



UNIVERSIDAD TÉCNICA DE AMBATO

**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E
INDUSTRIAL**

CARRERA DE SOFTWARE

Tema:

**DESARROLLO DE UNA APLICACION PARA LA GENERACIÓN DE
ILUSTRACIONES 2D DE FIGURAS HUMANAS UTILIZANDO RECURSOS
BASADOS EN INTELIGENCIA ARTIFICIAL Y SU EFECTO EN LOS
TIEMPOS DE CREACIÓN**

Trabajo de titulación modalidad Proyecto de Investigación, presentado previo a la
obtención del título de Ingeniero en Software

ÁREA: Desarrollo de Software

LÍNEA DE INVESTIGACIÓN: Software, Tecnologías de la Información y
Ciencia de Datos

AUTOR: Mauricio Fernando Villafuerte Grijalva

TUTOR: Ing. Santiago David Jara Moya, MSc

Ambato - Ecuador

enero – 2026

APROBACIÓN DEL TUTOR

En calidad de tutor del trabajo de titulación con el tema: DESARROLLO DE UNA APLICACION PARA LA GENERACIÓN DE ILUSTRACIONES 2D DE FIGURAS HUMANAS UTILIZANDO RECURSOS BASADOS EN INTELIGENCIA ARTIFICIAL Y SU EFECTO EN LOS TIEMPOS DE CREACIÓN, desarrollado bajo la modalidad Proyecto de Investigación por el señor Mauricio Fernando Villafuerte Grijalva, estudiante de la Carrera de Software, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 17 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.3 del instructivo del reglamento referido.

Ambato, enero 2026.



**Santiago David Jara
Moya**

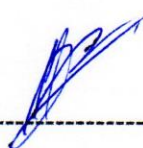


Ing. Santiago David Jara Moya, MSc
TUTOR

AUTORÍA

El presente trabajo de titulación con el tema: DESARROLLO DE UNA APLICACION PARA LA GENERACIÓN DE ILUSTRACIONES 2D DE FIGURAS HUMANAS UTILIZANDO RECURSOS BASADOS EN INTELIGENCIA ARTIFICIAL Y SU EFECTO EN LOS TIEMPOS DE CREACIÓN es absolutamente original, auténtico y personal y ha observado los preceptos establecidos en la Disposición General Quinta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, enero 2026.



Mauricio Fernando Villafuerte Grijalva

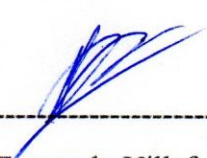
C.C. 1804450383

AUTOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato para que reproduzca total o parcialmente este trabajo de titulación dentro de las regulaciones legales e institucionales correspondientes. Además, cedo todos mis derechos de autor a favor de la institución con el propósito de su difusión pública, por lo tanto, autorizo su publicación en el repositorio virtual institucional como un documento disponible para la lectura y uso con fines académicos e investigativos de acuerdo con la Disposición General Cuarta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato.

Ambato, enero 2026.



Mauricio Fernando Villafuerte Grijalva

C.C. 1804450383

AUTOR

APROBACIÓN DEL TRIBUNAL DE GRADO

En calidad de par calificador del informe final del trabajo de titulación presentado por el/la señor Mauricio Fernando Villafuerte Grijalva, estudiante de la Carrera de Software, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado DESARROLLO DE UNA APLICACION PARA LA GENERACIÓN DE ILUSTRACIONES 2D DE FIGURAS HUMANAS UTILIZANDO RECURSOS BASADOS EN INTELIGENCIA ARTIFICIAL Y SU EFECTO EN LOS TIEMPOS DE CREACIÓN, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 19 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.4 del instructivo del reglamento referido. Para cuya constancia suscribimos, conjuntamente con el señor Presidente del Tribunal.

Ambato, enero 2026.



Ing. Franklin Oswaldo Mayorga Mayorga Mg.
PRESIDENTE DEL TRIBUNAL



Ing. Clay Fernando Aldás Flores, Mg.

PROFESOR CALIFICADOR



Ing. Elsa Pilar Urrutia Urrutia

PROFESOR CALIFICADOR

DEDICATORIA

Dedico la materialización de este logro a mi familia, quienes siempre me han ayudado y dado su apoyo en los momentos difíciles. También a quienes me ayudaron aun sin ser conscientes de ello a seguir adelante.

AGRADECIMIENTO

A mis seres queridos, por su apoyo constante y por mantenerme cuerdo a lo largo de este camino.

A mis profesores, cuya dedicación y enseñanzas han enriquecido mi formación.

A mi tutor, por su orientación y guía durante el desarrollo de este proyecto.

ÍNDICE GENERAL DE CONTENIDOS

PORTADA.....	i
APROBACIÓN DEL TUTOR.....	ii
AUTORÍA.....	iii
DERECHOS DE AUTOR	iv
APROBACIÓN DEL TRIBUNAL DE GRADO.....	v
DEDICATORIA.....	vi
AGRADECIMIENTO	vii
ÍNDICE GENERAL DE CONTENIDOS	viii
ÍNDICE DE TABLAS.....	xii
ÍNDICE DE FIGURAS.....	xiii
ÍNDICE DE ANEXOS	xvi
RESUMEN EJECUTIVO	xvii
ABSTRACT	xviii
CAPÍTULO I. MARCO TEÓRICO.....	1
1.1 Tema de investigación.....	1
1.1.1 Planteamiento del problema.....	1
1.2 Antecedentes investigativos	2
1.3 Fundamentación teórica	10

1.3.1 Representación visual de personas en formato bidimensional.....	10
1.3.2 Lenguaje de programación.....	11
1.3.3 Librerías tecnológicas	11
1.3.4 Metodología Kanban.....	11
1.3.5 Inteligencia artificial	12
1.3.6 Captura de postura.....	12
1.3.7 Modelo 2D humano.....	12
1.4 Objetivos	13
1.4.1 Objetivo general.....	13
1.4.2 Objetivos específicos	13
CAPÍTULO II. METODOLOGÍA.....	14
2.1 Materiales.....	14
2.1.1 Modalidad de la investigación	14
2.1.2 Población y muestra	15
2.1.3 Recolección de información.....	18
2.1.4 Procesamiento y análisis de datos	34
CAPÍTULO III. RESULTADOS Y DISCUSIÓN.....	35
3.1 Análisis de herramientas	35

3.1.1 Análisis del uso de herramientas para generación de imágenes	35
3.1.2 Análisis de metodologías de desarrollo.....	37
3.1.3 Análisis de herramientas de desarrollo	40
3.2 Definición de requerimientos	44
3.3 Desarrollo de la aplicación.....	46
3.3.1 Investigación de las características de herramientas preexistentes	46
3.3.2 Definición de requerimientos funcionales y no funcionales	47
3.3.3 Diagrama de procesos actual y el propuesto	47
3.3.4 Diseño de la interfaz de la aplicación	50
3.3.5 Diseño del modelo de figura humana 2D por defecto.....	53
3.3.6 Desarrollo del módulo de la ventana principal	56
3.3.7 Desarrollo del módulo del modelo 2D	58
3.3.8 Desarrollo del módulo de detección de postura	60
3.3.9 Finalización de la interfaz y opciones de personalización	62
3.4 Implementación de la aplicación.....	65
3.4.1 Arquitectura de la aplicación	65
3.4.2 Inicialización de la aplicación y generación del ejecutable	67
3.5 Validación con usuarios	70

3.5.1 Reducción de la dificultad para creación de ilustraciones	70
3.5.2 Reducción del tiempo requerido	72
CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES	74
4.1 Conclusiones	74
4.2 Recomendaciones.....	75
REFERENCIAS BIBLIOGRÁFICAS.....	76
ANEXOS.....	78

ÍNDICE DE TABLAS

Tabla 1. Escala de Likert para evaluación de preguntas	19
Tabla 2. Promedio de los criterios evaluados en la encuesta de Likert.....	19
Tabla 3. Comparativa cualitativa de Herramientas de edición tradicionales.	35
Tabla 4. Comparación de recursos para la detección de posturas humanas.....	41
Tabla 5. Comparación de librerías gráficas y multimedia de Python.	43
Tabla 6. Planificación de requerimientos del sistema por iteraciones	45

ÍNDICE DE FIGURAS

Figura 1. Frecuencia de librerías utilizadas.....	9
Figura 2. Frecuencia de lenguajes de programación utilizados	10
Figura 3. Resultados de la pregunta 1	20
Figura 4. Resultados de la pregunta 2	21
Figura 5. Resultados de la pregunta 3	22
Figura 6. Resultados de la pregunta 4	23
Figura 7. Resultados de la pregunta 5	24
Figura 8. Resultados de la pregunta 6	25
Figura 9. Resultados de la pregunta 7	26
Figura 10. Resultados de la pregunta 8	27
Figura 11. Resultados de la pregunta 9	28
Figura 12. Resultados de la pregunta 10	29
Figura 13. Resultados de la pregunta 11	30
Figura 14. Resultados de la pregunta 12	31
Figura 15. Resultados de la pregunta 13	32
Figura 16. Resultados de la pregunta 14	33
Figura 17. Polígono de frecuencias de creación de ilustraciones con Krita en minutos.	34
Figura 18. Ilustración del tablero Kanban.....	38
Figura 19. Tablero Kanban del desarrollo.....	39

Figura 20. Diagrama de flujo de Krita	48
Figura 21. Diagrama de flujo de la aplicación desarrollada.....	49
Figura 22. Interfaz principal del prototipo	50
Figura 23. Interfaz de personalización del modelo 2D	51
Figura 24. Interfaz de personalización de expresiones del modelo 2D.....	52
Figura 25. Interfaz de personalización de fondo	53
Figura 26. Estructura del modelo 2D de figura humana	54
Figura 27. Modelo 2D de figura humana por defecto terminada.....	55
Figura 28. Expresiones de modelo 2D de figura humana por defecto	55
Figura 29. Segunda apariencia del modelo 2D de figura humana	56
Figura 30. Interfaz del módulo ventana principal	57
Figura 31. Módulo del modelo 2D en la interfaz.....	58
Figura 32. Ajuste y rotación de una extremidad del modelo 2D en la interfaz.....	60
Figura 33. Módulo de detección de postura en la interfaz	62
Figura 34. Interfaz y funcionalidad completadas.....	65
Figura 35. Estructura de directorios y módulos de la aplicación	67
Figura 36. Fotograma del tutorial de la aplicación	68
Figura 37. Ejecución del comando de PyInstaller en el terminal.....	70
Figura 38. Aplicación desarrollada en ejecución	71
Figura 39. Polígono de frecuencias de los tiempos de creación de ilustraciones con la aplicación desarrollada en segundos	72

Figura 40. Comparación de promedios de tiempos de creación de ilustraciones en minutos.....	73
Figura 41. Vista del documento de los resultados y la submuestra.	83
Figura 42. Portada del video tutorial.....	87

ÍNDICE DE ANEXOS

Anexo A. Encuesta para la evaluación de la viabilidad y aceptación.....	78
Anexo B. Resultados de la encuesta y de la submuestra.....	83
Anexo C. Levantamiento de requerimientos – IEEE 830	83
Anexo D. Video tutorial de la aplicación desarrollada.	87

RESUMEN EJECUTIVO

Las ilustraciones 2D de figuras humanas constituyen un medio ampliamente utilizado en presentaciones, materiales informativos y señalética debido a su capacidad para representar escenas y conceptos de manera clara. No obstante, a pesar de su aparente simplicidad visual, su elaboración suele requerir conocimientos intermedios o avanzados en herramientas de edición o ilustración digital, lo que dificulta su creación para usuarios sin experiencia previa.

Para facilitar el acceso se desarrolló una aplicación orientada a la generación de ilustraciones 2D de figuras humanas mediante recursos basados en Inteligencia Artificial. La herramienta incorpora un modelo 2D y automatización de procesos repetitivos y complejos. El desarrollo se realizó bajo la metodología Kanban y utilizando Python como lenguaje principal. Para la interfaz y la interacción del modelo se empleó Pygame, mientras que MediaPipe se usó como componente de Inteligencia Artificial para la captura y transferencia de posturas de fotografías de referencia.

Los resultados mostraron una reducción del 85% en el tiempo necesario para generar una ilustración en comparación con un flujo de trabajo tradicional en herramientas, además de una simplificación significativa del proceso creativo. La aplicación permite ajustar posiciones del modelo, importar una fotografía para replicar su postura, personalizar la apariencia del personaje y exportar la ilustración resultante con fondo transparente.

El proyecto demostró que la integración de técnicas de Inteligencia Artificial con interfaces simplificadas puede optimizar de forma sustancial la generación de ilustraciones 2D, haciéndola accesible para usuarios sin experiencia y reduciendo tanto la complejidad como el tiempo requerido en la creación de imágenes.

Palabras claves: Ilustraciones 2D, modelo 2D, Inteligencia Artificial, tiempo de ilustraciones, procesos repetitivos, interacción de usuario.

ABSTRACT

The 2D illustrations of human figures are a widely used medium in presentations, informational materials, and signage due to their ability to clearly represent scenes and concepts. However, despite their apparent visual simplicity, their creation often requires intermediate or advanced knowledge of digital editing or illustration tools, making them difficult for users without prior experience.

To facilitate access, an application was developed for generating 2D illustrations of human figures using artificial intelligence-based resources. The tool incorporates a 2D model and automates repetitive and complex processes. Development was carried out using the Kanban methodology and Python as the primary language. Pygame was used for the model's interface and interaction, while MediaPipe was used as the AI component for capturing and transferring poses from reference photographs.

The results showed an 85% reduction in the time required to generate an illustration compared to a traditional workflow using various tools, as well as a significant simplification of the creative process. The application allows users to adjust the model's position, import a photograph to replicate its pose, customize the character's appearance, and export the resulting illustration with a transparent background.

The project demonstrated that integrating artificial intelligence techniques with simplified interfaces can substantially optimize the generation of 2D illustrations, making it accessible to inexperienced users and reducing both the complexity and time required for image creation.

Keywords: 2D illustrations, 2D model, Artificial Intelligence, illustration time, repetitive processes, user interaction.

CAPÍTULO I. MARCO TEÓRICO

1.1 Tema de investigación

DESARROLLO DE UNA APLICACION PARA LA GENERACIÓN DE ILUSTRACIONES 2D DE FIGURAS HUMANAS UTILIZANDO RECURSOS BASADOS EN INTELIGENCIA ARTIFICIAL Y SU EFECTO EN LOS TIEMPOS DE CREACIÓN

1.1.1 Planteamiento del problema

La representación visual se ha convertido en un componente fundamental en la creación de contenidos educativos, informativos y de comunicación. Las ilustraciones digitales en formato 2D, especialmente aquellas que muestran figuras humanas, se utilizan ampliamente para facilitar la comprensión de conceptos en distintas áreas. Sin embargo, muchas de las herramientas existentes para generar este tipo de imágenes están orientadas a usuarios con experiencia en diseño gráfico, lo que limita su uso por parte de personas sin formación técnica. Esto genera una brecha entre la demanda de recursos visuales simples y las opciones disponibles que puedan ser utilizadas de forma práctica por un público general.

En Ecuador, esta situación se presenta con particular claridad debido a restricciones en el acceso a software especializado y a la falta de capacitación técnica en herramientas de diseño por parte de muchos usuarios. Aunque existe una necesidad real de incorporar elementos visuales en documentos y presentaciones, las herramientas disponibles suelen tener interfaces complejas y requieren conocimientos específicos. Las soluciones basadas en inteligencia artificial representan una alternativa, pero su aplicación no siempre ofrece resultados adecuados y, en muchos casos, requiere realizar varios intentos antes de obtener una imagen aceptable, lo que implica pérdida de tiempo y recursos.

En el entorno universitario, esta situación puede llegar a ser un obstáculo significativo para estudiantes y docentes que necesiten de la generación de ilustraciones de figuras humanas para sus trabajos, presentaciones o materiales didácticos. En áreas distintas al diseño gráfico, la mayoría de los usuarios carece de las competencias técnicas necesarias para manejar software especializado, lo que limita la calidad de los recursos visuales. Esta carencia puede derivar en la exclusión de elementos gráficos esenciales o en la utilización de imágenes inadecuadas.

1.2 Antecedentes investigativos

En el desarrollo de esta investigación resulto indispensable revisar los antecedentes que han explorado la aplicación de técnicas de inteligencia artificial en la generación de ilustraciones digitales, particularmente aquellas orientadas a la representación de figuras humanas en 2D. Esta revisión permite identificar los avances logrados en el área, así como reconocer las limitaciones y enfoques diferenciadores en comparación con el presente proyecto. A partir de este análisis se establecieron las bases teóricas y metodológicas que sustentaron la propuesta y orientaron el estudio de su efecto en los tiempos de creación.

En [1] propone un sistema de captura de movimiento en tiempo real para convertir los movimientos humanos en animaciones de personajes virtuales en 2D(two-dimensional). Dividen el método en dos fases: la primera se centra en el diseño y creación del personaje 2D utilizando herramientas gráficas como Photoshop y su despliegue en Unity mientras que la segunda construye un sistema para detectar y rastrear los movimientos de la parte superior del cuerpo (cabeza, brazos y expresiones faciales) mediante una cámara de portátil simulando en tiempo real los movimientos en Unity a través de protocolos TCP(Transmission Control Protocol). Los autores implementan técnicas de visión por computadora utilizando OpenCV y Mediapipe para detectar las articulaciones del cuerpo y enviar los datos al personaje virtual en Unity. A diferencia de métodos tradicionales que requieren sensores físicos esta propuesta se apoya únicamente en cámaras comunes lo que simplifica la implementación. El sistema está enfocado en aplicaciones de animación en tiempo real

con especial interés en producciones 2D por su simplicidad y rapidez en el proceso de configuración.

En [2] presenta un método para estimar poses humanas en 3D(three-dimensional) usando un dispositivo monocular. La técnica utiliza el método OpenPose para obtener la pose en 2D a partir de videos continuos capturados. Luego, una red especializada que fusiona canales de profundidad calcula la pose en 3D con mayor precisión. Se diseñó una red de filtrado que optimiza y suaviza la estimación mediante una estrategia de ventana deslizante. Además, se aplican métodos para retargetizar esqueletos y mejorar la animación de personajes virtuales en tiempo real. La validación del método se realizó en el conjunto de datos Human3.6M, ampliamente reconocido en la comunidad. Los resultados muestran un error medio por articulación de 40.5 mm, cifra inferior a la de múltiples métodos benchmark comparados.

En [3] se ha utilizado el framework MediaPipe (v0.8.10) junto con el plugin MediaPipe Unity (v0.10.1) para la detección y seguimiento de landmarks faciales, de manos y cuerpo, permitiendo obtener datos 3D precisos para la manipulación de avatares digitales. Investigaciones con 18 participantes, con una edad promedio de 32.17 años, demostraron que la integración de estos datos con un rig generado en Blender mediante el add-on Rigify mejora significativamente la precisión de la animación. Se logró reducir errores de posición hasta en un 52.35% y errores angulares en un 6.45%, en comparación con métodos convencionales. Las evaluaciones subjetivas mediante la prueba estadística Wilcoxon indicaron que estas mejoras aportan animaciones más naturales y suaves. Los resultados muestran que la rigidez y artificialidad de los movimientos en modelos 3D disminuyen notablemente.

En [4] se analiza la evolución de las técnicas para la edición de mallas 3D, desde los métodos geométricos tradicionales hasta las recientes propuestas basadas en aprendizaje profundo. Las técnicas clásicas incluyen métodos ARAP(As-Rigid-As-Possible) y Laplacian-based que permiten deformaciones flexibles y controladas de las mallas. Se explica que los métodos geométricos presentan limitaciones para capturar detalles complejos y la diversidad de formas. Los autores destacan el impacto de los repositorios públicos con modelos 3D, los cuales han facilitado el entrenamiento de redes neuronales. Las redes neuronales se utilizan para aprender deformaciones

realistas de manera automática, superando las limitaciones de las técnicas manuales. Los métodos recientes permiten una edición más eficiente y adaptable para distintos tipos de mallas. El artículo concluye que la combinación de técnicas clásicas y aprendizaje profundo ha impulsado avances significativos en la edición 3D.

En [5] se analizan los avances en modelos generativos profundos para la síntesis de imágenes condicionales a partir de múltiples modalidades. El estudio presenta arquitecturas híbridas que combinan GANs(Generative Adversarial Networks) y VAEs(Variational Autoencoders), las cuales mejoran la calidad visual de las imágenes generadas. Se describe la incorporación de mecanismos de atención y normalización adaptativa espacial para optimizar la coherencia espacial en las escenas sintetizadas. Se emplean métricas objetivas como FID(Fréchet Inception Distance), IS(Inception Score), PSNR(Peak Signal-to-Noise Ratio) y SSIM(Structural Similarity Index Measure) para evaluar el desempeño de los modelos. Las métricas permiten comparar resultados en bases de datos que contienen hasta 1,2 millones de imágenes. Las bases de datos consideradas incluyen imágenes de objetos, humanos y vehículos, lo que garantiza una amplia cobertura de escenarios. La investigación destaca la importancia de la combinación multimodal para mejorar la síntesis de imágenes.

En [6] los métodos más usados para la estimación 3D de la postura de la mano a partir de una imagen de profundidad 2D convierten dicha imagen en coordenadas 3D. Una técnica convierte el mapa de profundidad 2D en una forma volumétrica 3D basada en una función de distancia firmada truncada. Moon et al. desarrollaron un modelo con redes 3D convolucionales que estima la probabilidad por vóxel para cada punto clave. El error promedio de distancia 3D con mapas de profundidad 2D es de 18.85 mm, mientras que con cuadrículas voxelizadas 3D se reduce a 16.78 mm. El refinamiento de la localización mejora el error de 11.14 mm a 9.22 mm. La aplicación de un ensamble de épocas disminuye aún más el error a 8.42 mm. El modelo V2V-PoseNet(Voxel-to-Voxel Pose Network) utiliza datos voxelizados para estimar la probabilidad por vóxel de cada punto clave en la mano.

En [7] se analizan avances en inteligencia artificial aplicados al dibujo con redes neuronales profundas y modelos de difusión. Estas tecnologías reducen barreras técnicas en el dibujo, aunque la evaluación estética sigue siendo un reto. Se emplean

modelos foundation que combinan procesamiento de imágenes, gráficos 3D y lenguaje natural para ofrecer soporte interactivo en tiempo real. Las herramientas permiten integrar bocetos digitales, movimientos corporales y descripciones en lenguaje natural. El proceso asistido enfrenta dificultades por la falta de habilidades técnicas y artísticas en los usuarios. Se propone el framework CIS(Contents - Intention understanding - Support), que organiza el proceso en contenido, comprensión de intención y soporte. Este enfoque mejora la colaboración entre humanos y máquinas mediante retroalimentación visual continua y soporte en tiempo real.

En [8] la generación automática de activos gráficos para entornos digitales ha ganado relevancia en los últimos años. La mayoría de los estudios se centran en activos 3D, tales como interiores, exteriores, hard-surface, escenarios y personajes. Los trabajos utilizan herramientas como Blender, Unity y Unreal Engine, que facilitan la creación y automatización de modelos tridimensionales. Se aplican técnicas avanzadas de aprendizaje profundo, incluyendo GANs y CNNs(Convolutional Neural Networks) . Los métodos de renderizado diferenciable mejoran la calidad visual y la eficiencia del proceso. La incorporación de modelos de texto a imagen, como DALL·E, amplía las posibilidades de síntesis visual en 2D y 3D. Esta área combina visión computacional, gráficos por computadora y aprendizaje automático, creando un campo interdisciplinario en constante evolución.

En [9] se destaca el uso de Transformers para mejorar la precisión en la estimación de postura humana a partir de imágenes. Se aplican técnicas de aprendizaje profundo que permiten clasificar y segmentar imágenes de manera automática. Además, se emplean modelos que combinan métodos manuales y automáticos para etiquetar imágenes con mayor exactitud. Los sistemas descritos facilitan la manipulación de imágenes segmentadas para ajustar posturas de forma interactiva mediante dispositivos táctiles y con mouse. La investigación resalta la importancia de interfaces intuitivas que optimizan la edición y generación de contenido gráfico. Estas metodologías buscan incrementar la rapidez y la precisión en la generación de modelos gráficos 2D personalizados. Finalmente, la integración de modelos gráficos con aprendizaje automático representa un avance importante en esta área.

En [10] el ámbito de la detección de objetos 3D en visión monocular, la arquitectura propuesta por el artículo presenta mejoras significativas en el rendimiento en comparación con métodos previos. En las evaluaciones realizadas sobre el conjunto de datos KITTI, la técnica introducida, denominada MonoSAID, logró una AP3D(Average Precision 3D) de 23.91% en el nivel "Fácil", 16.93% en el nivel "Moderado" y 14.19% en el nivel "Difícil" para la categoría de automóviles, superando a métodos como MonoCon, que obtuvo 22.88%, 16.46% y 13.95%, respectivamente. En la categoría de peatones, MonoSAID mostró un rendimiento destacado, alcanzando un AP3D de 13.93% en el nivel "Fácil", 9.32% en el nivel "Moderado" y 7.12% en el nivel "Difícil". Del mismo modo, en la categoría de ciclistas logró un AP3D de 4.99%, 2.72% y 2.36% para cada nivel de dificultad. Para el conjunto de datos Waymo se evidenció un desempeño superior, con una precisión media de 10.60% en el nivel 1 y 10.52% en el nivel 2, lo cual destaca su capacidad de generalización en escenarios más complejos. Estas mejoras son atribuidas al uso del módulo ABWE(Adaptive Bin Width Estimation) para la estimación adaptativa de anchos de bin y al módulo MSPE(Multi-Scale Perception Element) para una percepción espacial multiescala, optimizando así la capacidad de detección en contextos variados y desafiantes.

En [11] se presentan técnicas de HPE(Human Pose Estimation) que se dividen en métodos de volumen implícito y métodos de volumen aparente. Los métodos de volumen implícito representan los esqueletos en un sistema de coordenadas 3D, mientras que los métodos de volumen aparente utilizan superficies para encapsular las coordenadas tridimensionales de las articulaciones. En la estimación de poses en 2D, algunos modelos han alcanzado un mAP(Mean Average Precision) de hasta 75.6% en conjuntos de datos reconocidos. Otros enfoques basados en redes gráficas han logrado mAP cercanos al 67.6%. Ciertos modelos han conseguido valores de mAP superiores al 70% en múltiples conjuntos de datos, incluso alcanzando un 85.1% en algunos casos. Los métodos de volumen implícito requieren menos parámetros y entrenamiento en comparación con los métodos de volumen aparente. Sin embargo, estos métodos presentan limitaciones para aplicaciones en realidad virtual y juegos que requieren representaciones tridimensionales detalladas. Por esta razón, se siguen desarrollando técnicas que mejoren la precisión y la eficiencia de la estimación de poses humanas.

En [12] se abordan múltiples avances en la detección y análisis de objetos en entornos artísticos, destacando el uso de conjuntos de datos especializados. El conjunto de datos WikiArt Paintings contiene 81,000 imágenes distribuidas en 27 clases específicas, lo cual facilita la detección de objetos en obras de arte. El conjunto de datos People-Art cuenta con 4,500 imágenes orientadas a la detección de personas en escenarios artísticos; por lo tanto, permite un enfoque detallado en la identificación de figuras humanas. El modelo Styled Deeplabv3 se ha implementado para tareas de segmentación semántica y ha mostrado un desempeño considerable en la clasificación y segmentación de objetos dentro de contextos artísticos. Los modelos HRNet(High-Resolution Network) y DeepPose se han utilizado de manera efectiva en la estimación de poses humanas en obras de arte; en consecuencia, se ha logrado un reconocimiento preciso de posturas humanas. Técnicas como PiFU(Pixel-aligned Implicit Function) han permitido la reconstrucción 3D de figuras humanas a partir de imágenes bidimensionales, lo cual representa un avance importante en la interpretación de escenas artísticas complejas. Los desarrollos mencionados han mejorado la segmentación y el reconocimiento en entornos artísticos, impulsando la manipulación y generación de contenido visual mediante modelos de aprendizaje profundo.

En [13] se propone un sistema con sliders guiados por trayectorias para explorar los tiempos de inbetween entre fotogramas clave. Se realizó un estudio con seis participantes cuyas edades promedian 31.2 años y una desviación estándar de 9.42. Los resultados indican que la utilidad y la facilidad de uso del slider tradicional obtuvieron puntuaciones promedio de 3.83 y 3.00, respectivamente, mientras que el slider guiado sin función de guía alcanzó 5.83 y 6.00. Cuando se añadió la función de guía, ambas puntuaciones mejoraron a 6.00 con desviaciones estándar bajas, lo que evidencia una satisfacción general alta. La diferencia en las valoraciones fue estadísticamente significativa con valores p menores a 0.05 en pruebas de hipótesis de dos colas. Los usuarios destacaron que el slider guiado facilita la comprensión intuitiva del movimiento, y la función de guía reduce la barrera para usuarios sin experiencia previa. No obstante, se identificaron áreas de mejora, como la visualización para movimientos complejos y la inclusión de más tipos de guía.

En [14] se presenta EPEdit(Efficient Photo Editor), una aplicación web que integra modelos de inteligencia artificial para facilitar la edición de imágenes. El sistema

utiliza algoritmos de edición de imágenes zero-shot basados en el modelo Stable Diffusion, lo que elimina la necesidad de ajustes adicionales y reduce el consumo de recursos. EPedit ofrece funciones como generación de imágenes, reemplazo y eliminación de objetos, modificación de fondos, cambios en la pose y perspectiva, y edición específica por regiones, todo guiado mediante máscaras y comandos de texto. La interfaz de usuario fue diseñada para ser accesible, permitiendo su uso inmediato sin procesos complicados de inicio de sesión. En un estudio con 24 participantes entre 20 y 30 años, el sistema mostró tiempos de procesamiento de 12 a 15 segundos por muestra usando una GPU(Graphics Processing Unit) A-100. Los usuarios evaluaron positivamente la facilidad de uso, la adecuación a sus expectativas y la satisfacción general, destacando que EPedit supera en usabilidad y velocidad a programas tradicionales como Photoshop. La arquitectura del sistema está basada en el patrón Modelo-Vista-Controlador, con un frontend desarrollado en ReactJS y un backend en Express que se comunica con un servidor Python encargado del procesamiento de los modelos de IA(Artificial Intelligence).

En [15] se propone un modelo completamente neuronal que incorpora restricciones físicas para capturar movimientos humanos de manera realista y precisa a partir de un solo video monocular. Este enfoque emplea un controlador PD(Proportional-Derivative) neuronal llamado DyNet, el cual ajusta dinámicamente los coeficientes de ganancia en función de la pose objetivo y la pose actual; como resultado, se mitigan los retrasos en movimientos rápidos sin sacrificar la suavidad en las transiciones. El modelo introduce una GRFNet(Ground Reaction Force Network), la cual replica de forma realista las fuerzas externas necesarias para controlar el movimiento global del cuerpo y mantener el equilibrio. Las evaluaciones cuantitativas en los conjuntos de datos Human3.6M, MPI-INF-3DHP(Max Planck Institute for Informatics - 3D Human Pose) y DeepCap demuestran mejoras notables en métricas de precisión. Por ejemplo, se reporta un MPJPE(Mean Per Joint Position Error) de 52.6 mm, superando a métodos previos como PhysCap, VNect y VIBE en escenarios complejos y con transiciones rápidas. El modelo utiliza la Rigid Body Dynamics Library para el cálculo de cantidades dinámicas; se emplea Pybullet para la visualización y PyTorch permite la implementación de la red neuronal.

En la **Figura 1**. Frecuencia de librerías utilizadas se presenta un gráfico de barras que representa la cantidad de veces que distintas librerías fueron utilizadas en los antecedentes de la investigación. Las barras más altas corresponden a PyTorch y ROIAlign, cada una con ocho menciones, lo que refleja su amplio uso en trabajos relacionados con el aprendizaje profundo y la estimación precisa de poses. MediaPipe y TensorFlow, ambas con cuatro menciones, muestran una presencia significativa en estudios que emplean modelos de visión por computadora de alto nivel. Detectron2, con tres menciones, se vincula a tareas de segmentación y detección de objetos. OpenCV, con dos menciones, se ha empleado en aplicaciones más generales de procesamiento de imágenes. Finalmente, ShapeNet aparece una sola vez, indicando un uso más limitado en el análisis de modelos tridimensionales.

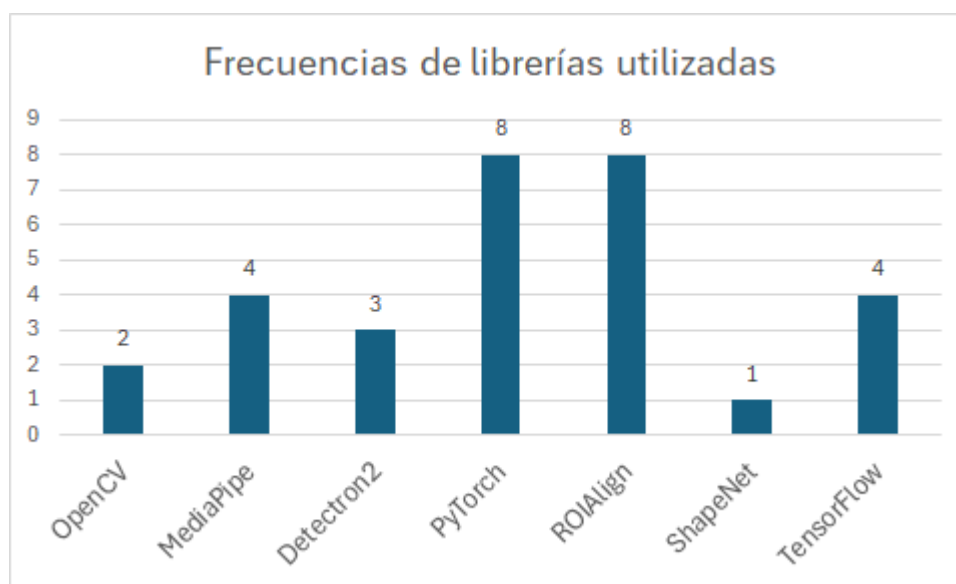


Figura 1. Frecuencia de librerías utilizadas

En la **Figura 2**. Frecuencia de lenguajes de programación utilizados se presenta un gráfico de barras que muestra la frecuencia con la que se utilizaron distintos lenguajes de programación en los antecedentes revisados. Python destaca de forma significativa con doce menciones, lo que refleja su amplio uso en proyectos vinculados al desarrollo de herramientas basadas en inteligencia artificial, procesamiento de imágenes y

visualización de datos. C++ aparece en dos casos, asociado generalmente a tareas que requieren alto rendimiento computacional. Por su parte, C# y JavaScript registran una sola mención cada uno, lo que indica un uso mucho más limitado. Esta distribución sugiere una clara preferencia por Python debido a su versatilidad, simplicidad y compatibilidad con diversas librerías especializadas.

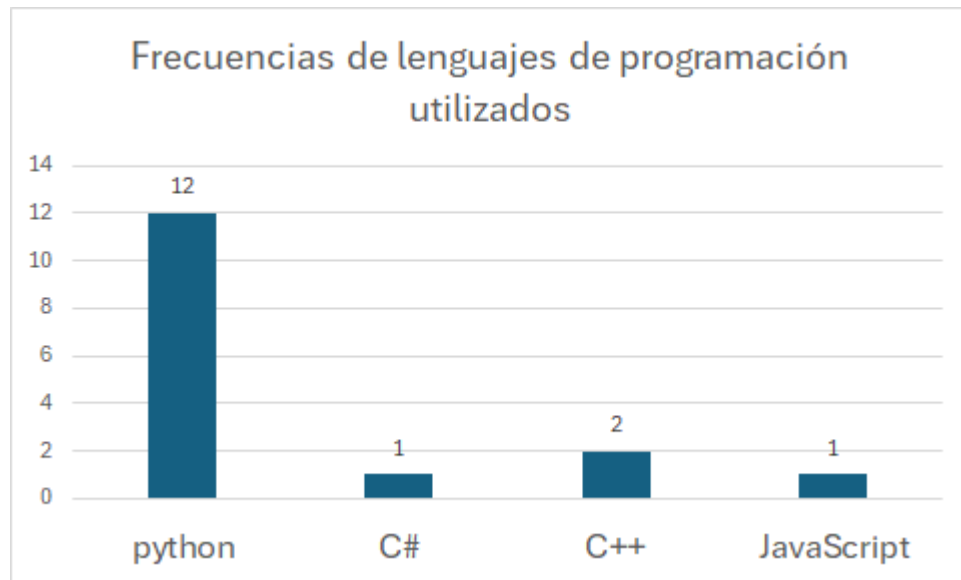


Figura 2. Frecuencia de lenguajes de programación utilizados

1.3 Fundamentación teórica

1.3.1 Representación visual de personas en formato bidimensional

La representación visual de personas en formato bidimensional es un recurso ampliamente utilizado en materiales educativos, presentaciones y medios digitales. Este proyecto adopto como base estética el estilo conocido como Corporate Memphis, caracterizado por el uso de formas geométricas simples, colores planos y proporciones exageradas. Este tipo de ilustración permite representar figuras humanas de manera simplificada pero reconocible, lo que facilita su uso en contextos visuales sin necesidad de un alto grado de detalle. Además, su estructura modular y clara lo hace ideal para ser manipulado dentro de interfaces gráficas interactivas.

1.3.2 Lenguaje de programación

El desarrollo de aplicaciones interactivas requiere el uso de un lenguaje de programación que permita estructurar de forma clara la lógica del sistema y gestionar la interacción con diferentes librerías y recursos. En este proyecto se empleará un lenguaje orientado a objetos que facilita la manipulación de gráficos, el control de eventos y la integración con librerías de inteligencia artificial [9].

1.3.3 Librerías tecnológicas

Las librerías tecnológicas son colecciones de funciones, clases y herramientas preconstruidas que permiten realizar tareas complejas sin desarrollar todo desde cero. Estas librerías encapsulan lógica avanzada en módulos reutilizables, y resultan fundamentales en el desarrollo de software moderno. En este proyecto, se emplearán librerías que ofrecen capacidades como la manipulación de gráficos 2D, el manejo de eventos del usuario y la detección automatizada de estructuras visuales. Su integración agiliza el desarrollo, reduce errores y permite centrar el trabajo en los aspectos funcionales de la aplicación[1].

1.3.4 Metodología Kanban

El desarrollo del proyecto seguirá una metodología ágil basada en el enfoque Kanban. Esta metodología organiza el flujo de trabajo mediante un sistema visual de tarjetas que permite distribuir las tareas en distintas etapas: por hacer, en proceso y finalizado. Kanban proporciona flexibilidad, visibilidad y control sobre el progreso del proyecto, y se adapta bien a entornos donde los requerimientos pueden cambiar con el tiempo. Su uso facilita la gestión eficiente del tiempo, la priorización de tareas y la mejora continua durante el ciclo de desarrollo.

1.3.5 Inteligencia artificial

La inteligencia artificial, aplicada al campo de la visión por computadora, permite procesar imágenes y extraer información relevante de ellas mediante modelos de aprendizaje automático. En este proyecto, se utilizarán recursos basados en IA para detectar posturas humanas. Estos modelos pueden identificar la posición de distintas partes del cuerpo humano a partir de una imagen. Esta información se usará para reflejar los movimientos de las imágenes en un modelo gráfico 2D, permitiendo una relación directa entre los datos visuales capturados y la manipulación del modelo dentro de la aplicación [7].

1.3.6 Captura de postura

La captura del estado de un cuerpo humano en una imagen es una técnica utilizada para registrar y analizar la postura de personas en un momento concreto, mediante fotografías o representaciones. En el ámbito digital, este proceso permite identificar la posición de articulaciones y extremidades, lo que hace posible replicar posturas humanas en entornos virtuales. En el marco de este proyecto, la captura de movimiento se aplicará mediante recursos basados en IA, que procesan imágenes y extraen información corporal de forma simple y rápida [6].

1.3.7 Modelo 2D humano

El modelo 2D humano constituye la representación gráfica simplificada de la anatomía de una persona en un entorno bidimensional. Este modelo se compone de segmentos modulares que corresponden a las diferentes partes del cuerpo, como cabeza, tronco, brazos y piernas, permitiendo su manipulación de manera independiente o conjunta. En este proyecto, el modelo 2D humano funcionará como el eje central de la aplicación, ya que sobre él se proyectarán las posturas capturadas y se realizarán las interacciones del usuario [1].

1.4 Objetivos

1.4.1 Objetivo general

Implementar una aplicación para la generación de ilustraciones 2D de figuras humanas mediante recursos basados en Inteligencia Artificial, con el fin de reducir la dificultad y el tiempo requerido para la creación de imágenes por parte de usuarios sin experiencia.

1.4.2 Objetivos específicos

- Analizar el uso de herramientas para la generación de imágenes, identificando las características que facilitan su utilización y simplifican el proceso de creación de ilustraciones para usuarios sin experiencia.
- Definir los requerimientos funcionales de la aplicación para la mejora de la usabilidad en la generación de ilustraciones 2D de figuras humanas.
- Desarrollar una aplicación que integre las tecnologías adecuadas para la generación de ilustraciones 2D de figuras humanas mediante recursos basados en inteligencia artificial.

CAPÍTULO II. METODOLOGÍA

2.1 Materiales

Para la recolección de información del proyecto se aplicaron encuestas a estudiantes de las distintas facultades de la Universidad Técnica de Ambato, con el propósito de evaluar la viabilidad de la propuesta y recopilar datos relevantes para la definición de los requerimientos en relación con la generación de ilustraciones 2D de figuras humanas mediante recursos basados en inteligencia artificial; adicionalmente, se llevaron a cabo pruebas de tiempo de elaboración de ilustraciones 2D con una herramienta ya existente, aplicadas a una submuestra, con el fin de registrar y analizar los tiempos de creación, lo que, en conjunto con el método cuantitativo inicial, permitió determinar el nivel de aceptación del proyecto y obtener información clave para su desarrollo. El formato de la encuesta se encuentra en el **Anexo A**. Encuesta para la evaluación de la viabilidad y aceptación

2.1.1 Modalidad de la investigación

El desarrollo de la investigación será utilizando un enfoque cuantitativo, que se caracteriza por el análisis numérico y estadístico de los datos recolectados. Este enfoque permite medir y evaluar variables específicas de forma objetiva y precisa, como la dificultad percibida por los usuarios y el tiempo requerido para crear ilustraciones 2D. De esta manera, se pueden establecer relaciones claras entre las variables y valorar la eficacia de la aplicación mediante indicadores medibles y comparables.

- ***Modalidad aplicada***

La modalidad aplicada consiste en la utilización de herramientas y recursos tecnológicos ya existentes para adaptarlos y ponerlos en práctica, de modo que puedan ser probados directamente por los usuarios. Esta modalidad busca resolver problemas concretos mediante soluciones funcionales que se implementan y evalúan en contextos

reales, facilitando así la creación de ilustraciones 2D accesibles para personas sin experiencia técnica.

- ***Modalidad bibliográfica***

La modalidad bibliográfica consiste en la revisión y análisis de fuentes documentales relevantes para sustentar teóricamente los conceptos clave del proyecto. Esta revisión incluye libros, artículos científicos, manuales técnicos y documentación oficial sobre ilustración digital, inteligencia artificial, interacción humano-computador y metodologías de desarrollo. La modalidad bibliográfica permite contextualizar la propuesta dentro del marco teórico existente, identificar antecedentes y orientar el diseño de la solución desde fundamentos previamente establecidos por la literatura especializada.

2.1.2 Población y muestra

Para el desarrollo de este proyecto de titulación se definió la población de la Universidad Técnica de Ambato, que está conformada por 18.000 estudiantes. Con el propósito de establecer la viabilidad del proyecto y determinar los requerimientos específicos, se decidió aplicar encuestas a una muestra representativa de dicha población. Adicionalmente, se seleccionó una submuestra dentro de la muestra general, a la cual se le aplicaron pruebas orientadas a registrar los tiempos de creación de ilustraciones 2D de figuras humanas mediante herramientas tradicionales. El cálculo del tamaño de la muestra se realizó empleando la fórmula siguiente:

Fórmula del tamaño de muestra para poblaciones finitas:

$$n = \frac{N * Z^2 * p * q}{Z^2 * p * q + (N - 1) * e^2}$$

- n: Tamaño de la muestra necesario.
- N: Tamaño de la población total. En este caso, N=18000.

- Z: Valor crítico de la distribución normal correspondiente al nivel de confianza deseado. Para un nivel de confianza del 95%, $Z = 1,96$.
- p: Probabilidad de éxito. Se utiliza $p = 0.5$.
- q: Probabilidad de fracaso, complementaria de p. Por lo tanto, $q=0,5$.
- e: Margen de error deseado. Se utiliza $e = 0.05$.

Sustituyendo los valores en la fórmula:

$$n = \frac{(18000) * (1,96)^2 * (0.5) * (0.5)}{(1,96^2) * (0.5) * (0.5) + (18000 - 1) * (0.05)^2}$$

Resolviendo paso a paso:

$$Z^2 = (1.96)^2 = 3.8416$$

$$p * q = 0.5 * 0.5 = 0.25$$

$$Z^2 * p * q = (3.8416) * (0.25) = 0.9604$$

$$(N - 1) * e^2 = (18000 - 1) * (0.05)^2 = (17999)(0.0025) = 44.9975$$

Sustituyendo estos valores:

$$n = \frac{18000 * 0.9604}{0.9604 + 44.9975} = \frac{17287.2}{45.9579} = 376.15$$

Por lo tanto, el tamaño de la muestra necesario es aproximadamente:

$$n \approx 377$$

Cálculo de la Submuestra:

$$n = \frac{N * Z^2 * p * q}{Z^2 * p * q + (N - 1) * e^2}$$

- n: Tamaño de la muestra necesario.

- N: Tamaño de la población total. En este caso, N=377.
- Z: Valor crítico de la distribución normal correspondiente al nivel de confianza deseado. Para un nivel de confianza del 95%, $Z = 1,96$.
- p: Probabilidad de éxito. Se utiliza $p = 0.1$.
- q: Probabilidad de fracaso, complementaria de p. Por lo tanto, $q=0,9$.
- e: Margen de error deseado. Se utiliza $e = 0.1$.

Sustituyendo los valores en la fórmula:

$$n = \frac{(377) * (1,96)^2 * (0.1) * (0.9)}{(1,96^2) * (0.1) * (0.9) + (377 - 1) * (0.1)^2}$$

Resolviendo paso a paso:

$$Z^2 = (1.96)^2 = 3.8416$$

$$p * q = 0.5 * 0.5 = 0.25$$

$$Z^2 * p * q = (3.8416) * (0.25) = 0.9604$$

$$(N - 1) * e^2 = (18000 - 1) * (0.05)^2 = (17999)(0.0025) = 44.9975$$

Sustituyendo estos valores:

$$n = \frac{18000 * 0.9604}{0.9604 + 44.9975} = \frac{17287.2}{45.9579} = 31.7471$$

Por lo tanto, el tamaño de la muestra necesario es aproximadamente:

$$n \approx 32$$

2.1.3 Recolección de información

La información se recolectó mediante la aplicación de encuestas a los estudiantes de la Universidad Técnica de Ambato, quienes representan la población objetivo del proyecto. Estas encuestas tuvieron como finalidad determinar la viabilidad del proyecto y recopilar los requerimientos necesarios para el desarrollo del sistema de generación de ilustraciones 2D de figuras humanas con inteligencia artificial. Adicionalmente, a una submuestra seleccionada se le aplicaron pruebas de tiempo de creación de ilustraciones 2D utilizando herramientas tradicionales, con el propósito de obtener datos concretos sobre los procesos actuales y establecer parámetros de comparación. En el **Anexo B**, se presenta la fórmula empleada para el cálculo del tamaño de la muestra.

a. Validación del instrumento

La validación del instrumento de recolección de datos se llevó a cabo mediante una encuesta con una escala tipo Likert de 5 niveles, que abarcó desde la menor frecuencia o dificultad percibida hasta la mayor frecuencia o utilidad reconocida. El cuestionario estuvo compuesto por 14 ítems orientados a evaluar el conocimiento y uso de herramientas de edición de imagen, la percepción de dificultad al crear ilustraciones de figuras humanas y la utilidad de diferentes funcionalidades dentro de una aplicación de fácil acceso. La encuesta se encuentra en el Anexo A

Para comprobar la consistencia interna de la encuesta, se aplicó el coeficiente alfa de Cronbach, se procesaron las respuestas obtenidas en los ítems del cuestionario. El análisis estadístico arrojó un coeficiente alfa de Cronbach de 0,812, lo que, según la escala de interpretación del coeficiente, indica que el instrumento posee una adecuada consistencia interna y un nivel alto de confiabilidad. El cálculo de este coeficiente se encuentra en el Anexo B

Para la validación de la encuesta antes de su aplicación, se realizó una revisión por parte de docentes, utilizando una escala de Likert diseñada para evaluar los ítems de la encuesta. Esta evaluación permitió calificar tres dimensiones específicas de cada pregunta: claridad, consistencia e importancia, en relación con la generación de

ilustraciones 2D de figuras humanas. La información obtenida a través de esta herramienta proporcionó datos relevantes para efectuar ajustes a la formulación de las preguntas de la encuesta.

Tabla 1. Escala de Likert para evaluación de preguntas

Criterios	Muy baja	Baja	Medio	Alta	Muy Alta
Claridad	1	2	3	4	5
Consistencia	1	2	3	4	5
Importancia	1	2	3	4	5

Previo a la aplicación de la encuesta, se llevó a cabo una revisión de la encuesta. La información recabada se analizó a través de una escala de Likert, la cual evaluó la claridad, consistencia e importancia de cada pregunta de la encuesta. Los resultados de esta evaluación permitieron calcular los promedios para cada una de las dimensiones mencionadas, los cuales se presentan en la tabla siguiente:

Tabla 2. Promedio de los criterios evaluados en la encuesta de Likert

Pregunta	Claridad	Consistencia	Importancia
P1	4.3	4.5	4.5
P2	4.6	4.5	4.4
P3	4.2	4.6	4.4
P4	4.4	4.3	4.5
P5	4.3	4.4	4.6
P6	4.4	4.2	4.3
P7	4.6	4.5	4.8
P8	4.8	4.3	4.4
P9	4.3	4.2	4.3
P10	4.2	4.6	4.4
P11	4.5	4.3	4.4
P12	4.4	4.6	4.5
P13	4.5	4.4	4.6
P14	4.6	4.5	4.5

El análisis realizado muestra que los ítems de la encuesta tienen altos niveles de claridad, consistencia e importancia, con promedios que generalmente oscilan entre 4.2 y 4.8. Estos resultados sugieren que las preguntas son comprensibles, coherentes y

relevantes en relación con el tema de la generación de ilustraciones 2D de figuras humanas. La información recopilada ofrece una base sólida para ajustar y mejorar la formulación de los ítems, asegurando que la encuesta sea clara y efectiva antes de aplicarla a los estudiantes.

b. Resultados de las encuestas aplicadas a los estudiantes de la Universidad Técnica de Ambato.

Pregunta 1: ¿Con qué frecuencia utiliza herramientas de edición de imagen para tareas o presentaciones?

Opciones de la encuesta en la pregunta 1: A) Nunca, B) Rara vez, C) A veces, D) Casi siempre, E) Muy frecuentemente

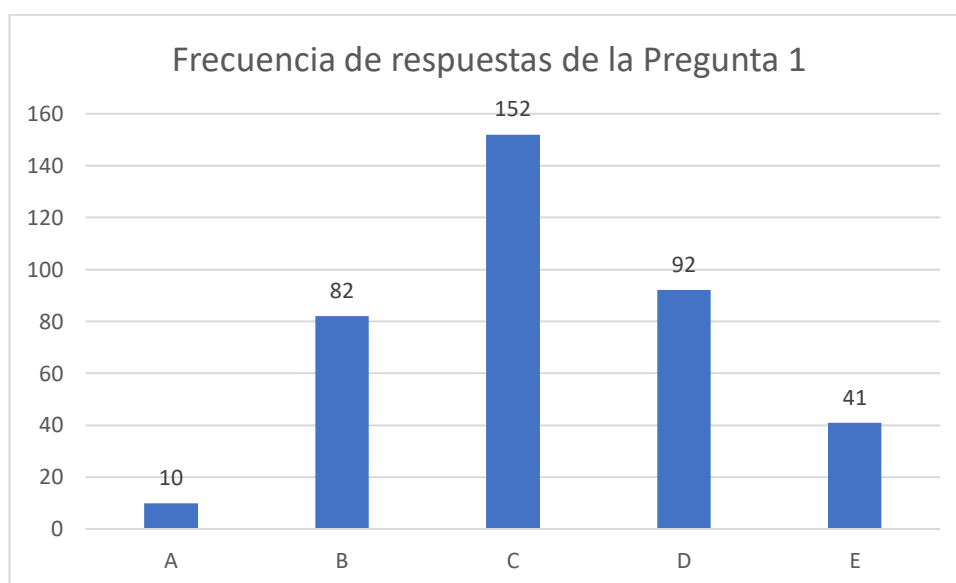


Figura 3. Resultados de la pregunta 1

Análisis:

La mayoría de los estudiantes utiliza herramientas de edición de imagen de manera moderada, eligiendo principalmente las opciones “A veces” y “Casi siempre”. Una cantidad menor de estudiantes reporta un uso bajo, con “Rara vez” o “Nunca”, mientras que solo algunos las utilizan con alta frecuencia, seleccionando “Muy

frecuentemente”. En conjunto, los datos muestran que el uso de estas herramientas es común entre los estudiantes, aunque no todos dependen de ellas de manera constante.

Pregunta 2: ¿Conoce sobre herramientas que permiten crear personajes o figuras humanas en 2D?

Opciones de la encuesta en la pregunta 2: A) Nada, B) Poco, C) Algo, D) Bastante, E) Mucho

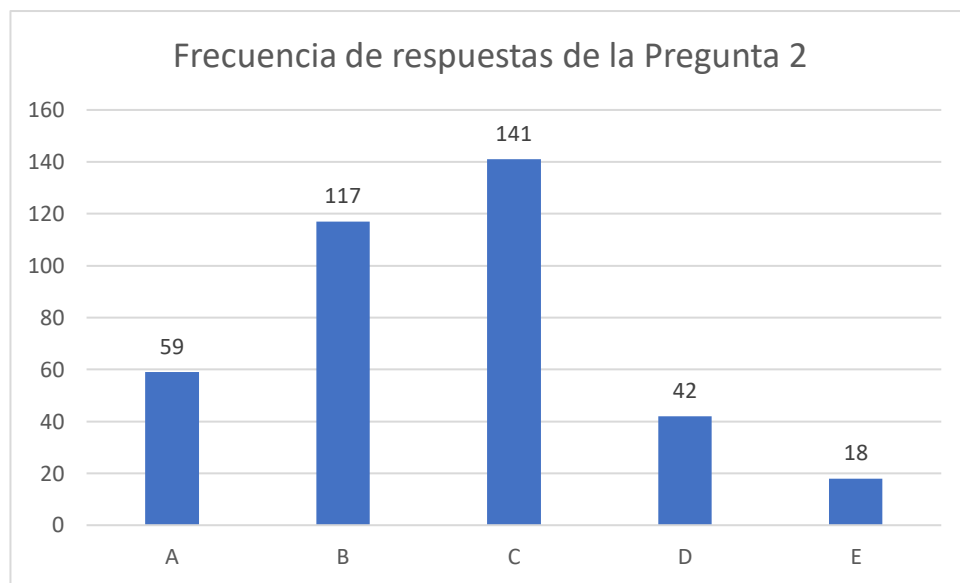


Figura 4. Resultados de la pregunta 2

Análisis:

La mayoría de los estudiantes posee un conocimiento limitado sobre herramientas para crear personajes o figuras humanas en 2D, concentrándose principalmente en las opciones “Poco” y “Algo”. Un grupo menor indica tener un conocimiento más avanzado, seleccionando “Bastante” o “Mucho”, mientras que una parte de los estudiantes no tiene conocimiento sobre estas herramientas, eligiendo “Nada”. En conjunto, los resultados muestran que, aunque algunos estudiantes cuentan con cierta familiaridad, la comprensión general sobre este tipo de herramientas es todavía baja.

Pregunta 3: ¿Qué tan familiarizado está con herramientas como PowerPoint, Canva, Photoshop u otras similares?

Opciones de la encuesta en la pregunta 3: A) Nada familiarizado, B) Poco familiarizado, C) Medianamente familiarizado, D) Bastante familiarizado, E) Muy familiarizado

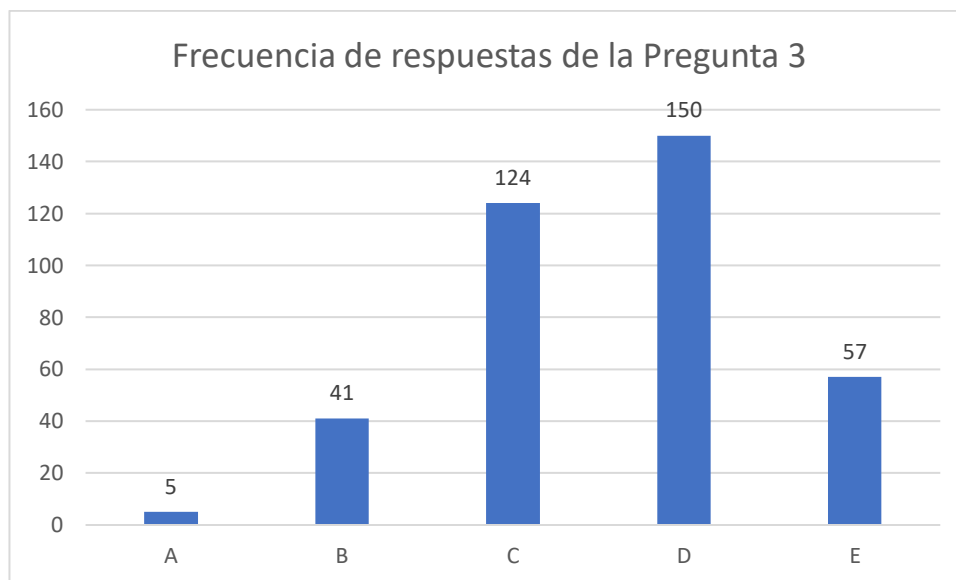


Figura 5. Resultados de la pregunta 3

Análisis:

La mayoría de los estudiantes muestra un nivel moderado a alto de familiaridad con las herramientas mencionadas, concentrándose principalmente en las opciones “Medianamente familiarizado” y “Bastante familiarizado”. Un grupo menor indica tener un conocimiento muy limitado, seleccionando “Nada familiarizado” o “Poco familiarizado”, mientras que algunos estudiantes se consideran “Muy familiarizados”. En conjunto, los datos reflejan que la mayoría de los estudiantes posee experiencia suficiente para utilizar estas herramientas en sus tareas y presentaciones.

Pregunta 4: ¿Con qué frecuencia has tenido dificultades para crear ilustraciones gráficas en herramientas de edición de imagen?

Opciones de la encuesta en la pregunta 4: A) Nunca, B) Rara vez, C) A veces, D) Casi siempre, E) Siempre

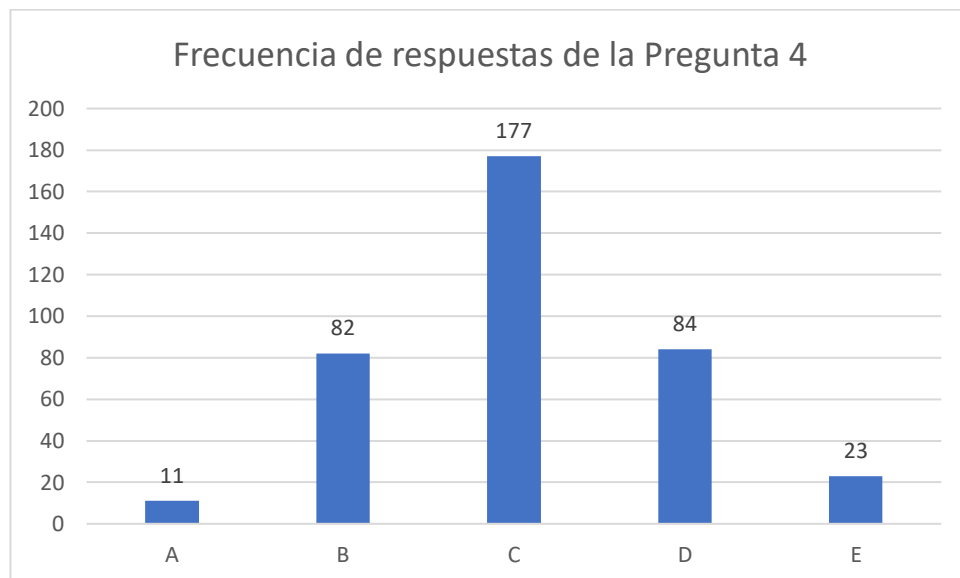


Figura 6. Resultados de la pregunta 4

Análisis:

La mayoría de los estudiantes experimenta dificultades ocasionales al crear ilustraciones gráficas en herramientas de edición de imagen, eligiendo principalmente las opciones “A veces” y “Casi siempre”. Un grupo menor reporta enfrentar problemas con poca frecuencia, seleccionando “Nunca” o “Rara vez”, mientras que solo algunos indican tener dificultades constantes, eligiendo “Siempre”. En conjunto, los resultados muestran que, aunque no todos los estudiantes tienen problemas regularmente, la creación de ilustraciones gráficas puede presentar desafíos para una parte significativa.

Pregunta 5: ¿Qué nivel de dificultad percibe al usar herramientas de edición de imagen tradicionales sin experiencia previa?

Opciones de la encuesta en la pregunta 5: A) Muy bajo, B) Bajo, C) Medio, D) Alto, E) Muy alto

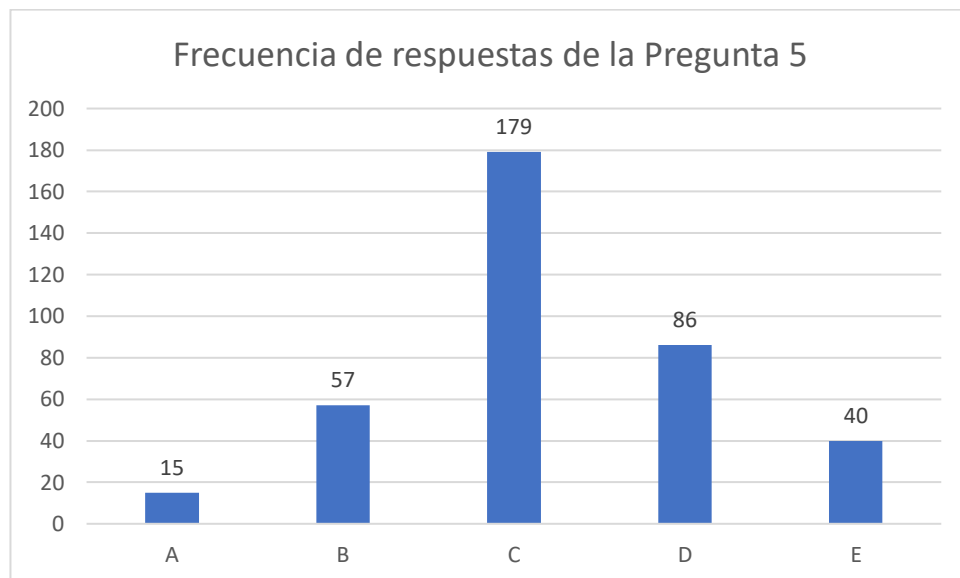


Figura 7. Resultados de la pregunta 5

Análisis:

La mayoría de los estudiantes percibe un nivel medio de dificultad al utilizar herramientas de edición de imagen tradicionales sin experiencia previa, reflejado principalmente en la opción “Medio”. Un grupo menor considera que la dificultad es baja, eligiendo “Muy bajo” o “Bajo”, mientras que algunos estudiantes perciben un nivel alto o muy alto de dificultad. En conjunto, los resultados indican que, aunque la mayoría encuentra las herramientas manejables, hay una tendencia que apunta a que de forma general hay más quienes tienen dificultades.

Pregunta 6: ¿Qué tan fácil le resulta actualmente crear una ilustración de personajes humanos usando herramientas que conoce?

Opciones de la encuesta en la pregunta 6: A) Muy difícil, B) Difícil, C) Medio, D) Fácil, E) Muy fácil

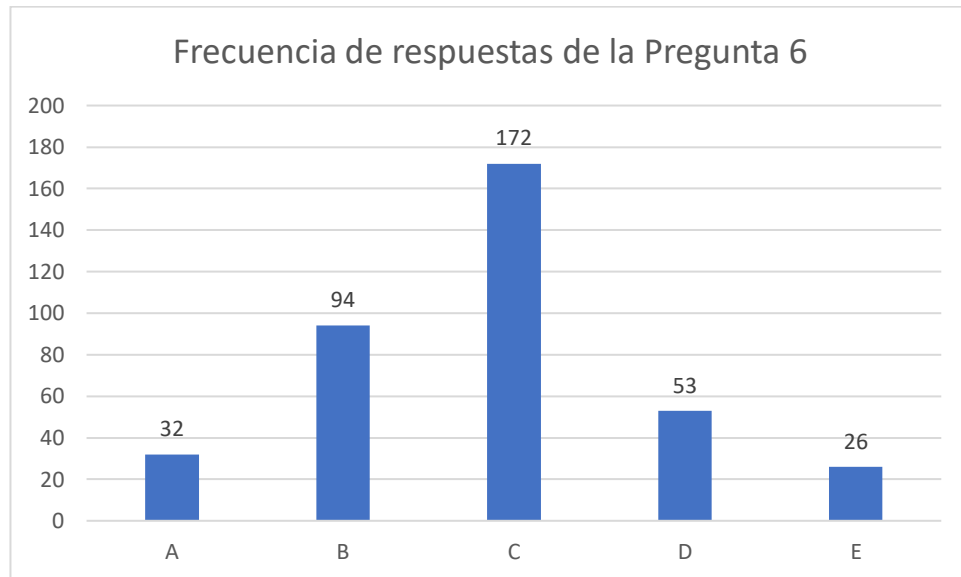


Figura 8. Resultados de la pregunta 6

Análisis:

La mayoría de los estudiantes considera que crear ilustraciones de personajes humanos con las herramientas que conocen resulta de dificultad media, reflejado principalmente en la opción “Medio”. Un grupo menor percibe el proceso como difícil o muy difícil, mientras que algunos estudiantes lo encuentran fácil o muy fácil. En conjunto, los resultados muestran que, aunque la experiencia actual permite cierto manejo de las herramientas, la creación de ilustraciones sigue representando un reto para muchos estudiantes.

Pregunta 7: ¿Cuánto tiempo estima que te toma normalmente crear una ilustración gráfica sencilla de personajes humanos?

Opciones de la encuesta en la pregunta 7: A) Menos de 10 minutos, B) Entre 10 y 20 minutos, C) Entre 20 y 40 minutos, D) Entre 40 minutos y 1 hora, E) Más de 1 hora

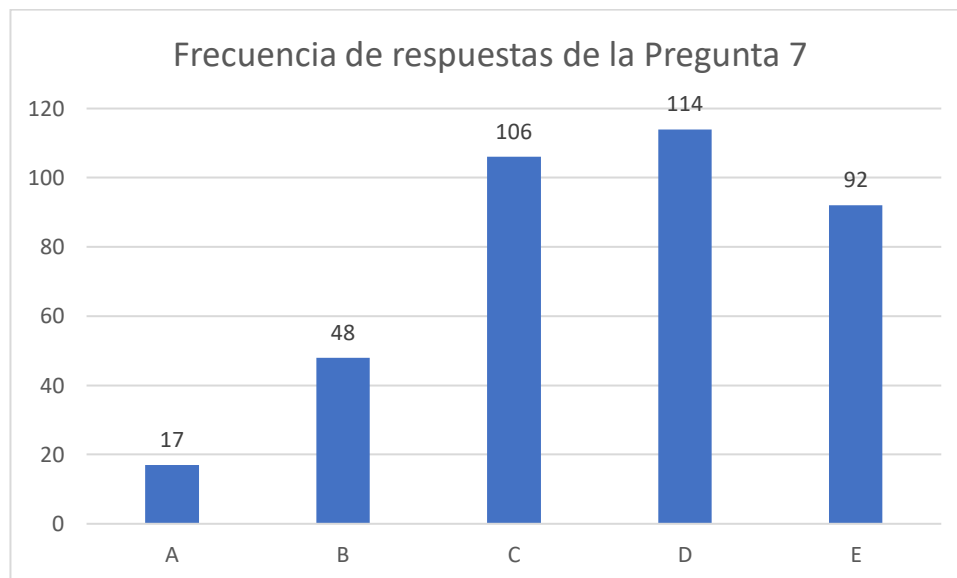


Figura 9. Resultados de la pregunta 7

Análisis:

La mayoría de los estudiantes estima que crear una ilustración gráfica sencilla de personajes humanos les toma entre 20 minutos y 1 hora, concentrándose principalmente en las opciones “Entre 20 y 40 minutos” y “Entre 40 minutos y 1 hora”. Un grupo menor indica que tarda menos de 20 minutos, mientras que otros reportan que el proceso supera la hora. En conjunto, los resultados muestran que la creación de ilustraciones requiere un tiempo considerable para la mayoría de los estudiantes, evidenciando que no es un proceso rápido con las herramientas que utilizan actualmente.

Pregunta 8: ¿Qué tan probable es que utilice una herramienta que le facilite crear ilustraciones didácticas de personajes humanos sin necesidad de conocimientos técnicos?

Opciones de la encuesta en la pregunta 8: A) Nada probable, B) Poco probable, C) Neutral, D) Probable, E) Muy probable

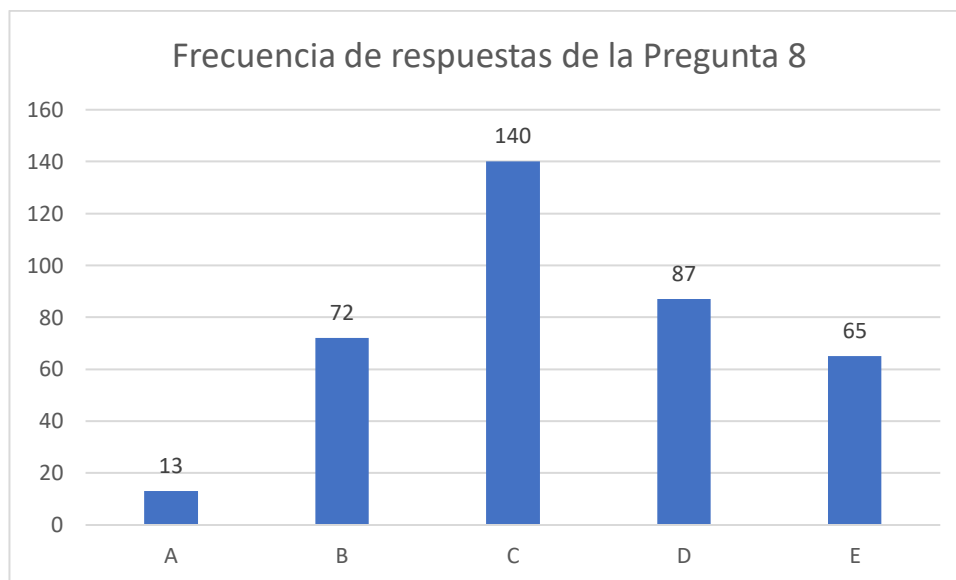


Figura 10. Resultados de la pregunta 8

Análisis:

La mayoría de los estudiantes se muestra neutral o favorable respecto al uso de una herramienta que facilite la creación de ilustraciones didácticas de personajes humanos sin necesidad de conocimientos técnicos, concentrándose principalmente en las opciones “Neutral” y “Probable”. Un grupo menor considera poco probable o nada probable su uso, mientras que algunos estudiantes lo perciben como muy probable. En conjunto, los resultados indican que existe un interés general en contar con herramientas que simplifiquen la creación de ilustraciones.

Pregunta 9: ¿Qué tan importante le sería que la aplicación tenga una interfaz simple y fácil de entender desde el primer uso?

Opciones de la encuesta en la pregunta 9: A) Nada importante, B) Poco importante, C) Neutral, D) Importante, E) Muy importante

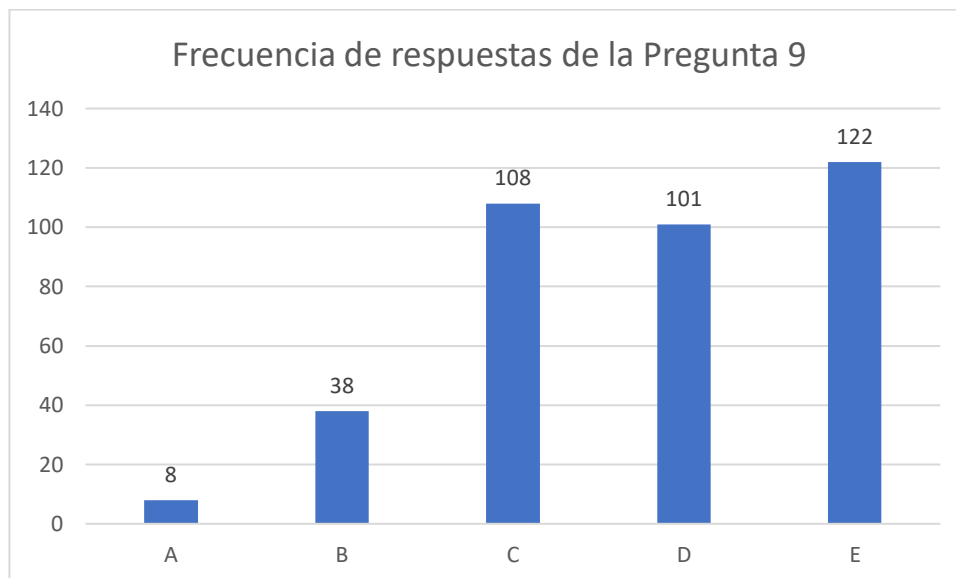


Figura 11. Resultados de la pregunta 9

Análisis:

La mayoría de los estudiantes considera importante o muy importante que la aplicación cuente con una interfaz simple y fácil de entender desde el primer uso, concentrándose principalmente en las opciones “Importante” y “Muy importante”. Un grupo menor mantiene una postura neutral, mientras que pocos consideran que esto sea poco o nada importante. En conjunto, los resultados reflejan que la facilidad de uso es un factor relevante para los estudiantes al momento de interactuar con la aplicación.

Pregunta 10: ¿Qué tan útil le sería que puedas mover partes del cuerpo de una figura humana directamente con el mouse (por ejemplo, brazos o piernas)?

Opciones de la encuesta en la pregunta 10: A) Nada útil, B) Poco útil, C) Neutral, D) Útil, E) Muy útil

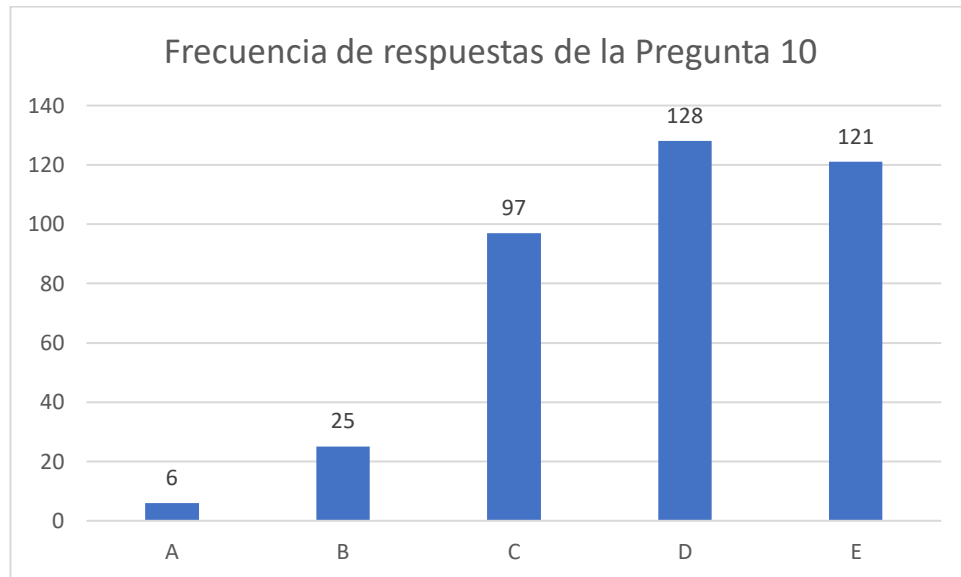


Figura 12. Resultados de la pregunta 10

Análisis:

La mayoría de los estudiantes considera útil o muy útil la posibilidad de mover partes del cuerpo de una figura humana directamente con el mouse, concentrándose principalmente en las opciones “Útil” y “Muy útil”. Un grupo menor mantiene una postura neutral, mientras que pocos consideran esta función poco o nada útil. En conjunto, los resultados muestran que la interacción directa con las figuras sería una característica apreciada y relevante para los estudiantes al crear ilustraciones.

Pregunta 11: ¿Qué tipo de retroalimentación visual le gustaría recibir al interactuar con la figura (por ejemplo, al seleccionar una parte del cuerpo)?

Opciones de la encuesta en la pregunta 11: A) No necesito retroalimentación, B) Cambio de color, C) Contorno resaltado, D) Etiqueta con nombre de la parte, E) Sonido

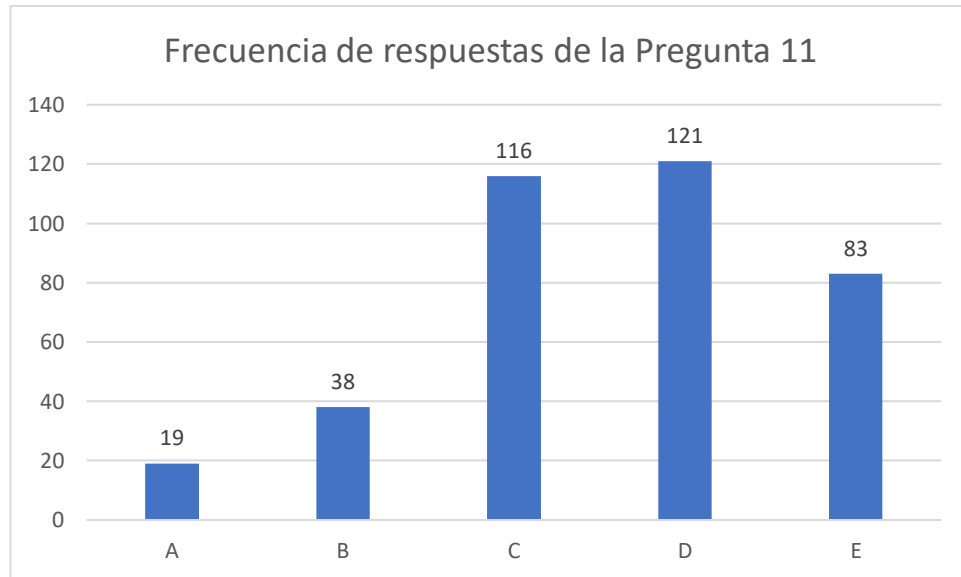


Figura 13. Resultados de la pregunta 11

Análisis:

La mayoría de los estudiantes prefiere recibir retroalimentación visual al interactuar con la figura, destacando principalmente las opciones “Etiqueta con nombre de la parte” y “Contorno resaltado”. Un grupo menor considera suficiente un cambio de color o un sonido, mientras que pocos indican que no necesitan retroalimentación. En conjunto, los resultados muestran que proporcionar señales visuales claras al seleccionar partes de la figura sería una característica valorada y útil para mejorar la interacción con la herramienta.

Pregunta 12: ¿Qué tan útil le sería que la aplicación permita reiniciar el proceso en cualquier momento?

Opciones de la encuesta en la pregunta 12: A) Nada útil, B) Poco útil, C) Neutral, D) Útil, E) Muy útil

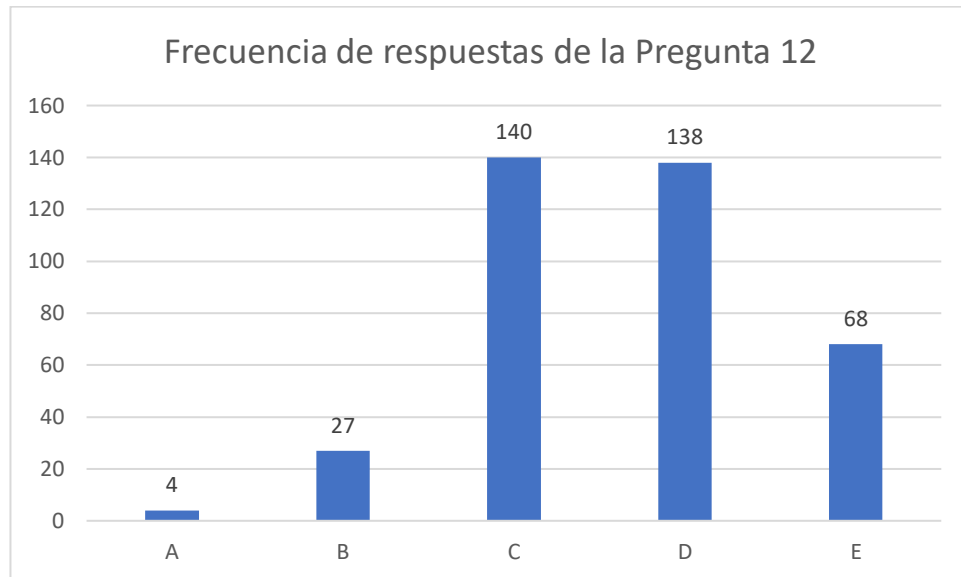


Figura 14. Resultados de la pregunta 12

Análisis:

La mayoría de los estudiantes considera que sería útil que la aplicación permita reiniciar el proceso en cualquier momento, concentrándose principalmente en las opciones “Útil” y “Neutral”. Un grupo menor percibe esta función como muy útil, mientras que pocos la consideran poco o nada útil. En conjunto, los resultados indican que la posibilidad de reiniciar el proceso es una característica valorada por los estudiantes, aunque no todos la perciben como esencial.

Pregunta 13: ¿Qué tan útil le sería que la aplicación permita replicar una postura a partir de una fotografía?

Opciones de la encuesta en la pregunta 13: A) Nada útil, B) Poco útil, C) Neutral, D) Útil, E) Muy útil

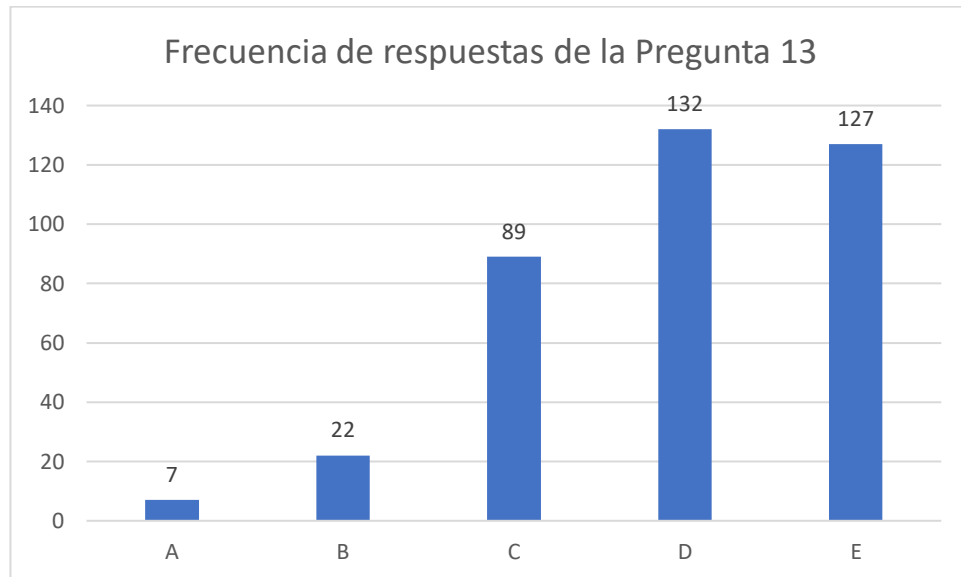


Figura 15. Resultados de la pregunta 13

Análisis:

La mayoría de los estudiantes considera útil o muy útil que la aplicación permita replicar una postura a partir de una fotografía, concentrándose principalmente en las opciones “Útil” y “Muy útil”. Un grupo menor mantiene una postura neutral, mientras que pocos la consideran poco o nada útil. En conjunto, los resultados muestran que esta función sería apreciada y considerada valiosa para facilitar la creación de ilustraciones de figuras humanas.

Pregunta 14: ¿Una herramienta como esa debería incluir instrucciones o ayudas contextuales al usar cada herramienta?

Opciones de la encuesta en la pregunta 14: A) No, B) Poco, C) Neutral, D) Mucho, E) Sí

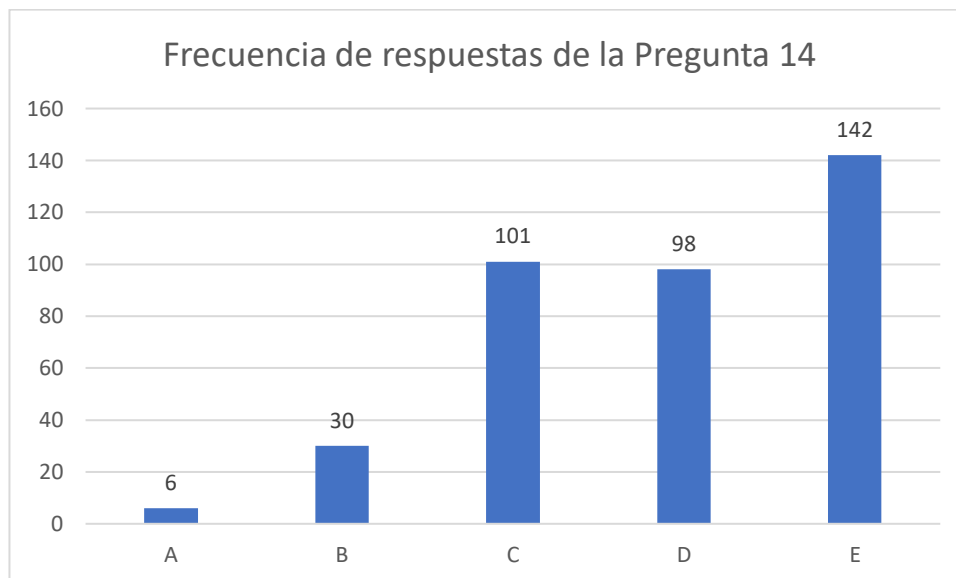


Figura 16. Resultados de la pregunta 14

Análisis:

La mayoría de los estudiantes considera que la herramienta debería incluir instrucciones o ayudas contextuales al utilizar cada función, concentrándose principalmente en las opciones “Sí” y “Mucho”. Un grupo menor mantiene una postura neutral, mientras que pocos consideran que estas ayudas no son necesarias o son poco relevantes. En conjunto, los resultados indican que contar con instrucciones o asistencia contextual sería una característica valorada para facilitar el uso de la aplicación.

2.1.4 Procesamiento y análisis de datos

Los datos obtenidos en las encuestas fueron organizados y procesados para calcular frecuencias y porcentajes de cada respuesta, permitiendo identificar tendencias en el uso y percepción de las herramientas de edición de imágenes. El análisis evidenció que en torno al 75% de estudiantes utiliza estas herramientas con frecuencia media con tendencia a moderada, pero reporta un conocimiento limitado sobre la creación de personajes en 2D, así como dificultades ocasionales para elaborar ilustraciones.



Figura 17. Polígono de frecuencias de creación de ilustraciones con Krita en minutos.

La **Figura 17**. Polígono de frecuencias de creación de ilustraciones con Krita presenta los resultados detallados de los tiempos de creación de ilustraciones, indicando para cada participante el número de minutos empleados y el porcentaje de frecuencia en la curva de distribución. El tiempo promedio registrado fue de 6,24 minutos con una desviación estándar de 1,45 minutos, concentrándose la mayor parte de los estudiantes en valores intermedios (alrededor de 6 a 7 minutos). Solo unos pocos alcanzaron tiempos muy bajos (entre 3 y 4 minutos) o muy altos (superiores a 8 minutos), lo que evidencia una distribución cercana a la normal y refuerza la necesidad de una herramienta que optimice el proceso de creación.

CAPÍTULO III. RESULTADOS Y DISCUSIÓN

3.1 Análisis de herramientas

Para el desarrollo del proyecto se tuvo que seleccionar un conjunto de herramientas con compatibilidad y facilidad de implementación entre sí. El análisis de estas herramientas se basó en antecedentes, los recursos disponibles y la integración con librerías. De esta manera, se definieron las herramientas más adecuadas para cumplir con los objetivos del proyecto.

3.1.1 Análisis del uso de herramientas para generación de imágenes

En esta sección se establecen las opciones actuales de creación de ilustraciones humanas en 2D, las cuales se componen de un conjunto de características que permiten evaluar la complejidad percibida por los usuarios y coordinadores académicos, contrastando dichas características entre las distintas herramientas disponibles.

Tabla 3. Comparativa cualitativa de herramientas de edición tradicionales.

Herramienta / Software	Adobe Photoshop	Krita	GIMP	Stable Diffusion	MidJourney	Canva
Tipo de herramienta	Editor gráfico profesional	Software de ilustración digital	Editor de imágenes	Generador de imágenes IA (texto a imagen)	Generador de imágenes IA (texto a imagen)	Diseño gráfico en línea
Facilidad de uso (para principiantes)	Media-Baja (interfaz compleja)	Media (intuitivo para dibujo)	Media	Alta (interfaz web o local sencilla)	Alta (a través de Discord)	Muy alta
Requiere conocimientos de diseño	Sí, alto	Sí, medio	Sí, medio	No	No	No

Capacidades de generación automática	Limitada (<i>Generative Fill</i> , <i>Firefly</i>)	No	No	Sí – creación a partir de <i>prompts</i>	Sí – creación inmediata por texto	Sí – <i>Magic Design</i> , <i>Text to Image</i>
Permite editar libremente una imagen generada	Sí – edición avanzada con capas y herramientas precisas	Sí – edición completa pero manual	Sí – edición completa, pero requiere práctica	Parcial – requiere configuraciones	No directo – solo variaciones	Sí – editor integrado muy simple
Capacidad de procesamiento y peso de la aplicación	Alta; instalación ~4 GB	Media; instalación ~300 MB	Media-Baja; instalación ~200 MB	Muy alta; instalación de modelos >4 GB	Baja local; solo requiere navegador/Discord; sin instalación	Baja local ; app ligera en móvil (~100 MB)
Licencia / Costo	De pago (suscripción)	Gratuito / Código abierto	Gratuito / Código abierto	Gratuito (open source)	De pago (planes mensuales)	Freemium
Soporte de IA	Parcial (IA en versiones recientes)	No	No	Sí, IA generativa	Sí	Sí

Las distintas herramientas analizadas presentan tendencias diferenciadas según su enfoque principal. Aquellas orientadas a la edición de imágenes requieren un nivel de conocimiento de medio a alto para ser utilizadas de manera adecuada, ya que ofrecen amplias capacidades para modificar e incluso intervenir elementos generados. No obstante, esta flexibilidad implica una mayor demanda de procesamiento y un mayor espacio de almacenamiento, lo que puede limitar su uso en equipos con recursos limitados.

Mientras que, las herramientas centradas en la generación automática de imágenes suelen ser ligeras y operan de manera remota en un servidor, reduciendo la necesidad de un hardware potente. No obstante, su funcionamiento limita la edición directa de las imágenes generadas haciendo que cualquier modificación generalmente requiere nuevas instrucciones o ajustes en los parámetros, lo que puede producir resultados diferentes a los esperados.

En cuanto a la creación de ilustraciones 2D de figuras humanas, las herramientas tradicionales facilitan el trabajo del usuario al ofrecer un control preciso y constante sobre cada parte de la composición, permitiendo editar cualquier elemento en cualquier momento. Por su parte, las herramientas que emplean IA simplifican y aceleran significativamente el proceso al automatizar tareas complejas. Por ello, la opción óptima es usar la IA para agilizar el flujo de trabajo sin reducir el control que el usuario tiene sobre los elementos de la ilustración.

Considerando las características funcionales de las herramientas analizadas y la accesibilidad necesaria para el desarrollo del proyecto, **se optó por seleccionar Krita como software de referencia**. Esta elección se basa en sus requerimientos de sistema moderados, que facilitan su ejecución en distintos entornos de hardware sin comprometer el rendimiento, así como una interfaz, que permite realizar procesos de ilustración y edición de manera eficiente. Además, su licencia de código abierto ofrece la posibilidad de explorar y comprender su estructura interna sin restricciones comerciales, lo que permite establecer comparaciones y validar las funcionalidades del software desarrollado durante la fase de pruebas.

3.1.2 Análisis de metodologías de desarrollo

En la actualidad existen múltiples metodologías ágiles que pueden aplicarse en la planificación y ejecución de proyectos de investigación, Estas metodologías definen distintas estrategias para la organización y distribución del trabajo, la comunicación entre los miembros del equipo y la gestión de las entregas parciales del producto. La elección de una metodología está condicionada al tipo de proyecto, los recursos disponibles y la flexibilidad durante su desarrollo.

Para el desarrollo de la aplicación para la generación de ilustraciones 2D de figuras humanas utilizando recursos basados en inteligencia artificial, se decidió usar la metodología **Kanban**. Esta decisión se realizó basándose en las características del proyecto, en el que se tuvieron que realizar procesos de diseño, desarrollo y pruebas, en un periodo de tiempo determinado, lo que requiere una planificación adaptable y

un control visual del progreso. Kanban resulta apropiada porque no impone una estructura rígida de iteraciones ni roles fijos, lo que facilita el ajuste continuo de las tareas conforme evolucionan los requerimientos y resultados. Además, permite una gestión eficiente del flujo de trabajo sin necesidad de procesos complejos de coordinación, algo especialmente útil en entornos académicos donde los tiempos y recursos pueden variar.

El método Kanban tiene sus raíces en los sistemas de producción orientados a la eficiencia y la optimización de recursos. Su adaptación al ámbito del desarrollo de software daba la capacidad de pasar esos principios a la gestión de tareas, priorizando la visualización del trabajo y la mejora continua. Su principal objetivo es mantener un flujo de desarrollo estable, reducir retrasos y garantizar que las tareas se completen de manera ordenada y progresiva. La implementación de Kanban se lleva a cabo mediante tableros visuales que representan el estado de cada tarea a través de columnas como “Pendiente”, “En desarrollo” y “Completada”. Este sistema permite observar de forma directa el desarrollo del proyecto, detectar sobrecargas o bloqueos y establecer prioridades de forma más efectiva. Asimismo, el control del trabajo en curso contribuye a evitar la acumulación de tareas sin finalizar y favorece un ritmo de trabajo equilibrado.

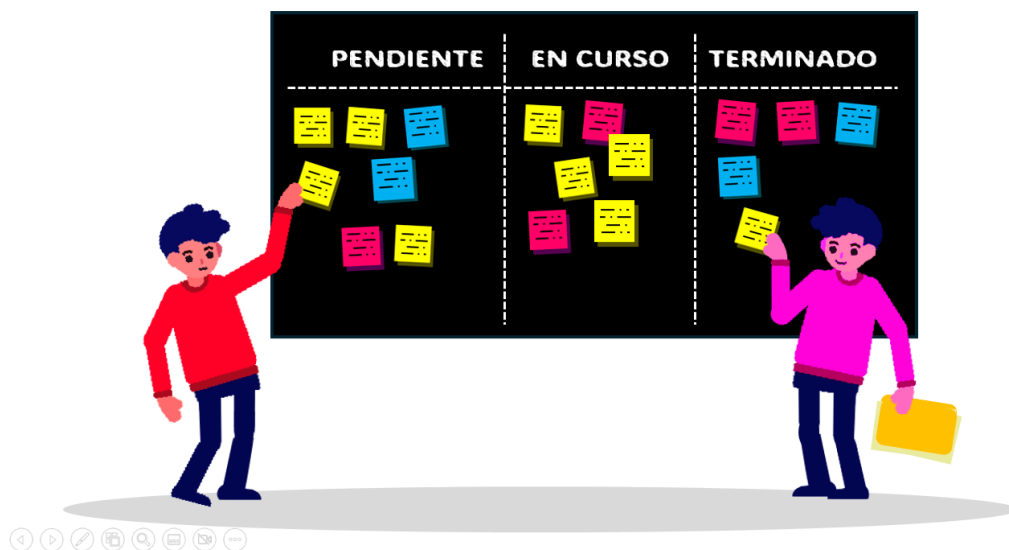


Figura 18. Ilustración del tablero Kanban

Durante el proceso de desarrollo de la aplicación, el uso de Kanban permitió estructurar las etapas principales del proyecto de manera ordenada. En primer lugar, se abordó la recopilación y análisis de los datos, seguido por el diseño de la interfaz y desarrollo del sistema. El uso de esta metodología facilitó la gestión visual del avance del trabajo y permitió realizar ajustes oportunos por los ajustes en los tiempos de desarrollo. La flexibilidad del enfoque Kanban fue muy importante para mantener la coherencia entre los objetivos iniciales y los resultados alcanzados, en el desarrollo.

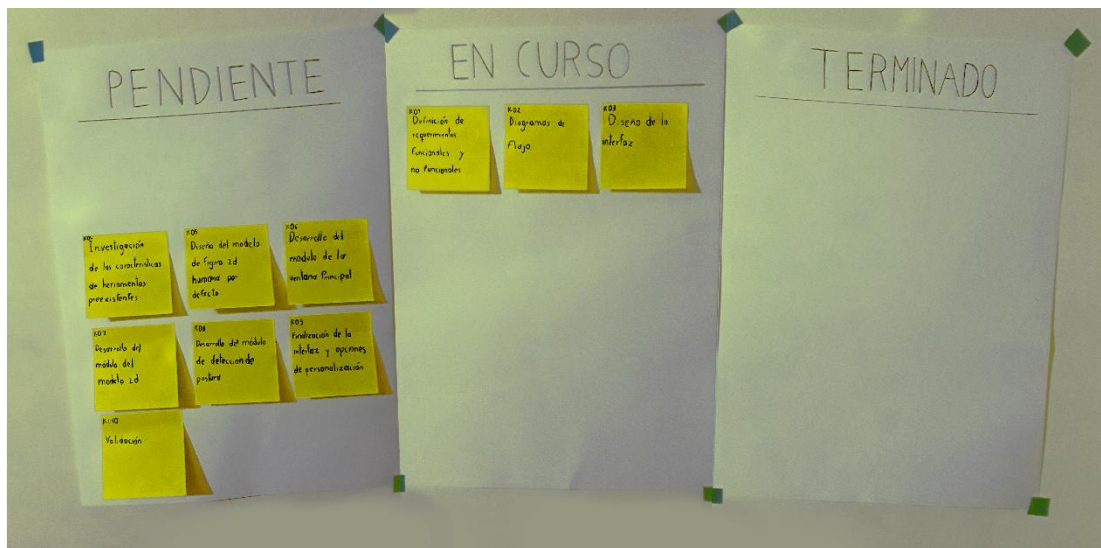


Figura 19. Tablero Kanban del desarrollo

En la **Figura 19**. Tablero Kanban del desarrollo se presenta una fotografía del tablero Kanban utilizado durante el ciclo el desarrollo. Dicho tablero se utilizó como herramienta principal para la planificación, seguimiento y control del flujo de trabajo. A continuación, se detallan las tareas registradas en el tablero Kanban, organizadas conforme al desarrollo:

- ***K01. Investigación de las características de herramientas preexistentes***
- ***K02. Definición de requerimientos funcionales y no funcionales***
- ***K03. Diagrama de procesos actual y el propuesto***
- ***K04. ¡Error! No se encuentra el origen de la referencia.***
- ***K05. Diseño del modelo de figura humana 2D por defecto***

- ***K06. Desarrollo del módulo de la ventana principal***
- ***K07. Desarrollo del módulo del modelo 2D***
- ***K08. Desarrollo del módulo de detección de postura***
- ***K09. Finalización de la interfaz y opciones de personalización***
- ***K10. Implementación de la aplicación***
- ***K11. Validación con usuarios***

3.1.3 Análisis de herramientas de desarrollo

Para el desarrollo de la aplicación se seleccionaron herramientas adecuadas para programas de recursos interactivos como el modelo 2D y compatibles con recursos de IA. Estas herramientas fueron elegidas tomando en cuenta la facilidad de integración, la disponibilidad de recursos y su capacidad para manejar tanto el procesamiento de imágenes como la manipulación de modelos 2D.

Visual Studio Code

El entorno principal utilizado fue Visual Studio Code, fácil compatibilidad con los equipos y compatibilidad con extensiones orientadas al desarrollo en Python. Su interfaz simple permitió desarrollar el código fuente, realizar depuración y mantener una estructura modular durante las etapas de implementación. Además, su soporte para control de versiones facilitó el seguimiento de los cambios en el proyecto.

Python

El lenguaje principal fue Python, seleccionado basándose en los resultados de **Figura 2**. Frecuencia de lenguajes de programación utilizados por su versatilidad, legibilidad y extensa comunidad de desarrolladores, en concreto la versión 3.11 por la compatibilidad de versiones con las librerías. Su amplio ecosistema de librerías lo hace ideal para proyectos que tengan recursos basados en IA, procesamiento de imágenes y

gráficos 2D. Python permitió combinar el reconocimiento de posturas humanas con la representación visual del modelo de manera eficiente y estructurada.

MediaPipe

La detección de posturas humanas mediante visión por computadora se apoya principalmente en el uso de librerías y frameworks especializados que facilitan el procesamiento de imágenes y la implementación de modelos de aprendizaje profundo. Estas herramientas proporcionan funciones optimizadas para la captura y análisis de video, así como modelos entrenados previamente o entornos flexibles para el desarrollo de soluciones personalizadas, permitiendo reducir los tiempos de implementación y mejorar la precisión de los resultados.

Tabla 4. Comparación de recursos para la detección de posturas humanas.

Herramienta / Framework	Tipo de herramienta	Uso en detección de posturas	Ventajas principales	Desventajas / Limitaciones
OpenCV	Librería de visión por computadora	Puede usarse como apoyo lectura de cámara, procesamiento de imágenes.	Muy rápida, compatible con otras librerías.	No incluye modelos preentrenados para pose humana.
MediaPipe	Librería de Google para visión en tiempo real	Estimación pose 2D y 3D lista para usar.	Detecta posturas en tiempo real con alta precisión y bajo consumo.	Difícil de personalizar o reentrenar.
Detectron2	Framework de visión profunda	Implementa modelos avanzados para mapear cada píxel del cuerpo humano en 3D.	Resultados muy precisos y configurables, permite reentrenar.	Instalación compleja; requiere GPU; más pesado y lento en tiempo real.
PyTorch	Framework de deep learning	Sirve como base para implementar redes de detección de pose.	Total flexibilidad para crear o entrenar modelos personalizados.	No incluye detección de pose lista.
RoIAlign	Operación dentro de	Se usa internamente en	Mejora la precisión	No se usa sola.

	modelos de detección	redes para extraer regiones de interés del cuerpo.	espacial en modelos de detección de partes del cuerpo.	
ShapeNet	Base de datos 3D	No detecta pose humana, pero se puede usar para entrenar modelos que estimen o reconstruyan posturas en 3D.	Ideal para investigación en reconstrucción 3D del cuerpo humano.	No apta para detección directa de posturas.
TensorFlow	Framework de deep learning	Permite implementar modelos o redes personalizadas.	Tiene modelos preentrenados fáciles de usar, compatible con GPU y móviles.	Requiere configuración inicial y conocimiento básico de redes neuronales.

La librería MediaPipe fue seleccionada por el reconocimiento, especialización en reconocimiento de posturas humanas y eficiencia en tiempo real. A diferencia de otras alternativas más complejas como Detectron2 o PyTorch, MediaPipe es una opción optimizada y lista para utilizarse, capaz de detectar puntos de referencia corporales de forma rápida y estable, incluso en dispositivos con recursos limitados. Estos puntos fueron procesados por la aplicación y ajustados para que el modelo 2D adoptara la postura correspondiente. Su precisión y facilidad de integración fueron factores cruciales para su elección.

MediaPipe es una librería desarrollada por Google que proporciona herramientas y modelos listos para usar en tareas de visión por computadora. Dentro de sus capacidades, incluye un sistema de detección de postura que identifica y rastrea los puntos clave del cuerpo humano en imágenes. Gracias a esta función, MediaPipe permite capturar la postura de una persona de manera rápida y eficiente, lo que resulta útil para aplicaciones interactivas y análisis de movimiento.

Pygame

Las herramientas para el desarrollo de aplicaciones gráficas interactivas permiten crear entornos visuales, gestionar eventos de entrada y manipular elementos gráficos en pantalla. Estas soluciones varían en peso, complejidad y nivel de abstracción, ofreciendo desde opciones ligeras orientadas a gráficos 2D hasta entornos más

completos con sistemas de interfaz y control avanzado del renderizado. La elección de una herramienta adecuada influye directamente en la facilidad de desarrollo, el rendimiento y las capacidades de interacción de la aplicación.

Tabla 5. Comparación de librerías gráficas y multimedia de Python.

Librería	Pygame	Arcade	Kivy	Godot	PyOpenGL
Peso aproximado	4–6 MB	15–20 MB	50–100 MB	80–120 MB	5 MB
Complejidad	Baja a media	Baja a media	Media a alta	Media	Muy alta
¿Crea ventanas?	si	si	si	si	no
¿Botones/interfaz nativa?	No nativo (se programan manualmente)	No nativo (pero se pueden crear manualmente)	Sí, completo sistema de interfaz	Sí, con sistema de interfaz integrado	No
Manipular imágenes con el mouse	Se puede detectar clics, arrastrar, etc.	Mejor soporte para arrastrar/soltar	Soporte táctil, gestos, arrastre, pinch-zoom	Soporte pleno mediante nodos	Se debe implementar desde cero
Conectar objetos entre sí	imágenes, grupos y colisiones	imagenes, físicas y escenas	elementos interactivos y jerargicos	Sistema de nodos y escenas	Controlas de todo lo que se dibuja en pantalla, pixel por pixel, usando

					la GPU directamente.
Otros aspectos técnicos	Muy flexible para 2D; sonido integrado.	Basado en OpenGL; alto rendimiento; API moderna.	Ideal para aplicaciones; soporta Android/iOS.	Motor 2D/3D; edición visual.	control total de GPU.

Entre las alternativas que se consideraron para el desarrollo se analizaron en función de sus capacidades técnicas, requisitos de implementación y carga de recursos. La opción seleccionada fue Pygame, ya que contaba con las funciones necesarias para la gestión gráfica e interactiva del sistema, presenta un nivel de complejidad bajo en su uso y un consumo mínimo de memoria. Además, al tratarse de una librería orientada al desarrollo de aplicaciones interactivas y videojuegos, ofrece un conjunto de funciones especializadas en la manipulación de elementos visuales y la captura de eventos del usuario.

Para la representación y manipulación del modelo 2D se utilizó Pygame, una biblioteca enfocada en el manejo de gráficos y eventos en tiempo real. A través de esta herramienta se implementó la interfaz además de las funciones necesarias para dibujar, mover y rotar las diferentes partes del cuerpo, así como la interacción mediante el mouse. Pygame proporcionó la base visual sobre la cual se desarrolló la dinámica interactiva de la aplicación.

3.2 Definición de requerimientos

Iniciando la primera etapa del desarrollo del sistema propuesto, se realizó el levantamiento de los requerimientos funcionales y no funcionales, siguiendo los parámetros establecidos por la norma IEEE 830. En esta etapa se definieron las características principales del sistema mediante la encuesta presentada en el Anexo A.

Encuesta para la evaluación de la viabilidad y aceptación Anexo A, la cual permitió recopilar las necesidades, expectativas y preferencias de los usuarios potenciales respecto a la creación de ilustraciones 2D asistidas por inteligencia artificial.

Secuencialmente, los requerimientos fueron documentados en el Anexo C. Levantamiento de requerimientos – IEEE 830, aplicando la estructura que se sugiere en la norma IEEE 830. Cada requerimiento se organizó con su identificador, nombre, características, descripción, requerimientos no funcionales asociados, nivel de prioridad, criterios de aceptación y observaciones relevantes. Esta documentación constituye una base para las etapas posteriores del desarrollo, garantizando su coherencia con los objetivos del proyecto.

Tabla 6. Planificación de requerimientos del sistema por iteraciones

N.	ID	Requerimiento	Tiempo
1	RF01	Modelo 2D manipulable por el usuario	5 días
2	RF02	Cuadro de análisis de postura con imágenes	5 días
3	RF03	Guardar modelo como imagen PNG	3 días
4	RF04	Fondo personalizable del modelo	3 días
5	RF05	Restablecer postura por defecto	3 días
6	RF06	Selector de apariencias del modelo	3 días

7	RF07	Selector de expresiones faciales	3 días
---	------	----------------------------------	--------

3.3 Desarrollo de la aplicación

En esta sección se presenta el proceso integral de desarrollo de la aplicación, abordando la investigación preliminar de herramientas existentes, la definición de requerimientos, la estructuración de los procesos operativos, el diseño de la interfaz, recursos visuales y la construcción de los módulos funcionales. Asimismo, se detalla la integración de cada componente, su validación y los resultados obtenidos durante la evaluación del sistema.

3.3.1 Investigación de las características de herramientas preexistentes

Para establecer una base comparativa y determinar los procesos que debían incorporarse o mejorarse en la aplicación, se realizó un análisis técnico de diversas herramientas utilizadas para la creación de ilustraciones de figuras humanas en 2D. Este estudio abarcó herramientas de edición digital, generadores asistidos por inteligencia artificial y plataformas de diseño gráfico, permitiendo identificar funciones comunes y enfoques utilizados en los procesos de ilustración.

Las características principales, tiempo de elaboración, facilidad de uso, complejidad operativa, precisión en la edición, fluidez del flujo de trabajo y requerimientos técnicos, fueron contrastadas y organizadas en la **Tabla 3**. Comparativa cualitativa de herramientas de edición tradicionales, lo que permitió definir lineamientos iniciales para orientar el desarrollo de la herramienta propuesta.

3.3.2 Definición de requerimientos funcionales y no funcionales

A partir de la información obtenida en la investigación previa y de los datos recolectados mediante encuestas, se establecieron los requerimientos funcionales y no funcionales que guiarían el desarrollo del sistema. Estos fueron redactados siguiendo la estructura del estándar ANSI/IEEE 830-1998, documentando de forma formal y estructurada los objetivos del sistema, sus funciones, restricciones, prioridades, interfaces, usuarios y criterios de validación.

El proceso permitió delimitar con claridad los elementos esenciales del sistema, así como las expectativas de rendimiento, usabilidad y compatibilidad. La documentación detallada se organizó en el **Anexo C. Levantamiento de requerimientos – IEEE 830**, donde se encuentra la especificación técnica completa.

3.3.3 Diagrama de procesos actual y el propuesto

Los procesos que se llevan a cabo al realizar una ilustración 2D de una figura humana son un conjunto de acciones que realiza el usuario. En esta sección se presenta una descripción de esos procesos y el funcionamiento interno de la aplicación, detallando los procesos que permiten su operación. Se abordan los mecanismos de despliegue de la interfaz, la interacción con el modelo humano 2D, la gestión visual y las herramientas que integran el flujo completo del sistema.

Se desarrollaron los diagramas con el fin de representar de manera clara el proceso de creación de ilustraciones. **Figura 20.** Diagrama de flujo de Krita describe el flujo de trabajo seguido durante la prueba realizada a la submuestra utilizando la herramienta Krita, mientras que la **Figura 21.** Diagrama de flujo de la aplicación desarrollada muestra el proceso equivalente utilizando la aplicación desarrollada.

La comparación entre ambos diagramas permitió evidenciar la optimización del flujo de trabajo, destacando la reducción de pasos manuales y la automatización de procesos que en Krita requerían múltiples acciones repetitivas.

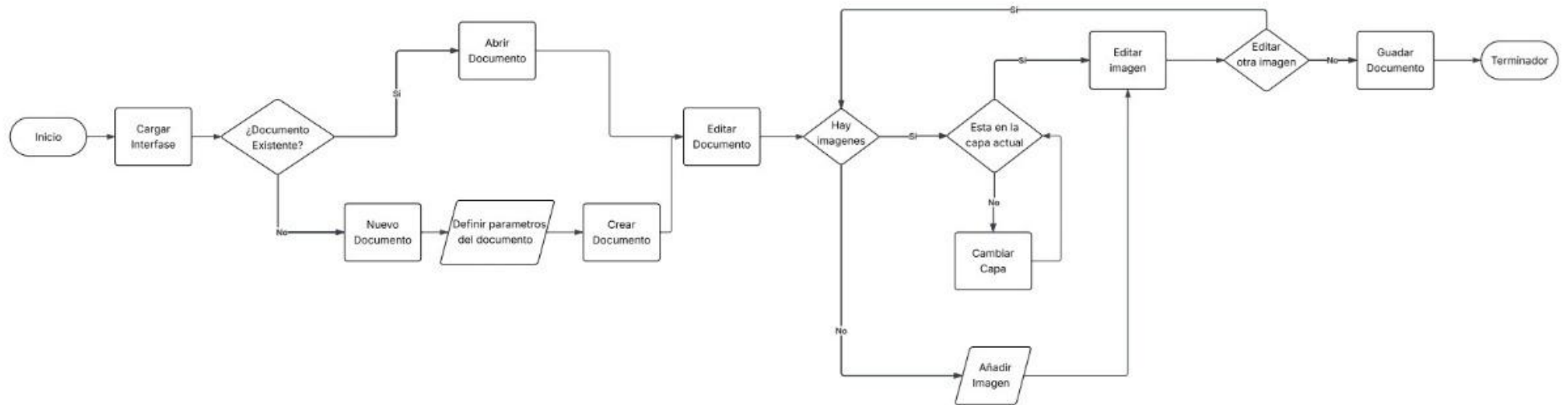


Figura 20. Diagrama de flujo de Krita

La **Figura 20**. Diagrama de flujo de Krita presenta el diagrama de flujo que describe el funcionamiento de Krita, la herramienta de edición empleada en la submuestra. En este esquema se detalla el proceso seguido durante la creación de las ilustraciones de figuras humanas en 2D, abarcando desde la apertura del programa hasta los ajustes finales y el guardado del proyecto. Durante este flujo de trabajo fue necesario insertar imágenes en nueve ocasiones y realizar múltiples movimientos en cada imagen para lograr una disposición satisfactoria.

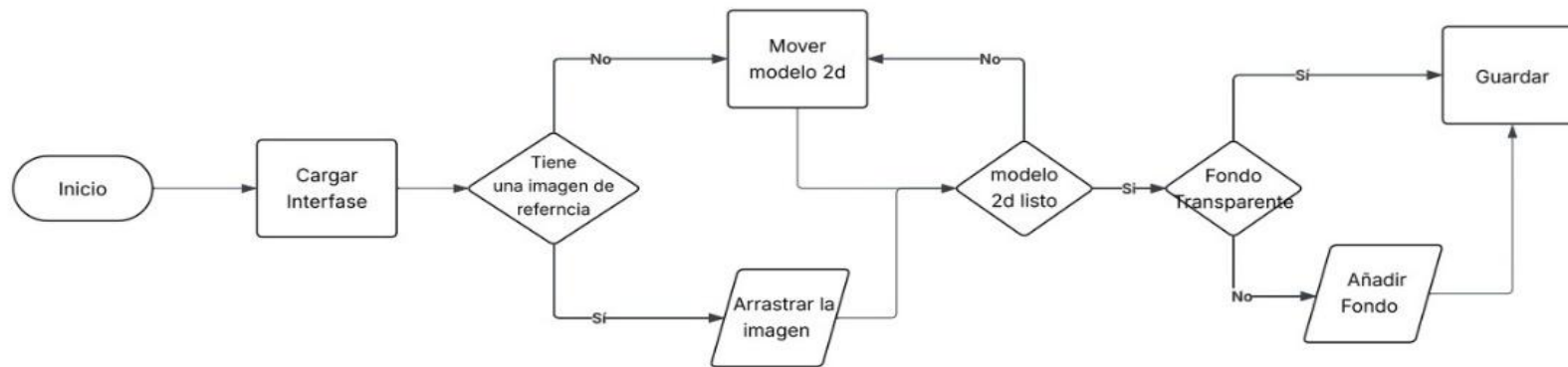


Figura 21. Diagrama de flujo de la aplicación desarrollada

La **Figura 21**. Diagrama de flujo de la aplicación desarrollada presenta el diagrama de flujo de la aplicación desarrollada, en el cual se visualiza la secuencia de procesos que intervienen en la generación de ilustraciones 2D de figuras humanas. Este diagrama permite comprender de manera estructurada cómo interactúan los distintos módulos del sistema, reflejando así la lógica interna y el flujo operativo de la aplicación, además de destacar que el número de iteraciones para conseguir el mismo resultado que con Krita puede ser de solamente 1

3.3.4 Diseño de la interfaz de la aplicación

El diseño preliminar de la interfaz se llevó a cabo en Figma, se utilizó esta herramienta por su capacidad para estructurar visualmente la aplicación de forma clara, rápida, fácil aprendizaje y gratuidad. En este entorno se definió una organización compuesta por dos áreas principales: un lienzo a la izquierda para la visualización del modelo 2D y, a la derecha, un panel de herramientas que agrupa la carga de imágenes para detección de postura, las opciones de personalización del modelo, la configuración del fondo y la función de exportación. El uso de Figma permitió planificar una interfaz funcional antes de implementar cualquier componente, asegurando una distribución lógica de elementos y una experiencia de uso intuitiva, incluso para usuarios sin conocimientos previos en diseño.

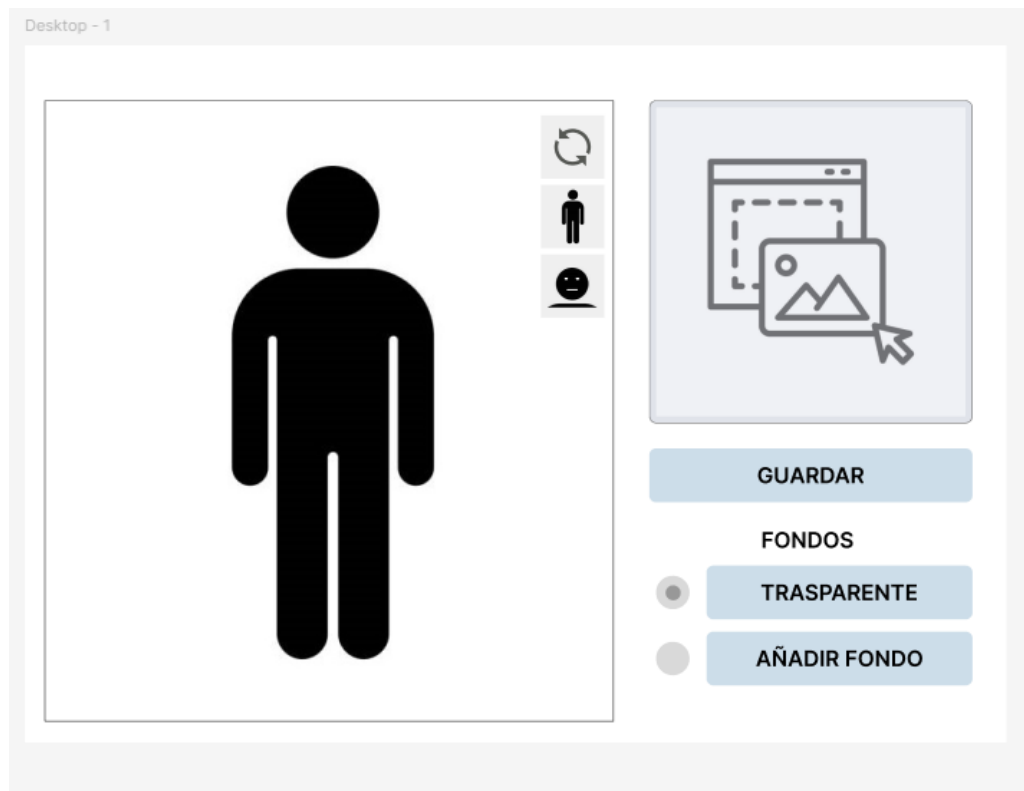


Figura 22. Interfaz principal del prototipo

En la **Figura 22**. Interfaz principal del prototipo se muestra la pantalla principal del prototipo en Figma, la cual integra los procesos previstos en la **Figura 21**. Diagrama de flujo de la aplicación desarrollada para la aplicación. En el área izquierda de la ventana se encuentra el lienzo destinado a la visualización del modelo 2D,

acompañado en su esquina superior derecha por tres botones representados mediante iconos: el primero restablece el modelo a su estado inicial, mientras que los otros dos despliegan opciones de personalización del modelo 2D y de expresiones. En el panel derecho se ubica, en la parte superior, un cuadro gris con un icono central destinado a la importación de imágenes mediante arrastre para el procesamiento de postura; debajo se sitúa el botón “Guardar”, encargado de la exportación de la ilustración generada. Finalmente, la sección de configuración del fondo compuesta por un título y un checkbox que determina el tipo de fondo de la ilustración, configurado por defecto como transparente.

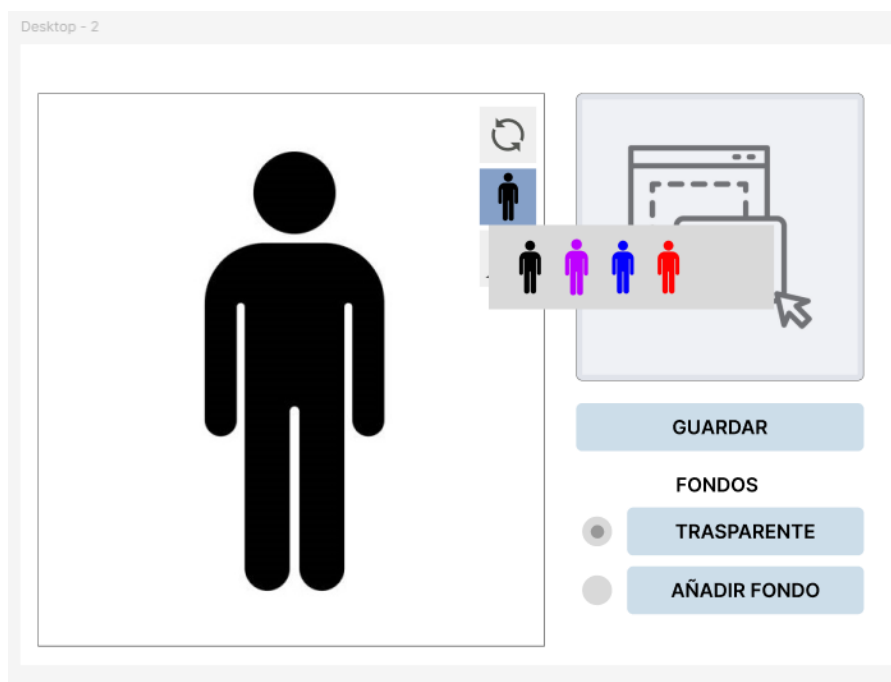


Figura 23. Interfaz de personalización del modelo 2D

En la **Figura 23**. Interfaz de personalización del modelo 2D se muestra la funcionalidad del segundo botón del lienzo, al activarse despliega un panel de selección que presenta la galería completa de las apariencias disponibles para el modelo 2D, permitiendo al usuario cambiar la apariencia actual del modelo 2D mediante una interfaz organizada en miniaturas.

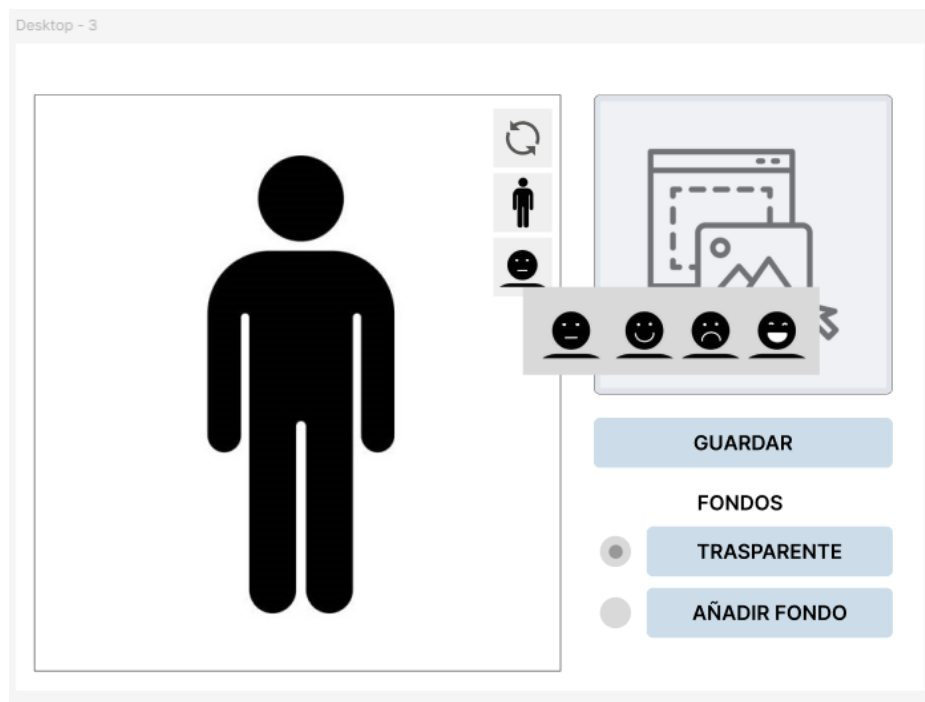


Figura 24. Interfaz de personalización de expresiones del modelo 2D

En la **Figura 24**. Interfaz de personalización de expresiones del modelo 2D se presenta la funcionalidad del tercer botón del lienzo, el cual, al activarse, despliega un panel de selección similar al del botón anterior, pero centrado en las expresiones faciales. Este panel muestra las distintas variantes de expresión disponibles para la apariencia seleccionada del modelo 2D, permitiendo al usuario modificarla mediante la elección directa de una de las miniaturas mostradas.

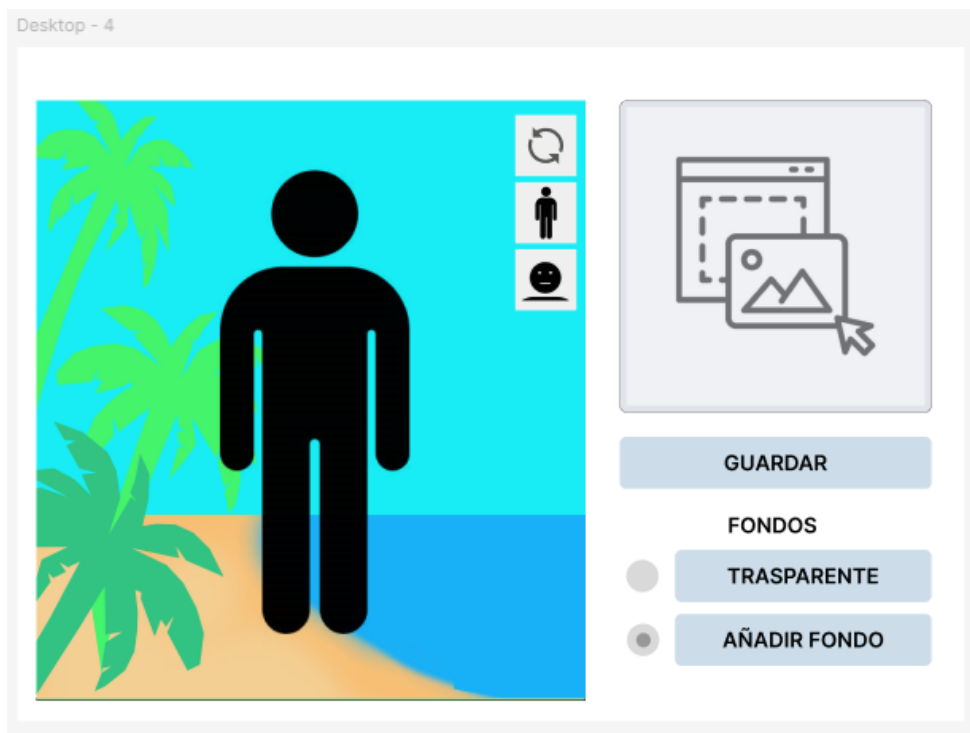


Figura 25. Interfaz de personalización de fondo

En la **Figura 25**. Interfaz de personalización de fondo se muestra la modificación de la interfaz al habilitar la opción “Añadir fondo” mediante el checkbox. Al activarse esta función, el área interna del lienzo se actualiza para integrar la imagen seleccionada como fondo, ajustándola automáticamente a las dimensiones del lienzo para garantizar una cobertura completa y proporcional.

3.3.5 Diseño del modelo de figura humana 2D por defecto

El modelo humano 2D por defecto fue desarrollado en Krita mediante la creación de nueve imágenes independientes que representan las distintas partes anatómicas: brazos, antebrazos, muslos y piernas, mientras que la cabeza y el torso se elaboraron como una sola imagen. Cada componente fue diseñado de manera modular, que se pueda ensamblar en las articulaciones, lo que permite rotaciones controladas con precisión.

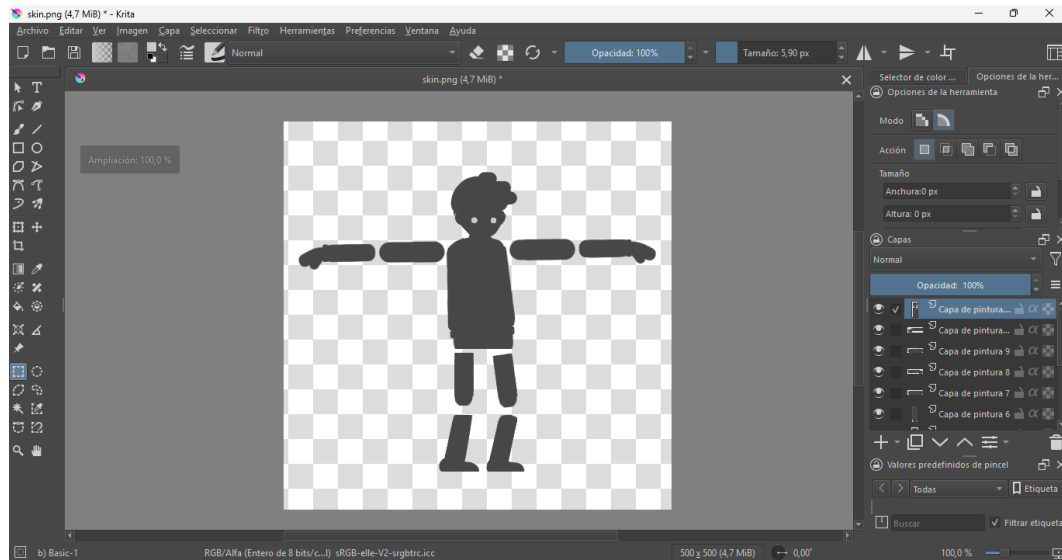


Figura 26. Estructura del modelo 2D de figura humana

Una vez definidas las distintas partes del modelo 2D, como se observa en la **Figura 26**. Estructura del modelo 2D de figura humana, se procedió a configurar su esquema de colores. La porción superior, correspondiente a la camiseta de manga larga, se diseñó con un tono rojo de alto contraste (RGB 255, 1, 38), complementado con una variante de menor saturación para resaltar detalles en el cuello y las mangas (RGB 171, 10, 30). La sección inferior, que abarca pantalón y zapatos, se representó con un azul profundo (RGB 2, 5, 80), tono que también se empleó para el cabello. La piel se definió con un rosado base (RGB 254, 128, 129), aplicando una versión desaturada en zonas específicas como el cuello y la oreja derecha (RGB 219, 70, 96) para generar contraste. La elección de una paleta de colores reducida busca maximizar el contraste entre los elementos y su fácil identificación.

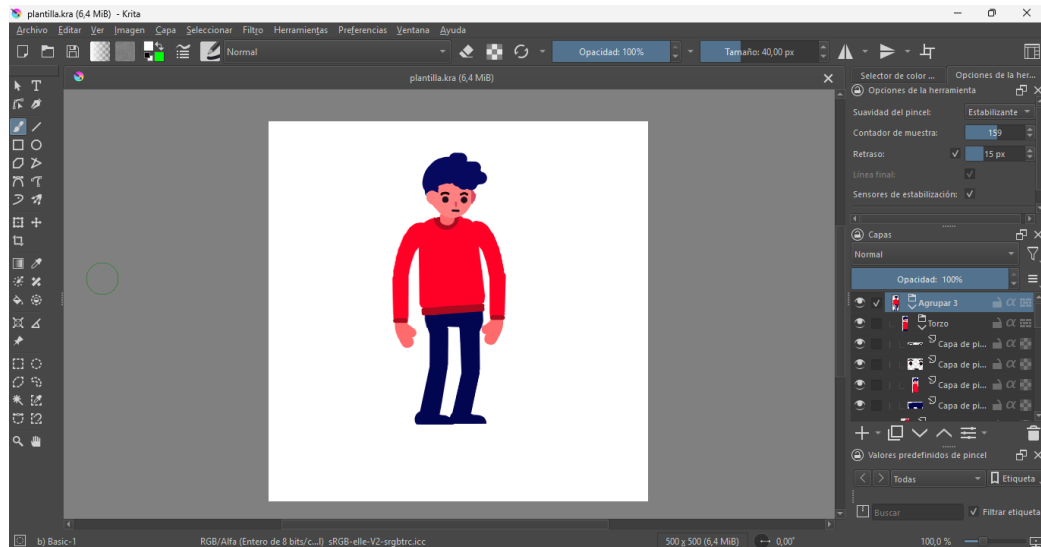


Figura 27. Modelo 2D de figura humana por defecto terminada

El elemento que combina el torso y la cabeza del modelo 2D se editó en Krita para permitir la representación de distintas expresiones faciales. La expresión por defecto se definió inicialmente, y las variantes se generaron las líneas de los rasgos faciales, específicamente ojos, cejas y boca. Estos cambios permitieron obtener, con modificaciones mínimas, expresiones como feliz, triste, sorprendido y enojado, manteniendo la consistencia con la estructura del modelo 2D.

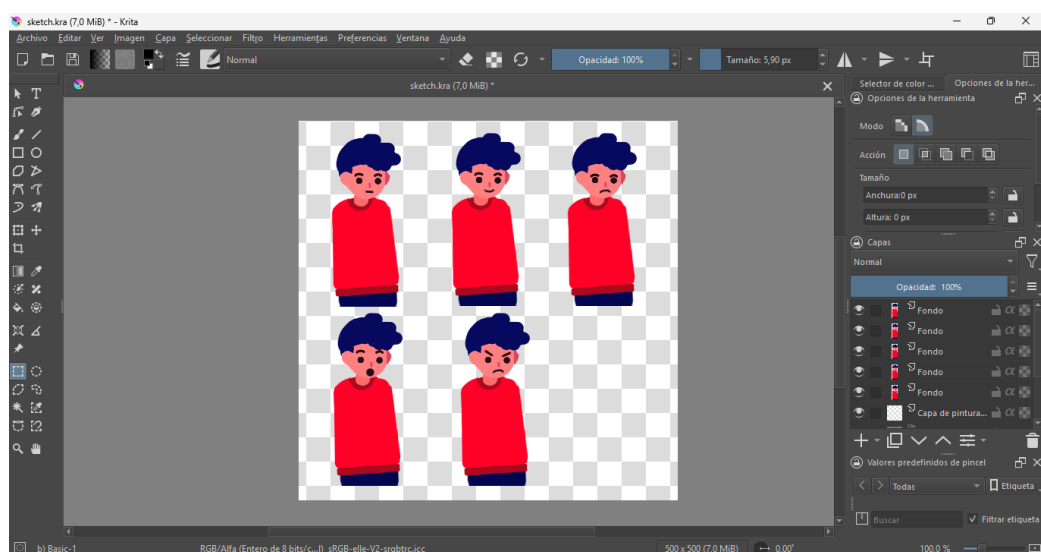


Figura 28. Expresiones de modelo 2D de figura humana por defecto

SE implemento una segunda apariencia del modelo 2D, generando una variante a partir del modelo existente, aprovechando la apariencia del modelo 2D previamente definida.

Esta nueva apariencia se creó mediante ajustes controlados en los niveles de balance de color, lo que permitió modificar la paleta cromática sin alterar la geometría ni la disposición de las partes del cuerpo. De esta manera, se logró una representación visual distinta que mantiene la coherencia estructural del modelo original, proporcionando al usuario una opción adicional de personalización y ampliando la versatilidad estética del prototipo.

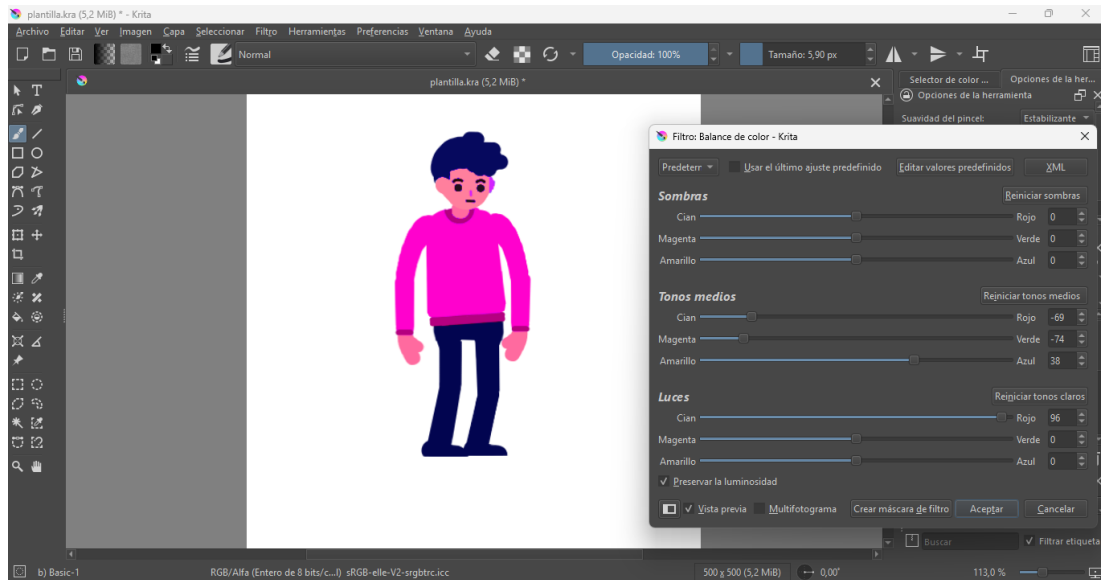


Figura 29. Segunda apariencia del modelo 2D de figura humana

3.3.6 Desarrollo del módulo de la ventana principal

El desarrollo comenzó con la configuración del entorno gráfico mediante Pygame, que actuó como el motor encargado del manejo de superficies, renderizado de imágenes y gestión de eventos. Se estableció una ventana principal con dimensiones fijas y dos áreas, el lienzo donde se dibuja la composición completa del modelo 2D y el panel que se utilizaría para los elementos de configuración como botones y el cuadro para pasar imágenes. Esta estructura permitió dividir el área principal de ilustración del panel lateral de herramientas, facilitando la actualización independiente de cada elemento.

Al iniciar la aplicación, se construye la ventana principal utilizando los recursos gráficos de Pygame. Las dimensiones de la interfaz son fijas y se definen como

800×600 píxeles mediante la función `pygame.display.set_mode()` (establece el tamaño de la ventana). Esta ventana se divide en dos áreas principales:

- **Lienzo:** Zona donde se dibuja en el lado izquierdo y actualiza el modelo humano 2D en cada fotograma. Aquí se ejecutan todas las interacciones directas del usuario con el personaje.
- **Panel lateral:** Área destinada a botones en el lado derecho, selectores, herramientas de análisis de postura y las opciones para guardar y gestionar fondos.

Durante esta etapa inicial, el sistema carga todos los recursos necesarios para operar. Esto incluye:

- La inicialización de superficies mediante `pygame.Surface` (crea un espacio gráfico para dibujar).
- La definición de áreas interactivas mediante `pygame.Rect` (representa zonas rectangulares que permiten detectar clics o colisiones).

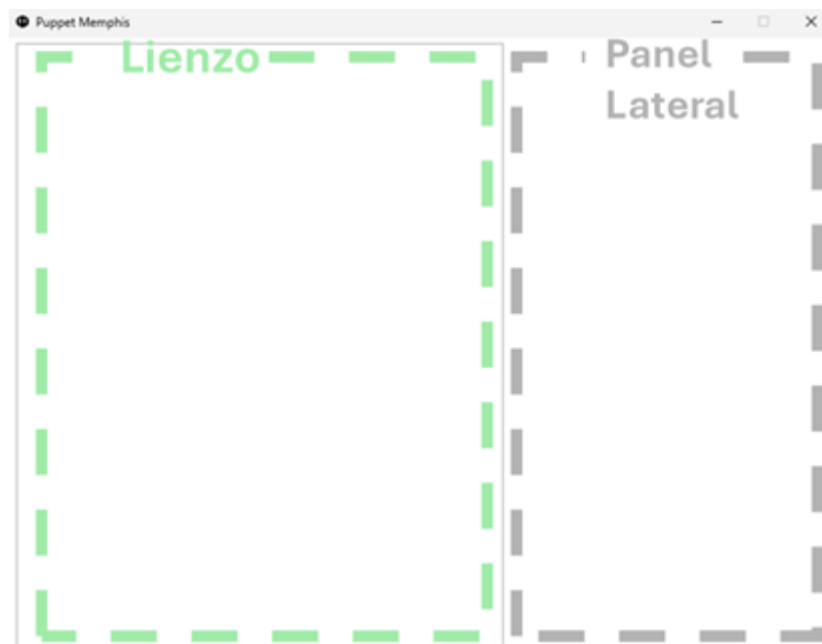


Figura 30. Interfaz del módulo ventana principal

3.3.7 Desarrollo del módulo del modelo 2D

Para el módulo del modelo 2D se utilizó Pygame como sistema de renderizado y manipulación gráfica. Cada parte del cuerpo se trató como una imagen independiente cargada en memoria, asociada a una coordenada en el lienzo conserva su tamaño, punto de origen y posición. Pygame permitió organizar estas partes en una estructura jerárquica, en la que la ubicación de cada extremidad depende de la posición de su articulación superior. La rotación de las extremidades se gestionó mediante las capacidades de transformación de Pygame, que permiten redibujar las piezas en distintos ángulos preservando su eje en la articulación. Esta rotación se aplicó en cada ciclo, ajustando la posición relativa mediante el vector articulación-extremidad para garantizar movimientos precisos y coherentes.

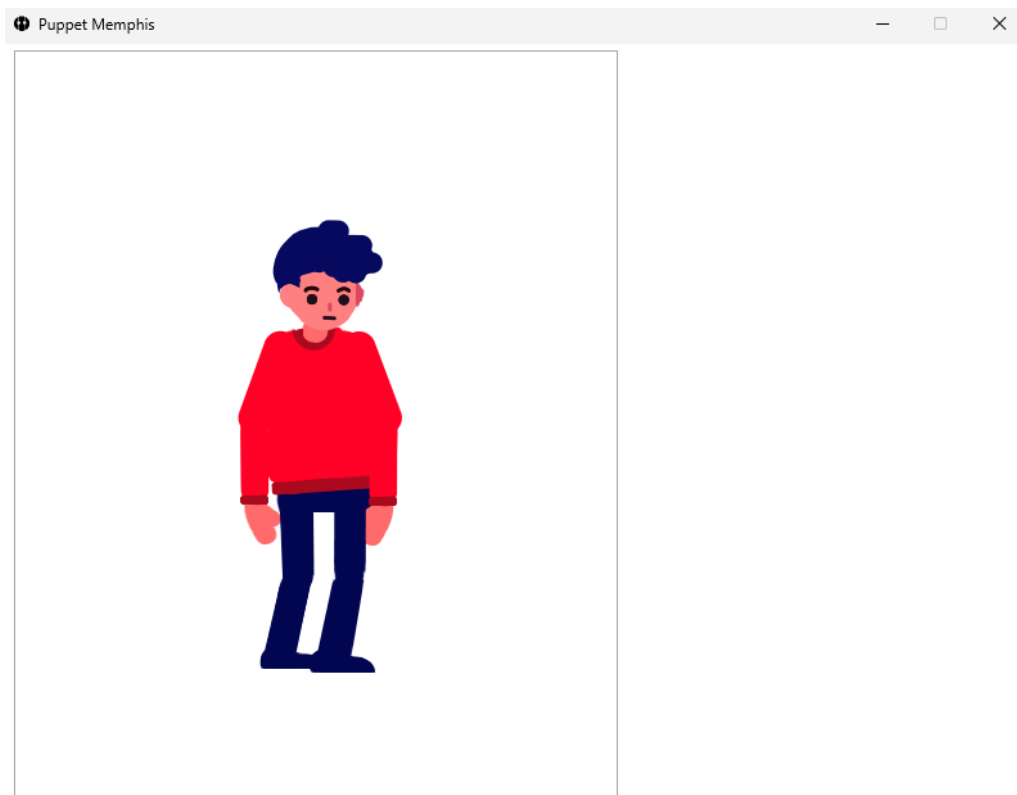


Figura 31. Modulo del modelo 2D en la interfaz

Una vez cargada la ventana principal, se inicializa el modelo 2D por defecto, compuesto las imágenes que representan las distintas partes del cuerpo humano, como se ve en la **Figura 31**. Modulo del modelo 2D en la interfaz, cada una con su propio

eje y un punto de unión que determina su comportamiento rotacional respecto a las otras imágenes. La interacción con el modelo se realiza mediante el mouse, utilizando `pygame.mouse.get_pos()` para obtener la posición actual del cursor y el evento `MOUSEBUTTONDOWN` para detectar clics sobre áreas activas del modelo.

Las acciones principales se estructuran de la siguiente manera:

Movimiento:

- Clic izquierdo: permite arrastrar el modelo completo dentro del lienzo sin alterar su postura. El sistema toma la posición del cursor y recalcula las coordenadas globales del modelo para desplazarlo suavemente en la pantalla.

Rotación:

- Clic derecho sobre una extremidad: el programa identifica la parte seleccionada mediante detección de colisiones con su rectángulo de interacción, además de resaltar la extremidad.
- A medida que el mouse se mueve, el sistema calcula el ángulo entre la posición del cursor y el eje de la extremidad.
- Finalmente, se actualiza la rotación aplicando las transformaciones sobre la superficie asociada.

Este módulo también gestiona la detección de interacciones entre el cursor y las áreas definidas del modelo, el reescalado de elementos, su transparencia y el orden de renderizado sobre la superficie principal, asegurando así una manipulación visual completa del modelo 2D.

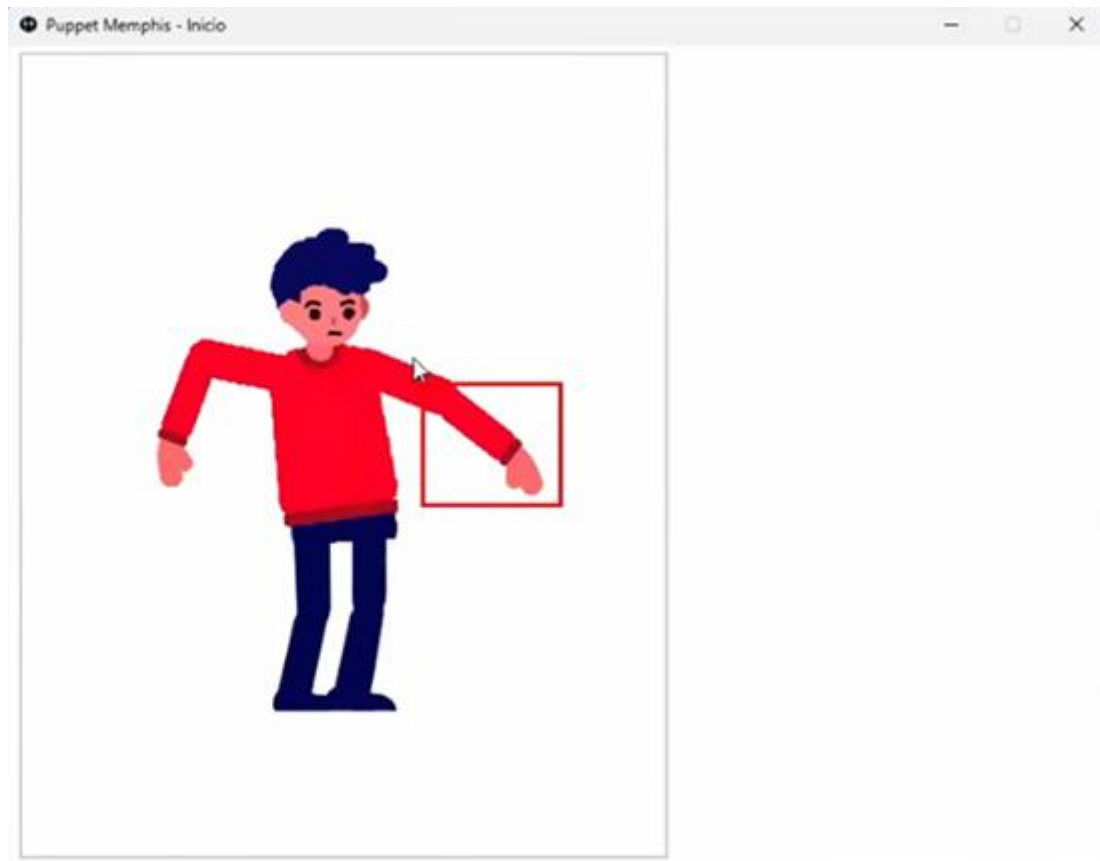


Figura 32. Ajuste y rotación de una extremidad del modelo 2D en la interfaz

3.3.8 Desarrollo del módulo de detección de postura

El módulo de detección de postura se desarrolló mediante la integración de MediaPipe Pose con el modelo humano 2D, permitiendo extraer información de la postura partir de fotografías de personas reales. Esta integración combina la capacidad de detección automática de articulaciones de MediaPipe con la manipulación y renderizado gráfico de Pygame, proporcionando un sistema que replica las poses humanas en el modelo 2D.

El proceso comienza cuando el usuario arrastra un archivo de imagen a la ventana de la aplicación al cuadro interactivo en la parte superior del panel mediante **pygame.DROPPFILE**. La imagen se carga usando OpenCV con **cv2.imread** y se transforma a formato RGB con **cv2.cvtColor** para que este en el formato que requiere MediaPipe. La función principal de detección es

mp.solutions.pose.Pose(static_image_mode=True).process(), que devuelve un objeto nombrado **results** con **pose_landmarks**, siendo los landmarks los puntos clave de la postura como las articulaciones y cabeza. Cada landmark tiene coordenadas normalizadas (x, y) que representan la posición relativa de cada articulación.

Estas coordenadas se pasan a la función **map_pose_to_model()**, que:

- Escala los puntos para adaptarlos al tamaño de la ventana de Pygame, ajustando su posición mediante un cálculo basado en las dimensiones del lienzo.
- Reasigna los landmarks a las partes del modelo 2D, definiendo posiciones de inicio y ejes de rotación para cada extremidad.
- Calcula los ángulos de las extremidades y del torso para asegurar que la orientación del modelo coincida con la postura detectada.

La interfaz permite visualizar simultáneamente:

- La imagen original con landmarks de MediaPipe sobrepuestos para validar la detección.
- El modelo 2D replicando la misma postura, listo para continuar con edición o exportación.

Este flujo asegura una reproducción precisa y visualmente coherente de poses humanas, integrando de forma eficiente la detección automática de MediaPipe con la manipulación gráfica en Pygame.

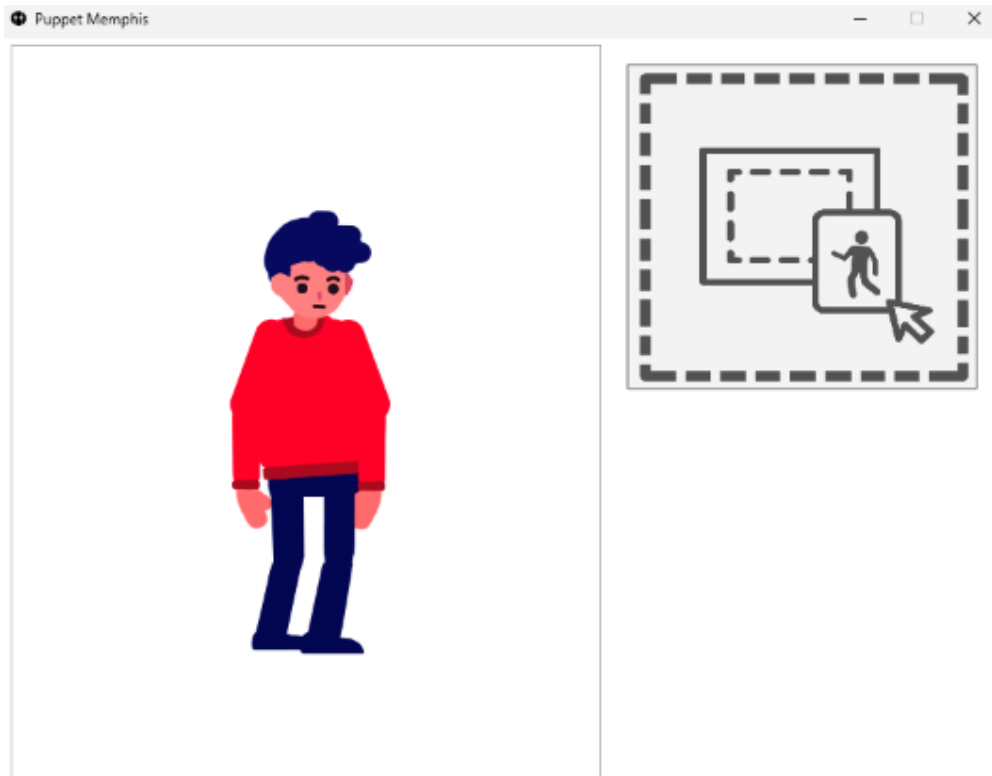


Figura 33. Módulo de detección de postura en la interfaz

3.3.9 Finalización de la interfaz y opciones de personalización

La finalización de la interfaz se completó con la incorporación de un conjunto de botones y áreas interactivas, construidas utilizando las funcionalidades de Pygame. Estos elementos se integraron en la ventana principal, proporcionando controles para restablecer la postura, cambiar la skin del modelo, modificar expresiones faciales, gestionar el fondo de la ilustración y exportar el modelo 2D como imagen. Los módulos de interacción y personalización disponibles incluyen:

Control y personalización del modelo

En la misma área del lienzo se definieron tres botones interactivos, cada uno definido mediante un **pygame.Rect** para detectar clics. Estos botones permiten modificar tanto el estado como la apariencia del modelo sin alterar su funcionamiento interno:

- **Restablecer postura:** Devuelve todas las partes del cuerpo a sus posiciones y ángulos originales de inicio de la aplicación.
- **Selector de skins:** Muestra distintas variaciones gráficas del modelo; al elegir una, se reemplazan las imágenes del cuerpo sin cambiar la postura actual.
- **Selector de expresiones:** Permite seleccionar expresiones faciales distintas, aplicadas directamente al modelo.

Estas herramientas incrementan el nivel de personalización de la aplicación al permitir cambiar el modelo 2D sin afectar su estructura interna ni la lógica de sus articulaciones.

Gestión del fondo

La configuración del fondo se implementó mediante un checkbox que ofrece dos opciones: “transparente”, seleccionada por defecto, y “añadir fondo”. Al seleccionar la segunda opción, el sistema permite agregar imágenes como fondo detrás del modelo, garantizando que cada ilustración pueda incorporar imágenes que acompañen al modelo.

En la parte inferior del panel se encuentra la sección destinada a la gestión del fondo. Por defecto, el lienzo se renderiza sobre una Surface de Pygame (un objeto gráfico que permite dibujar y controlar transparencia), mostrando el modelo 2D sin fondo.

El usuario puede activar la opción “Añadir fondo”, lo cual abre un explorador utilizando **Tkinter** para abrir un explorador de archivos, permitiendo al usuario seleccionar la imagen de fondo Una vez elegida:

- Se carga mediante **pygame.image.load()** que lee una imagen almacenada.
- Se adapta automáticamente al tamaño del lienzo.

El fondo nunca interfiere con los elementos interactivos ni con los cálculos de postura.

Exportación de ilustraciones

Para la exportación, se utilizó la capacidad interna de Pygame para capturar directamente el contenido de la superficie principal, esto permitió generar archivos PNG manteniendo la transparencia del lienzo, la resolución exacta del área de dibujo y el estado visual completo del modelo tal como se mostraba en pantalla. Gracias a esta integración, el módulo de personalización ofreció un control completo sobre la apariencia final, asegurando que cualquier configuración seleccionada por el usuario se reflejara fielmente en la ilustración exportada.

El botón “Guardar” genera una imagen final de la ilustración. El sistema combina:

- El fondo transparente, exceptuando cuando se añadió uno.
- Todas las partes del modelo 2D, con su postura, apariencia y expresión actual.

Para organizar y diferenciar los archivos exportados, se emplea la librería **datetime**, que permite obtener la fecha y hora actual y usarla en el nombre del archivo. Esto asegura que cada ilustración se guarde con un identificador único, evitando sobrescribir archivos previos.

La imagen se guarda como PNG, lo que permite conservar la transparencia cuando no se ha seleccionado un fondo teniendo como nombre de archivo la letra i por ilustración seguido por la fecha.

La ilustración generada se almacena en la carpeta predeterminada denominada **“Ilustraciones”**, utilizando como nombre de archivo la marca temporal correspondiente al momento de su creación, lo que permite una correcta organización y evita la sobreescritura de archivos.

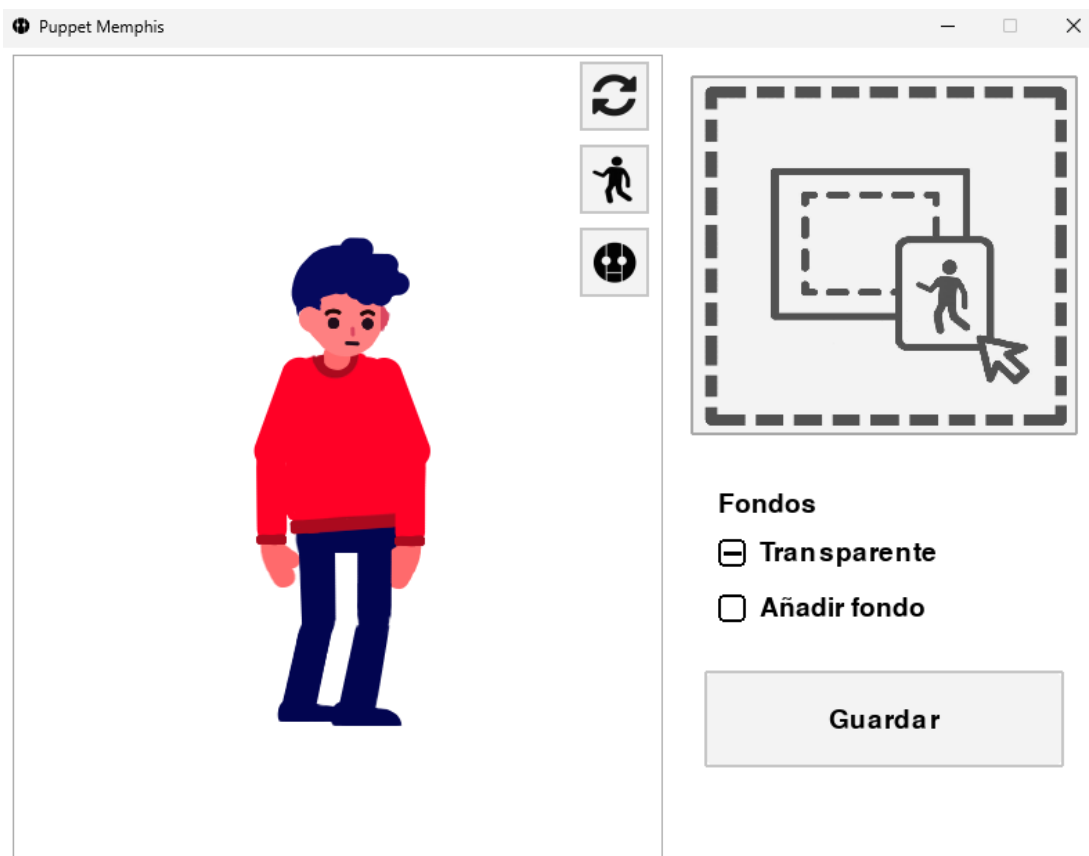


Figura 34. Interfaz y funcionalidad completadas

3.4 Implementación de la aplicación

Esta sección describe cómo se llevó a cabo la implementación de la aplicación, incluyendo las herramientas, librerías y módulos utilizados para desarrollar un sistema de manipulación de un modelo 2D con soporte de detección de postura mediante Inteligencia Artificial. Se explican tanto la arquitectura interna de la aplicación, que organiza los módulos y recursos, como el proceso de generación del ejecutable, destinado al usuario final.

3.4.1 Arquitectura de la aplicación

La arquitectura de la aplicación se diseñó para integrar de manera eficiente la manipulación de un modelo 2D con un sistema de detección de postura basado en

Inteligencia Artificial. El diseño modular permite separar responsabilidades entre los distintos componentes y facilita la escalabilidad y el mantenimiento del código.

Estructura de directorios y módulos principales:

- **/assets:** Contiene recursos generales como imágenes y sonidos necesarios para la aplicación.
- **/icons:** Iconos de la interfaz y botones interactivos.
- **/Ilustraciones:** Carpeta de salida donde se guardan las ilustraciones generadas por el usuario.
- **/Models:** Almacena los diferentes skins y expresiones del modelo 2D.
- **/skin01/faces y /skin02/faces:** Subcarpetas de Models con imágenes para distintas expresiones faciales del modelo.
- **main.py:** Módulo principal que ejecuta la aplicación, importando librerías como Pygame, OpenCV, Tkinter, OS, Sys y Datetime para la gestión de la interfaz, carga de imágenes y exportación de ilustraciones.
- **Launcher.py:** Módulo encargado de la pantalla inicial y la selección de archivos; utiliza Pygame, OS y OpenCV.
- **model_mophet.py:** Contiene la lógica del modelo 2D y la integración con detección de postura mediante MediaPipe Pose, así como cálculos de ángulos con math y representación gráfica en Pygame.

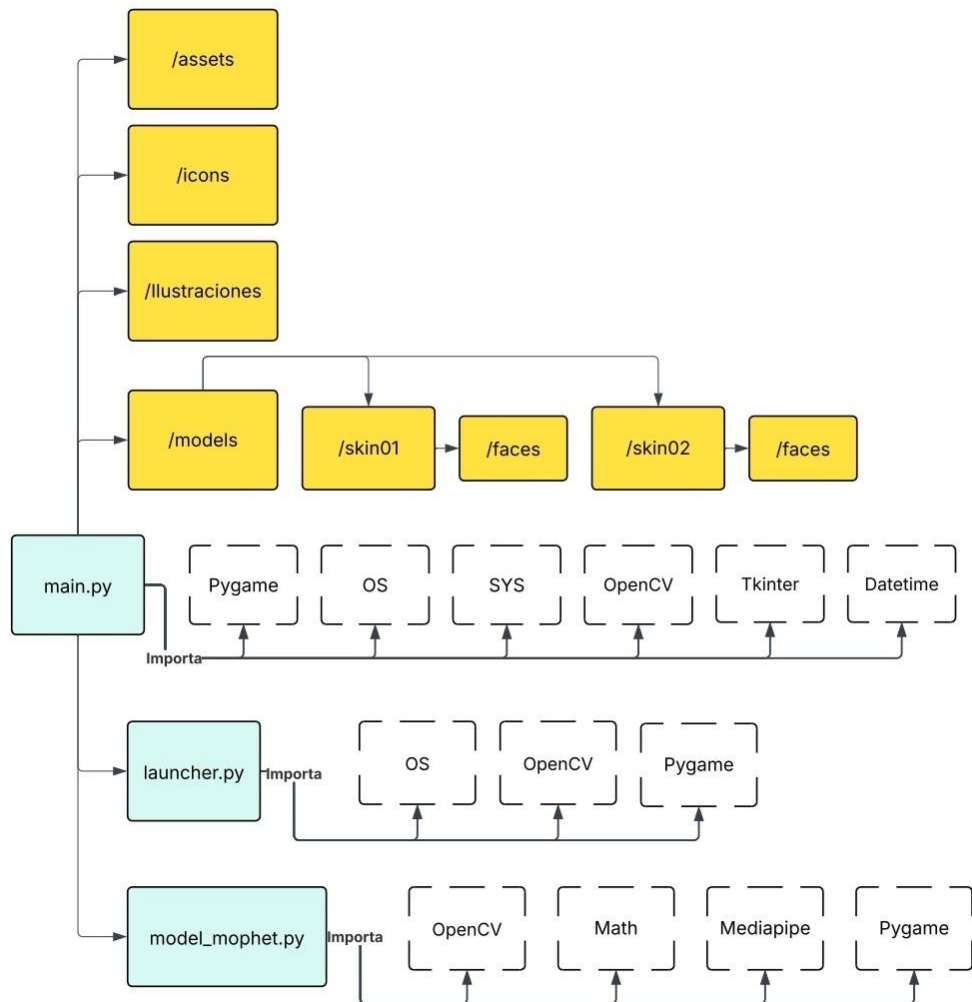


Figura 35. Estructura de directorios y módulos de la aplicación

3.4.2 Inicialización de la aplicación y generación del ejecutable

Una vez completada la aplicación, las funcionalidades principales se orientan a facilitar la comprensión y el uso por parte del usuario final. Esto permite una rápida inserción y operación de la aplicación en cualquier equipo, asegurando que los usuarios puedan interactuar con el modelo 2D y sus herramientas sin requerir conocimientos previos.

Introducción del usuario a la aplicación

Para guiar al usuario en la utilización de la aplicación, se implementó un video tutorial introductorio de 2 minutos y 37 segundos, adjunto en el **Anexo D**. Video tutorial de la aplicación desarrollada., accesible desde la pantalla de inicio. Este tutorial explica de manera secuencial las funciones disponibles, incluyendo:

1. La disposición y organización de los elementos de la interfaz.
2. La manipulación directa del modelo 2D mediante controles interactivos.
3. La restauración del modelo 2D a su postura inicial.
4. La personalización del modelo y la selección de expresiones faciales.
5. La incorporación de fotografías de referencia para adoptar posturas humanas.
6. La configuración del fondo de la ilustración, incluyendo opciones de transparencia y selección de imágenes externas.
7. La exportación del modelo 2D como ilustración PNG, manteniendo resolución, transparencia y el estado visual completo del modelo.

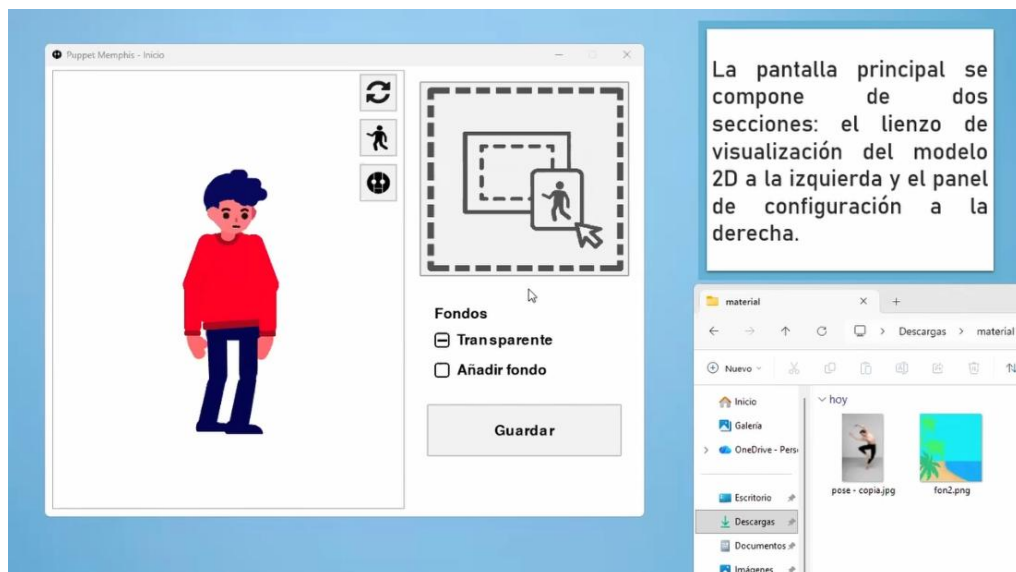


Figura 36. Fotograma del tutorial de la aplicación

Este enfoque asegura que el usuario comprenda rápidamente el funcionamiento de la aplicación y pueda operar todas las herramientas de manera intuitiva antes de pasar a la ejecución final y exportación de ilustraciones.

Compilación de la aplicación

Una vez completado el desarrollo de la aplicación, el siguiente paso era preparar una versión ejecutable para el usuario final. Esto es necesario porque, en su estado de desarrollo, la aplicación se ejecuta directamente mediante el intérprete de Python en el terminal, lo cual no resulta práctico para usuarios sin conocimientos de programación, además de necesitar contar con las librerías instaladas.

Para generar el ejecutable, se utilizó PyInstaller, una herramienta que permite empaquetar scripts de Python junto con sus dependencias en un único archivo ejecutable. Esto asegura que la aplicación pueda ejecutarse en cualquier equipo con Windows, sin necesidad de instalar Python ni librerías adicionales.

La creación del ejecutable implicó configurar varias características en el comando:

- **noconsole:** elimina la ventana de terminal, de modo que la aplicación se ejecute únicamente como ventana gráfica.
- **name="Puppet Memphis":** asigna un nombre personalizado al archivo ejecutable.
- **icon=puppet_icon.ico:** define el ícono del ejecutable.
- **add-data:** incorpora archivos y carpetas externas necesarias para el correcto funcionamiento de la aplicación, incluyendo recursos gráficos (assets, icons, Ilustraciones) y módulos internos (models, scripts .py), así como los módulos de MediaPipe necesarios para la detección de postura.

La entrada principal de PyInstaller fue main.py, ya que este script contiene el flujo de ejecución principal de la aplicación.

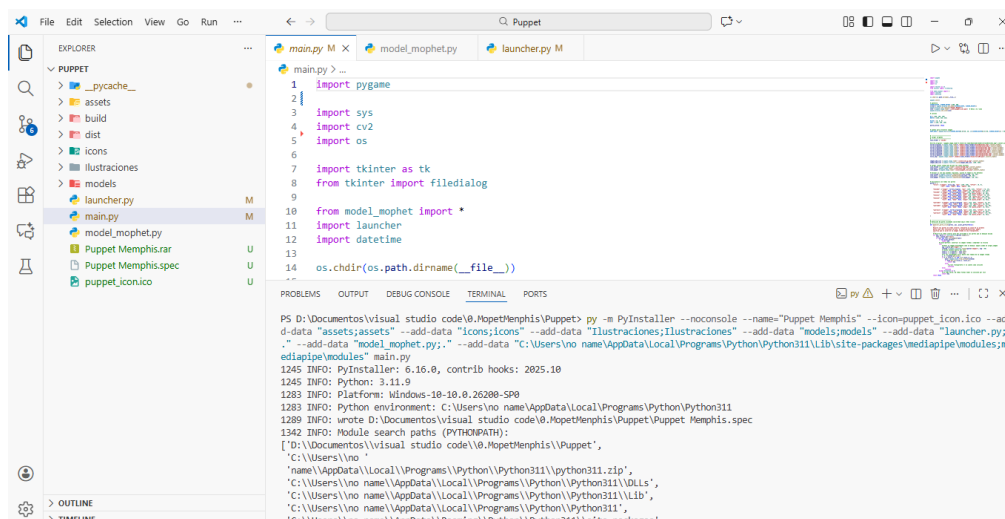


Figura 37. Ejecución del comando de PyInstaller en el terminal

Este proceso permitió obtener un archivo ejecutable independiente que mantiene la funcionalidad completa de la aplicación, incluyendo la carga de modelos 2D, la integración con MediaPipe para la detección de postura, la interfaz gráfica desarrollada con Pygame y la capacidad de personalización y exportación de ilustraciones.

El resultado final garantiza que cualquier usuario pueda ejecutar la aplicación de manera sencilla, sin necesidad de instalar Python ni configurar dependencias adicionales, cumpliendo así con los objetivos de accesibilidad y facilidad de uso.

3.5 Validación con usuarios

En esta sección se analiza la dificultad de uso y reducción de tiempo al crear ilustraciones 2D utilizando la aplicación desarrollada, comparándola con otra herramienta.

3.5.1 Reducción de la dificultad para creación de ilustraciones

La creación de ilustraciones 2D de figuras humanas en herramientas de edición implica múltiples procesos complejos y un elevado número de interacciones manuales con las herramientas gráficas. En Krita, la elaboración de una ilustración requiere ejecutar

secuencias repetitivas de acciones, como se puede ver en la **Figura 20**. Diagrama de flujo de Krita, desde agregar imágenes base hasta ajustar la posición y tamaño de cada elemento hasta alcanzar un resultado satisfactorio. Solo el proceso de agregar las imágenes que componen la figura humana puede implicar 9 repeticiones, sin contar los ajustes posteriores en proporciones y alineaciones.

En contraste, la aplicación desarrollada simplifica considerablemente este flujo de trabajo, permitiendo completar la ilustración con un único ciclo de los procesos de la aplicación, como se puede ver en la **Figura 21**. Diagrama de flujo de la aplicación desarrollada, integrando la detección automática de postura y la manipulación directa del modelo 2D. Esto reduce drásticamente la complejidad operativa y facilita la interacción del usuario.

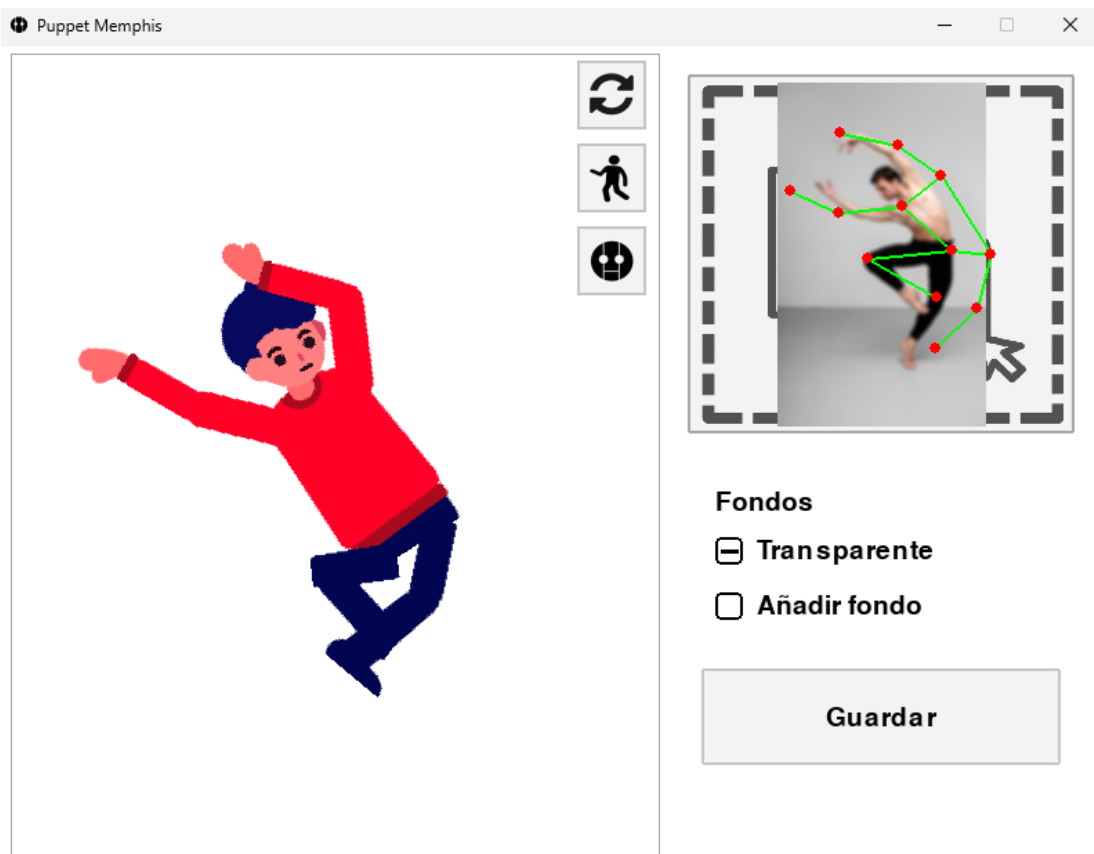


Figura 38. Aplicación desarrollada en ejecución

3.5.2 Reducción del tiempo requerido

Para evaluar el desempeño de la aplicación, se llevó a cabo una validación experimental con la submuestra de participantes replicando la misma tarea previamente realizada en Krita, utilizando la aplicación desarrollada. Durante la prueba, se registraron los tiempos de elaboración de cada ilustración, comparando los resultados entre ambas herramientas.

Los datos mostraron una reducción significativa del tiempo de creación, con un promedio de 6.25 minutos utilizando Krita, frente a 0.92 minutos al emplear la aplicación desarrollada. Esto representa una reducción del **85%** en el tiempo de ejecución, evidenciando la eficiencia y facilidad de uso de la herramienta desarrollada.

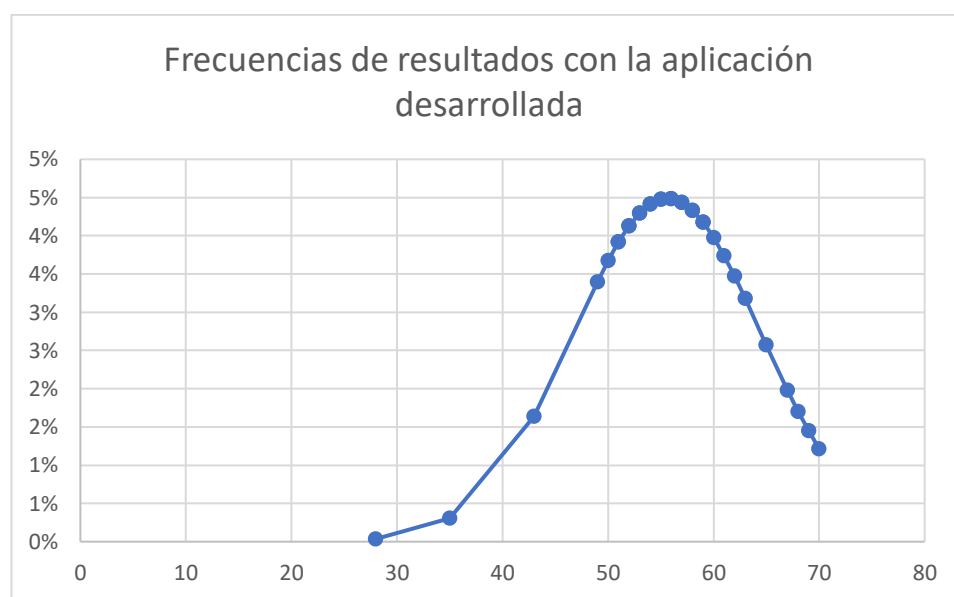


Figura 39. Polígono de frecuencias de los tiempos de creación de ilustraciones con la aplicación desarrollada en segundos

La **Figura 39**. Polígono de frecuencias de los tiempos de creación de ilustraciones con la aplicación desarrollada muestra los resultados detallados del tiempo requerido por cada participante para completar la creación de ilustraciones, indicando tanto los segundos consumidos como la distribución porcentual de frecuencia. El tiempo promedio registrado fue de 55,6 segundos, con una desviación estándar de 8,88 segundos. La mayor concentración de estudiantes se ubicó en valores intermedios, aproximadamente entre 50 y 60 segundos, mientras que solo unos pocos participantes

lograron completar la tarea en tiempos significativamente menores, cercanos a los 30 segundos.

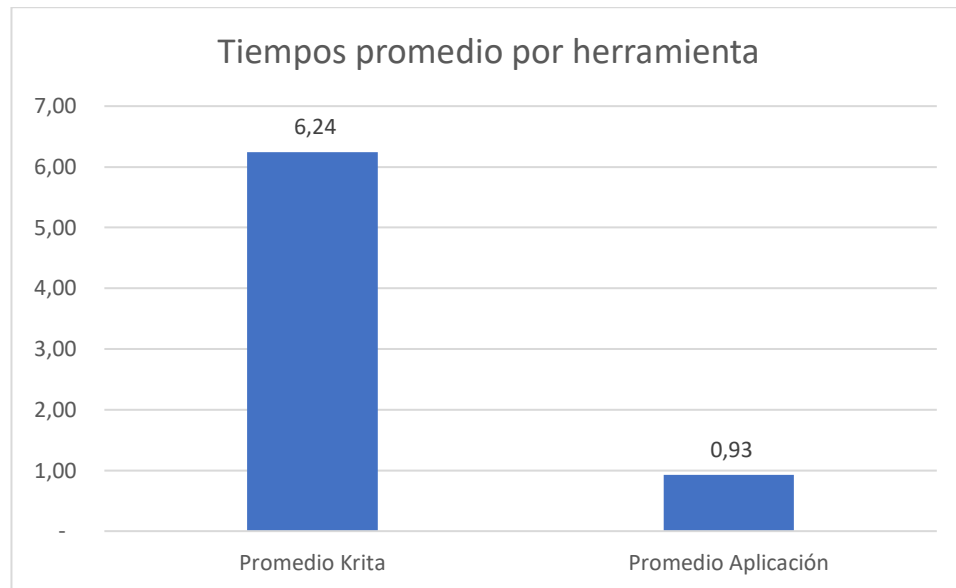


Figura 40. Comparación de promedios de tiempos de creación de ilustraciones en minutos

En la **Figura 40**. Comparación de promedios de tiempos de creación de ilustraciones en minutos se muestra la comparación de los tiempos promedio usados en la creación de ilustraciones 2D de figuras humanas. Los resultados evidencian que el proceso realizado con Krita registra un promedio de 6,24 minutos, mientras que la aplicación desarrollada alcanza un promedio de 0,93 minutos. Esto representa una disminución de 5,31 minutos en el tiempo de producción, equivalente a reducir el tiempo de creación a solo el 15% del original.

CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- El análisis realizado sobre diversas herramientas de edición y generación de imágenes permitió identificar patrones claros en cuanto a su nivel de complejidad y facilidad de uso. Plataformas profesionales como Adobe Photoshop, junto con alternativas de código abierto como Krita, requieren un nivel de experiencia mayor debido a la necesidad de manejar capas, configuraciones avanzadas y procesos manuales. En contraste, soluciones basadas en IA como Stable Diffusion o MidJourney, simplifican la creación al permitir generar contenido a partir de texto, aunque su capacidad de edición suele ser más limitada. Herramientas más accesibles, como Canva, destacan por su interfaz intuitiva y el uso de plantillas preconstruidas, lo que reduce significativamente la curva de aprendizaje. Esta comparación permitió identificar qué elementos como el soporte de IA, la disponibilidad de recursos predefinidos y la claridad de la interfaz resultan esenciales para disminuir la complejidad y evitar procesos repetitivos, especialmente para usuarios sin experiencia previa.
- La definición de los requerimientos funcionales estableció las directrices para la interacción del usuario con la aplicación, priorizando una operatividad sencilla, la personalización de modelos 2D, la incorporación de fondos y la exportación de ilustraciones en formatos estándar. Estos lineamientos permitieron orientar el desarrollo hacia una experiencia de uso fluida y con una curva de aprendizaje mínima, optimizando además la eficiencia en la generación de ilustraciones de figuras humanas en 2D. Como resultado, la aplicación desarrollada logró reducir el tiempo de creación en aproximadamente un 85% en comparación con el proceso equivalente realizado en Krita.
- El desarrollo de la aplicación se realizó integrando tecnologías de inteligencia artificial, como MediaPipe, junto con herramientas gráficas y de interfaz,

logrando una generación automática de poses humanas en modelos 2D. La implementación demostró que la combinación de IA y gráficos interactivos reduce significativamente el tiempo y la complejidad del proceso de ilustración, proporcionando un sistema funcional y adaptable a las necesidades de usuarios sin experiencia previa.

- La implementación de la aplicación para la generación de ilustraciones 2D de figuras humanas utilizando recursos de inteligencia artificial demostró una mejora significativa en la accesibilidad del proceso creativo para usuarios sin experiencia en herramientas de edición. La solución desarrollada ofrece una interfaz simplificada, un ejecutable de bajo peso que facilita su despliegue y uso, y evidencia una reducción importante en el tiempo requerido para la creación de las imágenes.

4.2 Recomendaciones

- Se recomienda identificar aquellas tareas en otros procesos más allá de la generación de imágenes, pero que aún requieren intervención o toma de decisiones por parte del usuario, y evaluar cómo la Inteligencia Artificial podría asistir en estas actividades. Esta implementación permitiría reducir la carga manual manteniendo la interacción y la personalización.
- Se sugiere fomentar la difusión y uso de la aplicación en entornos educativos. La promoción podría incluir demostraciones, o ilustraciones creadas con la ayuda de la aplicación para mejorar la productividad y la experiencia de aprendizaje. Esto permitirá aumentar la adopción y obtener retroalimentación valiosa para futuras mejoras.
- Se recomienda que investigaciones futuras amplíen el alcance del estudio, explorando no solo la generación de imágenes con recursos de IA, sino también la implementación de técnicas para la creación de animaciones 2D a partir de secuencias de video de personas, lo que permitiría evaluar nuevas posibilidades de automatización y análisis en entornos gráficos.

REFERENCIAS BIBLIOGRÁFICAS

- [1] K. T. Nguyen, L. X. Vu, y T. T. Q. Nguyen, «Turning Human into 2D Virtual Character Using Real-Time Motion Capture», en *Cooperative Design, Visualization, and Engineering*, Y. Luo, Ed., Cham: Springer Nature Switzerland, 2023, pp. 214-224.
- [2] A. Yang, G. Liu, W. Naeem, D. Wu, Y. Zhou, y L. Chen, «A monocular 3D human pose estimation approach for virtual character skeleton retargeting», *J. Ambient Intell. Humaniz. Comput.*, vol. 14, n.º 7, pp. 9563-9574, 2023, doi: 10.1007/s12652-023-04629-2.
- [3] A. Cannavò, R. Pesando, y F. Lamberti, «A Framework for Animating Customized Avatars from Monocular Videos in Virtual Try-On Applications», en *Extended Reality*, L. T. De Paolis, P. Arpaia, y M. Sacco, Eds., Cham: Springer Nature Switzerland, 2023, pp. 69-88.
- [4] Y.-J. Yuan, Y.-K. Lai, T. Wu, L. Gao, y L. Liu, «A Revisit of Shape Editing Techniques: From the Geometric to the Neural Viewpoint», *J. Comput. Sci. Technol.*, vol. 36, n.º 3, pp. 520-554, 2021, doi: 10.1007/s11390-021-1414-9.
- [5] Y. Xue, Y.-C. Guo, H. Zhang, T. Xu, S.-H. Zhang, y X. Huang, «Deep image synthesis from intuitive user input: A review and perspectives», *Comput. Vis. Media*, vol. 8, n.º 1, pp. 3-31, 2022, doi: 10.1007/s41095-021-0234-8.
- [6] M. Toshpulatov, W. Lee, S. Lee, y A. H. Roudsari, «Human pose, hand and mesh estimation using deep learning: a survey», *J. Supercomput.*, vol. 78, n.º 6, pp. 7616-7654, 2022, doi: 10.1007/s11227-021-04184-7.
- [7] X. Sun y J. Qin, «AI for Supporting the Freedom of Drawing», *Mach. Intell. Res.*, vol. 21, n.º 1, pp. 63-88, 2024, doi: 10.1007/s11633-023-1438-4.
- [8] K. Fukaya, D. Daylamani-Zad, y H. Agius, «Intelligent Generation of Graphical Game Assets: A Conceptual Framework and Systematic Review of the State of the Art», *ACM Comput. Surv.*, vol. 57, dic. 2024, doi: 10.1145/3708499.
- [9] J. Yao, J. Chen, L. Niu, y B. Sheng, «Scene-aware Human Pose Generation using Transformer», oct. 2023, pp. 2847-2855. doi: 10.1145/3581783.3612439.
- [10] C. Xia *et al.*, «MonoSAID: Monocular 3D Object Detection based on Scene-Level Adaptive Instance Depth Estimation», *J. Intell. Robot. Syst.*, vol. 110, n.º 1, p. 2, 2023, doi: 10.1007/s10846-023-02027-6.
- [11] Z. Gao, J. Chen, Y. Liu, Y. Jin, y D. Tian, «A systematic survey on human pose estimation: upstream and downstream tasks, approaches, lightweight models, and prospects», *Artif. Intell. Rev.*, vol. 58, n.º 3, p. 68, 2025, doi: 10.1007/s10462-024-11060-2.
- [12] M. Vijendran, J. Deng, S. Chen, E. S. L. Ho, y H. P. H. Shum, «Artificial intelligence for geometry-based feature extraction, analysis and synthesis in artistic

images: a survey», *Artif. Intell. Rev.*, vol. 58, n.º 2, p. 64, 2024, doi: 10.1007/s10462-024-11051-3.

[13] T. Fukusato, A. Maejima, T. Igarashi, y T. Yotsukura, «Exploring inbetween charts with trajectory-guided sliders for cutout animation», *Multimed. Tools Appl.*, vol. 83, n.º 15, pp. 44581-44594, 2024, doi: 10.1007/s11042-023-17354-x.

[14] H.-P. Nguyen *et al.*, «EPedit: Redefining Image Editing with Generative AI and User-Centric Design», en *Information and Communication Technology*, W. Buntine, M. Fjeld, T. Tran, M.-T. Tran, B. Huynh Thi Thanh, y T. Miyoshi, Eds., Singapore: Springer Nature Singapore, 2025, pp. 272-283.

[15] S. Shimada, V. Golyanik, W. Xu, P. Pérez, y C. Theobalt, «Neural monocular 3D human motion capture with physical awareness», *ACM Trans. Graph.*, vol. 40, pp. 1-15, ago. 2021, doi: 10.1145/3476576.3476640.

ANEXOS

Anexo A. Encuesta para la evaluación de la viabilidad y aceptación

Esta encuesta recogió datos para evaluar la viabilidad y aceptación de una aplicación para generar ilustraciones 2D de figuras humanas con una interfaz accesible. Está diseñada según tres objetivos específicos relacionados con el análisis de herramientas existentes, la evaluación del rendimiento para usuarios sin experiencia y la validación de la solución en contextos educativos. Las preguntas son cerradas y permiten obtener resultados cuantificables para guiar el desarrollo del sistema.

Encuesta

1. ¿Con qué frecuencia utiliza herramientas de edición de imagen para tareas o presentaciones?

- Nunca
- Rara vez
- Normalmente
- Frecuentemente
- Muy frecuentemente

2. ¿Conoce sobre herramientas que permiten crear personajes o figuras humanas en 2D?

- Nada
- Poco
- Algo
- Bastante
- Mucho

3. ¿Qué tan familiarizado está con herramientas como PowerPoint, Canva, Photoshop u otras similares?

- Nada familiarizado
- Poco familiarizado
- Medianamente familiarizado
- Bastante familiarizado
- Mucho familiarizado

4. ¿Con qué frecuencia has tenido dificultades para crear ilustraciones gráficas en herramientas de edición de imagen?

- Nunca
- Rara vez
- A veces
- Frecuentemente
- Siempre

5. ¿Qué nivel de dificultad percibe al usar herramientas de edición de imagen tradicionales sin experiencia previa?

- Muy bajo
- Bajo
- Medio
- Alto
- Muy alto

6. ¿Qué tan fácil le resulta actualmente crear una ilustración de personajes humanos usando herramientas que conoce?

- Muy difícil
- Difícil
- Medio
- Fácil
- Muy fácil

7. ¿Cuánto tiempo estima que te toma normalmente crear una ilustración gráfica sencilla de personajes humanos?

- Menos de 10 minutos
- Entre 10 y 20 minutos
- Entre 20 y 40 minutos
- Entre 40 minutos y 1 hora
- Más de 1 hora

8. ¿Qué tan probable es que utilice una herramienta que le facilite crear ilustraciones didácticas de personajes humanos sin necesidad de conocimientos técnicos?

- Nada probable
- Poco probable
- Neutral
- Probable
- Muy probable

9. ¿Qué tan importante le sería que la aplicación tenga una interfaz simple y fácil de entender desde el primer uso?

- Nada importante
- Poco importante
- Neutral
- Importante
- Muy importante

10. ¿Qué tan útil le sería que puedas mover partes del cuerpo de una figura humana directamente con el mouse (por ejemplo, brazos o piernas)?

- Nada útil
- Poco útil
- Neutral
- Útil
- Muy útil

11. ¿Qué tipo de retroalimentación visual le gustaría recibir al interactuar con la figura (por ejemplo, al seleccionar una parte del cuerpo)?

- Cambio de color
- Contorno resaltado
- Etiqueta con nombre de la parte
- Sonido

- No necesito retroalimentación

12. ¿Qué tan útil le sería que la aplicación permita reiniciar el proceso en cualquier momento?

- Nada útil
- Poco útil
- Neutral
- Útil
- Muy útil

13. ¿Qué tan útil le sería que la aplicación permita replicar una postura corporal a partir de una fotografía?

- Nada útil
- Poco útil
- Neutral
- Útil
- Muy útil

14. ¿Una herramienta como esa debería incluir instrucciones o ayudas contextuales al usar cada herramienta?

- No
- Poco
- Neutral
- Sí
- Mucho

Anexo B. Resultados de la encuesta y de la submuestra.

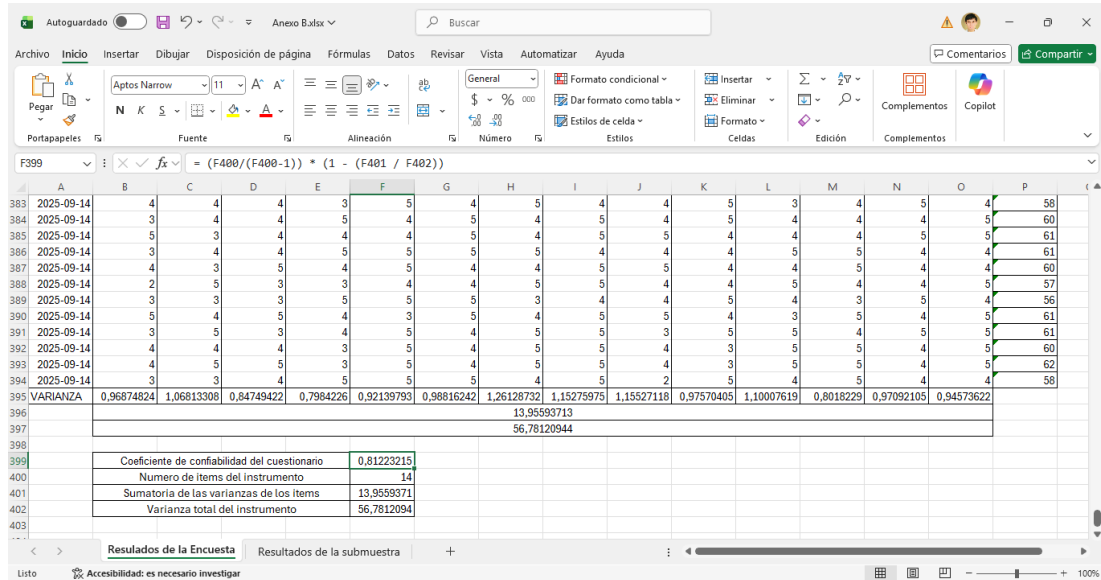


Figura 41. Vista del documento de los resultados y la submuestra.

- Vínculo al documento completo: [Anexo B.xlsx](#)

Anexo C. Levantamiento de requerimientos – IEEE 830

Mediante el estándar IEEE específicamente el ANSI/IEEE 830-1998, referente a la Práctica Recomendada para la Especificación de Requisitos de Software, se elaboró la definición de los requerimientos del sistema de generación de ilustraciones 2D de figuras humanas mediante el uso de recursos basados en inteligencia artificial. El propósito de esta especificación es describir de manera estructurada los requerimientos funcionales y no funcionales para la base para el diseño y desarrollo de la aplicación, siguiendo las necesidades de los usuarios sin experiencia en programas de edición de imágenes. A continuación, se presentan detalladamente los requerimientos del sistema.

Identificador	RF01
Nombre	Modelo 2D manipulable por el usuario
Características	Permite al usuario manipular un modelo 2D de una figura humana compuesta por imágenes creadas previamente, cuyas partes estén conectadas.

Descripción	El sistema debe permitir que el usuario mueva las partes del modelo haciendo clic izquierdo y las rote con clic derecho, manteniendo las conexiones entre las articulaciones sin desalinearse.
Requerimientos no funcionales	RNF01: El sistema debe resaltar visualmente las partes del cuerpo al ser seleccionadas, permitiendo al usuario identificar fácilmente qué elemento está activo dentro del modelo 2D.
Prioridad del requerimiento	Alta
Criterios de aceptación	- El modelo puede moverse y rotarse con los clics correspondientes. - Las articulaciones mantienen la conexión entre partes.
Observaciones	Al iniciar el programa el modelo debe tener una postura definida por defecto.

Identificador	RF02
Nombre	Cuadro de análisis de postura con imágenes
Características	Permite al usuario arrastrar imágenes de personas a un cuadro destinado al análisis de postura
Descripción	El sistema debe detectar la postura de la persona en la imagen usando un recurso de inteligencia artificial y adaptar la postura del modelo 2D a una posición similar, mostrando la fotografía con el esquema del modelo superpuesto.
Requerimientos no funcionales	RNF01: El reconocimiento de postura debe realizarse en menos de tres segundos para darle al usuario una experiencia ágil. RNF02: La superposición del esquema debe ser visualmente clara para que el usuario identifique fácilmente la postura detectada.
Prioridad del requerimiento	Alta
Criterios de aceptación	- El sistema detecta correctamente la postura de la persona. - El modelo 2D adopta la postura identificada. - La imagen original permanece visible con el esquema superpuesto.
Observaciones	Para la funcionalidad de este requerimiento se utilizó la Liberia MediaPipe.

Identificador	RF03
Nombre	Guardar modelo como imagen PNG
Características	Permite al usuario guardar el modelo 2D en su postura actual como una imagen en formato PNG.
Descripción	El sistema debe capturar la postura actual del modelo y generar un archivo PNG con la imagen correspondiente del modelo 2D.
Requerimientos no funcionales	RNF01: El proceso no debe demorar más de dos segundos para garantizar una respuesta rápida para mantener la fluidez de uso del usuario.
Prioridad del requerimiento	Alta
Criterios de aceptación	- Se genera correctamente el archivo PNG. - El archivo contiene la postura y apariencia actual del modelo.
Observaciones	El fondo de la imagen será transparente por defecto al guardarse.

Identificador	RF04
Nombre	Fondo personalizable del modelo
Características	Permite al usuario seleccionar una imagen como un fondo detrás del modelo 2D mediante un checkbox.
Descripción	El sistema tiene el fondo "Transparente" seleccionado por defecto y, al seleccionar la opción del checkbox "Añadir fondo", abre una ventana que permite elegir una imagen para establecerla como fondo.
Requerimientos no funcionales	RNF01: Las imágenes deben adaptarse al tamaño del lienzo para asegurar una visualización completa al usuario.
Prioridad del requerimiento	Alto
Criterios de aceptación	- El fondo predeterminado es transparente. - Al marcar el checkbox, se puede elegir una imagen. - El fondo se muestra correctamente detrás del modelo.
Observaciones	Debe soportar formatos JPG y PNG.

Identificador	RF05
Nombre	Restablecer postura por defecto
Características	Permite al usuario devolver el modelo 2D a su postura inicial por defecto mediante un botón.
Descripción	El sistema debe restablecer la posición y rotación de todas las partes del modelo al presionar el botón.
Requerimientos no funcionales	RNF01: La acción debe ejecutarse en menos de un segundo para ofrecer una respuesta inmediata al usuario.
Prioridad del requerimiento	Media
Criterios de aceptación	- El modelo vuelve a la postura inicial. - No se altera la apariencia ni expresiones activas.
Observaciones	La postura por defecto no puede cambiarse.

Identificador	RF06
Nombre	Selector de apariencias del modelo
Características	Permite al usuario cambiar la apariencia visual del modelo 2D mediante un botón que despliega una galería de opciones.
Descripción	El sistema debe desplegar un conjunto de variaciones de la apariencia del modelo al presionar el botón correspondiente y aplicar la apariencia seleccionada sin alterar la postura actual del modelo.
Requerimientos no funcionales	RNF01: Las apariencias deben cargarse en menos de dos segundos para garantizar una transición fluida entre modelos al usuario.
Prioridad del requerimiento	Media
Criterios de aceptación	- El usuario puede cambiar la apariencia del modelo. - El modelo mantiene sus posiciones y articulaciones.
Observaciones	Las apariencias deben almacenarse en carpetas organizadas y ser compatibles con el formato de los modelos definidos por defecto.

Identificador	RF07
Nombre	Selector de expresiones faciales
Características	Permite al usuario cambiar la expresión facial del modelo 2D mediante un botón que despliega una galería de opciones.
Descripción	El sistema debe mostrar las expresiones disponibles según el modelo seleccionado y aplicarlas instantáneamente al rostro del modelo.
Requerimientos no funcionales	RNF01: El cambio de expresión debe ejecutarse instantáneamente para que el usuario perciba de forma inmediata el cambio. RNF02: Las expresiones deben corresponderse al modelo activo, para dar un proceso dinámico al usuario.
Prioridad del requerimiento	Media
Criterios de aceptación	- El botón muestra expresiones disponibles. - Las expresiones se aplican correctamente. - Las opciones cambian según el modelo activo.
Observaciones	Cada modelo debe tener un conjunto propio de expresiones predefinidas.

Anexo D. Video tutorial de la aplicación desarrollada.

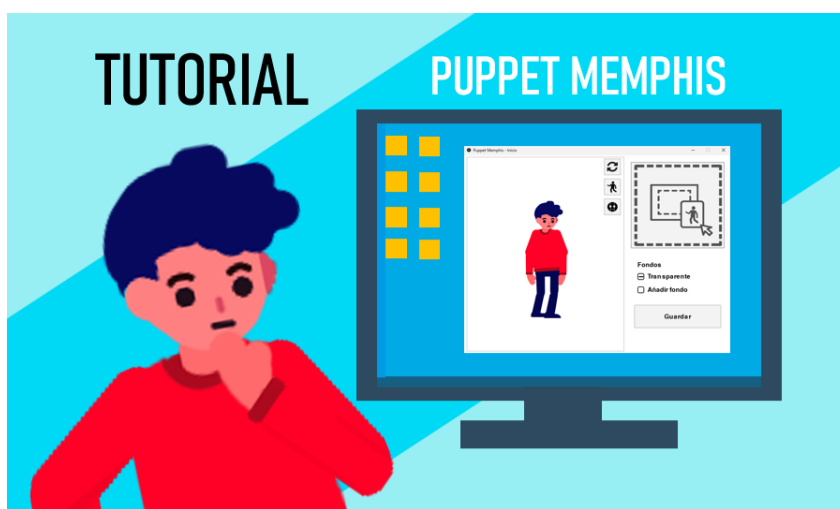


Figura 42. Portada del video tutorial.

- Vínculo al documento completo: tutorial.mp4