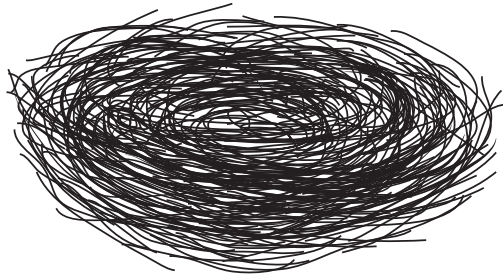




**ESPOL**  
*"Impulsando la sociedad del conocimiento"*

# DESARROLLO DE UN SISTEMA DE RENDERIZACIÓN DE ANIMACIONES 3D EN LA NUBE, MEDIANTE EL USO DE TECNOLOGÍAS DE CÓDIGO LIBRE.



**LIWEB**

**Licenciatura en Diseño Web y  
Aplicaciones Multimedia**

---

## **BROCHURE DE PROCESOS**

Tema:

Cometa - Render EDCOM

Autores:

Ronny David Valdivieso Mato

Ivonne Rosalía Burgos Benites

Paralelo # 1

**Firma del Profesor**

.....

MSc. Diego Carrera G.

**Contenido:**

**1. GENERALIDADES**

- 1.1 Introducción
- 1.2 Antecedentes
- 1.3 Definición del problema
- 1.4 Justificación
- 1.5 Objetivos general y específicos

**2. PROYECTO**

- 2.1 Metodología
- 2.2 Investigación y evaluación
- 2.3 Diseño
  - 2.3.1 Conceptos
  - 2.3.2 Línea gráfica
- 2.4 Desarrollo
  - 2.4.1 Cliente
  - 2.4.2 Servidor
- 2.5 Presupuesto
- 2.6 Cronograma

**3. CONCLUSIÓN**

- 3.1 Resultados obtenidos
- 3.2 Conclusión

**4. BIBLIOGRAFÍA**

- 4.1 Bibliografía

4 Desarrollo de un sistema de renderización de animaciones 3D en la nube, mediante el uso de tecnologías de código libre.



1

## GENERALIDADES

El sistema de Render como Servicio (RaaS) permite la renderización de animaciones 3D desde el navegador. Es una herramienta completa y robusta sin embargo su uso es sencillo e intuitivo, debido a su diseño minimalista.

# 1.1 INTRODUCCIÓN



En el inicio, la animación fue realizada como lo que ahora se conoce "técnica tradicional". El lápiz y el papel eran los mejores aliados del animador, sin embargo, la llegada del computador generó un impacto disruptivo en la forma del arte. (Jones & Oliff, 2007). Los gráficos por computadora permiten tener mayor control sobre las animaciones, aún así, existe un reto mayor dentro de la industria que es, la renderización final de las piezas digitales.

Renderizar, es el proceso de convertir un modelo 3D en efectos visuales y tangibles. Una vez el animador haya producido la animación con la ayuda de un software especializado, debe usar una forma altamente eficiente de convertir los modelos en imágenes (Wu, Xu & Vassilev, 2016). Para alcanzar una conversión de imágenes fiel al modelado, el render requiere de una configuración computacional de alta gama (Ver figura 1).



Figura 1. Proceso básico de renderizado. Fuente: flaticon.com



Figura 2. Animador trabajando. Fuente: <http://bit.ly/1XPLXtY>

Por lo tanto, este trabajo plantea una herramienta de código libre que provea un servicio de render (RaaS) simple e intuitivo, para ser utilizado por la academia y la industria, optimizando recursos del estado y privados para potenciar esta industria.

## 1.2 ANTECEDENTES



La animación es una de las industrias de mayor crecimiento y su aplicación no se limita a las películas, por el contrario es aprovechada tanto por los sectores del entretenimiento, marketing y más.

En los últimos años, la computación en la nube se ha convertido en la solución a muchos problemas, y entre ellos el proceso de renderización de animaciones.



Figura 3. Representación de renderización en la nube.  
Fuente: flaticon.com

En la actualidad se encuentran disponibles algunos servicios de render en la nube, la mayoría de ellos como servicios de pago, por ejemplo:



Figura 4. Logo de RebusFarm.  
Fuente: <http://bit.ly/2tbytes2>



Figura 5. Logo de RevUp Render. Fuente: <http://bit.ly/2F79Ut9>



Figura 6. Logo de Fox Renderfarm.  
Fuente: <http://bit.ly/2taYIKx>



Figura 7. Logo de Renderwow.  
Fuente: <https://goo.gl/Y6Nzm1>



Figura 8. Logo ZYNC.  
Fuente: <http://bit.ly/1tCellt>

# 1.3 PROBLEMA



El actual proceso de renderizado de imágenes 3D es complejo y lento. Cada detalle que se agrega en una animación como iluminación, texturas o agua implica mayor procesamiento, por ende el tiempo de renderizado aumenta.v



Figura 9. Representación de avance en el tiempo. Fuente: flaticon.com



Figura 10. Modelado de personaje en Blender.  
Fuente: <https://goo.gl/sMzd9B>

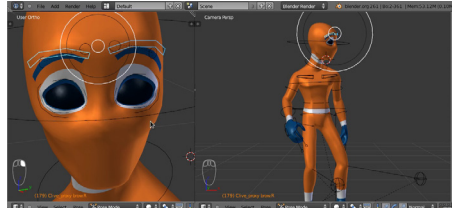


Figura 11. Animación de personaje en Blender.  
Fuente: <https://goo.gl/KZdXLv>



Figura 12. Sombreado de piel en Blender.  
Fuente: <https://goo.gl/FM3pL8>



## 1.4 JUSTIFICACIÓN



### Tiempo

La renderización de imágenes necesita mucho tiempo (especialmente animación 3D) debido a la complejidad de las escenas y los efectos especiales.



Figura 13. Representación de tiempo.  
Fuente: <http://bit.ly/2FLwzMQ>



Figura 14. Representación de recursos.  
Fuente: <http://bit.ly/2HV1Ayi>

### Recursos

Configurar un sistema RaaS implica el manejo de recursos complejos.

### Conocimientos

Conocimientos computacionales de alta gama.



Figura 15. Representación de conocimiento.  
Fuente: <http://bit.ly/2CSvrUc>

Por lo tanto es importante contar con un sistema que no sólo reduzca los tiempos de renderización, si no que resuelva los problemas con la administración de recursos complejos, mejore la eficiencia en la producción de animaciones y optimice los costos.

# 1.5 OBJETIVOS GENERAL Y ESPECÍFICOS



## General

Desarrollar una solución de Render como Servicio mediante el uso de tecnologías de código libre para la mejora de los procesos de renderización de multimedios 3D.



Figura 16. Representación de objetivo. Fuente : <http://bit.ly/2FcKei5>

## Específicos

1. Configurar el entorno de trabajo de la herramienta de cómputo remoto Afanasy para el renderizado de 3D modelados en el software Blender. (<https://goo.gl/aLPBEr>)
2. Instalar Owncloud del lado del servidor como plataforma de almacenamiento y sincronización de archivos para la gestión de los proyectos de animación de los usuarios finales. (<https://goo.gl/NFxCmd>)
3. Desarrollar plugins **“Render”** y **“Mis Trabajos”** para el envío y posterior visualización de los proyectos renderizados.
4. Realizar pruebas con usuarios reales para la evaluación del correcto funcionamiento del sistema y todos sus componentes.



Figura 17. Representación de tareas. Fuente: <http://bit.ly/2F6taqK>

2

# PROYECTO



## 2.1 METODOLOGÍA



Figura 18. Representación de trabajo en equipo. Fuente: <http://bit.ly/2vt2U4H>

La metodología SCRUM, es ágil y flexible para gestionar el desarrollo de software. En SCRUM se realizan avances parciales y regulares, priorizando el beneficio que aporta al usuario del proyecto. Por tal motivo es ideal para proyectos en entornos complejos, donde los requisitos cambian o se modifican y la innovación, la competitividad, la flexibilidad y la productividad son primordiales.



Figura 19. Representación de procesamiento.  
Fuente: <http://bit.ly/2HV1Ayi>

1. Configuración de herramienta de control de procesamiento remoto.



Figura 20. Representación de almacenamiento en la nube. Fuente: <http://bit.ly/2HV1Ayi>

2. Configuración de plataforma de almacenamiento en la nube.



Figura 21. Representación de programación.  
Fuente: <http://bit.ly/2HV1Ayi>

3. Programación de plugins.



Figura 22. Representación de renderización.  
Fuente: <http://bit.ly/2HV1Ayi>

4. Pruebas de Renderización.



## 2.2 INVESTIGACIÓN Y EVALUACIÓN

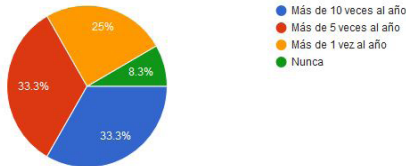


Con el fin de validar la idea, se realizaron encuestas a animadores tanto profesionales como estudiantes, para comprobar la aceptación de un sistema de este tipo.

Las encuestas demostraron que los artistas **SÍ** están interesados en utilizar una solución RaaS, para agilizar los procesos de renderización.

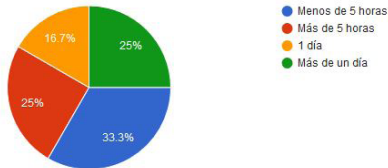
¿Con qué frecuencia realiza procesos de renderizado 3D?

12 respuestas



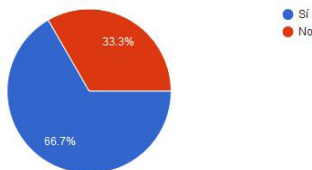
¿Cuánto tiempo mínimo necesita para llevar a cabo un proceso de renderizado?

12 respuestas



¿Estaría interesado en utilizar un aplicación de Render como Servicio?

12 respuestas



## 2.3 DISEÑO



### 2.3.1 CONCEPTOS

#### Concepto comunicacional

Sistema robusto, confiable y fácil de usar.



Figura 23. Representación de transmitir ideas. Fuente: <https://goo.gl/eLenif>



Figura 24. Representación de creatividad. Fuente: <https://goo.gl/Vw8oQG>

#### Concepto creativo

Cometa es un sistema RaaS que permite la renderización de animaciones 3D. Es una herramienta completa y robusta sin embargo su uso es sencillo e intuitivo, debido a su diseño minimalista.

## 2.4.2 LÍNEA GRÁFICA

### Estilo

Plano y minimalista

### Logo



### Colores

#222e6a

Principal

#eff0f5

Fondos

#5964b3

Secundario

#d9d9d9

Sombras

#ff8500

Complementario

Se escogió al azul como color principal debido a que transmite sentimientos de seguridad y confianza. Además, es un color ampliamente utilizado en las marcas de la industria tecnológica y con buenos resultados.

Adicionalmente se ha incluido al naranja como color complementario para darle un toque más relajado, accesible y amigable al proyecto.

Los tonos grises son para fondos y textos.

### Iconos

#### Iconos de plugin



Render



Mis trabajos

#### Iconos de plugin



### Tipografía

Sans-serif o Palo Seco, ideal para pantallas digitales.

# Myriad Pro

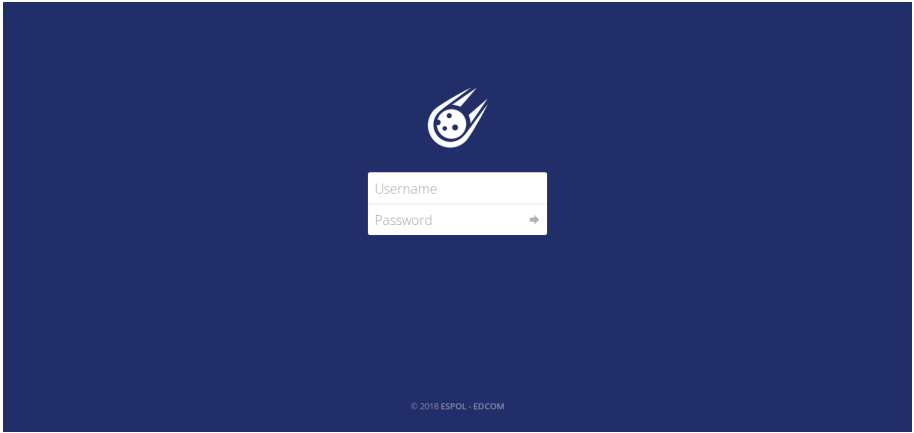


Figura 25. Inicio de sesión de la aplicación. Elaboración propia.

1. En el diseño de la pantalla de "Inicio de sesión". Contiene el logo de Cometa y un formulario de autenticación. (Ver figura 25).

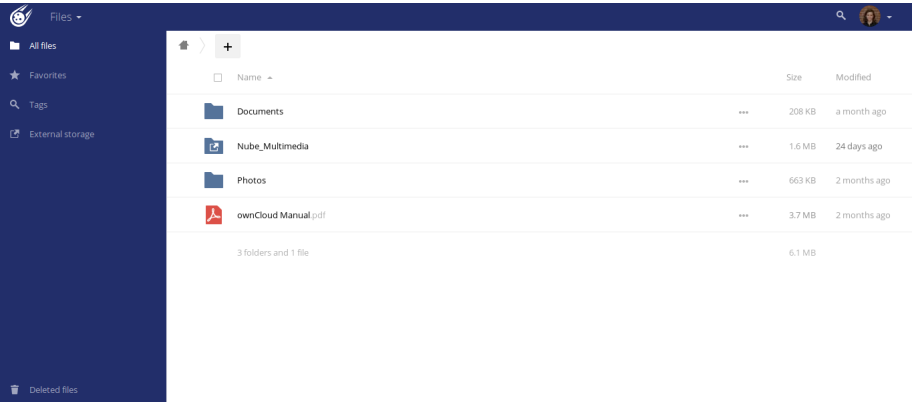


Figura 26. Directorio principal de archivos. Elaboración propia.

2. En el directorio principal, se utilizan los iconos del sistema ubicados en el la barra lateral. Los iconos utilizados son: una **carpeta** para la opción ver todos los archivos, una **estrella** para la opción favoritos, una **lupa** para la búsqueda por etiquetas y el icono de almacenamiento externo.



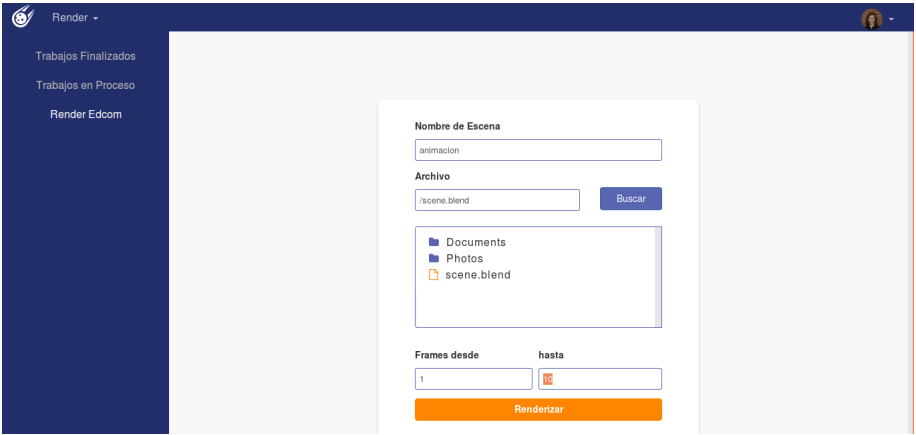


Figura 27. Formulario del plugin Render. Elaboración propia.

3. En el plugin Render EDCOM, se implementó un formulario sencillo que cuenta con 4 campos, dos botones y un visor de directorios que permite la búsqueda de archivos para enviar a renderizar. (Ver figura 27).

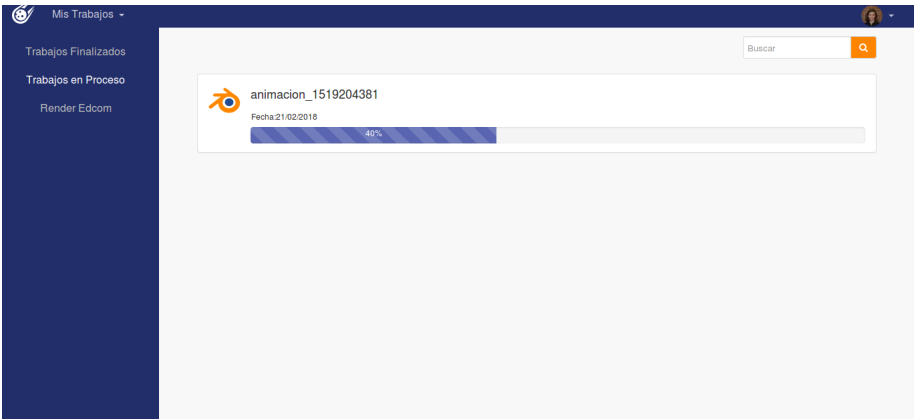


Figura 28. Barra de progreso del plugin Mis trabajos. Elaboración propia.

4. Barra de progreso que muestra el porcentaje de avance de la renderización. (Ver figura 28).

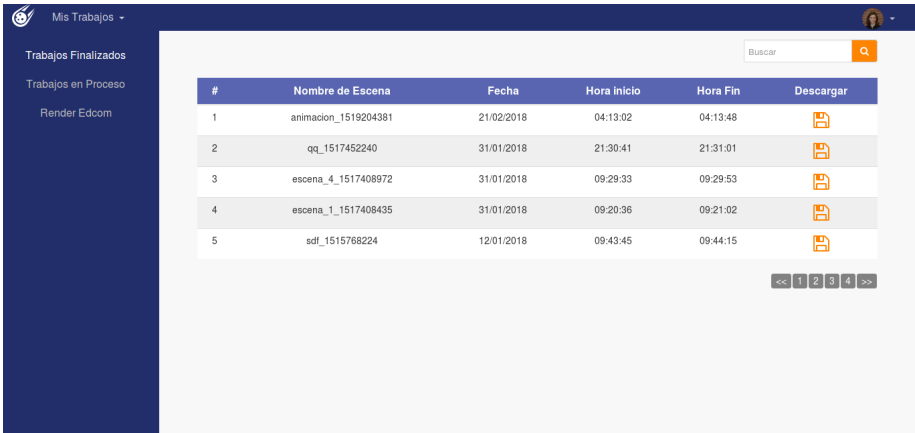


Figura 29. Lista de trabajos renderizados del plugin Mis trabajos. Elaboración propia.

5. Tabla que contiene información general de los trabajos finalizados, cada trabajo tiene la opción que permite guardar los archivos en una carpeta personal. También cuenta con paginación para agrupar cierta cantidad de trabajos y un buscador en la parte superior derecha que permite filtrar por el nombre de la escena. (Ver figura 29).

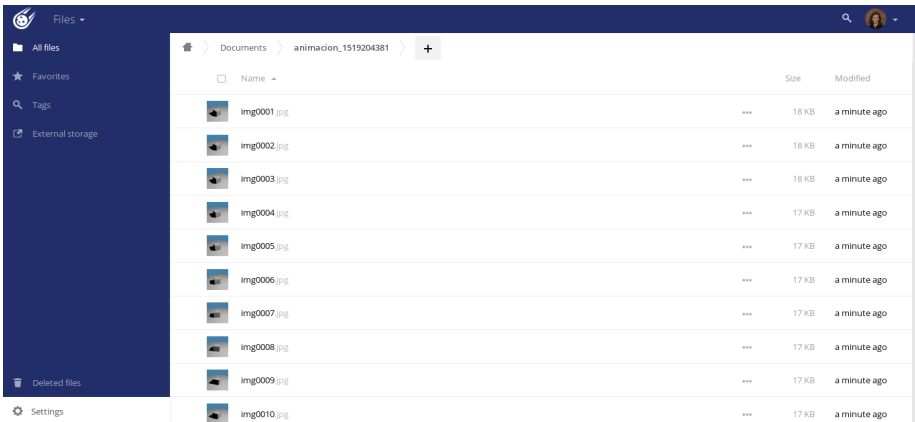


Figura 30. Carpeta que contiene las imágenes renderizadas. Elaboración propia.

6. En la figura 30, está la carpeta que contiene las imágenes guardadas de un trabajo.

## 2.4 DESARROLLO



Para el desarrollo del sistema se realizó una integración entre dos herramientas de código libre. **CGRU - Afanasy** que es una herramienta de cómputo remoto para animaciones y **Owncloud** que es una plataforma de almacenamiento y sincronización de archivos. La integración se la hizo sobre el sistema operativo Ubuntu y el software de modelado y animación 3D Blender.



Figura 31. Logo de Owncloud.  
Fuente: <http://bit.ly/2CVdipa>



FREE AND OPEN SOURCE  
RENDER FARM MANAGER

Figura 32. Logo de CGRU AFANASY.  
Fuente: <http://bit.ly/2GWOmzT>



Figura 33. Logo de Ubuntu.  
Fuente: <https://goo.gl/gTmy1w>



Figura 34. Logo de Blender.  
Fuente: <https://goo.gl/nUDnMi>

La integración de las herramientas fue posible con el uso de los siguientes lenguajes y tecnologías de programación:

### Backend



### Frontend



El proyecto se desarrolló en las instalaciones de EDCOM, con un equipo Intel Core i7. Para las pruebas de renderización se hizo uso de los laboratorios de EDCOM y finalmente la totalidad del sistema está alojado en un servidor de ESPOL el cual se puede acceder mediante la URL [www.cometa.espol.edu.ec](http://www.cometa.espol.edu.ec).

Desarrollo de un sistema de renderización de animaciones 3D en la nube, mediante el uso de tecnologías de código libre.

## Arquitectura

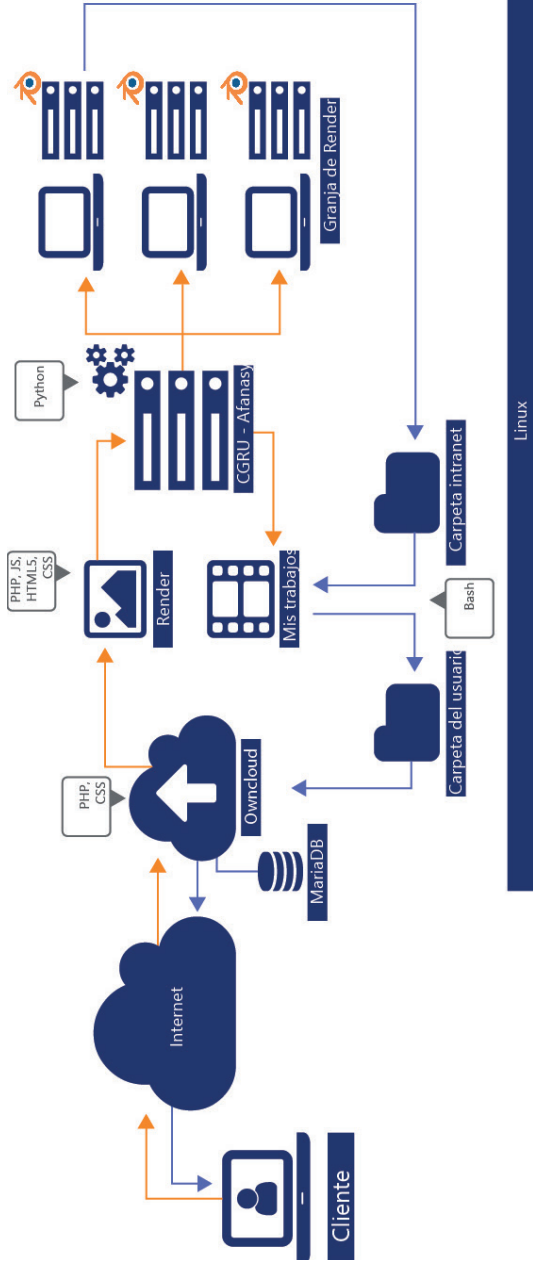


Figura 37. Arquitectura de Cometa. Elaboración propia.



## 2.4.1 CLIENTE



1. El usuario inicia sesión (ver figura 25) y visualiza el directorio de carpetas (ver figura 26) que tiene sincronizada con el cliente de escritorio que provee Owncloud. De esta manera el animador guarda sus archivos localmente y automáticamente se sincroniza en la nube.
2. Luego en el menú de aplicaciones ubicado en la parte superior izquierda, el usuario elige el plugin **Render** para renderizar sus animaciones. Como se muestra en la figura 27, el plugin Render cuenta con un formulario que permite seleccionar el archivo, definir el rango de frames a tomarse en cuenta y enviar a **"Renderizar"**.
3. Inmediatamente se iniciará el plugin **"Mis Trabajos"**, que se encarga de mostrar el estado del proceso de renderizado en la sección **"Trabajos en Proceso"**. (Ver figura 28).
4. Cuando la renderización culmina, los trabajos se listan en la sección **Trabajos Finalizados** tal como se muestra en la figura 29 y al hacer clic en el icono de **Guardar** se sincronizan los archivos en la nube con la máquina del cliente.
5. La carpeta con los archivos renderizados se guardará con el nombre de la escena y un código único generado por la herramienta para evitar conflictos de nombres. (Ver figura 30).

## 2.4.2 SERVIDOR



```

- 8796 bytes.
Wed 21 Feb 04:10.41: Job registered: "sdf_1515768224"[3]: admin2@ronny-js[15] -
9745 bytes.
Wed 21 Feb 04:10.41: Job registered: "escena_1_1517408435"[7]: admin2@ronny-js[1
6] - 8811 bytes.
Wed 21 Feb 04:10.41: Job registered: "escena_4_1517408972"[10]: admin2@ronny-js[
17] - 8811 bytes.
Wed 21 Feb 04:10.41: Job registered: "render_8_1517409362"[11]: ronny@ronny-js[3
] - 8842 bytes.
Wed 21 Feb 04:10.41: INFO      24 jobs registered from store.
Wed 21 Feb 04:10.41: Job registered: "qq_1517452240"[18]: admin2@ronny-js[18] -
8766 bytes.
Wed 21 Feb 04:10.41: INFO      Raising 10 threads to process incoming connections.
..
Wed 21 Feb 04:10.41: INFO      Raising 10 threads to read/write sockets...
Wed 21 Feb 04:10.41: INFO      Raised 10 threads.
Wed 21 Feb 04:10.41: INFO      Run thread started.
Available addresses:
IP = '0.0.0.0'
IPv6 = '::'

```

Figura 38. Proceso afserver iniciado. Elaboración propia

1. Una máquina de la Granja de Render se encarga de ser el monitor principal. Este nodo corre un servicio llamado **Afserver**, quién es el encargado de escuchar si hay trabajos y enviarlo a los otros nodos esclavos. (Ver figura 38)

```

RLIMIT_NPROC: Processes (threads) limit: 23107
WARNING: SERVER ADDRESS ID NOT CONFIGURED, USING 127.0.0.1
Network Interfaces:
  enp2s0: f0:79:59:24:a1:ea
  wlp3s0: ac:b5:7d:2e:5c:e7
         IPv4 Address: 192.168.0.104
         IPv6 Address: fe80:::a715:5775:bb17:cb9d
ON
Host:
  OS="unix linux 64"
  Capacity = 0, Max Tasks = 0, Power = 0

Resources:
  CPU = 1933 MHz x4
        2% usr, 0% nice, 0% sys, 97% idle, 0% iow, 0% irq, 0% sirq
  load average:  0.2  0.6  0.5
  Memory: 5851 Mb / 4262 Mb free (cache 1229 Mb, buffers 281 Mb)
  Swap: 6026 Mb / 0 Mb used
  Network: Recieved 0 Kb/sec, Send 0 Kb/sec
  IO: Read 0 Kb/sec, Write 0 Kb/sec, Busy = 0%

```

Figura 39. Proceso afrender iniciado. Elaboración propia

3. Cada nodo esclavo de la granja corre un servicio llamado **Afrender**, con el cual se comunica con el monitor principal para realizar las tareas asignadas. Cada nodo debe tener instalado el motor de Blender para poder renderizar las imágenes. (Ver figura 39).



```
Wed 21 Feb 04:57.31: INFO Started PID=6841POST: convert -identify "/var/www/h
tml/owncloud/Nube_Multimedia/admin2/animacion_2_1519207048/img0001.jpg" -thumbn
ail "100x100^" -alpha Deactivate -gravity center -extent 100x100 -colorspace sRG
B "/var/tmp/afanasy/tasks/25.0.0_1/img0001.jpg.jpg" (Running tasks:1)
Wed 21 Feb 04:57.32: INFO Finished PID=6841: Exit Code=0 Status=0
Wed 21 Feb 04:57.33: INFO Started PID=6847 [blender:1000] admin2: animacion_2
_1519207048[blenderRender][frame 2]
Wed 21 Feb 04:57.35: INFO Finished PID=6847: Exit Code=0 Status=0
Wed 21 Feb 04:57.35: INFO Started PID=6867POST: convert -identify "/var/www/h
tml/owncloud/Nube_Multimedia/admin2/animacion_2_1519207048/img0002.jpg" -thumbn
ail "100x100^" -alpha Deactivate -gravity center -extent 100x100 -colorspace sRG
B "/var/tmp/afanasy/tasks/25.0.1_2/img0002.jpg.jpg" (Running tasks:1)
Wed 21 Feb 04:57.36: INFO Finished PID=6867: Exit Code=0 Status=0
Wed 21 Feb 04:57.37: INFO Started PID=6874 [blender:1000] admin2: animacion_2
_1519207048[blenderRender][frame 3]
Wed 21 Feb 04:57.39: INFO Finished PID=6874: Exit Code=0 Status=0
Wed 21 Feb 04:57.39: INFO Started PID=6898POST: convert -identify "/var/www/h
tml/owncloud/Nube_Multimedia/admin2/animacion_2_1519207048/img0003.jpg" -thumbn
ail "100x100^" -alpha Deactivate -gravity center -extent 100x100 -colorspace sRG
B "/var/tmp/afanasy/tasks/25.0.2_3/img0003.jpg.jpg" (Running tasks:1)
```

Figura 40. Detalle del proceso de renderizado. Elaboración propia

**4.** Al momento que se inicia un proceso de renderizado, el monitor verifica el estado de los nodos y envía las tareas. Todos los nodos esclavos están conectados a una carpeta intranet que funciona como repositorio temporal de los archivos. En el momento que el usuario da clic en el icono de Guardar del plugin Mis Trabajos, la carpeta se borra del repositorio temporal para no sobrecargar la capacidad de almacenamiento del servidor.

## 2.5 DETALLE DE PRESUPUESTO



### SERVICIOS PROFESIONALES

Cargo	Costo	Tiempo de Trabajo	Cantidad	Total
Jefe de Sistemas	\$1,740.00	5	1	\$8,700.00
Programador y diseñar multimedia / web	\$850	2	2	\$3,400.00
				<b>\$12,100.00</b>

### LICENCIAS

Rubro	Costo	Tiempo de Trabajo	Cantidad	Total
Licencia Adobe Illustrator CC	\$30	5	1	\$150.00
Licencia Adobe Photoshop CC	\$30	5	1	\$150.00
				<b>\$300.00</b>

### VIÁTICOS

Rubro	Costo	Tiempo de Trabajo	Cantidad	Total
Transporte	24	5	2	\$240.00
Alimentación	60	5	2	\$600.00
				<b>\$840.00</b>



## 2.6 CRONOGRAMA



### OCTUBRE

	1	2	3	4
Instalar Afanasy				
Establecer variables de entornos y archivos de configuración				
Instalar Blender en los nodos				
Instalar Owncloud 9.0.4.				
Crear carpeta compartida entre entre nodos y servidor.				
Configurar OC para el trabajo conjunto con Afanasy.				

### NOVIEMBRE

	1	2	3	4
Crear la estructura base de los plugins con el framework de desarrollo de OC.				
Personalizar la interfaz de los plugins de acuerdo a sus funciones.				
Enviar nuevo trabajo de renderización a la nube multimedia mediante formulario de OC.				
Agregar funcionalidad para subir los resultados de render a la carpeta OC del usuario				

## DICIEMBRE

	1	2	3	4
Enviar nuevo trabajo de renderización a la nube multimedia mediante formulario de OC.				
Agregar funcionalidad para subir los resultados de render a la carpeta OC del usuario				

## ENERO

	1	2	3	4
Obtener trabajos reales de animación 3D				
Medir los tiempos de procesamiento				

3

## CONCLUSIÓN



## 3.1 RESULTADOS OBTENIDOS

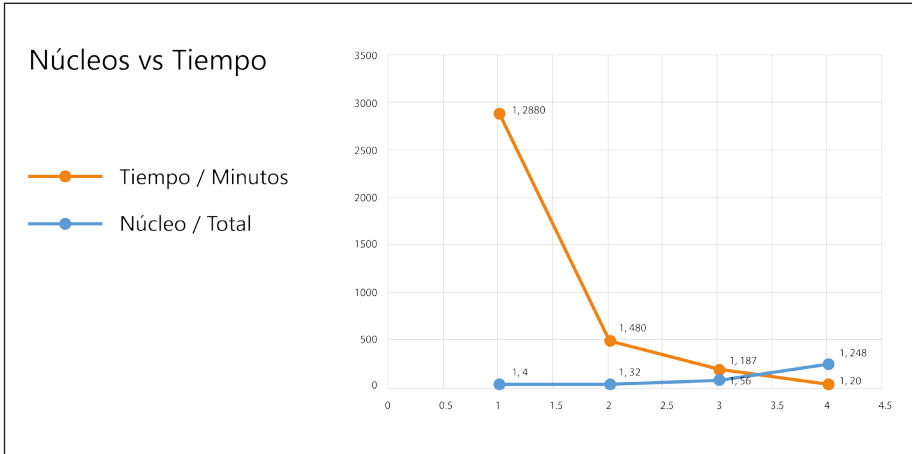


Figura 41. Resultados de pruebas de procesamiento. Elaboración propia.

Esta gráfica muestra una relación entre núcleos del procesador y tiempo de renderización, en la cual a mayor cantidad de procesadores, menor es el tiempo de renderizado de una animación en 3D.

## 3.2 CONCLUSIONES



De acuerdo a las evaluaciones de las encuestas se determinó que hay un buen porcentaje de personas dedicadas al modelado y animación en 3D, pero se encuentran con el problema de que, el tiempo que se necesita para renderizar un trabajo es mucho.

Los resultados obtenidos en las pruebas de funcionamiento de Cometa han demostrado que es posible acelerar el proceso de renderizado, gracias a la integración de diferentes tecnologías que configuradas correctamente pueden trabajar entre sí, optimizando los procesos y acortando los tiempos de renderización.



4

# BIBLIOGRAFÍA

## 4.1 BIBLIOGRAFÍA



Wu, J., Xu, H., & Vassilev, T. (2016). Design of 3D Rendering Platform Based on Cloud Computing. 2016 4th International Conference on Enterprise Systems (ES). doi:10.1109/es.2016.26

Patil, G. V., & Deshpande, S. L. (2016). Distributed rendering system for 3D animations with Blender. 2016 IEEE International Conference on Advances in Electronics, Communication and Computer Technology (ICAECCT). doi:10.1109/icaecct.2016.7942562

Jones, A., & Oliff, J. (2008). Thinking animation: bridging the gap between 2D and CG. Boston, MA: Thomson Course Technology.

Liu, Z., & Zou, H. (2015). AzureRender: A Cloud-Based Parallel and Distributed Rendering System. 2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems. doi:10.1109/hpcc-css-icess.2015.328

### Sitios web

R. (n.d.). Art Wanted! Featuring Deny Maulana Mallik. Retrieved January 31, 2018, from <https://us.rebusfarm.net/es>

R. (n.d.). RevUp Render - Cloud Computing for Designers. Retrieved January 31, 2018, from <https://www.revuprender.com/>

R. (n.d.). Render Farm In The Cloud. Retrieved January 31, 2018, from <https://www.renderrocket.com/features.php>

ZYNC, I. (n.d.). Google - Google Cloud Visual Effects Rendering. Retrieved February 01, 2018, from <https://www.zyncrender.com/>

Shenzhen Rayvision Technology (n.d.). Cloud Render Farm Online | Fox Render Farm. Retrieved February 01, 2018, from <http://www.foxrenderfarm.com/>

Garagefarm. (n.d.). Cloud Rendering Services | GarageFarm.NET Render Farm. Retrieved February 01, 2018, from <https://garagefarm.net/>



