

Loop animado 3D estilo cartoon



Grado en Ingeniería Multimedia

Trabajo Fin de Grado

Autor:
Araceli Lucía Rebollo Domínguez

Tutor/es:
José Vicente Berná Martínez

Julio 2023



Universitat d'Alacant
Universidad de Alicante

Resumen

Para este trabajo de fin de grado la autora ha querido reflejar los conocimientos adquiridos durante el grado de Ingeniería Multimedia. Para ello, ha desarrollado un *loop* animado estilo *cartoon* en 3D en el que se desarrollan múltiples tareas que ponen en práctica estos conocimientos.

Este *loop* animado tiene como protagonista a su gata Kely. Consiste en una animación corta que se repite en bucle de forma infinita sin que se note el corte cuando vuelve al inicio de la reproducción. Se ha realizado una gestión y seguimiento de las tareas del proyecto. También, se han realizado todas las etapas correspondientes a un *pipeline* 3D de animación, que incluye la preproducción, producción y postproducción, además de la memoria del proyecto. Para el trabajo final no se utilizaron recursos externos, todo se realizó a mano por la autora. Se han usado las herramientas *software* necesarias para realizar todas las tareas del proyecto. Se ha logrado realizar el proyecto con herramientas gratuitas utilizadas en la industria profesional.

La animación consiste en una gata viendo redes sociales usando una tableta. Esta animación convertida en bucle esconde una moraleja que representa la cantidad de horas que una persona puede pasar navegando en redes sociales sin darse cuenta. Aunque es una animación corta, se han realizado muchos detalles que dotan de gran calidad a la animación. La gata al estar basada en una gata conocida, la autora ha podido recrear sus rasgos más característicos en el modelo 3D del animal, pero en una versión *cartoon* que simplifica la realidad. Se han realizado estudios para poder recrear la anatomía, los movimientos, la iluminación, los materiales y texturas con el máximo realismo posible dentro de este estilo. Se puede apreciar en los resultados: el detalle de los reflejos, la iluminación de la escena, la calidad de los modelos 3D y la de la animación.

Se ha dedicado mucho tiempo y esfuerzo en realizar el proyecto, al haber realizado todos los pasos del *pipeline* 3D, a mano, sin usar otros recursos externos. Finalmente, esto ha sido muy satisfactorio haber puesto en práctica los conocimientos, y haber obtenido un resultado óptimo, comparado con el nivel del que partía la autora en este ámbito al comenzar el trabajo.

Motivación, justificación y objetivo general

Realmente, echando la vista atrás, todo empezó en el instituto. Cuando pasé de 4º ESO y me preguntaron si iba a hacer bachillerato o algún grado medio. Bien, pues en ese entonces, estaba muy perdida todo lo que sabía era que quería hacer algo relacionado con ciencias. Me gustaban las matemáticas, el arte y los gatos, animales en general, pero hagamos hincapié en estos específicamente, ya que estarán presentes en la animación. Así pues, hice bachillerato de ciencias mientras iba pensando que quería hacer en el futuro. Fue en ese momento en el que descubrí el arte digital: diseño gráfico, *concept artist*, ilustrador digital, animador 2D y 3D, y modelado por computador. Es cierto que bachiller no me dejó mucho tiempo para explorar y aprender sobre estos temas, pero aun así ya tenía más claro por donde encaminarlo y pude practicarlo un poco. Al acabar, en una excursión, fue cuando me presentaron finalmente esta carrera de Ingeniería Multimedia. Que, además, estaba en San Vicente del Raspeig, Alicante donde yo había estado estudiando, por tanto, quedaba muy cerca. Mi oportunidad, porque la carrera mezclaba casi todos mis intereses: ciencias, matemáticas y también dejaba hueco para el arte (faltaban los animales), perfecto.

Por una parte, y para ser sinceros, la idea, surgió en primero de carrera, después de un largo fin de semana con mi gata hospitalizada. Esa mañana le iban a dar el alta cuando me llamaron para informarme de la mala noticia de que había fallecido, recuerdo que tenía el examen práctico de Matemáticas II, desconsolada, lo hice y, en su honor, me prometí que acabaría el grado y le dedicaría el trabajo final con un trabajo artístico en 3D. Conforme pasaron los cursos, me di cuenta que aunque en la carrera había asignaturas que dan ciertas bases para diseño, modelado y animación como Fundamentos de la Física, Fundamentos del Diseño Gráfico, Matemáticas II, Modelado y Animación por Computador, Gráficos por Computador, Postproducción Digital y Realidad Virtual. Pero para dedicarte a ello falta mucha información, claro, hay tantos conocimientos que no se puede dedicar tanto tiempo a uno específico, tiene que ser uno mismo el que, si le interesa, se embarque en esa aventura. Así que aun más convencida quería poner en práctica e investigar más este tema. Asimismo, en los dos primeros veranos me apunté a cursos relacionados con este tema que me parece muy interesante, el primero era sobre modelado orgánico y el segundo sobre animación 2D.

Por otra parte, obviamente este tema me interesa mucho, me gustaría dedicarme a alguna salida de este mundillo. E igualmente creo que es un sector que está en auge, y en concreto, los *loops* se utilizan mucho en distintos ámbitos. Por ejemplo, en publicidad, los carteles digitales que

vemos por la calle o las carteleras de las películas que cambian los anuncios y consecutivamente vuelven a aparecer. Otro ejemplo serían los videojuegos, muchas veces, las pantallas de carga y/o de inicio usan este recurso para que visualmente sea más dinámico y el jugador no se confunda y crea que se ha colapsado. En redes sociales (RRSS) es un recurso del que se abusa mucho con *gifs*, *stickers*, o incluso los propios artistas para mostrar sus trabajos en el CV de una manera rápida y dinámica.

También, he pensado que, vale voy a hacer un *loop*, pero ¿de qué? Vale, pues he decidido que al ser un bucle va a tener que conectarse el inicio con el final de forma que no se sepa cuando ha vuelto al comienzo, aunque esto no es una regla estricta para los *loops*, los hay que simplemente se nota cuando lo hacen, esto ya es como deseé el autor. En mi caso quiero que no se note, y a la vez que transmita algo, así que pensé en relacionarlo con las RRSS y la cantidad de horas que a veces uno se queda mirándolas, se pasa el tiempo y ni te das cuenta. Entonces, en el ámbito de la publicidad, servirse de un efecto reiterativo como este para reflejarlo, me parece muy irónico, original y divertido representarlo de esta manera.

En resumen, el objetivo es investigar y aprender qué es y cómo se hace un *loop* animado 3D realizando todas las fases de un proyecto de animación. Y simultáneamente, concienciar con humor y de una manera artística la cantidad de tiempo que, muchas veces sin darnos cuenta, nos absorben las redes sociales. Por tanto, creo que, para mí, será una forma muy útil de aprender y mostrar mis capacidades adquiridas en el grado, a la vez que me propongo un reto.

Agradecimientos

Quiero agradecer en primer lugar, y, sobre todo, a mi hermano, mi mejor amigo, por estar siempre ahí, escucharme, intentar entenderme y apoyarme, gracias, realmente gracias, te quiero. A mi madre, porque es la otra persona que siempre ha estado ahí, incluso cuando yo pensaba que no había nadie, gracias por apoyarme, y creer en mí, te quiero. A mi padre, por ser la persona que me enseñó los primeros problemas matemáticos e hizo que me gustasen las matemáticas con aquel cuaderno azul y rosa de Mickey Mouse, también por apoyarme y animarme a hacer esta titulación. A mi hermana, aunque a veces un poco distanciadas, eres importante para mí, y me gusta compartir conocimientos y aficiones contigo.

Agradecer a mis amigos, porque ellos también han sido la fuerza que me ha ayudado a conseguir llegar hasta aquí. A Kevin y Saul, por estar ahí en los momentos que me ha hecho falta, por escucharme y ayudar a despejarme cuando lo he necesitado, quedar con vosotros siempre me anima. A Javi Corral siempre amable y bueno, sé que puedo contar contigo, un amigo de toda la vida, aunque nuestros caminos nos hayan separado muchas veces, siempre hemos encontrado la forma de mantenernos en contacto y espero que siga siendo así. A Vicky, pese a que estés algo lejos ahora no significa que no te sienta aquí mismo, desde que te conozco cambiaste mi forma de ver la vida a mejor, y te lo agradezco mucho. A Lilian y su hijo Maxi, por hacerme ver la vida con otros ojos. A Javi García pese a las dificultades, hemos conseguido mantener el contacto y me alegro muchísimo, porque eres el tesoro más grande que me llevo de la carrera, gracias por todas esas tardes tan entretenidas y de risoterapia que tenemos, y por asumir por mí que podía realizar con éxito este trabajo. A Charo, la mejor compañera de trabajo que he tenido, gracias por pensar en mí e invitarme a salir de mi zona de confort. A todo mi grupo de ABP, porque pese a los baches y dificultades, seguimos (aunque indecisos) adelante, y conseguimos lograr nuestros objetivos en mayor o menor medida. Sin crear mal ambiente entre nosotros, y una de las conclusiones que saqué de aquel trabajo es que eso no fue para todos tan fácil como se nos hizo a nosotros ese aspecto, que lo hizo mucho más llevadero. A Lucrecia, por ser tan amable y buena, amar tan desmesuradamente a los animales y enseñarme a hacerlo a mí también. Podéis contar conmigo para lo que necesitéis.

Por último, como no podía ser de otra manera a mis animales, que también son familia. En concreto, agradecer a mis gatos que literalmente son mi sombra en casa, constantemente están conmigo e incluso ellos mismos aparecen cuando no me encuentro bien. Especialmente a mi gata Kely, descansa en paz bonita, me hubiese gustado que hubieses estado conmigo todos

estos años de carrera, pero igualmente haré que estés ahí como el personaje principal del *loop*, en tu honor.

Citas

¿Qué cosa más grande que tener a alguien con quien te atrevas a hablar como contigo mismo?

Cicerón

Ser sincero no es decir todo lo que se piensa, sino no decir nunca lo contrario de lo que se piensa.

André Maurois

Si abordas una situación como asunto de vida o muerte, morirás muchas veces.

Adam Smith

Lo que sabemos es una gota de agua; lo que ignoramos es el océano.

Isaac Newton

Dedicatoria

A mi hermano, a mi madre, a mi padre y a mis animales.

*Sobre todo, a Kely, mi gata que falleció cuando estaba en primero de carrera, y desde entonces,
fue que pensé que acabaría el grado y le dedicaría mi trabajo final.*

Índice de contenidos

Resumen.....	2
Motivación, justificación y objetivo general	3
Agradecimientos	5
Citas.....	7
Dedicatoria.....	8
Índice de contenidos.....	9
Índice de figuras.....	12
Índice de tablas.....	18
1. Introducción.....	19
2. Planificación.....	21
3. Estado del arte.....	23
3.1. Cómo se realiza un loop animado.....	23
3.2. Conceptos básicos.....	23
3.3. Principios de la animación.....	27
3.4. Preproducción.....	37
3.4.1. Brainstorming.....	38
3.4.2. Moodboard	39
3.4.3. Guion.....	40
3.4.4. Diseño de personajes, elementos y entornos (bocetado)	42
3.5. Producción.....	44
3.5.1. Modelado	45
3.5.2. Retopología	53
3.5.3. Rigging y Skinning.....	54
3.5.4. Materiales y Texturizado.....	59
3.5.5. Partículas.....	69

3.5.6.	Iluminación y Sombreado.....	70
3.5.7.	Animación.....	75
3.5.8.	Renderizado	76
3.6.	Postproducción	77
3.6.1.	Composición.....	77
3.6.2.	Corrección de color y Etalonaje	78
3.6.3.	Motion Graphics y VFX.....	78
3.6.4.	Audio	79
3.7.	Antecedentes	80
3.8.	Principales herramientas software	87
4.	Objetivos.....	95
5.	Metodología	97
6.	Herramientas utilizadas.....	104
7.	Preproducción	106
7.1.	Brainstorming.....	106
7.2.	Moodboard	107
7.3.	Guion	108
7.4.	Diseño de personajes, elementos y entorno (bocetado).....	109
8.	Producción.....	114
8.1.	Modelado	114
8.1.1.	Preparación del proyecto	116
8.1.2.	Modelo de la tableta	120
8.1.3.	Modelo de la gata Kely.....	127
8.1.4.	Modelo del escenario.....	140
8.2.	Retopología	141
8.3.	Materiales y Texturizado.....	146
8.3.1.	Modelo de la tableta	146
8.3.2.	Modelo de la gata Kely.....	149

8.3.3. Modelo del escenario.....	155
8.4. Rigging y Skinning.....	156
8.5. Iluminación y Sombreado.....	163
8.6. Animación.....	164
8.7. Partículas.....	170
8.8. Renderizado	172
9. Postproducción.....	174
9.1. Composición y Motion Graphics	174
10. Resultado final.....	177
11. Conclusiones y trabajo futuro	184
Bibliografía	185
Recursos externos utilizados en el desarrollo del <i>loop</i>	192
Lista de Acrónimos y Abreviaturas.....	195
Apéndice I: Planificación semanal en Trello.....	197
Apéndice II: Informe de Toggl con las horas empleadas.....	211

Índice de figuras

Figura 1. Cronograma de tareas en una hoja de cálculo.....	22
Figura 2. Ejemplo del principio Estirar y encoger con una pelota.....	28
Figura 3. Ejemplo de Anticipación. La fase se muestra en el dibujo de la derecha	29
Figura 4. Ejemplo de Puesta en escena con una bailarina en el foco del escenario.....	29
Figura 5. Ejemplo de cómo se va dibujando cada fotograma en Animación directa.....	30
Figura 6. Ejemplo de poses clave en la técnica Pose a Pose	31
Figura 7. Ejemplo de Acción complementaria y superpuesta con las orejas de un perro	32
Figura 8. Ejemplo de Acelerar con una pelota y la línea temporal	32
Figura 9. Ejemplo de Desacelerar con una pelota y la línea temporal.....	32
Figura 10. Ejemplo del principio Arcos con un columpio.....	33
Figura 11. Ejemplo del principio Acción Secundaria con un personaje esperando	34
Figura 12. Ejemplo de Timing con una bola y su movimiento	34
Figura 13. Ejemplo del principio Exageración con los ojos de un personaje	36
Figura 14. Ejemplo del principio Dibujo sólido con un personaje.....	36
Figura 15. Ejemplo del principio Atractivo sin aplicar	37
Figura 16. Ejemplo del principio Atractivo aplicado	37
Figura 17. Pipeline de animación 3D.....	38
Figura 18. Ejemplo de moodboard con anotaciones	40
Figura 19. Ejemplo fragmento de storyboard de la película Peter Pan de Disney.....	42
Figura 20. Ejemplo de Model Sheet del personaje Bart de los Simpson	43
Figura 21. Ejemplo Concept Art de un prop en su Model Sheet.....	44
Figura 22. Ejemplo Concept Art de un escenario en su Model Sheet.....	44
Figura 23. Elementos que forman un polígono en un modelo 3D.....	46
Figura 24. Ejemplo de high poly, mid poly y low poly.....	46
Figura 25. Proyección ortogonal	47
Figura 26. Proyección en perspectiva	47
Figura 27. Ejemplo Primitivas Básicas en Blender.....	48
Figura 28. Ejemplo Box Modelling de personaje low poly	49
Figura 29. Ejemplo de Modelado Poly by Poly.....	49
Figura 30. Ejemplo de Modelado con Nurbs.....	50
Figura 31. Ejemplo Esculpido en 3D	51
Figura 32. Ejemplo Texto en 3D	51

Figura 33. Ejemplo Modelado con Revoluciones 360º	52
Figura 34. Ejemplo Metaballs en Blender	52
Figura 35. Ejemplo de Fotogrametría de una estatua	53
Figura 36. Ejemplo Modelado Paramétrico de un terreno	53
Figura 37. Ejemplo de un modelo y su esqueleto rigging	55
Figura 38. Ejemplo de A pose y T pose.....	56
Figura 39. Ejemplo root joint y sus joints hijas.....	56
Figura 40. Ejemplo rigging facial	57
Figura 41. Ejemplo rigging facial y su interfaz.....	57
Figura 42. Ejemplo distribución de pesos en Blanco y Negro	59
Figura 43. Ejemplo distribución de pesos en Color	59
Figura 44. Código de colores del degradado de pesos en Blender	59
Figura 45. Ejemplo mapas de textura para un tronco de árbol	66
Figura 46. Ejemplo de plantilla de papercraft.....	67
Figura 47. Ejemplo de papercraft montado	67
Figura 48. Ejemplo costuras de Suzanne, mona de Blender	67
Figura 49. Ejemplo de cómo usar la textura de chequeo (izda. errores y dcha. arreglada)	68
Figura 50. Ejemplo unwrap en el espacio UV de Suzanne, mona de Blender.....	68
Figura 51. Ejemplo aplicar texturas a Suzanne, mona de Blender.....	69
Figura 52. Ejemplo fase de render (dos primeras) y fase de simulación (última).....	70
Figura 53. Modelo de iluminación local	72
Figura 54. Imagen a) creada con Radiosity y Ray-tracing y b) solo Ray-tracing.....	73
Figura 55. Captura del primer modelo 3D, un cuerpo femenino realizado en ZBrush	81
Figura 56. Casa de LEGO hecha en 3ds Max.....	83
Figura 57. Cafetería de Apple hacia el interior hecho en 3ds Max	83
Figura 58. Cafetería de Apple hacia el exterior hecho en 3ds Max.....	83
Figura 59, Captura de la animación imitando intro Pixar en 3ds Max	83
Figura 60. Captura animación correr fotograma 11 de una silueta	84
Figura 61. Captura animación andar y diálogo de un personaje propio.....	84
Figura 62. Coche de juguete realizado en Blender	84
Figura 63. Entorno subterráneo inspirado en la película Arthur y los Minimoys	84
Figura 64. Modelo 3D nave industrial	85
Figura 65. Modelo 3D garajes industriales.....	85
Figura 66. Ejercicio de anatomía calavera en 3/4	86
Figura 67. Ejercicio diseño de un escenario	86

Figura 68. Trabajo final diseño de un personaje.....	86
Figura 69. Ejercicio modelado de un Pokémon diseñado por un compañero	86
Figura 70. Reto modelado piruleta de golosinas sin aplicar modificadores	86
Figura 71. Modelado de una flor.....	86
Figura 72. Escultura de Ugly Dog de la película UglyDolls: Extraordinariamente feos	87
Figura 73. Práctica ilustración vectorial de mis gatos	87
Figura 74. Esquema metodología Scrum.....	97
Figura 75. Esquema metodología Kanban.....	98
Figura 76. Organización inicial del flujo de trabajo de este proyecto en Trello.....	101
Figura 77. Recopilación de toda la lista de tareas inicial en Trello por partes.....	101
Figura 78. Ejemplo del tiempo empleado cada día y de la semana en Toggl	102
Figura 79. Ejemplo muestra de las horas totales empleadas en cada tarea en Toggl	102
Figura 80. Moodboard inicial del proyecto realizado con PureRef.....	107
Figura 81. Storyboard del proyecto página 1	108
Figura 82. Storyboard del proyecto página 2	109
Figura 83. Boceto de Kely frente y perfil.....	110
Figura 84. Boceto concept art Kely	110
Figura 85. Boceto de la tableta	111
Figura 86. Bocetos de la app con y sin like en la pantalla de la tableta	112
Figura 87. Fotografía de Kely real para crear la paleta de color	113
Figura 88. Concepts art de Kely, la app con el like y la tableta	113
Figura 89. Comprobar si la malla es correcta y está formada por quads.....	115
Figura 90. Configuración de la comprobación de la malla formada por quads	115
Figura 91. Ejemplo comprobación orientación correcta de las caras del modelo	115
Figura 92. Opción para que se vean redondos o planos los bordes de un objeto	116
Figura 93. Opción In Front para cualquier objeto en Blender	118
Figura 94. Organización de las colecciones en Blender	119
Figura 95. Configuración de la referencia vista frontal de la tableta.....	120
Figura 96. Configuración de la referencia vista trasera de la tableta	121
Figura 97. Vista delantera del objeto inicial de la tableta.....	123
Figura 98. Vista trasera del objeto inicial de la tableta.....	123
Figura 99. Objeto de la pantalla de la app dentro de la tableta	124
Figura 100. Ejemplo del objeto botón de delante de la pantalla.....	125
Figura 101. Objeto cámara de la tableta.....	126
Figura 102. Modelo entero de la tableta con los objetos que lo forman	127

Figura 103. Configuración de las referencias de la gata	128
Figura 104. Blocking del modelo de la gata sin mirror.....	130
Figura 105. Blocking del modelo de la gata completo	130
Figura 106. Referencia forma de Kely real de frente	132
Figura 107. Referencia forma de Kely real de perfil.....	132
Figura 108. Modo escultura: pinceles, simetría y esculpido el modelo de Kely	132
Figura 109. Resto de la lista de pinceles para el modo escultura y objeto ojo visible.....	133
Figura 110. Unión de las partes del cuerpo de la gata y Remesh de la malla.....	134
Figura 111. Modelo de la gata high poly suavizado y con los detalles	135
Figura 112. Comparación de la topología para crear una esfera correctamente	136
Figura 113. Pasos para convertir un cubo en una esfera que esté formada por quads	136
Figura 114. Estudio anatómico de un ojo.....	137
Figura 115. Partes de un ojo por capas.....	137
Figura 116. Esfera modelada con la parte del iris, pupila, esclerótica, etc.....	137
Figura 117. Colocación de los objetos del ojo y la forma que tienen (de perfil y low poly)	137
Figura 118. Modelo final de los ojos de la gata con transparencia.....	138
Figura 119. Configuración de las Nurbs de los bigotes y cejas.....	139
Figura 120. Modelo high poly final de la gata con el Mirror de los bigotes y cejas	140
Figura 121. Modelo del escenario.....	141
Figura 122. Preparar la herramienta Poly build para hacer la retopología.....	142
Figura 123. Preparación de la retopología con los loops de polígonos	143
Figura 124. Retopología completa y dirección de los polígonos.....	144
Figura 125. Retopología completa sin el Mirror	144
Figura 126. Comparación de los modelos high poly, low poly y con la proyección.....	145
Figura 127. Comparación en modo alámbrico de la topología	146
Figura 128. Texturas para la animación de la pantalla de la tableta	147
Figura 129. Configuración material cristal de la tableta	148
Figura 130. Configuración material carcasa de la tableta.....	148
Figura 131. Configuración material App Catisgram de la tableta	149
Figura 132. Configuración material de los bigotes de la gata	151
Figura 133. Configuración material de la córnea del ojo de la gata	151
Figura 134. Configuración material del iris del ojo de la gata.....	152
Figura 135. Configuración material de la pupila del ojo de la gata.....	152
Figura 136. Modelo de la gata con las costuras marcadas y el Unwrap a la izquierda.....	153
Figura 137. Comprobación de errores con la textura Checker UV	154

Figura 138. Espacio de trabajo Texture Paint con el modelo de la gata pintado.....	155
Figura 139. Configuración material del escenario	156
Figura 140. Referencia estudio del esqueleto de un gato	157
Figura 141. Rigging de la gata vista perfil.....	158
Figura 142. Referencia estudio anatómico muscular de un gato	159
Figura 143. Muestra del pesado de vértices para el hueso del muslo de la gata	161
Figura 144. Dependencia de objetos y huesos de la gata nombrados en el Outliner	162
Figura 145. Disposición de luces en la escena y configuración de la luz solar.....	164
Figura 146. Configuración de las luces que forman el triángulo de luces.....	164
Figura 147. Configuración de la cámara.....	165
Figura 148. Referencia gato Sphynx sentado de perfil	166
Figura 149. Referencia gato Sphynx sentado de frente	166
Figura 150. GIF de un gato real usando un móvil con la pata	167
Figura 151. Panel Graph Editor para copiar transformaciones de un hueso a otro	169
Figura 152. Animación de la gata	170
Figura 153. Animación de la pantalla de la tableta y final de la gata.....	170
Figura 154. Sistema de partículas: los corazones que salen de la pantalla	171
Figura 155. Configuración del sistema de partículas de los corazones.....	172
Figura 156. Configuración del Render.....	173
Figura 157. Configuración Output.....	173
Figura 158. Aplicación Adobe Express en el menú.....	174
Figura 159. Edición de vídeo con Adobe Express, añadiendo texto	175
Figura 160. Acceder a la función Convertir MP4 a GIF de Adobe Express	176
Figura 161. Configuración para convertir MP4 a GIF con Adobe Express.....	176
Figura 162. Muestra de resultado final como imagen con el enlace al vídeo	177
Figura 163. GIF de un gato real para compararlo con el resultado final.....	178
Figura 164. Foto de Kely real para comparar con el resultado final	182
Figura 165. Muestra de resultado final como imagen con las partículas	182
Figura 166. Lista de los vídeos del proceso de Producción con el enlace al vídeo	183
Figura 167. Inicio de la planificación del sprint 1 en Trello.....	198
Figura 168. Fin de la planificación del sprint 1 en Trello.....	198
Figura 169. Inicio de la planificación del sprint 2 en Trello.....	199
Figura 170. Fin de la planificación del sprint 2 en Trello.....	199
Figura 171. Inicio de la planificación del sprint 3 en Trello.....	200
Figura 172. Fin de la planificación del sprint 3 en Trello.....	200

Figura 173. Inicio de la planificación del sprint 4 en Trello.....	201
Figura 174. Fin de la planificación del sprint 4 en Trello.....	201
Figura 175. Inicio de la planificación del sprint 5 en Trello.....	202
Figura 176. Fin de la planificación del sprint 5 en Trello.....	202
Figura 177. Inicio de la planificación del sprint 6 en Trello.....	203
Figura 178. Fin de la planificación del sprint 6 en Trello.....	203
Figura 179. Inicio de la planificación del sprint 7 en Trello (Parte 1).....	204
Figura 180. Inicio de la planificación del sprint 7 en Trello (Parte 2).....	204
Figura 181. Fin de la planificación del sprint 7 en Trello (Parte 1).....	205
Figura 182. Fin de la planificación del sprint 7 en Trello (Parte 2).....	205
Figura 183. Inicio de la planificación del sprint 8 en Trello (Parte 1).....	206
Figura 184. Inicio de la planificación del sprint 8 en Trello (Parte 2).....	206
Figura 185. Fin de la planificación del sprint 8 en Trello.....	207
Figura 186. Inicio de la planificación del sprint 9 en Trello (Parte 1).....	207
Figura 187. Inicio de la planificación del sprint 9 en Trello (Parte 2).....	208
Figura 188. Fin de la planificación del sprint 9 en Trello.....	208
Figura 189. Inicio de la planificación del sprint 10 en Trello.....	209
Figura 190. Fin de la planificación del sprint 10 en Trello (Parte 1).....	209
Figura 191. Fin de la planificación del sprint 10 en Trello (Parte 2).....	210
Figura 192. Informe de Toggl con las horas empleadas (Parte 1).....	211
Figura 193. Informe de Toggl con las horas empleadas (Parte 2).....	211
Figura 194. Informe de Toggl con las horas empleadas (Parte 3).....	212
Figura 195. Informe de Toggl con las horas empleadas (Parte 4).....	212
Figura 196. Informe de Toggl con las horas empleadas (Parte 5).....	213
Figura 197. Informe de Toggl con las horas empleadas (Parte 6: final).....	213

Índice de tablas

Tabla 1. Lista de Acrónimos y Abreviaturas empleados en el documento	195
--	-----

1. Introducción

Este proyecto consiste en realizar un *loop* animado 3D en estilo *cartoon*. Para explicarlo por partes, será una animación, es decir una secuencia de dibujos que se reproducen uno detrás de otro creando la ilusión de movimiento de la escena; el estilo *cartoon* se refiere a la apariencia que tendrá de dibujo animado, pero en 3 dimensiones en vez de 2; y, el *loop* se refiere a que esa animación se repetirá de principio a fin de forma continua e indefinidamente. Además, mediante este proyecto visual, se pretende concienciar del tiempo que una persona puede pasar usando las redes sociales en el móvil sin darse cuenta.

Loop, traducido como bucle, en diseño y animación es un efecto visual que se utiliza en distintos medios y para fines muy diversos. Como, por ejemplo, los carteles digitales publicitarios de la calle o estadios que muestran un anuncio detrás de otro y los repiten una y otra vez. En las redes sociales también son muy solicitados, se usan para realizar las animaciones de los *stickers*, emojis o sin ir más lejos, es como el formato gif. Otro ejemplo, es en los videojuegos para las animaciones de elementos del menú para que sea más atractivo a la vista, o para los fondos del juego llamen la atención lo justo para que visualmente resulte realista. Puede que, a menudo, no sea algo en lo que alguien se fije inicialmente, pero la animación del viento, el movimiento de las hojas de los árboles, la lluvia, cascadas, etcétera, emplean este efecto para hacer esas animaciones de los escenarios. Con esto consiguen que el juego sea mucho más inmersivo al reflejar acciones que en la vida real pasarían a tu alrededor en esos escenarios. Asimismo, en las pantallas de carga de los videojuegos o webs, para que el usuario no piense que ha colapsado. También, para animar los logos y aperturas de las empresas y/o marcas para los vídeos de sus canales en medios sociales. Incluso hoy en día los artistas lo emplean para realizar una presentación rápida de sus trabajos en los currículums o sus páginas web de presentación personal.

La animación es un proceso para dar la sensación de movimiento mediante una sucesión de imágenes, dibujos u otros objetos inanimados, como, entre otros materiales, la plastilina. Es una ilusión óptica que se puede conseguir con numerosas técnicas como: los dibujos animados por fotogramas, *stop motion*, *cut-out* o recortes, *claymation* (con arcilla o plastilina), rotoscopia, animación por computador, o *time lapse* entre otras. Existe un conjunto de principios conocidos como *Los doce Principios básicos de la animación* de Disney, publicados por los animadores Ollie Johnston y Frank Thomas en su libro *The illusion of life: Disney Animation*. Actualmente, la animación no sólo se usa como método de entretenimiento, sino que se ha abierto paso en

múltiples disciplinas ya sea por ejemplo en educación e investigación para que el contenido sea más visual y se entienda mejor; en marketing digital y publicidad; en medicina, para ayudar a los pacientes que necesitan rehabilitación, o para que los médicos y cirujanos sean capaces de practicar con modelos 3D sus avances y/o enseñanzas nuevas; y otras aplicaciones. Hay diversos estudios de animación y muchos muy conocidos, que se dedican a la producción en masa de estas técnicas, pero también existen otros oficios relacionados que las usan como, por ejemplo, para los mencionados anteriormente o para videojuegos.

Los gráficos en 3D por ordenador, son una forma en la que se puede representar, en tres dimensiones, datos geométricos almacenados en el ordenador y proyectarlos en una pantalla, que es bidimensional. La fotografía y la escultura son artes muy parecidos a los gráficos tridimensionales. Aunque todavía existen debates sobre si se considera o no arte el producido por computadora, lo cierto es que muchas industrias lo usan, y cada vez más se está abriendo un hueco en el mercado en diferentes formatos, como en cine y videojuegos.

Luego, por una parte, están los dibujos animados, conocidos con el anglicismo *cartoon*, son una categoría de las obras de animación, estas normalmente son realizadas en 2 dimensiones con una técnica que consiste en una secuencia de fotogramas dibujados que crearán esa ilusión de movimiento. Por otra parte, el estilo *cartoon*, es una de las maneras en la que se puede representar nuestro arte, que se basa en los dibujos animados y las reglas que se siguen en ellos. Es decir, que se basa en dibujo animado 2D, las expresiones y exageraciones de este, y lo traslada al tipo de arte que se esté creando, en este caso a un modelo 3D animado. Es más, se utilizan las técnicas de color que tiene: los colores planos, pocas texturas e incluso, el line art se logra obtener si el artista lo desea para obtener esa apariencia que recuerda a los dibujos animados.

Por último, la animación, sea del estilo que sea y de cualquier gráfico, consigue transmitir mensajes en numerosos aspectos, y/o para varios propósitos, ya sea, difundir información, enseñar, emocionar, asustar, aterrorizar, ilusionar, vender un producto, recordar o para algo similar al espectador. En este proyecto se pretende concienciar de que a veces sin caer en ello, las personas pasan demasiadas horas mirando el móvil en redes sociales. Aprovechando el hecho de que un bucle se repite indefinidamente, se ha pensado en que de modo irónico puede representar el tiempo que se pierde en ellas sin darse cuenta. Existen distintas corporaciones, empresas y asociaciones que se comprometen a intentar enseñar y difundir este mensaje, y otros, para lograr una digitalización sostenible, responsable, segura, no discriminatoria y universal para los ciudadanos. Como el proyecto *Levanta la Cabeza* de Atresmedia.

2. Planificación

En este capítulo se explicará la planificación temporal que pretende seguirse para la realización de este trabajo. Esta planificación es muy importante para poder llevar a cabo un trabajo de esta magnitud, ya que el tiempo es un factor limitante para su finalización y entrega. Además, de este modo, cuando se complete el proyecto se procederá a analizar, junto a los resultados, el desarrollo total y si se ha podido o no seguir esta planificación. También, al hacer esta planificación inicial, se creará una guía de los pasos a seguir y que tareas deben realizarse en cada periodo de tiempo.

Se ha realizado un cronograma de tareas en una hoja de cálculo con: los puntos de la memoria, reuniones con el tutor, revisión del trabajo final y las fechas importantes del TFG como solicitar la defensa, entregar la memoria y la asignación del día de la defensa.

En la tabla de la hoja de cálculo que se ve a continuación, se ha colocado en la primera columna las actividades que engloban todas las tareas que deben realizarse para lograr finalizar el proyecto, incluidas las fechas importantes como se ha mencionado anteriormente. Se ha estimado que el trabajo de cada semana, de lunes a viernes, debe ser de aproximadamente 40 horas, que son 8 horas al día. Para cumplir las 300 horas que abarca la asignatura de TFG de Ingeniería Multimedia. Si se cumplen las horas diarias, teniendo en cuenta las horas que debe comprender el trabajo, este debería poder realizarse en más o menos 8 semanas o lo que es lo mismo en 38 días. También se ha tenido en cuenta para calcular el tiempo que conllevará un margen de error que pueda producirse por eventos, circunstancias u otro tipo de impedimentos y retrasos. Por eso, en vez de solo 2 meses, se han tenido en cuenta 4 meses. En las siguientes columnas se ha dividido el tiempo en semanas. Cada semana cuenta como 5 días laborables y, alrededor de 4 semanas se cuenta como un mes, para saber más o menos el periodo de tiempo que debe abarcar cada actividad y cuándo debería estar finalizada. Y se ha marcado en cada una de las actividades, por bloques de contenido (que se ha tenido en cuenta como similar), el periodo en el que deben estar activas esas tareas, y cuando deberían empezar y terminar como fechas límites. Por último, como en un principio no se sabía en que convocatoria iba a ser posible entregarlo, no se puso nombre a los meses, ni fecha a las semanas, así que en la ilustración se refiere a cada mes como *Mes "X"* y a las semanas de la misma manera. Por último, hay que aclarar que las tareas nombradas como Reunión con el/la tutor/a inicial o parcial se han colocado más o menos donde se cree que es el momento idóneo de obtener feedback de como avanza el proyecto, y se hará un día de esa semana conforme le venga mejor al/a la tutor/a.

ACTIVIDADES (40 horas semanales)	Mes 1				Mes 2				Mes 3				Mes 4			
	Semana 1	Semana 2	Semana 3	Semana 4	Semana 5	Semana 6	Semana 7	Semana 8	Semana 9	Semana 10	Semana 11	Semana 12	Semana 13	Semana 14	Semana 15	Semana 16
Reunión inicial tutor/a																
Motivación, Justificación y Objetivo general																
Agradecimientos																
Citas																
Dedicatoria																
Planificación																
Reunión tutor/a revisión memoria parcial																
Introducción																
Estado del Arte																
Objetivos																
Metodología																
Reunión tutor/a revisión memoria parcial																
Preproducción																
Producción																
Postproducción																
Reunión tutor/a revisión memoria parcial																
Resultado final																
Revisión de propuesta de cambios, indicaciones del tutor/a, etc																
Conclusiones																
Referencias																
Reunión tutor/a revisión memoria final																
Solicitar Defensa																
Entrega memoria final																
Defensa																

Figura 1. Cronograma de tareas en una hoja de cálculo
(Fuente propia: realizado en una hoja de cálculo)

Más adelante, en el capítulo de *Metodología*, de este mismo documento, se explicarán y desglosarán estas actividades en tareas para llevar un mejor control de la planificación. También, se explicarán las herramientas utilizadas para la gestión del proyecto y del tiempo empleado en el trabajo para llevar a cabo esta planificación.

3. Estado del arte

En este capítulo se explicará la parte teórica de qué es y cómo se realiza un loop, y de las fases que existen en los proyectos de animación, que son los que se necesitarán para realizarlo. En definitiva, se estudiará y documentará toda la información obtenida que pueda resultar útil para el desarrollo de este proyecto. Se descubrirá el contexto, los problemas, cómo lo abordan con posibles soluciones, que factores intervienen y los recursos necesarios.

3.1. Cómo se realiza un loop animado

Para saber cómo se realiza un loop animado, lo primero que se debe hacer es investigar qué es un loop animado. Un loop animado no es más que una animación que se repite en bucle. Por tanto, primero habrá que realizar la animación y hacer que el inicio y el final de esta sean el mismo, así se puede conseguir el efecto visual de que al reproducirla una y otra vez se cree la ilusión de que esa animación no acaba nunca. Pero esto es más difícil de lo que parece a simple vista, ya que hay que calcularlo muy bien para que no se note raro y no se vea un corte que deje identificar que es la misma animación repetida infinitamente. Entonces, la diferencia principal con una animación corriente es que hay que hacer un buen diseño del bucle, y cuadrar bien los tiempos de cada imagen o frame, para que el espectador no pueda localizar el comienzo y el final de la animación. Así pues, a continuación, se explicará qué es una animación, cómo se hace y luego que se tendría que hacer para hacer un bucle perfecto.

3.2. Conceptos básicos

Para comprender mejor qué es la animación, es necesario saber unos conceptos básicos que rigen la animación. No son muchos, pero ayudarán a comprender todo lo que vendrá más tarde. Primero, se hablará del concepto de persistencia retiniana y del fenómeno Phi, de la diferencia entre fotograma y FPS y, por último, de los conceptos escena, secuencia, acto y plano.

Se llama **persistencia retiniana**, a un supuesto fenómeno visual de la vista humana, que permite que una imagen que se vea permanezca, antes de que desaparezca completamente, en la retina una décima de segundo. Todo esto permite que todo lo que ve, en este caso un humano, sea tratado como una secuencia de imágenes continua, gracias a la que puede calcular la dirección y velocidad a la que va un objeto que se está desplazando. Además de este, también existe el

fenómeno Phi (φ), que es uno de los principios de la Gestalt. Esta ilusión óptica hace creer que unas imágenes que en realidad están fijas, se mueven, esto demuestra que el cerebro humano es capaz de sintetizar y llenar espacios vacíos para simplificar las cosas y que se puedan entender mejor, mostrando una realidad que no es del todo cierta. Sin estos fenómenos relacionados con la vista, se vería todo como una secuencia de imágenes independientes y estáticas. El ojo humano puede ver aproximadamente 10 imágenes por segundo, estas quedan superpuestas en la retina y el cerebro realiza una simplificación de todos los estímulos que le llegan, uniéndolas entre ellas para crear una sola acción móvil sencilla que pueda entender. Es por esto, que se puede disfrutar del básicamente todo el entretenimiento digital, ya que es lo que se usa en cine para proyectar siempre más de 10 imágenes por segundo (normalmente entre 24 o 25) y el cerebro es el que crea la ilusión del movimiento.

Por otra parte, se llama **fotograma** (o *frame*) a cada una de las imágenes estáticas de las que se compone una película. Gracias al fenómeno anterior, se produce la sensación de movimiento cuando se reproducen a una determinada velocidad. Es decir, un fotograma es la unidad mínima de registro en una animación. Sin embargo, los **FPS** o fotogramas por segundo, son el número de fotogramas que se reproducen en un segundo. Este número señala la velocidad e influye en el tiempo. Actualmente, el estándar que se usa en el cine y televisión es de 24-120 FPS. También depende de la resolución, a mayor resolución necesitará más FPS para que sea imperceptible el engaño. En animaciones en tiempo real, como los videojuegos o portales de vídeo, se utilizan ratios entre 30-120 FPS (normalmente a 60 FPS se consigue un resultado óptimo).

En comunicación audiovisual, una **escena**, también llamada secuencia mecánica o secuencia de producción, es una unidad de espacio y tiempo, es decir, un fragmento de la historia que transcurre de forma ininterrumpida dentro del mismo escenario y plano. Un conjunto de escenas forma una unidad narrativa a la que se llama **secuencia** o secuencia dramática. Varias secuencias forman un **acto**, con tres actos se consigue contar una historia convencional. Estos actos serían: la introducción, el nudo o desarrollo y el desenlace.

En proyectos audiovisuales, como cine, animación, un corto, un video, un documental e incluso para fotografía, existen distintos tipos de **plano** en los que se puede mostrar una escena. A continuación, se explicarán algunos tipos de plano que existen.

Según el encuadre:

- **Gran Plano General.** En él apenas se distinguen las figuras humanas.
- **Plano General.** Se utiliza para contextualizar la narración.

- **Plano de Conjunto.** En este plano se sitúa a los sujetos en una zona más cercana a la cámara que al resto de personajes u objetos. La cámara suele acercarse lentamente a los sujetos principales, puede encuadrar la figura entera o casi entera de los sujetos. Es más cerrado que el Plano General.
- **Plano Entero o Figura.** En plano donde se encuadra perfectamente al sujeto. El límite superior de la imagen coincide con la cabeza y el inferior con los pies. En ocasiones puede que haya un pequeño hueco desde el límite superior a la cabeza y/o de los pies al límite inferior.
- **Plano Americano** o $\frac{3}{4}$ (tres cuartos). El encuadre va desde la cabeza, puede haber hueco por encima de la cabeza, hasta justo por debajo de las rodillas.
- **Plano Medio o Normal.** El encuadre muestra al/a los personaje/s de la cabeza a la cintura. Este plano permite transmitir relaciones entre los sujetos y que estos puedan usar sus manos para gestualizar.
- **Plano Medio Corto.** La figura humana se encuadra desde la cabeza hasta mitad del torso o debajo del pecho.
- **Primer Plano.** El plano muestra la cabeza hasta los hombros, dándole más detalle al rostro. Se usa para mostrar las emociones y generar intimidad.
- **Primerísimo Primer Plano.** Se capta el rostro desde la parte superior de la cabeza hasta el mentón. La cara del personaje es lo más importante, y hace que sea muy expresiva enfatizando más que el plano del anterior punto.
- **Plano Detalle.** Permite mostrar objetos o partes del cuerpo de algún ser vivo, para centrarse en ellos. Es un plano muy narrativo, se usa para darle importancia a ese objeto o parte del ser vivo, en un momento dado, porque intenta transmitir su gran valor en la historia.

Según el ángulo de la cámara:

- **Plano Cenital o Zenital.** Hoy en día también se le relaciona con el plano aéreo. Se realiza desde la parte superior de los sujetos u objetos. Tiene un ángulo de 90º totalmente perpendicular al suelo.
- **Plano Picado.** La cámara se sitúa por encima de los ojos inclinada hacia el suelo en un ángulo de más o menos 45º para encuadrar al sujeto u objeto.
- **Plano Contrapicado.** La cámara se sitúa por debajo de los ojos y se inclina hacia el techo para encuadrar al sujeto u objeto.

- **Plano Nadir.** Es el contrario al Zenital. La cámara se sitúa en el suelo mirando hacia el techo totalmente por debajo del sujeto u objeto. Crea más dramatismo y dinamismo a la imagen.
- **Plano en Escorzo.** Es usualmente utilizado para grabar conversaciones entre dos sujetos. Se sitúa la cámara detrás de uno de los hablantes dejando ver la parte trasera de su cabeza y al sujeto que habla enfrente.
- **Plano Perfil o Plano Lateral.** Se coloca la cámara en un lateral del sujeto u objeto mostrando su perfil.
- **Plano Frontal.** La cámara se sitúa frente al sujeto u objeto sin ningún tipo de ángulo, es el ángulo neutro.
- **Plano Dorsal.** La cámara se sitúa detrás del sujeto y muestra lo que este está viendo. Se puede ver al actor, pero este no mira a cámara.
- **Plano Holandés.** La cámara se inclina sobre su propio eje unos 45°. Se usaba mucho en cine expresionista.

Según el punto de vista:

- **Plano Objetivo.** La cámara es invisible para los personajes, es el plano “normal”.
- **Plano Subjetivo.** La cámara muestra el punto de vista de un personaje, como si el espectador se metiese en su piel. También puede ser desde el punto de vista de un objeto, en este caso se llamaría **Plano Objetual**.

Otros tipos de plano:

- **Maestro o El Plano Master.** Cuando se graba un plano muy largo y sin cortes. Se usa mucho en televisión, en cine para las escenas de acción o con muchos extras. Suele utilizar planos generales porque abarcan más porción de la escena. A la hora de montar la grabación en postproducción consigue tenerlo todo más a mano.
- **Plano Contraplano.** Se usa para conversaciones entre los personajes. Se graba cada vez al sujeto que habla en ese momento y después al otro cuando hable con el mismo plano en dirección opuesta.
- **Plano Múltiple o Pantalla Dividida.** Se muestran dos o más planos simultáneamente que tengan algún tipo de relación entre ellos.
- **Plano de Fotograma Congelado.** Se produce cuando en la sala de montaje se congela un fotograma para alargar el tiempo de este y poder sincronizar la grabación con el audio, por ejemplo.

- **Plano Recurso.** Se utiliza para dar más dinamismo. Se deja el audio y se van mostrando intercaladamente al personaje que está hablando y detalles de los gestos de sus manos, su entorno, etc. Se usa mucho en documentales de investigación y noticias.

3.3. Principios de la animación

Walt Disney fue uno de los pioneros de la animación, y aunque ya haya avanzado la tecnología y los procesos de una forma tan considerable, aun ayudan los descubrimientos y técnicas que usaban antes. Él contrató a un especialista en la animación llamado Don Graham para que diera clases a todo el equipo de animadores, y así consiguió que las animaciones fuesen más fluidas y tuviesen un gran nivel de calidad. Estas clases ayudaron a los artistas a darse cuenta de la importancia que tenía usar la naturaleza como referencia, prestar atención a los movimientos de los seres vivos y analizar cada una de las acciones que realizaban. Aunque no es tan fácil como memorizarlas y ponerse a trabajar, dependiendo de cada animación se deben tener en cuenta unos factores u otros.

En la época dorada de la animación, entre 1928 a 1973, coincidiendo con la era dorada de Hollywood, la productora Walt Disney Animation se hizo un renombre con grandes films como Blancanieves, Pinocho, Dumbo o Bambi entre otros, esto hizo que la compañía se hiciese muy famosa. Alrededor de 1950 a medida que los intereses de la compañía se extendían, Disney empezó a delegar funciones en sus colaboradores más cercanos. Así es cómo nació la figura de *Los nueve ancianos de Disney*. Todos habían sido animadores y hoy en día aún siguen dejando su huella tanto en temas administrativos como en la animación. Ellos establecieron los 12 principios básicos de la animación, que más tarde gracias a dos de ellos, Ollie Johnston y Frank Thomas, darían a conocer públicamente en 1981 con su libro *The illusion of life: Disney Animation*. El objetivo de estos principios es que las animaciones sean más naturales, realistas y más agradables visualmente para el ojo humano. Algunos de estos se basan en la física y otros en años de estudio del diseño, dibujo, la fotografía o el cine. Aunque fueron concebidos por la animación tradicional dibujada a mano, se aplican de igual forma para la animación digital, el *stop motion* o la animación en 3D.

A continuación, se explicarán brevemente estos 12 principios de la animación:

1. Estirar y encoger (*Squash and stretch*)

Este es uno de los principios más importantes, ya que trata de otorgar a las entidades de peso y volumen dotándolos de elasticidad y flexibilidad, también ayuda a indicar de qué material están

hechos. El volumen debe quedar inalterable cuando se estire o se encoja, no es posible que de repente tenga más volumen o menos, sino siempre el mismo, pero deformándolo. Se puede usar para indicar una dirección de la acción y aplicarse a objetos más simples o incluso geométricos, como el del dibujo de la pelota que se ve a continuación, o a algo mucho más complejo, como la musculatura de un cuerpo humano. Este efecto también puede usarse de una forma mucho más exagerada para dar efecto cómico.

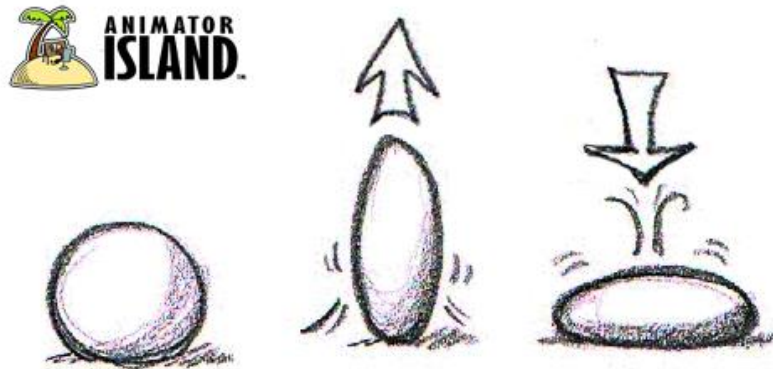


Figura 2. Ejemplo del principio Estirar y encoger con una pelota
(Fuente: <https://www.animatorisland.com/say-no-to-stickmen/sandsball/>)

2. Anticipación (*Anticipation*)

Las acciones en animación suelen dividirse en tres partes, la preparación del movimiento, la acción y lo que precede a la acción. Se llama anticipación a la primera de las partes. Se utiliza para captar la atención del espectador y darle una idea de lo que está a punto de ocurrir. Ayuda a entender las dimensiones físicas de la entidad que va a realizar la acción. Dependiendo de lo que va a ocurrir (la acción en sí), el periodo de tiempo de la anticipación variará. Un ejemplo de esto sería cuando vas a batear una pelota en beisbol. La anticipación sería cuando hechas el bate hacia atrás para coger impulso, la acción golpear la pelota y lo que precede sería que el bate se mueve un poco más adelante una vez la golpea, que sería la consecuencia de la acción. A veces en animación también se usa como efecto cómico o para dar más carisma a un movimiento.

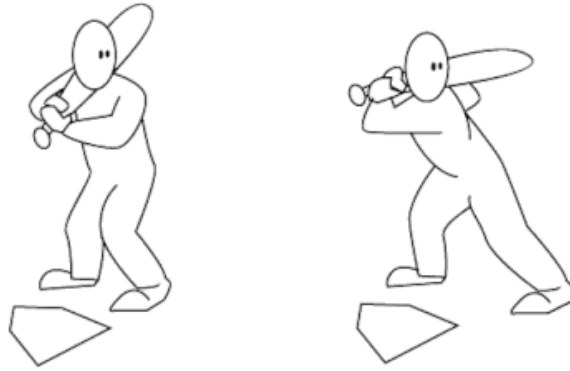


Figura 3. Ejemplo de Anticipación. La fase se muestra en el dibujo de la derecha
(Fuente: <http://katrina-ictx-animation.weebly.com/principles-of-animation.html>)

3. Puesta en escena (Staging)

Consiste en presentar la escena para que sea clara y fácil de entender. Busca centrar la mirada del espectador hacia el punto de interés. Si ocurren demasiadas cosas a la vez, el espectador no sabría en qué debe fijarse y la acción quedaría eclipsada. Es importante también no contradecir la acción con otras u otros elementos visuales. También puede ayudar a transmitir una sensación concreta en una escena. La dirección de la luz de una escena y la composición (la ley de los tercios, la dirección de la mirada, el encuadre, los colores, el ángulo de la cámara, el desenfoque, etc.) tienen mucho que ver en esto ya que transmiten cómo es la situación que están viviendo los personajes y lo que sienten.



Figura 4. Ejemplo de Puesta en escena con una bailarina en el foco del escenario
(Fuente: en la cabecera de la página <https://darvideo.tv/dictionary/staging/>)

4. Animación directa y Pose a pose (Straingt Ahead Action and Pose to Pose)

Son dos técnicas compatibles entre sí que se usan en animación. Ambas se utilizan de forma combinada.

La primera, **animación directa** consiste en que como ya se ha mencionado anteriormente, se sabe que para que el ojo humano perciba bien la acción del movimiento animado, esta debe

constar de al menos 24 FPS. Pues el animador dibujaría cada pose de forma lineal y continua, sin saltar a otra pose que vaya a suceder mucho más adelante. Esta técnica a veces tiene algunas desventajas como que el animador pierda la perspectiva de la entidad y se vea más grande o pequeña de lo que debería ser, en algunos *frames*; o que la acción no termine en el lugar en el espacio donde debería empezar la siguiente acción. Pero la primera desventaja, para la animación en 3D, ya no es un problema, ya que no dibujas la entidad cada vez, sino que modificas el modelo que tienes para cada fotograma. Esta técnica deja más libertad de creatividad al animador, y hace que se vea más fluida y natural. Normalmente, es una buena técnica para usar en explosiones, animación del agua, del fuego, del pelo, o elementos de este tipo.

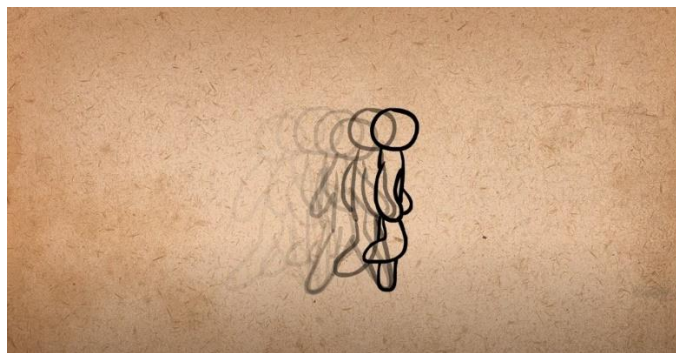


Figura 5. Ejemplo de cómo se va dibujando cada fotograma en Animación directa
(Fuente: captura del minuto 0:20 del video https://www.youtube.com/watch?v=v8quCbt4C-c&ab_channel=AlanBeckerTutorials)

La segunda técnica, **pose a pose**, consiste en realizar los dibujos de las poses clave (las más importantes) que marcan la acción e ir rellenando la transición de una a otra con el resto de los dibujos. Este proceso de rellenar las poses faltantes se llama *twining*, y consiste en que primero dibujas las siguientes poses más importantes que predecirían a la pose clave, luego los siguientes, así hasta rellenar todos los fotogramas necesarios para la acción. En la animación digital, ya sea en 3D o 2D, esta técnica ha tenido mucho más brillo, ya que el animador solo tiene que posar el modelo en los *keyframes* (fotogramas clave) y el ordenador complementará las transiciones de uno a otro. Si en algún momento, necesita retocar algún fotograma en concreto, podrá hacerlo fácilmente. Esta técnica asegura que la entidad no vaya a necesitar demasiados *frames* para esa acción, o que llegue al lugar en el espacio indicado para realizar la siguiente acción, o que no haya cambios de volumen en la entidad. Es más organizada que la anterior, pero deja menos libertad para la creatividad del animador.

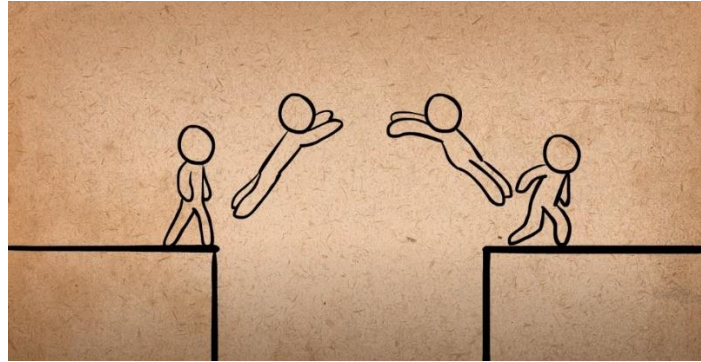


Figura 6. Ejemplo de poses clave en la técnica Pose a Pose
(Fuente: captura del minuto 0:43 del vídeo https://www.youtube.com/watch?v=v8quCbt4C-c&ab_channel=AlanBeckerTutorials)

5. Acción complementaria y Acción superpuesta (*Follow Through and Overlapping Action*)

Estos dos principios están relacionados con el principio de la inercia, que es la propiedad que tienen los cuerpos de permanecer en su estado de reposo o en movimientos relativos.

La **acción complementaria** muestra que los distintos elementos de una entidad deben seguir su movimiento incluso unos segundos después de que esta finalice su acción. Recordando el ejemplo del principio de anticipación, el de batear una pelota de beisbol, la acción complementaria sería la parte del movimiento que hay después de golpear la pelota. Tanto el brazo del bateador como el bate seguirían el movimiento un poco más en la misma dirección en la que batearon la pelota, y luego volverían hacia atrás, al punto en el que estarían en su estado de reposo.

Por su parte, la **acción superpuesta**, sería lo que les ocurre a otros elementos a causa de otra acción. Es decir, cómo afectaría la acción de la entidad a los distintos elementos que formen parte de ella. Estos elementos podrían ser: el pelo, la ropa, una cuerda que cuelga de la entidad, las extremidades de un personaje, etc. Por ejemplo, si se anima a un personaje con el pelo largo corriendo, al frenar se vería como el pelo sigue la dirección en la que estaba corriendo el personaje incluso unos segundos después hasta que llegue a su estado en reposo. El pelo sigue al cuerpo, pero con su propio ritmo.

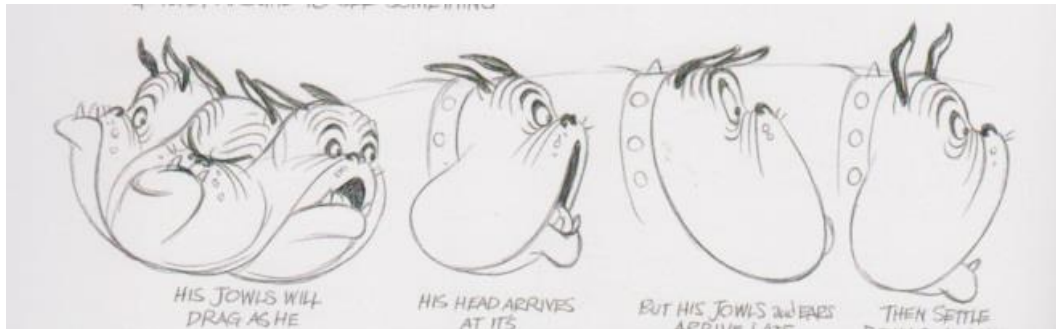


Figura 7. Ejemplo de Acción complementaria y superpuesta con las orejas de un perro
(Fuente: <https://animation2012.weebly.com/follow-through--overlapping-action.html>)

6. Acelerar y desacelerar (Slow In and Slow Out)

En la naturaleza, ninguna entidad entra en movimiento o en su estado de reposo de forma espontánea, sino que puede iniciar un movimiento desde la pose original (**aceleración**), o ir cada vez más despacio hasta llegar a la pose deseada (**desaceleración**). Para lograr este efecto se añaden más poses clave al principio y al final de una acción. Si, por ejemplo, en una escena una entidad lleva una velocidad uniforme, normalmente se dejan unos instantes al inicio y al final de su paso por la escena sin que se vea a la entidad, dando a entender que ya llevaba esa velocidad desde antes de entrar en el encuadre y sigue teniéndola al salir de él.

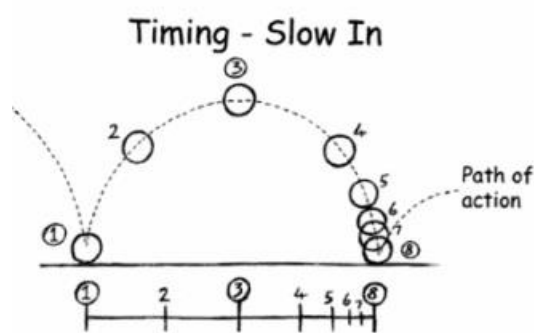


Figura 8. Ejemplo de Acelerar con una pelota y la línea temporal
(Fuente: <https://animation2012.weebly.com/slow-in--slow-out.html>)

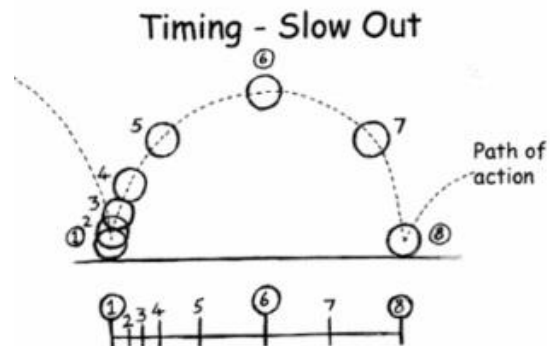


Figura 9. Ejemplo de Desacelerar con una pelota y la línea temporal
(Fuente: <https://animation2012.weebly.com/slow-in--slow-out.html>)

7. Arcos (Arcs)

En el mundo real los movimientos predominantes suelen seguir una trayectoria circular o curva, y en esto se basa este principio. Igual que en diseño los objetos con curvas resultan más agradables a la vista, en animación pasa lo mismo, y los movimientos con trayectorias curva se ven más naturales y fluidos. Al igual que en diseño e ilustración los objetos puntiagudos o con

líneas rectas suelen usarse para transmitir peligro, materiales rígidos, rugosidad, o inquietud; y los objetos formados por arcos o círculos transmiten tranquilidad, suavidad, o caos.

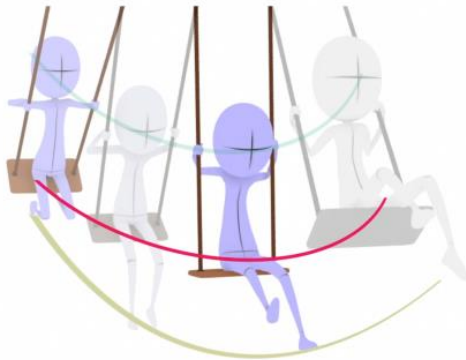


Figura 10. Ejemplo del principio Arcos con un columpio
(Fuente: <https://iwanttobeanimator.wordpress.com/tag/arcs/>)

8. Acción secundaria (*Secondary Action*)

Como su nombre indica, trata de agregar una acción extra a la acción principal, para dar más valor o mayor peso visual a la acción de la entidad. Esta acción debe ser coherente siempre con la acción principal y la escena. Si en algún momento la acción secundaria opaca a la acción principal es mejor prescindir de ella. Lo más interesante de este principio, es que, aunque es una acción con menos importancia, muchas veces es imprescindible para comunicar al espectador el estado de ánimo de un personaje, su personalidad o los sentimientos al público. A diferencia de las acciones complementarias y superpuestas estas no están ligadas al movimiento de la entidad, no son las consecuencias de la acción principal, por tanto, no deben confundirse con estas. En cambio, esta acción tiene que ver con el papel que desempeña y lo que representa la entidad. Para que quede más claro, sería como la típica escena de la risa de una bruja. La acción principal sería la risa, el movimiento de los labios y la boca; las acciones complementarias y superpuestas serían los movimientos de la cabeza y músculos de la cara; y la acción secundaria podría ser que se viese el movimiento de los dedos de la mano izquierda acariciando los dedos de la mano derecha, y viceversa, que le aportaría más énfasis de maldad a la risa.

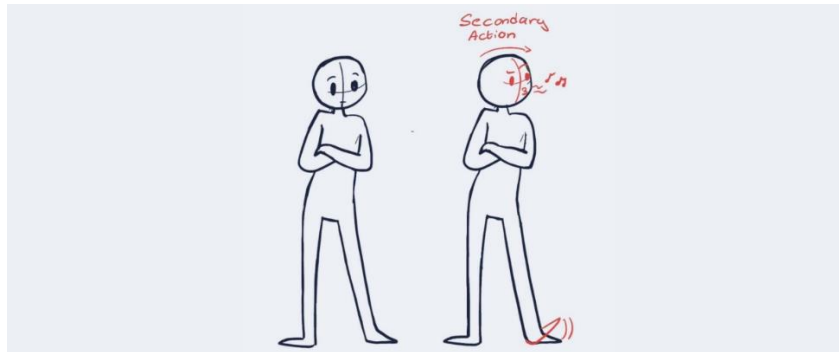


Figura 11. Ejemplo del principio Acción Secundaria con un personaje esperando
(Fuente: <https://darvideo.tv/blog/recovering-the-classics-famous-12-principles-of-animation/>)

9. **Timing** (o Sincronización)

Es la sincronización de los tiempos de cada acción en la escena. La velocidad está determinada por la forma en la que se distribuyan los fotogramas clave. Para distribuir las poses clave hay que tener en cuenta la masa y el peso de las entidades, ya que determinan el impulso que necesita para moverse. Los patrones de movimiento también cambiarán para ayudar a transmitir: emociones, la personalidad de un personaje, sus sentimientos en esa escena, o sus reacciones. Por ejemplo, si es una entidad más pesada necesitará un timing mayor; o si es más ligera uno menor. Es decir, en la primera las poses clave se verán distribuidas, en la línea del tiempo, como el movimiento de una acción acelerada; y la segunda como una desacelerada. Otro ejemplo sería, según los sentimientos, si se quiere transmitir que está triste, se podrían disponer las poses clave de forma más separadas unas de otras; o si es una acción más frenética y alegre, podrían estar más juntos los *keyframes*. U otro caso, según la personalidad del personaje, si es un villano que atemoriza con solo sus pisadas, será más pausada, sus fotogramas clave estarán dispuestos de una forma más uniforme en la línea temporal.

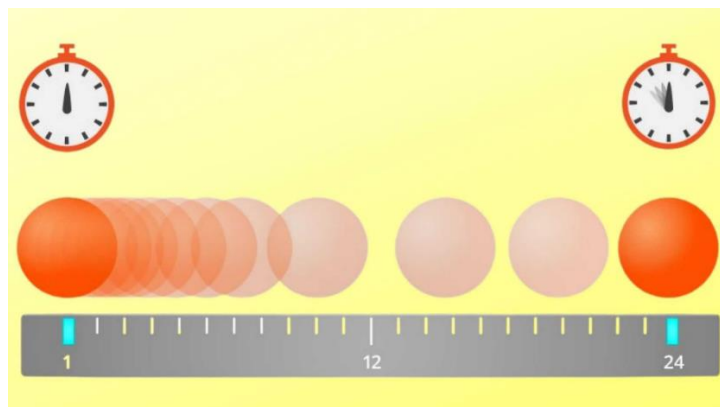


Figura 12. Ejemplo de Timing con una bola y su movimiento
(Fuente: en la cabecera de la página <https://darvideo.tv/dictionary/timing/>)

10. **Exageración** (*Exaggeration*)

Antiguamente, en la animación tradicional donde cada fotograma se dibujaba a mano, este recurso era muy importante porque ayudaba a los animadores a que sus trabajos quedasen más atractivos y llamativos, disimulando los posibles errores de desplazamiento de las entidades de un fotograma a otro. Pero, aun así, este principio sigue siendo importante para muchas otras cosas actualmente.

En la animación, no basta con solo realizar movimientos realistas, es necesario agregarle personalidad, y esto se puede conseguir con este recurso. Este principio sirve para acentuar una acción. Hay que tener cuidado y usar este principio con precaución, porque si se abusa de él puede no lograrse los resultados esperados y que la animación en conjunto quede de poca calidad, antinatural y no fluida. Por tanto, no debe usarse de forma arbitraria. No es necesario que siempre sea una exageración extrema, a veces basta con usar la dosis justa dentro de lo que se está animando, logrando los objetivos deseados. Se consigue realizar mediante las distorsiones del cuerpo de una entidad con la finalidad de generar un efecto dramático o cómico. También se puede aplicar al escenario distorsionando el espacio y sus formas. Igualmente, a la cantidad de elementos en una escena, pero esto no implica que deban estar desordenados, y/o en caos, deben estar, al menos en cierta medida, en armonía para que el espectador pueda seguir la escena sin saturar su atención en demasiados puntos focales. Asimismo, agregarla con elementos visuales que se añadan a la escena de forma complementaria, que pueden estar o no en el mismo plano de la realidad que los personajes, siendo apuntes para el espectador. Como podrían ser textos, carteles, bocadillos de pensamiento del personaje, colores saturados o desaturados, o incluso, elementos más contemporáneos como emoticonos, stickers, memes o gifs. Este principio siempre ha marcado una distinción clave de que se puede lograr con la animación y, sin embargo, no se puede obtener en otros ámbitos como con la imagen real. El principio de la exageración hace que el animador pueda dar rienda suelta a su creatividad, y romper con las reglas de la física y la materia del mundo real. Las animaciones de los *Looney Tunes*, están repletas de la utilización de este recurso.



Figura 13. Ejemplo del principio Exageración con los ojos de un personaje
(Fuente: <https://nutchelleblog.wordpress.com/2015/11/22/exaggeration/>)

11. Dibujo sólido (Solid Drawing)

Está relacionado con el diseño, la posición y el movimiento de las entidades. Implica que las entidades deben crearse siempre teniendo en cuenta un espacio tridimensional, para que simule la realidad tal y como la percibe un ser humano, y ayude al espectador a entender los volúmenes de las entidades, los movimientos de estas y sus desplazamientos por la escena. Aquí se pone en práctica los conceptos de la masa, el volumen, el peso, la perspectiva y la iluminación.

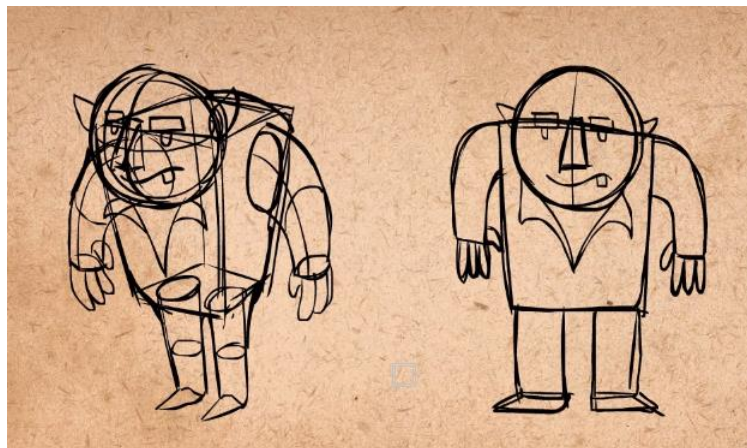


Figura 14. Ejemplo del principio Dibujo sólido con un personaje
(Fuente: <https://www.artsandjustice.org/digital-arts-12-principles-of-animation/>)

12. Atractivo (Appeal)

Este principio no se refiere a que todos los personajes tengan que ser guapos o que caigan bien. Lo que determina si un personaje es guapo o feo, si cae bien o mal, es muy subjetivo, y dependerá de cada espectador, de sus gustos y opiniones. Un villano tiene su atractivo, un personaje odioso también tiene su atractivo. Este principio se refiere a la atracción que crea la entidad en el espectador, lo cautivador que sea, por qué llama la atención, qué lo hace interesante. Puede ser por su diseño, su personalidad, su forma de desplazarse, su carisma, sus colores, etc. Es cierto que este principio es algo subjetivo, porque, aunque en diseño e ilustración

hay algunas pautas que se pueden seguir para transmitir ciertas emociones, al final también depende de la época, la cultura, y, a fin de cuentas, del espectador.

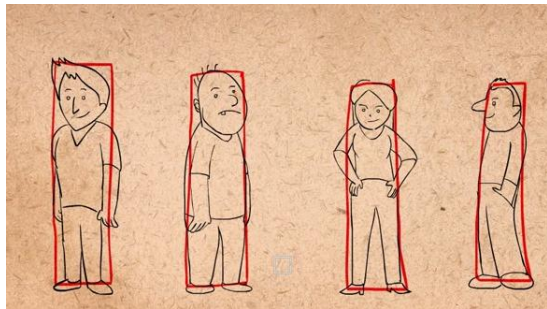


Figura 15. Ejemplo del principio Atractivo sin aplicar
(Fuente: <https://www.artsandjustice.org/animation-12-principles-of-animation/>)

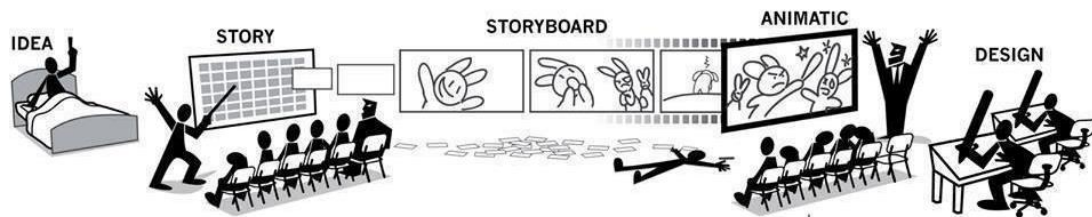


Figura 16. Ejemplo del principio Atractivo aplicado
(Fuente: <https://www.artsandjustice.org/animation-12-principles-of-animation/>)

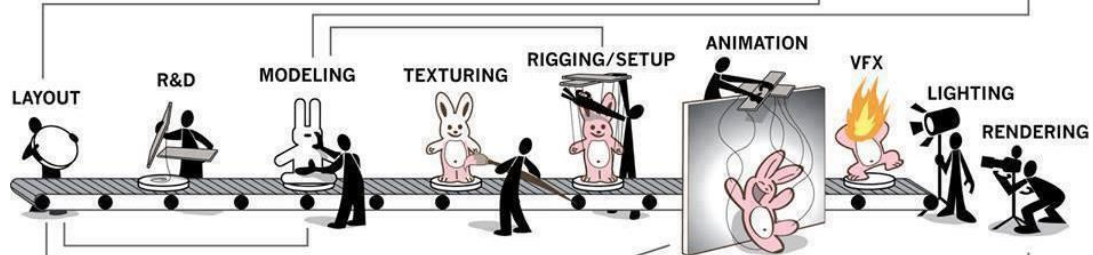
3.4. Preproducción

Los proyectos como la creación de una animación, de una película, una *intro* para vídeos, una canción, etc. Siguen una elaboración de proyecto lineal. Es decir, no son como por ejemplo crear una aplicación o un videojuego donde el programador debe intentar tener en cuenta todas las posibles acciones que vaya a intentar realizar el usuario para prevenir fallos, ya que las acciones dependen de ellos (proyectos no lineales). No, en una película, en una animación, en un vídeo, o en una canción todas las acciones que existan ocurren siempre de la misma manera en el mismo momento del tiempo. Por tanto, se estructuran de la misma forma en las siguientes fases: preproducción, producción, y postproducción. En cada fase se realizarán las tareas necesarias dependiendo de las que requiera en concreto cada proyecto. Mientras que algunas de las tareas del proyecto dependerán de otras, y muchas otras se podrán hacer en paralelo con otras; normalmente las fases si dependen de que la fase anterior se haya terminado para poder comenzar la siguiente. En este y los siguientes dos apartados se explicarán algunas de las tareas que se realizan en proyectos de animación 3D. Aunque no es necesario ni que todos los proyectos de animación 3D las realicen todas, ni tampoco son las únicas que existen.

PRE-PRODUCTION



PRODUCTION



POST-PRODUCTION

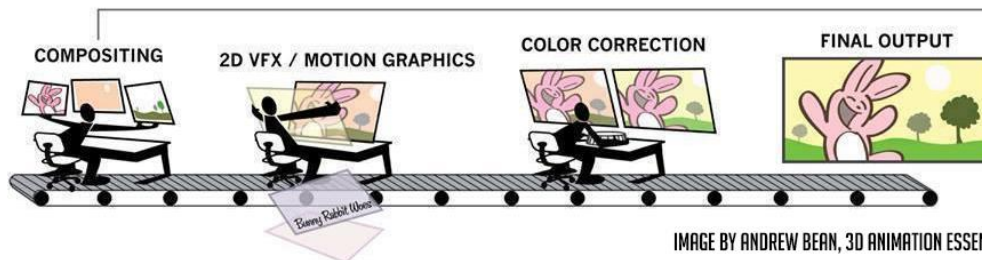


IMAGE BY ANDREW BEAN, 3D ANIMATION ESSENTIALS (2012)

Figura 17. Pipeline de animación 3D

(Fuente: <https://animeight.com/2018/02/21/3d-production-pipeline/>)

La preproducción es el conjunto de tareas previas que se deben realizar para planificar y organizar un proyecto de este estilo. Entre ellas se encuentran las tareas relacionadas con los capítulos de *Planificación*, *Objetivos* y *Metodología* de este documento, y las que se explican a continuación.

3.4.1. Brainstorming

Lo primero que se debe hacer en la etapa de preproducción, es determinar la planificación del proyecto, los objetivos y definir cómo será el flujo de trabajo. Una vez se tenga esto claro, se puede definir la idea del proyecto. Una forma de hacerlo es mediante la técnica del *brainstorming*.

El *brainstorming* es la técnica más conocida y aceptada en todo el mundo, es utilizada sobre todo en trabajos colaborativos, pero es perfectamente factible para trabajos individuales.

Mediante esta técnica se consigue que todos los miembros involucrados participen, se expresen, aporten sus propias ideas y den rienda suelta a su creatividad.

Consiste en reunir al grupo, lo más aconsejable es que cada uno pertenezca a un área de conocimiento diferente, para que pueda aportar ese punto de vista. Se plantea un problema, se puede explicar de dónde proviene el problema, para qué se necesita una solución y lo que haga falta para preparar a todo el grupo y que tengan la información necesaria. Luego todos deben aportar ideas espontáneamente para dar posibles soluciones al problema, y un moderador debe apuntarlas todas. Las reglas son que todas las ideas son aceptadas, nadie las critica, solo se escuchan y se aportan mejoras, y sin perder de vista el objetivo final. A continuación, cuando acaba el tiempo establecido para la lluvia de ideas, se trabaja en cada una de las ideas para ver si son factibles y si se pueden mejorar. Por último, se evalúan y entre todos se escoge la solución más adecuada para el problema. A partir de aquí se empieza a desarrollar la idea, y con esto se consigue encaminar el trabajo, ya pueden realizarse el resto de las tareas de la etapa de la preproducción.

Existen variantes de esta técnica, pero el objetivo es el mismo. Se puede hacer que en vez de decir las ideas en voz alta las escriban en un papel, que terminen las ideas en un papel de la persona a su derecha, listar una serie de preguntas sobre el problema a ver qué responde cada uno, etc. Pero al final obtienes una posible solución dada entre todos para el problema, y esta solución será la que habrá que desarrollar.

3.4.2. Moodboard

Un Moodboard es una herramienta creativa para todo el proyecto en general, aunque se pueden realizar *moodboards* concretos, pero normalmente se crea uno y se organiza por categorías. Se usan mucho en trabajos artísticos. Consiste en realizar un tablero de inspiración para el proyecto, es decir, una representación visual en formato collage con imágenes, un tablero organizado con todas las referencias necesarias. También se puede escribir anotaciones en texto. En él se recogen ideas para diseños, tipografías, texturas, paletas de colores, composiciones, estilos de modelos 3D y todos los elementos necesarios para la composición artística. Esto ayuda a que todo el equipo e incluso el cliente puedan visualizar mejor el estilo de lo que se quiere conseguir, y al fijarse todos los departamentos en lo mismo, se consigue que sea coherente y armónico todo el proyecto.



Figura 18. Ejemplo de moodboard con anotaciones
 (Fuente: <https://gamesartworld.wordpress.com/2015/01/30/first-ideas-down-week-18/>)

3.4.3. Guion

Todo proyecto audiovisual parte de una idea escrita en papel. Para poder transmitir algo mediante imágenes primero se debe expresar por escrito. El guion es una herramienta estructurada donde se cuenta una historia que será convertida en un producto audiovisual. Debe proporcionar a todo el equipo de producción información como: la historia de los personajes; los lugares, escenarios y localizaciones de cada escena; los diálogos; y estar definidos todos los elementos necesarios para la producción. No se escribe igual un guion para un programa de televisión que para una película o una serie, porque no todas esas producciones necesitan la misma información. A continuación, se explicará cómo sería un guion para películas, que es más o menos igual que para un corto, y, por tanto, se puede escribir el del bucle basándose en él.

La construcción del guion se puede realizar en distintos formatos. Aunque no es necesario escribirlo en todos ellos, sí es aconsejable, sobre todo para producciones muy grandes. Los distintos formatos serían:

- Escribir **el tema**, es decir, la idea principal del proyecto audiovisual de forma clara, y debe resumirse en una o dos frases.
- **La sinopsis** o argumento, que es la síntesis de la historia. Marca la trama fundamental de la historia, como son los personajes protagonistas y el espacio y el tiempo donde transcurren los hechos. Normalmente se usa para promocionar el producto más tarde.
- **La escaleta** donde se representa el guion de forma esquemática por escenas ordenadas cronológicamente. Es un pre-guion muy importante para que todo el equipo pueda trabajar. Se diferencia del guion en que este solo mostraría el esqueleto mientras que el guion sería mucho más completo y detallado. Es la forma simple de escribir el guion por pasos ordenados de lo que sucederá en cada escena. Para cine, cortometrajes y animación, se muestra por puntos de forma muy resumida cada escena. Puede añadirse encabezados, donde se indica si la escena es en exterior o interior, si es de día o de noche y el decorado donde transcurre la acción. No suelen incluirse los diálogos.
- **El guion literario**. Este sería mucho más completo, con todos los detalles que se necesitan para el rodaje.
- **El guion técnico** que muestra la planificación del guion literario para llevarlo a la pantalla. Se incluirían los movimientos, ángulos y posiciones de la/s cámara/s, la iluminación, los encuadres, los efectos visuales, sonidos, música y los modos de transición.
- **El storyboard** o guion gráfico es una serie de ilustraciones secuenciadas y ordenadas cronológicamente que sirven para definir una historia de manera visual. Con este guion gráfico se estructura el proyecto lineal antes de crearlo. Puede ayudar a visualizarlo y estudiar si la composición de cámaras, luces y música está bien organizada y pensada y va a ser atractiva. Normalmente en un storyboard se indica para cada escena: el número de la escena; la duración; y la ilustración que muestra lo que se verá y desde qué ángulo. Pero también opcionalmente se puede mostrar: una breve explicación de la escena; el diálogo de los personajes; y qué sonido/s y/o música/s suenan en esos instantes.



Figura 19. Ejemplo fragmento de storyboard de la película Peter Pan de Disney
(Fuente: <https://www.pinterest.es/pin/399483429420848158/>)

3.4.4. Diseño de personajes, elementos y entornos (bocetado)

Existe una especialidad en arte denominada *concept art*. Los *concept artist*, como su nombre indica, son los artistas conceptuales que diseñan las ideas del proyecto mediante la ilustración. Con esto se consigue trabajar la parte visual del proyecto, para que el resto del equipo tenga en fases posteriores una misma idea de la que partir para realizar sus tareas. En este momento se diseñan los personajes, los *props* o elementos, y los entornos o escenarios. Normalmente en un estudio hay artistas especializados para realizar los *concepts* de cada tipo de diseño. Para este proceso es muy importante la imaginación. Los *concepts arts* pueden estar hechos en formato digital o tradicional, aunque hoy en día lo más común es hacerlo en medios digitales porque termina siendo más práctico para todo el proyecto y el equipo.

Para la creación de los diseños es necesario tener cierta base en: dibujo, saber dibujar en distintas perspectivas, de volumen, la anatomía; diseño, como transmitir emociones, saber de composición, la teoría del color; diferentes estilos (esto será según el propio proyecto lo que necesite) como el *cartoon*, el realista, el semirealista, la caricatura, el entintado, etc.; y saber buscar referencias e inspiración. Dependiendo del proyecto, a veces los organizadores, el cliente, la empresa o el estudio darán al *concept artist* una idea más o menos de cómo quieren el diseño de cada cosa, proporcionándole documentos con la información, o un moodboard, o descripciones, o referencias, o algo similar. Pero en muchas ocasiones esto también es trabajo

de investigación de los artistas. Es importante tener cuidado con el diseño y no dejar nada al azar.

Para el diseño de personajes se puede empezar por escribir en un papel cómo es su fisionomía, el estilo de ropa que usa, su personalidad, su rol en la historia, su trasfondo, el periodo de tiempo y el entorno en el que se sitúa, sus habilidades y a qué se dedica, para poder dejarlo reflejado en su diseño. Una vez se tienen todas esas cosas claras y apuntadas, y junto con el *moodboard*, se procede a hacer varios bocetos. En ocasiones se utiliza la técnica de los *thumbnails* para realizar las primeras figuras de los bocetos en blanco y negro, y ver cuál funciona mejor. Una vez hechos los bocetos y estén aprobados, se elegirá de entre ellos que estilo de ropa, que peinado, que paleta de color, que pose de presentación de personaje y otras características serán las definitivas para ese personaje. Una vez elegido todo esto, se puede pasar a crear la *Model Sheet*. Es una hoja de referencia del diseño del personaje con la que luego trabajará el resto del equipo. En ella también se marcan ciertas indicaciones básicas que deban permanecer sin cambios porque son importantes para ese personaje, o que deban artísticamente siempre hacerse de alguna manera en concreto. Aparte, se hará el concept art con todo lo elegido en la pose elegida para que se pueda tener como referencia más adelante.

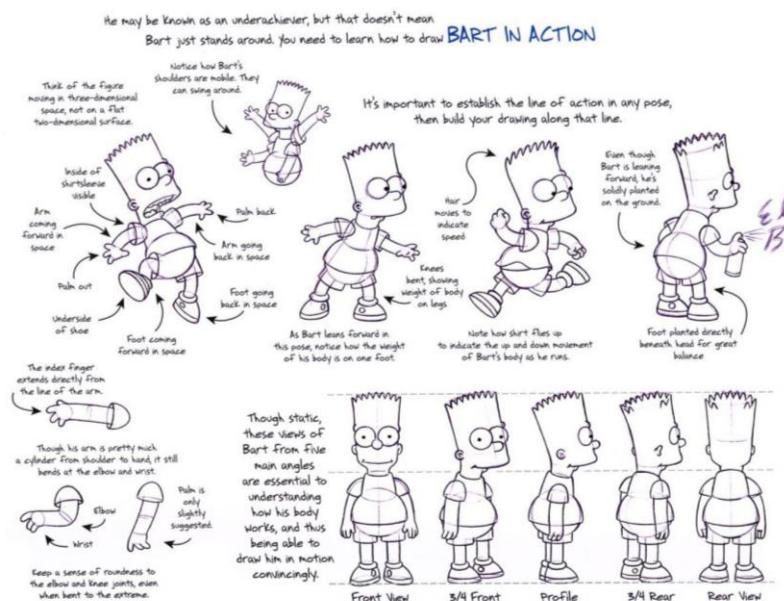


Figura 20. Ejemplo de Model Sheet del personaje Bart de los Simpson
(Fuente: <https://www.pinterest.es/pin/301248662554397641/>)

Para el diseño de props o elementos y de entornos o escenarios se tiene en cuenta si cuando se pase al 3D van a ser modelos para un videojuego, una película, etc. Porque en el caso de los videojuegos, hay que tener muy en cuenta el número de polígonos, según el proyecto se marcarán cuántos se pueden usar para cada tipo de elemento incluido para los personajes, y

muchos de los detalles se tendrán que añadir mediante las técnicas de texturizado que se explican más adelante. Como con los personajes, hay que tener a mano el moodboard y en este caso la ficha de cosas que hay que tener en cuenta serán, de colores, la época en la que se ambienta, si está en exterior o interior, el estilo si es realista, *cartoon*, semirealista, etc. Con todo eso hacer los bocetos y cuando se escojan las partes que formarán parte del diseño final, hacer el *concept art* y si es preciso, se pueden hacer planos arquitectónicos del escenario. Estos ayudarán a la hora de modelar a saber cómo es la distribución y las medidas por dentro y por fuera del escenario. En ocasiones, dependiendo del proyecto y cuál sea la finalidad, el entorno puede ser un simple plano de un color, si lo que se quiere conseguir es que el personaje y las acciones que hagan sean el punto de atención. En este caso solo se debe elegir un color de fondo que no atraiga más la mirada que el propio personaje. Pero debe quedar armónico y visualmente atractivo e incluso se le pueden añadir algún que otro elemento decorativo, siempre teniendo en cuenta el objetivo final, que no llame más la atención que el personaje, si lo hace es mejor prescindir de ello.

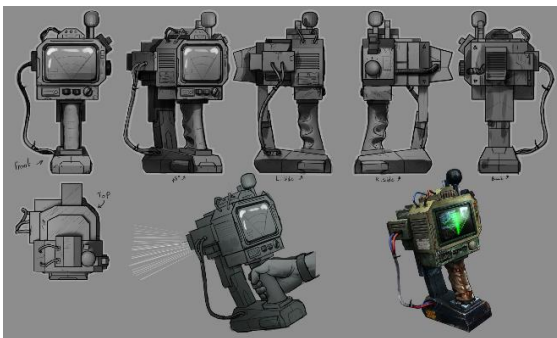


Figura 21. Ejemplo Concept Art de un prop en su Model Sheet
(Fuente: <https://www.artlex.com/blog/prop-concept-art/>)



Figura 22. Ejemplo Concept Art de un escenario en su Model Sheet
(Fuente: <https://twitter.com/UArtsSchool/status/1339961679413727239>)

3.5. Producción

La producción es el conjunto de tareas que se deben realizar para llevar a cabo el proyecto audiovisual que se ha planificado. Con toda la documentación y diseños realizados en la etapa anterior, se pueden realizar las tareas de esta etapa. Entre ellas se encuentran las tareas que se explican a continuación. Aunque no son las únicas que existen ni todos los proyectos tienen que realizarlas todas, solo las necesarias para el proyecto que se está realizando.

3.5.1. Modelado

Para poder realizar la animación, es importante realizar los modelos o modelos de los personajes que se van a animar, los *props* u objetos que se verán por el escenario y puede que los usen los personajes, y el escenario en el que se localizarán. Por eso, una de las primeras fases es el modelado 3D.

Normalmente primero se detalla una lista de los modelos necesarios para el proyecto, separados por personajes, *props* y escenarios. Existen especialistas que se encargan de modelar cada uno de los tipos de modelos, a los que se llaman *character artist*, *prop artist* y *environment artist* respectivamente.

Se llama **objeto** a cada uno de los elementos que forman parte de una escena 3D como una luz, una cámara, una línea curva (*spline*) o un modelo formado por una sola malla. Los objetos se pueden agrupar o guardar en colecciones, para facilitar su manejo. Se usa un entorno con un espacio tridimensional representando las coordenadas con los ejes *x*, *y* y *z* (dependiendo del programa, cada una de las letras, estará representando el alto, ancho y la profundidad), donde se dispondrán los objetos y los elementos que los forman. Los modelos 3D están formados por 3 tipos de elementos: **vértices**, **aristas** y **caras**. Estos elementos forman polígonos, un polígono es una cara que puede estar formada por mínimo 3 vértices unidos por 3 aristas, formando un triángulo, al que se llama *tri*. Normalmente, para tener una malla correcta se utilizan **quads**, que son polígonos de una cara formada por 4 vértices unidos por 4 aristas. Los polígonos de más de 4 lados, se llaman ***n-gon***. Los vértices reciben el nombre de **pole de X** según el número de aristas que se unan a ellos. Para un modelo realizado con quads, la mayoría de los vértices serán poles de 4 y algunos tendrán que ser poles de 3 o de 5 pero no de más si se quiere lograr tener una topología correcta. Usar un tipo específico de polígono, o al menos intentar que todos sean formados de la misma manera, evita tener errores posteriores en la malla como: que no se rompa, ni se creen huecos entre los vértices; que no se les de la vuelta a las caras; que no se creen dobleces extraños en algunas partes del modelo; o, en general, que no tengan la topología correcta para su posterior animación. Se llama **topología** a la organización de los polígonos de una malla de un objeto. Se posiciona en un entorno en 3D, muchos polígonos para obtener las mallas de los modelos deseadas. Estas mallas de los modelos se llaman ***high poly*** o alta poligonación cuando están más detalladas, es decir, tienen un mayor número de polígonos, más número de vértices, caras y aristas; y cuando son de baja poligonación, se llaman ***low poly*** y tendrán menos.

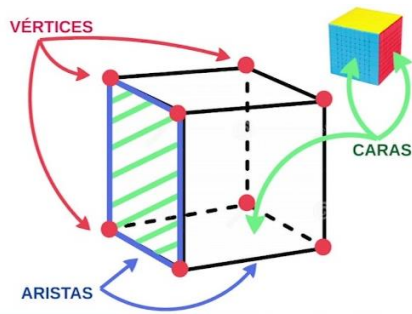


Figura 23. Elementos que forman un polígono en un modelo 3D

(Fuente: captura del minuto 1:28 del video

https://www.youtube.com/watch?v=JkeMR4W7-f4&ab_channel=RicardoL.Mart%C3%A9ndezRes%C3%A9ndiz)

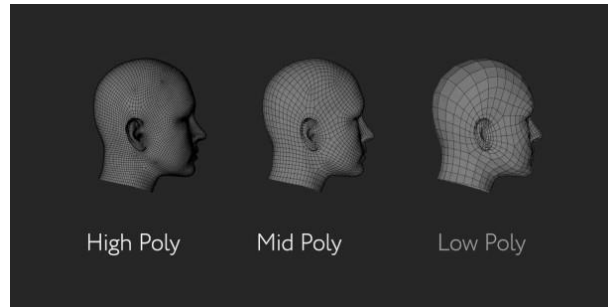


Figura 24. Ejemplo de high poly, mid poly y low poly

(Fuente:

<https://shop6602.sosoutremer.org/category?name=low%20poly%20mid%20poly%20high%20poly>)

Por ejemplo, para videojuegos, usar los *quads* es la mejor opción, puesto que, al introducir las mallas de los modelos en el juego, estas se triangulan, es decir, todos los polígonos se dividen para que formen triángulos. Si son *quads*, todos los polígonos se convertirán de forma correcta en polígonos triangulares (*tris*). Si, por ejemplo, se hiciese la malla directamente en polígonos triangulares, al introducir la malla en el motor del videojuego se volvería a triangular dejando una malla con demasiados polígonos y con posibles fallos. En los videojuegos es importante conseguir que los objetos estén formados por mallas con el menor número de polígonos posible. Porque si tienen demasiados, a la tarjeta gráfica le costará mover los modelos y se verá el juego con bajos FPS, dando una mala experiencia al usuario.

Los objetos se pueden ver en el entorno por una ventana que funciona con casi los mismos parámetros que un objeto cámara. Hay dos tipos de **vista: global**, con la que se ven todos los objetos de la escena; o **local** que aísla un solo objeto para su visualización. Se puede cambiar en la configuración del programa: la **distancia focal**, para configurar a partir de que distancia de la lente de la cámara se dejarán de visualizar los objetos; la **profundidad de campo**, a partir de qué distancias los objetos se ven enfocados y nítidos, y cuándo no. También suele haber una rejilla donde se posicionan los objetos y elementos en el entorno 3D, se puede configurar el tamaño y divisiones de esta. Es posible alternar la vista global y la local. Aunque también se puede crear un objeto cámara y visualizar la escena desde ese punto de vista, pero no se podrá modificar las mallas de los modelos ni elementos de la escena desde ahí, solo posicionar y configurar la propia cámara. Los objetos se pueden visualizar en el entorno 3D en dos tipos de **proyecciones de cámara: ortográfica**, proyecta los modelos desde una distancia infinita a un plano; o en

perspectiva, una visión más realista, los objetos que estén más cerca de la cámara se verán más grandes y conforme se alejen de ella se harán más pequeños por la distancia.

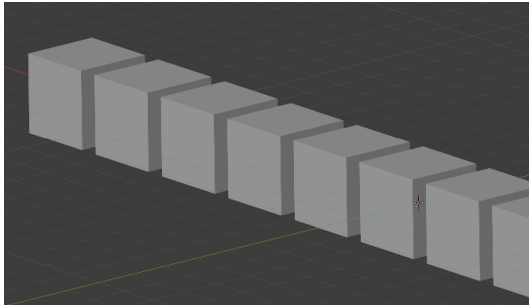


Figura 25. Proyección ortogonal
(Fuente:

<https://docs.blender.org/manual/es/latest/editor/s/3dview/navigate/projections.html>)

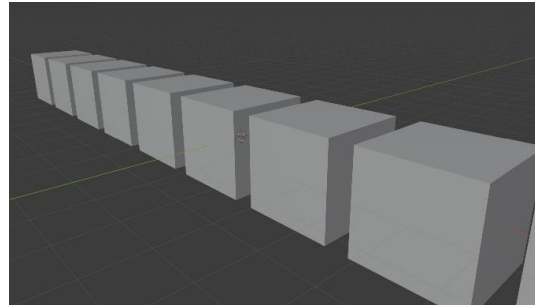


Figura 26. Proyección en perspectiva
(Fuente:

<https://docs.blender.org/manual/es/latest/editor/s/3dview/navigate/projections.html>)

También se pueden cambiar las **vistas de la cámara**, según su posición, para ver: la frontal, la trasera, del lado derecho, del lado izquierdo, la parte de arriba, y la parte de abajo del objeto. Se usan 3 movimientos para desplazarse por el entorno 3D y poder ver los modelos y la escena desde cualquier punto y vista, estos **movimientos de cámara** se llaman:

- **Paneo**. Es un tipo de movimiento de cámara sin desplazamiento: hacia arriba, hacia abajo, hacia los lados o en diagonal. Se usa para fijar un objeto o mostrar una parte de la escena.
- **Travelling**. La cámara se desplaza de lugar: hacia delante, hacia atrás o hacia los lados.
- **Zoom**. Puede ser para acercarse o alejarse de un objeto o elemento de la escena.

Además, se pueden visualizar los modelos con: **modo alámbrico**, donde las caras no se ven y así se pueden ver todos los vértices y aristas de detrás de las caras; ver los objetos en **modo sólido**, los modelos se visualizan con un color que se puede configurar, en este modo se ven las caras y no dejan ver las aristas ni vértices de detrás; el **modo transparencia** que hace que el color sólido sea transparente, deja ver los vértices y aristas de atrás, pero se ven semitransparentes las caras; **modo materiales**, se ven los materiales (pero no texturas) y la configuración de la iluminación de la escena de forma sencilla; **modo texturas**, se ven las texturas proyectadas en los modelos y la configuración de la iluminación de la escena de forma más realista de cómo será en el render final de la escena. Normalmente, en todos los programas se puede combinar cualquier modo con el modo transparencia.

Existen 3 tipos de **transformaciones geométricas** que se pueden aplicar a los objetos o a los elementos que lo forman, que son **mover**, **rotar** y **escalar**. Además de esto, se utilizan otras

herramientas para **cortar, extruir, unir, separar**, etc., y otras como **pinceles, modificadores** o las **propiedades** de un objeto o sus elementos, como su posición en el espacio, para manipularlo.

Se diferencian dos tipos de modelos:

- **Modelos sólidos o físicos:** son modelos que se basan en líneas rectas, ángulos y formas más rígidas. Los objetos físicos en el mundo real como una pantalla, un ratón, una verja, una silla, y demás fabricaciones del ser humano, suelen entrar en este tipo de modelos. Suelen ser modelos con simetría.
- **Modelos orgánicos:** los modelos que presentan muchas curvas. Aquí entran los personajes, las telas, animales, y demás modelos que no tienen una simetría clara ni son tan rígidos.

Para realizar los modelos se pueden seguir distintos tipos de métodos. También se pueden combinar cualquiera de estos métodos con otros para conseguir el resultado deseado. A continuación, se explicarán algunos de los que hay:

- **Modelado 3D mediante Primitivas básicas.** En todos los programas de modelado vienen por defecto unos objetos básicos como un cubo, una esfera, un cono, un cilindro, una línea, un plano, etc. A estos elementos se les llaman primitivas. Se pueden usar y combinar entre ellas para formar el modelo. Normalmente para figuras básicas de modelos físicos.

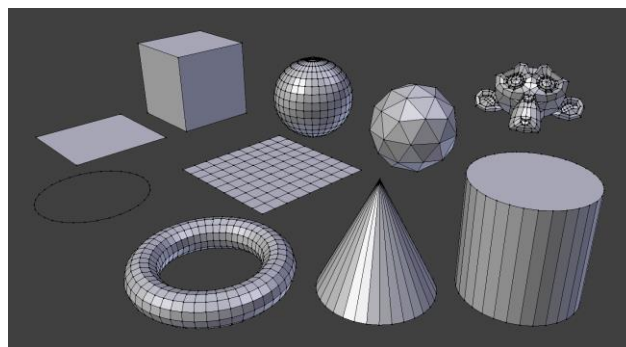


Figura 27. Ejemplo Primitivas Básicas en Blender

(Fuente: <https://docs.blender.org/manual/es/latest/modeling/meshes/primitives.html>)

- **Box Modelling o Modelado de cajas.** Consiste en partir de un cubo (una primitiva básica) y manipularlo extruyendo caras (crear un grosor a la cara o caras del objeto), rotando, escalando y moviendo vértices y aristas, y haciendo cortes para conseguir formas más complejas. Con este método se puede llegar incluso a modelar personajes y objetos más orgánicos. Por ejemplo, los personajes *low poly* podrían modelarse con este método.

Muchas veces, se usa este método combinado con la escultura digital 3D, que se verá más adelante. Este se usa para generar la forma básica del objeto como base, y luego mediante la escultura digital 3D se definen los detalles.

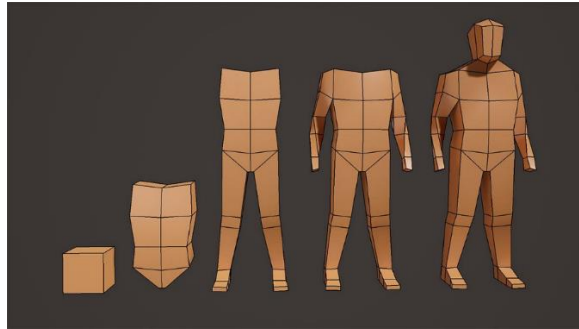


Figura 28. Ejemplo Box Modelling de personaje low poly
(Fuente: <https://shop13020.okawaballet.com/category?name=box%20modeling>)

- **Modelado 3D Poly by Poly o Poligonal.** Consiste en crear a mano los polígonos uno a uno, desde el primer vértice al último. Normalmente se usan *quads*, colocas los vértices, los unes creando una cara con 4 aristas y vas creando la forma de la malla. Se crean *loops* de polígonos en las zonas con curvas y circunferencias, que son *arrays* o hileras de polígonos que siguen el contorno de ese tipo de zonas, y poco a poco se van uniendo al resto de partes. Es un proceso más tedioso, pero proporciona mallas más flexibles y productivas para luego poder manipular el objeto por ejemplo en animación. Es decir, se consiguen mallas con una topología muy buena. Este método sirve para modelar todo tipo de objetos ya sean orgánicos o sólidos.

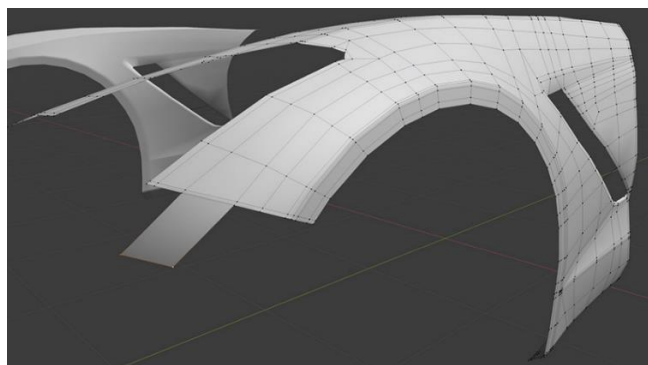


Figura 29. Ejemplo de Modelado Poly by Poly
(Fuente: <https://monstruosdeldiseno.com/podcast/tipos-tecnicas-modelado-3d>)

- **Modelado 3D con Nurbs.** Se denominan Nurbs a las curvas. Con este método mediante las curvas de Bézier (*B-splines*), que son el mismo tipo de curvas que se pueden encontrar en programas de ilustración vectorial, como Inkscape, o Illustrator, o como la

herramienta pluma de Photoshop, tienen puntos de ancla que ayudan a manipularlas para generar la forma de la curva deseada. A partir de esa curva, se pueden crear objetos. Con las curvas dibujas el contorno de la figura y después la extruyes o con ayuda de otra curva puedes indicar con qué forma debe extenderse la curva para crear la superficie del objeto. Esta técnica se usa mucho en modelado CAD (diseño asistido por computadora) para generar piezas mecánicas e industriales.

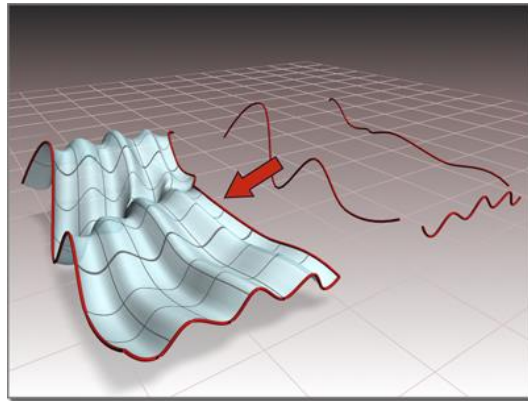


Figura 30. Ejemplo de Modelado con Nurbs

(Fuente: <https://help.autodesk.com/view/3DSMAX/2023/ENU/?guid=GUID-DD3B220A-7D44-4C55-A274-002F25976D04>)

- **Escultura Digital 3D.** Permite modelar los objetos como si se trabajara con arcilla o plastilina en el ordenador. Creas una esfera con suficiente polígonos para poder manipularla sin que se rompa o de errores, y mediante herramientas como los pinceles de la escultura tradicional, creas la malla de un objeto. Se modela el objeto primero con menor resolución y poco a poco con subdivisiones se crea más resolución y se van esculpiendo los detalles. En este tipo de proceso no se tiene tanto en cuenta el número de polígonos o la topología. Por lo que, a no ser que solo se use para crear la figura, normalmente, luego habrá que hacerles una retopología que se explicará más adelante. Este método se usa para modelar objetos orgánicos.

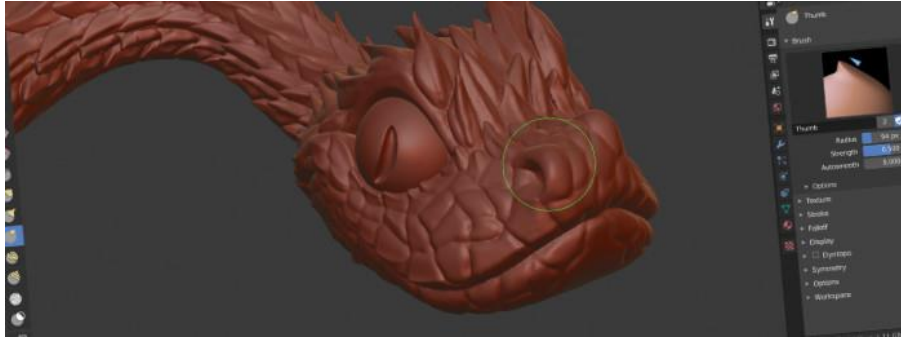


Figura 31. Ejemplo Esculpido en 3D

(Fuente: <https://monstruosdeldisenio.com/podcast/que-es-escultura-digital-3d>)

- **Texto 3D con Extrusiones.** Con una herramienta de texto similar a las que se usan en programas de ilustración o fotografía, creas un texto con la forma deseada y luego aplicas una extrusión. Así obtienes un texto en 3D.

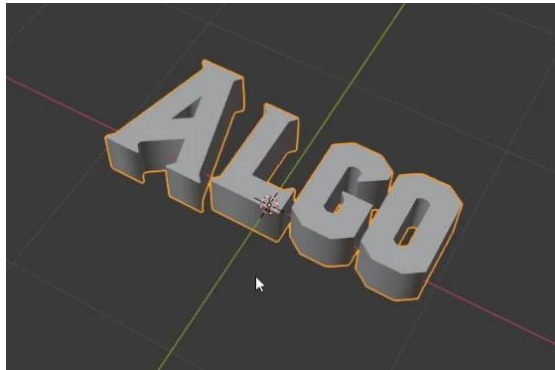


Figura 32. Ejemplo Texto en 3D

(Fuente: <https://gamedevtraum.com/es/tutoriales-y-curiosidades-sobre-blender/como-crear-texto-3d-en-blender-con-cualquier-fuente/>)

- **Modelado 3D con Revoluciones.** Con las curvas de Bézier, creas el contorno de un objeto, y generas el volumen recubriendo con polígonos aplicados mediante una revolución con los grados deseados. Con este método se pueden crear fácilmente recipientes como vasijas, vasos, copas, etc.

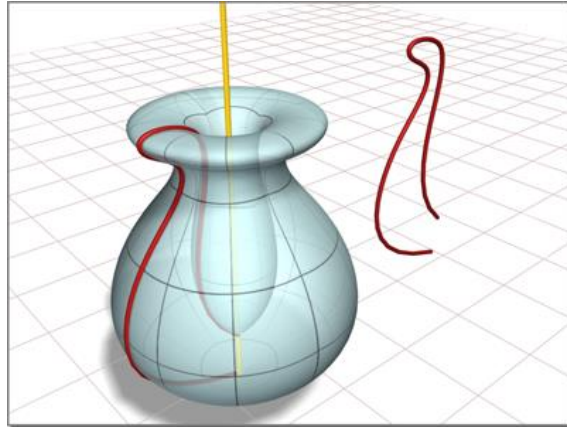


Figura 33. Ejemplo Modelado con Revoluciones 360º

(Fuente: <https://help.autodesk.com/view/3DSMAX/2023/ENU/?guid=GUID-2D4547C0-6956-4544-8CED-26BED9962B1C>)

- **Modelado con Metaballs** (Metabolas). Las Metaballs son un tipo de objeto 3D que funciona como si fuese un fluido, como el agua o mercurio, que, al estar cerca unas bolas de otras, estas se unen por proximidad. Este método no es muy usado, pero suele utilizarse para generar objetos líquidos o animaciones de líquidos.

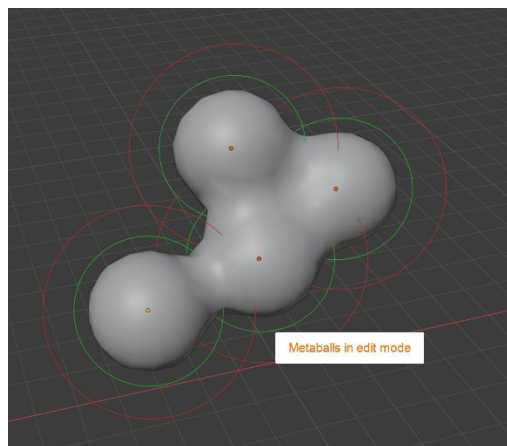


Figura 34. Ejemplo Metaballs en Blender

(Fuente: <https://artisticrender.com/how-to-use-metaballs-in-blender/>)

- **Fotogrametría.** Consiste fotografiar un objeto en la vida real desde todos los ángulos posibles. El objeto debe permanecer sin alterar su posición y la cámara es la que debe girar alrededor de este fotografiándolo en todos los ángulos. Luego en programas especializados, estas fotografías se usan como si se hubiese escaneado el objeto, y a partir de ellas, se crea la malla del objeto. También se usan para esta técnica: escáneres especializados, para una mayor precisión, o drones, para superficies más grandes. Esto proporciona una malla con una topología muy sucia, por tanto, también habrá que hacer retopología y limpiar vértices que puedan haber quedado flotando sin pertenecer a ninguna malla.

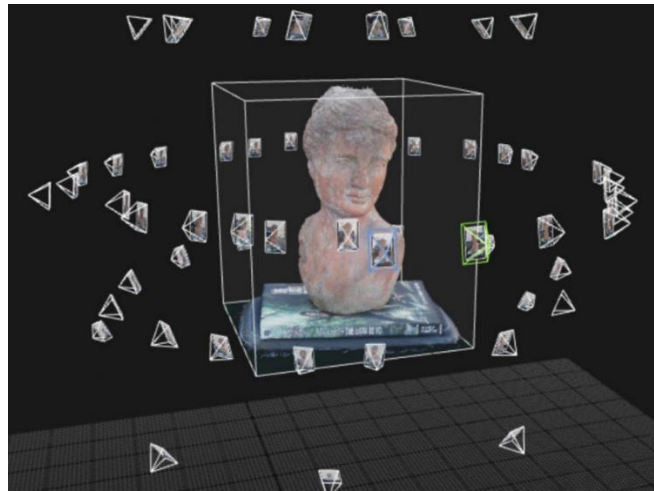


Figura 35. Ejemplo de Fotogrametría de una estatua

(Fuente: <https://learn.nebulargroup.com/kb/usos-de-la-fotogrametria-en-la-impresion-3d/>)

- **Modelado 3D Paramétrico.** A partir de una primitiva básica como un plano, se usan las propiedades del objeto y se aplican modificadores para modificar su aspecto. Los valores de los modificadores se pueden ajustar, y también se pueden usar imágenes y texturas para generar motivos que sean los que se apliquen con el modificador. Este método suele usarse para la creación de terrenos, como escenarios montañosos, por ejemplo.

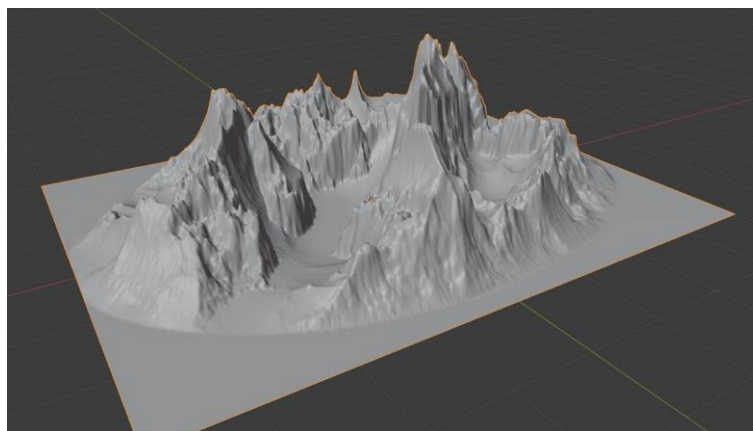


Figura 36. Ejemplo Modelado Paramétrico de un terreno

(Fuente: captura del minuto 22:13 del vídeo

https://www.youtube.com/watch?v=FAhuzxTtjMI&ab_channel=BerglMir)

3.5.2. Retopología

Para poder explicar qué es la retopología, primero se debe recordar qué es la topología de un modelo 3D. La **topología** se refiere a las características poligonales del modelo, es decir, al número de polígonos que lo forman y cómo están organizados para formarlo. En los programas 3D se puede usar el modo alámbrico o wireframe para ver en todo momento la topología del

modelo, esto permite ver los vértices y aristas que lo forman, y las caras se ven en modo invisible para que no molesten.

Entonces cuando se habla de **retopología**, consiste en cambiar la topología de un modelo ya existente. Con este método se puede reordenar y reorganizar los polígonos de la malla para que tengan una determinada topología que cumpla con los requisitos para, por ejemplo, poder animarla, o porque es necesario reducir la cantidad de polígonos, o porque se necesita que los *loops* poligonales sigan la forma del objeto de una determinada manera. Es importante mantener una copia del modelo original, ya que generalmente tiene más detalles y sirve para realizar el texturizado del modelos que se explicará más adelante. Usualmente se usa el modelado Poly by Poly o Poligonal para realizar la retopología.

Para saber en qué casos es importante realizarla, debe fijarse en los siguientes datos:

1. Modelos de alta poligonación que no vayan a ser animados, pero se necesite reducir el número de polígonos para optimizar la escena sin perder la forma original.
2. Modelos que vayan a ser animados, especialmente los personajes, para que al hacerlo la geometría se deforme y transforme correctamente sin dar lugar a fallos en la malla.
3. Modelos en los que la topología no acompañe a la forma del objeto, esto produce que se vean poco definidos los detalles, y se debe realizar con el fin de marcar mejor esas formas.

También, es importante añadir más topología, con *loops* poligonales en forma circular, que sigan la forma del objeto, en las zonas del modelo por donde se sabe que el objeto va a ser manipulado. Por ejemplo, si se habla de un modelo 3D de una figura humanoide, se pueden añadir esos **loops extra** en: las articulaciones, para que al flexionarlas no se deforme la malla; alrededor de la boca y los ojos para que sus expresiones se vean más naturales, etc.

3.5.3. Rigging y Skinning

Existen muchas técnicas de animación. El **rigging** (abreviado como *rig*) es un proceso previo a la animación para crearle un esqueleto al modelo y conseguir automatizar la animación más tarde. Facilita el trabajo a los animadores creándoles unos puntos de control o controladores en el modelo que puedan usar para animarlos con las transformaciones básicas, es la base para poder moverlos de manera más sencilla. El *rigging* no se aplica a todos los modelos, solo a los que van a necesitar una animación más compleja como los personajes. Se puede decir que este proceso

tiene como objetivo crear uniones que imiten las funcionalidades de un esqueleto real, aunque no es necesario replicarlo exactamente igual.

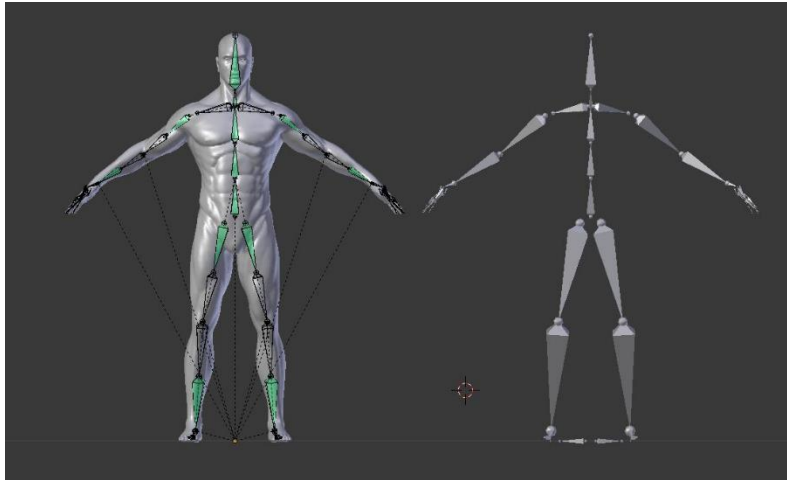


Figura 37. Ejemplo de un modelo y su esqueleto rigging
(Fuente: <https://blenderartists.org/t/rigify-help/532751>)

Para los personajes, principalmente, se crean dos conjuntos de rigging: el **corporal** y el **facial**. En videojuegos, para optimizar, el facial solo se usa en las animaciones de transición donde se le ve la cara al personaje.

El **rigging corporal** consiste en la construcción general del esqueleto que tendrá el cuerpo del personaje. La complejidad dependerá del nivel de detalle que necesite la animación. El modelo que se usa en el proceso debe tener una pose fija, normalmente en A o T. Para la pose A, llamada **A pose**, se coloca el cuerpo del modelo en forma de letra A con los brazos extendidos, pero ligeramente separados del torso y las piernas separadas. Para la pose T, llamada **T pose**, los brazos tienen que estar colocados formando una T. Pero esta pose es menos recomendable, ya que no es una pose en la que generalmente un cuerpo real se encuentre en estado de relajación, esto puede ocasionar que al trabajar con ella la malla en su punto de origen se encuentre muy estirada y al flexionarla forme huecos o pliegues extraños.

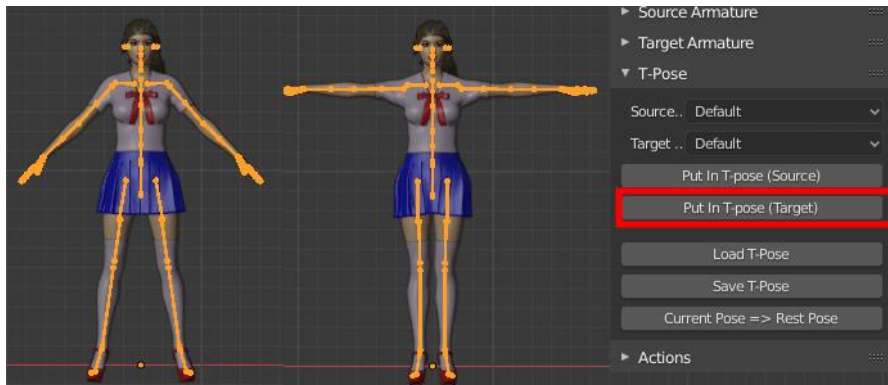


Figura 38. Ejemplo de A pose y T pose
 (Fuente: <http://diffeomorphic.blogspot.com/p/bvh-poses-panel.html>)

Una vez posado el modelo, se colocan unos puntos de unión llamados **joints** que funcionan como controladores de los huesos del esqueleto humano, y serán los que se utilicen para realizar las animaciones. Casi siempre se colocan en las posiciones de las articulaciones y partes del cuerpo que se puedan rotar o flexionar. La unión más importante, llamada **root joint**, se colocará en la parte más importante, y todas las demás dependerán de esta. Esto quiere decir que, si mueves o rotas esa unión, todas las uniones que dependan de ella, llamadas hijas, rotarán también acompañando a esa rotación. Por tanto, normalmente esta unión se sitúa en la cadera y las demás se crean siendo hijas de estas, o hijas de las hijas de esta unión y así sucesivamente, todas tendrán una unión padre menos la **root joint**. A todos los **joints** se les puede poner un nombre para que la lectura del esqueleto sea más clara y sencilla.

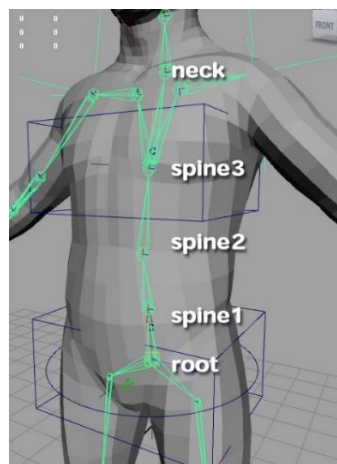


Figura 39. Ejemplo root joint y sus joints hijas
 (Fuente: http://www.mason3d.com/IK_spine_tutorial.html)

El **rigging facial** funciona de la misma manera, solo que se centra en el rostro y los músculos de la cara. Está pensado para captar las expresiones del personaje, según la finalidad de la animación este rigging será más o menos complejo. Es necesario conocer bien la anatomía y cómo funciona para conseguir crear un **rigging** realista y creíble. Ya que la parte de la

gesticulación de la cara es la que más sentimientos transmite, por eso el trabajo debe ser lo más minucioso posible. Los controladores de la cara pueden ser ligeramente diferentes, funcionan igual, pero pueden ser simplemente circulares, dependerá del programa que se utilice. Además, como el rigging anterior seguirán un sistema jerárquico que los conectará debidamente para un buen funcionamiento. Asimismo, para este *rigging*, se suele realizar una interfaz con los controladores faciales para posteriormente poder animar el rostro desde esa interfaz. Este tipo de interfaz también se puede realizar para el *rigging* corporal o partes de él si es necesaria.



Figura 40. Ejemplo rigging facial
(Fuente: <https://blenderartists.org/t/rigify-beginner-facial-rig-key-not-found-controller-not-showing-up/1285224>)

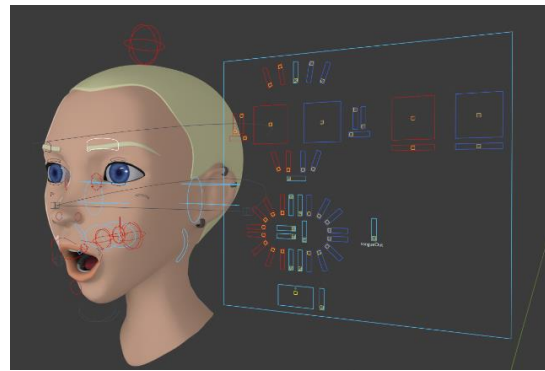


Figura 41. Ejemplo rigging facial y su interfaz
(Fuente: <https://shop04004.cafeteca.net/category?name=blender%20face%20rig>)

Una vez se terminen de crear todos los huesos del cuerpo y la cara, habrá que unirlos a la malla del modelo 3D, este proceso se llama **skinning**. Consiste en indicar que las uniones son los huesos del *skeleton* y la malla es la *skin* (piel). A partir de aquí el programa hará una interpretación de cómo debe doblarse el modelo por cada unión. Esta interpretación lo más común es que no acierte, y por eso se procederá a: crear otros controladores para simplificar el esqueleto (para manejarlo más intuitivamente con ellos, y no con todos los controladores que lleva el esqueleto), y hacer una distribución de pesos.

Principalmente hay dos sistemas de configuración para crear los controladores que agrupan los huesos del esqueleto: sistema IK o FK. Este paso no es obligatorio, la distribución de pesos sí lo es. Puede que haya producciones de animaciones muy cortas y/o mucho más sencillas que prefieran animar directamente con el esqueleto. Aunque esto no es lo más recomendable porque puede provocar problemas en el futuro como que se tarde más tiempo de lo esperado en realizar la animación. Normalmente, sobre todo para grandes producciones, es importante crear los dos para cada modelo que vaya a tener una animación compleja. Así el animador podrá intercambiar entre un sistema y otro, para usarlos dependiendo en cada situación lo que necesite para posar y mover al personaje. Por una parte, el **sistema FK** (*Forwardk Kinematics*) o

de cinemática directa, es el más sencillo, esta configuración hace que a la hora de animar al personaje el animador crea la pose del personaje desde los huesos más generales a los más particulares. Es decir, que los huesos hijos heredan el movimiento (posición, rotación y escalado) de los padres, se concatenan las rotaciones de los padres para calcular la de los hijos. Si se anima directamente con los controladores del esqueleto (los huesos), se estará usando el sistema FK que es con el que funcionan por defecto, pero habrá muchos más al no haberlos simplificado. Por otra parte, el **sistema IK** (*Inverse Kinematics*) o de cinemática inversa, es lo contrario que el anterior. Se posiciona, por ejemplo, en un brazo la posición final de la mano y el ordenador calcula la interpolación para colocar la cadena de huesos hasta la mano en su posible configuración. Este sistema hace que puedas animar de forma más libre ya que posicionas la mano sin que modifique el movimiento del codo y el hombro. En ambos sistemas se crean restricciones a los controladores para que no se puedan rotar en direcciones que no acompañan al movimiento natural. Por ejemplo, el codo no puede rotar hacia atrás, si se hiciese es porque lo requiere la animación de ese momento, pero para que sea más fácil se usan las restricciones. Si se necesitase porque el personaje es contorsionista esto se tendría en cuenta antes, en el diseño para crear el *rig* del personaje. Así se obtienen unos pocos controladores o una interfaz, que simplifican el esqueleto de huesos con las restricciones para facilitar la tarea a los animadores. Pero además para que estas restricciones se apliquen debidamente, es necesario realizar el siguiente proceso.

La **distribución de pesos** es un proceso que consiste en pintar de distintos tonos la malla que se va a animar para indicarle cómo y hasta dónde va a afectar que se doble por esa unión colocada previamente. Así cada vez que se seleccione una de las uniones, marcará que sección del modelo se verá afectada. Hay dos formas de pintar los pesos sobre el modelo:

- **Blanco y Negro.** Es el método más sencillo para marcar los pesos, se realiza un degradado con estos dos colores. El color que más se acerque al blanco será la zona más propensa a doblarse y la que se acerque más al negro, la que permanezca más inmóvil.
- **Colores.** Se pintan los pesos con un gradiente de color que va de cálidos a fríos, del rojo, pasando por el amarillo y el verde hasta el azul. La parte pintada de rojo o que más se acerque a este color será la más propensa a doblarse y la que más se acerque al azul la más inmóvil.



Figura 42. Ejemplo distribución de pesos en Blanco y Negro

(Fuente: captura del minuto 1:19 del vídeo https://www.youtube.com/watch?v=oGZWeKSB3aU&ab_channel=AlejandroCelis)

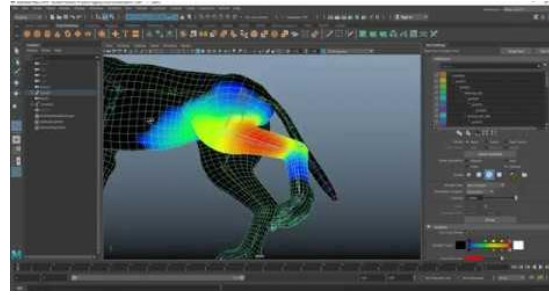


Figura 43. Ejemplo distribución de pesos en Color

(Fuente: captura del minuto 24:11 del vídeo https://www.youtube.com/watch?v=AHcz4wh3Xas&ab_channel=JuanmaCode)

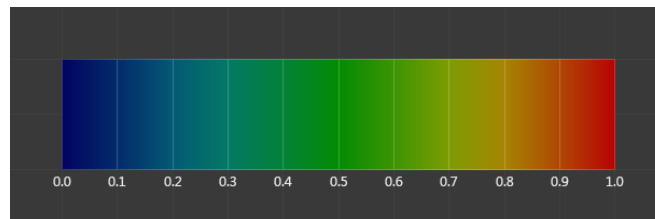


Figura 44. Código de colores del degradado de pesos en Blender

(Fuente: https://docs.blender.org/manual/es/3.5/sculpt_paint/weight_paint/introduction.html)

Una vez se pinten correctamente cada *joint* con su distribución de pesos, el modelo estará listo para ser animado.

3.5.4. Materiales y Texturizado

El texturizado es un proceso que se puede hacer en paralelo a la retopología, puesto que solo necesita la malla del modelo original con todos sus detalles. Para este proceso se pueden usar materiales, distintos tipos de texturas o materiales inteligentes. Puedes usar texturas de imágenes reales para obtener los resultados deseados, que serán aplicadas al objeto mediante un proceso de proyección. Hay distintos tipos de programas que trabajan principalmente de 3 formas distintas:

- Están los **sistemas PBR**, son los más avanzados, y otorgan al modelo las propiedades físicas de los materiales mediante materiales inteligentes y texturas, también se puede pintar, pero pintas el material, como si tuvieses las propiedades de un metal o una madera en el pincel.
- Programas **de pintura**, con ellos se simula pintando en las distintas texturas los materiales, pero no otorgas al objeto las propiedades de esos materiales, es una simulación.

- Y programas **con materiales** ya predefinidos que puedes arrastrar al objeto o seleccionar en qué polígonos se va a aplicar, estos programas también puedes modificar algunas características de estos materiales o crearlos, pero solo con los parámetros que tenga disponibles, igual que puedes proyectar texturas en el modelo que te ayuden a crear materiales más complejos y creíbles.

Algunos programas tienen varias de estas funciones disponibles para que puedas elegir según cada proyecto lo que necesite y/o combinarlas.

El texturizado consiste en una serie de imágenes que se aplican a los modelos y sirven para: dar color, relieve, simular un tipo de material cuando incide la luz sobre la superficie del objeto, etc. a los modelos. Es necesario tener unos conceptos básicos de imagen digital para realizar el texturizado de un modelo 3D. Estos conceptos básicos incluyen:

- ¿Cómo funciona el **sistema de coordenadas** de un espacio tridimensional? Se representa con los ejes x, y, y z la anchura, la altura y la profundidad. Como se ha mencionado anteriormente, dependerá de cada programa cómo nombra a cada eje, en algunos el eje z es la altura, y el eje y la profundidad; y en otros el eje z es la profundidad, y el eje y la altura. Para las texturas, lo importante es saber cómo funciona, puesto que algunos mapas de textura utilizan esta información para saber cómo afecta la luz al incidir en la superficie del objeto que tiene la textura. Ya que puede que el mapa cree un relieve en el modelo o simule un tipo de material u otro que hará que al incidir la luz según el ángulo se vea distinto. Asimismo, servirá para comprender cómo funciona el mapeado UV más adelante.
- Los **perfiles de color RGB y CMYK**. Principalmente en el ordenador se trabaja con estos dos perfiles de color. El primero está diseñado para trabajar en digital, y el segundo para trabajos que vayan a ser impresos. Por una parte, RGB son las siglas en inglés de los colores rojo, verde y azul respectivamente. Estos son los colores primarios de la luz. El funcionamiento de este perfil consiste en la síntesis aditiva, se mezclan colores para obtener el color deseado. Generalmente las texturas se trabajan con este perfil. Por otra parte, CMYK son las siglas en inglés de los colores cian, magenta y amarillo respectivamente, y la K representa al negro. Estos son los colores primarios de la síntesis sustractiva.
- La **profundidad de bits** indica los niveles por canal que tiene cada color, es decir, el espectro de color que abarca con esa cantidad de bits. A mayor número de bits, mayor es la paleta de colores que es capaz de representar. Esto también influye a la hora de

guardar un archivo de imagen, ya que a más profundidad más pesará el archivo. Cada bit equivale a dos niveles que pueden ser 0 ó 1. Los valores de la profundidad siempre serán divisores o múltiplos de 8. Por ejemplo, una profundidad de 8 bits puede representar 256 niveles por canal de color, una de 12 bits puede representar 4.096 niveles por canal de color, y una de 16 bits puede representar 65.536 niveles por canal de color.

- Los **formatos de imagen** que existen y qué características tiene cada uno. Existen gran variedad de formatos en los que se puede guardar un archivo de imagen. Cada uno tiene unas características y es importante conocerlas para saber cuál es conveniente en cada caso. Normalmente para texturas los más útiles son los descritos a continuación. **JPG** que es muy manejable y ocupa poco espacio gracias a su método de compresión. **PNG** cuya calidad es mayor y con él se puede representar el canal *alpha*, que son las transparencias en una imagen, por ello es más pesado. Además, este formato solo soporta el perfil de color RGB, así que no se puede cometer el error de escoger el perfil erróneo. **TIFF** que es un formato muy versátil gracias a que su compresión no es destructiva. Contiene etiqueta que indican a los programas cómo debe ser leída la imagen, esto optimiza mucho los tiempos de carga. Asimismo, puede conservar varias imágenes en un mismo archivo, como las capas de Photoshop. Por último, **TGA** es un formato que permite guardar información valiosa como el canal *alpha*, los valores gamma y los metadatos en forma de texto. Puede conservar bastante bien la calidad de la imagen.
- ¿Qué es la **resolución** y cómo afecta esto a las imágenes? A mayor resolución mayor calidad se obtiene en la imagen, pero también pesa más el archivo. Cuando una imagen tiene poca resolución y se aprecia, se dice que está pixelada, y se debe evitar. La resolución en una imagen es la cantidad de píxeles que tiene por unidad de longitud. Se mide en píxeles por pulgada (ppp).
- ¿Qué es un **tile** o una textura *tileable*? Muchas veces se necesita usar una textura cuadrada que pueda cubrir una superficie pegando una imagen a esa misma imagen, como un mosaico, esto es a lo que se dice que es *tileable*. Es una imagen que puede repetirse al lado de sí misma formando un patrón.
- ¿Qué significa que una textura sea sin costuras o **seamless**? Cuando se crea un mosaico con una textura cuadrada, es decir, con una textura *tileable*, y además no se notan los cortes de una imagen a otra, se dice que no se ven las costuras. Para que esto funcione

la textura debe acabar y empezar de la misma forma en sus bordes, así se consigue que no se vean las costuras, a esta textura tileable se le dice que es *seamless*.

Una vez se tienen estos conceptos claros. Se puede explicar qué tipos de texturas hay y qué utilidades tiene cada una. Las propiedades de las texturas son las mismas que se usan para los materiales, la diferencia es que el material suele aplicarlo a los polígonos seleccionados enteros o al objeto que se lo apliques; y las texturas se puede pintar y elegir qué parte actúa y cómo de cada mapa, ya que se guardan en imágenes que se aplican al modelo. Generalmente se usa el nombre de mapas o *maps* para referirse a este tipo de archivos en diseño 3D. Algunos tipos de mapas que existen son:

- ***Diffuse map o Base color***. Se trata de una imagen que contiene la información del color base que tendrá el objeto o mejor dicho como dispersará la luz ese color por su superficie. Este mapa es bastante antiguo, ahora normalmente se usa el *albedo*. No tiene por qué usar solo colores planos, puede tener tonos sombreados o incluso una fotografía. Este mapa controla la forma en que la luz se dispersa por la superficie del objeto. Define cómo la superficie absorbe y refleja la luz, esto ayuda a agregar detalles y complejidad a la apariencia del objeto.
- ***Albedo map***. Albedo es un término que se usa en física para referirse a la proporción de luz que es reflejada por un objeto. Este mapa como el anterior otorga el color base a un objeto antes de ser iluminado o sombreado. La diferencia con el *diffuse map* es que este define el color que parecería tener el objeto bajo una luz brillante distribuida uniformemente. Por ejemplo, un objeto con este mapa completamente blanco y bajo una luz difusa, parecería ser de un blanco mate brillante; y con este mapa completamente negro y bajo la misma luz difusa, parecería ser de un negro mate oscuro. En ocasiones, el *diffuse map* y el *albedo* los cuentan como uno solo en algunos programas.
- ***Specular map***. Los niveles de este mapa se utilizan para definir cómo de brillante es la superficie del modelo. Cuanto menor sea el valor, más mate parecerá el material de la superficie, y cuanto mayor sea el valor, más brillante parecerá. Así que indica las zonas que pueden tener mayor o menor brillo. Si usas la información de color del *diffuse map*, el mapa estará muy saturado, y cuanto más saturado esté el mapa, más colorido se verá el modelo en el render. Pero si necesitas un color más neutral puedes usar la información inversa de color del *diffuse map*.
- ***Metalness map o Metallic***. Como su nombre indica, este mapa otorga las propiedades de un metal al objeto. Es una textura en blanco y negro, el blanco hace que se comporte

como un metal y el negro no. Ayudará a diferenciar las propiedades que diferencian a los materiales metálicos, conductores de electricidad, como el oro, la plata, bronce, cobre aluminio, etc. de los dieléctricos, materiales aislantes, como el plástico, la goma, madera, cerámica, tela, etc. Los materiales metálicos no tienen luz difusa generalmente, hay una parte de luz que absorben y el resto la reflejan; son más reflectantes gracias a que su superficie es más lisa, esto hace que brillen más; y los metales se corrompen, es decir, les puede salir óxido o manchas verdosas como al cobre y al bronce, así que en algunas zonas pueden tener el aspecto de material dieléctrico.

- **Roughness map.** Determina la rugosidad o la suavidad de la superficie del modelo, afectando a cómo se ven los reflejos en el objeto, si se ve más nítido o desenfocado.
- **Glossiness map.** Modula la cantidad de brillo que tiene un objeto cuando incide en él la luz especular. Así se puede tener puntos en la superficie del objeto que reflejen más o menos luz especular que otros.
- **Bump map.** Este mapa crea la ilusión de profundidad en la superficie del objeto utilizando un truco de iluminación, pero el detalle que crean es falso, no se añade resolución extra al modelo. Hoy en día este mapa no suele usarse, se usa el *normal* o el *height*. Estas imágenes usan una escala de color de grises para indicarle dos cosas, arriba o abajo según si el gris se acerca más al blanco o al negro respectivamente, creando así la ilusión de profundidad. Son geniales para pequeños detalles como poros, arrugas en la piel o grietas en superficies sólidas. El problema que tienen es que el engaño se pierde muy fácilmente dependiendo del ángulo de la cámara. Al no modificar la malla del modelo, si se sitúa la cámara de perfil a la superficie donde esté creada esa ilusión, no se apreciará el volumen ni la profundidad.
- **Normal map.** Son como la versión mejorada de los *bump*, pero el detalle que crean sigue siendo falso. A diferencia del *bump*, este no usa una escala de grises, sino que usa la información RGB que corresponde con los ejes x, y y z en el espacio 3D. Esta información le dice la dirección exacta en la que están orientadas las superficies del mapa para cada polígono, es decir, le dice al polígono cómo debe ser sombreado cuando la luz lo alcance. Estos mapas se pueden crear con la técnica *bake*, que consiste en transmitir información de un modelo *high poly* a un *low poly*, que se aplica al modelo. Aunque hay métodos más sencillos como trabajar con una imagen normal en un programa de edición de imágenes o de dibujo y luego aplicar el filtro adecuado que la convierta en un mapa de normal.

- **Displacement map.** Sirve para crear detalles adicionales en los modelos. Desplaza la malla a la que ha sido aplicado. Aunque normalmente hoy en día usan más el *height map*. Al igual que el bump este utiliza la escala de grises para indicar la información, pero este mapa sí deforma la geometría del modelo. Normalmente, para poder utilizar este mapa sin problemas se debe realizar geometría real en el modelo y después exportar de él el mapa, es por eso por lo que se debe guardar el modelo *high poly* que es el que tiene más detalles. El resultado será bastante superior a los anteriores, al no ser un engaño. Dará igual el ángulo de la cámara. Aunque al no ser una ilusión, el ordenador requerirá más esfuerzo y tiempo para procesar esta información, esta es la desventaja que tiene frente al *bump* o el *normal*.
- **Height map.** Es casi lo mismo que el *displacement map*. Se usa en el mismo contexto que el anterior solo que este no desplaza la superficie creando triángulos entre los *texels* (unidad mínima de una textura aplicada a una superficie), sino que modifica la altura de los vértices de la malla para crear los detalles.
- **Emissive map.** Este mapa otorga la capacidad al objeto o a la parte que se le aplique de emitir luz. Se puede escoger el color de la luz que emitirá el objeto. Esto puede servir para simular que los faros de un coche están encendidos o que es una joya mágica que brilla, por ejemplo.
- **Ambient Occlusion map.** Se usa para indicarle al modelo qué áreas deben recibir mayor o menor luz indirecta. La luz indirecta viene dada por la iluminación ambiental y los reflejos. Este mapa indica que zonas están más hacia dentro o hacia fuera, como si fuesen grietas, y, por tanto, donde llega más o menos luz. Este mapa utiliza la información en escala de grises, donde el blanco es que recibe más luz y el negro el que menos.
- **Anisotropic map.** Este mapa controla la forma de los reflejos especulares de las luces en tiempo real. A menudo se utiliza para sombrear el cabello de los personajes, y crear los típicos reflejos que se ven exagerados en el estilo manga o anime en el pelo. Los reflejos de este mapa se caracterizan por ser trazos direccionales largos en vez de los típicos circulares.
- **Alpha.** Este mapa se usa para indicar al programa 3D que partes del modelo son visibles y cuáles no. Antiguamente cuando se usaba el *diffuse map* este solo tenía las propiedades de color RGB, y este canal servía para aportar el canal *alpha*. Es por esto por lo que ahora es más utilizado el *opacity map* que es una versión mejorada de este. Este mapa también usa la escala de grises, siendo de color blanco lo que vaya a ser

totalmente opaco y negro lo transparente. Se puede jugar con distintas tonalidades de gris para dejar pasar una mayor o menor cantidad de luz a través del modelo. Este mapa puede ser muy útil para elementos como el césped, las hojas de los árboles, ya que no vas a modelar una a una todas las hojas de las plantas, o también para crear efectos especiales como el humo o las nubes del cielo.

- **Opacity map.** Este mapa funciona como el anterior, es una versión más avanzada. La diferencia entre los dos reside en el que el anterior para otorgar la transparencia a un objeto usa la codificación del canal alpha en una textura RGBA, mientras que este, imita más propiedades físicas de los objetos transparentes como puede ser el cómo dejan pasar la luz y cómo afecta esa luz a otros objetos del entorno. Sigue usando el mapa en escala de grises que indica si no deja pasar nada de luz siendo opaco, si el objeto solo es traslúcido, o si deja pasar toda la luz haciéndolo transparente. Se usa en las hojas de los árboles para no tener que modelarlas todas, los vidrios, etc.
- **Flipbook.** Sigue el mismo funcionamiento que los libritos donde los animadores dibujan cada una de las ilustraciones y al pasarlas rápido se crea la ilusión del movimiento del dibujo. Se trata de un archivo que contiene un conjunto de imágenes que al reproducirse sucesivamente forman una animación. Al aplicarlo a un objeto, se verá que la animación se repite infinitamente, como un loop animado. Este mapa combinado con el *alpha*, se puede simular el efecto de humo, por ejemplo. Existen otras opciones a parte de este mapa para conseguir el mismo efecto, pero con archivos de vídeo para textura.

Como se puede observar, hay muchos tipos diferentes de mapas, que ayudan a crear diferentes detalles al objeto, y eso que aquí solo se explican unos cuantos. Algunos sirven para funciones muy parecidas y ahí el artista decide cuál es el mejor para ese proyecto, otros se han dejado de usar con el tiempo porque existen versiones mejoradas. Pero si se hiciese un juego para una consola antigua puede que tocarse usarlos, ya que son los que soportan, así que conviene al menos conocerlos. En la mayoría de las ocasiones, no vas a usar en un objeto solo un tipo de mapa, sino que se usan varios de ellos combinados en un mismo modelo para obtener el resultado deseado.

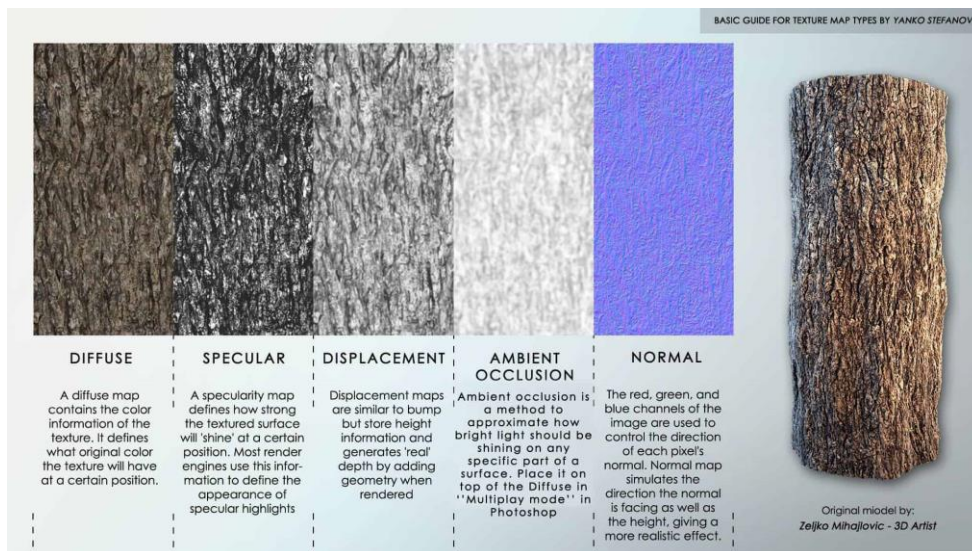


Figura 45. Ejemplo mapas de textura para un tronco de árbol
(Fuente: <https://cgtricks.com/basic-guide-for-textures-map-types-yanko-stefanov/>)

Ahora que se han explicado algunos tipos de textura que existen, se verá cómo funciona el mapeado UV. El **mapeado UV** sirve para aplicar correctamente, en la posición que se quiera situar la textura 2D al modelo 3D. Mediante este proceso se pueden aplicar las texturas sobre el modelo 3D por proyección. Se utilizan las letras U y V para las coordenadas 2D de la textura, ya que las letras x, y y z se usan para el espacio tridimensional. El proceso consiste en asignar píxeles de la imagen al polígono del modelo, una vez asignados, el modelo estará listo para el render y usará esa información para darle ese aspecto en la imagen final. Esto solo es necesario usarlo en texturas complejas, los materiales simples que se aplican al objeto no necesitan una proyección, ya que aplican la información por toda la malla uniformemente. Tampoco los necesitas si usas técnicas de pintura que simulan las propiedades del material, porque para ellas usas el *vertex color* y pintas los polígonos guardando la información en los elementos de la malla, aunque tendrás la opción de exportar la información como textura, que es lo más recomendable.

Para poder texturizar el modelo 3D se debe definir porque zonas se separa para crear las texturas, es decir, se tiene que crear costuras por donde se abrirá el modelo para generar la textura en 2D de sus superficie, a esto se le denomina *unwrap*, que significa desenvolver. Para entenderlo de una forma más sencilla, imagina que funciona como el *papercraft*, las figuras 3D en papel que se crean haciendo pliegues y uniéndolos con pegamento.

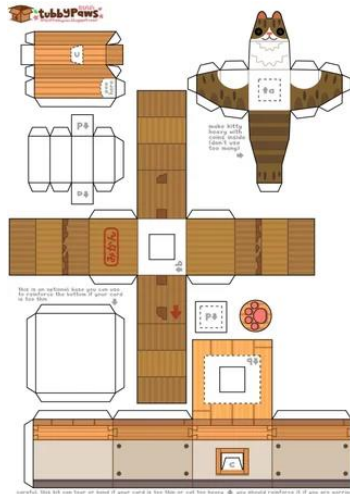


Figura 46. Ejemplo de plantilla de papercraft

(Fuente:

<https://ar.pinterest.com/pin/851391504537390582/>)



Figura 47. Ejemplo de papercraft montado

(Fuente:

<https://www.instructables.com/Cat-in-Box-Papercraft/>)

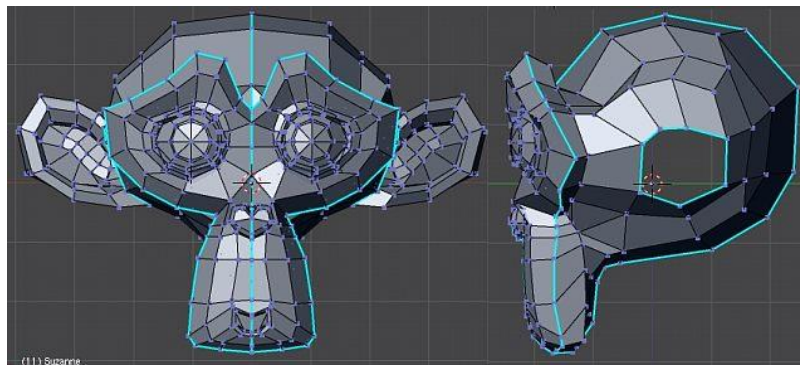


Figura 48. Ejemplo costuras de Suzanne, mona de Blender

(Fuente: <https://docs.blender.org/api/html/x5336.html>)

Pues igual, pero con el modelo 3D, marcas las costuras por donde se abrirá para extenderse en una imagen 2D. Estas costuras se hacen en las aristas, para evitar errores, normalmente se escogen aristas que estén escondidas para que no se vean cortes al aplicar la textura en el modelo. Los modelos se pueden dividir en varias piezas distintas, que se llaman islas, generalmente separadas por el tipo de material que se le aplique a cada una, aunque esto dependerá del proceso y necesidades del proyecto en cuestión. Se deben hacer costuras hasta que el modelo en 2D quede lo más extendido y plano posible para evitar errores, existen herramientas como texturas *checker* para ver esos fallos. Son texturas que tienen dibujadas cuadraditos en blanco y negro o de colores que ayudan a ver si están bien colocadas las costuras, para ello se debe comprobar que los cuadraditos en el modelo 3D se vean de una forma coherente, ordenados y del mismo tamaño.

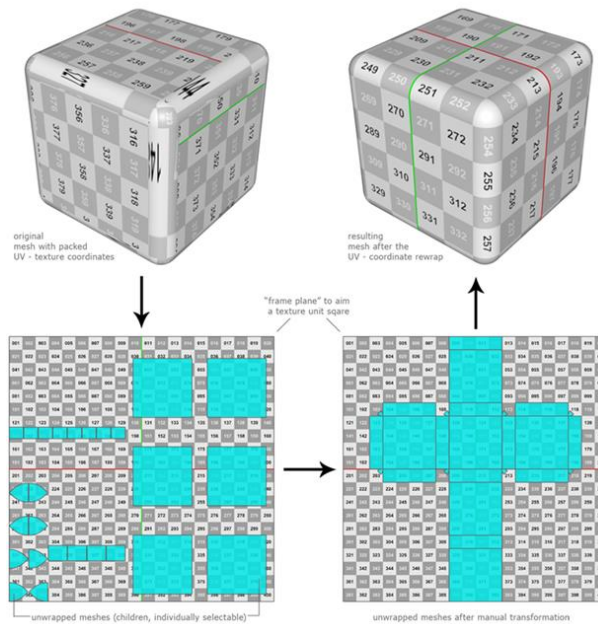


Figura 49. Ejemplo de cómo usar la textura de chequeo (izda. errores y dcha. arreglada)
 (Fuente: <https://wiki.mcneel.com/labs/uvwrap>)

Una vez se ha realizado el unwrap, ya está listo para exportar las texturas y/o hacer el *bakeo* de las mismas en el programa de texturizado elegido. A partir de aquí se aplicará lo aprendido en este mismo apartado del documento, más arriba, sobre cada tipo de textura y para qué sirven, para texturizarlo. Luego se procederá a aplicarle la textura al modelo 3D, pero con los diferentes mapas y materiales aplicados y se verá el resultado final del modelo.

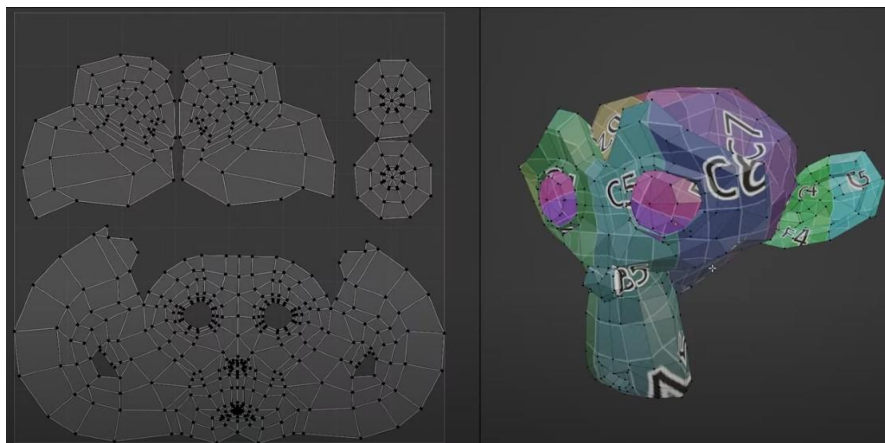


Figura 50. Ejemplo unwrap en el espacio UV de Suzanne, mona de Blender
 (Fuente: captura del minuto 1:07 del video
https://www.youtube.com/watch?v=jnzhNdWoXMq&ab_channel=GrantAbbitt)

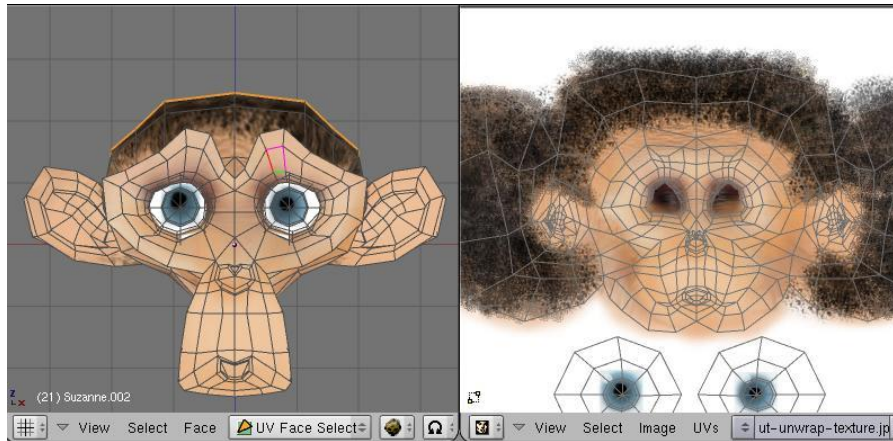


Figura 51. Ejemplo aplicar texturas a Suzanne, mona de Blender
 (Fuente: <https://docs.blender.org/api/html/x5336.html>)

3.5.5. Partículas

Los sistemas de partículas son una técnica que se utiliza en gráficos generados por computador y videojuegos. Se pueden usar aplicando un gran número de pequeños *sprites* (pequeñas texturas 2D que se integran en la escena), modelos 3D u otro tipo de objetos, para simular determinados efectos. Se basan en los sistemas de partículas de la mecánica básica, de ahí su nombre. Generalmente tratan de simular sistemas caóticos, fenómenos de la naturaleza o procesos causados por reacciones químicas como pueden ser: el fuego, explosiones, humo, agua en movimiento, chispas, hojas que caen, rocas que caen, nubes, niebla, nieve, polvo, estrellas y galaxias, brillos, efectos mágicos, etc. Asimismo, se utilizan para recrear algo que tenga muchas hebras como el pelo o la hierba.

En los programas 3D se colocan objetos que actúan como fuente de las partículas, y a los que se les atribuye parámetros para controlarlas. Estos objetos se llaman emisores de partículas. De este emisor provienen las partículas, su posición determina donde se generan y hacia donde se mueven. Cualquier modelo 3D puede ser utilizado como emisor. Existen diferentes tipos de emisores:

- **Omni.** Emite partículas desde el centro a todas las direcciones.
- **Volume.** Con este emisor las partículas parten de una figura geométrica concreta. Se puede elegir el tipo de figura y configurar sus parámetros.
- **Surface/Curve.** Para este emisor se crea primero un objeto curva o un modelo determinado. Luego se selecciona y se elige de cuál de los dos se trata. Las partículas

nacerán partiendo de ese modelo o curva. Es muy útil para hacer que las partículas nazcan de una forma muy determinada.

Los parámetros que las controlan pueden ser: la velocidad a la que se generan desde la fuente, la esperanza de vida, el número de partículas por unidad de tiempo, la forma de las partículas, la velocidad inicial con la que se generan y la velocidad final con la que desaparecen, el color o colores si se eligen varios, el comportamiento casual (el recorrido que tienen), influencias de agentes externos sobre ellas (como el viento o la gravedad), la colisión con otras superficies, y otras propiedades físicas y químicas que se pueden simular.

Para que el ordenador no tenga que hacer tantos cálculos cuando se están configurando, existen dos fases distintas para las partículas:

- **Fase de actuación y simulación** la que se utiliza mientras aún se trabaja en la escena, esta requiere menos cálculos y usa simulaciones de las partículas, pero sin ser los modelos o texturas seleccionados, para que el programa no necesite tantos recursos.
- **Fase de render** que se utiliza cuando se obtienen los resultados deseados en la fase anterior. En esta fase las partículas se renderizan una a una como los objetos o texturas seleccionadas. Estarán programados para que siempre se vean frontales a la cámara.

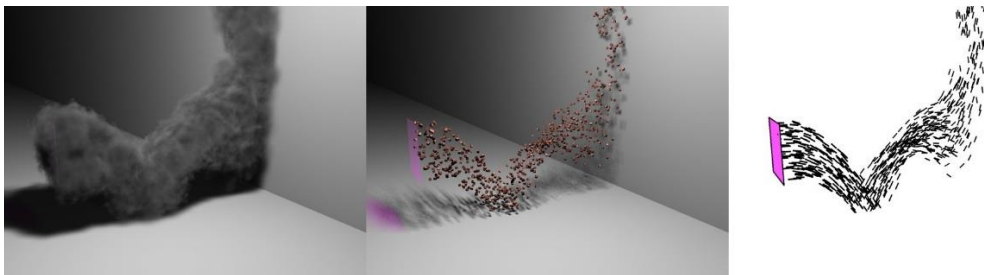


Figura 52. Ejemplo fase de render (dos primeras) y fase de simulación (última)
(Fuente: https://es.wikipedia.org/wiki/Sistema_de_part%C3%ADculas_%28software%29)

Las partículas no tienen por qué ser animadas necesariamente, pueden ser estáticas, como el cabello. Existen programas especializados solo para la creación, configuración y/o animación de partículas.

3.5.6. Iluminación y Sombreado

En los programas de modelado hay una colección de herramientas y técnicas que se usan para simular la luz en un entorno tridimensional digital. La iluminación es un proceso muy importante para la visualización de los modelos, si no se agregase ningún tipo de luz, en el render (la imagen

final) se vería toda la imagen en negro, no se apreciaría el modelo. Asimismo, es muy importante conocer cómo funciona la luz en el mundo real gracias a la física para poder usar estos conocimientos para simularla en este tipo de entornos, igual que se usa en fotografía y cine. Con suficientes conocimientos de la luz y cómo funciona se pueden llegar a crear imágenes muy realistas en entornos 3D, como si fuesen una fotografía de un espacio en el mundo real. Muchas veces la mala implementación de esta hace que toda la escena quede poco o nada convincente, con un acabado plano que no se acerca a la realidad. La calidad del acabado del trabajo final depende mucho de este conocimiento. La iluminación tiene un papel muy importante en la composición y la puesta en escena, que es uno de los principios básicos de la animación que se vieron en el apartado 3 de este capítulo (*3.3 Principios de la animación*).

La configuración de luces que se contemplan día a día en el mundo real es una combinación compleja de luz ambiental sumada a fuentes de luz adicionales. Gran parte de esa luz a lo largo del día la proporciona el Sol, que es una estrella en constante reacción de fusión, lo que hace que libere grandes cantidades de energía en forma de luz y calor que llegan a los cuerpos del sistema solar.

Para otorgar a la escena 3D de luz, además de los conocimientos físicos de la luz y cómo funciona como ya se ha mencionado, se necesita establecer una combinación de fuentes de luz para la escena. Los artistas tienen una variedad de fuentes de luz, herramientas y técnicas para poder cubrir cada necesidad según el tipo de escenario e iluminación que deseen imitar. Para poder imitar lo que ocurre en el mundo real se añaden varias luces, tantas como sean necesarias para la escena y de los tipos correspondientes configuradas según lo que intente imitar cada una de ellas en la escena. La suma de todas estas luces será la iluminación final de la escena, como pasa en el mundo real. Se siguen las mismas técnicas para posicionarlas y configurarlas que las que se usan en estudios de grabación, de fotografía o en el teatro. Como puede ser la técnica de **los tres puntos** donde se coloca una luz principal, una de relleno y una trasera para dar un efecto de profundidad en el escenario.

La mayor cantidad de luz que recibe una superficie proviene de fuentes directas de emisión de luz, es decir las que van desde el foco de luz directamente al objeto (**iluminación directa**). La luz reflejada por otras superficies de la escena que reciben los objetos y crean brillos en ellos, se llama **iluminación indirecta**. Algunas veces la luz es bloqueada por otros objetos de la escena, entonces, se crean **sombras** por el objeto que bloquea la luz a otros objetos o partes de la escena. Dado un rayo de luz que incide sobre un punto de una superficie se calcula cuánto de ese rayo de luz será reflejado y en qué dirección. En gráficos por computador, existen dos tipos

de modelos: **modelos de iluminación o lighting**, que otorgan los valores que determinan el color de una superficie; y los **modelos de sombreado o shaders** que determinan donde se aplica el modelo de iluminación y qué información se recibe de este. A su vez, los modelos de iluminación se dividen en dos categorías: iluminación local e iluminación global.

En el modelo de **iluminación local** y de **shading** (como se colorea la superficie) solo se considera la luz incidente, la posición del observador y las propiedades de los materiales de las distintas superficies para obtener el color de los objetos. Este modelo es el que se usa en videojuegos, ya que es un método rápido y simple para calcular la cantidad de luz, que no genera una mala experiencia de juego porque la computadora puede calcularlo rápido. Se basa en 3 parámetros que se suman para obtener la luz final que ilumina el objeto:

- **Ambient.** La luz ambiental que es la luz proveniente del entorno. Ilumina el objeto de manera uniforme en todas las direcciones y lo refleja de la misma manera, obteniendo un color uniforme global en la superficie del objeto.
- **Diffuse.** La iluminación no brillante y sombras. Considera el ángulo de incidencia de la luz sobre la superficie para obtener, en función de la orientación del foco de luz, una iluminación que varía sobre la superficie.
- **Specular.** El punto de brillo del objeto que se crea por las reflexiones directas sobre la superficie del objeto provenientes de la fuente de luz.

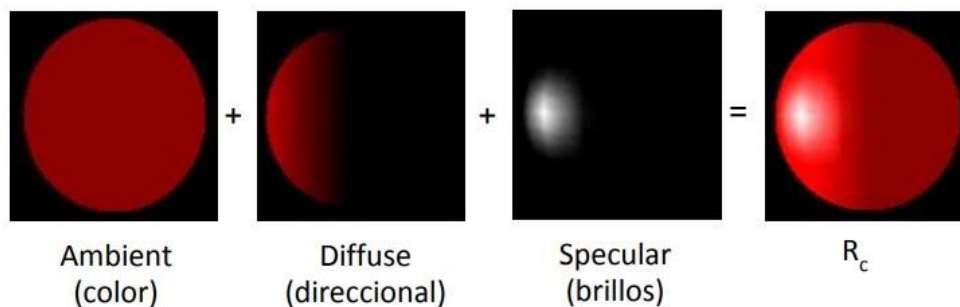


Figura 53. Modelo de iluminación local

(Fuente: captura de las diapositivas del tema de Render de la asignatura de Modelado y Animación por Computador del grado de Ingeniería Multimedia por Miguel Davia Aracil)

A este modelo también se le puede añadir un cuarto parámetro, **opacity** que se sumaría si es necesario crear transparencias para el objeto.

En cambio, en los modelos de **iluminación global** se tiene en cuenta la interacción de la luz proveniente de todas las superficies de la escena, a lo que se denomina **radiosity**. Estos modelos buscan acercarse al realismo, se usan en fotografía y en cine, es un modelo más avanzado. Dos de estos modelos son:

- **Ray-tracing** o trazado de rayos que es un algoritmo de síntesis de imágenes 3D. Se trazan rayos desde la cámara del render hasta la escena para calcular la luz en la imagen final. Se calculan las intersecciones del rayo con los diferentes objetos de la escena y aquella intersección que esté más cerca del observador determina cuál es el objeto visible con un proceso de sombreado. Tiene en cuenta reflexiones, refracciones o sombras proyectadas. Simula la reflexión de la luz una sola vez para cada superficie difusa, así que un objeto de la escena puede ser iluminado por otros a su alrededor. Las imágenes resultantes son muy reales, tanto que son demasiado perfectas, pero es un cálculo muy lento.
- **Radiosity**. Es la energía radiante, la cantidad de energía que irradia una superficie por unidad de tiempo. Este método representa el desarrollo de modelos basados en propiedades físicas, el uso de métodos computacionales más rigurosos y la continua tensión entre interactividad (videojuegos) y el realismo por computación. Toda la energía emitida o reflejada por las superficies es tomada en cuenta por la reflexión o absorción en las otras superficies.

Estos dos métodos se pueden usar de manera combinada para obtener mejores resultados.

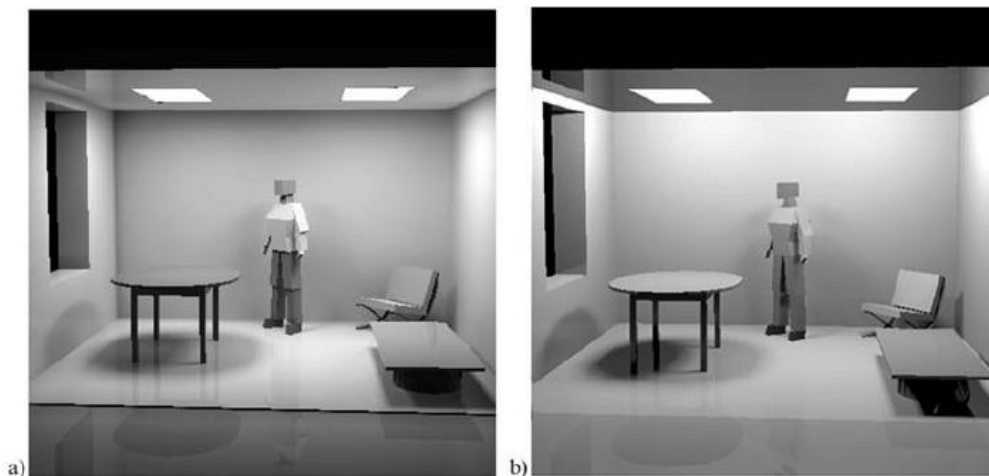


Figura 54. Imagen a) creada con Radiosity y Ray-tracing y b) solo Ray-tracing
(Fuente: https://www.researchgate.net/figure/a-Lightscape-image-created-using-a-radiosity-and-ray-trace-solution-b-DIRSIG-image_fig9_251190152)

Ahora que se ha explicado cómo se calcula y funciona la luz en entornos tridimensionales digitales. Se verán los tipos de focos de luz (objetos tipo luz) que se pueden colocar en una escena 3D para iluminarla:

- **Luz puntual (Point Light)** es una luz que sale de un punto del espacio. Es una luz omnidireccional, emite luz en todas las direcciones. Tiene parámetros que pueden

cambiar la distancia en la que se atenúa la luz, su color, su potencia, el número máximo de rayos rebotan, y si se quiere que cree sombras. Se puede usar para iluminar como lo haría una bombilla, un led o una vela situadas a una cierta distancia, por ejemplo.

- **Luz direccional** (*Directional Light*) es un haz de rayos paralelos. La posición del foco no es importante para el resultado, solo la dirección de los rayos. Este objeto tiene una línea discontinua que se puede rotar para cambiar la dirección de los rayos de luz. Es la única luz que influye sobre el aspecto del cielo y el horizonte de la escena. Tiene parámetros para cambiar su color, la intensidad, el ángulo, el número máximo de rayos rebotan, y si se quiere que cree sombras. Se usa para simular el Sol o la Luna.
- **Luz de foco** (*Spot Light*) es un cono de luz que parte de un punto. Funciona como un foco de un escenario o la luz de una farola o de un flexo o una linterna, puedes cambiar la dirección rotando el objeto, también se puede escalar para modificar la apertura del foco y cambiar el ancho del degradado de la sombra en el borde. Tiene parámetros para cambiar el color, la potencia, el radio y el fundido del borde, el tamaño del foco, el número máximo de rayos rebotan, y si se quiere que cree sombras.
- **Luz de área** (*Area Light*) simula una luz proveniente de una superficie. Tiene parámetros para cambiar el color, la potencia, la forma de la superficie que emite la luz, su tamaño, el ángulo de dispersión del haz de luz, el número máximo de rayos rebotan, y si se quiere que cree sombras. Las luces que se pueden simular son las luces de estudio que usan los fotógrafos, una lámpara fluorescente, un panel retroiluminado o la de una ventana, por ejemplo.
- **Luz ambiental** (*Ambient Light*) es una luz omnidireccional desde su ubicación en el espacio y además la luz brilla de manera uniforme en todas las direcciones desde todas las direcciones. Esta luz no genera sombras. Se puede usar para simular luz ambiental en una habitación que tenga una luz directa (una lámpara) y una luz indirecta (la luz que se refleja de la lámpara a las paredes y el techo).
- **Luz volumétrica** (*Volume Light*) es una luz que tiene una representación visual del espacio que ocupa, es decir, tiene un espacio limitado en el que actúa. Emite rayos omnidireccionales desde un punto, pero solo ilumina superficies dentro de un volumen. Las luces volumétricas no crean sombras de hardware. Se puede usar como luz negativa para eliminar o disminuir la iluminación o para aclarar las sombras.

3.5.7. Animación

La animación es el movimiento expresado con imágenes que no son tomadas directamente de la realidad. La ilusión del movimiento se consigue disponiendo de forma consecutiva los fotogramas. Como ya se comentó en el apartado 3 de este mismo capítulo (3.3 *Principios de la animación*), se tienen en cuenta los mismos **principios básicos de la animación** tradicional en la animación por computador o en cualquier otro tipo, como la realizada con plastilina, por ejemplo. Estos principios han sido adaptados a cómo se trabaja en la animación por computador.

Una vez realizado el *rigging* y *skinning* del modelo o modelos, se puede comenzar a animar. Primero se debe hacer una comprobación de los movimientos. Para ello, se usa el *rigging* ya realizado para detectar cualquier anomalía. En caso de que hubiese algún problema con los movimientos, que no se ejecuten como deberían, o en la malla, que no se deforme correctamente con los movimientos, aún habría tiempo para arreglar esos fallos.

A continuación, se **configura la escena** con todo lo necesario, se indican: las unidades que se utilizarán, como la rotación en grados; los FPS; y la duración de la animación en el *timeline* (línea de tiempo). Es importante para ello saber calcular cuánto durará una animación y cómo se mide en fotogramas. Una vez este todo configurado y el *rigging* funcione correctamente, se utilizan los controladores e interfaces que se hayan creado para **animar el modelo**, como ya se explicó en ese apartado (3.5.3 *Rigging*), dependiendo de lo que se necesite se va intercambiando entre el sistema FK e IK, si se han realizado ambos, sino el sistema FK es el más sencillo para articular al personaje en cada una de las poses clave de la animación. Además, si no se crean los controladores para agrupar los huesos, el esqueleto por defecto funciona como el sistema FK, si se anima directamente con él.

Se guarda la pose inicial para poder resetear el modelo a esa pose en cualquier momento, también se pueden crear otras poses, nombrarlas y guardarlas para usarlas posteriormente si son poses muy recurrentes o es necesario para el proyecto.

Los animadores, previamente a obtener el modelo o modelos con el *rig*, pueden realizar los **dibujos de las poses clave** de cada animación en las que posarán los modelos. Como los dibujos de las poses clave para la animación de caminar, para la de coger un objeto del suelo, para pegar un puñetazo, etc., todas las que necesiten para completar las acciones del storyboard.

Eso dibujos les servirán de guía para posicionarlas en la línea de tiempo. Una vez hayan **creado las poses clave** y hayan grabado cada una en el fotograma o fotogramas correspondientes en el que debe estar el modelo en esa posición, el ordenador calculará la interpolación entre las poses

y moverá en los fotogramas intermedios los huesos para que cambien de una posición a otra, generando la animación. En cada fotograma para guardar su correspondiente pose clave, se guardan la información de los atributos de los controladores y del modelo. Se pueden guardar solo los cambios de los controladores que hayan modificado algún atributo, incluso solo de ese atributo del controlador, o de todos los controladores, aunque no hayan cambiado. También se puede copiar y pegar en otro *frame* esa configuración de uno o varios atributos para que se repita en otro fotograma. Se pueden editar en cualquier momento los atributos de cualquier modelo en cualquier fotograma. Asimismo, se pueden crear Nurbs para crear railes para que los objetos se muevan por ellas en la forma que se les haya dado y guardar los atributos del objeto en cada fotograma de la misma forma que antes. Por último, en la línea de tiempo se utilizan gráficos con curvas para suavizar las transiciones de una pose a otra.

De la misma forma se pueden animar otros tipos de objetos como partículas, metabolas, cámaras o luces en la escena. Todos los objetos pueden cambiar cualquiera de sus atributos en cualquier *frame*. A modo de ejemplo, así se pueden crear animaciones con la cámara en movimiento, o que la luz de una linterna acompañe al modelo de la linterna para que pueda alumbrar.

3.5.8. Renderizado

Una vez realizadas las texturas, la iluminación, los distintos movimientos de cada elemento de la escena 3D para la animación final, y todo esté acabado y correcto, se procede a renderizar la animación o la escena. Consiste en que a partir de la vista del objeto u objetos cámara de la escena tridimensional el ordenador genere todos los fotogramas resultantes o los que se le indiquen.

Previamente, se tienen que **configurar los parámetros** con los que se generará la escena: si se quiere que realice directamente el vídeo en un **formato** .mp4, o si se prefiere que genere las imágenes de cada *frame* sueltas en distintos .png; la **relación de aspecto** (proporción entre ancho y alto de la pantalla) del vídeo; la **resolución**; y la **carpeta** del ordenador donde se guardará, entre otros. Antiguamente solo se renderizaban en formato imagen y luego se concatenaban todas en un programa editor de vídeo, ahora muchos programas ya lo exportan directamente en un formato de vídeo. También se puede escoger según el programa el motor con el que se realizará el render o renders, para tener más parámetros para configurarlos. Este tipo de programas o complementos para un programa (*plugins*) añaden nuevas funcionalidades, para poder configurar parámetros adicionales, como podrían ser: la **calidad de la iluminación** o

de los materiales, filtros o más **parámetros de la cámara**, a veces tienen sus propios objetos para iluminar la escena o sus propios materiales para colocar a las mallas, etc. Asimismo, se pueden realizar tareas de la postproducción (que se explicará en el siguiente apartado), como la corrección de color, sin necesidad de usar otro programa adicional para la posterior edición del vídeo, se haría la edición final del vídeo directamente en el programa si lo permite.

3.6. Postproducción

La postproducción digital es el conjunto de tareas que se deben llevar a cabo para la edición y corrección final del proyecto. Con el render de la animación realizado en la fase de producción, se realizan las tareas de esta etapa. A continuación, se explican algunas de las tareas que se pueden llevar a cabo en esta fase del proyecto. Aunque no son las únicas que existen ni todos los proyectos tienen que realizarlas todas, solo las necesarias para la finalización del proyecto.

3.6.1. Composición

Muchas veces en animación 3D el renderizado se hace por capas. Normalmente cada capa agrupará elementos que tengan características en común. Además, cada una corresponderá a un plano distinto de la escena, es decir, los elementos que estén más lejos de la cámara serán una, los siguientes más cercanos otra, y así sucesivamente hasta los que estén más cerca de la cámara en la escena. Por ejemplo, en una capa de render estará el escenario estático de cada fotograma, en otra capa los elementos del escenario que tengan movimiento, en otra capa la animación de los personajes secundarios, en otra capa la animación de los personajes principales de la animación, en otra capa los efectos especiales y así. Principalmente se hace así por las siguientes razones: quitarle carga de computación al ordenador, al no tener en cuenta todos los elementos en un solo render; si hay algún error no tener que volver a hacer todos los renders de toda la escena completa, sino solo de lo que se haya modificado; y porque así en la postproducción los elementos son más manejables si se quieren añadir otros elementos en 2D entre los distintos planos.

La **composición** es precisamente el proceso en el que se recopilan toda esta serie de elementos gráficos y se unen todos ellos en una sola imagen o vídeo. Aunque a continuación en los siguientes apartados se habla del audio, *motion graphics* y los VFX, estos también son elementos que se unen y se tienen en cuenta para sincronizarlos junto al resto de elementos gráficos en este proceso para generar el producto final. Este proceso se realiza en programas especializados

en la edición de vídeo. En ocasiones se utilizan en este proceso de la postproducción técnicas como el *matte painting*, CGI, la rotoscopia, etc. Por ejemplo, para crear los entornos de las películas de ciencia ficción, entre otras finalidades. Son técnicas que usan imágenes del mundo real o fotografía y se modifican para crear contenido nuevo, en la industria de la fotografía y publicidad también se usan. Asimismo, en este proceso será cuando se genere el *loop* haciendo coincidir los primeros fotogramas del vídeo con los últimos en la edición de este.

3.6.2. Corrección de color y Etalonaje

La **corrección de color** consiste en una vez se han unido todas las piezas que forman parte de la animación, se modifican los parámetros de la temperatura o la iluminación del vídeo para que se aprecien todos los elementos correctamente. Básicamente, se igualan los colores de los diferentes renders, que como ya se explicó, pueden exportarse en .mp4, es decir cada capa del render es un archivo vídeo, que componen la obra final para crear uniformidad y se vea más atractivo. Así se consigue que la imagen del video sea óptima.

Después de la corrección de color, se hace una gradación de color o **etalonaje**. Es una de las fases más importantes de la postproducción. Consiste en definir un tipo de estética para el proyecto audiovisual, es decir, con este proceso se consigue contar una historia con un estilo determinado y uniforme en toda la pieza. Hace que se creen ambientes propios, se conviertan los días en noches, se destaquen el estado de ánimo de los personajes, etc. Esto puede crear un cambio radical con mucha influencia en la audiencia para la forma de contar esa historia. Ayuda a comunicar cada sentimiento en las escenas, simplemente con el color. A dar una aspecto más frío, o hacer que un escenario transmita alegría, etc. Esto se consigue gracias a las técnicas de composición, a la teoría del color y los conocimientos que se estudian en diseño, cine e ilustración sobre como influencia una imagen en las personas y como se transmiten esos sentimientos.

3.6.3. Motion Graphics y VFX

En este proceso se añaden al vídeo todos los elementos necesarios en 2D. Los títulos, los créditos, VFX para terminar de retocar el vídeo, animaciones de imágenes u otros elementos, etc. Los **motion graphics** es un tipo de animación, pero se basa en las animaciones de estos elementos: imágenes, títulos, subtítulos, logos, se pueden añadir stickers, emoticonos, o los elementos que se necesiten para terminar la composición de la obra de forma creativa. Siempre

deben añadirse teniendo en cuenta la coherencia y si están aportando información valiosa, que son términos que se estudian en diseño. Es importante si se añaden textos tener en cuenta los conocimientos sobre tipografía, color y composición para que sean legibles y claros. Todos estos elementos forman parte de cómo se cuenta la historia o la narrativa del producto audiovisual resultante, y se puede aportar mucha información de esta manera. Por su parte, los **VFX** (efectos visuales) se tienen en cuenta en todo el *pipeline* de la animación. Se utilizan diversas técnicas para crear imágenes o manipular las que ya disponemos. Por último, hay que tener en cuenta que los efectos visuales y los efectos especiales son cosas diferentes. Los efectos especiales (FX) son los que se han realizado durante la producción y no se modifican en la postproducción.

3.6.4. Audio

El sonido es una parte muy importante en diversos medios audiovisuales. Con él se consigue la inmersión del usuario y es una gran ayuda para la narrativa. Se llama **banda sonora** al conjunto de todos los sonidos que acompañan a la parte visual.

En los **medios no lineales** como videojuegos, realidad aumentada, web, etc., no se puede saber cuándo sucederá cada evento sonoro, por tanto, a partir de los sonidos que el diseñador ha construido, se programan los sonidos y estos se generan sobre la marcha según las acciones del usuario. Sin embargo, en los **medios lineales** (cine, vídeo, animación, TV, etc.) siempre se sabe cuándo va a suceder cada evento sonoro, en este caso, la banda sonora es un producto final del trabajo de edición. Aunque dependiendo del proyecto, hay que tener en cuenta que, si es una animación, los animadores necesitarán las voces de los personajes, para poder animar las bocas de los personajes, por tanto, estas deberán estar grabadas antes de animarlos, y posteriormente en la postproducción se unirán y sincronizarán con el resto de las pistas de audio al apartado visual.

Existen principalmente 3 **tipos de sonidos** básicos que al final constituyen la banda sonora:

- **Efectos sonoros (SFX)**. Que pueden ser efectos duros o enlatados; los *foleys* que son generados especialmente para ese proyecto y no son reutilizables en otros; sonidos de ambiente que no provienen de los actores/personajes ni objetos de la escena; y de diseño que son creaciones peculiares.
- **Voces (diálogos)**. Hacen referencia a las voces de los actores y/o personajes de la historia, pero no a los sonidos ambientales que se puedan añadir porque simulan que hay gente de fondo hablando, por ejemplo. Normalmente en películas de imagen real

estas voces se toman como referencia y luego se doblan en estudio. Mientras que en animación y videojuegos se generan directamente en estudio.

- **Música.** Las composiciones originales o prestadas para el proyecto. Puede verse de tres maneras: música en la escena porque forma parte de ella; música de acompañamiento; y canciones que los personajes cantan o suenan.

En el caso de los **loops animados**, en ocasiones no tienen ningún tipo de canal de audio, ya que si el propósito es generar una animación en bucle que sirva para enviarla por redes sociales, no suele llevar sonidos, como los stickers o gifs. Además, el formato .gif no permiten audio. Sin embargo, esto no significa que no existan bucles animados con sonido, lo único que serán exportados en formatos que permitan guardar canales de audio como pueden ser los formatos .mp4 o .avi. En este caso, además de sincronizar los fotogramas para que la animación se reproduzca en bucle de manera infinita sin que se noten cortes en la imagen, esto también deberá hacerse con el audio. Editar la banda sonora para que se sincronice con el apartado visual y suene en bucle sin que se noten saltos en el audio, haciendo que el final y el inicio se complementen sonando de forma coherente y secuencial. Este tipo de *loops* se utilizan por ejemplo en las pantallas de inicio o en los menús de los videojuegos; en cabeceras para vídeos de streaming que estén esperando para empezar y pongan una animación en bucle; y muchas más utilidades.

3.7. Antecedentes

En este apartado se mostrarán algunos proyectos realizados en el pasado que tengan relación con este, para mostrar la experiencia y conocimientos previos de este tema que se aplican al proyecto.

A lo largo del grado en Ingeniería Multimedia, se han cursado ciertas asignaturas en las que se han realizado proyectos relacionados con el diseño, la animación y el sonido. También se han realizado cursos intermedios relacionados con el sector. A continuación, se mostrarán los proyectos en orden cronológico y algunas imágenes con ejemplos.

En el primer curso del grado, en la asignatura de **Fundamentos del Diseño Gráfico** se aprendieron las bases del diseño y los pasos necesarios para realizar un video publicitario junto con el diseño de una web. El trabajo se hizo en equipo junto con José Francisco Valdés Sirvent y María del Rosario Fernández García. Se puso en práctica la correcta realización de un *brainstorming*, un guion, el *storyboard*, la producción y postproducción del vídeo realizado con

la herramienta de PowToon.com. El vídeo publicitario iba sobre un concurso ficticio para participar en la creación de un videojuego colaborativo con la comunidad. También se realizaron: el diseño y la documentación del logo; los *mockups* (las propuestas) del diseño de la web para participar en el concurso; y un ejemplo interactivo de cómo se vería la web.

En verano, después de terminar el primer curso, se realizó un curso sobre la **Introducción al desarrollo de personajes 3D para videojuegos** en la Universidad Miguel Hernández. En este curso se aprendieron los conceptos del diseño de personajes 3D para videojuegos y cómo introducir los modelos en Unity. Con el programa de ZBrush y Photoshop, se hizo el modelo 3D de un personaje femenino en 3D sin cabello (porque no daba tiempo), se le hizo una chaqueta, se texturizó, se agregó un rigging al modelo de una página web de recursos, y se introdujo para poder probarlo en el entorno de Unity (motor de videojuego). Fue la primera vez que usaba un programa de modelado y que realizaba un modelo 3D.



*Figura 55. Captura del primer modelo 3D, un cuerpo femenino realizado en ZBrush
(Fuente propia: realizado con [ZBrush](#))*

En segundo curso del grado, en la asignatura de **Sistemas Multimedia** se aprendió sobre las distintas formas de gestionar un trabajo en grupo. Para el segundo trabajo, se elaboró una animación 3D y la web junto con María del Rosario Fernández García, Francisco Javier García Fernández y Clara Rodríguez Prado. El proyecto iba sobre la inauguración ficticia de un cine novedoso en el que dos o más personas podrían reservar una sala y ver una película a elección sin preocuparse de que otras personas los molestasen porque la sala la reservaba el grupo. Se realizó el logo de la empresa, el modelo del cine en 3D y una animación mostrando cómo era por dentro con el programa SketchUp, además de “la web oficial” del cine.

En la asignatura de **Modelado y Animación por Computador** de ese mismo curso se realizaron 3 prácticas relacionadas con el modelado y la animación. En esta asignatura se vieron los fundamentos básicos sobre esta área y cómo llevarlo a la práctica. La primera práctica se realizó el modelo 3D de una casa como si fuese hecha de piezas LEGO, esta práctica sirvió para aprender el funcionamiento del programa Autodesk 3ds Max y crear los primeros objetos con primitivas básicas y las funciones principales de un programa de modelado. Para la segunda práctica, se tenía que diseñar una cafetería en 3D con el estilo identificativo de la empresa de Apple, en el mismo programa que la práctica anterior. Además, como reto, había que introducir una serie de elementos como los taburetes y los apliques de la pared de forma obligatoria en el diseño. Para la última práctica de la asignatura, se tenía que realizar en ese mismo programa una animación que imitase la intro de la empresa Pixar, pero el texto podía ser distinto y en vez de la bombilla de Pixar, había que hacer un robot humanoide con cabeza de bombilla. Igualmente, al final de la animación obligatoria se podía alargar con una animación diseñada por el alumno. En ese proyecto se cambió la palabra de Pixar por Kely que era el nombre de mi gata y se añadieron a los lados del texto dos elementos decorativos. Se modeló el robot humanoide como indicaba el enunciado, y se aprovechó para que el final de la animación de saltar sobre la letra *E* del nombre de la gata, hubiese un apagón, pero el robot tenía un interruptor para poder volver a encender la luz (la del escenario y la suya propia de su cabeza) todo con motivo de gato. Asimismo, se realizó un trabajo en grupo sobre como programar un script, para el programa que se había estado usando en la asignatura, que añadiese una utilidad. Este trabajo se realizó junto con la colaboración de Francisco Javier García Fernández y María del Rosario Fernández García. Se realizó un script con partículas (se programa una nueva herramienta para el programa que ayude al modelador o animador) con el que se podían generar una lluvia de estrellas, para el programa que se ha mencionado que se usaba en la asignatura. Se podía editar una luna, las nubes del cielo, el número de estrellas, su forma y color y la animación que seguían.



Figura 56. Casa de LEGO hecha en 3ds Max
(Fuente propia: realizado con [Autodesk 3ds Max](#))



Figura 57. Cafetería de Apple hacia el interior
hecho en 3ds Max
(Fuente propia: realizado con [Autodesk 3ds Max](#))



Figura 58. Cafetería de Apple hacia el exterior
hecho en 3ds Max
(Fuente propia: realizado con [Autodesk 3ds Max](#))



Figura 59, Captura de la animación imitando intro Pixar en 3ds Max
(Fuente propia: realizado con [Autodesk 3ds Max](#))

En el verano después de terminar el segundo curso, se llevó a cabo un curso sobre la animación 2D, llamado **Animación en Acción**, en la Universidad Miguel Hernández. Se introdujo a la animación digital en 2D, y se realizó una práctica de animación 2D de un péndulo, ejercicios de *timing*, una animación de correr de un personaje, y otra sobre andar y luego hablar (animación

del diálogo) de otro personaje en Toon Boom Harmony. Esta era el primer contacto que se tuvo con la animación 2D.

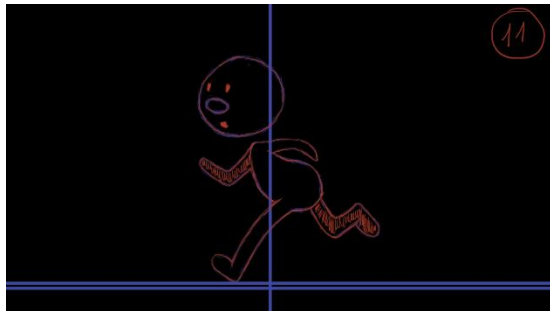


Figura 60. Captura animación correr fotograma 11 de una silueta
(Fuente propia: realizado con [Toon Boom Harmony](#))

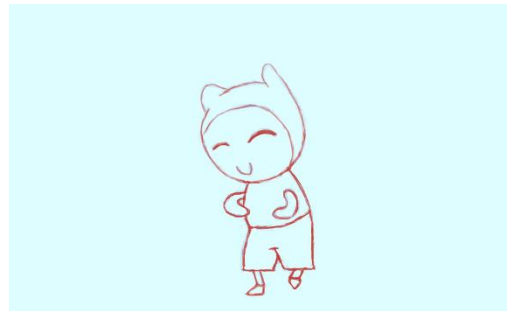


Figura 61. Captura animación andar y diálogo de un personaje propio
(Fuente propia: realizado con [Toon Boom Harmony](#))

En el tercer curso del grado, en la asignatura de **Gráficos por Computador**, había que programar, como si fuese para un videojuego, el movimiento de un coche, modelarlo en 3D con Blender, y también un entorno por el que se pudiese conducir. Esta práctica se basó en la película de Arthur y los Minimoys. Se realizó el modelo del coche como uno de juguete parecido a los que salían en esta serie de películas, y un escenario inspirado en el mundo subterráneo diminuto de los Minimoys.

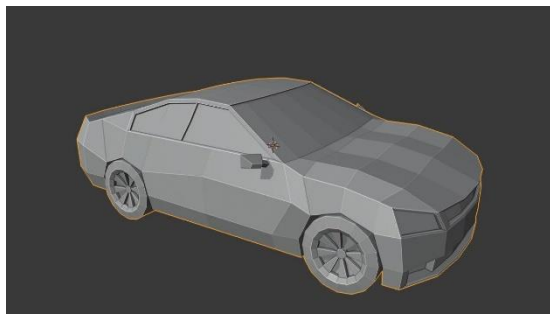


Figura 62. Coche de juguete realizado en Blender
(Fuente propia: realizado en [Blender](#))

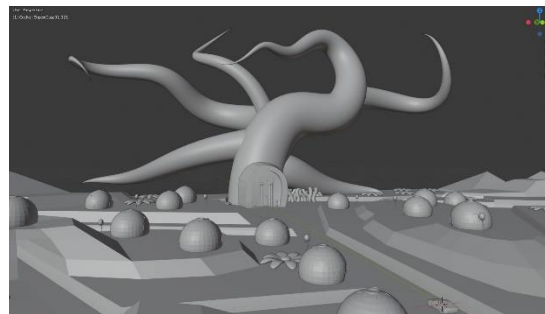


Figura 63. Entorno subterráneo inspirado en la película Arthur y los Minimoys
(Fuente propia: realizado en [Blender](#))

En verano se realizó la asignatura de **Prácticas en Empresa II**. Se realizó en una empresa de marketing digital, y aunque gran parte de las prácticas fue para programación web, hubo una parte de creación de contenidos para las webs, donde se escribieron artículos para diferentes blogs, se retocaron fotografías y se realizó un vídeo publicitario para las redes sociales. Esta parte de las prácticas requirió conocimientos sobre los programas de Adobe: Photoshop, Premiere Pro y After Effects, con ellos se realizaron: los trabajos de fotografía, arreglando las fotos y corrigiendo el color; y se hizo el vídeo publicitario para la portada de Facebook de una clínica dental.

En el cuarto curso del grado, todas las asignaturas de este curso se unificaban para realizar un proyecto en grupo mediante el **ABP**, que consistió en la realización de un videojuego junto con Francisco de Asís Prieto Buendía, Carlos Benito Santiago Portas, José Manuel Manresa Guirao y Jose Antonio Sansano Melero. El videojuego era un *shooter* (de disparos) en primera persona con mecánicas de sigilo estilo *low poly*. En ese proyecto se realizaron los modelos 3D para el videojuego, el diseño de los logos del grupo y del juego, los vídeos del logo del grupo, los créditos, el *making of* y el tráiler, y la programación del juego y del motor gráfico. Mis principales funciones en el proyecto fueron: la programación del motor gráfico en el lenguaje de C++, es decir el software usado para dibujar los gráficos del juego en la pantalla; la organización, recopilación, edición y creación de los sonidos del proyecto con Audacity y FMOD, tanto los del videojuego como los de los vídeos, junto con Paco; y la documentación y maquetación en Word de los distintos documentos del proyecto junto con mis compañeros.

En verano, después de terminar el ABP, se realizó la asignatura de **Prácticas en Empresa I**, en una empresa que crea herramientas software útiles para otras empresas. En esta empresa, se realizaron modelos 3D en Blender y el texturizado en Adobe Substance 3D Painter, de unas naves industriales que se introdujeron en una aplicación realizada en Unity, para que una empresa de Alabama pudiese tener un mejor control sobre sus actividades viéndolas en tiempo real.



Figura 64. Modelo 3D nave industrial
(Fuente propia: modelo realizado en [Blender](#) y texturizado con [Substance 3D Painter](#))

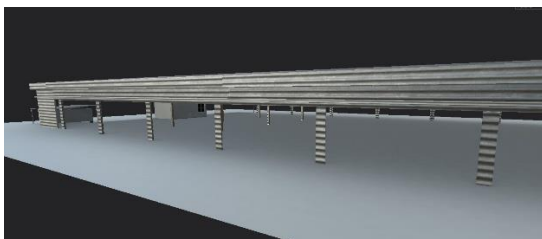


Figura 65. Modelo 3D garajes industriales
(Fuente propia: modelo realizado en [Blender](#) y texturizado con [Substance 3D Painter](#))

Después del cuarto curso del grado, se comenzó un doble curso superior sobre **Diseño y Modelado 3D para Videojuegos, y Animación 3D y Rigging para Videojuegos**. Donde se están estudiado más en profundidad algunos de los conceptos y se están realizando prácticas relacionadas con este área. Se han usado herramientas como Photoshop, Illustrator, Maya, Blender y Substance 3D Painter. Este curso se está realizando, de momento los apartados que se han visto han sido diseño, *concept art* y modelado.



Figura 66. Ejercicio de anatomía calavera en 3/4
(Fuente propia: realizado en [Photoshop](#))



Figura 67. Ejercicio diseño de un escenario
(Fuente propia: realizado en [Photoshop](#))



Figura 68. Trabajo final diseño de un personaje
(Fuente propia: realizado en [Photoshop](#))



Figura 69. Ejercicio modelado de un Pokémon
diseñado por un compañero
(Fuente propia: realizado en [Blender](#))

Además, a lo largo de estos años se han realizado **otros trabajos** para practicar diseño, ilustración, concept art, modelado, animación y cómo funcionan algunos de los programas de la industria por cuenta propia. En estos trabajos se han realizado logos en dibujo vectorial, bocetos de dibujo, prácticas de anatomía y dibujo, retoque de imágenes, y modelado 3D.



Figura 70. Reto modelado piruleta de
golosinas sin aplicar modificadores
(Fuente propia: realizado en [Blender](#) sin
aplicar modificadores)

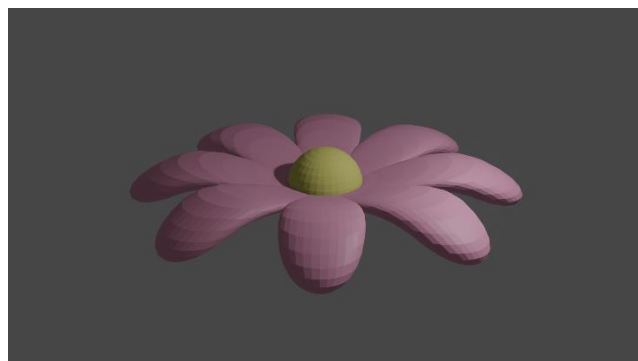


Figura 71. Modelado de una flor
(Fuente propia: realizado en [Blender](#))



Figura 72. Escultura de Ugly Dog de la película UglyDolls: Extraordinariamente feos
(Fuente propia: realizado en [Blender](#))



Figura 73. Práctica ilustración vectorial de mis gatos
(Fuente propia: realizado en [Illustrator](#))

Con todo esto se pretende mostrar el avance gracias a las asignaturas y el trabajo realizado, y se quiere poner todos esos conocimientos en práctica para este proyecto como trabajo final del grado. En este caso realizando una animación en bucle para ver qué se debería tener en cuenta y cómo para que no se note que la animación termina y comienza una y otra vez, creando así un “bucle perfecto” como se les dice en la industria a estos tipos de *loops* que no se nota cuando han vuelto a empezar.

3.8. Principales herramientas software

En la industria de los gráficos por computador y pintura digital existen múltiples tecnologías. Normalmente, se usan distintos programas y herramientas según la tarea que se vaya a realizar. Hay unas tecnologías más específicas para unas tareas que otras como las que su finalidad es una muy concreta, y otras son más versátiles. Muchas veces es necesario conocerlas para saber qué necesitas y el proceso que quieres seguir. Ya que algunas no tienen la posibilidad de exportar los elementos gráficos generados en los formatos necesarios para usarlos correctamente en otro programa para la siguiente tarea; o una te ofrece más ayudas y es más sencilla de usar y aprender que otro programa; entre otros casos. Conocerlas de antemano y haber practicado con ellas, agiliza mucho las tareas, ya que suelen tener atajos de teclado diferente, los nombres de las funciones en cada programa pueden cambiar de una a otra, y puede que no tengan las mismas. Además, también son muy distintos sus precios. Por todo esto en este apartado se analizarán las tecnologías más usadas dentro del sector para el diseño, el modelado y la animación 3D.

Primero se verán en conjunto las principales tecnologías utilizadas principalmente para la fotografía, el diseño, *concept art* e ilustración. Hoy en día, todas estas tecnologías suelen ser compatibles con el control para trabajar una tableta o pantalla gráfica para una mayor precisión y manipulación. Por eso, aunque solo se detalle en alguna por un caso especial, las demás también se pueden configurar el control del lápiz para estos tipos de dispositivos de entrada.

- **Adobe Photoshop.** Es el editor de imágenes líder en la categoría de fotografía y gráficos. Es tan popular que el formato de archivo original de este programa, .psd o .psb (para archivos grandes), se considera como un estándar en la industria. A veces las empresas incluso pueden pedir ese archivo, y aunque hay otras aplicaciones que exportan su trabajo a este formato, no todos los filtros y capas son igual de compatibles que si se hace directamente desde el programa original del formato. Existen alternativas como exportar en archivo .tiff o .pdf o similar que luego se importe a este programa y así no tener problemas de compatibilidad. Pero lo más recomendable si piden ese formato es porque esperan que trabajes con él. Además, ha supuesto durante mucho tiempo un referente para otros programas, puesto que los usuarios de este si necesitaban cambiar la curva de aprendizaje era más leve si las funciones del nuevo programa se parecían lo máximo posible a las de este. Asimismo, tiene herramientas para realizar animaciones 2D y retoques de gráficos en 3D. Normalmente, el problema de este software es el precio, igual que el resto de las herramientas de Adobe. Aunque ofrecen diversos planes y paquetes de aplicaciones para empresas, estudiantes y profesores, y centros educativos y universidades entre los que elegir, además de una prueba gratuita de 7 días. Disponible para macOS y Windows.
- **GIMP.** Es la alternativa de distribución gratuita a Photoshop más conocida, además puede exportar en formato .psd. Es un software para retoque fotográfico, composición de la imagen y creación de imágenes. Aunque no tiene tantas funciones y herramientas disponibles para la pintura digital. La interfaz es bastante intuitiva. Aunque puede ser diferente a la de Photoshop, existe GIMPshop que es la misma aplicación, pero con apariencia mucho más similar a Photoshop. Está disponible para Windows, Linux y Mac, además de su versión portable.
- **Krita.** Es una aplicación completa para dibujo y pintura digital creativa. Es de código abierto, gratis y está disponible en todas las plataformas de escritorio. Su interfaz es visualmente parecida a la de Photoshop. Hay algunas funciones no compatibles al pasar un archivo de este programa a Photoshop por eso se debe tener en cuenta al exportar a un .psd. Pero es una muy buena alternativa gratis al programa de Adobe para la pintura

digital. Incluye herramientas aptas para la realización de animaciones 2D. Está disponible para dispositivos Windows (con versión de instalador o portable), macOS y Linux.

- **Clip Studio Paint.** Conocida anteriormente por Manga Studio o ComicStudio. Es una aplicación utilizada para la creación digital de cómics y manga. Sus herramientas están muy orientadas a este estilo de dibujo y ofrecen una variedad de herramientas y funciones para ese tipo de trabajos. Tiene disponibles herramientas para realizar animación 2D. Asimismo, permite exportar archivos .psd. Es una aplicación de pago pero que ofrece la posibilidad de una compra única con licencia perpetua, además de una versión de prueba gratuita de entre 3 y 6 meses según el sistema. Para iPhone tiene una versión gratuita. Disponible para sistemas Windows, macOS, iPad, iPhone, Galaxy, Android y Chromebook.
- **OpenCanvas.** Es un programa de edición de imágenes y software de pintura apto para expertos y principiantes. Este programa incluye una función que permite grabar el procedimiento del dibujo, últimamente esto es muy utilizado por los artistas para mostrar su flujo de trabajo con un *speedpaint* en las redes sociales. Admite el uso de tabletas de dibujo externas como dispositivo de entrada. Es muy popular en Japón, y algunas marcas de tabletas gráficas regalan una licencia o una prueba de este programa con la compra de alguno de sus productos. Tiene disponible una prueba gratuita y la posibilidad de compra única (licencia perpetua) o compra con actualización. Solo disponible para sistemas Windows.
- **Corel Painter.** Es una aplicación de arte digital creada para simular la apariencia y comportamiento del dibujo y pintura tradicionales. Es compatible para exportar archivos .psd. Posibilidad de compra única (licencia perpetua) o compra por actualización. Disponible para plataformas Windows y Mac.
- **ArtRage.** Es una alternativa muy económica a Corel Painter, con una interfaz sencilla y elegante. Un programa de pintura orientado al arte tradicional, capaz de abrir y guardar archivos de Photoshop. Admite el uso de tabletas de dibujo externas como dispositivo de entrada. Por la compra de alguno de sus productos, algunas marcas de tabletas gráficas regalan una licencia o una prueba de este programa. Disponible compra única (licencia perpetua). Para dispositivos Windows, Mac, Android, iPhone y iPad.
- **Adobe Illustrator.** Es un editor de gráficos vectoriales, apto para la ilustración y dibujo vectorial. Incluye funciones para maquetación de documentos. Completamente compatible con el formato Photoshop, se puede importar un archivo .psd con todas sus

capas, de la misma forma que se puede importar en Photoshop un archivo .ai (formato de archivo de Illustrator). El problema es el mismo que el resto de las herramientas de Adobe, el precio. Aunque ofrecen diversos planes y paquetes de aplicaciones para empresas, estudiantes y profesores, y centros educativos y universidades entre los que elegir. Disponible para macOS y Windows.

- **Inkscape.** Es una alternativa a Illustrator gratuita y de código abierto. Un programa de gráficos vectoriales. Si estás acostumbrado al programa de Adobe la curva de aprendizaje, aunque más leve porque las herramientas y conceptos son casi los mismos, puede ser un poco costosa. Disponible para Windows y macOS.

Aunque se debe tener en cuenta que muchas de las grandes empresas de animación cuentan con sus propias tecnologías, se continuará con el análisis de las disponibles en el mercado para todo tipo de usuarios. Para modelado y animación 3D las tecnologías cambian según el objeto que se vaya a modelar, si es sólido u orgánico, y depende de las técnicas de modelado hay programas mejor recomendados que otros. Para la escultura digital los programas son compatibles con tabletas y pantallas gráficas para una mayor precisión en los trabajos. Para retopología y *rigging*, generalmente existen *add-ons* (complementos) compatibles con los programas para facilitar el proceso. Aunque hay programas que ya tienen funciones que lo hacen más fácil. Para texturizado existen programas enfocados solo a este sector. Pero siempre se pueden exportar las texturas, y si el estilo final que se quiera dar al modelo lo permite, utilizar programas de pintura digital como los que se mencionaron anteriormente. A continuación, se analizarán algunas de las tecnologías utilizadas para modelado, retopología, rigging, texturizado y animación 3D.

- **Autodesk Maya.** Este software es una poderosa solución que integra herramientas para modelado, efectos especiales, renderizado y animación 3D para videojuegos, TV, películas y más. Se pueden lograr acabados muy realistas y profesionales. Aunque tiene herramientas para el modelado orgánico con la escultura digital, se especializa en modelos sólidos realizado mediante *box modelling*, primitivas básicas, paramétrico, con Nurbs y mucho más. Es reconocido como uno de los mejores software de la industria. Tiene muchas ventajas a la hora de realizar la tarea de retopología, sus herramientas brindan al artista de grandes facilidades. Además, es compatible con otras aplicaciones de la compañía Autodesk, que facilitan el flujo de trabajo. Entre ellos la aplicación Arnold, que es un motor de renderizado que, mediante el trazado de rayos y muchas opciones disponibles, eleva la calidad del resultado final. El problema, como el del resto de aplicaciones de Autodesk, es su elevado coste. Aunque ofrece diversos planes y

paquetes, además de la posibilidad de una licencia educativa gratuita para estudiantes o profesores. Disponible para Windows, Linux y Mac.

- **Autodesk 3ds Max.** Otra aplicación de la misma compañía, pero orientada a la realización de modelos, FX, animaciones y renders 3D para videojuegos y visualización de diseños de arquitectura. Este programa es mejor para el modelado sólido. Es compatible con otras aplicaciones de la compañía Autodesk, que facilitan el flujo de trabajo. El problema sigue siendo el elevado coste. Ofrece diversos planes y paquetes, además de la posibilidad de una licencia educativa gratuita para estudiantes o profesores. Disponible para Windows.
- **AutoCAD.** Es otra aplicación de la compañía Autodesk. Orientada para la realización de modelos de piezas industriales. Los ingenieros y arquitectos profesionales de la construcción lo utilizan para dibujar y anotar geometría en 2D y 3D sólidos, comparar planos, sustitución de bloques, recuento de objetos, crear tablas de planificación y mucho más. Ofrece diversos planes y paquetes, además de la posibilidad de una licencia educativa gratuita para estudiantes o profesores. Disponible para Windows y Mac.
- **SketchUp.** Es un programa de modelado 3D orientado a la creación de modelos de arquitectura, diseño de interiores, modelos de ingeniería civil y mecánica. Viene con una galería de modelos 3D que permite buscar y compartir modelos fácilmente. Es muy fácil de manejar. Tiene disponible una versión gratuita para web, y otras versiones de pago con funcionalidades adicionales. Disponible para Windows y macOS.
- **Blender.** Es el programa de código abierto y gratuito de creación 3D más reconocido. Tiene soporte para todo el proceso de producción de un proyecto 3D, todo el *pipeline* de animación 3D: modelado, retopología, *rigging*, texturizado, simulación de partículas (FX), edición de vídeo, composición, e incluso herramientas para animación 2D y de rotoscopia. Tiene herramientas para todo tipo de modelado: por escultura digital, modelado paramétrico, box modelling, con Nurbs, primitivas y mucho más. Es fácil de usar y hay miles de contenido en línea gratis que pueden ayudar a su uso y aprendizaje. Al ser tan extenso, no se especializa mucho en ninguna tarea. Disponible para plataformas Windows, macOS, Linux y cualquiera con soporte para gráficos OpenGL.
- **ZBrush.** Es un programa de escultura digital y pintura 3D que ha revolucionado la industria con sus potentes funciones y flujos de trabajo. Con una interfaz elegante y una gran cantidad de herramientas. Al principio puede resultar un poco abrumador el aprendizaje, pero se obtienen modelos de alta calidad. Es capaz de manejar modelos de altísimas cantidades de polígonos. Dispone de diversas opciones de pago e incluso una

licencia perpetua de compra única del programa. Apto para las plataformas Windows y Mac.

- **Cinema 4D.** Es un software de animación, modelado, simulación de partículas y renderizado 3D de la misma compañía que ZBrush. De gama alta, destacado por su interfaz flexible y facilidad de uso. Cuanta con diversas opciones de pago por suscripción y con una compra única (licencia perpetua). Disponible para plataformas Windows, macOS y Linux.
- **Houdini.** Es un software que cubre todas las áreas principales de la producción 3D, como el modelado, animación, efectos especiales, dinámica de fluidos, iluminación y render 3D. Se especializa sobre todo en la animación y manejo de sistemas de partículas y animación de fluidos para FX. Se obtiene mediante una compra única con licencia perpetua, pero dispone de una versión gratuita para proyectos no comerciales. Disponible para los sistemas Windows, Mac y Linux.
- **Rhinoceros 3D.** Es una aplicación con la que se puede crear, editar, analizar, documentar, renderizar, animar y traducir curvas y superficies. Basada en el modelado paramétrico con Nurbs. Usada para arquitectura, creación rápida de prototipos, diseño industrial, naval y de joyas, y otros parecidos. Aunque también se usa para industrias de diseño gráfico y multimedia. Uso disponible bajo la adquisición de una licencia perpetua de pago único. Apto para Windows y Mac.
- **Marvelous Designer.** Es un programa especializado en la creación de modelos 3D de telas y prendas de ropa para personajes y escenarios. Mejora la calidad de este tipo de elementos y ahorra tiempo. Replica muy bien las propiedades físicas de los tejidos hasta el último accesorio, botón o pliegue. Muy utilizado tanto en industrias de cine, animación y videojuegos, como en publicidad y la industria de la moda. Dispone de diversos planes de pago para empresas, personal independiente o academias, además de una versión de prueba de 30 días. Disponible para Windows y Mac.
- **Autodesk Mudbox.** Es un programa de pintura 3D de texturas y de modelado mediante escultura digital. Ofrece la posibilidad de crear modelos orgánicos y sólidos altamente detallados. Tiene un sistema de capas que posibilita la pintura digital sobre el modelo como en los típicos programas de dibujo digital 2D. No permite el texturizado por materiales ni paramétrico ni sistemas PBR. Dispone de distintos planes de pago y una prueba gratuita. Apto para plataformas Windows y Mac.
- **Mari.** Es una herramienta creativa de pintura de texturas. Diseñado para manejar proyectos complejos y altamente detallados. Tiene herramientas y funciones

especializadas para el texturizado de modelos 3D. La interfaz es receptiva y fluida. Se puede pintar directamente encima de los modelos y ver los resultados de inmediato, esto hace que sea mejor opción que aplicaciones de pintura digital 2D. Tiene diversos planes de pago entre los que se encuentra la posibilidad de una licencia permanente de pago único. Además de una prueba gratuita de 30 días y una versión gratuita no comercial. Disponible para Windows y Linux.

- **Adobe Substance Painter.** Es una herramienta que utiliza el sistema PBR para texturizar modelos 3D. Es intuitivo y tiene una complejidad media de aprendizaje. Tiene múltiples herramientas y funcionalidades para lograr acabados de alta calidad. Posee la capacidad de pintar los modelos con pinceles avanzados y materiales inteligentes que se adaptan automáticamente al modelo. Además, al texturizar con materiales que simulan las propiedades físicas de los que están basados en la vida real, se puede no tener en cuenta los distintos trucos, ilusiones ópticas y engaños que se realizaban en otro tipo de programas. Estos engaños eran necesarios para que parecieran que dependiendo de la iluminación del entorno donde se pusiera el modelo, hiciesen el efecto deseado del material. Pero esto ya no es necesario con programas que trabajan sistemas PBR como este. Presenta múltiples opciones de planes y paquetes de compra como el resto de las aplicaciones de la compañía. Disponible para sistemas Windows, macOS y Linux.

Finalmente, para las tareas de la postproducción, se analizarán algunas tecnologías para edición de vídeo, audio, motion graphics y FX.

- **Adobe Premiere Pro.** Es un editor de vídeo muy potente. Contiene características y herramientas para la administración de proyectos audiovisuales. Permite la edición, composición, corrección y gradación de color de la parte visual, a la vez que ofrece la posibilidad de editar, añadir efectos y otras funciones a la parte auditiva. Una de las más usadas en el sector. Su problema es el coste de las aplicaciones de Adobe. Pero ofrece diversos planes y paquetes de pago. Disponible para plataformas Windows y macOS.
- **Adobe After Effects.** Posiblemente uno de los mejores editores de efectos de vídeo en la industria. Brinda a los usuarios un conjunto sólido de herramientas para que profesionales y principiantes hagan lo necesario para crear los efectos visuales deseados para vídeos y películas. Como las demás aplicaciones de Adobe cuenta con los distintos planes y paquetes de pago. Disponible para Windows y macOS.
- **DaVinci Resolve.** Suite completa para la postproducción digital. Con herramientas para la corrección de color, composición, efectos visuales, edición del audio, etc. Permite la colaboración multiusuario en vivo en el mismo proyecto. Utilizada en diversos sectores

como publicidad, TV, cine y animación entre otros. Es una versión gratuita, pero además tiene otra versión de pago único de una licencia perpetua con más funcionalidades. Disponible para Windows, Linux y Mac

- **Wondershare Filmora.** Es un software de edición de vídeo doméstico. Muy fácil de usar con funcionalidades especiales que permiten combinar clips de vídeo, audio, texto y aplicar efectos visuales. Tiene una versión gratuita bajo una licencia EULA y una opción de pago con distintos planes a elegir, entre ellos un pago único de una licencia perpetua. Disponible para Windows y Mac.
- **Adobe Express.** Es una herramienta de Adobe que puede usarse desde el navegador o desde la app para dispositivos móviles. Se conecta con el resto de las herramientas de la compañía. Ahora mismo está en su fase beta, pero es perfectamente funcional. En esta herramienta se pueden usar las distintas plantillas que ofrece o comenzar con tu propio contenido, y diseñar los contenidos para las redes sociales como las cabeceras, las portadas, logos, historias, imágenes, vídeos, etc., con los formatos correctos. El contenido se puede editar, añadir efectos, textos, gráficos y otros elementos. Se guarda el proyecto en la nube y se descarga cuando sea necesario. Se debe crear una cuenta de Adobe ID para usarlo. Tiene diferentes planes de pago y un plan gratuito para siempre con las funciones básicas. Disponible en web, y para dispositivos móviles iOS y Android.
- **Adobe Audition.** Es una aplicación que permite editar, mezclar, grabar y restaurar audio. Incluye herramientas como multipista, forma de onda y visualización espectral del audio. Se usa en industrias tanto de cine, TV, animación y publicidad como de producción musical. Dispone de diversos planes y paquetes de pago, y la posibilidad de una prueba gratuita de 7 días. Disponible para Windows y macOS.
- **Audacity.** Es un programa gratuito de código abierto para grabar y editar sonidos. También se puede usar para grabar audio en vivo, convertir grabaciones, editar archivos de sonido, cambiar el tono, la velocidad y otras propiedades del audio, y mucho más. Existen diversos tutoriales y ayudas gratis para aprender a manejar el programa. Permite la visualización de la forma de onda y del espectro del audio, y permite la edición multipista. Disponible para Windows, Linux y Mac.

En el capítulo de *Herramientas Utilizadas* de este documento, se incluirán solo las que se han empleado a lo largo del trabajo.

4. Objetivos

Uno de los dos objetivos principales, y más importante para este proyecto será aprender a realizar un *loop* animado en 3D estilo *cartoon* con todas las fases que sean necesarias para llevarlo a cabo, y dejarlas reflejadas en este mismo documento. Asimismo, el otro objetivo principal será que el tema de la animación tendrá que reflejar una concienciación del tiempo que puede pasar una persona en redes sociales sin darse cuenta. Esto se logrará gracias a todos los conocimientos adquiridos a lo largo de la carrera en Ingeniería Multimedia en la Universidad de Alicante.

Para ello, se plantean los siguientes subobjetivos que ayudarán a lograr alcanzar los objetivos principales:

- Investigar, estudiar y desarrollar los procesos a seguir para realizar una animación 3D en bucle, ampliando así los conocimientos obtenidos en el grado.
- Usar una metodología apta para el desarrollo del trabajo.
- Analizar las herramientas software que deben usarse para llevar a cabo el proyecto sin tener que depender demasiado de licencias, intentando usar software libre y de código abierto. Para que sea lo más asequible posible, es decir, el proyecto final se consiga realizar con un presupuesto de bajo coste.
- Buscar ejemplos de experiencias similares que abarquen el mensaje de lo que se quiere transmitir, para poder estudiar las propuestas existentes en el mercado y ver qué funciona y qué no.
- Aumentar la capacidad de resolución de problemas ante los impedimentos que puedan surgir en el proceso.
- Realizar los pasos necesarios para crear la idea del *loop* animado en la etapa de la preproducción.
- Efectuar todos los pasos que sean necesarios para generar la idea concebida en la producción. Aprendiendo a realizar las tareas que no se habían practicado hasta ahora en el proceso.
- Realizar los ajustes y ediciones necesarias para completar la animación del *loop* en la fase de la postproducción del trabajo.
- Crear un contenido atractivo visualmente para todo tipo de público al que le gusten los gatos, ya que va a intentar concienciar sobre el uso de las redes sociales.

- Documentar la investigación y los procesos más importantes seguidos para dejar constancia del funcionamiento, materiales empleados, los problemas, y las soluciones elegidas entre las posibles para que se pueda tener en cuenta para proyectos similares en el futuro.

5. Metodología

En este capítulo, se va a explicar la metodología que se pretende seguir para la elaboración de todo el proyecto, su organización y flujo de trabajo, así como las tecnologías que se usarán a lo largo de este.

Después de estudiarlo detenidamente, para este proyecto la metodología escogida se basará en Scrumban, que está inspirada en los procesos de Scrum y Kanban. Ya que permite usar ciclos de sprint, agregar tareas individuales a la planificación, y también mantener un seguimiento del progreso visual y de una forma clara.

Por una parte, Scrum consiste en hacer sprints cortos, entre 1 o 2 semanas, aunque esto puede depender del proyecto, para crear ciclos de trabajo. Normalmente, se usa para proyectos colaborativos, donde se elige o hay un Scrum Master, el gerente del proyecto, que se encargará de organizar las reuniones diarias, los sprints y los análisis del trabajo después de cada sprint, que garantizan que las tareas puedan llevarse a cabo a tiempo. Esta metodología es algo más estricta que Kanban. A continuación, en el esquema se puede ver cómo se trabajaría con Scrum.

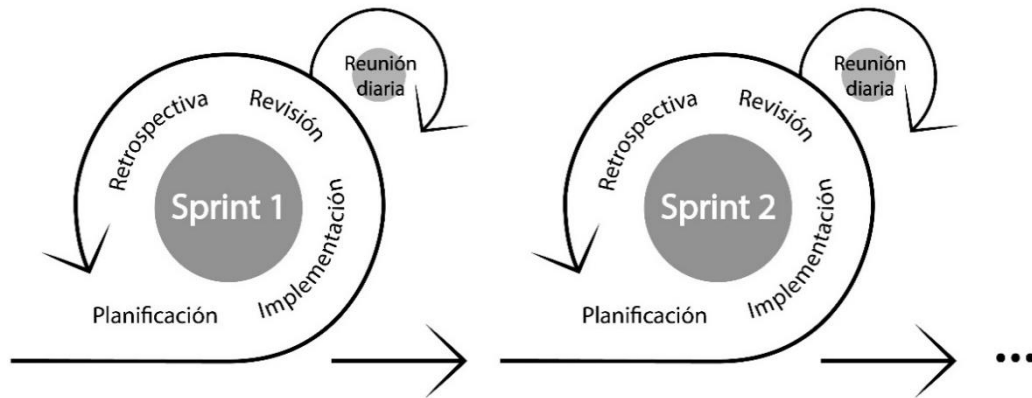


Figura 74. Esquema metodología Scrum
(Fuente propia: realizado con [Inkscape](#))

Por otra parte, Kanban permite representar en un tablero el flujo de trabajo y su progreso. Las tareas se escriben en tarjetas y van cambiando de columna dependiendo el estado en el que se encuentren, también se pueden indicar fechas importantes para esa tarea, pasos, quien es el encargado, y otros detalles. Esto ayuda a que sean más visuales: las tareas pendientes, las finalizadas, las que están en progreso, etcétera. Las columnas pueden ser de cualquier cosa que ayude a categorizarlas, dependerá de cada proyecto cual será la forma de organizarse. Así poder saber si se están creando cuellos de botella, si las tareas con mayor prioridad avanzan o, en

definitiva, si se están logrando o no los objetivos. Es más flexible que Scrum, ya que esta no tiene por qué tener fechas de inicio y finalización como sí tiene Scrum en los sprints, no hay reuniones diarias y es algo más adaptativa. Igualmente, esta metodología también se usa en proyectos colaborativos y es muy buena para equipos que trabajen en remoto, ya que todos pueden ver el tablero y saber cómo está avanzando el proyecto en cualquier momento. Igual que para el método anterior, se mostrará un esquema de cómo se vería un tablero de Kanban.

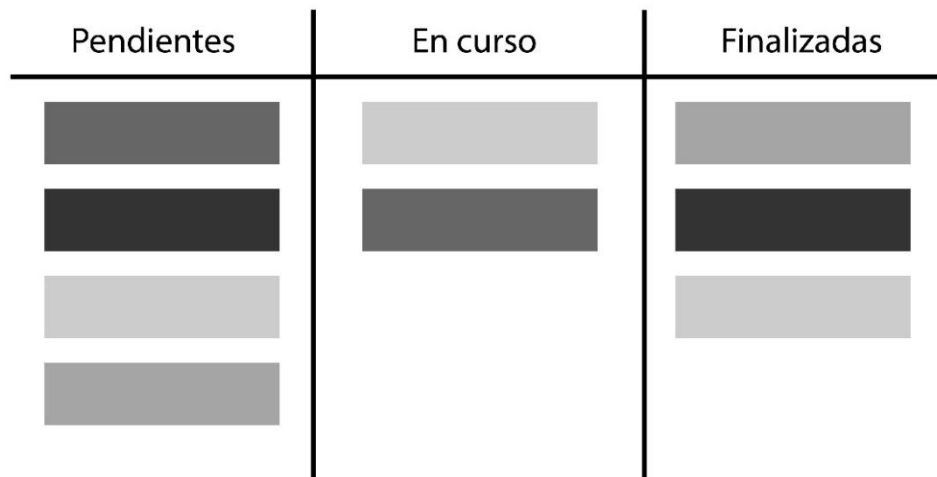


Figura 75. Esquema metodología Kanban
(Fuente propia: realizado con [Inkscape](#))

Por tanto, Scrumban logra dividir el proyecto en tareas más pequeñas, usando ciclos de sprint, y a la vez, se pueden agregar tareas individuales y hace que la planificación mantenga una estructura visual simple y clara.

La opción por la que se ha elegido esta metodología es porque aparte de que teóricamente sean las más indicadas para proyectos de software e ingeniería, también ayudará mucho a organizar un trabajo en el que el tiempo es clave y no se conoce desde el inicio todas las tareas que deberán de realizarse ni cuánto tiempo real necesitará cada una, ya que el propio trabajo ha incluido un trabajo de investigación. Esto podía hacer que llegados a un punto apareciesen tareas nuevas que no se habían tenido en cuenta y otras que a lo mejor se pensaba que eran necesarias, pero realmente no lo fuesen tanto, así que durante el proceso se aprende cuáles son las tareas clave que deben priorizarse y si alguna crea dependencias. Así también este sistema permite añadir más si es necesario y reorganizar el trabajo. Pero a la vez se puede controlar que no se use demasiado tiempo en una, cuando esa no debería requerir tanto tiempo, y visualmente también se podrá ver cómo avanza.

Entonces, para organizar el flujo de trabajo basándolo en la metodología mencionada anteriormente, se ha usado la herramienta online **Trello**, para crear el tablero Kanban y la herramienta online **Toggl Track**, para hacer un seguimiento del tiempo empleado en cada tarea. Se ha determinado que los sprints serán de 1 semana, al inicio de cada uno se realizará la planificación de este y se considerarán las que deberían realizarse esa semana. Al inicio y al final de cada sprint se realizará una captura de pantalla para poder documentar el progreso y saber si se han cumplido los objetivos establecidos. Además, habrá una serie de reuniones con el tutor que no se harán semanalmente, pero servirán para que haya una retroalimentación del progreso y un seguimiento. Para la organización, en Trello, se ha creado un tablero para el Trabajo de Fin de Grado, con el nombre de este. También, se han creado las tarjetas, con sus respectivas fechas de inicio y fin, los *checklist* (lista de comprobación del progreso de la tarea) de los pasos que hay que seguir para realizarla, la descripción, con el miembro que realizará la tarea, que en este caso siempre será el mismo, y los archivos adjuntos, si los necesita. Igualmente, se han categorizado las tareas mediante el sistema de etiquetas que son las mismas que se usarán en Toggl junto con el mismo nombre de la tarea para saber a cuál hace referencia a ese tiempo.

Por tanto, tanto en Toggl como en Trello el nombre de proyecto es *Loop animado 3D estilo cartoon*. Las etiquetas utilizadas son:

- **Animación 2D.** Para, en un principio, todas las tareas que tengan que ver con la animación de las redes sociales que seguramente tenga que estar en 2D. Como no se sabe todavía la composición de la escena no es posible saber si serán necesarias esas imágenes.
- **Animación 3D.** Para todas las que tengan que ver con la animación 3D, incluidas las cámaras si las hubiere.
- **Diseño 2D.** Para las relacionadas con el diseño inicial del *loop*, como el Moodboard, storyboard, etc.
- **Documentación.** Para las que deben realizarse para la correcta entrega del trabajo como el vídeo, el resumen, rellenar y firmar el documento de responsabilidad y autoría, etc.
- **Iluminación.** Las que tiene que ver con la iluminación de la escena del *loop*.
- **Ilustración.** Para las ilustraciones finales que se verán en la pantalla, si son necesarias, ya que no se sabe aún la composición de la escena, no es posible saber si se verá la pantalla con las redes sociales.
- **Información.** Esta categoría especial se usó para la tarea de informarse de cómo se debe afrontar el TFG, realizando el curso que estaba disponible en Moodle.

- **Investigación.** Para las de investigación de los pasos para realizar un *loop*, la historia del modelado y animación, el proceso de realizar un TFG o conocer las herramientas software necesarias.
- **Materiales y Texturas.** Para las que tengan que ver con aplicar a los modelos 3D sus texturas y/o materiales.
- **Memoria.** Para todas las que se refieran a escribir un punto de este documento que es la memoria del proyecto.
- **Modelado 3D.** Las que sean de modelar los personajes, elementos y el entorno.
- **Planificación.** Las que tienen que ver con la planificación y organización del flujo de trabajo, como organizar los sprints semanales.
- **Render.** Para las de preparar y realizar los renders necesarios, y la edición final del *loop*.
- **Reunión tutor.** Las citas con el tutor para ver el progreso del proyecto y obtener retroalimentación.
- **Rigging.** Para las relacionadas con la preparación de los modelos que tengan que ser animados.
- **Trámites.** Para tener presente las tareas y fechas de: que el tutor revise el trabajo, solicitar la defensa, entregar el trabajo y el día de la defensa.
- **VFX.** Las relacionadas con añadir los efectos visuales que sean necesarios.

Asimismo, en Trello se han organizado las siguientes columnas:

- **Tareas nuevas (No contempladas).** Para añadir, en el futuro, las que no se hubiesen considerado desde el inicio.
- **Lista de Tareas.** Donde se han añadido todas las tarjetas de las tareas iniciales que en un principio se han contemplado como las necesarias para realizar el proyecto, y se han rellenado los datos necesarios de cada una.
- **Sprint Semanal.** Donde pasarán a estar todas las que se escojan en la planificación que se hará cada vez que empiece un sprint, para ser finalizadas en esa misma semana.
- **En desarrollo.** Donde se colocarán las tarjetas de las que estén en curso en ese momento.
- **Terminado Sprint Semanal.** Donde pasarán a estar las tarjetas de las tareas de esa semana que ya se hayan concluido.
- **Terminado Proyecto.** Donde a final de cada sprint pasarán todas las tareas que se hayan conseguido finalizar durante el sprint, o, lo que es lo mismo, todas las que se vayan terminando del proyecto.

- **Impedimentos.** En esta columna se añadirán tarjetas de los problemas que vayan surgiendo durante el proceso.

A continuación, se mostrará la organización de Trello inicial y como en la captura no era posible mostrar todas las tareas, se hicieron distintas capturas que se recogen en la segunda imagen cada columna siendo la continuación de la anterior.

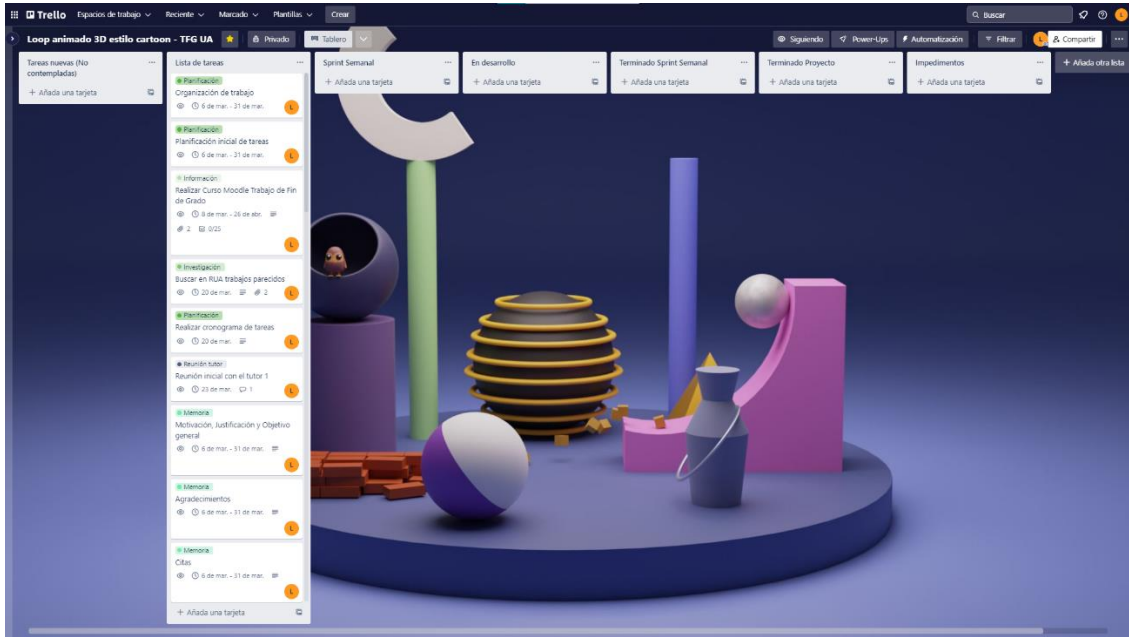


Figura 76. Organización inicial del flujo de trabajo de este proyecto en Trello
(Fuente propia: realizado en [Trello](#))

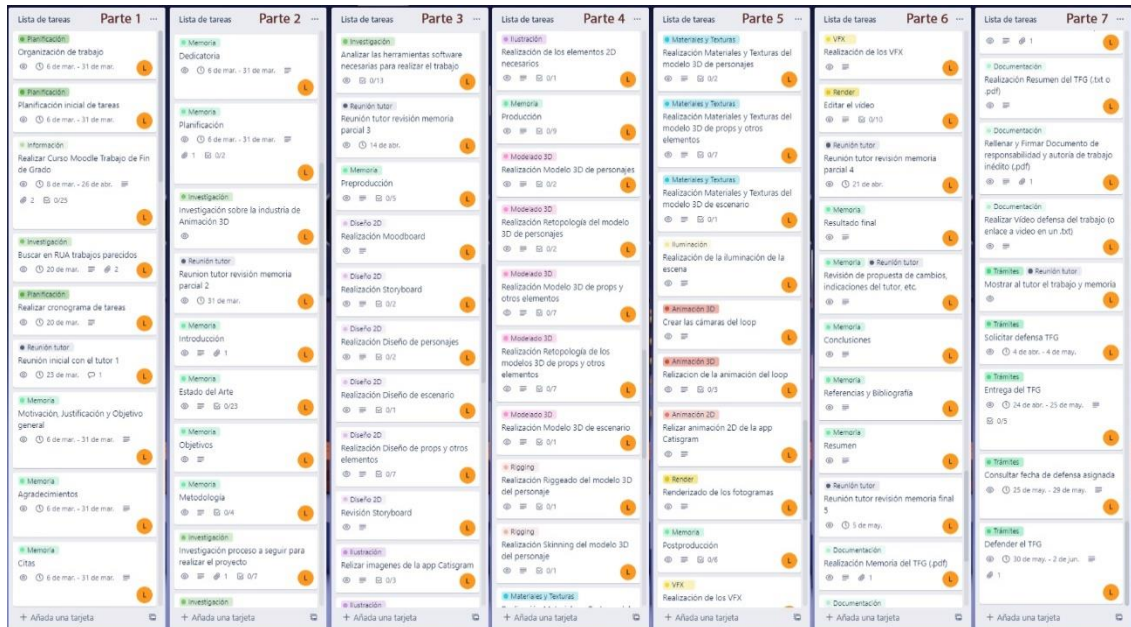


Figura 77. Recopilación de toda la lista de tareas inicial en Trello por partes
(Fuente propia: realizado con [GIMP](#))

Se ha añadido un **Apéndice I: Planificación semanal en Trello** al final del documento que muestra las capturas de pantalla del progreso del trabajo por semanas en Trello de este proyecto.

Por último, se muestran dos capturas de ejemplo, una de cómo Toggl muestra el tiempo de trabajo cada día, y otra del tiempo empleado en cada tarea.

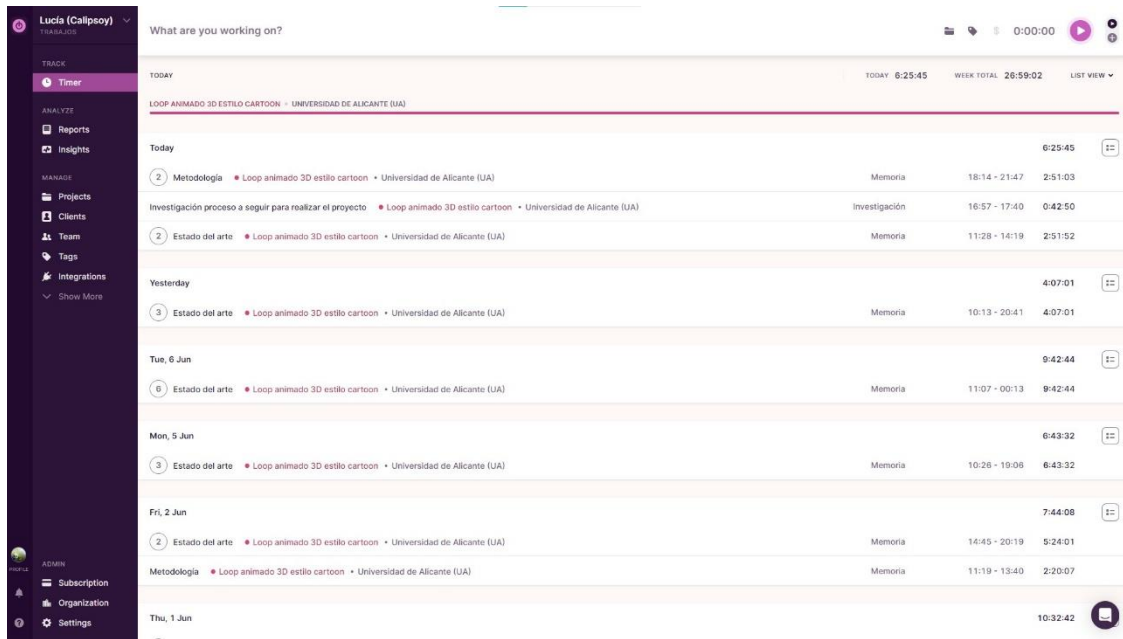


Figura 78. Ejemplo del tiempo empleado cada día y de la semana en Toggl (Fuente propia: realizado en [Toggl](#))

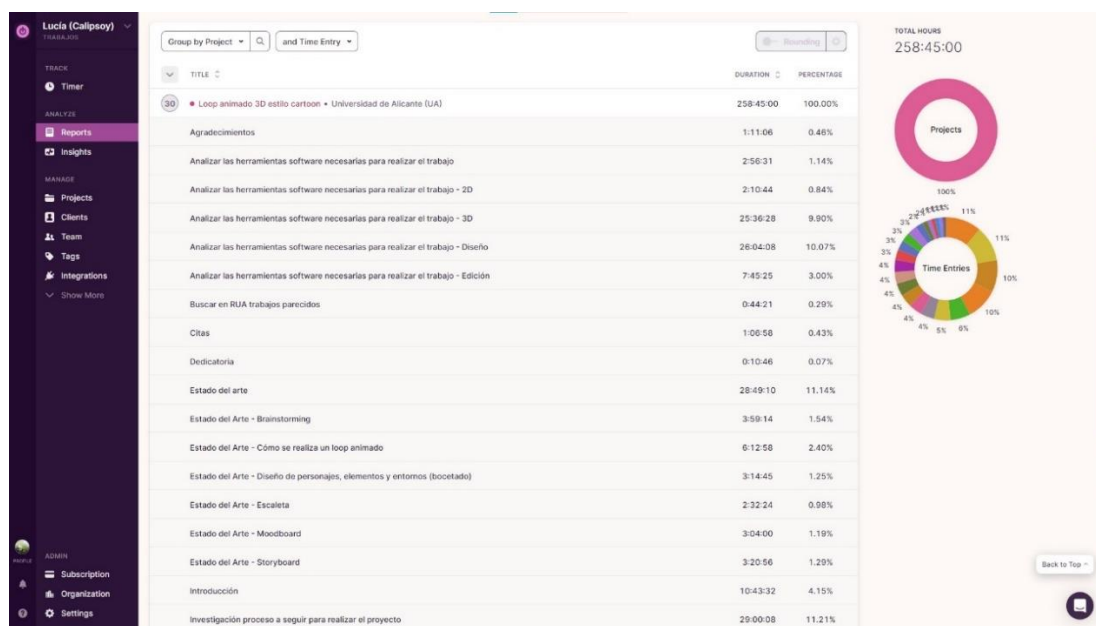


Figura 79. Ejemplo muestra de las horas totales empleadas en cada tarea en Toggl (Fuente propia: realizado en [Toggl](#))

En el capítulo de **Resultado** de este documento se incluirán las capturas de las horas empleadas en cada tarea y las totales en el proyecto.

6. Herramientas utilizadas

En este capítulo se listarán todas las herramientas y recursos utilizados a lo largo del proyecto para su desarrollo.

- **Word.** Para la redacción y maquetación de este documento se ha usado con la licencia para estudiantes otorgada por la universidad. Creado por Microsoft.
- **Excel.** Para la planificación del tiempo en el cronograma se ha usado con la licencia para estudiantes otorgada por la universidad. Creado por Microsoft.
- **Inkscape.** Para dibujar los esquemas en dibujo vectorial de las metodologías Scrum y Kanban. Desarrollado por el Equipo de Inkscape, software libre y de código abierto.
- **GIMP.** Para retoque de imágenes y para crear los collages de capturas de pantalla. Programa libre y gratuito desarrollado por el Equipo de desarrollo de GIMP.
- **Toggl Track.** Para gestionar y medir el tiempo dedicado a cada tarea del proyecto. Es una herramienta online, se ha usado el plan *free*. Desarrollado por el equipo de Toggl.
- **Trello.** Para la organización del flujo del trabajo, planificación de tareas y como tablero para llevar a cabo la metodología Scrumban. Es una herramienta online, se ha usado el plan *free*. Desarrollado por Trello Inc.
- **Toby Mini.** Es una extensión para el navegador que organiza las pestañas, se ha usado el plan *free*. Así solo se tienen abiertas las necesarias, y si, por ejemplo, se usa un programa de edición de vídeo o modelado a la vez que el navegador, esto hace que no se sature tanto el ordenador. También es útil si cambias de equipo solo tienes que iniciar sesión, y estarán tus pestañas organizadas y guardadas por colecciones y etiquetas. Además, ayuda para tener en cuenta en la bibliografía todos los recursos utilizados.
- **PureRef.** Es una herramienta para artistas para crear los tableros de inspiración, de referencias e ideas, es decir, los moodboards. Es gratuito y está sujeto bajo una licencia EULA. Creado por Natascha y Viktor de Idyllic Pixel.
- **Krita.** Para realizar el *storyboard*, los diseños de los distintos elementos de la animación y las ilustraciones de la app que mira la gata. Es un software de pintura digital e ilustración, libre y gratuito. Desarrollado por el Equipo de Krita.
- **Blender.** Para la realización de los modelos en 3D, la retopología, el *rigging*, *skinning*, texturizado, la creación de partículas, la iluminación, la animación y renderizado del *loop*. Es un programa gratuito y libre con múltiples opciones. Desarrollado por la Fundación Blender.

- **Adobe Express.** Para la realización de la postproducción del vídeo final y la conversión a GIF. Es una herramienta online (en estado beta ahora mismo), se ha usado el plan gratuito. Desarrollado por la compañía de Adobe.

7. Preproducción

La preproducción es la fase en la que se realizarán todas las tareas necesarias para tener clara la idea del proyecto en las siguientes fases.

En este capítulo quedarán reflejados los pasos más importantes del desarrollo de la primera fase, cómo se han realizado cada una de las tareas de la preproducción.

7.1. Brainstorming

Para llevar a cabo el brainstorming, es decir, la lluvia de ideas de qué iba a ir la animación 3D del *loop*, se tuvieron en cuenta los objetivos mencionados en el capítulo *Objetivos* de este documento.

Para empezar, primero se juntaron las ideas clave que se tenía en el momento en un documento de texto. Estas ideas eran las que se mencionan a continuación. Por una parte, la animación del *loop* estaba destinada a ser un recurso para usarse en las redes sociales, es decir, una animación corta que se pueda enviar entre amigos como **sticker** en movimiento o **gif**. La idea principal es que la **protagonista** fuese mi **gata** Kely. Esto supone que la historia que cuente la animación será representada con un gato mediante el uso literario de la **personificación**, es decir, el gato realizará acciones con cualidades, actitudes y características propias de los seres racionales. Por otra parte, el mensaje que se quiere transmitir con ironía, aprovechando que es un *loop* y puede reproducirse la animación sin encontrar el final, es de **concienciar sobre** la cantidad de horas que una persona puede pasar mirando **el móvil y las redes sociales** sin darse cuenta. Por tanto, con todas estas ideas listadas en un archivo, se procedió a hacer una búsqueda en internet sobre gifs y proyectos relacionados con estos elementos, para obtener ideas de cómo podría ser la historia que cuente el corto. También se realizó una recopilación de otros trabajos de fin de grado que tuviesen un tema parecido para saber cómo abordar la memoria y la organización de los capítulos de esta.

Una vez se tuvieron todos estos datos, ya estaba definido el problema. Solo faltaba responder a la pregunta ¿cómo puedo hacer para representarlo? Entonces, se empezaron a escribir las distintas ideas que surgieron. Después se pensó en cada una detenidamente para saber cuál de todas era la mejor para llevarla a cabo. Se hizo un planteamiento de lo que se necesitaría para realizar cada una, para saber si la propuesta era viable, y una estimación del tiempo que se emplearía. Finalmente, se concluyó que **la idea** sería hacer que la gata estuviese mirando en una

tableta o móvil una red social, inspirada en Instagram, pero de gatos, llamada Catisgram, donde estuviese pasando fotos y dándole *like*.

Para que tenga un diseño limpio, claro y sencillo, que no esté saturado, la gata estará sobre un fondo de color plano o con transparencia. Toda la animación se realizará estilo *cartoon*, para conseguir sencillez, ya que los *stickers* en redes sociales deben poder verse a escalas relativamente pequeñas, y si se crean demasiados detalles podría perderse la atención del público en otras cosas y no conseguir transmitir lo que se pretende.

7.2. Moodboard

Para organizar un **tablero de inspiración**, se usó la herramienta PureRef. Se realizó una búsqueda por plataformas como Pinterest, Freepik, Adobe Stock y Google imágenes en busca de inspiración. Una vez realizada se pusieron las imágenes seleccionadas en el tablero, se añadieron fotos de Kely, imágenes de referencia de anatomía de gatos, y se organizaron las diferentes secciones mediante títulos. A este moodboard se le añadirán más adelante, cuando se realicen, las imágenes de los diseños de los personajes, los elementos, y la paleta de color, para poder usar el tablero completo en la fase de producción y postproducción.

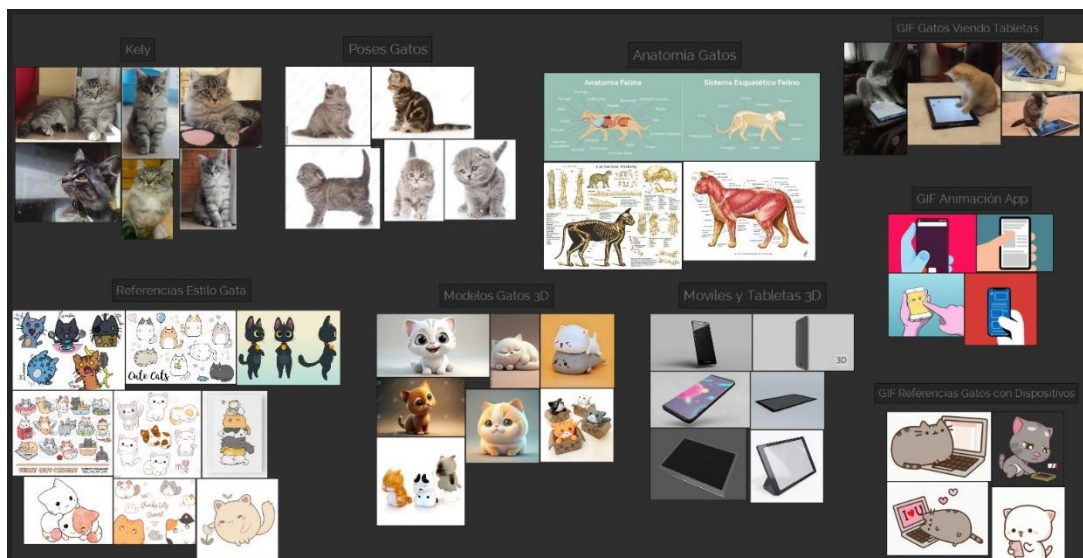


Figura 80. Moodboard inicial del proyecto realizado con PureRef
(Fuente propia: realizado con [PureRef](#))

Las fuentes de las imágenes que se ven en el moodboard se encuentran en el subapartado *Recursos externos utilizados en el desarrollo del loop*, incluido en el capítulo de la *Bibliografía* de este documento.

7.3. Guion

Para definir bien la trama de la animación se decidió realizar un guion en forma de **storyboard**. Ya que al ser una animación corta y sin sonido, es la mejor forma de preparar la idea y la distribución de las distintas escenas de una manera visual. Posteriormente este storyboard será una guía clave para la producción del proyecto. Como en este momento no se había realizado el diseño de la gata, se realizó a modo de bocetos con formas sencillas.

Primero se realizó una plantilla para *storyboards* en Inkscape y posteriormente en Krita se rellenó la plantilla y se realizaron los bocetos de las escenas. En total son 9 escenas muy cortas que expresan los pasos de la animación a realizar.

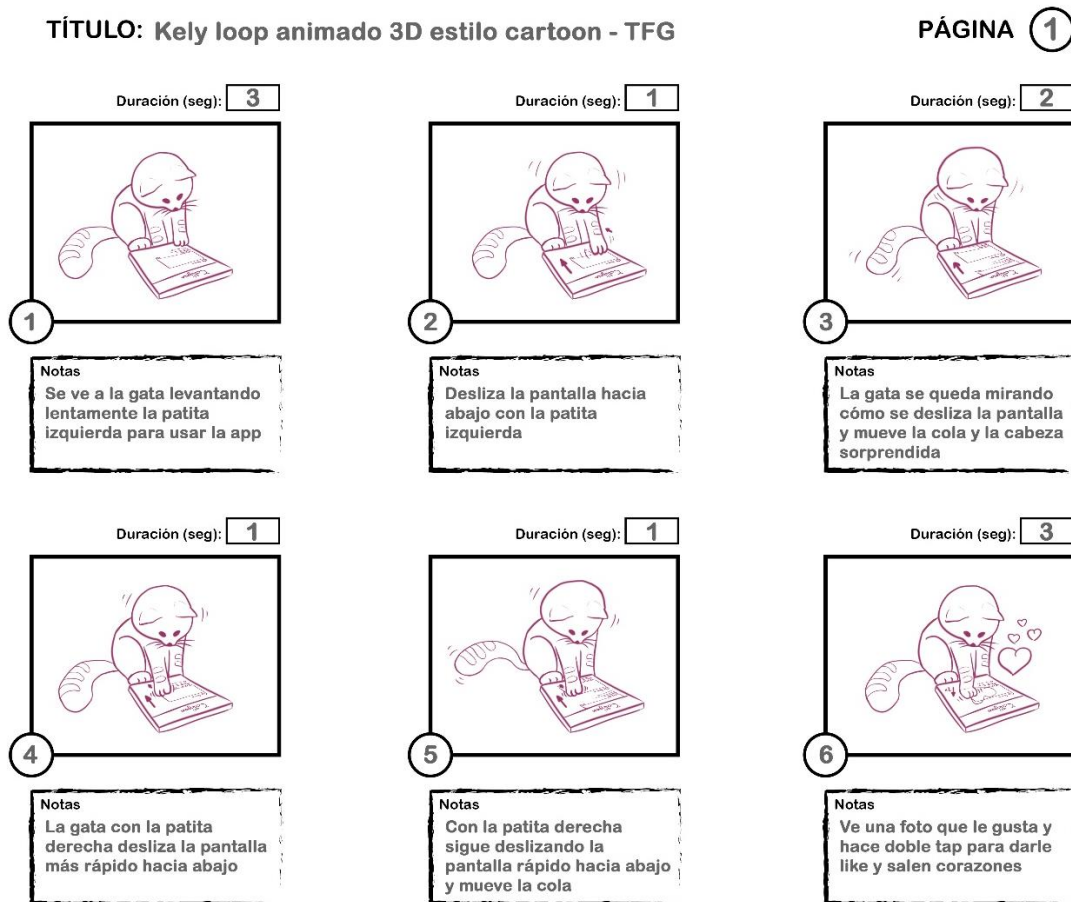


Figura 81. Storyboard del proyecto página 1
(Fuente propia: plantilla realizada en [Inkscape](#) y rellenada en [Krita](#))

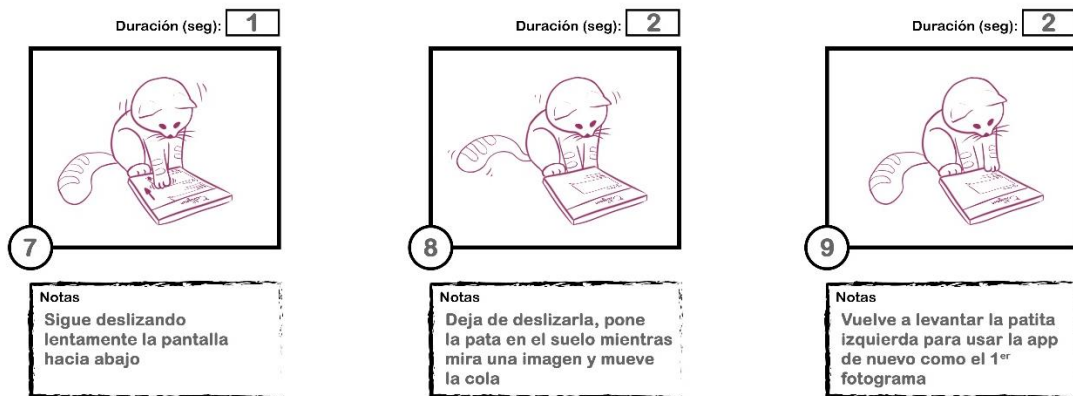
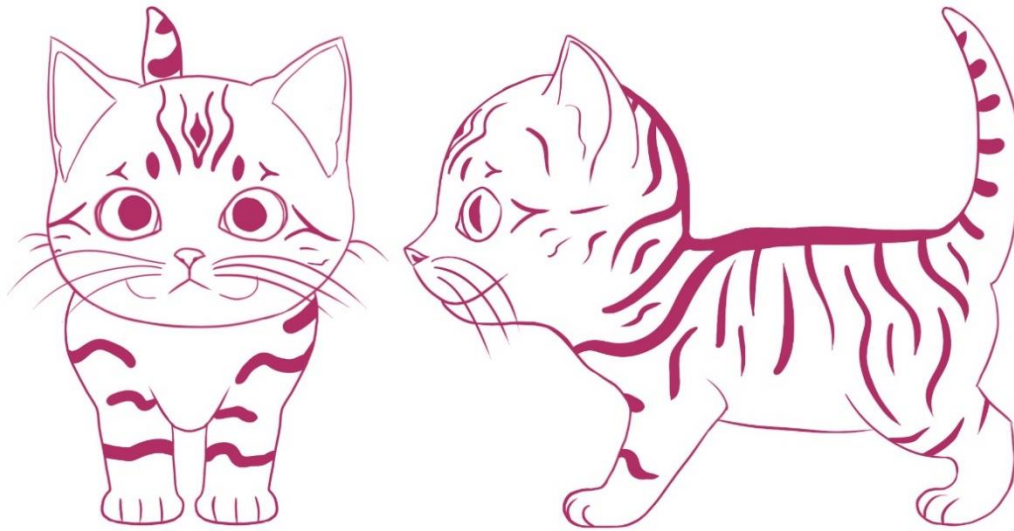


Figura 82. Storyboard del proyecto página 2
 (Fuente propia: plantilla realizada en [Inkscape](#) y rellena en [Krita](#))

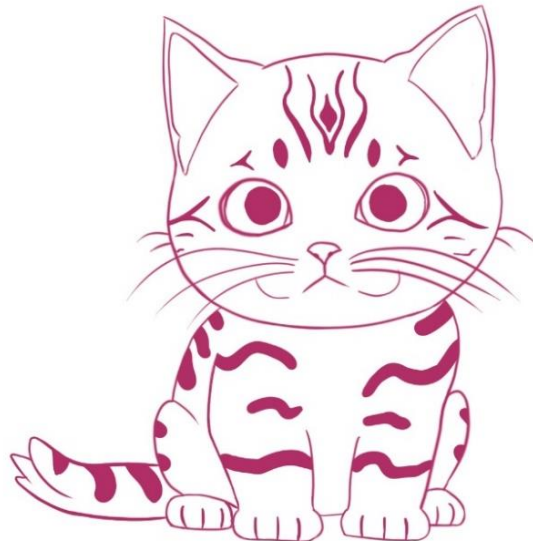
7.4. Diseño de personajes, elementos y entorno (bocetado)

Esta tarea es muy importante, ya que en ella se definirá la apariencia de todos los elementos necesarios para poder llevar a cabo la producción del proyecto. Este proceso se llevó a cabo en la herramienta Krita.

Primero se realizaron los **bocetos** de la gata protagonista **Kely**. Se realizaron los bocetos de la vista frontal y de perfil para luego poder usarlos en el modelado. También se hizo un boceto del concept del personaje para posteriormente poder indicar los colores del personaje en él. Se buscaba que fuera un estilo cartoon, sencillo sin muchos detalles y lindo. Por eso se decidió fijarse en la anatomía de gatos pequeños y crearle una cabeza más grande como si fuese estilo *chibi*. En japonés *chibi* significa pequeño, y se usa para representar de forma divertida y linda a los personajes, haciéndolos pequeños con la cabeza grande y pocos detalles en el diseño, como una simplificación de este.



*Figura 83. Boceto de Kely frente y perfil
(Fuente propia: realizado con [Krita](#))*



*Figura 84. Boceto concept art Kely
(Fuente propia: realizado con [Krita](#))*

Luego se realizaron los **bocetos de la tableta** que usará la gata en la animación. Se hicieron la vista frontal y trasera, las necesarias para modelarla, y el boceto del concept para posteriormente añadirle color y saber la paleta de color que se usará. Se inventó un logo con motivo felino para la marca del dispositivo, llamado *PeT*. Que significa mascota en inglés, pero la T final está en mayúsculas porque viene de la palabra tableta. Se usó la cursiva y elementos espirales en las letras para que el diseño fuese elegante y lindo. Para que se notase que era con motivo felino, se agregaron orejitas encima de la P y en la horizontal de la T.

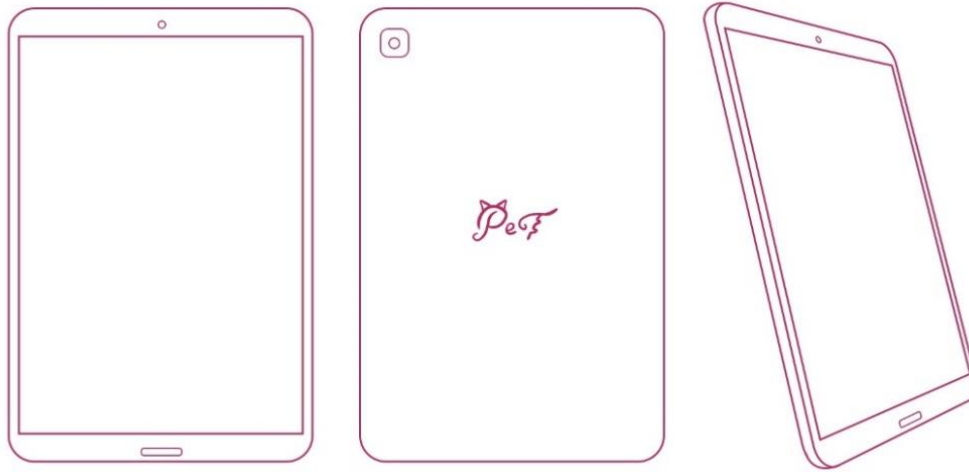


Figura 85. Boceto de la tableta
(Fuente propia: realizado con [Krita](#))

Posteriormente se realizó los **bocetos** del diseño de la **app Catisgram** que estará viendo la gata en la tableta. El nombre de la app viene de la mezcla de la palabra *cat*, gato en inglés, y del nombre de la aplicación Instagram, como guiño a que es una red social como esa, pero por y para gatos. Como la intención es que la aplicación no llame más la atención que la gata y sus movimientos, el diseño de la app consiste en formas geométricas sencillas. Del mismo modo que se hizo el logo de la tableta, se realizó el de la app, se usó un diseño para la tipografía que estuviese hecho a mano, con formas redondas y espirales para que fuese mona. También se añadió a la C una orejita de gato. Además, se realizó el diseño del corazón para cuando la gata le de like a una imagen y esa misma imagen se usará de textura en la pantalla de la tableta, y para el sistema de partículas de los corazones que se verán saliendo del dispositivo cuando le de *double tap* a la pantalla.

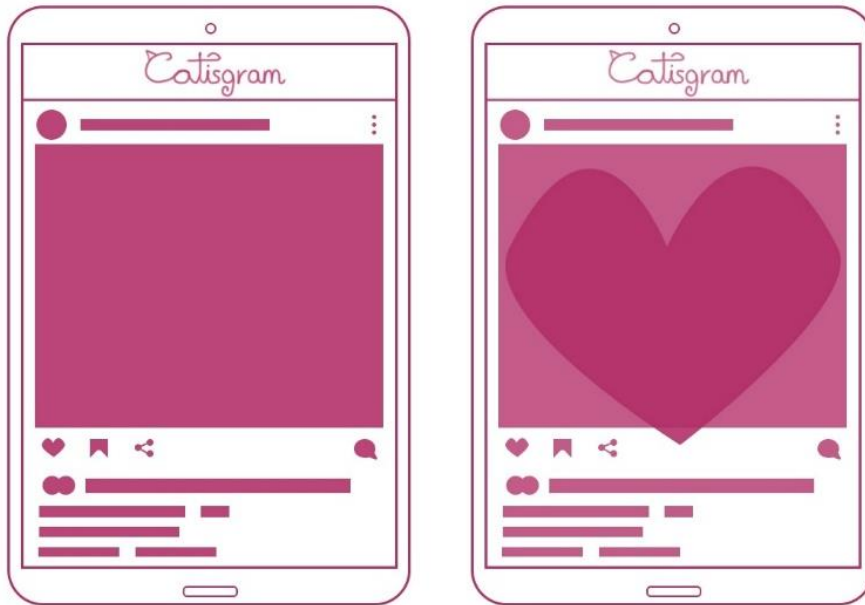


Figura 86. Bocetos de la app con y sin like en la pantalla de la tableta
(Fuente propia: realizado con [Krita](#))

Por último, se terminaron los **concepts** del personaje y los elementos para establecer la **paleta de color** de cada uno. Los colores de la gata se escogieron a partir de fotografías de Kely, adecuándolos al diseño y a lo que se quiere transmitir. Los de la tableta, a partir de los colores de línea de la gata, sabiendo cuál era el umbral de oscuros y claros. La app en escala de grises para simular las figuras y textos y el corazón en una mezcla de rosa y rojo para que aporte más énfasis a la sensación que se quiere transmitir de que es una animación linda, tranquila y suave. También, se escogió un color plano para el fondo, para así conseguir que el entorno no llame más la atención que la gata y las acciones que realice. Este color de fondo fue escogido para dar una sensación de tranquilidad y que a la vez crease contraste con el resto de las paletas de color.



Figura 87. Fotografía de Kely real para crear la paleta de color
(Fuente propia: realizado con una cámara)

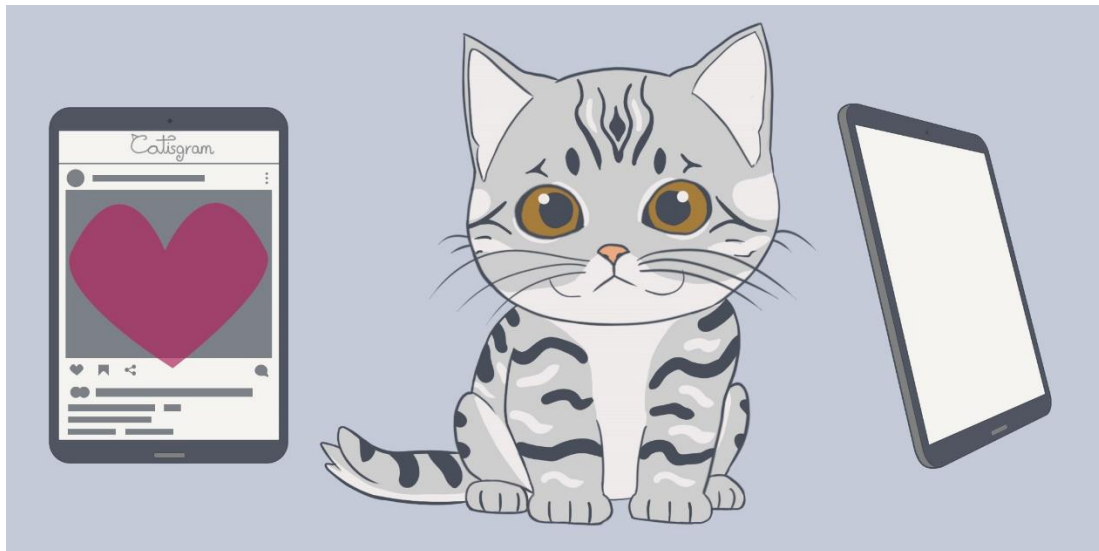


Figura 88. Concepts art de Kely, la app con el like y la tableta
(Fuente propia: realizado con [Krita](#))

8. Producción

La producción es la fase en la que se realizarán todas las tareas necesarias para llevar a cabo la idea que se ha concebido en la fase anterior del proyecto, obteniendo como resultado los distintos elementos gráficos que lo compondrán.

En este capítulo quedarán reflejados los pasos más importantes del desarrollo de esta fase, cómo se han realizado cada una de las tareas de la producción. Todas estas tareas fueron realizadas con el programa Blender en la versión 3.6, menos las imágenes de la app que se añadieron como texturas a la pantalla de la tableta, estas se realizaron con Krita. El programa de Blender estaba configurado en la interfaz de las preferencias con el lenguaje *English (English)*. Por eso, en este documento las direcciones a las herramientas y otras opciones se escriben en inglés.

Aunque durante el proceso se usan muchos atajos de teclado y/o comandos, en este documento se escribirá la dirección completa a cada herramienta y/u opción, para que se entiendan mejor las herramientas utilizadas, y pueda servir la memoria como una guía accesible (por si alguien tiene otra configuración de los atajos de teclado que no sea la que se usó).

8.1. Modelado

Como los **modelos** se querían crear pensando en que el proyecto es una animación, se realizaron todos los detalles necesarios para hacerlos de **alta calidad** con mallas de **alta poligonación**, pero optimizadas, sin que hubiese polígonos de más sin sentido. Se tuvo en cuenta que las **mallas** de los modelos fuesen en todo momento **correctas**, es decir, no se le creasen agujeros, ni hiciesen cosas extrañas las caras y estuviesen **formadas por quads**. Durante todo el proceso, cada cierto tiempo o al finalizar cada fase de modelado se **comprueba** con la herramienta *Clean Up* que hay en el *Edit Mode* (modo de edición de los elementos que forman el objeto), con todos los vértices del modelo seleccionados en *Select > Select All by Trait > Faces by Sides*, luego sale una opción bajo a la izquierda en el viewport donde se selecciona el número de vértices, en este caso 4 para comprobar los quads y el tipo que si es menos que, igual a, mayor que o no igual al número de vértices que se ha señalado. Si hay caras que cumplan la condición escogida, serán las únicas que queden señaladas. Si no están formadas por 4 vértices, habrá que editarlas para que cumplan esa condición. Siempre teniendo en cuenta el modelo y su función, hay ocasiones

especiales en las que es preferible tener un n -gon o un tri si va a ser más fácil el manejo de los elementos de la malla del objeto, o si tiene sentido por la forma de esa parte del modelo.

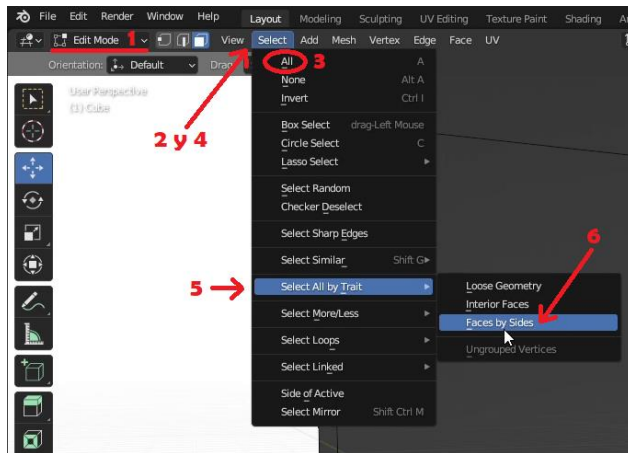


Figura 89. Comprobar si la malla es correcta y está formada por quads
(Fuente propia: realizado en [Blender](#))

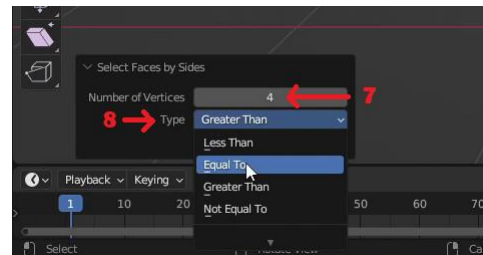


Figura 90. Configuración de la comprobación de la malla formada por quads
(Fuente propia: realizado en [Blender](#))

También, se **comprobaba la correcta orientación de las caras** con la opción en *Viewport Overlays > Face Orientation*, si las caras externas se ven azules y las interiores del modelo rojas, está correcto, si no, las caras que estén al revés hay que cambiar la orientación.

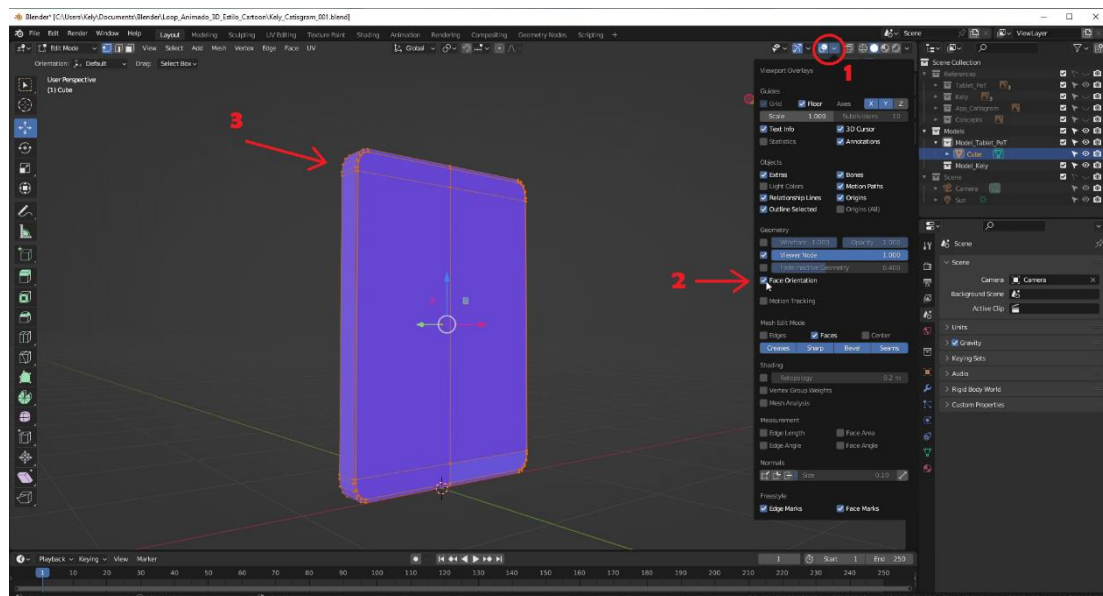


Figura 91. Ejemplo comprobación orientación correcta de las caras del modelo
(Fuente propia: realizado en [Blender](#))

Asimismo, para **suavizar visualmente los bordes de los objetos** sin tener que crear demasiada geometría se puede usar la opción *Object > Shade Smooth* en el *Object Mode*, y si se necesita **desactivar la opción**, para que se vean más planos, se utiliza *Object > Shade Flat*, para que cree o quite esa ilusión de bordes redondeados en el objeto seleccionado.

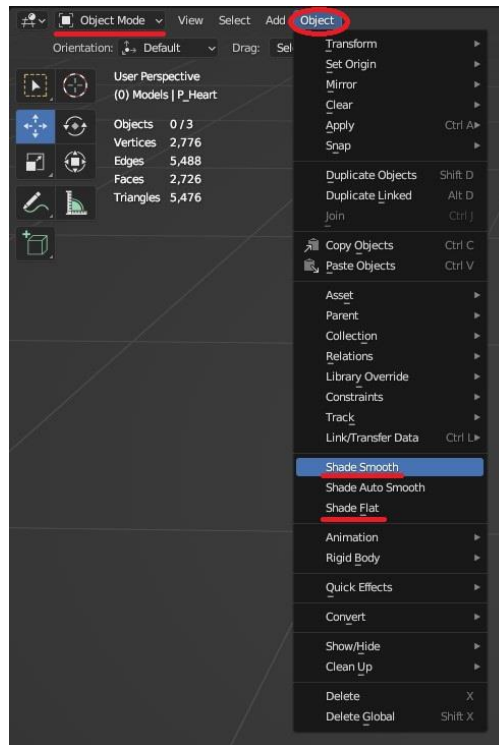


Figura 92. Opción para que se vean redondos o planos los bordes de un objeto
(Fuente propia: realizado en [Blender](#))

8.1.1. Preparación del proyecto

Primero de todo se preparó la **carpeta del proyecto** para realizar el trabajo, para tener un orden, siempre se crean todas las carpetas, aunque luego se queden vacías. Se creó la carpeta principal del proyecto llamada **Loop_Animado_3D_Estilo_Cartoon**, que dentro contendría:

- Los **archivos .blend** (de Blender) de las distintas versiones que se irían guardando del proyecto. Se numeran según el orden del progreso del archivo.
- Las **carpetas**:
 - **Animation**. Donde se guardarán los distintos vídeos (las pruebas y el vídeo final) que se generen en el render.
 - **External Models**. Donde se incluirán los modelos que se usen en el proyecto provenientes de otras fuentes. En este proyecto no se usaron otros tipos de recursos externos, así que esta carpeta se quedó vacía.
 - **Materials**. Esta carpeta suele contener materiales que se usen externos ya sean propios creados en otras aplicaciones o descargados como recurso externo. Pero en este proyecto no se usaron otros materiales aparte de los configurados en Blender.

- **Objects.** Aquí se guardarán las distintas exportaciones de los modelos, materiales, luces y cámaras creados en el proyecto.
- **References.** Aquí se incluyen las referencias creadas en la fase de preproducción, necesarias para realizar los modelos y saber las paletas de color que se utilizarán en el texturizado y materiales.
- **Renders.** Aquí se guardarán los distintos renders generados en las pruebas de iluminación, materiales, texturas y los del modelo final.
- **Textures.** En esta carpeta se guardarán los distintos mapas de textura y las texturas necesarias para el proyecto.

Una vez organizada la carpeta del proyecto, se introdujeron los bocetos y concepts en la carpeta de *References*. Se abre Blender y se **configura el archivo inicial**. Para ello se borran los objetos que vienen por defecto creados (un cubo, una luz y una cámara). Se organizan los distintos elementos en colecciones. Existen una serie de colores que se pueden aplicar a las colecciones para visualmente diferenciarlas mejor. Hay una colección principal llamada **Scene Collection**. Dentro de esta se organizaron las siguientes colecciones:

- **References.**
 - **Tablet_PeT.** Se incluirán las referencias de bocetos de la tableta, la de frente y la trasera.
 - **Kely.** Se incluirán las referencias de bocetos de la gata, la de frente y perfil.
 - **App_Catisgram.** Se incluirán las referencias de bocetos de la app.
 - **Concepts.** Se incluirán los concepts para poder sacar las paletas de color.
- **Models.**
 - **Model_Tablet_PeT.** Los objetos necesarios que conformen la tableta.
 - **Model_Kely.** Los objetos que conformen el modelo de la gata.
 - **Retopology.** Los modelos de alta poligonación y el de la retopología que no sean los que se usen en el render final.
 - **Stage.** Los objetos necesarios para formar el escenario de la escena.
- **Scene.** Los objetos necesarios para las luces de la escena y el objeto cámara.
- **Locked.** Esta colección se irá usando en cada fase de la realización del proyecto para introducir los modelos que no se estén usando en ese momento, y se bloquearán para no editarlos sin querer. Es para cuando se necesite ver estos objetos porque junto con el que se esté editando forman el objeto final.

Una vez creadas las colecciones se arrastran desde la carpeta de *References* las imágenes de los bocetos, y se colocan en el espacio 3D correctamente. Se aplica cierta transparencia para que no molesten al modelar y poder ver bien así el modelo con el que se está trabajando. Se puede activar la opción de **In front** en *Editor Type* la opción *Properties* > *Object* > *Viewport Display* > *In front*, cuando sea necesario para que se esté en la vista que se esté se vea la referencia por delante del modelo. Esta opción se puede usar en cualquier tipo de objeto en Blender, se usará también en el *rigging* para ver el esqueleto.

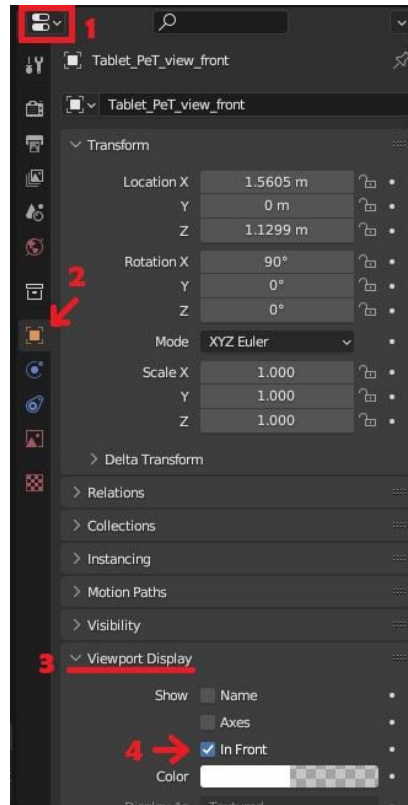


Figura 93. Opción *In Front* para cualquier objeto en Blender
(Fuente propia: realizado en [Blender](#))

Una vez introducidas las referencias, la organización del proyecto en el *Editor Type* la opción *Outliner* queda como se muestra en la siguiente figura:

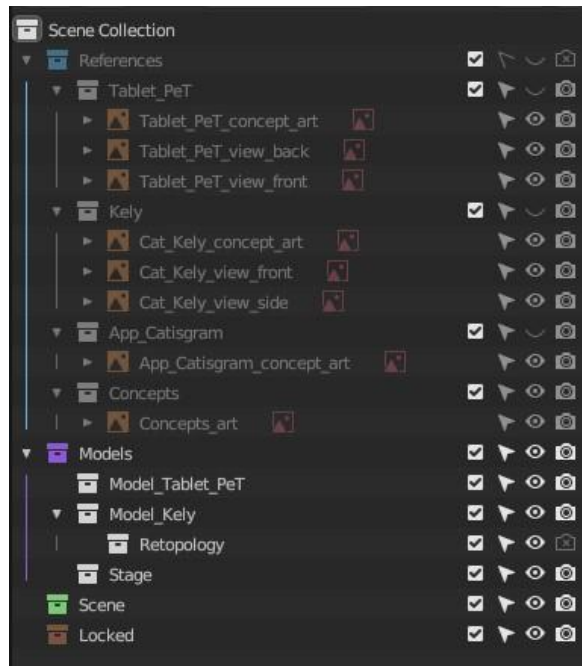


Figura 94. Organización de las colecciones en Blender
(Fuente propia: realizado en [Blender](#))

Con las flechitas que hay al lado izquierdo de las colecciones, que dentro contienen otra colección u objeto, y de los objetos, sirven para plegar (flechita señalando hacia la derecha) y desplegar (flechita señalando hacia abajo) lo que contienen dentro, así es más sencillo buscar y encontrar los elementos si están organizados. En el caso de los objetos si se despliegan se pueden ver los modificadores (en el *Editor Type* la opción *Properties > Modifiers > Add Modifier*) si no se han aplicado, las referencias externas al programa que esté usando el objeto como las imágenes, y otras propiedades del objeto se muestran aquí. Los iconos que se muestran a la derecha sirven para:

- El **cuadrado con la verificación** es solo para las carpetas de las colecciones si está activado se muestra en el *viewport* (espacio 3D) todo lo que contenga la colección, sino no. Si no se muestran en el *viewport* además no se pueden seleccionar ni editar.
- La **flecha** si está activada significa que se pueden seleccionar el objeto u objetos de la colección, sino es que están bloqueados y no pueden seleccionarse, y, por tanto, no se pueden editar.
- El **ojo** si está activado (abierto) se ve el objeto o colección en el *viewport*, si está desactivado (cerrado) no.
- La **cámara** si está activada (con el objetivo sin tachar) ese objeto o colección se verá en el render, si está desactivada (con una cruz en el objetivo) no saldrán en el render final.

Una vez se tiene esto, ya está organizado el archivo inicial, y se puede pasar a modelar los objetos que compondrán la escena.

8.1.2. Modelo de la tableta

Para el **modelo de la tableta**. Ahora se verán las **referencias** que se colocaron en el entorno 3D, en una captura se ve las referencias en la **vista frontal (front)** que se ha configurado como se mencionó anteriormente con transparencia y para que se vea por delante de los objetos, y la **vista trasera (back)** se ha añadido la misma imagen girada sobre el eje Z (altura) con la configuración de la transparencia y que esta imagen solo sea visible en esta vista, para que no se mezclen en todas las vistas y confundan. Para que esto sea así en la **configuración** hay que escoger que el lado visible de la imagen que se vea sea el *Front* porque se ha girado 180º sobre el eje Z para que el dibujo estuviese derecho en la vista trasera, y se quiere que sea visible la imagen de frente, que en este caso está mirando hacia el eje Y (la profundidad en dirección positiva), cuando los objetos normalmente miran al eje -Y (la profundidad en dirección negativa). Es por eso por lo que se ve en la imagen frontal un marco negro que pertenece a la imagen trasera, pero no muestra la imagen en ninguna vista que no sea la trasera.

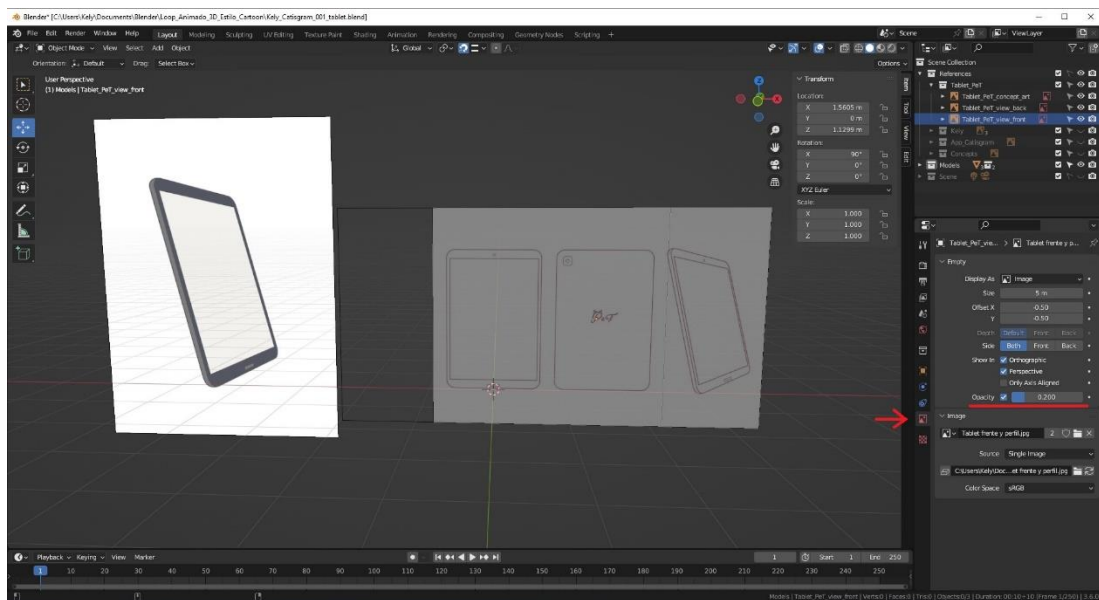


Figura 95. Configuración de la referencia vista frontal de la tableta
(Fuente propia: realizado en [Blender](#))

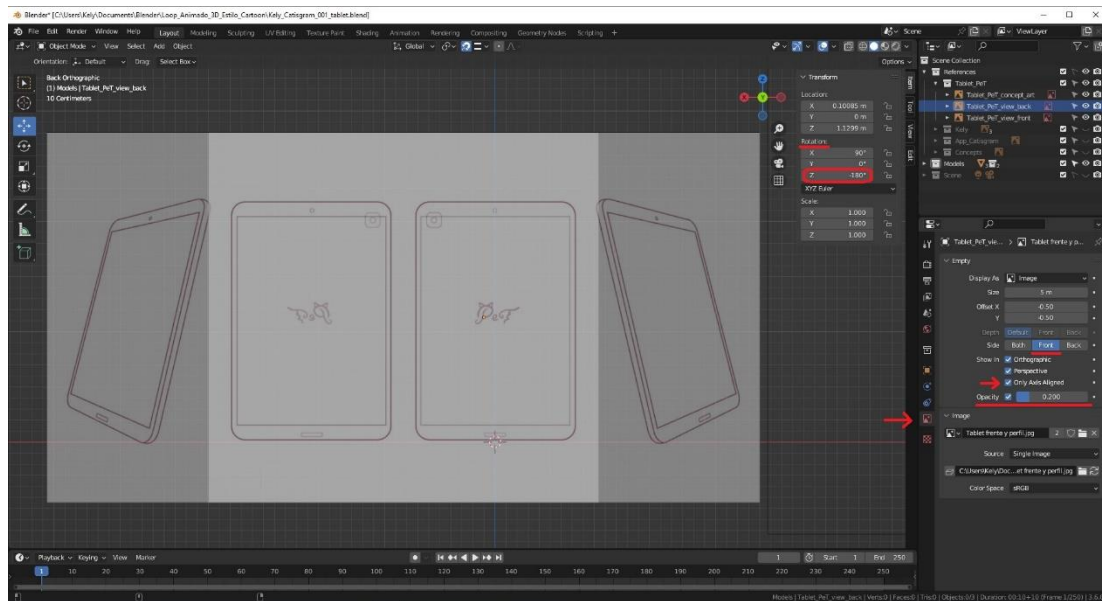


Figura 96. Configuración de la referencia vista trasera de la tableta
(Fuente propia: realizado en [Blender](#))

A continuación, se hizo el modelo 3D de la tableta a partir de las referencias. Como es un **modelo sólido** se usó el método de **box modelling**. Se creó un **cubo** con **Add > Mesh > Cube** en el **Object Mode** que es el modo que se usa para colocar los distintos objetos en el menú. Una vez creado, se cambiaron las escalas en los 3 ejes para que tuviese el tamaño y forma general de la imagen de referencia y se le dio la profundidad deseada, quedando un prisma rectangular con las dimensiones del boceto. Una vez hecho eso se pasó al **Edit Mode**, donde se pueden modificar, añadir y eliminar vértices, aristas y caras del objeto específico que pase a este modo. Se realizaron dos **loops horizontales** con la herramienta **Loop Cut** (cortar un loop en la malla) en el modelo y se colocaron a uno en la parte de arriba de lo que ocuparía el alto de la pantalla de la tableta, y el otro en la parte de abajo. Luego se hizo otro **loop** esta vez **vertical** que pasase por el centro de la tableta, porque agiliza el trabajo ya que facilita trabajar con simetrías, pero más tarde se borraría este loop para disminuir la cantidad de polígonos de este modelo y así optimizarlo. A continuación, en las aristas de los bordes de la tableta se aplicó la herramienta **Bevel Edges** para **redondear los bordes** y que el modelo sea más realista. Luego se seleccionan las caras de la parte delantera de la tableta, donde va el cristal de la pantalla y se hace un pequeño **Insert Faces** hacia dentro que **crea un borde en la pantalla donde va el cristal**, en el mismo plano en el que se encuentran las caras seleccionadas sin cambiar su posición en Y. Sin deseleccionar estas caras, se hace una pequeña **extrusión hacia dentro** del teléfono con la herramienta **Extrude Region**, otro pequeño **Insert Faces** hacia dentro para hacer **otro borde** y se vuelve a **extruir** esta vez **hacia afuera**, con un poco más de altura de la que tenía al principio el

cristal de la pantalla. Así se crea una pequeña ranura que simula la que hay en los móviles y tabletas donde se conecta el cristal de la pantalla a la carcasa. Siguiendo, se seleccionan las **caras laterales que formal la profundidad de la tableta** y se hace un *Insert Faces* para **crear dos bordes en el interior** y que al añadir el modificador en el *Editor Type* la opción *Properties > Modifiers > Add Modifier > Subdivision Surface* (que divide los polígonos de la malla del objeto para crear geometría que suavice y cree una malla de alta poligonación), respete la profundidad del objeto. Se añade el modificador mencionado con la cantidad de veces que **dividirá la malla para el render**, en este caso se escogió 3 por cómo se buscaba que se viese el acabado. Este modificador no es necesario aplicarlo, así se podrá volver a la malla de baja poligonación si es necesario retocarla. Igual que se ha hecho con el lateral de la tableta, se seleccionan las **caras de la parte trasera de la tableta** y se introduce un *Insert Faces*, por la misma razón que se realizó en el lateral. Se borra el *loop* vertical que se hizo anteriormente. Se hacen las **comprobaciones** de orientación de las caras y quads. En este caso se puede comprobar que hay *n-gons* en las caras delanteras y traseras de la pantalla de la tableta. Para la parte trasera esto se arregló fácilmente seleccionando, de dos en dos, los vértices (que están enfrentados en la horizontal) de los extremos de las caras de los *n-gons*, y con la herramienta *Vertex > Connect Vertex Path* que une los vértices creando las aristas y vértices necesarios para cortar las caras que haya en medio de ellos. Pero para la parte delantera (aunque se podría arreglar de la misma forma) no se hizo, porque dejándola así facilitará el trabajo de más adelante para crear la pantalla donde se verá la app y al añadir los materiales. Además, al ser un objeto que en la animación no se va a doblar, es rígido, que haya *n-gons* ahí no va a generar ningún error en el render de la animación. Por tanto, se podrían haber dejado los de la parte trasera también como *n-gons*. Pero se decidió formar los *quads* para mostrar que tiene una sencilla solución, y que el modelo tenga una topología más correcta con menos *n-gons*. Para terminar esta parte del modelo se cambia al *Object Mode*, y se **nombra el objeto** en el *Outliner* como *Tablet*.

Para mostrar la figura de la tableta que se ve a continuación, se ha usado el *Edit Mode* con selección de caras, con el *Viewport Shading* que muestra el objeto en modo sólido, sin transparencias, y así se ven las aristas que forman el objeto.

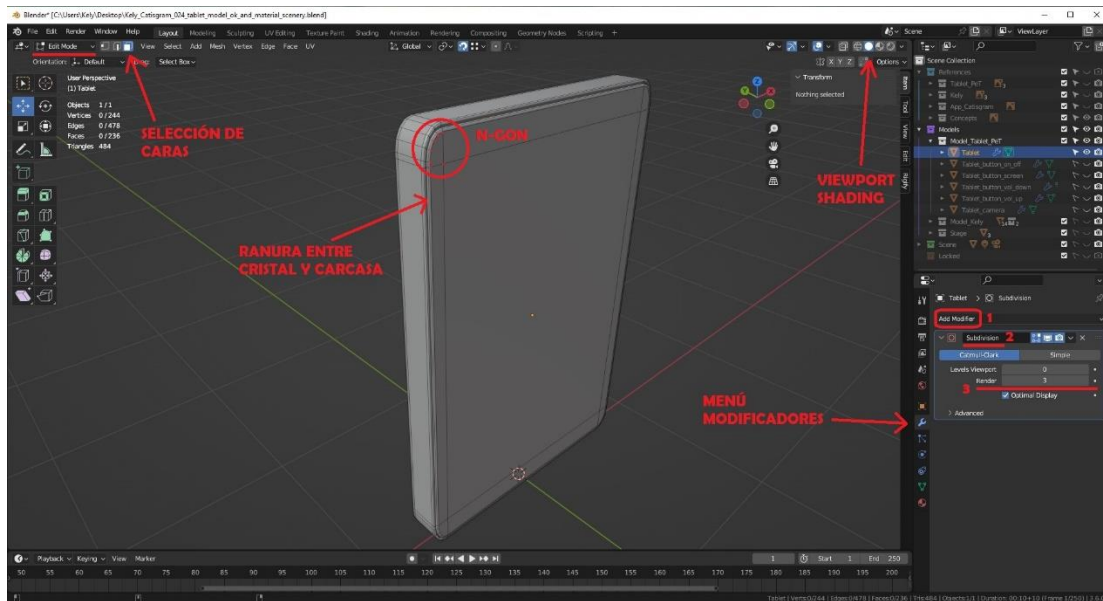


Figura 97. Vista delantera del objeto inicial de la tableta
(Fuente propia: realizado en [Blender](#))

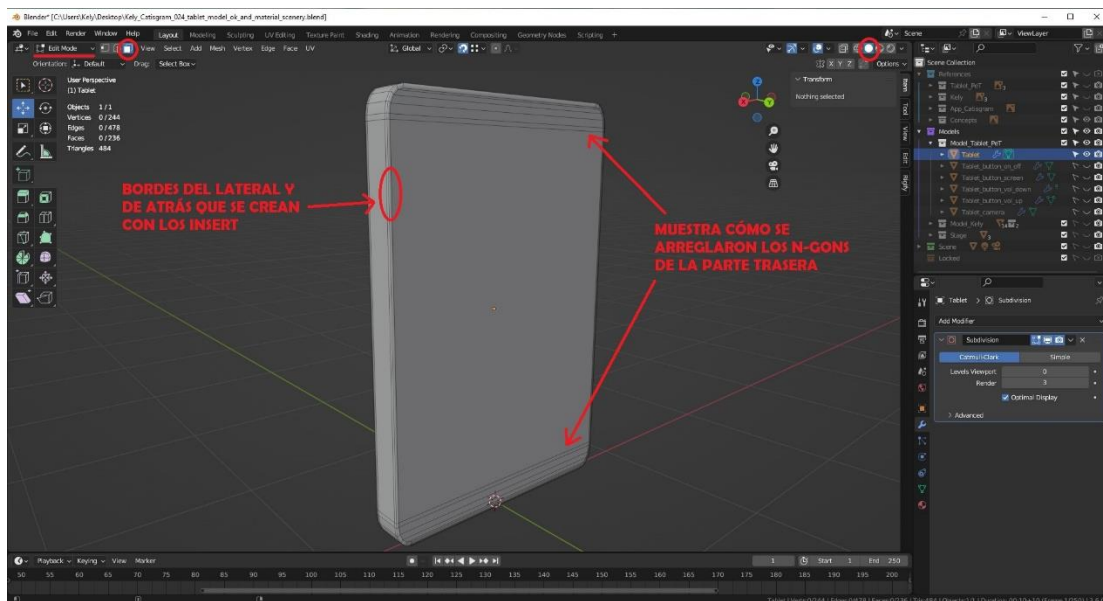


Figura 98. Vista trasera del objeto inicial de la tableta
(Fuente propia: realizado en [Blender](#))

A continuación, se selecciona el objeto *Tablet*, y se pasa al Edit Mode. Se **seleccionan las caras frontales del cristal**, con la opción Mesh > Duplicate, se duplica la cara, sin deseccionarla se va a la opción Mesh > Separate > Selection, para que **separe esta cara del objeto** y forme parte de un objeto independiente. En el Object Mode, **se mueve** este objeto un poco hacia el interior de la tableta para posteriormente poder **usarlo para mostrar lo que se verá en la pantalla del dispositivo**. Se nombra el objeto como *Tablet_screen*.

Para mostrar la figura del objeto de la pantalla de la app, se seleccionó el objeto *Tablet_screen* para que se pueda apreciar su dimensión, que es igual al cristal de la pantalla de la tableta, y que está introducida dentro del modelo de la *Tablet*. Se puso el *viewport* en modo *Toggle X-Ray* que ofrece una visión como de rayos X, para que se transparenten los objetos y se vea a través de ellos, así se puede apreciar mejor que el objeto seleccionado está dentro del otro.

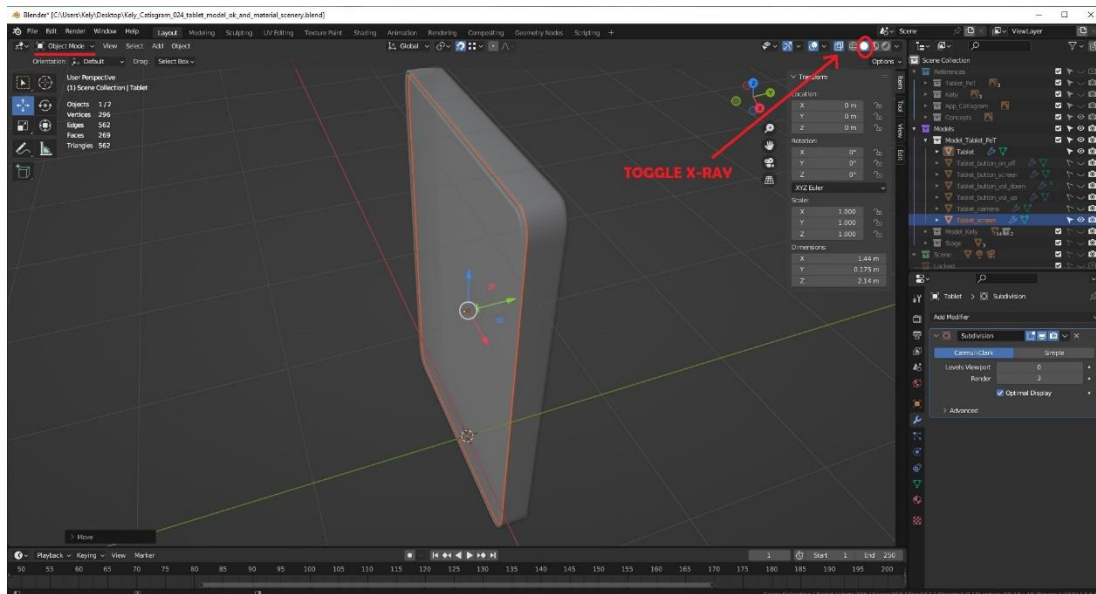


Figura 99. Objeto de la pantalla de la app dentro de la tableta
(Fuente propia: realizado en [Blender](#))

Para el **botón** de delante de la pantalla, se crea otro cubo. **Se escala** en los distintos ejes para que tenga las **dimensiones** que tiene en el boceto, con la profundidad deseada, y **se coloca** a la altura que estará en la pantalla del objeto *Tablet* creado anteriormente. En el *Edit Mode*, se seleccionan las aristas de las esquinas del botón y se hace un *Bevel Edges* para **redondear los bordes**. Como para el cuerpo de la tableta, se seleccionan las caras laterales del botón y se hace un *Insert Faces* para **crear los dos bordes** y que así las opciones de *Shade Smooth* o *Subdivision Surface*, que se puedan añadir en el futuro, **respeten la profundidad del botón**. En este caso en las caras delanteras y traseras del botón no se hace un *Insert Faces* porque la intención es que se redondeen y quede abombado el botón por el frente. Luego se sale del modo edición al Object Mode. **Se nombra** en el *Outliner* como *Tablet_button_screen*. Aunque en el boceto no se dibujaron, **se van a añadir 3 botones: dos** para simular los que se usan para subir y bajar el **volumen** del dispositivo, que se añadirán en el lateral derecho; y otro al lado izquierdo, a la misma altura que el de subir el volumen, para que sea el **botón de apagar y encender** el dispositivo y el de bloqueo de pantalla. Así estéticamente parecerá más una tableta. Para ello, **se duplica el objeto del botón 3 veces** seleccionándolo y clicando la opción *Object > Duplicate*

Objects, y se colocan en el sitio especificado anteriormente. Después se nombran como: *Tablet_button_vol_up*, *Tablet_button_vol_down* y *Tablet_button_on_off*.

Ahora se muestra una figura del botón de la pantalla para que se vea la topología, en el modo edición, al tener aplicado el modificador *Subdivision Surface* se ven las aristas reales del objeto, y la malla del objeto se ve con una forma redondeada, como sería si se aplicase el modificador, pero no se aplica para poder tener la malla con menos poligonación accesible.

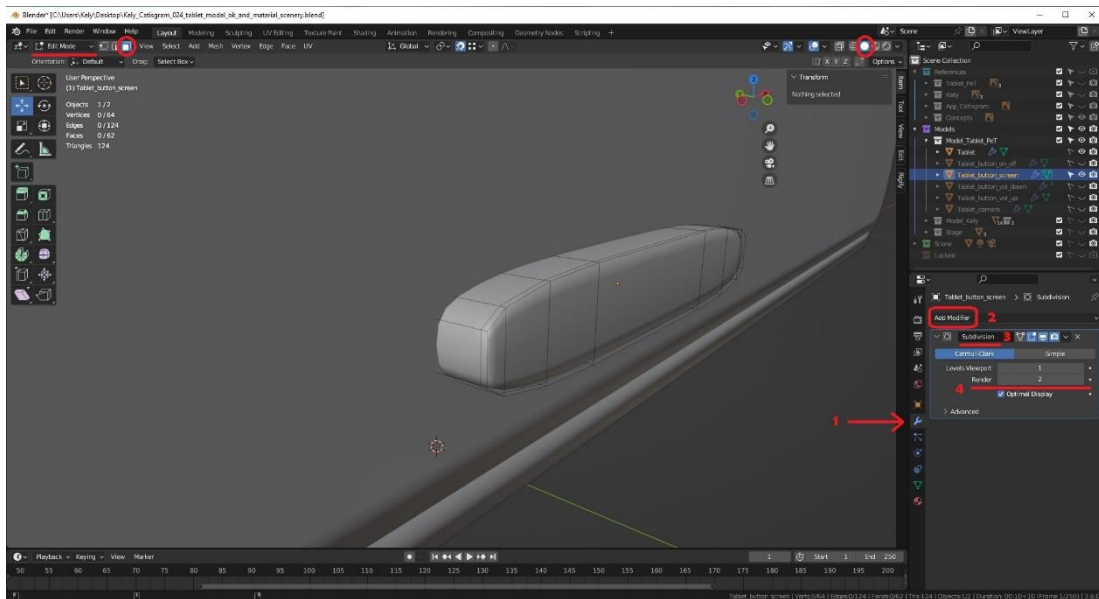


Figura 100. Ejemplo del objeto botón de delante de la pantalla
(Fuente propia: realizado en [Blender](#))

Para la cámara se siguen los mismos pasos que con el objeto *Tablet*, pero con las dimensiones de la cámara. Se nombra el objeto como *Tablet_camera*. E incluso si no se quisiese añadir, se podría no hacer, ya que la tableta va a estar apoyada en el suelo boca arriba y la cámara de detrás no se verá. Por este motivo además no se hará como con la pantalla que se ha creado un objeto en el interior para luego mostrar las imágenes de la aplicación, esto se podría aplicar en la cámara para texturizar el objetivo que está protegido por el cristal, pero no se ha realizado este paso porque no es necesario en este proyecto. Si se quisiese hacer un modelo de tableta o móvil más detallado y realista preparado para cualquier otro proyecto se podría hacer de esta manera.



Figura 101. Objeto cámara de la tableta
(Fuente propia: realizado en [Blender](#))

Para finalizar el modelo 3D de la tableta, se realizan las **comprobaciones** mencionadas en **cada uno de los objetos**. Por último, **se seleccionan los objetos: *Tablet*, *Tablet_button_vol_up*, *Tablet_button_vol_down*, *Tablet_button_on_off* y *Tablet_camera***, **se unen en un solo objeto** con la opción *Object > Join*. **Se nombra** el objeto como *Tablet_union*. Estos objetos se unen porque para la animación se les añadirá el mismo material, así que se pueden unir para facilitar el manejo de todo el objeto y agilizar el trabajo más tarde. Además, que son elementos que siempre van a estar en esa posición de la tableta. Igualmente, si se quisiesen separar en objetos distintos se podría seleccionar una de las caras de por ejemplo uno de los botones, y con la opción *Select > Select Linked > Linked* se seleccionaría solo los polígonos de ese botón y con la opción *Mesh > Separate > Selection*, se podrían volver a separar del objeto. El objeto *Tablet_screen* no se une a los demás, porque este se quiere que se visualice con *Shade Flat*, y el resto con *Shade Smooth*.

Para la figura del modelo de la tableta final, se seleccionó el objeto *Tablet_screen* para que se pueda apreciar su dimensión, que es igual al cristal de la pantalla de la tableta, y que está introducida dentro del dispositivo, es el objeto que se ve seleccionado. Se puso el *viewport* en modo *Toggle X-Ray* para que se trasparenten los objetos y se vea a través de ellos, así se pueden ver todos los botones y la cámara en sus posiciones, aunque estén detrás de la tableta.

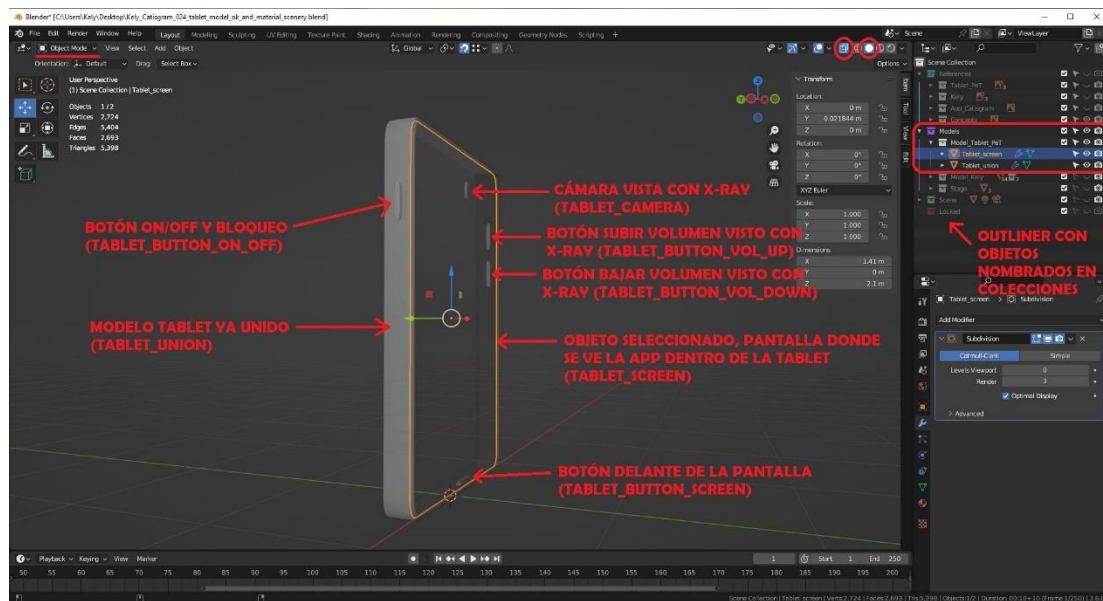


Figura 102. Modelo entero de la tableta con los objetos que lo forman
(Fuente propia: realizado en [Blender](#))

8.1.3. Modelo de la gata Kely

Para el **modelo de la gata**. Ahora se verán las **referencias** que se colocaron en el entorno 3D, en una captura se ve las referencias en la **vista frontal** (*front*) que se ha **configurado** como se mencionó anteriormente con transparencia y para que se vea por delante de los objetos, y la **vista lateral** (*right and left*) se ha añadido la misma imagen girada sobre el eje Z (altura) 90° con la configuración de la transparencia. Como se ve en el lado izquierdo y derecho, se usa esa misma imagen para los dos lados.

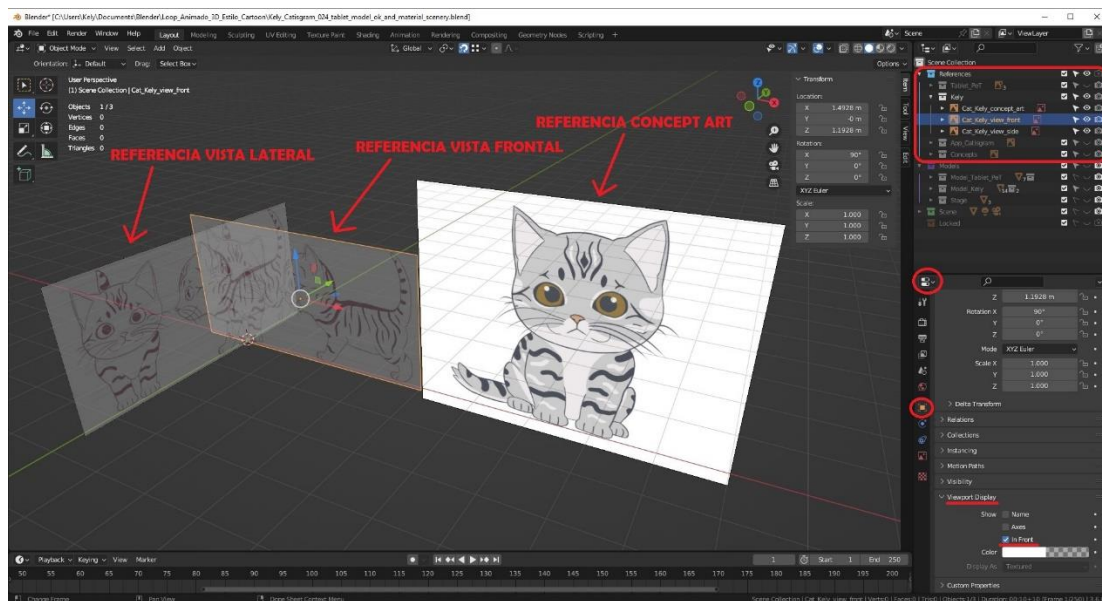


Figura 103. Configuración de las referencias de la gata
(Fuente propia: realizado en [Blender](#))

A continuación, se realizó el modelo 3D de la gata a partir de las referencias. Como es un **modelo orgánico** se usó el método de **escultura digital**, para conseguir que el modelo tuviese una forma más natural de un gato. Para ello primero mediante **primitivas básicas**, en este caso cubos, porque así es más fácil que la malla se forme con *quads*, se hizo el **blocking** del animal, que es la forma básica que se crea con primitivas dándoles una primera forma general, para posteriormente poder detallar cada parte del cuerpo. Para realizar el blocking, se creó **un cubo** con **Add > Mesh > Cube para la cabeza y cuello**, se colocó en la posición 3D donde en las referencias está la cabeza. A ese cubo se le añadió un **modificador** en el *Editor Type* la opción *Properties > Modifiers > Add Modifier > Multiresolution*, este modificador permite **subdividir la malla** tantas veces como se necesite, pero tiene herramientas que hace que puedas **volver a las subdivisiones anteriores**. De manera que al final se tiene el resultado en *high poly*, pero puedes volver a la versión de *mid poly* y *low poly* si es necesario. De hecho, se puede poner la versión *high poly* solo para el render y también mientras se está en el *Sculpt Mode*, y en el *Object Mode* se vea la versión *low poly* del modelo para que el programa no tenga que mover tantos polígonos y se pueda trabajar más cómodamente. A este modificador se le da **una primera subdivisión** de la malla del cubo que hace que sea más redonda, se escala con las **dimensiones de la cabeza** del gato, y **se nombra** como *Kely_head*. A continuación, se hace lo mismo con otro **cubo para el cuerpo**, llamado *Kely_body*, y otro **para la cola**, llamado *Kely_tail*. Luego **otro cubo para una de las orejas**, llamado *Kely_ear*. Además del modificador *Multiresolution*, en este caso **se añade otro modificador** en el *Editor Type* la opción *Properties > Modifiers > Add Modifier > Mirror*, que permite crear una **simetría del objeto**, y si seleccionas desde que objeto se quiere hacer la

simetría se crea a la misma distancia de ese objeto en el eje que se escoja. Por tanto, se escoge **la cabeza** como objeto desde el que se quiere hacer el *Mirror* en el eje X. Así con que se modele una oreja, se modelará la otra. Esto mismo que se hace para la oreja, se hace para las **patas delanteras**, llamado *Kely_front_leg*, y para **las traseras** (*Kely_back_leg*) con un *Mirror* en el eje X **desde el objeto del cuerpo**. Para terminar de **preparar los objetos**, que formarán cada una de las partes del cuerpo de la gata, **para el siguiente paso**, es importante “limpiar” o resetear sus transformaciones de rotación y escala. Aunque para estar más seguros de que no dará problemas en el futuro, se aplicaron todas las transformaciones, no solo rotación y escala. Esto sirve para que los escalado y rotaciones que se le ha hecho al objeto no interfieran en los cálculos futuros de desplazamiento (mover, rotar o escalar) de los vértices de la malla. Así, reconoce como localización, rotación y escalado inicial en todos los ejes (es decir, todos estarán a 0, 0, 0). De esta forma, internamente no tiene que hacer los cálculos desde el escalado, rotación y localización que tenga el objeto por haber realizado la primera forma inicial de lo que representará, sino que lo hace como si ese estado fuese el inicial (0, 0, 0), y parte desde ahí. Así se evitan errores en el futuro al rotar o escalar partes de la malla. También se evitan problemas que pueden surgir en el modo escultura en los que, si se tienen valores en la escala, a veces, los pinceles se escalan con esos valores, y quedan ovalados en vez de redondos como forma por defecto. Esto puede provocar comportamientos aleatorios al usarlos en la malla, por no tener el total control sobre la forma del pincel. Por tanto, esto se realiza en el *Object Mode* seleccionando los objetos que se quiere resetear su escala (de uno en uno o todos a la vez), en este caso cada uno de los que forma parte del cuerpo de la gata. Antes, si se quiere **poner el pivote en el centro del objeto**, se coloca con *Object > Set Origin > Origin to Geometry*, para que los nuevos cálculos que se realicen al objeto se hagan desde ese punto. Luego se **resetean las transformaciones** desde la opción *Object > Apply > All Transforms to Deltas*. Con esto ya se tiene el *blocking* listo y los objetos preparados para el siguiente paso.

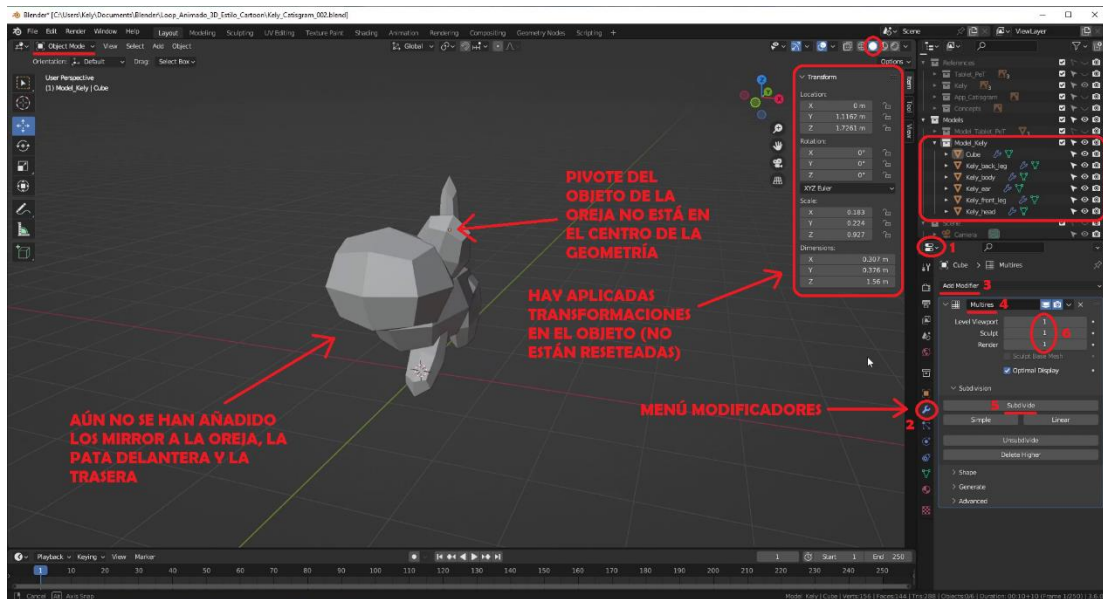


Figura 104. Blocking del modelo de la gata sin mirror
(Fuente propia: realizado en [Blender](#))

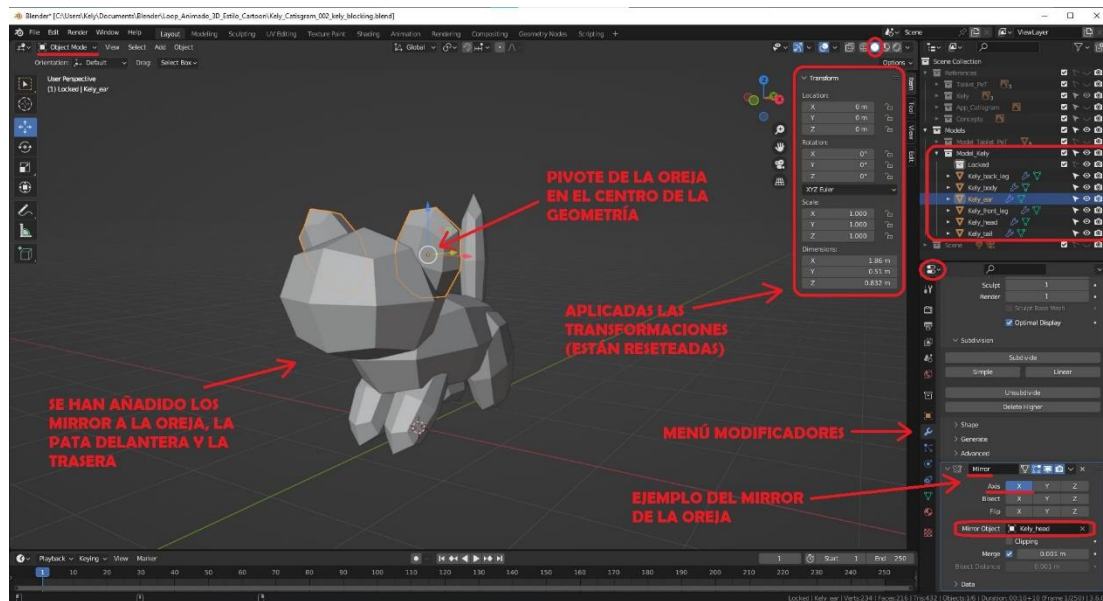


Figura 105. Blocking del modelo de la gata completo
(Fuente propia: realizado en [Blender](#))

Una vez el *blocking* está hecho, y se ha preparado cada una de las partes del modelo. Se seleccionan parte por parte y en el **Sculpt Mode** se **esculpen con los pinceles** que hay como herramientas. Estos pinceles son muy parecidos en todos los programas de escultura, y no hay reglas exactas de cómo usarlos. Depende de lo que quieras hacer y la mejor forma se escoja para hacerlo. Para dibujar algún detalle de la malla se usa el pincel *Draw*. Si se quiere añadir más malla para trabajar, pues si es en el modelo general, con el modificador *Multiresolution* que se añadió anteriormente, se le da a subdividir tanto como se necesite, y si solo es en una parte del

modelo, se puede usar el pincel *Clay*. Si se necesita mover la malla para colocar una parte el pincel *Elastic Deform*. Si necesitas hacer algún borde más notable en la malla lo pellizas con el pincel *Pinch*. Si necesitas hacer un pliegue el *Crease*. Si necesitas aplanar el *Flatten*, para inflar una zona el *Inflate*, y para suavizar el pincel *Smooth*, entre otros, pero estos son los más usados. Cada uno de los pinceles **se pueden configurar** la forma del pincel, añadirles texturas, elegir el radio que ocupan (si se necesita que cubra una zona más amplia del modelo o más pequeña), la fuerza con la que se aplica el pincel, crearles tu propia forma del dibujo que harán al pintar con ellos, y se pueden usar en versión positiva o en negativa, es decir, si, por ejemplo, usas el pincel *Inflate* en positivo crearás un bulto que sale hacia fuera del modelo, si lo usas en negativo se creará un bulto hacia dentro. Se pueden **usar máscaras** para trabajar zonas sin estropear el resto de la malla sin querer. También se puede **trabajar en simetría** con los ejes X, Y, Z o en varios de ellos a la vez, para agilizar trabajo si el modelo lo permite. Es un proceso laborioso en el que es necesario tener paciencia y practicar mucho. Existen muchas otras formas de esculpir, se puede no usar el modificador *Multiresolution* y en su lugar utilizar una malla base con bastante densidad y usar las herramientas de *Dyntopo* y *Remesh*, si se necesitase más densidad en una zona, y luego ir haciendo la malla uniforme para poder seguir trabajando con un número de polígonos que la gráfica del ordenador sea capaz de manejar, por ejemplo. Pero se ha escogido este método porque así se puede tener todos los niveles de resolución e ir cambiándolos si no necesita más detalle, y porque es el modo más parecido a cómo se trabaja en el programa ZBrush especializado en este método de modelado y usado en la industria de forma profesional.

Además de las referencias que se realizaron y colocaron en el viewport, para esculpir, es importante **tener en cuenta el estudio anatómico** de los gatos que se realizó en la fase de preproducción. Esto ayudará a darle realismo a la anatomía del gato para que coja la apariencia de uno. Como en todo trabajo artístico, es necesario este tipo de estudios para que luego poder crear la ilusión óptica ante el ojo humano de que dé la sensación de ser un gato y el cerebro lo compare inconscientemente con uno. Además, como en este caso se trata de una gata conocida, ha sido posible fijarse directamente en fotografías de dicha gata. Como se explicó en la fase de preproducción, en concreto, de fotos de cuando era un bebé para fijarse en la forma de la gata para hacerla estilo cartoon.

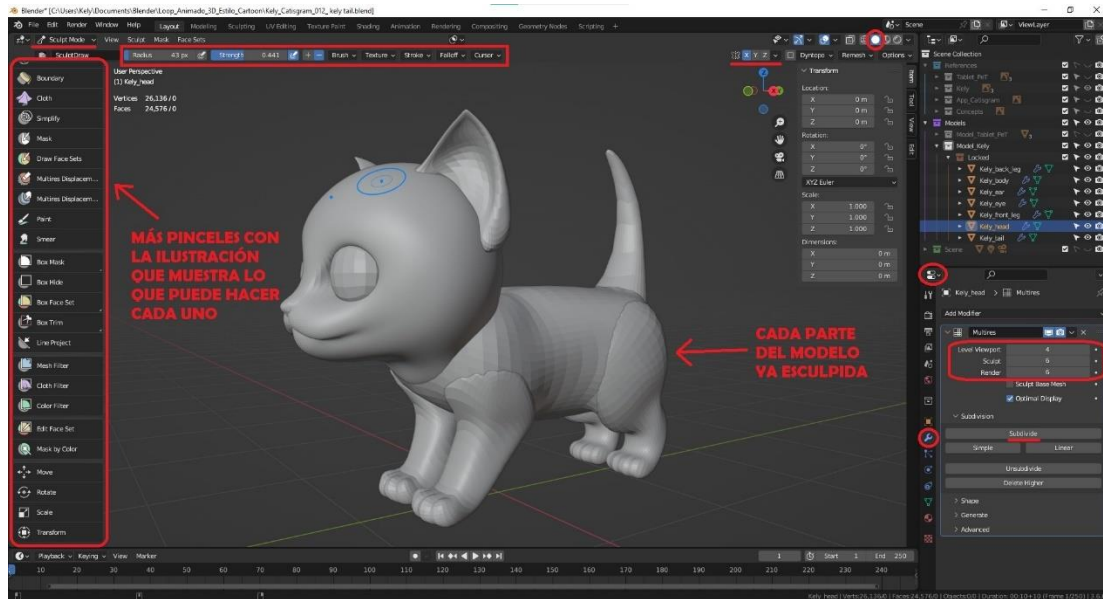


Figura 109. Resto de la lista de pinceles para el modo escultura y objeto ojo visible (Fuente propia: realizado en [Blender](#))

Después de tener cada una de las partes esculpidas, hay que unir las en una única malla. Para ello se comprueban los pivotes de nuevo, se vuelven a colocar cada uno en el centro de la geometría, y se comprueban las transformaciones, por si se ha aplicado alguna y hay que resetearla como se hizo antes de esculpir las. Luego, se seleccionan todas las partes, menos el ojo que irá separado, y se unen en un solo objeto con la opción *Object > Join*. Una vez unidas, se añade un modificador al modelo en el *Editor Type* la opción *Properties > Modifiers > Add Modifier > Remesh*, este modificador unifica los polígonos de la malla para que tengan la misma densidad en todas las partes del modelo, hace que en el interior se borren las partes sobrantes que había cuando eran partes separadas, y se puede elegir el tamaño de los polígonos, por si se quisiese hacer un *Remesh* para obtener una versión low poly. En este caso, se quiere coger todo el detalle de la malla de alta poligonación que se pueda, así que se pone un tamaño en el *Voxel Size* de 0'01 m y se aplica el modificador.

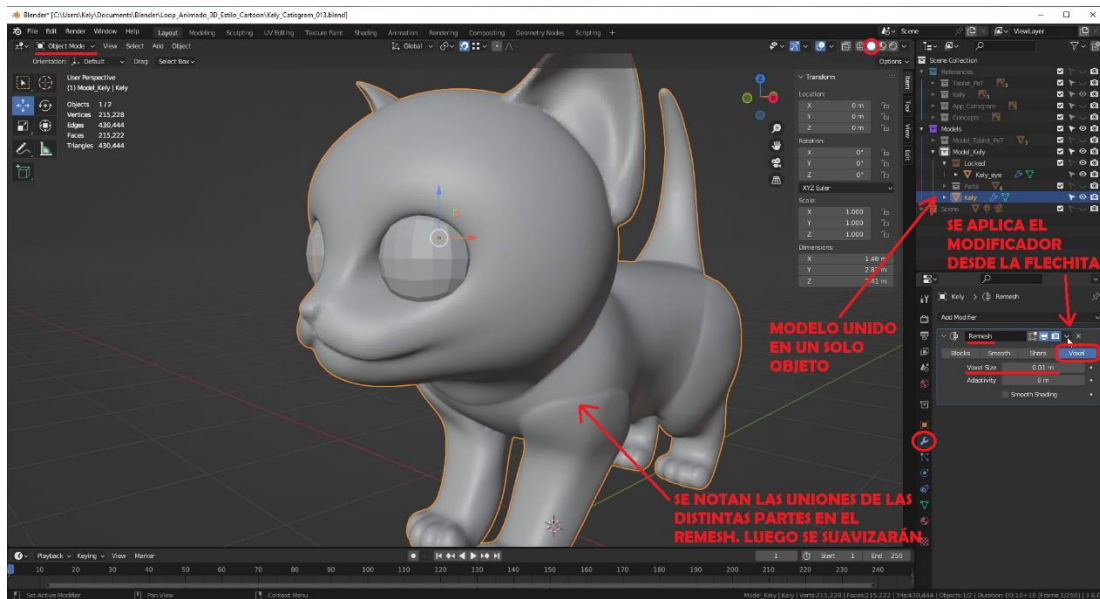


Figura 110. Unión de las partes del cuerpo de la gata y Remesh de la malla
(Fuente propia: realizado en [Blender](#))

Una vez hecho esto, se vuelve a **poner el pivote en el centro** de la geometría y **comprobar las transformaciones** por si hay que resetearlas, es importante porque se volverá a usar el pincel. En el *Sculpt Mode* se **suavizan las zonas de unión** de las partes que eran objetos distintos anteriormente con el pincel *Smooth*, para que no parezcan partes distintas y coja la apariencia de un objeto único y como se tiene que ver el gato entero. Se puede **recolocar y hacer más detalles** en la malla con los pinceles si es necesario, ahora que todo es un único objeto. Cuando se terminan los detalles, se obtiene el modelo del gato en su versión de alta calidad y poligonación, *high poly*. Se nombra como *Kely_high_poly*.



Figura 111. Modelo de la gata high poly suavizado y con los detalles
(Fuente propia: realizado en [Blender](#))

Ahora queda, por una parte, el modelo de los **ojos de la gata**. Primero, se explicará **cómo crear una esfera con topología correcta** en Blender, es decir, una esfera formada por *quads*. Para ello, se crea **un cubo** en el *Object Mode* con *Add > Mesh > Cube* para que esté formado por *quads*. A este cubo se le añade un modificador en el *Editor Type* la opción *Properties > Modifiers > Add Modifier > Subdivision Surface* para darle **más subdivisión a la malla** del cubo y que se redondee, pero como se puede comprobar no tiene la forma de una esfera, está ovalada. Para **convertirlo en una esfera**, en el *Edit Mode*, se seleccionan todas las caras del objeto y se usa la opción *Mesh > Transform > To Sphere* y se le da al número 1 del teclado numérico y al intro, para que lo haga a escala 1:1. Así se obtiene una esfera hecha mediante *quads*.

Ahora se verán dos figuras, en la primera se muestra los tipos de esferas de objetos primitivos en Blender la *UV Sphere* y la *Ico Sphere*, que no tienen una topología correcta con la que poder trabajar, ya que su malla contiene polígonos formados por 3 vértices (*tris*), y al lado un *Cube* con el que se consigue realizar una esfera formada por *quads*. La otra imagen muestra los pasos y el proceso de convertir ese cubo en una esfera.

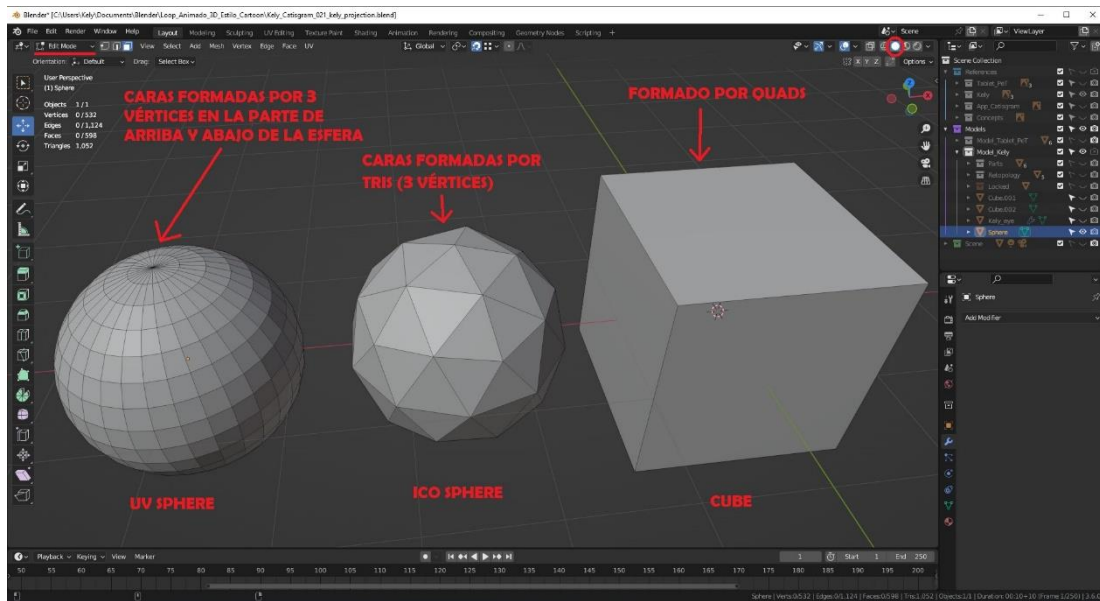


Figura 112. Comparación de la topología para crear una esfera correctamente (Fuente propia: realizado en [Blender](#))

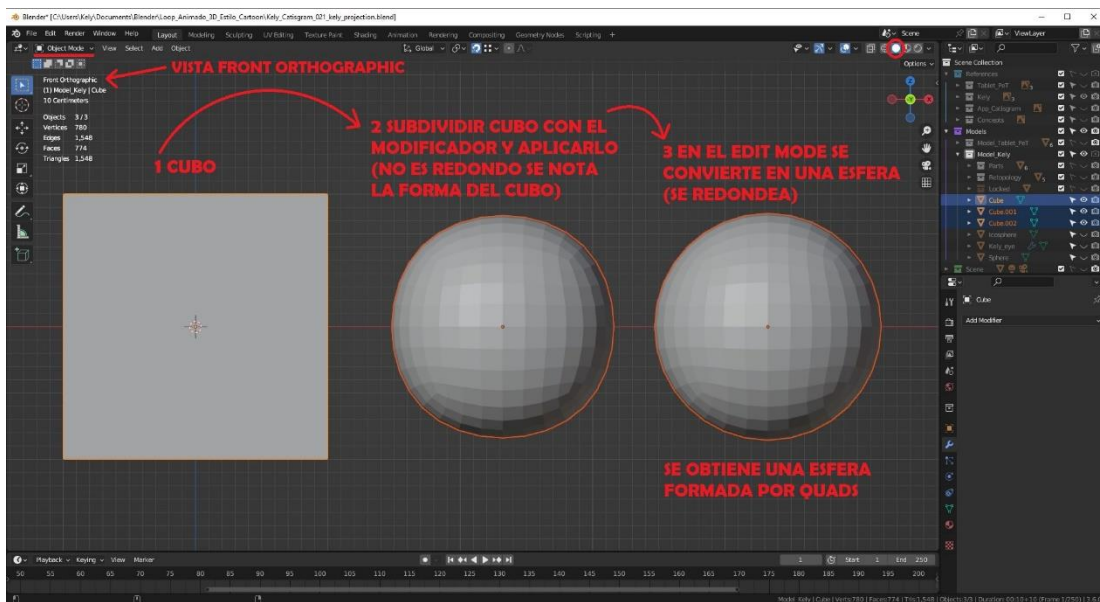


Figura 113. Pasos para convertir un cubo en una esfera que esté formada por quads (Fuente propia: realizado en [Blender](#))

A continuación, **se duplica** esta esfera que tiene la topología correcta para trabajar con ella. Se necesitan 2 de estas esferas para realizar el modelo de los ojos del gato. La primera será para crear la parte de la córnea, la parte cristalina transparente del ojo donde se crean los reflejos y brillos, y la otra esfera será el interior con el iris, la pupila, la esclerótica, el cristalino y demás partes del ojo. A la **segunda esfera**, la del iris, la pupila y el resto de las partes interiores, para que sea más realista el ojo, se le da **más subdivisión** con el modificador *Subdivision Surface*. Se **comprueba el pivote y transformaciones** para usar el *Sculpt Mode* con el pincel *Inflate* en negativo se hace un pequeño **huevo hacia dentro** de la malla con la **forma y tamaño de la pupila**.

Esto se hace así porque se ha estudiado la **anatomía de los ojos felinos**, y cuando miras de perfil a un gato se puede apreciar muy bien que la superficie del iris, la pupila y el cristalino se ve como a mitad de la parte transparente, y la parte transparente crea un efecto lupa de la pupila cuando ves los ojos del gato de frente. Por eso se usan dos esferas para crear este efecto en el ojo con los materiales más adelante.

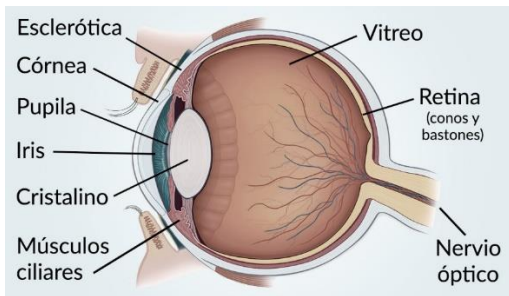


Figura 114. Estudio anatómico de un ojo
(Fuente: <https://www.optibejar.com/l/salud-visual-cristalino/>)

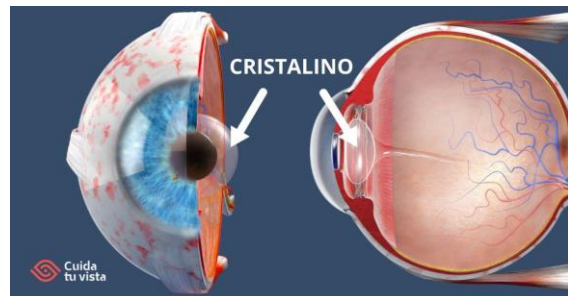


Figura 115. Partes de un ojo por capas
(Fuente: <https://cuidatuvista.com/cristalino-del-ojo-funcion/>)

A la **primera esfera**, la parte de la córnea que va por delante de la pupila, o bien, se le puede **dar una forma** en la parte de atrás para colocarla delante de la otra pero que la parte de la pupila esté dentro, o hacer que sea más grande y la otra esfera se pueda introducir entera dentro de esta. Pero como la segunda opción hace un poco menos realista el ojo, se ha decidido implementar la primera opción. Por último, **a cada esfera se le añade** un modificador *Mirror* desde el modelo del gato, para hacer el otro ojo por **simetría**, se nombran los objetos como *Kely_eye* y *Kely_eye_pupil*. Y ya se tiene los modelos de los ojos del gato.

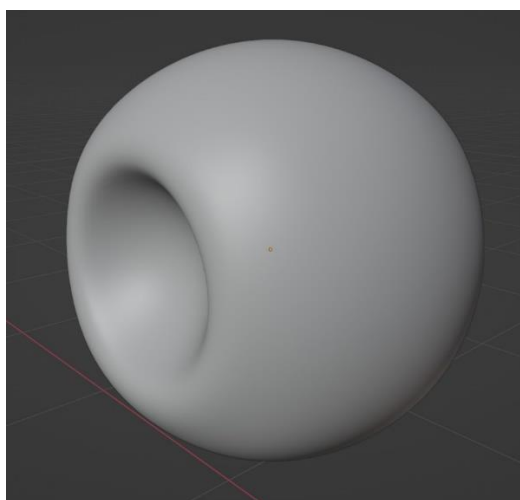


Figura 116. Esfera modelada con la parte del iris, pupila, esclerótica, etc
(Fuente propia: realizado en [Blender](#))

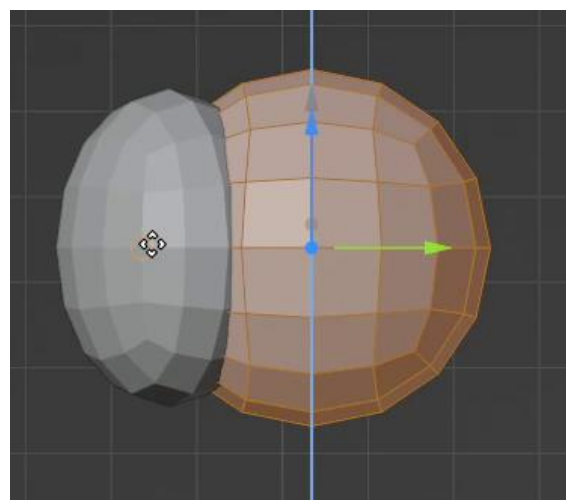


Figura 117. Colocación de los objetos del ojo y la forma que tienen (de perfil y low poly)
(Fuente propia: realizado en [Blender](#))

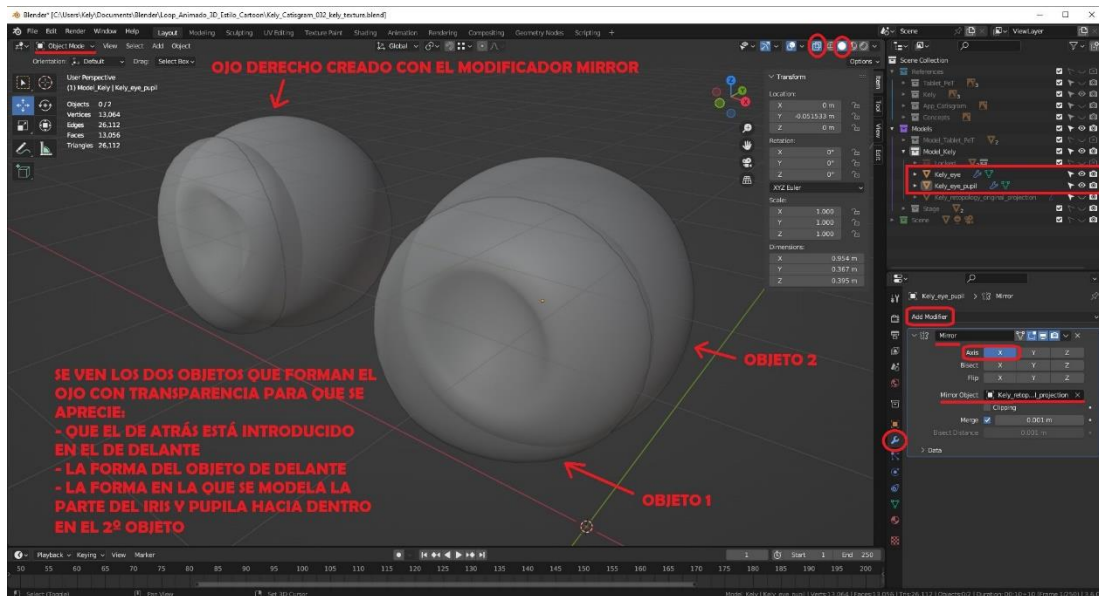


Figura 118. Modelo final de los ojos de la gata con transparencia
 (Fuente propia: realizado en [Blender](#))

Por otra parte, para terminar, los **bigotes y cejas de la gata**. Para realizarlos se ha usado el método de **modelado con Nurbs**. Para cada pelo se crea una curva de Bézier en el *Object Mode* con *Add > Curve > Bezier*. Se edita para darle forma en el *Edit Mode* con los puntos de ancla, de la misma forma que las herramientas vectoriales en un programa de dibujo, pero en un espacio con 3 dimensiones. Una vez ese pelo tenga la forma que se busca, se crea la siguiente curva para hacer lo mismo, así hasta obtener todos los pelos que se quieran añadir. Se añade un **grosor a cada uno de los pelos** en la configuración de sus parámetros en el *Editor Type* la opción *Properties > Data*. En el apartado *Shape* poner:

- En *Resolution Preview U* al menos 12 para que sean curvas y no se quedan como pelos tiesos y rectos, que no serían naturales.
- En *Curve Deform* activar *Radius* para indicar la forma de la curva.

En el apartado *Geometry*, poner:

- Poner un *Extrude* de 0'001 m, para darle el grosor al pelo, pero que sea un pelo finito.
- En *Bevel* marcar la opción *Round*, para que la extrusión tenga una forma redonda alrededor de la curva. Poner en *Depth* un 0'001 m para el radio que redondea la extrusión, y en *Resolution* un 4 para que la extrusión sea un cilindro más redondo al tener más aristas. Marcar la opción de *Fill Caps* para que los tubitos de los pelos tengan una tapa en los extremos del cilindro.



Figura 119. Configuración de las Nurbs de los bigotes y cejas
(Fuente propia: realizado en [Blender](#) y montado con [GIMP](#))

Luego se **transforman las curvas en una malla** seleccionándolas, con la opción *Object > Convert > Mesh*, para poder agregarles materiales más adelante. **Se unen en un único objeto** con la opción *Object > Join*. Por último, se añade un modificador *Mirror* desde el objeto del cuerpo del gato para que se creen **con simetría** los bigotes y cejas de la otra parte de la cabeza. **Se coloca el pivote en medio de la geometría y se resetean las transformaciones. Se nombra** el objeto como *L_moustache* (L de left, ya que es la parte de la cabeza en la que se hicieron los pelos, la otra se hizo con el *Mirror*). Finalmente, se obtiene el modelo del gato de alta calidad formado por cuatro objetos.

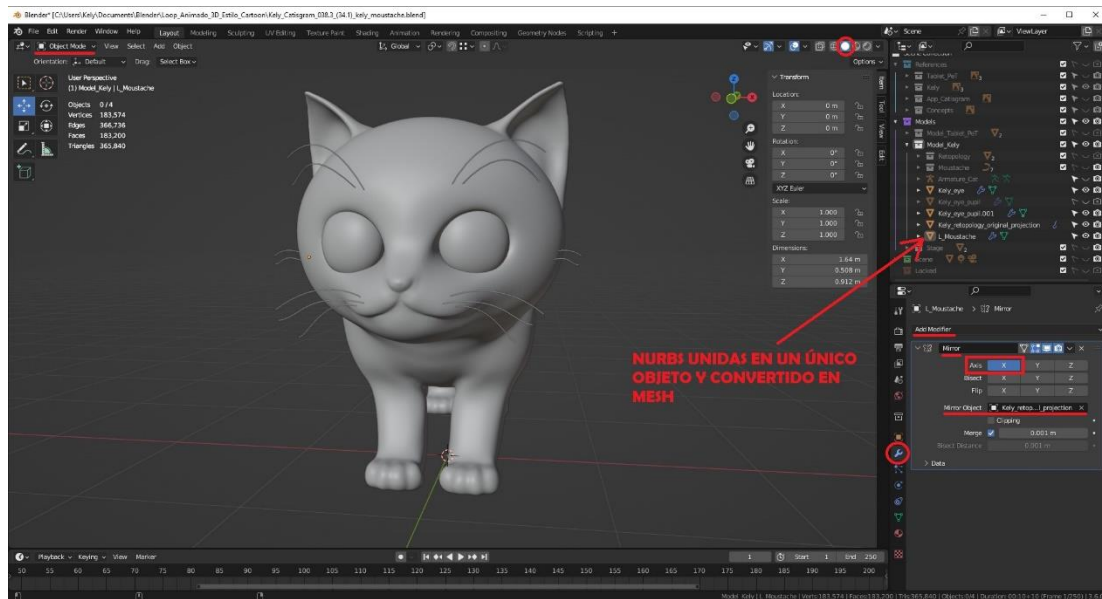


Figura 120. Modelo high poly final de la gata con el *Mirror* de los bigotes y cejas
(Fuente propia: realizado en [Blender](#))

8.1.4. Modelo del escenario

Para realizar el escenario, se ha usado el método de **primitivas básicas**. Se crean **dos planos** en el Object Mode con *Add > Mesh > Plane*, uno en posición vertical **para la pared** del fondo y otro en horizontal **para el suelo**. **Se colocan y se escalan** en los ejes X y Z para que se salgan del plano de la cámara cuando se cree y coloque. Es importante que los planos creen una intersección entre ellos y quede en la parte que no se verá del escenario un poco de malla, para evitar que en el render se pueda crear un hueco vacío donde se conectan, y parezca que no están juntos el suelo y la pared. **Se centran los pivotes en las geometrías y se resetean las transformaciones**. Por último, **se nombran P_{wall} y P_{floor}** , P de plano.

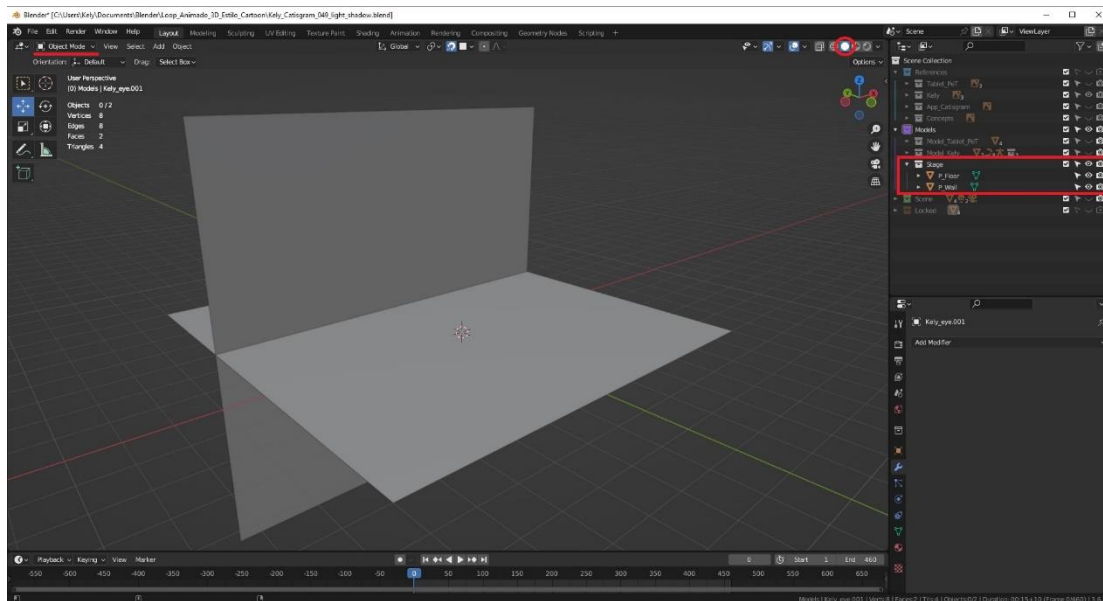


Figura 121. Modelo del escenario
(Fuente propia: realizado en [Blender](#))

8.2. Retopología

Para más tarde poder texturizar y animar el modelo de la gata correctamente, es muy importante **organizar la topología del modelo**, es decir, realizar la retopología. Se deben **colocar los loops de polígonos y**, en general, los **polígonos para que sigan la forma del modelo**. Se crean *loops* extras en las partes que se articulen y/o se puedan doblar, para que no de errores la malla al animar el personaje, al estirar y flexionar esas partes. Esto se hace solo con el modelo de la gata porque el resto de los objetos y modelos se han realizado con métodos que ya los dejaban con una topología correcta formada por quads. En el caso de los objetos orgánicos, como el de la gata, para que se parezca más a un gato, se hacen por escultura. Haber tenido en cuenta que la malla estuviese formada solo por quads, en su caso esto sirve para que ahora al hacer con la retopología la versión *low poly*, luego al proyectar los detalles de la malla *high poly* en la nueva topología no se generen errores en el proceso, y deje proyectar los detalles que se habían hecho en la escultura.

Para realizar la retopología del modelo de la gata, es necesario dejar el **modelo *high poly***, que se realizó anteriormente, **visible en el viewport** (se puede bloquear para no editarlo sin querer). Se configura la herramienta de encajado (*Snap*) de Blender que proporciona la capacidad de que, al mover, rotar o escalar los elementos de una malla o de un objeto se imanten a vértices, caras, aristas, a la rejilla del espacio 3D, o al centro de una arista, etc. que haya en la escena, o a varios de ellos, para poder trabajar la retopología. **Se configura el *Snap*** para que se imante a

Vertex y Face. **Se crea un plano** (porque es un *quad*) en el *Object Mode* con *Add > Mesh > Plane*, en el *Edit Mode* y **con el *Snap* activado**, se **coloca** el primer polígono sobre el modelo *high poly*. Se configura el objeto del plano para que sea siempre **visible por delante del modelo *high poly***, en *Editor Type* la opción *Properties > Object > Viewport Display > In front*.

Para agilizar la tarea y como este modelo al ser cartoon lo permite, se **añade un modificador de simetría al plano** en el *Editor Type* la opción *Properties > Modifiers > Add Modifier > Mirror* para solo tener que hacer la retopología de la mitad del cuerpo. Además, así se puede cumplir un requisito importante para los modelos que vayan a ser animados, que es que tengan una simetría y haya un *loop* de aristas que divida el modelo por el centro. Con ese plano, en el *Edit Mode* se usa la herramienta *Poly build* para ir **construyendo el resto de los polígonos**, colocando los vértices con el *Snap* activado sobre la malla *high poly* para posicionarlos. Es importante **activar la opción *Merge*** al usar *Poly Build* para que, al ir creando los polígonos, los vértices que ocupen la misma posición se unan en uno solo y no se creen vértices de más que hagan huecos en la malla.

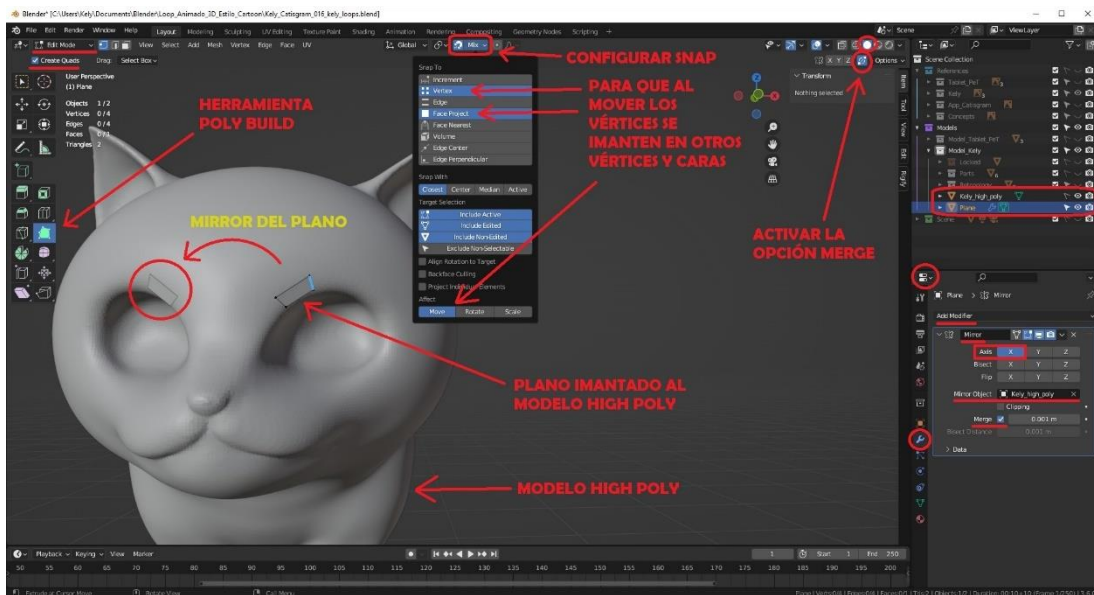


Figura 122. Preparar la herramienta *Poly build* para hacer la retopología (Fuente propia: realizado en [Blender](#))

Para facilitar la tarea hay que **organizar el proceso** de la retopología. Para ello **primero se crean los *loops*** en las zonas que se flexionan del modelo y pensando en las zonas en las que se tendrán que crear las costuras en la fase de texturizado, y, por tanto, se necesita que tengan una serie de aristas que las envuelva (un *loop* de aristas). Estos *loops* es importante contar y hacer que el **número de polígonos** que los forman **sea par**, para luego poder rellenar con *quads* el resto del modelo. Ya que, si no se hiciese así, más tarde no se podría cerrar la malla con un *quad* por no

haber calculado bien y se tendría que volver a empezar toda la zona del *loop*. Este tipo de errores no se podrían intentar arreglar porque ocasionan errores en cadena en toda la malla si los intentas arreglar más tarde, así que hay que tenerlo en cuenta desde el principio. A continuación, se ve una figura con todos los *loops* de polígonos hechos en las zonas en las que se necesitan. Como cada *loop* tiene el *Mirror* y la opción *In front* activada, se ven los *loops* de las dos partes del cuerpo como si estuviesen por delante del modelo *high poly*.

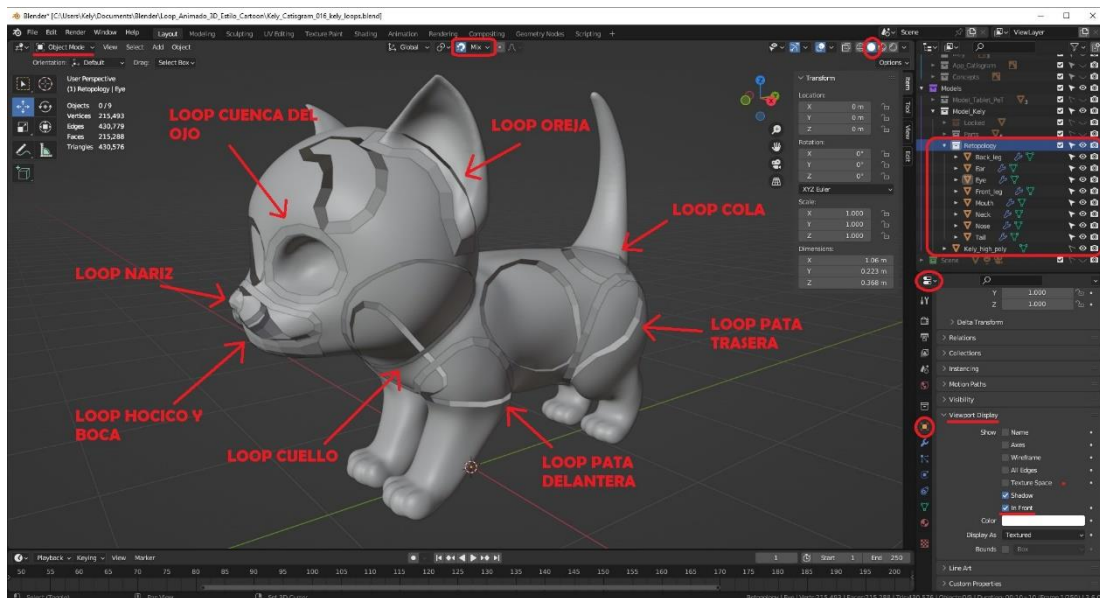


Figura 123. Preparación de la retopología con los loops de polígonos
(Fuente propia: realizado en [Blender](#))

Una vez se tienen los *loops* de cada zona hechos, se va rellenando cada parte del cuerpo con polígonos que siguen las formas del cuerpo y teniendo en cuenta que los vértices del centro (los que dividen el modelo en dos) estén alineados en este caso en el eje X en la coordenada 0. Este es un proceso que lleva mucho tiempo y es delicado, pero es importante para que se pueda texturizar y se vea bien el modelo al animarlo. Las siguientes figuras muestran el modelo con la retopología hecha y para que se pueda apreciar que la otra mitad del modelo se hace con el modificador *Mirror*.

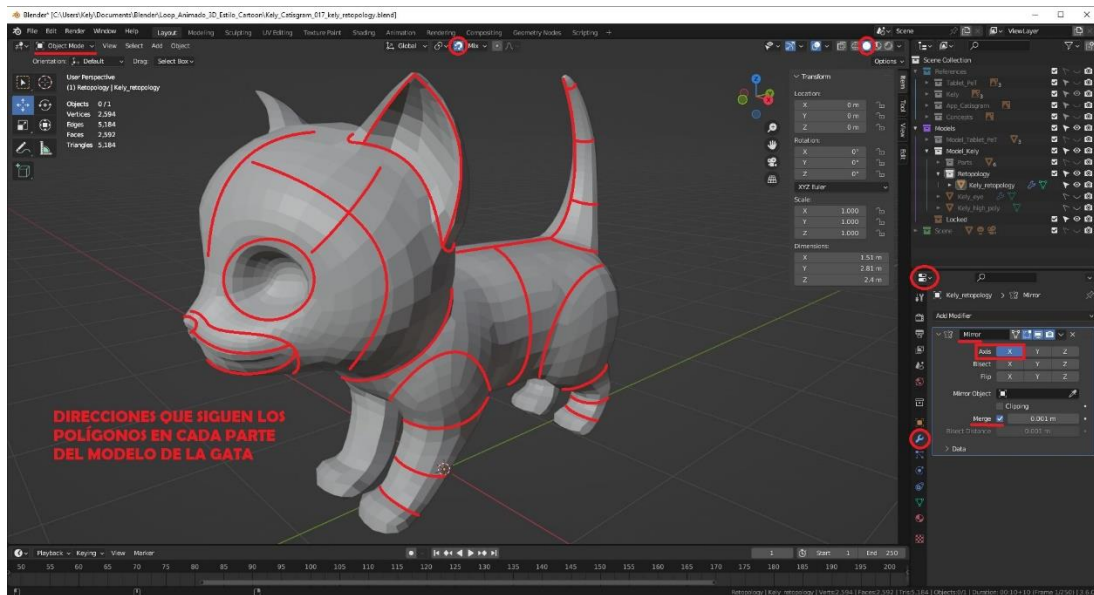


Figura 124. Retopología completa y dirección de los polígonos
(Fuente propia: realizado en [Blender](#))

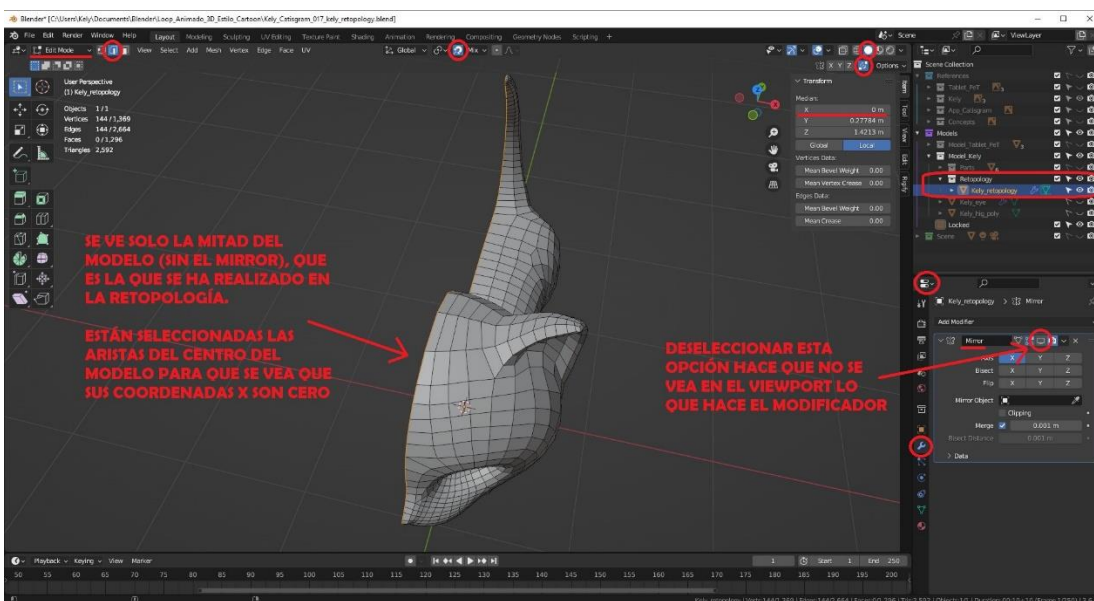


Figura 125. Retopología completa sin el Mirror
(Fuente propia: realizado en [Blender](#))

Como se puede ver, se ha conseguido una versión **low poly** con la **topología correcta** del modelo de la gata, pero se han perdido los pequeños detalles que se hicieron al esculpir. Para poder volver a **conseguir esos detalles**, se necesita **aplicar el Mirror** y **añadir el modificador Multiresolution** en el Editor Type la opción **Properties > Modifiers > Add Modifier > Multiresolution**. Se **subdivide la malla** para que tenga una resolución igual aproximadamente al modelo **high poly**. Se selecciona en el **Object Mode** el objeto **high poly** y luego se selecciona también el modelo **low poly** (sin deseleccionar el anterior) en el modificador **Multiresolution** que se ha añadido. En la subdivisión que cumpla con el requisito de ser igual a la del modelo **high**

poly, se le da a la opción que está en el apartado *Shape > Reshape*, que **proyectará la malla *high poly* sobre la *low poly***. Esto hace que en esa subdivisión se marquen los detalles de la malla *high poly* en la nueva malla. Así no se pierden los detalles del modelo y a la vez se tiene una malla con una correcta topología. Luego **se comprueba que el pivote** esté en el centro de la geometría y se **resetean las transformaciones**. Por último, se muestran dos figuras. En una se puede ver como se hace la proyección y la comparación entre: la malla *high poly*, la malla *low poly* de topología correcta, y con la malla de topología correcta con la proyección de los detalles hecha. La segunda figura muestra la misma comparación, pero mostrando la densidad de la malla con la estructura alámbrica de los modelos (*Wireframe* activado). Como se puede ver, en la malla de topología correcta con la proyección hecha, con el modificador *Multiresolution* se consigue que, aunque se vean todos los detalles de la malla *high poly*, la malla en realidad solo está formada por la misma cantidad de vértices que la *low poly*. Por tanto, no cuesta tanto mover los vértices del modelo cuando se manipula en el programa, porque son pocos vértices y no tiene que hacer demasiados cálculos que podrían hacer colapsar el programa. Por último, **se nombra el objeto** que tiene la topología correcta y la proyección en el *Outliner* como *Kely_retopology_original_projection*.

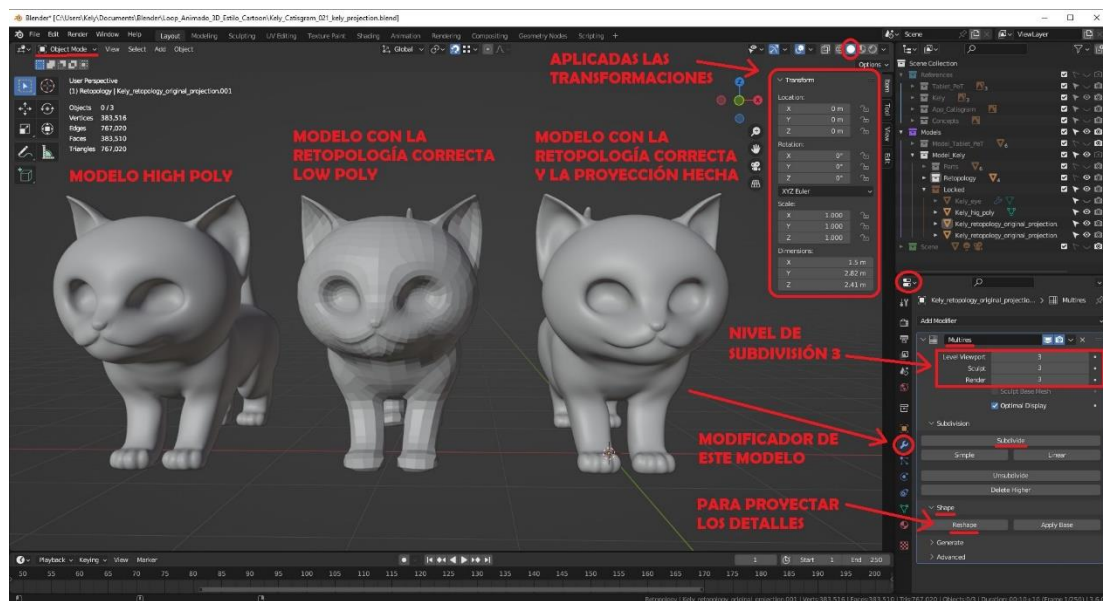


Figura 126. Comparación de los modelos *high poly*, *low poly* y con la proyección (Fuente propia: realizado en [Blender](#))

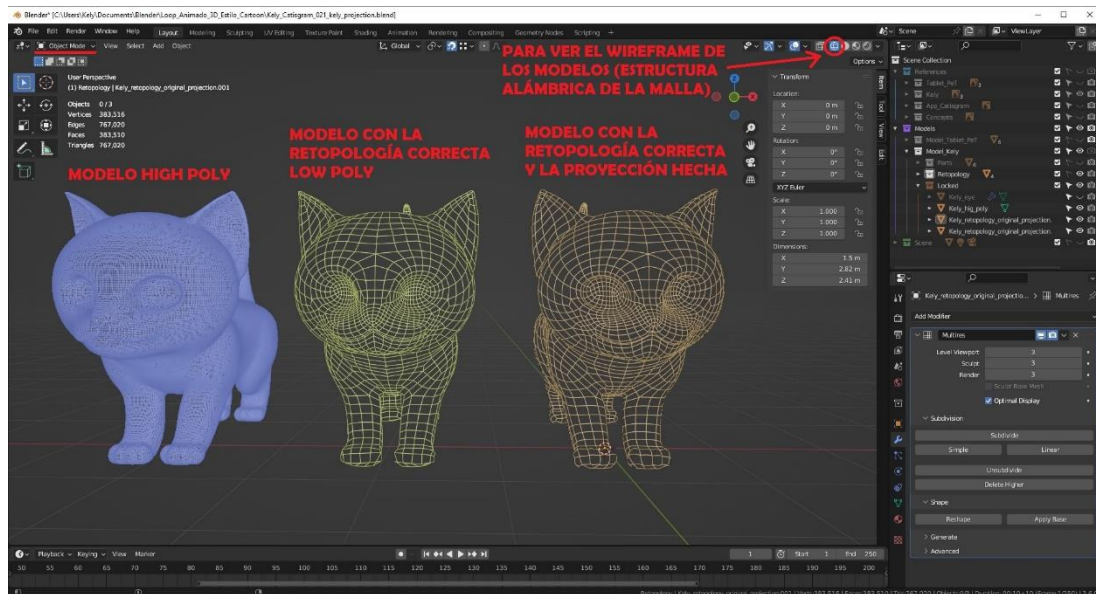


Figura 127. Comparación en modo alámbrico de la topología (Fuente propia: realizado en [Blender](#))

Este modelo con la topología correcta y la proyección será el que se texturice y se utilice en la animación final.

8.3. Materiales y Texturizado

Este proceso se puede empezar una vez se tenga la **topología** de la malla de los objetos **correcta**.

8.3.1. Modelo de la tableta

Para la **tableta** se utilizaron **materiales** y la configuración de sus propiedades mediante nodos. Para ello se usó el **espacio de trabajo Shading**. Entre las opciones del *Editor Type: Properties > Material* y el *Shader Editor* se configuraron y crearon los distintos materiales.

Para la **aplicación Catisgram** se prepararon 7 **texturas** de la app en el programa de dibujo **Krita**, para poder más adelante en la animación ir intercambiándolas y crear la animación de la pantalla de la tableta. Las texturas que se crearon, 4 son los frames de la animación de la app (imágenes de la fila de arriba de la figura), y 3 son los frames de la animación de palpar el corazón en la pantalla cuando la gata le dé a *Me gusta* (imágenes de la fila de debajo de la figura). Estas son las texturas que se crearon y que luego se irán intercambiando en la pantalla del dispositivo:

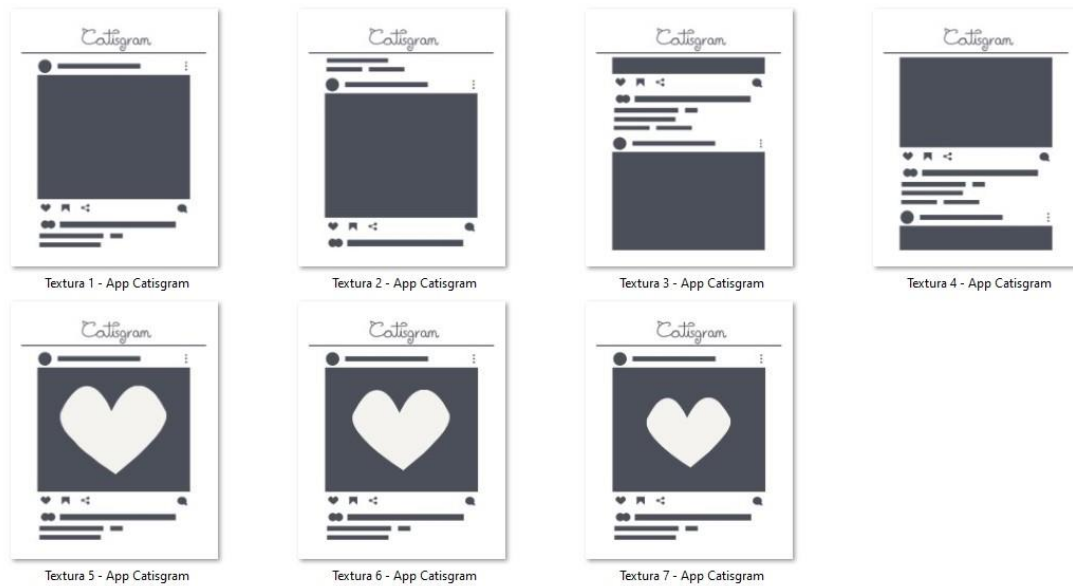


Figura 128. Texturas para la animación de la pantalla de la tableta
(Fuente propia: realizadas con [Krita](#))

En total para el modelo de la tableta se usaron 3 **materiales**:

- Para **el cristal** de la pantalla se creó un material al que se llamó *M_Tablet_glass*. Se usó una plantilla *Glass BSDF* de Blender.
- Para **la carcasa** se creó un material al que se llamó *M_Tablet_base*. Se usó una plantilla *Principled BSDF* de Blender. Se escogió el color negro del material según los *concepts* que se habían hecho en la fase de preproducción. El parámetro *Metallic* se aplicó entero, se puso a valor 1 (va de 0 a 1), para que pareciese metálico y tuviese reflejos y otras propiedades de los metales. El *Specular* se puso a mitad (0'5) para conseguir los brillos. El *Roughness* se puso bajito (0'3) para que fuese como un metal que parece plástico.
- Para **la app** que se ve en el móvil se creó un material al que se llamó *M_Tablet_screen*. Se usó una plantilla *Principled BSDF* de Blender a la que se le añadió un nodo de textura. Desde ese mismo nodo se configura que imagen de la app se ve en cada momento. Para que se viese la textura (imagen que se había realizado en Krita) se conectó el canal del *Color* y el canal *Alpha* de la textura con el del material. Además, para que la pantalla produjese luz se añadió un nodo *Emission* y se subió el valor a 2. Para unir el nodo *Emission* y el *Principled BSDF* se usó un nodo *Add Shader* que se conectó con el nodo de salida para que se viesen los dos efectos aplicados en ese mismo material.

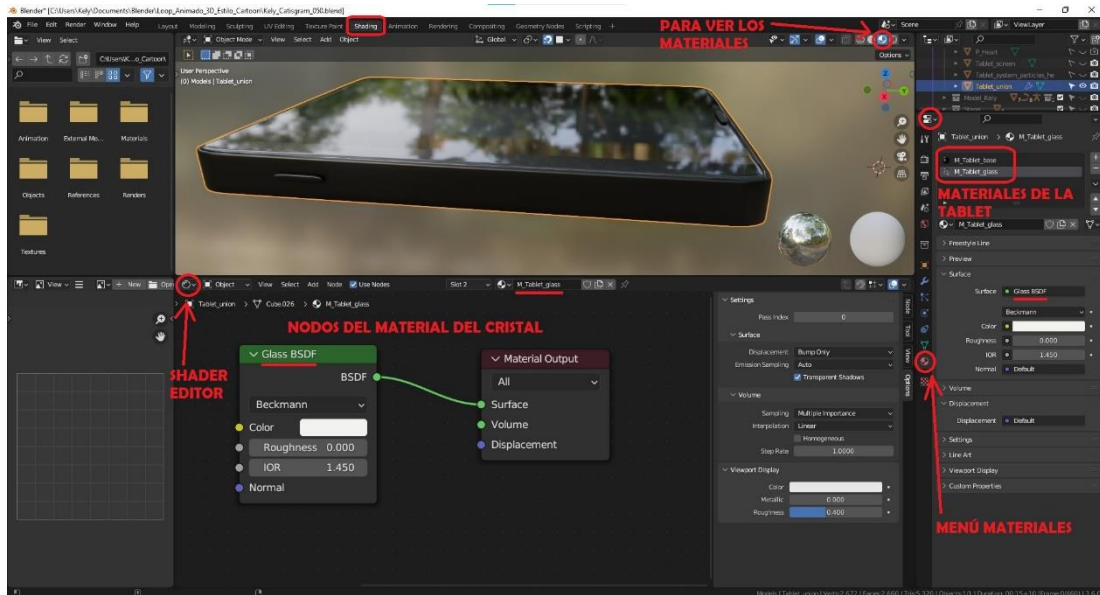


Figura 129. Configuración material cristal de la tableta
(Fuente propia: realizado en [Blender](#))

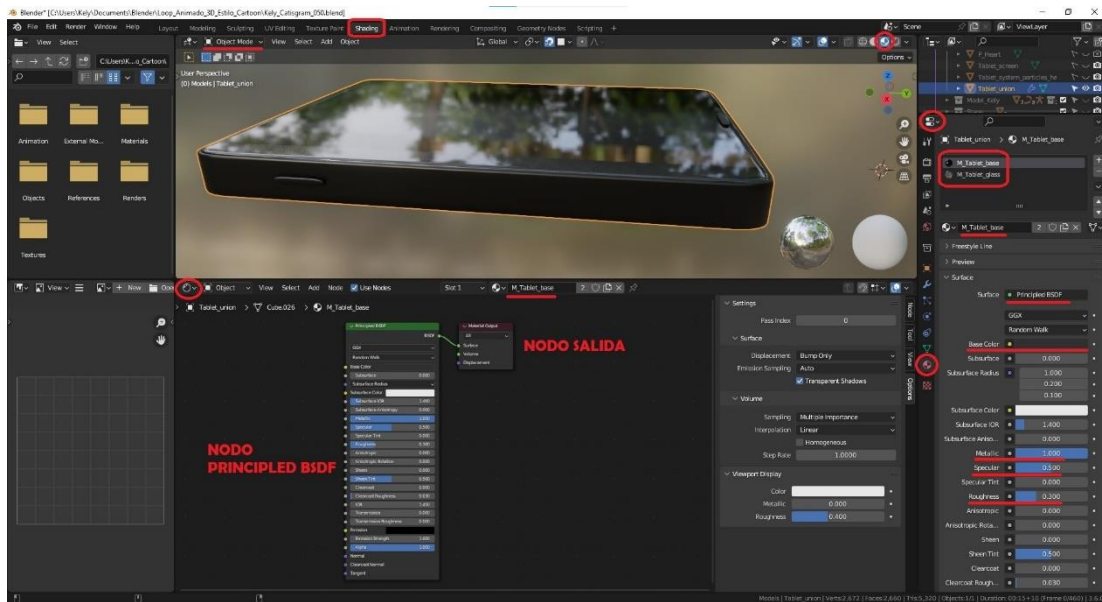


Figura 130. Configuración material carcasa de la tableta
(Fuente propia: realizado en [Blender](#))

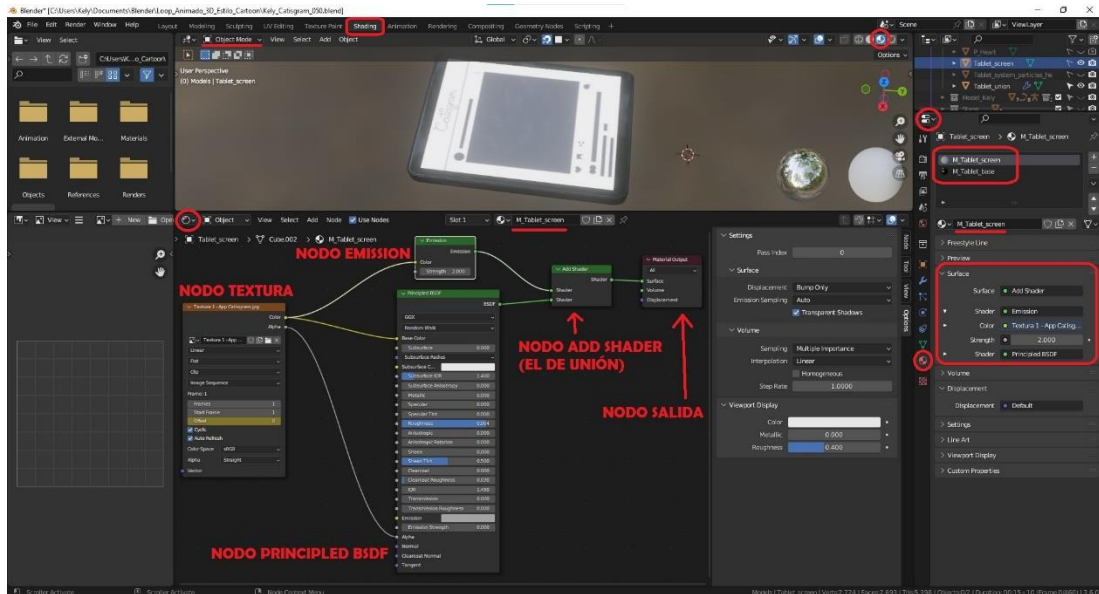


Figura 131. Configuración material App Catisgram de la tableta
(Fuente propia: realizado en [Blender](#))

8.3.2. Modelo de la gata Kely

Para el modelo de **la gata**, los objetos de **los ojos y bigotes** usaron **materiales**, y el modelo del **cuerpo** se **texturizó** en el propio programa Blender.

Para las partes del modelo que usan **materiales**. Se usó el **espacio de trabajo Shading**. Entre las opciones del *Editor Type: Properties > Material* y el *Shader Editor* se configuraron y crearon los distintos materiales.

En total para el modelo de la gata se usaron **4 materiales**:

- Para **los bigotes** se creó un material al que se llamó *M_Moustache*. Se usó una plantilla *Principled BSDF*. Se escogió el color gris claro del material según los *concepts* que se habían hecho en la fase de preproducción. El parámetro *Metallic* se puso a 0 para que no fuese nada metálico. El *Specular* se puso bajito (0'3) para conseguir que hubiese algún brillo, pero no demasiado fuerte. El *Roughness* se puso alto (0'8) para que se comportase como un pelo, que tiene una textura rugosa.
- Para **la córnea** se creó un material al que se llamó *M_Kely_eye_glass*. Se usó una plantilla *Principled BSDF* en vez de la de cristal para poder configurarlo a gusto personal. El parámetro *Metallic* se puso a 0 para que no fuese nada metálico. El *Specular* se puso bajito (0'163) para conseguir que hubiese algún brillo flojito. El *Roughness* se puso bajo

(0'250) para que se comportase como una pantalla algo sucia del uso y por eso tiene una textura rugosa. Por último, el parámetro *Transmission* se aplicó entero, se puso a valor 1 (va de 0 a 1), para crear un cristal transparente en el que se reflejan el resto de los objetos de la escena.

- Para **el iris** se creó un material al que se llamó *M_Kely_eye*. Se usó una plantilla *Principled BSDF*. Se escogió el color caramelo para el material según los *concepts* que se habían hecho. El parámetro *Metallic* se puso a 0 para que no fuese nada metálico. El *Specular* se puso a la mitad (0'5) para conseguir que se formen brillos con la luz, pero no demasiados. El *Roughness* también se puso a la mitad (0'5) para mantener un equilibrio entre los brillos que se pudiesen generar cuando le dé la luz y la textura rugosa del iris. Por último, se escogió un color para el parámetro *Emission* un poco más oscuro que el color caramelo que se había escogido de base. Se subió el parámetro un poco (0'2) *Emission Strenght* para que con la oscuridad de la escena los ojos emitan una luz tenue y muy sutil de ese color. Así simular como se les ven los ojos brillantes a los gatos en la oscuridad.
- Para **la pupila** se creó un material al que se llamó *M_Kely_eye_pupil*. Se usó una plantilla *Principled BSDF*. Se escogió el color gris oscuro (casi negro) para el material según los *concepts* que se habían hecho. Los parámetros *Metallic*, *Specular* y *Roughness* se pusieron igual que en el iris, ya que la pupila está en el centro del iris y tiene esas mismas propiedades. No se utilizó en este caso los parámetros *Emission* y *Emission Strenght* porque cuando ves los ojos de un gato en la oscuridad lo que les brilla es el iris y por eso se ve el color de este.

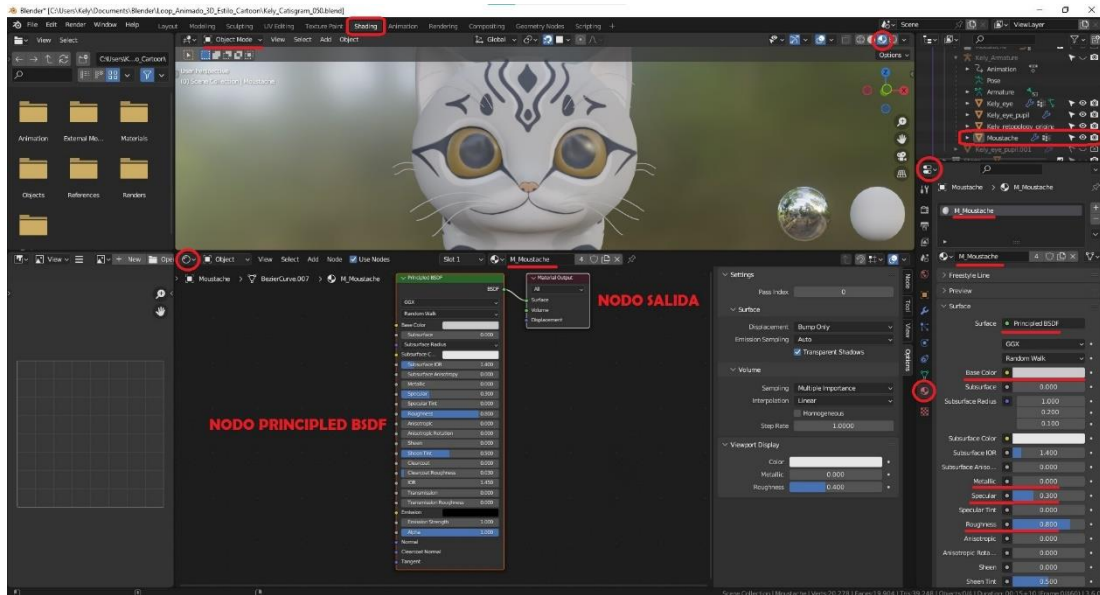


Figura 132. Configuración material de los bigotes de la gata
(Fuente propia: realizado en [Blender](#))

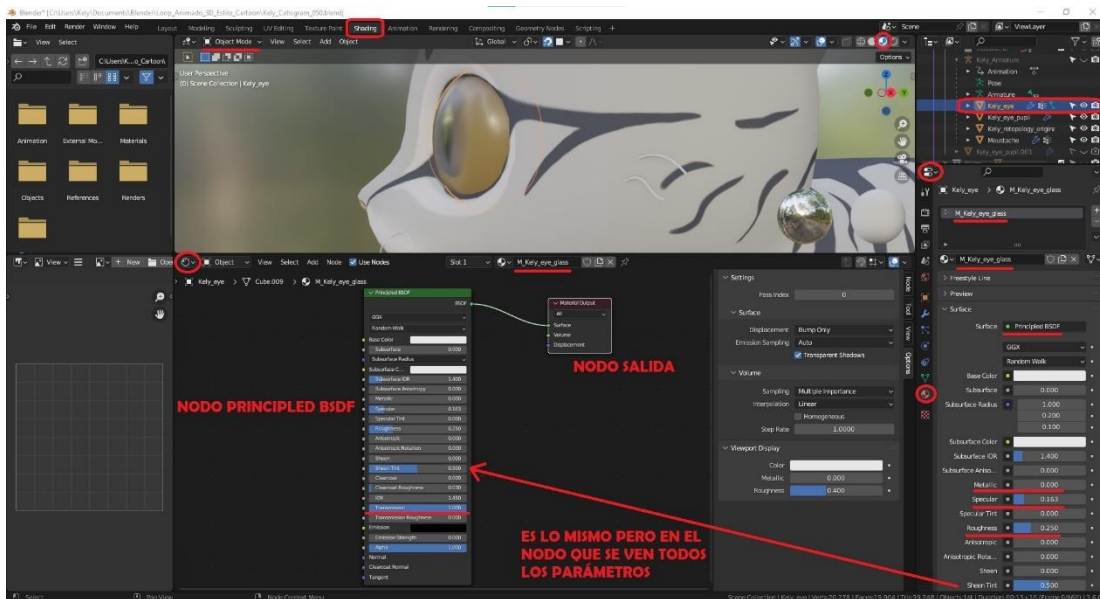


Figura 133. Configuración material de la córnea del ojo de la gata
(Fuente propia: realizado en [Blender](#))

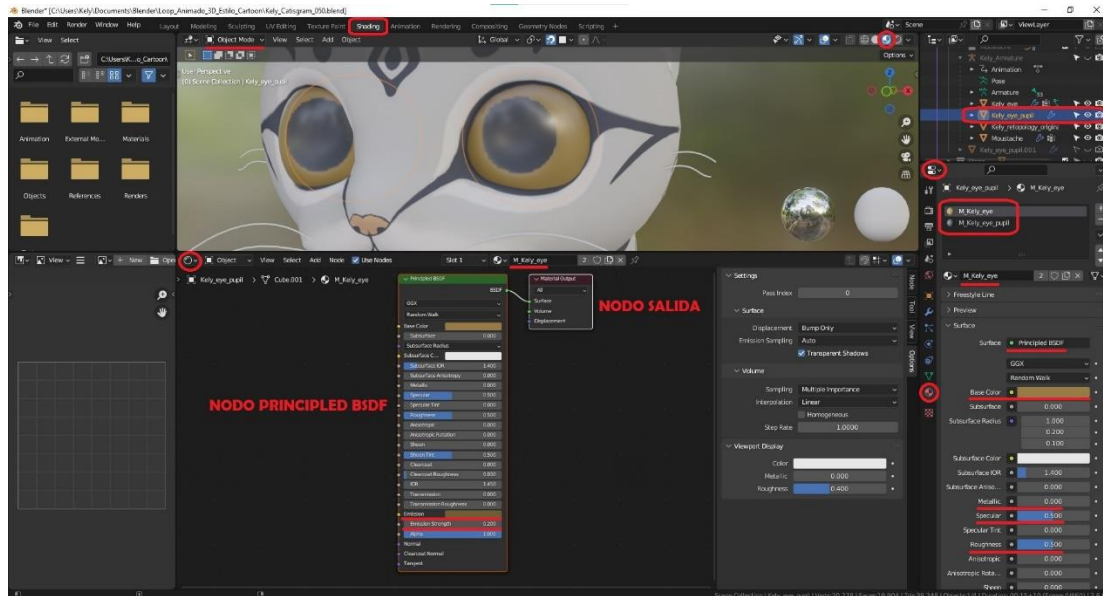


Figura 134. Configuración material del iris del ojo de la gata
(Fuente propia: realizado en [Blender](#))

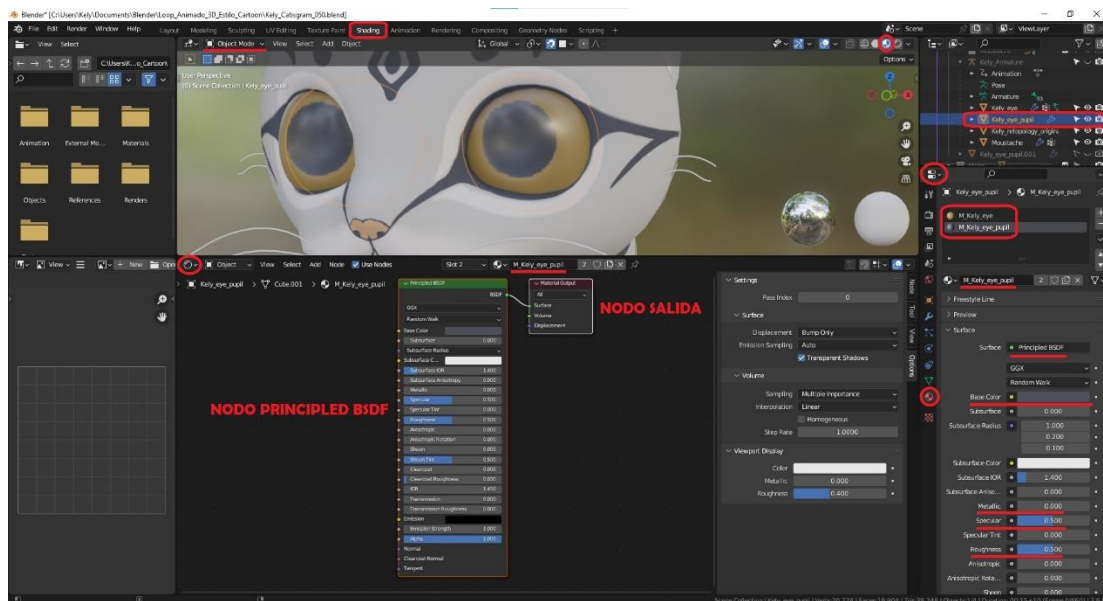


Figura 135. Configuración material de la pupila del ojo de la gata
(Fuente propia: realizado en [Blender](#))

Para la parte del modelo que se **texturizó**. Se usaron dos **espacios de trabajo**. Por una parte, el espacio de trabajo *UV Editing* para crear las costuras en el modelo, y, por otra parte, el espacio de trabajo *Texture Paint* para pintar sobre el modelo.

Primero había que **marcar las costuras** en el modelo para poder pintarlo. Estas costuras se marcan en las zonas por las que se abrirá el modelo para que se represente en 2D. Para ello en el espacio *UV Editing*, en el *Edit Mode* se seleccionaban las aristas que se iban a marcar en el

menú vas a la opción *Edge > Mark Seam*. Una vez se tenían todas las costuras marcadas, se seleccionaban todas las aristas del modelo y se usaba la opción *UV > Unwrap* que **desplegaba todas las partes del modelo** que se habían hecho las costuras en el panel de la izquierda en 2D. Como se explicó en el estado del arte, cada objeto en este panel es una isla.

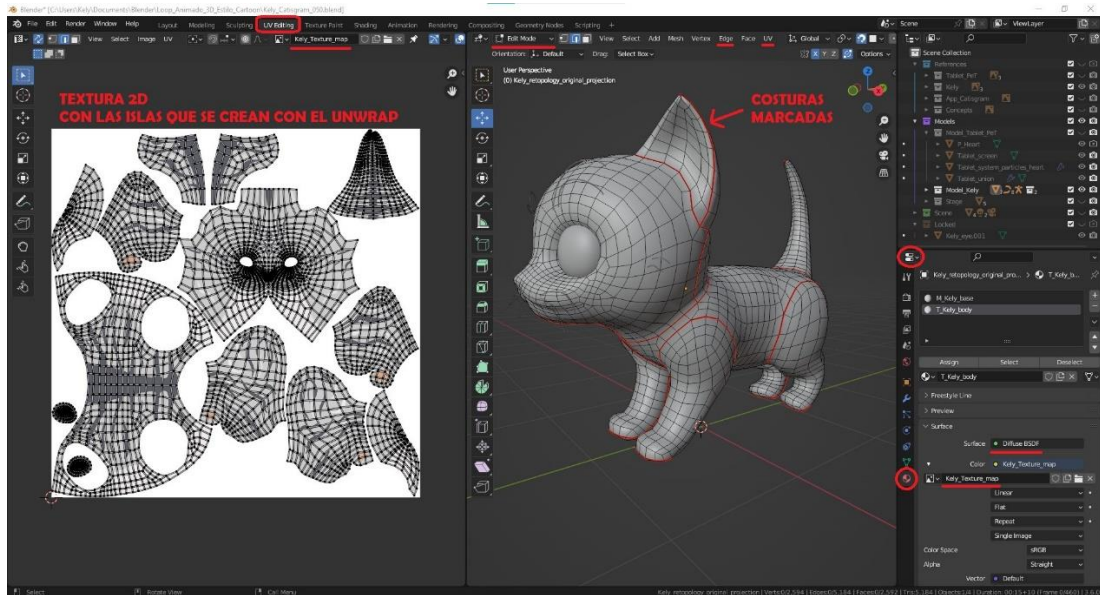


Figura 136. Modelo de la gata con las costuras marcadas y el Unwrap a la izquierda (Fuente propia: realizado en [Blender](#))

Luego se **escalan y colocan las islas** para que todas queden con una **resolución** más o menos uniforme. Para **comprobar** esto y que se han realizado las **costuras** en las aristas correctas, se crea una **textura de comprobación** del tipo *Checker UV* con las dimensiones que se realizará la textura que se vaya a pintar, en este caso se hizo a 4k (4096 x 4096). Esta textura se aplica al modelo en el **espacio de trabajo Shading** creando un nodo de textura que se conecta a la salida del material y eligiendo la imagen de la textura que se ha creado. Igual que se hizo con la textura de la app de la tableta, pero sin el nodo *Emission*. Después vuelves al **espacio de trabajo UV Editing** y compruebas como queda. Si es necesario se modifican las costuras marcadas, se vuelve a dar al *Unwrap* cada vez que se cambie una, y se colocan y escalan las islas de nuevo.

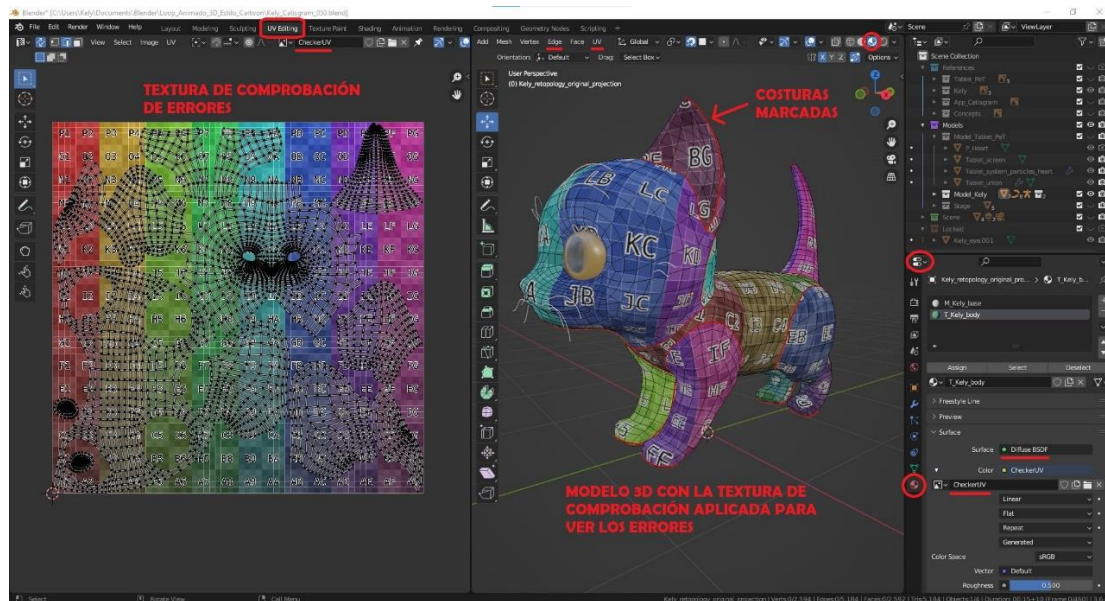


Figura 137. Comprobación de errores con la textura Checker UV
(Fuente propia: realizado en [Blender](#))

Finalmente, cuando ya se ha comprobado con la textura de comprobación que todo está bien, se **crea otra textura** de la misma resolución con un color base como el blanco. Se cambia en el material la textura de comprobación por esta nueva. Se pasa al **espacio de trabajo Texture Paint**, donde se **crea una paleta** con los colores que se usaron en el concept en la gata y se **pinta** sobre el modelo 3D y/o en la textura 2D cómo se quiera ver en el modelo. Los pinceles se pueden configurar la fuerza, el radio, la textura o la forma de la brocha como en un programa de pintura o dibujo digital. Se pueden usar las **herramientas**: pincel, desenfocar, emborronar, también de clonar, bote de pintura o enmascarar una zona igual que en esos programas. Es un proceso delicado en el que hay que invertir mucho tiempo hasta lograr los resultados esperados. En el espacio a la izquierda donde se ve la textura 2D, en el menú se puede **guardar** en una carpeta del ordenador la textura 2D pintada con la opción *Image > Save As*.

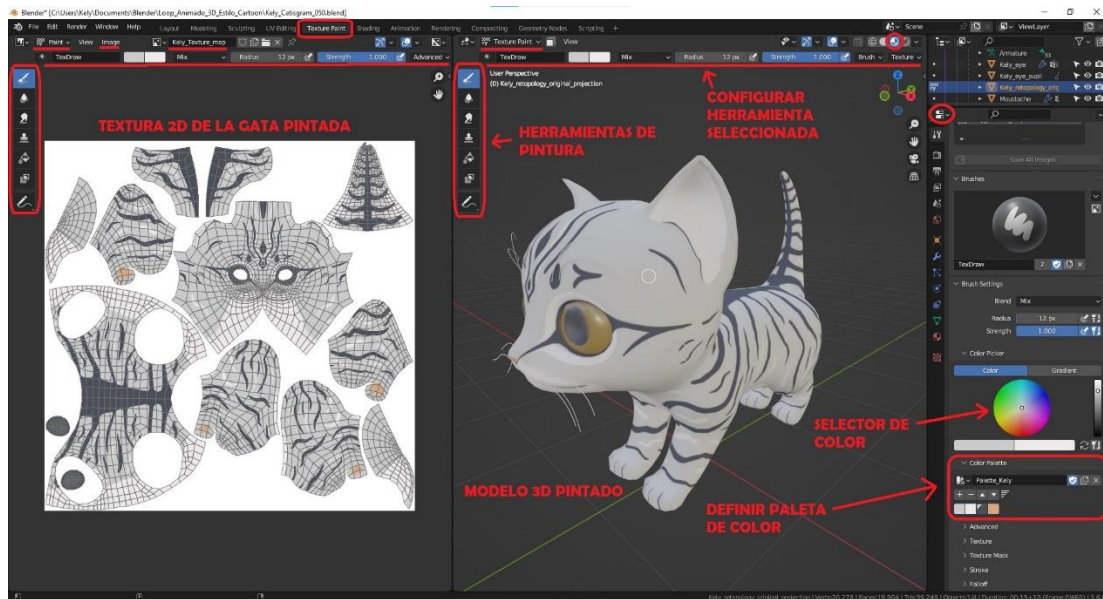


Figura 138. Espacio de trabajo Texture Paint con el modelo de la gata pintado
(Fuente propia: realizado en [Blender](#))

Con esto ya estarían todos los materiales y texturas del modelo de la gata.

8.3.3. Modelo del escenario

Para el **escenario** se creó y configuró 1 **material**. Se configuró las propiedades del material mediante nodos. Se usó el **espacio de trabajo Shading**. Entre las opciones del *Editor Type: Properties > Material* y el *Shader Editor* se configuró y creó este material que se aplicó al suelo y a la pared.

El material se nombró *M_Scenery*. Se usó una plantilla *Principled BSDF* de Blender. Se escogió el color lila del material según los concepts que se habían hecho en la fase de preproducción. El parámetro *Metallic* se puso a 0 para que no fuese nada metálico. El *Specular* se puso a un nivel bajo (0'1) suficiente para que cree algún brillo con la luz. El parámetro *Roughness* se puso alto (0'8) para simular que es una pared rugosa y por eso tampoco crea tantos brillos ni reflejos.

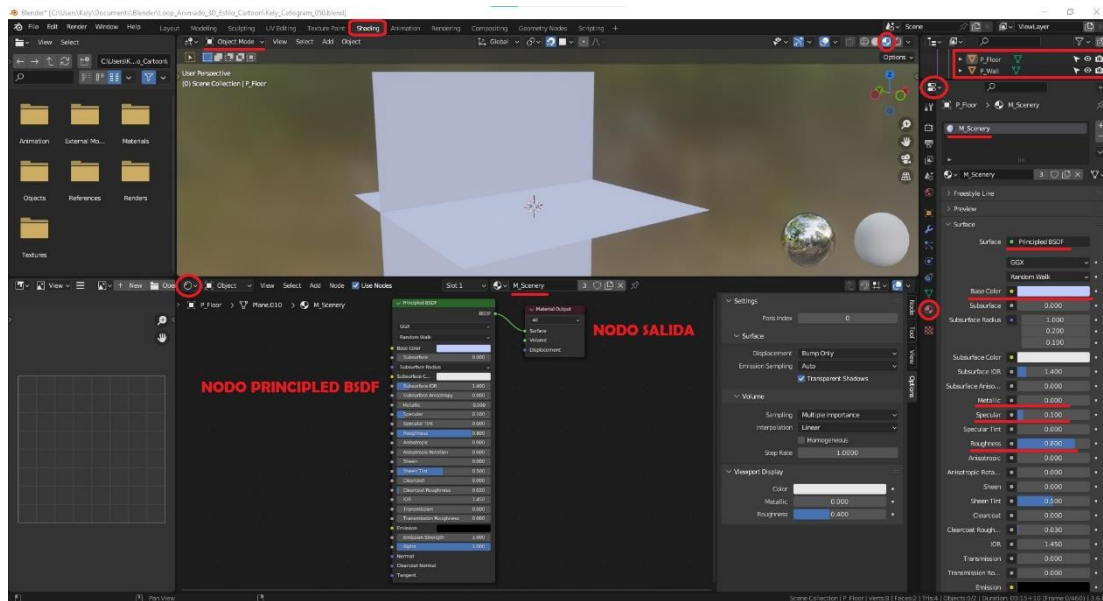


Figura 139. Configuración material del escenario
(Fuente propia: realizado en [Blender](#))

8.4. Rigging y Skinning

Este proceso se puede empezar una vez se tenga la **topología** de la malla de los objetos **correcta**. Para esta animación, como solo se moverá **la gata** este proceso solo se ha hecho con este modelo. Ya que la tableta permanecerá en el suelo y se animará solo la pantalla de la app, pero se explicará en el apartado de animación que hay más adelante cómo se hace. Es importante **comprobar** que **los pivotes** estén en el centro de sus geometrías y que se han aplicado (reseteado) las **transformaciones** a cada objeto.

El **rigging** se realiza **pensando en los movimientos** que, en este caso, pueda realizar un gato. Y también, en todo lo que se imagine que el animador pueda necesitar para animarlo. Incluso, aunque luego haya huesos que luego no utilice en esa animación. De esta forma el modelo queda **preparado para otras animaciones** en las que pueda intervenir en el futuro. Así que, como se vio en el capítulo de *Estado del arte* de este documento, es momento de pensar y tener en cuenta **los principios de la animación**, y la clase de movimientos que existen o se puedan querer hacer. A la vez que se aplica el **estudio anatómico** del esqueleto de un gato que se realizó en la fase de preproducción. Con esto se consigue realismo en la animación, ya que se usa la anatomía real de un gato para simular los huesos que se crean en la estructura del esqueleto del modelo de la gata.

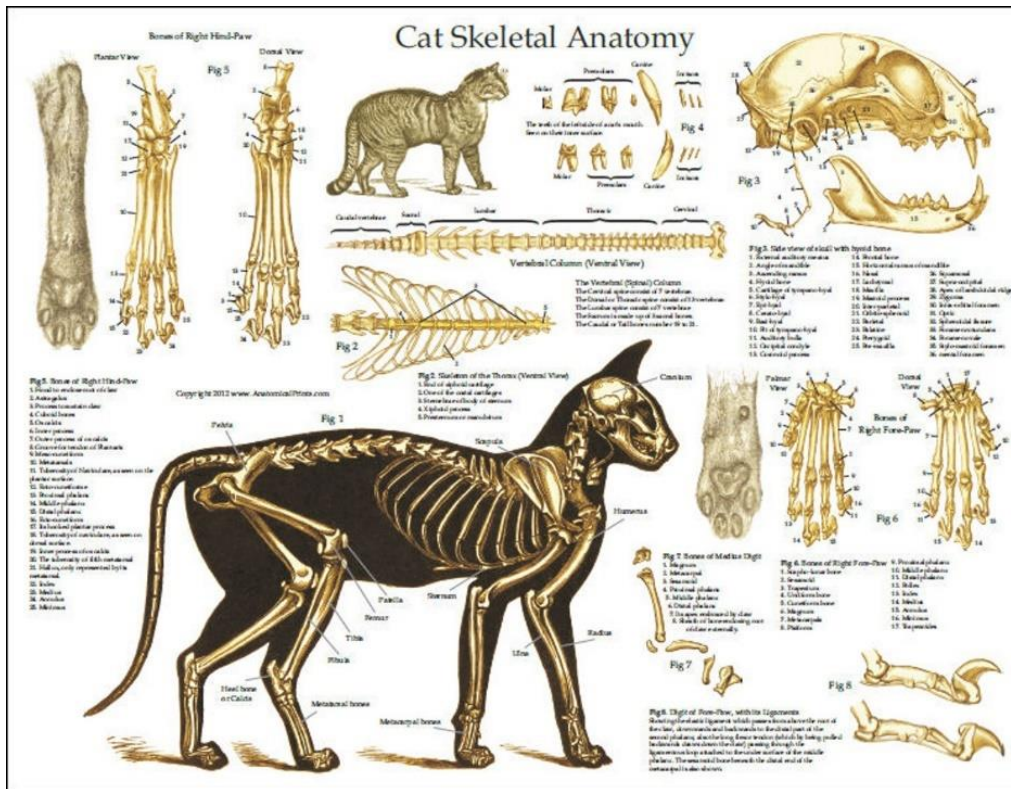


Figura 140. Referencia estudio del esqueleto de un gato
(Fuente: <https://clinicalcharts.com/products/cat-skeleton-anatomy-poster.html>)

Cuando se tiene todo listo y comprobado. Se **crea un hueso** en el *Object Mode* con la opción del menú *Add > Armature > Single Bone*. Ahora en el *Edit Mode*, con la **vista right** o *left* (se accede en el menú *View > Viewpoint > Right*) para asegurarse que todos los huesos quedan en el centro del modelo, se coloca y escala el primer hueso. En este caso, el hueso es el que va **al final de la columna vertebral** antes de que empiecen los huesos de la cola de la gata. A partir de ese hueso, se **extruyen el resto de los huesos** de la columna hasta la cabeza (incluyendo los del hocico para la nariz y mandíbula). Al extruir ya quedan **emparentados** entre ellos automáticamente. Hay que **colocarlos y escalarlos** para que tengan las dimensiones correctas. Una vez se tengan estos huesos, con el menú *Add > Armature > Single Bone* siguiendo en el *Edit Mode*, se **crea otro hueso** para hacer otra de las hileras de huesos, por ejemplo, la de la cola. Luego, de la misma forma, se realizan **las hileras de huesos** de: la pata delantera, la de la trasera y los de las orejas. Se cambia a la vista front desde el menú *View > Viewpoint > Front*, para **colocar mejor los huesos** con ayuda de ambas vistas (*front* y *right*). Esto es para los huesos que van en una de las dos partes del cuerpo, se escogió colocarlos a la derecha. También se añaden otros **huesos sueltos** que se encargarán de: controlar el **pecho**, por si hay que animar la respiración de la gata; la **barriga** que puede servir por si en la animación adelgazase o metiese barriga o al comer se hinchase, o cualquier cosa que se pueda imaginar; y un hueso con el que se pueda manejar el

esqueleto entero, por si se quiere mover, rotar o escalar el modelo entero en algún momento. Este es el hueso llamado **root**. Además de estos huesos, se pueden añadir otros para aplicarlas **las funciones IK** y poder animar de esta manera. Para este proyecto, se han añadido estos huesos que son: un controlador en el codo de cada pata, y otro hueso en el tobillo de cada pata, por si en el futuro se necesitasen, pero no se han implementado las funciones IK porque no se iba a usar este método de animación. Así el *rigging* y *skinning* queda preparado y solo habría que añadir esas funciones cuando se necesiten en el futuro. A continuación, **se emparentan todas las hileras de huesos y los huesos sueltos**, cada una con el hueso correspondiente del que tengan que depender y **de la forma que se tenga que conectar** en cada caso. Esto se realiza seleccionando primero el hueso que será el hijo y luego manteniendo la selección, el hueso padre, y usando la opción del menú *Armature > Parent > Make*. Con esto aparecen dos opciones que son: *Keep Offset* que se utiliza cuando deban estar emparentados, pero no conectados físicamente; o la opción *Connected*, que se usa para que estén emparentados y conectados físicamente (el final del hueso del padre comparte localización con el inicio del hueso hijo).

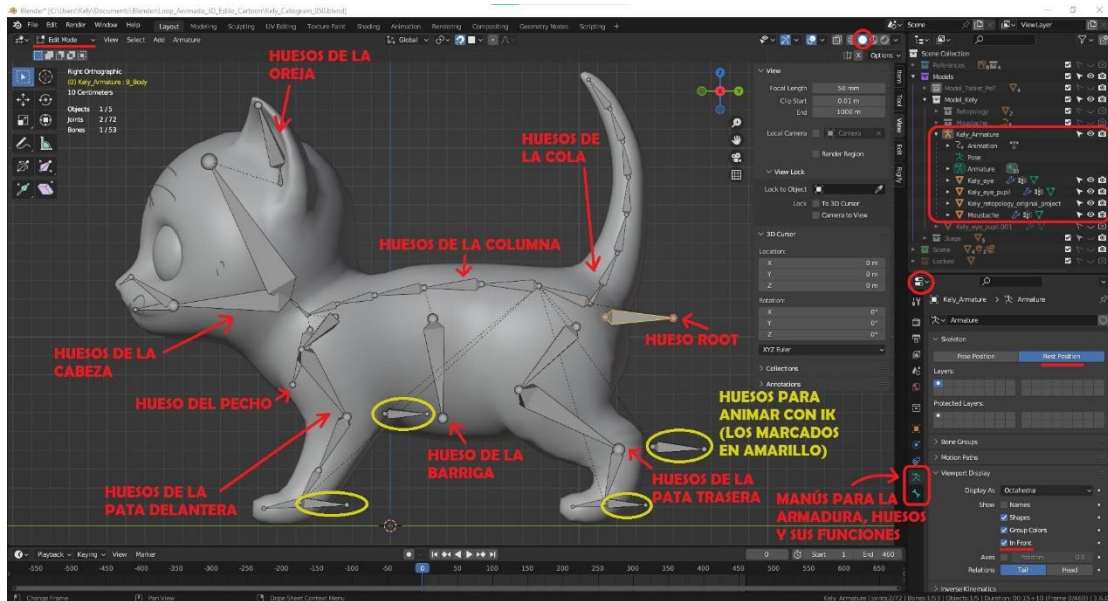


Figura 141. Rigging de la gata vista perfil
(Fuente propia: realizado en [Blender](#))

Luego, en el *Pose Mode*, se **comprueba** que todos **los huesos funcionan** correctamente como cabría esperar. Una vez se ha comprobado, en el *Outliner* **se nombra cada hueso**, es importante además nombrar los huesos que se han colocado en la **parte derecha** con algún tipo de **sufijo** que indique que son los de esa parte (como **_r** de *right*). Cuando todos estén nombrados, en el *Edit Mode*, se seleccionan los huesos que se colocaron a la derecha y se va a la opción del menú *Armature > Symmetrize*, para que se **creen los huesos de la izquierda** igual que los de esta parte. Además, se puede comprobar que los huesos de la izquierda tendrán el mismo nombre que los

siguiente acción. Con todo eso seleccionado se va a la opción *Object > Parent > With Automatic Weights* que **realizará un pesado de vértices de forma automática**. Pero, en el *Pose Mode* al comprobar cómo se deforma el modelo al mover los huesos, se puede ver que **no lo realiza correctamente**. Esto sucede siempre, a no ser que el modelo fuese un objeto sólido, sus partes estén separadas en mallas diferentes y se pueda hacer el pesado de vértices automático conectando cada parte del modelo con su correspondiente hueso individualmente. Pero no es el caso de nuestro modelo, ya que es un modelo orgánico y no se puede seleccionar por partes y conectar cada una a su correspondiente hueso. Por tanto, en este tipo de modelos, el pesado de vértices automático, siempre falla, ya que lo hace el programa interpretando a que parte del modelo afecta cada hueso, pero no puede saber para que sirve cada hueso y que extensión de la malla del modelo de verdad se tiene que ver afectada por cada hueso.

Así, que **hay que corregir el pesado de vértices**. Para ello, se selecciona el *rigging* y luego el objeto del modelo que se vaya a trabajar (el modelo está formado por varios objetos, los bigotes, los dos objetos de los ojos y el del cuerpo del gato), en ese orden para que se pueda editar y muestre para cada hueso que se seleccione a qué está afectando en el modelo. Se **cambia al modo Weight Paint**, donde se podrán **pintar con el degradado** de colores de rojo a azul, que se explicó en el estado del arte, cuánto y a qué parte afecta cada hueso. El **rojo es que afecta completamente**, y el **azul oscuro que no afecta** a esa parte. Se selecciona el hueso que se vaya a editar y con las herramientas se arregla a qué parte del modelo afecta ese hueso. Hay que tener en cuenta, que cuando mueves un hueso padre, para comprobar, se moverá la parte de la malla a la que afecten los huesos hijos también. Esto es importante tenerlo en cuenta para saber que, a veces, el error puede estar en los hijos. Con la **herramienta del pincel** se cambian los colores del degradado en la configuración de *Weight* donde 0 es el azul oscuro y 1 el rojo. Existen **otras herramientas** para desenfocar, emborronar, hacer un promedio, un degradado o la herramienta del cuentagotas. A la hora de pintar el pesado de vértices, se debe saber que aquí pintar no funciona como cuando se estaba texturizando el modelo. Aquí pintas los vértices, si hay zonas en las que hace falta algún vértice más habrá que subdividir la malla para que sea más preciso pintar la zona a la que afecta. Si no hay un vértice no se podrá pintar en ese espacio, y si no se quiere hacer una subdivisión porque el modelo tiene que ser *low poly*, por ejemplo, habrá que conformarse con pintar el vértice más próximo que vaya a hacer que el movimiento quede mejor. Hay que **ir realizando este proceso con todos los huesos y objetos del modelo**. Se puede **usar la simetría** para que al editar los huesos de la parte derecha se modifique de la misma manera los de la izquierda. Es un proceso que requiere paciencia y lleva muchas horas, pero una

vez que se termine de comprobar cada uno de los huesos con todos los objetos que forman el modelo, se tendrá listo el modelo para animarlo.

Por último, se muestran dos figuras, la primera muestra cómo se ve el peso de vértices en el objeto del gato para el hueso que afecta al muslo; y la segunda cómo se muestra en el *Outliner* la dependencia de huesos y que dentro de la armadura al conectarlo con los objetos del modelo que hacen de piel, estos crean una dependencia con la armadura y por eso están dentro de ella.

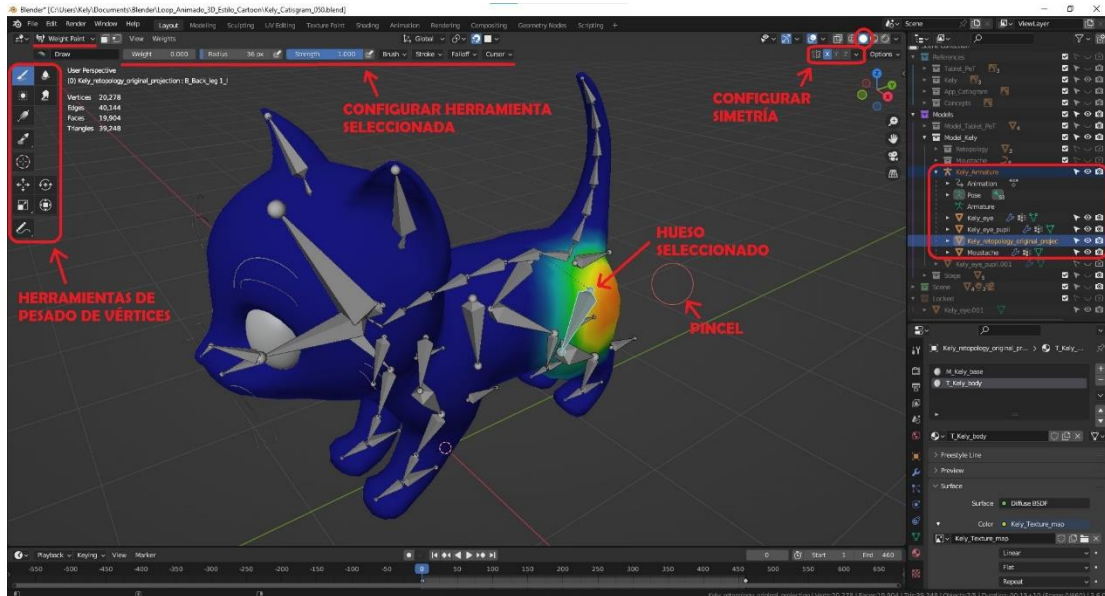


Figura 143. Muestra del pesado de vértices para el hueso del muslo de la gata
(Fuente propia: realizado en [Blender](#))

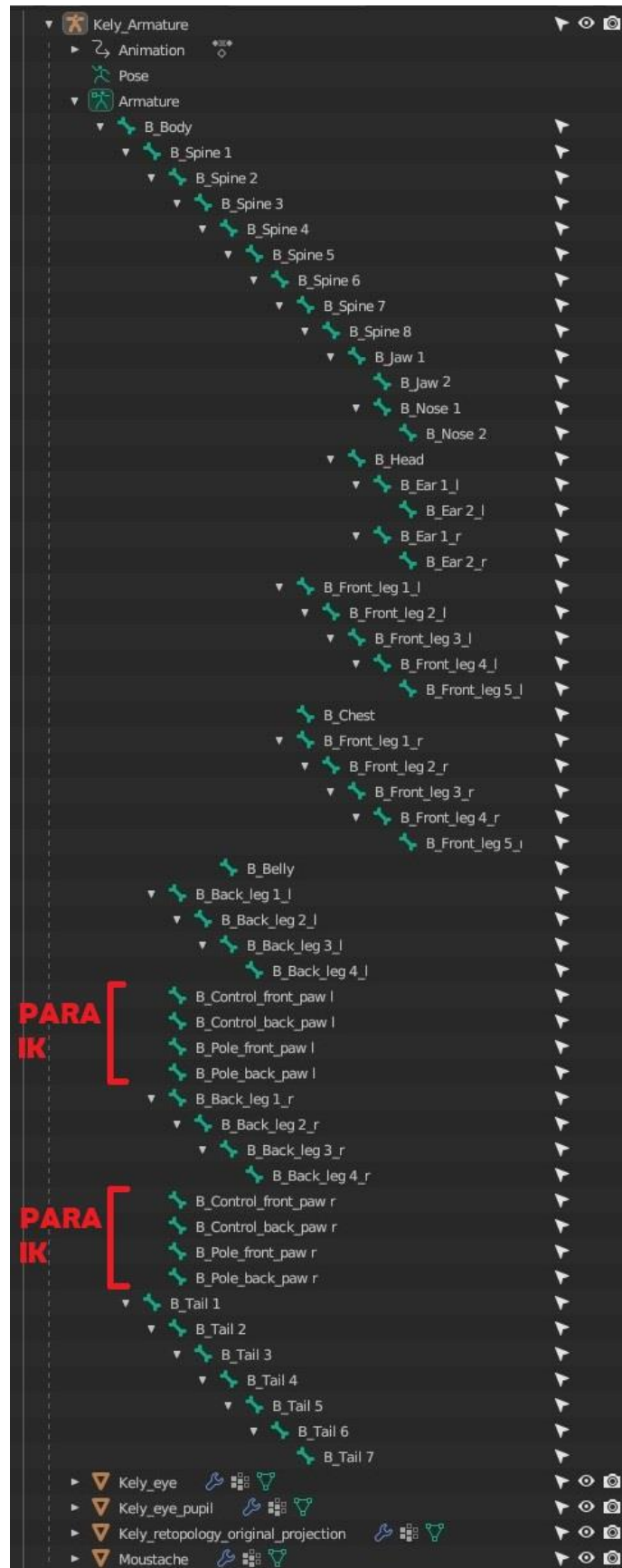


Figura 144. Dependencia de objetos y huesos de la gata nombrados en el Outliner (Fuente propia: realizado en [Blender](#))

8.5. Iluminación y Sombreado

Para iluminar la escena se han usado **4 luces**. La primera es **una luz de tipo Sun**. Funciona como la luz solar. Es la encargada de generar **las sombras del gato** en el fondo del escenario, el resto de las luces se han configurado para que no generen sombras, así no aparecerán varias en distintas direcciones. El color de esta luz es blanco, y tiene una fuerza de 0'2 para crear una sombra de luz de día en la animación. Se ha colocado frente al gato con un ángulo de 0'526º.

Las otras 3 luces que se han colocado son para **darle más valor a la escena**, así el render será de **mejor calidad**. La posición de estas luces se basa en los triángulos de luces que usan en **estudios de fotografía** y de grabación para **darle protagonismo al sujeto que iluminan**. La **configuración** y uso de estas 3 luces se explica a continuación. Una de las luces es de **tipo Area**. Ésta se encuentra detrás del gato iluminando en dirección a la cámara. Es de tipo área porque se usa para iluminar los objetos de la escena desde atrás y generar un brillo en la silueta para separar los objetos del color del fondo. Así quedan mejor definidas las siluetas de los objetos de la escena. Tiene una potencia de 50 W. Tiene una forma *Square* (cuadrada) que ocupa todo el plano del fondo del escenario. Las **otras dos luces**, como se va a renderizar con el motor de *Cycles* incluido en el programa de Blender, son **2 planos** creados con *Add > Mesh > Plane*. Se han configurado dos **materiales** con un nodo *Emission* para que **emitan luz**. Uno emite **una luz azul violácea** (el que está a la derecha mirando al gato de frente), y el otro **una luz blanca** (el que está a la izquierda mirando al gato de frente). La luz azul violácea tiene una fuerza de 8 y la blanca de 6. Para que estas luces no generen sombras se ha desactivado las opciones *Shadow* y *Camera* en el *Editor Type* la opción *Properties > Object > Visibility > Ray Visibility*. Estas dos luces generan los brillos en los objetos que le dan más valor, a la vez que la luz azul violácea añade un toque de color a la escena y a todo lo que la compone.

Por último, se mostrarán unas figuras con la configuración y disposición de estas luces en la escena.

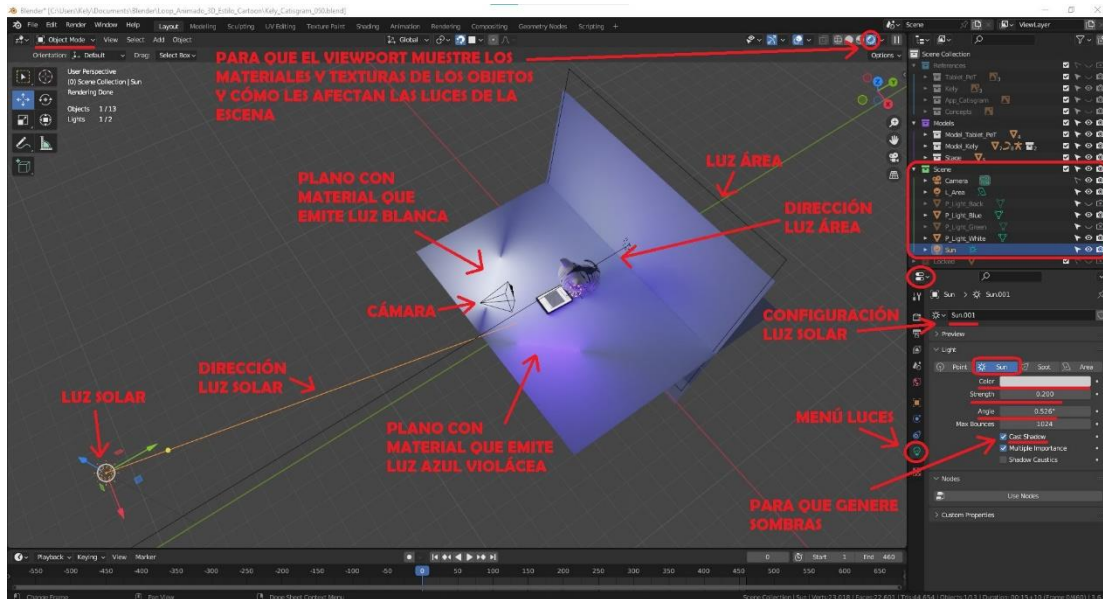


Figura 145. Disposición de luces en la escena y configuración de la luz solar
(Fuente propia: realizado en [Blender](#))



Figura 146. Configuración de las luces que forman el triángulo de luces
(Fuente propia: realizado en [Blender](#))

8.6. Animación

Este proceso se puede empezar una vez se tenga todo lo que va a intervenir en la animación preparado, menos el sistema de partículas que se puede configurar su animación en su propio menú de configuración como se explicará en su apartado.

Lo primero de todo, es **crear la cámara** para poder ir viendo cómo queda la animación durante todo el proceso. Cómo es una animación estática que se usará como recurso en redes sociales,

solo es necesaria una cámara que **estará estática**. Para crearla, en el menú se usa la opción *Add > Camera*. Para poder ver lo que se ve desde la cámara, **se cambia la vista** desde el menú *View > Cameras > Active Camera* o desde *View > Viewpoint > Camera*. Para poder **colocar la cámara** cómodamente, en el panel de la derecha del *viewport*, en *View* se activa la opción que hay en *View Lock > Lock > Camera to View*, de esta forma si se cambia la posición de la vista, se estará cambiando la posición y la lente de la cámara. Una vez que esté colocada es importante desmarcar esta opción (*Camera to View*) para no modificarla sin querer.

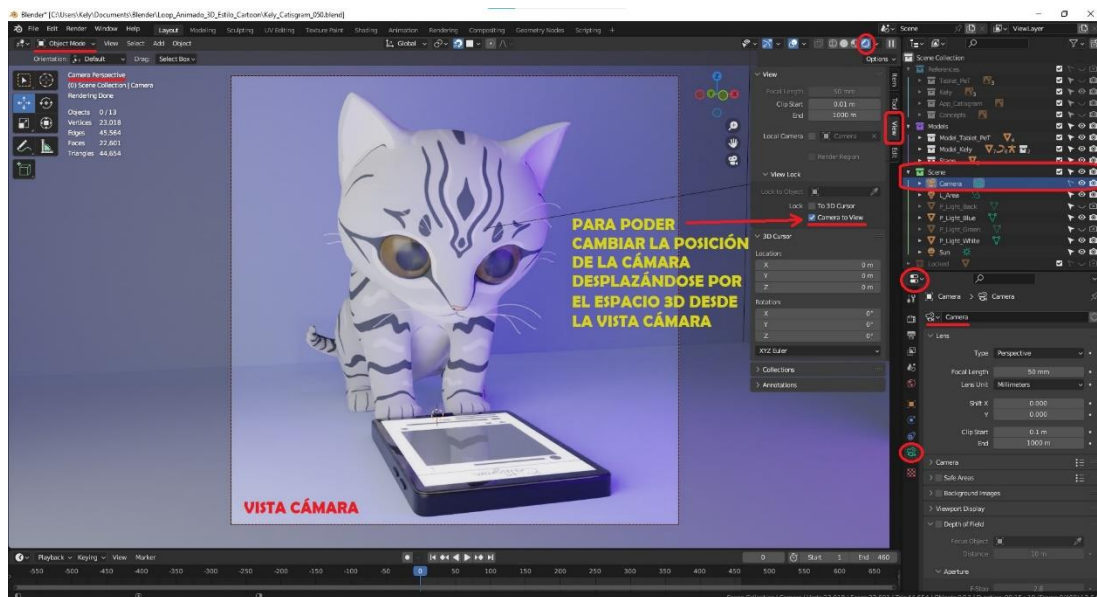


Figura 147. Configuración de la cámara
(Fuente propia: realizado en [Blender](#))

Ahora que hay una cámara con la que se puede ir comprobando cómo queda la animación, se va al **espacio de trabajo Animation**, donde se animará al gato. Para **animar al gato**, se tiene que seleccionar el objeto de su armadura (el *rigging*) y cambiar al **Pose Mode**. Ahora en el *Editor Type* la opción **Dope Sheet**, se irá mostrando una lista desplegable de los objetos que se animen y que partes cambian de estos objetos en cada frame. Para el primer frame se posiciona el gato sentado. Para hacerlo es mejor ir **cambiando entre las vistas left, right, front y back** en el menú *View > Viewpoint* para poder estar seguros de que al rotar y mover los huesos lo hacen en la dirección deseada y no se giran hacia otro eje que no deben.

Para poder sentar al gato en una pose que quede natural, se han usado **referencias de gatos de la raza Sphynx sentados**, ya que estos gatos no tienen pelaje y se puede ver mejor la anatomía del gato como queda en esta pose. Además, como el modelo del gato 3D que se ha realizado es estilo cartoon y no tiene pelo, la pose de estar sentado deberá parecerse a la de esta raza de gatos. En las figuras de abajo se puede apreciar que los gatos al sentarse dejan un espacio entre

el codo de las patas traseras y la parte de debajo de la espalda por donde pasan la cola que hace que **se sienten como si estuviesen de cuclillas**. También, se puede apreciar la forma de **la curva que crea la columna vertebral** del gato en esta pose. Por último, se puede ver que suelen **acercar mucho las patas delanteras a las traseras** en esta posición. Todo esto se ha tenido en cuenta para sentar a la gata y que quede lo más natural y cercano a la realidad posible.



Figura 148. Referencia gato Sphynx sentado de perfil
(Fuente: <https://www.istockphoto.com/es/foto/gato-sphynx-sentado-disparo-de-estudio-fondo-blanco-qm1193691730-339629435>)



Figura 149. Referencia gato Sphynx sentado de frente
(Fuente: https://www.freepik.es/fotos-premium/gato-esfinge_6665489.htm)

Una vez, que se haya posado a la gata sentada, se ha fijado en los GIF que se buscaron para el **moodboard el movimiento de los gatos reales ante un dispositivo** como una tableta o un móvil. Con estos GIF, se ha estudiado el movimiento de las patas y cómo se doblan al tocar las pantallas de los dispositivos. Como se puede ver en el GIF de la imagen siguiente si se clica el enlace de la fuente, se puede apreciar **cómo se comportan las articulaciones de las patas**. Más o menos los huesos de las patas de un gato funcionan de la misma forma que los de los humanos, solo cambian sus formas y posiciones, mientras los humanos tienen una **posición relajada de los huesos** recta y seguidos unos de otros; los de los gatos tienen una posición relajada con una forma de zigzag que, de manera simplificada va desde el omoplato al hombro hacia delante (del cuerpo), desde el hombro al codo hacia atrás y desde el codo a los dedos hacia delante. Por tanto, en el modelo, el primer hueso de la pata, el que está más arriba, será el que simule los movimientos de la escapula u omoplato, con el que pueden acercar o alejar la pata hacia el dispositivo, y subir o bajar todos los huesos de la pata. El siguiente hueso será el húmero que tiene la articulación del hombro (se puede articular de esa manera). Luego el hueso que está en la posición del cúbito y radio, éste, aunque estiren mucho la pata hacia delante, siempre suele marcar el codo hacia atrás en la pata, porque la articulación no permite que lo roten en dirección

contraria (como la articulación del codo de los humanos). Este hueso además se puede rotar sobre su propio eje para hacer que muestren la almohadilla de la pata hacia arriba, que es lo que hacen cuando agarran algo con la pata. Luego está el hueso de los metacarpos que es la parte de la pata antes de los huesos de los dedos, este hueso pueden articularlo hacia atrás y hacia adelante. Por último, el hueso que de manera simplificada hará el papel de los dedos que se puede articular hacia dentro de la almohadilla de la pata. En los GIF, también se ha estudiado **los movimientos del resto del cuerpo** que hace el gato al usar el móvil. Se puede apreciar que mueve **la cabeza** al mover las patas, y dirige la mirada al dispositivo como si le extrañase ver que se mueve el interior de la pantalla del objeto cuando lo toca. También se aprecia que da pequeños golpes secos, lentos y sutiles con **la cola**, de vez en cuando, y cuando le divierte ver que se mueven cosas en el dispositivo, esos movimientos se aprecian más y los realiza con más energía. Todos estos movimientos y cómo se articula cada hueso, se ha tenido en cuenta en la animación para crear un movimiento de la gata lo más realista posible.



Figura 150. GIF de un gato real usando un móvil con la pata
(Fuente: <https://tenor.com/es/view/funny-animals-cute-cats-animals-aww-gif-12603952>)

Asimismo, hay que tener en cuenta **los principios de la animación** que se explican en el capítulo de *Estado del arte* (en concreto, el 3.3 *Principios de la animación*) de este documento. Un ejemplo entre otros sería que se implementó el principio de anticipación en la pata de la gata justo antes de que le de *like* a la imagen haciendo que mueva de lado a lado, la parte de la pata que equivale a la mano de un humano, para que sutilmente de a entender (inconscientemente el ojo humano envía esas señales al cerebro cuando se mira la animación) que va a darle golpecitos a la pantalla esta vez en lugar de deslizar la pata sobre ésta de nuevo.

Teniendo todo el estudio y la teoría en mente, y con el modelo de la gata ya sentado, en el bloque del *Dope Sheet* **se posiciona en el frame 0**, se seleccionan todos los huesos y se va a *Key*

> *Insert Keyframes*. Con esto **se guardan las transformaciones de todos los huesos en esa posición** para ese frame. Esta será la primera pose de la animación y, por tanto, la última, ya que la animación tiene que comenzar y terminar en la misma pose para crear el efecto de *loop* animado (animación en bucle infinito). Ahora se va moviendo en la línea de *frames* el marcador del *frame* en el que se encuentra la animación, y se activa la opción *Auto Keying*. Esta opción guardará de forma automática las transformaciones que se hagan a los huesos en el frame en el que se encuentre el marcador. Como estará activada, hay que tener cuidado cada vez que se vaya a mover, rotar o escalar un hueso de que el marcador este en el frame que corresponda. A continuación, se va **mirando el storyboard** que se hizo en la preproducción **para saber el tiempo de cada pose clave y la pose clave**. Sabiendo que 25 fotogramas equivalen a 1 segundo de animación a 30 fps que es como se configurará el render. Se van realizando cada pose clave en su correspondiente fotograma. El programa completará el movimiento que realiza cada hueso para pasar de la posición de una pose clave a otra por **interpolación**. Pero esta interpolación, a veces, realiza movimientos que no deberían verse así (genera errores en la animación). Así que se pueden **realizar poses intermedias** para mejorar y corregir el movimiento en la animación. Por ejemplo, muchas veces al posicionar la pata en la pantalla, la punta de los dedos atravesaba la pantalla y había que corregir esa pose para que no lo hiciese, igual cuando la pata volvía a apoyarse en el suelo.

Hay una forma de **copiar las transformaciones de los huesos de una parte de la simetría del cuerpo a la otra** (si los huesos son los mismos, pero al otro lado del cuerpo). Esto se usó cuando ya se había logrado el movimiento por primera vez de la pata izquierda deslizándose por la pantalla correctamente, para copiarlo en la pata derecha. Para ello, en el viewport hay que tener seleccionados los huesos de los que se quiere copiar el movimiento. A continuación, cambiar el *Editor Type* a la opción *Graph Editor*, donde se puede ver en un gráfico como se guardan las transformaciones de los huesos seleccionados. Ahí seleccionar todo el movimiento del frame al frame que se quiere copiar. Se copia como en cualquier programa, *botón derecho del ratón* > *Copy* (o presionando en el teclado Ctrl + C en Windows). Luego en el viewport se seleccionan los mismos huesos, pero de la otra pata, se vuelve a cambiar el *Editor Type* a la opción *Dope Sheet*, se coloca el marcador en el frame que se quiere pegar y se pega con el *botón derecho del ratón* > *Paste* (o presionando en el teclado Ctrl + V en Windows). Esto agilizará el proceso de la animación ya que solo se tendrá que corregir las poses intermedias si lo requiere. Para **copiar las transformaciones de los huesos de una misma pata en otros frames**, se pueden usar las opciones de copiar y pegar en el *Editor Type* la opción *Dope Sheet*, sin tener que cambiar el editor.

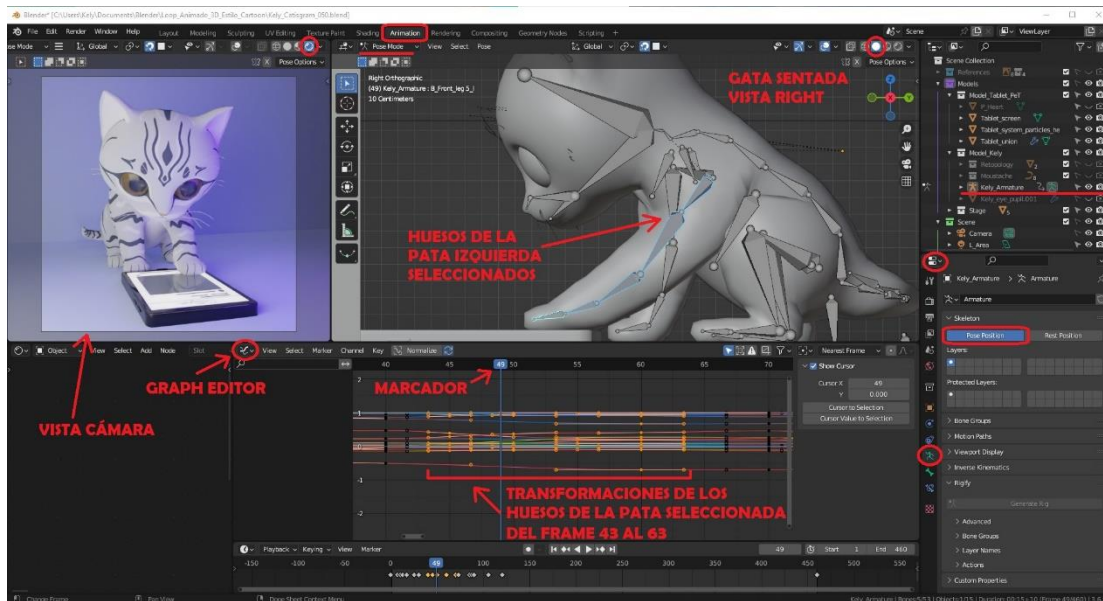


Figura 151. Panel Graph Editor para copiar transformaciones de un hueso a otro
(Fuente propia: realizado en [Blender](#))

Con todos estos datos, se realiza la animación de la gata. Para realizar la **animación de la app**. Hay que seleccionar el objeto de la pantalla en el *viewport*. Se coloca el marcador donde se vaya a cambiar la textura de la app, y se cambia en el *Editor Type* la opción *Shader Editor*. Aquí se configura para que la animación de estas texturas sea: *Linear*, para que la transición de una textura a otra no pase por las texturas del medio, sino que se produzca de golpe; *Flat* porque se va a hacer la proyección de la textura sobre un objeto plano (que es la pantalla); *Clip*, para que, si la imagen es más grande del objeto en el que se proyecta, se recorten las partes sobrantes; e *Image Sequence*, para indicar que la textura proviene de la selección de varias imágenes. Se selecciona el fotograma 1, y con la opción *Offset* se va **cambiando** en los *frames* correspondientes, **el número de la textura que se tiene que ver en la pantalla** en ese momento. En las siguientes figuras se pueden ver todas las transformaciones que se han guardado en que frames y a que huesos y objeto pertenecen.

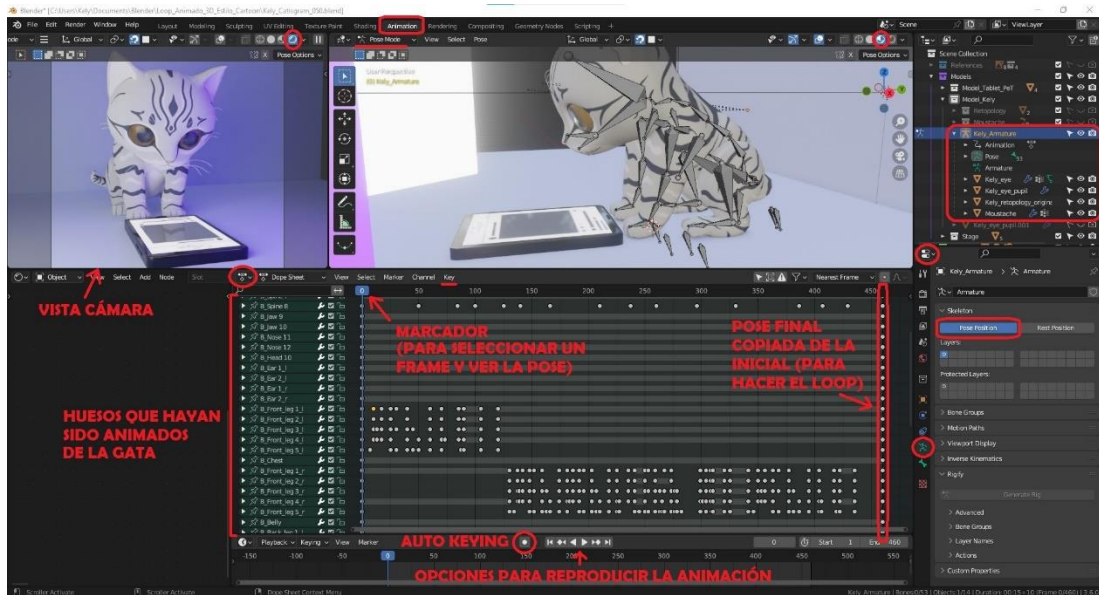


Figura 152. Animación de la gata
(Fuente propia: realizado en [Blender](#))

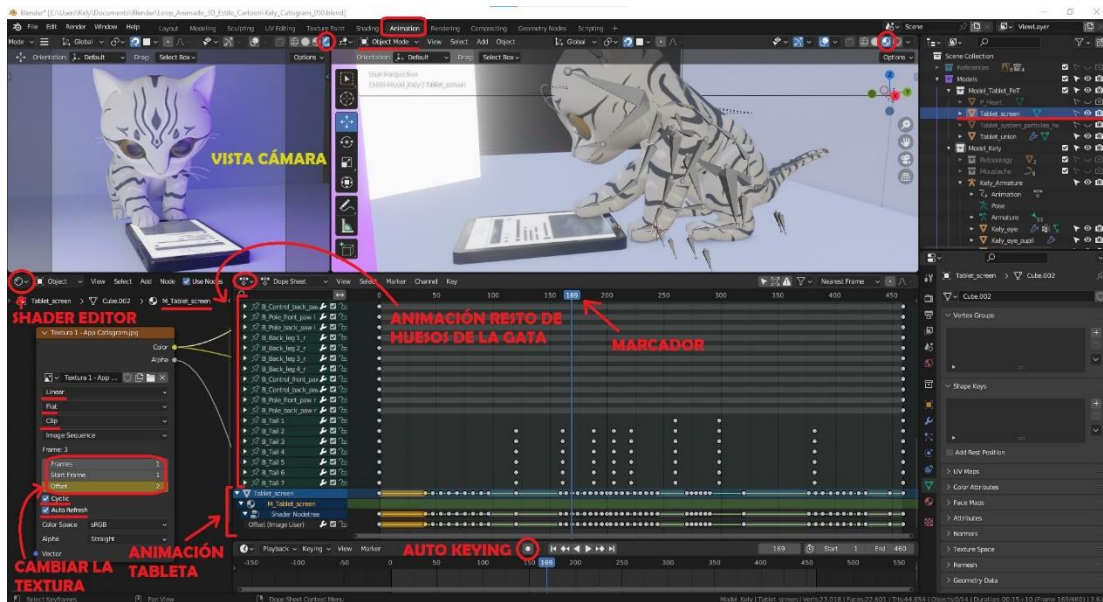


Figura 153. Animación de la pantalla de la tableta y final de la gata
(Fuente propia: realizado en [Blender](#))

8.7. Partículas

El sistema de partículas, en el caso de esta animación, se usará para crear **los corazones que saldrán por la pantalla de la tableta y su animación cuando el gato le de like a una imagen de la app Catisgram.**

Para ello se ha **creado un plano Add > Mesh > Plane** en el que se ha incluido **la textura del corazón** de la misma forma que se pusieron las texturas en la pantalla de la app y sobre el

modelo de la gata en el menú del *Editor Type: Properties > Material* y en el espacio de trabajo *Shader Editor* para crear el nodo de la textura que se conectará a la salida.

Luego se ha **creado otro plano** con las dimensiones de la imagen de la app Catisgram. Este plano se ha colocado detrás de la pantalla de la app, introducido en el teléfono. Este objeto **será el emisor de partículas**, para ello, hay que añadirle un modificador en el *Editor Type* la opción *Properties > Modifiers > Add Modifier > Particle System*. Para **configurar el sistema de partículas** hay que ir al apartado en el *Editor Type* la opción *Properties > Particles*. Se ha escogido que el objeto es un *Emitter*. Se han configurado las siguientes opciones dentro de este menú:

- En el apartado **Emission**. El número de partículas que creará para la animación será 50. **La animación de partículas** empezará en el frame 274 hasta el 320 (esto se sabe porque en la animación es cuando coincide que sale la animación del corazón blanco de la app en la pantalla). El tiempo de vida de las partículas será de 50.
- En el apartado **Render**. Se ha escogido que se renderice como *Object* con una escala de 0'2 y un número aleatorio para el tamaño de cada corazón de 1. Así cada una de las partículas será de un tamaño distinto. En las opciones del *Object*, se ha escogido el objeto del plano con la textura del corazón que se creó anteriormente, para que las partículas aparezcan con esta forma. También se han marcado la opción de la rotación de ese objeto para que las partículas aparezcan de frente a la cámara.
- En el apartado **Field Weights**. Se ha puesto a 0 el valor de la gravedad para que las partículas salgan hacia arriba desde el emisor.

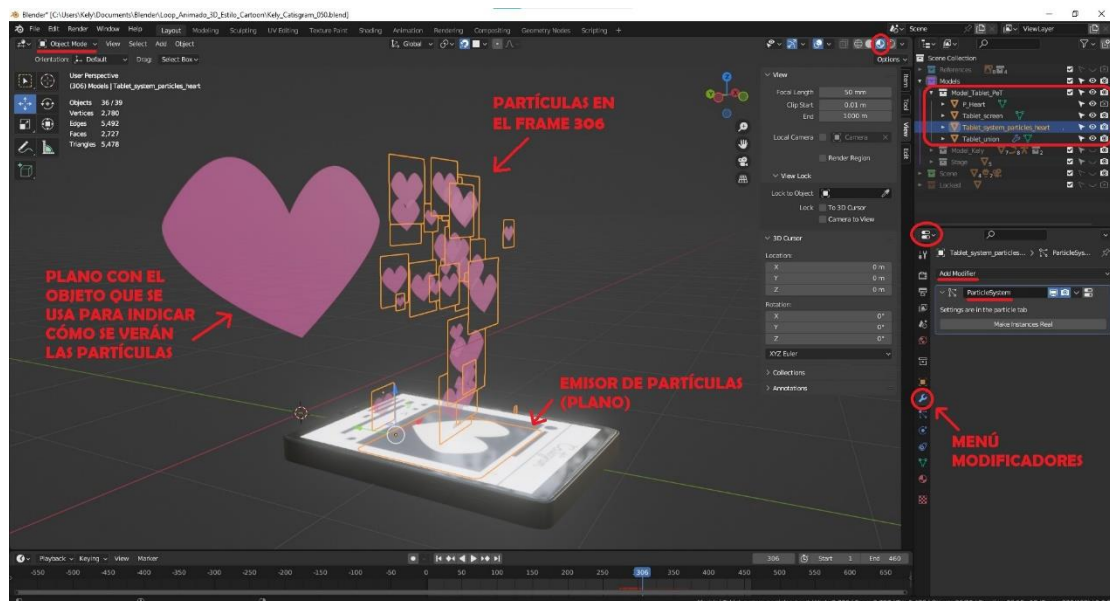


Figura 154. Sistema de partículas: los corazones que salen de la pantalla (Fuente propia: realizado en [Blender](#))

CONFIGURACIÓN DE LAS PARTÍCULAS EN EL MENÚ DE PARTICLES

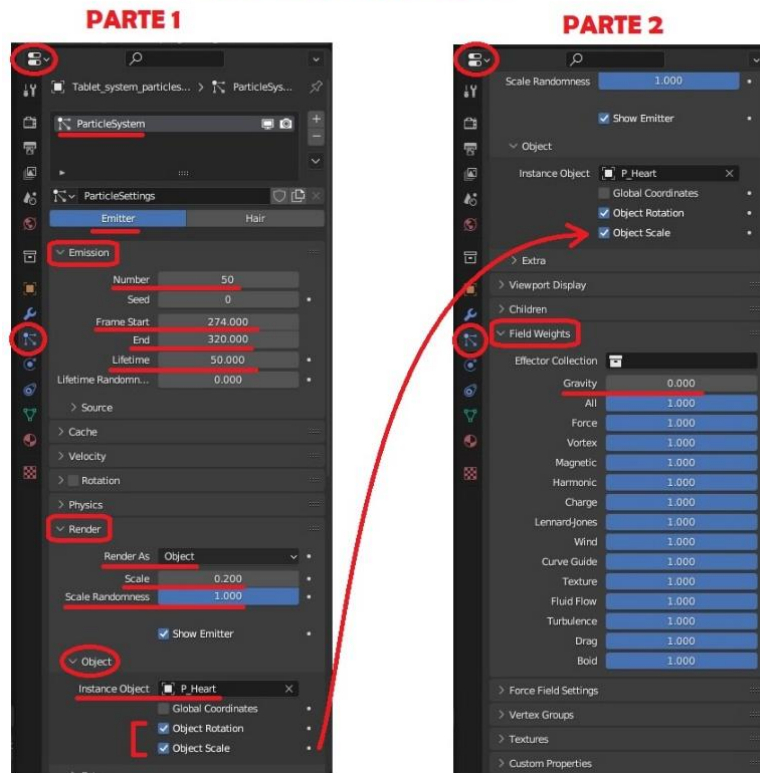


Figura 155. Configuración del sistema de partículas de los corazones
(Fuente propia: realizado en [Blender](#))

8.8. Renderizado

Para el render se usó, como ya se ha mencionado, el **motor** del render **Cycles** de Blender, que usa el trazado de rayos. La configuración que muestra la siguiente figura en el *Editor Type*, en las opciones *Properties > Render*, y *Properties > Output* es las que se utilizó para esta tarea una vez la animación estaba completada.

En el menú del **Render** se ha escogido el motor de *Cycles*, que el renderizado lo haga por GPU, y se han activado las opciones de: el *Noise Threshold* a 0'01 y el *Denoise* para que no haya ruido en la imagen; el *Simplify* para la configuración de las subdivisiones de los objetos; y *Motion Blur* para el desenfocado de movimiento. Todo ello hace que funcionen las luces y se vean los materiales que se han configurado (como los cristales y los reflejos en ellos).

En el menú del **Output**, se ha escogido un formato HD (1080 x1080) del 100% y una relación de aspecto 1:1 para que sea cuadrado, que es el formato de los GIF normalmente. Se ha indicado el frame inicial 1 y el final 460 y que vaya de frame en frame la animación. Se ha seleccionado la

carpeta de salida donde guardará en el ordenador la animación. El formato de la salida será en vídeo FFmpeg Video con color RGB. Codificado con formato MPEG-4, con códec H.264 de buena calidad (suficientemente alta para un GIF) con un intervalo de fotogramas clave de 18, y sin audio.

CONFIGURACIÓN MENÚ RENDER

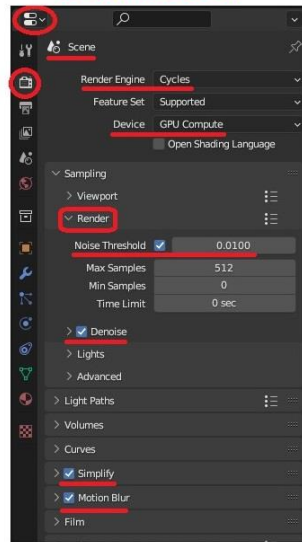
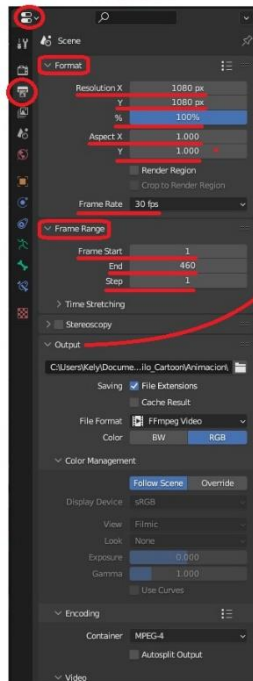


Figura 156. Configuración del Render
(Fuente propia: realizado en [Blender](#))

CONFIGURACIÓN MENÚ OUTPUT

PARTE 1



PARTE 2

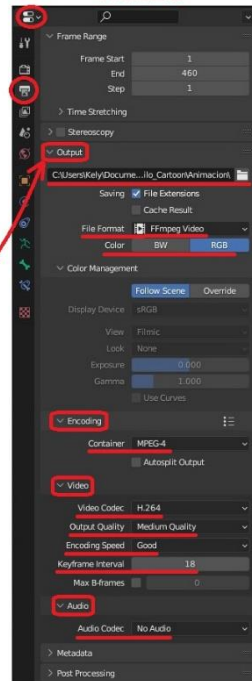


Figura 157. Configuración Output
(Fuente propia: realizado en [Blender](#))

9. Postproducción

La postproducción es la última fase, en la que se reúnen todos los elementos gráficos obtenidos en la fase anterior, se juntan en una única composición final y se pueden añadir los arreglos y retoques finales para dar como resultado el producto final que es de lo que trata el proyecto.

En este capítulo quedarán reflejados los pasos realizados en el desarrollo de esta fase, cómo se han realizado cada una de las tareas de la postproducción. Todas estas tareas fueron realizadas con la aplicación web Adobe Express, que ahora mismo está en fase beta. Se usó una cuenta de Adobe de la que ya se disponía anteriormente.

9.1. Composición y Motion Graphics

Para la **composición**, dentro de Adobe Express se fue al apartado de *Vídeos* del menú y se escogió la opción de *Empieza a partir de un vídeo*. Se escogió el **vídeo** en formato .mp4 que se había generado con el render en Blender en la fase anterior.

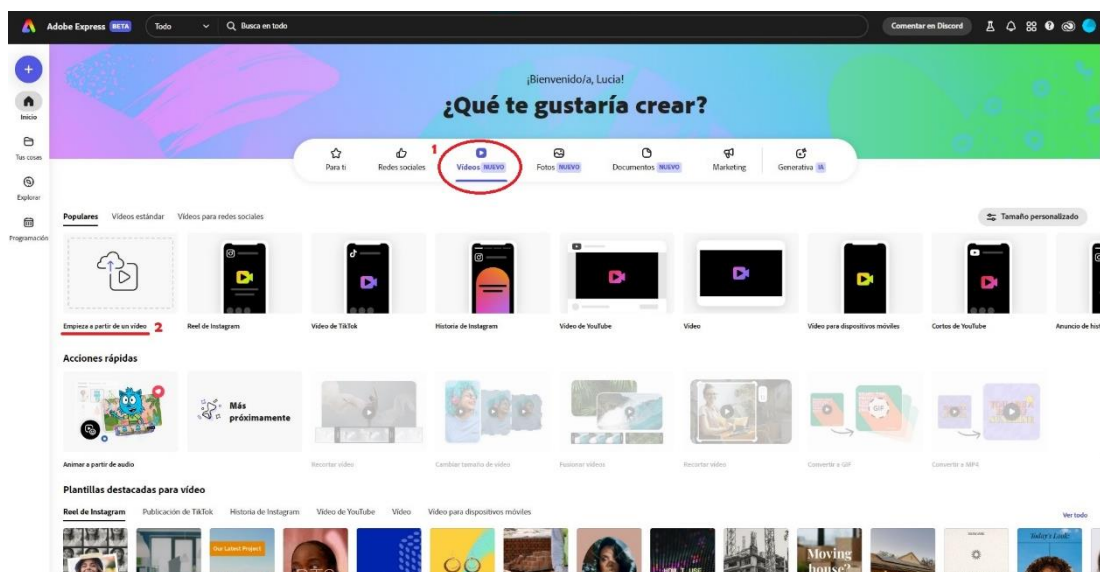


Figura 158. Aplicación Adobe Express en el menú
(Fuente propia: realizado con [Adobe Express](#))

Una vez escogido y cargado en la página, como **elementos visuales**, se añadió un **texto** que pone *2023@Calipsoy* en la esquina superior derecha. Este texto se añadió a modo de marca de agua como firma, ya que *Calipsoy* es el nombre de usuario en redes sociales, por eso lleva el @ delante, y 2023 indica el año en el que se realizó la animación en este caso. Para este texto se utilizó la **fuente** de *Vollkorn* con el formato Bold Italic (negrita y cursiva). Esta fuente también

está disponible en el sistema Windows por defecto, y, por tanto, en cualquier programa que permita escoger la fuente del texto. Se escogió la negrita para que sea más fácil la lectura, y en cursiva porque es parte de la marca personal de la autora, ya que define un estilo que simpatiza con los trabajos que suele realizar. Además, es el mismo motivo por el que se escogió la fuente, representa actualmente el estilo del arte de la autora. El código de **color** de relleno de texto que se escogió en hexadecimal fue el HEX #B4C2FF, que es un color que en la zona donde se colocó el texto se diferenciaba muy bien del fondo para facilitar su lectura y a la vez no molesta visualmente en el vídeo, ni llama más la atención que el contenido principal. Así que se consigue lo que se pretendía añadiéndolo al vídeo. Una vez finalizado se da al botón superior de la derecha de **descargar**, se escoge el formato, en este caso .mp4.

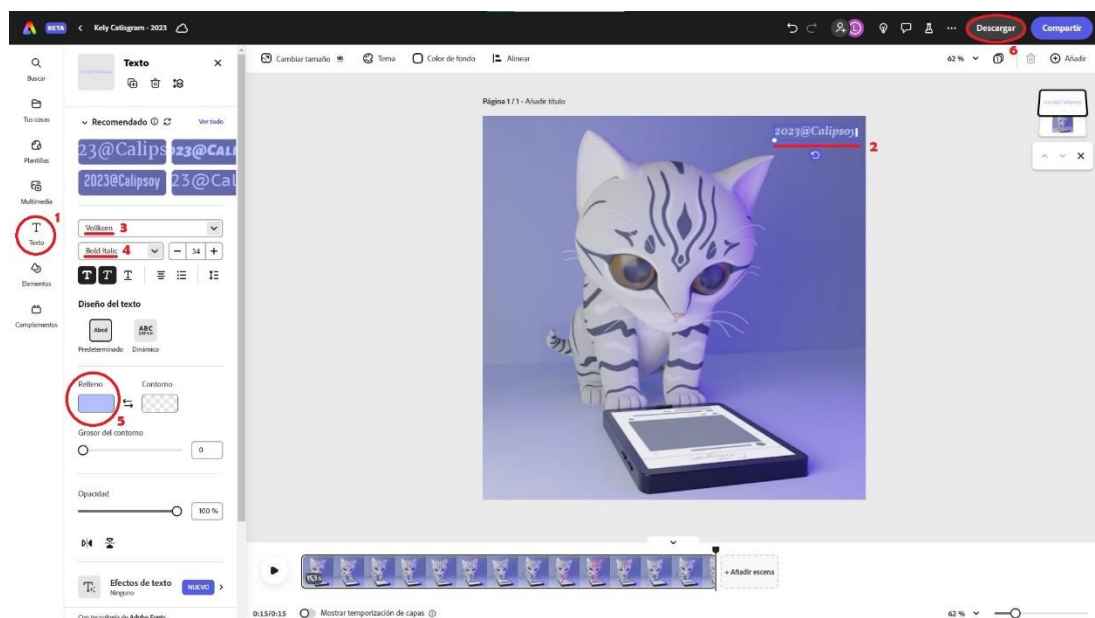


Figura 159. Edición de vídeo con Adobe Express, añadiendo texto
(Fuente propia: realizado con [Adobe Express](#))

Por último, para **finalizar la composición**, se volvió al menú de la página de Adobe y en *Funciones* se escogió la opción de **Convertir MP4 a GIF**. Se arrastró el vídeo que se había generado anteriormente con los motion graphics (el texto). Una vez se cargó en la página, se escogió el **tamaño de archivo** y el **tamaño Cuadrado (1:1)**, que es el mismo que tiene el vídeo desde el render, así se evita que salgan las franjas negras que se añaden a algunos vídeos porque en este proyecto no interesan, y la mejor forma es la cuadrada para las redes sociales, y se **descarga**. Este paso se realizó 3 veces para descargarlo en los 3 tamaños de archivos disponibles, grande, mediano y pequeño. Finalmente, así se obtiene la animación en un formato de imagen que se reproduce en bucle infinitamente, y es posible usarlo para enviarlo por las distintas redes sociales.

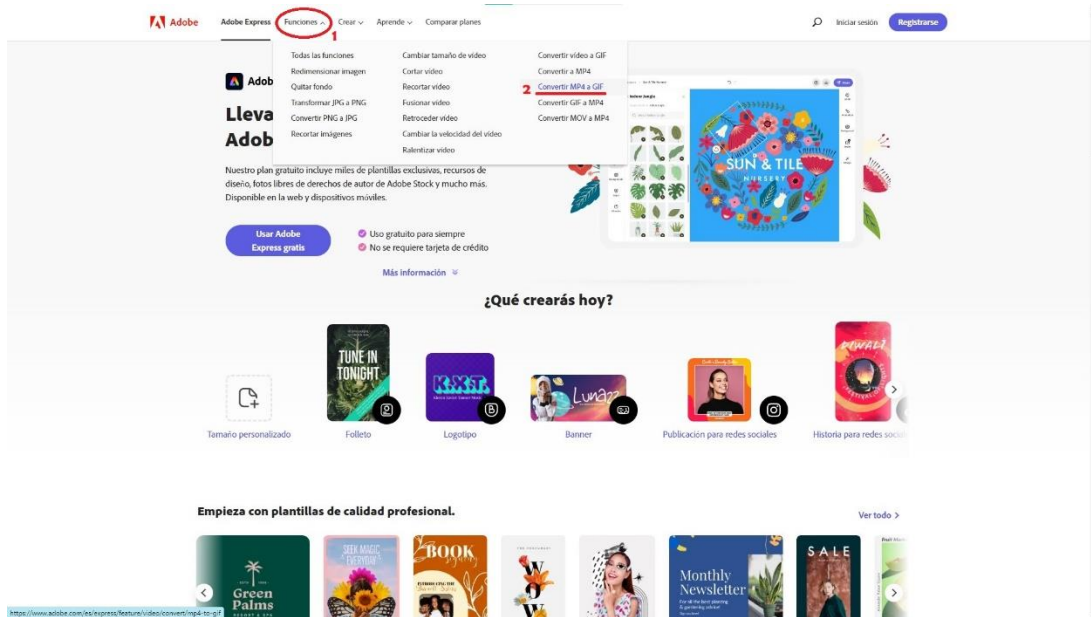


Figura 160. Acceder a la función Convertir MP4 a GIF de Adobe Express (Fuente propia: realizado con [Adobe Express](#))

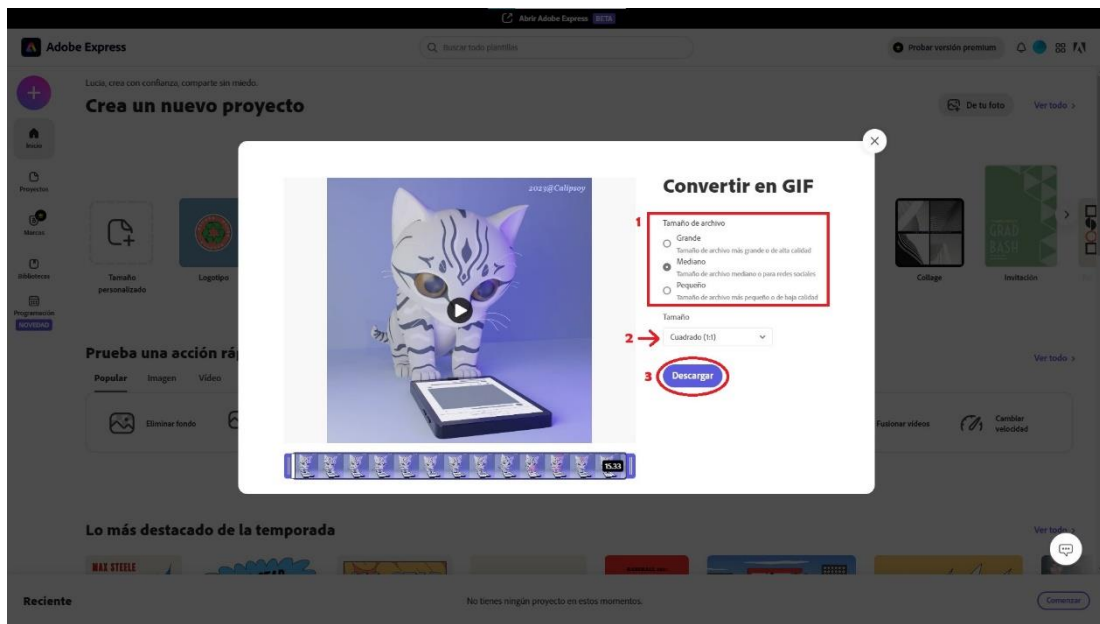


Figura 161. Configuración para convertir MP4 a GIF con Adobe Express (Fuente propia: realizado con [Adobe Express](#))

10. Resultado final

En este capítulo se analizarán los resultados finales obtenidos al realizar el proyecto. Para ello, se mostrarán los resultados y se realizará un análisis de los objetivos principales que tenía este proyecto descritos en el capítulo de *Objetivos* de este documento.

Se ha subido a la plataforma de YouTube el vídeo de la animación que se repite 3 veces para que se pueda observar que el bucle ha conseguido lograr el objetivo de que se puede reproducir en bucle y no se notan cortes en la animación al comenzarla de nuevo. El enlace al vídeo es el siguiente (clicando en la imagen):



*Figura 162. Muestra de resultado final como imagen con el enlace al vídeo
(Fuente propia: subido a [YouTube](#))*

Las texturas de la app *Catisgram*, se han realizado a propósito de esta manera, sin incluir texto e imágenes, con el fin de que no llame más la atención el contenido de la app que los movimientos que realiza el gato. Sin embargo, se ha añadido el detalle de poner el nombre de la app como diseño del logo, para aportar un detalle sutil a la animación, que no roba protagonismo a la gata.

El resultado se puede comparar con el GIF del gato real usando un dispositivo móvil que aparece en el apartado de *Animación* del capítulo de *Producción*. Se vuelve a colocar el GIF a continuación con el fin de tener el enlace a mano y poder compararlos.



Figura 163. GIF de un gato real para compararlo con el resultado final
(Fuente: <https://tenor.com/es/view/funny-animals-cute-cats-animals-aww-gif-12603952>)

A continuación, el análisis de los objetivos. Por una parte, uno de los objetivos principales del proyecto, consistía en **aprender a realizar un *loop* animado en 3D estilo cartoon con todas las fases** que fuesen necesarias para llevarlo a cabo. Este objetivo ha sido cumplido con éxito ya que se ha conseguido realizar el *loop* animado desempeñando todas las fases del proyecto. A parte de realizar todas las fases, éstas **han sido realizadas por la autora a mano** con el objetivo de poner en práctica los conocimientos adquiridos y practicarlos en el proceso, ya que nunca se habían realizado todas las fases para un mismo trabajo en proyectos anteriores. De hecho, algunas de las fases como la retopología o *rigging* era la primera vez que se realizaban. Aunque se sabe que existen herramientas que realizan automáticamente algunas de las tareas, se decidió no utilizarlas por varias razones:

1. Que la mayoría son herramientas o complementos de los programas son de pago y no se quería utilizar programas de pago para demostrar que se puede realizar una animación con un presupuesto bajo.
2. Se quería aprender a realizar el proceso de primera mano para saber cómo solventar los problemas que puedan surgir utilizando estas herramientas en el futuro.
3. Las herramientas gratuitas que se podían usar para estas tareas no realizaban correctamente su función. Por ejemplo, en retopología existe una aplicación llamada *Instant Meshes* que realiza una retopología del modelo, y es capaz de crear una malla con *quads* a partir del modelo. Pero no es capaz de colocar los *loops* de polígonos

siguiendo las formas del objeto correctamente y, por tanto, no se puede realizar correctamente la tarea de texturizado con la malla que genera.

También **se quería dejar reflejado en este documento todo el proceso y el estudio realizado** para que pueda servir como una guía de apoyo para futuros proyectos, y también como una guía educativa con lo básico para quien quiera empezar a aprender animación y modelado en el mudillo 3D. Con toda la información teórica que recoge este documento cumple ese objetivo, y al escribir paso por paso cómo se ha realizado el proyecto de animación del *loop*, se obtiene una guía que ayuda a saber, para proyectos similares, qué pasos se realizan, en qué hay que fijarse, y cómo se pueden evitar algunos problemas que puedan surgir.

Por otra parte, el otro objetivo principal era el de **reflejar la cantidad de horas que una persona puede pasar viendo redes sociales**, de forma irónica y con humor. Ya que todos los recursos artísticos se realizan con la intención de transmitir algún tipo de sentimiento, sensación o mensaje por parte del autor. Esto se ha logrado creando un recurso artístico, el *loop* animado, en el que se puede ver a la gata navegando por una red social, y al ser una animación con un bucle perfecto (que no se notan cortes cuando vuelve a comenzar) se puede reproducir de forma indefinida, refleja así el tiempo que se puede pasar en las redes sociales sin darse cuenta. Es una forma sutil y divertida de representarlo. Se puede decir que el *loop* animado ha conseguido contar esta moraleja.

Además de los objetivos principales, se han logrado **los subobjetivos** del proyecto, se describirá cómo se han logrado en la siguiente lista (se han escrito en el mismo orden en el que están redactados en su capítulo correspondiente):

- Se han **utilizado y ampliado los conocimientos adquiridos en el grado** en las siguientes asignaturas (entre paréntesis se explica para qué han servido los conocimientos de cada una): Fundamentos de la Física (para animar al personaje y crear el sistema de partículas), Fundamentos del Diseño Gráfico (para realizar las tareas de la fase de *Preproducción*), Sistemas Multimedia (para gestionar el proyecto y poner en práctica los pasos de resolución de problemas), Análisis y Especificación de Sistemas Multimedia (para realizar la planificación de forma apropiada, y escoger la metodología adecuada al proyecto y llevarla a cabo), Modelado y Animación por Computador (para realizar las tareas de la fase de *Producción*), Gráficos por Computador (para saber resolver los problemas técnicos encontrados en la fase de *Producción*), Imagen y Vídeo por Computador (para escoger los formatos de imagen y vídeo adecuados), las asignaturas de Sonido y Música por Computador, y Técnicas para

el Diseño Sonoro (para explicar el subapartado de *Audio* en el *Estado del Arte > Postproducción*), Proyectos Multimedia (para tener en cuenta los derechos de atribución y gestionar el proyecto) y Postproducción Digital (para explicar y realizar la fase de *Postproducción*).

- Se ha puesto en práctica la **metodología** y planificación explicadas en sus propios capítulos. Siguiéndola, y aunque han surgido algunos impedimentos nombrados más abajo, se ha podido realizar todas las tareas programadas. En el capítulo al final del documento del *Apéndice I: Planificación semanal en Trello* se puede ver la progresión por *sprints* del proyecto.
- También se hizo el análisis de las **herramientas software** que existen para conocer las distintas herramientas, cuáles eran las mejores y de más bajo coste para realizar cada fase del proyecto. Con eso se ha logrado que el **coste final del proyecto sea bajo**, ya que todos los programas que se han usado son gratuitos, y todos los recursos necesarios se han realizado a mano (como las texturas de la app para la pantalla, los bocetos o los materiales de los objetos), es decir, no se han usado recursos de pago.
- Se han realizado **estudios de anatomía y de otros recursos artísticos** que intentan transmitir la misma idea que este proyecto. Al compararlos, se puede decir que este loop animado ha logrado el objetivo. Ya que finalmente se ha creado un GIF de alta calidad. En el que se aprecian los pequeños detalles como que: la pantalla se refleja en los ojos de la gata; en la pantalla del móvil se reflejan todos los objetos (incluidas las partículas); las luces hacen que la escena sea visible claramente y aporta calidad; la animación es fluida y consigue imitar los movimientos naturales de un gato; la estética ha conseguido parecerse a otras con el estilo que se buscaba para que transmita tranquilidad y fuese linda con ayuda de los colores utilizados; se entiende la animación y las acciones del gato; y además, se ha conseguido que la gata sea una versión *cartoon* de la gata real en la que se basa el modelo (Kely la gata de la autora), pareciéndose mucho a ella e imitando sus rasgos más característicos.
- Se han conseguido **resolver los problemas** que surgieron a lo largo del trabajo. Hubo materiales como el cristal de la pantalla de la tableta y el de la córnea del ojo del gato que no se conseguía que tuviesen reflejos, pero finalmente sí pudieron realizarse. El *rigging* y *skinning* hubo que realizarlos 3 veces porque no se conseguía crear el esqueleto de manera correcta y luego porque no funcionaba bien al conectarlo al modelo. El render del vídeo se tuvo que realizar varias veces (con la cantidad de horas que le lleva al ordenador) porque al ver los primeros resultados, no cumplían con lo

deseado. Fue difícil encontrar un programa de edición de vídeo que fuese gratuito y exportase en formato GIF. Por último, como la autora estaba realizando un curso de grado superior a la vez que este trabajo, hubo un periodo de tiempo que ese curso impidió trabajar en este proyecto, pero finalmente se logró realizar para la fecha de entrega.

- Se realizaron a mano todas las tareas necesarias para la fase de **Preproducción**.
- Se realizaron a mano todas las tareas necesarias para la fase de **Producción**. Como se ha mencionado anteriormente, algunas se pusieron en práctica por primera vez en este trabajo.
- Se realizaron a mano todas las tareas necesarias para la fase de **Postproducción**.
- El GIF **se ha compartido** con amigos por redes sociales como WhatsApp y Telegram, los comentarios han sido halagadores. Además, así se ha comprobado que realmente tiene un uso como recurso para las conversaciones de chats.
- Como ya se ha comentado, **este documento recoge toda la investigación** realizada y se explica **todo el proceso** que se ha seguido para dejar constancia de cómo se ha realizado y que se ha necesitado.

A continuación, para que se vean mejor los detalles que se han indicado en los puntos anteriores se muestran una imagen de Kely (la gata en la que se basa el modelo 3D en estilo *cartoon*), y un render que corresponde a uno de los frames de la animación del *loop* animado cuando salen las partículas para que se vea que también se reflejan en la pantalla de la tableta y en los ojos de la gata.



Figura 164. Foto de Kely real para comparar con el resultado final
(Fuente propia: realizado con una cámara)



Figura 165. Muestra de resultado final como imagen con las partículas
(Fuente propia: realizado en [Blender](#))

También cabe mencionar, las horas empleadas en la realización del proyecto con un total de 620 horas. Se ha añadido un apéndice al final del documento, **Apéndice II: Informe de Toggl con las horas empleadas**, donde se muestran las capturas de pantalla del informe de horas en la herramienta Toggl, con el total de las horas empleadas en el proyecto y cuánto se ha

invertido en cada una de las tareas. Aunque el proyecto ha llevado muchas más horas (aproximadamente el doble) de las que comprende la asignatura del Trabajo Final de Grado en Ingeniería Multimedia, este no era un aspecto importante para la autora, ya que se le dio más importancia al hecho de aprender el proceso y lograr realizarlo con un buen acabado.

Para finalizar, aunque no formaba parte del proyecto inicialmente. Para poder realizar las tareas sin problemas y poder recordar todos los pasos para explicarlos en el documento se capturó en vídeo la pantalla cuando se realizaban las tareas de la *Producción*. Como material complementario del trabajo, se subirán estos vídeos en una lista de reproducción de YouTube junto al resultado final para que se pueda ver cómo se ha realizado cada tarea de esta fase, los problemas que han surgido y cómo se solventaron. Los vídeos se podrán ver en el siguiente enlace (clicar en la imagen):

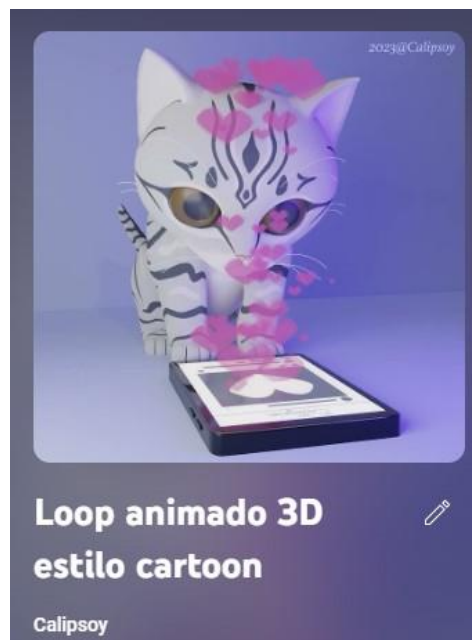


Figura 166. Lista de los vídeos del proceso de Producción con el enlace al vídeo
(Fuente propia: subido a [YouTube](#))

Nota: no se han incluido los vídeo de los intentos de realizar algunas tareas que tuvieron que realizarse de nuevo. Asimismo, se han cortado las partes del vídeo en la que se realizaban búsquedas y sin querer se quedaba grabando la pantalla.

11. Conclusiones y trabajo futuro

Como conclusión, se han logrado alcanzar todos los objetivos del proyecto. Aunque como todo trabajo artístico siempre es mejorable, éste ha servido para practicar los conceptos teóricos e incluso los que no se habían podido realizar a lo largo del grado por el tiempo que tiene cada asignatura para realizarse. A lo largo del proyecto, se ha aprendido mucho sobre el arduo trabajo que lleva realizar una animación. Se han podido poner en práctica y ampliar los conocimientos adquiridos en el grado. Aunque el proyecto ha llevado muchas más horas de las que comprende la asignatura del Trabajo Final de Grado en Ingeniería Multimedia, no ha sido un tema preocupante porque se le daba más importancia al hecho de aprender el proceso y lograr realizarlo con un buen acabado. Ha sido muy satisfactorio llevar a cabo este trabajo logrando los objetivos, y haber podido incluir a Kely en él.

Se ha podido compartir el GIF y obtener *feedback* de la gente cercana. Más adelante habrá que ver cómo reaccionan otras personas. Como trabajo futuro, y que ya se ha mencionado en el apartado de *Resultado final*, como material complementario a este proyecto, se irán subiendo los vídeos que se grabaron realizando las tareas del proceso de *Producción*, en el programa Blender, en [la lista de reproducción](#) que se menciona al final de ese mismo capítulo de este documento. Asimismo, en un futuro, estaría bien volver a realizar la animación del *loop* para ver lo que se ha mejorado, y, por ejemplo, cambiar el estilo general a un acabado más realista creándole pelaje a la gata y aprendiendo sobre cómo se realiza esto en 3D. También se podría realizar una animación más larga donde la protagonista siga siendo la gata y de esta forma practicar más la animación para mejorar este aspecto y que cada vez los movimientos queden más fluidos y naturales. Para seguir mejorando en el futuro, se planea realizar otros proyectos que incluso puede que se dediquen a una sola tarea específica del pipeline 3D para seguir aprendiendo. También, se ha podido añadir este proyecto en el portfolio para el currículum.

Bibliografía

- Aho, A., & Haav, K. (2006). *toggl*. Obtenido de toggl: <https://toggl.com/>
- Allen, E., & Murdock, K. L. (2011). *Body language: advanced 3D character rigging*. Indianapolis: John Wiley & Sons.
- Ambient Design Ltd. (2004). *ArtRage*. Obtenido de ArtRage: <https://www.artrage.com/>
- Asana, T. (16 de Noviembre de 2022). *asana*. Obtenido de Las 12 metodologías más populares para la gestión de proyectos: <https://asana.com/es/resources/project-management-methodologies>
- Atlassian. (2010). *trello*. Obtenido de trello: <https://trello.com/>
- Autodesk, Inc. (1982). *AutoCAD*. Obtenido de Autodesk: <https://www.autodesk.es/products/autocad/overview?term=1-YEAR&tab=subscription>
- Autodesk, Inc. (1990). *Autodesk 3ds Max*. Obtenido de Autodesk: <https://www.autodesk.es/products/3ds-max/overview?term=1-YEAR&tab=subscription>
- Autodesk, Inc., & Alias Systems Corporation. (1998). *Autodesk Maya*. Obtenido de Autodesk: https://www.autodesk.es/products/maya/overview?mktvar002=4417848|SEM|18873799116|142041545246|kwd-298197271104&utm_source=GGL&utm_medium=SEM&utm_campaign=GGL_DEC_Maya_EMEA_ES_eComm_SEM_BR_NA_MIX_0000_4417848_Nurture&utm_id=4417848&utm_term=kwd-29819727110
- Autodesk, Inc., & Skymatter, Inc. (2007). *Mudbox*. Obtenido de Autodesk: <https://www.autodesk.es/products/mudbox/overview?term=1-YEAR&tab=subscription>
- Beane, A. (2012). *3D Animation Essentials*. John Wiley & Sons.
- Blackmagic Design. (2004). *DaVinci Resolve*. Obtenido de Blackmagic Design: <https://www.blackmagicdesign.com/es/products/davinciresolve>
- Bouvier, N., Hull, G., Jones, A., Antonov, V., & Wade, D. P. (2006). *Concept Art: d'artiste. Digital Artist Master Class*. Bulgaria: Ballistic.

- Branham, R. (2017). *Houdini Fast Track Vol 1: Fundamentals*. Los Ángeles: Gnomon Workshop.
- Burnet, J. (1880). *Practical hints on composition in painting*. No. 187.
- Carrasco, E. (17 de Noviembre de 2022). *animum3d*. Obtenido de animum3d: <https://www.animum3d.com/blog/que-debemos-saber-sobre-el-estilo-cartoon/#:~:text=Este%20es%20el%20estilo%20que,exagerados%20que%20delatan%20osu%20personalidad>
- CEI: Escuela Diseño y Marketing. (14 de Mayo de 2021). *cei*. Obtenido de cei.es: <https://cei.es/vfx-que-es/#:~:text=%C2%BFQu%C3%A9%20es%20el%20VFX%3F,la%20etapa%20de%20postproducci%C3%B3n%20cinematogr%C3%A1fica>.
- Celsys, Inc. (2001). *Clip Studio Paint*. Obtenido de Clip Studio: <https://www.clipstudio.net/es/>
- CLO Virtual Fashion Inc. (2019). *Marvelous Designer*. Obtenido de Marvelous Designer: <https://marvelousdesigner.com/>
- Cohn, M. (2010). *Succeeding with agile: software development using Scrum*. Pearson Education.
- Constant, I. (2020). *domestika*. Obtenido de domestika.org: <https://www.domestika.org/es/blog/4241-motion-graphics-que-es-y-para-que-sirve>
- Corel Corporation, Zimmer, M., & Hedges, T. (1991). *Corel Painter*. Obtenido de painterartist: https://www.painterartist.com/en/product/painter/?sourceid=ptr2022-xx-ppc_brkws&x-vehicle=ppc_brkws&trial=false&utm_source=google&utm_medium=cpc&utm_campaign=pt-dd-all-adwordspc&utm_content=144004202365&utm_term=corel%20painter&utm_id=15550553000&gclid=C
- Dannenberg, R., Mazzoni, D., & Audacity Team. (2000). *Audacity*. Obtenido de Audacity: <https://www.audacityteam.org/>
- Davia Aracil, M. (2018). Diapositivas de la asignatura Modelado y Animación por Computador. San Vicente del Raspeig, Alicante, España: 2º Curso del Grado Ingeniería Multimedia en la Universidad de Alicante.
- Deemer, P., Benefield, G., Larman, C., & Vodde, B. (2010). *The SCRUM Primer*. Scrum Training Institute.

- Equipo de Documentación de Blender. (17 de Mayo de 2023). *Manual de usuario de Blender*. Obtenido de blender.org: <http://docs.blender.org>
- Escalpés Jover, F. J. (2016). Diapositivas de la asignatura Fundamentos del Diseño Gráfico. San Vicente del Raspeig, Alicante, España: 1er Curso del Grado Ingeniería Multimedia en la Universidad de Alicante.
- Escuela Trazos, & @Trazos. (30 de Enero de 2023). *Escuela Trazos*. Obtenido de trazos.net: <https://trazos.net/diferencias-entre-etalonaje-y-correccion-de-color/#:~:text=en%20la%20postproducci%C3%B3n,-,%C2%BFQu%C3%A9%20es%20la%20correcci%C3%B3n%20de%20color%3F,se%20vea%20lo%20mejor%20posible.>
- Foundry, & Roper Technologies, Inc. (2010). *Mari*. Obtenido de Foundry: <https://www.fox.com/products/mari>
- Fundación Blender. (1994). *Blender*. Obtenido de Blender: <https://www.blender.org/>
- GR, M., DonBlasP, Mansoncc, Isegar, SeroBOT, Aldair, . . . Varios Colaboradores. (21 de Mayo de 2023). *wikipedia*. Obtenido de wikipedia: <https://es.wikipedia.org/wiki/Animaci%C3%B3n>
- historiadelcine.es. (12 de Abril de 2023). *historiadelcine*. Obtenido de historiadelcine: <https://historiadelcine.es/glosario-terminos-cinematograficos/tipos-planos-en-cine/>
- Idyllic Pixel, Natascha, & Viktor. (2012). *PureRef*. Obtenido de PureRef: <https://www.pureref.com/>
- Iñesta Quereda, J. M., & Ponce de León Amador, P. J. (2019). Temario de la asignatura Técnicas para el Diseño Sonoro. San Vicente del Raspeig, Alicante, España: 4º Curso del Grado Ingeniería Multimedia en la Universidad de Alicante.
- Johnston, O., & Thomas, F. (1995). *The illusion of life: Disney animation*. New York: Hyperion.
- Keller, E., Palamar, T., & Honn, A. (2010). *Mastering Autodesk Maya 2011*. United Kingdom: John Wiley & Sons.
- Kimball, S., Mattis, P., Natterer, M., Jehan, & Varios Colaboradores. (1996). *GIMP*. Obtenido de GIMP: <https://www.gimp.org/>

- Lasseter, J. (Agosto de 1987). *Principles of traditional animation applied to 3D computer animation*. Proceedings of the 14th annual conference on Computer graphics and interactive techniques (pp. 35-44).
- Luhta, E., & Roy, K. (2013). *How to Cheat in Maya 2013: Tools and Techniques for Character Animation*. United States: Taylor & Francis.
- Luque Gálvez, M. (2010). *Conceptos y fundamentos de diseño en 3D*. Antequera: IC Editorial.
- Márquez Ruiz, A. (2016). Temario de la asignatura Fundamentos de la Física. San Vicente del Raspeig, Alicante, España: 1er Curso del Grado Ingeniería Multimedia en la Universidad de Alicante.
- Maxon Computer GmbH. (1990). *Cinema 4D*. Obtenido de Maxon: <https://www.maxon.net/es/cinema-4d>
- Maxon Computer GmbH, & Pixologic. (1999). *ZBrush*. Obtenido de Maxon: <https://www.maxon.net/es/zbrush>
- Mayorga, A., SeroBot, Charlyfar, Bombay, C., Santas20, SuperJoe, . . . Varios Colaboradores. (10 de Junio de 2023). *wikipedia*. Obtenido de wikipedia: https://es.wikipedia.org/wiki/Dibujo_animado
- McDermott, W. (2018). *The PBR guide: A handbook for physically based rendering*. United Kingdom: Allegorithmic.
- Meno25, Violetwtf, Malikshorts, Anne, B., Seeworldcefe000, Apparition11, . . . Varios Colaboradores. (25 de Abril de 2023). *science-things*. Obtenido de science-things: https://www.science-things.com/terms/3d_computer_graphics.htm
- Microsoft Corporation. (1983). *Word*. Obtenido de Microsoft: <https://www.microsoft.com/es-es/microsoft-365/word?activetab=tabs%3afaqheaderregion3>
- Microsoft Corporation. (1985). *Excel*. Obtenido de Microsoft: <https://www.microsoft.com/es-es/microsoft-365/excel>
- Mora, D. (26 de Noviembre de 2021). *monstruosdeldiseño*. Obtenido de monstruosdeldiseño: <https://monstruosdeldisenio.com/podcast/tipos-tecnicas-modelado-3d>
- Murdock, K. (2018). *Autodesk Maya 2019 Basics Guide*. United States: SDC Publications.

- Narayan, S., Belsky, S., Chakravarthy, A., Chen, G., Durn, D., Rao, D., & Wadhvani, D. (1987). *Adobe Illustrator*. Obtenido de Adobe: https://www.adobe.com/es/products/illustrator.html?gclid=Cj0KCQjwnf-kBhCnARIsAFIlg491bVILBmM1_BuNt7s1Vy645SmLV_qIEOoGQIHsok98jJ4vYGBYZJh0aA piPEALw_wcB&mv=search&mv=search&sdid=KCJMVLf6&ef_id=Cj0KCQjwnf-kBhCnARIsAFIlg491bVILBmM1_BuNt7s1Vy645SmLV_qIEOoGQIHsok
- Narayan, S., Belsky, S., Chakravarthy, A., Chen, G., Durn, D., Rao, D., & Wadhvani, D. (1990). *Adobe Photoshop*. Obtenido de Adobe: https://www.adobe.com/es/products/photoshop/landpb.html?gclid=Cj0KCQjwnf-kBhCnARIsAFIlg493M91g8BsG451FOJfI-0HoQ9vAzG9c7q7OFVqEz9tL9FvAwNV-Ew9caAsEnEALw_wcB&mv=search&s_kwid=AL!3085!3!441887029669!e!!g!!adobe%20photoshop!1445901735!56657232416&mv=search&sd
- Narayan, S., Belsky, S., Chakravarthy, A., Chen, G., Durn, D., Rao, D., & Wadhvani, D. (1993). *Adobe After Effects*. Obtenido de Adobe: https://www.adobe.com/es/products/aftereffects/landpb.html?gclid=Cj0KCQjwnf-kBhCnARIsAFIlg492j-irMaH-WEp0z1ykgCKgiBmBApKq0Z0egWsW97KmWCZZO9bnRby8aAkhNEALw_wcB&mv=search&mv=search&sdid=MYYBRYZH&ef_id=Cj0KCQjwnf-kBhCnARIsAFIlg492j-irMaH-WEp0z1ykgCKgiBmBApKq0Z
- Narayan, S., Belsky, S., Chakravarthy, A., Chen, G., Durn, D., Rao, D., & Wadhvani, D. (2003). *Adobe Audition*. Obtenido de Adobe: https://www.adobe.com/es/products/audition.html?gclid=Cj0KCQjwnf-kBhCnARIsAFIlg4926rhxBp7vL4GHJ3NrS9ZkfxS-1J_fJCQpm2ANyNBTSIVx-YYd26U0aAjDVEALw_wcB&mv=search&mv=search&sdid=MC95SLNN&ef_id=Cj0KCQjwnf-kBhCnARIsAFIlg4926rhxBp7vL4GHJ3NrS9ZkfxS-1J_fJCQpm2ANyNBTS
- Narayan, S., Belsky, S., Chakravarthy, A., Chen, G., Durn, D., Rao, D., & Wadhvani, D. (2003). *Adobe Premiere Pro*. Obtenido de Adobe: https://www.adobe.com/es/products/premiere.html?gclid=Cj0KCQjwnf-kBhCnARIsAFIlg491mNEJvTsa2Elmbd0HjSmOtt6iA2ztul2b_UTvAgI5x4E8_IkA58RYaAusvEALw_wcB&mv=search&mv=search&sdid=LQLZT7BT&ef_id=Cj0KCQjwnf-kBhCnARIsAFIlg491mNEJvTsa2Elmbd0HjSmOtt6iA2ztul2b_UTvAgI5x
- Narayan, S., Belsky, S., Chakravarthy, A., Chen, G., Durn, D., Rao, D., & Wadhvani, D. (s.f.). *Adobe Express*. Obtenido de Adobe: <https://www.adobe.com/es/express/>

- Narayan, S., Belsky, S., Chakravarthy, A., Chen, G., Durn, D., Rao, D., & Wadhvani, D. (s.f.). *Adobe Substance 3D Painter*. Obtenido de Adobe: <https://www.adobe.com/es/products/substance3d-painter.html>
- O'Hailey, T. (2018). *Rig it right! Maya animation rigging concepts*. United States: Routledge.
- Ortiz Zamora, J. (2019). Diapositivas de la asignatura Postproducción Digital. San Vicente del Raspeig, Alicante, España: 4º Curso del Grado Ingeniería Multimedia en la Universidad de Alicante.
- Owens, M., de Hessel, R., Neuhauser, J., Bah, T., Jeanmougin, M., Gould, T., . . . Hurst, N. (2006). *inkscape*. Obtenido de inkscape: <https://inkscape.org/es/>
- Pérez, L. F. (22 de Diciembre de 2018). *aprendercine*. Obtenido de aprendercine.com: <https://aprendercine.com/escena-y-secuencia-diferencias/>
- PGN Corp. (2000). *openCanvas*. Obtenido de portalgraphics: <https://portalgraphics.net/en/oc/>
- Puchol García, J. A. (2018). Diapositivas de la asignatura Gráficos por Computador. San Vicente del Raspeig, Alicante, España: 3er Curso del Grado Ingeniería Multimedia en la Universidad de Alicante.
- Reeves, W. T. (1983). *Particle systems—a technique for modeling a class of fuzzy objects*. ACM Transactions On Graphics (TOG), 2(2), 91-108.
- Robert McNeel & Associates. (1993). *Rhinoceros 3D*. Obtenido de Rhino3D: <https://www.rhino3d.com/es/>
- Rodríguez, H. (20 de Mayo de 2021). *crehana*. Obtenido de crehana: <https://www.crehana.com/blog/estilo-vida/como-dibujar-cartoon/>
- S.A. , A. (22 de Febrero de 2023). *compromiso.atresmedia*. Obtenido de compromiso.atresmedia: <https://compromiso.atresmedia.com/levanta-la-cabeza/>
- Sale, T., & Betti, C. (2007). *Drawing: A contemporary approach*. Cengage Learning.
- Schwaber, K. (2004). *Agile project management with Scrum*. Microsoft press.
- Sgia, M., Bronard, L., Josefina, Idict, W., Ciget.cfg, M., Cinformmtz, J., . . . Pinar. (3 de Diciembre de 2022). *ecured*. Obtenido de ecured: https://www.ecured.cu/Gr%C3%A1ficos_3D_por_computadora#Tesselation_y_Mallas
- Side Effects Software Inc. (1996). *Houdini*. Obtenido de SideFX: <https://www.sidefx.com/>

- The Krita Team. (2005). *Krita*. Obtenido de Krita: <https://krita.org/es/>
- Tillman, B. (2012). *Creative character design*. Crc Press.
- TobyForTabs. (2016). *Toby Mini*. Obtenido de Chrome Web Store: <https://chrome.google.com/webstore/detail/toby-mini/gfcdgfhkelkdmglklfbndgopaihmoeci>
- Torres, A. (31 de Marzo de 2021). *esdesignbarcelona*. Obtenido de esdesignbarcelona: <https://www.esdesignbarcelona.com/actualidad/motion-design/como-crear-un-loop-o-bucle-con-after-effects#:~:text=Un%20loop%20o%20bucle%20es,de%20los%20efectos%20m%C3%A1s%20frecuentes>
- Toudou. (2017). *Substance Painter for Blender User*. United Kingdom: Amazon Kindle Books.
- Trimble, Last Software, & Google. (2000). *SketchUp*. Obtenido de SketchUp: <https://www.sketchup.com/es>
- Vázquez Pérez, S. (2018). Temario de la asignatura Análisis y Especificación de Sistemas Multimedia. San Vicente del Raspeig, Alicante, España: 2º Curso del Grado Ingeniería Multimedia en la Universidad de Alicante.
- Vista Oftalmólogos. (3 de Agosto de 2019). *vistaoftalmologos*. Obtenido de vistaoftalmologos: <https://www.vistaoftalmologos.es/la-persistencia-retiniana/#:~:text=La%20persistencia%20retiniana%20permite%20que,de%20im%C3%A1genes%20est%C3%A1ticas%20e%20independientes>
- Wondershare Software Ltd. (2019). *Wondershare Filmora*. Obtenido de Wondershare Filmora: <https://filmora.wondershare.net/>

Recursos externos utilizados en el desarrollo del *loop*

Para el trabajo final **no se utilizaron recursos externos**, todo se realizó a mano por la autora. Pero en la fase de preproducción, para hacer el moodboard se realizó una búsqueda para obtener **imágenes de referencia** que se usaron para fijarse en ellas como **inspiración** o, por ejemplo, el **estudio anatómico**. En este apartado se incluyen las **URL** de esas **imágenes** separadas en los mismos bloques que se encuentran en el **moodboard**:

- **Kely.** Las imágenes de este bloque son de **fuentes propias** realizadas con una cámara a la gata real en la que se basa el personaje principal de la animación, que además fue la mascota de la autora.
- **Poses Gatos.** Se usaron para fijarse en las formas de un gato bebé.
 - Fuente imagen 1: <https://stock.adobe.com/es/images/baby-british-shorthair-kitten-sitting-in-profile-and-looking-up/69533287>
 - Fuente imagen 2: https://es.123rf.com/photo_25596881_gatito-escoc%C3%A9s-en-el-perfil-de-aislados-en-fondo-blanco.html
 - Fuente imagen 3: https://es.123rf.com/photo_24497272_gatito-brit%C3%A9nico-de-pelo-corto-se-coloca-en-perfil-aislado-en-blanco.html
 - Fuente imagen 4: https://es.123rf.com/photo_33270965_peque%C3%B1o-gatito-de-pie-delante-aislado-en-fondo-blanco.html
 - Fuente imagen 5: https://es.123rf.com/photo_37101279_escoc%C3%A9s-del-gatito-reci%C3%A9n-nacido-lindo-que-se-coloca-en-el-frente-aislado-en-fondo-blanco.html
- **Anatomía Gatos.** Estas imágenes se usaron para hacer el estudio anatómico de un gato.
 - Fuente imagen 1: <https://www.hepper.com/es/cuantos-musculos-tiene-un-gato/>
 - Fuente imagen 2: <https://clinicalcharts.com/products/cat-skeleton-anatomy-poster.html>
 - Fuente imagen 3: <https://twitter.com/ElGatoCarey/status/1181715215601340416>
- **GIF Gatos Viendo Tabletas.** Estas imágenes en movimiento se usaron para fijarse en cómo gatos reales se comportaban delante de las pantallas de los dispositivos y ver sus movimientos.
 - Fuente imagen en movimiento 1: <https://tenor.com/es/view/texting-cat-phone-phone-gif-7305828>
 - Fuente imagen en movimiento 2: <https://tenor.com/es/view/cat-playing-phone-video-games-cute-cat-gif-11850115>
 - Fuente imagen en movimiento 3: <https://www.pinterest.es/pin/546694842248966468/>
 - Fuente imagen en movimiento 4: <https://tenor.com/es/view/smartphone-cat-kitten-katze-telephone-gif-5822569>
- **GIF Animación App.** Estas imágenes en movimiento se usaron como referencias para estudiar la animación del movimiento de los elementos de la app en un dispositivo cuando alguien navega por ellas.

- Fuente imagen en movimiento 1: <https://slate.com/technology/2018/05/iphone-x-android-p-gestures-start-new-era-of-smartphone-operations.html>
- Fuente imagen en movimiento 2: <https://uxdesign.cc/how-and-should-we-stop-the-infinite-scroll-66141fcb0768>
- Fuente imagen en movimiento 3: <https://wifflegif.com/gifs/496989-transformation-animaterial-gif>
- Fuente imagen en movimiento 4: https://www.behance.net/gallery/121212685/Various-Gif-Animations?locale=es_ES
- **Referencias Estilo Gata.** Estas imágenes se usaron para buscar inspiración para el estilo que se usaría para ilustrar a la gata en las referencias y los *concept art*.
 - Fuente propia imagen 1: realizada con [Adobe Illustrator](#).
 - Fuente imagen 2: <https://www.creativefabrica.com/es/product/kawaii-cats-clipart/>
 - Fuente imagen 3: <https://www.pinterest.es/pin/821695894507050363/>
 - Fuente imagen 4: <https://www.pinterest.es/pin/567875834275699227/>
 - Fuente imagen 5: <https://www.istockphoto.com/es/vector/conjunto-de-lindos-gatos-estilo-kawaii-colecci%C3%B3n-de-gatitos-encantadores-en-gm1397764004-452095582>
 - Fuente imagen 6: <https://www.redbubble.com/es/i/cuaderno/Torre-de-gatitos-de-Flakey/43482689.RXH2R>
 - Fuente imagen 7: <https://www.pinterest.es/pin/217932069444730561/>
 - Fuente imagen 8: <https://www.etsy.com/es/listing/1225674485/cute-cats-clipart-cat-png-cute-chonky>
 - Fuente imagen 9: <https://www.vecteezy.com/vector-art/3808428-cute-happy-kawaii-cat>
- **Modelos Gatos 3D.** Estas imágenes se usaron para inspirarse en los acabados que se le podrían dar al estilo de la gata cuando se pasase a las 3 dimensiones.
 - Fuente imagen 1: https://www.freepik.es/fotos-premium/gato-ojos-grandes-ojos-grandes-sienta-sobre-superficie-blanca_42062217.htm
 - Fuente imagen 2: https://stock.adobe.com/images/white-cat-laying-down-with-its-eyes-closed.-generative-ai/568794459?as_campaign=TinEye&as_content=tineye_match&clickref=1100lwWkdJQZ&mv=affiliate&mv2=pz&as_camptype=backlink&as_channel=affiliate&as_source=partnerize&as_campaign=tineye
 - Fuente imagen 3: <https://www.pinterest.es/pin/780248704199632935/>
 - Fuente imagen 4: <https://es.dreamstime.com/lindo-gatito-naranja-d-con-ojos-grandes-trazo-rayado-image266043745>
 - Fuente imagen 5: https://www.freepik.com/premium-photo/adorable-cute-chubby-cat-3d-render_37489858.htm#query=cute%20cat%203d&position=30&from_view=search&track=ais
 - Fuente imagen 6: <https://www.ebay.es/itm/124156315050>
 - Fuente imagen 7: <https://www.kosiaka.co/figura-de-gato-preocupado>
- **Móviles y Tablet 3D.** Estas imágenes se usaron como referencia para ver modelos 3D de distintos dispositivos y ver sus acabados.

- Fuente imagen 1: <https://free3d.com/es/modelo-3d/cellular-9829.html>
 - Fuente imagen 2: <https://3dlancer.net/es/freemodels/telefonos-moviles/3dmodel-telefono-celular-33587>
 - Fuente imagen 3: <https://digitaldeleon.com/ciencia-y-tecnologia/recomendacion-app-movil/>
 - Fuente imagen 4: <https://www.cgtrader.com/free-3d-models/electronics/computer/tablet--4>
 - Fuente imagen 5: <https://es.3dexport.com/3dmodel-tablet-85783.htm>
 - Fuente imagen 6: <https://es.3dexport.com/3dmodel-digital-tablet-with-case-mock-up-02-266359.htm>
- **GIF Referencias Gatos con Dispositivos.** Estas imágenes en movimiento se usaron como estudio de mercado de proyectos parecidos que hubiera ya en la web. Sirvieron para saber: qué había hasta ahora, qué funcionaba (gustaba y quedaba bien) y qué no, qué podría aportar nuevo este proyecto y cómo se habían solventado algunas dudas de movimientos que pudieran surgir.
 - Fuente imagen en movimiento 1: <https://www.pinterest.es/pin/712061391085797612/>
 - Fuente imagen en movimiento 2: <https://ar.pinterest.com/pin/772437773566743453/>
 - Fuente imagen en movimiento 3: <https://www.pinterest.es/pin/310678074307922076/>
 - Fuente imagen en movimiento 4: <https://gifcop.com/cute-cat-gif/>

Lista de Acrónimos y Abreviaturas

Los acrónimos y abreviaturas que se usan a lo largo del documento se muestran en la siguiente tabla ordenada alfabéticamente:

Tabla 1. Lista de Acrónimos y Abreviaturas empleados en el documento

2D	2 Dimensiones
3D	3 Dimensiones
4D	4 Dimensiones
ABP	Aprendizaje Basado en Proyectos
AVI	Audio Video Interleave
CAD	Computer-Aided Design
CGI	Computer Generated Imagery
CMYK	Cyan Magenta Yellow Black
CV	Currículum Vitae
ESO	Educación Secundaria Obligatoria
EULA	End-User License Agreement
FK	Forwardk Kinematics
FPS	Fotogramas por Segundo
FX	Efectos especiales
GIF	Graphics Interchange Format
IK	Inverse Kinematics
JPG/JPEG	Joint Photographic Experts Group
PBR	Physically Based Rendering
PDF	Portable Document Format
PNG	Portable Network Graphics
ppp	Píxeles por pulgada
PSB	Photoshop Large Document
PSD	Photoshop Document
RGB	Red Green Blue
RGBA	Red Green Blue Alpha
Rig	Rigging
RRSS	Redes Sociales
SFX	Efectos sonoros

TFG	Trabajo Fin de Grado
TGA	Truevision Graphics Adapter
TIFF	Tagged Image File Format
TV	Televisión
VFX	Efectos visuales

Apéndice I: Planificación semanal en Trello

A lo largo de todo el trabajo, se hizo un **seguimiento de la planificación en Trello**. Para ello se realizaban **capturas de pantalla** de la herramienta **al inicio y al final de cada semana** en la que se había trabajado en el proyecto. Hubo un periodo de tiempo entre abril y junio que no se trabajó en este proyecto porque se estaba realizando otro trabajo final para un grado medio que estaba realizando la autora.

En las siguientes capturas de pantalla muestran en la columna *Lista de tareas* las tareas que quedaban por hacer en cada momento del proyecto. En la columna *Sprint Semanal*, el inicio de la planificación de tareas semanal para la semana en cuestión. En la columna *En desarrollo* las que estaban en ese estado en ese momento. Finalmente, qué tareas se pudieron acabar en esa semana, en la columna *Terminado Sprint Semanal*. Luego estas tareas finalizadas, para el comienzo de la siguiente semana se pasaban a la columna de *Terminado Proyecto* como se explica en el capítulo de *Metodología* de este documento.

Además, hay una columna al principio, *Tareas nuevas (no contempladas)*, en las que aparecen escritas las tareas que no se habían contemplado al principio de la planificación y se añadieron después en el trabajo; y otra al final, *Impedimentos*, esta sirvió para ir apuntando todo lo que había producido un retraso en el trabajo o que había originado problemas que fueron más difícil de resolver en un momento dado.

Sprint 1: semana del 20 al 26 de marzo de 2023. En las siguientes figuras, se muestra: en la primera captura las tareas que se planificaron realizar en esa semana; y en la segunda, al final de la semana se lograron realizar todas las tareas propuestas.

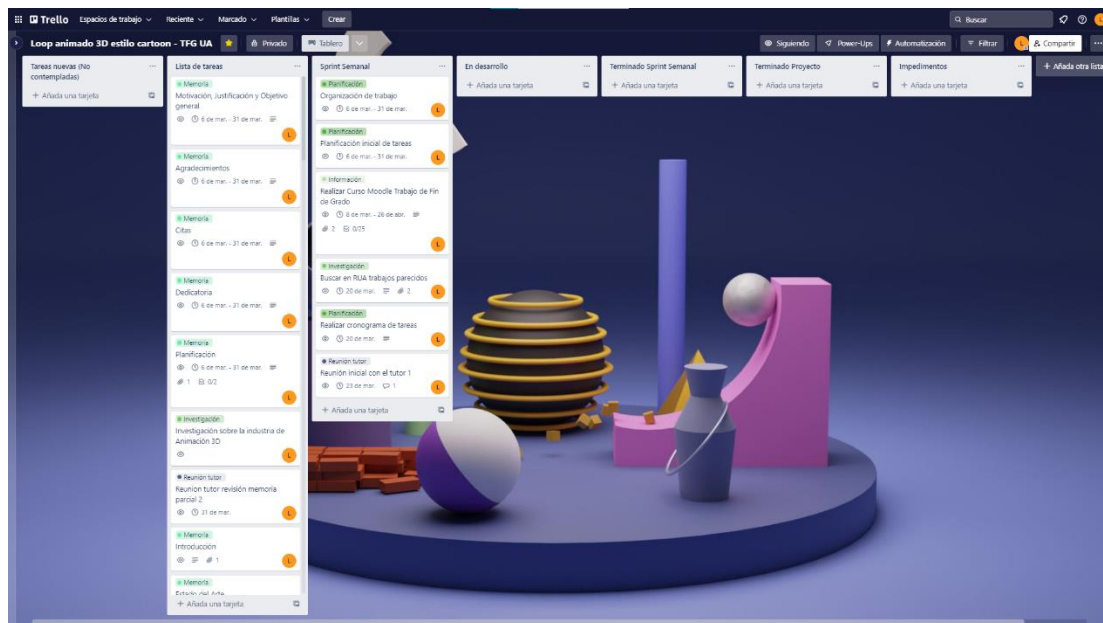


Figura 167. Inicio de la planificación del sprint 1 en Trello
(Fuente propia: realizado en [Trello](#))

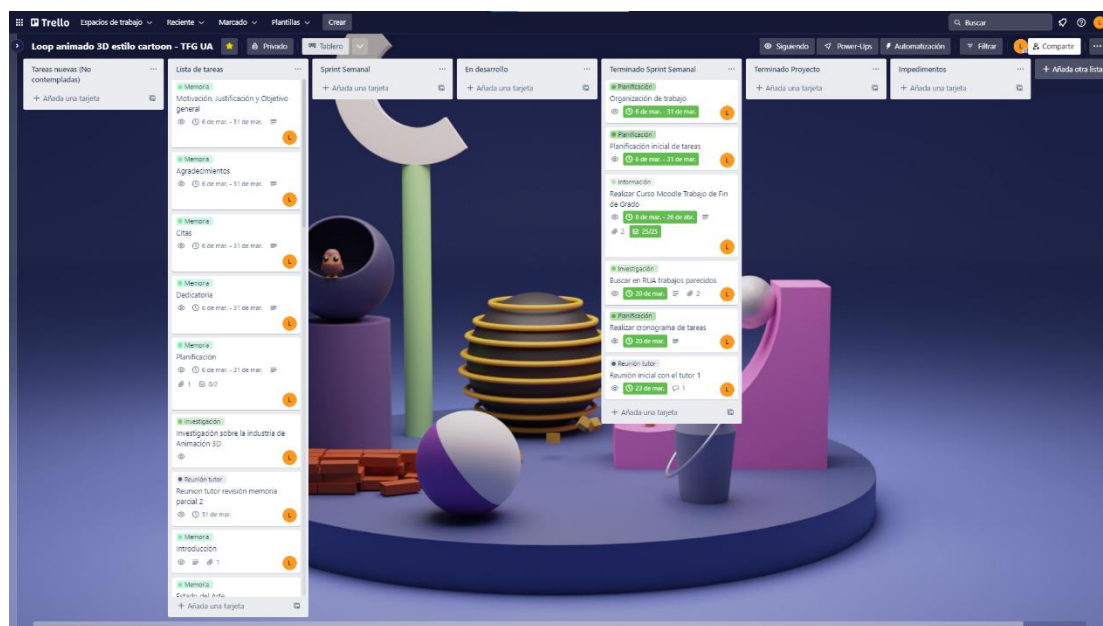


Figura 168. Fin de la planificación del sprint 1 en Trello
(Fuente propia: realizado en [Trello](#))

Sprint 2: semana del 27 de marzo al 2 de abril de 2023. En las siguientes figuras, se muestra: en la primera captura las tareas que se planificaron realizar en esa semana y las que ya estaban terminadas de la semana anterior; y en la segunda, al final de la semana se lograron realizar todas las tareas propuestas menos una que se quedó en proceso.

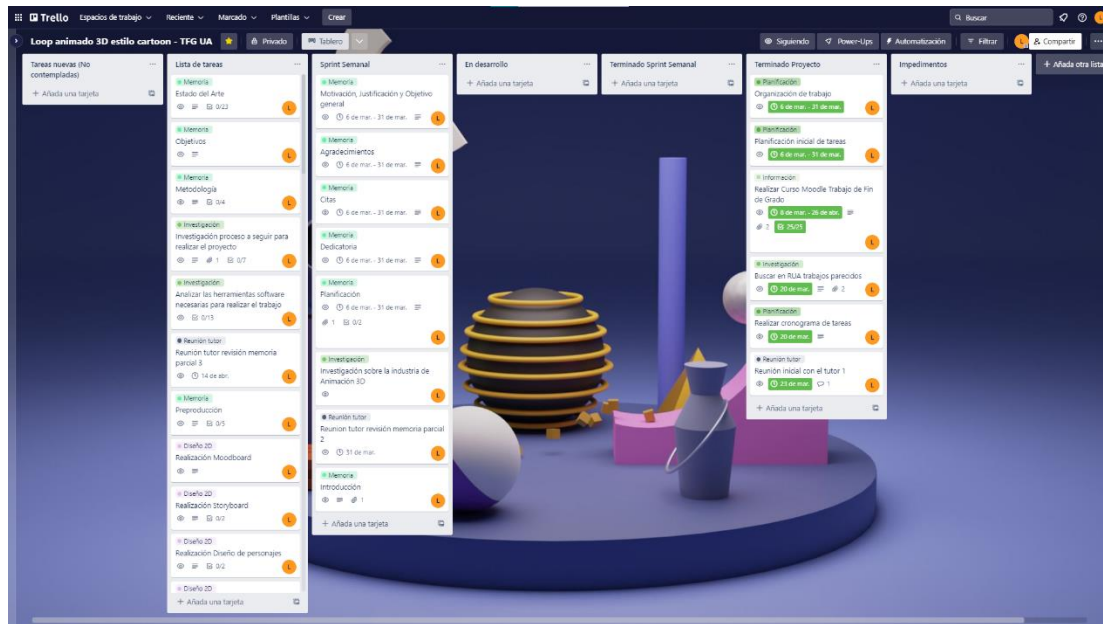


Figura 169. Inicio de la planificación del sprint 2 en Trello
(Fuente propia: realizado en [Trello](#))

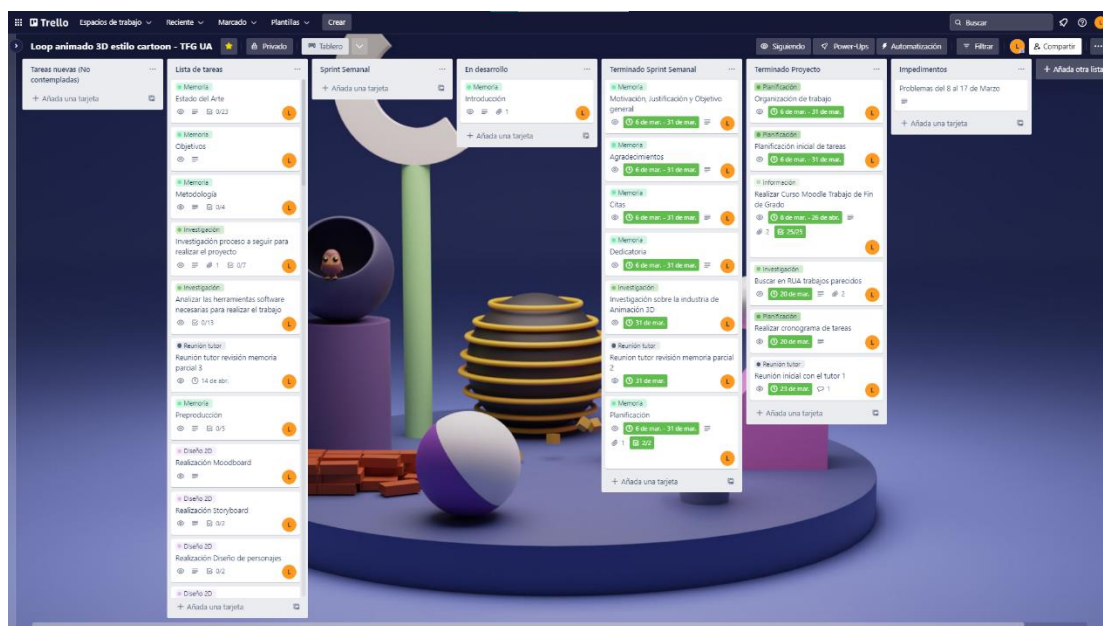


Figura 170. Fin de la planificación del sprint 2 en Trello
(Fuente propia: realizado en [Trello](#))

Sprint 3: semana del 3 al 9 de abril de 2023. En las siguientes figuras se muestra: la primera captura las tareas que se planificaron realizar en esa semana; y en la segunda, al final de la semana se lograron realizar todas las tareas propuestas.

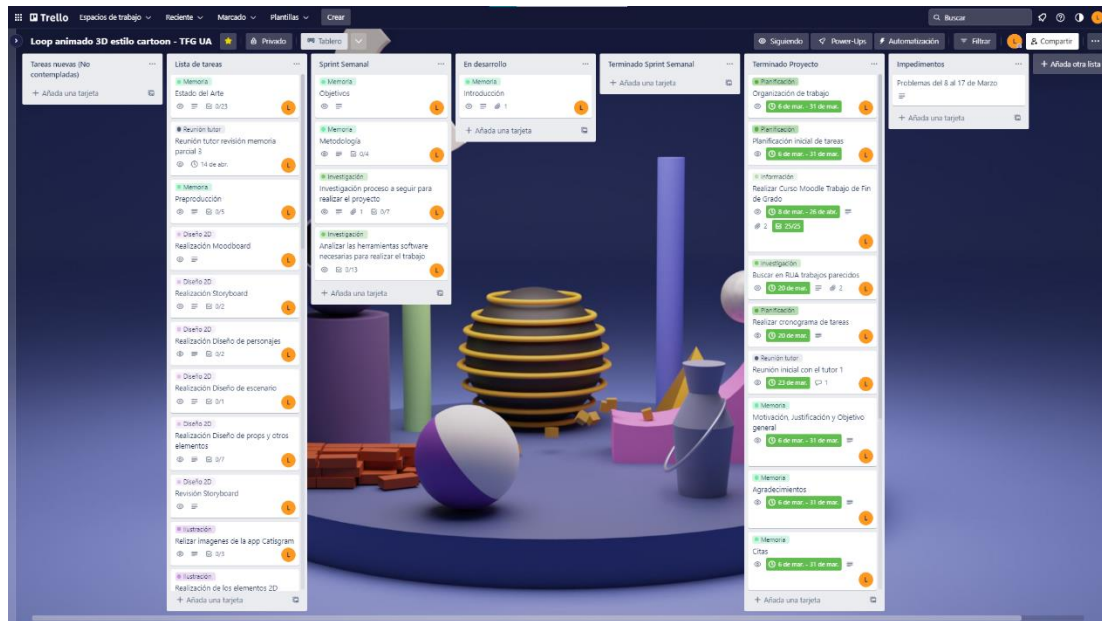


Figura 171. Inicio de la planificación del sprint 3 en Trello
(Fuente propia: realizado en [Trello](#))

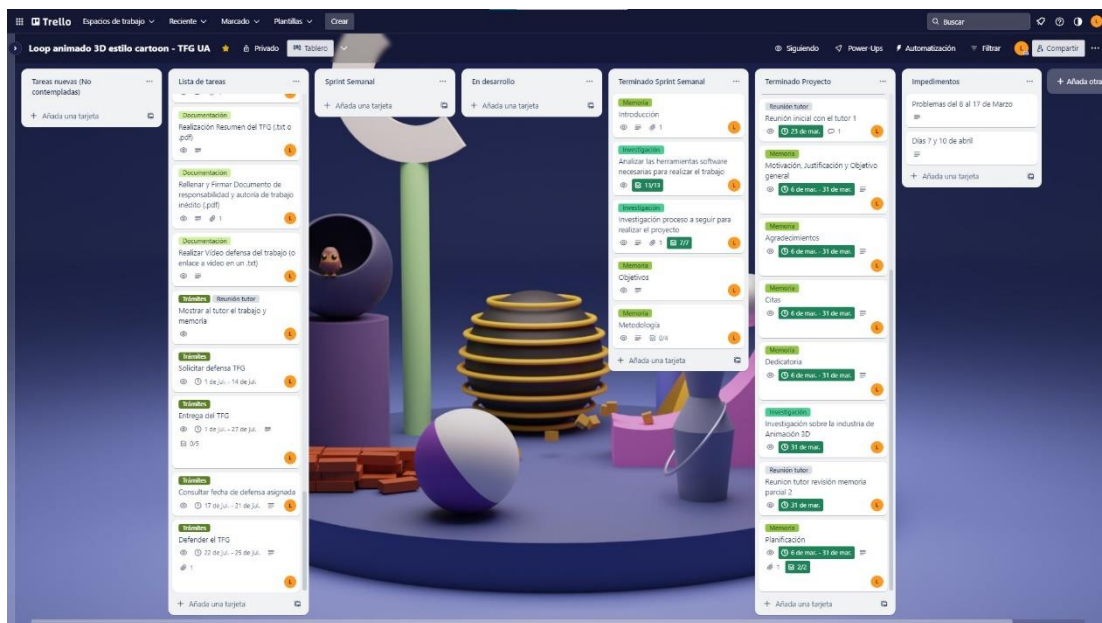


Figura 172. Fin de la planificación del sprint 3 en Trello
(Fuente propia: realizado en [Trello](#))

Después de este sprint hubo un parón en el proyecto que duró del 10 de abril al 11 de junio. Porque se estuvo realizando el trabajo final para el curso superior que se estaba cursando al mismo tiempo.

Sprint 4: semana del 12 al 18 de junio de 2023. En las siguientes figuras se muestra: en la primera captura las tareas que se planificaron realizar en esa semana; y en la segunda, al final de la semana, las que se comenzaron, pero no se lograron acabar para este *sprint*, porque en la

planificación inicial del *sprint* se calculó mal el tiempo que llevarían esas tareas. Por eso, en los siguientes *sprints* hubo que invertir tiempo hasta conseguir terminarlas.

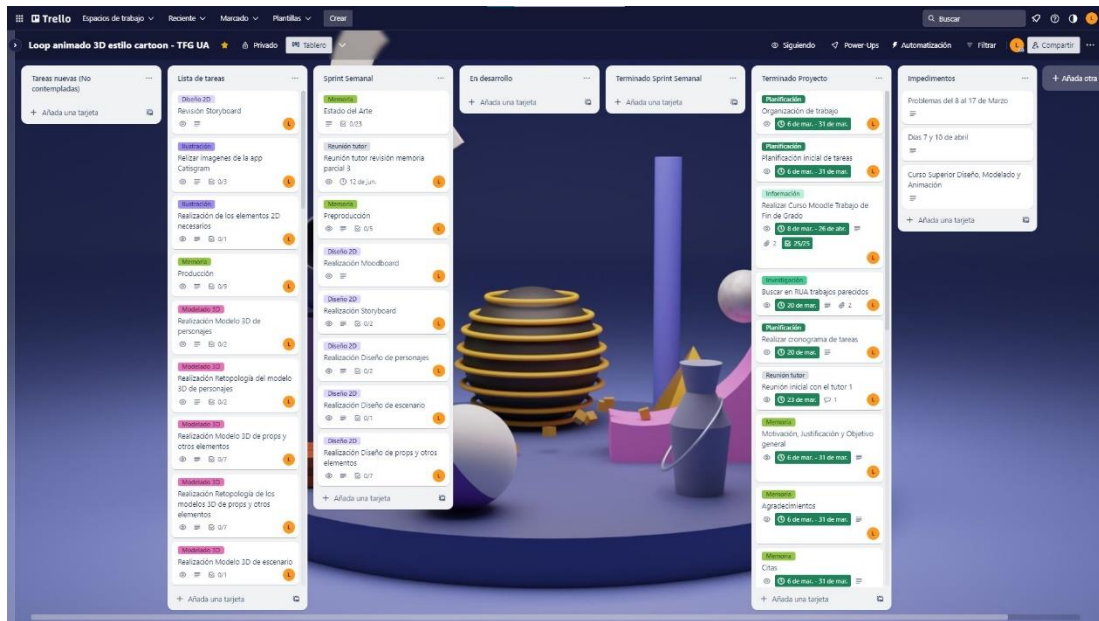


Figura 173. Inicio de la planificación del *sprint* 4 en Trello
(Fuente propia: realizado en [Trello](#))

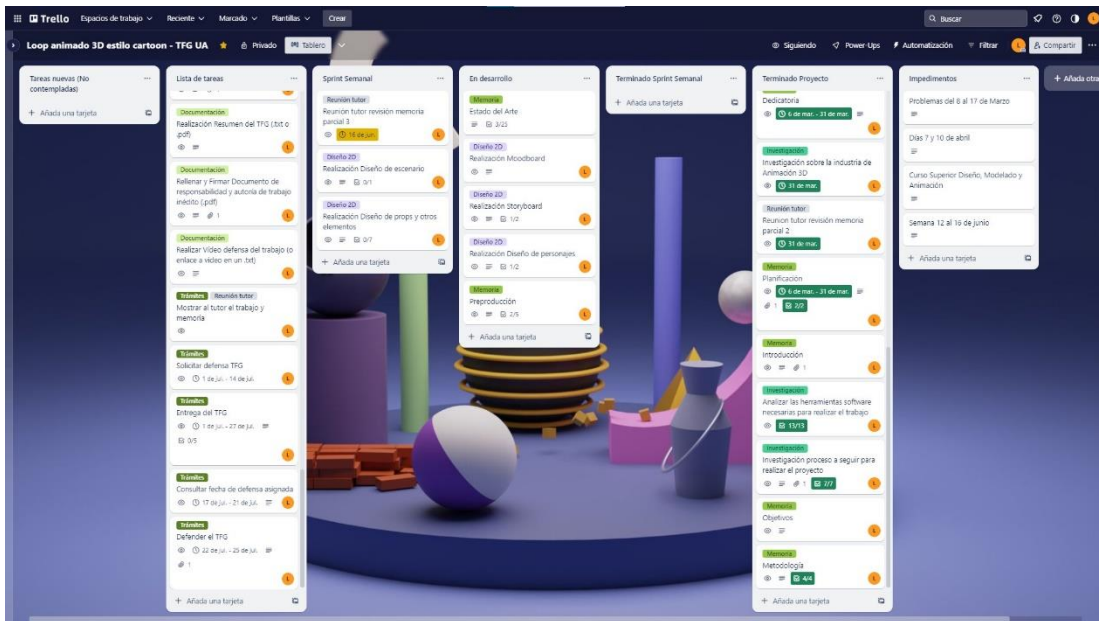


Figura 174. Fin de la planificación del *sprint* 4 en Trello
(Fuente propia: realizado en [Trello](#))

Sprint 5: semana del 19 al 25 de junio de 2023. Se mantuvo la planificación de la semana anterior para avanzar esas tareas que duraban más de lo esperado. En las siguientes figuras, se muestra: en la primera captura las tareas que se planificaron realizar en esa semana (que era la que se

arrastraba del *sprint* anterior); y en la segunda, al final de la semana, aunque no se logró terminar ninguna tarea, sí que se había conseguido avanzar mucho en esas tareas.

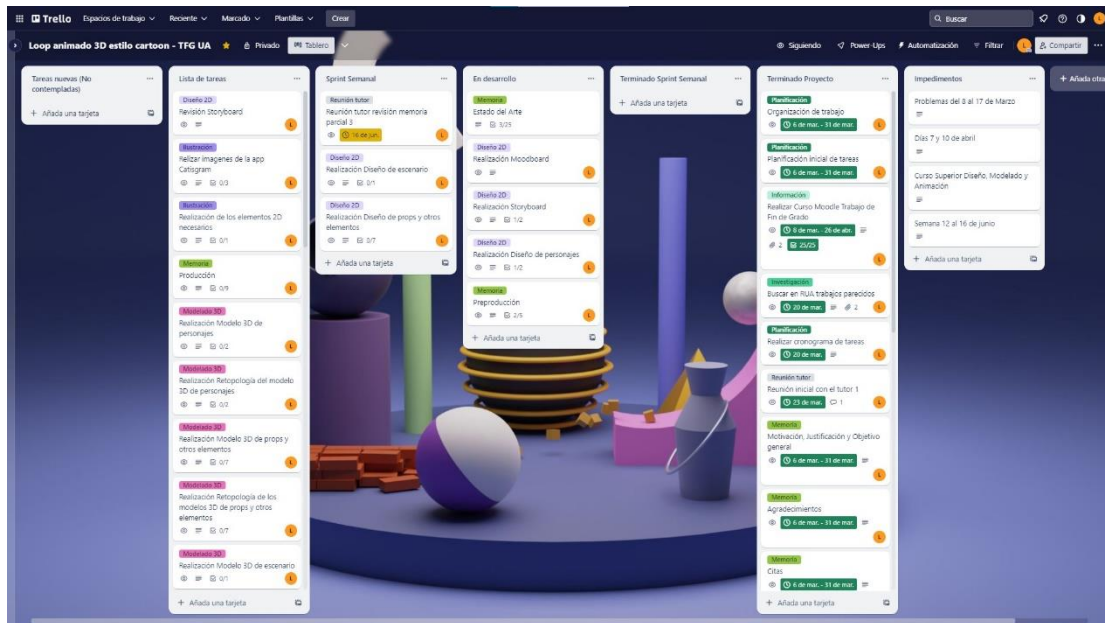


Figura 175. Inicio de la planificación del sprint 5 en Trello (Fuente propia: realizado en [Trello](#))

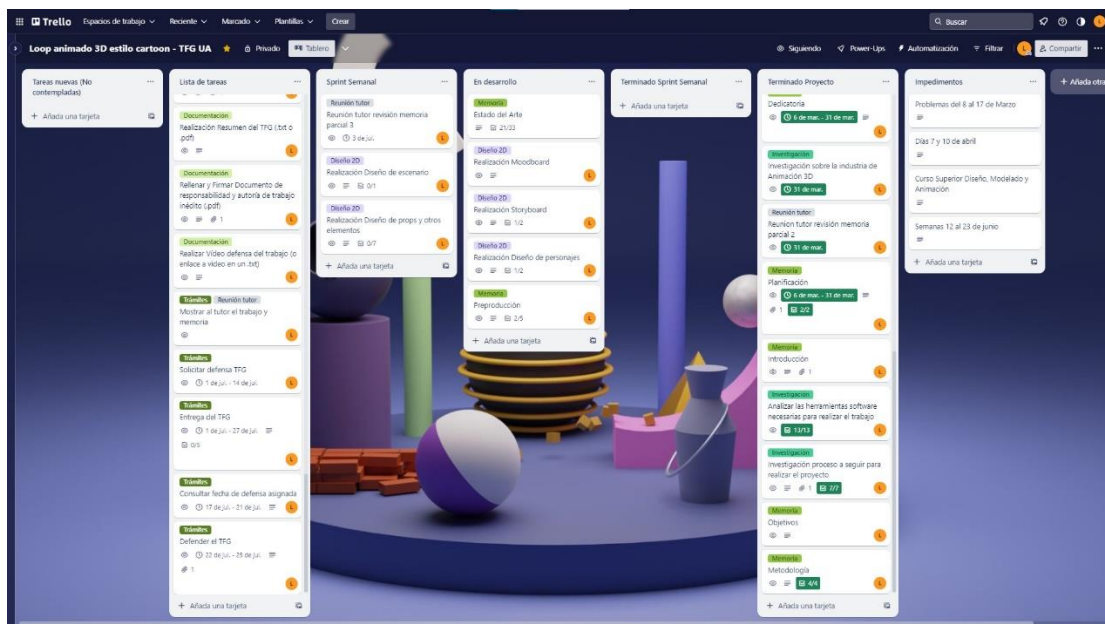


Figura 176. Fin de la planificación del sprint 5 en Trello (Fuente propia: realizado en [Trello](#))

Sprint 6: semana del 26 de junio al 2 de julio de 2023. En esta *sprint* se logró terminar la tarea que provocó que se alargara la planificación de los *sprints* anteriores por calcular mal el tiempo. En las siguientes figuras, se muestra: en la primera captura las tareas que se planificaron realizar en esa semana (que era la que se arrastraba de los *sprints* anteriores); y en la segunda, al final

de la semana, se logró terminar la tarea que más tiempo llevaba y se había calculado mal el tiempo estimado que duraría realmente.

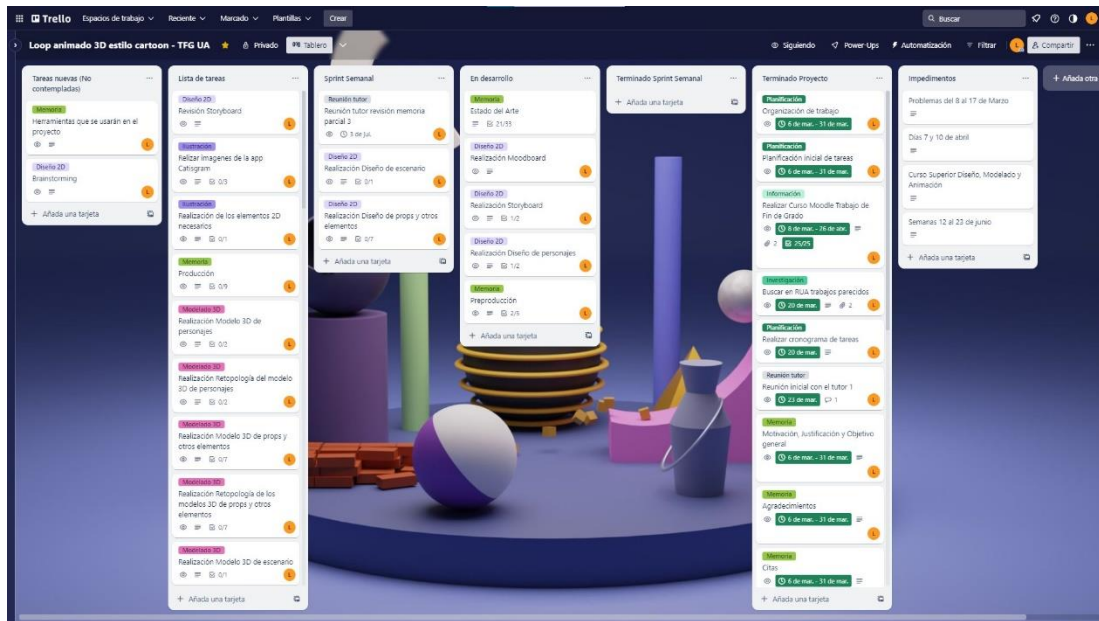


Figura 177. Inicio de la planificación del sprint 6 en Trello
(Fuente propia: realizado en [Trello](#))

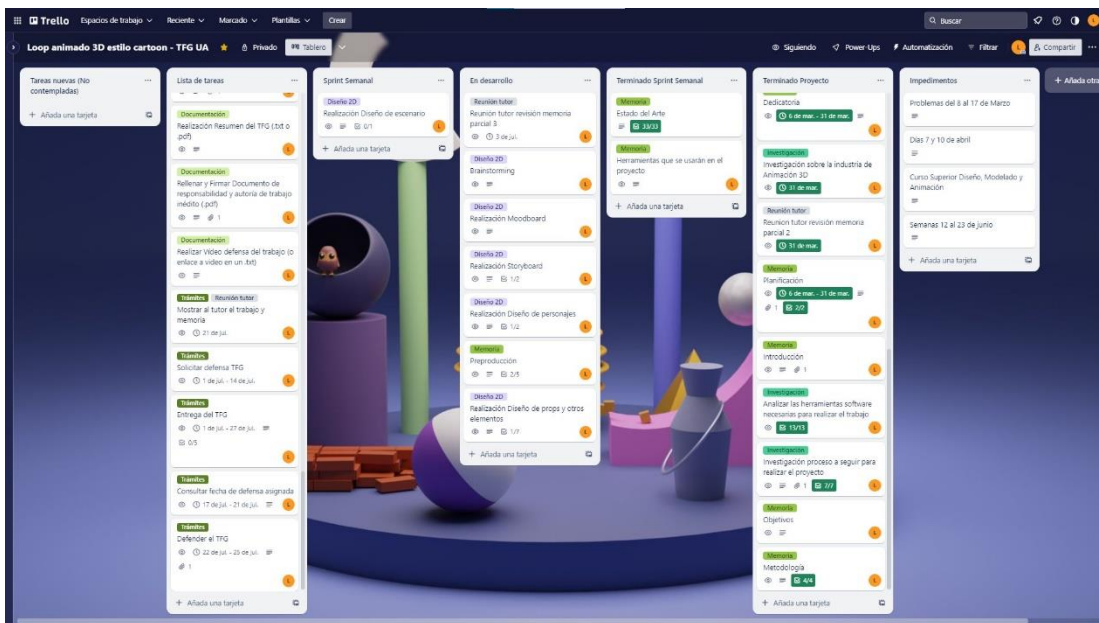


Figura 178. Fin de la planificación del sprint 6 en Trello
(Fuente propia: realizado en [Trello](#))

Sprint 7: semana del 3 al 9 de julio de 2023. Esta semana se utilizó para volver a encauzar el proyecto y volver a planificar los *sprints* siguientes para que no volviese a pasar lo mismo que con la tarea que duró más tiempo de lo esperado. En las siguientes figuras, se muestra: en las dos primeras las tareas que se planificaron realizar en esa semana; y en las dos últimas, al final

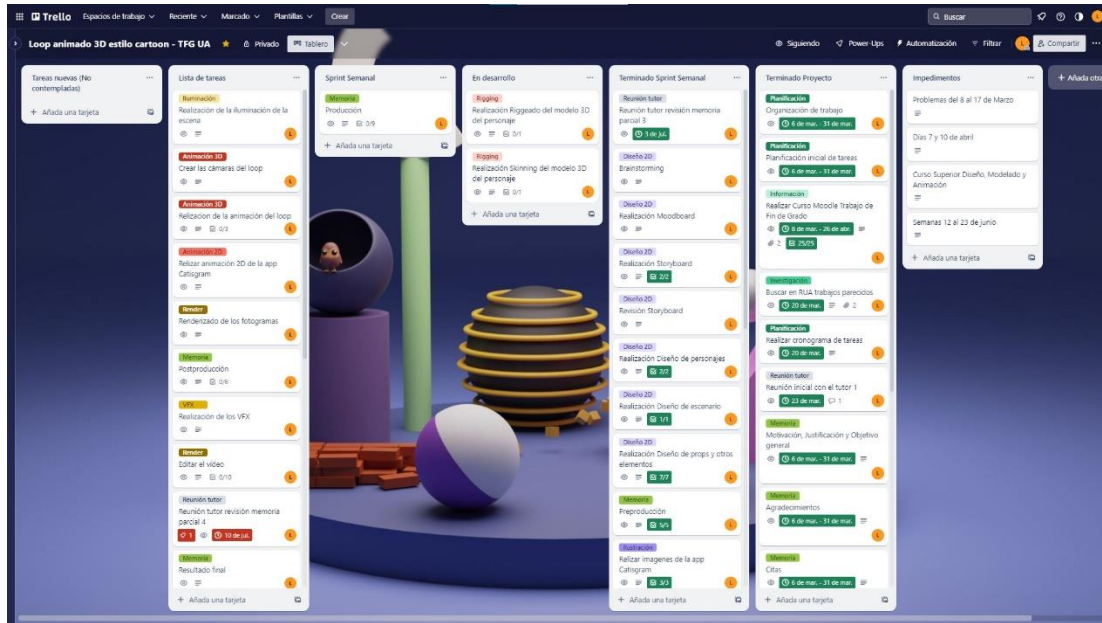


Figura 181. Fin de la planificación del sprint 7 en Trello (Parte 1)
(Fuente propia: realizado en [Trello](#))

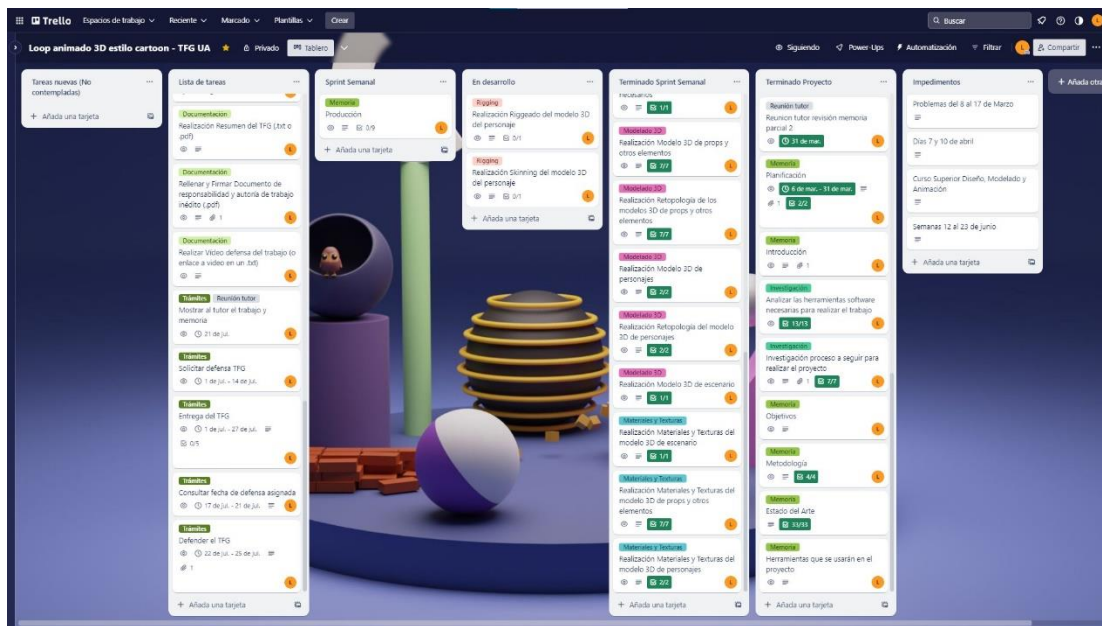


Figura 182. Fin de la planificación del sprint 7 en Trello (Parte 2)
(Fuente propia: realizado en [Trello](#))

Sprint 8: semana del 10 al 16 de julio de 2023. En las siguientes figuras, se muestra: en las dos primeras las tareas que se planificaron realizar en esa semana; y en la última, al final de la semana, todas las tareas que se lograron terminar esa semana. Como el ordenador tardó más de lo esperado en renderizar por la calidad del video, los materiales y luces utilizadas, las tareas de la postproducción no se pudieron realizar hasta el *sprint* siguiente.

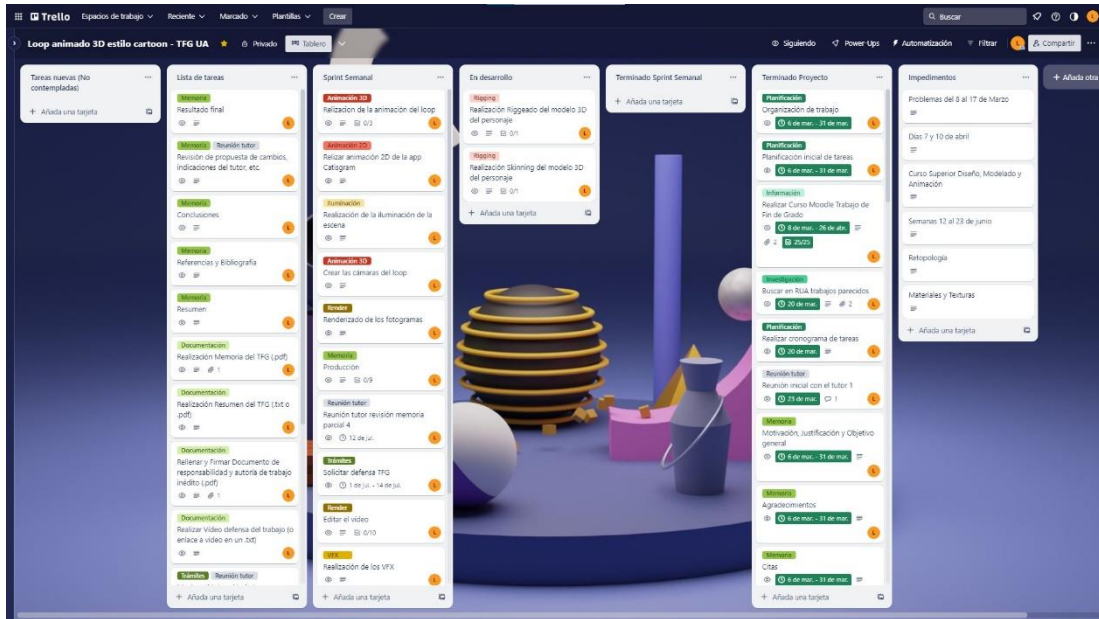


Figura 183. Inicio de la planificación del sprint 8 en Trello (Parte 1)
(Fuente propia: realizado en [Trello](#))

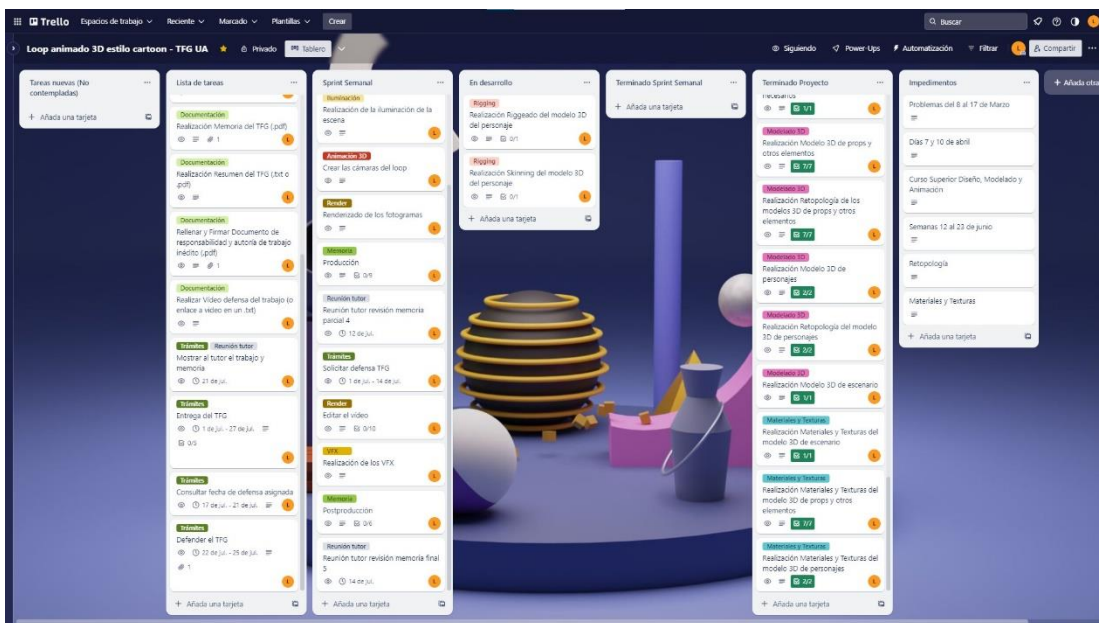


Figura 184. Inicio de la planificación del sprint 8 en Trello (Parte 2)
(Fuente propia: realizado en [Trello](#))

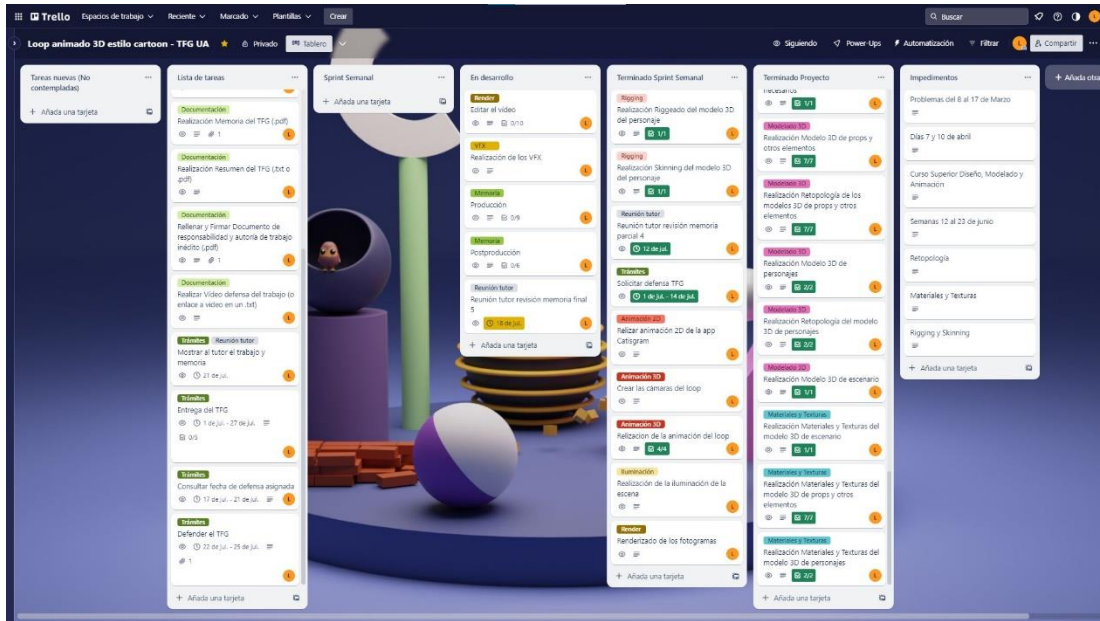


Figura 185. Fin de la planificación del sprint 8 en Trello (Fuente propia: realizado en [Trello](#))

Sprint 9: semana del 17 al 23 de julio de 2023. En las siguientes figuras, se muestra: en las dos primeras las tareas que se planificaron realizar en esa semana; y en la última, al final de la semana, todas las tareas que se lograron terminar esa semana. Para este sprint se incluyeron en la planificación inicial demasiadas tareas y por eso no se lograron terminar todas. Las tareas que quedaron sin terminar al final del *sprint* tenían que ver con la correcta preparación de la entrega del trabajo.

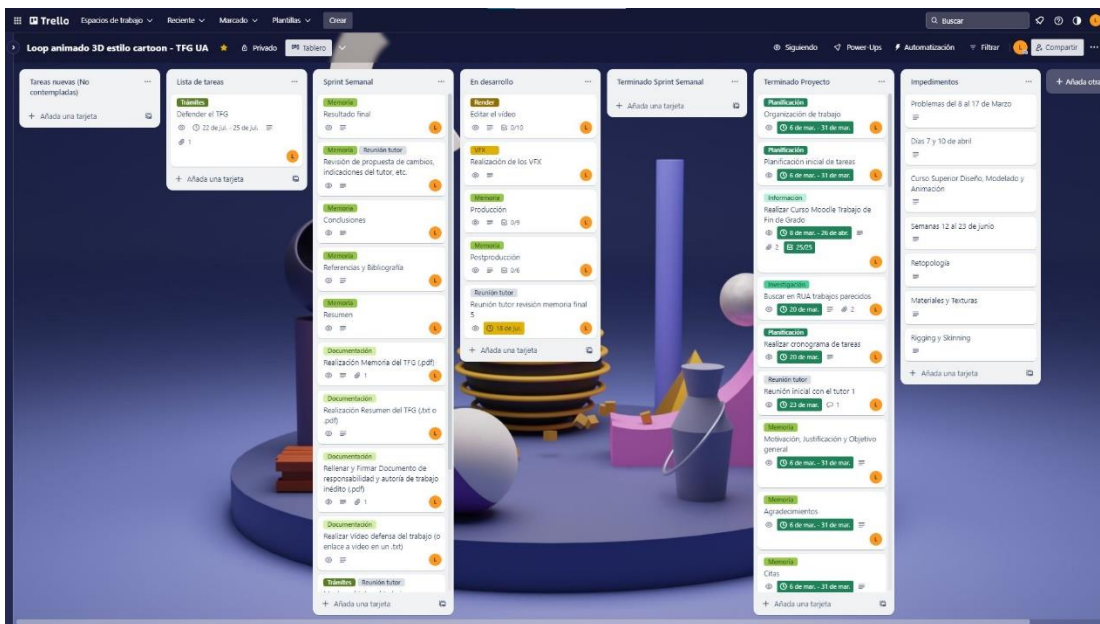


Figura 186. Inicio de la planificación del sprint 9 en Trello (Parte 1) (Fuente propia: realizado en [Trello](#))

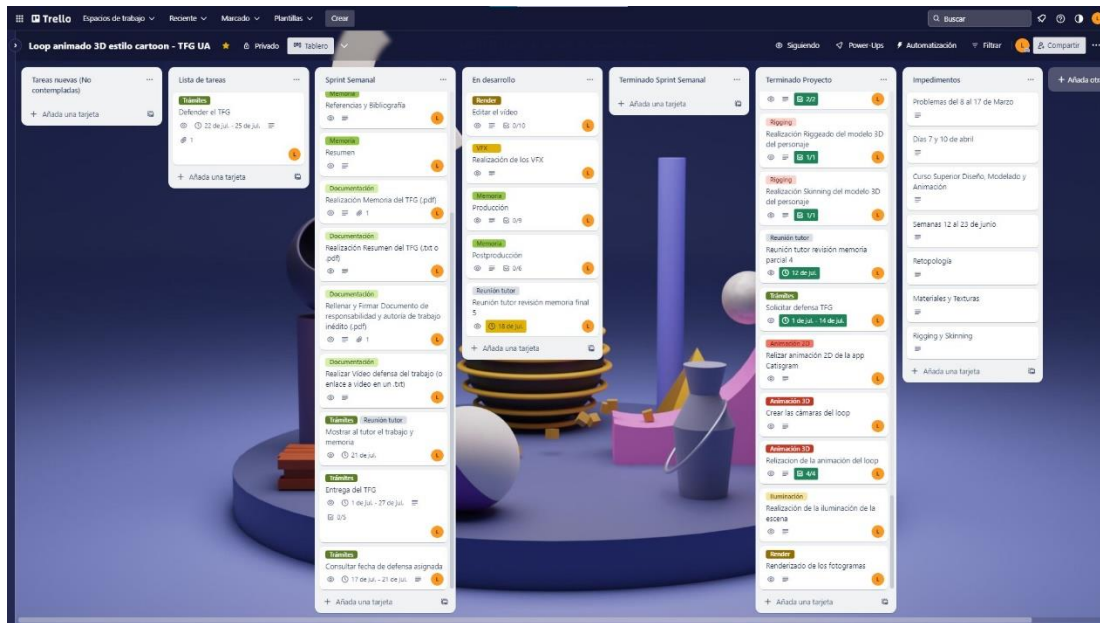


Figura 187. Inicio de la planificación del sprint 9 en Trello (Parte 2)
(Fuente propia: realizado en [Trello](#))

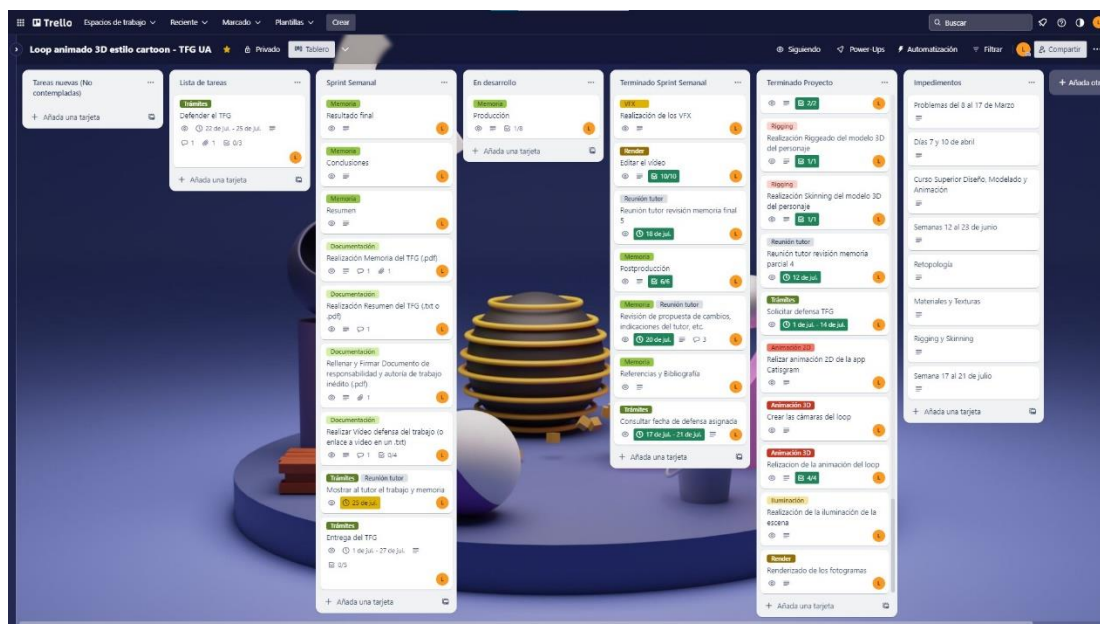


Figura 188. Fin de la planificación del sprint 9 en Trello
(Fuente propia: realizado en [Trello](#))

Sprint 10: semana del 24 al 27 de julio de 2023. Esta semana era más corta porque el 27 era el día de la entrega final del proyecto. En las siguientes figuras, se muestra: en la primera las tareas que se planificaron realizar en esa semana (que se arrastraron del *sprint* anterior); y en las dos últimas, al final de la semana, como se lograron realizar todas las tareas del proyecto. La tarea escrita en la columna *Lista de tareas* tiene que ver con la defensa del trabajo que se realizará el día indicado en septiembre de 2023.

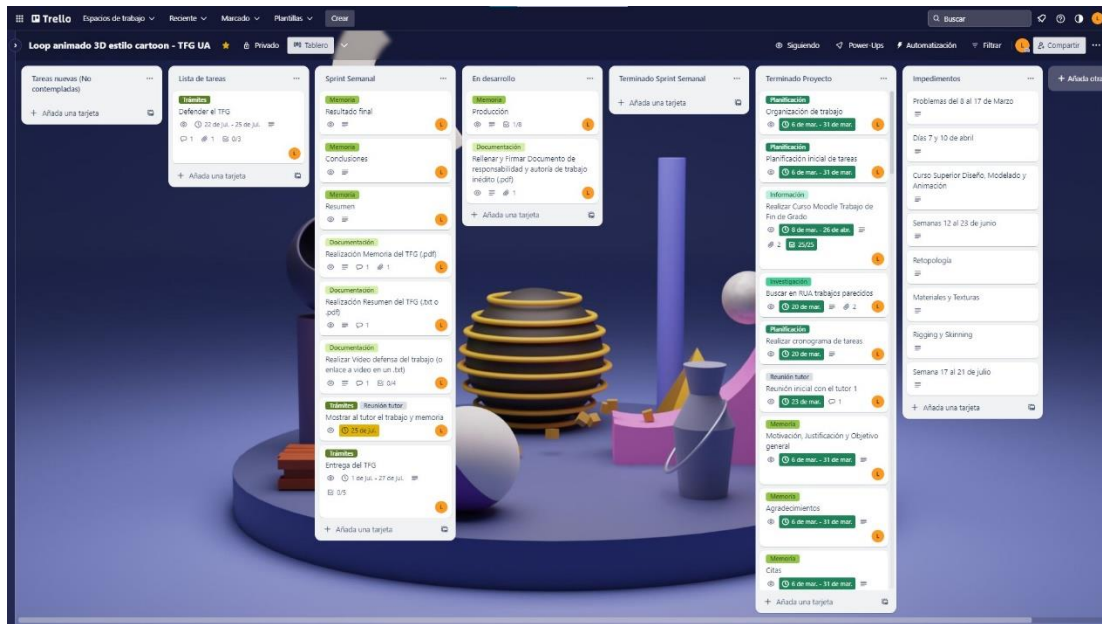


Figura 189. Inicio de la planificación del sprint 10 en Trello
(Fuente propia: realizado en [Trello](#))

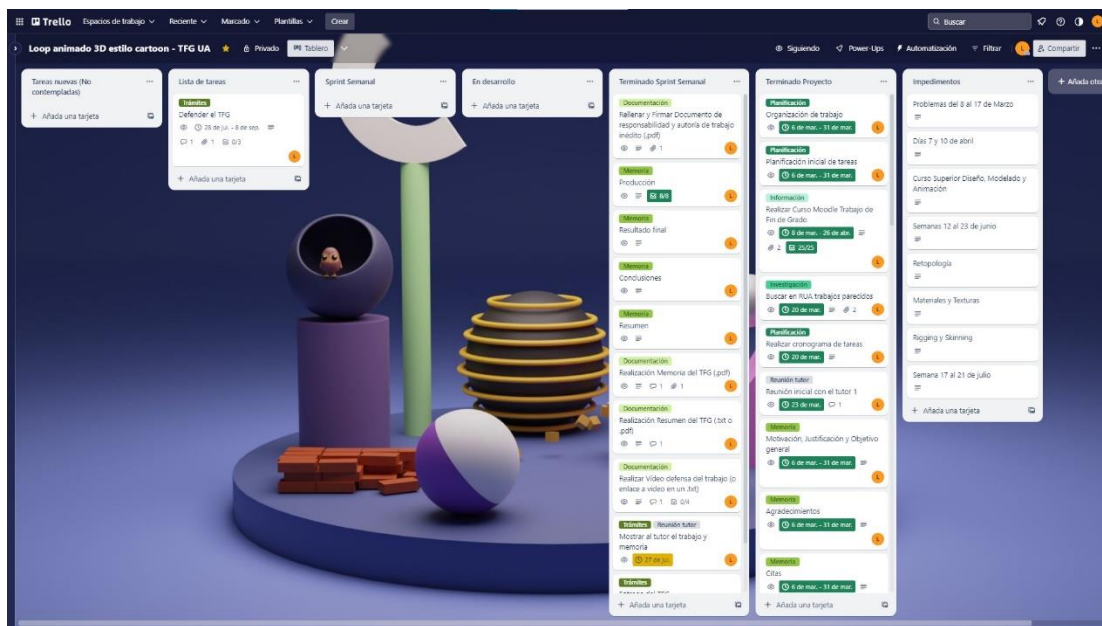


Figura 190. Fin de la planificación del sprint 10 en Trello (Parte 1)
(Fuente propia: realizado en [Trello](#))

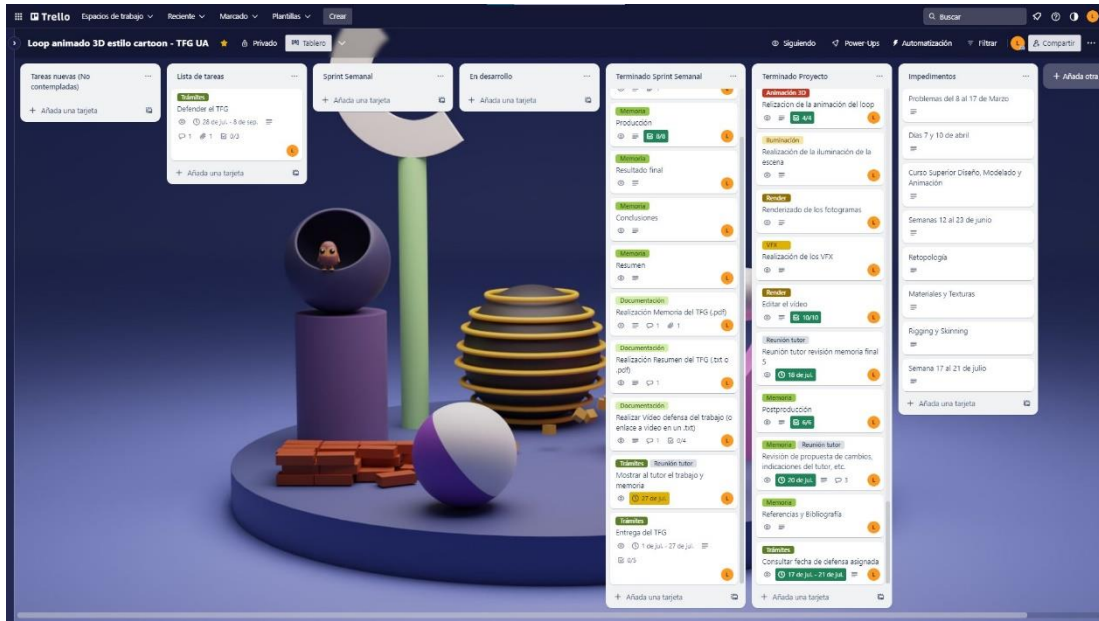


Figura 191. Fin de la planificación del sprint 10 en Trello (Parte 2)
(Fuente propia: realizado en [Trello](#))

Apéndice II: Informe de Toggl con las horas empleadas

A continuación, se muestran las capturas de pantalla del informe final de Toggl de las horas empleadas para cada tarea del proyecto, y las horas totales.

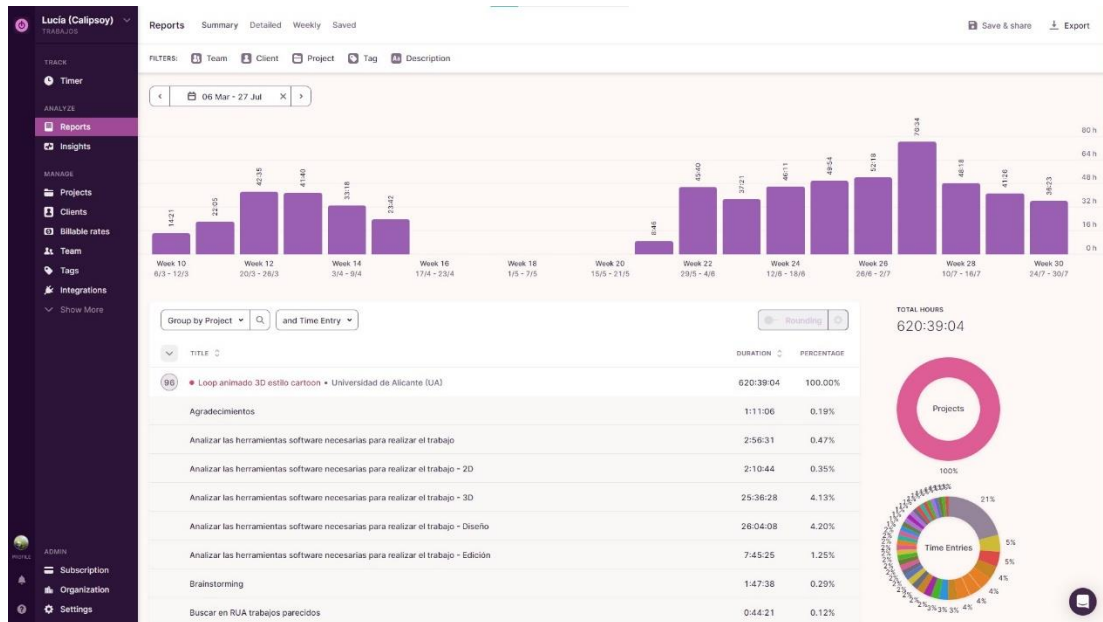


Figura 192. Informe de Toggl con las horas empleadas (Parte 1)
(Fuente propia: realizado con [Toggl](#))

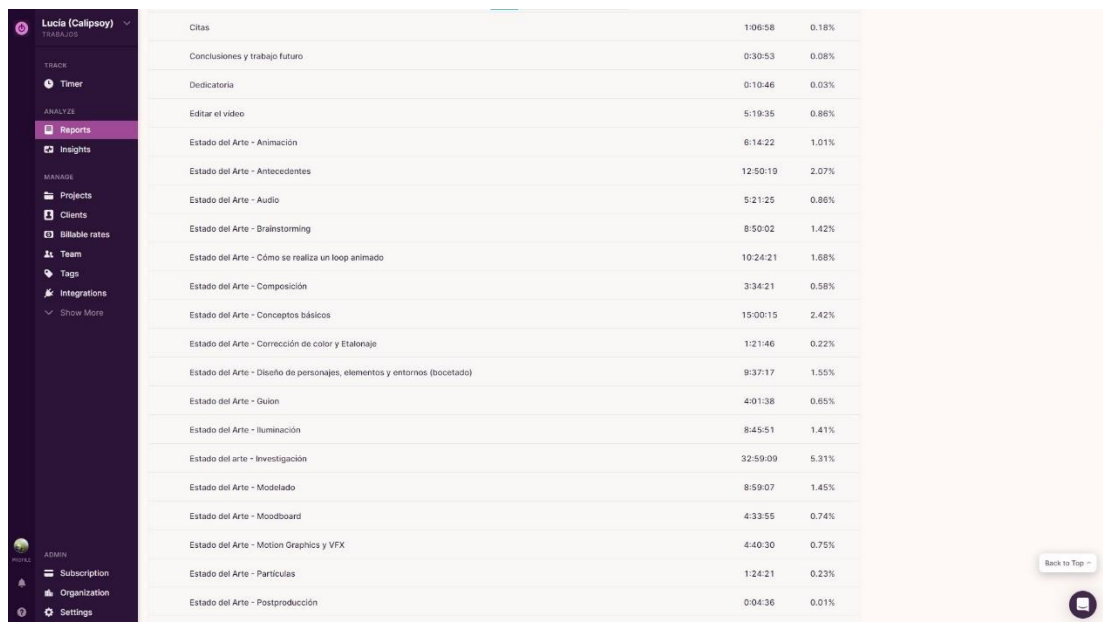


Figura 193. Informe de Toggl con las horas empleadas (Parte 2)
(Fuente propia: realizado con [Toggl](#))

Tarea	Horas	Porcentaje
Estado del Arte - Preproducción	3:28:54	0.56%
Estado del Arte - Principales herramientas software	9:28:14	1.53%
Estado del Arte - Principios de la animación	24:29:04	3.94%
Estado del Arte - Producción	0:18:59	0.05%
Estado del Arte - Renderizado	2:18:08	0.37%
Estado del Arte - Retopología	2:31:23	0.41%
Estado del Arte - Rigging	4:51:06	0.78%
Estado del Arte - Storyboard	4:09:58	0.67%
Estado del Arte - Texturizado	10:00:52	1.61%
Herramientas utilizadas	2:25:11	0.39%
Introducción	10:43:32	1.73%
Investigación proceso a seguir para realizar el proyecto	29:39:07	4.78%
Investigación sobre la industria de Animación 3D	10:30:24	1.69%
Metodología	17:15:18	2.78%
Motivación, Justificación y Objetivo general	2:09:18	0.35%
Objetivos	9:51:05	1.59%
Organización de trabajo	9:38:29	1.55%
Planificación	19:38:56	3.17%
Planificación inicial de tareas	10:04:31	1.62%
Postproducción	0:27:31	0.07%
Postproducción - Composición y Motion Graphics	1:03:51	0.17%

Figura 194. Informe de Toggl con las horas empleadas (Parte 3)
(Fuente propia: realizado con [Toggl](#))

Tarea	Horas	Porcentaje
Preproducción	0:07:33	0.02%
Preproducción - Brainstorming	1:15:25	0.20%
Preproducción - Diseño de personajes, elementos y entorno (bocetado)	1:07:02	0.18%
Preproducción - Moodboard	0:53:50	0.14%
Preproducción - Storyboard	0:25:29	0.07%
Producción	0:36:54	0.10%
Producción - Animación	2:59:01	0.48%
Producción - Iluminación y Sombreado	1:59:07	0.32%
Producción - Materiales y Texturizado	4:29:05	0.72%
Producción - Modelado	24:01:14	3.87%
Producción - Partículas	1:06:29	0.18%
Producción - Render	1:40:51	0.27%
Producción - Retopología	7:46:11	1.25%
Producción - Rigging y Skinning	4:52:50	0.78%
Realización de la iluminación de la escena	1:08:09	0.18%
Realización de los elementos 2D necesarios	1:03:03	0.17%
Realización de los VFX	7:30:00	1.21%
Realización Diseño de escenario	0:34:26	0.09%
Realización Diseño de personajes	7:45:26	1.25%
Realización Diseño de props y otros elementos	2:54:31	0.47%
Realización Materiales y Texturas del modelo 3D de escenario	0:09:30	0.03%

Figura 195. Informe de Toggl con las horas empleadas (Parte 4)
(Fuente propia: realizado con [Toggl](#))

Tarea	Horas	Porcentaje
Realización Materiales y Texturas del modelo 3D de personajes	8:40:21	1.40%
Realización Materiales y Texturas del modelo 3D de props y otros elementos	5:07:13	0.82%
Realización Memoria del TFG - Apéndice I: Planificación semanal en Trello	3:03:40	0.49%
Realización Modelo 3D de escenario	0:32:49	0.09%
Realización Modelo 3D de personajes	16:30:46	2.66%
Realización Modelo 3D de props y otros elementos	4:51:53	0.78%
Realización Moodboard	1:22:08	0.22%
Realización Retopología del modelo 3D de personajes	15:02:12	2.42%
Realización Riggado del modelo 3D del personaje	12:06:41	1.95%
Realización Skinning del modelo 3D del personaje	10:22:14	1.67%
Realización Storyboard	5:40:53	0.92%
Realizar cronograma de tareas	7:56:03	1.28%
Realizar Curso Moodle Trabajo de Fin de Grado	8:28:09	1.37%
Referencias	9:11:11	1.48%
Realización de la animación del loop	7:40:04	1.24%
Realizar animación 2D de la app Calligram	5:20:52	0.86%
Realizar imagenes de la app Calligram	1:24:46	0.23%
Rellenar y Firmar Documento de responsabilidad y autoría de trabajo inédito (.pdf)	0:15:13	0.04%
Rendizado de los fotogramas	6:40:11	1.07%
Resultado final	3:01:18	0.49%
Resumen	1:00:06	0.16%
Reunión inicial con el tutor 1 - José Vicente Berná	2:53:32	0.47%

Figura 196. Informe de Toggl con las horas empleadas (Parte 5)
(Fuente propia: realizado con [Toggl](#))

Tarea	Horas	Porcentaje
Realización Moodboard	1:22:08	0.22%
Realización Retopología del modelo 3D de personajes	15:02:12	2.42%
Realización Riggado del modelo 3D del personaje	12:06:41	1.95%
Realización Skinning del modelo 3D del personaje	10:22:14	1.67%
Realización Storyboard	5:40:53	0.92%
Realizar cronograma de tareas	7:56:03	1.28%
Realizar Curso Moodle Trabajo de Fin de Grado	8:28:09	1.37%
Referencias	9:11:11	1.48%
Realización de la animación del loop	7:40:04	1.24%
Realizar animación 2D de la app Calligram	5:20:52	0.86%
Realizar imagenes de la app Calligram	1:24:46	0.23%
Rellenar y Firmar Documento de responsabilidad y autoría de trabajo inédito (.pdf)	0:15:13	0.04%
Rendizado de los fotogramas	6:40:11	1.07%
Resultado final	3:01:18	0.49%
Resumen	1:00:06	0.16%
Reunión inicial con el tutor 1 - José Vicente Berná	2:53:32	0.47%
Reunion tutor revisión memoria parcial 2	1:15:00	0.20%
Revisión de propuesta de cambios, indicaciones del tutor, etc.	0:56:14	0.15%
Revisión de trabajo	7:36:50	1.23%

Figura 197. Informe de Toggl con las horas empleadas (Parte 6: final)
(Fuente propia: realizado con [Toggl](#))