



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior de Jaén

Trabajo Fin de Grado

HERRAMIENTA PARA LA SIMULACIÓN DE TEJIDOS

Alumno: Pedro Antonio Reca Perales

Tutores: Prof. D. Jiménez Pérez, Juan Roberto
Prof. D. Jurado Rodríguez, Juan Manuel

Dpto: Informática

Septiembre, 2021

Agradecimientos

En primer lugar a mi familia, en particular a mis padres, mi hermana y mis abuelas, que han estado a lo largo de todo este tiempo apoyándome y animándome a seguir adelante.

A mis tutores Roberto y Juanma, por su ayuda en la planificación, información y organización en este Trabajo de Fin de Grado.

A todos mis amigos y compañeros que siempre me han apoyado durante esta etapa.

A todos los que han evaluado el software desarrollado, por dedicarme su tiempo y sus comentarios.

Por último, expresar mi más sentido agradecimiento a la Universidad de Jaén y su profesorado por acogerme dentro de sus aulas y hacerme sentir como en casa. Sin duda, ha sido un período de aprendizaje científico y personal.

Resumen

En este trabajo proponemos una herramienta para la simulación virtual de complementos de moda, particularmente bufandas y pañuelos, incluyendo componentes como los clásicos flecos. El motivo es ayudar a futuros usuarios no expertos que visualizar el comportamiento de estas prendas. Se ha desarrollado un plugin para Blender específico para resolver esta problemática y se han modelado distintos ejemplos de uso.

Índice general

1. Introducción	1
1.1. Introducción	1
1.2. Estado del arte	1
1.2.1. Trimirror	2
1.3. Objetivos	3
1.4. Metodología	3
1.4.1. Metodologías de desarrollo software tradicionales	4
1.4.1.1. Waterfall(cascada):	4
1.4.1.2. Prototipado:	4
1.4.1.3. Espiral:	4
1.4.1.4. Incremental:	5
1.4.1.5. Diseño rápido de aplicaciones (RAD):	5
1.4.2. Metodologías de desarrollo software ágiles	5
1.4.3. Kanban:	5
1.4.4. Scrum:	6
1.4.5. Lean:	6
1.4.6. Programación extrema (XP):	6
1.5. Planificación	6
1.6. Presupuesto	8
1.7. Tecnologías utilizadas	9
1.7.1. Blender	9
1.7.2. Makehuman	9
1.7.3. Mixamo	9
1.7.4. Gimp	10
1.7.5. Normal map online	10
2. Desarrollo del proyecto	11
2.1. Desarrollo del proyecto	11
2.2. Iteración 1: Primeros pasos	11
2.2.1. Análisis de requisitos funcionales y no funcionales	11
2.2.1.1. Funcionales	11
2.2.2. Implementación	11
2.2.2.1. Comportamiento de Tejido	12
2.2.2.2. Generación del pelo asociado a la prenda	13
2.2.3. Resultados y pruebas	15
2.2.3.1. Entregables	15
2.2.3.2. Dificultades	16
2.3. Iteración 2: Realización del plug-in	16
2.3.1. Análisis de requisitos funcionales y no funcionales	16
2.3.1.1. Funcionales	16
2.3.1.2. No funcionales	17
2.3.2. Diseño	17

2.3.2.1.	Interfaz	17
2.3.2.1.1.	Panel selección material	17
2.3.2.1.2.	Panel flecos	17
2.3.3.	Historia de usuario	18
2.3.4.	Implementación	18
2.3.4.1.	Comportamiento de tela	19
2.3.4.1.1.	Suavizado de la prenda	20
2.3.4.2.	Generación del pelo asociado a la prenda	21
2.3.4.3.	Inserción de diferentes tipos de materiales/tejidos	22
2.3.5.	Resultados y pruebas	23
2.3.5.1.	Entregables	23
2.3.6.	Dificultades	25
2.3.6.0.1.	Selección de los vértices	26
2.3.6.0.2.	Prenda que atraviesa objeto	26
2.3.6.0.3.	Comportamiento del pelo	26
2.4.	Iteración 3: Personalización del complemento	26
2.4.1.	Análisis de requisitos funcionales y no funcionales	26
2.4.1.1.	Funcionales	26
2.4.1.2.	No funcionales	27
2.4.2.	Diseño	27
2.4.2.1.	Interfaz	27
2.4.3.	Historia de usuario	27
2.4.4.	Implementación	28
2.4.4.1.	Bump mapping	28
2.4.4.2.	Shader editor	28
2.4.4.2.1.	Nodo principled BSDF	29
2.4.4.2.2.	Nodo normal map	30
2.4.4.2.3.	Nodo bump	31
2.4.4.2.4.	Nodos imagen	32
2.4.4.2.5.	Nodo mapping	32
2.4.4.3.	Aplicado de una imagen propia a nuestro complemento	33
2.4.4.3.1.	Creación de el mapeado de alturas (Height Map)	34
2.4.4.3.2.	Creación de la imagen normal	36
2.4.5.	Resultados y pruebas	37
2.4.5.1.	Entregables	37
2.4.5.2.	Dificultades	40
2.5.	Iteración 4: Realización de simulaciones e inclusión del viento en la escena	41
2.5.1.	Análisis de requisitos funcionales y no funcionales	41
2.5.1.1.	Funcionales	41
2.5.1.2.	No funcionales	41
2.5.2.	Diseño	41
2.5.2.1.	Interfaz	41
2.5.3.	Historia de usuario	42
2.5.4.	Implementación	42
2.5.4.1.	Añadir un individuo a la escena	42

2.5.4.2. Animar nuestro modelo	43
2.5.4.2.1. Animar modelo en Mixamo	43
2.5.5. Resultados y pruebas	45
2.5.5.1. Entregables	45
2.5.5.2. Dificultades	48
3. Validación del proyecto	49
3.1. Usuario 1	49
3.2. Usuario 2	49
3.3. Usuario 3	50
3.4. Conclusiones tras el feedback	51
4. Modelo de negocio	52
4.1. Objetivos	52
4.2. Análisis del entorno	52
4.3. Plan de marketing	52
5. Conclusiones y trabajo futuro	53
6. Manual de usuario	54
6.1. Instalación	54
6.2. Lanzamiento del script	60
6.3. Añadir otros modelos a la escena	65
6.4. Renderizado de la animación	70
7. Bibliografía	74

Índice de figuras

1.1. Ejemplo del modelo de la tienda	2
1.2. Captura de la herramienta de TriMirror [12]	3
1.3. Diagrama de Gantt.	7
2.1. Primera bufanda realizada en el modo de Modeling	12
2.2. Solo una cara por lado	12
2.3. Divido en muchas caras	12
2.4. Bufanda que no se ve realista, hay que aplicar los suavizados.	13
2.5. Bufanda suavizada.	13
2.6. Vértices seleccionados de un extremo de la bufanda, lugar donde ira el pelo que simulara los flecos.	14
2.7. Imagen del sistema de partículas con la opción de pelo	15
2.8. Imagen donde se une el grupo de vértices a el sistema de partículas, para que aparezca el pelo en una zona determinada	15
2.9. Imagen del resultado de la iteración 1, si haces clic sobre la imagen aparecerá un vídeo con la animación.	16
2.10. Panel de selección de material.	17
2.11. Panel para la activar o desactivar los flecos de la prenda.	18
2.12. Código con el que se crea un cubo, que está escalado para que tenga la forma de una bufanda.	18
2.13. Elementos básicos de la escena generados en el Script.	19
2.14. Izquierda objeto con el modificador de tejido pero sin aplicar nada de suavizado. Derecha objeto con los modificadores y opciones de suavizado aplicadas.	20
2.15. Observamos los vértices seleccionados de un extremo de la bufanda, que serán añadidos a el grupo de vértices.	21
2.16. Imagen con todos los parámetros y configuración del sistema de partículas que simula pelo en nuestro complemento.	22
2.17. Captura del código donde se puede observar el ejemplo de dos materiales diferentes Denim y piel	23
2.18. Primer resultado en la iteración 2 donde la prenda se incrusta en el objeto en el que cae y los pelos de la prenda no se comportan de manera realista, si haces clic en la imagen te llevará a un vídeo de este ejemplo	24
2.19. Resultado final de la iteración 2, donde la prenda tiene un comportamiento adecuado y el pelo asociado se comporta con normalidad, al hacer clic sobre la imagen te redirigirá a un vídeo con este resultado.	25
2.20. Ventana que aparece al pulsar el botón de añadir textura, compuesta de un input para introducir el nombre de la textura deseada y un botón para confirmar.	27
2.21. Ejemplo de las imágenes necesarias para poder aplicar una textura correctamente. Izq:Imagen Color, Centro:Height, Dcha:Normal	28

2.22.shader editor con todos los nodos necesarios para aplicar nuestra textura.	29
2.23.Nodo Principled BSDF	30
2.24.Nodo Normal Map.	31
2.25.Nodo Bump.	31
2.26.Nodos Imagen de nuestro proyecto, donde cargamos las imágenes de color, normal y Height Map.	32
2.27.Mapping Node.	33
2.28.Imagen obtenida desde un dispositivo móvil para aplicar a nuestro objeto.	34
2.29.Ventana de niveles que hay que ajustar.	35
2.30.Mapeado de alturas calculado de nuestra imagen.	35
2.31.Imagen de la pagina para calcular la imagen normal.	36
2.32.Imagen normal de nuestra imagen.	36
2.33.Textura sin la técnica de Bump Mapping.	37
2.34.Textura con la técnica de Bump Mapping.	37
2.35.Textura sin ajustar usando el nodo mapping.	37
2.36.Textura ajustada gracias al nodo mapping.	37
2.37.Imagen del complemento con una textura creada a partir de una foto con un dispositivo móvil.	38
2.38.Ejemplo del complemento con una textura.	39
2.39.Ejemplo del complemento con una textura.	39
2.40.Ejemplo del complemento con una textura.	40
2.41.Ejemplo del complemento con una textura.	40
2.42.Panel para la activar o desactivar el viento en la escena.	42
2.43.Imagen del programa Makehuman [7] que nos permite crear el modelo de una persona.	43
2.44.Exportar modelo desde el programa de Makehuman. [7]	43
2.45.Menú de animaciones de Mixamo [6].	44
2.46.Ventana de previsualización que aparece tras cargar nuestro personaje en Mixamo [6]	44
2.47.Ventana de descarga de Mixamo con las opciones disponibles.	45
2.48.Imagen de la prueba 1 donde se ve la simulación de la caída de la bufanda sobre un ovalo y un plano, todo generado en el script.	45
2.49.Imagen de la simulación de como se comporta la prenda al haber viento en la escena.	46
2.50.Imagen de la simulación de la caída de la bufanda sobre una persona que está en una posición fija.	47
2.51.Imagen de la prueba en la que vemos el comportamiento de la bufanda en una persona que esta andando.	47
6.1. Pagina Web de Blender donde se ve la opción de descargar el programa, con las versiones para los diferentes sistemas operativos	54
6.2. Instalador de Blender	55
6.3. Ejecución instalador de Blender	55
6.4. Ventana de instalador de Blender	56
6.5. Ventana de instalador de Blender con los términos de licencia.	57

6.6. Ventana de instalador de Blender con el directorio de instalación. . . .	58
6.7. Ultima ventana de instalación de Blender.	59
6.8. Blender instalado.	60
6.9. Primer menú que aparece al abrir Blender.	61
6.10. Imagen del código en Gitlab que podrá ser descargado gratuitamente.	61
6.11. Imagen del apartado de scripting donde la flecha roja indica que hay que pulsar para crear un nuevo fichero de código.	62
6.12. Ventana de guardado de fichero donde se indicara el nombre del archivo y el directorio donde se guardará.	62
6.13. Captura con una flecha que indica que botón hay que clicar para ejecutar el script.	63
6.14. Captura de la escena tras ejecutar el script en el modo layout, indicando que botón pulsar para activar los paneles de nuestra interfaz.	63
6.15. Captura de la escena tras ejecutar el script en el modo layout, con los paneles activados.	64
6.16. Captura del código del script donde vemos la ruta de las imágenes.	64
6.17. Previsualización de la animación en el modo layout.	65
6.18. Importar animación o modelo.	66
6.19. Ventana para importar animación o modelo.	67
6.20. Animación importada.	67
6.21. Eliminar esqueleto de la escena.	68
6.22. Esqueleto eliminado de la escena.	69
6.23. Elementos que tendrán que estar seleccionados para poder mover o escalar correctamente a la animación/modelo.	70
6.24. Captura donde se ven los parámetros modificables para el render como: directorio donde se guarda el resultado, formato y frames donde empezara y acabara la animación.	71
6.25. Animation Render.	72
6.26. Ventana de previsualización de renderizado mientras se realiza el render.	72
6.27. Resultados en formato vídeo, en el directorio elegido.	73

1. Introducción

1.1. Introducción

La digitalización de la ropa es un objetivo presente en el campo de los Gráficos y la Animación 3D, con importantes aplicaciones en muchos ámbitos como: marketing, videojuegos, pasarelas virtuales y diseño de moda.

Se han producido importantes avances en el mundo de la moda, tanto a nivel de simulación de movimiento, como en la calidad y realismo de los tejidos. No obstante, este sector todavía precisa de expertos tanto para el modelado como para el diseño de la simulación. Con este trabajo se pretende ofrecer al usuario una aplicación añadida (*plug-in*) en la que sin necesidad de un conocimiento avanzado poder realizar simulaciones virtuales de complementos tales como bufandas, pañuelos, etc. De la prenda elegida, el usuario podrá modificar el material de la que está compuesta (algodón, seda, piel, goma), la textura o poder decidir si la prenda tiene o no sus flecos típicos. Adicionalmente, se han añadido efectos como viento para poder comprobar como se comporta el tejido ante determinadas situaciones.

La herramienta elegida para el desarrollo de este trabajo es Blender 2.92.0 [8] que es un programa gratuito, de código abierto y multiplataforma que está enfocado principalmente al modelado, iluminación, animación y creación de gráficos en 3D. Blender posee un apartado de scripting donde se puede ampliar la funcionalidad, en áreas como animación, visualización, creación de objetos, etc.

1.2. Estado del arte

La simulación física de diferentes tipos de materiales del mundo real ha sido un tema ampliamente estudiado en Informática Gráfica. Del mismo modo, el uso de contenido 3D en la web es cada vez más frecuente, surgiendo la necesidad de la generación de contenido virtual fotorrealista. Destacar el avance en el realismo y comportamiento de los materiales. Por ejemplo hoy en día existen tiendas de ropa online en las que puedes crear tus propios tejidos con medidas personalizadas, ya que tienen la opción de crear tu propio avatar 3D en el que puedes ver cómo te sienta la ropa, lo cómoda que es y cómo complementa tu silueta, en la Figura 1.1 se puede apreciar un ejemplo de un modelo de una tienda que tiene esta función. También estará animada para poder ver cómo se mueve el tejido tal y como lo hace en la vida real. Ejemplo de una herramienta online que ofrece estos servicios [12].



Figura 1.1: Ejemplo del modelo de la tienda

Hoy por hoy existen Add-ons en los que se puede ver como se comporta la ropa en modelos humanos articulados [19]. Incluso existen datasets con secuencias de modelos humanos vestidos que contienen una gran variabilidad en el tipo de prenda, la forma, el tamaño, el tejido, etc. La ropa se podrá simular sobre miles de secuencias de poses y formas corporales [16]. Además existen técnicas que permiten abordar eficazmente las colisiones entre la prenda y el cuerpo ya que es uno de los principales problemas en la simulación de tejidos/ropa [20]. También encontramos otros métodos que dotan de más realismo a las arrugas de las telas [18] y otras técnicas para colocar con más precisión una prenda sobre un cuerpo basada en el procesamiento de nubes de puntos 3D con redes profundas para extraer características de ésta [17]. Existen métodos que planifican el movimiento que conectan directamente las manipulaciones de la tela con los cambios de la misma, mediante una red neuronal [21]. Estos métodos la entrenan para poder predecir el estado de la tela después de la manipulación/movimiento [15].

1.2.1. Trimirror

Herramienta que hemos puesto de ejemplo antes, Trimirror [12] es un sistema de prueba virtual verdadero, sin compromisos y en tiempo real, que permite a los consumidores y a los diseñadores experimentar la ropa de la vida real en sus precisos modelos virtuales en movimiento, así como la visualización instantánea del ajuste en plataformas online, de escritorio o móviles. Esto les permite ver dónde la prenda queda ajustada o suelta, dónde sobresale o se hunde, y cómo se comporta cuando se mueven o se colocan en varias poses, y por lo tanto lleva toda la experiencia de compra en la tienda y en línea a un nuevo nivel de compromiso y entretenimiento. En la Figura 1.2 se puede observar una imagen de la web de esta

herramienta.

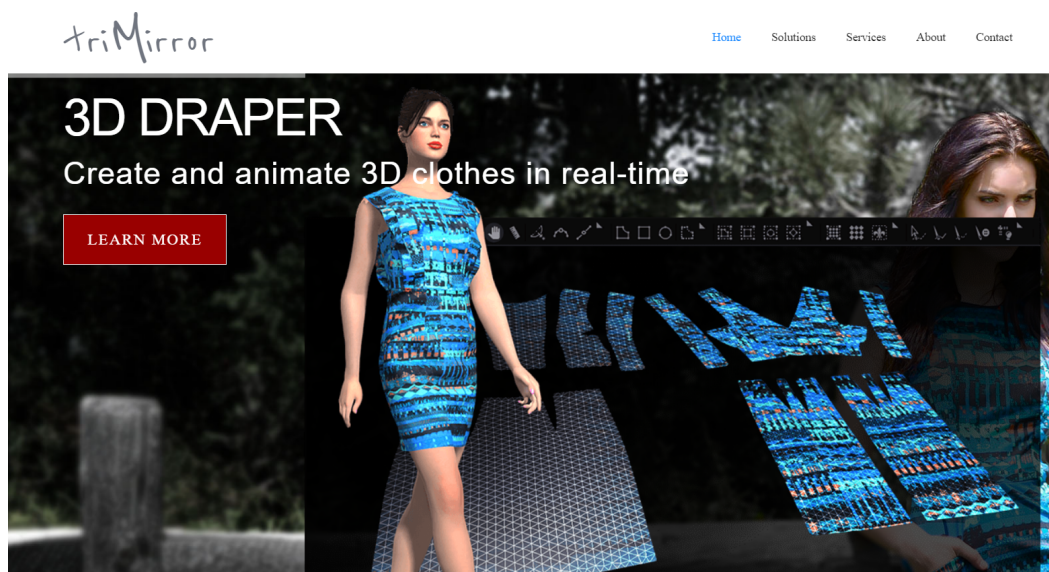


Figura 1.2: Captura de la herramienta de TriMirror [12]

Tiene algunas opciones muy útiles como: Draper3D es una herramienta para diseñadores de moda, animadores, desarrolladores de juegos y artistas 3D. En la que se pueden crear fácilmente prendas en 3D a partir de patrones en 2D, ver cómo se ajustan y se mueven en tiempo real, con diferentes tejidos y colores, y en diferentes tamaños y formas de cuerpo.

También tiene una solución para el comercio electrónico en pequeños comercios de moda online en los cuales TriMirror se puede integrar fácilmente en sus plataformas. Además permite a los usuarios el uso de la herramienta en smartphones o tablets.

1.3. Objetivos

El objetivo de este trabajo es desarrollar una herramienta que ayude a comprender y llevar a cabo una simulación para el modelado de tejidos, simulando el movimiento e interacción de estos. Con este trabajo se pretende ofrecer al usuario una aplicación añadida (*plug-in*) en el que pueda ver como interactúan diferentes tipos de tejidos/materiales en diferentes situaciones, ofreciendo la opción de poder cambiarlos siempre que quiera.

1.4. Metodología

En esta sección se describirá la metodología de desarrollo de software empleada para nuestro proyecto, en primer lugar mencionaremos las diferentes metodologías y por último justificaremos la que ha sido elegida para este trabajo.

En la actualidad se pueden diferenciar dos grandes grupos de metodologías de desarrollo de software: las ágiles y las tradicionales. A continuación, se explican las características de cada una de ellas

1.4.1. Metodologías de desarrollo software tradicionales

Las metodologías de desarrollo de software tradicionales se caracterizan por definir total y rígidamente los requisitos al inicio de los proyectos de ingeniería de software. Los ciclos de desarrollo son poco flexibles y no permiten realizar cambios, al contrario que las metodologías ágiles, lo que ha propiciado el incremento en el uso de las segundas. Las etapas se suceden una tras otra y no se puede empezar la siguiente sin terminar la anterior. Tampoco se puede volver hacia atrás una vez se ha cambiado de etapa. Estas metodologías, no se adaptan nada bien a los cambios, y el mundo actual cambia constantemente.

Las principales metodologías tradicionales son:

1.4.1.1. Waterfall(cascada):

Es una metodología en la que las etapas se organizan de arriba a abajo, de ahí el nombre. Se desarrollan las diferentes funciones en etapas diferenciadas y obedeciendo un riguroso orden. Antes de cada etapa se debe revisar el producto para ver si está listo para pasar a la siguiente fase. Los requisitos y especificaciones iniciales no están predispuestos para cambiarse, por lo que no se pueden ver los resultados hasta que el proyecto ya esté bastante avanzado.

1.4.1.2. Prototipado:

Se basa en la construcción de un prototipo de software que se construye rápidamente para que los usuarios puedan probarlo y aportar feedback. Así, se puede arreglar lo que está mal e incluir otros requerimientos que puedan surgir. Es un modelo iterativo que se basa en el método de prueba y error para comprender las especificidades del producto.

1.4.1.3. Espiral:

Es una combinación de los dos modelos anteriores, que añade el concepto de análisis de riesgo. Se divide en cuatro etapas: planificación, análisis de riesgo, desarrollo de prototipo y evaluación del cliente. El nombre de esta metodología da nombre a su funcionamiento, ya que se van procesando las etapas en forma de espiral. Cuanto más cerca del centro se está, más avanzado está el proyecto.

1.4.1.4. Incremental:

En esta metodología de desarrollo de software se va construyendo el producto final de manera progresiva. En cada etapa incremental se agrega una nueva funcionalidad, lo que permite ver resultados de una forma más rápida en comparación con el modelo en cascada. El software se puede empezar a utilizar incluso antes de que se complete totalmente y, en general, es mucho más flexible que las demás metodologías.

1.4.1.5. Diseño rápido de aplicaciones (RAD):

Esta metodología permite desarrollar software de alta calidad en un corto periodo de tiempo. Los costes son mucho más altos y el desarrollo más flexible, aunque requiere una mayor intervención de los usuarios. Por otro lado, el código puede contener más errores, y sus funciones son limitadas debido al poco tiempo del que se dispone para desarrollarlas. El objetivo es iterar el menor número posible de veces para conseguir una aplicación completa de forma rápida.

1.4.2. Metodologías de desarrollo software ágiles

Estas metodologías son las más utilizadas hoy en día debido a su alta flexibilidad y agilidad. Los equipos de trabajo que las utilizan son mucho más productivos y eficientes, ya que saben lo que tienen que hacer en cada momento. Además, la metodología permite adaptar el software a las necesidades que van surgiendo por el camino, lo que facilita construir aplicaciones más funcionales. Las metodologías ágiles se basan en la metodología incremental, en la que en cada ciclo de desarrollo se van agregando nuevas funcionalidades a la aplicación final. Sin embargo, los ciclos son mucho más cortos y rápidos, por lo que se van agregando pequeñas funcionalidades en lugar de grandes cambios.

Este tipo de metodologías permite construir equipos de trabajo autosuficientes e independientes que se reúnen cada poco tiempo para poner en común las novedades. Poco a poco, se va construyendo y puliendo el producto final, a la vez que el cliente puede ir aportando nuevos requerimientos o correcciones, ya que puede comprobar cómo avanza el proyecto en tiempo real.

Las principales metodologías ágiles son:

1.4.3. Kanban:

Consiste en dividir las tareas en porciones mínimas y organizarlas en un tablero de trabajo dividido en tareas pendientes, en curso y finalizadas. De esta forma, se crea un flujo de trabajo muy visual basado en tareas prioritarias e incrementando el valor del producto.

1.4.4. Scrum:

Es también una metodología incremental que divide los requisitos y tareas de forma similar a Kanban. Se itera sobre bloques de tiempos cortos y fijos (entre dos y cuatro semanas) para conseguir un resultado completo en cada iteración. Las etapas son: planificación de la iteración (planning sprint), ejecución (sprint), reunión diaria (daily meeting) y demostración de resultados (sprint review). Cada iteración por estas etapas se denomina también sprint.

1.4.5. Lean:

Está configurado para que pequeños equipos de desarrollo muy capacitados elaboren cualquier tarea en poco tiempo. Los activos más importantes son las personas y su compromiso, relegando así a un segundo plano el tiempo y los costes. El aprendizaje, las reacciones rápidas y potenciar el equipo son fundamentales.

1.4.6. Programación extrema (XP):

Es una metodología de desarrollo de software basada en las relaciones interpersonales, que se consideran la clave del éxito. Su principal objetivo es crear un buen ambiente de trabajo en equipo y que haya un feedback constante del cliente. El trabajo se basa en 12 conceptos: diseño sencillo, testing, refactorización y codificación con estándares, propiedad colectiva del código, programación en parejas, integración continua, entregas semanales e integridad con el cliente, cliente in situ, entregas frecuentes y planificación.

1.5. Planificación

Para el desarrollo se utilizará una metodología ágil Scrum. Se han establecido diferentes iteraciones en pequeños lapsos de tiempo. Todas estas iteraciones están vinculadas de forma que cada una suponga un avance con respecto a la anterior.

Las iteraciones en las que se divide nuestro proyecto son:

- **Iteración 1**
 - Elección y conocimiento del entorno de trabajo.
 - Realización del plug-in.
 - **Iteración 2**
 - Creación del complemento y comportamiento de tejido ropa. Aquí se creara nuestro modelo y se le asignara el comportamiento de tela.
 - Pelo del complemento. Durante esta iteración se añadirá a nuestro complemento pelo, usando un sistema de partículas.

- Inserción de los diferentes tipos de tejidos/materiales. En esta fase se añaden diferentes tipos de materiales y telas, para que el usuario pueda ver el comportamiento de la prenda con diferentes tipos de tejido.
- **Iteración 3**
- Personalizar la bufanda. Durante esta iteración se añadirá la opción de que el usuario seleccione una imagen que desee aplicar al complemento.
- **Iteración 4**
- Inclusión de la opción de activar viento en la escena.
- Realización de simulaciones y añadir otros elementos/animaciones.
- Documentación final del proyecto.

En la Figura 1.3 podemos ver un diagrama de Gantt con el tiempo que se dedicará a cada una de las iteraciones. Cada una tiene una semana de margen para que, en el caso de que en la reunión con los tutores, si se comenta algún aspecto a cambiar, poder realizar alguna modificación o cambio. En esta semana también redactaremos parte de la documentación de las iteraciones y se realizaran pruebas o ajustes de parámetros de lo avanzado durante estas semanas.

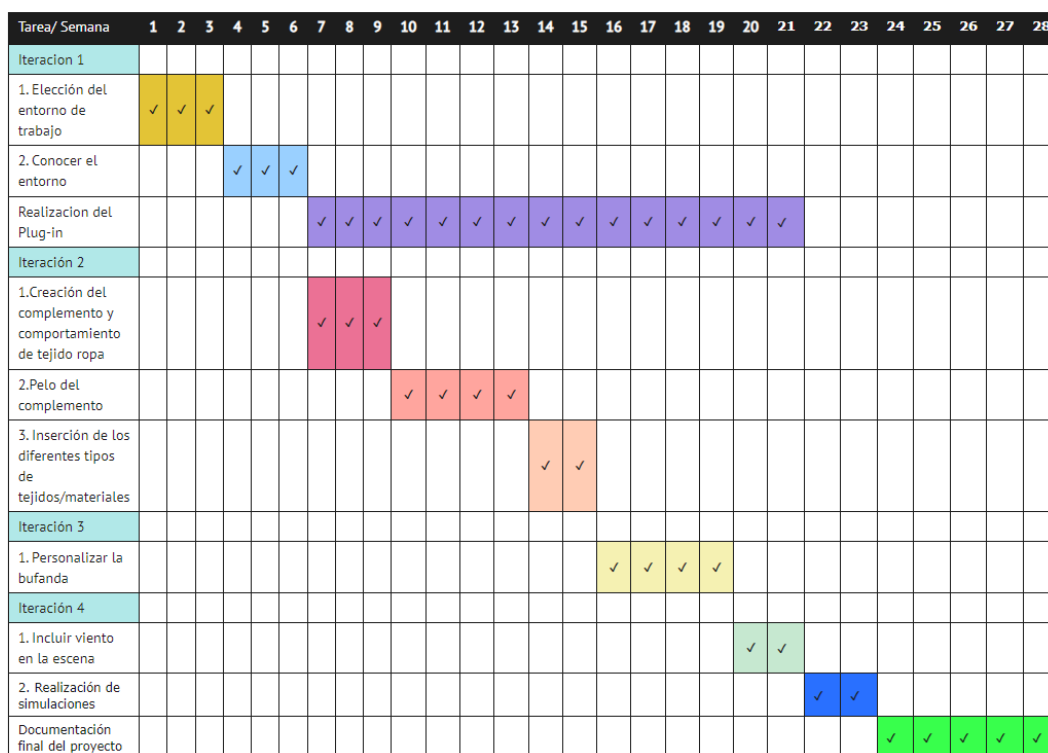


Figura 1.3: Diagrama de Gantt.

1.6. Presupuesto

Para hacer una estimación del coste del proyecto debemos de hacer una estimación de los costes de software, los recursos materiales y coste del personal.

En la tabla 1.1 podemos ver el coste de los programas usados durante la realización del proyecto. Estos son: Blender [8] en el que hemos realizado la mayor parte del proyecto, Gimp [3] que ha sido utilizado para crear imágenes de alturas (height map) para poder aplicar una textura creada a partir de una foto a nuestro modelo, Makehuman [7] usado para crear el modelo de una persona, Mixamo [6] empleado para animar nuestro personaje a la hora de hacer simulaciones y por último Paint para editar algunas imágenes en la documentación. Todos estos programas son de uso gratuito por lo que tienen un coste de 0€.

Programa	Objetivo	Coste
Blender	Modelado y creación del plug-in	0
Gimp	Creación de imágenes de alturas	0
Makehuman	Creación de un modelo de persona	0
Mixamo	Animación de modelos	0
Paint	Edición de imágenes	0

Cuadro 1.1: Presupuesto estimado de los programas usados

En la tabla 1.2 se pueden observar los costes materiales necesarios para poder realizar el proyecto. Para el ADSL que se cobra mensualmente y tiene una factura media de 60€/mes, calcularemos el coste por hora que será de 0.0833€, lo multiplicaremos por el total de horas del proyecto (300), obtenemos un consumo de 25€. La luz y material de oficina tendrán un coste de 30€ aproximadamente en total a lo largo de las 300 horas de proyecto. Nuestro equipo informático con el que se desarrollará el software podrá costar alrededor de 800€, con una vida media de 4 años, nuestro proyecto tiene una duración de 28 semanas (más de medio año), por lo que el coste estimado del equipo de desarrollo será de unos 107€.

Concepto	Objetivo	Coste
Equipo de Desarrollo	Ordenador portátil, ratón, monitor	107€
ADSL	Conexión a internet	25€
Electricidad y material de oficina	Luz y consumo	20€

Cuadro 1.2: Presupuesto estimado de los costes materiales

En la última tabla 1.3 veremos una estimación de los costes de personal para la parte de programación, el salario de un programador junior en España [10] es de 1.556,66€ netos al mes, se ha estimado que el total de horas realizando esta labor han sido de 250 horas. El trabajo de diseñador 3D en España [11] es de 1.994,83€ netos al mes, el tiempo estimado realizando esa función ha sido de 50 horas. Aproximadamente lo que tiene que abonar una empresa a la Seguridad Social por un

empleado al mes corresponden con una cifra aproximada al 30 por ciento del sueldo bruto. Con este calculo la empresa deberá abonar aproximadamente unos 640€.

Rol	Objetivo	Horas	Coste
Programador junior	Programar el Plug-in	250	1.712,326€
Diseñador 3D	Modelo de algunos elementos y trabajo con Blender	50	415,41€
Seguridad Social	Gastos de Seguridad Social	0	640€

Cuadro 1.3: Presupuesto estimado de los costes de personal

El presupuesto inicial del proyecto será de 2.919,736€. Algunos de los gastos citados anteriormente son estimaciones y pueden variar durante el desarrollo del proyecto, puede haber algún gasto imprevisto, lo cual obliga a presupuestar un 15 por ciento mas para tener un margen de seguridad.

Finalmente, el presupuesto del proyecto es de 3.357,69€.

1.7. Tecnologías utilizadas

1.7.1. Blender

Herramienta principal [8] donde se ha realizado la mayor parte del proyecto, la hemos usado para desarrollar nuestro plug-in, modelar y hacer simulaciones y pruebas. Es un programa gratuito, de código abierto y multiplataforma que está enfocado principalmente al modelado, iluminación, animación y creación de gráficos en 3D. Blender siempre ha mantenido una estrecha relación con el lenguaje Python, lo cual permite a los desarrolladores voluntarios programar de forma sencilla. Gracias a esto hemos podido realizar nuestra aplicación.

1.7.2. Makehuman

Makehuman [7] es una aplicación de gráficos 3D por computadora para prototipos de humanos fotorealísticos para ser utilizados en gráficos por computador. Es desarrollado por una comunidad de programadores, artistas y académicos interesados en el modelado tridimensional de personajes. En nuestro proyecto hemos usado esta herramienta para descargar un prototipo de un individuo para la realización de pruebas.

1.7.3. Mixamo

Mixamo [6] es una plataforma para encontrar archivos 3D y animaciones gratis, para ser usado en cualquier proyecto, además tiene un opción en la que puedes subir tu propio modelo y poder añadirle una de las animaciones disponibles en la

pagina. En nuestro proyecto la hemos usado para asignarle una animación a nuestro individuo.

1.7.4. Gimp

GIMP [3] es un programa de distribución gratuita que permite realizar tareas como retoque fotográfico, creación de imágenes y composición. En nuestro proyecto lo hemos usado para crear una imagen de mapeado de alturas.

1.7.5. Normal map online

Es una herramienta online [4] que permite crear una imagen normal a partir de cualquier imagen. En nuestro proyecto la hemos usado para obtener una imagen normal.

2. Desarrollo del proyecto

2.1. Desarrollo del proyecto

En este apartado se procederá a detallar las iteraciones que han comprendido el desarrollo de nuestra aplicación.

2.2. Iteración 1: Primeros pasos

En esta primera iteración se llevará a cabo un proceso de conocimiento de Blender mas a fondo. El objetivo es realizar diferentes pruebas en el apartado de modelado. Con el fin de comprender todo lo necesario para crear una prenda con sus respectivas propiedades y diferentes parámetros.

2.2.1. Análisis de requisitos funcionales y no funcionales

2.2.1.1. Funcionales

- Conocer el entorno.
- Crear objetos rígidos y de tipo tela y ver su comportamiento.
- Funcionamiento básico en objetos de tipo tela.
- Ver como interactúan/colisionan los modelos realizados.

2.2.2. Implementación

En primer lugar se crearán los objetos que pertenecerán a la escena. A todos estos se les tiene que añadir el modificador de colisión de Blender que permite que los objetos choquen e interactúen entre si. En este primer ejemplo tendremos dos esferas, un plano y nuestra prenda que es el objeto principal. En la Figura [2.1](#) se pueden observar todos los elementos de la escena.

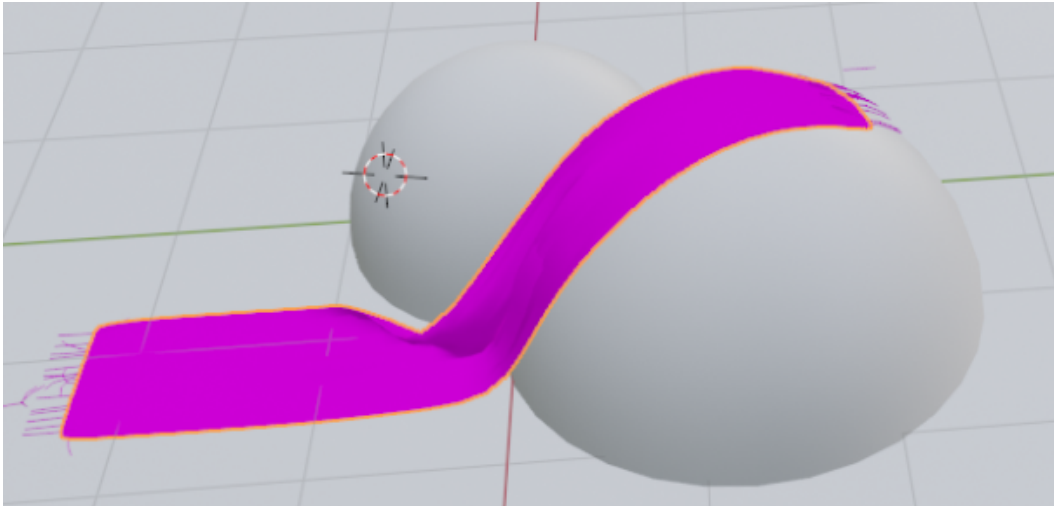


Figura 2.1: Primera bufanda realizada en el modo de Modeling

2.2.2.1. Comportamiento de Tejido

Para que nuestra prenda tenga el comportamiento de la tela tendremos que incluirle el modificador de tejido en propiedades físicas. Tras esto en el modo edición deberemos dividir la prenda en mas caras para que, en la simulación se puedan ver con mejor calidad los pliegues de la tela. Para subdividir la figura en mas caras, en el modo edición hay que tener seleccionado el objeto deseado y hacer clic derecho en subdividir. En las imágenes [2.2](#) y [2.3](#) podremos ver como se vería en el modo edición.

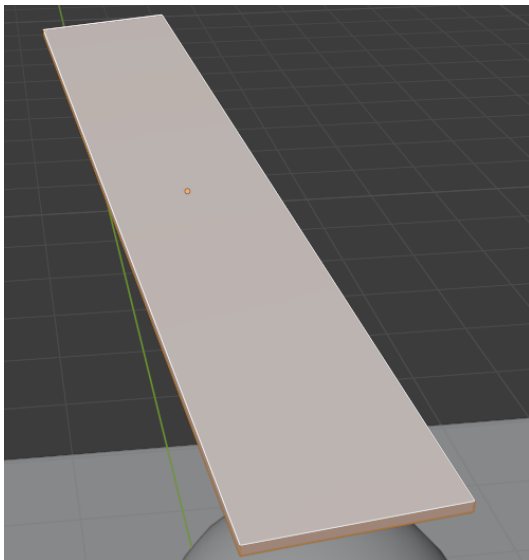


Figura 2.2: Solo una cara por lado

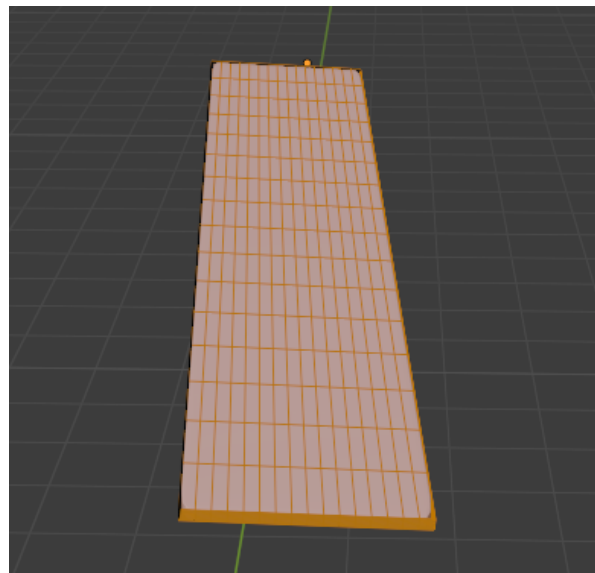


Figura 2.3: Divido en muchas caras

Pero no es suficiente ya que al hacer la simulación nos encontramos con una prenda que no se ve realista, como se puede apreciar en la imagen [2.4](#). Esto es

debido a que la bufanda no está suavizada por lo que habrá que aplicar un suavizado de polígonos y añadir el modificador de subdivision surface que le da una apariencia más pulida a la superficie.

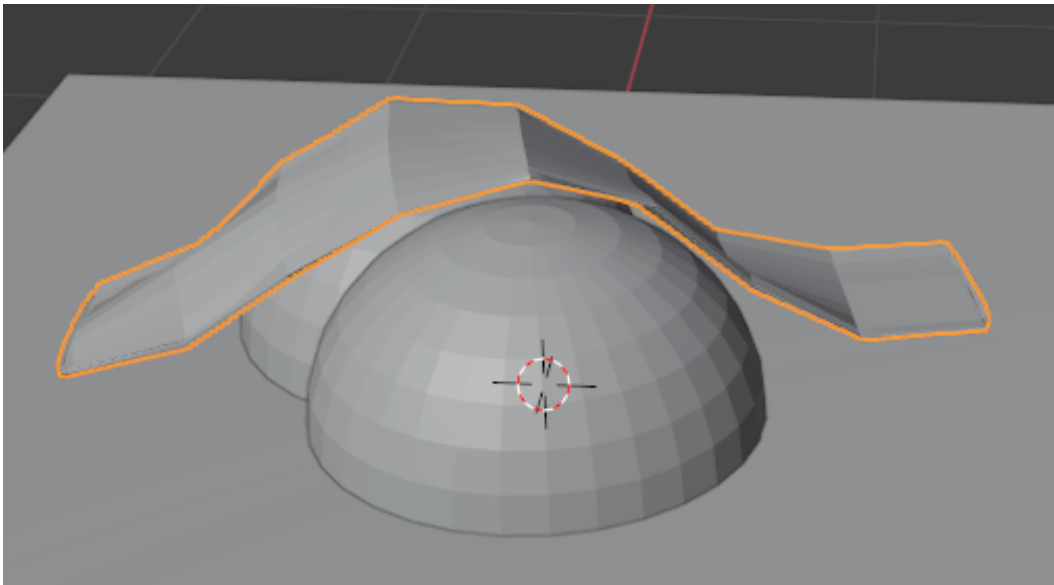


Figura 2.4: Bufanda que no se ve realista, hay que aplicar los suavizados.

Tras añadir estas modificaciones la bufanda quedaría con la apariencia correcta. En la Figura 2.5 se observaría la prenda tras aplicar los métodos de suavizado.

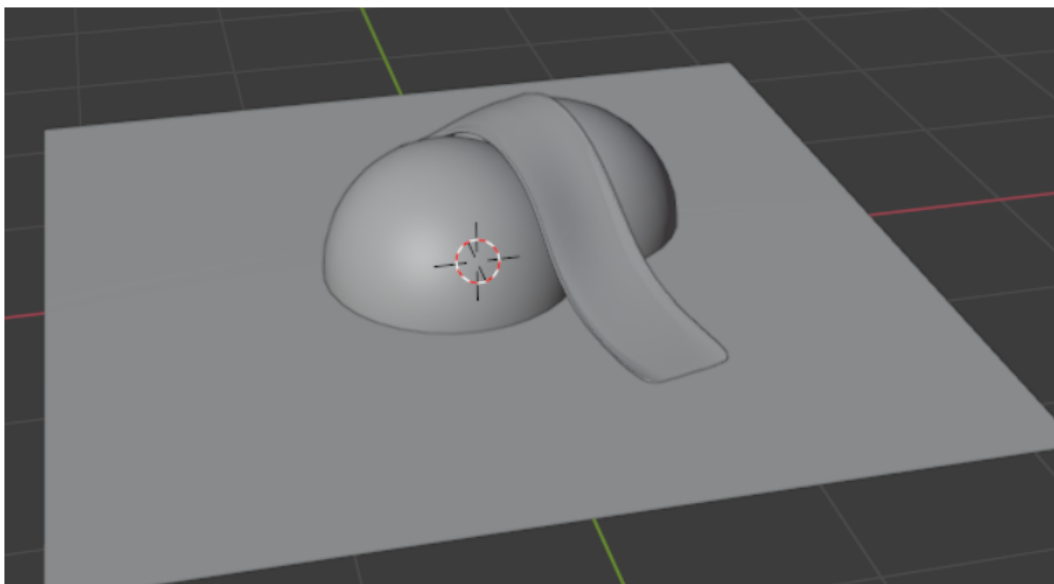


Figura 2.5: Bufanda suavizada.

2.2.2.2. Generación del pelo asociado a la prenda

Para incluir los flecos de la bufanda debemos de crear un grupo de vértices e incluiremos los vértices donde queremos que se sitúen los flecos, en nuestro

caso seleccionaremos los extremos de la prenda. En la Figura 2.6 observamos los vértices seleccionados (color naranja), donde ira el sistema de partículas.

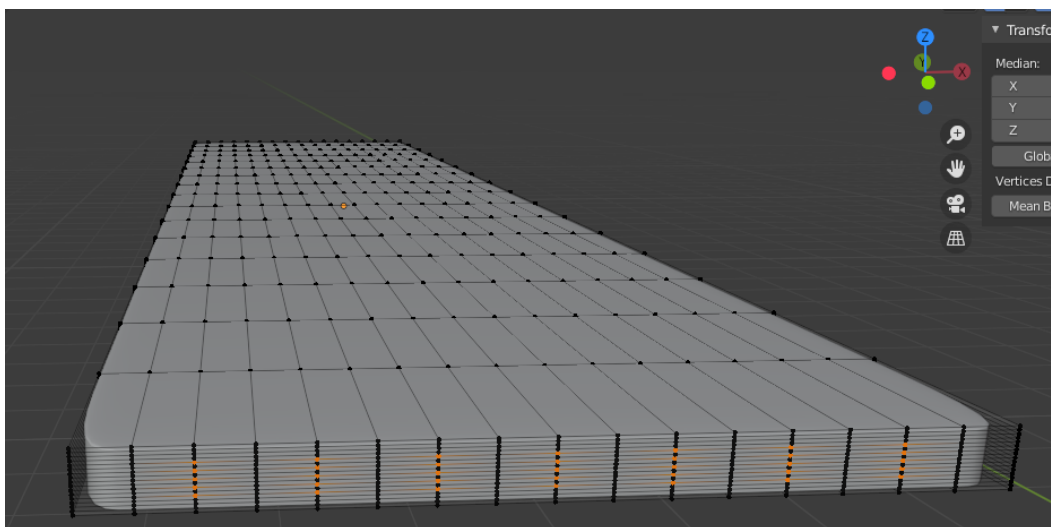


Figura 2.6: Vértices seleccionados de un extremo de la bufanda, lugar donde ira el pelo que simulara los flecos.

Para aplicar un sistema de partículas a nuestro objeto hay que crear uno en la opción de particle properties, ahí seleccionaremos todos los parámetros necesarios para la simulación del pelo. En las Figura 2.7 podemos observar la opción de crear el sistema de partículas y en la Figura 2.8 observamos la opción donde se relaciona el grupo de vértices creado previamente, para que solo aparezca pelo donde deseamos.

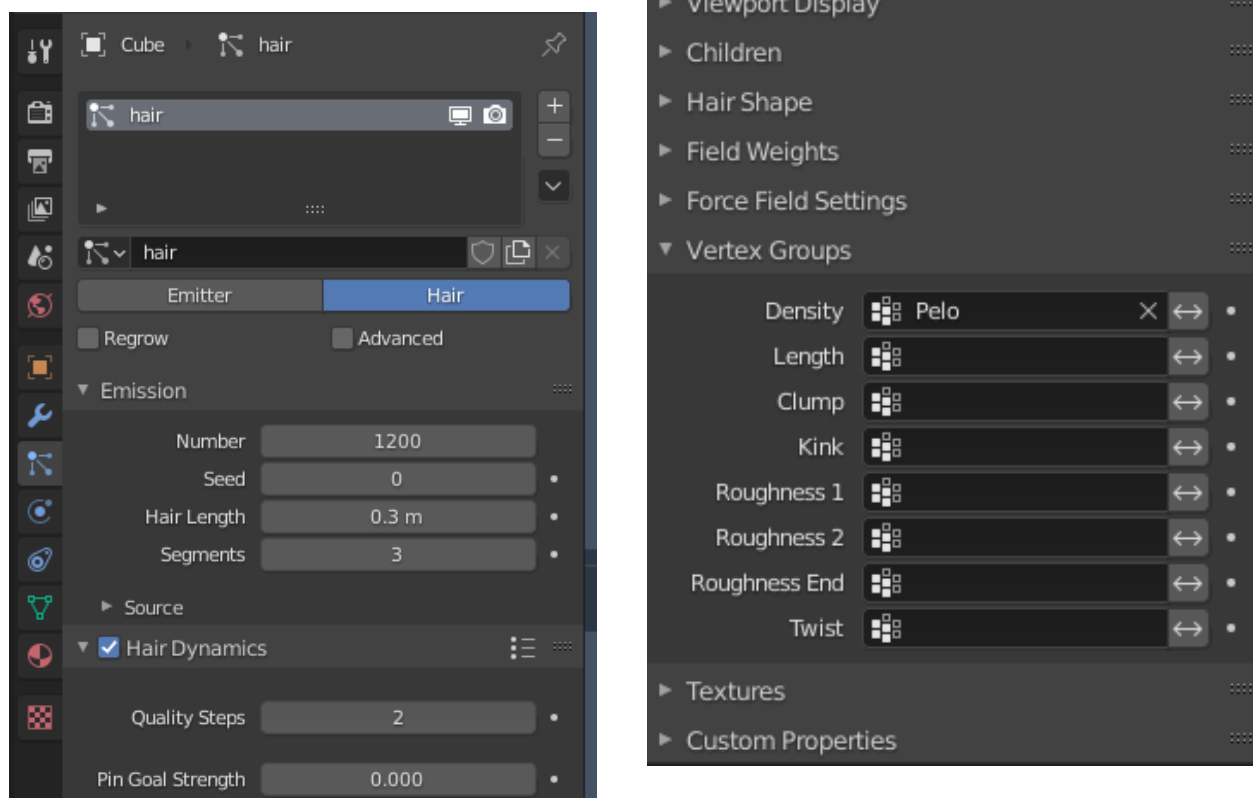


Figura 2.7: Imagen del sistema de partículas con la opción de pelo

Figura 2.8: Imagen donde se une el grupo de vértices a el sistema de partículas, para que aparezca el pelo en una zona determinada

2.2.3. Resultados y pruebas

2.2.3.1. Entregables

El primer resultado de la primera iteración, es una primera prueba realizada en el modo de modelado, esta prueba nos ha ayudado para conocer el entorno, ver como interactúan todos los objetos entre si, especialmente el comportamiento del objeto de tipo tela. En la Figura [2.2.3.1](#) se puede apreciar una imagen de nuestro primer resultado, en el que si haces clic sobre la imagen te redirigirá a un vídeo, donde se puede visualizar la animación.

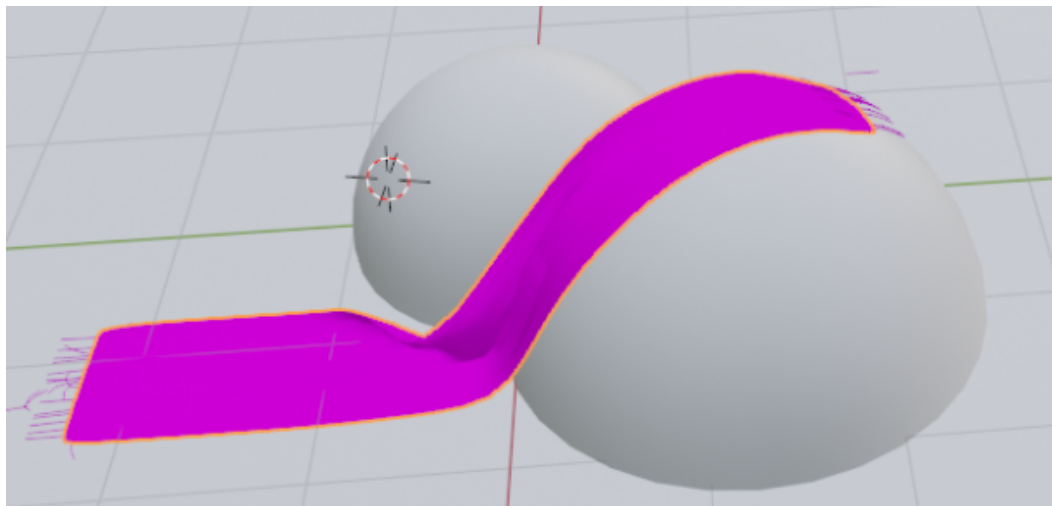


Figura 2.9: Imagen del resultado de la iteración 1, si haces clic sobre la imagen aparecerá un vídeo con la animación.

2.2.3.2. Dificultades

El desarrollo de esta iteración ha ido según lo previsto, gracias a el amplio material que hay en internet he podido crear este primer modelo, no obstante el comportamiento del pelo(flecos) no ha sido el esperado, pero al ser una primera prueba, esto se resolverá mas adelante.

Algunos de los tutoriales que me ayudaron en esta iteración [13] y [14]

2.3. Iteración 2: Realización del plug-in

Esta iteración es donde se realizará la mayor parte del plug-in, durante está lo estimado será, tener el complemento con diferentes opciones de tipos de tejido/-materiales a seleccionar y añadir una opción de poder añadir pelo al complemento.

2.3.1. Análisis de requisitos funcionales y no funcionales

2.3.1.1. Funcionales

- La herramienta tendrá una prenda que tenga comportamiento de tela.
- La herramienta tendrá la opción de que pueda el usuario pueda incluir un sistema de partículas (pelo) en su objeto.
- El sistema debe de tener la opción de que el usuario pueda seleccionar el tipo de material del que esta compuesta la prenda.
- Garantizar una interfaz sencilla e intuitiva al usuario para seleccionar el tipo de material y añadir pelo al complemento

2.3.1.2. No funcionales

- La aplicación no necesita conexión a internet.
- La herramienta se podrá usar en cualquier sistema operativo.

2.3.2. Diseño

2.3.2.1. Interfaz

Para poder hacer cambios en la escena una vez ejecutado el plug-in, hemos incluido una interfaz basada en paneles que permiten modificar nuestro objeto o la escena para tener simulaciones en diferentes ámbitos.

2.3.2.1.1. Panel selección material

En este panel podremos seleccionar el material del que esta compuesto nuestro objeto, los materiales seleccionables son: Algodón, seda, denim, goma y piel. Estos materiales estarán disponibles para seleccionarlos en una lista desplegable. Al pulsar en añadir se efectuarán los cambios adaptando las propiedades a el material escogido. En la Figura 2.10 podremos ver el panel donde se aprecia la lista desplegable y el botón para añadir.



Figura 2.10: Panel de selección de material.

2.3.2.1.2. Panel flecos

La función de este panel es la de añadir o eliminar los flecos típicos de una bufanda, si al pulsar el botón el pelo está desactivado, se creará un sistema de partículas de tipo hair (pelo) con todos los parámetros ajustados a los flecos de una bufanda, por ejemplo: la longitud, cantidad, calidad del pelo, etc. También se asignara el grupo de vértices definido antes a el sistema de partículas para que el pelo aparezca solo en los extremos de la prenda. El control para saber si el sistema de partículas esta activado lo comprobaremos con una variable baliza. En la Figura

2.11 podemos observar el panel para añadir o quitar los flecos, compuesto por un botón para activar o desactivar el sistema de partículas.



Figura 2.11: Panel para la activar o desactivar los flecos de la prenda.

2.3.3. Historia de usuario

- Como usuario quiero poder realizar una simulación de un complemento de ropa, para ver su comportamiento.
- Como usuario quiero tener la opción de seleccionar el material del que esta compuesto el complemento, para observar la prenda con diferentes tipos de materiales.
- Como usuario me gustaría poder elegir si mi prenda tiene pelo o no, para ver como quedaría este en la prenda.

2.3.4. Implementación

Nuestro objeto principal será la prenda, aunque también se crearan otros modelos, que servirán para que una vez ejecutado el programa nuestro complemento pueda interactuar con algo. En primer lugar crearemos los objetos básicos de la escena. En nuestro caso hemos creado un plano y una especie de óvalo/esfera y como no, el complemento de ropa. En la Figura 2.13 vemos los elementos básicos de la escena.

```
bpy.ops.mesh.primitive_cube_add(location=(0,0,7),scale=(0.7,3.5,0.05),rotation=(0,0,0))
```

Figura 2.12: Código con el que se crea un cubo, que está escalado para que tenga la forma de una bufanda.

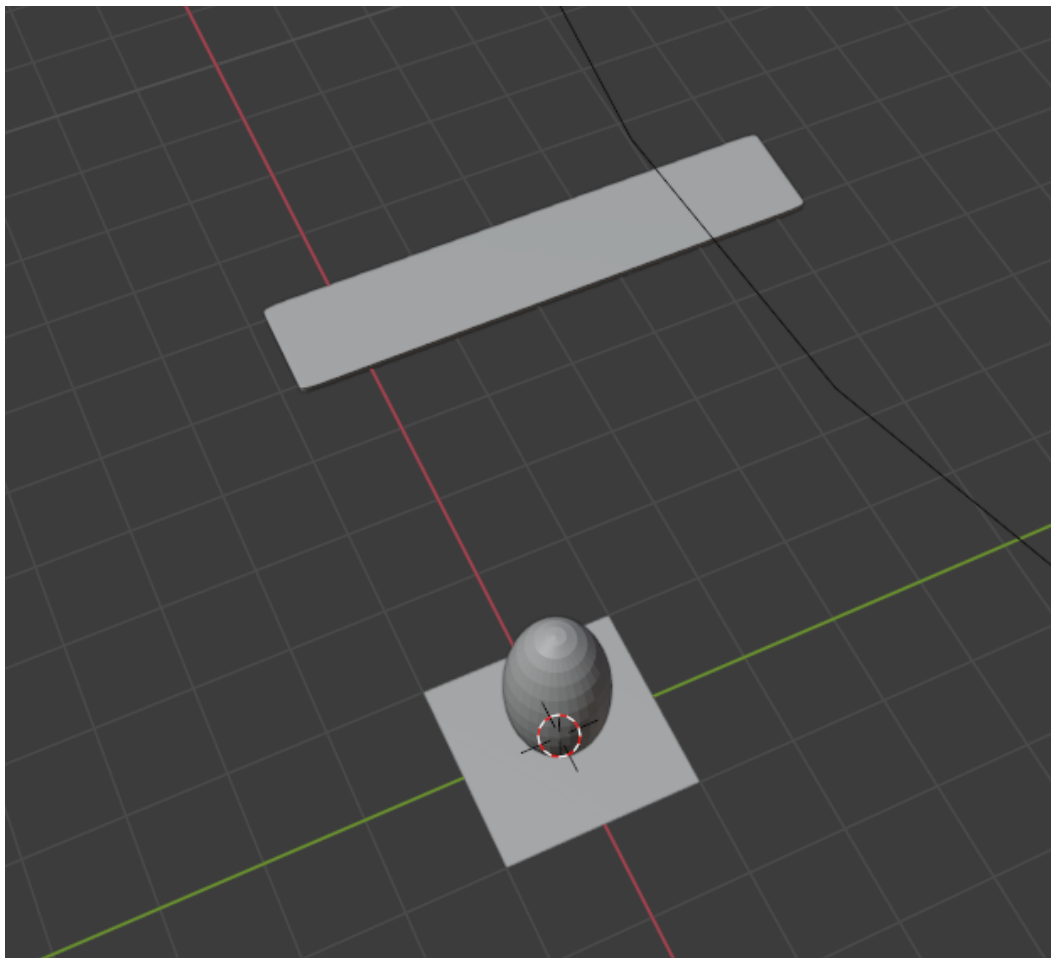


Figura 2.13: Elementos básicos de la escena generados en el Script.

Para que todos los objetos de nuestro entorno colisionen, será necesario añadir el modificador de colisión de Blender a todos los elementos de la escena, en este caso hemos creado 3 objetos por lo que, lo hemos incluido en cada uno de estos, podemos ver un extracto de código con un ejemplo de como se añade el modificador de colisión a un objeto [2.1](#).

```
bpy.data.objects['Cube'].modifiers.new("ColisionBufanda", "COLLISION")
```

Cuadro 2.1: Código que añade al objeto indicado el modificador de colisión.

2.3.4.1. Comportamiento de tela

Una vez creada la prenda habrá que asignarle el modificador de tejido que tiene Blender, para que esta adquiera las propiedades de la tela. En el extracto de código [2.2](#) vemos como se añadiría dicho modificador a nuestro objeto.

```
bufanda.modifiers.new("Ropa", "CLOTH")
```

Cuadro 2.2: Código que añade al objeto indicado el modificador de tejido.

Pero con añadir únicamente este modificador no es suficiente, para que la prenda se adapte mejor a todo tipo de superficies habrá que aplicar una serie de técnicas de suavizado a nuestro objeto.

2.3.4.1.1. Suavizado de la prenda

En primer lugar habrá que dividir las caras de la bufanda en caras mas pequeñas, dándole un mayor numero de caras para que a la hora de colisionar o caer sobre algo se comporte de manera mas realista. Para dividir las caras en nuestro código, hay que poner el edit mode para poder subdividir las caras del objeto ya que esto se hace en el modo edición. En este caso he decidido subdividir 4 veces las caras ya que era suficiente para conseguir un comportamiento realista. Tras aplicar estas subdivisiones tendremos que volver al object mode para poder seguir trabajando en el suavizado del objeto. El siguiente paso para suavizar sera aplicar la opción shade smooth que se usa para dar una apariencia mas curvada y no tan rígida. Para aplicarlo en nuestro script, tendremos que recorrer todo el polígono e ir activando la opción de suavizado (smooth). Por último también he añadido el modificador subdivision surface, cuya función es suavizar la diferencia entre todas las caras del polígono.

En la Figura 2.14 podremos ver la diferencia entre una prenda con el suavizado aplicado y otra sin aplicar.

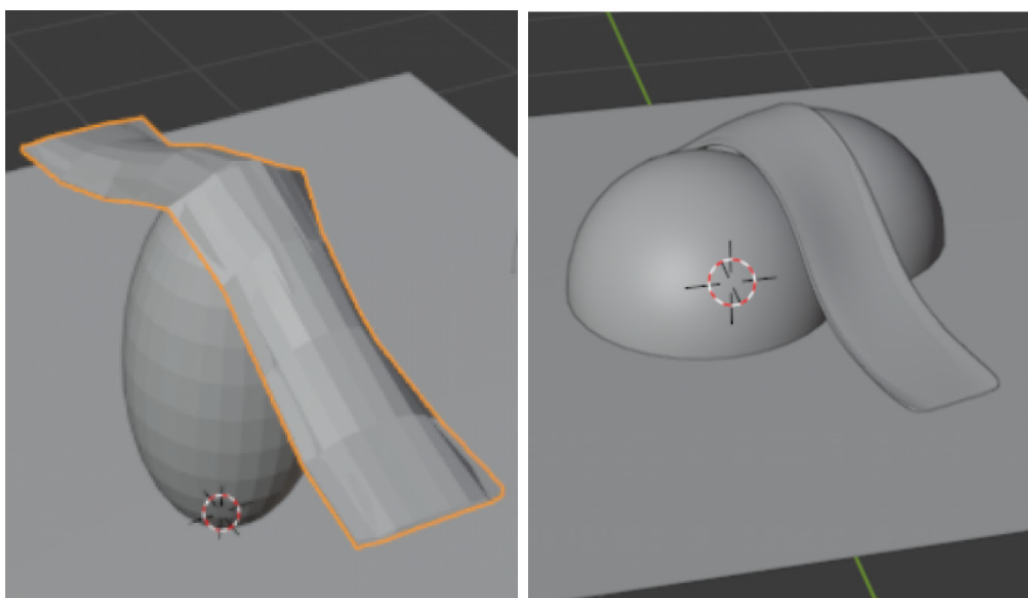


Figura 2.14: Izquierda objeto con el modificador de tejido pero sin aplicar nada de suavizado. Derecha objeto con los modificadores y opciones de suavizado aplicadas.

2.3.4.2. Generación del pelo asociado a la prenda

Para poder incluir pelo a nuestro complemento es necesario definir un grupo de vértices, que irá ligado al sistema de partículas (flecos de la bufanda) con la creación del grupo de vértices lo que logramos es situar nuestro pelo en el lugar que deseemos, en este caso, los extremos de la prenda. Si el grupo de vértices no estuviera definido, el sistema de partículas estaría presente en todo el objeto. Para poder conocer los vértices deseados para después incluirlos en el código, deberemos pre-seleccionarlos antes, activando la opción de enumerar vértices en Blender. En la Figura 2.15 se puede apreciar como se verían los vértices seleccionados de nuestro objeto (de un extremo), estos vértices deberemos de anotarlos para después poder incluirlos en el nuestro script y posteriormente asociar el grupo de vértices a el sistema de partículas.

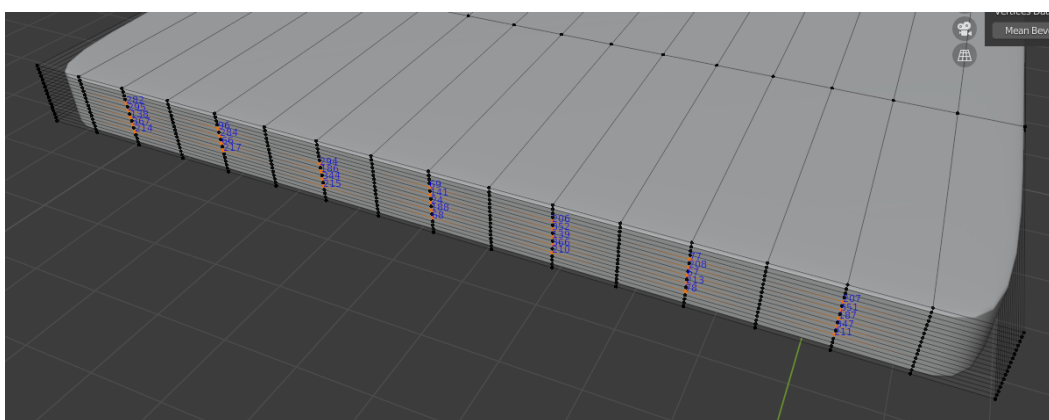


Figura 2.15: Observamos los vértices seleccionados de un extremo de la bufanda, que serán añadidos a el grupo de vértices.

El sistema de partículas tendrá todo lo necesario para la simulación del pelo, densidad, longitud, masa y otros parámetros que hemos adaptado para que tenga un comportamiento lo mas real posible. En la imagen 2.16 observamos todos los parámetros que se ha ajustado después de un gran numero de pruebas, para que el pelo se comporte de la manera mas real posible.

```
#Sistema de particulas que simulara los flecos de la bufanda(pelos)
psys = bufanda.modifiers.new("hair", 'PARTICLE_SYSTEM').particle_system
psys.settings.type = 'HAIR'
#Configuracion de los parametros del eplo
bufanda.show_instancer_for_render = False
psys.settings.use_strand_primitive = True
psys.settings.radius_scale = 15
psys.settings.hair_step = 3
psys.settings.count = 1200
psys.settings.hair_length = 0.3
bufanda.particle_systems["hair"].use_hair_dynamics = True

#Calidad de el pelo
bufanda.particle_systems["hair"].cloth.settings.quality = 2

#Masa del pelo
bufanda.particle_systems["hair"].cloth.settings.mass = 5

#Asigacion de donde crecera el pelo
bufanda.particle_systems["hair"].vertex_group_density = "Pelo"
```

Figura 2.16: Imagen con todos los parámetros y configuración del sistema de partículas que simula pelo en nuestro complemento.

2.3.4.3. Inserción de diferentes tipos de materiales/tejidos

En esta ultima tarea de la iteración 2, se han incluido diferentes tipos de materiales, con el fin de que al ejecutar el plug-in, el usuario pueda seleccionar el material que mas se ajuste a sus deseos o pueda probar todos para ver el comportamiento de cada uno. Para nuestro proyecto se han incluido los materiales que trae Blender por defecto que son: Algodón, seda, denim, goma y piel. Todos estos tienen sus parámetros específicos y al seleccionar uno u otro los parámetros de nuestro material cambian en función del material seleccionado, en la imagen [2.17](#) un ejemplo de dos materiales con sus respectivas características.

```
#Denim
if mytool.my_enum == 'DEN':
    bufanda.modifiers["Ropa"].settings.quality = 12
    bufanda.modifiers["Ropa"].settings.mass = 1
    bufanda.modifiers["Ropa"].settings.tension_stiffness = 40
    bufanda.modifiers["Ropa"].settings.compression_stiffness = 40
    bufanda.modifiers["Ropa"].settings.shear_stiffness = 40
    bufanda.modifiers["Ropa"].settings.bending_stiffness = 10
    bufanda.modifiers["Ropa"].settings.tension_damping = 25
    bufanda.modifiers["Ropa"].settings.compression_damping = 25
    bufanda.modifiers["Ropa"].settings.shear_damping = 25
    bufanda.modifiers["Ropa"].settings.air_damping = 1

#Leather
if mytool.my_enum == 'LEA':
    bufanda.modifiers["Ropa"].settings.quality = 15
    bufanda.modifiers["Ropa"].settings.mass = 0.4
    bufanda.modifiers["Ropa"].settings.tension_stiffness = 80
    bufanda.modifiers["Ropa"].settings.compression_stiffness = 80
    bufanda.modifiers["Ropa"].settings.shear_stiffness = 80
    bufanda.modifiers["Ropa"].settings.bending_stiffness = 150
    bufanda.modifiers["Ropa"].settings.tension_damping = 25
    bufanda.modifiers["Ropa"].settings.compression_damping = 25
    bufanda.modifiers["Ropa"].settings.shear_damping = 25
    bufanda.modifiers["Ropa"].settings.air_damping = 1
```

Figura 2.17: Captura del código donde se puede observar el ejemplo de dos materiales diferentes Denim y piel

2.3.5. Resultados y pruebas

2.3.5.1. Entregables

En primer lugar obtuvimos este resultado [2.18](#) donde la prenda se incrustaba en el objeto y los pelos se comportaban de una manera muy extraña de manera muy poco realista. En la Figura [2.18](#) si clicas en la imagen te lleva a un vídeo donde se puede ver mejor este resultado.

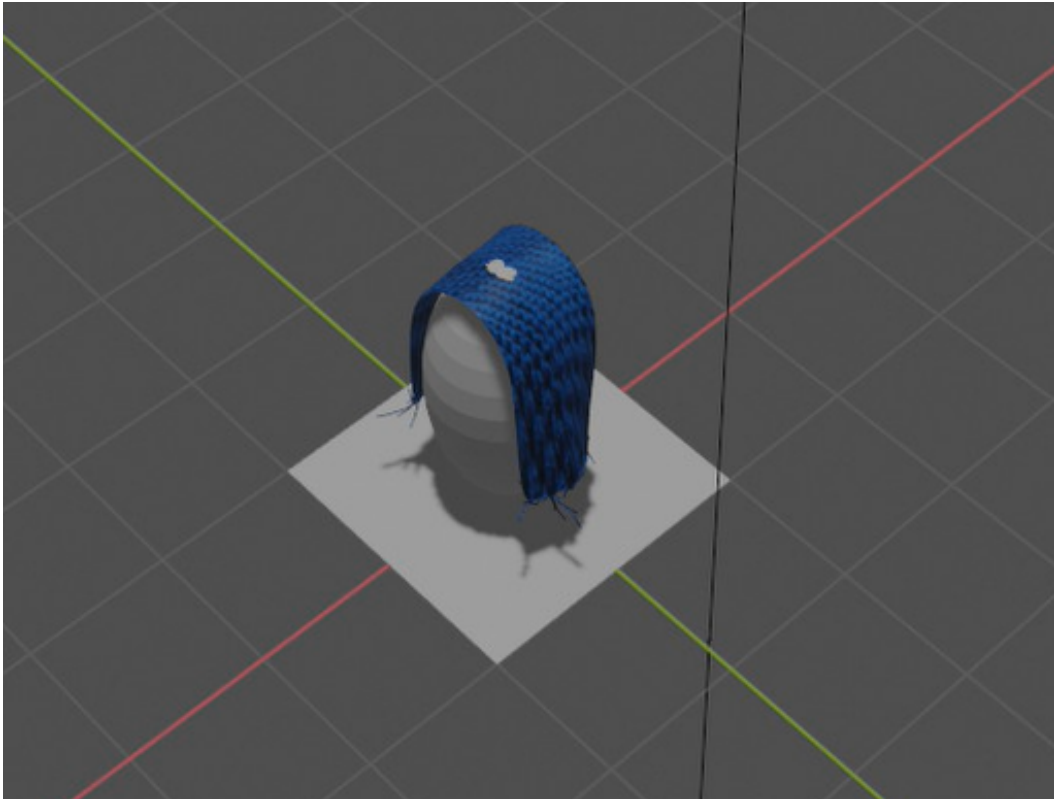


Figura 2.18: Primer resultado en la iteración 2 donde la prenda se incrusta en el objeto en el que cae y los pelos de la prenda no se comportan de manera realista, si haces clic en la imagen te llevará a un vídeo de este ejemplo

Tras un gran numero de pruebas y ajustes conseguimos un comportamiento a priori bastante real, con un buen comportamiento del tejido y del pelo. Algunos de los parámetros que hemos ajustado han sido: calidad de colisión, distancia de colisión, calidad del pelo y masa del pelo. Todos estos ajustes están presentes en nuestro código. En la Figura 2.19 vemos este resultado y si clicas sobre la imagen te redirigirá a un vídeo donde podemos observar mejor este resultado.

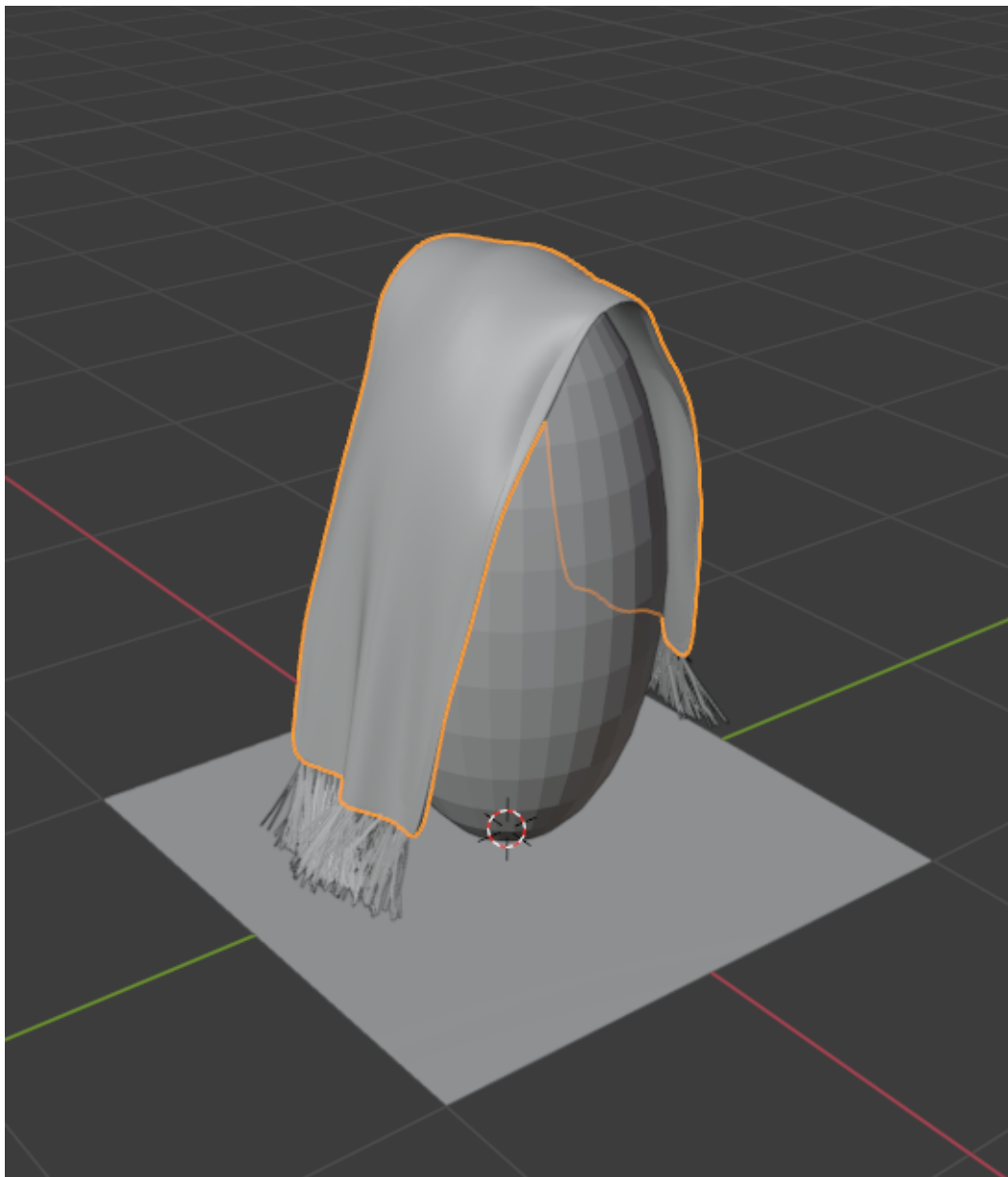


Figura 2.19: Resultado final de la iteración 2, donde la prenda tiene un comportamiento adecuado y el pelo asociado se comporta con normalidad, al hacer clic sobre la imagen te redirigirá a un vídeo con este resultado.

Por ultimo se han incluido unos vídeos en una lista de reproducción [5], en la que se puede ver el comportamiento de la prenda, con los diferentes tipos de materiales, excepto el algodón que ya esta incluido en la Figura anterior 2.19.

2.3.6. Dificultades

Debido a que hay poco material en internet para trasladar a código algunas de las utilidades de Blender, surgieron una serie de dificultades que fueron estas:

2.3.6.0.1. Selección de los vértices

Fue un problema, ya que primero hay que preseleccionarlos para incluirlos después en nuestro grupo, lo que pasó fue que al principio estaba seleccionando las caras del objeto en lugar de los vértices y no situaba el pelo correctamente, ya que había que seleccionar los vértices en lugar de las caras.

2.3.6.0.2. Prenda que atraviesa objeto

Como podemos ver en el primer resultado [2.18](#), la prenda atraviesa el objeto en la simulación. En un principio no sabía el por que de esto, pero tras probar tocando los parámetros de las colisiones, el parámetro que no estaba bien ajustado era el distancia de colisión, este parámetro lo que hace es ajustar la distancia que otro objeto debe alcanzar a la tela, para que la tela sea repelida, si es un valor pequeño puede dar errores, que es lo pasaba en nuestro caso. Los valores grandes darán resultados irreales si son demasiado grandes y pueden ser lentos, lo mejor es encontrar un buen valor intermedio, que fue finalmente lo que insertamos.

2.3.6.0.3. Comportamiento del pelo

Tal y como observamos también en el primer resultado [2.18](#) el comportamiento del pelo, no es el adecuado ya que, dista mucho de un comportamiento real, estos no se comportan de manera adecuada al caer sobre el objeto, para solucionar esto hemos ajustado los parámetros de nuestro sistema de partículas. Los que hemos ajustado han sido el de Mass (peso) que aporta al pelo un mayor peso y el de calidad que define con que calidad quieres que tenga el pelo en la simulación.

2.4. Iteración 3: Personalización del complemento

Durante esta iteración el objetivo será poder aplicar una imagen a nuestro complemento para complementar la simulación de una manera mas completa y real. También es una buena opción por si el usuario desea ver como quedaría la prenda con diferentes estilos/colores.

2.4.1. Análisis de requisitos funcionales y no funcionales

2.4.1.1. Funcionales

- La herramienta debe de tener la opción de que el usuario pueda añadir una textura a la prenda.

- Nuestra herramienta será capaz de trabajar con imágenes de color, normales, mapeados de alturas, incluso imágenes realizadas por el usuario.
- Garantizar una interfaz sencilla e intuitiva al usuario para añadir una textura.

2.4.1.2. No funcionales

- La aplicación no necesita conexión a internet.
- La herramienta se podrá usar en cualquier sistema operativo.

2.4.2. Diseño

2.4.2.1. Interfaz

Hemos modificado nuestro panel de materiales de la iteración anterior, para que a la hora de pulsar el botón de añadir textura aparezca una ventana pequeña con un Input de texto, en el que el usuario debe escribir el nombre de la textura seleccionada, una vez introducida esta nueva ventana tendrá un botón de para confirmar lo introducido, al pulsarlo se aplicara el material seleccionado y la textura escogida. En la Figura 2.10 podremos ver el panel donde se aprecia la lista desplegable(iteración anterior) y el botón para añadir la textura y en la Figura 2.4.2.1 podremos apreciar la ventana donde aparece el Input para introducir la imagen seleccionada y un botón para confirmar.

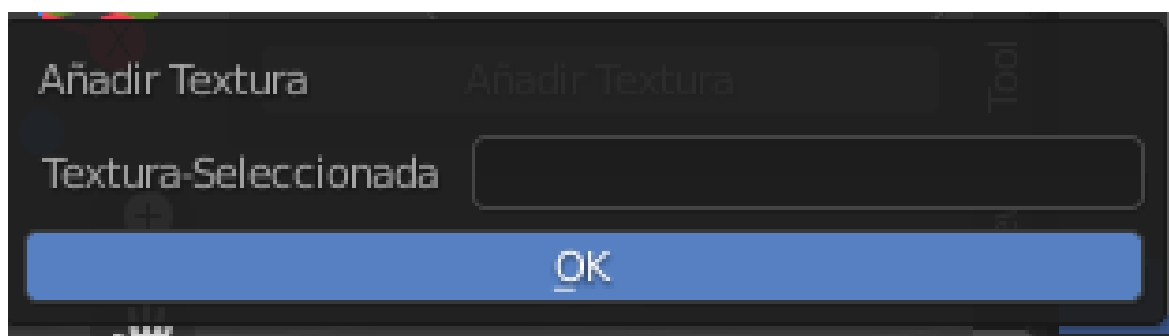


Figura 2.20: Ventana que aparece al pulsar el botón de añadir textura, compuesta de un input para introducir el nombre de la textura deseada y un botón para confirmar.

2.4.3. Historia de usuario

- Como usuario quiero poder añadir una textura a mi prenda, para ver como quedaría esta con diferentes tipos de imágenes y texturas.
- Como usuario quiero poder sacar una foto con mi dispositivo móvil y aplicarla a mi complemento, para observar como quedaría una imagen de un tejido que tengo en persona en la prenda.

2.4.4. Implementación

Tras asignar directamente una foto a la prenda vi que esto no era suficiente por lo que empecé a investigar sobre el uso de editar materiales y usar materiales en Blender y dí con la función de shader editor, que se utiliza para editar los materiales que se usan en el renderizado, se define por un árbol de nodos que en nuestro caso habrá que aplicarlo a código.

2.4.4.1. Bump mapping

Para dar mas realismo a nuestro material usaremos la técnica de Bump mapping que consiste en: Simular detalles en las superficies como golpes, bultos, surcos o rayones, gracias al uso de la luz. Para aplicar esta técnica son necesarias la imagen Normal (Normal Map) y el mapeado de alturas (Height Map). Ambas imágenes contienen datos que representan un detalle aparente en la superficie. pero cada uno almacena estos datos de una manera diferente. En la Figura 2.21 podemos ver las tres imágenes que se aplicaran a nuestro objeto. Podemos observar la imagen de Color que será la que aplicará el color a nuestro material, el Height map (mapeado de alturas), que es la textura negra y blanca, dónde cada píxel representa la cantidad que ese punto en la superficie debería ser elevado. Cuando más blanco sea el color del píxel, más elevada aparecerá el área. Por ultimo podemos ver un Normal Map que es una textura RGB, dónde cada píxel almacena una dirección, estas direcciones se denominan normales. Estas texturas tienden a tener un tinte morado-azul, debido a la manera que el vector es almacenado en valores RGB.

En nuestro modelo he decidido incluir ambos tipos de imágenes para que la textura sea mas realista.



Figura 2.21: Ejemplo de las imágenes necesarias para poder aplicar una textura correctamente. Izq:Imagen Color, Centro:Height, Dcha:Normal

2.4.4.2. Shader editor

En la Figura 2.22 podemos ver el shader editor, con todos los nodos necesarios para aplicar correctamente cualquier textura.

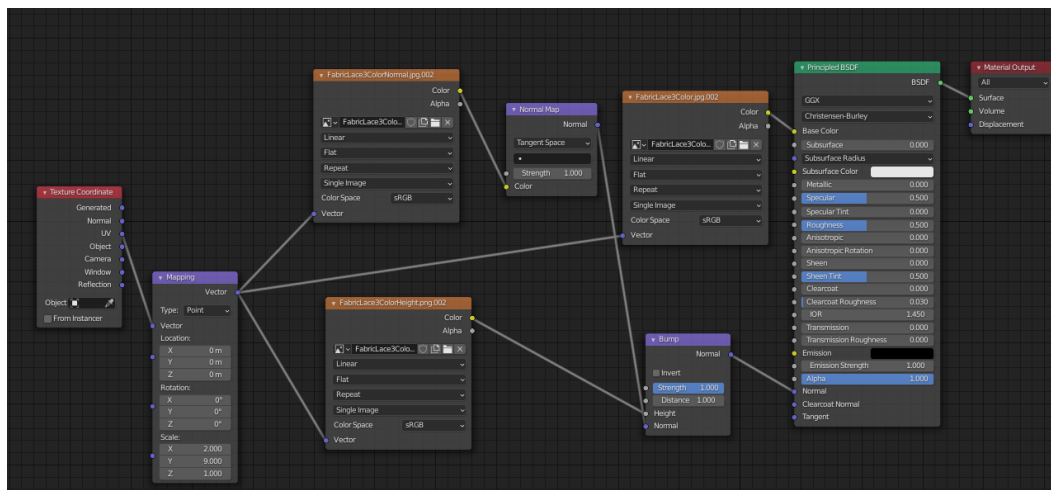


Figura 2.22: shader editor con todos los nodos necesarios para aplicar nuestra textura.

Ahora pasaremos a ver algunos de estos nodos mas importantes y su función.

2.4.4.2.1. Nodo principled BSDF

Es el nodo principal que está compuesto por una serie de slots a los que podemos añadir texturas/imágenes que pueden ser por ejemplo: Para el color, occlusion, normal, height... Este nodo será al que se le atribuirá todo lo relacionado con la textura e irá unido al objeto en concreto. En la Figura 2.23 podemos ver este nodo.

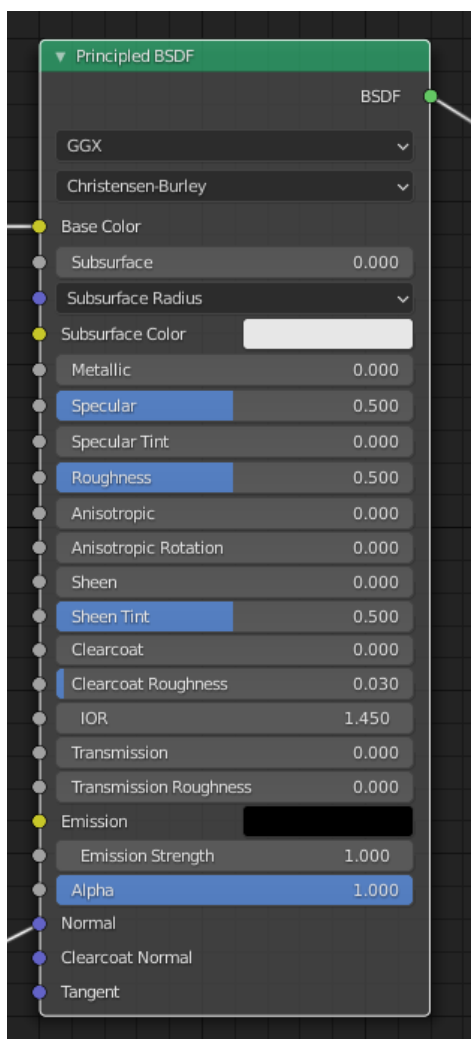


Figura 2.23: Nodo Principled BSDF

2.4.4.2.2. Nodo normal map

El nodo Normal Map genera una normal a partir de una imagen Normal Map. Se encadena con un nodo de textura de imagen en la entrada de color, para especificar la imagen del mapa normal. En la Figura 2.24 podemos observar este modo en el shader editor. La entrada strength sera la fuerza del efecto de mapeo normal.

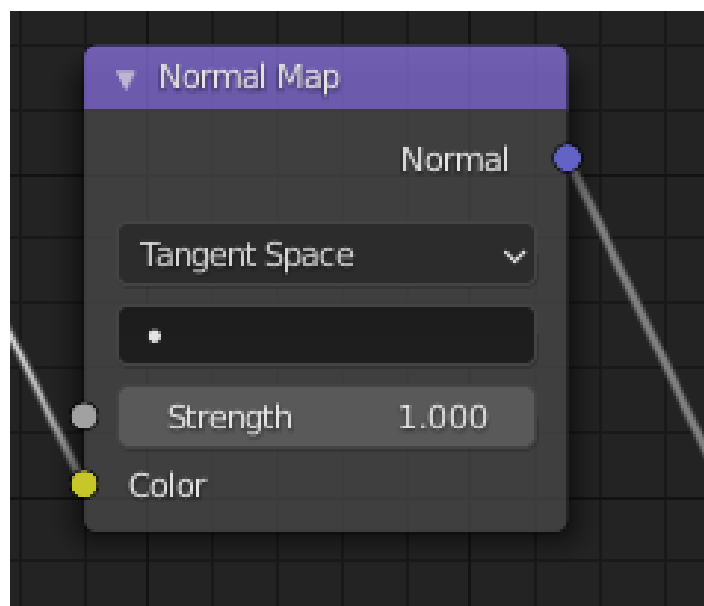


Figura 2.24: Nodo Normal Map.

2.4.4.2.3. Nodo bump

El nodo Bump genera una normal a partir de una textura de altura (Height Map). Se encadena con un nodo de textura de imagen (Height Map) en la entrada de height y con el nodo Normal descrito. La entrada strength será la fuerza del efecto de mapeo de relieve y la entrada distance será un multiplicador del valor de la altura para controlar la distancia total del mapeado de relieve. En la Figura 2.25 podemos una imagen de este nodo.

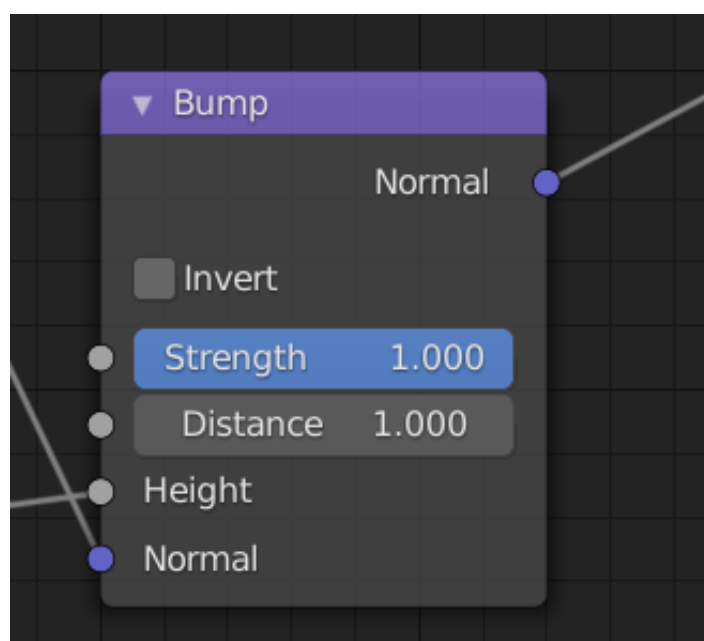


Figura 2.25: Nodo Bump.

2.4.4.2.4. Nodos imagen

Estos nodos son los usados para cargar cualquier imagen soportada por Blender. En nuestro caso hemos aplicado 3 nodos imagen, uno para cargar el color, otro para la imagen Normal y otro para el mapeado de alturas (Height Map). En la Figura 2.26 podemos ver todos los usados en nuestro proyecto.

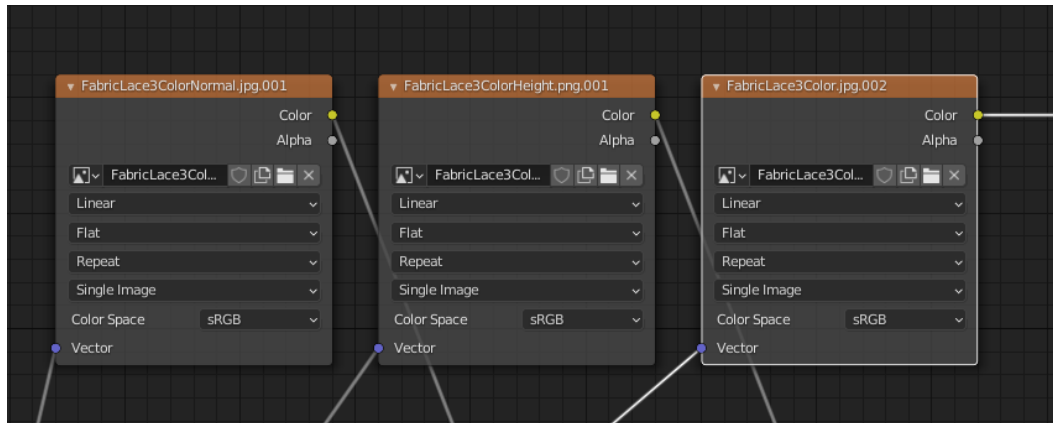


Figura 2.26: Nodos Imagen de nuestro proyecto, donde cargamos las imágenes de color, normal y Height Map.

2.4.4.2.5. Nodo mapping

El nodo de mapeo(Mapping) se utiliza para transformar una imagen o textura. Por ejemplo, puedes usarlo para mover, rotar o escalar texturas. En la Figura 2.27 se puede ver el nodo de mapeo, en esta ocasión nosotros hemos necesitado escalar en la X y en Y para que las texturas se adapten mejor a el objeto

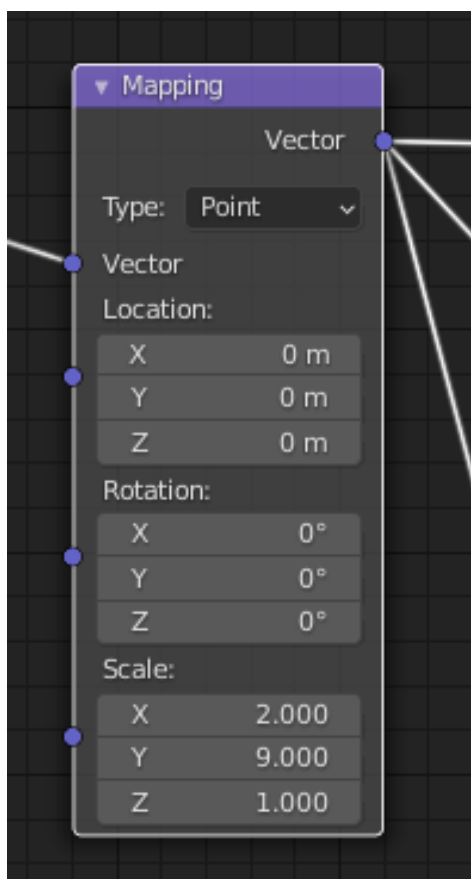


Figura 2.27: Mapping Node.

2.4.4.3. Aplicado de una imagen propia a nuestro complemento

Para poder aplicar nuestras propias texturas, por ejemplo de un tejido que tengamos en la vida real, deberemos sacar una foto del tejido en la que se pueda apreciar bien. En mi caso he sacado fotos con mi dispositivo móvil de un tejido que tenía por casa para hacer la prueba. En la Figura 2.28 podemos ver la imagen obtenida con el móvil.



Figura 2.28: Imagen obtenida desde un dispositivo móvil para aplicar a nuestro objeto.

Tras tener la imagen capturada deberemos obtener la imagen normal y el mapeado de alturas de esta.

2.4.4.3.1. Creación de el mapeado de alturas (Height Map)

Para la creación del mapa de alturas a partir de una imagen, deberemos tener instalado Gimp [3] y seguir estos pasos:

1. Abrir Gimp y la imagen de la textura de la que queremos generar el mapeado de alturas.
2. Seleccionamos Imagen - Modo - Escala de grises.
3. Ahora tendremos que seleccionar Colores - Niveles para que aparezca el cuadro de diálogo del histograma. Nos aparecerá una ventana con el histograma que debemos ajustar. En los niveles de entrada el nivel de la izquierda habrá que ajustarlo donde el histograma empiece a tener altura, el nivel de la derecha habrá que ajustarlo donde el histograma baje los valores de entrada, el nivel del centro se debe ajustar hasta obtener un resultado adecuado. En la Figura 2.29 vemos la ventana de niveles que tenemos que ajustar.

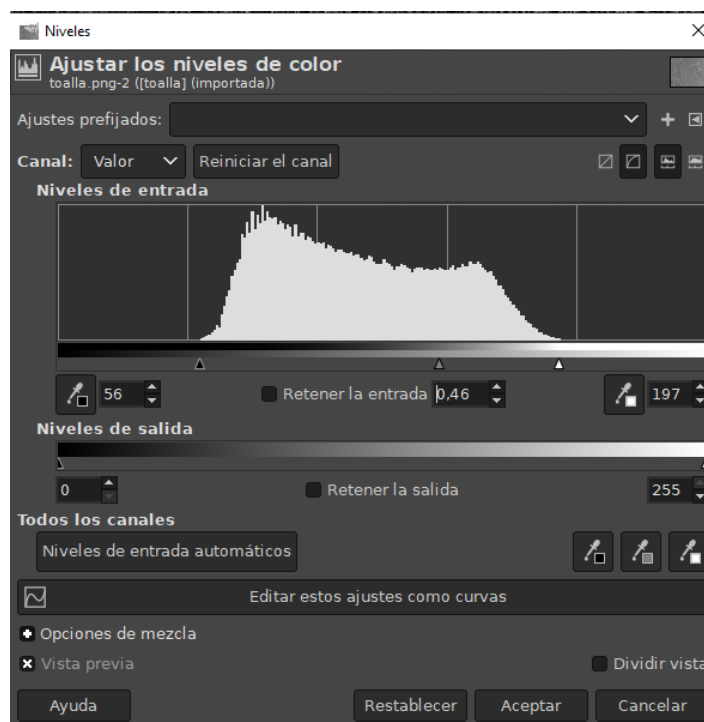


Figura 2.29: Ventana de niveles que hay que ajustar.

4. Una vez todo ajustado podremos exportar la imagen en Archivo - Exportar como y guardamos el resultado.
5. En la Figura 2.30 observamos la imagen resultante nuestro mapeado de alturas.

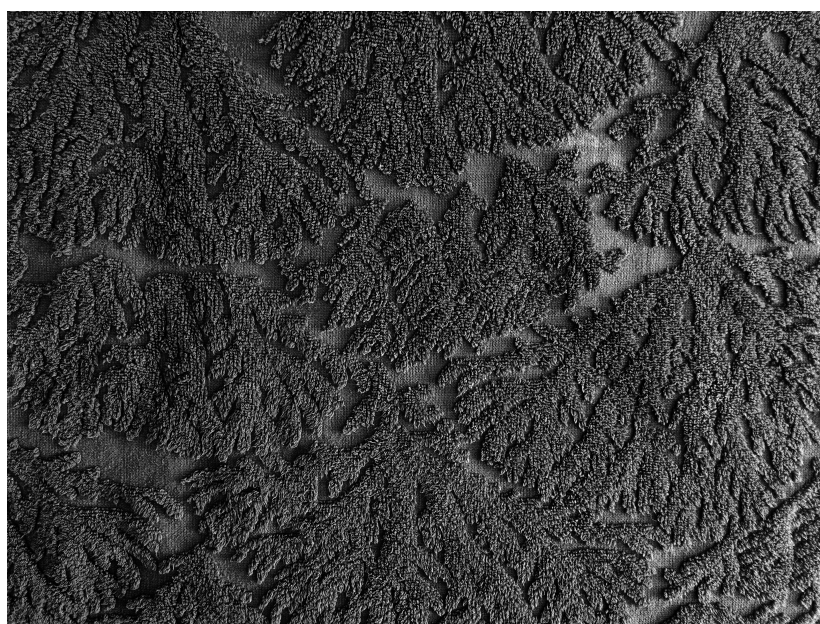


Figura 2.30: Mapeado de alturas calculado de nuestra imagen.

2.4.4.3.2. Creación de la imagen normal

Para la creación de la imagen normal a partir de la nuestra tendremos que seguir estos pasos:

1. Visitamos esta pagina [4] en la que te crea automáticamente una imagen normal a partir de la nuestra.
2. Subimos nuestra imagen a la página y desactivamos Displacement, Ao y Specular. En la Figura 2.31 observamos la pagina con la configuración correcta.

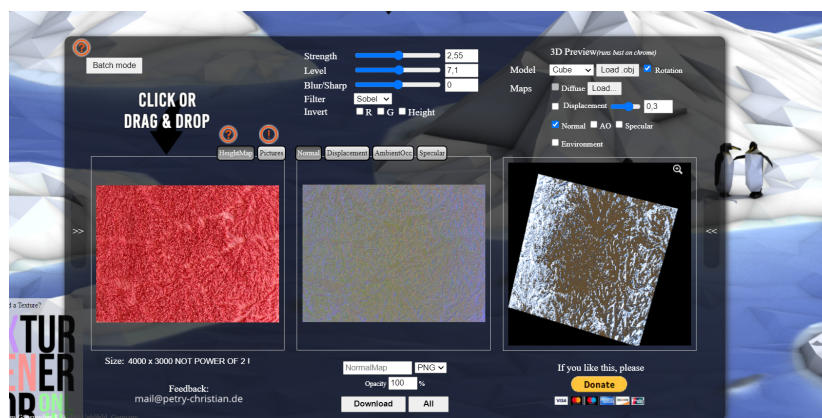


Figura 2.31: Imagen de la pagina para calcular la imagen normal.

3. Resultado final de la imagen Normal 2.32

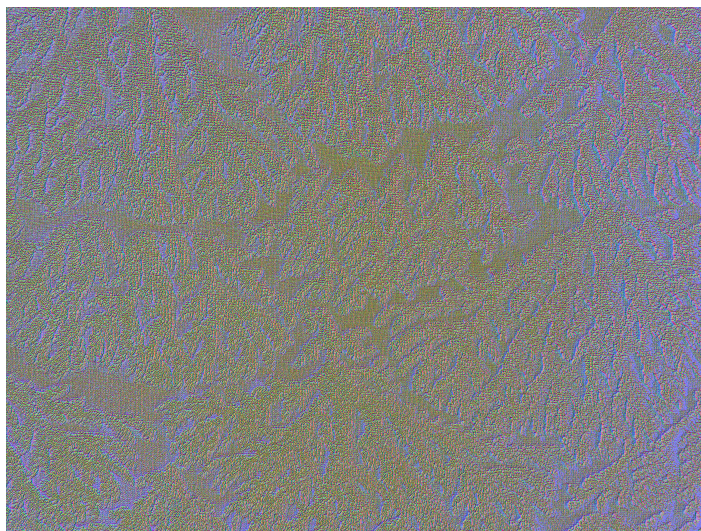


Figura 2.32: Imagen normal de nuestra imagen.

2.4.5. Resultados y pruebas

2.4.5.1. Entregables

En primer lugar mostrar la diferencia entre aplicar la técnica de bump mapping y no aplicarla, en la Figura 2.4.5.1 se puede apreciar esta diferencia.

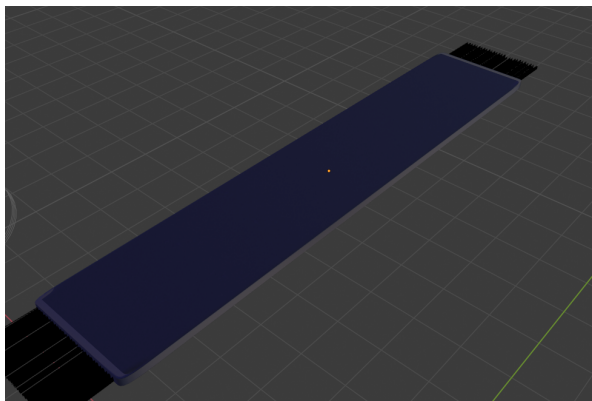


Figura 2.33: Textura sin la técnica de Bump Mapping.

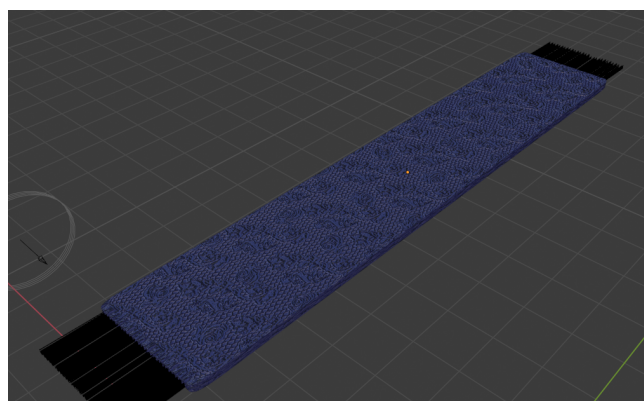


Figura 2.34: Textura con la técnica de Bump Mapping.

Comentar también la diferencia entre ajustar la textura gracias al nodo mapping y no ajustarla, gracias a esto hemos conseguido una mejor adaptación de la textura a nuestro objeto. En la Figura 2.4.5.1 podemos ver esta diferencia.

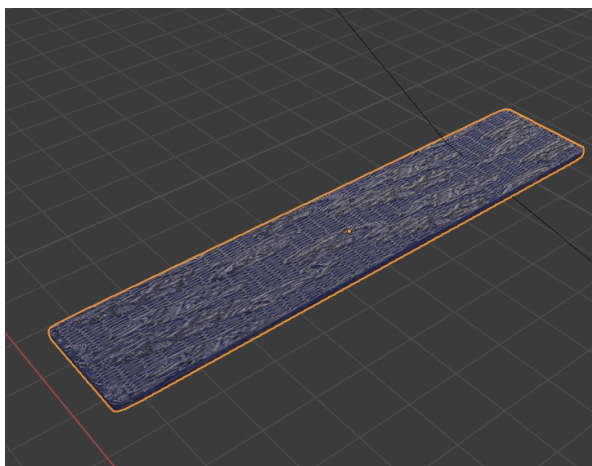


Figura 2.35: Textura sin ajustar usando el nodo mapping.

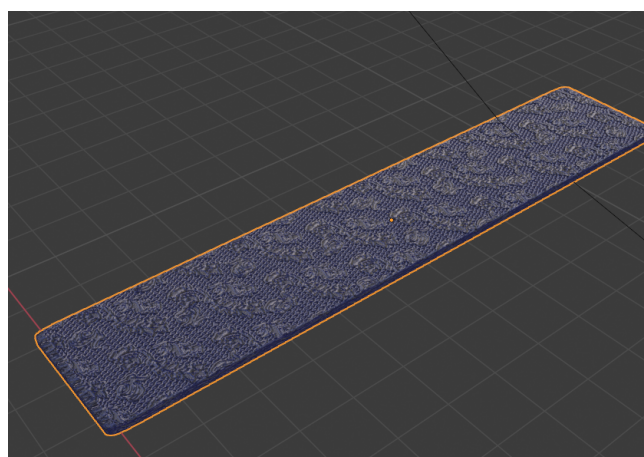


Figura 2.36: Textura ajustada gracias al nodo mapping.

Mostrar un ejemplo también de nuestro complemento con una imagen sacada con nuestro dispositivo móvil 2.37.

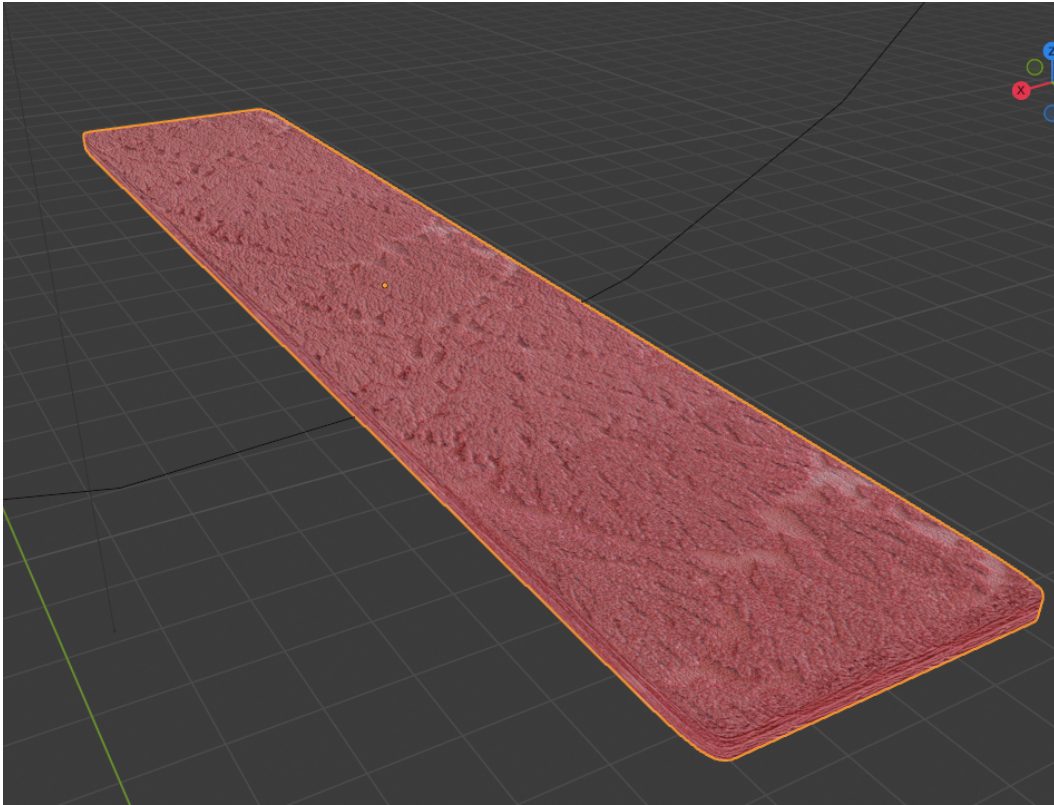


Figura 2.37: Imagen del complemento con una textura creada a partir de una foto con un dispositivo móvil.

Por ultimo mostrar otros ejemplos de nuestro objeto con otros tipos de texturas [2.38](#) [2.39](#) [2.40](#) [2.41](#).

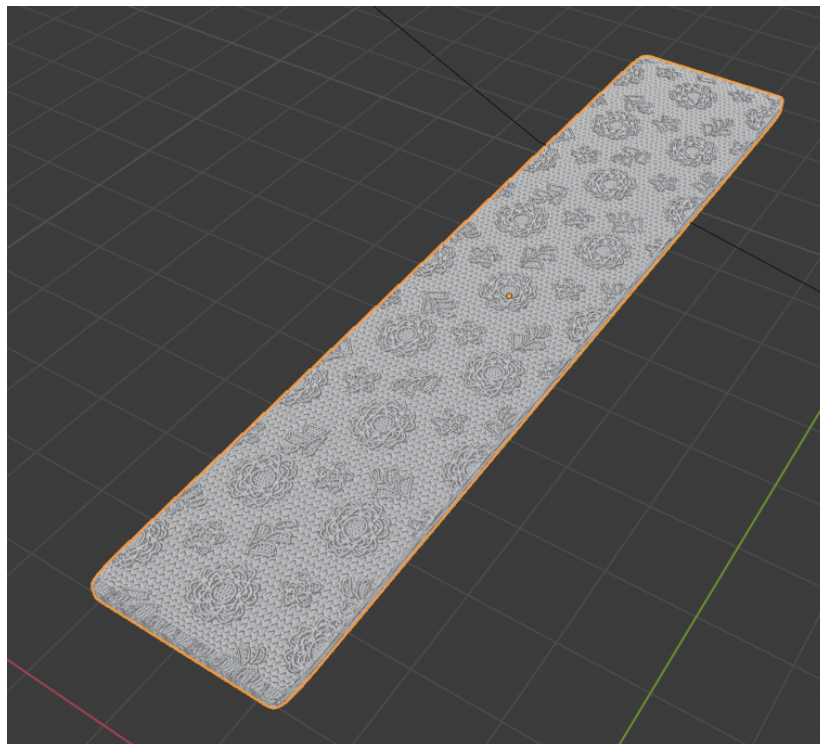


Figura 2.38: Ejemplo del complemento con una textura.

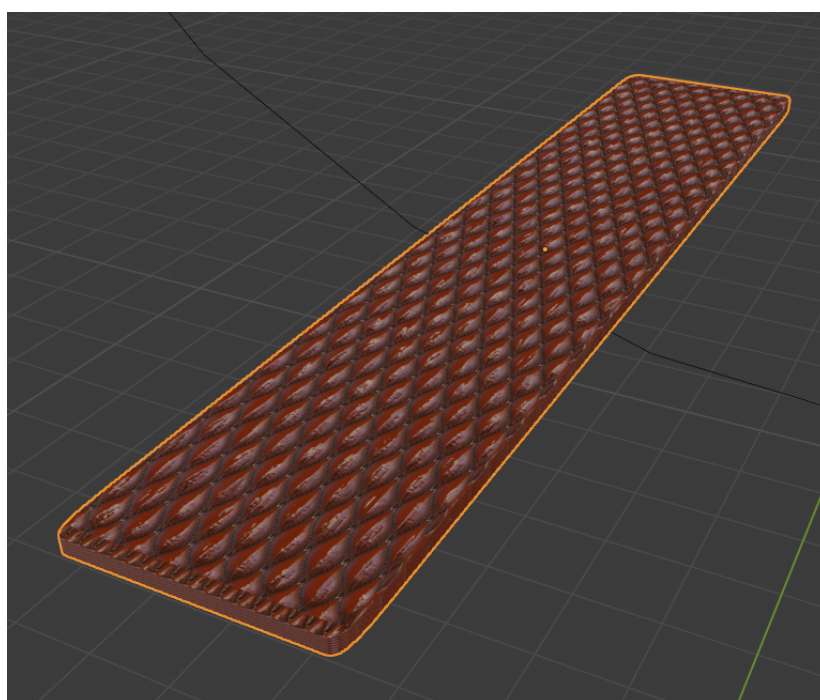


Figura 2.39: Ejemplo del complemento con una textura.

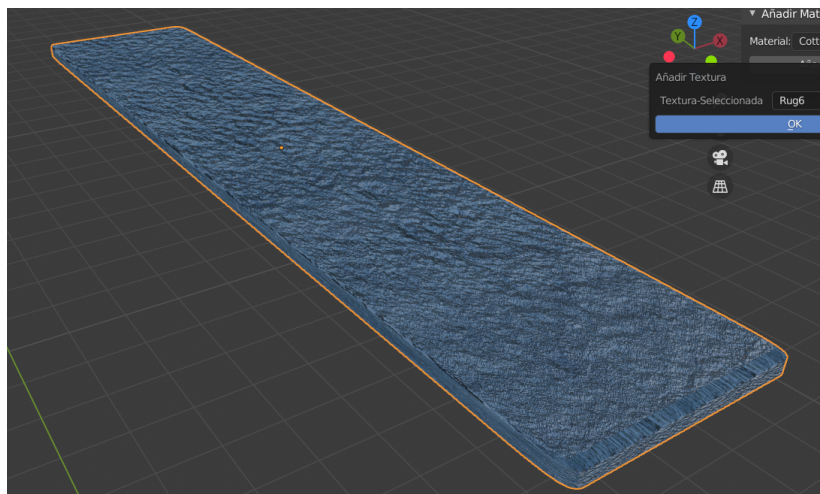


Figura 2.40: Ejemplo del complemento con una textura.

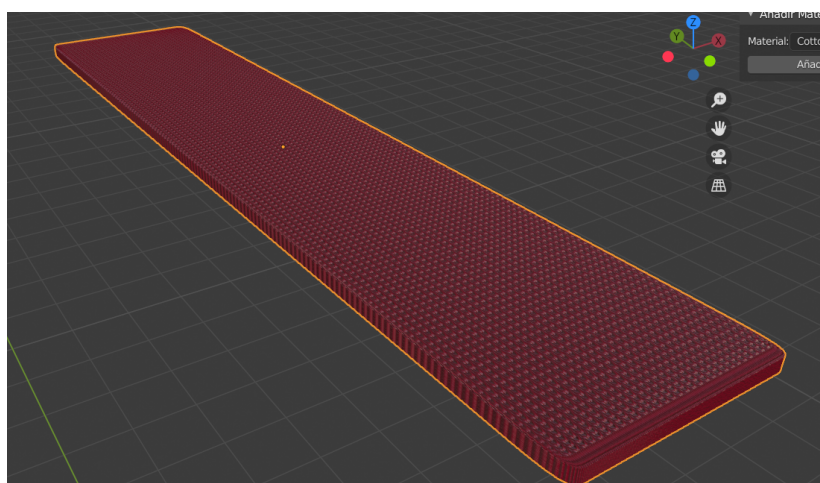


Figura 2.41: Ejemplo del complemento con una textura.

2.4.5.2. Dificultades

Durante esta iteración la mayor dificultad fue que tuve que descubrir que se podían ajustar las imágenes(texturas) en el nodo mapping, eso ayudó bastante ya que en las primeras pruebas no se apreciaban correctamente las imágenes en nuestro objeto.

Destacar también la técnica de bump mapping, que estaba prevista desde el inicio su uso pero, durante esta iteración se ha tenido que conocer más a fondo y quería hacerlo constar en el transcurso de esta.

2.5. Iteración 4: Realización de simulaciones e inclusión del viento en la escena

El objetivo durante esta última iteración será realizar simulaciones con nuestro complemento, modificando los elementos de la escena para poder apreciar nuestro modelo en diferentes situaciones y ámbitos, también incluiremos la opción de activar viento en la escena, una vez ejecutado el plug-in.

2.5.1. Análisis de requisitos funcionales y no funcionales

2.5.1.1. Funcionales

- El sistema debe permitir al usuario poder trasladar, rotar y escalar los objetos de la escena.
- El sistema permitirá al usuario poder borrar cualquier elemento de la escena.
- La herramienta permitirá al usuario hacer pruebas y simulaciones con los diferentes elementos y opciones de la escena.
- El sistema permitirá al usuario añadir elementos, modelos y animaciones a la escena.
- El sistema debe garantizar una interfaz sencilla e intuitiva al usuario, para añadir elementos, modelos y animaciones a la escena.

2.5.1.2. No funcionales

- La aplicación no necesita conexión a internet.
- La herramienta se podrá usar en cualquier sistema operativo.

2.5.2. Diseño

2.5.2.1. Interfaz

Hemos diseñado un último panel cuya función es la de activar o desactivar viento en la escena. El control para saber si el viento está activado se comprobará también con una variable baliza. Si pulsamos el botón y el viento no está activado la fuerza del viento se incrementará para que se aprecie en la escena. En caso de querer desactivarlo la fuerza será 0. En la Figura [2.42](#) podemos observar el panel para añadir o eliminar el viento de la escena, está compuesto por un botón para activar o desactivarlo.



Figura 2.42: Panel para la activar o desactivar el viento en la escena.

2.5.3. Historia de usuario

- Como usuario quiero tener la opción de añadir viento en la escena para tener un ámbito distinto.
- Como usuario quiero poder añadir añadir otros elementos y animaciones en la escena para poder observar el comportamiento de la prenda en situaciones diferentes.

2.5.4. Implementación

En primer lugar incluiremos la opción de añadir viento en la escena, el fin de esto, es que el usuario pueda comprobar el comportamiento de la prenda en una situación diferente. Durante la iteración además de hacer pruebas con el elemento del viento, se han incluido otro tipo de objetos para tener algo mas de variedad en las simulaciones, pero estas no están disponibles en nuestro plug-in, se tienen que añadir manualmente a la escena.

2.5.4.1. Añadir un individuo a la escena

Al modelar complementos de ropa, como es lógico tendremos que realizar pruebas con modelados de persona, por lo que tendremos que crear el modelo de una figura humana. Para la creación de este modelo accedemos a la pagina de Makehuman [7] y descargamos el programa. Éste nos permite generar el modelo de un individuo fácilmente.

En la Figura 2.43 podremos ver una imagen del programa donde se puede observar el gran numero de opciones que tiene para personalizar el modelo.

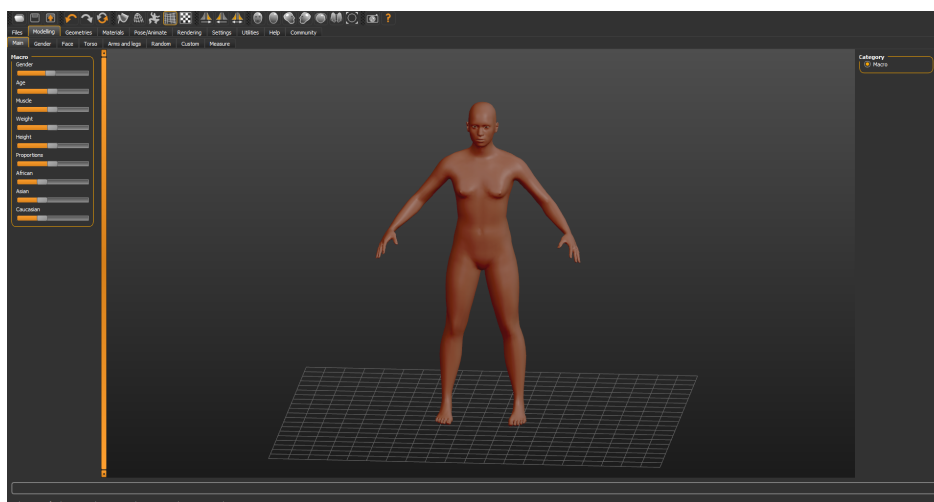


Figura 2.43: Imagen del programa Makehuman [7] que nos permite crear el modelo de una persona.

Para exportar el modelo sera necesario hacerlo en formato fbx para poder añadirlo a nuestra escena principal. En las Figura 2.44 podemos observar como se exportaría en el programa de Makehuman.

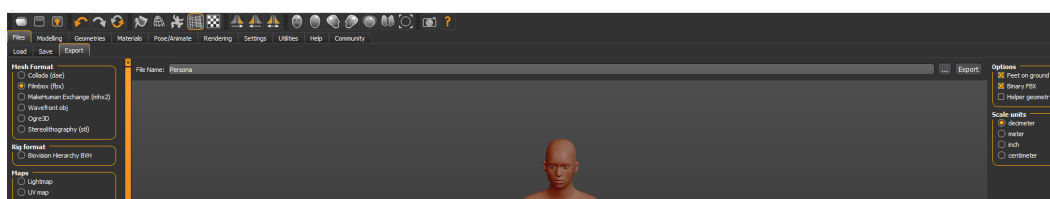


Figura 2.44: Exportar modelo desde el programa de Makehuman. [7]

Una vez tengamos el modelo incluido en nuestra escena sera necesario añadirle el modificador de colisión para que pueda interactuar con los modelos de la escena.

2.5.4.2. Animar nuestro modelo

En el caso que deseemos animar a nuestro individuo, para poder tener otra perspectiva del comportamiento de la prenda, hemos hecho uso de la herramienta Mixamo [6], que la podemos encontrar un gran numero de animaciones gratis, además tiene una opción en la que puedes subir tu propio modelo y poder añadirle una de las animaciones disponibles en la pagina.

2.5.4.2.1. Animar modelo en Mixamo

A continuación los pasos necesarios para animar nuestro modelo en Mixamo [6]

- Entrar en la página y registrarse ya que si no estás registrado, no podremos descargar animaciones.
- Una vez registrado, podemos acceder a el apartado de animaciones una vez ahí nos aparecerá un menú con varias opciones, una de ellas será la de subir nuestro propio personaje. En la Figura 2.45 podremos ver este menú.

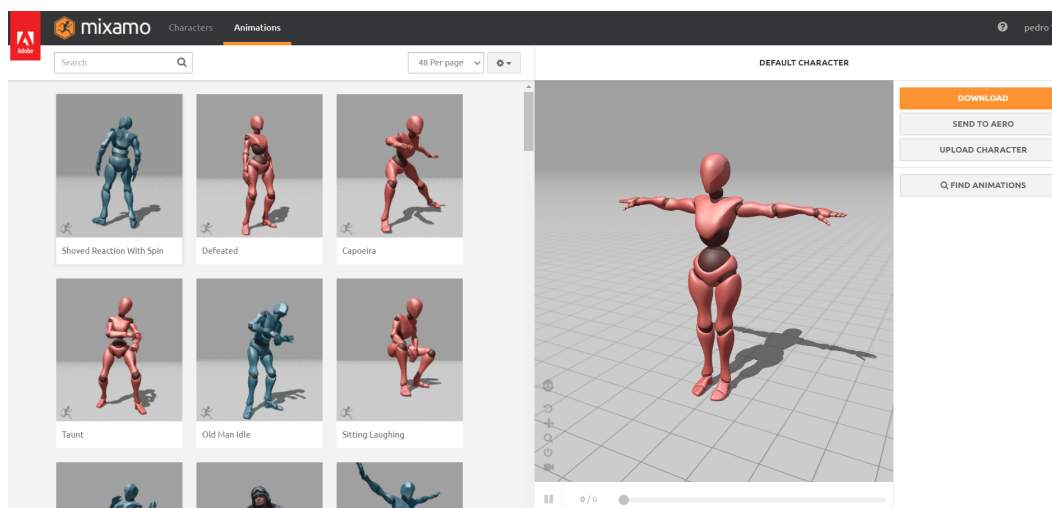


Figura 2.45: Menú de animaciones de Mixamo [6].

- Seleccionamos el personaje que queremos animar y lo cargamos. En la Figura 2.46 observamos la ventana que aparece y nos permite la previsualización de nuestro personaje.

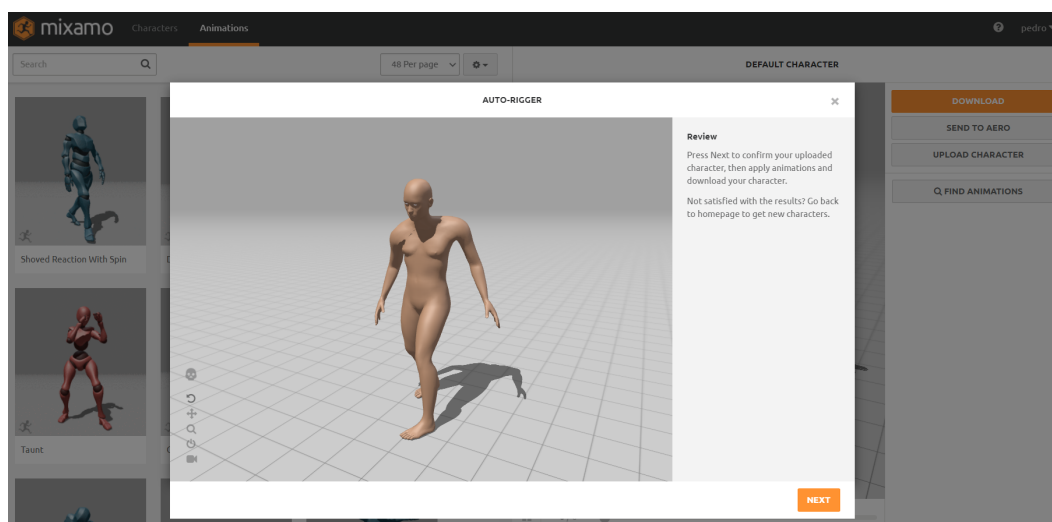


Figura 2.46: Ventana de previsualización que aparece tras cargar nuestro personaje en Mixamo [6]

- Cuando tengamos al personaje correctamente subido, podemos seleccionar cualquier animación y ver el comportamiento, en nuestro caso seleccionamos

una sencilla y simple de nuestro individuo andando, una vez decidida, la podremos descargar. En la Figura 2.47 observamos la ventana de descarga, tendremos que seleccionar el formato fbx.

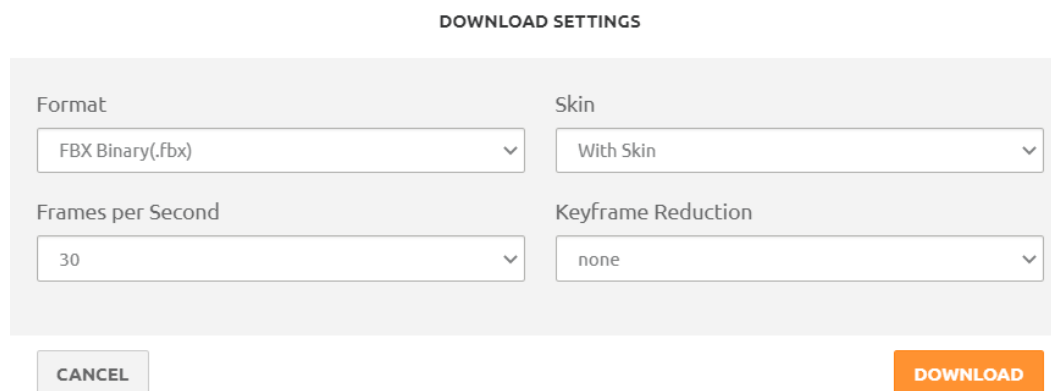


Figura 2.47: Ventana de descarga de Mixamo con las opciones disponibles.

2.5.5. Resultados y pruebas

2.5.5.1. Entregables

Simulación de una caída sobre los objetos que se generan en el script. Esta es la prueba mas sencilla ya que es la generada con los objetos que vienen por defecto en el script. La bufanda caerá sobre una especie de óvalo, tras esto si no consigue mantenerse en el óvalo caerá en el plano. En la Figura 2.5.5.1 se muestra la prenda cayendo sobre los objetos generados.

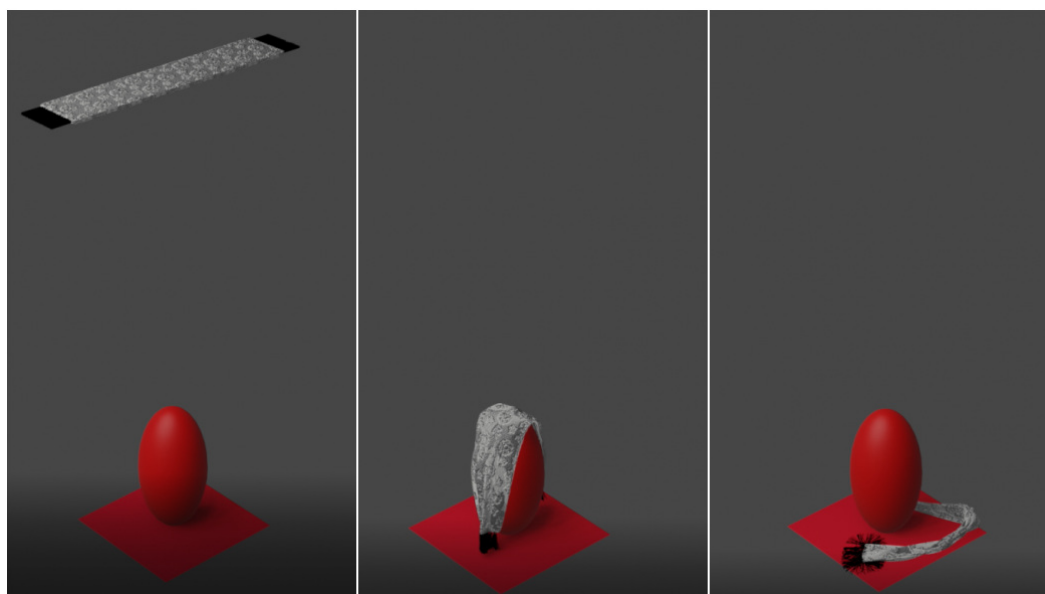


Figura 2.48: Imagen de la prueba 1 donde se ve la simulación de la caída de la bufanda sobre un ovalo y un plano, todo generado en el script.

Simulación ante la reacción de la bufanda con el viento. Aquí activaremos el viento desde el panel descrito anteriormente y podremos ver el comportamiento de el objeto cuando aparece esta novedad en la escena. En las imágenes de ejemplo de esta prueba el viento tendrá una fuerza de 6500 unidades. En la Figura 2.5.5.1 se muestra dicha simulación.

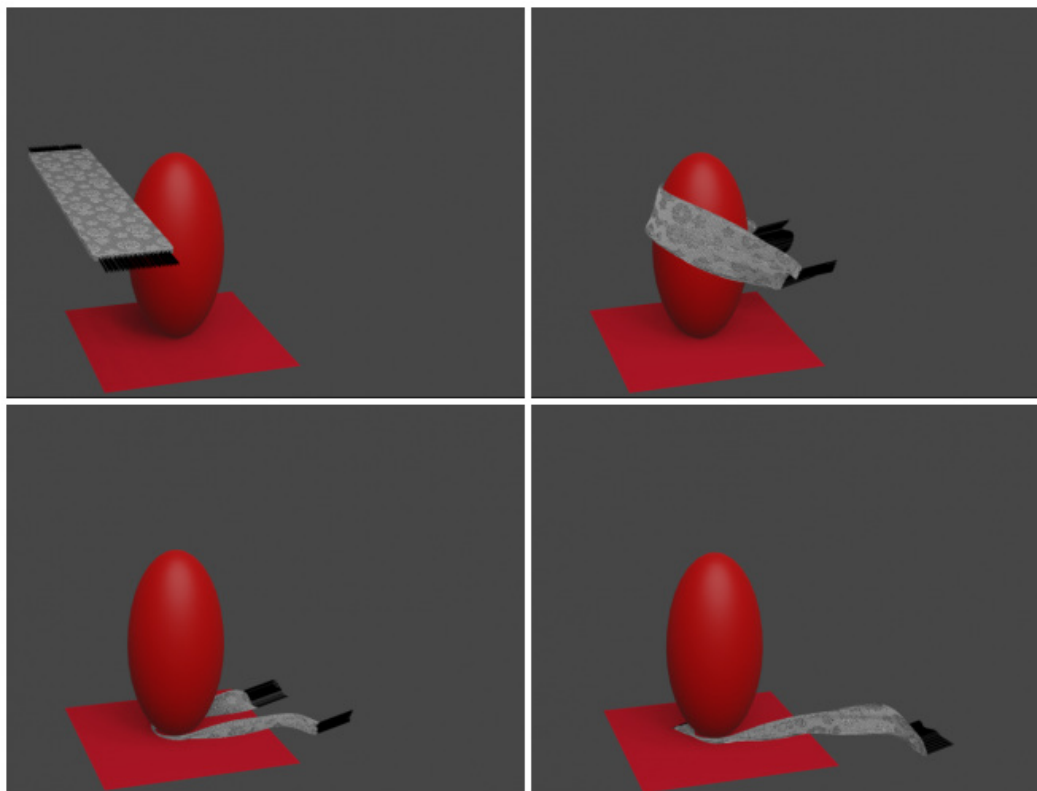


Figura 2.49: Imagen de la simulación de como se comporta la prenda al haber viento en la escena.

Simulación de una caída sobre un sujeto que esta quieto. Para esta prueba sera necesario el modelo de una persona. En nuestro caso obtuvimos el modelo desde el programa de Makehuman [7] como hemos descrito anteriormente. En la Figura 2.5.5.1 podremos ver las imágenes de esta prueba.

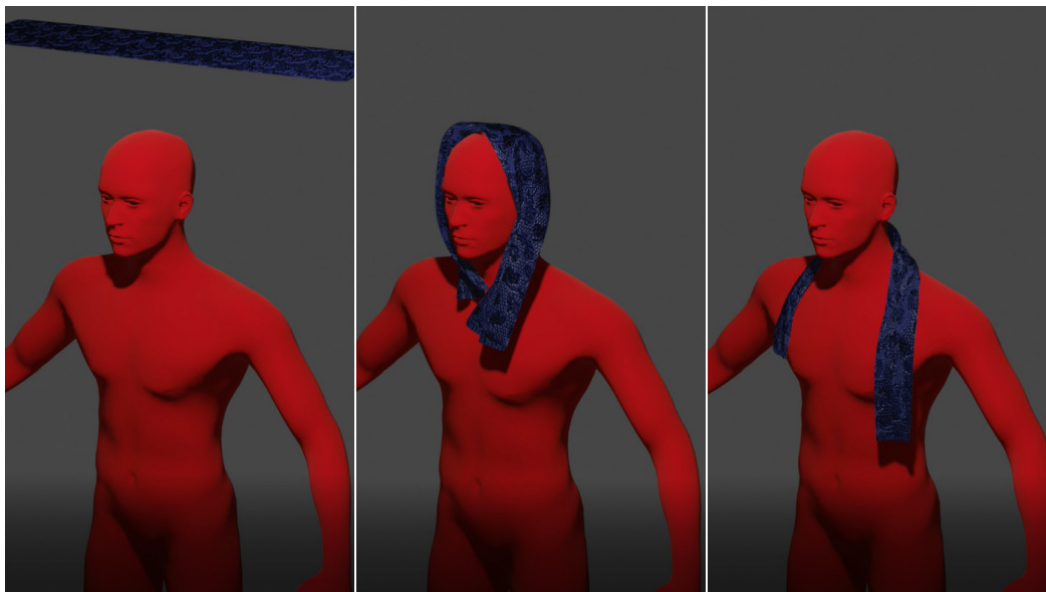


Figura 2.50: Imagen de la simulación de la caída de la bufanda sobre una persona que está en una posición fija.

Simulación de como se comporta la bufanda ante una persona en movimiento. En este caso la persona anterior tendrá que ser animada para poder ver como se comporta la bufanda ante esta situación. Para animar a la persona accedimos a la pagina de Mixamo [6] y lo animamos con una animación sencilla en la que el personaje anda. En la Figura 2.5.5.1 vemos algunas imágenes de esta simulación.



Figura 2.51: Imagen de la prueba en la que vemos el comportamiento de la bufanda en una persona que esta andando.

2.5.5.2. Dificultades

La principal dificultad fue realizar la última simulación, ya que me resultó muy difícil situar la bufanda en un lugar específico para que cayera y se adaptara bien sobre el individuo, es uno de los aspectos a mejorar en un futuro.

3. Validación del proyecto

En esta sección hemos presentado nuestro proyecto a 3 personas de diferentes perfiles, para que lo prueben y tener un feedback de nuestro plug-in. El tiempo medio entre las personas ha sido entre 25 y 40 minutos con la instalación del entorno y uso del plug-in.

3.1. Usuario 1

Describe brevemente los errores que consideras que has encontrado:

El programa presenta fallo cuando se inicia dos veces consecutivas ya que que es necesario borrar los elementos presentes en la ventana.

¿Qué aspectos y/o elementos te han gustado o llamado la atención?

La capacidad de customización de la bufanda es uno de los elementos más relevantes en el proyecto. La posibilidad de cambiar apariencias y texturas lo dota de gran interés desde el punto de vista comercial.

Otro elemento reseñable es la viabilidad de introducir viento en la animación. Visualmente proporciona muchas posibilidades.

¿Qué aspectos y/o elementos no te han gustado?

Desde el punto de vista de usuario convencional con conocimiento bajo en programación puede llegar a resultar poco intuitivo la interfaz de Blender. En cuanto al programa en sí, el proceso de subir fotos al programa para customizar la bufanda resulta algo tedioso. Es necesario incluir dos imágenes, cada una con un formato distinto.

¿Qué mejorarías?

La posibilidad de trabajar con otro tipo de prendas.

La introducción de un fragmento código en la fase de customización que sea capaz de reconocer el formato de imagen subida (jpg o png) y cree un archivo con la misma fotos pero en el formato restante.

Observaciones adicionales:

Puliendo los detalles mencionados, el programa informático desarrollado tiene cualidades para ser usado en proyectos de moda o marketing.

3.2. Usuario 2

Describe brevemente los errores que consideras que has encontrado:

A la hora de instalar Blender, en el manual de usuario no se especifica nada acerca del idioma, sin embargo, si el idioma es Español en lugar de Inglés el programa nos dará un error proveniente del script.

¿Qué aspectos y/o elementos te han gustado o llamado la atención?

La posibilidad de personalizar la bufanda, pudiendo añadir texturas, flecos, etc. De esta forma se puede comprobar cómo se comporta la misma dependiendo de los parámetros de configuración establecidos.

La forma de poder ejecutar el programa y realizar una primera visualización de la animación es bastante sencilla, considero que cualquier usuario siguiendo el manual podría hacerlo fácilmente.

¿Qué aspectos y/o elementos no te han gustado?

La necesidad de incluir las imágenes normales y de altura cada una en un formato diferente (jpg y png).

¿Qué mejorarías?

La disposición de los botones de material y texturas, viento y flecos creo que podría mejorarse ya que me han resultado un poco incómodo acceder a dichos botones.

Observaciones adicionales:

3.3. Usuario 3

Describe brevemente los errores que consideras que has encontrado:

El apartado del tutorial debería incluir algo más de instrucciones sobre el uso de la interfaz al aplicar un material a la bufanda, como dónde se encuentra la interfaz en el programa y qué debería ponerse en la parte de texto. Además debería de especificarse cómo se ha de colocar el fichero .blender y las imágenes de las texturas para que no haya problema con la búsqueda de ficheros al aplicar el material.

¿Qué aspectos y/o elementos te han gustado o llamado la atención?

El detalle del viento es muy curioso y se asemeja muy fielmente a las físicas de una ráfaga de viento real. El movimiento de la bufanda al caer y tocar el cuerpo que se encuentra debajo de ella también es llamativo. Hasta lo que he comentado en el primer punto, el tutorial está perfectamente explicado.

¿Qué aspectos y/o elementos no te han gustado?

No he visto ningún apartado que no me haya gustado.

¿Qué mejorarías? La parte del tutorial que he mencionado en el primer punto, por lo demás es un gran trabajo en mi opinión.

Observaciones adicionales: Ninguna.

3.4. Conclusiones tras el feedback

Todos los usuarios se han quejado de que al ejecutar el plug-in no encontraban la interfaz de nuestra aplicación, esto es debido a que Blender “oculta” estas ventanas cuando el programa está recién instalado, por lo que hay que pulsar un botón para que aparezcan nuestros paneles, en los que se pueden modificar la prenda y la escena. Para resolver este problema he incluido en el manual de usuario unas instrucciones mas detalladas sobre como activar estos paneles.

4. Modelo de negocio

Este trabajo incluye un estudio sobre el posible impacto que puede tener esta aplicación en el mercado en un futuro.

4.1. Objetivos

El objetivo principal es que nuestra herramienta pueda facilitar a los usuarios que quieran comprar algo en una tienda de moda online, dando la posibilidad a estos de ver como se adaptaría el producto a un cuerpo, o observar el modelo en ámbitos diferentes. De este modo la empresa ofrecería un servicio que a los usuarios gustaría bastante y facilitaría la compra de sus productos, ganando así una buena opinión y crítica sobre la empresa.

4.2. Análisis del entorno

Para establecer un modelo de negocio hay que evaluar a que entorno va dirigido. En este caso la aplicación esta mas enfocada a empresas que quieran ofrecer una herramienta a sus usuarios de manera online, en los que estos puedan realizar pruebas y simulaciones, por ejemplo: de todo el catalogo de la tienda.

En el año 2018, se registraron aproximadamente 9.000 empresas dentro del sector de la moda en España [9]. Siendo muy optimistas y estimando que el 80 % tienen pagina web, nuestra herramienta podrían llegar a usarla unas 7.200 empresas.

4.3. Plan de marketing

El plan de mercado consistiría en ofrecer nuestra herramienta de manera temporal a pequeñas tiendas de ropa, para comprobar que los usuarios hacen uso de ella, tras esto, podremos ofrecer esta herramienta a estas mismas empresas fijando un precio. En un futuro si la herramienta funciona se podría ofrecer a grandes compañías como: Inditex, Nike, Adidas, Christian Dior, etc.

5. Conclusiones y trabajo futuro

Los avances conseguidos con este script, han permitido lograr el objetivo principal que era poder simular diferentes tipos de materiales o tejidos. Además desarrollamos una serie de funcionalidades para que el usuario pueda probar la herramienta. Gracias a Blender hemos logrado obtener una animación realista de nuestro complemento en diferentes ámbitos.

En un futuro se podrá mejorar el comportamiento de los materiales, en especial su realismo. Una opción es crear un material específico con sus propios parámetros personalizados y no cogiendo los que tiene Blender por defecto. De este modo, podremos mejorar el comportamiento de nuestro tejido. Otras mejoras que se podrían incluir en un futuro serían: aumentar el número de prendas, dando la posibilidad al usuario de crear otro tipo de ropa, por ejemplo: camisetas, chaquetas, pantalones, etc. También podremos dotar nuestro modelo de persona humana, en un principio "desnudo", por un avatar personalizado en el cual si el usuario desea probar cualquier tipo de prenda pueda hacerlo o incluir un modelo de un avatar vestido en el caso que se deseen probar solo complementos. Todos estos posibles avances, tienen relación con el mundo de la moda actual, ya que hoy en día cada vez son más las tiendas que deciden incluir estas opciones en sus webs o portales, con el fin de facilitar a los usuarios la compra de sus catálogos.

Además hemos publicado públicamente la herramienta en Gitlab, para que cualquier usuario interesado pueda acceder, mediante este enlace [2]. También se ha realizado un estudio en el que algunos usuarios de diferentes perfiles han probado la herramienta, para ver poder el feedback y estudiar posibles mejoras.

6. Manual de usuario

En esta sección se mostrarán los pasos para poder hacer uso del Script.

6.1. Instalación

Para instalar Blender tendremos que acceder a su pagina web en el apartado de descargas [1] y elegir la que se adapte a nuestro equipo dependiendo del sistema operativo de nuestro ordenador, importante tiene que ser la versión 2.92 de Blender.

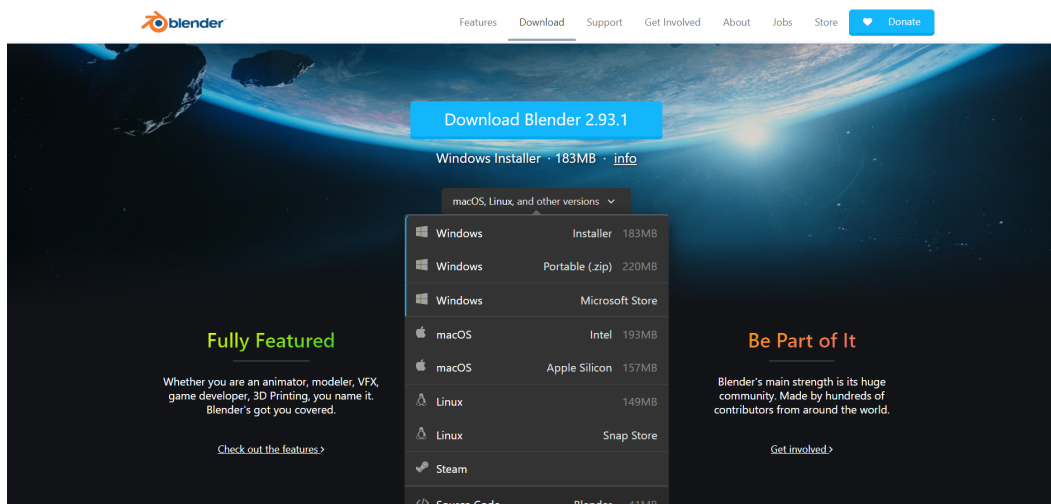


Figura 6.1: Pagina Web de Blender donde se ve la opción de descargar el programa, con las versiones para los diferentes sistemas operativos .

Una vez descargado ejecutamos el instalador.

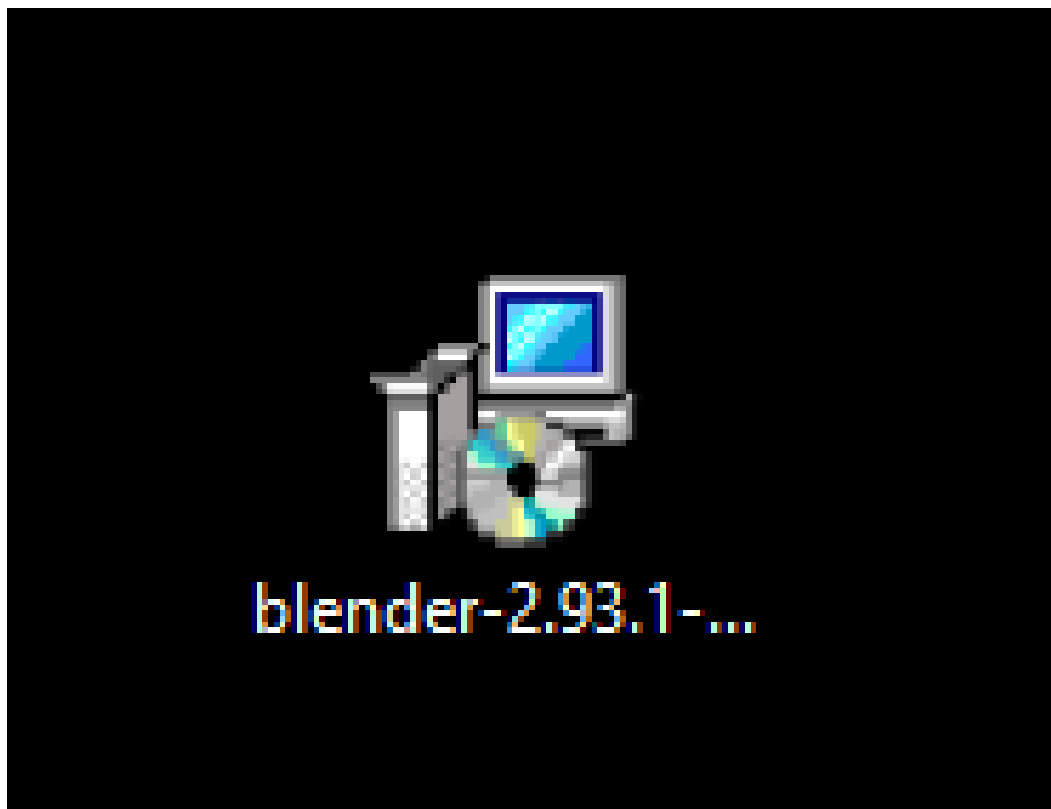


Figura 6.2: Instalador de Blender .

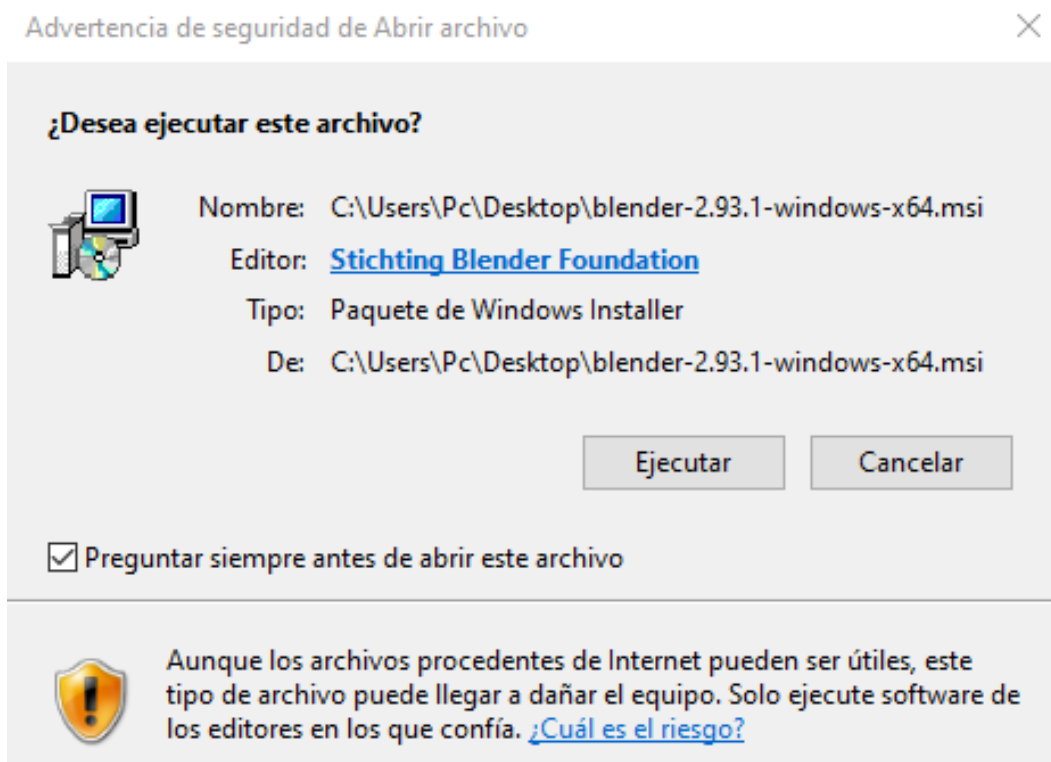


Figura 6.3: Ejecución instalador de Blender .

Cuando ejecutamos el instalador aparecerá la ventana de instalación, tendremos que clicar next para continuar con la instalación.

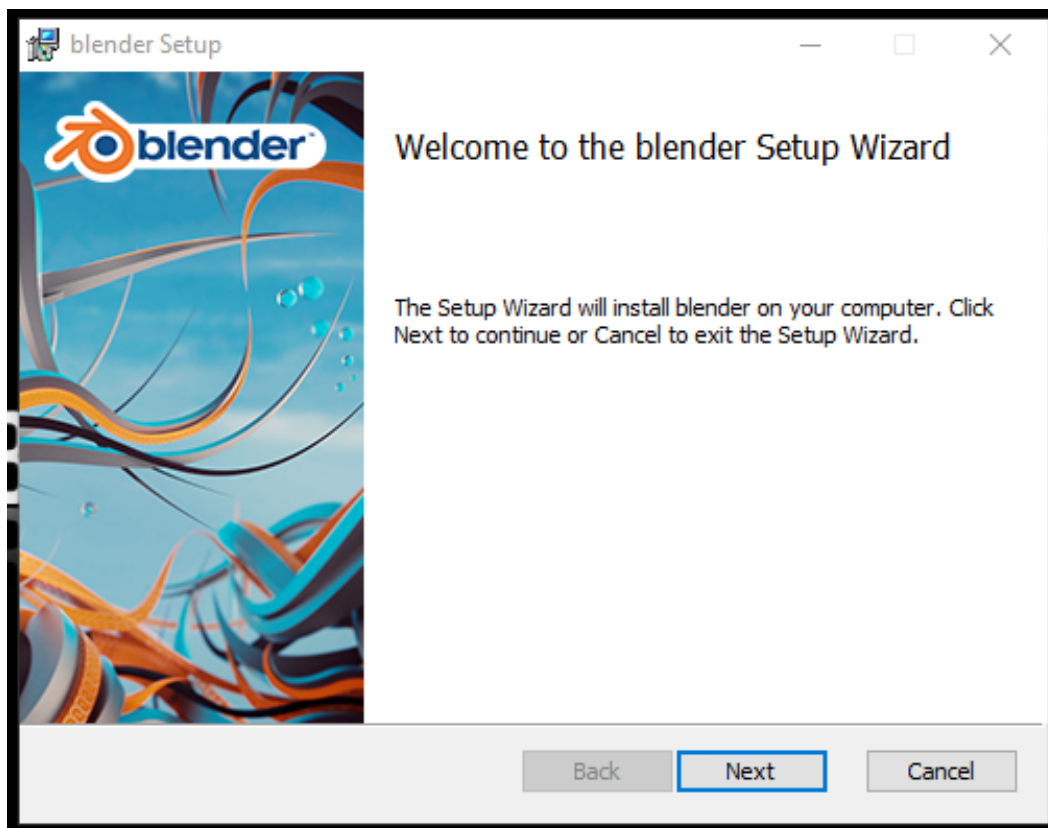


Figura 6.4: Ventana de instalador de Blender .

En la siguiente ventana aparecen los términos de licencia para continuar hay que aceptarlos y clicar en next.

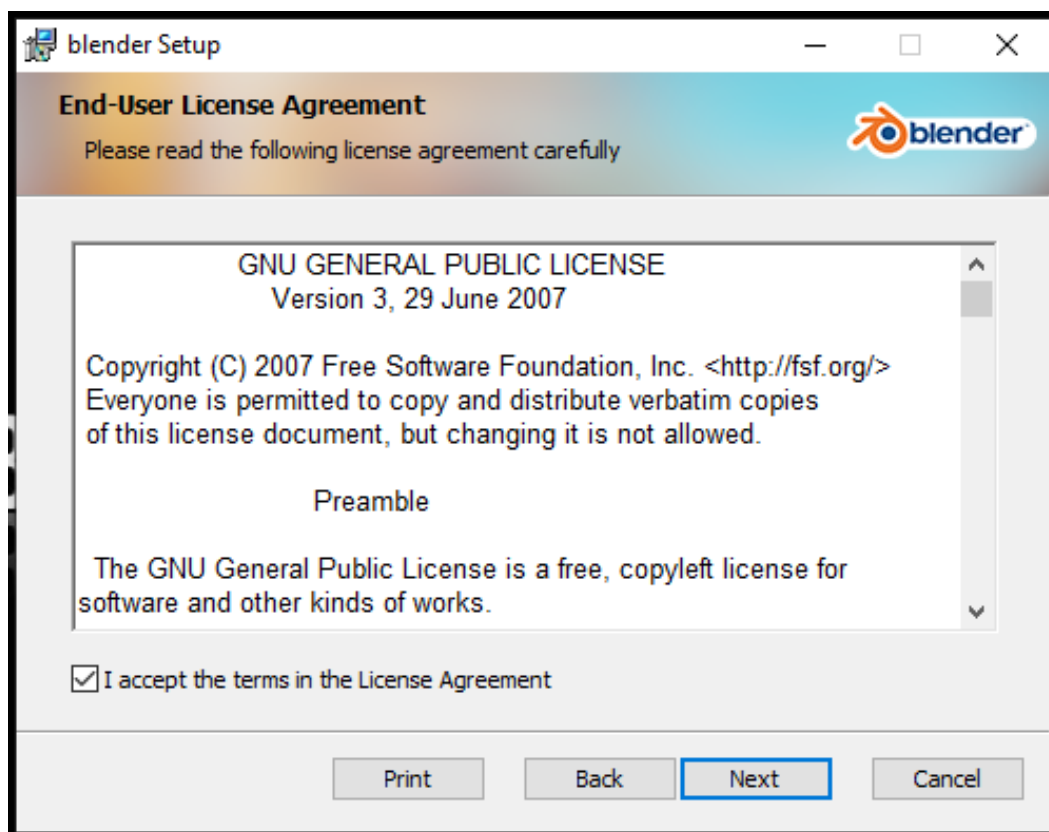


Figura 6.5: Ventana de instalador de Blender con los términos de licencia.

El siguiente paso es indicar donde queremos que sea instalado el programa.

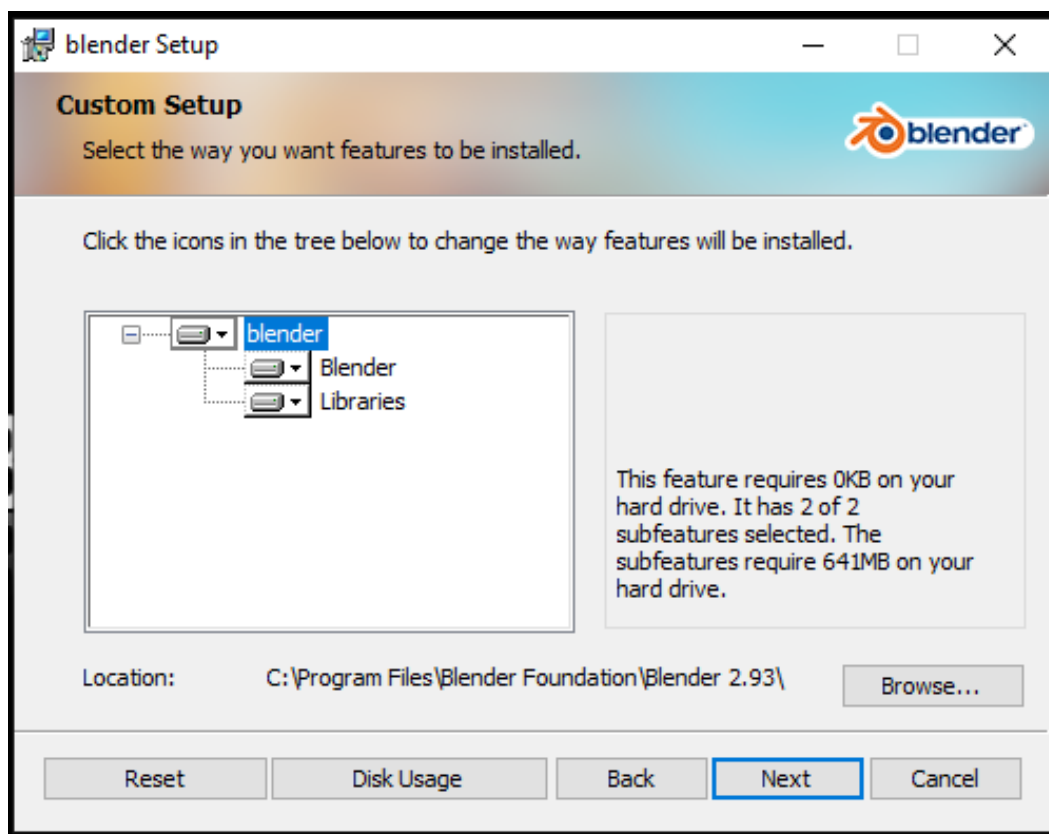


Figura 6.6: Ventana de instalador de Blender con el directorio de instalación.

Por ultimo solo tendremos que clicar en install para completar la instalación.

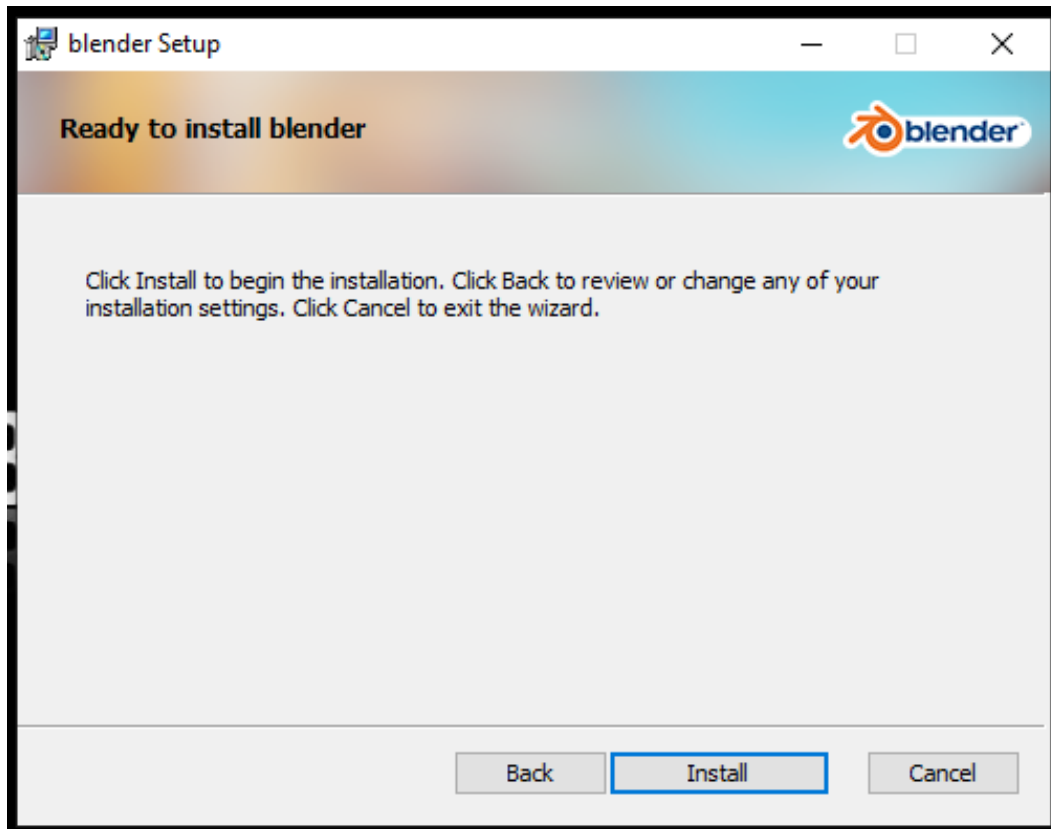


Figura 6.7: Última ventana de instalación de Blender.

Blender se habrá instalado con éxito.

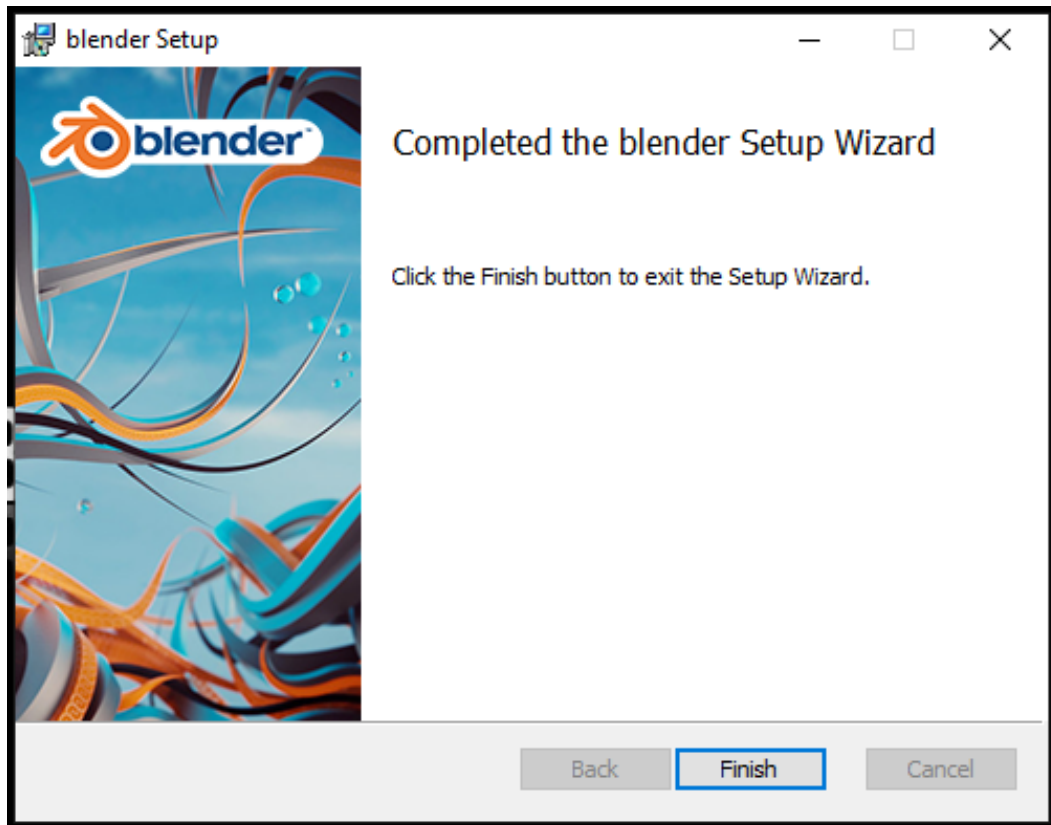


Figura 6.8: Blender instalado.

6.2. Lanzamiento del script

Antes de lanzar el script debemos eliminar el cubo por defecto que aparece en la escena para poder ejecutar el código.

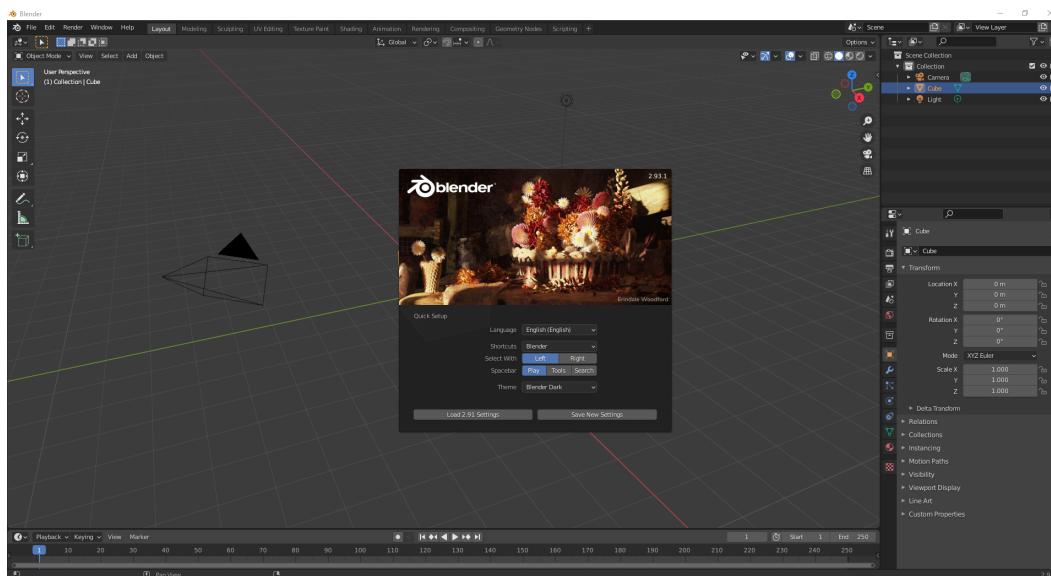


Figura 6.9: Primer menú que aparece al abrir Blender.

Para obtener el código, este ha sido subido a la plataforma de Gitlab para que todo el mundo pueda descargarlo gratuitamente y ejecutarlo libremente [2].

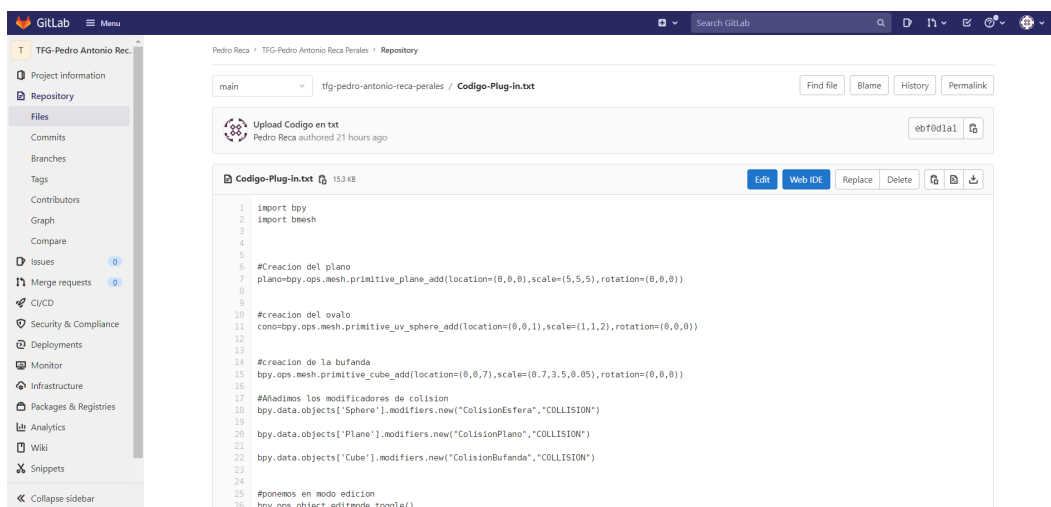


Figura 6.10: Imagen del código en Gitlab que podrá ser descargado gratuitamente.

Cuando tengamos el código descargado habrá que copiarlo en el apartado scripting de Blender y crear un nuevo fichero de código. En la Figura 6.11 podemos observar el apartado de scripting en Blender y la línea roja indica donde hay que pulsar para crear un nuevo fichero de código.

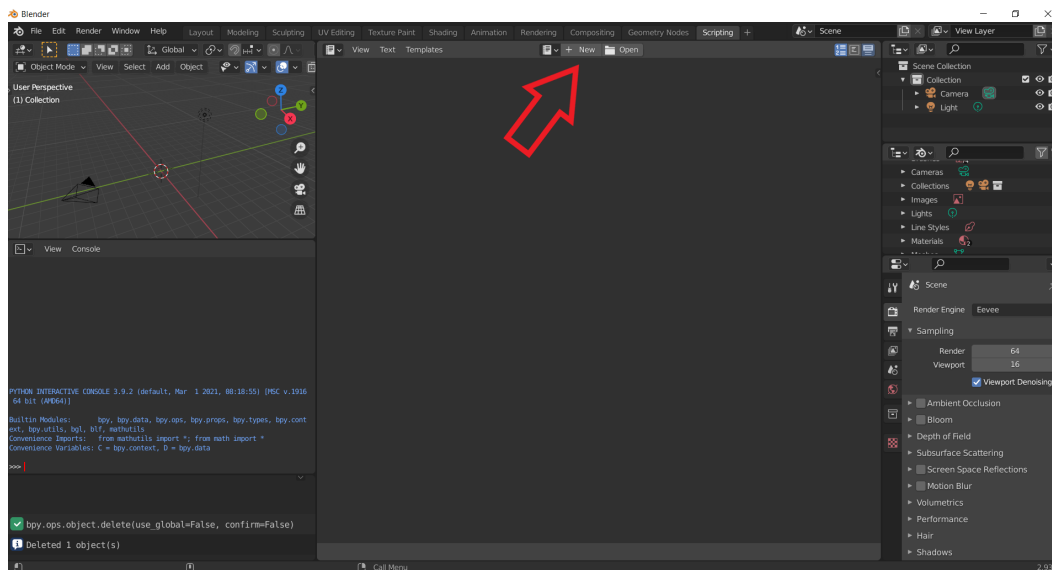


Figura 6.11: Imagen del apartado de scripting donde la flecha roja indica que hay que pulsar para crear un nuevo fichero de código.

Al tener ya un fichero de código nuevo podremos copiar el código proporcionado que contiene todo para poner en marcha el plug-in y deberemos guardar nuestro fichero en File - Save y aparecerá una ventana como la de la Figura 6.12 en la que se debe de poner el nombre del fichero y el directorio donde se guardará.

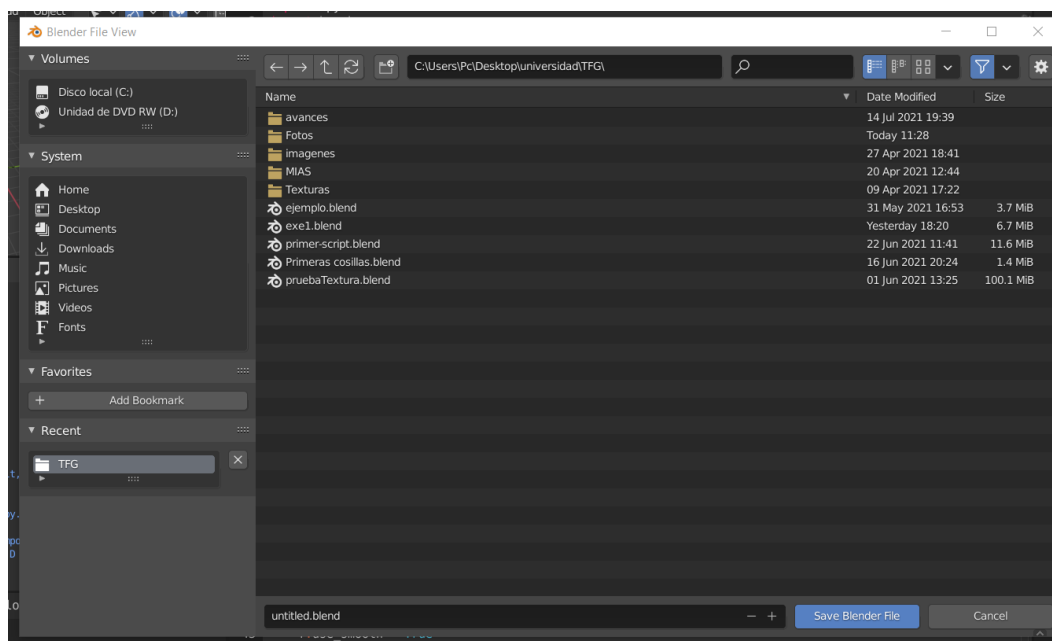


Figura 6.12: Ventana de guardado de fichero donde se indicara el nombre del archivo y el directorio donde se guardará.

Cuando hayamos guardado nuestro proyecto podremos poner en marcha el plug-in, que se ejecutará pulsando el botón de Run Script en el apartado de scrip-

ting. En la Figura 6.13 se verá una flecha roja que indica cual es el botón de Run Script.

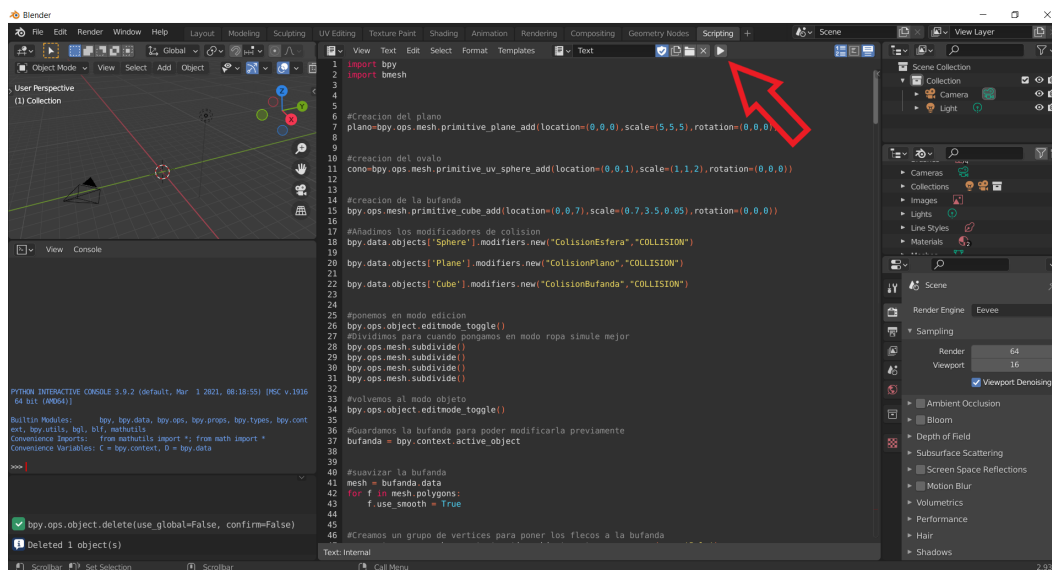


Figura 6.13: Captura con una flecha que indica que botón hay que clicar para ejecutar el script.

Al ejecutar se crean todos los elementos de la escena y podremos ir al modo layout para ver la escena mas grande y mejor. Para que aparezca la interfaz con los paneles diseñados, habrá que pulsar un botón que esta indicado en la imagen 6.14 con una flecha roja, una vez pulsado aparecerán todos los paneles diseñados para modificar/editar nuestro objeto y la escena.

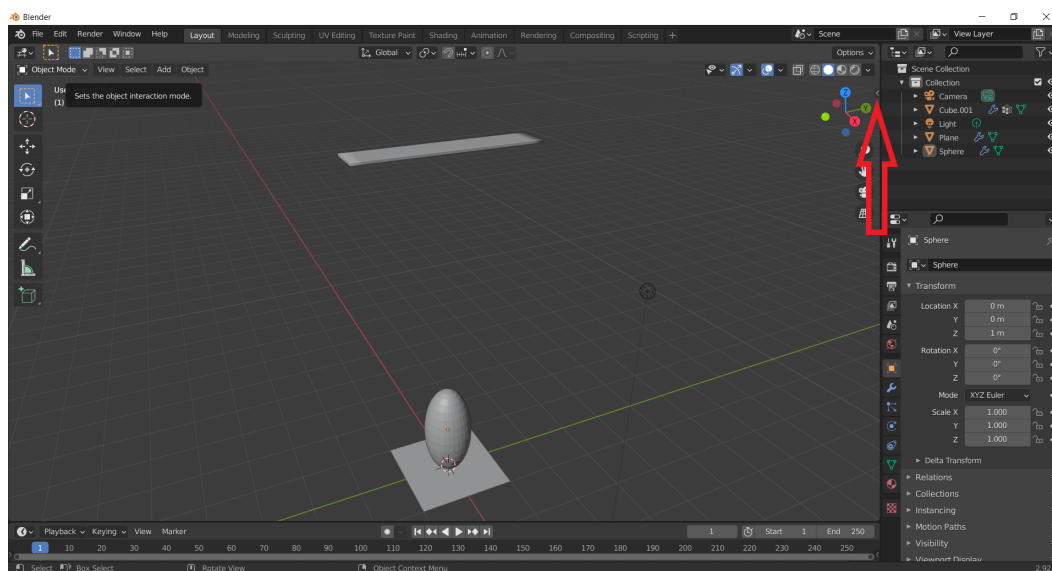


Figura 6.14: Captura de la escena tras ejecutar el script en el modo layout, indicando que botón pulsar para activar los paneles de nuestra interfaz.

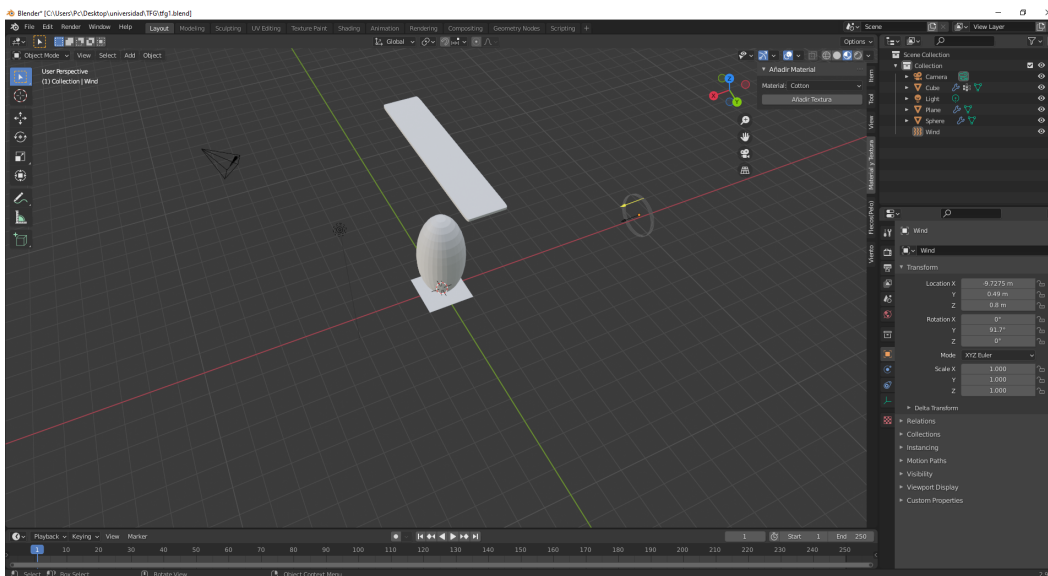


Figura 6.15: Captura de la escena tras ejecutar el script en el modo layout, con los paneles activados.

Ya podremos hacer uso de los paneles creados, aunque para el panel de Añadir una textura será necesario modificar el script con la ruta donde el usuario que use el plug-in tenga las imágenes de las texturas. En la Figura 6.16 vemos como filepath es el directorio donde se encuentra nuestro proyecto guardado y le incluimos la carpeta imágenes que es donde se encuentran todas mis texturas, por ejemplo si yo tengo mi archivo .blend en el escritorio, en el escritorio debe haber una carpeta llamada imágenes, donde estarán todas las texturas que se quieran usar. Para poder aplicar una textura debemos tener 3 imágenes que son: la imagen de color de la textura que deberá tener un formato .jpg, la imagen normal de la textura que deberá tener un formato .jpg y el mapeado de alturas que tendrá un formato .png. Importante el nombre de las estas dos ultimas imágenes deberá ser nombre de nuestra imagen de color + Normal/Height, por ejemplo nuestra imagen de color se llama TelaAzul.jpg pues las imágenes de alturas y normal deberán estar guardadas en nuestro dispositivo como: TelaAzulNormal.jpg y TelaAzulHeight.png para que nuestro script pueda reconocerlas correctamente

```
#Cargamos la imagen de la textura que ha sido introducida en el input de texto
img = bpy.data.images.load(filepath = "//imagenes\\"+str(self.col)+".jpg")
#se la aplicamos al nodo
img_node.image = img
#Cargamos las imagenes normales y de altura pertenecientes a la textura
img_n = bpy.data.images.load(filepath = "//imagenes\\"+str(self.col)+"Normal.jpg")
img_h = bpy.data.images.load(filepath = "//imagenes\\"+str(self.col)+"Height.png")
```

Figura 6.16: Captura del código del script donde vemos la ruta de las imágenes.

Para poder ver la simulación de la escena antes de renderizar, en el modo de layout podremos tener una previsualización de la animación, en la que podemos seleccionar el frame donde empieza la simulación y cuando acaba, también podemos pararla, reanudarla o rebobinarla. En la Figura 6.17 abajo observamos una

barra que indica por donde avanza la animación y los botones para parar, reanudar o rebobinar.

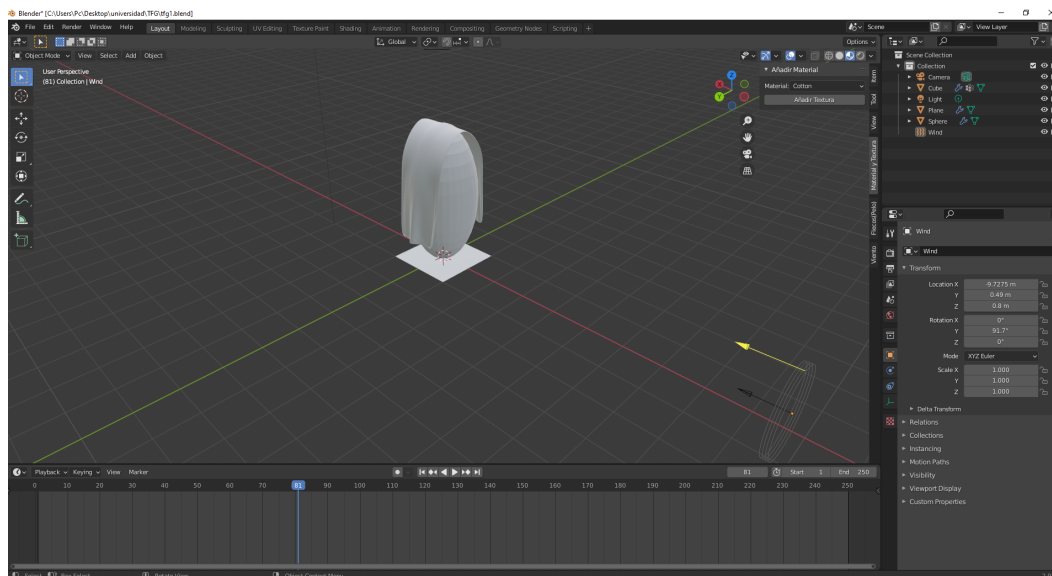


Figura 6.17: Previsualización de la animación en el modo layout.

6.3. Añadir otros modelos a la escena

Como se vio en la sección ?? explicamos como crear un modelo de una persona y su respectiva animación para incluirlo en la escena habrá que seguir estos pasos. En File - Import - fbx (formato en el que exportamos nuestro modelos o animaciones).

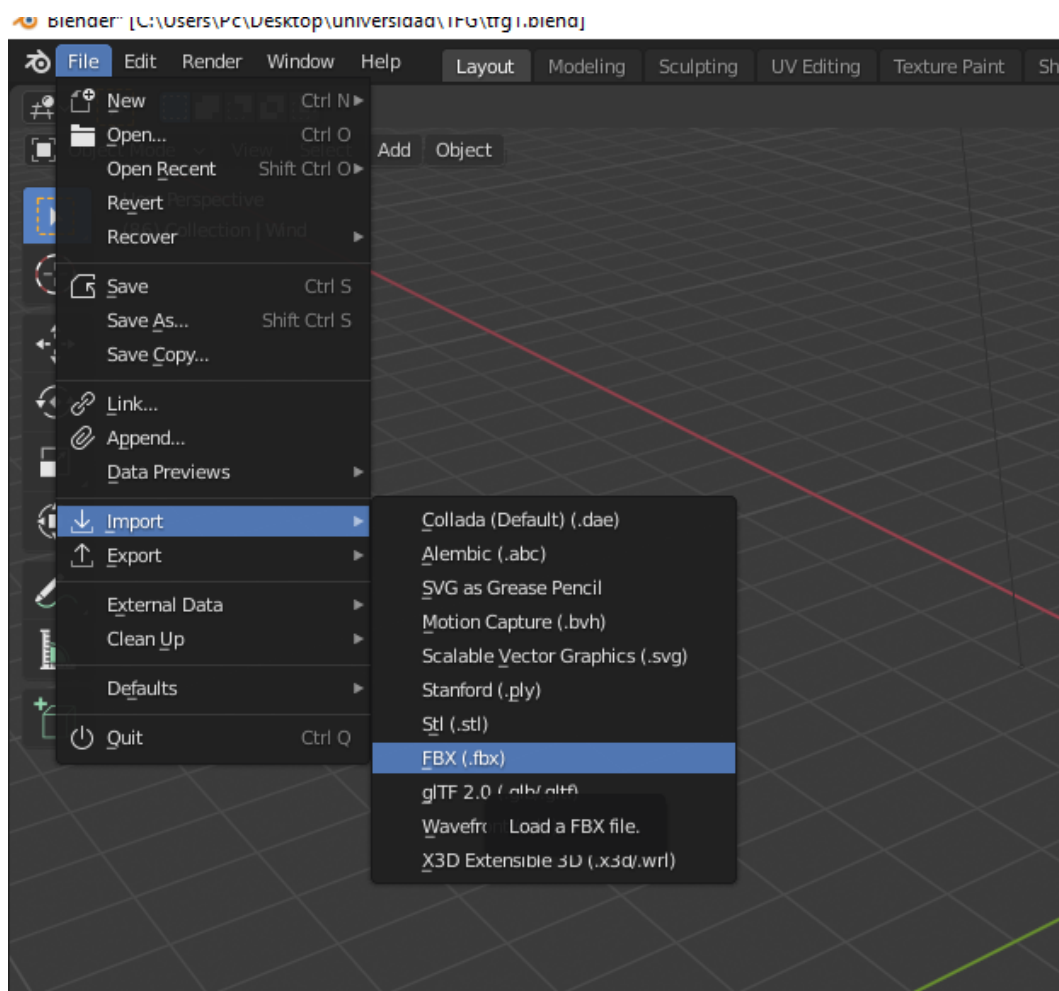


Figura 6.18: Importar animación o modelo.

Al clicar aparecerá una ventana donde debemos buscar nuestro modelo/animación en el directorio donde lo guardásemos cuando lo tengamos lo seleccionamos y pulsamos el botón Import FBX.

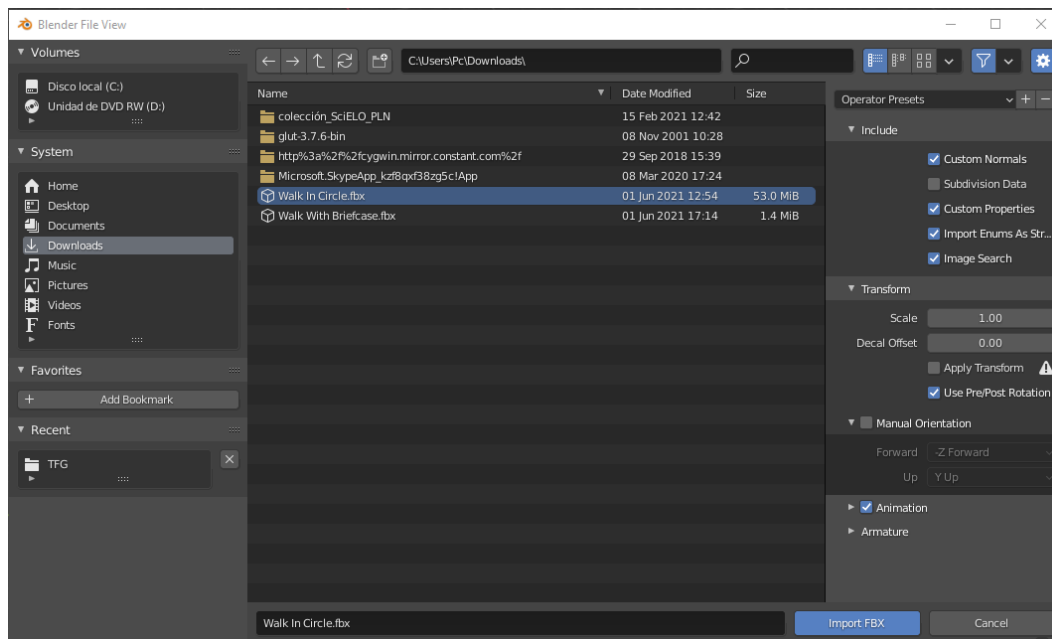


Figura 6.19: Ventana para importar animación o modelo.

Ya habremos importado nuestro modelo

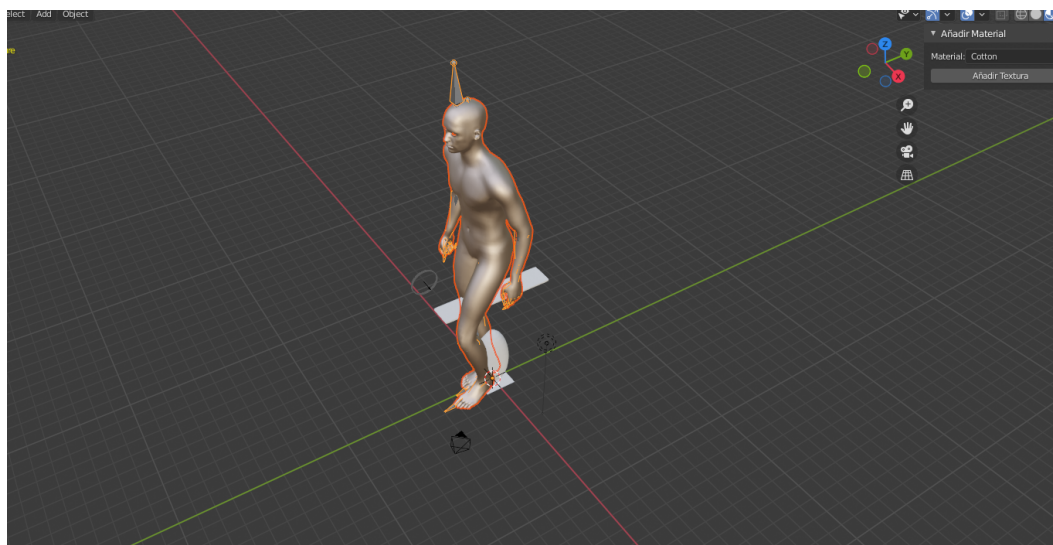


Figura 6.20: Animación importada.

Como observamos en la Figura 6.20 observamos que se ha importado bien pero se ve la armature que es como el esqueleto del objeto por lo que esto habrá que retirarlo de la escena. En la Figura 6.22 podremos ver una flecha roja que señala una especie de ojo, este ojo al clicar sobre el eliminara la visibilidad de el armature (esqueleto) de la escena.

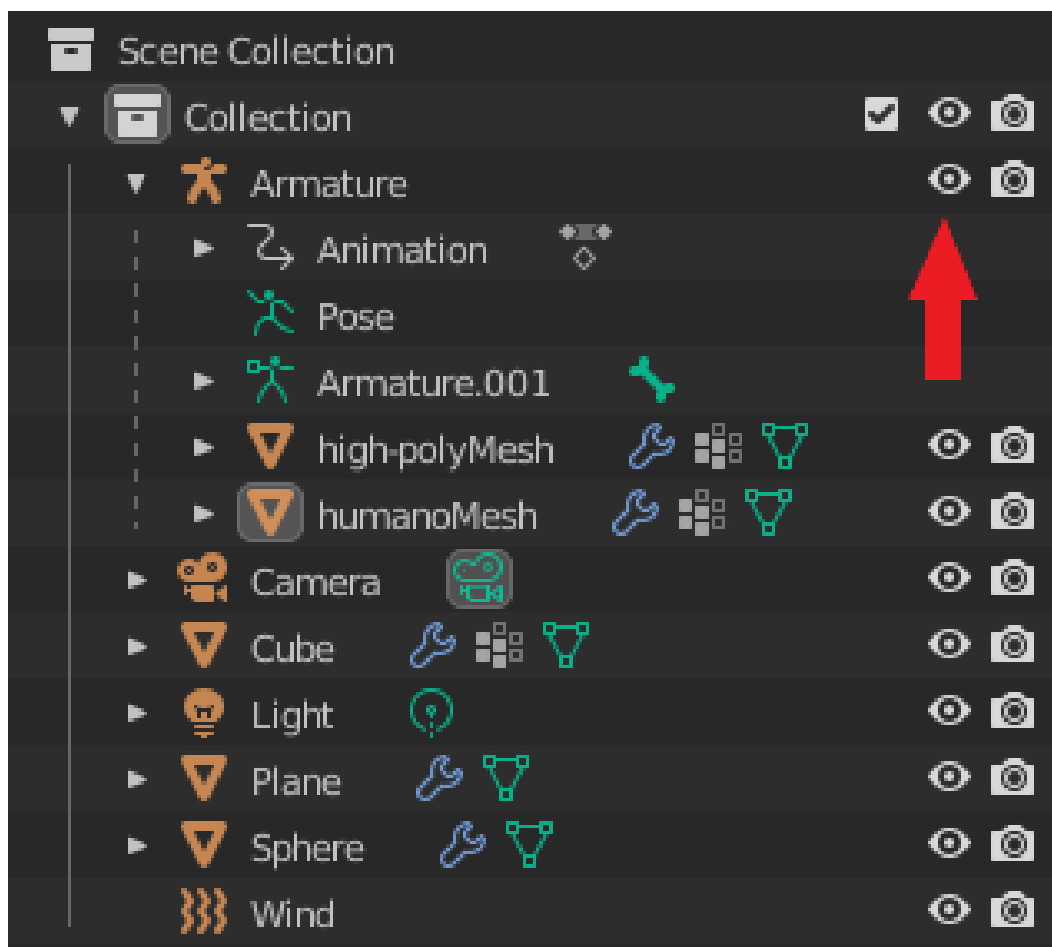


Figura 6.21: Eliminar esqueleto de la escena.

El esqueleto del objeto será no visible y no aparecerá en la simulación.

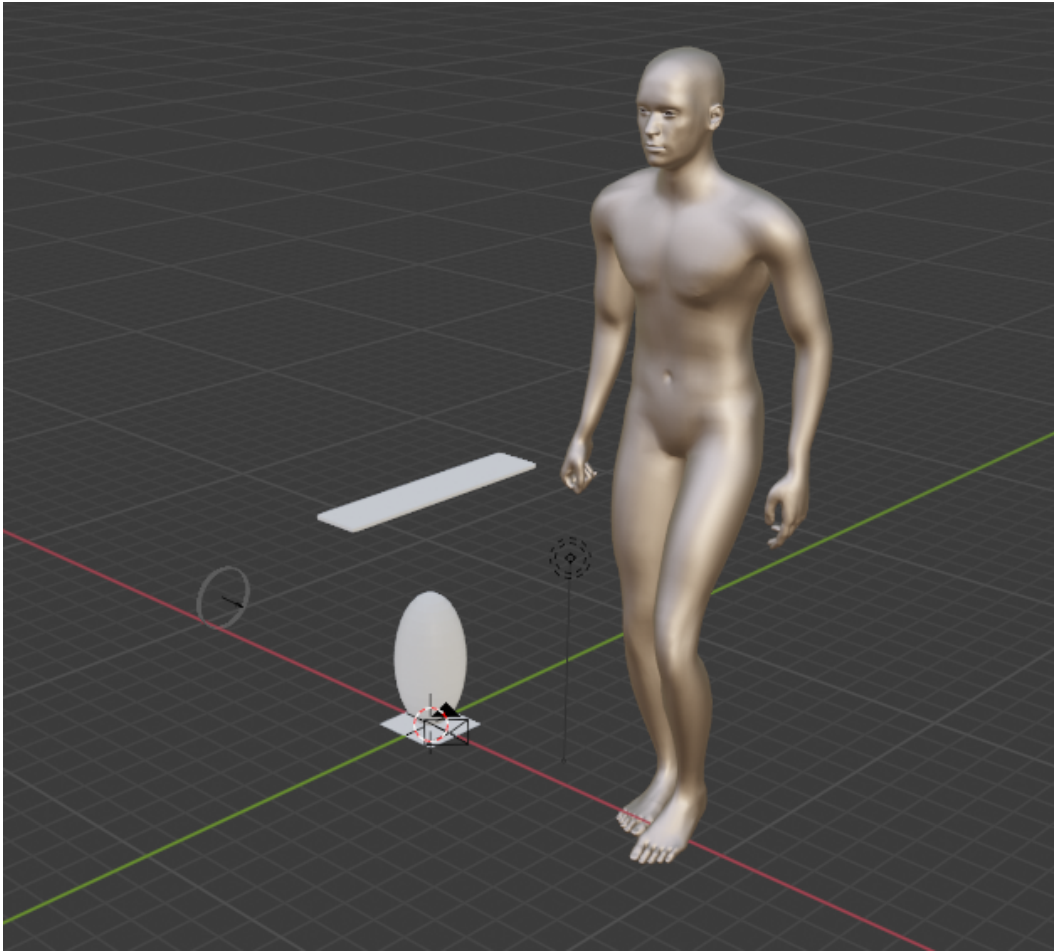


Figura 6.22: Esqueleto eliminado de la escena.

Importante para mover o escalar el modelo, el esqueleto tendrá que estar visible y tendremos que seleccionar la animación en la escena, si no el modelo se deformará. En la Figura [6.25](#) vemos en azul los elementos que tenemos que seleccionar para que se apliquen los cambios correctamente.

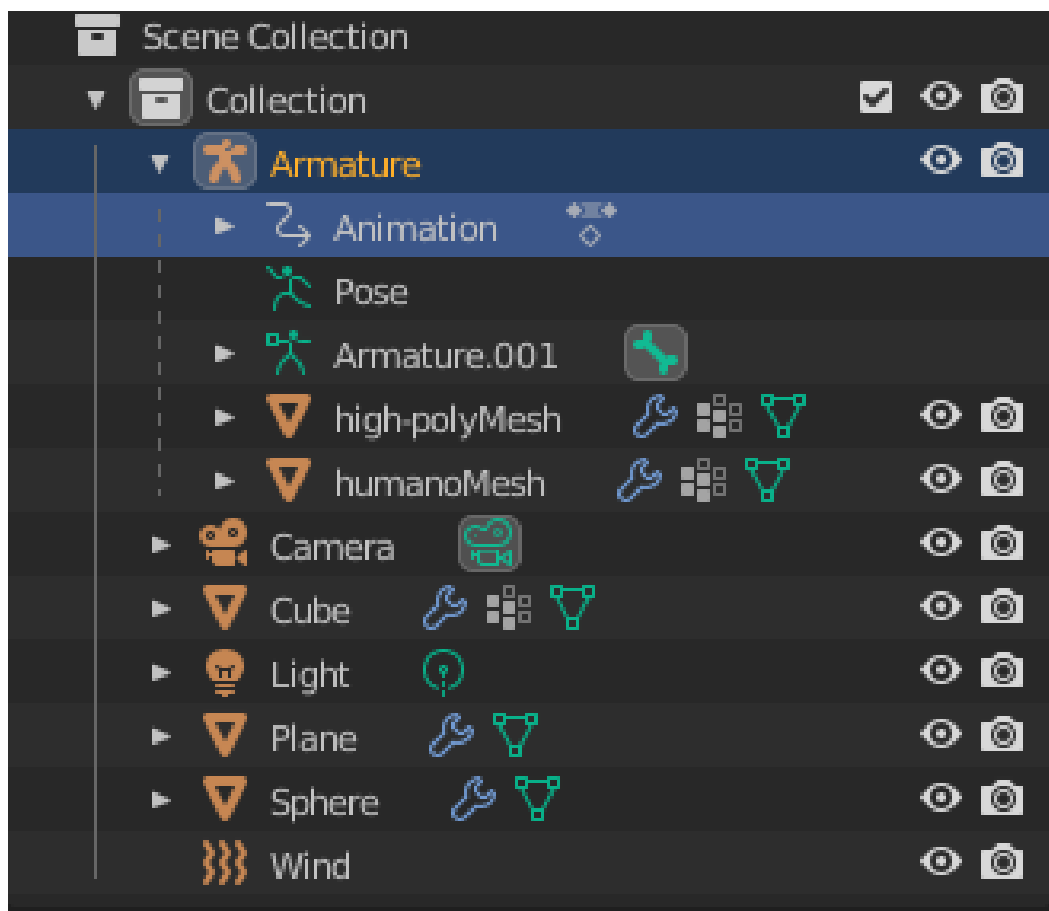


Figura 6.23: Elementos que tendrán que estar seleccionados para poder mover o escalar correctamente a la animación/modelo.

6.4. Renderizado de la animación

Una vez tengamos nuestra animación como deseamos, podremos exportarla para poder guardarla o tenerla en formato de vídeo. Para ajustar los parámetros de el render en las output properties, podemos ajustar el directorio donde se guardara nuestro render, el formato, y los frames donde empezará y acabará la animación. En la Figura 6.24 se observa todos estos parámetros mencionados.

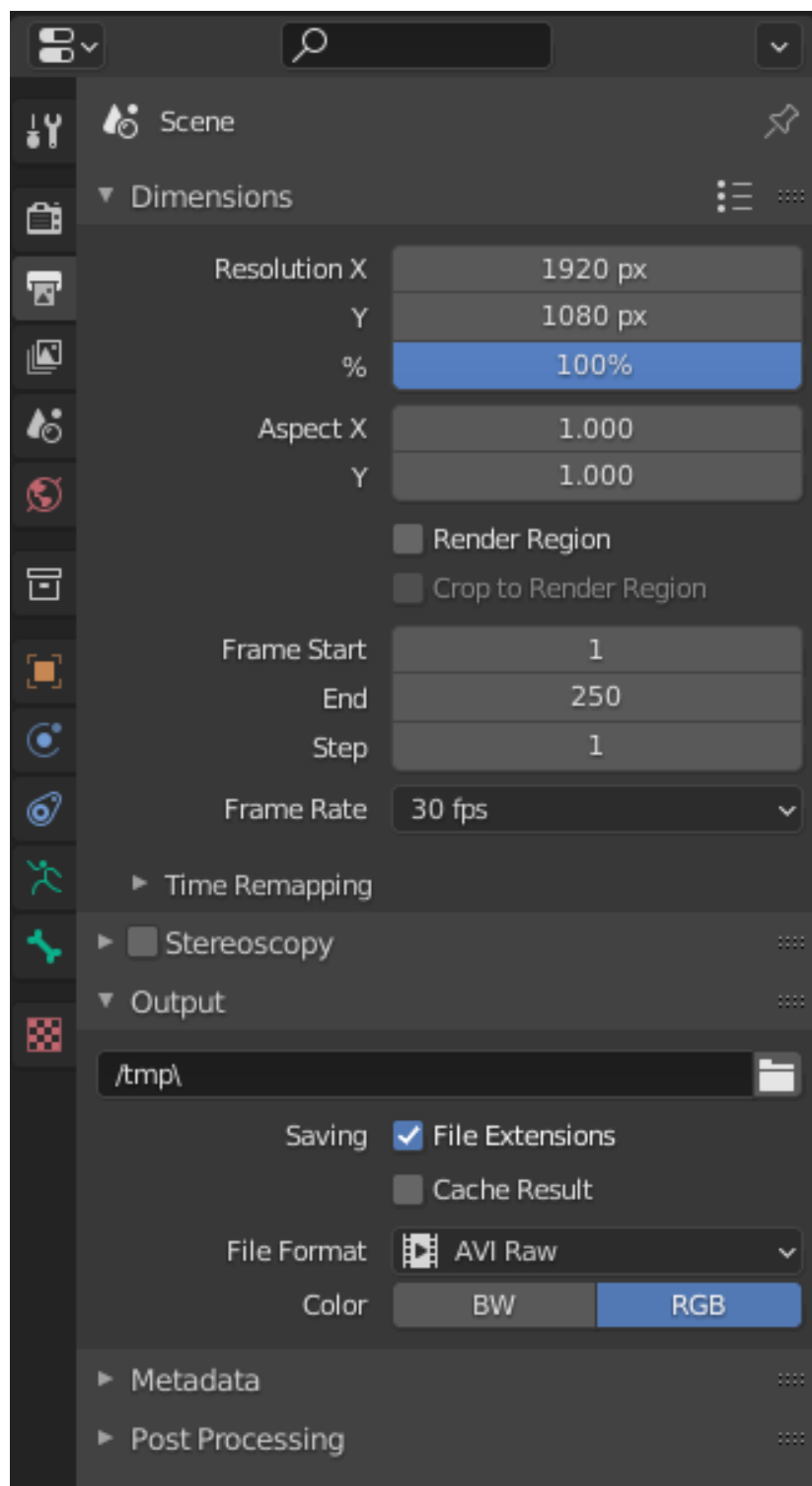


Figura 6.24: Captura donde se ven los parámetros modificables para el render como: directorio donde se guarda el resultado, formato y frames donde empezara y acabara la animación.

Cuando hayamos configurado estos parámetros como queramos ya podremos realizar el render que sera seleccionando Render - Render Animation

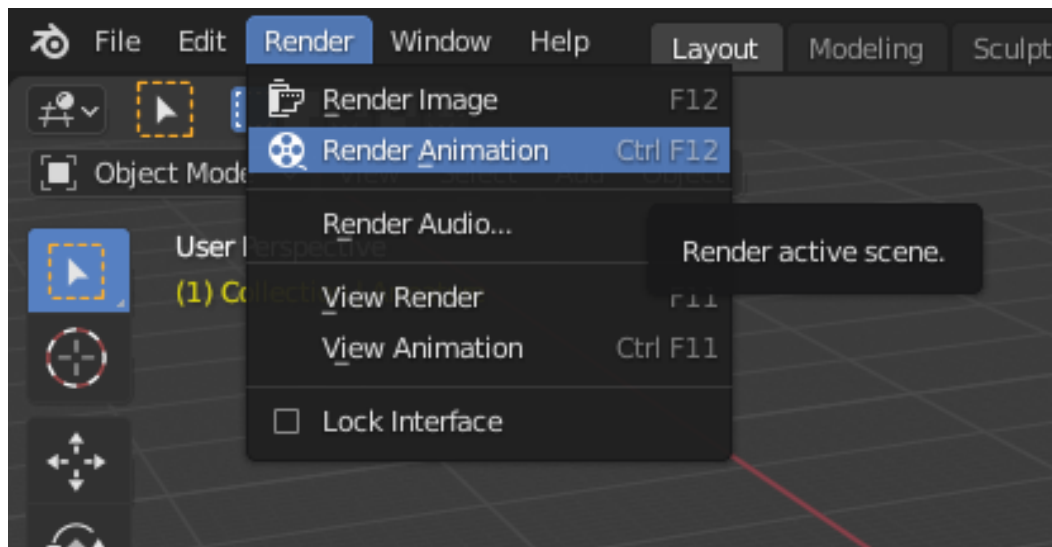


Figura 6.25: Animation Render.

Se abrirá una ventana nueva con la previsualización de la animación mientras se carga el renderizado en la Figura 6.26 observamos esta ventana.

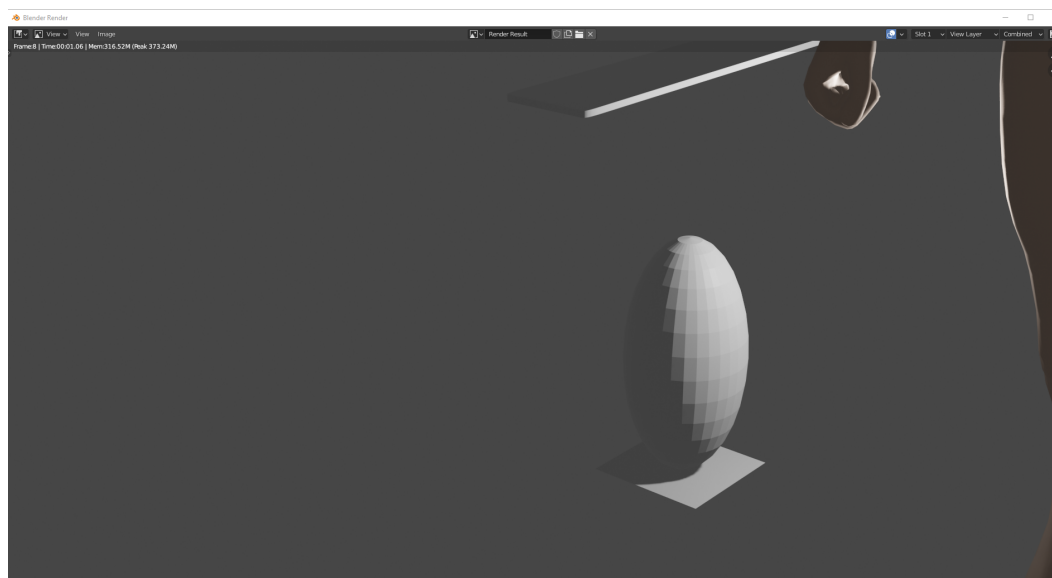


Figura 6.26: Ventana de previsualización de renderizado mientras se realiza el render.

Cuando termine podremos ver la animación que se encontrará en el directorio donde indicamos que se guardarían los resultados 6.24.

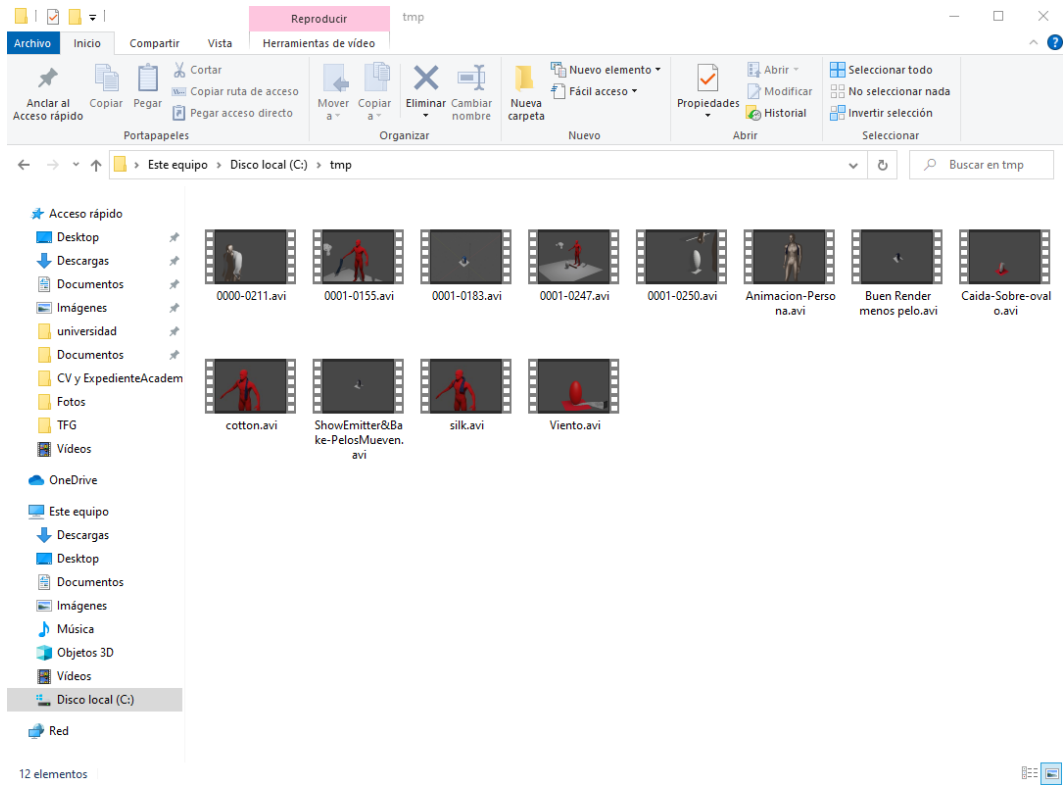


Figura 6.27: Resultados en formato vídeo, en el directorio elegido.

7. Bibliografía

- [1] URL <https://download.blender.org/release/Blender2.92/>.
- [2] URL <https://gitlab.com/pedroreca98/tfg-pedro-antonio-reca-perales/-/blob/main/Codigo-Plug-in.txt>.
- [3] URL <https://www.gimp.org/>.
- [4] URL <https://cpetry.github.io/NormalMap-Online/>.
- [5] URL <https://www.youtube.com/playlist?list=PL2nKh5kZkXJVCd7fmSL4QLzZWFL75orbB>.
- [6] URL <https://www.mixamo.com/#/>.
- [7] URL <http://www.makehumancommunity.org/>.
- [8] URL <https://www.blender.org/>.
- [9] URL <https://es.statista.com/estadisticas/477233/numero-de-empresas-del-sector-de-confeccion-de-prendas-de-vestir-en-espana/>.
- [10] . URL <https://es.indeed.com/career/programador-junior/salaries>.
- [11] . URL https://es.indeed.com/career/dise%C3%B1ador-3d/salaries?from=top_sb.
- [12] URL <https://www.trimirror.com/>.
- [13] . URL https://www.youtube.com/watch?v=ExwpFivx0Lk&ab_channel=Cinematix.
- [14] . URL https://www.youtube.com/watch?v=BI9_bQdZ30c&t=139s&ab_channel=TutsByKai.
- [15] Solvi Arnold and Kimitoshi Yamazaki. Fast and flexible multi-step cloth manipulation planning using an encode-manipulate-decode network (em* d net). *Frontiers in neurorobotics*, 13:22, 2019.
- [16] Hugo Bertiche, Meysam Madadi, and Sergio Escalera. Cloth3d: Clothed 3d humans. In *European Conference on Computer Vision*, pages 344–359. Springer, 2020.
- [17] Erhan Gundogdu, Victor Constantin, Amrollah Seifoddini, Minh Dang, Mathieu Salzmann, and Pascal Fua. Garnet: A two-stream network for fast and accurate 3d cloth draping. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8739–8748, 2019.

- [18] Damien Rohmer, Tiberiu Popa, Marie-Paule Cani, Stefanie Hahmann, and Alla Sheffer. Animation wrinkling: augmenting coarse cloth simulations with realistic-looking wrinkles. *ACM Transactions on Graphics (TOG)*, 29(6):1–8, 2010.
- [19] Jordi Sanchez-Riera, Aniol Civit, Marta Altarriba, and Francesc Moreno-Noguer. Avatar: Blender add-on for fast creation of 3d human models. *arXiv preprint arXiv:2103.14507*, 2021.
- [20] Igor Santesteban, Nils Thürey, Miguel A. Otaduy, and Dan Casas. Self-supervised collision handling via generative 3d garment models for virtual try-on, 2021. URL <http://gmr.v.es/Publications/2021/STOC21>.
- [21] Daisuke Tanaka, Solvi Arnold, and Kimitoshi Yamazaki. Emd net: An encode–manipulate–decode network for cloth manipulation. *IEEE Robotics and Automation Letters*, 3(3):1771–1778, 2018.