UNIVERSIDAD NACIONAL MICAELA BASTIDAS DE APURÍMAC

FACULTAD DE INGENIERÍA

ESCUELA ACADÉMICO PROFESIONAL DE INGENIERÍA INFORMÁTICA Y SISTEMAS

"RENDERIZACIÓN DEL MODELADO 3D CON EL FRAMEWORK THREEJS DEL TEMPLO COLONIAL DE CHUQUINGA EN DISPOSITIVOS MÓVILES, AYMARAES - APURÍMAC 2016"

TESIS PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO INFORMÁTICO Y SISTEMAS

PRESENTADO POR:

BACH. PIMENTEL PALOMINO, MICHAEL ALEXANDER

ABANCAY – APURÍMAC - PERÚ

2018

UNIVERSIDAD NACIONAL MICAELA BASTIDAS DE APURÍMAC

FACULTAD DE INGENIERÍA

ESCUELA ACADÉMICO PROFESIONAL DE INGENIERÍA

INFORMÁTICA Y SISTEMAS

RENDERIZACIÓN DEL MODELADO 3D CON EL FRAMEWORK THREEJS DEL TEMPLO COLONIAL DE CHUQUINGA EN DISPOSITIVOS MÓVILES, AYMARAES – APURÍMAC 2016

Presentado por el bachiller: PIMENTEL PALOMINO MICHAEL ALEXANDER, a la Escuela Académico Profesional de Ingeniería Informática y Sistemas, sustentado y aprobado el 25 de mayo del 2018 para obtener el titulo profesional de:

INGENIERO INFORMÁTICO Y SISTEMAS

Ante el jurado integrado por:

Presidente :

Mg. Erecli Ordonez Ramos

Primer miembro :

Mg. Marleny Peralta Ascue

Segundo miembro :

Ing. Ebert Gómez Aiquipa

Asesor :

M.Sc. Ecler Mamani Vilca

DEDICATORIA:

A mis queridos padres: Asunta Palomino Sánchez y Alejandrino Pimentel Patiño, por ser la fuente de mi motivación e inspiración para superarme profesionalmente; por creer y confiar siempre en mí apoyándome en todas las decisiones que he tomado en la vida.

A mis hermanos, por su paciencia, comprensión y motivación; aportes fundamentales para terminar este trabajo.

AGRADECIMIENTO:

A mis padres, por haberme brindado una educación de calidad y confiar en mis capacidades. No defraudare sus expectativas.

A mis hermanos y hermanas, Carlos, Héctor, Fernando, Alicia, Norka, Elizabeth, Tania, a quienes exhorto a mantener una visión de éxito en sus vida mediante el estudios continuo.

A mi asesor de tesis Ing. **Ecler Mamani Vilca**, por sus sugerencias, consejos y enseñanzas, que me sirvió mucho para afirmar este trabajo. En ellos a la "UNIVERSIDAD NACIONAL MICAELA BASTIDAS DE APURÍMAC" por haberme formado profesionalmente.

ÍNDICE

ÍNDICE DE TABLAS

ÍNDICE DE GRÁFICOS

RESUMEN

La investigación se centró en conocer de qué manera influye la memoria interna, memoria RAM y GPU de los dispositivos móviles en el tiempo de renderización y la frecuencia de imágenes (FPS) del modelado 3D con el Framework ThreeJS del Templo Colonial de Chuquinga Aymaraes Apurímac 2016.

Para el modelado 3D del Templo Colonial de Chuquinga, se utilizó herramientas como: AutoCAD, con la cual se generaron las medidas adecuadas en base al plano catastral facilitado por la Municipalidad Provincial de Aymaraes, la librería ThreeJS, para crear objetos 3D, 3ds Max para el modelado de los muros del templo y WebGL para el renderizado final.

El tipo de investigación es aplicada, con un nivel explicativo y un diseño experimental con 30 corridas (dispositivos móviles), tres factores (memoria interna, memoria RAM y GPU) y dos niveles (básico y óptimo), donde se evaluó el tiempo de renderización y la frecuencia de imágenes (FPS).

Teniendo como resultado de la investigación a un 95% de confiabilidad se obtuvo: Que el factor o característica más influyente en el tiempo de renderizado es la GPU debido a que disminuye el tiempo de renderizado cuando la frecuencia del reloj central es mayor a 500 Mhz en todos los dispositivos móviles que tienen un nivel óptimo, mientras que en dispositivos de nivel básico cuando disminuye los Mhz de la GPU, se incrementa el tiempo de renderizado ocasionando que el usuario demore en acceder al aplicativo. También se encontró que el tiempo promedio de renderizado es 1.56 segundos.

En la frecuencia de imágenes (FPS), el promedio es de 52.29 FPS y el factor o característica más influyente es la GPU es decir, aumenta la frecuencia de imágenes (FPS) cuando el reloj central es mayor a 500 Mhz en todos los dispositivos móviles que tienen un nivel óptimo, mientras que en

dispositivos de nivel básico cuando disminuye los Mhz de la GPU, baja la frecuencia de imágenes (FPS) sin embargo no ocasiona distorsión en las animaciones y texturas.

Palabras claves: Modelado 3D, WebGL, tiempo de renderización, frecuencia de imágenes (FPS), dispositivos móviles, ThreeJS, niveles.

ABSTRACT

The research focused on knowing how the internal memory, RAM memory and GPU of mobile devices influence the rendering time and image frequency (FPS) of 3D modeling with the ThreeJS Framework of the Colonial Temple of Chuquinga Aymaraes - Apurímac 2016.

For the 3D modeling of the Colonial Temple of Chuquinga, tools were used such as: AutoCAD, with which the appropriate measures were generated based on the cadastral plan facilitated by the Provincial Municipality of Aymaraes, the ThreeJS library, to create 3D objects, 3ds Max for the modeling of the temple walls and WebGL for the final rendering.

The type of research is applied, with a correlational-explanatory level and an experimental design with 30 runs (mobile devices), three factors (internal memory, RAM and GPU) and two levels (basic and optimal), where time was evaluated of rendering and the frequency of images (FPS).

Taking as a result of the research a 95% reliability was obtained: That the factor or characteristic more influential in the rendering time is the GPU because it decreases the rendering time when the frequency of the central clock is higher than 500 Mhz in all mobile devices that have an optimal level, while in basic level devices when the Mhz of the GPU decreases, the rendering time increases, causing the user to delay in accessing the application. It was also found that the average rendering time is 1.56 seconds.

In the frequency of images (FPS), the average is of 52.29 FPS and the factor or characteristic more influential is the GPU that is to say, it increases the frequency of images (FPS) when the central clock is greater than 500 Mhz in all the mobile devices that have an optimal level,

while in devices of basic level when it diminishes the Mhz of the GPU, lowers the frequency of images (FPS) nevertheless does not cause distortion in the animations and textures.

Keywords: 3D modeling, WebGL, rendering time, image frequency (FPS), mobile devices, ThreeJS, levels.

INTRODUCCIÓN

Hoy en día, la mayoría de las personas optan por adquirir dispositivos móviles debido a su gran portabilidad, estos permiten visualizar aplicaciones 3D en sitios web móvil, logrando que los usuarios puedan darse cuenta de las limitaciones de las características del dispositivo móvil y los conflictos que generan al ejecutar aplicativos interactivos 3D como: la demora en acceder a estos sitios, problemas en el proceso de carga de textura, proceso de renderizado lento, relentización de procesos que en ese momento se están ejecutando, poca fluidez en animaciones, etc. En consecuencia, los usuarios quedan poco satisfecho con el funcionamiento al ver que sus equipos no cumplen con los requisitos para ver sitio web móvil en 3D.

Por ello, se desarrolló un aplicativo web 3D del Templo Colonial de Chuquinga con el Framework ThreeJS para evaluar cómo las características de los dispositivos móviles (la memoria interna, memoria RAM y la GPU) influyen en el tiempo de renderización y la frecuencia de imágenes (FPS).

Como resultados de esta investigación al 95% de confiabilidad se obtuvo que la GPU es la característica más importante de los dispositivos móviles que influyen en el tiempo de renderizado, registrando un tiempo promedio con características óptimas de 1.55 segundos y con características básicas de 2.53 segundos y para la frecuencia imágenes (FPS) utilizando los mismos factores se obtuvo la GPU es la característica que más influye en la frecuencia de imágenes (FPS) donde los dispositivos con nivel óptimo registra 52.29 FPS y con nivel básico 32.76 FPS.

Para culminar, este trabajo tiene una estructura dividida en 6 capítulos, los cuales son: Planteamiento del problema, objetivos, marco referencial, hipótesis y variables, metodología de investigación, resultados, conclusiones, recomendaciones y anexo.

CAPÍTULO I

PLANTEAMIENTO DEL PROBLEMA

1.1 Definición y formulación del problema

En la actualidad, existe una gran demanda de aplicaciones web móviles y una de las más conocidas son los aplicativos interactivos 3D, los cuales al ejecutarse en dispositivos móviles con características básicas traen inconvenientes como: En la memoria interna, cuando está saturada demora en acceder a aplicaciones 3D, problemas en el proceso de carga de textura, proceso de renderizado lento; en la memoria RAM (Random Access Memory) al gestionar múltiples aplicaciones en segundo plano ésta se satura y se vuelven lentos los procesos que se están ejecutando en ese momento, impidiendo la fluidez del aplicativo; y finalmente la GPU (Unidad de procesamiento gráfico), una de las parte más importantes del móvil, dedica exclusivamente al procesamiento gráfico, si tiene bajo Mhz o no posee GPU, tiene poca fluidez en la animación, poco realismo para crear texturas complejas debido a los cálculos grandes que éstas realizan. En consecuencia, los usuarios quedan poco satisfecho con el funcionamiento, al ver que sus equipos no cumplen con los requisitos.

Es importante conocer que a lo largo del tiempo los gráficos en la web que se muestran en dispositivos móviles han experimentado diferentes cambios; anteriormente solo eran textos hasta que apareció el software Macromedia flash, siendo el comienzo de una nueva era de las páginas web interactivas, utilizando plugins o recursos externos como adobe Flash o Applets de Java, para aprovechar la aceleración mediante GPU, permitiendo un mejor renderizado y animaciones en aplicaciones web interactivas para dispositivos móviles en 3D, utilizando grandes cantidades de recursos de hardware, lo que genera que las aplicaciones web para dispositivos móviles, tengan problemas en su funcionamiento las cuales son: Renderización

no completada en dispositivos móviles, esto se produce debido a la complejidad de objeto del escenario y realismo con imágenes de alta calidad del aplicativo, en consecuencia el móvil realiza más trabajo para mostrar el resultado final, produciéndose en muchos casos ante la demora del renderizado, cierre inesperado del aplicativo, pantalla negra sin imagen ni animación.

Cuando el aplicativo web ya realizó la renderización en el dispositivo es decir ya está en funcionamiento, la frecuencia de imágenes (FPS) en un dispositivo son bajos y menores a 20 FPS, generando animaciones poco fluidas. También se puede observar que la bajada drástica de fotogramas ocasiona distorsión de la animación en la escena 3D.

Por lo cual se propone evaluar la renderización y la frecuencia de imágenes (FPS) del modelado 3D con el Framework ThreeJS del Templo Colonial de Chuquinga en dispositivos móviles, en relación a su memoria interna, RAM y GPU.

En ese atender nos lleva a formular las siguientes interrogantes:

1.1.1 Problema general

¿De qué manera los dispositivos móviles, influyen significativamente en el tiempo de renderización del modelado 3D con el Framework ThreeJS del Templo Colonial de Chuquinga Aymaraes - Apurímac 2016?

1.1.2 Problemas específicos

- ¿De qué manera la memoria interna de los dispositivos móviles influyen significativamente en el tiempo de renderización del modelado 3D con el Framework ThreeJS del Templo Colonial de Chuquinga, Aymaraes- Apurímac 2016?

- ¿De qué manera la memoria RAM de los dispositivos móviles influyen significativamente en el tiempo de renderización del modelado 3D con el Framework ThreeJS del Templo Colonial de Chuquinga, Aymaraes- Apurímac 2016?
- ¿De qué manera la GPU de los dispositivos móviles influyen significativamente en el tiempo de renderización del modelado 3D con el Framework ThreeJS del Templo Colonial de Chuquinga, Aymaraes- Apurímac 2016?
- ¿De qué manera la memoria interna de los dispositivos móviles influyen significativamente en la frecuencia de imágenes por segundos del modelado 3D con el Framework ThreeJS del Templo Colonial de Chuquinga, Aymaraes- Apurímac 2016?
- ¿De qué manera la memoria RAM de los dispositivos móviles influyen significativamente en la frecuencia de imágenes por segundos del modelado 3D con el Framework ThreeJS del Templo Colonial de Chuquinga, Aymaraes- Apurímac 2016?
- ¿De qué manera la GPU de los dispositivos móviles influyen significativamente en la frecuencia de imágenes por segundos del modelado 3D con el Framework ThreeJS del Templo Colonial de Chuquinga, Aymaraes- Apurímac 2016?

1.2 Justificación

Esta investigación pretende evaluar la renderización del modelado 3D con el Framework ThreeJS del Templo Colonial de Chuquinga en dispositivos móviles, para tener un tiempo óptimo y mejorar la velocidad de fotogramas por segundo en función a las características de los dispositivos móviles como, la memoria interna, la RAM y la GPU.

La importancia de este estudio es reunir toda la información del renderizado y frecuencia de imágenes (FPS) del modelado 3D de distintos dispositivos móviles con esta nueva tecnología y saber las características necesarias que debe de tener el dispositivo para su ejecución.

ThreeJS y su motor de Render WebGL, facilitó la creación de modelados 3D en distintos formatos como obj (formato de archivos para objetos tridimensionales), Collada dae (formato xml para almacenar datos 3D), FBX (formato de archivo 3D independiente de la plataforma que proporciona acceso al contenido de cualquier paquete de software), etc; que luego de convertirlos al formato JSON (formato de texto ligero para intercambio de datos) permitieron un mejor renderizado en los dispositivos móviles por ser un formato ligero de intercambio 3D, sin utilizar demasiada cantidad de recursos. Además para su funcionamiento de la aplicación web 3D en dispositivos móviles, no se hizo uso de plugins adicionales como en los casos tradicionales.

La contribución de esta investigación en lo académico es generar conocimientos de los pasos a seguir para desarrollar aplicaciones web 3D en dispositivos móviles y conocer que formatos de intercambio 3D son óptimos con el uso del Framework ThreeJS en dispositivos móviles para un óptimo renderizado usando recurso gratuitos es decir con licencia libre.

1.3 Limitación

A. El desarrollo del aplicativo web 3D se limitó en la utilización de imágenes con calidad media debido al costo de la cámara profesional.

B. El detalle de las texturas del Templo Colonial de Chuquinga en el modelado se limitó por motivos de posición de la toma de fotos, reflejos de luz y el poco tiempo de visita al templo.

C. La aplicación web 3D funcionó en versiones recientes de navegadores como:

- Superiores a la versión de Google Chrome 10.0.648.127.
- Superiores a la versión de Internet Explorer 9.
- Superiores a la versión Mozilla Firefox versión 4.0. 54.

Debido a que en versiones antiguas de navegadores no soportan WebGL.

CAPÍTULO II
OBJETIVOS

2.1 Objetivo

2.1.1 Objetivo general

Evaluar cómo los dispositivos móviles, influyen significativamente en el tiempo de renderización del modelado 3D con el Framework ThreeJS del Templo Colonial de Chuquinga Aymaraes - Apurímac 2016

2.1.2 Objetivos específicos

- Evaluar cómo la memoria interna, influye significativamente en el tiempo de renderización del modelado 3D con el Framework ThreeJS del Templo Colonial de Chuquinga Aymaraes - Apurímac 2016
- Evaluar cómo la memoria RAM, influye significativamente en el tiempo de renderización del modelado 3D con el Framework ThreeJS del Templo Colonial de Chuquinga Aymaraes - Apurímac 2016
- Evaluar cómo la GPU, influye significativamente en el tiempo de renderización del modelado 3D con el Framework ThreeJS del Templo Colonial de Chuquinga Aymaraes - Apurímac 2016
- Evaluar cómo la memoria interna, influye significativamente en la frecuencia de imágenes por segundos del modelado 3D con el Framework ThreeJS del Templo Colonial de Chuquinga Aymaraes - Apurímac 2016
- Evaluar cómo la memoria RAM, influye significativamente en la frecuencia de imágenes por segundos del modelado 3D con el Framework ThreeJS del Templo Colonial de Chuquinga Aymaraes - Apurímac 2016

- Evaluar cómo la GPU, influye significativamente en la frecuencia de imágenes por segundos del modelado 3D con el Framework ThreeJS del Templo Colonial de Chuquinga Aymaraes - Apurímac 2016

CAPÍTULO III
MARCO REFERENCIAL

3.1 Antecedentes de la investigación

3.1.1 En el exterior

A. En el año 2014, Javier Villamonte Pereira, realizó la tesis para optar el título de Ingeniero de Informática y Sistemas, que lleva como título **"Gráficos 3D en Interfaces Web"**, de la universidad San Cristóbal de la Laguna, en Tenerife.

En esta tesis el principal problema que describe es la visualización de sitio desigual en los distintos navegadores y el poco control sobre el diseño de parte de los desarrolladores.

El objetivo central de este trabajo es realizar un estudio del estado de los gráficos 3D en Interfaces Web, comparando las distintas librerías disponibles. Para poder hacer una mejor valoración del estado de los gráficos 3D en Interfaces Web, una meta practica en cuya ejecución sea posible comparar y valorar cualitativamente las características de cada una de las herramientas.

Durante las fases de análisis previas al desarrollo se han evaluado distintas librerías, con algunas de las cuales se han hecho pruebas, lo que ha permitido extraer sus fortalezas y debilidades. Además, los requerimientos del proceso han servido para saber qué es lo que se puede hacer con este tipo de tecnologías. El resultado final fue la explicación de lo que ocurre internamente y cómo se relacionan las clases en cada caso de uso en los gráficos 3D haciendo uso de las librerías.

B. A finales del año 2011, Jorge Izquierdo Ciges, realizo la tesis para optar el título de Ingeniero de Informática, que lleva como título **"Visualización interactiva de escenas tridimensionales con OpenSceneGraph sobre dispositivos Android"**, de la Universidad Politécnica de Valencia, en España.

En esta tesis el principal problema que describe es la representación interactiva de escenas en Android. Esto limita inicialmente a la tasa de representación que se debe alcanzar como mínimo de diez frames por segundo. Aunque Esta tasa es muy limitada lo cual ocasiona que no se ajuste a ella en cualquier situación, siempre existe un punto en el que una escena será incapaz de visualizarse en ese límite. El objetivo central de este trabajo es "portabilizar" la librería OSG y emplearla, sin ninguna arquitectura externa o de apoyo, para representar escenas complejas con una tasa de dibujo interactiva. Esto se realizará empleando el sistema operativo Android. Como resultados de esta investigación se realizaron una serie de cambios en los archivos del código fuente de la librería que arreglaban los problemas provocados por las diferentes distribuciones de Linux y Android. Además de los cambios para la incompatibilidad, se generaron scripts de compilación NDK Android para OSG. Debido a que la librería empleada para gestionar la cadena de compilación y con el objetivo de no añadir complejidad al mantenimiento se generaron funciones y macros en CMake que permiten generar archivos compilados NDK Android desde los scripts CMake originales.

C. En el año 2014, Ana María Quinteros Gomez, realizó la tesis para optar el título de Ingeniera Física que lleva como título **"Metodología para la evaluación de técnicas de renderizado 3D en sistemas de visualización de imágenes médicas"**, de la Universidad Tecnológica de Pereira Facultad de Ingenierías Eléctrica, Electrónicas, Física y Ciencias de la Computación de Pereira, en Colombia.

En esta tesis el principal problema que describe es la inexistencia de una técnica única para generar imágenes tridimensionales en un sistema de visualización médica, puesto que cada técnica depende de la aplicación, el objetivo central de este trabajo es el diseño de una metodología para la evaluación entre técnicas de renderizado 3D en un sistema y su objetivo específico es identificar las diferentes técnicas, para implementar el renderizado 3D.

Las conclusiones son la comparación entre técnicas de rendering para imágenes diagnóstico, como lo fueron en este caso Matlab y C# debido a esto se encontró que la técnicas de rendering de volumen son más usados y ofrecen una mejor visualización de lo que se requiere estudiar también se pudo realizar reconstrucciones de diferentes partes del paciente en este caso de la cabeza, no solo de la capa externa. También se llegó a la conclusión que la ventaja del rendering de superficie en C# es la buena calidad de imágenes pero poco precisa ya que solo muestra el exterior de la imagen y eso hace que se desperdicie espacio de estudio.

D. En el año 2015, Rafael Gaitán **"Visualización interactiva 3D en dispositivos móvil"**, Univ. Politécnica de Valencia.

En esta tesis el principal problema que describe es la limitación de los dispositivos móviles referidos a la potencia de cálculo como de tamaño de la memoria y a los dispositivos que se encuentra en mercado que disponen de aceleración gráfica por hardware.

El objetivo central es desarrollar nuevos modelos que permitan almacenar grandes escenas en memoria secundaria, de modo que se pueden visualizar, de forma interactiva, conjuntos de datos enormes utilizando técnicas de simplificación dependiente de la vista. Todas estas técnicas pueden ser aplicadas a la visualización de escenas 3D complejas en dispositivos móviles, debido a las carencias de potencia de cálculo y de memoria que tienen. Existen, principalmente, tres técnicas para visualizar en 3D sobre dispositivos móviles: visualización basada en la imagen, que utiliza imágenes precalculadas para sustituir geometría y obtener efectos realistas a bajo coste visualización basada en el punto, que visualiza geometrías complejas procesando un conjunto de puntos sobre la superficie de los objetos y visualización basada en polígonos.

En el desarrollo de esta tesis se utilizó la técnica de multirresolución, Los resultados obtenidos muestran un número de imágenes por segundo que varía mucho en las escenas, esto es debido a la cantidad de geometría visible que hay en cada momento, y por lo tanto se realizan muchos envíos de geometría, de ahí los dientes de sierra que aparecen en la gráfica, en el futuro sería interesante mejorar el funcionamiento de la caché de geometría, de modo que se puedan

almacenar en ella nodos multirresolución en vez de triángulos, utilizando técnicas de simplificación dependiente de la vista.

3.2 Marco Teórico

3.2.1 Conceptos iniciales sobre el modelado en 3D

A. Modelo en 3D

Es un formato que contiene la información necesaria para renderizar un objeto en 3 Dimensiones. Este archivo contiene dos tipos de información:

- **La geometría.** Forma del objeto.
- **Los atributos de la superficie.** Es el contenido del objeto 3D que permite que esté perfectamente coloreado de modo que aparente estar hecho de un determinado material. La información de la geometría del modelo define la superficie del objeto como una lista de polígonos planos que comparten lados y vértices. Cuando modelemos, si el objeto 3D no comparte lados y vértices comunes, habrá roturas en el mismo, y se perderá la sensación de continuidad en su superficie (Escobedo, 2013).

B. Tipos de modelos 3D

a) Modelos representados por polígonos

Es uno de los sistemas utilizado por el ordenador para representar cualquier objeto.

b) Modelos definidos por sus curvas matemáticas: NURBS y Patch

Son superficies curvas definidas matemáticamente. Se representan como una figura poligonal de muchos lados, pero también representarse como una función matemática entre dos variables x e y.

Evidentemente el usuario no tiene que vérselas con engorrosas fórmulas, sino que de la misma forma que en un programa vectorial resulta sencillo trazar curvas en un modelador no poligonal se disponen de diferentes tipos de herramientas para crear superficies curvas complejas (Escobedo, 2013).

C. La escena

La escena en 3D es el archivo que contiene toda la información necesaria para identificar y posicionar todos los modelos, luces y cámaras para su renderización (Escobedo, 2013).

D. Sistemas de coordenadas

Una escena puede identificarse con las coordenadas en 3 dimensiones del espacio en las cuales tiene lugar la renderización. Este espacio a menudo se llama sistema de coordenadas universal. Pero al operar con los objetos de la escena, las cuales se utilizar diferentes sistemas de coordenadas (Escobedo, 2013).

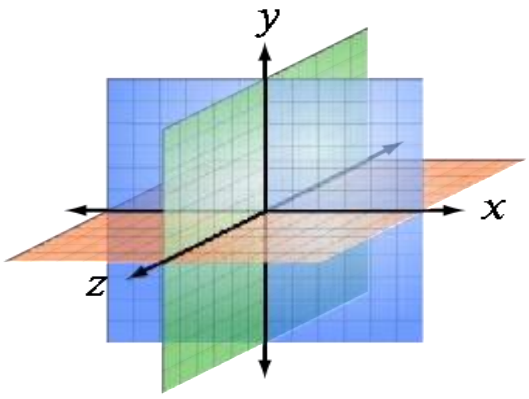


FIGURA N° 1 : Sistema de coordenadas
Fuente: Escobedo, (2013).

Eje X: siempre hacia la derecha de la vista.

Eje Y: siempre apunta hacia arriba.

Eje Z: siempre apunta hacia el usuario.

E. Sistema de coordenadas local a un objeto

Sistemas de coordenadas locales. Cada objeto tiene su sistema de coordenadas, evidenciado en la imagen por sus pivotes.

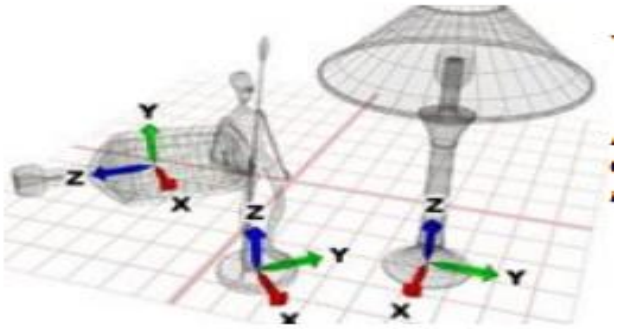


FIGURA N° 2 : Sistema de coordenadas local.
Fuente: Escobedo, (2013).

- **Iluminación de una escena**

La iluminación correcta de la escena es fundamental para imprimirle realismo. La iluminación se trata de colocar las luces adecuadas y modificar sus parámetros para obtener el resultado deseado (Escobedo, 2013).

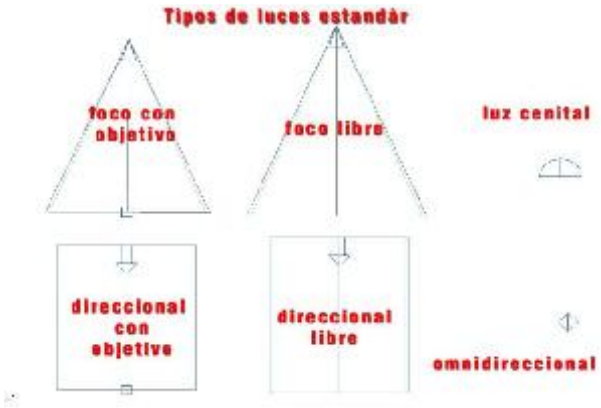


FIGURA N° 3 : Tipos de luces.
Fuente: Escobedo, (2013).

F. Materiales y texturas

Cuando modelamos un objeto, su superficie queda por defecto de un color uniforme y liso. Con los materiales y texturas haremos que adquiera el realismo necesario.

Los materiales junto con la iluminación, forma una parte esencial para la realización de imágenes realistas, ya que los materiales son los que dan vida a una escena, ya que no es lo mismo crear una esfera con un material predeterminado, que representar la misma esfera con un material tipo vidrio, metálico, etcétera. Una parte esencial para la creación de materiales son los mapas, o imágenes de referencia las cuales son la parte medular de un material, ya que de estás depende la calidad del material. La aplicación de materiales va de la mano del modificador Mapa UVW, para la correcta visualización de los materiales. Los materiales no únicamente sirven para dotar de vida a una escena u objeto, mediante ellos también se pueden lograr efectos como: desplazamientos, relieve (Escobedo, 2013).

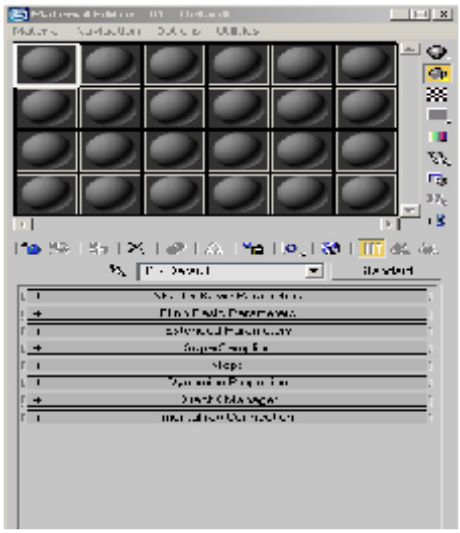


FIGURA N° 4 : Editor de materiales de 3Ds Max.
Fuente: Escobedo, (2013).

G. Render

Es el proceso final que genera la imagen o animación a partir de la escena creada. Esto puede ser comparado con tomar una foto o filmar una escena real. El software puede simular efectos cinematográficos producidos por las imperfecciones mecánicas de la fotografía real, aportando realismo con su presencia debido a la costumbre de los seres humanos a ellos. Esta fase requiere una gran capacidad de cómputo ya que simula procesos físicos complejos, por lo tanto, es natural que con la mejora de la capacidad de los ordenadores a lo largo del tiempo también mejorara el realismo de los render.

El render es un proceso necesario para experimentar los efectos de la iluminación, así como sus ajustes para una mayor simulación de la realidad, puede decirse que el render es en base a experimentación, el cual va de la mano de la iluminación y visualización de materiales (Escobedo,2013).

FIGURA N° 5 : *Casa renderizado con Mental Ray.*
Fuente: Escobedo, (2013).

H. Cámaras

Las cámaras son objetos que nos permiten observar una escena desde una determinada posición y vista mediante el visor cámara, los objetos cámaras son similares a las cámaras de la vida real, mediante las cámaras se pueden generar imágenes fijas, o recorridos a lo largo de la escena generando videos (Escobedo, 2013).

- **Target camera** (Cámara con objetivo).- Este tipo de cámaras está conformado por dos elementos que son la cámara y el objetivo, en general es la más usada ya que se puede animar fácilmente, únicamente moviendo el objetivo, aunque la cámara también se puede animar o transformar.
- **Free camera** (Cámara libre).- Este tipo de cámara está compuesto de un sólo elemento que es la cámara, la cual carece de objetivo, por consiguiente, únicamente encuadra hacia donde apunta la cámara, este tipo de cámara es ideal para crear recorridos a través de escenarios (Escobedo, 2013).

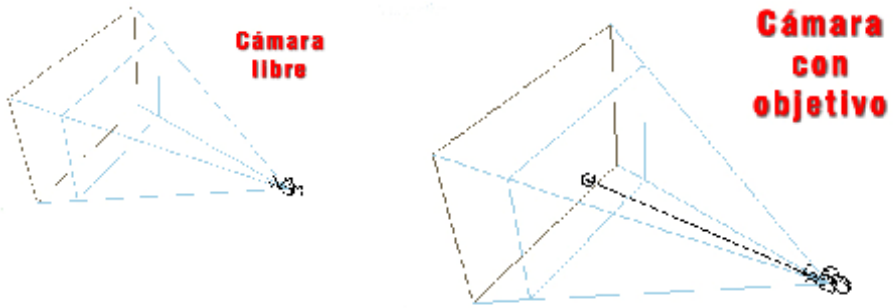


FIGURA N° 6 : *Tipo de cámara con objetivo.*
Fuente: Escobedo, (2013).

I. **Extensión de archivo**

- **Collada (COLLAborative Design Activity).-** Es un formato de archivo de intercambio de 3D utilizado para el intercambio de activos digitales entre varios programas de gráficos.
- **JSON (JavaScript Object Notation)**.- Es un formato de archivo de intercambio de 3D, ligero y rápido (por tanto muy útil para desarrollos web). Los datos en formato JSON pueden ser utilizados por prácticamente todos los lenguajes de programación (como Java, C#, C, C++, PHP, JavaScript, Python, etc.).

3.2.2 Transformaciones y movimientos

- **Move.-** Nos permite desplazar cualquier objeto dentro de los distintos visores, al momento de seleccionar un objeto y tratar de seleccionar ejes que nos permite restringir, en que eje se requiere desplazar (Escobedo, 2013).
- **Rotate.-** Nos permite rotar objetos en cualquier de los distintos ejes, al momento de seleccionar aparecen una serie de círculos los cuales representan los distintos ejes (Escobedo, 2013).
- **Scale.-** Permite reducir o aumentar el tamaño de un objeto, ya sea conservando la proporción, o únicamente en algún eje. Hay 3 formas distintas de escalar objetos (Escobedo, 2013).

  a) Escala uniformemente.- Los objetos reducen o aumentan su tamaño de forma uniforme en los 3 ejes conservando las proporciones del objeto, cuando se requiere escalar una determinada cantidad (Escobedo, 2013).

b) Escala no uniformemente.- Nos permite reducir o incrementar el tamaño de un objeto dependiendo del eje de restricción que esté activado (Escobedo, 2013).

3.2.3 Conceptos de geometría en modelado 3D

A. Polígonos

En modelación, consideramos un polígono a cualquier forma plana y cerrada (con su primer y último vértice perfectamente coincidentes). Un polígono también puede ser una figura 2D, una forma cerrada cuyos primer y último vértice coinciden (Escobedo, 2013).

B. Creación de polígonos regulares

En un programa 3D, tendremos la opción de crear shapes 2D como polígonos regulares. En 3ds Max se llama Ngon, y modificando los parámetros desde el panel de comandos podremos decidir de cuántos lados deseamos crear el polígono, entre otras opciones (Escobedo, 2013).

3.2.4 Vértice de un polígono

Es el punto que define el inicio o final de un segmento. Dentro de la figura viene representado por un pequeño trazo que corta perpendicularmente a su línea de contorno. Todo segmento tiene un vértice final y otro final situados en los extremos del mismo. En un objeto sólido, los vértices de los polígonos forman parte de la malla y suponen puntos en común entre los diferentes polígonos del objeto 3D (Escobedo, 2013).

3.2.5 Lenguaje de programación

Es un conjunto de órdenes o comandos que describen el proceso deseado. Cada lenguaje tiene sus instrucciones y enunciados verbales propios, que combinan para formar los programas de cómputo (Kernighan & Ritchie, 1991).

Los lenguajes de programación no son aplicaciones, sino herramientas que permite construir y adecuar aplicaciones, entre uno de ellos tenemos a:

A. JavaScript

Es un lenguaje de lado del cliente, implementado como parte del navegador web, permitiendo mejoras en la interfaz del usuario. JavaScript permite a los navegadores ser capaces de reconocer objetos en una página HTML a través del DOM. (Powell, 2002).

3.2.6 WebGL

WebGL es una especificación estándar que está siendo desarrollada actualmente para mostrar gráficos en 3D en navegadores web. WebGL permite mostrar gráficos en 3D acelerados por hardware (GPU) en páginas web, sin la necesidad de plugins en cualquier plataforma que soporte OpenGL 2.0 u OpenGL ES 2.0.

A diferencia de otras tecnologías de gráficos 3D, como OpenGL y Direct3D, con WebGL se pueden construir páginas web para que puedan ser ejecutadas directamente en los navegadores sin necesidad de instalar ningún plugins o bibliotecas especializadas (Matsuda & Lea ,2013).

3.2.7 Librerías

Es un conjunto de subprogramas utilizados para desarrollar software. Las bibliotecas contienen código y datos, que proporcionan servicios a programas independientes, es decir, pasan a formar parte de éstos. Esto permite que el código y los datos se compartan y puedan modificarse de forma modular (Dirksen, 2015).

A. ThreeJS

Es una librería bastante liviana y muy eficiente para generar y animar gráficos en 3D dentro del navegador, aprovechando las grandes novedades que nos ofrece HTML5 para la generación de contenidos multimedia.

Aprovecha tanto las capacidades de HTML5 que es capaz de generar escenas 3D con WebGL, Canvas (2D) y SVG (Dirksen, 2015).

- **Requisitos para ThreeJS**

  ThreeJS no tiene ninguna dependencia a otras bibliotecas, por lo que se puede ejecutarse de manera independiente. Sin embargo, para aprovechar ThreeJS al máximo, necesita un navegador que admita el estándar WebGL (Dirksen, 2014).

  La siguiente tabla proporciona una descripción de los navegadores que son compatibles.

TABLA N° 1 : *Navegadores compatibles con ThreeJS*

Navegador	Descripción
Internet Explore	Admite WebGL desde la versión 11.
Mozila Firefox	WebGL es compatible con la versión 4.
Google Chrome	Admite WebGL desde la versión 10
Safari	Admite WebGL desde la versión 5.1 y superiores.
Opera	Admite WebGL desde la versión 12.0.

Fuente: Elaboración propia.

3.2.8 Software

A. 3ds Max

Es un software en el cual es posible realizar cualquier objeto en 3 dimensiones, desde el más simple hasta el más complejo y fantasioso objeto que se deseé, para después representarlo en formato de imágenes, o en formato de animación ya sea como una secuencia de imágenes o en formato de video, además permite la creación de efectos visuales, creación de gráficos para video juegos, se puede decir que cualquier cosa que imagine se puede realizar con 3ds Max, la única restricción para hacer cualquier cosa es la de conocer a fondo el programa y sus diferentes características, así como la creatividad que cada persona posea, para poder desarrollar aquellos elementos de carácter irreal , así como la de tratar de emular elementos de la vida real(Escobedo,2013).

Como 3ds Max es un software para la creación y previsualización de elementos visuales, es usado en diferentes campos de la vida en general, entre los más comunes son: arquitectura para ver las edificaciones antes de construirlas sobre el terreno, cine y televisión en la realización de comerciales donde se incluya personajes ficticios en entornos reales, en efectos especiales, con la creación de efectos que

sería imposible realizarlos en la vida real debido a su peligrosidad, modelos para video juegos, hoy en día la mayor parte de los video juegos incluyen gráficas en 3 dimensiones, que son realizados en las diferentes softwares de diseño tridimensional.

B. AutoCAD

Es una aplicación universal de diseño y dibujo asistido por computadora. Las aplicaciones de CAD son herramientas muy potentes. La velocidad y facilidad con las que un dibujo puede ser preparado y modificado con un ordenador presenta enorme ventaja frente al dibujo manual. AutoCAD ofrece esta sofisticada tecnología a los usuarios de ordenadores personales (Olivier, 2009).

Con AutoCAD se puede crear prácticamente todo tipo de diseños. AutoCAD se puede utilizar en numerosos campos de aplicación:

- Dibujos arquitectónicos variados.
- Dibujos de aplicaciones electrónicas, químicas, ingeniería civil, mecánica y aeroespacial así como en la industria de la automoción,
- Diseño naval.
- Dibujos de arquitectura de interiores.
- Organigrama y gráficos de toda clase.
- Proyectos y presentaciones.
- Industria técnica y planos de montaje.
- Logos de empresa.

3.2.9 Dispositivos móviles

Aparato de pequeño tamaño, con algunas capacidades de procesamiento, con conexión permanente o intermitente a una red, con memoria limitada, que ha sido diseñado específicamente para una función, pero que puede llevar a cabo otras funciones más generales (Morillo, 2005).

Se trata de todo aquel dispositivo de fácil traslado, que dispone de determinadas presentaciones que ayudan en el desempeño diario de determinadas presentaciones que nos ayudan en el desempeño diario de nuestras actividades laborales y/o académicas (Luna, 2014).

A. Tipos de dispositivos móviles

El término dispositivo móvil cubre un amplio rango de dispositivos electrónicos de consumo. Normalmente, por dispositivo móvil nos referimos a un dispositivo que puede conectarse a internet. Algunos de estos tipos dispositivos son los siguientes (Morillo, 2005).

- Teléfonos móviles.
- Smartphones.
- Tablet PC
- Tablet.

B. Características generales de los dispositivos móviles

- **Memoria interna**

  Es el espacio donde se almacena el sistema operativo, aplicaciones nativas, software y otras funcionalidades con las que el equipo saldrá a la venta, dejando un campo libre para que el usuario guarde más información. Su capacidad de almacenaje se presenta en GB (Gigabytes) (Prado & Lamas, 2014).

- **Memoria RAM (Random Access Memory)**

  Esta es una memoria muy rápida, tradicionalmente utilizada para almacenar aplicaciones y datos temporales mientras utilizamos el dispositivo. Su capacidad de almacenaje se presenta en GB (Gigabytes) y MB (megabit) (Prado & Lanas, 2014).

  Los elementos que componen la memoria RAM, son:

  - ✓ **Registro de intercambio.** Recibe los datos en operaciones de lectura y almacena los datos en las operaciones de escritura.
  - ✓ **Registro de direcciones.** Contiene las direcciones de la celda o posición de la memoria a la que se va a acceder.
  - ✓ **Sector de memoria.** Se activa cada vez que hay que leer o escribir conectando la celda o posición de memoria con el registro de intercambio.
  - ✓ **Señal de control.** Indica si una operación es de lectura o escritura.

- **GPU (unidad de procesamiento gráfico)**

  Se trata de un procesador que se dedica exclusivamente al procesamiento de gráficos u operaciones. Lo que hace la GPU es aligerar el trabajo de la CPU, sobre todo a la hora de abrir juegos o aplicaciones con gráficos interactivos 3 D. Su velocidad del núcleo se mide en Mhz y Ghz (Prado & Lanas, 2014).

C. Web móvil

Se entiende aquella accesible y disponible desde cualquier punto en el que exista una conexión inalámbrica a la red, en cualquier momento y con cualquier tipo de dispositivo con capacidad para ello (Tosete, 2008).

También se define como un sitio cuyo diseño, navegación, contenidos y servicios están optimizados para ser accedidos y consumidos a través de un dispositivo móvil, entendiendo por dispositivo móvil cualquiera que pueda ser utilizado en movilidad. No se debe pensar en una web móvil como una versión distinta o reducida de la versión web clásica. Hay que tener en cuenta las características del dispositivo con el que se accede para adaptar la información y servicios aprovechando las ventajas de la movilidad.

3.2.10 Metodología de desarrollo orientado a prototipo

Es un tipo de metodología para el desarrollo de software en la cual se desarrolla una maqueta del producto. Esta maqueta o prototipo desarrollada por el equipo y refinada junto al cliente permite idealmente especificar los requerimientos del producto. Inicialmente se comienza con la definición de alto nivel de los requisitos junto al cliente. En base al entendimiento de los requerimientos se desarrolla un prototipo que será presentado al cliente, quien probará la maqueta y emitirá su

opinión, de acuerdo a la cual se iterará sobre el proceso descrito o se procederá a la implementación definitiva del producto. Así el prototipo evolucionará refinando los requerimientos en cada interacción hasta lograr la aprobación del cliente (Guillermo, 2015).

El objetivo principal del proceso descrito es disminuir los riesgos que podrían traer aparejada la falta de claridad de los requerimientos. Con esto se busca orientar al cliente/usuario con la definición de los requisitos.

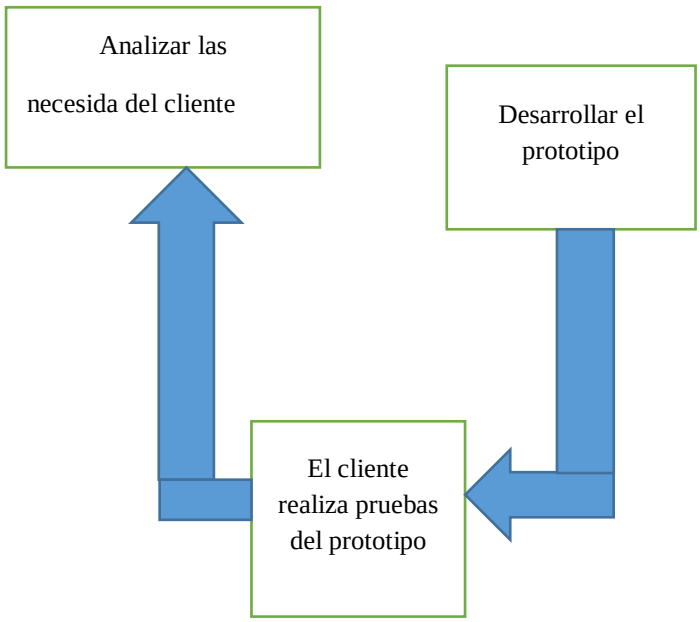


FIGURA N° 7 : Metodología de prototipo.
Fuente: Elaboración propia.

A. Fases de la metodología orientada a prototipos

1. **Investigación preliminar:** Se determina el problema, los efectos que tendrán sobre la organización, y se identifica la idea general de la solución para realizar un estudio de factibilidad.

2. **Definición de los requerimientos del sistema:** Esta fase es la más importante del ciclo de vida del método de prototipos, el objetivo es determinar todos los requerimientos y deseos de los usuarios en relación al proyecto que se está deseando implementar. Aquí el desarrollador interactúa con el usuario y sus necesidades mediante la construcción, demostración y retroalimentaciones del prototipo (Jiménez & Mendoza, 2014).

3. **Diseño técnico:** En esta etapa el sistema debe ser rediseñado y tener la respectiva documentación, guiándose en los estándares que tiene la organización la cual servirá como ayuda en mantenciones futuras del mismo. En este punto existen dos etapas :

   - Producción de una documentación de diseño la cual específica y describe la estructura del software, interfaces de usuarios, funciones y el control de flujo.
   - Producción de todo lo requerido para promover cualquier mantención futura del software.

4. **Programación y prueba:** En esta etapa es donde los cambios identificados en el diseño técnico son implementados y probados para asegurar la corrección y completitud de los mismos con respecto a los requerimientos. Las pruebas serán de realizarse tantas veces sea necesarias para verificar cualquier tipo de anomalía en el sistema (Jiménez & Mendoza, 2014).

5. **Operación y mantenimiento:** En esta fase se realiza la instalación y mantención del software, la complejidad en este caso resulta menor ya que en las etapas anteriores los usuarios han trabajado con el sistema al momento de

hacer las pruebas de prototipos, además la mantención también debería ser una fase menos importante, ya que se supone que el refinamiento del prototipo permitiría una mejor claridad en los requerimientos, mediante lo cual las mantenciones perfectivas se reducirían. Si existiese el caso en el cual se requiera una manutención, entonces el proceso de prototipo es repetido y se definirá un nuevo conjunto de requerimientos (Jiménez & Mendoza, 2014).

6. Razones para usar este modelo

Con este modelo se puede ilustrar los formatos de datos de entrada, mensajes, informes y diálogos al usuario, mediante lo cual se logra un mejor entendimiento de las necesidades. Se logra una exploración de los aspectos técnicos del producto propuesto (Jiménez & Mendoza, 2014).

Ventajas

- Existe una reducción de la incertidumbre y del riesgo.
- Se reduce el tiempo y costos.
- Hay incremento en la aceptación del nuevo sistema.
- Mejora la administración de proyectos.
- Existe mayor comunicación entre los desarrolladores y el usuario.

Desventajas

- Se depende de las herramientas de software para el éxito ya que la necesidad de disminución de incertidumbre depende de las iteraciones del prototipo, entre más iteraciones existan mejor y este último se logra mediante el uso de mejores herramientas lo que hace a este proceso dependiente de las mismas.

- No es posible usar la metodología en todos los sistemas.
- Puede existir una mala interpretación que pueden hacer los usuarios del prototipo, al cual pueden confundir con el sistema terminado.

B. Pasos para construir un prototipo de software

PASO 1: Evaluar la petición del software y determinar si el programa a desarrollar es un buen candidato para construir un prototipo. Debido a que el cliente debe interaccionar con el prototipo en los últimos pasos, es esencial que: 1) el cliente participe en la evaluación y refinamiento del prototipo, y 2) el cliente sea capaz de tomar decisiones de requerimientos de una forma oportuna. Finalmente, la naturaleza del proyecto de desarrollo tendrá una fuerte influencia en la eficacia del prototipo (Jiménez & Mendoza, 2014).

PASO 2: Dado un proyecto candidato aceptable, el analista desarrolla una representación abreviada de los requerimientos. Antes de que pueda comenzar la construcción de un prototipo, el analista debe representar los dominios funcionales y de información del programa y desarrollar un método razonable de partición. La aplicación de estos principios de análisis fundamentales, pueden realizarse mediante los métodos de análisis de requerimientos (Jiménez & Mendoza, 2014).

PASO 3: Después de que se haya revisado la representación de los requerimientos, se crea un conjunto de especificaciones de diseño abreviadas para el prototipo. El diseño debe ocurrir antes de que comience la construcción del prototipo. Sin embargo, el diseño de un prototipo se enfoca normalmente hacia la arquitectura a nivel superior y a los aspectos de diseño

de datos, en vez de hacia el diseño procedimental detallado. (Jiménez & Mendoza, 2014).

PASO 4: El software del prototipo se crea, prueba y refina idealmente, los bloques de construcción de software preexisten se utilizan para crear el prototipo de una forma rápida. Para las aplicaciones interactivas con el hombre, es posible crear un prototipo en papel que describa la interacción hombre-máquina usando una serie de hojas de historia (Jiménez & Mendoza, 2014).

PASO 5: Una vez que el prototipo ha sido probado, se presenta al cliente, el cual "conduce la prueba" de la aplicación y sugiere modificaciones. Este paso es el núcleo del método de construcción de prototipo. Es aquí donde el cliente puede examinar una representación implementada de los requerimientos del programa, sugerir modificaciones que harán al programa cumplir mejor las necesidades reales (Jiménez & Mendoza, 2014).

PASO 6: Los pasos 4 y 5 se repiten iterativamente hasta que todos los requerimientos estén formalizados o hasta que el prototipo haya evolucionado hacia un sistema de producción (Jiménez & Mendoza, 2014).

3.2.11 Arquitectura Colonial

Define a la Arquitectura Colonial como un conjunto de manifestaciones arquitectónicas que surgieron en América Latina desde el descubrimiento del continente en 1492 hasta la independencia del mismo a principios del siglo XIX (Alejandra, 2007). A comienzos del siglo XVI puede decirse que se terminó la conquista de América en su mayor parte. Sobre ruinas de grandes imperios

precolombinos como México, se preparan los cimientos de la nueva civilización hispanoamericana.

El arte en Latinoamérica va a ser fundamentalmente religioso, marcado por el poder de las principales órdenes religiosas llegadas del viejo continente. En el trazado reticular de las ciudades, a través de los españoles que los proponen, aparecen las plazas y los monumentos. La iglesia edificada junto a la plaza central de las poblaciones se encuentra como punto de referencia del espacio urbano. Pese a la uniformidad que las órdenes religiosas van a intentar aportar, las nuevas formas artísticas van cambiando de acuerdo a la región étnica y geográfica.

3.2.12 Arte Colonial

Inmediatamente después de la conquista de México Tenochtitlán, por los españoles, nació un arte Colonial, esencialmente religioso que buscaba propiciar la evangelización cristiana de los pueblos conquistados. Este arte Colonial también es denominado Novohispano o arte de la Nueva España y reflejó en un inicio los ideales político-religiosos, dentro de la tradición europea. Sin embargo, paulatinamente, aparecieron elementos indígenas siempre más marcados, hasta el surgimiento de un arte inconfundible, con una trayectoria y proyección propias. El arte colonial es producto de la imposición de las formas de vida europea a los pueblos indígenas.

La iglesia católica en su afán de evangelizador, patrocino el desarrollo de todas las artes, por lo que el arte civil careció de importancia, salvo en lo que se refiere a la arquitectura. Bajo la dirección de los frailes se elevaron conventos y se crearon instalaciones que no eran conocidas en Europa.

En el desarrollo del arte barroco novohispano fueron importantes los siguientes factores:

- Imposición de la forma de vida Europea a los pueblos indígenas
- La Iglesia, en su afán de evangelizador; patrocina el desarrollo de las artes.
- Los frailes dictan los parámetros para la expresión artística
- Para los conquistadores fue la oportunidad de crear una sociedad con un nuevo estilo de vida.
- Para los indígenas implico un cambio tajante de vida, ya que se vieron obligados a renunciar a sus creencias ancestrales, sus costumbres y ritos.
- La mezcla de la cultura indígena y europea, que dio como resultado un sincretismo de extraordinaria riqueza, el cual se ve reflejado en el arte del siglo XVI.
- Los indígenas aportaron sus conocimientos de diversas técnicas, de dominio de los recursos naturales, como colorantes y aglutinantes, etc.
- El conjunto conventual es el representante más significativo de la arquitectura del siglo XVI, debido que en él se refleja con mayor nitidez las características y necesidades de la sociedad que surgió a raíz de la fusión de las culturas indígenas y europea. Dichos conjuntos arquitectónicos se componían de diferentes partes:

Características de la arquitectura Colonial

- Diferenciarán tanto por los materiales utilizados para la construcción.
- La aplicación de nuevas formas artísticas.
- Adaptando a las variaciones étnicas y geográficas.
- Se emplea una tipología en virtud de la función.
- El nacimiento de esta nueva civilización hispanoamericana surge paralelo a grandes estilos artísticos desarrollados en Europa a finales del siglo XV.
- El gótico, renacimiento, barroco y neoclasicismo fueron los estilos que influenciaron en la topología arquitectónica de la Arquitectura Colonial.
- Los ejemplos de traza gótica que encontramos en Latinoamérica son por ello escasos y muy directamente emparentados con el primer renacimiento del siglo XVI.
- La catedral de Santo Domingo (1521-1537), como la Iglesia de San Francisco y la de la Merced, así como algunas portadas y edificios civiles en la República Dominicana.
- El barroco en Hispanoamérica es esencialmente decorativo. México y Perú son las dos ciudades en donde hubo más intensidad de este estilo. Mientras que en México hubo un buen manejo de los materiales, como son la piedra y el yeso creando con estos ricas policromías. El elemento de mayor importancia fue la cúpula, presente en todos los templos.

3.2.13 La arquitectura religiosa en el Perú Colonial

El establecimiento de los dominios españoles en los territorios americanos tuvo una de sus más significativas manifestaciones en la fundación de ciudades por los conquistadores, y por supuesto en toda la transculturización realizada por ellos desde ese histórico momento, una de cuyas muestras son las abundantes edificaciones dentro de las nuevas colonias. Caracterizaciones e imprimieron sus sellos a los hechos urbanos arquitectónico de esa época (denominada Época Colonial-Siglo XVI a XIX) siendo una expresión fiel de una situación histórica dada y de su realidad económica y social, su mundo físico, su técnica y su creencia. Sin embargo, esta arquitectura europea, más propiamente español, experimento determinados cambios por razones de índoles social, cultural e histórica, geográfica y climatológica, especialmente si se toma en consideración que la arquitectura de la civilización inca poseía patrones y una simbología diferente (Jiménez, 2010).

Elemento representativo de esta etapa histórica es la arquitectura religiosa, un claro ejemplo que la dimensión religiosa de la vida social y supremacía de la Iglesia Católica tuvieron una profunda incidencia en la España de la época y por consiguiente en el nuevo territorio conquistador de américa. Las características físicas de los templos de esta época mantenían criterios religiosos diferentes a los cuales, como la realización de la ceremonia litúrgica en latín, pero no eran entendido por la mayoría de la población, y a espaldas de la feligresía, pues esta era parte "anexa" de la celebración, pero sin tener una real compresión y participación del rito religioso.

3.2.14 Templos Coloniales en Aymaraes

A. Templo Colonial de Chuquinga

- Es un pequeño pueblo de la provincia apurimeña de Aymaraes, ubicado en la margen izquierda del rio Chalhuanca a 2870 m.s.n.m. luego de cruenta batalla en Chuquinga, del 21 de mayo de 1554, en los dos siglos, Chuquinga fue el punto de partida para el ascenso a la mina de oro y asentamiento minero de Huaillaripa en las punas de la comunidad. Durante el auge de la explotación de las minas en el siglo XVII fue construido el hermoso templo de Huaillaripa, ahora en ruina, y la Iglesia de Chuquinga, posiblemente en remplazo con una edificación más rustica (Hostnig, 2006). La selección de este templo para el modelado forma parte de la interculturidad y promover este atractivo como las tendencias del desarrollo de software educativo intercultural (Mamani & Ibarra & Ordoñez , 2012)

FIGURA N° 8 : Templos Colonial de chuquinga, cuyo centro se encuentra a 2800 msnm.
Fuente:(Hostnig, 2006).

B. Características del Templo

El Templo de Chuquinga, ergido bajo la advocación de San Pedro, presenta una planta de distribución longitudinal, con tres volúmenes adosados al cuerpo principal, en el muro lateral sur, correspondiente a los ambientes del baptisterio[1], la capilla y la sacristía. El muro norte, orientado hacia el rio Chalhuanca, presenta un desarrollo longitudinal pronunciado, interrumpido solamente por cuatro contra fuentes y la portada lateral (Hostnig, 2006).

C. Descripción del análisis de las tejas pintadas de Chuquinga

Las tejas con pintura pertenecen a la tipología de tejas coloniales de forma curva y tamaño relativamente uniforme, mide entre 45 y 50 cm de largo, 21 a 24 cm en el extremo angosto y la curvatura de 9 a 12 cm, medidos desde el centro de la teja hasta la línea base imaginaria (Hostnig, 2006).

FIGURA N° 9 : Línea en zigzag entrecruzadas separadas por líneas horizontales graficadas sobre las tejas.
Fuente: (Hostnig, 2006).

[1] Zona en el interior de una iglesia donde se encuentra la pila bautismal y tiene lugar la ceremonia del bautismo.

FIGURA N° 10: Motivos en forma de ganchos y anclas.
Fuente: (Hostnig, 2006).

FIGURA N° 11: Reptil con serpientes.
Fuente: (Hosting, 2006).

FIGURA N° 12: Tejas pintadas con características similares a las de San Blas, Cusco.
Fuente: (Hostnig, 2006).

3.3 Marco Conceptual

A. Dispositivo móvil. Es un tipo de computadora de bolsillo o computadora de mano con capacidad de procesamiento, con conexión a internet, con memoria diseñado específicamente para una función.

B. Fotogrametría digital. Es un conjunto de técnicas que, mediante una cámara fotográfica, permiten deducir una proyección cónica de la imagen, sus dimensiones y la ubicación de una zona.

C. Gráficos 3D. El término se referirse al proceso de crear dichos gráficos, o el campo de estudio de técnicas y tecnología relacionadas con los gráficos tridimensionales.

D. Motor de render. Puede ser un plugin o software independiente que nos permite generar una vista realista de un 3D.

E. Modelo en 3D. Conjunto de líneas vértices que forman un objeto.

F. Plugins. Son aplicaciones adicionales que se ejecutan por aplicaciones principales.

G. Renderización. Es el proceso de generar una imagen o video mediante el cálculo de iluminación partiendo de un modelo 3D.

H. Sistema. Un sistema que es un conjunto de componentes que interaccionan entre sí para lograr objetivos en común. Nuestra sociedad está rodeada de sistemas. Sistema es una colección de componentes interrelacionados que trabajan conjuntamente para cumplir algún objetivo.

I. Turismo. Conjunto de acciones, actividades que una persona realiza y se lleva a cabo cuando una persona viaja y pernocta en un lugar diferente de su residencia.

J. **ThreeJS.** Es un Framework liviano escrita en JavaScript con un motor gráfico para la Web.

CAPÍTULO IV
HIPÓTESIS Y VARIABLES

4.1 Formulación de la hipótesis

Los dispositivos móviles influyen significativamente en el tiempo de renderización del modelado 3D con el Framework ThreeJS del Templo Colonial de Chuquinga, Aymaraes-Apurímac 2016.

Hipótesis especificas

- La memoria interna de los dispositivos móviles influyen significativamente en el tiempo de renderización del modelado 3D con el Framework ThreeJS del Templo Colonial de Chuquinga, Aymaraes- Apurímac 2016.
- La memoria RAM de los dispositivos móviles influyen significativamente en el tiempo de renderización del modelado 3D con el Framework ThreeJS del Templo Colonial de Chuquinga, Aymaraes- Apurímac 2016.
- La GPU de los dispositivos móviles influyen significativamente en el tiempo de renderización del modelado 3D con el Framework ThreeJS del Templo Colonial de Chuquinga, Aymaraes- Apurímac 2016.
- La memoria interna de los dispositivos móviles influyen significativamente en la frecuencia de imágenes por segundos del modelado 3D con el Framework ThreeJS del Templo Colonial de Chuquinga, Aymaraes- Apurímac 2016.
- La memoria RAM de los dispositivos móviles influyen significativamente en la frecuencia de imágenes por segundos del modelado 3D con el Framework ThreeJS del Templo Colonial de Chuquinga, Aymaraes- Apurímac 2016.

- La GPU de los dispositivos móviles influyen significativamente en la frecuencia de imágenes por segundos del modelado 3D con el Framework ThreeJS del Templo Colonial de Chuquinga, Aymaraes- Apurímac 2016.

4.2 Sistemas de variables

TABLA N° 2: *Definición Sistema de variables*

Variable	Dimensión	Indicador	Escala de medición
Variable Independiente Dispositivo móvil. Definición: Aparato pequeño, con algunas capacidades de procesamiento, con conexión permanente o intermitente a una red, con memoria limitada, que ha sido diseñado específicamente para una función, pero que puede llevar a cabo otras funciones más generales	Características de dispositivos móviles	Memoria interna	➢ 16 GB ➢ 32 GB ➢ 64 GB ➢ 128 GB
		Memoria Ram	➢ 512 MB ➢ 1 GB ➢ 3 GB ➢ 4 GB
		GPU	➢ Básicos ➢ Óptimos
Variable Dependiente Renderizacion del Modelado 3D con el Framework ThreeJS del Templo Colonial de Chuquinga Definición: El render es el proceso de producir imágenes desde una vista de modelos tridimensionales, en una escena 3D. En palabras sencillas, es "tomar una foto" de la escena. Una animación es una serie de render Secuenciados.	Rederización	Tiempo de renderizado	- Tiempo Segundos (0- 60)
		Frecuencia de imagen en un dispositivo móvil.	- FPS(0-60)

Fuente: Elaboración propia.

CAPÍTULO V
METODOLOGÍA DE LA INVESTIGACIÓN

5.1 Tipos y niveles de investigación

5.1.1 Tipo de investigación

La presente investigación que se realizó es aplicada, puesto que su aporte está dirigido al tiempo de renderización y a la frecuencia de imágenes (FPS) del modelado 3D del Templo Colonial de Chuquinga, ubicado en la provincia de Aymaraes-Apurímac 2016, haciendo uso del Framework ThreeJS en dispositivos móviles.

5.1.2 Nivel de investigación

El nivel de investigación que se realizó es explicativo debido a que se quiere conocer el comportamiento de una variable respecto a los demás, es decir se establecerá las causas del fenómeno que se estudia y los efectos de estos en la renderización y la frecuencia de imágenes (FPS) del modelado 3D del Templo Chuquinga en dispositivos móviles haciendo uso Framework ThreeJS, Aymaraes- Apurímac 2016.

5.2 Método y diseño de investigación

5.2.1 Método de investigación

El método usado en esta investigación es el método inductivo porque permitió describir, detallar, especificar todos los aspectos del desarrollo del modelado 3D del Templo Colonial de Chuquinga con el Framework ThreeJS; ya que permite responder a las preguntas planteadas.

5.2.2 Diseño de investigación

El diseño de investigación que se utilizó es el diseño experimental; ya que se realizó una acción y luego se observó la consecuencia al manipular de manera intencional una o más variables independiente (causa) para analizar la consecuencia de tal manipulación de una o más variables dependientes (efecto).

Nomenclatura

Este diseño podría diagramarse de la siguiente manera:

Causa Efecto

X----------Y

(Variables independientes) (Variable dependientes)

$X_i = memoria\ interna$	$Y_1 = tiempo\ de\ renderizado$
$X_j = memoria\ RAM$	$Y_2 = frecuencia\ de\ imágenes\ (FPS)$
$X_k = GPU$	

5.3 Población y muestra

5.3.1 Población

La población de esta investigación es el número personas con dispositivos móviles que residen en la provincia de Aymaraes- Apurímac.

Se tiene una población de 29569 personas según el censo realizado por el INEI en el 2007, de las cuales se consideraron solo la zona urbana siendo el 43,2 %. Esto nos da un total de 12,773 personas con y sin dispositivos móviles que tengan un nivel de conocimiento básico con respecto a dispositivos y de acuerdo al censo son el 15% del total de la población urbana dando un total de N=1915 personas.

5.3.2 Muestra

El tipo de muestra es no probabilística por cuotas debido a que se asemeja con el muestreo aleatorio estratificado que no tiene el carácter de aleatoriedad, es decir se fijan unas cuotas que consiste en el números de personas con dispositivos móviles que reúnen una determinada condición(Hernández & Fernández & Baptista, 2014), como son.

- Que tengan dispositivos móviles originales.
- Dispositivos móviles que sean superiores o igual a gama baja.
- Que los dispositivos móviles sean de la marca Samsung, LG, Sony, Nokia, Motorola, Huawei, Bitel, Mobile, Doogee, HTC y Xioami.

n= 30 dispositivos móviles.

5.4 Material de la investigación

5.4.1 Técnicas e instrumentos de investigación:

La técnica utilizada es:

- La observación, siendo un elemento fundamental que permitió obtener la información de las características de los dispositivos móvil (memoria interna, memoria Ram y GPU), tiempo de renderización y frecuencia de imágenes (FPS) para registrarla y proseguir con su análisis.

El instrumento utilizado es:

- Registros de datos del tiempo de renderización y frecuencia de imágenes (FPS) de los dispositivos móviles.

5.5 Procedimientos de la investigación

I Etapa:

Se desarrolló la aplicación web móvil utilizando la Metodología de Desarrollo Orientado a Prototipos, por ser esta la que mejor se adapta a los objetivos que se pretenden lograr.

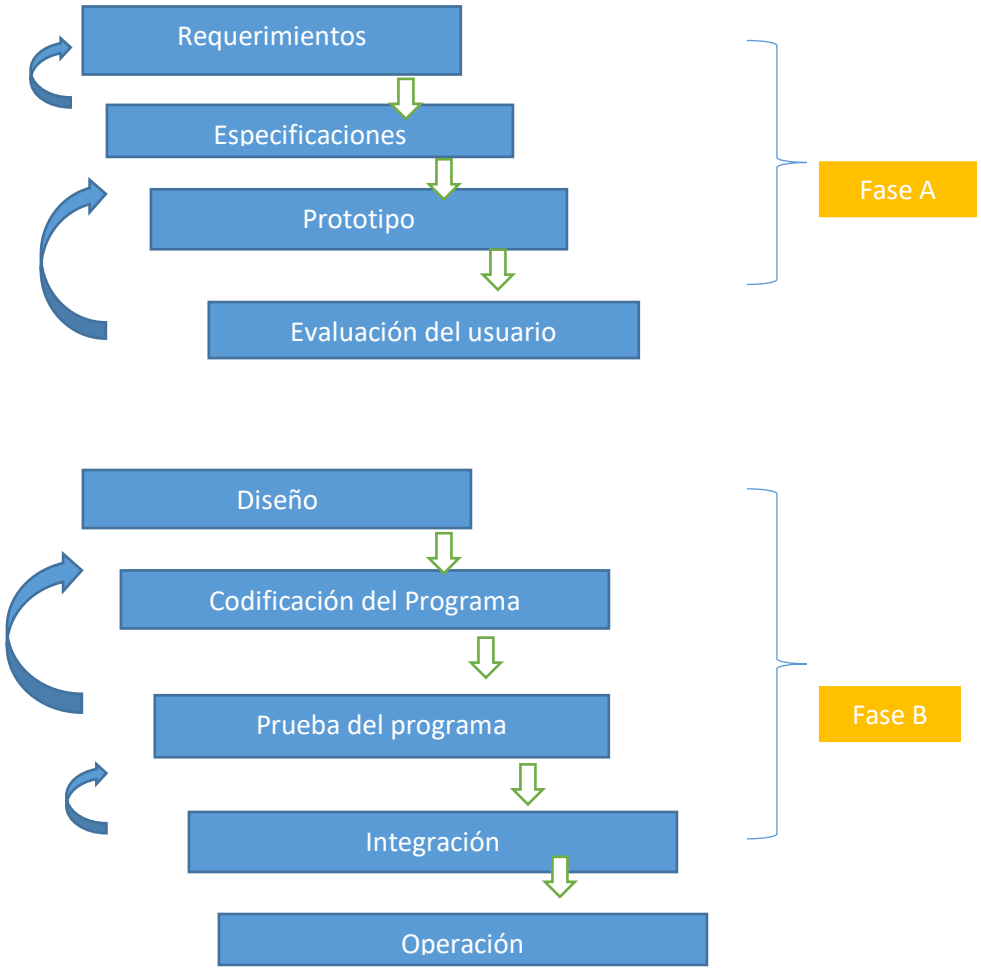


FIGURA N° 13: Fases de metodología orientada a prototipos
Fuente: Elaboración propia

FASE A: Las principales actividades que se realizó en esta primera fase son: Análisis de los requerimientos del sistema, integración de los requerimientos, selección del entorno y creación de la geometría, al finalizar esta fase el proyecto prototipo estará listo para ser evaluado.

FASE B: Se inició con la conversión de la geometría y se trabajó con la implementación de tecnologías de entorno donde se detallaron aspectos visuales y se integraron las distintas etapas hasta dejarlo culminado.

II Etapa Prueba y detección de errores:

- Pruebas en distintos dispositivos móviles.
- Evaluación de la aplicación en dispositivos.

III Etapa Implantación:

El modelado del templo colonial de Chuquinga se encuentra alojado en un hosting cuyo dominio es www.spaziour.com

IV Etapa Tratamiento de datos:

En esta etapa se realizó un tratamiento de los datos tomados de diferentes dispositivos móviles al ejecutar la aplicación.

CAPÍTULO VI
RESULTADOS

6.1 Análisis e interpretación de datos

6.1.1 Descripción de resultados de las hipótesis de investigación

A. Para el tiempo de renderización en dispositivos móviles

a) Selección de factores y variables de respuesta

- Identificar qué factores afectan al tiempo de renderización del Templo Colonial de Chuquinga utilizando el Framework ThreeJS como se muestra en la tabla N° 3.
- Definir los niveles "óptimos" en el que debe fijarse para cada factor.
- Identificar los factores que no afectan al renderizado del Templo Colonial de Chuquinga utilizando el Framework ThreeJS a fin de ahorrar costos de prueba.

Se escogió los factores que se desean estudiar y los valores que pueden tomar (dominio experimental). Una vez analizado el proceso de obtención del tiempo de renderizado en los dispositivos móviles, se encontraron los factores que se muestra en la siguiente tabla N° 3:

TABLA N° 3 : *Factores y rangos*

FACTOR	TIPO	RANGO
Memoria interna	Controlable	[16GB < , ≥ 16GB]
Memoria RAM	Controlable	[2GB < , ≥ 2GB]
GPU	Controlable	[500 MHZ≤ , >500 MHZ]
Internet	No controlable	

Fuente: Elaboración propia.

Cada factor tiene niveles y un nivel es un rango de variación cuyo centro es el nivel teórico de tal factor, para lo cual hay que tener en cuenta que cada factor deba tener al menos dos niveles de prueba que realmente sean diferentes. A continuación se muestra la tabla de factores con sus respectivos niveles:

TABLA N° 4 : *Factores y niveles para de tiempo de renderizado*

FACTOR	TIPO	RANGO	DOMINIO EXPERIMENTAL	
			Nivel (-)	Nivel(+)
Memoria interna	Controlable	[16GB < , ≥ 16GB]	16GB <	≥ 16GB
Memoria RAM	Controlable	[2GB < , ≥ 2GB]	2GB <	≥ 2GB
GPU	Controlable	[500MHZ≤,>500MHZ]	500MHZ ≤	> 500MHZ

Fuente: Elaboración propia.

b) Selección de matriz de experimentos

Se tienen 3 factores y cada uno con dos niveles extremos uno negativo y otro positivo como se muestra en la tabla N° 5.

Una vez establecido los factores con sus respectivos niveles, se diseña la matriz de datos o experimentos, estos arreglos son diseños factoriales, la cual está representada por el número de corridas dando un total de 30 corridas, también se muestran los 3 factores memoria interna, memoria RAM y GPU; para conocer el efecto de un factor es suficiente con hacerlo variar entre dos valores – y + que son los extremos del dominio experimental y se muestran en la tabla N° 6.

TABLA N° 5 : *Dominio experimental*

FACTORES	DOMINIO EXPERIMENTAL	
	NIVEL(-)	NIVEL(+)
Memoria interna	16GB <	≥ 16GB
Memoria RAM	2GB <	≥ 2GB
GPU	500MHZ ≤	> 500MHZ

Fuente: Elaboración propia.

TABLA N° 6 : Matriz de experimentos con dos niveles y tres factores memoria interna, memoria RAM y GPU.

MATRIZ DE EXPERIMENTOS			
N° de corridas	**Memoria interna**	**Memoria RAM**	**GPU**
1	+	+	+
2	-	-	-
3	+	+	+
4	-	-	-
5	+	+	+
6	+	-	+
7	+	+	+
8	+	+	+
9	+	+	+
10	+	+	+
11	-	-	-
12	-	-	-
13	-	-	-
14	-	-	-
15	-	+	+
16	+	+	+
17	-	-	-
18	-	+	+
19	+	+	+
20	+	-	+
21	-	+	-
22	-	-	-
23	-	+	-
24	-	-	-
25	+	+	+
26	-	-	-
27	-	-	-
28	+	-	+
29	-	-	+
30	-	+	+

Fuente: Elaboración propia.

Donde:

\+ = Nivel óptimo
\- = Nivel básico

TABLA N° 7 : Plan de experimentación con característica de dispositivos móviles

MATRIZ DE EXPERIMENTOS			
N° de corridas	**Memoria interna**	**Memoria RAM**	**GPU**
1	≥ 16GB	≥ 2GB	> 500MHZ
2	16GB <	2GB <	500MHZ ≤
3	≥ 16GB	≥ 2GB	> 500MHZ
4	16GB <	2GB <	500MHZ ≤
5	≥ 16GB	≥ 2GB	> 500MHZ
6	≥ 16GB	2GB <	> 500MHZ
7	≥ 16GB	≥ 2GB	> 500MHZ
8	≥ 16GB	≥ 2GB	> 500MHZ
9	≥ 16GB	≥ 2GB	> 500MHZ
10	≥ 16GB	≥ 2GB	> 500MHZ
11	16GB <	2GB <	500MHZ ≤
12	16GB <	2GB <	500MHZ ≤
13	16GB <	2GB <	500MHZ ≤
14	16GB <	2GB <	500MHZ ≤
15	16GB <	≥ 2GB	> 500MHZ
16	≥ 16GB	≥ 2GB	> 500MHZ
17	16GB <	2GB <	500MHZ ≤
18	16GB <	≥ 2GB	> 500MHZ
19	≥ 16GB	≥ 2GB	> 500MHZ
20	≥ 16GB	2GB <	> 500MHZ
21	16GB <	≥ 2GB	500MHZ ≤
22	16GB <	2GB <	500MHZ ≤
23	16GB <	≥ 2GB	500MHZ ≤
24	16GB <	2GB <	500MHZ ≤
25	≥ 16GB	≥ 2GB	> 500MHZ
26	16GB <	2GB <	500MHZ ≤
27	16GB <	2GB <	500MHZ ≤
28	≥ 16GB	2GB <	> 500MHZ
29	16GB <	2GB <	> 500MHZ
30	16GB <	≥ 2GB	> 500MHZ

Fuente: Elaboración propia.

Seguidamente, se obtuvo la respuesta del tiempo de renderizado en dispositivos móviles para cada corrida es decir las combinaciones de los tres factores (memoria interna, memoria RAM y GPU) con sus dos niveles (básico y óptimo) como se muestra en la tabla N° 8.

TABLA N° 8 : Respuesta del tiempo de renderizado de cada combinación

N° de corridas	Memoria interna	Memoria RAM	GPU	Respuesta del tiempo de renderizado
1	≥ 16GB	≥ 2GB	> 500MHZ	2.1 segundos
2	16GB <	2GB <	500MHZ ≤	2.9 segundos
3	≥ 16GB	≥ 2GB	> 500MHZ	1.5 segundos
4	16GB <	2GB <	500MHZ ≤	2.26 segundos
5	≥ 16GB	≥ 2GB	> 500MHZ	2.2 segundos
6	≥ 16GB	2GB <	> 500MHZ	2.3 segundos
7	≥ 16GB	≥ 2GB	> 500MHZ	1.3 segundos
8	≥ 16GB	≥ 2GB	> 500MHZ	1.95 segundos
9	≥ 16GB	≥ 2GB	> 500MHZ	1.5 segundos
10	≥ 16GB	≥ 2GB	> 500MHZ	1.882 segundos
11	16GB <	2GB <	500MHZ ≤	1.78 segundos
12	16GB <	2GB <	500MHZ ≤	2.8 segundos
13	16GB <	2GB <	500MHZ ≤	2.122 segundos
14	16GB <	2GB <	500MHZ ≤	2.597 segundos
15	16GB <	≥ 2GB	> 500MHZ	1.1924 segundos
16	≥ 16GB	≥ 2GB	> 500MHZ	0.852 segundos
17	16GB <	2GB <	500MHZ ≤	0.66 segundos
18	16GB <	≥ 2GB	> 500MHZ	2.267 segundos
19	≥ 16GB	≥ 2GB	> 500MHZ	1.354 segundos
20	≥ 16GB	2GB <	> 500MHZ	1.573 segundos
21	16GB <	≥ 2GB	500MHZ ≤	4.174 segundos
22	16GB <	2GB <	500MHZ ≤	4.161 segundos
23	16GB <	≥ 2GB	500MHZ ≤	2.152 segundos
24	16GB <	2GB <	500MHZ ≤	3.147 segundos
25	≥ 16GB	≥ 2GB	> 500MHZ	1.891 segundos
26	16GB <	2GB <	500MHZ ≤	2.923 segundos
27	16GB <	2GB <	500MHZ ≤	1.3 segundos
28	≥ 16GB	2GB <	> 500MHZ	0.9 segundos
29	16GB <	2GB <	> 500MHZ	0.8 segundos
30	16GB <	≥ 2GB	> 500MHZ	0.92 segundos

Fuente: Elaboración propia.

B. Para frecuencia de imágenes(FPS) en dispositivos móviles

a) Selección de factores y variables de respuesta

- Identificar qué factores afectan a la frecuencia de imágenes del modelado 3D del Templo Colonial de Chuquinga utilizando el Framework ThreeJS en dispositivos móviles como se muestra en la tabla N° 9.
- Definir los niveles "óptimos" en el que debe fijarse para cada factor.
- Identificar los factores que no afectan a la frecuencia de imágenes del modelado 3D del Templo Colonial de Chuquinga utilizando el Framework ThreeJS en dispositivos móviles a fin de ahorrar costos de prueba.

Se escoge los factores que se desean estudiar y los valores que pueden tomar (dominio experimental). Una vez analizado la frecuencia de imágenes del modelado 3D del Templo Colonial de Chuquinga utilizando el Framework ThreeJS en dispositivos móviles, se encontraron los factores que se muestra en la siguiente tabla N° 9:

TABLA N° 9 : *Factores y rangos*

FACTOR	TIPO	RANGO
Memoria interna	Controlable	[16GB < , ≥ 16GB]
Memoria RAM	Controlable	[2GB < , ≥ 2GB]
GPU	Controlable	[500 MHZ≤ , >500 MHZ]
Internet	No controlable	

Fuente: Elaboración propia.

Cada factor tiene niveles y un nivel es un rango de variación cuyo centro es el nivel teórico de tal factor, para lo cual hay que tener en cuenta que cada factor (memoria interna, memoria RAM y GPU) debe tener al menos dos niveles (básico y óptimo) de prueba que realmente sean diferentes.

A continuación se muestra la tabla de factores con sus respectivos niveles para la variable respuesta (frecuencia de imágenes) con el aplicativo web en dispositivos móviles:

TABLA N° 10 : *Factores y niveles para la frecuencia de imágenes (FPS)*

FACTOR	TIPO	RANGO	DOMINIO EXPERIMENTAL	
			Nivel (-)	Nivel(+)
Memoria interna	Controlable	[16GB < , ≥ 16GB]	16GB <	≥ 16GB
Memoria RAM	Controlable	[2GB < , ≥ 2GB]	2GB <	≥ 2GB
GPU	Controlable	Básico, óptimo	500MHZ ≤	> 500MHZ

Fuente: Elaboración propia.

b) Selección de matriz de experimentos

Se tienen 3 factores y cada uno con dos niveles extremos uno negativo y otro positivo como se muestra en la tabla N° 11.

Una vez establecido los factores con sus respectivos niveles, se diseña la matriz de datos o experimentos, estos arreglos son diseños factoriales, la cual está representada por el número de corridas dando un total de 30 corridas, también se muestran los 3 factores memoria interna, memoria RAM y GPU; para conocer el efecto de un factor es suficiente con hacerlo variar entre dos valores – y + que son los extremos del dominio experimental y se muestran en la tabla N° 12.

TABLA N° 11 : *Dominio experimental*

FACTORES	DOMINIO EXPERIMENTAL	
	NIVEL(-)	NIVEL(+)
Memoria interna	16GB <	≥ 16GB
Memoria RAM	2GB <	≥ 2GB
GPU	500MHZ ≤	> 500MHZ

Fuente: Elaboración propia.

TABLA N° 12 : Matriz de experimentos con dos niveles y tres factores memoria interna, memoria RAM y GPU.

MATRIZ DE EXPERIMENTOS			
N° de corridas	**Memoria interna**	**Memoria RAM**	**GPU**
1	+	+	+
2	-	-	-
3	+	+	+
4	-	-	-
5	+	+	+
6	+	-	+
7	+	+	+
8	+	+	+
9	+	+	+
10	+	+	+
11	-	-	-
12	-	-	-
13	-	-	-
14	-	-	-
15	-	+	+
16	+	+	+
17	-	-	-
18	-	+	+
19	+	+	+
20	+	-	+
21	-	+	-
22	-	-	-
23	-	+	-
24	-	-	-
25	+	+	+
26	-	-	-
27	-	-	-
28	+	-	+
29	-	-	+
30	-	+	+

Fuente: Elaboración propia.

Donde:

\+ = Nivel óptimo
\- = Nivel básico

TABLA N° 13 : Plan de experimentación con característica de dispositivos móviles

MATRIZ DE EXPERIMENTOS			
N° de corridas	**Memoria interna**	**Memoria RAM**	**GPU**
1	≥ 16GB	≥ 2GB	> 500MHZ
2	16GB <	2GB <	500MHZ ≤
3	≥ 16GB	≥ 2GB	> 500MHZ
4	16GB <	2GB <	500MHZ ≤
5	≥ 16GB	≥ 2GB	> 500MHZ
6	≥ 16GB	2GB <	> 500MHZ
7	≥ 16GB	≥ 2GB	> 500MHZ
8	≥ 16GB	≥ 2GB	> 500MHZ
9	≥ 16GB	≥ 2GB	> 500MHZ
10	≥ 16GB	≥ 2GB	> 500MHZ
11	16GB <	2GB <	500MHZ ≤
12	16GB <	2GB <	500MHZ ≤
13	16GB <	2GB <	500MHZ ≤
14	16GB <	2GB <	500MHZ ≤
15	16GB <	≥ 2GB	> 500MHZ
16	≥ 16GB	≥ 2GB	> 500MHZ
17	16GB <	2GB <	500MHZ ≤
18	16GB <	≥ 2GB	> 500MHZ
19	≥ 16GB	≥ 2GB	> 500MHZ
20	≥ 16GB	2GB <	> 500MHZ
21	16GB <	≥ 2GB	500MHZ ≤
22	16GB <	2GB <	500MHZ ≤
23	16GB <	≥ 2GB	500MHZ ≤
24	16GB <	2GB <	500MHZ ≤
25	≥ 16GB	≥ 2GB	> 500MHZ
26	16GB <	2GB <	500MHZ ≤
27	16GB <	2GB <	500MHZ ≤
28	≥ 16GB	2GB <	> 500MHZ
29	16GB <	2GB <	> 500MHZ
30	16GB <	≥ 2GB	> 500MHZ

Fuente: Elaboración propia.

Seguidamente, se obtuvo la respuesta de la frecuencia de imágenes (FPS) en dispositivos móviles con el aplicativo para cada corrida es decir las combinaciones de los tres factores con su dos nivel o domino experimental como se muestra en la tabla N° 14.

TABLA N° 14 : *Respuesta de la frecuencia de imágenes generada por cada dispositivo móvil con el aplicativo web móvil*

N° de corridas	Memoria interna	Memoria RAM	GPU	Respuesta frecuencia de imágenes (FPS)
1	≥ 16GB	≥ 2GB	> 500MHZ	58
2	16GB <	2GB <	500MHZ ≤	30
3	≥ 16GB	≥ 2GB	> 500MHZ	55
4	16GB <	2GB <	500MHZ ≤	35
5	≥ 16GB	≥ 2GB	> 500MHZ	50
6	≥ 16GB	2GB <	> 500MHZ	56
7	≥ 16GB	≥ 2GB	> 500MHZ	47
8	≥ 16GB	≥ 2GB	> 500MHZ	59
9	≥ 16GB	≥ 2GB	> 500MHZ	60
10	≥ 16GB	≥ 2GB	> 500MHZ	57
11	16GB <	2GB <	500MHZ ≤	55
12	16GB <	2GB <	500MHZ ≤	39
13	16GB <	2GB <	500MHZ ≤	30
14	16GB <	2GB <	500MHZ ≤	30
15	16GB <	≥ 2GB	> 500MHZ	55
16	≥ 16GB	≥ 2GB	> 500MHZ	60
17	16GB <	2GB <	500MHZ ≤	25
18	16GB <	≥ 2GB	> 500MHZ	55
19	≥ 16GB	≥ 2GB	> 500MHZ	50
20	≥ 16GB	2GB <	> 500MHZ	30
21	16GB <	≥ 2GB	500MHZ ≤	13
22	16GB <	2GB <	500MHZ ≤	25
23	16GB <	≥ 2GB	500MHZ ≤	50
24	16GB <	2GB <	500MHZ ≤	24
25	≥ 16GB	≥ 2GB	> 500MHZ	60
26	16GB <	2GB <	500MHZ ≤	40
27	16GB <	2GB <	500MHZ ≤	30
28	≥ 16GB	2GB <	> 500MHZ	24
29	16GB <	2GB <	> 500MHZ	55
30	16GB <	≥ 2GB	> 500MHZ	58

Fuente: Elaboración propia.

6.1.2 Contrastación de la hipótesis

A. Prueba de hipótesis: Tiempo de renderizado en dispositivos móviles

1. Planteamiento de hipótesis estadísticas

Se tiene el siguiente modelo estructural de ANOVA:

$Y_{ijk} = \mu + \alpha_i + \beta_j + \gamma_k + \varepsilon_{ijk}$

Donde:

Y_{ijk} = Respuesta del tiempo de renderizado sujeto a combinaciones.

μ = La media común a todos los datos del experimento.

ε_{ijk} = Error experimental o aleatorio de muestreo.

α_i = Efecto o impacto del i nivel de la variable del tratamiento memoria interna.

β_j = Efecto o impacto del j nivel de la variable del tratamiento memoria RAM.

γ_K = Efecto o impacto del k nivel de la variable del tratamiento GPU.

Para el caso de la formulación de la hipótesis en esta investigación se planteó las siguientes hipótesis.

H_0: $\alpha_i = \beta_j = \delta_k = 0$ (memoria interna, memoria RAM, GPU no influyen significativamente en el tiempo de renderizado del modelado 3D con el Framework ThreeJS del Templo Colonial de Chuquinga).

H_1: $\alpha_i \neq \beta_j \neq \delta_k \neq 0$ (memoria interna, memoria RAM, GPU influyen significativamente en el tiempo de renderizado del modelado 3D con el Framework ThreeJS del Templo Colonial de Chuquinga).

En este caso las hipótesis de interés es saber si los factores que interactúan en el tiempo de renderizado en dispositivos móviles, se comportan de igual formal o existe alguna diferencia en al menos uno de ellos en cuanto al efecto de la variable de respuesta (tiempo de renderizado).

2. Nivel de significancia

El nivel de significancia es de 5%, donde α = 0.05

3. Estadística

Es un diseño factorial por lo cual se realizó el análisis estadístico de ANOVA con tres factores (memoria interna, memoria RAM y GPU) en el programa de ingeniería minitab y se tiene las siguientes fórmulas:

$$\alpha_i = \frac{1}{I}\sum_{j=0}^{I} Y_{ij(k)} - \bar{Y}\ldots = \bar{Y}_{i..} - \bar{Y}..$$

$$\beta_j = \frac{1}{I}\sum_{i=0}^{I} Y_{ij(k)} - \bar{Y}\ldots = \bar{Y}_{.j.} - \bar{Y}..$$

$$\gamma_K = \frac{1}{I}\sum_{i=1}^{I}\sum_{j=0}^{I} Y_{ij(k)} - \bar{Y}\ldots = \bar{Y}_{\ldots k} - \bar{Y}..$$

Tabla de ANOVA

Factores	S.C	Varianzas	Test F	p - v.
Memoria interna	$I\sum_I \alpha_i^2$=SCE(α)	$S_\alpha^2 = \frac{SCE_{(\alpha)}}{I-1}$	$F_\alpha = \frac{S_\alpha^2}{S_R^2}$	¿?
Memoria RAM	$I\sum_I \beta_j^2$=SCE(β)	$S_\beta^2 = \frac{SCE_{(\beta)}}{I-1}$	$F_\beta = \frac{S_\beta^2}{S_R^2}$	¿?
GPU	$I\sum_I \gamma_k^2$=SCE(γ)	$S_\gamma^2 = \frac{SCE_{(\gamma)}}{I-1}$	$F_\gamma = \frac{S_\gamma^2}{S_R^2}$	¿?

Seguidamente, se procedió a desarrollar el experimento real de la investigación con el diseño factorial y los datos mostrados en la tabla N° 15.

TABLA N° 15 : *Resumen del diseño factorial completo para el tiempo de renderizado*

Diseño factorial completo		
Corridas	30	
Bloques	1	
Factor 1	memoria interna	Nivel • 16GB < donde (-) • ≥ 16GB donde (+)
Factor 2	memoria RAM	Nivel • 2GB < donde (-) • ≥ 2GB donde (+)
Factor 3	GPU	Nivel • 500 MHZ ≤ (-) • > 500MHZ (+)
Total de factores y niveles	3 factores, 2 niveles por cada factor.	

Fuente: Elaboración propia.

Resultados del análisis de varianza del tiempo de renderizado

TABLA N° 16 : *Análisis de varianza del tiempo de renderizado*

Fuente	Suma de cuadrados	Df	Media de cuadrados	F	P-Value
Memoria interna (α_i)	2.7037	1	0.54818	0.89	0.354
Memoria RAM (β_j)	0.0240	1	0.62825	1.02	0.322
GPU (Υ_K)	5.4451	1	5.74865	9.36	0.005

Fuente: Elaboración propia.

Nota: Si el p-value (valor de probabilidad) se aproxima a 1 el valor no es significativo, si P-value se aproxima a 0 el valor es significativo es decir el factor tiene mayor efecto sobre el tiempo de renderizado.

4. Decisión

Resultados de estimación de los efectos promedios para el tiempo de renderización

Los resultados de la estimación de efectos promedios mostrados en la tabla N° 17 se obtuvieron procesando tiempo de renderización registrados en cada dispositivo móvil para lo cual se utilizó el software minitab.

TABLA N° 17 : *Estimación de los efectos promedios para el tiempo de renderizado*

Estimación de los efectos promedios del tiempo de renderizado	
Memoria interna (α_i)	0.4921
Memoria RAM (β_j)	0.4011
GPU(γ_K)	-1.6630

Fuente: Elaboración propia.

- **Resultados de los efectos principales para el tiempo de renderización**

Los resultados se obtuvieron en base a la tabla N° 14 de efectos principales como se muestra en la tabla N° 18.

TABLA N° 18 : *Medias de la gráfica de efectos principales del tiempo de renderización*

Nivel	Memoria Interna	Memoria Ram	GPU
	Media	Media	Media
-	2.244 segundos	2.1482 segundos	2.53662 segundos
+	1.6386 segundos	1.8156 segundos	1.55773 segundos

Fuente: Elaboración propia.

- **Gráfico de efectos principales para tiempo de renderización**

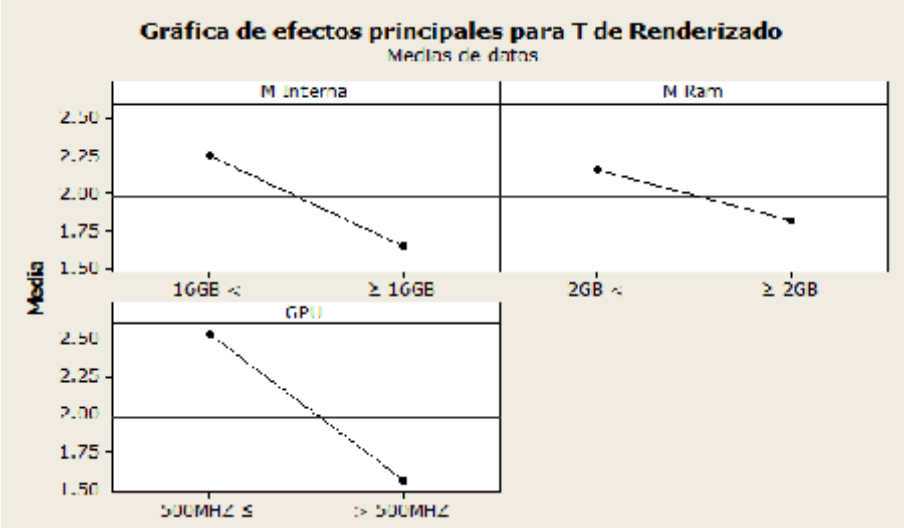


FIGURA N° 14 : Gráfico de efectos principales para el tiempo de renderizado en dispositivos móviles
Fuente: Elaboración propia.

Interpretación: En la figura N° 14 se observa que el factor GPU es el que más efecto tiene sobre el tiempo de renderizado en dispositivos móviles, puesto que la línea no es horizontal, en consecuencia hay efecto principal (diferentes niveles de un factor afecta la respuesta de manera diferente).

Los dispositivos con frecuencia del reloj central menor o igual a 500 Mhz tienen una tasa media mayor a los dispositivos que tienen una frecuencia del reloj central mayor a 500 Mhz, esto indica que cuando en un dispositivo móvil hay mayor frecuencia del reloj central del factor GPU disminuye el tiempo de renderizado caso contrario cuando hay menor frecuencia del reloj central aumenta el tiempo de renderizado, ocasionando que los usuarios que accedan al aplicativo demoren en ver el resultado final del renderizado con toda la textura y animación.

La memoria interna y la memoria RAM parecen afectar en el tiempo de renderizado debido a que sus pendientes son pequeñas por los cual se deberán evaluar en el diagrama de Pareto y la gráfica de efectos estandarizados para ver si estos dos factores afectan significativamente o no en el tiempo de renderizado.

Cálculo de la suma de cuadrados para el análisis de varianza para el tiempo de renderizado.

$$SS_{Total} = SS_{\alpha_i} + SS_{\beta_j} + SS_{\delta_K} + SS_{\alpha_i \beta_j} + SS_{\alpha_i \delta_K} + SS_{\beta_j \delta_K} + SS_{\alpha_i \beta_j \delta_K} + SS_{error}$$

$$SS_{\alpha_i} = (CONTRASTE(\alpha_i))^2 / (2)^K * n$$
$$SS_{\beta_j} = (CONTRASTE(\beta_j))^2 / (2)^K * n$$
$$SS_{\delta_K} = (CONTRASTE(\delta_K))^2 / (2)^K * n$$
$$SS_{\alpha_i \beta_j} = (CONTRASTE(\alpha_i \beta_j))^2 / (2)^K * n$$
$$SS_{\alpha_i \delta_K} = (CONTRASTE(\alpha_i \delta_K))^2 / (2)^K * n$$
$$SS_{\beta_j \delta_K} = (CONTRASTE(\beta_j \delta_K))^2 / (2)^K * n$$
$$SS_{\alpha_i \beta_j \delta_K} = (CONTRASTE(\alpha_i \beta_j \delta_K))^2 / (2)^K * n$$

$$SS_{Error} = SS_{Total} - SS_{\alpha_i} - SS_{\beta_j} - SS_{\delta_K} - SS_{\alpha_i \beta_j} - SS_{\alpha_i \delta_K} - SS_{\beta_j \delta_K} - SS_{\alpha_i \beta_j \delta_K}$$

$$SS_{Error} = 14.7326$$

- **Diagrama de Pareto de efectos estandarizados**

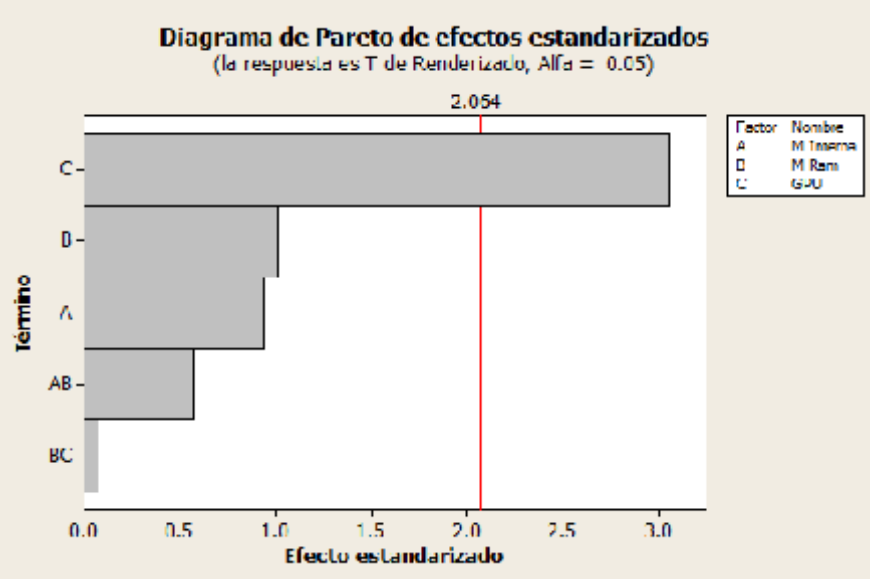


FIGURA N° 15 : Diagrama de Pareto de efectos estandarizados.
Fuente: Elaboración propia.

Interpretación

En la figura N° 15 de Pareto, las barras que cruzan la línea de referencia son significativas es decir la barra C la cual es el factor GPU es el único que cruza la línea de referencia. Este factor es estadísticamente significativo en el nivel de 0.05.

Este gráfico nos ayuda a determinar cuáles efectos son grandes, pero no determina cuál efecto aumenta o disminuye la respuesta del tiempo de renderizado, para ello se utilizó el gráfico de probabilidad de los efectos estandarizados que permite examinar la magnitud y dirección de los efectos de esta investigación.

- **Gráfico normal de los efectos**

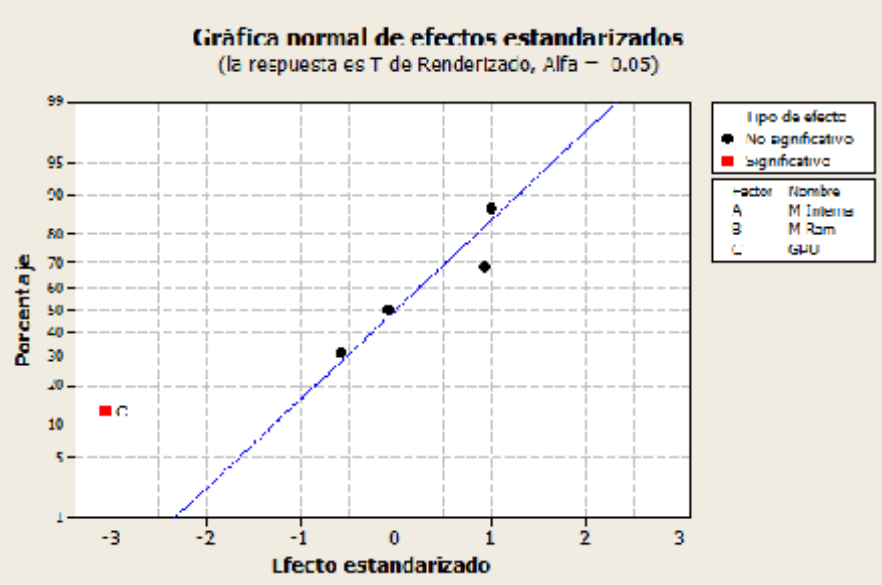


FIGURA N° 16 : Gráfico normal de efectos estandarizados.
Fuente: Elaboración propia.

Interpretación

La figura N° 16 nos permitió determinar la magnitud, dirección e importancia de los efectos. En esta investigación la GPU que representa el punto rojo C es el único factor significativo en el nivel 0.05. Este punto se encuentra alejado y su forma es diferente en comparación con los otros factores no significativos.

Además, el factor GPU tiene un efecto estandarizado negativo debido a que se encuentra en el lado izquierdo de la línea vertical esto indica que cuando la frecuencia

del reloj central es mayor a 500 Mhz, el tiempo de renderizado disminuye en dispositivos móviles.

TABLA N° 19 : Valores extremos de la media del efecto principal para GPU

Nivel	Media
-	2.53662
+	1.55773

Fuente: Elaboración propia.

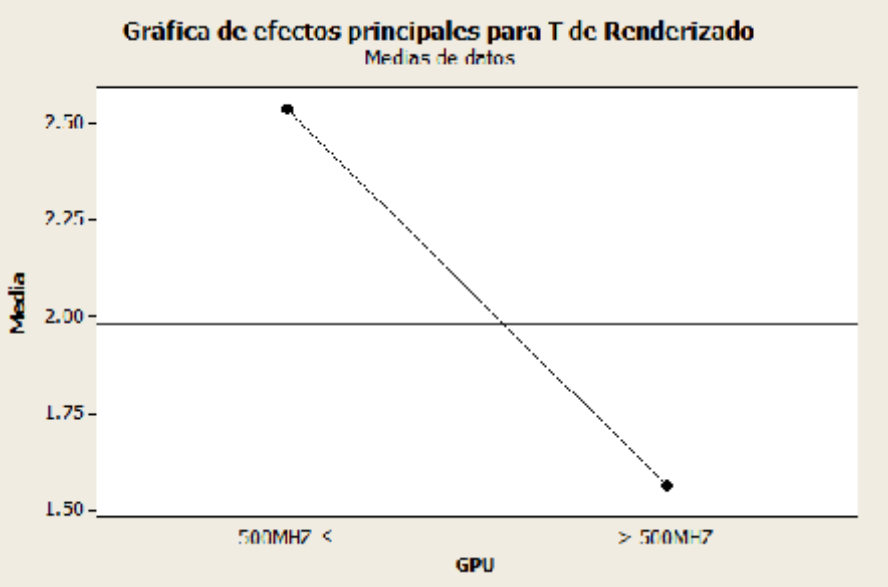


FIGURA N° 17 : Gráfico de efectos principales para la media del tiempo de renderizado en dispositivos móviles.
Fuente: Elaboración Propia.

Interpretación

La figura N° 17 de efectos principales para factor GPU se analizó por ser el único factor que influye en el tiempo de renderizado, el cual es significativo.

Se puede ver que el nivel básico de la frecuencia del reloj central menor o igual a 500 Mhz de la GPU tuvo una tasa media de tiempo de renderizado mayor al nivel óptimo de la frecuencia del reloj central mayor a 500 Mhz.

También, se observa que es un efecto negativo cuando cambia de nivel básica a nivel óptimo de GPU habiendo menor tiempo de renderización en dispositivos móviles.

Análisis de datos

Factor: Memoria interna (α_i)

Con un 95% de confiabilidad en el análisis de varianza se obtuvo que el p-value=0.354. Al evaluar obtenemos que 0.354> 0.05 entonces se rechaza la hipótesis alterna y se acepta la hipótesis nula, por lo que se afirma que modelado 3D del Templo Colonial de Chuquinga con ThreeJS según la memoria interna no influye significativamente en el tiempo de rendizado en dispositivos móviles.

Factor: Memoria RAM (β_j)

Con un 95% de confiabilidad en el análisis de varianza se obtuvo que el p-value=0.322. Al evaluar obtenemos que 0.322> 0.05 entonces se rechaza la hipótesis alterna y se acepta la hipótesis nula, por lo que se afirma que modelado 3D del Templo Colonial de Chuquinga con ThreeJS según la memoria RAM no influye significativamente en el tiempo de renderizado en dispositivos móviles.

Factor: GPU (β_K)

Con un 95% de confiabilidad en el análisis de varianza se obtuvo que el p-value=0.005. Al evaluar obtenemos que 0.005<0.05 entonces se acepta la hipótesis alterna y se rechaza la hipótesis nula, por lo que se afirma que

modelado 3D del Templo Colonial de Chuquinga con ThreeJS según la GPU influye significativamente en el tiempo de renderizado en dispositivos móviles.

B. Prueba de hipótesis: Frecuencia de imágenes (FPS) en dispositivos móviles

1. Planteamiento de hipótesis

Se tiene el siguiente modelo estructural de ANOVA

$Y_{ijk} = \mu + \alpha_i + \beta_j + \gamma_k + \varepsilon_{ijk}$

Donde:

Y_{ijk} = Respuesta de la frecuencia sujeto a combinaciones.
μ = La media común a todos los datos del experimento.
ε_{ijk} = Error experimental o aleatorio de muestreo.
α_i = Efecto o impacto del i nivel de la variable del tratamiento memoria interna.
β_j = Efecto o impacto del j nivel de la variable del tratamiento memoria RAM.
γ_k = Efecto o impacto del k nivel de la variable del tratamiento GPU.

Para el caso de la formulación de la hipótesis en esta investigación se planteó las siguientes hipótesis.

H_0: $\alpha_i = \beta_j = \delta_k = 0$ (memoria interna, memoria RAM, GPU no influyen significativamente en la frecuencia de imágenes del modelado 3D con el Framework ThreeJS del Templo Colonial de Chuquinga)

H_1: $\alpha_i \neq \beta_j \neq \delta_k \neq 0$ (memoria interna, memoria RAM, GPU influyen significativamente en la frecuencia de imágenes del modelado 3D con el Framework ThreeJS del Templo Colonial de Chuquinga).

En este caso las hipótesis de interés es saber si los factores que interactúan en la frecuencia de imágenes (FPS) en dispositivos móviles del aplicativo, se

comportan de igual formal o existe alguna diferencia en al menos uno de ellos en cuanto al efecto de la variable de respuesta.

2. Nivel de significancia

El nivel de significancia es de 5%, donde α = 0.05

3. Estadística

Es un diseño factorial por lo cual se realizó el análisis estadístico de ANOVA con tres factores (memoria interna, memoria RAM y GPU) en el programa de ingeniería minitab y se tiene las siguientes fórmulas:

$$\alpha_i = \frac{1}{I}\sum_{j=0}^{I} Y_{ij(k)} - \bar{Y}\ldots = \bar{Y}_{i..} - \bar{Y}..$$
$$\beta_j = \frac{1}{I}\sum_{i=0}^{I} Y_{ij(k)} - \bar{Y}\ldots = \bar{Y}_{.j.} - \bar{Y}..$$
$$\gamma_K = \frac{1}{I}\sum_{i=1}^{I}\sum_{j=0}^{I} Y_{ij(k)} - \bar{Y}\ldots = \bar{Y}_{\ldots k} - \bar{Y}..$$

Tabla de ANOVA

Factores	S.C	Varianzas	Test F	p - v.
Memoria interna	$I\sum_I \alpha_i^2$=SCE(α)	$S_\alpha^2 = \frac{SCE_{(\alpha)}}{I-1}$	$F_\alpha = \frac{S_\alpha^2}{S_R^2}$	¿?
Memoria RAM	$I\sum_I \beta_j^2$=SCE(β)	$S_\beta^2 = \frac{SCE_{(\beta)}}{I-1}$	$F_\beta = \frac{S_\beta^2}{S_R^2}$	¿?
GPU	$I\sum_I y_k^2$=SCE(γ)	$S_\gamma^2 = \frac{SCE_{(\gamma)}}{I-1}$	$F_\gamma = \frac{S_\gamma^2}{S_R^2}$	¿?

Seguidamente, se procedió a desarrollar el experimento real de la investigación con el diseño factorial y los datos mostrados en la tabla N° 20.

TABLA N° 20 : Resumen del diseño factorial completo para la frecuencia de imágenes en dispositivos móviles

Diseño factorial completo		
Corridas	30	
Bloques	1	
Factor 1	memoria interna	Nivel • 16GB < donde (-) • ≥ 16GB donde (+)
Factor 2	memoria RAM	Nivel • 2GB < donde (-) • ≥ 2GB donde (+)
Factor 3	GPU	Nivel • 500 MHZ ≤ (-) • > 500MHZ (+)
Total de factores y niveles	3 factores, 2 niveles por cada factor.	

Fuente: Elaboración propia.

Análisis de varianza de frecuencia de imágenes (FPS) en dispositivos móviles con el aplicativo 3D.

TABLA N° 21 : Análisis de varianza de la frecuencia de imágenes (FPS)

Fuente	**Suma de cuadrados**	**Df**	**Media de cuadrados**	**F-Ratio**	**P-Value**
Memoria interna (α_i)	1255.39	1	198.64	2.07	0.163
Memoria RAM (β_j)	1203.36	1	296.73	3.09	0.091
GPU (β_K)	781.73	1	1123.69	11.72	0.002

Fuente: Elaboración propia.

Nota: Si el p-value (valor de probabilidad) se aproxima a 1 el valor no es significativo, si P-value se aproxima a 0 el valor es significativo es decir el factor tiene mayor efecto sobre la variable respuesta de frecuencia de imágenes (FPS).

4. Decisión

- **Resultados de estimación de los efectos promedios para el tiempo de renderización**

  Los resultados de la estimación de efectos promedios mostrados en la tabla N° 22 se obtuvieron procesando la frecuencia de imágenes (FPS) registrados en cada dispositivo móvil para lo cual se utilizó el software minitab.

TABLA N° 22 : Estimación de los efectos promedios para la frecuencia de imágenes (FPS)

Promedios para la frecuencia de imágenes (FPS)	
Memoria interna (α_i)	-9367
Memoria RAM (β_j)	8.717
GPU(δ_K)	23.250

Fuente: Elaboración Propia.

- **Resultados de los efectos principales para la frecuencia de imágenes(FPS)**

TABLA N° 23 : *Medias del gráfico de efectos principales de la frecuencia de imágenes (FPS).*

Nivel	Memoria Interna	Memoria Ram	GPU
	Media	Media	Media
-	38.1765	35.5	32.7692
+	51.2308	52.4667	52.2941

Fuente: Elaboración propia.

- **Gráfico de efectos principales para la frecuencia de imágenes(FPS)**

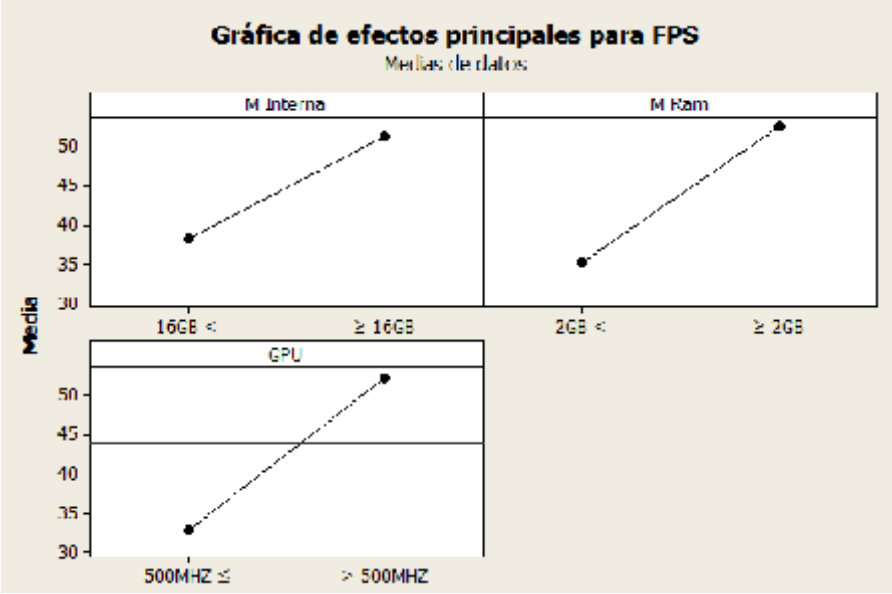


FIGURA N° 18 : Gráfico de efectos principales para la media de frecuencia de imágenes en dispositivos móviles.
Fuente: Elaboración propia.

Interpretación: En la figura N° 18 se observa que el factor GPU es el que más efecto tiene sobre la respuesta de fotogramas por segundo (FPS) en dispositivos móviles, puesto que la línea no es horizontal en consecuencia hay efecto principal (diferentes niveles de un factor afecta la respuesta de manera diferente).

Los dispositivos con frecuencia del reloj central menor o igual a 500 Mhz tienen una tasa media menor a los dispositivos que tienen una frecuencia del reloj central mayor a 500 Mhz, esto indica que cuando en un dispositivo móvil hay mayor frecuencia del reloj central del factor GPU, aumenta la frecuencia de imágenes(FPS) y lo contrario sucede cuando hay menor frecuencia del reloj central, disminuye la frecuencia de imágenes (FPS) ocasionando que al interactuar con el aplicativo exista baja fluidez en la animación.

La memoria interna y la memoria RAM parecen afectar en la frecuencia de imágenes (FPS) debido a que sus pendientes son pequeñas por los cual se deberán evaluar en el diagrama de Pareto y la gráfica de efectos estandarizados para ver si estos dos factores afectan significativamente o no en la frecuencia de imágenes (FPS).

Cálculo de la suma de cuadrados para el análisis de varianza para el tiempo de renderizado.

$$SS_{Total} = SS_{\alpha_i} + SS_{\beta_j} + SS_{\delta_K} + SS_{\alpha_i \beta_j} + SS_{\alpha_i \delta_K} + SS_{\beta_j \delta_K} + SS_{\alpha_i \beta_j \delta_K} + SS_{error}$$

$$SS_{\alpha_i} = (CONTRASTE(\boldsymbol{\alpha_i}))^2 / (2)^K * n$$
$$SS_{\beta_j} = (CONTRASTE(\boldsymbol{\beta_j}))^2 / (2)^K * n$$
$$SS_{\delta_K} = (CONTRASTE(\boldsymbol{\delta_K}))^2 / (2)^K * n$$
$$SS_{\alpha_i \beta_j} = (CONTRASTE(\boldsymbol{\alpha_i \beta_j}))^2 / (2)^K * n$$
$$SS_{\alpha_i \delta_K} = (CONTRASTE(\boldsymbol{\alpha_i \delta_K}))^2 / (2)^K * n$$
$$SS_{\beta_j \delta_K} = (CONTRASTE(\boldsymbol{\beta_j \delta_K}))^2 / (2)^K * n$$
$$SS_{\alpha_i \beta_j \delta_K} = (CONTRASTE(\boldsymbol{\alpha_i \beta_j \delta_K}))^2 / (2)^K * n$$

$$SS_{Error} = SS_{Total} - SS_{\alpha_i} - SS_{\beta_j} - SS_{\delta_K} - SS_{\alpha_i \beta_j} - SS_{\alpha_i \delta_K} - SS_{\beta_j \delta_K} - SS_{\alpha_i \beta_j \delta_K}$$

$$SS_{Error} = 2301.57$$

- **Diagrama de Pareto de efectos estandarizados**

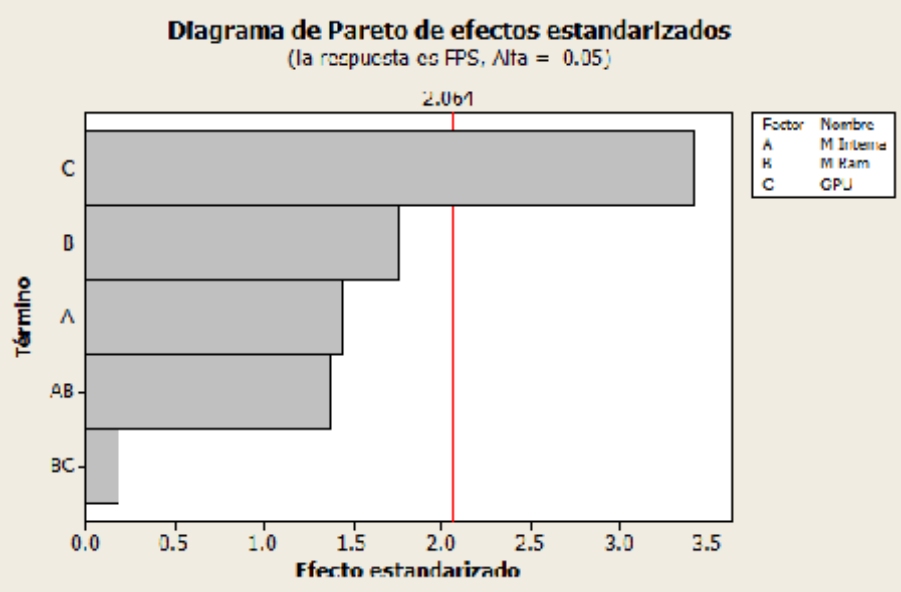


FIGURA N° 19 : Diagrama de Pareto de efectos estandarizados.
Fuente: Elaboración propia.

Interpretación

En la figura N° 19 del diagrama de Pareto, las barras que cruzan la línea de referencia son significativas es decir la barra C la cual es el factor GPU es el único que cruza la línea de referencia. Este factor es estadísticamente significativo en el nivel de 0.05. Este gráfico nos ayuda a determinar cuáles efectos son grandes, pero no determina cuál efecto aumenta o disminuye la respuesta de la frecuencia de imágenes (FPS), para ello se utilizó el gráfico de probabilidad de los efectos estandarizados que permite examinar la magnitud y dirección de los efectos de esta investigación.

- **Gráfico normal de los efectos estandarizados**

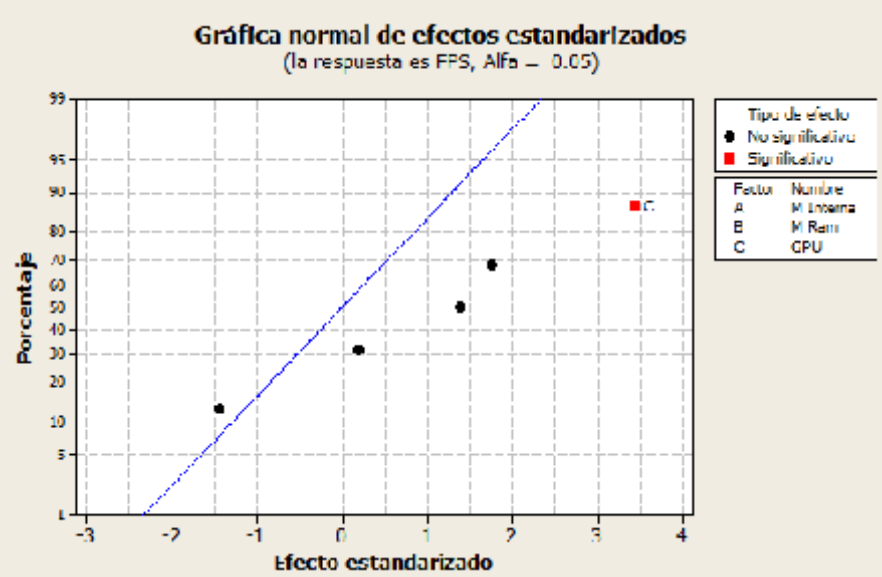


FIGURA N° 20 : Gráfico normal de efectos estandarizados para la frecuencia de imágenes (FPS).
Fuente: Elaboración propia.

Interpretación factor GPU

La figura N° 20 nos permitió determinar la magnitud, dirección e importancia de los efectos. En esta investigación la GPU que representa el punto rojo C es el único factor significativo en el nivel 0.05. Este punto se encuentra alejado y su forma es diferente

en comparación con los otros factores no significativos (memoria interna y memoria RAM).

Además, el factor GPU tiene un efecto estandarizado positivo debido a que se encuentra en el lado derecho de la línea vertical esto indica que cuando la frecuencia del reloj central es mayor a 500 Mhz, la frecuencia de imágenes (FPS) aumenta en dispositivos móviles.

TABLA N° 24 : Valores extremos de la media del efecto principal del factor GPU

Nivel	Media
-	32.7692
+	52.2941

Fuente: Elaboración propia

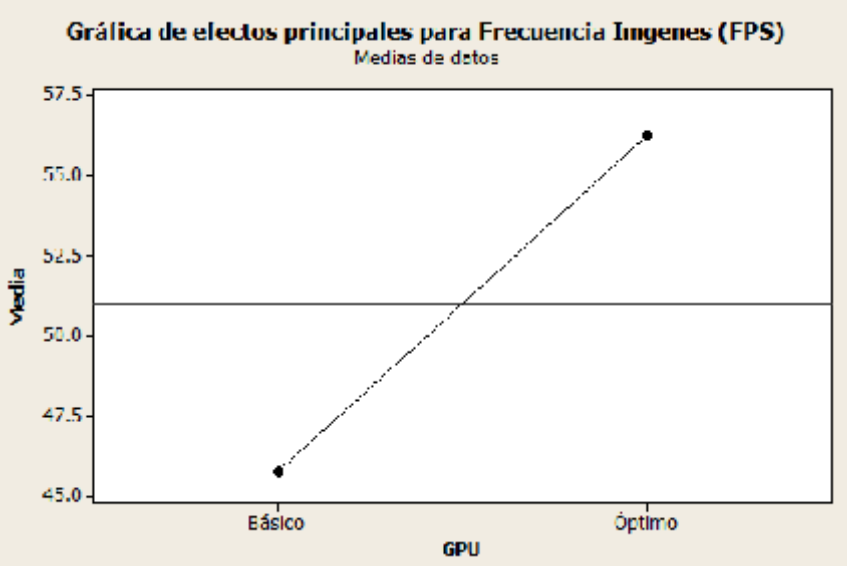


FIGURA N° 21 : Gráfico de efectos principales para frecuencia de imágenes (FPS).
Fuente: Elaboración propia.

Interpretación GPU

La figura N° 21 de efectos principales para factor GPU se analizó por ser el único factor que influye en el tiempo de renderizado, el cual es significativo.

Se puede ver que el nivel básico de la frecuencia del reloj central menor o igual a 500 Mhz de la GPU tuvo una tasa media de tiempo de renderizado menor al nivel óptimo de la frecuencia del reloj central mayor a 500 Mhz.

También, se observa que es un efecto positivo cuando cambia de nivel básica a nivel óptimo de GPU habiendo mayor frecuencia de imágenes (FPS) en dispositivos móviles con el aplicativo 3D.

Análisis de datos

Factor: Memoria interna (α_i)

Con un 95% de confiabilidad en el análisis de varianza se obtuvo que el p-value=0.163. Al evaluar obtenemos que 0.163 > 0.05 entonces se rechaza la hipótesis alterna y se acepta la hipótesis nula, por lo que se afirma que modelado 3D del Templo Colonial de Chuquinga con ThreeJS según la memoria interna no influye significativamente en la frecuencia de imágenes (FPS) en dispositivos móviles.

Factor: Memoria RAM (β_j)

Con un 95% de confiabilidad en el análisis de varianza se obtuvo que el p-value=0.091. Al evaluar obtenemos que 0.091>0.05 entonces se rechaza la hipótesis alterna y se acepta la hipótesis nula, por lo que se afirma que modelado 3D del Templo Colonial de Chuquinga con ThreeJS según la memoria RAM no influye significativamente en la frecuencia de imágenes (FPS) en dispositivos móviles.

Factor: GPU (Υ_K)

Con un 95% de confiabilidad en el análisis de varianza se obtuvo que el p-value=0.002. Al evaluar obtenemos que 0.002<0.05 entonces se acepta la hipótesis alterna y se rechaza la hipótesis nula, por lo que se afirma que modelado 3D del Templo Colonial de Chuquinga con ThreeJS según la GPU influye significativamente en la frecuencia de imágenes (FPS) en dispositivos móviles.

6.2 Desarrollo de la Aplicación móvil "VirtualTemp"

6.2.1 Introducción

Este proyecto de investigación consiste en el desarrollo un sitio web móvil "VirtualTemp". Donde el objetivo general es evaluar cómo influye el modelado 3D del Templo Colonial de Chuquinga con el Framework ThreeJS en el tiempo de renderización y frecuencia de imágenes (FPS) en dispositivos móviles. Las cuales se describirá desde la selección del sitio hasta el modelado de cada una de las estructuras y herramientas que se utilizaron para mejorar el renderizado.

6.2.2 Análisis de requerimientos del sistema

En vista de que se desea crear un entorno virtual que detalle el Templo Colonial de Chuquinga. Se toma una exigencia en desarrollar cierto nivel de detalle en su interior que conservan restos importantes de pinturas murales, entre ellas la alegoría de Árboles de la vida, sencillas pinturas que cubren las columnas del arco triunfal, tejas pintas. Por lo tanto, y considerando que los resultados deben ser vistos en dispositivos con bajas características además de una buena textura, el aplicativo móvil debe cumplir con las siguientes características.

- Alto nivel de detalle en las estructuras arquitectónicas.
- Buena calidad en las texturas.
- Buen manejo de luces y sombras.
- Interacción con el entorno.
- Buenos desempeños en dispositivos móviles con baja características.

En el mercado existe muchas herramientas pero no hay una con la cual podamos cubrir todas las necesidades, por lo tanto se utilizará una serie de ellas, las cuales fueron seleccionadas en base al buen manejo y conocimiento.

- **Herramientas**

Herramientas utilizadas para el aplicativo web móvil 3D

TABLA N° 25 : Herramientas para el desarrollo

	Elemento	Selección
	Framework MVC para Php.	Laravel 5.6
	IDE de programación	Sublime text
	Diseño de Interfaz	Bootstrap
	Control de versiones	Github
	Lenguajes de programación	Php
	Software de modelado 3D	3Ds Max
	Software de diseño CAD	Autocad
	Framework para generar gráficos 3D	ThreeJS
	Editor Gráfico	Photoshop cc

Fuente: Elaboración propia.

6.2.3 Modelado 3D

A. Selección del templo colonial

En esta primera etapa se selecciona el entorno virtual a crear. Desde los inicios, la idea fue trabajar con los templos coloniales de Apurímac.

Después de analizar varios templos de Apurímac, se optó por el Templo Colonial de Chuquinga, ubicada en el distrito de Chalhuanca, provincia de Aymaraes. En una comunidad campesina quechua, cuyo centro poblado se encuentra a 2800 msnm, sobre una terraza elevada en la margen izquierda del rio Chalhuanca. Los factores que se consideraron para la selección son: accesibilidad, que tenga detalles y gran variedad de diseños, importancia del lugar. Todos estos requisitos los reunió el Templo Colonial de Chuquinga: Es un templo de la época virreinal y está hecha de piedra, barro de mampostería ordinario y adobe. En el interior existen algunos murales y en la nave existen retablos y ornacinas de barro, en las cuales encontramos esculturas religiosas. Eso sin duda le da una importancia indiscutible. Tiene un plano con información de sus medidas.

B. Creación del Templo Colonial de Chuquinga

Esta segunda etapa consiste en la creación de todos los objetos y elementos que conforman el modelado 3D del templo. En esta fase es donde empieza a tener forma el sitio web móvil “VirtualTemp” y las tareas a desarrollar principalmente son el modelado y su texturizado.

Para el desarrollo el modelado 3D del templo colonial se usó la herramienta de modelado tridimensional 3ds Max, ofrece gran capacidad para crear cualquier cosa existente en la vida real, las cual fue utilizado para algunas figuras

geométricas complejas, como sillas, puerta de entrada y los muros del templo, todo lo demás fue realizado directamente en el aplicativo con la Framework ThreeJS lo cual facilitó el desarrollo del modelado del Templo Colonial de Chuquinga.

- **Modelado**

  El modelado del Templo Colonial de Chuquinga, se diseñó en base al plano catastral Chalhuanca 2004, que se muestra en las figura N° 22 y N° 23.

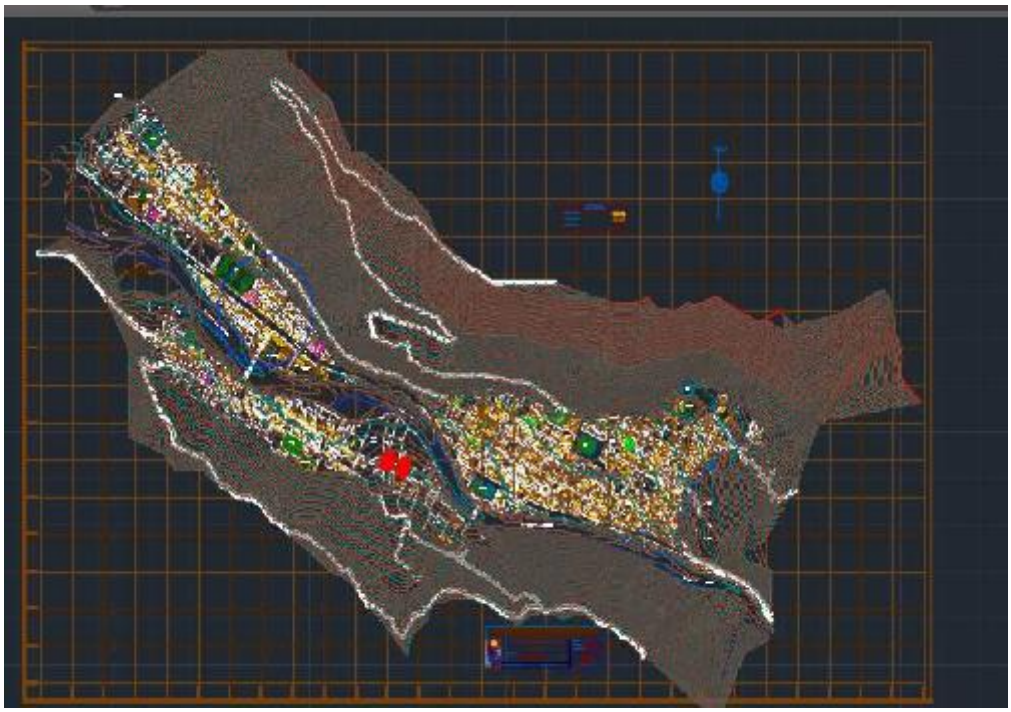


FIGURA N° 22 : Plano catastral de Chalhuanca.
Fuente: Municipalidad Provincial de Aymaraes.

La figura N° 24 muestra parte del plano catastral del Templo Colonial de Chuquinga.

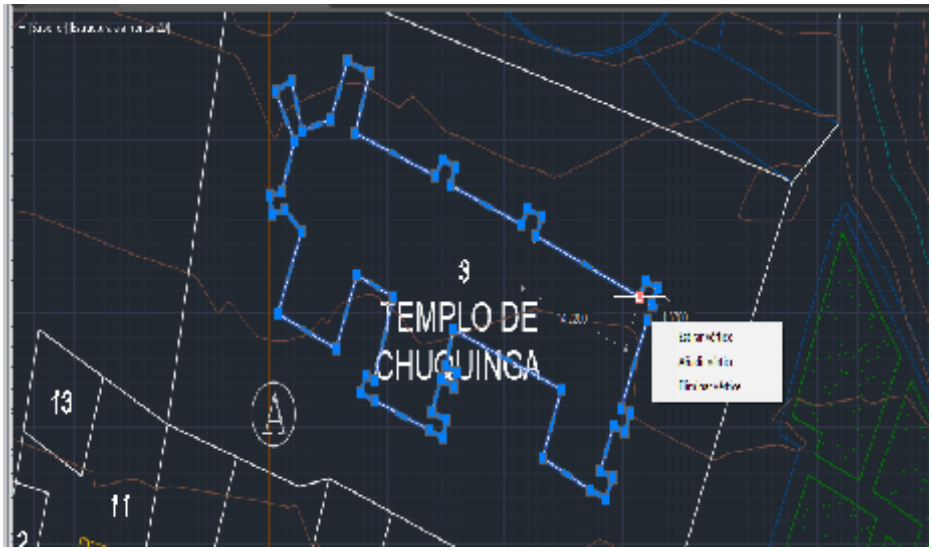


FIGURA N° 23 : Plano del Templo Colonial de Chuquinga.
Fuente: Municipalidad Provincial de Aymaraes.

Todas las partes del modelado 3D del Templo Colonial de Chuquinga, fueron creadas con la ayuda de las medidas que nos proporcionaron, en el plano de AutoCAD.

Este plano fue obtenido de la Municipalidad Provincial de Aymaraes, con la finalidad de hacer un modelado 3D con las medidas más próximas posibles al diseño real.

Dentro del plano en AutoCAD, podemos encontrar las siguientes unidades de medidas configuradas: Longitud (decimal), precisión (0.0000), unidad de escala (metros). Las medidas son las siguientes: El contorno del Templo Colonial mide una aproximado de 191.49 metros, el grosor de las paredes miden 0.031 metros, el baptisterio mide 7.17 metros, bautisterio 8.69 metros, y la que sigue después del bautisterios mide 6.39 metros.

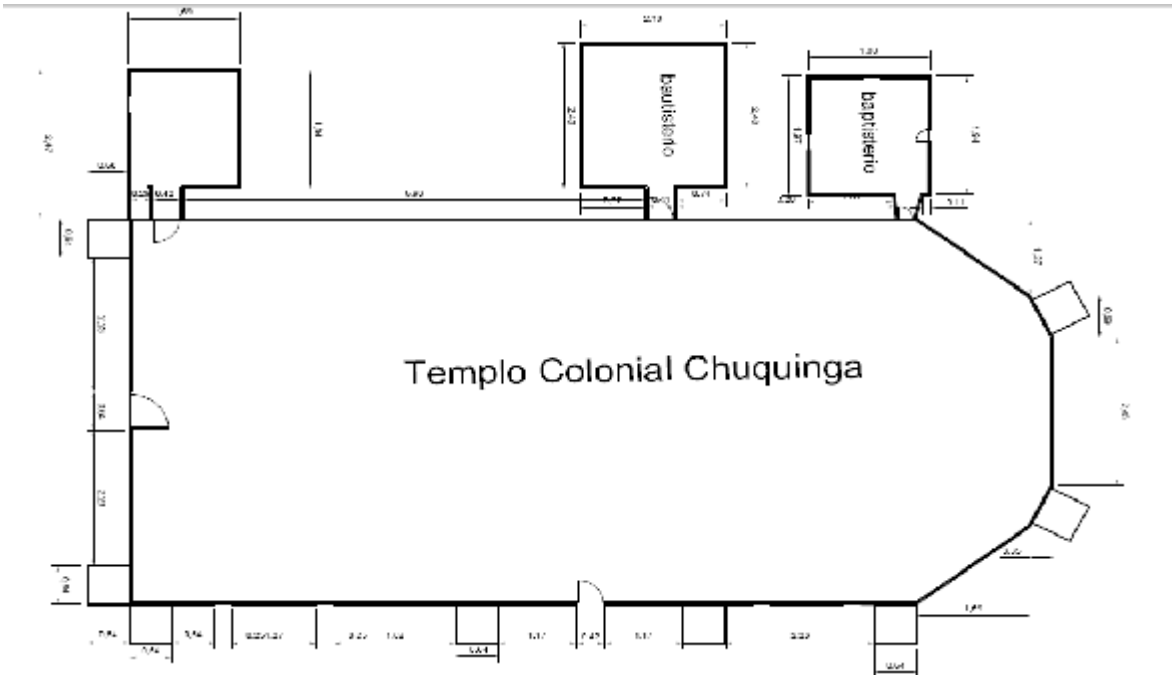


FIGURA N° 24 : Plano del Templo Colonial con sus medidas.

El modelado consiste en ir dando forma a objetos individuales que ya culminados serán exportados en un formato ligero, para poder ser utilizados en el aplicativo. En esta etapa se modelará a través de la técnica editable poly que consiste en modificar totalmente a nuestro placer los polígonos a través de vértices, aristas y borde. Su uso es similar al de un objeto de malla editable.

Las geometrías de las partes del templo serán creadas con pocos polígonos, para que el dispositivo móvil lo procese de forma rápida con el Framework ThreeJS, la cual utiliza WebGL, que es un API implementada en JavaScript para la renderización de gráficos en 3D dentro de cualquier navegador web.

C. Modelo y ambiente del Templo Colonial Chuquinga

Para crear el Templo Colonial en 3D y su ambiente, se realizó varios pasos:

- **Diseño del plano detallado del templo.** Creación del plano del Templo Colonial de Chuquinga, a partir del plano catastral de Chalhuanca. Con medidas exactas.
- **Importación del plano.** La importación del plano con las medidas para la generación de modelado 3D.
- **Modelado.** Generación del templo en 3D a partir del plano de autocad, con el apoyo de la herramienta de modelado.
- **Texturizado.** Creación de texturas a partir de fotos tomadas del templo la cual serán parte del modelado y serán mapeados para poder colocar la textura al lugar que corresponda.
- **Escenario.** Creación del escenario.
- Modelando y texturizando los elementos 3D del Templo Colonial de Chuquinga con el Framework ThreeJS.
- Integración de todos los modelados 3D del Templo Colonial de Chuquinga con el Framework ThreeJS.

Estos pasos son lo que a continuación se representaran de una manera clara para su desarrollo.

A) Diseño del plano detallado del templo

En todo proceso de diseño de un espacio arquitectónico existente, el primer paso siempre es tomar las medidas y reflejar en un croquis mediante cotas y anotaciones. Para la elaboración del Templo Colonial Chuquinga, se pudo obtener las medidas y un plano individual a partir del plano catastral de Chalhuanca, como se muestra en la figura N° 25.

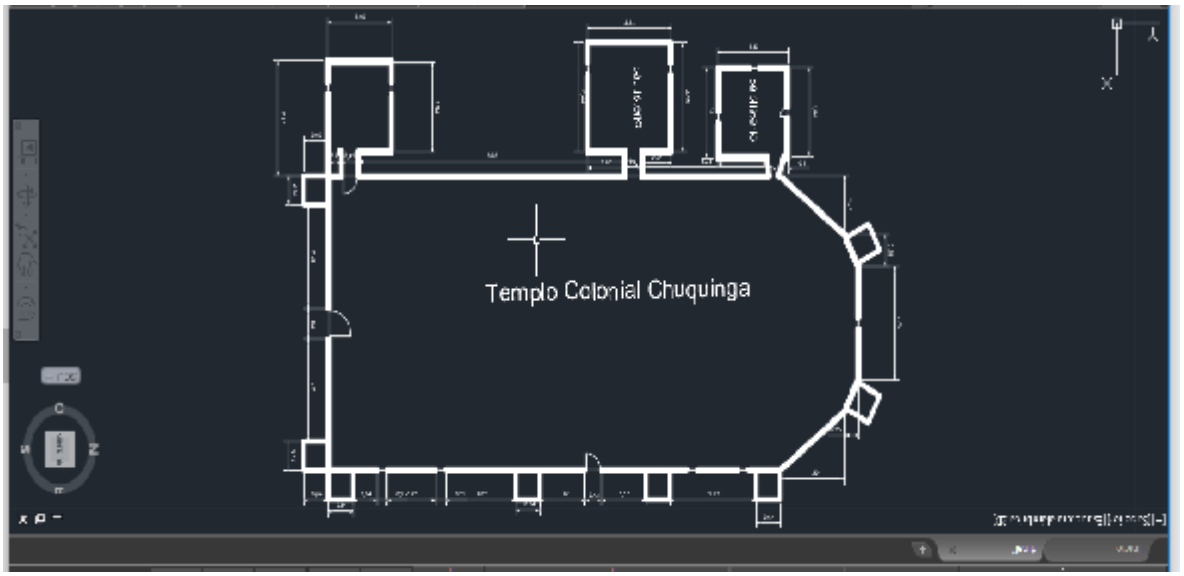


FIGURA N° 25 : Plano del Templo Colonial con sus medidas.
Fuente: Elaboración propia.

B) Importación del plano

El proceso de importación se realizó del software de AutoCAD al software 3ds Max, para poder ser modelado, como se muestra en las figuras N° 26, N° 27 y N° 28.

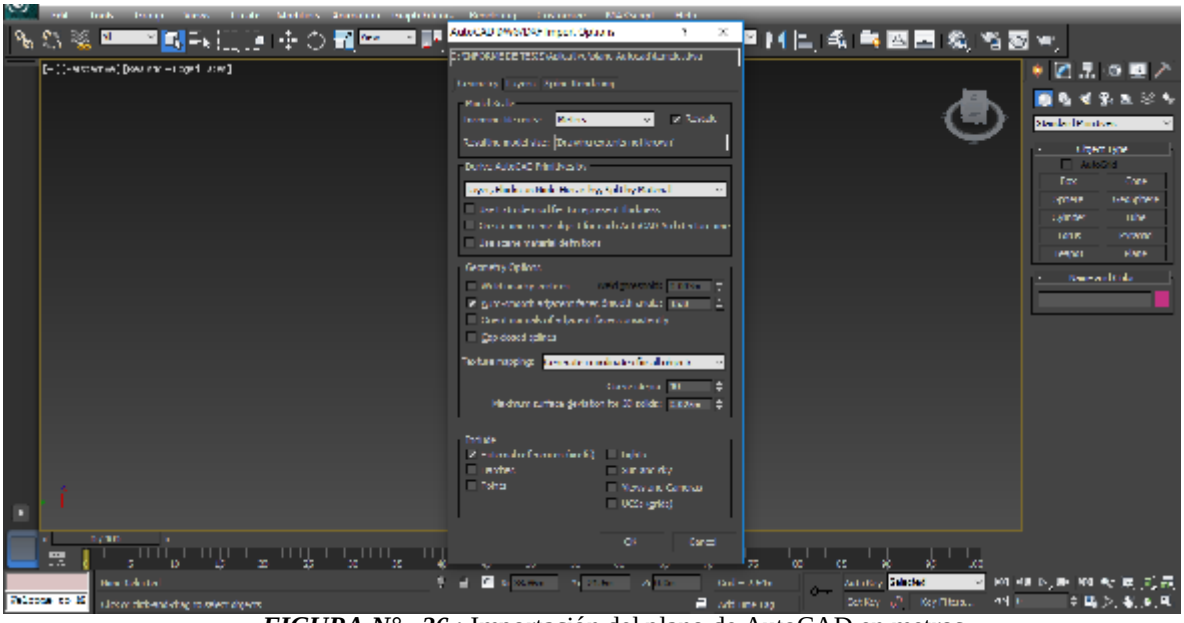


FIGURA N° 26 : Importación del plano de AutoCAD en metros.
Fuente: Elaboración propia.

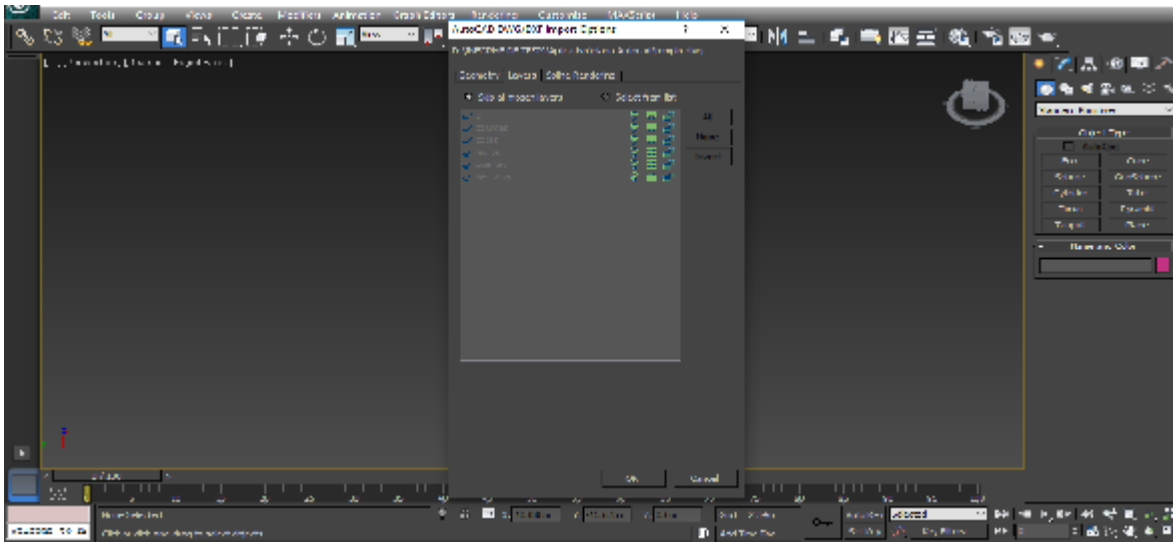


FIGURA N° 27 : Selección de las capas: muros, columnas, puertas, ventanas.
Fuente: Elaboración propia.

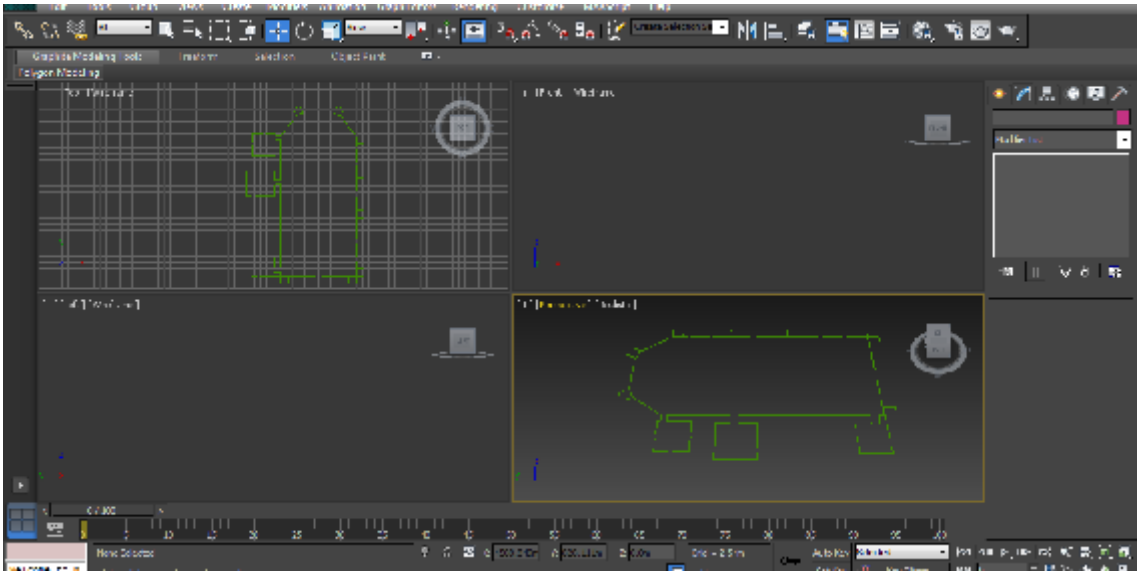


FIGURA N° 28 : Resultado final de la importación, en diferentes vistas: Top, front, left, perspectiva.
Fuente: Elaboración propia.

C) Modelado del Templo Colonial de Chuquinga

En el modelado del Templo Colonial de Chuquinga se utilizó la técnica editable poly, el cual permitió la extrusión de paredes, puertas y columnas como se muestra en las siguientes figuras N° 29 y N° 30.

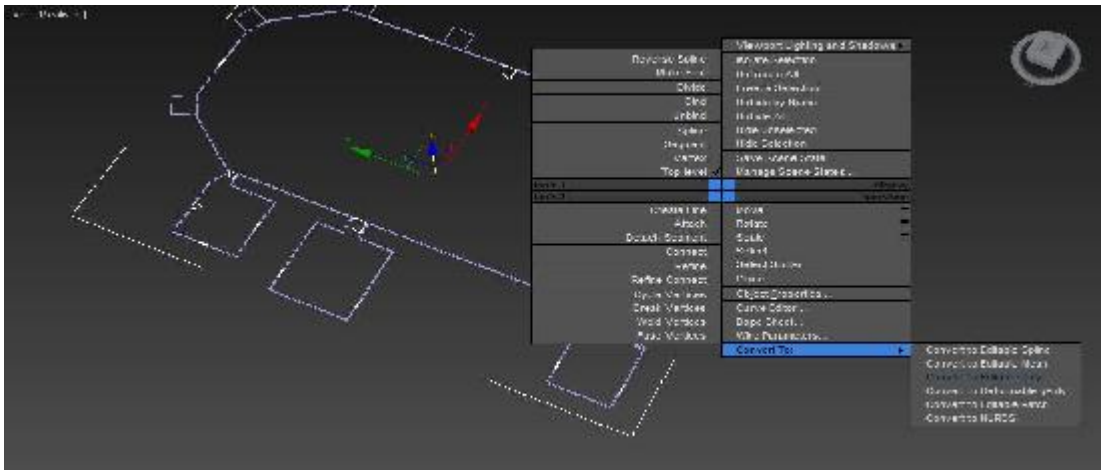


FIGURA N° 29 : Aplicando el modificador editable poly, su función son variables: para deformar paredes, crear objetos nuevos. La modificación se realiza a través de vértices, aristas y caras.
Fuente: Elaboración propia.

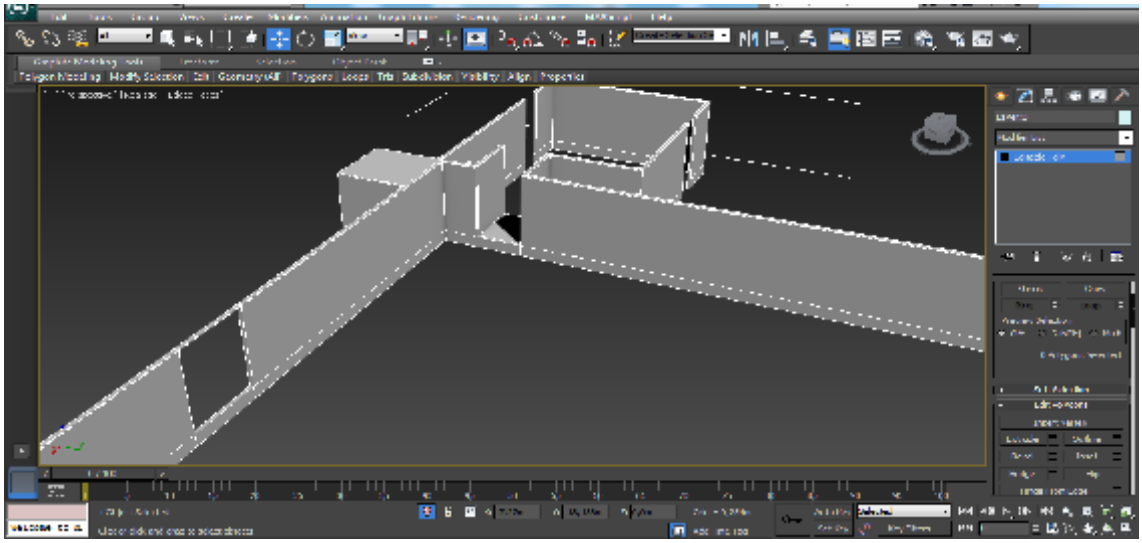


FIGURA N° 30 : Muros Levantados con el modificador editable poly, opción extrude, la cual nos permite dar volumen al plano importado de AutoCAD.
Fuente: Elaboración propia.

Después de aplicar la extrusión se procede a dar forma a los muros que rodean al templo, aplicando las transformaciones geométricas en los diferentes ejes como se muestra en la figura N° 31.

FIGURA N° 31 : Modificación de las columnas.
Fuente: Elaboración propia.

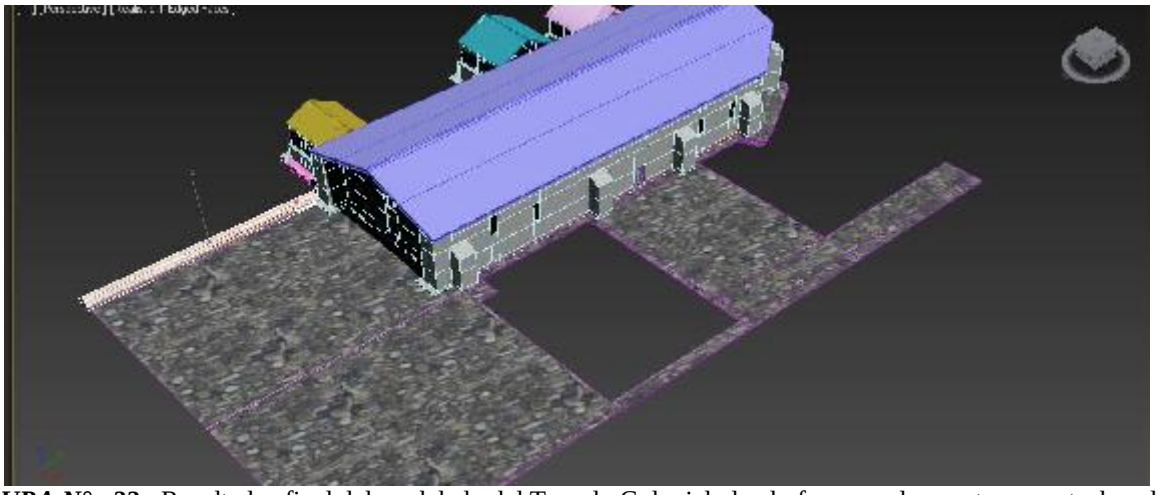


FIGURA N° 32 : Resultados final del modelado del Templo Colonial, dando forma en los contornos y techos del templo.
Fuente: Elaboración propia.

- **Modelado de las partes internas del Templo Colonial de Chuquinga, soporte del segundo piso**

  El objeto bidimensional fue creado partir de una foto tomada en el templo y utilizando el modificador lathe de 3ds Max se generó el objeto a partir de una forma bidimensional, girándola sobre uno de sus ejes dando forma al soporte.

FIGURA N° 33 : Modelo de soporte de la azotea del segundo piso.
Fuente: Elaboración propia.

Después de haber generado el soporte, se pasó a reducir el peso del modelado el cual consiste en ir eliminar aristas y vértices que conforman polígonos sin que esta pierda el detalle del objeto como se muestra en la figura N° 34.

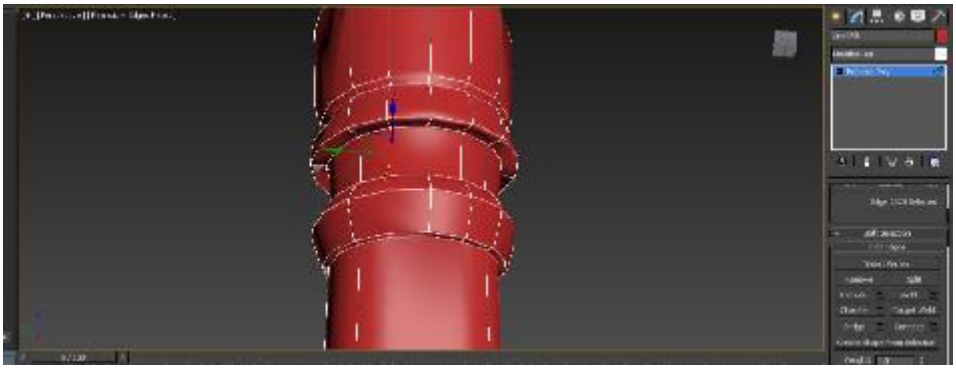


FIGURA N° 34 : Proceso de reducción del peso del soporte de la azotea.

Fuente: Elaboración propia.

- **Modelado de la parte atrás interna del Templo Colonial de Chuquinga**

El modelado se realizó a partir de una imagen tomada en templo, con la cual se pudo generan un objeto bidimensional para luego convertirla tridimensional con la herramienta extrude la cual añadió profundidad a la forma como se muestra en la figura N° 35.

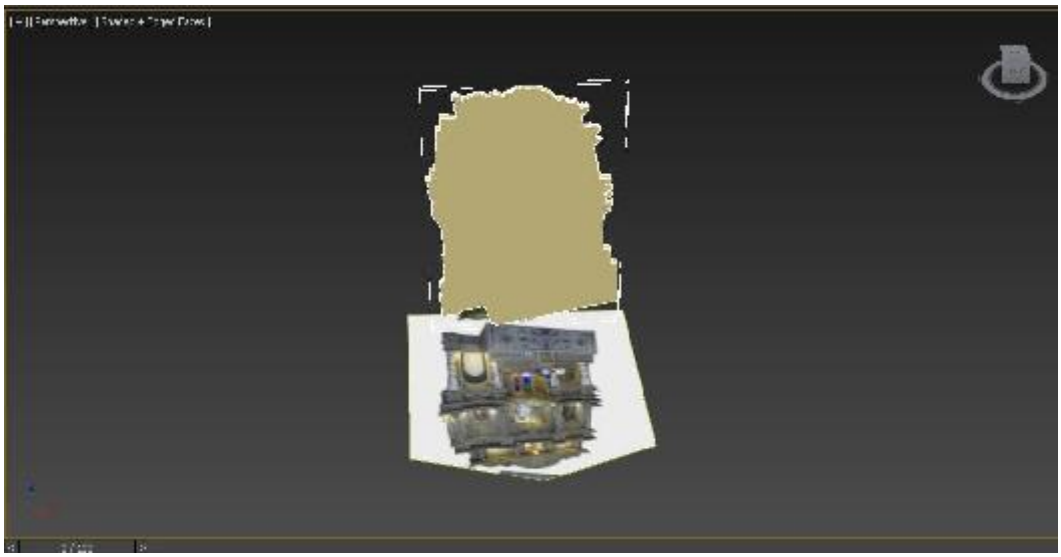


FIGURA N° 35: Objetos bidimensional convertida en tridimensional.

Fuente: Elaboración propia.

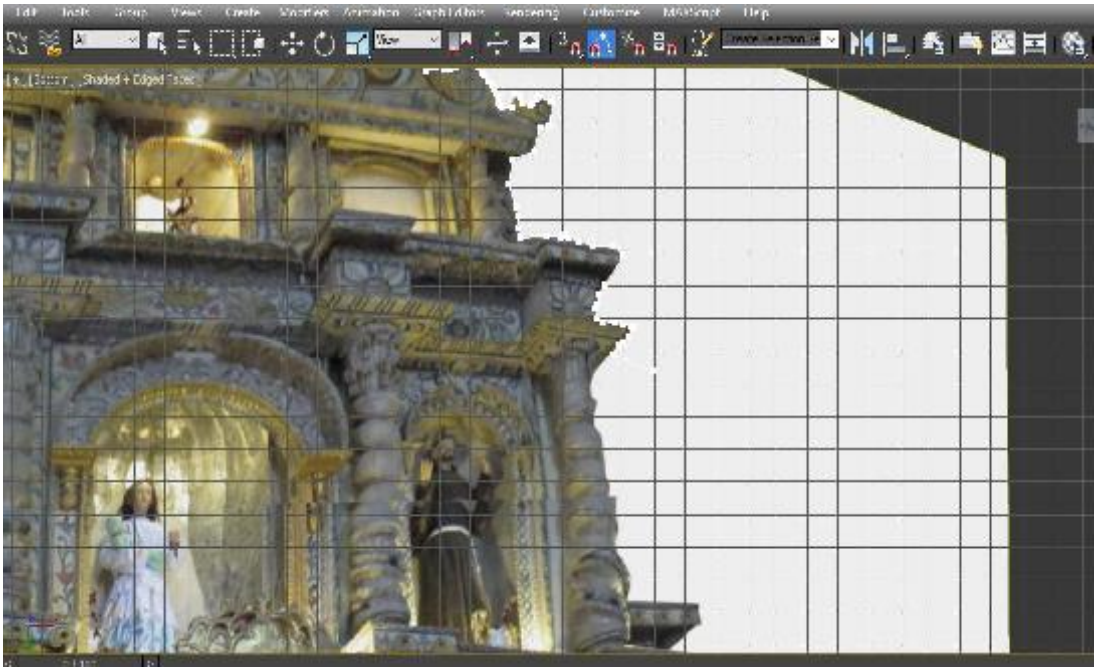


FIGURA N° 36 : Resultado de objeto tridimensional con la textura.
Fuente: Elaboración propia.

D) Texturizado del Templo Colonial de Chuquinga

Para el texturizado del Templo Colonial de Chuquinga se utilizó la técnica UVW mapping la cual permitió pintar las superficie en base a una textura sólida. Esto es útil para aplicar texturas complejas a cada una de las partes creadas del templo. Esta técnica de mapeo implica la creación de un mapa de plantilla para luego añadir la textura que se extrajeron de las fotos tomadas del Templo Colonial de Chuquinga.

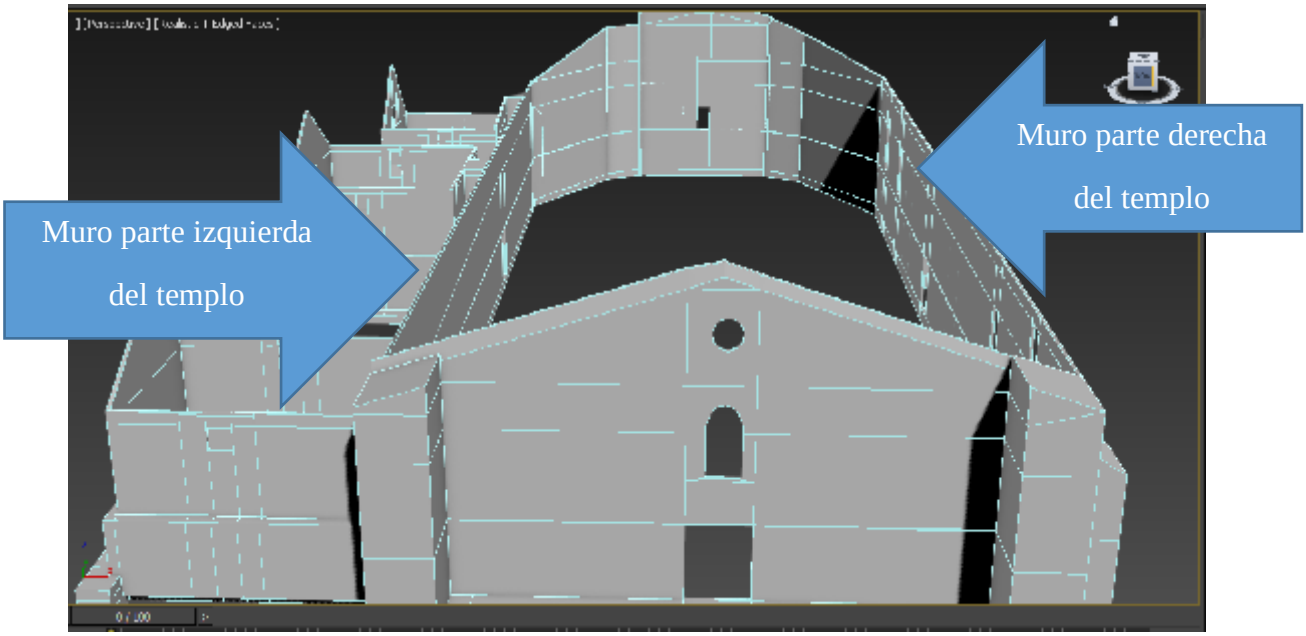


FIGURA N° 37 : Partes del Templo Colonial de Chuquinga para aplicar la textura.
Fuente: Elaboración propia.

- **Texturizado de la parte derecha del templo**

  Se generó la plantilla con la herramienta UVW mapping donde se puede apreciar todas las partes del lado derecho del templo. Los polígonos que conforman al objeto tuvieron que ser cuidadosamente ordenados para que sea más fácil el acoplamiento de las imágenes. Un vez listo se procedió a la exportación en formato jpeg, para aplicar la textura correspondiente en photoshop cc como se muestra en las figuras N° 38, N° 39, N° 40 y N° 41.

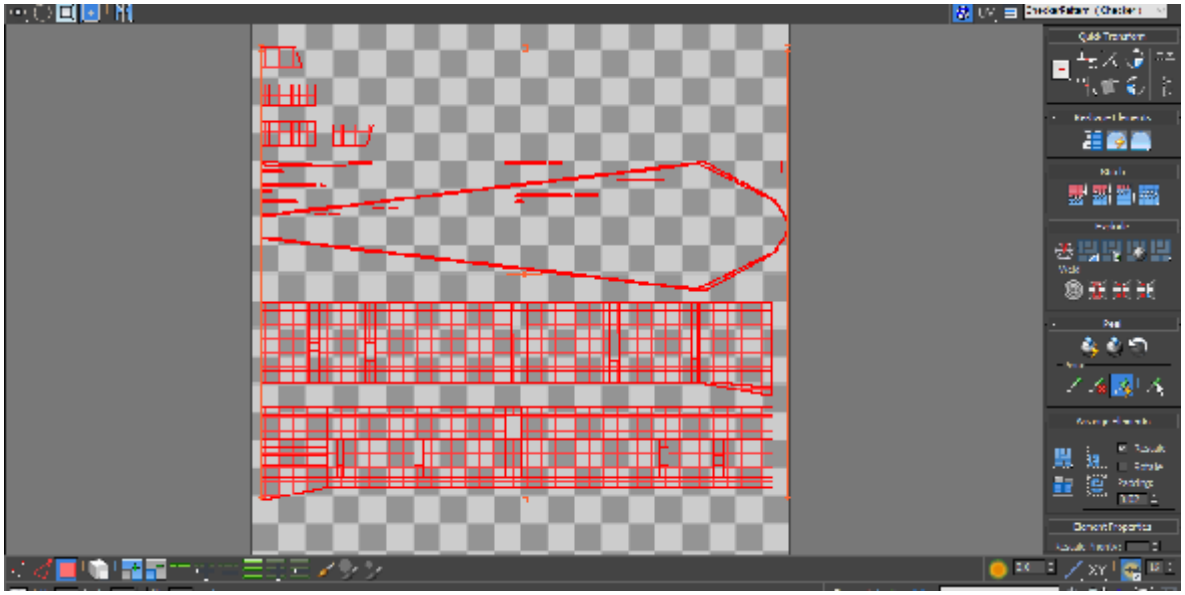


FIGURA N° 38 : Generación de la plantilla con la herramienta UVM mapping de la parte derecha del templo.
Fuente: Elaboración propia.

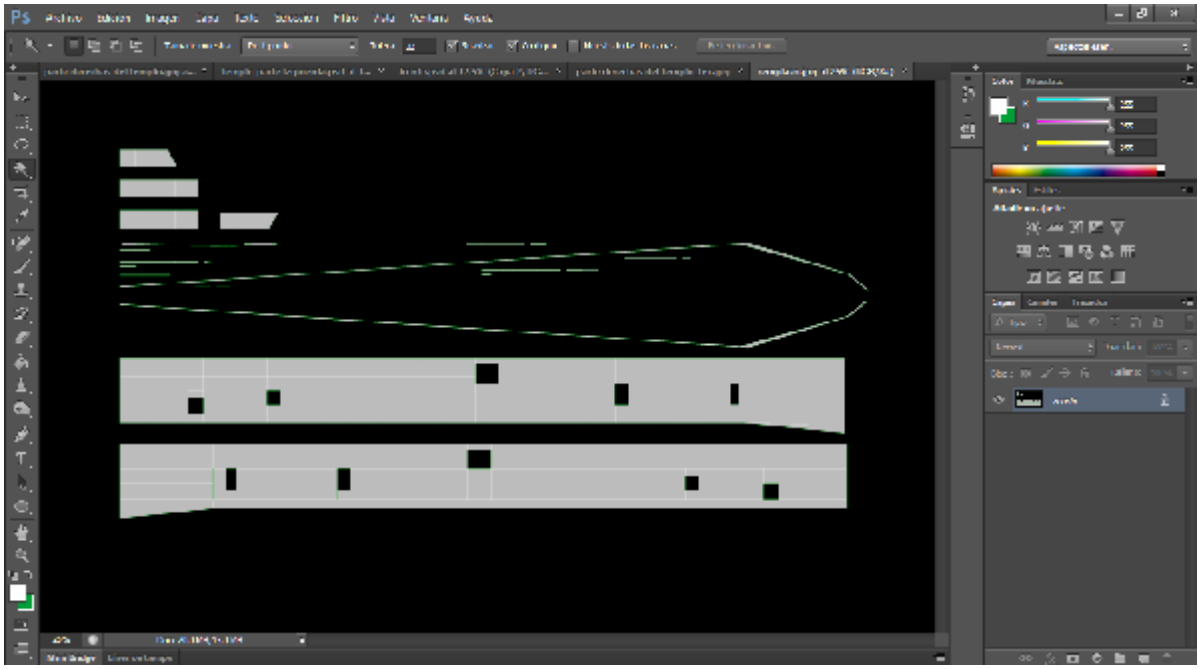


FIGURA N° 39 : Plantilla en formato jpeg de la parte derecha del templo.
Fuente: Elaboración propia.

Ya generada la plantilla en formato jpeg, se coloca la textura en cada una de las partes correspondientes, como se muestra a continuación.

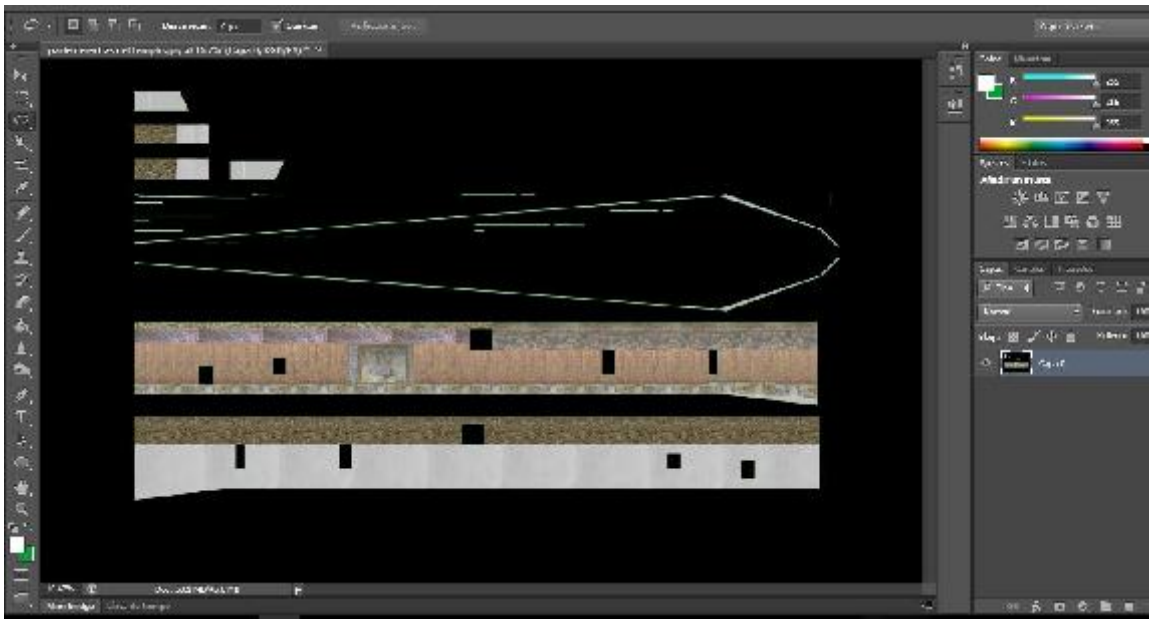


FIGURA N° 40 : Textura aplicada en el editor de imágenes.
Fuente: Elaboración propia.

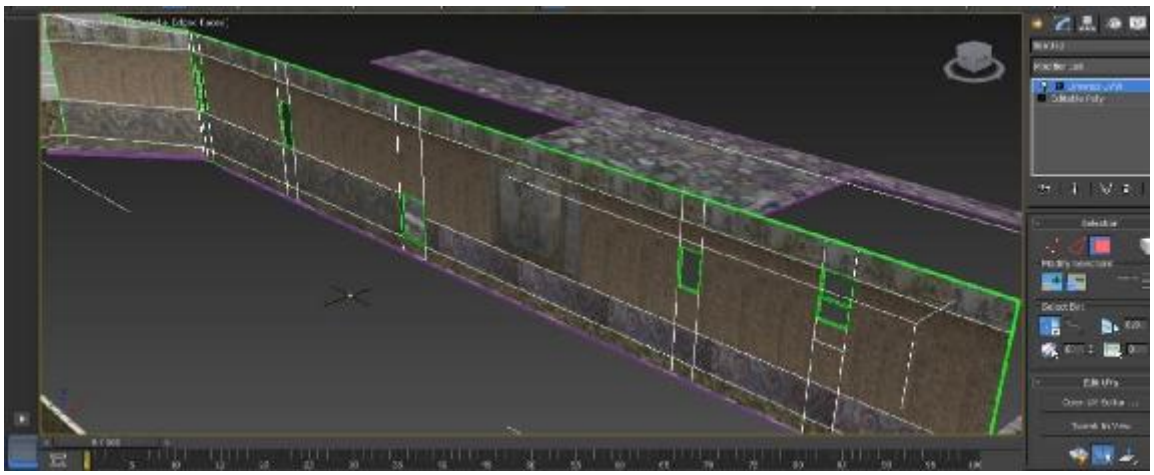


FIGURA N° 41 : Textura aplicada a la parte derecha del templo.
Fuente: Elaboración propia.

- **Texturizado de la parte izquierda del templo**

Utilizando la técnica UVW mapping se puede apreciar todas las partes del lado izquierdo del templo como se muestra en la figura N° 42. Los polígonos de la pared tuvieron que ser cuidadosamente ordenados para que sea más fácil el acoplamiento de las imágenes en el modelado. Un vez listo se procedió a la exportación en formato jpeg, para aplicar la textura correspondiente en photoshop cc como se muestra en las figuras N° 42, N° 43 y N° 44.

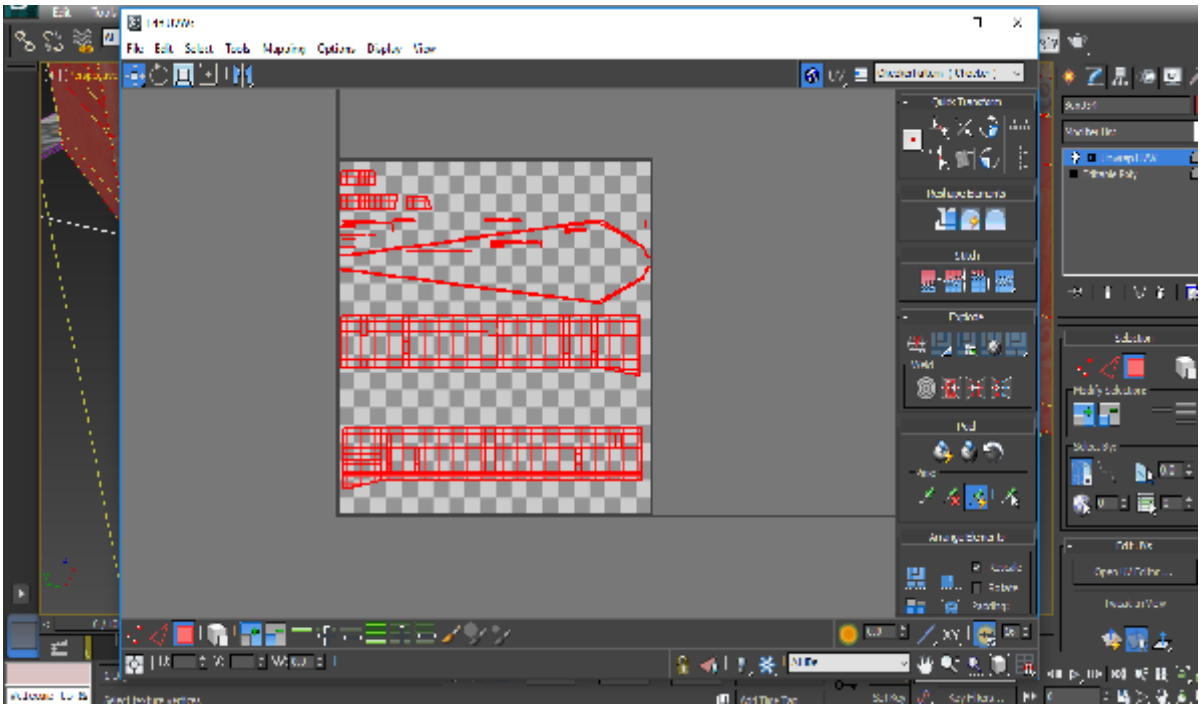


FIGURA N° 42 : Generación de la plantilla con la herramienta UVW mapping de la parte izquierda del templo.
Fuente: Elaboración propia.

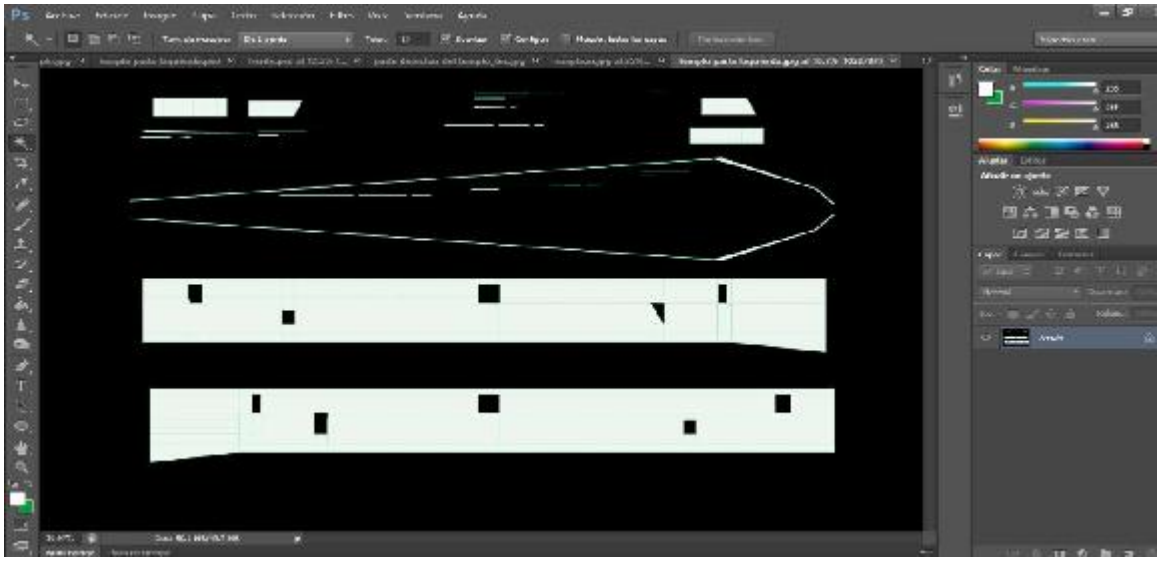


FIGURA N° 43 : Plantilla en formato jpeg de la parte izquierda del templo.
Fuente: Elaboración propia.

Ya generada la plantilla en formato jpeg, se procedió a colocar la textura en cada una de las partes correspondientes, como se muestra a continuación en la figura N° 44.

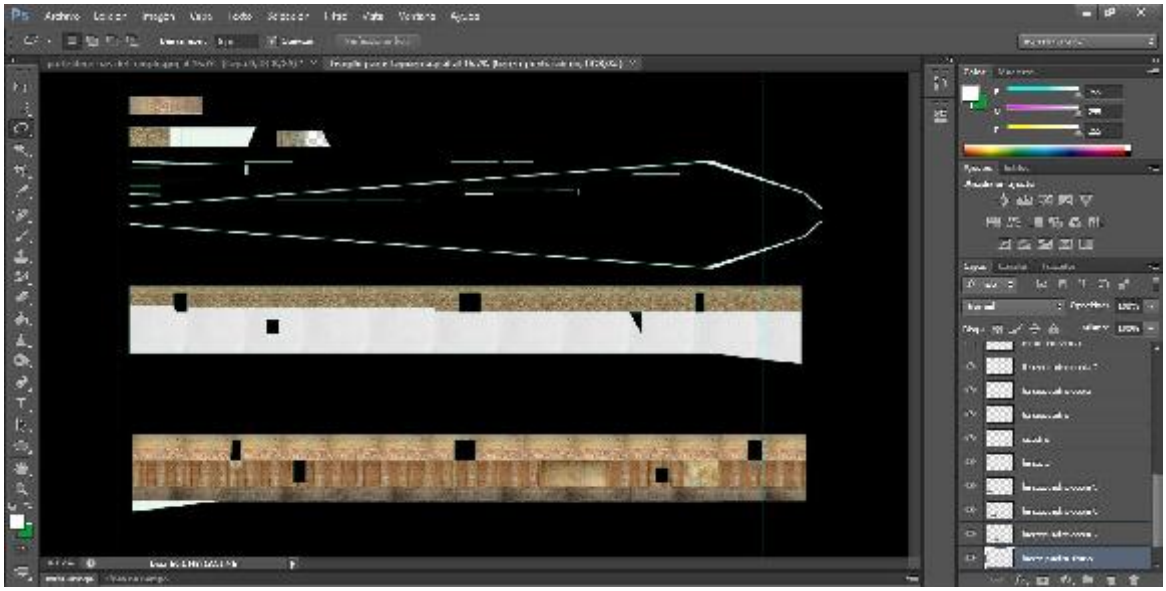


FIGURA N° 44 : Textura aplicada a la parte izquierda del templo.
Fuente: Elaboración propia.

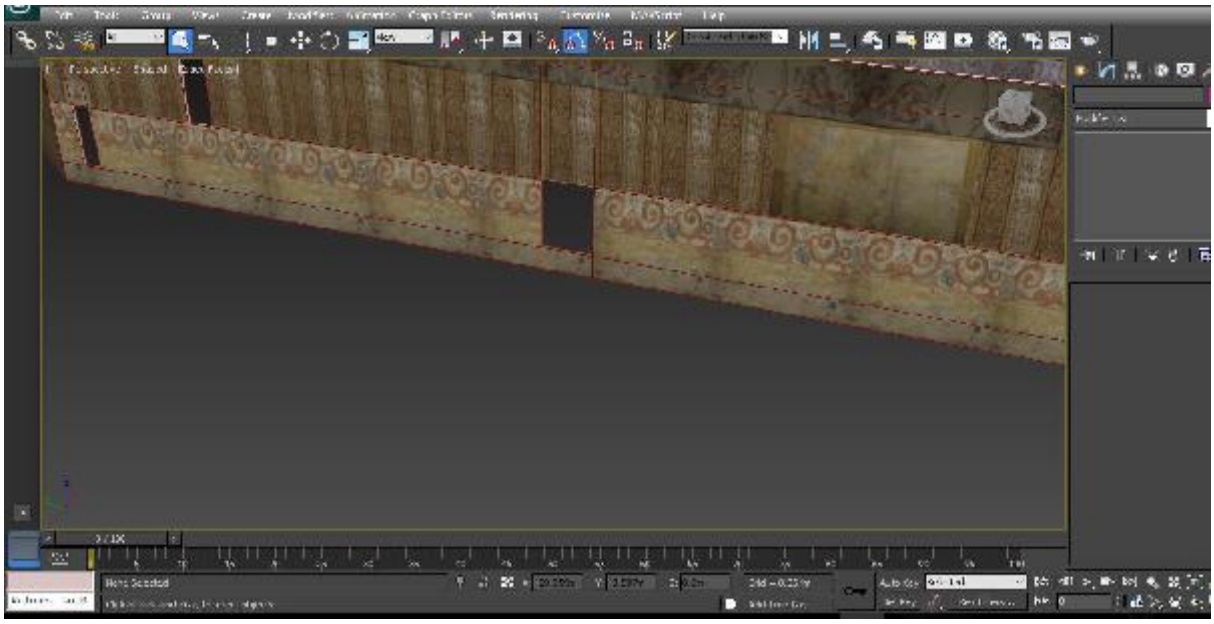


FIGURA N° 45 : Resultados de la plantilla aplicada al muro izquierda del templo.
Fuente: Elaboración propia.

- **Texturizado de la parte delantera del templo**

La parte delantera del templo fue texturizado teniendo en cuenta la forma del objeto debido a que es posible cometer errores con la técnica UVW mapping. Los múltiples polígonos que conforman la maya de la parte delantera tuvieron que ser ordenados de acuerdo al objeto para que no tengamos problemas en el acoplamiento de las imágenes como se muestra en la figura N° 46 y N° 47 .Un vez listo se procedió a la exportación en formato jpeg, para aplicar la textura correspondiente en photoshop cc.

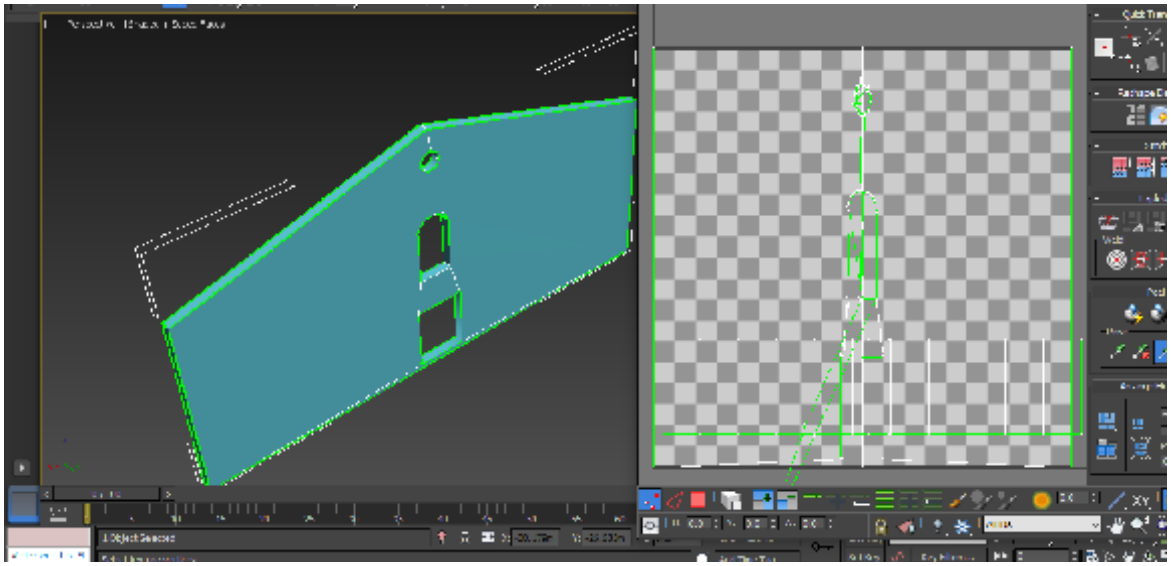


FIGURA N° 46 : Generación de la plantilla con la herramienta UVW mapping de la parte delantera del templos.
Fuente: Elaboración propia.

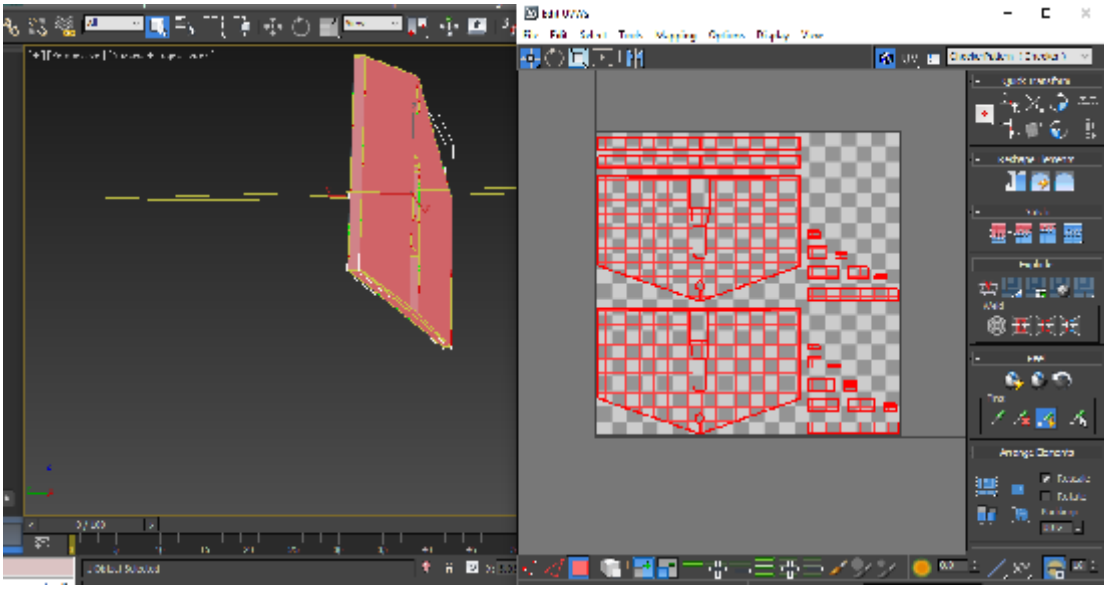


FIGURA N° 47 : Plantilla final ordenada para aplicar la textura en el editor.
Fuente: Elaboración propia.

- **Texturizado de los muros delanteros del templo**

  Aplicando la técnica UVW mapping se puede apreciar las partes del muro, agrupando y separado. Un vez listo se procedió a la exportación en formato jpeg, para aplicar la textura correspondiente en photoshop cc como se muestra en la figuras N° 48 y N° 49.

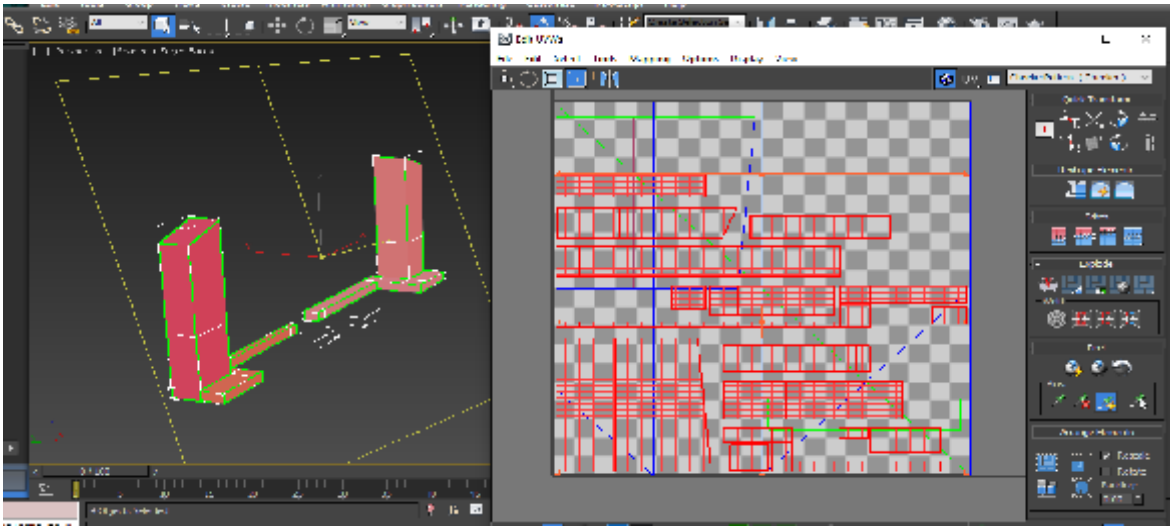


FIGURA N° 48 : Generación de la plantilla con la herramienta UVM mapping, de los dos muros delanteras.
Fuente: Elaboración propia.

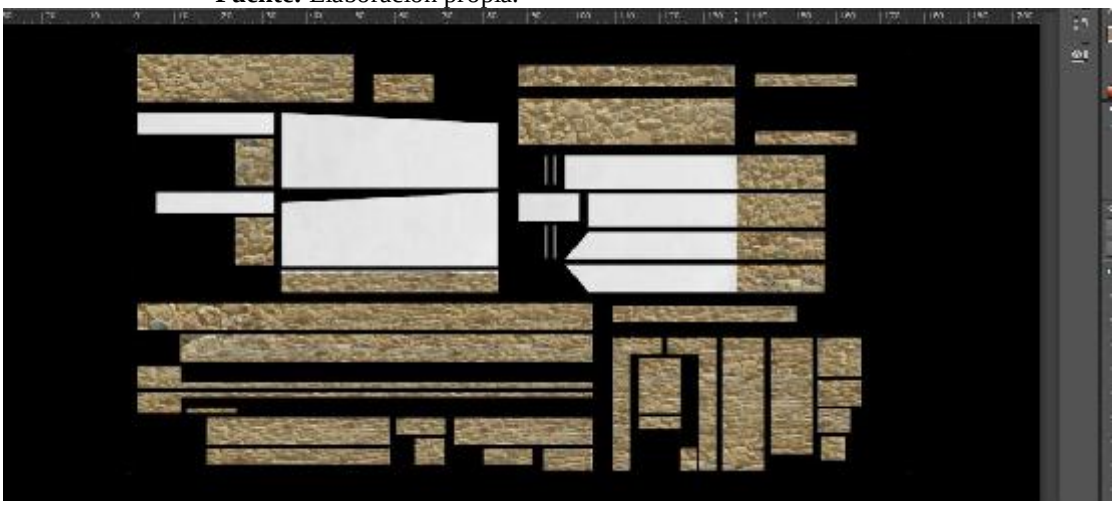


FIGURA N° 49 : Textura aplicada a los dos muros delanteros del templo.
Fuente: Elaboración propia.

- **Texturizado del arco del templo**

  La parte del arco de templo fue texturizado teniendo en cuenta la forma del objeto debido a que es posible cometer errores con la técnica UVW mapping. Los múltiples polígonos que conforman la maya de la parte del arco tuvieron que ser ordenados de acuerdo al objeto para que no tengamos problemas en el texturizado de las imágenes como se muestra en la figura N° 50 y N° 51 .Un vez listo se procedió a la exportación en formato jpeg, para aplicar la textura correspondiente en photoshop cc.

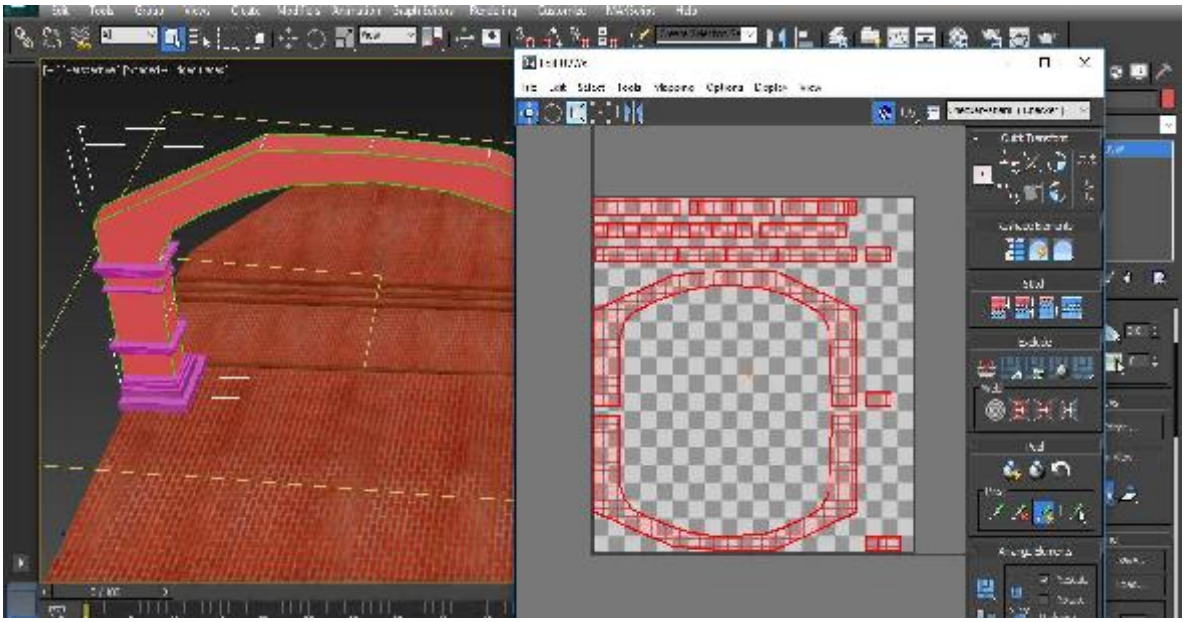


FIGURA N° 50 : Generación de la plantilla con la herramienta UVW mapping del arco interno del templo.
Fuente: Elaboración propia.

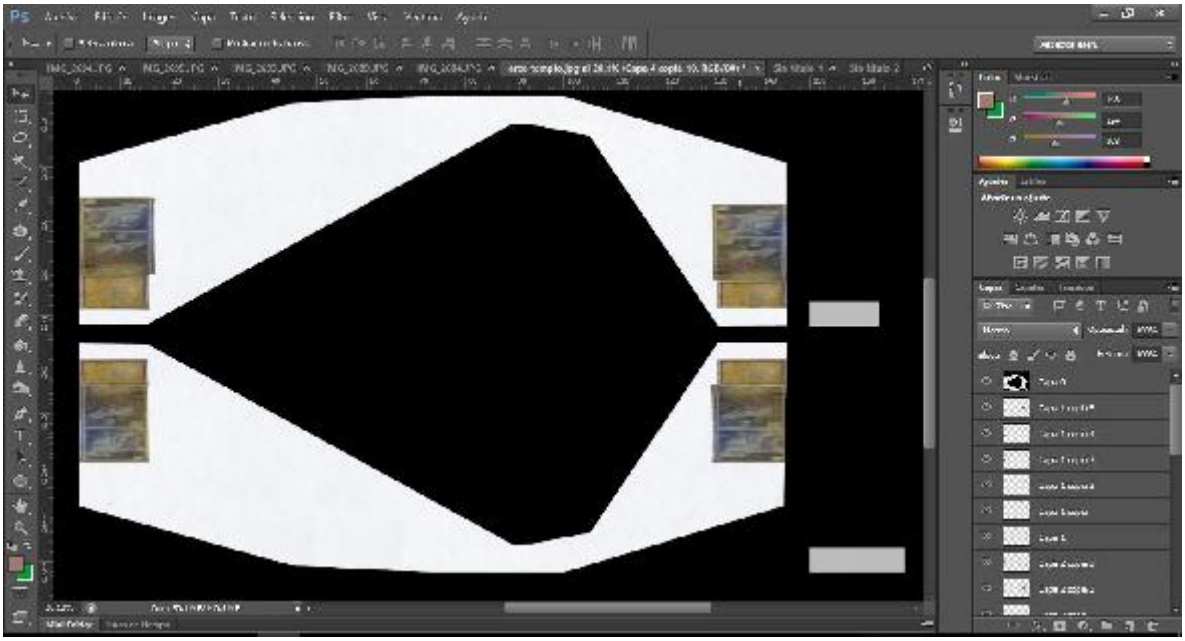


FIGURA N° 51 : Textura del templo colonial, aplicada a la plantilla generada con UVW mapping.
Fuente: Elaboración propia.

FIGURA N° 52 : Resultado del arco interno con textura.
Fuente: Elaboración propia.

E) Creación del escenario

Antes de iniciar con la creación del escenario, se elaboró una estructura básica que maneja el Framework ThreeJS, que nos permitirá integrar todos los objetos modelados en 3D del Templo Colonial de Chuquinga, aceleradas por el hardware, animadas e interactivas para ejecutarse cada dispositivo móvil a través del navegador, sin la necesidad de plugins, gracias al uso de WebGL.

➢ Estructura básica del Framework ThreeJS

En esta estructura se trabajó con tres cosas muy importantes como son: escena, cámara y renderizador para poder renderizar la escena con la cámara, mostrando el resultado de los modelados exportados 3D creados con el 3ds Max y también los elementos 3D creados directamente con el Framework, basados en WebGL, del Templo Colonial de Chuquinga de la provincia de Aymaraes.

```
<html>
    <head>
        <title>My first three.js app</title>
        <style>
            body { margin: 0; }
            canvas { width: 100%; height: 100% }
        </style>
    </head>
    <body>
        <script src="js/three.js"></script>
        <script>
            var scene = new THREE.Scene();
            var camera = new THREE.PerspectiveCamera( 75, window.innerWidth/window.innerHeight, 0.1, 1000 );

            var renderer = new THREE.WebGLRenderer();
            renderer.setSize( window.innerWidth, window.innerHeight );
            document.body.appendChild( renderer.domElement );

            var geometry = new THREE.BoxGeometry( 1, 1, 1 );
            var material = new THREE.MeshBasicMaterial( { color: 0x00ff00 } );
            var cube = new THREE.Mesh( geometry, material );
            scene.add( cube );

            camera.position.z = 5;

            var animate = function () {
                requestAnimationFrame( animate );

                cube.rotation.x += 0.1;
                cube.rotation.y += 0.1;

                renderer.render(scene, camera);
            };

            animate();
        </script>
    </body>
```

FIGURA N° 53 : Estructura básica de la inicialización para renderizacion con WebGL en ThreeJS

A continuación se detallará las configuraciones que tendrá el sitio web móvil del modelado 3D del Templo Colonial de Chuquinga.

- **Configuración de la vista de cámara para modelado 3D**

  La cámara que se utilizó fue PerspectiveCamera el cual tiene cuatro parámetros las cuales son: El primero es el campo de visión, la cual es la extensión de la escena que se ve en la pantalla en cualquier momento al iniciar el aplicativo en el móvil, éste parámetro está en grados. El segundo es el aspecto, es decir el tamaño del ancho y alto que tendrá al iniciar el aplicativo.

  El tercero y el curto es el recorte de cerca y lejos respectivamente, lo que significa que los objetos que se encuentren más lejos no representarán el valor de lejos o más cerca que cerca. En el escenario del templo colonial se utilizará el valor de 1 y 1000000, el cual nos da mejor rendimiento.

```
renderer = new THREE.WebGLRenderer( { antialias: true } );
renderer.setPixelRatio( window.devicePixelRatio );
renderer.setSize( window.innerWidth, window.innerHeight );
container.appendChild( renderer.domElement );

scene = new THREE.Scene();

camera = new THREE.PerspectiveCamera( 45, window.innerWidth / window.innerHeight, 1, 1000000 );
camera.position.x = -400;
camera.position.z = 4000;
camera.position.y = 300;
```

FIGURA N° 54 : Configuración de la Cámara.
Fuente: Elaboración Propia.

- **Renderizador**

  ThreeJS viene con varios renderizadores como son: Canvas Render, la cual diseña sus objetos 3D sin utilizar WebGL, éste renderizador está quedando obsoleto, otro de los renderizadores es WebGL Render, muestra

sus escenas usando WebGL la cual se utilizó para esta investigación del modelado 3D del Templo Colonial de Chuquinga.

```
nction init() {

  renderer = new THREE.WebGLRenderer( { antialias: true } );
  renderer.setPixelRatio( window.devicePixelRatio );
  renderer.setSize( window.innerWidth, window.innerHeight);//renderer.setSize( window.innerWidth/2,
  window.innerHeight/2,false );

  container.appendChild( renderer.domElement );

  scene = new THREE.Scene();
```

FIGURA N° 55 : Renderizador WebGLRenderer del aplicativo 3D del Templo Colonial de Chuquinga.
Fuente: Elaboración propia.

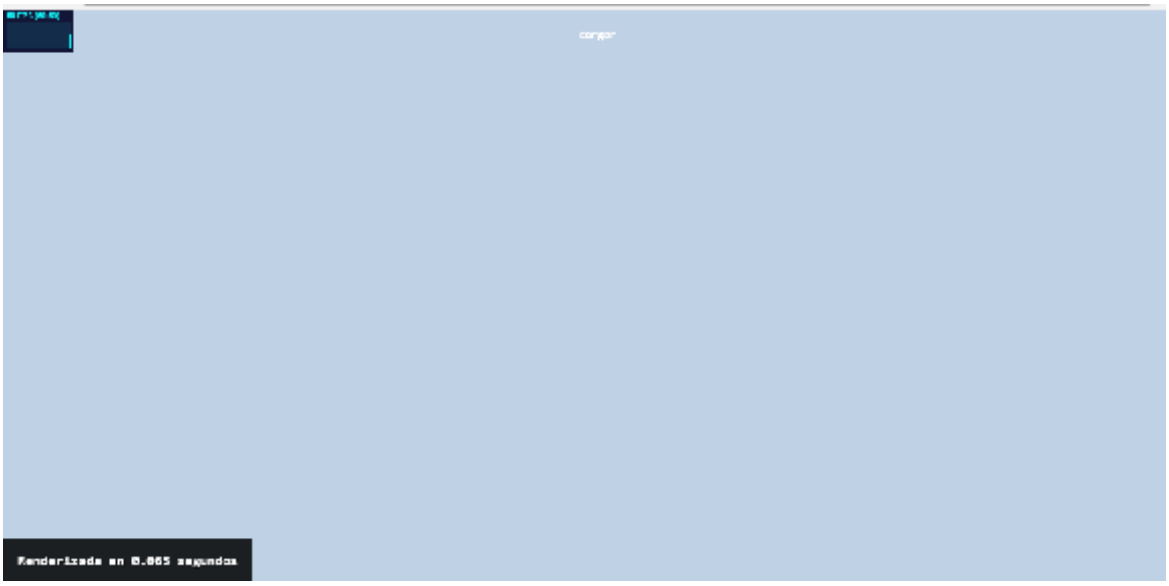


FIGURA N° 56 : Resultado de la configuración del escenario del modelado 3D del Templo Colonial de Chuquinga.
Fuente: Elaboración propia.

F) Modelando y texturizando de los elementos 3D del Templo Colonial de Chuquinga con el Framework ThreeJS

El modelado y texturizado de algunos elementos 3D del Templo Colonial de Chuquinga se realizó directamente en Framework ThreeJS, debido a la facilidad de algunos objetos, estos son muros que rodena al templo las cuales impiden el acceso al templo las cuales son: muro parte izquierda del templo, muro parte derecha, muro parte delantera y muros de la parte de

atrás. También se modelo directamente gradas y suelo externo que conforman el alrededor del Templo Colonial de Chuquinga otro de los objetos que se realizó directamente en el Framework fue el cielo y los cerros.

Para generar estos modelados se utilizó las transformaciones tridimensionales en el espacio o escenario 3D, las cuales son: Traslación, consiste en mover a cierta distancia los objetos, escala para la modificación del tamaño en los tres ejes, giros para la rotación de los objetos 3D.

- **Modelado del muro parte derecha del alrededor del Templo Colonial de Chuquinga**

  La parte derecha del contorno del Templo Colonial de Chuquinga fue creado en base a una matriz tridimensional, el cual permite controlar las tres transformaciones en las tres dimensione con exactitud e incluso a lo largo de uno o más ejes. El objeto se realizó con el método Shape de ThreeJS con una profundidad de 4000, números de segmento 5 y un grosor del bisel de 10, estos datos nos sirvieron junto con la matriz para poder configurar la data de extrusión y dar forma al objeto.

```
var Geometria=new THREE.Geometry();
var vertices=[[2,2,0],[50,2,0],[50,180,0],[2,180,0]];
```

FIGURA N° 57 : Matriz tridimensional del muro de la parte derecha del alrededor del Templo Colonial de Chuquinga.
Fuente: Elaboración propia.

```
var datos_extrusion={
                    amount:4000, //cantidad de profundidad izquieda
                    bevelEnabled:false, // activando bisel
                    bevelSegments:1, // segmentos del bisel
                    steps:5, // "profundidad y Núm. de segmentos que marcan la profundidad"
                    bevelThickness:10 // grosor del bisel
                    };
```

FIGURA N° 58 : Datos de extrusión del muro de la parte derecha del alrededor del Templo de Chuquinga.
Fuente: Elaboración propia.

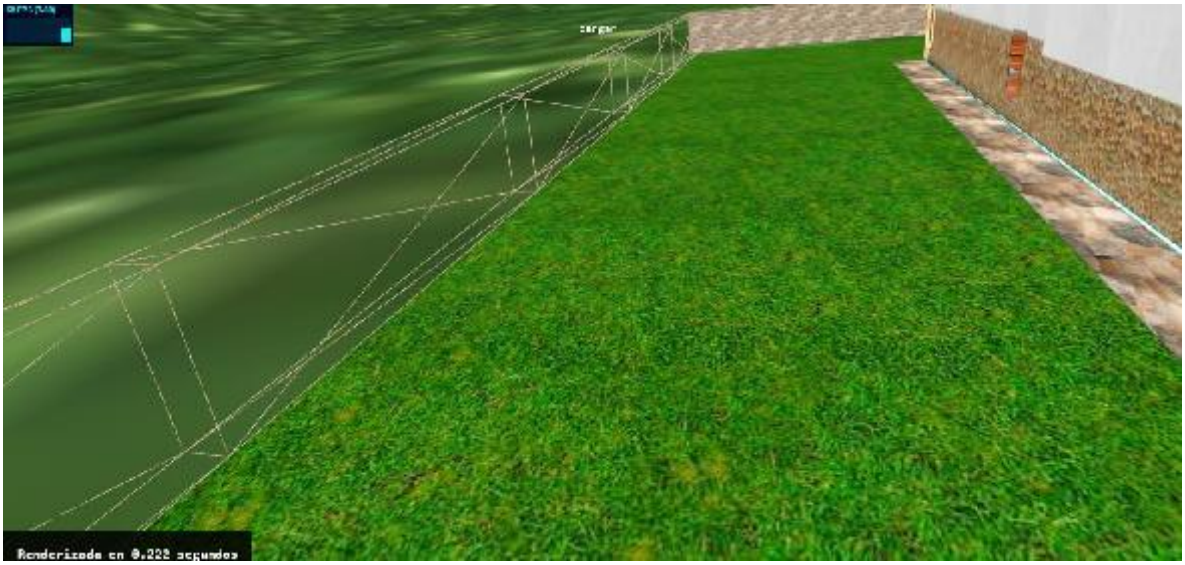


FIGURA N° 59 : Vista del muro derecho con los parámetros de extrusión.
Fuente: Elaboración propia.

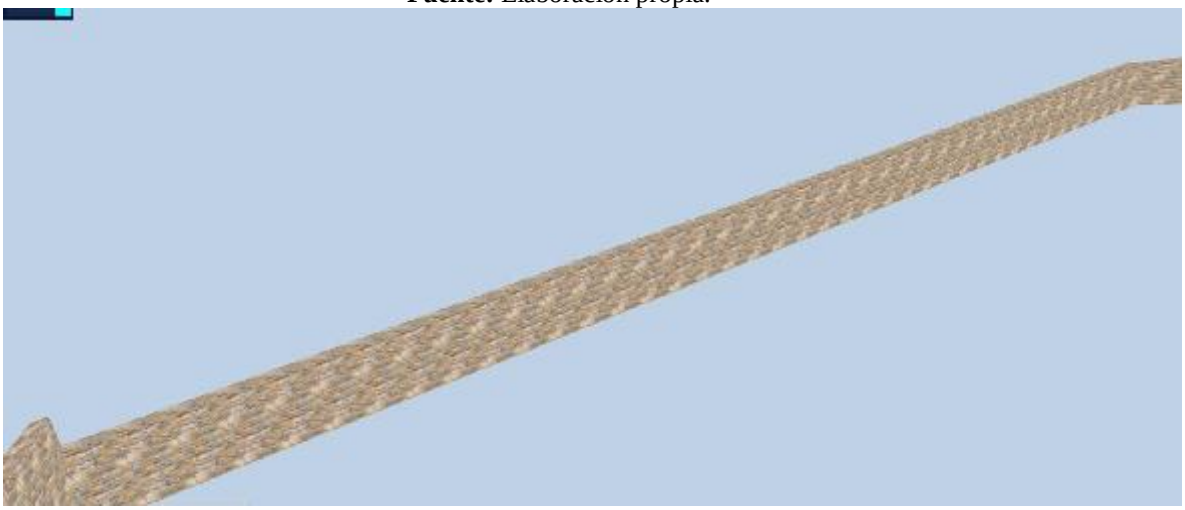


FIGURA N° 60 : Resultado final del muro derecho con su respectiva textura.
Fuente: Elaboración propia.

- **Modelado del muro parte izquierda del alrededor del Templo Colonial de Chuquinga**

  La parte izquierda del contorno del templo esta fue creado en base a una matriz tridimensional. El objeto se realizó con el método Shape de ThreeJS con una profundidad de 4000, números de segmento 5 y un grosor del bisel de 10, estos datos nos sirvieron junto con la matriz para poder configurar la data de extrusión y dar forma a los objetos.

```
var Geometria=new THREE.Geometry();
var vertices=[[2,2,0],[50,2,0],[50,180,0],[2,180,0]];
```

FIGURA N° 61 : Matriz tridimensional del muro de la parte izquierda del alrededor del Templo Colonial de Chuquinga.
Fuente: Elaboración propia.

```
var datos_extrusion={
                    amount:4000, //cantidad de profundidad izquieda
                    bevelEnabled:false, // activando bisel
                    bevelSegments:1, // segmentos del bisel
                    steps:6, // "profundidad y Núm. de segmentos que marcan la profundidad"
                    bevelThickness:10 // grosor del bisel
                    };
```

FIGURA N° 62 : Datos de extracción de la parte izquierda del alrededor del Templo Colonial de Chuquinga.
Fuente: Elaboración propia.

FIGURA N° 63 : Vista del muro izquierdo con los parámetros de extrusión.
Fuente: Elaboración propia.

FIGURA N° 64 : Resultado final del muro izquierdo con su respectiva textura.
Fuente: Elaboración propia.

- **Modelado del muro parte delantera del alrededor del Templo Colonial de Chuquinga**

  La parte delantera del contorno del templo fue creado en base a 2 matrices tridimensionales, el cual permitió controlar las tres transformaciones en las tres dimensione con exactitud e incluso a lo largo de uno o más ejes. El objeto se realizó con el método Shape de ThreeJS con una profundidad de 1300, números de segmento 5 y un grosor del bisel de 10, estos datos nos sirvieron junto con la matriz para poder configurar la data de extrusión y dar forma a los objetos.

```
var Geometria=new THREE.Geometry();
var vertices=[[2,2,0],[70,2,0],[70,35,0],[35,35,0],[35,250,0],[2,250,0]];
```

FIGURA N° 65 : Primera matriz tridimensional de muro delantero del alrededor del Templo Colonial de Chuquinga.
Fuente: Elaboración propia.

```
var Geometria=new THREE.Geometry();
var vertices=[[2,2,0],[70,2,0],[70,35,0],[35,35,0],[35,250,0],[2,250,0]];
```

FIGURA N° 66 : Segunda matriz tridimensional de muro delantero del alrededor del templo Colonial de Chuquinga
Fuente: Elaboración propia.

```
var datos_extrusion={
                    amount:1300, //cantidad de profundidad
                    bevelEnabled:false,
                    bevelSegments:1,
                    steps:5,
                    bevelThickness:100
                    };
```

FIGURA N° 67 : Datos de extracción de las dos partes delantera del Templo Colonial de Chuquinga.
Fuente: Elaboración propia.

FIGURA N° 68 : Vista de los dos muros delanteros con los parámetros de extrusión.
Fuente: Elaboración propia.

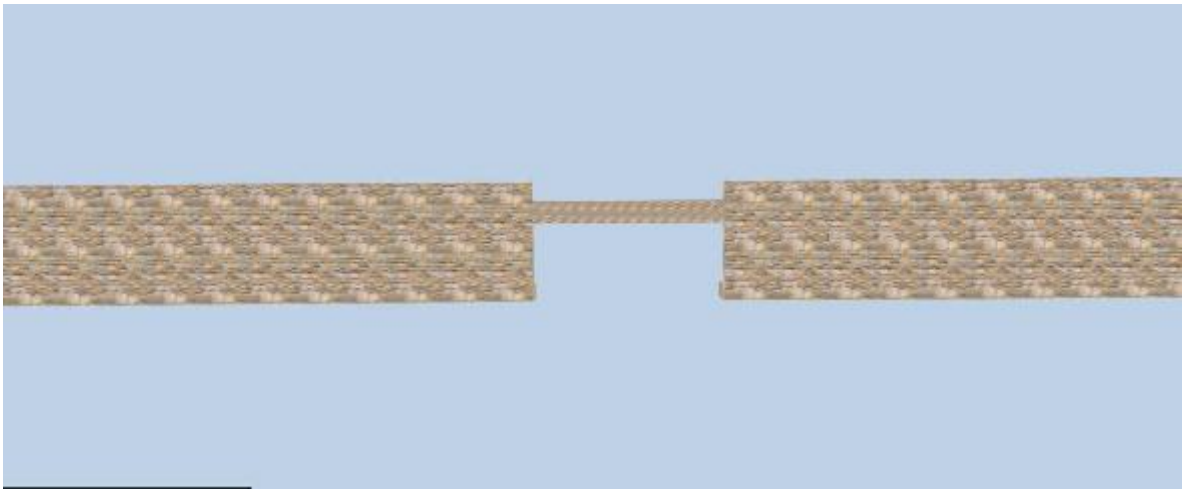


FIGURA N° 69 : Resultado final de los dos muros delanteros con su respectiva textura.
Fuente: Elaboración propia.

- **Gradas y el suelo externo que conforman el al rededor del Templo Colonial de Chuquinga**

  Las gradas y planos del Templo Colonial de Chuquinga, fueron modeladas en base a matrices tridimensional, el cual permitió controlar las tres transformaciones y la cual se tuvo que trabajarse con mucho detalle en la textura para no distorsionar las partes complejas que tiene el objeto 3D.

En cada una de estos elementos se trabajó con un determinado números de aristas y vértices para no sobrecargar el aplicativo del modelados 3D del Templo Colonial de Chuquinga.

A continuación se muestra el código la cual genera las gradas y niveles de un costado de la entrada del templo.

```
        var Geometria=new THREE.Geometry();
        var vertices=[[2,2,0],[70,2,0],[70,35,0],[35,35,0],[35,70,0],[2,70,0]];
        var long_vertices=vertices.length;
        var array_extrude=[];
        // xfxfxfxxdxdxdxfdxfdxfxxxdxdxdx
        for(i=0;i<long_vertices;i++){
                        x=vertices[i][0];
                        y=vertices[i][1];
                        z=vertices[i][2];
                        Vector=new THREE.Vector3(x,y,z);
                        Geometria.vertices.push(Vector);
                        array_extrude.push(Vector);
        }
        var forma_figura=new THREE.Shape(array_extrude);
        var datos_extrusion={
                                amount:500,
                                bevelEnabled:false,
                                bevelSegments:1,
                                steps:5,
                                bevelThickness:100
                                };
        var textura=THREE.ImageUtils.loadTexture('texturas/piedra/piedra.jpg' );
        var extrude_geometria=new THREE.ExtrudeGeometry(forma_figura,datos_extrusion);
        textura.repeat.set(0.02,0.02);
        textura.wrapS = textura.wrapT = THREE.repeatWrapping;
        var material = new
        THREE.MeshBasicMaterial({map:textura,side:THREE.DoubleSide,wireframe:false});
        var mallaextrusion=new THREE.Mesh(extrude_geometria,material);
        mallaextrusion.position.set(-1000,-19,1440);
scene.add(mallaextrusion);
```

FIGURA N° 70 : Gradas de la entrada al Templo Colonial de Chuquinga.
Fuente: Elaboración propia.

- **Cielo y cerros de la parte externa del Templo Colonial de Chuquinga**

  Para la creación del cielo se utilizó el formato de panorama, el cual se define como la proyección utilizada para mapear la escena 3D total o parcial. Hay varios tipos de proyecciones en ThreeJS las cuales son:

  - **Equirectangular**

    Es ampliamente utilizado, consiste en una sola imagen donde el ancho es exactamente dos veces a la altura.

  - **Cúbico**

    El formato cubico con la cual se trabajó en modelado 3D de Templo Colonial de Chuquinga usa 6 caras para llenar toda la esfera que rodela al templo. La imagen se asigna a las caras de los cubos que se ajustan perfectamente.

    A continuación se muestra el código fuente la cual genera el cielo utilizando la proyección cúbica.

```
var geometry = new THREE.CubeGeometry( 120000, 25000, 120000 );
var cubeMaterials =
                    [
new THREE.MeshBasicMaterial( { map: new THREE.TextureLoader( ).load( 'texturas/Skybox/front.jpg' ), side:
THREE.DoubleSide } ),
new THREE.MeshBasicMaterial( { map: new THREE.TextureLoader( ).load( 'texturas/Skybox/back.jpg' ), side:
THREE.DoubleSide } ),
new THREE.MeshBasicMaterial( { map: new THREE.TextureLoader( ).load( 'texturas/Skybox/up.jpg' ), side:
THREE.DoubleSide } ),
new THREE.MeshBasicMaterial( { map: new THREE.TextureLoader( ).load( 'texturas/Skybox/down.jpg' ), side:
THREE.DoubleSide } ),
new THREE.MeshBasicMaterial( { map: new THREE.TextureLoader( ).load( 'texturas/Skybox/right.jpg' ), side:
THREE.DoubleSide } ),
new THREE.MeshBasicMaterial( { map: new THREE.TextureLoader( ).load( 'texturas/Skybox/left.jpg' ), side:
THREE.DoubleSide } )
                    ];
var cubeMaterial = new THREE.MeshFaceMaterial( cubeMaterials );
var cube = new THREE.Mesh( geometry, cubeMaterial );
cube.position.set(-910,3940,1400);
scene.add( cube );
```

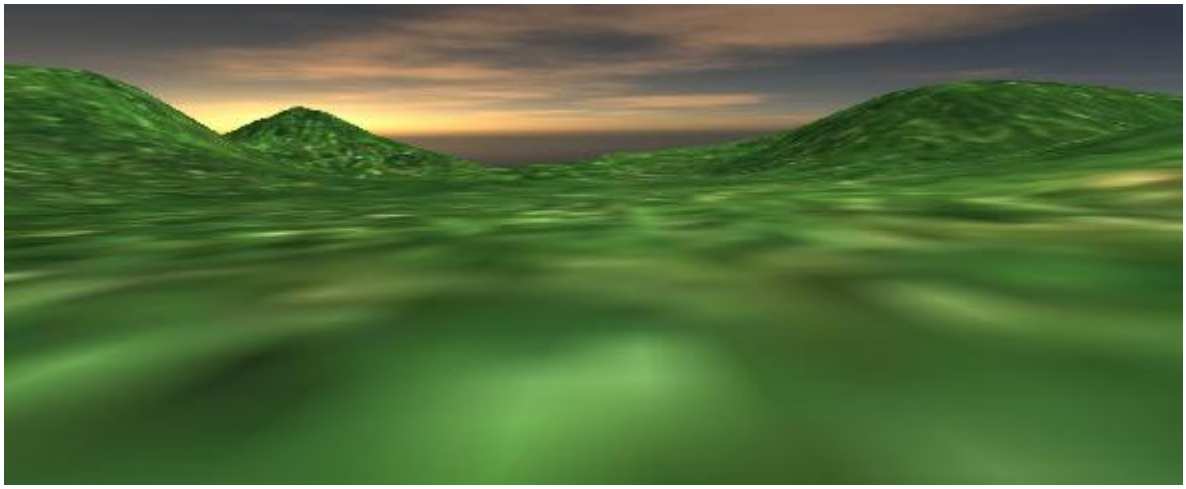


FIGURA N° 71 : Resultados de aplicar la proyección tipo cúbico para generar el ambiente.
Fuente: Elaboración propia.

G) Integración de todos los modelados 3D del Templo Colonial de Chuquinga con el Framework ThreeJS

El proceso de integración consiste en reunir todos objetos modelados en un mismo escenario utilizando el Framework ThreeJS para ello utilizaremos el complemento AssimpJSONLoade de ThreeJS con la cual cargaremos modelos creados en 3ds Max como son: muro parte derecha, muro parte izquierda, soporte del segundo piso, muro parte interna donde se están las esculturas y bancos del Templo Colonial de Chuquinga.

El complemento AssimpJSONLoade acepta archivos en formato JSON para cual se utilizó el exportador ThreeJSExport versión v0.8.

```
var loader=new THREE.JSONLoader();
                    modelo=function(object,color)
            {
                        asalogo=new THREE.Mesh
                                    (
                                      object,new
      THREE.MeshFaceMaterial(color)
                                    );
                         asalogo.position.set(0,-70,0);
                          scene.add(asalogo);
                    }
loader.load("templo.js",modelo);
```

FIGURA N° 72 : Exportador de 3ds Max.
Fuente: Elaboración propia

FIGURA N° 73 : Piso, pared interna y sillas exportadas en formato JSON e integradas a ThreeJS.
Fuente: Elaboración propia.

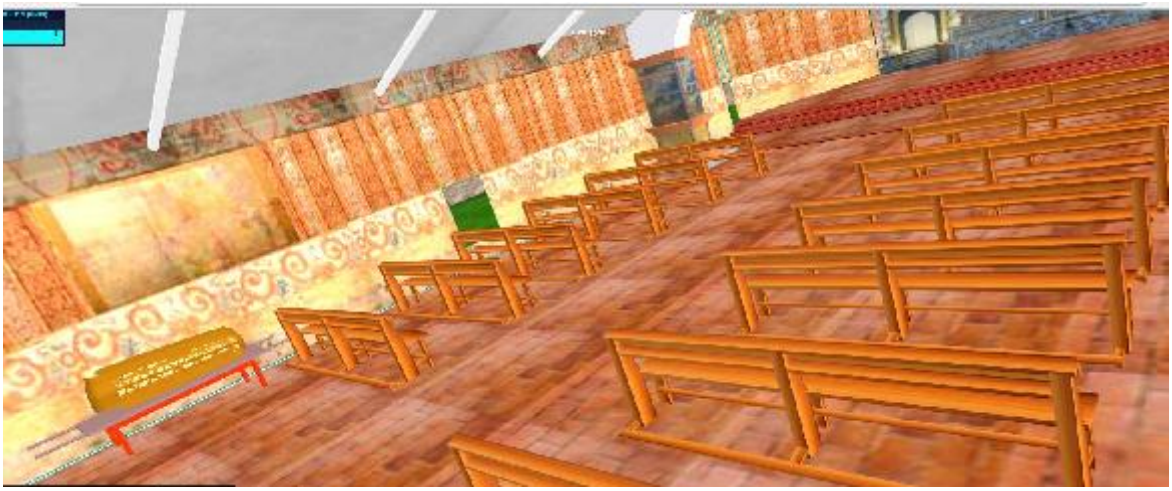


FIGURA N° 74 : Paredes y arcos exportados en formatos en JSON e integradas a ThreeJS.
Fuente: Elaboración propia.

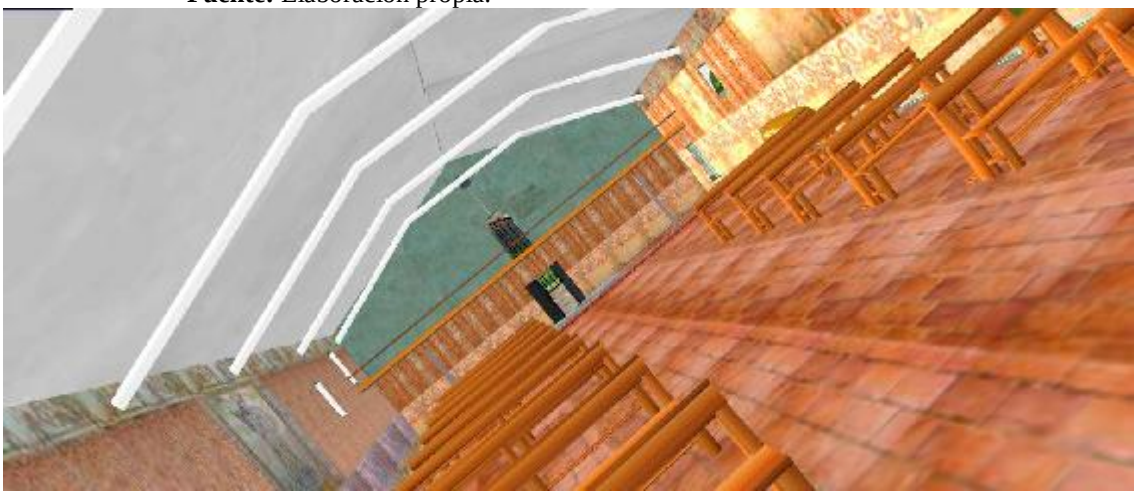


FIGURA N° 75 : Paredes delantera y soportes exportados en formatos en JSON e integradas a ThreeJS.
Fuente: Elaboración propia.

FIGURA N° 76 : Templo completo con modelos creados en 3ds max y ThreeJS.
Fuente: Elaboración propia.

FIGURA N° 77 : Primera capilla del Templo Colonial de Chuquinga con sus texturas.
Fuente: Elaboración propia.

FIGURA N° 78 : El bautisterio del Templo Colonial de Chuquinga con sus texturas.
Fuente: Elaboración propia.

FIGURA N° 79 : *Ventanas del Templo Colonial de Chuquinga.*
Fuente: Elaboración propia.

FIGURA N° 80 : Templo y escenario completo en ThreeJS.
Fuente: Elaboración propia.

CONCLUSIONES

Después de culminar con el trabajo de investigación, se llegaron a las siguientes conclusiones:

- Para el tiempo de renderización en dispositivos móviles del modelado 3D conforme al factor memoria interna, se observó que la memoria interna no supera la línea de referencia en el diagrama de Pareto, la cual nos indica que este factor no es significativo y en el análisis de varianza con tres factores se obtuvo el p-value=0.354, al ver que el valor de probabilidad es mayor al nivel de significancia (0.354> 0.05) se rechaza la hipótesis alterna y se acepta la hipótesis nula, afirmando que la memoria interna no influye significativamente en el tiempo de renderizado del modelo 3D del Templo Colonial de Chuquinga con ThreeJS según la prueba de hipótesis aplicada a un nivel de confiabilidad del 95% .
- Para el tiempo de renderización en dispositivos móviles del modelado 3D conforme al factor memoria RAM, se observó que la memoria RAM no supera la línea de referencia en el diagrama de Pareto, la cual nos indica que este factor no es significativo y en el análisis de varianza con tres factores se obtuvo el p-value=0.322, al ver que el valor de probabilidad es mayor al nivel de significancia (0.322> 0.05) se rechaza la hipótesis alterna y se acepta la hipótesis nula, afirmando que la memoria RAM no influye significativamente en el tiempo de renderizado del modelo 3D del Templo Colonial de Chuquinga con ThreeJS según la prueba de hipótesis aplicada a un nivel de confiabilidad del 95% .

- Para el tiempo de renderización en dispositivos móviles del modelado 3D conforme al factor GPU, existe un efecto principal en el factor GPU en la gráfica de efectos esto indica que cuando la frecuencia del reloj central es mayor a 500 Mhz, disminuye el tiempo de renderizado en los dispositivos móviles, también se ve que la GPU supera la línea de referencia en el diagrama de Pareto, la cual nos indica que este factor es significativo y en el análisis de varianza con tres factores se obtuvo el p-value=0.005, al ver que el valor de probabilidad es menor al nivel de significancia (0.005<0.05) se rechaza la hipótesis nula y se acepta la hipótesis alterna, afirmando que la GPU influye significativamente en el tiempo de renderizado del modelo 3D del Templo Colonial de Chuquinga con ThreeJS según la prueba de hipótesis aplicada a un nivel de confiabilidad del 95% .
- Para la frecuencia de imágenes (FPS) en dispositivos móviles del modelado 3D conforme al factor memoria interna, se observó que la memoria interna no supera la línea de referencia en el diagrama de Pareto, la cual nos indica que este factor no es significativo y en el análisis de varianza con tres factores se obtuvo el p-value=0.163, al ver que el valor de probabilidad es mayor al nivel de significancia (0.163> 0.05) se rechaza la hipótesis alterna y se acepta la hipótesis nula, afirmando que la memoria interna no influye significativamente en la frecuencia de imágenes (FPS) del modelo 3D del Templo Colonial de Chuquinga con ThreeJS según la prueba de hipótesis aplicada a un nivel de confiabilidad del 95%.

- Para la frecuencia de imágenes (FPS) en dispositivos móviles del modelado 3D conforme al factor memoria RAM, se observó que la memoria RAM no supera la línea de referencia en el diagrama de Pareto, la cual nos indica que este factor no es significativo y en el análisis de varianza con tres factores se obtuvo el p-value=0.091, al ver que el valor de probabilidad es mayor al nivel de significancia (0.091> 0.05) se rechaza la hipótesis alterna y se acepta la hipótesis nula, afirmando que la memoria RAM no influye significativamente en la frecuencia de imágenes (FPS) del modelo 3D del Templo Colonial de Chuquinga con ThreeJS según la prueba de hipótesis aplicada a un nivel de confiabilidad del 95%.
- Para la frecuencia de imágenes (FPS) en dispositivos móviles del modelado 3D conforme al factor GPU, existe un efecto principal en el factor GPU en la gráfica de efectos esto indica que cuando la frecuencia del reloj central es mayor a 500 Mhz, disminuye el tiempo de renderizado en los dispositivos móviles, también se ve que la GPU supera la línea de referencia en el diagrama de Pareto, la cual nos indica que este factor es significativo y en el análisis de varianza con tres factores se obtuvo el p-value=0.002, al ver que el valor de probabilidad es menor al nivel de significancia (0.002<0.05) se rechaza la hipótesis nula y se acepta la hipótesis alterna, afirmando que la GPU influye significativamente en la frecuencia de imágenes (FPS) del modelo 3D del Templo Colonial de Chuquinga con ThreeJS según la prueba de hipótesis aplicada a un nivel de confiabilidad del 95% .

RECOMENDACIONES

A continuación se plantea las recomendaciones para futuros trabajos que no pudieron ser implementadas en el trabajo de investigación.

- La textura utilizada en el aplicativo móvil VirtualTemp solo fue de calidad media. Por lo tanto se recomienda a futuros estudiantes trabajar con texturas de calidad alta para mejorar la apariencia en dispositivos móviles.
- El número de factores estudiados para esta investigación fue 3. Debido a esto un posible trabajo seria aumentar el número de factores como la CPU, núcleos del procesador, internet, etc. Lo cual aumentaría el numera de combinaciones y permitiría tener mejores resultados.
- El número de niveles para cada factor fue 2. Debido a esto un posible trabajo seria aumentar el número de niveles a básicos intermedios y óptimos, lo cual aumentaría número de tratamientos.
- Finalmente, un posible trabajo futuro podría ser la incorporación de controles y personajes para verificar el rendimiento del dispositivo con más elementos en el sitio web móvil.

BIBLIOGRAFÍA

- Alejandra, A. (2007). *Arquitectura y Urbanismo Colonial. Perú: vicens vives*, Perú s.a.c.
- Dirksen, J. (2015). *Learning Three.js: TheJavaScript 3D Library for WebGL* (2nd ed.). Madrid, España: Published by Packt Publishing Ltd.
- Dirksen, J. (2014). *Learning Three.js: TheJavaScript 3D Library for WebGL* (2nd ed.), Madrid, España: Published by Packt Publishing Ltd.
- Escobedo, N. (2013).*3D Studio Max en el proceso creativo de arquitectura* (Tesis de pregrado). Instituto Tecnológico de Durango. México.
- Gonzáles, C. (2006).*Diseño Gráfico 3D con software libre* (1st ed.). Madrid, España: Editorial Alfaomega.
- Gomez, A. (2010). *Reconstrucción 3D y realidad virtual* (6ta ed.), Madrid, España: Edición universidad de salaman.
- Guillermo, P. (2015).*Ingeniería de Software* (1st ed.). Madrid, España: Editorial Alfaomega.
- Hernández, S., & Fernández, C., & Baptista, L. (2014). *Metodología de la investigación* (6et ed.). Santa Fe, México: Interamericana Editores, S .A. DE C. V.
- Hostnig, R. (2006).*Tejas ornamentales del Templo de Chuquinga, Apurímac*. Aymaraes, Apurímac: El pino.
- Jiménez, R. (2010). *Correlación entre parámetros acústicos objetivos y características físicos arquitectónico en Templos católicos del periodo Colonial en las ciudades representativas del Perú (*Tesis de doctoral*)*. Universidad politécnica de Madrid, España.
- Jiménez, G., & Mendoza, F. (2014). *Animación y reconstrucción sobre la zona Prehispánica de Monte Albán*, Oaxaca (Tesis de pregrado). Universidad Nacional Autónoma de México, México.

- Kernighan, B., & Ritchie, D. (1991). *El lenguaje de programación C* (2nd ed.). Naucalpan de Juárez, México: Peason Educación.
- Luna, F. (2014). *Desarrollo Web para dispositivos móviles* (1st ed.). Buenos Aires, Argentina: Fox Andina.
- Matsuda, K., & Lea, R. (2013).*Programming Guide: Interactive 3D Graphics Programming With WebGl* (1st ed.). Míchigan, Estados Unidos: Person Education,Inc.
- Mamani, E., & Ibarra, M., & Ordoñez, E. (2012). *Propuesta de un enfoque para el desarrollo de software educativo intercultural.* Recuperado desde: https://www.academia.edu/29538468/Propuesta_de_un_enfoque_para_el_desarrollo_de_software_educativo_intercultural.
- Morillo, P. (2010). *Introducción a los dispositivos móviles* (6ta ed.). Madrid, España:alfaomega.
- Olivier, F. (2009).*AutoCAD 2009 Diseño, dibujo y presentación detallada todas las herramientas y funciones avanzadas*. Madrid, España: edición ENI.
- Powell, T. (2002). *Javascript: Manual de referencia*. Madrid, España: S.A mcgraw-hill - hill / Interamericana de España.
- Prado, A., & Lamas, N. (2016). *Fundamentos de la Informática* (1st ed.). San Fernando del Valle de Catamarca, Argentina: Editorial Científica Universitaria.
- Tosete, F. (2008).*Web Movil*, Madrid, España: Editorial Anuario ThinkEPI.

ANEXOS

Instrumento de recopilación de datos: Registro del tiempo de renderización con el aplicativo VirtualTemp en los dispositivos móviles.

TABLA N° 26 : *Registro del tiempo de renderizado en dispositivos móviles con el aplicativo VirtualTemp.*

N°	MARCA	MODELO	M. INTERNA	M. RAM	GPU	Frecuencia del reloj central (Mhz)	T. DE RENDERIZADO
1	Huawei	P8	16 GB	2 GB	Mali -T628 MP6	533	2.1 segundos
2	Bitel	B9503	8 GB	1 GB	ARM Mali-400 MP2	500	2.9 segundos
3	Huawei	P9 Lite	16 GB	2 GB	Mali-T860 MP2	700	1.50 segundos
4	Samsung	J1	4 GB	0.5 GB	ARM Mali-400 MP2	500	2.26 segundos
5	Lg	Nexus 5	32 GB	2 GB	Adreno 330	450	2.2 segundos
6	Lg	K8 2017	16 GB	1.5 GB	Mali- T720 MP2	600	2.3 segundos
7	Lg	LG XPOWE	16 GB	2 GB	Mali-T860 MP2	700	1.30 segundos
8	Samsung	S6 edge	32 GB	3 GB	Mali-T760	600	1.95 segundos
9	Samsung	j7 Neo 2017	16 GB	2 GB	Arm Mali-T830 MP2	700	1.5 segundos
10	Huawei	Y6 II	16 GB	2 GB	Arm Mali-450 MP4	650	1.882 segundos
11	Bitel	b8409	4 GB	0.5 GB	Adreno 304	400	1.78 segundos
12	Samsung	J1 mini	4 GB	1 GB	Mali-400MP2	500	2.8 segundos
13	Samsung	J3	4.64 GB	1.32 GB	Mali-400 MP2	500	2.122 segundos
14	Samsung	J1 mini	7.21 GB	1 GB	Mali-400MP2	500	2.597 segundos
15	Samsung	J7 Prime	10.95 GB	3 GB	Mali – T830	600	1.924 segundos
16	Samsung	s7 edge	32 GB	4 GB	Adreno 530	624	0.852 segundos
17	Bitel	b8409	4 GB	0.5 GB	No tiene		0.66 segundos
18	Xiaomi	Mi 5	3 GB	3 GB	Adreno 530	624	2.267 segundos
19	Huawei	P8 lite	16 GB	2 GB	Mali 450 MP4	650	1.354 segundos
20	Doogee	Valencia 2 y 100	16 GB	1 GB	Mali 450 MP4	650	1.573 segundos
21	htc	Htc desirc 530	8 GB	2 GB	Adreno 304	400	4.174 segundos
22	Samsung	Galaxy Ace 4 Neo	4 GB	0.5 GB	No tiene		4.161 segundos
23	Huawei	Y6	8 GB	2 GB	Adreno 304	400	2.152 segundos
24	Mobile	Ax675	4 GB	0.5 GB	Adreno 220	266	3.147 segundos
25	Huawei	P8	16 GB	2 GB	Mali-T628 MP4	533	1.891 segundos
26	Samsung	J1 Ace	8 GB	1 GB	Vivante GC7000UL	500	2.923 segundos
27	PowerVR	PowerVR	8 GB	1.5 GB	PowerVR G6430	450	1.3 segundos
28	Sony	Sony Xperia Z3	16 GB	1.5 GB	Adreno 330	550	0.9 segundos
29	Motorola	G5	8 GB	1.5 GB	Adreno 430	650	0.8 segundos
30	Sony	Sony Xperia Z5	8 GB	3 GB	Adreno 430	650	0.92 segundos

Instrumento de recopilación de datos: Registro de la frecuencia de imágenes con el aplicativo VirtualTemp en dispositivos móviles.

TABLA N° 27 : *Registro de la frecuencia de imágenes en dispositivos móviles con el aplicativo VitualTemp.*

N°	MARCA	MODELO	M. INTERNA	M. RAM	GPU	Frecuencia del reloj central (Mhz)	FPS
1	Huawei	P8	16 GB	2 GB	Mali -T628 MP6	533	58
2	Bitel	B9503	8 GB	1 GB	ARM Mali-400 MP2	500	30
3	Huawei	P9 Lite	16 GB	2 GB	Mali-T860 MP2	700	55
4	Samsung	J1	4 GB	0.5 GB	ARM Mali-400 MP2	500	35
5	Lg	Nexus 5	32 GB	2 GB	Adreno 330	450	50
6	Lg	K8 2017	16 GB	1.5 GB	Mali- T720 MP2	600	56
7	Lg	LG XPOWE	16 GB	2 GB	Mali-T860 MP2	700	47
8	Samsung	S6 edge	32 GB	3 GB	Mali-T760	600	59
9	Samsung	j7 Neo 2017	16 GB	2 GB	Arm Mali-T830 MP2	700	60
10	Huawei	Y6 II	16 GB	2 GB	Arm Mali-450 MP4	650	57
11	Bitel	b8409	4 GB	0.5 GB	Adreno 304	400	55
12	Samsung	J1 mini	4 GB	1 GB	Mali-400MP2	500	39
13	Samsung	J3	4.64 GB	1.32 GB	Mali-400 MP2	500	30
14	Samsung	J1 mini	8 GB	1 GB	Mali-400MP2	500	30
15	Samsung	J7 Prime	10.95 GB	3 GB	Mali – T830	600	55
16	Samsung	s7 edge	32 GB	4 GB	Adreno 530	624	60
17	Bitel	b8409	4 GB	0.5 GB	No tiene		25
18	Xiaomi	Mi 5	3 GB	3 GB	Adreno 530	624	55
19	Huawei	P8 lite	16 GB	2 GB	Mali 450 MP4	650	50
20	Doogee	Valencia 2 y 100	16 GB	1 GB	Mali 450 MP4	650	30
21	Htc	Htc desirc 530	8 GB	2 GB	Adreno 304	400	13
22	Samsung	Galaxy Ace 4 Neo	4 GB	0.5 GB	No tiene		25
23	Huawei	Y6	8 GB	2 GB	Adreno 304	400	50
24	Mobile	Ax675	4 GB	0.5 GB	Adreno 220	266	24
25	Huawei	P8	16 GB	2 GB	Mali-T628 MP4	533	60
26	Samsung	J1 Ace	8 GB	1 GB	Vivante GC7000UL	500	40
27	PowerVR	PowerVR	8 GB	1.5 GB	PowerVR G6430	450	30
28	Sony	Sony Xperia Z3	16 GB	1.5 GB	Adreno 330	550	24
29	Motorola	G5	8 GB	1.5 GB	Adreno 430	650	55
30	Sony	Sony Xperia Z5	8 GB	3 G GB	Adreno 430	650	58

Fuente: Elaboración propia

Parámetro de clasificación de dispositivos móviles

a) Memoria Interna

TABLA N° 28 : *Parámetro de clasificación según memoria interna*

Factor	Nivele	
	Básico	**Óptimo**
Memoria interna	16GB <	≥ 16GB

Fuente: Elaboración propia.

b) Memoria Ram

TABLA N° 29 : *Parámetro de clasificación según memoria RAM*

Factor	Nivele	
	Básico	**Óptimo**
Memoria RAM	2GB<	≥ 2GB

Fuente: Elaboración propia.

c) GPU

TABLA N° 30 : *Parámetro de clasificación según GPU*

Factor	Nivele	
	Básico	**Óptimo**
GPU	500 MHZ≤	>500 MHZ

Fuente: Elaboración propia.

Clasificación de teléfonos inteligentes, por rendimiento y especificación

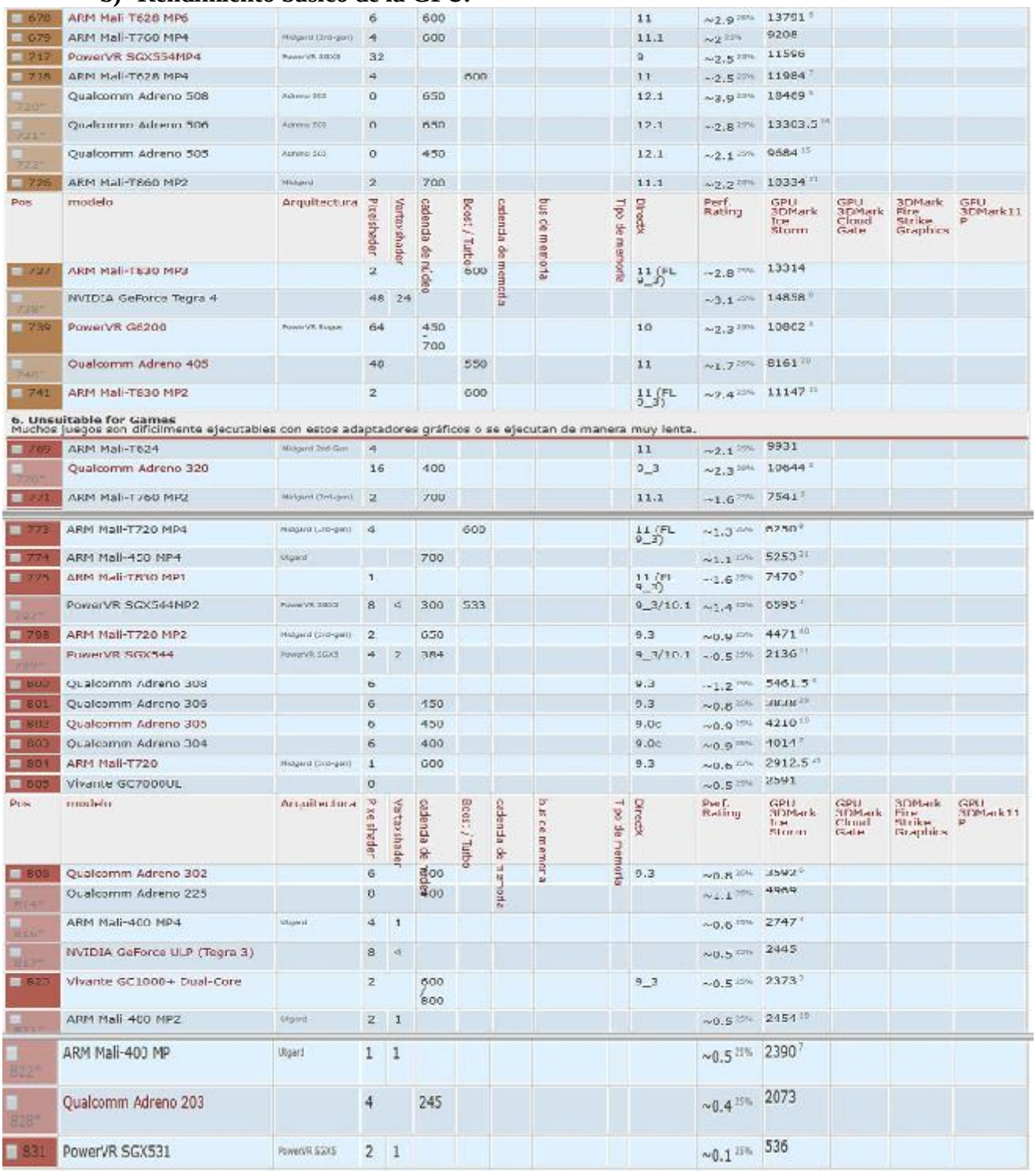
a) Rendimiento Óptimo de la GPU.

Pos	modelo	Arquitectura	Pixelshader	Vertexshader	cadencia de núcleo	Boost / Turbo	cadencia de memoria	bus de memoria	Tipo de memoria	DirectX	Perf. Rating	GPU 3DMark Ice Storm	GPU 3DMark Cloud Gate	3DMark Fire Strike Graphics	GPU 3DMark11 P
3. Low-Midrange Graphics Cards Estas tarjetas también deben ser capaces de aguantar todos los juegos actuales, pero la mayoría de ellos en configuraciones de detalles medios y bajos y con bajas resoluciones. Juegos más antiguos, o menos exigentes todavía pueden ser jugados con buena calidad de gráficos.															
254*	Apple A11 Bionic GPU	PowerVR Rogue	0								~23.8	112409			
254*	Apple A10X Fusion GPU / PowerVR	PowerVR Rogue	0								~23.4	110550			
283	Apple A9X / PowerVR Series 7XT	PowerVR Rogue	384							11_2	~10.7	50604			
312	NVIDIA Tegra X1 Maxwell GPU	Maxwell	256		1000		3200	64		11.2	~12.2	57697			
4. Barely Games-Capable Algunos juegos actuales no fastidiosos pueden ser jugados fluidamente con pequeños detalles.															
398	Apple A10 Fusion GPU / PowerVR	PowerVR Rogue	0								~13.4	63386			
415*	Qualcomm Adreno 540		0		710					12	~4.9	55725		518	786
416*	ARM Mali-G72 MP18	Bifrost	18		850						~10.1	47521.5			
417*	ARM Mali-G71 MP20	Bifrost	8			900					~7.7	36347			
418*	ARM Mali-G72 MP12	Bifrost	12		850						~7.2	33740			
419	ARM Mali-G71 MP8	Bifrost	8			900					~7.5	35418			
456	Qualcomm Adreno 530		0		624					11.1	~7	32824.5			
457	PowerVR GXA6850	PowerVR Rogue	256		450 (?)			128		10	~6.7	31542			
458	NVIDIA Tegra K1 Kepler GPU	Kepler	192		950					11	~7.9	37318.5			
459	Apple A9 / PowerVR GT7600	PowerVR Rogue	192							11_2	~9.2	43372.5			
460	ARM Mali-T880 MP12	Midgard (3rd-gen)	12			650				11.2 (FL 11_2)	~7	33031			
505	ARM Mali-T760 MP8	Midgard (3rd-gen)	8		700	772				11	~5.7	26964			
506*	ARM Mali-G71 MP2	Bifrost	2			770					~2.1	9862			
511	Qualcomm Adreno 430		192			650				11	~7.7	36316			
530	ARM Mali-T880 MP4	Midgard (4th-gen)	4			900				11.2 (FL 11_2)	~4.6	21577			
Pos	modelo	Arquitectura	Pixelshader	Vertexshader	cadencia de núcleo	Boost / Turbo	cadencia de memoria	bus de memoria	Tipo de memoria	DirectX	Perf. Rating	GPU 3DMark Ice Storm	GPU 3DMark Cloud Gate	3DMark Fire Strike Graphics	GPU 3DMark11 P
562	ARM Mali-T760 MP6	Midgard (3rd-gen)	6		700					11	~4.3	20244			
563	ARM Mali-T880 MP2	Midgard (4th-gen)	2			900				11.2 (FL 11_2)	~2.9	13708			
573	PowerVR GX6450	PowerVR Rogue	128		450 (?)			64		10	~5.1	23937			
574	Qualcomm Adreno 420		128			600				11	~4.4	20733			
594	Qualcomm Adreno 418		128			600				11.1	~4.9	23298.5			
5. Low End Graphics Cards Estos procesadores gráficos pueden mostrar únicamente juegos antiguos, fluidamente. Juegos actuales pueden ser presentados con detalles sustancialmente reducidos.															
596	Intel HD Graphics (Cherry Trail)	Intel Gen8	16		200	600		64/128		12 (FL 11_1)	1.5	19303	1783.5	225	298
672*	Qualcomm Adreno 510	Adreno 500	0							12.1	~4.6	21533			
673	Qualcomm Adreno 330		32		450	578				9_3	~3.8	18007			
674	PowerVR G6430	PowerVR Rogue	0							10	~3.9	18258			
675	PowerVR GX6250	PowerVR Rogue	64			700		64		10	~2.8	13446			
676	PowerVR G6400	PowerVR Rogue	0		400	533				10	~3.4	15933			
677	Intel HD Graphics (Bay Trail)	Ivy Bridge	4		311	896		32/64/128		11	1.1	14748.5	1228.5	149	185

FIGURA N° 81 : Listar de dispositivos móviles según rendimiento.

Fuente: Hinum, A.K., Hinum,S.,& Simon Leither,DI.J.(2018).Notebookcheck Publishing GmbH.Vienna.

b) Rendimiento básico de la GPU.



Pos	modelo	Arquitectura	Pixelshader	Vertexshader	cadencia de núcleo	Boost / Turbo	cadencia de memoria	bus de memoria	Tipo de memoria	Directx	Perf. Rating	GPU 3DMark Ice Storm	GPU 3DMark Cloud Gate	GPU 3DMark Fire Strike Graphics	GPU 3DMark11 P
678	ARM Mali T628 MP6		6		600					11	~2.9	13791			
679	ARM Mali-T760 MP4	Midgard (3rd-gen)	4		600					11.1	~2	9208			
717	PowerVR SGX554MP4	PowerVR SGX5	32							9	~2.5	11566			
718	ARM Mali-T628 MP4		4			600				11	~2.5	11984			
720*	Qualcomm Adreno 508	Adreno 5xx	0		650					12.1	~2.9	18469			
721*	Qualcomm Adreno 506	Adreno 5xx	0		650					12.1	~2.8	13303.5			
722*	Qualcomm Adreno 505	Adreno 5xx	0		450					12.1	~2.1	9684			
726	ARM Mali-T860 MP2	Midgard	2		700					11.1	~2.2	10334			
727	ARM Mali-T830 MP3		2			600				11 (FL 9_3)	~2.8	13314			
738*	NVIDIA GeForce Tegra 4		48	24							~3.1	14858			
739	PowerVR G6200	PowerVR Rogue	64		450 - 700					10	~2.3	10802			
740*	Qualcomm Adreno 405		40			550				11	~1.7	8161			
741	ARM Mali-T830 MP2		2			600				11 (FL 9_3)	~2.4	11147			
	6. Unsuitable for Games Muchos juegos son difícilmente ejecutables con estos adaptadores gráficos o se ejecutan de manera muy lenta.														
769	ARM Mali-T624	Midgard 2nd-Gen	4							11	~2.1	9931			
770*	Qualcomm Adreno 320		16		400					9_3	~2.3	10644			
771	ARM Mali-T760 MP2	Midgard (3rd-gen)	2		700					11.1	~1.6	7541			
773	ARM Mali-T720 MP4	Midgard (3rd-gen)	4			600				11 (FL 9_3)	~1.3	6250			
774	ARM Mali-450 MP4	Utgard			700						~1.1	5253			
775	ARM Mali-T830 MP1		1							11 (FL 9_3)	~1.6	7470			
797*	PowerVR SGX544MP2	PowerVR SGX5	8	4	300	533				9_3/10.1	~1.4	6595			
798	ARM Mali-T720 MP2	Midgard (3rd-gen)	2		650					9.3	~0.9	4471			
799*	PowerVR SGX544	PowerVR SGX5	4	2	384					9_3/10.1	~0.5	2136			
800	Qualcomm Adreno 308		6							9.3	~1.2	5461.5			
801	Qualcomm Adreno 306		6		450					9.3	~0.8	3808			
802	Qualcomm Adreno 305		6		450					9.0c	~0.9	4210			
803	Qualcomm Adreno 304		6		400					9.0c	~0.9	4014			
804	ARM Mali-T720	Midgard (3rd-gen)	1		600					9.3	~0.6	2912.5			
805	Vivante GC7000UL		0								~0.5	2591			
Pos	modelo	Arquitectura	Pixelshader	Vertexshader	cadencia de núcleo	Boost / Turbo	cadencia de memoria	bus de memoria	Tipo de memoria	Directx	Perf. Rating	GPU 3DMark Ice Storm	GPU 3DMark Cloud Gate	GPU 3DMark Fire Strike Graphics	GPU 3DMark11 P
806	Qualcomm Adreno 302		6		400					9.3	~0.8	3592			
814*	Qualcomm Adreno 225		0		400						~1.1	4884			
816*	ARM Mali-400 MP4	Utgard	4	1							~0.6	2747			
817*	NVIDIA GeForce ULP (Tegra 3)		8	4							~0.5	2445			
820	Vivante GC1000+ Dual-Core		2		600 / 800					9_3	~0.5	2373			
821*	ARM Mali-400 MP2	Utgard	2	1							~0.5	2454			
822*	ARM Mali-400 MP	Utgard	1	1							~0.5	2390			
828*	Qualcomm Adreno 203		4		245						~0.4	2073			
831	PowerVR SGX531	PowerVR SGX5	2	1							~0.1	536			

FIGURA N° 82 : Listar de dispositivos móviles según rendimiento.
Fuente: Hinum,A.K.,Hinum,S.,& Simon Leither,DI.J.(2018).Notebookcheck Publishing GmbH.Vienna.

Instrumento de recolección de datos: Registro del tiempo de renderizado en dispositivos móviles, clasificados según niveles de memoria interna, memoria RAM y GPU

TABLA N° 31 : *Registro del tiempo de renderizado clasificados según nivel*

N°	MARCA	MODELO	M. INTERNA	Nivel	M. RAM	Nivel	GPU	Velocidad de reloj del procesador gpu (Mhz)	Nivel	T. DE RENDERIZADO
1	Huawei	P8	16 GB	Óptimo	2 GB	Óptimo	Mali -T628 MP6	533	Óptimo	2.1 segundos
2	Bitel	B9503	8 GB	Básico	1 GB	Básico	ARM Mali-400 MP2	500	Básico	2.9 segundos
3	Huawei	P9 Lite	16 GB	Óptimo	2 GB	Óptimo	Mali-T860 MP2	700	Óptimo	1.50 segundos
4	Samsung	J1	4 GB	Básico	0.5 GB	Básico	ARM Mali-400 MP2	500	Básico	2.26 segundos
5	Lg	Nexus 5	32 GB	Óptimo	2 GB	Óptimo	Adreno 330	578	Óptimo	2.2 segundos
6	Lg	K8 2017	16 GB	Óptimo	1.5 GB	Básico	Mali- T720 MP2	600	Óptimo	2.3 segundos
7	Lg	LG XPOWE	16 GB	Óptimo	2 GB	Óptimo	Mali-T860 MP2	700	Óptimo	1.30 segundos
8	Samsung	S6 edge	32 GB	Óptimo	3 GB	Óptimo	Mali-T760	600	Óptimo	1.95 segundos
9	Samsung	j7 Neo 2017	16 GB	Óptimo	2 GB	Óptimo	Arm Mali-T830 MP2	700	Óptimo	1.5 segundos
10	Huawei	Y6 II	16 GB	Óptimo	2 GB	Óptimo	Arm Mali-450 MP4	650	Óptimo	1.882 segundos
11	Bitel	b8409	4 GB	Básico	0.5GB	Básico	Adreno 304	400	Básico	1.78 segundos
12	Samsung	J1 mini	4 GB	Básico	1 GB	Básico	Mali-400 MP2	500	Básico	2.8 segundos
13	Samsung	J3	4.64 GB	Básico	1.32 GB	Básico	Mali-400 MP2	500	Básico	2.122 segundos
14	Samsung	J1 mini	8 GB	Básico	1 GB	Básico	Mali-400MP2	500	Básico	2.597 segundos
15	Samsung	J7 Prime	10.95 GB	Básico	3 GB	Óptimo	Mali – T830	600	Óptimo	1.924 segundos
16	Samsung	s7 edge	32 GB	Óptimo	4 GB	Óptimo	Adreno 530	624	Óptimo	0.852 segundos
17	Bitel	b8409	4 GB	Básico	0.5 GB	Básico	No tiene		Básico	0.66 segundos
18	Xiaomi	Mi 5	3 GB	Básico	3 GB	Óptimo	Adreno 530	624	Óptimo	2.267 segundos
19	Huawei	P8 lite	16 GB	Óptimo	2 GB	Óptimo	Mali 450 MP4	650	Óptimo	1.354 Segundos
20	Doogee	Valencia 2 y 100	16 GB	Óptimo	1 GB	Básico	Mali 450 MP4	650	Óptimo	1.573 segundos
21	Htc	Htc desirc 530	8 GB	Básico	2 GB	Óptimo	Adreno 304	400	Básico	4.174 segundos
22	Samsung	Galaxy Ace 4 Neo	4 GB	Básico	0.5 GB	Básico	No tiene		Básico	4.161 segundos
23	Huawei	Y6	8 GB	Básico	2 GB	Óptimo	Adreno 304	400	Básico	2.152 segundos
24	Mobile	Ax675	4 GB	Básico	0.5 GB	Básico	Adreno 220	266	Básico	3.147 segundos
25	Huawei	P8	16 GB	Óptimo	2 GB	Óptimo	Mali-T628 MP4	533	Óptimo	1.891 segundos
26	Samsung	J1 Ace	8 GB	Básico	1 GB	Básico	Vivante GC7000UL	500	Básico	2.923 segundos
27	PowerVR	PowerVR	8 GB	Básico	1.5 GB	Básico	PowerVR G6430	450	Básico	1.3 segundos
28	Sony	Sony Xperia Z3	16 GB	Óptimo	1.5 GB	Básico	Adreno 330	550	Óptimo	0.9 segundos
29	Motorola	G5	8 GB	Básico	1.5 GB	Básico	Adreno 430	650	Óptimo	0.8 segundos
30	Sony	Sony Xperia Z5	8 GB	Básico	3G GB	Óptimo	Adreno 430	650	Óptimo	0.92 segundos

Fuente: Elaboración propia.

Instrumento de recolección de datos: Registro de frecuencia de imágenes (FPS) en dispositivos móviles, clasificados según niveles de memoria interna, memoria RAM y GPU.

TABLA N° 32 : *Registro de frecuencia de imágenes (FPS) clasificados según niveles*

N°	MARCA	MODELO	M. INTERNA	Nivel	M. RAM	Nivel	GPU	Velocidad de reloj del procesador gpu (Mhz)	Nivel	FPS
1	Huawei	P8	16 GB	Óptimo	2 GB	Óptimo	Mali -T628 MP6	533	Óptimo	58
2	Bitel	B9503	8 GB	Básico	1 GB	Básico	ARM Mali-400 MP2	500	Básico	30
3	Huawei	P9 Lite	16 GB	Óptimo	2 GB	Óptimo	Mali-T860 MP2	700	Óptimo	55
4	Samsung	J1	4 GB	Básico	0.5 GB	Básico	ARM Mali-400 MP2	500	Básico	35
5	Lg	Nexus 5	32 GB	Óptimo	2 GB	Óptimo	Adreno 330	578	Óptimo	50
6	Lg	K8 2017	16 GB	Óptimo	1.5 GB	Básico	Mali- T720 MP2	600	Óptimo	56
7	Lg	LG XPOWE	16 GB	Óptimo	2 GB	Óptimo	Mali-T860 MP2	700	Óptimo	47
8	Samsung	S6 edge	32 GB	Óptimo	3 GB	Óptimo	Mali-T760	600	Óptimo	59
9	Samsung	j7 Neo 2017	16 GB	Óptimo	2 GB	Óptimo	Arm Mali-T830 MP2	700	Óptimo	60
10	Huawei	Y6 II	16 GB	Óptimo	2 GB	Óptimo	Arm Mali-450 MP4	650	Óptimo	57
11	Bitel	b8409	4 GB	Básico	0.5 GB	Básico	Adreno 304	400	Básico	55
12	Samsung	J1 mini	4 GB	Básico	1 GB	Básico	Mali-400 MP2	500	Básico	39
13	Samsung	J3	4.64 GB	Básico	1.32 GB	Básico	Mali-400 MP2	500	Básico	30
14	Samsung	J1 mini	8 GB	Básico	1 GB	Básico	Mali-400 MP2	500	Básico	30
15	Samsung	J7 Prime	10.95 GB	Básico	3 GB	Óptimo	Mali – T830	600	Óptimo	55
16	Samsung	s7 edge	32 GB	Óptimo	4 GB	Óptimo	Adreno 530	624	Óptimo	60
17	Bitel	b8409	4 GB	Básico	0.5 GB	Básico	No tiene		Básico	25
18	Xiaomi	Mi 5	3 GB	Básico	3 GB	Óptimo	Adreno 530	624	Óptimo	55
19	Huawei	P8 lite	16 GB	Óptimo	2 GB	Óptimo	Mali 450 MP4	650	Óptimo	50
20	Doogee	Valencia 2 y 100	16 GB	Óptimo	1 GB	Básico	Mali 450 MP4	650	Óptimo	30
21	Htc	Htc desirc 530	8 GB	Básico	2 GB	Óptimo	Adreno 304	400	Básico	13
22	Samsung	Galaxy Ace 4 Neo	4 GB	Básico	0.5 GB	Básico	No tiene		Básico	25
23	Huawei	Y6	8 GB	Básico	2 GB	Óptimo	Adreno 304	400	Básico	50
24	Mobile	Ax675	4 GB	Básico	0.5 GB	Básico	Adreno 220	266	Básico	24
25	Huawei	P8	16 GB	Óptimo	2 GB	Óptimo	Mali-T628 MP4	533	Óptimo	60
26	Samsung	J1 Ace	8 GB	Básico	1 GB	Básico	Vivante GC7000UL	500	Básico	40
27	PowerVR	PowerVR	8 GB	Básico	1.5 GB	Básico	PowerVR G6430	450	Básico	30
28	Sony	Sony Xperia Z3	16 GB	Óptimo	1.5 GB	Básico	Adreno 330	550	Óptimo	24
29	Motorola	G5	8 GB	Básico	1.5 GB	Básico	Adreno 430	650	Óptimo	55
30	Sony	Sony Xperia Z5	8 GB	Básico	3GB	Óptimo	Adreno 430	650	Óptimo	58

Fuente: Elaboración propia.