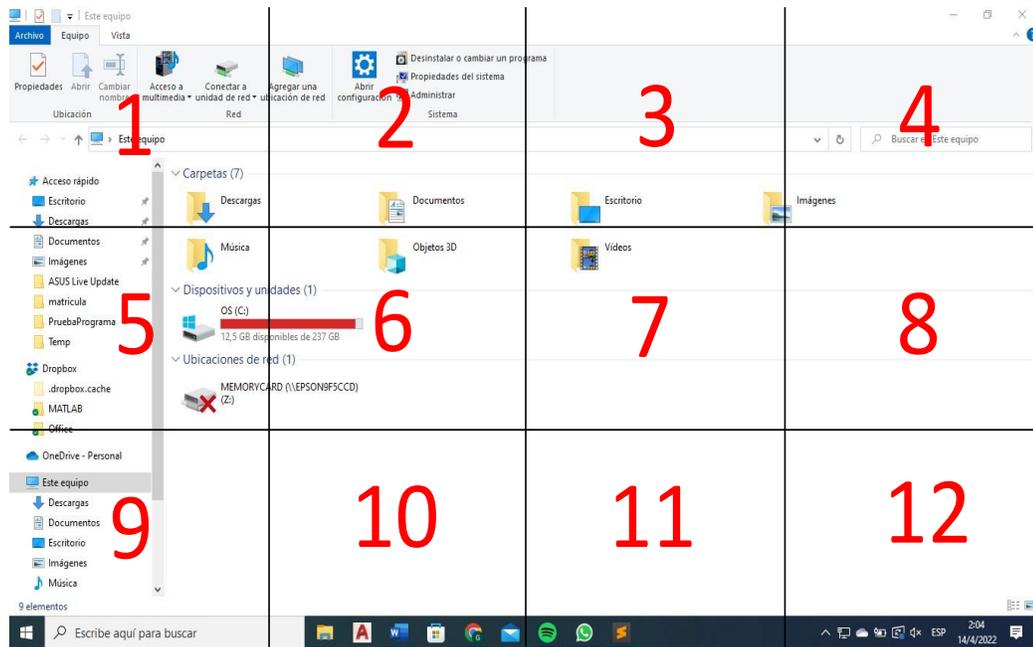
Hilo “comandos”

Este hilo está constituido por la función “Ejecutar_Comandos”, la cual recibe como entrada el objeto “Op” que proviene del hilo “audio”, con un valor diferente que depende del comando que se desee ejecutar. Las funcionalidades de este hilo están enfocadas en posicionar al cursor en determinadas zonas, se encarga de realizar la segmentación de pantalla, lo que facilita al usuario la navegación por la interfaz del computador. Para lograr esto, inicialmente se adquiere las dimensiones de la pantalla a través de la función “size” de “PyAutogui”. Se busca segmentar la pantalla en doce cuadrantes como se ve en la Figura 12, para lo que se definen cuatro variables que servirán para colocar las coordenadas, donde tendrá que posicionarse el cursor. La variable “a” es la cuarta parte del ancho de la pantalla, “b” es el tercio de la altura de la pantalla, “posA” es la mitad de “a” y “posB” es la mitad de “b”. Dentro de un bucle infinito, que se ejecuta hasta que el hilo “audio” acabe, se verifica si el objeto “Op” está vacío, entrando así a otro bucle que se ejecutará cuando esta condición sea falsa, y dependiendo del valor que se haya extraído del

objeto en la variable “var”, se ejecutará una acción de posicionamiento en una zona específica de la pantalla utilizando la función “moveTo”.

Figura 12

Posición de los segmentos en que se divide la pantalla



Para lograr la segmentación de pantalla se debe colocar las coordenadas a las que el cursor se moverá, por lo que se utilizaron las siguientes fórmulas como condiciones, dependiendo del valor que se almacena en “var”, el cual coincide con el número de cuadrante en donde se desea posicionarse:

- Si el usuario dice un número del 1-4, la coordenada será definida por:

$$\{posA + [a * (var - 1)], posB\}$$

- Si el usuario dice un número del 5-8, la coordenada será definida por:

$$\{posA + [a * (var - 5)], posB + b\}$$

- Si el usuario dice un número del 9-12, la coordenada será definida por:

$$\{posA + [a * (var - 9)], posB + 2b\}$$

Para valores de “Op” que estén entre 17-19, también se ejecutan acciones de posicionamiento que permiten llegar de forma más rápida a zonas de interés. En la Tabla 3, se detalla los comandos de voz que el usuario tiene que decir para enviar un valor de “Op”, que sirva para realizar alguna de las acciones de posicionamiento o segmentación.

Tabla 3

Comandos de segmentación y posicionamiento

Comando	Valor "Op"	Posición
uno	1	Cuadrante 1
dos	2	cuadrante 2
tres	3	Cuadrante 3
cuatro	4	Cuadrante 4
cinco	5	Cuadrante 5
seis	6	Cuadrante 6
siete	7	Cuadrante 7
ocho	8	Cuadrante 8
nueve	9	Cuadrante 9
diez	10	Cuadrante 10
once	11	Cuadrante 11
doce	12	Cuadrante 12
inicio	17	Realiza un clic en la esquina inferior izquierda, ícono de inicio.
escritorio	18	Realiza un clic en la esquina inferior derecha, sobre barra de acceso rápido a escritorio en Windows y posiciona el cursor en el centro de la pantalla.
centro	19	Posiciona el cursor en el centro de la pantalla.

A continuación, en la Figura 13 se presenta el diagrama de flujo de la función “Ejecutar_Comandos” del hilo “comandos”.

Figura 13

Diagrama de flujo de la función "Ejecutar_Comandos" del hilo "comandos"

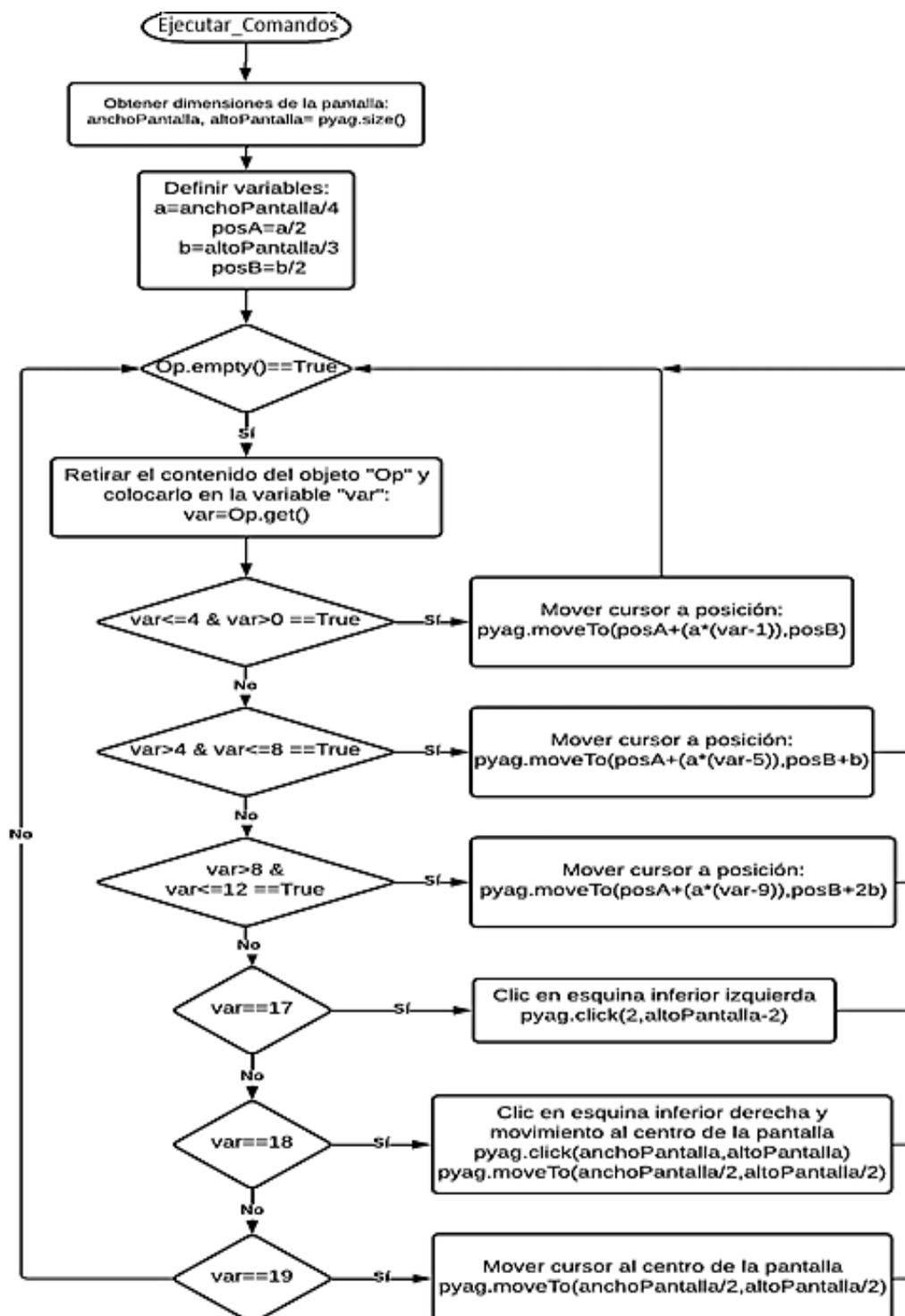
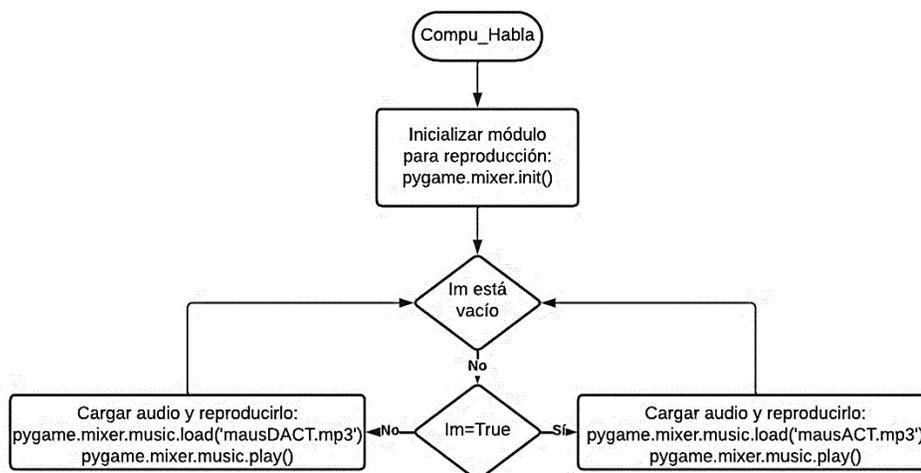


Figura 14

Diagrama de flujo de la función "Compu_Habla" del hilo "hablarcompu"



Por otro lado, la función "Compu_Habla2" es llamada únicamente desde el hilo "audio" cuando el modo dictado es activado o desactivado. Para cada caso se envía un parámetro "estado" a la función que puede ser 1 o 0, dependiendo del comando de voz que el usuario haya pronunciado, el primero es para reproducir un audio que permita indicar que el modo dictado se ha activado, reproduciendo una voz que dice "modo dictado activado", y el segundo que se ha desactivado diciendo "modo dictado desactivado". El diagrama de flujo (Figura 15), y el código de esta función se los puede ver a continuación.

```

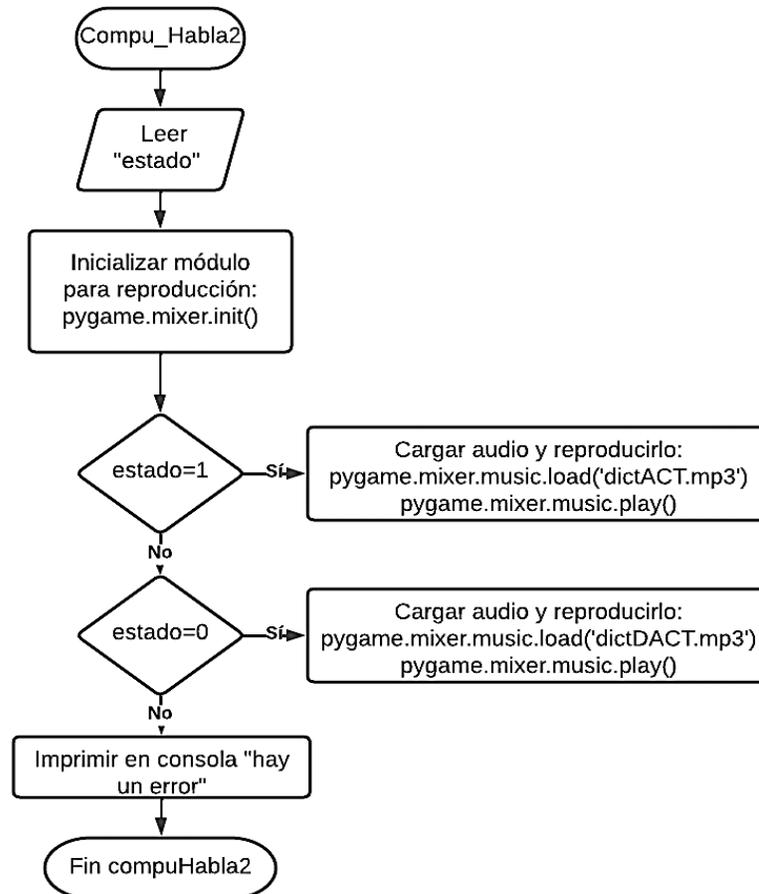
def Compu_Habla2(estado):
    pygame.mixer.init()

    if estado==1:
        pygame.mixer.music.load('dictACT.mp3')
        pygame.mixer.music.play()
        while pygame.mixer.music.get_busy():
            time.sleep(0.5)
        break
    elif estado==0:
        pygame.mixer.music.load('dictDACT.mp3')
        pygame.mixer.music.play()
        while pygame.mixer.music.get_busy():
            time.sleep(0.5)
        break
  
```

```
else:  
    print("hay un error")  
    time.sleep(0.5)
```

Figura 15

Diagrama de flujo de la función "Compu_Habla2"



Capítulo IV. Características del Hardware y plataformas de prueba

En este capítulo se mostrará las características de hardware del computador que se utilizó para el desarrollo del programa, y las pruebas de funcionamiento iniciales, siendo estas consideradas como las características mínimas para que el emulador tenga un buen desempeño. También se describirán otras plataformas en donde se realizaron pruebas de desempeño, con la finalidad de estudiar el funcionamiento del sistema usando otro tipo de hardware.

Características generales de hardware

En este capítulo se busca describir los requisitos de hardware necesarios para que el sistema desarrollado funcione de manera correcta, utilizando como referencia las características del computador en donde se ha desarrollado la programación del software, y gran parte de las pruebas de desempeño. El programa fue desarrollado en una laptop *ASUS Vivobook 15*, utilizando el sistema operativo *Windows 10 de 64 bits*. Como elementos fundamentales para el funcionamiento del emulador es necesario disponer de una cámara, un micrófono, un procesador y memoria RAM que soporte realizar tareas de procesamiento de imágenes continuas. Para realizar las pruebas de funcionamiento se utilizó la *webcam* integrada en el computador y un micrófono externo. A continuación, se detallará las características de dichos componentes.

Procesador y características generales

El computador portátil en el que se trabajó posee un procesador *Intel Core i5-8250U* y una memoria RAM de 8GB, la tarjeta gráfica viene integrada en el mismo chip del procesador y es una *Intel UHD Graphics*, el disco duro es de 256GB. Debido a estas características se puede considerar que el computador es de una gama media-baja, cuyo costo oscila entre 425 a 500 dólares.

Cámara. La cámara que se utilizó es una webcam estándar VGA, integrada en el computador (*USB2.0 VGA UVC WebCam*), la cual tiene una resolución característica de 640X480 píxeles.

Micrófono. Para receptor las señales de voz del usuario y que el algoritmo de detección de voz funcione de manera precisa, se utilizó un micrófono externo con la finalidad de que este se posicione lo más cercano a la boca del usuario. El dispositivo utilizado para las pruebas fue un micrófono *Genius MIC-01A* (ver Figura 16), el cual tiene un rango de frecuencia entre 100 y 11000 Hz.

Figura 16

Micrófono Genius MIC-01A



Otras plataformas de prueba

A continuación, se presentan las características básicas de otros computadores en donde se realizaron pruebas de desempeño.

Toshiba Satellite L 15W-B

Es una laptop de gama baja que dispone de un procesador Intel Pentium N3540 y una memoria RAM de 4GB. Funciona con el sistema operativo Windows 10 de 64 bits. La máquina posee cámara y micrófono interno de características básicas por lo que para realizar las pruebas se consideró el uso del micrófono de la Figura 16.

Biostar H61MHB

Es una computadora de escritorio de gama baja que dispone de un procesador Intel Core i3-3220 y una memoria RAM de 6GB. Funciona con el sistema operativo Windows 8 de 64 bits. Para realizar la prueba se utilizó una cámara externa de alta definición Adesso CyberTrack H4 (ver Figura 17) y el micrófono de la Figura 16.

Figura 17

Webcam Adesso CyberTrack H4



Nota. Tomado de *Adesso CyberTrack H4 1080p USB Webcam with Built-in Microphone*, por Adesso, 2022, <https://www.adesso.com/product/cybertrack-h4-1080p-hd-usb-webcam-with-built-in-microphone/>

Lenovo 81D6

Es una laptop de gama baja que dispone de un procesador AMD A9-425 y una memoria RAM de 8 GB. El sistema operativo con el que dispone es Windows 10. La computadora dispone de una webcam integrada VGA con la que se realizaron las pruebas, y además se utilizó el micrófono de la Figura 16.

Dell Inspiron 15 3515

Es una laptop de gama media que dispone de un procesador AMD Ryzen 5 3450U, con gráficos Radeon Vega Mobile Gfx y una memoria RAM de 8 GB. El sistema operativo con el que funciona es

Windows 11 de 64 bits. La computadora posee una cámara VGA y un micrófono que funciona de buena manera, por lo que no fue necesario utilizar dispositivos externos para las pruebas.

ASUS K53E

Es una laptop de gama alta antigua que cuenta con un procesador Inter Core i7-2670 y una memoria RAM de 8GB. El sistema operativo con el que funciona es Windows 7 de 64 bits. La cámara y el micrófono interno se encuentran en buen estado, aunque para las pruebas se intentó usar un micrófono externo.

Dell G3 3579

Es una laptop de gama alta que trabaja con un procesador Intel Core I7-8750H, con una tarjeta gráfica NVIDIA GeForce GTX 1050 y dispone de una memoria RAM de 16 GB. El computador funciona con sistema operativo Windows 10 de 64 bits. En cuanto a la cámara y el micrófono integrado, estos funcionan de una muy buena manera, por lo que no se requirió del uso de dispositivos externos.

Toshiba Satellite P55W-C

Es una laptop de gama alta que trabaja con un procesador Intel Core I7-6500H, dispone de una memoria RAM de 12 GB. Esta máquina funciona con un sistema operativo Windows 10 de 64 bits. En cuanto a la cámara y el micrófono integrado, estos funcionan de una muy buena manera, por lo que no se requirió del uso de dispositivos externos.

HP - HP Laptop 15-db1xxx

Es una laptop de gama baja que trabaja con un procesador AMD Ryzen 3 3200, dispone de una memoria RAM de 8 Gb. Funciona con un sistema operativo Windows 10 de 64 bits. Para realizar las pruebas se utilizó el mismo micrófono y cámara integrada en la computadora.

Toshiba Satellite P75-A7200

Es una laptop de gama alta que trabaja con un procesador Intel Core I7-4700MQ, dispone de una memoria RAM de 8 GB. Esta máquina funciona con un sistema operativo Windows 8 de 64 bits. En cuanto a la cámara y el micrófono integrado, estos funcionan de una muy buena manera, por lo que no se requirió del uso de dispositivos externos.

A continuación, se presentarán diferentes pruebas que se hicieron con el emulador, de las cuales varias fueron realizadas en los dispositivos nombrados en este capítulo. Se buscará validar el funcionamiento del emulador bajo diferentes situaciones, su rendimiento en computadores con diferentes características de hardware y software, y a través de pruebas en varios usuarios.

Capítulo V. Pruebas de desempeño y análisis de resultados

En este capítulo se muestran las diferentes pruebas que se realizaron para validar el funcionamiento y la eficacia del emulador. Se utilizó el programa en diferentes plataformas y situaciones para verificar como es su respuesta, y determinar cuáles son las condiciones óptimas para que el funcionamiento sea correcto. Además, se realizaron pruebas con diferentes personas para lograr medir su utilidad, y verificar el cumplimiento de los objetivos del proyecto. En cada prueba se presentará el análisis de los resultados que se obtuvieron.

Pruebas generales

A continuación, se detallarán las pruebas de carácter general que se realizaron con el emulador, las cuales consistieron en probar su desempeño con diferentes niveles de luminosidad, cuando hay más de un rostro en la imagen captada por la cámara y cuando hay interferencias de ruido causado por el habla de otras personas.

Variación de iluminación

El desempeño del emulador depende en gran parte de la claridad de la imagen, es por esto que se realizó una prueba con una cámara VGA para verificar los niveles de iluminación, en los cuales el sistema logre funcionar de manera adecuada. Para realizar estas pruebas se procedió a utilizar el emulador en diferentes ambientes con diferentes niveles de luz. Utilizando una aplicación móvil y el sensor de luminosidad, se obtuvieron resultados en *luxes* (lumen/m^2). La prueba consistió en lograr activar el movimiento del mouse con la apertura de la boca, si se logra realizar esta acción sin problema se considerará que se encuentra en un nivel de luz adecuado para el funcionamiento del emulador. A continuación, se presentarán las imágenes que fueron capturadas en los diferentes ambientes.

Figura 18

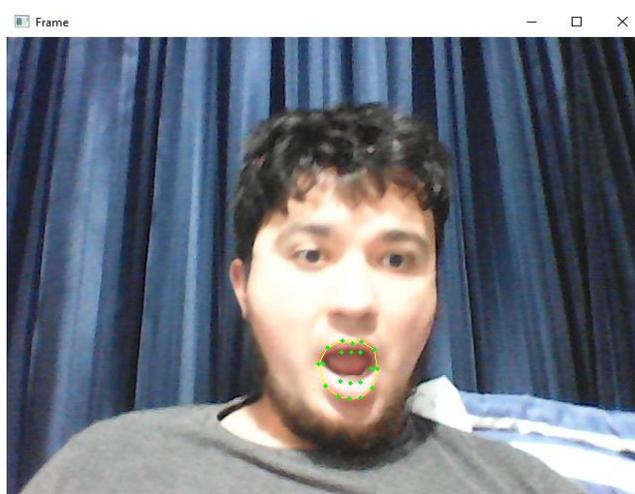
Rostro con iluminación de 469 luxes



En la Figura 18 se puede observar la captura de imagen del rostro con una iluminación de 469 luxes, para lograr este valor de luz se utilizó una lámpara de escritorio encendida. Al realizar la prueba, el desempeño del emulador fue muy bueno y la detección del rostro fue muy precisa por lo que, al abrir la boca, de forma inmediata se activó el movimiento del mouse.

Figura 19

Rostro con iluminación de 107 luxes



En la Figura 19 se pudo observar una imagen capturada con una luminosidad de 107 luxes, a estos niveles el rostro es reconocido sin problema y el predictor de gestos faciales logra obtener las coordenadas adecuadas para la boca, por lo que, al realizar la prueba, rápidamente se activó el movimiento del cursor.

Figura 20

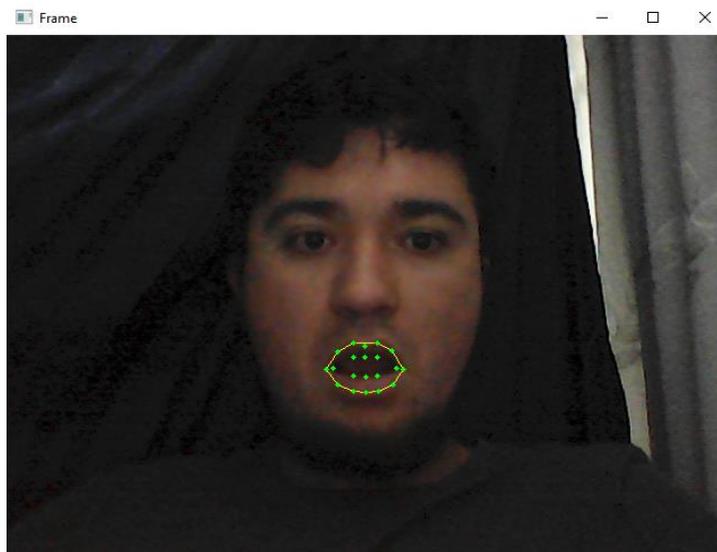
Rostro con iluminación de 19 luxes



En la Figura 20, la zona está iluminada con 19 luxes, en este caso, la detección de la boca tardó un tiempo considerable para poder activar el modo de movimiento del cursor. En comparación con las dos imágenes anteriores, se puede observar que el rostro se lo ve más oscuro, por lo que el algoritmo empieza a tener problemas en la detección y la predicción de las coordenadas de la boca las cuales llegan a ser imprecisas, en comparación a los dos casos anteriores. Sin embargo, se puede observar en la esquina superior izquierda una zona muy clara que proviene de un foco encendido, lo cual en este caso parece alterar el enfoque de la cámara, lo que quita iluminación a la imagen del rostro.

Figura 21

Rostro con iluminación de 5 luxes



En la Figura 21 se puede observar el rostro con una iluminación de 5 luxes, al momento de realizar la prueba el algoritmo detectó de una buena manera el rostro, por lo que logró realizar una buena predicción de las coordenadas de la boca, sin embargo, tardó más tiempo en detectar la apertura de la boca. A pesar de que en este ambiente se haya tenido un menor nivel de luminosidad, el resultado fue mejor a comparación del presentado en la figura anterior, esto debido a que el enfoque del rostro es mejor por lo que se ve con mayor claridad. Esto facilita la detección del rostro y, por consiguiente, la predicción de las coordenadas de la boca y la nariz.

En estas pruebas de iluminación se puede notar que el rendimiento del emulador es óptimo, mientras la cámara del computador detecte con claridad el rostro. Para esto se puede comparar las Figuras 18 y 19, en donde con una buena iluminación se puede obtener un rendimiento adecuado del detector y respuestas más rápidas a los gestos faciales. A detalle se puede ver que, en estas capturas, los labios se encuentran rodeados por los puntos de coordenadas y las líneas de forma precisa, esto no sucede de la misma forma en la Figura 20, en donde se puede ver que la predicción es incorrecta, ya que los

puntos y las líneas no encajan perfectamente con el labio inferior. Esto es debido al mal enfoque y ajuste de claridad del *frame* con el que trabaja el algoritmo, lo que genera un rostro más oscuro. En el caso de la Figura 21 se puede notar que la predicción de las coordenadas del labio es precisa, a pesar de que la imagen y la luminosidad sea menor, esto sucede gracias al enfoque y el ajuste de luminosidad, lo que permite mostrar un rostro más claro a comparación de la Figura 20.

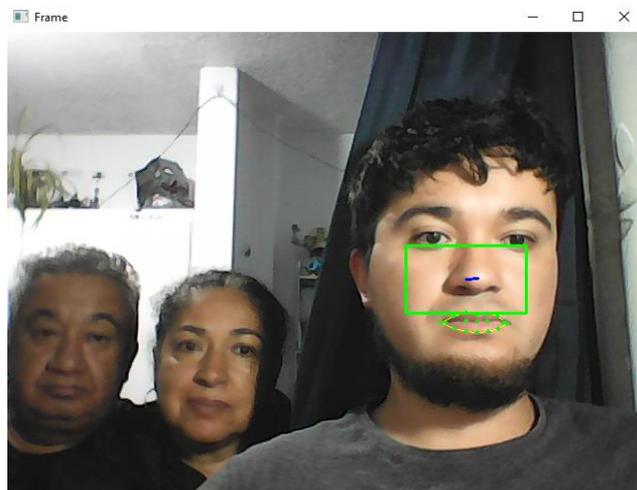
Prueba con varios rostros en el frame

Esta prueba consiste en probar el funcionamiento del emulador cuando el algoritmo detecta más de un rostro en el *frame*. Para esto se procedió a utilizar el sistema mientras está activado el movimiento del cursor y observar cómo es su funcionamiento cuando hay más personas alrededor.

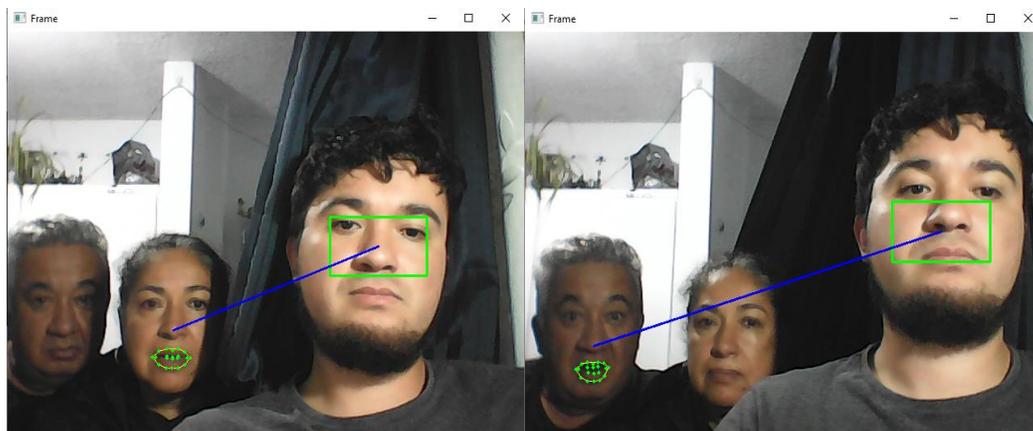
El algoritmo de detección de rostros se encarga de generar una lista con número de elementos igual al número de rostros presentes en un *frame*. El hilo "video" va realizando este procedimiento consecutivamente, por lo que la lista de los rostros se sustituye cada vez que analiza un *frame*, entonces, en el caso que otras personas aparezcan, es posible que sus rostros ocupen el primer lugar de la lista, el cual es el que se usa para realizar las funciones de movimiento del cursor. En la Figura 22, se puede observar que una de las tres personas ha activado el movimiento del cursor, ya que es su rostro el primero que está siendo detectado, sin embargo, en la Figura 23, se puede ver dos funcionamientos erróneos, debido a que se ha detectado primero otro rostro que no es el que inicialmente activó el movimiento. Esto generará que se realicen movimientos del cursor no deseados, por lo que es necesario que frente la cámara se encuentre un solo usuario.

Figura 22

Frame con más de un rostro (Funcionamiento normal)

**Figura 23**

Frames con más de un rostro (Funcionamiento erróneo)



Prueba con voces de más de una persona

Para realizar esta prueba se solicitó a dos personas que mantengan una conversación cerca del computador, mientras se estaba utilizando el emulador, esto con la finalidad de observar cómo influye el ruido y otras voces a la hora de ejecutar los comandos. A partir de esto se pudo observar que en este aspecto el sistema depende de la sensibilidad del micrófono que se esté usando, y de la distancia entre

las otras personas y el micrófono. Para las pruebas realizadas en el computador se utilizó su micrófono interno, que es de baja calidad, y el micrófono externo de la Figura 16. Mientras se estuvo usando el micrófono propio del computador, el sistema no captó otras voces aparte de la del usuario del emulador, sin embargo, el detector entendía otras palabras que no eran la de los comandos, por lo que no funcionaba correctamente. Cuando se conectó el micrófono externo el funcionamiento mejoró en cuanto a la detección de los comandos correctos, sin embargo, frecuentemente se mezclaban los comandos con palabras que habían pronunciado las personas que estaban hablando cerca. Cuando se alejaron del micrófono a mantener su conversación más lejos, el funcionamiento del emulador mejoró considerablemente. A partir de esto, se puede considerar necesario un ambiente moderadamente silencioso para lograr que el emulador funcione de buena manera.

Pruebas de inicio automático

Debido a que el emulador está enfocado para personas con discapacidades motrices, se buscó la manera de lograr que se inicie automáticamente al momento de abrir la sesión en el computador. Para lograr esto en sistemas operativos de Windows, simplemente se tuvo que acceder a la siguiente dirección:

C:\ProgramData\Microsoft\Windows\Start Menu\Programs\StartUp

En esta carpeta, se colocan todos los programas que se desea que se ejecuten al momento de iniciar la sesión. El procedimiento para lograr esto, consistió en copiar y pegar el acceso directo del emulador. Luego se reseteó la computadora y se verificó que el programa empezó a funcionar de forma automática.

Pruebas de desempeño en distintas plataformas

Con la finalidad de comprobar que el sistema funcione correctamente en distintas plataformas, se procedió a generar un archivo ejecutable del emulador y subirlo en la nube para que pueda ser compartido.

En esta sección se presentan pruebas de funcionalidad y rendimiento en varias computadoras de distintas gamas, con la finalidad de comparar como es el desempeño del emulador, a medida que las características de hardware y software varían. En este caso todas las máquinas tienen sistemas operativos de diferentes versiones de Windows. Se utilizaron los computadores especificados en el anterior capítulo, en donde se detalló sus características y la gama a la que pertenecen. Como parámetros para verificar que el funcionamiento del emulador sea correcto en cada una de las computadoras, se establecieron varios aspectos para analizar a través de las siguientes preguntas:

- ¿Cuál es el costo computacional con el emulador activo?
- ¿La velocidad de movimiento del cursor es la esperada?
- Usando el micrófono más adecuado, ¿es rápida la respuesta del emulador a los comandos de voz?
- ¿Funcionan correctamente todos los comandos de voz?
- ¿Es necesario el uso de cámara externa?
- ¿Es necesario el uso de micrófono externo?
- ¿El emulador funcionó con normalidad?

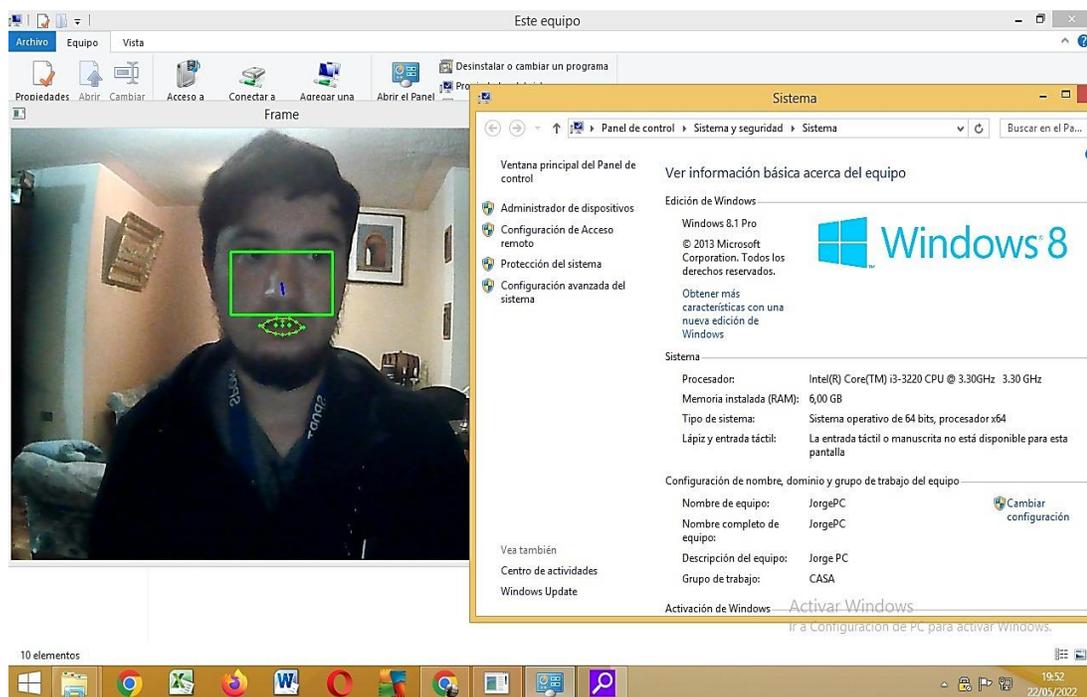
A continuación, se presentará el análisis individual del funcionamiento del emulador en cada una de las computadoras descritas.

Pruebas en computador Biostar H61MHB

Para utilizar el emulador en este computador se utilizó la cámara y el micrófono externo que se explicaron anteriormente. Las pruebas se realizaron en un ambiente silencioso y correctamente iluminado, lo que permitió una buena detección del rostro y de los comandos de voz. La prueba consistió en realizar la consulta de las características del computador utilizando el emulador (ver Figura 24). A partir de esto se pudo encontrar respuestas a las preguntas anteriormente planteadas.

Figura 24

Detalles básicos del computador Biostar H61MHB consultados usando el emulador



En cuanto al costo computacional, se puede observar en la Figura 25, que el programa ocupa un 23% de la capacidad de cómputo de la máquina y 239,5MB de los 6GB de memoria RAM disponibles. Esto indica que se podrá usar el emulador sin problema, siempre y cuando no se sature el computador con muchas aplicaciones abiertas a la vez.

Figura 25

Costo computacional del emulador en Biostar H61MHB

Nombre	Estado	51% CPU	40% Memoria	1% Disco	0% Red
Aplicaciones (4)					
Administrador de tareas		0,3%	8,9 MB	0 MB/s	0 Mbps
Emulador_Mouse		23,0%	239,5 MB	0 MB/s	0 Mbps
Explorador de Windows		0,1%	59,1 MB	0,1 MB/s	0 Mbps
Google Chrome		0%	55,6 MB	0 MB/s	0 Mbps
Procesos en segundo plano (44)					
Adobe Acrobat Update Service (...)		0%	0,8 MB	0 MB/s	0 Mbps
Aislamiento de gráficos de disp...		0,4%	5,5 MB	0 MB/s	0 Mbps
Aplicación de subsistema de cola		0%	10,3 MB	0 MB/s	0 Mbps
AVG Antivirus Installer (32 bits)		0%	0,1 MB	0 MB/s	0 Mbps
AVG Emergency Update (32 bits)		0%	0,2 MB	0 MB/s	0 Mbps
AVG Overseer		0%	0,1 MB	0 MB/s	0 Mbps
AVG Service (32 bits)		0%	22,9 MB	0 MB/s	0 Mbps
AvLaunch component (32 bits)		0%	0,1 MB	0 MB/s	0 Mbps

Al usar el emulador, se pudo verificar que la velocidad de movimiento del cursor fue la esperada y no causó ningún inconveniente a la hora de realizar la consulta de las características. La velocidad de respuesta a los comandos de voz fue buena en la mayoría de casos, con ciertas excepciones cuando el volumen de voz no fue el adecuado, por lo que se puede concluir que el desempeño en esta característica fue muy bueno. Para verificar que los comandos de voz funcionen de manera correcta, se los probó uno por uno y en su totalidad funcionaron, por lo que se puede decir que el emulador trabaja correctamente en el sistema operativo de Windows 8.

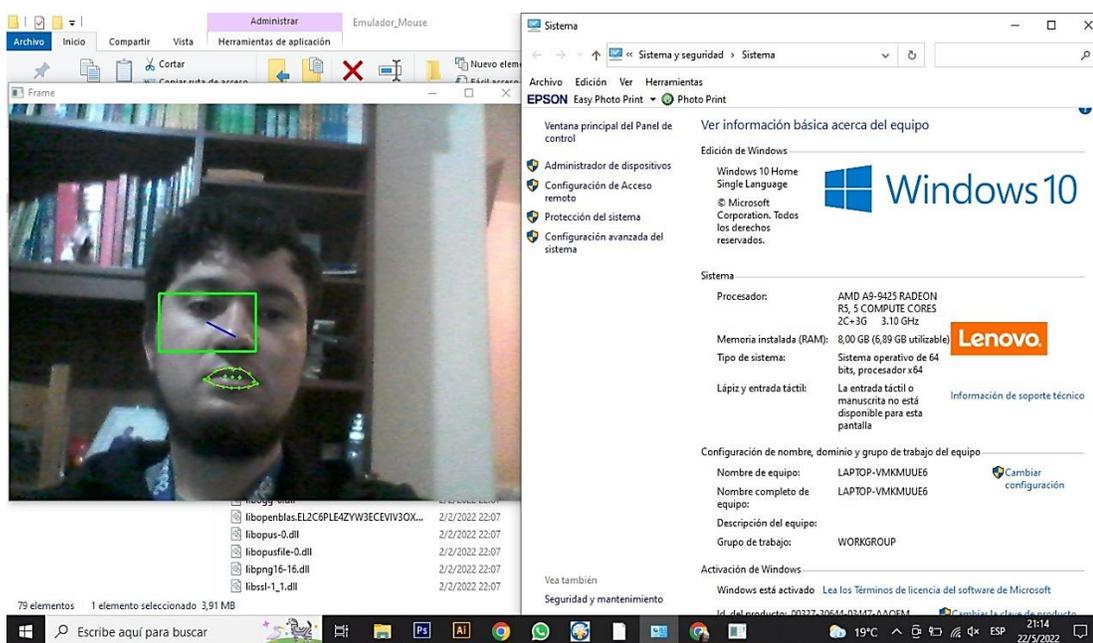
De manera general se pudo observar que el emulador funcionó con normalidad en este computador de escritorio, y no se presentó ninguna novedad en cuanto al desempeño del programa bajo las características de software y hardware utilizadas.

Pruebas en computador Lenovo 81D6

En este caso se utilizó la cámara integrada del computador y un micrófono externo. Las pruebas fueron realizadas en un ambiente silencioso e iluminado para evitar inconvenientes con el funcionamiento del emulador. Se realizó la consulta de las características básicas del computador (Figura 26), y en la ejecución se pudo notar que el movimiento del cursor era más lento de lo esperado.

Figura 26

Detalles básicos del computador Lenovo 81D6 consultados usando el emulador

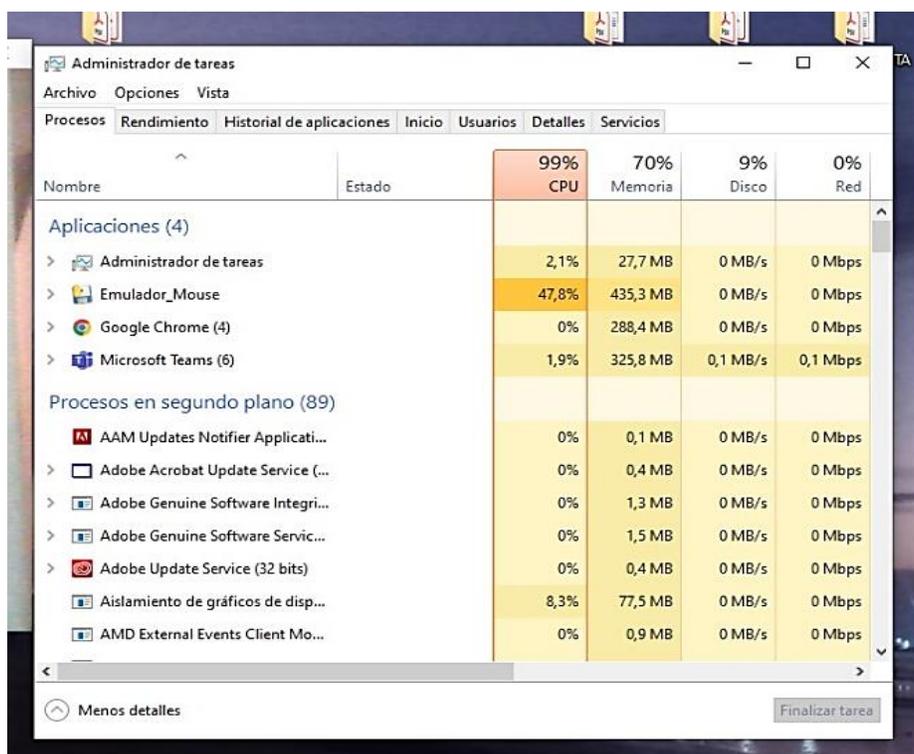


Cuando se verificó el costo computacional del emulador (Figura 27), se observó que la computadora estaba saturada, lo que generaba que el emulador realice el análisis de cada *frame* de forma más lenta. A la hora de usar los comandos de voz, estos se demoraban en ejecutarse más de lo normal, sin embargo, se pudo realizar todas las acciones y se verificó que todos los comandos funcionen correctamente. La principal novedad al realizar esta prueba fue la saturación del computador, llegando a un 99% de uso del CPU, y el emulador ocupando un considerable 47.8% de la capacidad. También se puede

notar que el computador tiene un uso excesivo de procesos en segundo plano, los cuales ocupan aproximadamente el otro 50% de la capacidad del CPU. A partir de esto, se puede considerar que el desempeño del emulador fue malo debido al estado saturado del computador.

Figura 27

Costo computacional del emulador en Lenovo 81D6

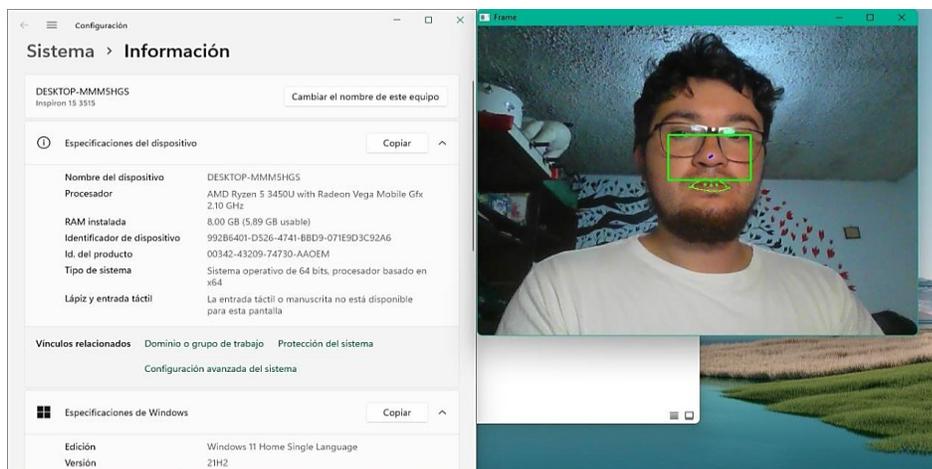


Pruebas en computador Dell Inspiron 15 3515

Para probar el emulador se utilizó el micrófono y la cámara interna del computador, ya que su funcionamiento fue suficiente y no se necesitó mejoras la calidad del video o del audio captado. En este caso las pruebas se realizaron en un ambiente moderadamente iluminado y silencioso. Se realizó la misma prueba en donde se buscó la información del sistema usando el emulador (ver Figura 28), y a breves rasgos se pudo notar que el sistema funcionó de buena manera.

Figura 28

Detalles básicos del computador Dell Inspiron 15 3515 consultados usando el emulador



En la Figura 29 se puede observar que el costo computacional del emulador, en este computador fue de 24,2% de la capacidad del CPU, y 474,5MB de las 8GB disponibles de memoria RAM. Esto indica que su uso no afectará la interacción con otras aplicaciones siempre y cuando no se utilicen demasiadas de forma simultánea.

Figura 29

Costo computacional del emulador en Dell Inspiron 15 3515

Nombre	Estado	54% CPU	89% Memoria	3% Disco	0% Red
Aplicaciones (9)					
Administrador de tareas		0,5%	26,9 MB	0 MB/s	0 Mbps
Configuración		0%	38,5 MB	0 MB/s	0 Mbps
Emulador_Mouse		24,2%	474,5 MB	0 MB/s	0 Mbps
Explorador de Windows (2)		0,2%	87,6 MB	0 MB/s	0 Mbps
Google Chrome (19)		4,2%	690,4 MB	6,4 MB/s	0,1 Mbps
Información del sistema		0%	2,2 MB	0 MB/s	0 Mbps
Microsoft Teams (9)		0%	222,6 MB	0 MB/s	0 Mbps
WinRAR archiver		0%	13,3 MB	0 MB/s	0 Mbps
Zotero (32 bits) (2)		0%	152,4 MB	0 MB/s	0 Mbps
Procesos en segundo plano (67)					
Adobe Acrobat Update Service (...)		0%	0,1 MB	0 MB/s	0 Mbps
AggregatorHost		0%	0,4 MB	0 MB/s	0 Mbps
Aislamiento de gráficos de disp...		7,0%	206,4 MB	0 MB/s	0 Mbps
AMD Crash Defender Service		0%	0,1 MB	0 MB/s	0 Mbps
AMD External Events Client Mo...		0%	0,7 MB	0 MB/s	0 Mbps
AMD External Events Service M...		0%	0,1 MB	0 MB/s	0 Mbps
Antimalware Service Executable		14,6%	162,2 MB	2,5 MB/s	0 Mbps

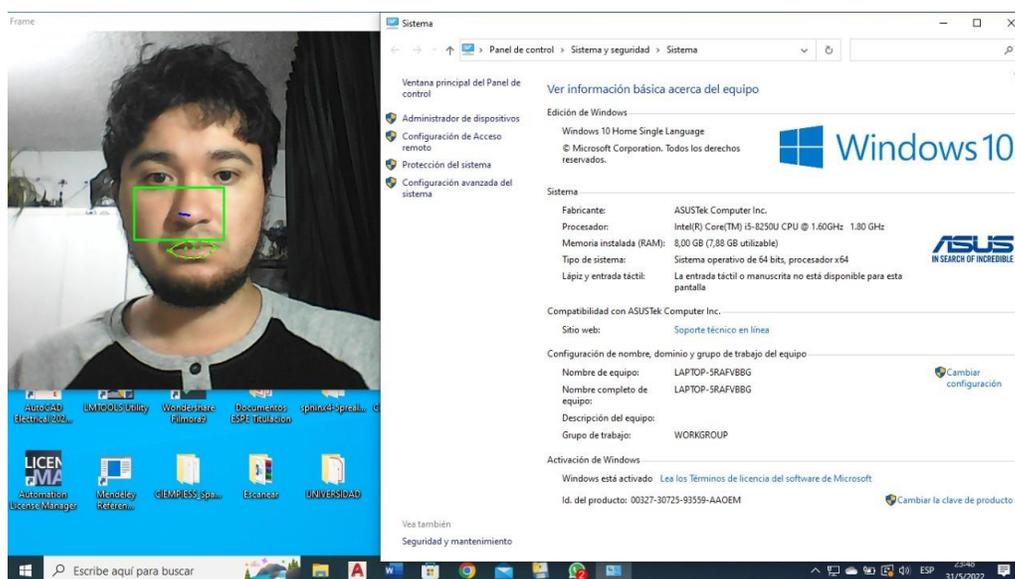
De manera general, el funcionamiento del emulador en este computador fue bueno, la velocidad de movimiento del cursor fue la esperada, y los comandos de voz funcionaban de forma inmediata, siempre y cuando se haya pronunciado las palabras a un volumen adecuado. El único inconveniente que se presentó fue en la ejecución de algunos comandos, específicamente los de “ventanas” y “escritorio”, los cuales no funcionaron debido a que las interfaces y algunos comandos para navegar en Windows 11 son diferentes a sus versiones antiguas.

Pruebas en computador ASUS Vivobook 15

Este fue el computador en donde se desarrolló el software y se realizaron la mayoría de pruebas de funcionamiento generales. Se utilizó la cámara integrada y un micrófono externo para su funcionamiento. Las pruebas se realizaron en un ambiente muy iluminado y silencioso. Se consultó la información básica del sistema (ver Figura 30), usando el emulador y se pudo notar que el funcionamiento fue muy bueno.

Figura 30

Detalles básicos del computador ASUS Vivobook 15 consultados usando el emulador



La velocidad de respuesta a los comandos de voz y la del movimiento del cursor fue rápida y no se notó que el costo computacional fuera un problema. Como se puede ver en la Figura 31, el emulador usa un 22% de los recursos del CPU, y 270MB de las 8GB de memoria RAM disponible, lo que no significa un problema para que pueda ser utilizado con normalidad. Todos los comandos de voz funcionaron perfectamente, y se puede ir evidenciando que en sistemas operativos de Windows 8 y 10, no existen problemas en este aspecto.

Figura 31

Costo computacional del emulador en ASUS Vivobook 15

Nombre	Estado	29%	64%	1%	0%	8
		CPU	Memoria	Disco	Red	Gi
Aplicaciones (6)						
Administrador de tareas		0,1%	29,6 MB	0 MB/s	0 Mbps	
Discord (32 bits) (4)		0,1%	116,9 MB	0 MB/s	0 Mbps	
Emulador_Mouse		22,0%	270,0 MB	0 MB/s	0 Mbps	
Explorador de Windows		0,1%	46,9 MB	0 MB/s	0 Mbps	
Microsoft Word		0%	105,6 MB	0 MB/s	0 Mbps	
WhatsApp (4)		0,1%	255,5 MB	0,1 MB/s	0 Mbps	
Procesos en segundo plano (1...)						
Activation Licensing Service (32 ...)		0%	1,0 MB	0 MB/s	0 Mbps	
Adobe Acrobat Update Service (...)		0%	0,1 MB	0 MB/s	0 Mbps	
Aislamiento de gráficos de disp...		1,1%	91,6 MB	0 MB/s	0 Mbps	
ALMPanelPlugin ActiveX comp...		0%	0,4 MB	0 MB/s	0 Mbps	
Antimalware Service Executable		1,0%	206,7 MB	0 MB/s	0 Mbps	
Aplicación de subsistema de cola		0%	3,8 MB	0 MB/s	0 Mbps	
Application Frame Host		0%	4,3 MB	0 MB/s	0 Mbps	
Application Web Server Daemo...		0%	0,2 MB	0 MB/s	0 Mbps	
ASLDR Service (32 bits)		0%	0,1 MB	0 MB/s	0 Mbps	
ASUS Battery Health Charging		0%	0,2 MB	0 MB/s	0 Mbps	

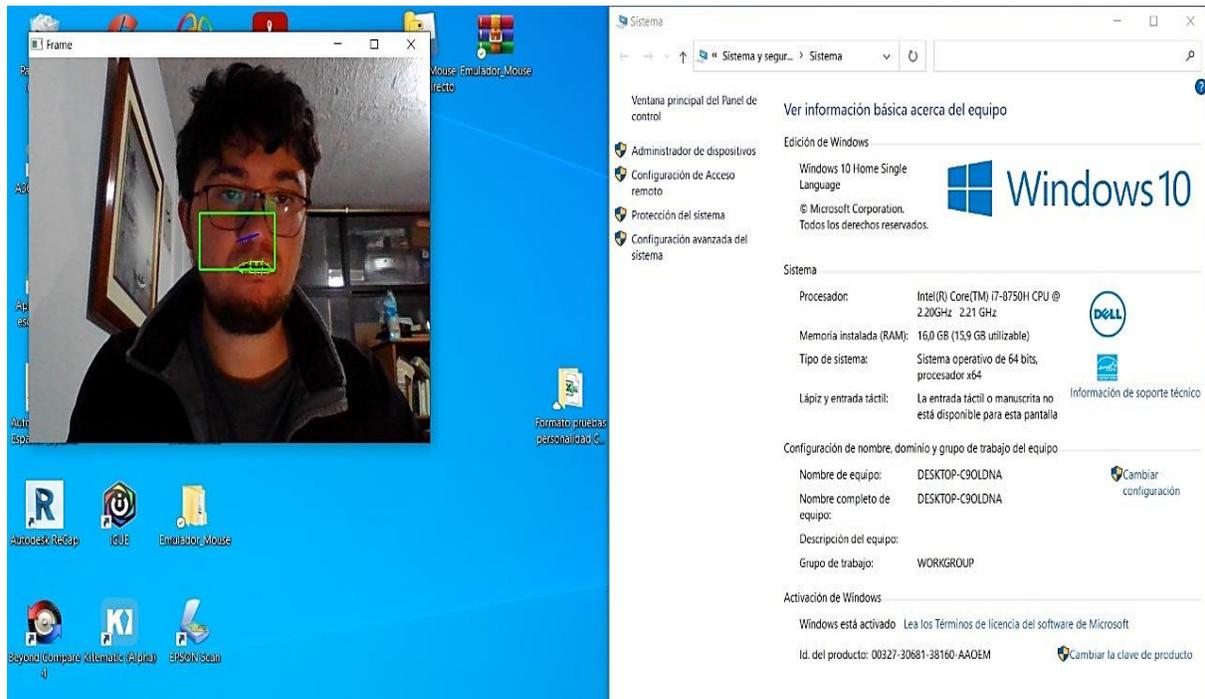
Pruebas en computador Dell G3 3579

Esta es la computadora con mejores características en la que se puso a prueba el emulador. Se utilizó su cámara y micrófono integrados, con los cuales se obtuvo buenas imágenes del rostro y se pudo

captar con precisión las palabras dichas por el usuario. A la hora de realizar la consulta de las características básicas del sistema (Figura 32), se notó que el desempeño del sistema fue muy bueno y rápido.

Figura 32

Detalles básicos del computador Dell G3 3579 consultados usando el emulador



En la Figura 33 se puede observar que el costo computacional del emulador representa el 15.4% de las capacidades del CPU, y apenas 300.2MB de las 16GB de memoria RAM disponibles. A partir de estos resultados se puede evidenciar que el uso de este emulador en una computadora con características de gama alta, no representa dificultad alguna. En cuanto a la prueba de los comandos de voz, todos funcionaron sin excepción, enfatizando que el sistema operativo es Windows 10. No existió novedad alguna al desarrollar las pruebas en este sistema.

Figura 33

Costo computacional del emulador en Dell G3 3579

Nombre	Estado	24% CPU	45% Memoria	1% Disco	0% Red	8% GPU	Motor d
Aplicaciones (3)							
> Administrador de tareas		0,2%	32,9 MB	0 MB/s	0 Mbps	0%	
> Emulador_Mouse		15,4%	300,2 MB	0 MB/s	0 Mbps	5,0%	GPU i
> Google Chrome (10)		0,7%	498,5 MB	0,1 MB/s	0,1 Mbps	0%	GPU i
Procesos en segundo plano (1...)							
> AcrobatNotificationClient (32 bi...	⊕	0%	3,8 MB	0 MB/s	0 Mbps	0%	
> Activation Licensing Service (32 ...		0%	0,7 MB	0 MB/s	0 Mbps	0%	
> Adaptador de rendimiento inve...		0,1%	1,8 MB	0 MB/s	0 Mbps	0%	
> Adobe Acrobat Update Service ...		0%	0,1 MB	0 MB/s	0 Mbps	0%	
Adobe Collaboration Synchroni...		0%	2,0 MB	0 MB/s	0 Mbps	0%	
Adobe Collaboration Synchroni...		0%	0,7 MB	0 MB/s	0 Mbps	0%	
Aislamiento de gráficos de disp...		2,6%	33,6 MB	0 MB/s	0 Mbps	0%	

Pruebas en computador Toshiba Satellite P55W-C

Esta es una computadora de gama alta en la cual se utilizó su cámara y micrófono integrados ya que estos fueron suficiente para que el emulador funcione correctamente. Al momento de realizar la prueba, se obtuvieron buenos resultados y no surgió ninguna novedad. En la Figura 34 se puede observar las características generales del computador y en la Figura 35 el costo computacional del programa mientras se está usando el emulador. El porcentaje de uso del CPU estuvo muy variable y en el momento en el que se realizó la captura estuvo en 38.3%, sin embargo, el funcionamiento del emulador fue muy bueno en términos generales. La respuesta a los comandos de voz y el movimiento del cursor, tuvieron velocidades adecuadas. En cuanto a la prueba de los comandos de voz, todos funcionaron adecuadamente y no existió novedad alguna.

Figura 34

Detalles básicos del computador Toshiba Satellite P55W-C consultados usando el emulador

Especificaciones del dispositivo

Nombre del dispositivo	DESKTOP-6NNQSKP
Procesador	Intel(R) Core(TM) i7-6500U CPU @ 2.50GHz 2.60 GHz
RAM instalada	12,0 GB (11,9 GB usable)
Identificador de dispositivo	0EE1CEA7-2F27-481F-B2A4-F1B8075419B1
Id. del producto	00325-95872-43030-AAOEM
Tipo de sistema	Sistema operativo de 64 bits, procesador basado en x64
Lápiz y entrada táctil	Compatibilidad de la función táctil con 10 puntos táctiles

Copiar

Cambiar el nombre de este equipo

Especificaciones de Windows

Edición	Windows 10 Home
Versión	21H2
Instalado el	14/5/2021
Compilación del sistema operativo	19044.1766
Experiencia	Windows Feature Experience Pack 120.2212.4180.0

Figura 35

Costo computacional del emulador en Toshiba Satellite P55W-C

Administrador de tareas

Archivo Opciones Vista

Procesos Rendimiento Historial de aplicaciones Inicio Usuarios Detalles Servicios

Nombre	Estado	60% CPU	75% Memoria	0% Disco	0% Red
> Google Chrome (10)		8,6%	1.194,1 MB	0,1 MB/s	0 Mbps
> Firefox (8)		3,7%	732,4 MB	0,1 MB/s	0 Mbps
> Emulador_Mouse.exe		38,3%	405,2 MB	0 MB/s	0 Mbps
Firefox		0,5%	370,1 MB	0 MB/s	0 Mbps
Firefox		0%	195,9 MB	0 MB/s	0 Mbps
Firefox		0,2%	181,2 MB	0 MB/s	0 Mbps
Google Chrome		0%	144,5 MB	0 MB/s	0 Mbps
OpenJDK Platform binary		0%	110,0 MB	0 MB/s	0 Mbps
Google Chrome		0,2%	100,5 MB	0 MB/s	0 Mbps
Google Chrome		0%	92,9 MB	0 MB/s	0 Mbps
Firefox		0%	72,8 MB	0 MB/s	0 Mbps
Administrador de ventanas del ...		1,4%	57,6 MB	0 MB/s	0 Mbps
> Adobe Acrobat DC (2)		0%	50,2 MB	0 MB/s	0 Mbps
Avast Antivirus		0%	47,8 MB	0 MB/s	0 Mbps

Menos detalles Finalizar tarea

Pruebas en computador Toshiba Satellite L 15W-B

Inicialmente para realizar las pruebas se procedió a descargar la carpeta que contenía el archivo ejecutable del emulador. Cuando se intentó abrir el emulador, el sistema no respondió y se generó un error (Figura 36), causado por que el computador no fue capaz de abrir el programa. En la Figura 37 se puede observar que las características del procesador y la memoria RAM del computador son muy básicos, por lo que se puede considerar que estas son las principales causas de que el sistema no haya podido funcionar.

Figura 36

Error generado al intentar usar el emulador en computador Toshiba Satellite L 15W-B

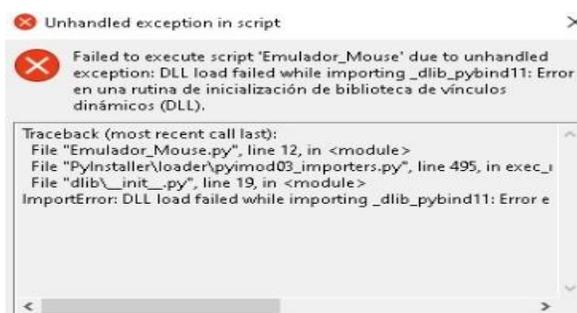


Figura 37

Detalles básicos del computador Toshiba Satellite L 15W-B

Ver información básica acerca del equipo

Edición de Windows

Windows 10 Home

© Microsoft Corporation. Todos los derechos reservados.

Sistema

Fabricante:	TOSHIBA
Modelo:	Satellite L15W-B
Procesador:	Intel(R) Pentium(R) CPU N3540 @ 2.16GHz 2.16 GHz
Memoria instalada (RAM):	4,00 GB (3,88 GB utilizable)
Tipo de sistema:	Sistema operativo de 64 bits, procesador x64
Lápiz y entrada táctil:	Compatibilidad de la función táctil con 10 puntos táctiles

Compatibilidad con TOSHIBA

Sitio web: [Soporte técnico en línea](#)

Configuración de nombre, dominio y grupo de trabajo del equipo

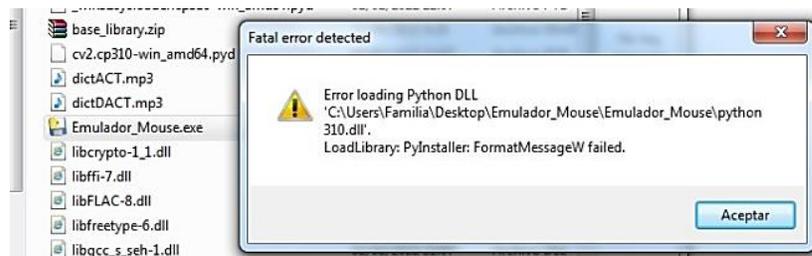
Nombre de equipo:	Bernardo
Nombre completo de equipo:	Bernardo
Descripción del equipo:	
Grupo de trabajo:	WORKGROUP

Pruebas en computador ASUS K53E

En este computador se intentó hacer la prueba de funcionamiento que se realizó en todos los otros dispositivos, sin embargo, al momento de intentar abrir el archivo ejecutable, se generó el error de la Figura 38, y no se pudo continuar con el procedimiento. Cabe recalcar que en la computadora ya se había instalado Python como intento de solución a este problema, sin embargo, no hubo resultados positivos.

Figura 38

Error generado al intentar usar el emulador en computador ASUS K53E



Como se puede observar en la Figura 39, las características demuestran un computador de gama alta antiguo, sin embargo, el sistema operativo es Windows 7 por lo que se puede considerar que, en este entorno, el emulador no podrá funcionar.

Figura 39

Detalles básicos del computador ASUS K53E

Ver información básica acerca del equipo

Edición de Windows

Windows 7 Ultimate
 Copyright © 2009 Microsoft Corporation. Reservados todos los derechos.
 Service Pack 1

Sistema

Fabricante:	Asus
Modelo:	k53
Evaluación:	Evaluación de la experiencia en Windows
Procesador:	Intel(R) Core(TM) i7-2670QM CPU @ 2.20GHz 2.20 GHz
Memoria instalada (RAM):	8,00 GB (6,79 GB utilizable)
Tipo de sistema:	Sistema operativo de 64 bits

Pruebas en computador HP – HP Laptop 15-db1xxx

Esta es una computadora de gama baja en la cual para realizar las pruebas se utilizó su cámara y micrófono integrados. Se obtuvieron buenos resultados y no surgió ninguna novedad. En la Figura 40 se puede observar las características generales del computador y en la Figura 41, el costo computacional mientras se está usando el emulador. El porcentaje de uso del CPU por parte del emulador, fue de 24%, esto se evidenció en el funcionamiento, ya que fue muy bueno en términos generales. La respuesta a los comandos de voz y el movimiento del cursor, tuvieron velocidades adecuadas. En cuanto a la prueba de los comandos de voz, todos funcionaron adecuadamente y no existió novedad alguna.

Figura 40

Detalles básicos del computador HP – HP Laptop 15-db1xxx consultados usando el emulador

Elemento	Valor
Nombre del SO	Microsoft Windows 10 Home Single Language
Versión	10.0.19044 Compilación 19044
Descripción adicional del SO	No disponible
Fabricante del SO	Microsoft Corporation
Nombre del sistema	LAPTOP-HVBQLV7U
Fabricante del sistema	HP
Modelo del sistema	HP Laptop 15-db1xxx
Tipo de sistema	x64-based PC
SKU del sistema	4PF46LA#ABM
Procesador	AMD Ryzen 3 3200U with Radeon Vega Mobile Gfx, 2600 Mhz, 2 procesadore...
Versión y fecha de BIOS	Insyde F.31, 1/7/2020
Versión de SMBIOS	3.1
Versión de controladora integr...	51.24
Modo de BIOS	UEFI
Fabricante de la placa base	HP
Producto placa base	85EA
Versión de la placa base	51.24
Rol de plataforma	Móvil
Estado de arranque seguro	Activado
Configuración de PCR7	Se necesita elevación de privilegios para ver
Directorio de Windows	C:\WINDOWS
Directorio del sistema	C:\WINDOWS\system32
Dispositivo de arranque	\Device\HarddiskVolume1
Configuración regional	México
Capa de abstracción de hardw...	Versión = "10.0.19041.1741"
Nombre de usuario	LAPTOP-HVBQLV7U\varg
Zona horaria	Hora est. Pacífico, Sudamérica
Memoria física instalada (RAM)	8,00 GB

Figura 41

Costo computacional del emulador en HP – HP Laptop 15-db1xxx

The screenshot shows the Windows Task Manager Performance tab. At the top, it displays overall system usage: CPU 48%, Memory 79%, Disk 18%, and Network 0%. Below this is a table listing running processes and their resource consumption.

Nombre	Estado	CPU	Memoria	Disco	Red
> Firefox (11)		7,9%	569,0 MB	5,3 MB/s	0 Mbps
> Microsoft Teams (7)		0%	393,5 MB	0,1 MB/s	0 Mbps
> Emulador_Mouse		24,0%	252,3 MB	0 MB/s	0 Mbps
> Antimalware Service Executable		0,5%	143,6 MB	0 MB/s	0 Mbps
Microsoft Teams		0%	89,5 MB	0 MB/s	0 Mbps
> Host de servicio: SysMain		0,1%	86,3 MB	0 MB/s	0 Mbps
Aislamiento de gráficos de disp...		4,1%	50,7 MB	0 MB/s	0 Mbps
> Explorador de Windows		0,5%	46,5 MB	0 MB/s	0 Mbps
Microsoft Teams		0%	37,8 MB	0 MB/s	0 Mbps
> Host de servicio: Servicio Frame...		1,4%	29,0 MB	0 MB/s	0 Mbps
> Administrador de tareas		1,1%	25,1 MB	0,1 MB/s	0 Mbps
Administrador de ventanas de e...		2,1%	21,2 MB	0 MB/s	0 Mbps
McAfee Module Core Service		0%	20,2 MB	0 MB/s	0 Mbps
> Inicio		0%	17,1 MB	0 MB/s	0 Mbps

Pruebas en computador Toshiba Satellite P75-A7200

Esta es una computadora de gama alta que dispone de un buen micrófono y cámara, por lo que no fue necesario el uso de dispositivos externos para realizar la prueba. En términos generales, se obtuvieron buenos resultados y no surgió novedad alguna. En la Figura 42 se muestran las características generales del computador y en la Figura 43, el costo computacional mientras se está usando el emulador. El porcentaje de uso del CPU mientras se está ejecutando el programa es de 13.3% lo cual demuestra que el computador no presenta problema alguno cuando se usa el emulador. La respuesta a los comandos de voz fue rápida y el movimiento del cursor tuvo la velocidad esperada. Se realizó la verificación de los comandos de voz, y se evidenció que todos funcionaron de manera adecuada.

Figura 42

Detalles básicos del computador Toshiba Satellite P75-A7200 consultados usando el emulador

View basic information about your computer

Windows edition

Windows 8.1

© 2013 Microsoft Corporation. All rights reserved.

[Get more features with a new edition of Windows](#)

System

Processor: Intel(R) Core(TM) i7-4700MQ CPU @ 2.40GHz 2.40 GHz

Installed memory (RAM): 8.00 GB (7.76 GB usable)

System type: 64-bit Operating System, x64-based processor

Pen and Touch: No Pen or Touch Input is available for this Display

Computer name, domain, and workgroup settings

Computer name: Matias

Full computer name: Matias

Computer description:

Workgroup: WORKGROUP

Figura 43

Costo computacional del emulador en Toshiba Satellite P75-A7200

File Options View						
Processes Performance App history Startup Users Details Services						
Name	Status	14% CPU	30% Memory	0% Disk	0% Network	
System		0.2%	0.3 MB	0.1 MB/s	0 Mbps	^
Service Host: Local Service (Peer...		0%	5.3 MB	0.1 MB/s	0 Mbps	
Service Host: Local Service (Net...		0%	16.2 MB	0.1 MB/s	0 Mbps	
Kaspersky Lab launcher		0%	35.6 MB	0 MB/s	0 Mbps	
Kaspersky Anti-Virus (32 bit)		0%	23.6 MB	0 MB/s	0 Mbps	
Google Chrome		0%	55.5 MB	0 MB/s	0 Mbps	
Desktop Window Manager		0%	16.5 MB	0 MB/s	0 Mbps	
Emulador_Mouse		13.3%	234.4 MB	0 MB/s	0 Mbps	
Service Host: Local System (Net...		0%	102.8 MB	0 MB/s	0 Mbps	
Windows Audio Device Graph Is...		0.9%	18.4 MB	0 MB/s	0 Mbps	
Service Host: Local Service (8)		0%	14.0 MB	0 MB/s	0 Mbps	
Windows Explorer		0%	25.6 MB	0 MB/s	0 Mbps	
Services and Controller app		0%	3.1 MB	0 MB/s	0 Mbps	
Kaspersky Secure Connection (3...		0%	1.3 MB	0 MB/s	0 Mbps	
Service Host: Windows Error Re...		0%	0.4 MB	0 MB/s	0 Mbps	v

Fewer details End task

En la Tabla 4, se presentará un resumen de los resultados obtenidos en las pruebas de desempeño realizadas en los diferentes computadores.

Tabla 4

Resultados de pruebas en diferentes computadores

Máquina	Gama	Procesador	RAM	Sistema operativo	Costo Computacional	Velocidad del cursor esperada	Respuesta a comandos de voz	Funcionan Todos los comandos	Cámara externa	Micrófono externo	Novedad
Biostar H61MHB	baja	Intel Core i3-3220	6GB	Windows 8	23%	si	muy buena	si	si	si	ninguna
Lenovo 81D6	baja	AMD A9-9425	8GB	Windows 10	47.8%	no	regular	si	no	si	Computador saturado
HP – HP Laptop 15-db1xxx	baja	AMD Ryzen 3 3200U	8GB	Windows 10	24%	si	buena	si	no	no	ninguna
Toshiba Satellite L 15W-B	baja	Intel Pentium N3540	4GB	Windows 10	-	-	-	-	-	-	El emulador no se pudo ejecutar.
Dell Inspiron 15 3515	media	AMD Ryzen 5 3450U	8GB	Windows 11	24.2%	si	muy buena	no	no	si	ninguna
ASUS Vivobook 15	media	Intel Core i5-8250	8GB	Windows 10	22%	si	muy buena	si	no	si	ninguna
Dell G3 3579	alta	Intel Core i7-8750H	16GB	Windows 10	15.4%	si	muy buena	si	no	no	ninguna
Toshiba Satellite P55W-C	alta	Intel Core i7-6500U	12GB	Windows 10	32.3%	si	muy buena	si	no	no	ninguna
Toshiba Satellite P75-A7200	alta	Intel Core i7-4700	8GB	Windows 8	13.8%	si	Muy buena	si	no	no	ninguna
ASUS K53E	alta	Intel Core i7-2670	8GB	Windows 7	-	-	-	-	-	-	El emulador no se pudo ejecutar.

Analizando los resultados presentados en la Tabla 4, se puede evidenciar que el emulador funciona de buena manera en varios computadores, ya sean de gama baja, media o alta. Si bien existen más problemas en los dispositivos de gama baja, existen casos como el del computador Biostar H61MHB, en el cual se obtuvo buenos resultados a pesar de no ser de alta calidad. En el caso del computador Lenovo

81D6, surgieron algunos problemas que se debieron principalmente a que el computador estaba muy saturado de aplicaciones ejecutándose en segundo plano, sin embargo, a pesar de la lentitud del movimiento del cursor y la respuesta a comandos de voz, el programa se pudo ejecutar en el computador sin generar errores.

Ahora, enfocándose en los dispositivos en donde el emulador no pudo ejecutarse, se puede establecer parámetros mínimos de funcionamiento, los cuales estarían basados específicamente en la calidad del procesador y en la versión de Windows que se utilice. Dicho esto, se puede decir que la versión de Windows mínima para que el sistema funcione es Windows 8, y el procesador del computador debe ser superior o igual a un Intel Core i3 o un AMD A9.

En cuanto al costo computacional, se puede evidenciar que en computadoras bien mantenidas se ocupa un 21.2% de los recursos del CPU, lo que no representa ningún inconveniente a la hora de usar el computador.

También se pudo analizar que el funcionamiento correcto de todos los comandos de voz dependerá de la versión de Windows que se esté ocupando, ya que en versiones de Windows de entre 8 y 10, funcionaron todos sin problema, pero en la versión más actual que es Windows 11, no todos los comandos funcionaron.

Se puede evidenciar que el uso de dispositivos externos está ligado a la gama del computador en el que se use el emulador, en los dispositivos de gama media y baja en los que se realizaron las pruebas se necesitó como mínimo que uno de los dos componentes sea externo. Para el caso del dispositivo de gama alta, no fue necesario su uso, ya que la cámara y el micrófono incorporados internamente son de buena calidad.

Pruebas en Linux

Para este apartado se procedió a instalar una máquina virtual en el computador *Dell G3 3579*, el cual presenta las mejores características de hardware a disposición. A la máquina virtual se la configuró con una memoria RAM de 6GB, un disco duro de 52GB y se instaló el sistema operativo Linux Mint 20.3. Las características generales con las que se trabajó se presentan en la Figura 44.

Figura 44

Características de la máquina virtual

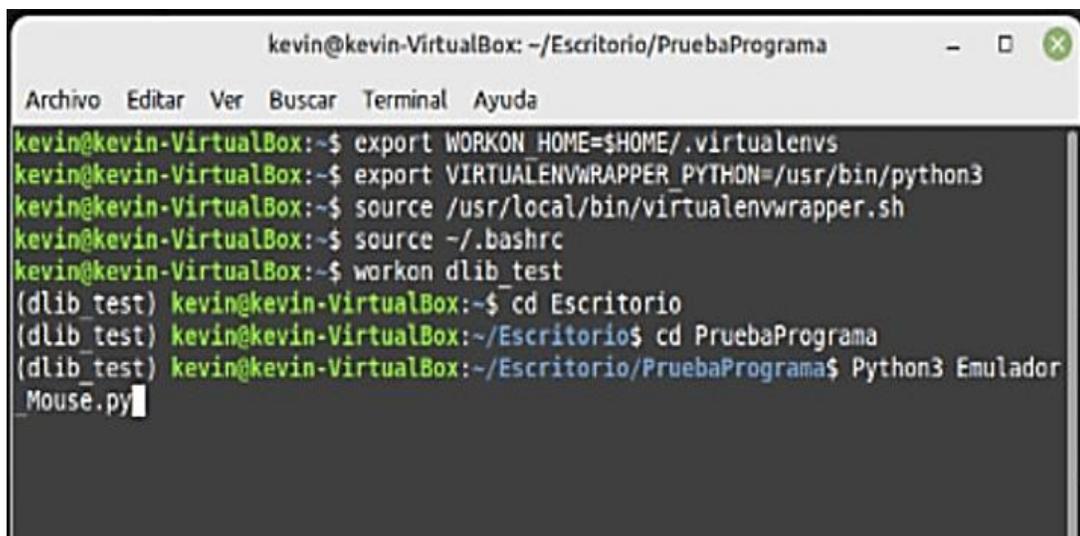
 General	
Nombre:	Linux Mint 20
Sistema operativo:	Ubuntu (64-bit)
 Sistema	
Memoria base:	6108 MB
Orden de arranque:	Disquete, Óptica, Disco duro
Aceleración:	VT-x/AMD-V, Paginación anidada, Paravirtualización KVM
 Pantalla	
Memoria de vídeo:	64 MB
Controlador gráfico:	VMSVGA
Aceleración:	3D
Servidor de escritorio remoto:	Inhabilitado
Grabación:	Inhabilitado
 Almacenamiento	
Controlador:	IDE
IDE secundario maestro:	[Unidad óptica] VBoxGuestAdditions.iso (58,69 MB)
Controlador:	SATA
Puerto SATA 0:	Linux Mint 20.vdi (Normal, 52,62 GB)
 Audio	
Controlador de anfitrión:	Windows DirectSound
Controlador:	ICH AC97
 Red	
Adaptador 1:	Intel PRO/1000 MT Desktop (NAT)
 USB	
Controlador USB:	OHCI, EHCI
Filtros de dispositivos:	0 (0 activo)
 Carpetas compartidas	
Ninguno	
 Descripción	
Ninguno	

Una vez que se instaló el sistema operativo, a través de la consola se procedió a crear un entorno virtual, con la finalidad de poder instalar y administrar las librerías de Python de una manera más sencilla. Cabe recalcar que en Linux Mint, Python ya viene instalado por defecto, por lo que no hace falta realizar

ninguna descarga. Una vez instaladas todas las librerías, se procedió a ejecutar el archivo de Python correspondiente al emulador utilizando los comandos presentados en la Figura 45.

Figura 45

Comandos para ejecutar el Emulador en Linux Mint 20.3



```
kevin@kevin-VirtualBox: ~/Escritorio/PruebaPrograma
Archivo Editar Ver Buscar Terminal Ayuda
kevin@kevin-VirtualBox:~$ export WORKON_HOME=$HOME/.virtualenvs
kevin@kevin-VirtualBox:~$ export VIRTUALENVWRAPPER_PYTHON=/usr/bin/python3
kevin@kevin-VirtualBox:~$ source /usr/local/bin/virtualenvwrapper.sh
kevin@kevin-VirtualBox:~$ source ~/.bashrc
kevin@kevin-VirtualBox:~$ workon dlib_test
(dlib_test) kevin@kevin-VirtualBox:~$ cd Escritorio
(dlib_test) kevin@kevin-VirtualBox:~/Escritorio$ cd PruebaPrograma
(dlib_test) kevin@kevin-VirtualBox:~/Escritorio/PruebaPrograma$ Python3 Emulador
_Mouse.py
```

El emulador se ejecutó con normalidad en el sistema operativo de Linux (Figura 46). Para verificar que el sistema se encuentre funcionando correctamente, se procedió a activar las funciones de la máquina virtual para acceder a los módulos de entrada/salida de audio y a la cámara del computador. En cuanto al *mouse* externo, una vez que se ejecutó el programa, se procedió a desactivarlo en la máquina virtual para que el emulador pueda funcionar sin ningún inconveniente.

Una vez que se comprobó que el emulador esté funcionando de forma adecuada, se procedió a realizar las pruebas de desempeño. Al abrir el monitor del sistema (Figura 47), se observó que el costo computacional fue de 72% de las capacidades de la máquina virtual. En cuanto a la movilidad del cursor y a la velocidad de detección de los comandos de voz, eran un poco más lenta de lo esperado, sin embargo, se pudo manejar el emulador de una manera aceptable. La disminución de las velocidades de

funcionamiento son producto del alto costo computacional, que se puede justificar por el mismo uso de la máquina virtual y las características de hardware que se le destinó.

Figura 46

Funcionamiento del emulador en Linux Mint 20.3

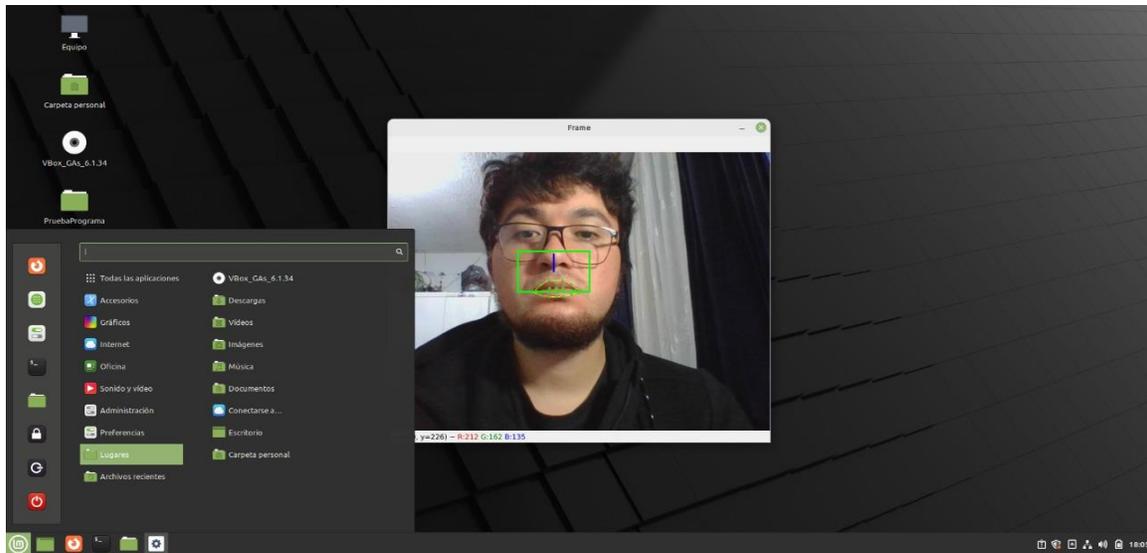
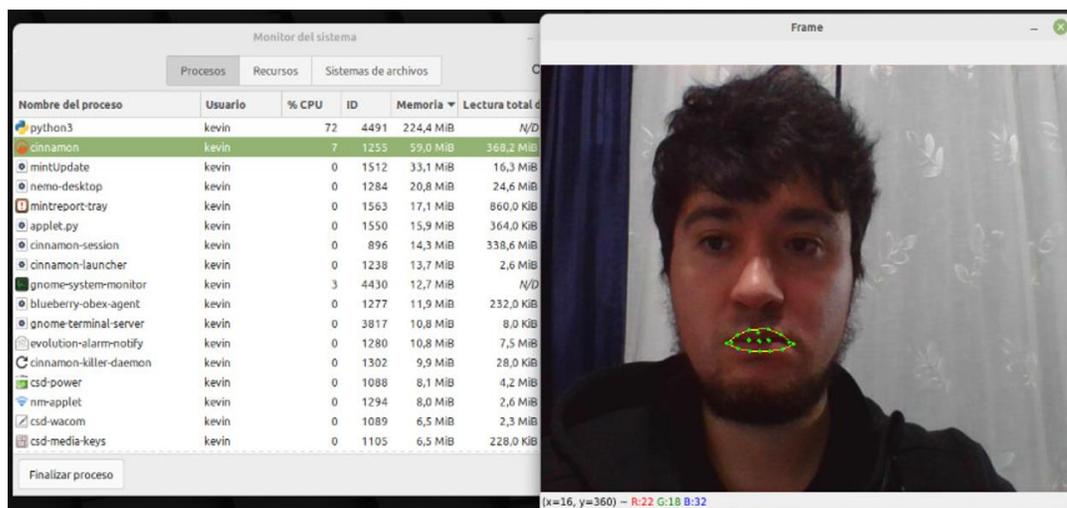


Figura 47

Costo computacional en Máquina Virtual con Linux Mint 20.3



En cuanto al uso de los comandos de voz, la mayoría funcionaron correctamente, con las excepciones de “ventanas” y “escritorio” que fueron desarrollados para funcionar en entornos de Windows. La detección del rostro y el reconocimiento de voz fue buena, debido a que la máquina virtual accedió a los componentes de hardware de la computadora con mejores características.

De forma general, se puede decir que el emulador funciona de manera correcta en sistemas operativos de distribución libre, como en este caso Linux Mint 20.3. Para mejorar la experiencia en estos entornos, se podría implementar comandos específicos que se adapten a las diferentes interfaces.

Pruebas en tarjeta Raspberry Pi 3 B

Con la finalidad de verificar como es el funcionamiento del emulador en sistemas embebidos, se realizaron pruebas de desempeño en una tarjeta Raspberry Pi 3 B con memoria RAM de 1GB.

Inicialmente, se procedió a instalar el sistema operativo *Raspberry Pi OS (32-bits)* más actual, en una tarjeta Micro SD a través del programa *Raspberry Pi Imager*. Una vez finalizada la instalación, se procedió a colocarla dentro del dispositivo y se conectaron todos los periféricos necesarios para hacer funcionar el emulador (ver Figura 48).

Figura 48

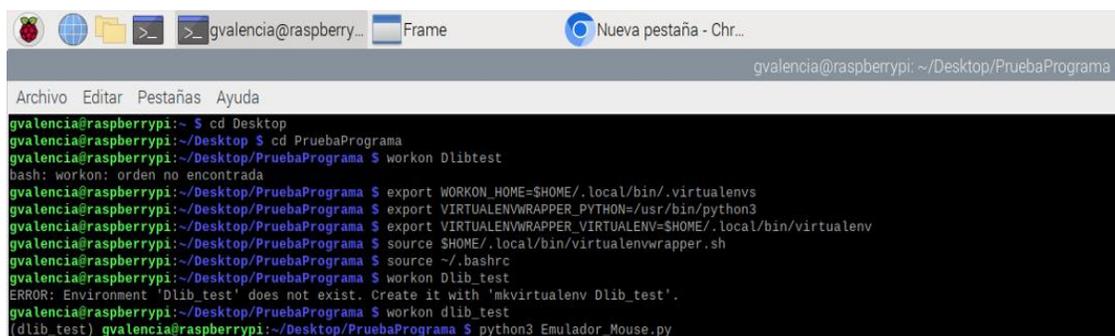
Raspberry Pi 3 B conectada a dispositivos periféricos de entrada.



Debido a que el sistema operativo utilizado en esta tarjeta Raspberry es similar a Linux, el procedimiento para poder ejecutar el emulador fue muy similar. Primero se creó un entorno virtual, el cual facilitó la instalación de todas las librerías utilizadas para desarrollar el proyecto. Una vez realizado esto, se ejecutó el programa a través de la terminal utilizando los comandos mostrados en la Figura 49.

Figura 49

Comandos utilizados para ejecutar el emulador en Raspberry Pi 3 B



```

gvalencia@raspberrypi:~ $ cd Desktop
gvalencia@raspberrypi:~/Desktop $ cd PruebaPrograma
gvalencia@raspberrypi:~/Desktop/PruebaPrograma $ workon Dlibtest
bash: workon: orden no encontrada
gvalencia@raspberrypi:~/Desktop/PruebaPrograma $ export WORKON_HOME=$HOME/.local/bin/.virtualenvs
gvalencia@raspberrypi:~/Desktop/PruebaPrograma $ export VIRTUALENVWRAPPER_PYTHON=/usr/bin/python3
gvalencia@raspberrypi:~/Desktop/PruebaPrograma $ export VIRTUALENVWRAPPER_VIRTUALENV=$HOME/.local/bin/virtualenv
gvalencia@raspberrypi:~/Desktop/PruebaPrograma $ source $HOME/.local/bin/virtualenvwrapper.sh
gvalencia@raspberrypi:~/Desktop/PruebaPrograma $ source ~/.bashrc
gvalencia@raspberrypi:~/Desktop/PruebaPrograma $ workon Dlib_test
ERROR: Environment 'Dlib_test' does not exist. Create it with 'mkvirtualenv Dlib_test'.
gvalencia@raspberrypi:~/Desktop/PruebaPrograma $ workon dlib_test
(dlib_test) gvalencia@raspberrypi:~/Desktop/PruebaPrograma $ python3 Emulador_Mouse.py

```

Luego que se ejecutó el programa, se observó que el emulador se abrió sin ningún problema (Figura 50). Al momento de querer iniciar el movimiento del cursor, se notó que el funcionamiento era muy lento, se tuvo que mantener abierta la boca por más tiempo de lo esperado para lograr realizar esta acción. Cuando se activó la función de movimiento, el cursor se movía a muy poca velocidad y resultó incomodo su uso. En cuanto a la detección de los comandos de voz, se utilizó un micrófono que viene integrado en la misma cámara de la Figura 17, la detección fue buena, sin embargo, se demoraba un tiempo considerable en dar una respuesta. Debido a la lentitud con la que se ejecutaba el programa, se procedió a revisar el costo computacional. En la Figura 51 se puede ver que la aplicación ocupa un alto porcentaje de las capacidades del computador (52%) y que gran parte de la memoria RAM disponible, está ocupada. A partir de esto se pudo notar que el funcionamiento del emulador en esta tarjeta fue deficiente debido a sus características limitadas, sin embargo, se podría suponer que, en modelos más actuales que dispongan de mejores características, la velocidad de respuesta podría mejorar.

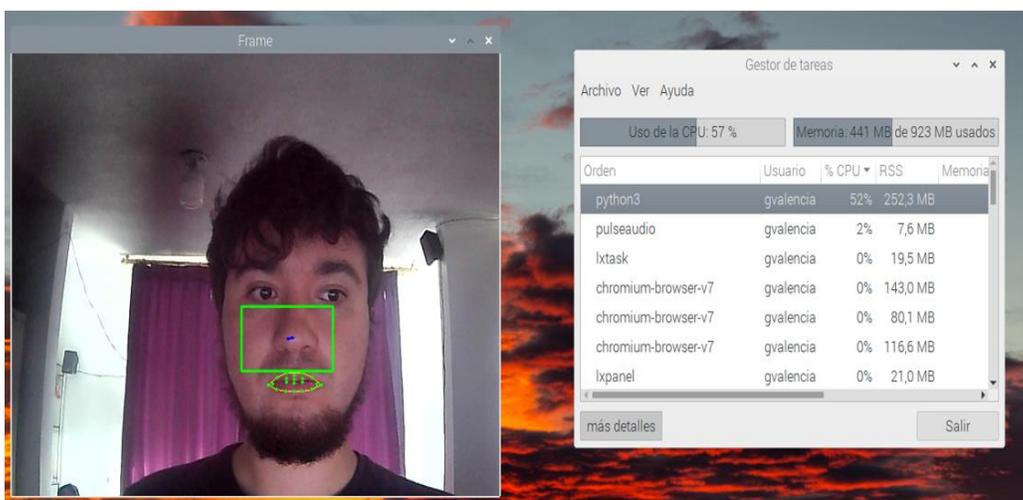
Figura 50

Emulador funcionando en Raspberry Pi 3 B



Figura 51

Costo computacional del emulador en Raspberry Pi 3 B



Pruebas en usuarios

Con la finalidad de verificar como es el funcionamiento del emulador en diferentes usuarios, se realizó un ejercicio a 24 personas de edades entre 22 a 82 años. Para realizar esta prueba inicialmente a cada usuario se le dio un tutorial personal, en donde se les enseñó los comandos de voz, las formas de realizar el movimiento del cursor, la manera correcta de posicionarse frente a la cámara y varias recomendaciones para un correcto funcionamiento. Luego se solicitó al usuario que realice un ejercicio, que consistió en consultar cuál es “la capital del Ecuador” en su navegador de preferencia, desde que se encuentra en la pantalla del escritorio, aplicando todos los gestos faciales y comandos necesarios.

A partir de este ejercicio, se realizó una encuesta con 13 preguntas para conocer la percepción del emulador en diferentes aspectos. Para realizar el análisis de los resultados de estas preguntas, se dividió el total de los encuestados en dos grupos, personas menores de 50 años y mayores a 50 años. El primero consta de 13 encuestados y el segundo de 10. La principal razón para realizar esta división es analizar cómo es la percepción del uso del emulador por gente joven, que desde pequeños han estado en contacto directo con la tecnología, y por gente mayor, los cuales no se criaron en ambientes tan tecnológicamente avanzados como los de ahora. A continuación, se presentarán las preguntas y el análisis de los resultados obtenidos en las encuestas.

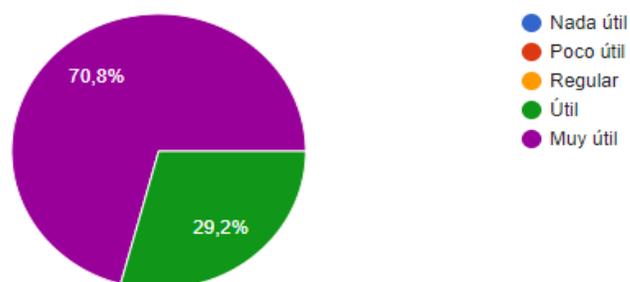
Pregunta 1: ¿Qué tan funcional considera que es el emulador para personas con discapacidades motrices?

Esta pregunta se la realizó con la finalidad de verificar si las personas consideran que el emulador sería de utilidad para personas que tengan discapacidades motrices. En la Figura 52 se puede observar que el 70.8% de los usuarios encuestados consideran que puede ser muy útil y el 29.2% restante, lo considera útil. Estos resultados indican que el emulador puede ser una herramienta funcional que ayude

a las personas con discapacidades motrices a manejar computadoras y, por lo tanto, poder aprovechar de sus funcionalidades.

Figura 52

Resultados Pregunta 1

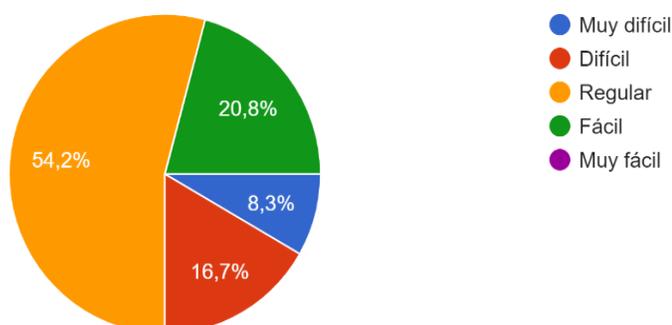


Pregunta 2: ¿Qué tan difícil fue la adaptación al uso del emulador?

Esta pregunta se la realizó con la intención de determinar qué tan difícil es acostumbrarse a usar el emulador, los resultados obtenidos del total de encuestados se los pueden observar en la Figura 53, y se evidencia que la respuesta que predomina es la de regular, con un 54,2%. En cuanto a las respuestas negativas, es decir, difícil y muy difícil suman un 25% que representan a los encuestados a los que les resultó complicado la adaptación. Por último, a un 20.8% le resultó fácil la adaptación.

Figura 53

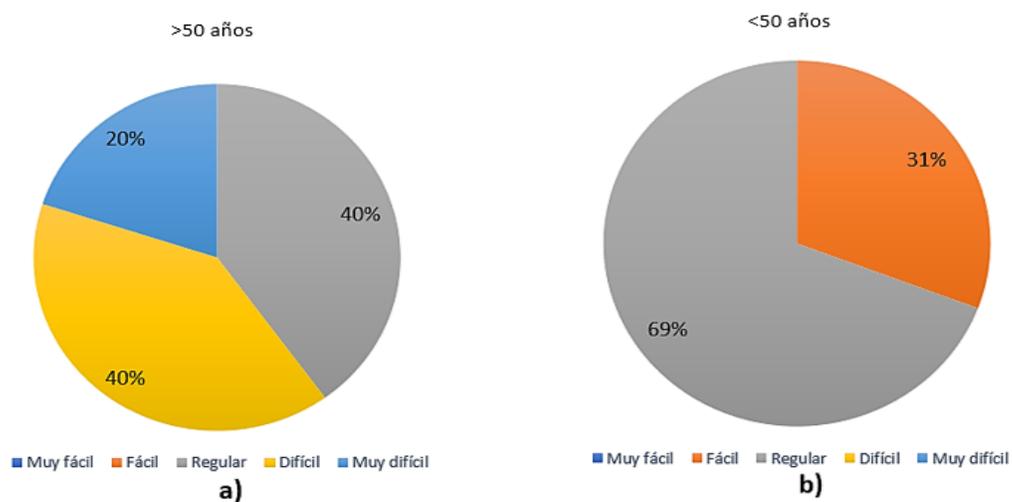
Resultados Pregunta 2



Realizando un análisis de acuerdo a las edades de los usuarios encuestados, se puede sacar conclusiones en cuanto a qué grupo de personas les resultó más difícil la adaptación al uso del emulador. En la Figura 54 se observa que, a las personas con edades mayores a 50 años se les hizo más complicado acostumbrarse a usar el emulador, esto se evidencia en el 60% de personas en este grupo que consideraron que la adaptación es difícil o muy difícil. En el caso de los usuarios menores a 50 años, al 100% de los encuestados no se les complicó este proceso.

Figura 54

Resultados por edades, Pregunta 2



Nota. a) Se analizan los datos únicamente para los usuarios con edades mayores a 50. b) Se analizan los datos únicamente para los usuarios con edades menores a 50 años.

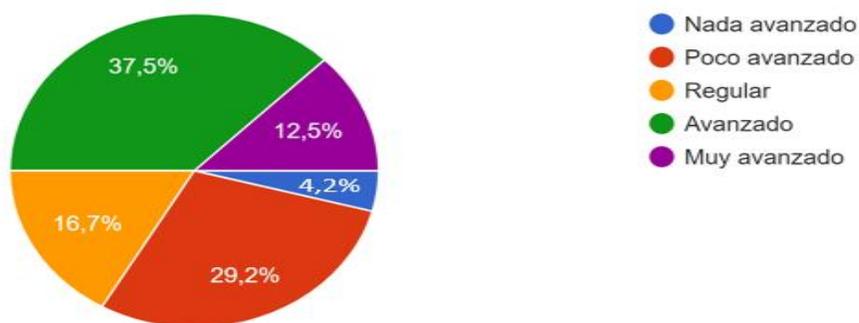
Se puede notar en esta pregunta que el proceso de adaptación de los usuarios depende en parte de su edad, esto puede ser debido a que sus conocimientos en cuanto al manejo de tecnología, no son tan avanzados como los de las personas jóvenes. En la siguiente pregunta se analizará este aspecto.

Pregunta 3: ¿Qué tan avanzados son sus conocimientos en cuanto al manejo del computador?

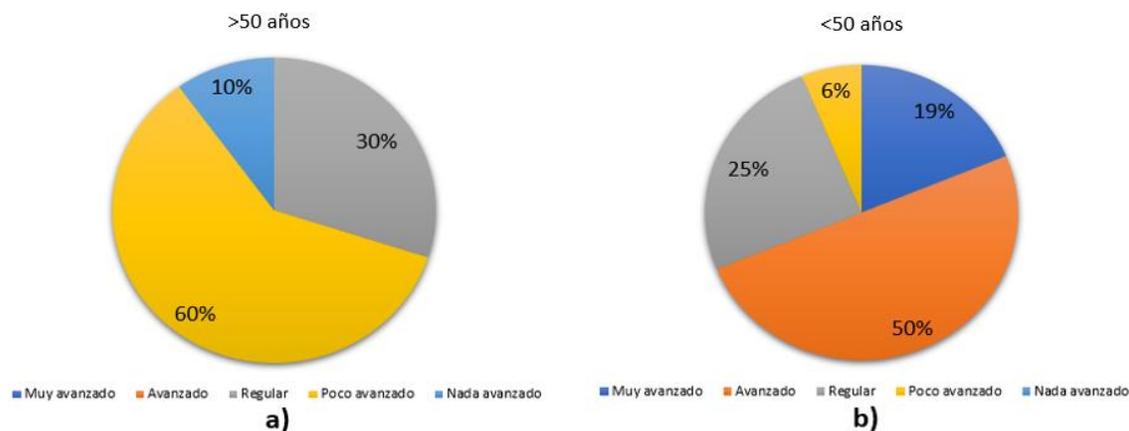
Esta pregunta se realiza con la finalidad de conocer cuál es la habilidad y conocimientos previos del uso de computadores, que cada usuario tiene, en la Figura 55, se muestran los resultados, donde un 66.7% representan usuarios con un mínimo de conocimientos regulares en cuanto al manejo de los equipos, lo que indica que se realizó las pruebas en su mayoría a personas que saben usar el computador.

Figura 55

Resultados Pregunta 3



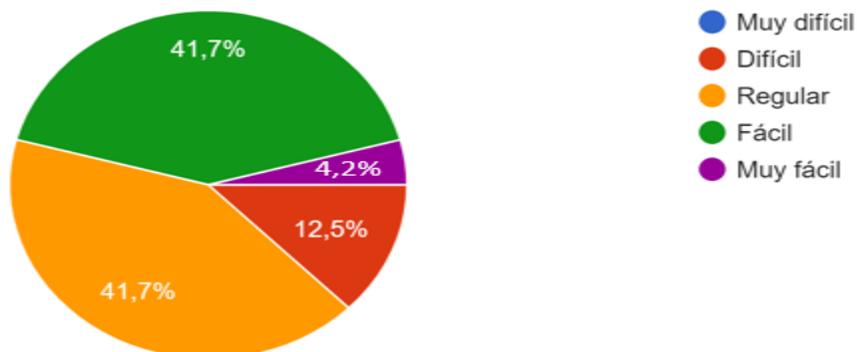
Realizando un análisis por edades de los resultados obtenidos en esta pregunta, en la Figura 56, se puede verificar que el 100% de los encuestados mayores a 50 años tienen conocimientos entre nada avanzados, poco avanzados y regulares. Para el caso de las personas menores a 50 años, el 94% tienen conocimientos entre regulares, avanzados y muy avanzados. A partir de estos resultados se puede decir de manera general que las personas encuestadas con mayor edad tienen menores conocimientos en cuanto al uso del computador. Este aspecto se puede relacionar con la pregunta anterior justificando el hecho de que, a las personas del grupo de mayor edad, les resulta más complicada la adaptación al uso del emulador.

Figura 56*Resultados por edades, Pregunta 3*

Nota. a) Se analizan los datos únicamente para los usuarios con edades mayores a 50. b) Se analizan los datos únicamente para los usuarios con edades menores a 50 años.

Pregunta 4: ¿Qué tan difícil fue la realización del ejercicio?

Con esta pregunta se busca analizar el grado de dificultad que representa usar el emulador en situaciones comunes, como es la realización de una consulta en un buscador de internet. Para este caso el usuario debe dirigir el cursor hacia el ícono de su navegador, hacer doble clic, activar el modo dictado, decir lo que desea buscar, activar el comando “enter”, apagar el modo dictado, mover hacia la primera opción presentada por el navegador y hacer clic. En el ejercicio propuesto se hace uso de la mayoría de comandos básicos del emulador, lo cual permite dar una percepción clara de qué tan difícil es manejar el sistema. En la Figura 57 se muestran los resultados generales de esta pregunta, que indican que a un 87.6% de los usuarios encuestados le resultó entre regular, fácil y muy fácil el uso del emulador a la hora de realizar este ejercicio. Esto indica que una vez que el usuario se ha adaptado al uso del emulador le resultará sencillo utilizarlo.

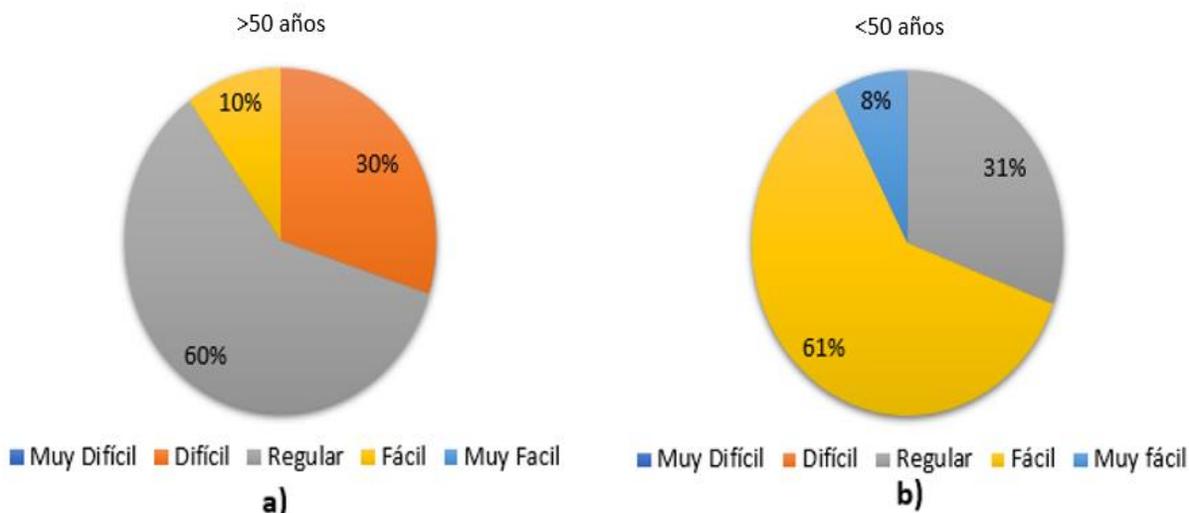
Figura 57*Resultados Pregunta 4*

Esta pregunta permite realizar un análisis por edades, el cual muestra a cuál grupo le resultó más sencillo utilizar el emulador para realizar el ejercicio. En la Figura 58 se puede observar por un lado que, a los usuarios mayores a 50 años en su mayoría (60%) les resultó regular en términos de dificultad la realización del ejercicio, sin embargo, a un 30% se les complicó un poco más y solo a un 10% le resultó fácil. En cuanto al 69% de encuestados menores a 50 años, el manejo del emulador les resultó entre fácil y muy fácil, al 31% restante de este grupo le resultó regular. A partir de estos resultados se puede decir que, los usuarios con mayor edad se les complicó más la realización de este ejercicio, sin embargo, basándose en el análisis de la Figura 57, se puede decir en términos generales que, el uso del emulador no resulta tan complicado ya que la mayoría de los usuarios lo considera entre regular y fácil.

Las siguientes preguntas que se presentarán están enfocadas en analizar ciertas características puntuales del emulador, como lo es la velocidad del movimiento del cursor, el reinicio del movimiento y los comandos de segmentación de pantalla.

Figura 58

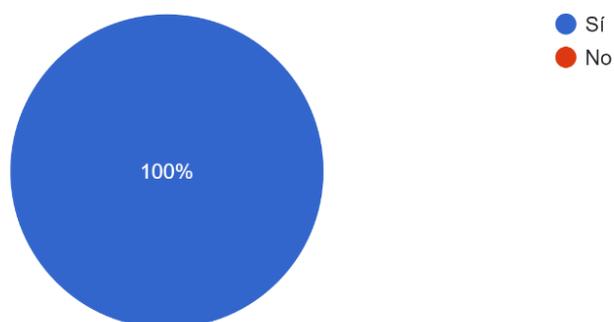
Resultados por edades, Pregunta 4



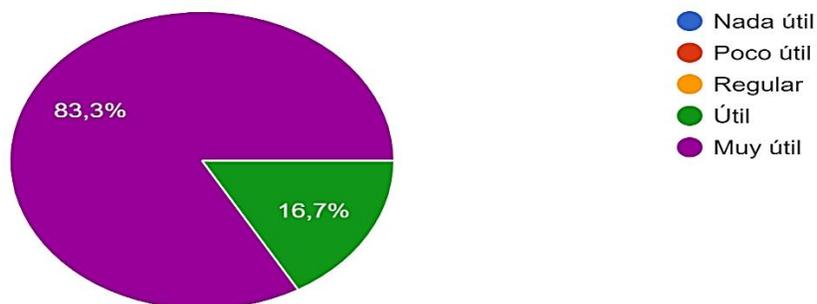
Nota. a) Se analizan los datos únicamente para los usuarios con edades mayores a 50. b) Se analizan los datos únicamente para los usuarios con edades menores a 50 años.

Pregunta 5: ¿Considera que la velocidad de movimiento del cursor es adecuada para facilitar la selección de objetos pequeños?

Esta pregunta cerrada se la realizó con la intención de verificar si el usuario se siente a gusto con la velocidad de movimiento del cursor, tomando en cuenta que si fuera más rápido se complicaría la selección de objetos pequeños en la pantalla. En la Figura 59 se puede observar que el 100% de los usuarios se sintieron a gusto, lo cual se podría justificar debido a que, el uso de los comandos de segmentación de pantalla permite moverse a diferentes zonas de forma rápida, y con la velocidad de movimiento del cursor se puede seleccionar objetos pequeños de forma sencilla.

Figura 59*Resultados Pregunta 5***Pregunta 6: ¿Qué tan útiles le resultaron los comandos de segmentación de pantalla?**

Como se pudo ver en la pregunta anterior, los usuarios se sintieron conformes con la velocidad de movimiento del cursor, y esto es en gran parte a que para realizar movimientos bruscos a través de diferentes zonas se utiliza los comandos de segmentación de pantalla. En la Figura 60 se puede ver respuestas positivas en su totalidad con un 82.6% de personas que consideraron que estos comandos fueron muy útiles, y el 17.4% restante, lo considero útil. Esto indica que el uso de los comandos de voz para moverse a través de la pantalla es una buena forma de realizar movimientos rápidos, y no perder precisión a la hora de querer seleccionar objetos pequeños con el cursor.

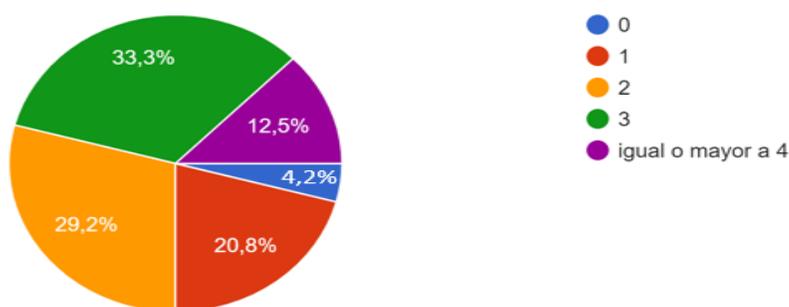
Figura 60*Resultados Pregunta 6*

Pregunta 7: ¿Cuántas veces tuvo que reiniciar el movimiento del cursor (manteniendo abierta la boca) para poder realizar el ejercicio?

El emulador permite al usuario resetear el punto de referencia de la nariz que facilita el movimiento del cursor, esto debido a que comúnmente cuando se está sentado frente a un computador, se realizan movimientos bruscos que pueden generar fallos a la hora de mover el cursor, es decir, se puede mover en direcciones no esperadas. Para dar solución a esto, el usuario puede mantener abierta su boca durante un tiempo y escuchar las señales de activación de movimiento del cursor para que se actualice la referencia. Durante el ejercicio realizado, varias personas tuvieron que recurrir a esta función, en la Figura 61 se muestra los resultados que indican que, la gran mayoría de usuarios tuvo que resetear el movimiento del sistema como mínimo una vez, un 12.5% tuvo que recurrir a esta función igual o más de cuatro veces. La mayoría de las respuestas se centran entre uno, dos y tres intentos, lo cual representan resultados intermedios, ya que lo óptimo es que el uso de esta función sea mínimo. Se considera que estos valores podrían disminuir, conforme el usuario se acostumbre más al uso del emulador.

Figura 61

Resultados pregunta 7



Las preguntas a continuación buscan relacionar las características de hardware usadas por el usuario (micrófono y webcam), con los resultados obtenidos en cuanto a la detección de rostros y de comandos de voz.

Pregunta 8: ¿Qué tan preciso considera que es la detección de los comandos de voz en su computador?

En la Figura 62 se puede observar buenos resultados en cuanto a la detección de los comandos de voz, ya que todos los usuarios consideraron como mínimo que esta fue regular y un 91.7% entre bueno y muy bueno. Esto indica que no existieron muchos inconvenientes en cuanto a la detección de los comandos de voz.

Pregunta 9: ¿Utilizó un micrófono externo para realizar la prueba?

El buen desempeño de los comandos de voz obtenidos del análisis de la anterior pregunta, se puede justificar gracias a los resultados que se observan en la Figura 63, en donde la mayoría de usuarios (62,5%), utilizó un micrófono externo, lo que garantiza que la señal de la voz que analice el programa sea más clara, dando así un mejor desempeño. Esto también se puede justificar, ya que algunos de los usuarios usaron computadoras de escritorio, las cuales necesitan de micrófonos externos para poder usar el emulador. Por otro lado, el 37.5% restante, no usó micrófono externo, y los buenos resultados en cuanto al desempeño de los comandos de voz puede ser, a que la prueba se la realizó en computadores de gama alta o gama media, con muy buenos micrófonos internos.

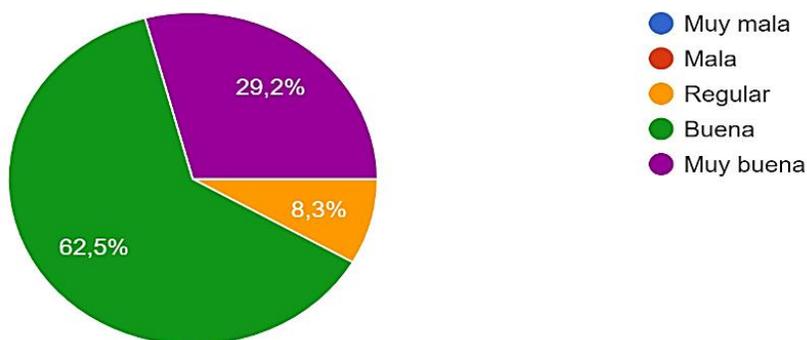
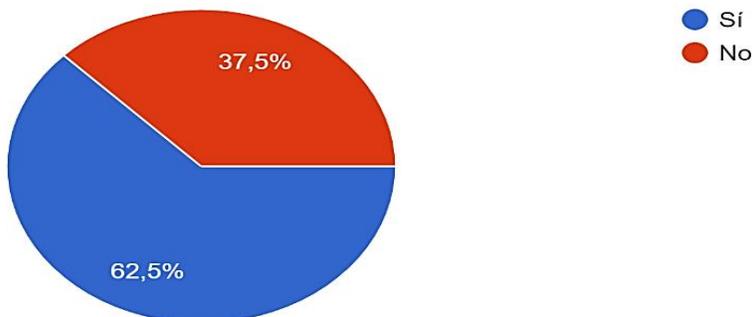
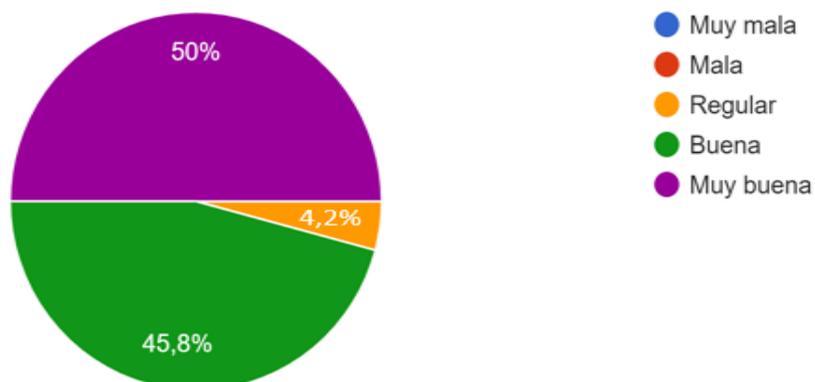
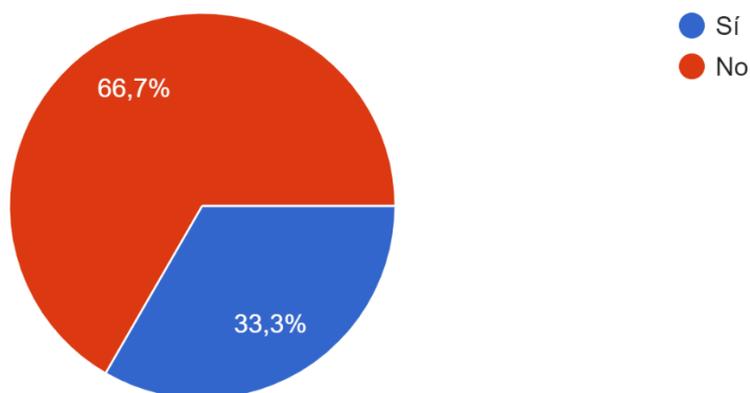
Figura 62*Resultados Pregunta 8*

Figura 63*Resultados Pregunta 9****Pregunta 10: ¿Qué tan preciso considera que fue la detección de su rostro?***

La Figura 64 muestra que un 95.8% de los encuestados consideró que la precisión a la hora de detectar sus rostros fue buena y muy buena, por otro lado, solo un 4,2% la consideró regular. Esto da un buen indicio de que la detección de los rostros no causó problemas.

Pregunta 11: ¿Utilizó una cámara externa para realizar la prueba?

En este caso, observando el diagrama de pastel de la Figura 65, se puede ver que el 66.7% de los usuarios no usaron una cámara externa, lo que puede indicar que las pruebas las realizaron en laptops con estos dispositivos internos. Por lo general la cámara que disponen este tipo de computadoras es básica (VGA), lo que es suficiente para lograr el buen desempeño del emulador en cuanto a la detección de rostros. El 33.3% restante utilizó cámaras externas, ya sea porque trabajan en computadoras de escritorio o las requieren en sus laptops. La calidad de estos dispositivos externos es generalmente mayor a la de los internos, por lo que se justifica los buenos resultados presentados en la pregunta anterior.

Figura 64*Resultados Pregunta 10***Figura 65***Resultados Pregunta 11****Pregunta 12: Cronometrar el tiempo que tarda en realizar el ejercicio***

La siguiente pregunta consistió en cronometrar el tiempo que tardan en realizar el ejercicio propuesto y registrar la respuesta en segundos. Se solicitó que se responda esta pregunta solo cuando el usuario se haya sentido debidamente adaptado al uso del emulador, con la finalidad de obtener los mejores resultados para el análisis. A partir de las respuestas recibidas, se obtuvo un tiempo promedio de

44.04 segundos. Por motivo de comparación, se realizó pruebas del mismo ejercicio, pero usando un mouse físico y tipeando a velocidad moderada las palabras con el teclado. Se cronometró un tiempo promedio aproximado de 12.4 segundos, lo cual resulta ser muy inferior al valor obtenido de los datos de los usuarios del emulador. Si bien es una gran diferencia, se debe considerar que el emulador está enfocado para personas con discapacidades motrices, por lo que no se espera que el desempeño a la hora de manejar el cursor con el emulador, sea mejor que cuando se usa un ratón común.

Pregunta 13: ¿Tuvo dificultades a la hora de realizar la prueba?

En esta pregunta se solicitó a los usuarios que especifiquen las dificultades que tuvieron a la hora de manejar el emulador. Entre algunas de las más comunes estuvo el hecho de tener que memorizarse varios comandos de voz, lo cual efectivamente representa una traba para lograr una correcta adaptación. Otra de las dificultades señaladas fue, la necesidad de hablar muy alto para que se entiendan los comandos de voz, sin embargo, esto depende del hardware que se esté utilizando. En cuanto a la segmentación de pantalla varios usuarios señalaron que, en un principio les fue difícil aprenderse los comandos y las posiciones para cada cuadrante. La posición del rostro también representó una dificultad para algunos usuarios, los cuales tuvieron que resetear varias veces el punto de referencia para que funcionen correctamente los movimientos verticales del cursor. En el modo dictado también tuvieron problemas ya que el sistema confundía algunos comandos, y los pasaba a texto.

Limitaciones del emulador

A partir de los resultados obtenidos en todas las pruebas descritas en este capítulo, se pudo encontrar varias limitaciones que presenta el emulador. Normalmente se buscaría comparar el uso de este sistema con un mouse físico, que sea controlado con la mano. Sin embargo, se debe considerar que este proyecto es realizado con la finalidad de brindar a personas con discapacidades motrices una herramienta que les permita mejora su interacción con la computadora. Se espera que les permita realizar

tareas comunes como: abrir archivos, navegar en internet, realizar consultas, ver videos, revisar correos electrónicos o usar redes sociales. Considerando esto, el uso del emulador no sería viable en aplicaciones que requieran un uso muy fluido del cursor, como puede ser el caso de videojuegos complejos.

En cuanto al modo dictado que se implementó, su uso no sería lo suficientemente fluido para poder redactar con facilidad documentos de texto formales. Esta función está pensada para que el usuario pueda ingresar texto en situaciones simples como, por ejemplo: consultas en el navegador, enviar mensajes en redes sociales, redactar correos electrónicos, entre otros. Estas acciones se las puede realizar solamente en español, debido a que este es el modelo de lenguaje que se utilizó para lograr la detección de la voz.

Como se pudo observar en las pruebas realizadas en distintos computadores, el emulador no funcionó en máquinas con procesadores inferiores a los Intel Core I3, ni tampoco en sistemas operativos antiguos como Windows 7.

Capítulo VI. Conclusiones y recomendaciones

Basándose en los objetivos especificados al inicio de este trabajo, y en los resultados obtenidos en las pruebas del emulador, en este capítulo se presentará diversas conclusiones y recomendaciones.

Conclusiones

En el presente trabajo de titulación se realizó el diseño, la implementación y la validación de un sistema que permite emular las funcionalidades de un mouse, a través del uso de algoritmos de visión artificial y reconocimiento de voz.

Se realizó un buen estudio del estado del arte sobre trabajos similares previamente realizados, esto facilitó el desarrollo de un enfoque para lograr que este trabajo sea innovador y original.

Se recurrió al uso de potentes librerías como OpenCV y Dlib, las cuales facilitaron la implementación de algoritmos de visión artificial, que permitieron desarrollar un sistema para controlar el movimiento del cursor y su activación. Se utilizó modelos desarrollados en Dlib para detectar los rostros y predecir los rasgos faciales.

Se utilizó la librería Vosk y sus modelos acústicos de lenguaje español, para implementar los algoritmos necesarios para lograr el reconocimiento de voz de una forma sencilla y sin la necesidad de que los usuarios tengan conexión a internet. Como complemento a esto, para mejorar el movimiento del cursor, se desarrolló funciones que permitieron segmentar la pantalla en una matriz de 4x3, y a través de diversos comandos de voz, se pudo acceder a cada uno de los cuadrantes formados.

Para lograr agrupar todas las funciones que conformaron el emulador, se recurrió al uso de programación por hilos, lo cual permitió que la ejecución de los comandos y la captura de video y audio funcionen de forma paralela.

Se realizó pruebas de funcionamiento del emulador variando la iluminación y se determinó que el sistema puede funcionar en entornos poco iluminados, sin embargo, dependerá en gran parte de que tan bien se logre enfocar el rostro del usuario. Mientras más bien iluminado esté el rostro, mejor será la detección y predicción de las características faciales. En entornos más oscuros se podrán generar fallos, debido a que la detección del rostro será más complicada.

Realizando pruebas en situaciones en las que el usuario se encuentra junto a otras personas, se determinó que, si la cámara llega a captar más de un rostro, podría generar fallos, en donde se podría sustituir la capacidad de control del movimiento del cursor de la persona que está usando el emulador. Por otro lado, si hay un ambiente bullicioso y el micrófono es muy sensible y preciso, pueden surgir dificultades para usar los comandos de voz.

Se realizaron pruebas de desempeño en varios computadores de diferentes características, se analizaron varios aspectos en donde se pudo determinar que, el emulador puede funcionar en dispositivos de todas las gamas, siendo mejor su desempeño mientras mejores sean los computadores. Esto permite concluir que el emulador cumple con el requerimiento de ser de bajo costo ya que puede funcionar en máquinas de gama baja, el único gasto adicional que representaría sería la adquisición de una cámara o micrófono externo, si fuese necesario. Además, su funcionamiento en máquinas con sistemas operativos de distribución libre como Linux, es igual de bueno que en Windows.

Mediante las pruebas en diferentes computadores se determinó que, el correcto funcionamiento del emulador se ve afectado por el estado de la máquina, esto debido a que, si se encuentra saturada, el emulador empezará a funcionar de forma lenta e imprecisa.

Se realizó pruebas de desempeño en computadores en los cuales el emulador no funcionó en su totalidad o parcialmente. Con procesadores inferiores a un Intel Core i3 y en máquinas con sistema operativo Windows 7, el emulador no funcionó. En el caso de las pruebas realizadas en la máquina virtual

y en la tarjeta Raspberry, con 6 y 1 GB de RAM respectivamente, el sistema funcionó, pero la respuesta no fue muy buena. A partir de estos resultados, se considera que las características mínimas del computador donde se ejecute el programa deberán ser: un procesador igual o mejor a un i3, una memoria RAM mayor a 6GB y sistema operativo superior a Windows 7.

El emulador se ha desarrollado para funcionar en distintos sistemas operativos. Si bien unos pocos comandos de voz funcionan únicamente en Windows, la mayoría funcionan de manera correcta en sistemas operativos similares a Linux.

Se logró validar la funcionalidad del emulador a través de pruebas a distintos usuarios, quienes a través de una encuesta consideraron que puede ser de gran utilidad para personas que padezcan de discapacidades motrices.

Mediante análisis por edades de las personas que realizaron las pruebas con el emulador, se pudo determinar que quienes poseen mayor edad, tienen más dificultad a la hora de adaptarse al uso del emulador, para los usuarios más jóvenes, esto les resultó mucho más sencillo.

En las pruebas a usuarios se les solicitó su opinión acerca de características específicas del emulador, como: la velocidad de movimiento, los comandos de segmentación de pantalla y la función de reseteo de movimiento del cursor. El 100% de los usuarios se sintió a gusto con la velocidad a la que se mueve el cursor y esto se complementó con los resultados positivos de la pregunta referente a la utilidad de los comandos de segmentación de pantalla, los cuales se desarrollaron con la finalidad de permitir un movimiento más rápido. En cuanto a la función de reseteo, los resultados fueron regulares, ya que un porcentaje considerable de usuarios la sobre utilizó. Sin embargo, se puede decir que, con mayor práctica, el uso de esta función se puede reducir considerablemente.

A través de la encuesta a usuarios se realizó un análisis para determinar cómo es el funcionamiento de la detección de comandos de voz y de los rostros. En ambos casos se obtuvo respuestas

positivas, lo que indica que con los dispositivos de hardware que se utilizaron, no existieron problemas considerables. También se les preguntó si se realizó la prueba con dispositivos externos, y los resultados mostraron que la mayoría de usuarios no utilizó una cámara externa, sin embargo, si utilizaron micrófonos externos. Esto indica que los requerimientos de hardware para lograr un buen desempeño del reconocimiento de rostros son menores a los necesarios para una buena detección de la voz.

A través de las encuestas se determinó que las dificultades que se presentaron más comúnmente en los usuarios estuvieron relacionadas con los comandos de voz, específicamente por la necesidad de memorizarse varias palabras, o tener que hablar muy alto para que se ejecuten los comandos. Otra dificultad que se evidenció fue el posicionamiento correcto del rostro, para evitar el uso excesivo de la función de reseteo.

Recomendaciones

A continuación, se detallan varias recomendaciones para lograr sacar el máximo de provecho al emulador de *mouse* presentado en este trabajo.

- Para un correcto funcionamiento del emulador, el usuario debe estar en un lugar bien iluminado y poco bullicioso.
- Se debe evitar que la cámara enfoque más de un rostro, ya que pueden surgir errores.
- Se recomienda sentarse a una distancia prudente de la cámara, para que el detector de rostros y el predictor funcionen correctamente.
- Si se está usando un micrófono interno y los comandos de voz no se ejecutan inmediatamente, o hay que hablar muy alto, se recomienda utilizar un micrófono externo.

- Si al momento de realizar un movimiento brusco con el cuerpo, y el cursor empieza a moverse sin sentido, se debe resetear la referencia o posicionar adecuadamente el cuerpo, para que el funcionamiento vuelva a ser normal.
- Una vez que se haya finalizado de usar el modo dictado, es necesario desactivarlo, ya que, si está activado en alguna aplicación, se puede simular que se presionan teclas que sirvan como comandos y por lo tanto se generarán acciones no deseadas.

Trabajos futuros

Este trabajo puede servir como base para desarrollar emuladores más sofisticados que puedan estar enfocados en otros sistemas operativos. Se pueden realizar emuladores similares para que funcionen en diferentes idiomas, trabajando con módulos de reconocimiento de voz online, los cuales pueden brindar mejores resultados en algunas situaciones.

Se puede implementar el uso de visión artificial y detección de voz de forma similar a la que se aplicó en este trabajo, para que personas con discapacidades motrices puedan controlar otro tipo de sistemas, como pueden ser vehículos o sillas de ruedas.

Bibliografía

Aggarwal, N. (2020, noviembre 13). *Queue in Python*. GeeksforGeeks.

<https://www.geeksforgeeks.org/queue-in-python/>

Alpha Cephei. (2020, Junio 6). *VOSK Offline Speech Recognition API*. Alpha Cephei.

<https://alphacephei.com/vosk/>

Alvear, V., Rosero, P., Peluffo, D., & Pijal, J. (2017). Internet de las Cosas y Visión Artificial, Funcionamiento y Aplicaciones: Revisión de Literatura. *Enfoque UTE*, 7(1), 244–256.

<http://ingenieria.ute.edu.ec/enfoqueute/>

Anderson, J. (2019, marzo 25). *An Intro to Threading in Python*. Real Python.

<https://realpython.com/intro-to-python-threading/>

Arango, H. (2010). La evolución del computador y el desarrollo de la informática: un reto hacia los nuevos modelos en la educación. *Publicaciones ICESI*, 1(21).

https://repository.icesi.edu.co/biblioteca_digital/bitstream/10906/2654/1/Evolucion_computador_desarrollo.pdf

Azevedo, L., & Nunes da Ponte, M. (2013). Periféricos de computador adaptados. In *La Interacción de Personas con Discapacidad con el Computador: Experiencias y Posibilidades en Iberoamérica* (1st ed., pp. 27–28). CYTED.

[https://segib.org/wp-content/uploads/Libro_CYTED_2013_2a_edicion_small-\(1\).pdf](https://segib.org/wp-content/uploads/Libro_CYTED_2013_2a_edicion_small-(1).pdf)

Barrios, K., López, J., Mendieta, S., Benavides, R., & Sáez, Y. (2018). Sistema de reconocimiento de voz: un enlace en la comunicación hombre-máquina. *Revista de Iniciación Científica*, 4(2), 92–

95. <https://doi.org/10.33412/REV-RIC.V4.0.1827>

- Chandra, A. (2018, octubre 7). *Mouse Cursor Control Using Facial Movements — An HCI Application*. Towards Data Science. <https://towardsdatascience.com/mouse-control-facial-movements-hci-app-c16b0494a971>
- Chávez, I. (2009). *Sistema automático de reconocimiento de voz* [Universidad Michoacana de San Nicolás de Hidalgo].
http://bibliotecavirtual.dgb.umich.mx:8083/xmlui/handle/DGB_UMICH/3424
- Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, 1*, 886–893. <https://doi.org/10.1109/CVPR.2005.177>
- de Lucena, S. E., & Conzelmann, P. (2019, agosto). Computer USB-Mouse emulation using EOG. *INSCIT 2019 - 4th International Symposium on Instrumentation Systems, Circuits and Transducers*. <https://doi.org/10.1109/INSCIT.2019.8868754>
- del Valle, L. (2016, mayo 23). *Visión artificial, OpenCV y Python, primeros pasos para analizar imágenes*. <https://programarfacil.com/podcast/81-vision-artificial-opencv-phyton/>
- Díaz, J., & Morales, C. (2019). *Desarrollo de una interfaz mediante señales EOG para el manejo de la computadora por parte de una persona con discapacidad en los miembros superiores*. [Universidad Peruana de Ciencias Aplicadas].
https://repositorioacademico.upc.edu.pe/bitstream/handle/10757/652754/D%C3%ADaz_SJ.pdf?sequence=11&isAllowed=y
- Espada, R., Moreno, R., & Morán, M. (2020). Educación inclusiva y TIC: Sistemas de barrido ocular para alumnado con parálisis cerebral en Educación Primaria. *Revista de La Facultad de Educación de Albacete*, 35(2), 171–190. <https://doi.org/10.18239/ENSAYOS.V35I2.2551>

- Esparza, E., & Avalos, B. (2003). Reconocimiento de voz. *Conciencia Tecnológica*, 1(22).
<https://www.redalyc.org/pdf/944/94402206.pdf>
- Espinoza, D., & Jorquera, P. (2015). *Reconocimiento Facial* [Pontificia Universidad Católica de Valparaíso]. http://opac.pucv.cl/pucv_txt/Txt-1000/UCD1453_01.pdf
- Ferrin, C., Mosquera, J., Pino, J., Moctezuma, L., Burgos, J., Aragón, L., & Loaiza, H. (2021). Interfaz humano-computador basada en gestos faciales y orientada a la aplicación WhatsApp para personas con limitación motriz de miembros superiores. *Tecnológicas*, 24(50), 72–96.
<https://doi.org/10.22430/22565337.1722>
- Fincher, J. (2019, septiembre 16). *PyGame: Introducción a la programación de juegos en Python*. Real Python. <https://realpython.com/pygame-a-primer/>
- Freire, T., Cavalieri, D., Sarcinelli, M., Azorín, J., Iáñez, E., & Úbeda, A. (2013). Interfaces basadas en movimientos oculares: electrooculografía (EOG) y videoo. In *La Interacción de Personas con Discapacidad con el Computador: Experiencias y Posibilidades en Iberoamérica* (1st ed., pp. 107–123). CYTED. [https://segib.org/wp-content/uploads/Libro_CYTED_2013_2a_edicion_small-\(1\).pdf](https://segib.org/wp-content/uploads/Libro_CYTED_2013_2a_edicion_small-(1).pdf)
- García, I., & Caranqui, V. (2015). La visión artificial y los campos de aplicación. *Tierra Infinita*, 1(1), 94–103. <https://doi.org/10.32645/26028131.76>
- Gimeno, R. (2010). *Estudio de Técnicas de Reconocimiento Facial*. [Universidad Politécnica de Catalunya].
https://upcommons.upc.edu/bitstream/handle/2099.1/9782/PFC_RogerGimeno.pdf
- Gür, D., Schäfer, N., Kupnik, M., & Beckerle, P. (2020). A Human–Computer Interface Replacing Mouse and Keyboard for Individuals with Limited Upper Limb Mobility. *Multimodal*

- Technologies and Interaction 2020, Vol. 4, Page 84, 4(4), 100.*
<https://doi.org/10.3390/MTI4040084>
- Kazemi, V., & Sullivan, J. (2014). One millisecond face alignment with an ensemble of regression trees. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1867–1874. <https://doi.org/10.1109/CVPR.2014.241>
- King, D. (2014, March 3). *dlib: A toolkit for making real world machine learning and data analysis applications in C++*. Github. <https://github.com/davisking/dlib>
- Loewenich, F., & Maire, F. (2007). Hands-free mouse-pointer manipulation using motion-tracking and speech recognition. *Australasian Computer-Human Interaction Conference*, 295–302.
<https://doi.org/10.1145/1324892.1324955>
- Lofaro, L. (2018, abril 16). *Working With JSON Data in Python*. Real Python.
<https://realpython.com/python-json/>
- Lund, M. E., Caltenco, H. A., Lontis, E. R., Christiensen, H. V., Bentsen, B., & Struijk, L. N. S. A. (2009). A framework for mouse and keyboard emulation in a tongue control system. *Proceedings of the 31st Annual International Conference of the IEEE Engineering in Medicine and Biology Society: Engineering the Future of Biomedicine, EMBC 2009*, 815–818.
<https://doi.org/10.1109/IEMBS.2009.5334055>
- Lupu, R. G., Ungureanu, F., & Siriteanu, V. (2013, noviembre). Eye tracking mouse for human computer interaction. *2013 E-Health and Bioengineering Conference*.
<https://doi.org/10.1109/EHB.2013.6707244>
- Majaranta, P., & Bulling, A. (2014). Eye Tracking and Eye-Based Human–Computer Interaction. *Advances in Physiological Computing*, 39–65. https://doi.org/10.1007/978-1-4471-6392-3_3

- Marín, R. (2020, febrero 12). *¿Qué es OpenCV? Instalación en Python y ejemplos básicos*. Revista Digital-INESEM. <https://revistadigital.inesem.es/informatica-y-tics/opencv/>
- Martinez, F., Portale, G., Klein, H., & Olmos, O. (2003). *Reconocimiento de voz, apuntes de cátedra para Introducción a la Inteligencia Artificial* (Vol. 1). Universidad Tecnológica Nacional. https://www.frba.utn.edu.ar/wp-content/uploads/2021/02/IA1_IntroReconocimientoVoz.pdf
- McFarland, D. J., Krusienski, D. J., Sarnacki, W. A., & Wolpaw, J. R. (2008). Emulation of computer mouse control with a noninvasive brain–computer interface. *Journal of Neural Engineering*, 5(2), 110. <https://doi.org/10.1088/1741-2560/5/2/001>
- Mosquera, J., Loaiza, H., Nope, S., & Restrepo, A. (2017). Identifying facial gestures to emulate a mouse: Navigation application on Facebook. *IEEE Latin America Transactions*, 15(1), 121–128. <https://doi.org/10.1109/TLA.2017.7827915>
- Mosquera, J., Loaiza, H., Nope, S., & Restrepo, A. (2020). Human-computer multimodal interface to internet navigation. *Disability and Rehabilitation: Assistive Technology*, 16(8), 807–820. <https://doi.org/10.1080/17483107.2020.1799440>
- PyAutoGUI. (2021, Julio 7). *PyAutoGUI documentation*. <https://pyautogui.readthedocs.io/en/latest/index.html>
- Pypi. (2017, marzo 18). *PyAudio*. <https://pypi.org/project/PyAudio/>
- Rev. (2021, junio 14). *All You Need to Know About Automatic Speech Recognition Transcription Models*. Rev. <https://www.rev.com/blog/guide-to-speech-recognition-transcription-models>
- Ronquillo, A. (2019, marzo 11). *A Beginner's Guide to the Python time Module*. Real Python. <https://realpython.com/python-time-module/>

- Rosebrock, A. (2014, noviembre 10). *Histogram of Oriented Gradients and Object Detection*. PyImageSearch. <https://pyimagesearch.com/2014/11/10/histogram-oriented-gradients-object-detection/>
- Rosebrock, A. (2015, enero 23). *imutils: A series of convenience functions to make basic image processing operations such as translation, rotation, resizing, skeletonization, and displaying Matplotlib images easier with OpenCV and Python*. Github. <https://github.com/PyImageSearch/imutils>
- Rosebrock, A. (2017, abril 3). *Facial landmarks with dlib, OpenCV, and Python*. PyImageSearch. <https://pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/>
- Sagonas, C., Tzimiropoulos, G., Zafeiriou, S., & Pantic, M. (2013). 300 faces in-the-wild challenge: The first facial landmark Localization Challenge. *Proceedings of the IEEE International Conference on Computer Vision*, 397–403. <https://doi.org/10.1109/ICCVW.2013.59>
- Sanabria S, J. J., & Archila D, J. F. (2011). Detección y análisis de movimiento usando visión artificial. *Scientia et Technica Año XVI*, 16(49), 180–188.
- Sánchez, A. (2020, octubre 4). *La librería Numpy*. Aprende Con Alf. <https://aprendeconalf.es/docencia/python/manual/numpy/>
- Sinha, K., Lalwani, B., Shah, R., & Jalora, M. (2015). Face Enable Mouse using Motion Detection and Speech Recognition. *International Journal for Innovative Research in Science & Technology*, 1(10), 141–143.
- Soukupova, T., & Cech, J. (2016). *Real-Time Eye Blink Detection using Facial Landmarks* (L. Cehovin, R. Mandeljc, & V. Struc, Eds.; Vol. 21). <https://vision.fe.uni-lj.si/cvww2016/proceedings/papers/05.pdf>

Zuluaga, C., Álvarez, D., & Orozco, Á. (2009). Técnicas de seguimiento de características faciales en secuencias de imágenes basadas en métodos de libre modelo. *Scientia et Technica*, 3(43), 13–18.

<https://dialnet.unirioja.es/servlet/articulo?codigo=4587881>

Apéndices