

RAE (Resumen Analítico Estructurado)

1. **TIPO DE DOCUMENTO:** Trabajo de grado para optar por el título de Ingeniero Multimedia
2. **TÍTULO:** Generador de personajes cartoon en el software Autodesk Maya para optimizar la eficiencia en el pipeline de animación 3D
3. **AUTOR (ES):** Santiago Molina Gutiérrez, William Felipe Muñoz Arias, Juan Felipe Quintero Perdomo
4. **LUGAR:** Bogotá, D.C
5. **FECHA:** Julio de 2022
6. **PALABRAS CLAVES:** Plugin, Autodesk Maya, Pipeline, Generador, Modelado, Personajes.
7. **DESCRIPCIÓN DEL TRABAJO:** Este proyecto busca optimizar el tiempo en el pipeline de modelado 3D implementado en películas animadas y videojuegos mediante un plugin el cual es un complemento que está desarrollado por medio del lenguaje de programación Python que se integra dentro del software Autodesk Maya, este plugin se maneja por medio de un panel con botones y sliders que crearán figuras 3D eficaz y eficientemente.
8. **LÍNEA DE INVESTIGACIÓN: LÍNEAS DE INVESTIGACIÓN:** Línea de Investigación de la USB: Tecnologías actuales Multimedia
Sub línea de Facultad de Ingeniería: Control de Procesos de diseño 3D.
Campo Temático del Programa: Generador de personajes cartoon en el software Autodesk Maya.
9. **METODOLOGÍA:** Metodología espiral realizando los procesos por iteraciones
10. **CONCLUSIONES:** My Toon Creator al ser un plugin no solo se convierte en un complemento que añade nuevas características o mejoras al programa, en este caso al software de Autodesk Maya, sino que también evita instalar aplicaciones de terceros los cuales podrían cumplir la misma función, pero requerirían mayor almacenamiento en el computador del usuario, afectando el rendimiento de este. Además, My Toon Creator fue creado usando el lenguaje de programación Python, el cual al ser un lenguaje muy versátil podría ser implementado en otro software de modelado 3D como 3Ds Max, Blender, entre otros. Al tener un panel de creación el cual está conformado con textos, botones, sliders e imágenes permite al usuario comprender rápidamente las funciones del plugin, además de otorgar libertad y comodidad a la hora de la creación y la personificación de sus personajes.

Generador de personajes cartoon en el software Autodesk Maya para optimizar la eficiencia en el pipeline de animación 3D

Santiago Molina Gutiérrez
William Felipe Muñoz Arias
Juan Felipe Quintero Perdomo

Trabajo de Grado presentado para optar al título de Ingeniero Multimedia

Asesor: Danny Bercelio González Pinzón, Especialista (Esp)
Co asesor: Johana Carolina Martínez Ballesteros, Especialista (Esp)



UNIVERSIDAD DE
SAN BUENAVENTURA
COLOMBIA

Universidad de San Buenaventura
Facultad de Ingeniería (Bogotá)
Ingeniería Multimedia
Bogotá D.C., Colombia
2022

CONTENIDO

RESUMEN.....	7
ABSTRACT	8
I. INTRODUCCIÓN	9
II. PLANTEAMIENTO DEL PROBLEMA.....	11
A. Antecedentes	11
III. JUSTIFICACIÓN.....	14
IV. OBJETIVOS	15
A. Objetivo general	15
B. Objetivos específicos	15
V. PROBLEMA DE INVESTIGACIÓN.....	16
VI. ALCANCES Y LIMITACIONES	17
A. Alcances	17
B. Limitaciones	17
Tabla 1. (Conceptos de herramientas 3D Autodesk Maya)	19
VII. METODOLOGÍA DE DESARROLLO	20
Tabla 2. (Ciclos de planificación del modelo espiral).....	21
VIII DISEÑO INGENIERIL.....	23
A. Creación de modelos 3D	25
IX RESULTADOS.....	42
Resultados obtenidos con el plugin en la creación de los personajes Cartoon 3D.....	52
X. DISCUSIÓN.....	54
XI. CONCLUSIONES	55
Referencias	56

INDICE DE TABLAS

Tabla 1. (Conceptos de herramientas 3D Autodesk Maya)	19
Tabla 2. (Ciclos de planificación del modelo espiral).....	21

LISTA DE FIGURAS

1 Figura	20
2 Figura	24
3 Figura	25
4 Figura	26
5 Figura	27
6 Figura	27
7 Figura	28
8 Figura	28
9 Figura	29
10 Figura	29
11 Figura	30
12 Figura	30
13 Figura	30
14 Figura	31
15 Figura	31
16 Figura	32
17 Figura	32
18 Figura	33
19 Figura	33
20 Figura	34
21 Figura	34
22 Figura	35

23 Figura	35
24 Figura	36
25 Figura	36
26 Figura	37
27 Figura	38
28 Figura	38
29 Figura	38
30 Figura	39
31 Figura	39
32 Figura	40
33 Figura	40
34 Figura	41
35 Figura	41
36 Figura	42
37 Figura	43
38 Figura	44
39 Figura	45
40 Figura	46
41 Figura	47
42 Figura	48
43 Figura	49
44 Figura	51
45 Figura	52
46 Figura	53
47 Figura	53

RESUMEN

Este proyecto busca optimizar el tiempo en el pipeline de modelado 3D implementado en películas animadas y videojuegos mediante un plugin el cual es un complemento que está desarrollado por medio del lenguaje de programación Python que se integra dentro del software Autodesk Maya, este plugin se maneja por medio de un panel con botones y sliders que crearán figuras 3D eficaz y eficientemente.

Para el desarrollo de este proyecto se implementa la metodología de desarrollo tipo espiral, ya que su principal uso es para el desarrollo de software, y este tiene como fin los ciclos de planificación, análisis de riesgo, implementación y evaluación, de esta forma su desarrollo es iterativo por consiguiente una vez terminado un proceso y este es aprobado se continúa con otro.

Como resultado se obtiene el respectivo plugin el cual está compuesto por dos carpetas, la primera con el código de programación que está ubicada en la carpeta de Autodesk Maya del computador y la segunda que está ubicada en el disco local C, la cual está compuesta por las imágenes de referencia que se usan en el panel como ayuda para la creación del personaje y los modelos 3D que se importan directamente en el software.

Para evaluar el desarrollo del plugin se realizan las respectivas pruebas de usuario, de las cuales se obtienen resultados positivos ya que es una herramienta que ahorra tiempo para los procesos mencionados anteriormente y que tiene presente la ventaja de que es instalada directamente en Autodesk Maya.

***Palabras clave* — Plugin, Autodesk Maya, Pipeline, Generador, Modelado, Personajes.**

ABSTRACT

This project seeks to optimize the time in the 3D modeling pipeline implemented in animated films and video games through a plugin which is a complement developed through the python programming language that is integrated into the Autodesk Maya software, this plugin is handle by a panel with buttons and sliders that create 3D figures effectively and efficiently.

For the development of this project, the spiral-type development methodology is implemented, since its main use is for software development, and its purpose is planning cycles, risk analysis, implementation and evaluation, in this way its development is iterative, therefore, once a process is finished and it is approved, it continues with another.

As a result, the respective plugin is obtained, which is make up of two folders, the first with the programming code that is located in the Autodesk Maya folder of the computer and the second is located in the local disk C, which is make up of the reference images that are used in the panel to help create the character and the 3D models that are imported directly into the software.

To evaluate the development of the plugin, the respective user tests are carry out, from which positive results are obtained since it is a tool that saves time for the processes mentioned above and has the advantage to be installed directly in Autodesk Maya.

Keywords — **Plugin, Autodesk Maya, Pipeline, Generador, Modeling, Characters.**

I. INTRODUCCIÓN

La presente investigación se refiere al tema de las aplicaciones utilizadas por las industrias que están enfocadas en áreas como las películas, videojuegos o animaciones, como, por ejemplo: Adobe Fuse CC, MakeHuman, Poser, IClone, etc.

La función principal de dichas aplicaciones es que permiten optimizar la eficiencia en términos de tiempo en el pipeline de animación 3D, la cual según la definición dada por (Mercado, 2020) “el pipeline es la organización y administración del paso de la información que existe dentro de una producción de computación gráfica “. Dado que estos programas son ajenos al software de modelado 3D como Maya, Blender, 3Ds Max, entre otros, tienen un costo adicional y ocupan una cantidad de almacenamiento significativa.

Teniendo en cuenta lo anterior, el objetivo de este proyecto es dar a conocer una solución por medio de un plugin el cual agrega funciones o características adicionales directamente en Autodesk Maya, evitando instalaciones de aplicaciones externas y cantidades de almacenamientos excesivas.

El plugin al ser creado mediante el lenguaje Python permite la posibilidad de implementarlo en diversos softwares de modelado 3D y además de cumplir con la función de optimizar el pipeline de animación 3D también cuenta con la característica de que la creación de los modelos tenga un estilo cartoon, esto debido a que es un estilo poco utilizado en dichas aplicaciones y muy conocido a nivel mundial.

Por consiguiente, el proyecto se realizó mediante la metodología de desarrollo tipo espiral, la cual es un modelo enfocado en el desarrollo de software y se caracteriza por los ciclos de planificación, análisis de riesgo, implementación y evaluación.

En el capítulo 1 se realizó la búsqueda de antecedentes sobre las aplicaciones enfocadas en la creación de los personajes 3D y en los problemas o deficiencias que tenían dichas aplicaciones. De la misma forma se explican las funciones utilizadas en el desarrollo del plugin en el marco conceptual y la metodología empleada.

El capítulo 2 se enfoca en el proceso de desarrollo en el diseño y creación del panel, en la creación de los personajes 3D con su estilo cartoon y la programación de las funciones de vinculación, limpieza y optimización de modelos al panel de creación.

En el capítulo 3 se analizaron los resultados de las pruebas de usuario realizadas a los estudiantes de la universidad de san Buenaventura de Bogotá, que estuvieran cursando desde cuarto semestre en adelante, esto con el fin de que tuvieran el conocimiento adecuado sobre el software de modelado 3D Autodesk Maya en el cual estaba implementado el plugin.

II. PLANTEAMIENTO DEL PROBLEMA

El proceso de personalizar un personaje puede llevar mucho tiempo si se quiere hacer alguno en específico es por esto que existen diferentes plugin o diferentes herramientas que sirven para generar personajes a partir de librerías o generadores para hacer más eficaz el proceso de modelado, sin embargo la mayoría de estos programas se basa en personajes humanos, es por esto que se quiere realizar esta herramienta, para optimizar el tiempo que se puede tardar creando un personaje en específico, por esta misma razón se quiere mejorar el pipeline con personajes cartoon que se puedan utilizar en animación, caricatura y videojuegos.

A. Antecedentes

En esta investigación se encontraron algunas referencias de programas que ayudarán en el estudio del diseño y modelado 3D tanto para topologías humanas como cartoon. A pesar de que no se han realizado una gran cantidad de estudios sobre el tema, (González, 2020) en su trabajo de grado sobre diseño y modelado de personajes, afirma que el modelado en 3D “es una representación geométrica de cualquier objeto tridimensional”. El proceso de modelado tiene básicamente dos maneras de realizarlo, manual o automáticamente; es decir, en la forma manual “se prepara la información geométrica para los gráficos 3D de forma similar a como se elabora en las artes plásticas y la escultura” (González, 2020) o se realiza bajo aplicaciones de modelado que son software especializados como es el caso de la forma automática.

Otros estudios como es el de (Asunción, 2017) el cual tiene el objetivo de buscar en bases teóricas, conceptos y técnicas a la hora de construir el personaje y además expone las fases para la creación de este, como lo son la preproducción, producción y postproducción del proyecto “Diseño y creación de un personaje 3D para un videojuego o animación”.

Para la realización de este trabajo fue necesario acudir a la información que permitiera evidenciar cómo poder hacer una herramienta (plugin) para generar modelos 3D en el software Autodesk maya. Como referencia se encontraron los siguientes programas:

Adobe fuse CC

La autora (Mathews, 2019) afirma que el programa Adobe Fuse CC permite crear personajes con rapidez empleando una biblioteca de contenido en 3D de gran calidad, personalizar el color, la textura y la forma de más de 280 atributos, incluido el cabello, los anteojos y los tejidos de la ropa; además de esto admite cambiar con facilidad el tamaño, las proporciones de un personaje, la ropa y el resto de los atributos se ajustarán automáticamente.

MakeHuman

Como lo expone (González, 2020) en su estudio sobre diseño y modelado de personajes la define como “una aplicación de gráficos 3D por computadora para prototipado de humanoides fotorrealistas para ser utilizados en diferentes aplicaciones de gráficos por ordenador”. Así mismo, menciona que tiene la función de modificar un modelo inicial en cuanto a atributos de forma.

Poser

Según la definición de la página web de Poser 12 (Poser, 2021) es una herramienta que le permite a diseñadores y artistas digitales a producir más fácilmente animaciones 3D e imágenes digitales; tanto así que desde 1995 se ha convertido en la herramienta más usada y referenciada por expertos en el tema ya que facilita y mejora la productividad del proyecto puesto en marcha.

iClone

“iClone es el software de animación 3D en tiempo real más rápido de la industria, que le ayuda a producir fácilmente animaciones profesionales para películas, animación, videojuegos, desarrollo de contenido, educación y arte. Integrado con las últimas tecnologías en tiempo real, iClone 7 simplifica el mundo de la animación 3D en un entorno de producción fácil de usar que combina la animación de personajes, el diseño de escenas y la narración cinematográfica; convirtiendo rápidamente su visión en realidad”. (Software, 2021)

Cartoon Character Designer

Esta herramienta permite según la tienda virtual Content store (Designer, 2021) modificar personajes cartoon base, es decir vienen 5 personajes con una respectiva topología en las que el usuario puede elegir entre ellas y agregar geometrías o cambiar su respectiva forma, como, por ejemplo, nariz, ojos, cachetes, boca, pelo, etc. Y tiene un costo de \$199.00 dólares.

III. JUSTIFICACIÓN

Los personajes 3D se usan en diversas áreas como las películas, animación y videojuegos. Por lo tanto, para facilitar la creación de personajes 3D existen varias herramientas y aplicaciones que realizan o facilitan este proceso, pero la mayoría se basa en la creación de personajes humanos, con esto presente se quiere realizar un generador de personajes 3D, pero enfocado al estilo cartoon, porque además de no haber muchas aplicaciones que realicen este estilo de personajes, estos pueden realizarse con geometrías más sencillas, lo cual puede facilitar la creación del generador.

Debido a que el bagaje de estudios sobre el tema en cuestión es muy corto, el presente proyecto tiene como objeto la optimización de recursos necesarios para el desarrollo y creación de personajes por medio de un plugin (aplicación que añade funcionalidad adicional o una nueva característica al software); esto con el fin de evitar descargarlo y que además contribuirá con el ahorro de memoria del dispositivo en uso ya que este vendrá adherido directamente al programa de modelado 3D. Debido a que gran parte de las aplicaciones se basan en la topología humana, el proyecto tendrá como principal enfoque el estilo cartoon, para así, poder crear personajes con características específicas que lo haga atractivo al público.

IV. OBJETIVOS

A. Objetivo general

Desarrollar una herramienta (plugin) capaz de generar personajes tipo cartoon a partir de botones, sliders, y selecciones, para optimizar el pipeline de animación 3D.

B. Objetivos específicos

- Diseñar el panel de creación de personajes en Autodesk maya.
- Generar modelos 3D de las partes del cuerpo.
- Programar las funciones de vinculación, limpieza y optimización de modelos. al panel de creación.
- Realizar pruebas de usuario y evaluación de las funciones de la herramienta.

V. PROBLEMA DE INVESTIGACIÓN

El proceso de personalizar un personaje puede llevar mucho tiempo si se quiere hacer alguno en específico es por esto que existen diferentes plugin o diferentes herramientas que sirven para generar personajes a partir de librerías o generadores para hacer más eficaz el proceso de modelado, sin embargo la mayoría de estos programas se basa en personajes humanos, es por esto que se quiere realizar esta herramienta, para optimizar el tiempo que se puede tardar creando un personaje en específico, por esta misma razón se quiere mejorar el pipeline con personajes cartoon que se puedan utilizar en animación, caricatura y videojuegos.

Para concluir, la pregunta a la cual este proyecto se enfocó es la siguiente: ¿Cómo se puede optimizar el pipeline a través de un plugin que mejore el tiempo de los interesados en crear personajes cartoon 3D para videojuegos y animación?

VI. ALCANCES Y LIMITACIONES

A. Alcances

- En esta fase del proyecto se desarrollará el módulo de modelado para permitir a futuro expandirlo a rigging y texturizado.
- Al realizarlo en el lenguaje Python se podrá implementar en otros tipos de plataforma 3D como (3DS Max, blender, etc.). Ya que Python es un lenguaje de programación multiplataforma.

B. Limitaciones

- Inicialmente las pruebas se llevarán a cabo con los estudiantes de la universidad en los semilleros de investigación.
- Principalmente se realiza el plugin en maya porque es la suite de 3D con la que cuenta la universidad.
- Por limitación de tiempo se desarrollará solamente la sección de modelado.

VII. MARCO CONCEPTUAL

Como ha sido expuesto por (González, 2020), en la creación de personajes 3D existen herramientas como adobe fuse, MakeHuman, Poser, entre otras; estas facilitan su desarrollo, y usan un modelo humano base en el que, por medio de una interfaz se modifica la topología o rasgos del modelo, como lo son los rasgos físicos del personaje. La función principal es optimizar el pipeline de desarrollo, ya que consiste en el proceso de creación de un personaje 3D.

Para esto se implementará un plugin de modelado 3D, el cual busca facilitar el trabajo de quien acude a él, evitando instalar más programas, puesto que puede hacer difícil el funcionamiento del equipo y la realización de personajes 3D con estilo cartoon en Autodesk maya. Esta herramienta tendrá incorporada una serie de paneles en los cuales el usuario podrá, por medio de ellos, seleccionar el estilo y modificar la topología del modelo cartoon a su gusto y al final el usuario escoja las características de su modelo, con las funciones mencionadas en la Tabla 1 según los términos de (Autodesk, 2018) se creará el modelo con las especificaciones del usuario y listo para ser usado.

Retopology	Permite reconstruir modelos que tengan técnicas esculpidas que se encuentran en el mismo programa que generan errores en ella.
Create	Esta herramienta permite crear polígonos tales como (esferas, cubos, curvas, cilindros, etc.). Estos objetos se pueden modificar para generar modelos funcionales.
Extrude	Esta herramienta permite seleccionar una cara del objeto y deformarla en base a vértices y caras seleccionadas
Combine	Esta herramienta permite combinar dos geometrías distintas como una sola pieza.
Poly Remesh	Permite que el modelo seleccionado sea topológicamente correcto. Buscando y arreglando geometrías que tengas más de 4 lados

Bridge	Esta herramienta permite que por medio de dos bordes de una misma figura se cree una cara entre ellos.
Bevel	Permite disminuir la nitidez de un borde dividiéndolo y expandiendo en una nueva cara en la geometría.

TABLA 1. (CONCEPTOS DE HERRAMIENTAS 3D AUTODESK MAYA)

Todo esto se podrá realizar mediante el lenguaje de programación Python, el cual como lo expone (VISUS, 2020) en su artículo sobre Python, está orientado a objetos y código abierto con una sintaxis medianamente sencilla y con una amplia biblioteca de herramientas; además de esto en su artículo declara que

“Python es ampliamente utilizado por empresas de todo el mundo para construir aplicaciones web, analizar datos, automatizar operaciones y crear aplicaciones empresariales fiables y escalables. Muchas compañías tecnológicas, entre las que se encuentran Google, Uber, Netflix y Facebook, llevan años utilizando este lenguaje de programación y construyendo su infraestructura tecnológica basándose en él. En el sector financiero, grandes firmas como Goldman Sachs, JP Morgan o PayPal lo consideran uno de los requisitos básicos en muchos de sus procesos de selección de personal”.

Teniendo en cuenta lo anterior para continuar con la creación del panel, es necesario conocer cuáles son los pasos para seguir con el fin de crear los modelos 3D para posteriormente implementarlos en el panel del plugin, como lo expone (Asunción, 2017), en primer lugar, se realiza la selección de referencias de los atributos físicos del personaje. En segundo lugar, se estudian las técnicas de modelado que se usarán para llevar a cabo este proceso de diseño de personajes, y finalmente, se realizan bocetos hasta definir el estilo deseado de los elementos, luego se procederá a crearlos usando los métodos y técnicas necesarias.

VII. METODOLOGÍA DE DESARROLLO

Para llevar a cabo este proyecto se va a usar la metodología de desarrollo tipo espiral, el cual es un modelo enfocado en el desarrollo de software. El modelo espiral según (Fariño, 2011) “es un modelo meta del ciclo de vida del software donde el esfuerzo del desarrollo es iterativo, tan pronto culmina un esfuerzo del desarrollo por ahí mismo comienza otro”.

En el modelo de espiral (Fariño, 2011), expone los siguientes cuatro pasos principales:

1. Determinar y fijar los objetivos
2. Análisis de riesgos
3. Desarrollar, verificar y validar
4. Planificar

Como se puede observar en la **Figura 1**, el modelo en espiral se caracteriza por los ciclos de planificación, análisis de riesgo, implementación y evaluación los cuales serán explicados a detalle en la **Tabla 2**. Este modelo será utilizado para dar continuidad al proyecto en cuestión.

1 Figura
Modelo espiral



Nota: (gems, 2019)

Planificación	Se determinan los objetivos y el alcance del ciclo que comienza, después de realizar su respectiva investigación, también se discuten las posibles alternativas y se especifican las condiciones como por ejemplo los sistemas operativos o los lenguajes de programación. Por cada iteración o ciclo realizado, se irá incrementado el tamaño del software entregado y su funcionalidad.
Análisis de Riesgo	Se realiza una evaluación de todo aquello que pueda afectar o influenciar el progreso del proyecto según el estado en que se encuentre y su grado de avance, también se evalúan las alternativas existentes. Para ello, se diseñan los prototipos o simulaciones que deberán ser validados en el ciclo.
Implementación	Los prototipos se amplían y se añaden funcionalidades. Se desarrolla y valida el software según los objetivos y alcances establecidos, los cuales están relacionados y condicionados con el análisis de riesgos anterior y se realizan una gran variedad de pruebas hasta que el software pueda ser implementado
Evaluación	Antes de realizar otro ciclo en el espiral, se evalúa lo sucedido en el ciclo anterior. Se realiza un análisis detallado sobre los riesgos detectados anteriormente y se verifica si dichos riesgos mencionados ya tuvieron una solución. Después se determina el avance del proyecto y hacia dónde debe enfocarse el próximo ciclo.

TABLA 2. (CICLOS DE PLANIFICACIÓN DEL MODELO ESPIRAL)

Usando la metodología Espiral como base se desarrollaron ciclos con cada uno de los objetivos mencionado anteriormente como por ejemplo el diseño del panel de creación de personajes.

En la fase de planeación se determinó que el panel debería estar conformado por botones, sliders, textos e imágenes, esto con el fin de que el usuario tuviera una interfaz más estética y fácil de entender, estableciendo Python como el lenguaje para la creación de dicho panel.

Se evaluó que, dependiendo de la cantidad de modelos y opciones de personalización, el panel se vería afectado y se deberían agregar más funciones al mismo, por consiguiente, se

desarrollaron prototipos del panel para después ser evaluados y validados. Así mismo se agregaron mejoras y funcionalidades basados en los errores evidenciados en los prototipos como la paleta de colores de la interfaz, el tamaño de letra, el orden de la interfaz entre otros.

Por último, se evaluó si el panel creado cumplía con los objetivos establecidos, esto con el fin de establecer un resultado y determinar el avance del proyecto con respecto a este ciclo y dependiendo de este resultado se establecen objetivos para el próximo ciclo.

VIII DISEÑO INGENIERIL

2.1 Diseño y creación del panel de la interfaz del plugin

Para la realización del panel se requirió un lenguaje de programación, en este caso el software Autodesk Maya brinda dos opciones las cuales son Python y MEL (maya embedded language) que según el significado en la página web del software (AUTODESK, 2020) es un lenguaje de programación en el corazón de Maya. La interfaz de usuario se creó principalmente utilizando este lenguaje de programación y este proporciona una manera fácil de ampliar la funcionalidad del software, teniendo en cuenta la documentación de comandos existentes en la página web oficial se realizaron varias pruebas con los dos lenguajes de programación mencionados y teniendo en cuenta los resultados se escogió el lenguaje Python, esto con el fin de poder implementar el plugin en un futuro en otros softwares de desarrollo 3D y adicionalmente Python tiene más guías de uso, las cuales permiten una mayor cantidad de comandos para la creación del panel.

La implementación del panel radica en la fácil y rápida creación de personajes cartoon en 3D para que los usuarios puedan elaborarlos de forma sencilla escogiendo las opciones dadas para generar creaciones auténticas. También, es importante crear un diseño de panel que sea novedoso y llamativo con el objetivo de facilitar el uso del plugin.

2.1.1 Proceso de selección para el lenguaje de programación

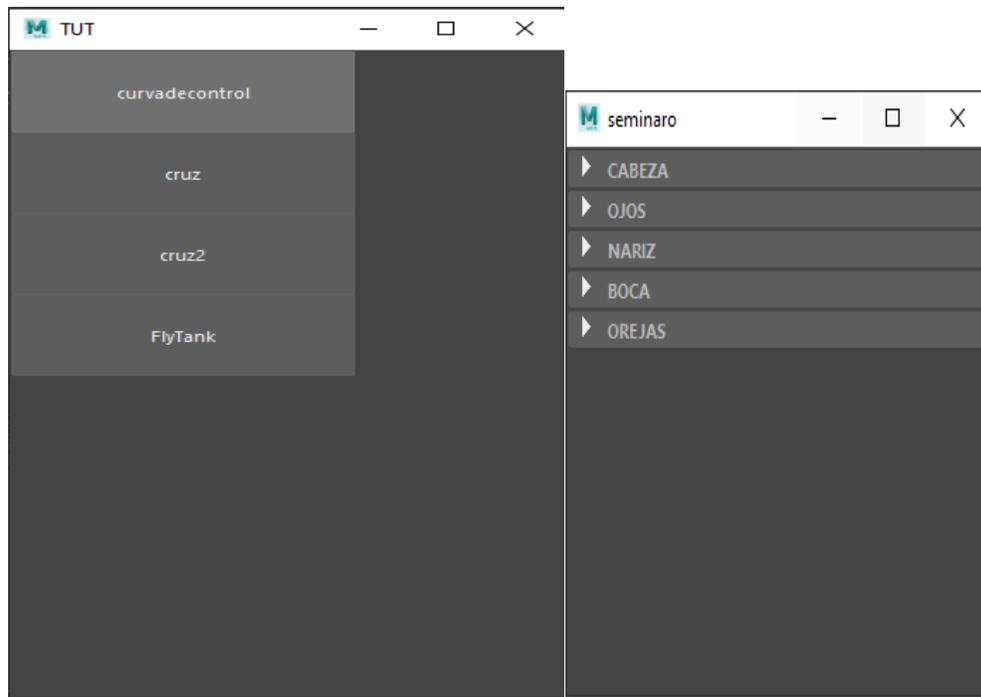
El proceso de selección para el lenguaje de programación se inició probando dos lenguajes de programación: MEL y Python; la principal diferencia entre estos dos lenguajes es que el primero es propio de la aplicación utilizada que es Autodesk Maya, mientras que el segundo es un lenguaje de programación universal.

Al realizar las pruebas con la creación de paneles y su funcionalidad, se puede ver como en **la figura 2** en el primer panel (TUT) se hace uso de MEL, el cual solo está conformado por botones y el panel no tiene ningún flujo para crear el personaje cartoon por consecuencia esto llevaría al que el usuario se perdiera en el proceso. Así mismo en la **figura 2** el segundo panel (seminario) se

utiliza el lenguaje Python el cual da más libertad de diseño debido a la documentación que trae este, dando así posibilidades de agrupar por secciones, haciendo que sea visualmente más agradable para el usuario, además de que el proceso de programación sea más fluido.

2 Figura

Panel implementado con MEL y Python



Nota: Elaboración propia

Luego de realizar las debidas pruebas de ambos lenguajes de programación, se tuvo en cuenta sus características para escoger uno de ellos, en este caso Python es más práctico ya que contiene más información que MEL y permite que sea implementado en softwares distintos a Autodesk Maya.

Para poder usar las funciones de MEL en el lenguaje Python, se implementó el comando “CMDC”, este se debe utilizar al principio del código para poder importar los comandos de un lenguaje a otro, de esta manera permite la programación en Python.

El siguiente proceso el cual es el de importación de modelos, se hace a partir de los botones de las diferentes secciones que se encuentran en el panel, estas están separadas como: cara, pelo, ojos, orejas y bocas. Este desarrollo consiste en llamar una función a partir de condicionales (if, elif) los cuales buscan la ruta donde está ubicada el objeto 3D en los archivos del computador y los importa dentro del software para poder ser utilizados en el panel; Gracias a estos condicionales se pueden hacer ajustes a las partes del rostro para que se acoplen entre sí a las diferentes formas de cabeza que se encuentran.

A. Creación de modelos 3D

Con el fin de iniciar el modelado 3D fue necesario buscar principalmente imágenes de referencia de las distintas partes del cuerpo en personajes cartoon para pasar a un proceso de selección en el cual se optó por escoger los estilos y diseños que más se acoplan al objetivo del presente proyecto como se muestra en la figura 3.

3 Figura

Referencia de cabellos

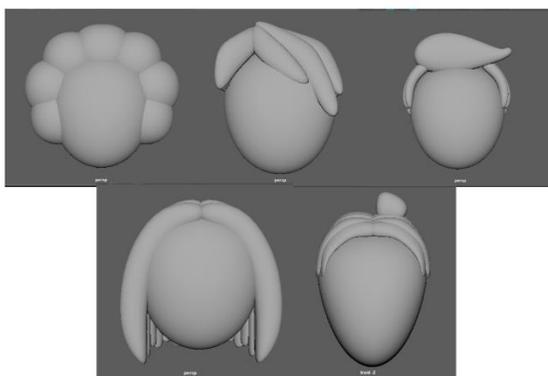


Nota: (Pinterest, 2015)

Posteriormente se inició el modelado de cada una de las referencias en el software, entre ellas: cabello, ojos, nariz, orejas, cabeza y boca; como lo muestra en la **figura 4**.

El principal motivo de la creación de cada modelo por separado está directamente relacionado con el panel para que a cada usuario se le facilite la selección de las partes para la creación de su personaje.

4 Figura *Modelos de cabellos*



Nota: Elaboración propia

2.3 Programar las funciones de vinculación, limpieza y optimización de modelos al panel de creación

Panel de creación

Primero se realizó la búsqueda de información del lenguaje de programación de MEL, el cual se encontró la documentación en la página principal de Autodesk maya junto a los distintos tipos de comandos. Posterior a esto, se creó una pestaña en la que se pueden agregar herramientas de Maya, dicha pestaña tiene el nombre de Shelf. Después se utilizó el Script Editor que permite crear código en el lenguaje Mel o en lenguaje Python y ejecutarlo en Maya. Esta característica se encuentra en la barra de menú principal: Windows > General Editor > Script Editor.

Basado en la documentación de comandos de MEL se realizaron prototipos del panel, después de escribir el código, se seleccionaba y con el botón del medio del mouse presionado se arrastraba hacia el Shelf creado para generar una nueva herramienta con el código como base. Después de realizar el panel, se crearon botones con el fin de agregarles las funciones necesarias para la importación de los modelos al software.

Por medio del bloque de código de la **figura 5** se realiza la creación del panel, con su respectivo nombre y cantidad de columnas, así mismo se introducen los botones con el fin de llamar las funciones requeridas tales como la importación de modelos, además si hay alguna otra ventana nombrada de la misma manera, la cierra y la vuelve a abrir automáticamente, como resultado final se obtiene el panel de la **figura 6**.

5 Figura

Bloque de código

```
if (`window -exists TUT`) deleteUI TUT;
window TUT;

columnLayout ;

button -w 200 -h 60 -l "Si se puede" -c "DOIT";
button -w 200 -h 60 -l "cube" -c "DOThis";
showWindow TUT;

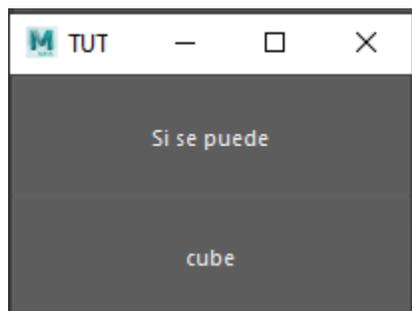
proc DOThis (){
print ("DOTHIS...");

file -import -type "OBJ" -ignoreVersion -ra true -mergeNamespacesOnClash false
  -namespace "panic" -options "mo=1;lo=0" -pr -importFrameRate
true -importTimeRange |"override" "C:/carpeta destino/modeloprueba.obj";
}
```

Nota: Elaboración propia

6 Figura

Panel importación de modelos 3D.



Nota: Elaboración propia

En estas líneas del bloque de código en la **figura 7** se realizó la creación de la ventana, pero usando el lenguaje Python, por tal motivo algunas líneas de código tienen la palabra “cmds” lo cual sirve para importar los comandos del lenguaje MEL a Python.

7 Figura

Creación de ventana

```
import maya.cmds as cmds

#INTERFACE

window_name = "proyecto"
window_title = "seminaro"
window_w = 300
window_h = 300

def create_window():
    if cmds.window(window_name, query=True, exists=True):
        cmds.deleteUI(window_name)
    cmds.window(window_name)
    cmds.window(window_name, edit=True, w=window_w, h=window_h, title= window_title)
    cmds.columnLayout("main_colum" , w=window_w, h=window_h)
    create_customUI()
    cmds.showwindow(window_name)
```

Nota: Elaboración propia

Posteriormente se implementa un botón imagen en el cual se importa una imagen desde los archivos del computador, se le asigna un tamaño y la función que realizará, ver **figura 8**, y el resultado se puede observar en la **figura 9**.

8 Figura

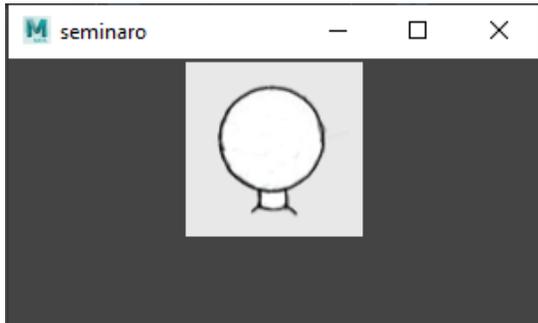
Tamaño imagen

```
cmds.symbolButton( image='C:\Imágenes y modelos MyToonCreator
\imagenes cabeza\cabeza1.png', w =100, h=100, command=crearcabeza1)
```

Nota: Elaboración propia

9 Figura

Botón imagen de la interfaz



Nota: Elaboración propia

En el siguiente bloque se implementa la función para importar el modelo de la cabeza número 1, para esto se usa el comando (cmds.file) con la dirección de la ruta donde se encuentra dicho modelo como se puede ver en la **Figura 10**.

10 Figura

Importar modelo

```
# CABEZA 1 ////////////////////////////////////////
def crearcabeza1(test):
    cmds.file('C:\Imágenes y modelos MyToonCreator\
|Modelo Cabeza\cabeza 1.obj', i=True)
```

Nota: Elaboración propia

Después se crean unos layouts con el fin de organizar el panel y distribuir de mejor manera las características que el usuario podrá usar para modificar en su personaje, para esto se implementó el código que se observa en la **figura 11**.

11 Figura

Creación desplegables

```
#BOTON CABEZA////////////////////////////////////
cmds.frameLayout("CABEZA", label="ELIGE UNA CABEZA ", w>window_w,
parent="mainShelfTab",bgc=(0.1490,0.1490,0.1490))
cmds.rowColumnLayout( numberOfColumns=3,
|columnWidth=[(1, 100), (2, 100), (3, 100)] )
```

Nota: Elaboración propia

Posteriormente como resultado de la figura anterior se puede apreciar que tanto los layouts para la interfaz relacionados con el elemento a modificar y los layouts de la imagen se presentan en las **figuras 12 & 13**, con un respectivo número de filas y columnas para mejorar la distribución de las imágenes que se usarán en el panel, en estas se encontrarán las opciones de cabeza y ojos que al desplegarlas se tendrán dos alternativas por elemento.

12 Figura

Layouts desplegables en la interfaz



Nota: elaboración propia

13 Figura

Layouts desplegables en la interfaz con imagen



Nota: elaboración propia

Vinculación e importación de los modelos 3D con el panel de creación.

Después, para poder cambiar las cabezas, a cada uno de los modelos se le asignó un grupo que se le denominó “CABEZA” tal y como se muestra en la **figura 14**. Esto con el fin de que al momento de presionar el botón el cual tiene asignado esa parte del cuerpo en específico, pueda borrar el elemento y lo asigna a la nueva cabeza con su respectivo grupo. También se les cambió el nombre a los modelos ya que al importarlos tenía el nombre predeterminado Mesh y no dejaba continuar con el proceso.

14 Figura

Se asigna el grupo cabeza al modelo

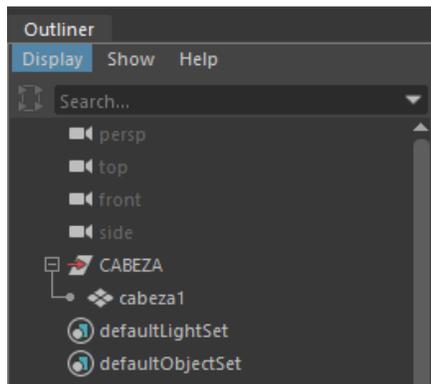
```
def crearcabeza1(test):
    if cmds.objExists('CABEZA'):
        cmds.select('CABEZA')
        cmds.delete('CABEZA')
        cmds.file('C:\carpeta destino\cabeza 1.obj', i=True)
        cmds.rename ('Mesh', 'cabeza')
        cmds.group( 'cabeza', n='CABEZA')
    else:|
        cmds.file('C:\carpeta destino\cabeza 1.obj', i=True)
        cmds.rename ('Mesh', 'cabeza')
        cmds.group( 'cabeza', n='CABEZA')
```

Nota: Elaboración propia

El resultado final del proceso realizado en la programación se puede observar en la **Figura 15**.

15 Figura

Creación de grupo



Nota: Elaboración propia

Para la realización de los ojos, primero el programa pregunta si existe una respectiva cabeza, esto con el fin de acomodar los ojos de manera correcta al modelo que escogió el usuario, después se le cambia el nombre como por ejemplo de cabeza 1 a A1 como se evidencia en la **figura 16**,

esto con el fin de mantener un orden, con motivo de que no se repitan los nombres ni genere un error inesperado; posteriormente se realiza el mismo procedimiento para importarlos.

16 Figura

Combinar el modelo de la cabeza con el modelo de los ojos

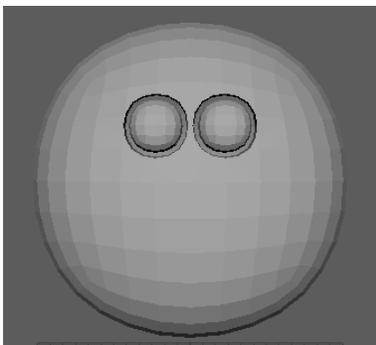
```
if cmds.objExists('cabeza1'):
    print("EXISTE CABEZA 1 Y SE PONEN LOS OJOS1")
    cmds.rename ('cabeza1', 'A1')
    cmds.file('C:\carpeta destino\ojosprueba.obj', i=True)
    cmds.rename ('Mesh', 'parpado')
    cmds.file('C:\carpeta destino\pupilaprueba.obj', i=True)
    cmds.rename ('Mesh', 'B1')
    cmds.polyCBoolOp( 'A1', 'parpado', op=2, n='A1B1' )
    cmds.group( 'A1B1', 'B1', n='OJOS')
```

Nota: Elaboración propia

Como los ojos tienen 2 modelos internos, uno que son los huecos que se realizan a la cabeza y estos se nombraron Párpados y el otro modelo que son las pupilas se les denominó como B1 y esto se le asigna a un grupo, en este caso llamado OJOS, y al final gracias a la función (polyCBoolOp) se realizó una booleana de diferencia entre los ojos y la cabeza anteriormente creada. En el grupo OJOS están las pupilas(B1) y el resultado de la diferencia entre la cabeza y los ojos como se muestra en la **figura 17**.

17 Figura

Resultado cabeza



Nota: Elaboración propia

Siguiendo con la realización de los botones, en las narices se hizo el mismo procedimiento que con los ojos primero se crea la función preguntando si existe el grupo NARIZ y si existe bórrelo, como se puede observar en la **figura 18**, después se pregunta si existe cabeza 1 o A1, cabeza 1 es cuando solo existe la cabeza y A1 es cuando ya hubo alguna combinación con la cabeza 1. Después dependiendo la cabeza se exporta la nariz seleccionada se acomoda a la cabeza y se le asigna el grupo NARIZ, de esta forma se obtiene como resultado en el outliner la **figura 19**.

18 Figura

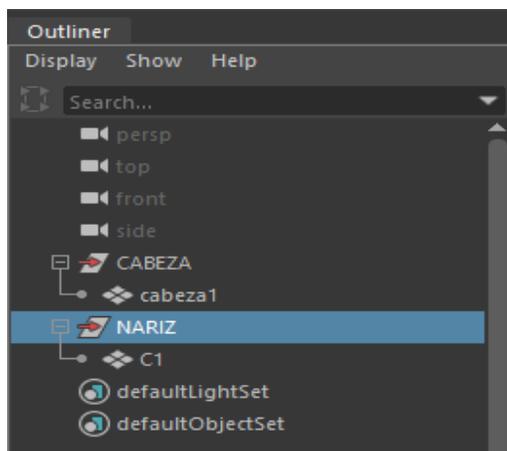
Creación de nariz

```
def crearnariz1(test):
    if cmds.objExists('CABEZA' and 'NARIZ'):
        print("CAMBIAR A NARIZ 1")
        cmds.delete('NARIZ')
    if cmds.objExists('A1') or cmds.objExists('cabeza1'):
        print("ARREGLAR NARIZ PARA CABEZA 1")
        cmds.file('C:\Imagenes y modelos MyToonCreator\Modelos Narizes\Nariz 1.obj', i=True)
        cmds.rename('Mesh', 'C1')
        cmds.group('C1', n='NARIZ')
        cmds.move(0, -0.807, -0.49, 'NARIZ', absolute=True)
        cmds FreezeTransformations('NARIZ')
```

Nota: Elaboración propia

19 Figura

Creación del grupo nariz



Nota: Elaboración propia

Para la creación de las bocas se realizó un procedimiento similar al de los ojos, se preguntaba qué cabeza existía y en base al modelo se exportaba la boca asignada, para después realizar el proceso de combinación, esto se ve reflejado en la **figura 20**. Al igual que los ojos, está dividida en dos partes, la caja, la cual sería el hueco de la boca y las muelas que hace referencia a los dientes, al final el resultado de la unión se nombró como A1D1 haciendo referencia a A1 como la cabeza 1 y D1 como la boca número 1, dicha combinación se puede notar en el outliner como se muestra en la **figura 21**.

20 Figura

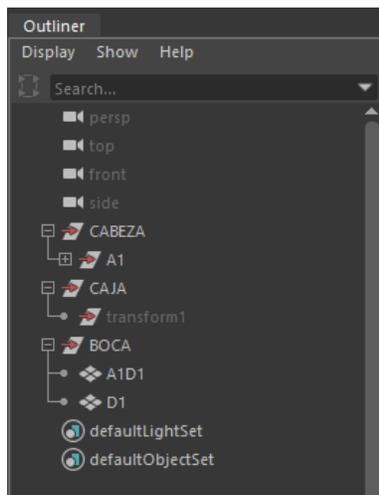
Creación de boca

```
def crearboca1(test):
    if cmds.objExists('cabeza1'):
        print("EXISTE CABEZA 1 Y SE PONEN LOS BOCA 1")
        cmds.rename ('cabeza1', 'A1')
        cmds.file('C:\carpeta destino\CajaBoca1.obj', i=True)
        cmds.rename ('Mesh', 'CAJA')
        cmds.file('C:\carpeta destino\Dientes y lengua 1.obj', i=True)
        cmds.rename ('Mesh', 'D1')
        cmds.polyCBoolOp( 'A1', 'CAJA', op=2, n='A1D1' )
        cmds.group( 'A1D1', 'D1', n='BOCA')
```

Nota: Elaboración propia.

21 Figura

Creación del grupo boca



Nota: Elaboración propia

Posteriormente para poder dar más libertad de diseño, se agregaron sliders para poder controlar la traslación y tamaño de los ojos, de esta forma el usuario lo puede ajustar a su propio gusto o necesidad.

Para la creación de las orejas se hizo también un procedimiento similar a los mencionados anteriormente, se creó una función, la cual preguntaba si ya existía un grupo llamado OREJAS, si dicho grupo existe, se borraba y luego se agregaban las orejas seleccionadas por el usuario, se le cambiaba el nombre a E1 y se le asignaba al grupo OREJAS. Ver **figura 22 & 23**.

22 Figura

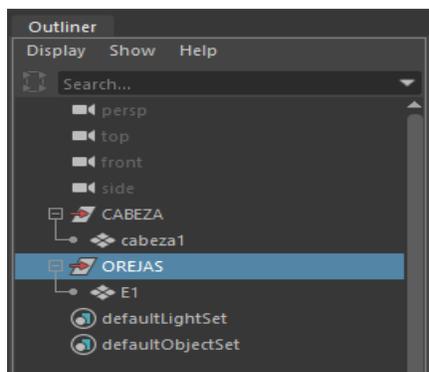
Creación del grupo boca

```
def crearorejas1(test):
    if cmds.objExists('CABEZA' and 'OREJAS'):
        print("CAMBIAR A OREJAS 1")
        cmds.delete('OREJAS')
    if cmds.objExists('cabeza1'):
        print("ARREGLAR OREJAS PARA CABEZA 1")
        cmds.file('C:\carpetas destino\orejas_duende.obj', i=True)
        cmds.rename('Mesh', 'E1')
        cmds.group('E1', n='OREJAS')
```

Nota: Elaboración propia

23 Figura

Creación del grupo orejas



Nota: Elaboración propia

Para la creación de los peinados, se realizó el mismo procedimiento que las orejas, con la diferencia que el grupo se llamaría PEINADO y los peinados tendrían las asignaciones de F1, F2, F3 y así respectivamente. Ver **figura 24 & 25**.

24 Figura

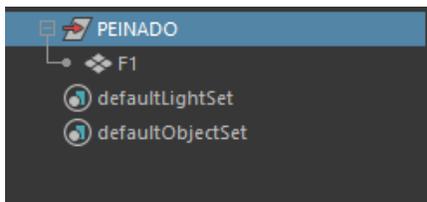
Creación de peinados

```
def crearpeinado1(test):
    if cmds.objExists('CABEZA' and 'PEINADO'):
        print("CAMBIAR A PEINADO 1")
        cmds.delete('PEINADO')
    if cmds.objExists('cabeza1'):
        print("ARREGLAR PEINADO PARA CABEZA 1")
        cmds.file('C:\carpeta destino\peinado1.obj', i=True)
        cmds.rename ('Mesh', 'F1')
        cmds.group( 'F1', n='PEINADO')
```

Nota: Elaboración propia

25 Figura

Creación del grupo peinados



Nota: Elaboración propia

Después a cada botón de la cabeza se le creó un código el cual preguntaba la existencia de los ojos, luego se seleccionaba ciertos grupos, con el fin de borrarlos y luego llamar las funciones de creación correspondientes, esto con el propósito de mantener el tipo de ojos, pero cambiando el tipo de cabeza. Se realizó un proceso similar con las narices, bocas, orejas y peinados como se aprecia en la **figura 26**.

26 Figura

Creación del grupo peinados

```

if cmds.objExists('B1'):
print(" existen los ojos1")
cmds.select('CABEZA','OJOS','parpado')
cmds.delete('CABEZA','OJOS','parpado')
crearcabeza1(test)
crearojos1(test)

if cmds.objExists('C1'):
print("ARREGLAR NARIZ PARA CABEZA 1")
cmds.select('NARIZ')
cmds.delete('NARIZ')
crearnariz1(test)

if cmds.objExists('D1'):
print("ARREGLAR Boca PARA CABEZA 2")
cmds.select('CABEZA','CAJA','BOCA')
cmds.delete('CABEZA','CAJA','BOCA')
crearcabeza1(test)
crearbocal1(test)

if cmds.objExists('E1'):
print("ARREGLAR OREJAS PARA CABEZA 1")
cmds.select('OREJAS')
cmds.delete('OREJAS')
crearorejas1(test)

if cmds.objExists('F1'):
print("ARREGLAR NARIZ PARA CABEZA 2")
cmds.select('PEINADO')
cmds.delete('PEINADO')
crearpeinado1(test)

```

```

if cmds.objExists('B1'):
print(" existen los ojos1")
cmds.select('CABEZA','OJOS','parpado')
cmds.delete('CABEZA','OJOS','parpado')
crearcabeza1(test)
crearojos1(test)

if cmds.objExists('C1'):
print("ARREGLAR NARIZ PARA CABEZA 1")
cmds.select('NARIZ')
cmds.delete('NARIZ')
crearnariz1(test)

if cmds.objExists('D1'):
print("ARREGLAR Boca PARA CABEZA 2")
cmds.select('CABEZA','CAJA','BOCA')
cmds.delete('CABEZA','CAJA','BOCA')
crearcabeza1(test)
crearbocal1(test)

if cmds.objExists('E1'):
print("ARREGLAR OREJAS PARA CABEZA 1")
cmds.select('OREJAS')
cmds.delete('OREJAS')
crearorejas1(test)

if cmds.objExists('F1'):
print("ARREGLAR NARIZ PARA CABEZA 2")
cmds.select('PEINADO')
cmds.delete('PEINADO')
crearpeinado1(test)

```

Nota: Elaboración propia

Para cada uno de los botones de los ojos, se implementaron unas líneas de código las cuales permiten que al tener una cabeza y elegir otro tipo de ojos, se mantenga la cabeza pero se cambien los ojos, esto se evidencia en la **figura 27**, también se realizó una función que preguntaba si existe una cabeza que ya tenía boca, se pondrían los ojos seleccionados y a esa combinación se le asignaba el nombre A1BD haciendo referencia que se creó la cabeza 1, con su respectivos ojos y bocas como se muestra en la **figura 28**, por último se implementó otra función la cual preguntaba si ya existe una cabeza con sus respectivos ojos y bocas, así asigna los ojos que requería el usuario pero

manteniendo la respectiva cabeza y boca seleccionadas anteriormente como se percibe en la **figura 29**. Dichas funciones mencionadas anteriormente se realizaron para cada cabeza con la boca asignada.

27 Figura

Creación de código botón ojos 1

```
def crearojos1(test):
    if cmds.objExists('A1') and cmds.objExists('OJOS'):
        print(" existe cabeza1 y OJOS")
        cmds.delete('OJOS','CABEZA','parpado')
        crearcabeza1(test)
        crearojos1(test)
```

Nota: Elaboración propia

28 Figura

Creación de código botón ojos 2

```
if cmds.objExists('A1D'):
    print(" existe cabeza1 y BOCA")
    cmds.file('C:\carpeta destino\ojosprueba.obj', i=True)
    cmds.rename ('Mesh', 'parpado')
    cmds.file('C:\carpeta destino\pupilaprueba.obj', i=True)
    cmds.rename ('Mesh', 'B1')
    cmds.polyCBoolOp( 'A1D', 'parpado', op=2, n='A1BD' )
    cmds.group( 'A1BD', 'B1', n='OJOS')
```

Nota: Elaboración propia

29 Figura

Creación de código botón ojos 3

```
if cmds.objExists('A1BD') and cmds.objExists('D1'):
    print(" existe cabeza1,BOCA Y OJOS")
    cmds.delete('OJOS','CABEZA','parpado','CAJA','BOCA')
    crearcabeza1(test)
    crearojos1(test)
    crearboca1(test)
```

Nota: Elaboración propia

Para cada uno de los botones de las bocas, se le creó un código el cual preguntaba la existencia de un modelo con un tipo de cabeza, ojos y boca, se borran los respectivos grupos y se llaman las funciones correspondientes, este proceso se realizó para todos los tipos de cabeza, pero con los distintos tipos de ojos, como se evidencia en el código de la **figura 30**. También se realizó una función que preguntaba la existencia de un modelo con un respectivo tipo de cabeza y boca, ver **figura 31**.

30 Figura

Creación de código botón Boca 1

```
def crearboca1(test):
    if cmds.objExists('A1BD') and cmds.objExists('B1'):
        print(" existe cabeza1,BOCA Y OJOS")
        cmds.select('OJOS','CABEZA','parpado','CAJA','BOCA')
        cmds.delete('OJOS','CABEZA','parpado','CAJA','BOCA')
        crearcabeza1(test)
        crearojos1(test)

    if cmds.objExists('A1BD') and cmds.objExists('B2'):
        print(" existe cabeza1,BOCA Y OJOS")
        cmds.select('OJOS','CABEZA','parpado','CAJA','BOCA')
        cmds.delete('OJOS','CABEZA','parpado','CAJA','BOCA')
        crearcabeza1(test)
        crearojos2(test)

    if cmds.objExists('A1BD') and cmds.objExists('B3'):
        print(" existe cabeza1,BOCA Y OJOS")
        cmds.select('OJOS','CABEZA','parpado','CAJA','BOCA')
        cmds.delete('OJOS','CABEZA','parpado','CAJA','BOCA')
        crearcabeza1(test)
        crearojos3(test)
```

Nota: Elaboración propia

31 Figura

Creación de código botón Boca 2

```
if cmds.objExists('A1') and cmds.objExists('BOCA') :
    print(" existe CABEZA 1 y BOCA")
    cmds.delete('CAJA','CABEZA','BOCA')
    crearcabeza1(test)
    crearboca1(test)

if cmds.objExists('A2') and cmds.objExists('BOCA') :
    print(" existe CABEZA 2 y BOCA")
    cmds.delete('CAJA','CABEZA','BOCA')
    crearcabeza2(test)
    crearboca1(test)

if cmds.objExists('A3') and cmds.objExists('BOCA') :
    print(" existe CABEZA 3 y BOCA")
    cmds.delete('CAJA','CABEZA','BOCA')
    crearcabeza3(test)
    crearboca1(test)
```

Nota: Elaboración propia

2.3.1 Limpieza y optimización de los modelos

Con el código de la **figura 32** se creó un cuadro de texto para que el usuario pueda asignarle un nombre al modelo y luego de que oprima la tecla enter, llame una función, la cual limpia el modelo y lo deja con el nombre asignado.

32 Figura

Creación del cuadro de texto en el panel

```
cmds.frameLayout("NOMBRE", label="DALE UN NOMBRE A TU PERSONAJE ",
w>window_w, parent="mainShelfTab",bgc=(0.1490,0.1490,0.1490))
cmds.columnLayout( columnAttach=('both', 20), rowSpacing=10, columnWidth=420)
cmds.text( label='NOMBRE',fn="boldLabelFont" )
name = cmds.textField()
cmds.textField( name, edit=True, cc=limpiar )
cmds.text( label='Cuando su personaje este listo ,escriba
el nombre que quiere ponerle al personaje y presione la tecla ENTER ',
|fn="boldLabelFont", ww=True)
```

Nota: Elaboración propia

A continuación, se creó una función para realizar la limpieza del modelo y se utilizaron las funciones de polyRemesh, polyRetopo y polyUnite, así el modelo queda como un solo objeto y con el nombre asignado. Esto se logró usando el código de la **figura 33**.

33 Figura

Creación de la función de limpieza

```
def limpiar(name):
    if cmds.objExists('A1BD'):
        if cmds.objExists('B0') and cmds.objExists('D0'):
            print("limpia resultado")
            cmds.select('A1BD')
            cmds.DeleteHistory( 'A1BD' )
            cmds.polyUnite( 'OJOS', 'BOCA', 'NARIZ', 'OREJAS', 'PEINADO', n='result' )
            cmds.DeleteHistory( 'result' )
            cmds.rename( 'result', '' + name)
        else:
            print("limpia resultado")
            cmds.select('A1BD')
            cmds.DeleteHistory( 'A1BD' )
            cmds.polyRemesh( me1=1,cot=20,smt=20 ,nds=0)
            cmds.select('A1BD')
            cmds.DeleteHistory( 'A1BD' )
            cmds.select('A1BD')
            cmds.polyRetopo( 'A1BD' )
            cmds.polyUnite( 'OJOS', 'BOCA', 'NARIZ', 'OREJAS', 'PEINADO', n='result' )
            cmds.DeleteHistory( 'result' )
            cmds.rename( 'result', '' + name)
```

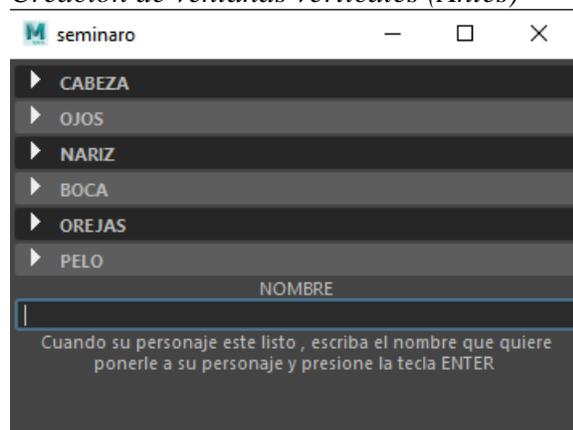
Nota: Elaboración propia

2.3.2 Actualización del panel

Haciendo la respectiva validación del diseño del panel de este primer esquema se encontró que se podría reestructurar su diseño para facilitarle al usuario la interacción con las pestañas y sliders de creación de las distintas partes del personaje cartoon. Esta actualización consistió en cambiar las opciones de ventanas verticales como se ve en la **figura 34** a ventanas independientes horizontales para una mejor visualización de las distintas opciones con las que se cuentan. También se implementó un logo y una introducción para darle una cálida bienvenida al usuario al momento de abrir el plugin que se visualiza en la **figura 35**.

34 Figura

Creación de ventanas verticales (Antes)



Nota: Elaboración propia

35 Figura

Creación de ventanas horizontales (Después)



Nota: Elaboración propia

De esta manera se concluye todo el proceso realizado a nivel visual y de funcionamiento, es decir, el código implementado para poder crear el panel, funcionalidad de botones para importar, unir, modificar en escala, rotación y posición los modelos, y se continúa con el análisis de resultados obtenidos.

IX RESULTADOS

Con el fin de evaluar y obtener una retroalimentación del trabajo realizado con el plugin creado, se llevó a cabo un taller la cual estaba dirigido a estudiantes de ingeniería Multimedia, de los cuales se logró un total de 35 participantes entre los semestres cuarto y décimo de la universidad de San Buenaventura sede Bogotá.

Para poder llevar a cabo este taller se le brindó información a cada estudiante mediante una página web la cual contenía la opción de descargar el plugin de forma manual o por medio de un instalador, el manual de usuario, una introducción del proyecto y su respectiva encuesta para evaluar el plugin. (Visitar la página web de MyToonCreator <https://juanfelipequintero53.wixsite.com/website> para mayor descripción).

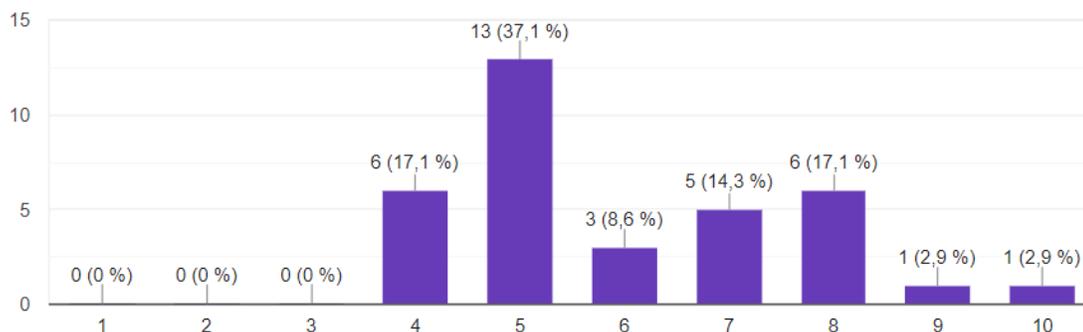
A continuación, se presentan los resultados obtenidos con sus respectivos análisis. Inicialmente se quiso conocer a qué semestre pertenecían, lo cual se puede ver reflejado en la **Figura 36**, que la mayoría de las personas encuestadas pertenecen a quinto semestre con un porcentaje del 37.1% (13 personas) y la minoría con un porcentaje cada uno del 2.9% (1 persona), se encuentra entre noveno y décimo semestre.

36 Figura

Gráfica de encuesta “Generador de personajes 3D” - Pregunta 1

1. ¿A que semestre pertenece?

35 respuestas



A pesar de que una gran cantidad de estudiantes pertenecen a semestres 4 & 5, este resultado se generó porque se tuvo como propósito evaluar el plugin desde varias experiencias de usuario con el software Autodesk Maya. De esta forma comprobar si podían conocer la interfaz en su totalidad o parcialmente tenía alguna interferencia en el uso del complemento, a lo cual se da respuesta con los siguientes resultados de las demás preguntas de la encuesta.

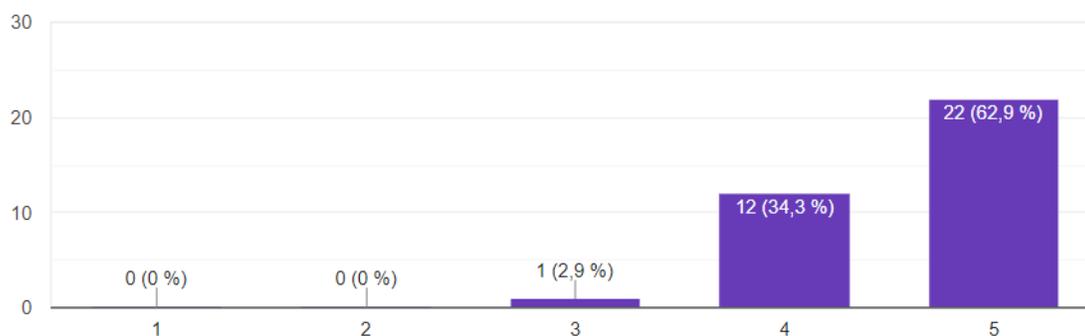
En la **Figura 37** se da a conocer el resultado del proceso de instalación del plugin, allí se puede evidenciar que el 62.9% (22 personas) de los usuarios consideró muy fácil la instalación del plugin, el 34.3 % (12 personas) lo consideró fácil y el 2.9% (1 persona) lo consideró moderado.

37 Figura

Gráfica de encuesta “Generador de personajes 3D” - Pregunta 2

2. Teniendo en cuenta su experiencia. ¿Qué tan fácil le pareció el proceso de instalación del plugin?

35 respuestas



Gracias a que se les proporcionó a los usuarios un manual en el que se les explicaba detalladamente paso a paso y con ayudas visuales, la instalación del plugin ya sea de manera manual o por medio del instalado. De esta manera se obtuvo un resultado muy positivo ya que se refleja que la mayoría de encuestados encontraron fácil este proceso de instalación.

Para la búsqueda del plugin ya instalado dentro del software de Autodesk maya, se puede observar en la **Figura 38**, que el 77.1% (27 personas), 17.1% (6 personas) y 5.7% (2 personas) de

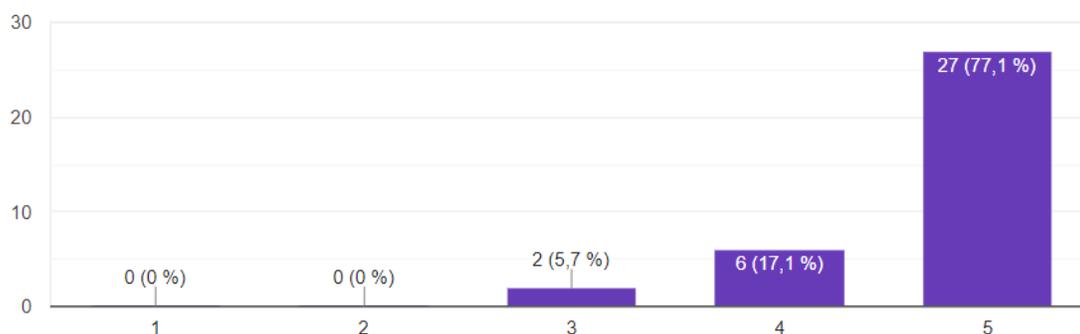
los usuarios encontró el plugin de manera muy rápida, rápida y en tiempo moderado respectivamente.

38 Figura

Gráfica de encuesta “Generador de personajes 3D” – Pregunta 3

3. ¿Qué tan rápido encontró el plugin dentro del software Autodesk Maya?

35 respuestas



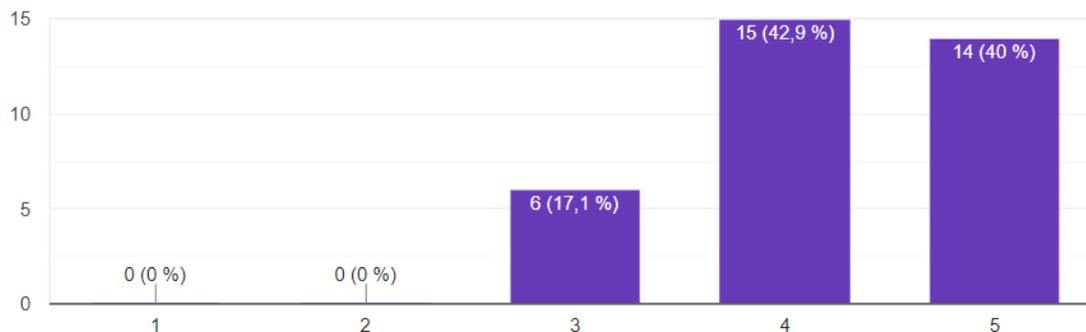
El resultado se logró gracias a que en el manual de usuario con ayudas visuales se le especificaba cómo iba a estar nombrado el plugin y donde iba a estar ubicado dentro del software Autodesk Maya, dándole mayor facilidad para ingresar al plugin e interactuar con él.

Con el fin evaluar la apariencia visual del plugin, en la **Figura 39** se analizó que al 40% (14 personas) de los usuarios les pareció muy agradable, el 42.9% (15 personas) le pareció agradable y el 17.1% (6 personas) lo encontró de apariencia moderada.

39 Figura*Gráfica de encuesta “Generador de personajes 3D” - Pregunta 4*

4. Al ver la interfaz gráfica del plugin, esta le pareció visualmente

35 respuestas



Se obtuvo este resultado debido a la forma en la que el plugin estaba estructurado, tenía un logo el cual lo identificaba, estaba organizado por secciones de creación, es decir una pestaña por cada parte del personaje, y se diferencian por colores, lo cual facilitaba el entendimiento y el orden de creación al usuario al momento de usar el plugin.

Las distintas opciones del panel con su funcionalidad fueron evaluadas en la **Figura 40**, se evidenció que el 82.9% (29 personas) de los usuarios les pareció muy fácil entenderlas e interactuar con ellas y el 17.1% (6 personas) le pareció fácil.

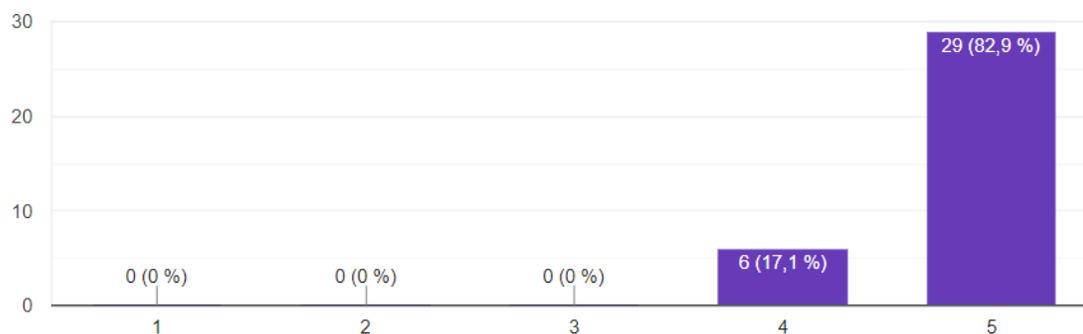
40 Figura

Gráfica de encuesta “Generador de personajes 3D” - Pregunta 5

5. Al desplegar el menú de opciones del panel de creación, ¿Qué tan fácil fue para usted entender sus funciones?



35 respuestas



El plugin está organizado por secciones y mediante títulos facilitando su comprensión, así mismo hay textos los cuales explican al usuario qué acciones pueden realizar para crear y modificar su personaje 3D.

En la interacción con el panel de creación del plugin, se muestra en la **Figura 41** que el 77,1% (27 personas) de los usuarios les pareció muy fácil y el 17,1% (6 personas) le pareció fácil la interacción con los distintos componentes del panel.

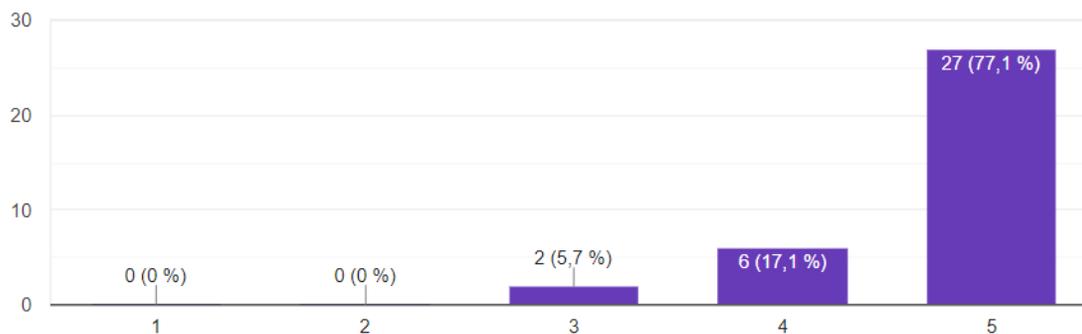
41 Figura

Gráfica de encuesta “Generador de personajes 3D” - Pregunta 6

6. ¿Considera usted que la interacción con los componentes que tiene el panel de creación fue fácil?



35 respuestas



Se tuvo un resultado positivo debido a que los botones de creación dentro del panel contaban con imágenes, de esta forma se logró dar al usuario una idea previa del estilo de los elementos que componen el personaje y mediante los sliders se le facilitó el proceso de modificar las partes, ya sea en escala o posición.

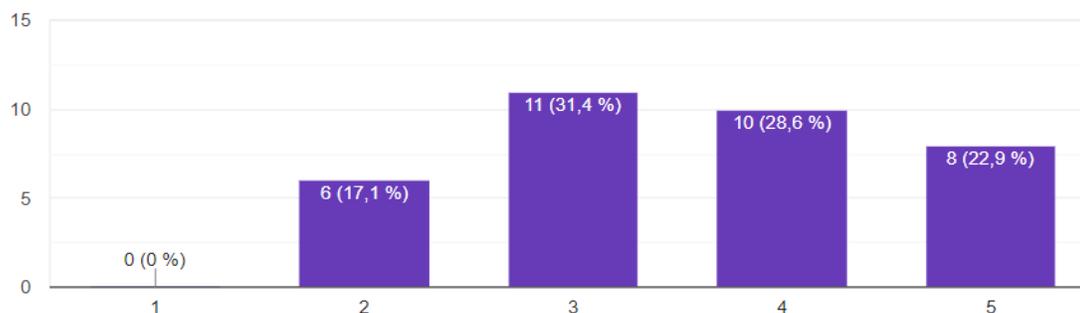
Para evaluar la diversidad en las opciones de creación de personajes, la **Figura 42** revela que el 31,4% (11 personas) de los usuarios considero moderada la diversidad, el 28,6% (10 personas) le pareció diversa, el 22,9 (8 personas) la considero muy diversa y el 17,1% (6 personas) la considere con poca diversidad.

42 Figura*Gráfica de encuesta “Generador de personajes 3D” - Pregunta 7*

7. ¿Siente que tiene hay diversidad en las opciones para la creación de los personajes?



35 respuestas



La mayoría de las personas no sintió una amplia diversidad entre las opciones de creación ya que se contaba con una limitación de solo 4 opciones en las diferentes secciones del panel. Esto se debe a que se enfocó más en el correcto funcionamiento del plugin que en las opciones de modelado.

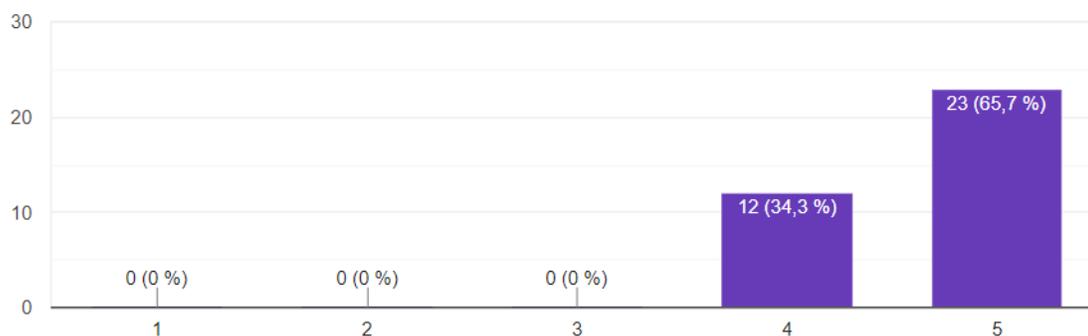
Enfocado en poder entender la facilidad con la que se creó el personaje por medio del plugin se da a conocer el resultado de la pregunta reflejada en la **Figura 43**, en esta se observó que el 65,7% (23 personas) de los usuarios les pareció muy fácil y el 34,3% (12 personas) le pareció fácil.

43 Figura

Gráfica de encuesta “Generador de personajes 3D” - Pregunta 8

8. Después de haber hecho uso del plugin en Mutodesk Maya, ¿Qué tan fácil considera que es la creación de personajes?

35 respuestas



Este resultado refleja que el plugin contaba con distintas funciones las cuales facilitaban el proceso de creación, además el usuario contaba con diferentes secciones del panel para escoger la que más se acomodaba a su gusto y el plugin combinaba y limpiaba el modelo para que estuviera óptimo para usarse y sin error alguno.

Para obtener retroalimentación a las oportunidades de mejora que vio el usuario frente al plugin, se formuló una pregunta abierta, para que se pudieran expresar de manera subjetiva, de esta forma como lo muestra los comentarios más destacados fueron:

- “Agregar más opciones de transformación de personaje o ambiente”.
- “Más objetos de creación, sin embargo, todo está muy bien”.
- “Más opciones, en cuanto a Boca o nariz sería una mejora”.
- “Más opciones para el cuerpo y arreglar una de las narices”.
- “Colocar opciones de cuerpo”
- “Añadir más opciones de personalización y dar la opción de crear algunas propias para el usuario”.

Teniendo en cuenta algunas de las respuestas, se evidenció que la mayoría de los usuarios ven como oportunidad de mejora la variedad de personajes y la personificación de los mismos, esto

se dio a causa del limitado número de modelos que tenía el plugin, el cual solo se enfocó en la parte de la cabeza.

Adicionalmente se pidió de forma abierta también que evaluaran si tuvieron algún inconveniente con el uso del plugin, se evidencio que el 51.5% (18 personas) encontró inconvenientes al momento de usar el plugin y el 48.5% (17 personas) no encontró inconvenientes.

Debido a la falta de advertencias dentro del manual, haciendo referencia al uso del plugin, los usuarios realizaban acciones indebidas en el proceso de crear el personaje tales como usar el comando del teclado Ctrl+Z, querer escalar o rotar un elemento manualmente, moverlo de su posición inicial, esto tuvo como consecuencia un mal funcionamiento del plugin o incluso generaba fallas del mismo programa Autodesk Maya, a continuación, se pondrán las respuestas destacadas para este punto:

- “Se me cerró Maya 2 veces, al ponerme a ver las ediciones”.
- “Al momento de escalar las pupilas El Maya Tira error y se cierra El programa”.
- “Crasheo una vez al hacer el personaje”.
- “Cuando aumentaba el tamaño de las pupilas se me crasheo, luego lo intenté de nuevo y ya no pasó, entonces no sabría por qué pasó”.
- “Al darle ctrl+z para quitar un objeto y crear uno nuevo, quedaron dos”.

De esta manera como ya se mencionó anteriormente, se debe mejorar el manual de usuario, teniendo en cuenta las acciones que no se pueden realizar dentro del programa.

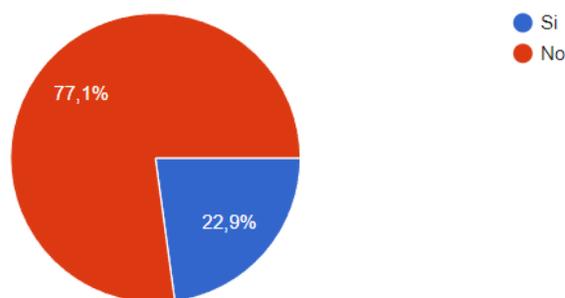
Para finalizar la encuesta se preguntó si se tuvo alguna limitación al crear el personaje 3D y cuáles fueron estas, de acuerdo con la **Figura 44** que representa una gráfica, se evidenció que el 77,1% (27 personas) de los usuarios no encontraron ninguna limitación y el 22,9 % (8 personas) encontró alguna limitación. Estas están enfocadas en la variedad de personajes y en la personificación de estos.

44 Figura

Gráfica de encuesta “Generador de personajes 3D” - Pregunta 11

11. ¿Sintió alguna limitación al momento de crear su personaje?

35 respuestas



Esto se dio, gracias a que en el manual se le explicó al usuario detalladamente el debido procedimiento a la hora, tanto de la instalación del plugin como en la creación de los personajes. Para poder validar la respuesta negativa por parte de los estudiantes, se optó por poner una casilla donde justificaran por qué sintieron alguna limitación, a lo cual se obtuvo como respuestas:

- “En cuanto a algunas opciones, me parecieron muy pocas”.
- “Las modificaciones de la boca”.
- “Un poco en cuanto a la personalización, pero la herramienta está muy bien desarrollada”.
- “muy pocas herramientas”.
- “serian mejores más expresiones”.
- “La cantidad de estilos disponibles”.
- “Quería más opciones”.

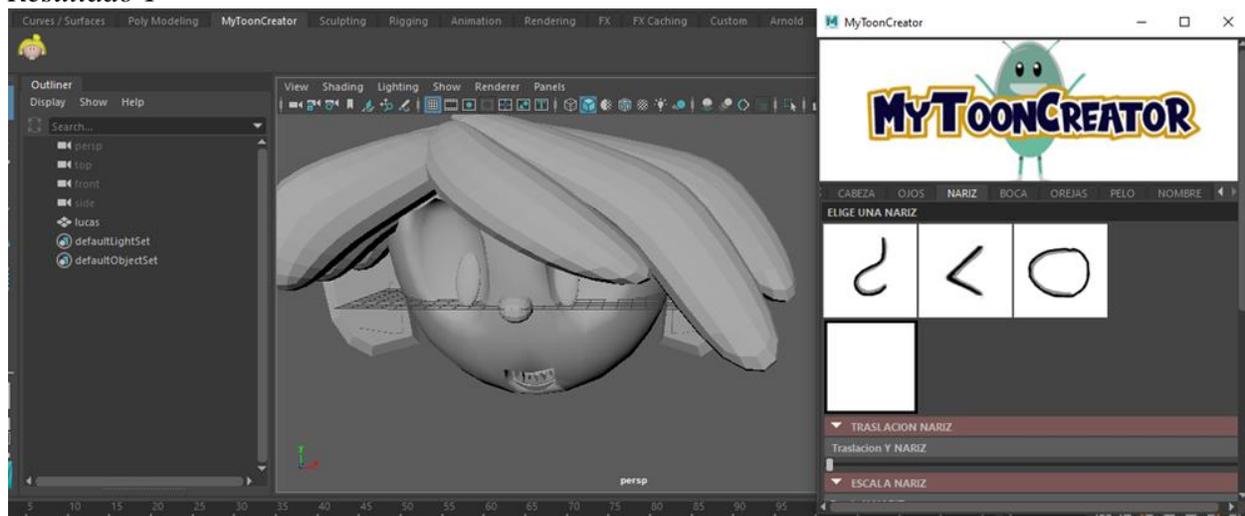
Estos limitantes se justifican, como se ha dicho durante el desarrollo de este documento, debido a la poca cantidad de opciones al crear su personaje y personalizarlo a su gusto.

A pesar de las sugerencias y puntos negativos de la herramienta, se pudo observar que en la encuesta se dieron respuestas positivas por parte de toda la audiencia, se quiso evaluar a la población de distintos semestres, teniendo en cuenta que a partir de cuarto semestre se comienza a hacer uso del software, esto con el fin de tener la opinión de los estudiantes que comienzan a conocer el programa y de los más experimentados en el uso de este.

Desde el primer paso que fue la descarga e instalación del plugin hasta la creación del personaje por medio de este se observó que los resultados que los encuestados estuvieron cómodos con el uso de este y aunque visualmente es sencillo, sus funciones facilitan procesos de modelado y optimización de las figuras geométricas que componen los personajes.

Resultados obtenidos con el plugin en la creación de los personajes Cartoon 3D

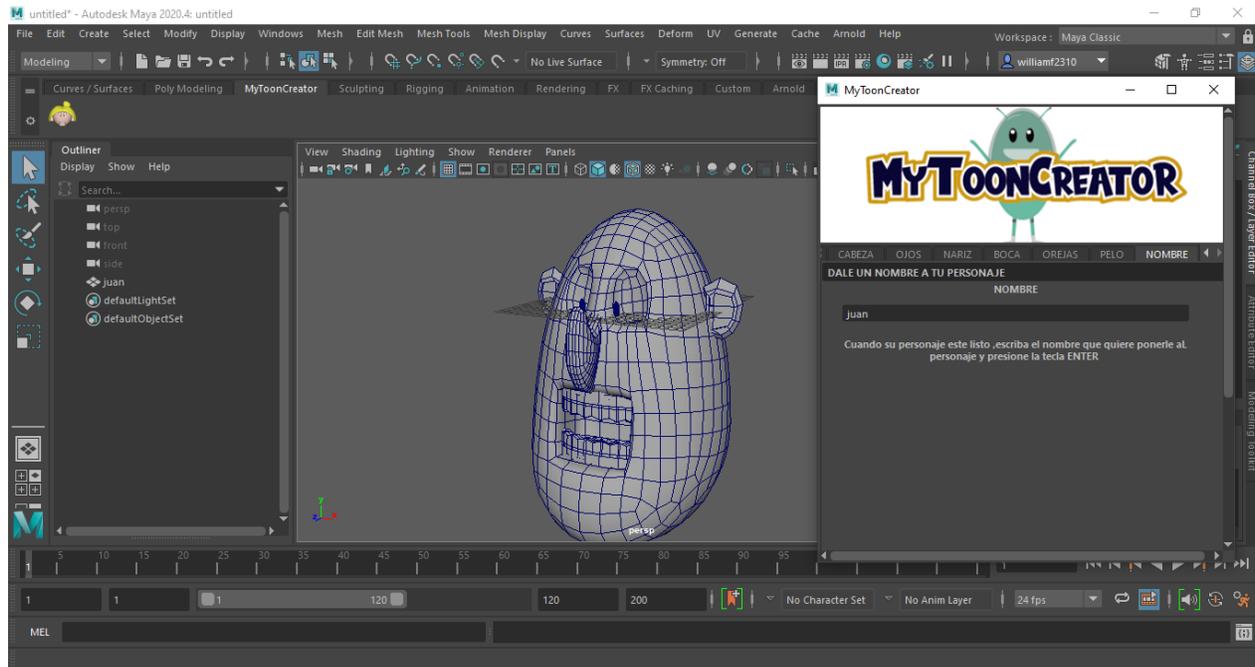
45 Figura Resultado 1



Nota: Elaboración propia

46 Figura

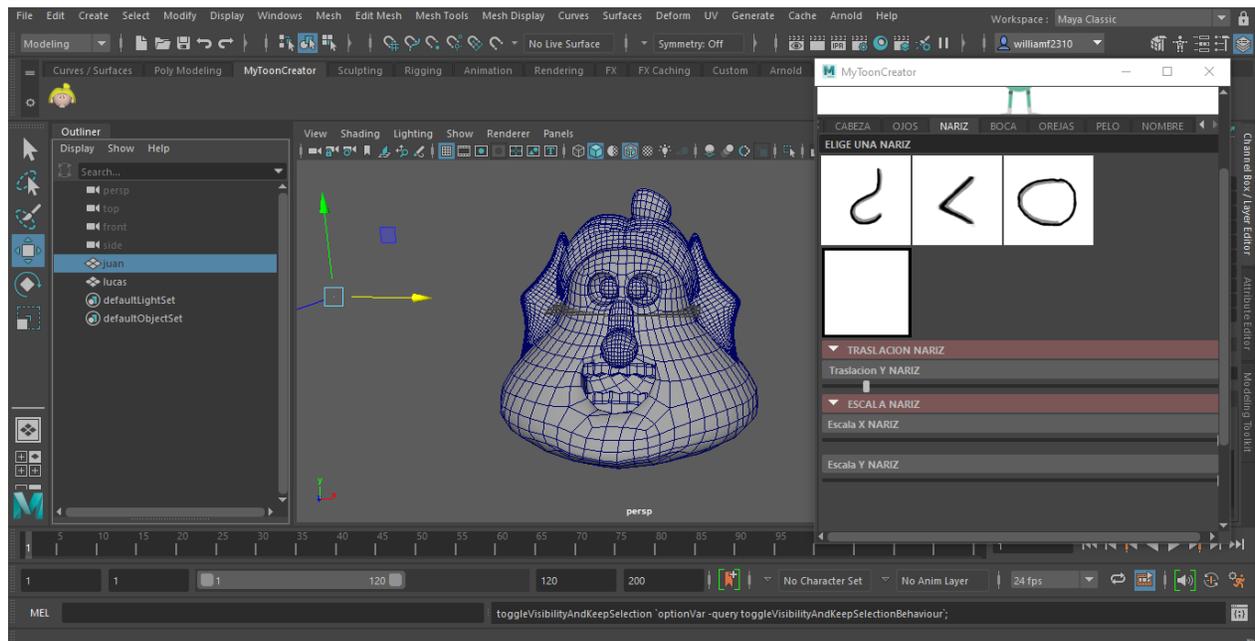
Resultado 2



Nota: Elaboración propia

47 Figura

Resultado 3



Nota: Elaboración propia

X. DISCUSIÓN

El plugin, el cual cuenta con un panel que tiene funciones de creación de personajes cartoon 3D listos para ser usados y con opciones de personalización, está compuesto por sliders, estos hacen que el proceso de edición del personaje sea más eficaz ya que se observa en tiempo real los cambios que se realizan.

Además la forma en que funciona este plugin es por medio de un instalador, el cual contiene carpetas que se instalan de forma automática dentro de los documentos del computador, estas carpetas contienen modelos 3D que se crearon anteriormente para luego poder ser implementados dentro del software por medio de botones, estos están representados por imágenes con el fin de dar una vista previa al modelo que se creará, una vez importados, estos podrán ser modificados en tamaño, rotación y traslación por los usuarios a gusto propio.

Siguiendo con el proceso de creación del personaje, el plugin de manera automática cuando el usuario le asigna un nombre a su modelo combinará y limpiará las geometrías que posteriormente se convertirán en el personaje manteniendo las características que le dio el usuario. El plugin podrá seguir siendo utilizado para crear más personajes, con la condición de que se debe cambiar el nombre y la posición del modelo creado en el espacio de trabajo.

De esta forma se optimiza el tiempo en el pipeline ya que no se tiene que crear un modelo 3D desde cero, sino que se hace uso de los recursos dados por el plugin, así mismo estos resultados positivos se ven reflejados en las encuestas, en las cuales los usuarios indicaron que el manejo es sencillo, contiene un buen diseño y es útil para la creación de sus personajes cartoon.

XI. CONCLUSIONES

My Toon Creator al ser un plugin no solo se convierte en un complemento que añade nuevas características o mejoras al programa, en este caso al software de Autodesk Maya, sino que también evita instalar aplicaciones de terceros los cuales podrían cumplir la misma función, pero requerirían mayor almacenamiento en el computador del usuario, afectando el rendimiento de este.

Además, My Toon Creator fue creado usando el lenguaje de programación Python, el cual al ser un lenguaje muy versátil podría ser implementado en otro software de modelado 3D como 3Ds Max, Blender, entre otros.

Al tener un panel de creación el cual está conformado con textos, botones, sliders e imágenes permite al usuario comprender rápidamente las funciones del plugin, además de otorgar libertad y comodidad a la hora de la creación y la personificación de sus personajes.

Al finalizar la creación del personaje, el panel tendrá una casilla que tendrá como función asignarle un nombre al modelo creado para luego poder optimizar y limpiar su topología, de esta forma el resultado estará acorde a las expectativas del usuario y podrá ser usado con otros propósitos u ámbitos como por ejemplo en el área de la animación.

Con base en los avances y encuestas realizadas en este proyecto, se manifiestan oportunidades de mejora en el mismo, como por ejemplo la implementación del cuerpo humano ya que este proyecto solo se enfocó en la cabeza, también de la inclusión de accesorios tales como gafas, sombreros, aretes, ropa, etc. La oportunidad de agregar texturas o colores a los rasgos de los personajes, incluyendo también la posibilidad de añadir el rigging a los personajes para poderlos animar o mejoras en la personificación.

REFERENCIAS

- Asunción, F. J. (2017). Diseño y creación de un personaje 3D para un videojuego o animación. Autodesk. (20 de 08 de 2018). *Autodesk*. Obtenido de <https://autode.sk/39dkhQ5>
- AUTODESK. (07 de 12 de 2020). *autodesk*. Obtenido de <https://autode.sk/3PbdMh2>
- Designer, C. C. (09 de 2021). *Toon Figures*. Obtenido de <https://bit.ly/3N76vx6>
- Fariño, G. (2011). *Modelo Espiral de un proyecto de desarrollo de software*.
- gems, A. (2019). *Metodología de desarrollo de software (III)*. Obtenido de <https://bit.ly/3M2lecr>
- González, M. A. (08 de 06 de 2020). Diseño y modelado de personajes para la reconstrucción histórica virtual de la Laguna en el siglo XVI. pág. 7.
- Mathews, J. (13 de 09 de 2019). *An Update on Adobe Fuse as Adobe Moves to the Future of 3D & AR Development*. Obtenido de <https://www.adobe.com/es/products/fuse.html>
- Mercado, P. (8 de 07 de 2020). *Industria Animación*. Obtenido de <https://bit.ly/3P5DlA2>
- Pinterest*. (2015). Obtenido de <https://bit.ly/3xYvAp8>
- Poser. (09 de 2021). *Poser Software*. Obtenido de <https://www.posersoftware.com/>
- Software, R. T. (09 de 2021). *Reallusion*. Obtenido de <https://www.reallusion.com/iclone/>
- VISUS, A. (10 de 2020). *ESIC*. Obtenido de <https://bit.ly/39W3ktT>