

# Evaluación de formatos de exportación e importación de nubes de puntos para su uso en motores gráficos de realidad virtual

## Evaluation of point cloud export and import formats for their use in virtual reality graphic engines

J. A. Romero-Guerrero <sup>a,\*</sup>, C. Reta <sup>b</sup>, D. Arenas-Islas <sup>a</sup>, G. E. Bautista-Orduña <sup>c</sup>, I. Juárez-Sosa <sup>c</sup>  
J. M. Suarez-Luna <sup>a</sup>

<sup>a</sup> Departamento Manufactura Virtual y Lean y Cad Cae, CIATEQ A. C., 42163 San Agustín Tlaxiaca, Hidalgo, México.

<sup>b</sup> Departamento de Tecnologías de la Información, Electrónica y Control, CONAHCYT-CIATEQ A. C., 42163 San Agustín Tlaxiaca, Hidalgo, México.

<sup>c</sup> Departamento Manufactura Virtual y Lean y Cad Cae, CIATEQ A. C., 76246 El Marqués, Querétaro, México.

### Resumen

La progresión natural de la tecnología en la captura de información de entornos ha llevado a un aumento en el uso de nubes de puntos en diversos sectores, como patrimonio arquitectónico, topografía, educación, industria minera y manufactura. En este trabajo se presentan los distintos formatos para exportar e importar información de nubes de puntos, con el fin de identificar la opción más adecuada para utilizarla en motores gráficos, considerando tanto el tamaño final como la calidad de la información obtenida con equipos de escáner láser terrestre, lo que permitió determinar cuáles son las opciones de exportación e importación que se aprovechan mejor en motores gráficos. La información obtenida resulta sumamente valiosa para el desarrollo de futuros proyectos de ingeniería con Realidad Virtual.

**Palabras Clave:** nube de puntos, realidad virtual, escaneo láser, Unreal Engine.

### Abstract

The natural progression of technology in capturing information from environments has led to an increased use of point clouds in various sectors, such as architectural heritage, surveying, education, mining, and manufacturing. This paper presents different formats for exporting and importing point cloud information to identify the most appropriate option for using it in graphic engines, considering both the final size and the quality of the information obtained with terrestrial laser scanning equipment. This enable us to determine which export and import options are best suited for graphic engines. The information obtained is extremely valuable for the development of future engineering projects with Virtual Reality.

**Keywords:** point cloud, virtual reality, laser scanning, Unreal Engine.

## 1. Introducción

Una nube de puntos es una representación digital de la forma y geometría de un objeto, superficie o ambiente. Cada punto dentro del conjunto de datos tiene coordenadas cartesianas en un espacio tridimensional y, en el caso de las nubes de puntos a color, también incluyen información de valores de color RGB (IDEA 2019). El término nube se utiliza porque, al igual que las nubes en la atmósfera, los puntos pueden estar dispersos en el espacio de manera aleatoria o podrían formar patrones

definidos. No necesariamente están organizados en una sola estructura específica y pueden ser irregulares en su distribución.

Este trabajo se centra en el estudio de equipos y técnicas de adquisición y representación de nubes de puntos para su integración en entornos digitales con realidad virtual. Estas tecnologías pueden tener múltiples aplicaciones prácticas,

\*Autor para la correspondencia: [adan.romero@ciateq.mx](mailto:adan.romero@ciateq.mx)

**Correo electrónico:** [adan.romero@ciateq.mx](mailto:adan.romero@ciateq.mx) (Jorge Adan Romero-Guerrero), [carolina.reta@ciateq.mx](mailto:carolina.reta@ciateq.mx) (Carolina Reta), [david.arenas@ciateq.mx](mailto:david.arenas@ciateq.mx) (David Arenas-Islas), [guillermo.bautista@ciateq.mx](mailto:guillermo.bautista@ciateq.mx) (Guillermo Egberto Bautista-Orduña), [ivan.juarez@ciateq.mx](mailto:ivan.juarez@ciateq.mx) (Iván Juárez-Sosa), [johovani.suarez@ciateq.mx](mailto:johovani.suarez@ciateq.mx) (Johovani Misael Suarez-Luna).

como recorridos virtuales y capacitación en instalaciones industriales. Esto permite a los usuarios interactuar con modelos digitales precisos de ambientes y objetos, lo que puede mejorar significativamente la eficiencia y la seguridad en el lugar de trabajo.

### 1.1 Tecnología de escaneo láser terrestre

Los sistemas de escaneo láser terrestre funcionan mediante movimientos oscilantes y cuentan con un sistema de espejos para desviar el haz del láser desde el instrumento en ángulos variables. Las coordenadas cartesianas en el espacio tridimensional están definidas por los ángulos de ubicación de los espejos en el momento en que se emite el haz del láser. La luz reflejada es detectada por un fotosensor sensible y convertida en una señal eléctrica. Al analizar esta señal mediante uno de los diferentes métodos, se determina la distancia al objeto (Romero G. y Cuellar V., 2015).

En la Figura 1 se representan los ángulos de ubicación del equipo de escaneo láser. Una vez asignado un eje de referencia inercial fijo, es decir, cuando se coloca el escáner en una posición fija con coordenadas  $x = y = z = 0$ , los ángulos resultantes ( $\alpha, \beta$ ) son los grados de libertad del sistema en cada eje y representan la ubicación del punto de interés,  $P$ .

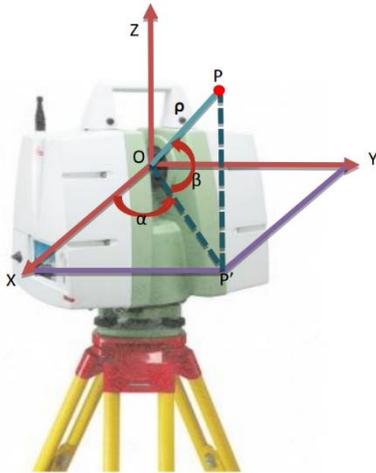


Figura 1: Asignación de un eje de referencia inercial fijo. Donde:  $\alpha$  representa el ángulo radial y  $\beta$  el ángulo axial del punto de interés y  $\rho$  es la distancia que existe entre el origen,  $O$  y el punto  $P$ .

Durante la captura de los puntos, se registran las coordenadas en forma polar. Posteriormente se realiza una conversión a coordenadas cartesianas mediante la ecuación (1). Es importante señalar que el proceso de conversión se lleva a cabo con la ayuda de un software especializado. Para los equipos utilizados fue el programa de Cyclone de Leica que permite la conversión precisa y rápida de las coordenadas de la nube de puntos para la representación del modelo en el espacio cartesiano.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \rho * \cos\alpha * \cos\beta \\ \rho * \sin\alpha * \cos\beta \\ \rho * \sin\beta \end{bmatrix} \quad (1)$$

#### 1.1.1 Tiempo de vuelo (Time-of-flight)

Para calcular la distancia ( $\rho$ ), el método de tiempo de vuelo multiplica la velocidad del pulso de luz ( $c$ ) por el tiempo que tarda en viajar desde el escáner hasta el objeto y de regreso al escáner ( $t$ ), como se muestra en la ecuación (2) (Maar y Hans M, 2021).

Una vez que se emite un pulso de luz, se activa el fotosensor del equipo y se inicia un temporizador que espera medir la luz entrante, ver Figura 2. Si el pulso de luz detectado supera un umbral específico, se reconoce como una señal reflejada del objeto y se utiliza la información de tiempo para calcular la distancia (Reshetyuk, 2006).

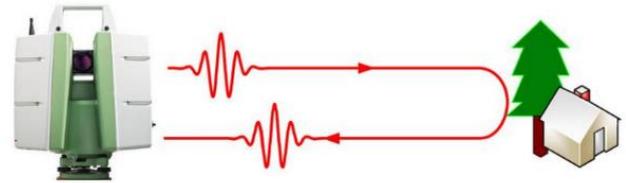


Figura 2: Principio de un escáner láser basado en el tiempo de vuelo.

$$\rho = \frac{1}{2} c * t \quad (2)$$

#### 1.1.2 Cambio de fase (Phase-Shift)

Los escáneres basados en cambio de fase utilizan un modulador que emite una señal periódica para contar con un espectro amplio en la potencia del láser, lo que la diferencia del método de tiempo de vuelo. Cuando esta señal sinusoidal moduladora se refleja en el objeto físico, se produce un desfase angular que se puede medir para determinar la distancia. Este ángulo de desfase, representado en la Figura 3, es una de las claves del funcionamiento de estos escáneres (Ramos et al., 2015).

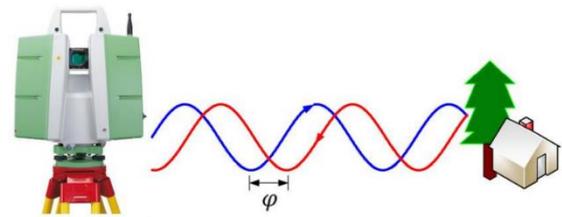


Figura 3: Principio de un escáner láser basado en cambio de fase.

La relación entre el desfase  $\varphi$  (radianes), la frecuencia modulada  $f_m$  (Hertz) y el tiempo de regreso  $t$  (segundos) está dada por la ecuación (3):

$$t = \frac{\varphi}{2\pi * f_m} \quad (3)$$

## 2. Metodología

### 2.1 Obtención de la nube de puntos

Se realizó un escaneo de la superficie exterior de tres edificios principales en un inmueble de 72.2 x 136 metros. Para esto, se empleó un escáner láser terrestre marca Leica modelo P20, el cual cuenta con las características mostradas en la Tabla 1.

Tabla 1: Información técnica de equipo utilizado

| Unidad Láser         |                                       |
|----------------------|---------------------------------------|
| Clasificación        | Láser clase 2                         |
| Longitud de onda     | 658 nm (visible) / 808 nm (invisible) |
| Rango                |                                       |
| Rango mínimo         | 0.4 metros                            |
| Rango máximo         | 120 metros                            |
| Adquisición de datos | Mas de 1,000,000 puntos por segundo   |

El escáner cuenta con diferentes configuraciones de escaneo, que se pueden ajustar según las necesidades del usuario y las características del ambiente a escanear. Estas configuraciones se detallan en la Tabla 2 y pueden variar según factores como la cantidad de información en el ambiente, si el escaneo es en un espacio abierto, si las referencias a escanear se encuentran a larga distancia, entre otros (Bitharis et al., 2022).

Tabla 2: Configuraciones escaneo escáner Leica P20

| Espaciado (mm a 10m) | Tiempo de escaneo y resolución (hh:mm:ss) |          |          |          |
|----------------------|---|----------|----------|----------|
|                      | Nivel de calidad                          |          |          |          |
|                      | 1   | 2        | 3        | 4        |
| 50                   | 00:20                                     | 00:20    | 00:28    | -        |
| 25                   | 00:33                                     | 00:33    | 00:53    | 01:43    |
| 12.5                 | 00:58                                     | 01:44    | 03:24    | 06:46    |
| 6.3                  | 01:49                                     | 03:25    | 06:46    | 13:30    |
| 3.1                  | 03:30                                     | 06:47    | 13:30    | 26:59:00 |
| 1.6                  | 13:33                                     | 27:04:00 | 54:07:00 | -        |
| 0.8                  | 54:07:00                                  | 01:48:13 | -        | -        |

En la Tabla 2, el espaciado se refiere a la densidad de la nube de puntos. Si seleccionamos el espaciado más grande, que es de 50 mm a 10 m, significa que, a una distancia de 10 metros desde el escáner, los puntos tendrán una separación entre ellos de 5 cm. En cambio, con una opción de espaciado menor, a 10 metros de distancia la separación entre ellos será de 0.8 mm.

Para este trabajo se seleccionó un espaciado de 6.3 mm a una distancia de 10 metros y un nivel de calidad nivel 3. Con estas configuraciones, el tiempo de escaneo por estación es de 6 minutos y 46 segundos, los resultados obtenidos se muestran en la Tabla 3 y la ubicación de los escaneos en la Figura 4.

Tabla 3: Cantidad de puntos obtenidos por estación

| No. Estación    | Cantidad de puntos | No. Estación | Cantidad de puntos |
|-----------------|--------------------|--------------|--------------------|
| Estación 1      | 3,641,326          | Estación 19  | 5,825,756          |
| Estación 2      | 4,098,822          | Estación 20  | 8,547,028          |
| Estación 3      | 4,138,016          | Estación 21  | 6,298,988          |
| Estación 4      | 5,826,699          | Estación 22  | 6,863,915          |
| Estación 5      | 8,838,433          | Estación 23  | 9,132,864          |
| Estación 6      | 8,636,700          | Estación 24  | 9,167,105          |
| Estación 7      | 7,589,947          | Estación 25  | 9,128,480          |
| Estación 8      | 6,351,499          | Estación 26  | 8,601,814          |
| Estación 9      | 5,495,135          | Estación 27  | 8,505,019          |
| Estación 10     | 8,772,853          | Estación 28  | 8,732,095          |
| Estación 11     | 15,551,095         | Estación 29  | 8,691,049          |
| Estación 12     | 14,893,910         | Estación 30  | 8,931,934          |
| Estación 13     | 14,912,210         | Estación 31  | 17,866,135         |
| Estación 14     | 15,583,383         | Estación 32  | 6,022,987          |
| Estación 15     | 16,138,740         | Estación 33  | 5,712,196          |
| Estación 16     | 19,244,081         | Estación 34  | 7,572,506          |
| Estación 17     | 5,977,258          | Estación 35  | 7,315,847          |
| Total de puntos |                    | 312,652,688  |                    |



Figura 4: Nube de puntos unificada de las 35 estaciones de escaneo.

## 2.2 Exportación de la nube de puntos utilizando distintos formatos

Posterior a la obtención de la nube de puntos, el siguiente paso consistió en exportar la información de la nube de puntos en los distintos formatos disponibles en el software de Cyclone y registrar su tamaño en KB, ver Tabla 4 (Heinrich-Wild-Strasse, 2018).

Tabla 4: Formatos exportados de Cyclone

| Formato | Tamaño en KB |
|---------|--------------|
| TXT     | 7,718,377    |
| LAS     | 7,895,526    |
| E57     | 8,842,450    |
| XYZ     | 9,641,467    |
| PTS     | 10,786,492   |
| SIM     | 16,009,465   |

La Tabla 4 muestra que el formato.txt es el más pequeño y el formato SIM el más grande. Sin embargo, es importante tener en cuenta que cada extensión cuenta con características de compresión e información distintas, por lo que no siempre el formato ideal para representar la nube de puntos es el que tiene menos peso en KB.

Para conocer la estructura de cada formato de la Tabla 4, se exporta una misma área de 1 x 1 metro a 10 metros de distancia del escaneo más cercano, conteniendo 3295 puntos en total. Se analiza la estructura de cada archivo exportado para determinar las diferencias en la información y características de compresión de cada formato, ver Figura 5.



Figura 5: Representación del metro cuadrado seleccionado para el estudio.

### 2.2.1 Formato TXT

El formato.txt exportado desde Cyclone incluye tres columnas que indican las coordenadas XYZ con una precisión de seis decimales como se observa en la Figura 6.

Sin embargo, en su mayoría, la nube de puntos tiene tonalidades de color verde que no se reflejan en este formato, lo que lo hace poco viable si queremos exportar una nube de puntos a color (Mosquera G, 2017).

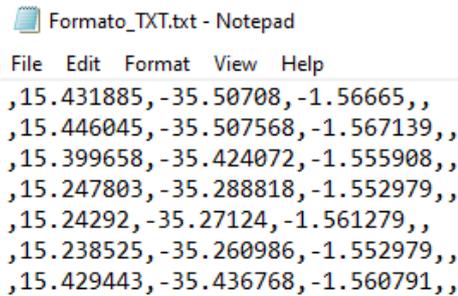


Figura 6: Representación en archivo de texto plano del formato TXT.

### 2.2.2 Formato LAS

El formato. Las (Lidar Application File) es un formato común para el almacenamiento y distribución de datos de nubes de puntos de LiDAR. Este formato es desarrollado y mantenido por la Asociación de Sistemas de Información Geoespacial (ASPRS) (Ramos, 2016).

A diferencia del formato.txt, el archivo en formato. Las utiliza una compresión binaria que dificulta la visualización de la información comprimida en un formato de texto legible. Sin embargo, existen visualizadores gráficos gratuitos para este formato, como la página <https://plas.io/>. Al importar el archivo en esta herramienta, es posible observar cómo se exporta la información y compararla con el software Cyclone, tal como se muestra en la Figura 7.

Se puede observar que la nube de puntos conserva el color del objeto escaneado y no se percibe una reducción en la cantidad de puntos con respecto al archivo original.

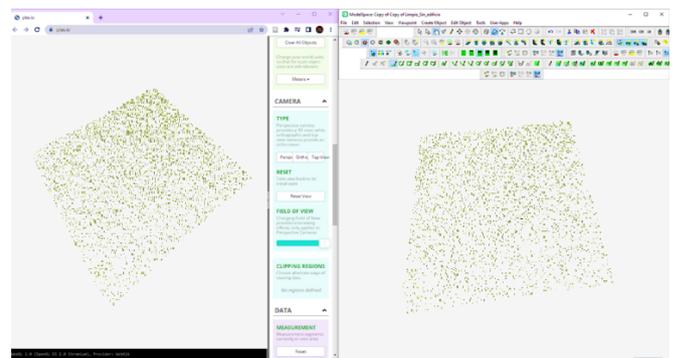


Figura 7: Comparación de nube de puntos. LAS vs nube de puntos en Cyclone.

### 2.2.3 Formato E57

El formato E57 utiliza una técnica de muestreo adaptativo para preservar la calidad de la nube de puntos mientras se reduce la cantidad de datos necesarios para almacenarla. La técnica de muestreo adaptativo es un enfoque utilizado para seleccionar de manera inteligente y eficiente una muestra representativa de datos a partir de un conjunto más grande. En lugar de tomar muestras uniformemente espaciadas, utiliza criterios específicos para elegir puntos estratégicos que mejor representen la característica o fenómeno que se está estudiando (Huber, 2011).

Para visualizar la información del archivo E57, se puede utilizar el sitio web Online Binary File Viewer de iamkate.com. La Figura 8 muestra detalles del encabezado obtenido.

```
<formatName type="String">![CDATA[ASTM E57 3D Imaging Data File]]</formatName>
<guid type="String">![CDATA[!{547E736D-0B74-4F89-8404-48C3387CE058}]]</guid>
<versionMajor type="Integer">1</versionMajor>
<versionMinor type="Integer"/>
<e57LibraryVersion type="String">![CDATA[unknown]]</e57LibraryVersion>
<coordinateMetadata type="String"/>
<creationDateTime type="Structure">
  <dateTimeValue type="Float">1.36595370895198965e+09</dateTimeValue>
  <isAtomicClockReferenced type="Integer"/>
</creationDateTime>
<data3D type="Vector" allowHeterogeneousChildren="1">
  <vectorChild type="Structure">
    <guid type="String">![CDATA[!{0000000000000001}]]</guid>
    <sensorSoftwareVersion type="String">![CDATA[!{Leica Geosystems}]]</sensorSoftwareVersion>
    <colorLimits type="Structure">
      <colorRedMaximum type="Integer">255</colorRedMaximum>
      <colorRedMinimum type="Integer"/>
      <colorGreenMaximum type="Integer">255</colorGreenMaximum>
      <colorGreenMinimum type="Integer"/>
      <colorBlueMaximum type="Integer">255</colorBlueMaximum>
      <colorBlueMinimum type="Integer"/>
    </colorLimits>
    <cartesianBounds type="Structure">
      <xMinimum type="Float">1.4000000000000000e+01</xMinimum>
      <xMaximum type="Float">1.7000000000000000e+01</xMaximum>
      <yMinimum type="Float">-3.6000000000000000e+01</yMinimum>
      <yMaximum type="Float">-3.3000000000000000e+01</yMaximum>
      <zMinimum type="Float">-2.0000000000000000e+00</zMinimum>
      <zMaximum type="Float">-1.0000000000000000e+00</zMaximum>
    </cartesianBounds>
    <pose type="Structure">
      <rotation type="Structure">
        <w type="Float">1.0000000000000000e+00</w>
        <x type="Float"/>
        <y type="Float"/>
        <z type="Float"/>
      </rotation>
      <translation type="Structure">
        <x type="Float"/>
        <y type="Float"/>
        <z type="Float"/>
      </translation>
    </pose>
    <points type="CompressedVector" fileOffset="48" recordCount="3295">
      <prototype type="Structure">
        <cartesianX type="ScaledInteger" minimum="-2147483647" maximum="2147483647" scale="1.1">
          <cartesianY type="ScaledInteger" minimum="-2147483647" maximum="2147483647" scale="1.1">
            <cartesianZ type="ScaledInteger" minimum="-2147483647" maximum="2147483647" scale="1.1">
              <colorRed type="Integer" minimum="0" maximum="255"/>
              <colorGreen type="Integer" minimum="0" maximum="255"/>
              <colorBlue type="Integer" minimum="0" maximum="255"/>
            </colorRed type="Integer" minimum="0" maximum="255"/>
          </colorGreen type="Integer" minimum="0" maximum="255"/>
        </colorBlue type="Integer" minimum="0" maximum="255"/>
      </prototype type="Structure">
    </points type="CompressedVector" fileOffset="48" recordCount="3295">
  </vectorChild type="Structure">
</data3D type="Vector" allowHeterogeneousChildren="1">
```

Figura 8: Encabezado de formato E57.

La Figura muestra detalles de la nube de puntos, incluyendo la cantidad total de puntos (en este caso, 3295), así como información sobre los rangos de color, el equipo utilizado para capturar la nube de puntos y la forma en que se interpretan las coordenadas cartesianas.

### 2.2.4 Formato XYZ

El formato.Xyz es uno de los formatos más simples y populares para almacenar nubes de puntos, consiste en una lista de coordenadas cartesianas x, y, z en formato de texto plano, donde cada línea representa un punto. Sin embargo, como se

muestra en la Figura 9 en el archivo exportado por Cyclone, que utiliza el formato.Xyz, no se incluye información sobre la topología o la conectividad de los puntos, ni la información de normales, intensidad o colores de la nube.

```
Formato_XYZ.xyz - Notepad
File Edit Format View Help
15.431885 -35.507080 -1.566650
15.446045 -35.507568 -1.567139
15.399658 -35.424072 -1.555908
15.247803 -35.288818 -1.552979
15.242920 -35.271240 -1.561279
15.238525 -35.260986 -1.552979
15.429443 -35.436768 -1.560791
15.359619 -35.406982 -1.552979
15.342041 -35.386963 -1.553955
15.349365 -35.392334 -1.553467
```

Figura 9: Estructura del formato XYZ.

### 2.2.5 Formato PTS

El formato.Pts es ampliamente utilizado en la industria para almacenar nubes de puntos. Similar al formato.Xyz, este formato consiste en una lista de coordenadas cartesianas x, y, z en formato de texto plano, donde cada línea representa un punto. Sin embargo, el formato. Pts va un paso más allá al permitir la inclusión de información adicional, como el color, la intensidad y la normal de cada punto (Ugarte Goicurúa, 2019).

En la Figura 10 se puede apreciar la estructura del formato. Pts exportado del software Cyclone. En el encabezado se muestra la cantidad de puntos presentes en la nube. A continuación, se describe por fila la información de cada punto en el siguiente orden: coordenadas X, Y, Z, normal y RGB. Esta organización de la información facilita la lectura y comprensión de los datos almacenados, por lo que el formato. Pts se considera de un formato de referencia para la nube de puntos.

```
Formato_PTS.pts - Notepad
File Edit Format View Help
3295
15.432 -35.507 -1.567 0 182 197 103
15.446 -35.508 -1.567 0 180 195 100
15.400 -35.424 -1.556 0 144 163 56
15.248 -35.289 -1.553 0 153 166 58
15.243 -35.271 -1.561 0 148 164 55
15.239 -35.261 -1.553 0 181 197 108
15.429 -35.437 -1.561 0 153 165 57
15.360 -35.407 -1.553 0 151 169 61
15.342 -35.387 -1.554 0 152 165 57
15.349 -35.392 -1.553 0 154 170 63
```

Figura 10: Estructura formato PTS.

### 2.2.6 Formato SIM

La estructura del archivo.sim es similar al formato.Pts, el primer campo "A00" es una identificación para indicar el comienzo del archivo. Luego, cada línea que comienza con "A01" contiene información sobre cada punto escaneado, que incluye su índice, coordenadas y, x, z en el sistema de referencia global sin embargo en este caso no hay coordenadas globales si no locales (en algunos casos puede ser útil para reconstruir la topología de la nube de puntos). Aunque puede

faltar información adicional, como color o normal, el formato. sim puede ser útil para algunos programas y aplicaciones específicas. El encabezado del archivo.sim se muestra en la Figura 11.

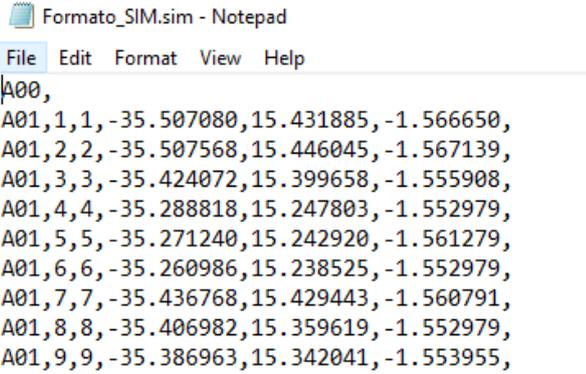


Figura 11: Estructura formato SIM.

### 2.3 Unión de estaciones de escaneo

Después de identificar que los formatos.Las, E57 y .Pts son los más completos para exportar la información, se procedió a unir las estaciones de escaneo. Existen varios algoritmos para realizar el registro de puntos, lo cual es necesario para afinar la alineación y unión de las nubes de puntos. El algoritmo que utiliza el software de Cyclone es Iterative Closest Point (ICP).

El algoritmo ICP se basa en minimizar la distancia media cuadrática (MSE) entre los puntos de dos nubes de puntos, como se define en la ecuación (4).

$$MSE = \frac{1}{N} \sum_{i=1}^N \|p_i - T(q_i)\|^2 \quad (4)$$

donde:

$p_i$  es el  $i$ -ésimo punto de la nube de puntos de referencia.

$q_i$  es el  $i$ -ésimo punto de la nube de puntos a registrar.

$T$  es una transformación rígida (rotación y traslación) que alinea la nube de puntos a registrar con la nube de puntos de referencia.

$N$  es el número total de puntos en las nubes de puntos.

El objetivo del algoritmo ICP es encontrar la transformación  $T$  que minimice el  $MSE$ . Para lograr esto, el algoritmo itera hasta que se alcanza un cierto criterio de convergencia, como una cantidad mínima de cambios en la transformación entre iteraciones (Xu et al., 2021).

### 2.4 Optimización de la nube de puntos

Posterior a la unión de la nube de puntos, se emplea un algoritmo de optimización para eliminar los puntos duplicados generados al momento de la unión. El algoritmo se basa en comparar la distancia entre cada par de puntos en la nube y eliminar aquellos que estén lo suficientemente cerca entre sí para ser considerados duplicados. La ecuación (5) presenta la métrica de distancia empleada en este trabajo (Sun et al., 2023).

$$Distancia = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \quad (5)$$

Donde  $(x_1, y_1, z_1)$  y  $(x_2, y_2, z_2)$  son las coordenadas de los dos puntos que se están comparando. Si la distancia entre dos puntos es menor que un umbral establecido, se consideran duplicados y se elimina uno de ellos. Este proceso se repite para cada par de puntos en la nube de puntos.

Con el propósito de este estudio, se eliminaron todos los puntos que tenían una distancia menor a 5mm.

Después de aplicar estas modificaciones, el número total de puntos se redujo a 52,094,616 como se observa en la Figura 12, lo que representa solo el 16% del número inicial de puntos que era de 312,652,688.

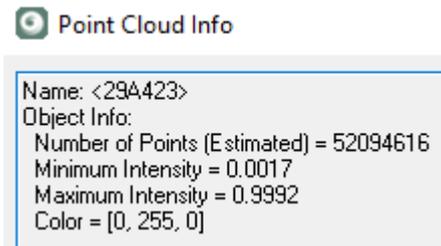


Figura 12: Información al momento de optimizar la nube en Cyclone.

Exportando la nube de puntos en los formatos.Las, E57 y .Pts los valores son los de la Tabla 5:

Tabla 5: Formatos exportados con nube optimizada

| Formato | Tamaño en KB |
|---------|--------------|
| E57     | 766,413      |
| LAS     | 1,322,716    |
| PTS     | 1,786,496    |

### 2.5 Importando la nube de puntos a un motor de juegos

Unreal Engine es un motor de juego desarrollado por Epic Games que se utiliza para crear videojuegos, aplicaciones de realidad virtual y aumentada, animaciones y visualizaciones arquitectónicas. Ofrece una gran ventaja, ya que cuenta con un plugin gratuito llamado "PointCloudPlugin" que es compatible con varios formatos de nube de puntos, incluyendo.Xyz, .Las, .Pts, lo que permite importar nubes de puntos de diferentes fuentes. Además, Unreal Engine es muy potente, con un sistema de renderizado avanzado y capacidad para trabajar en múltiples plataformas. Tiene un editor de niveles personalizable y fácil de usar, soporte para la creación de contenido en tiempo real, un lenguaje de programación visual llamado Blueprint y una gran comunidad de usuarios y desarrolladores que brindan soporte y recursos adicionales. Cabe destacar que para el desarrollo de este trabajo se utilizó la versión gratuita de Unreal Engine (Unreal Engine, 2023).

Para importar la nube de puntos, se creó un proyecto en tercera persona con los requisitos mínimos necesarios para poder mover al personaje alrededor de la nube de puntos mediante el teclado. En Unreal Engine, existen cinco tipos de calidad gráfica diferentes, y a medida que se aumenta la calidad gráfica, también aumentan los recursos requeridos por el sistema. En este caso, se eligió la calidad High, que sería considerada como la calidad media, esto se puede observar en la Figura 13.



Figura 13: Elección de calidad gráfica en proyecto.

Después de la creación del proyecto, se procedió a suprimir los objetos tridimensionales por defecto y se llevó a cabo el proceso de "cocinar el proyecto". Este proceso implica la recopilación de todos los activos del proyecto, como los modelos 3D, las texturas y los scripts, para compilarlos y optimizarlos de manera que se puedan utilizar de manera eficiente en el motor de juego. Durante el proceso de cocinado,

se generan versiones optimizadas de los archivos de contenido y se almacenan en un directorio específico del proyecto, lo que permite reducir el tiempo de carga y mejorar el rendimiento global del juego (Unreal Engine, 2021).

Posteriormente, después de optimizar el proyecto, se procedió a empaquetarlo, lo que implica la creación de un paquete ejecutable que contiene el proyecto completo y todos sus recursos. Este paquete es el archivo que se distribuye al usuario final, y en este caso es un archivo .exe para Windows. El paquete incluye todo lo necesario para que el usuario final pueda ejecutar el juego o la aplicación sin tener que instalar el motor de juego o descargar archivos adicionales, el tamaño del paquete creado alcanzó los 329 MB, ver Figura 14.

Este proceso se utilizará como punto de referencia para comparar el proyecto inicial con los elementos básicos, y posteriormente con los proyectos que contienen las tres variantes de nubes de puntos importadas, ver Figura 15.



Figura 14: Archivo ejecutable vacío que se usa de referencia.

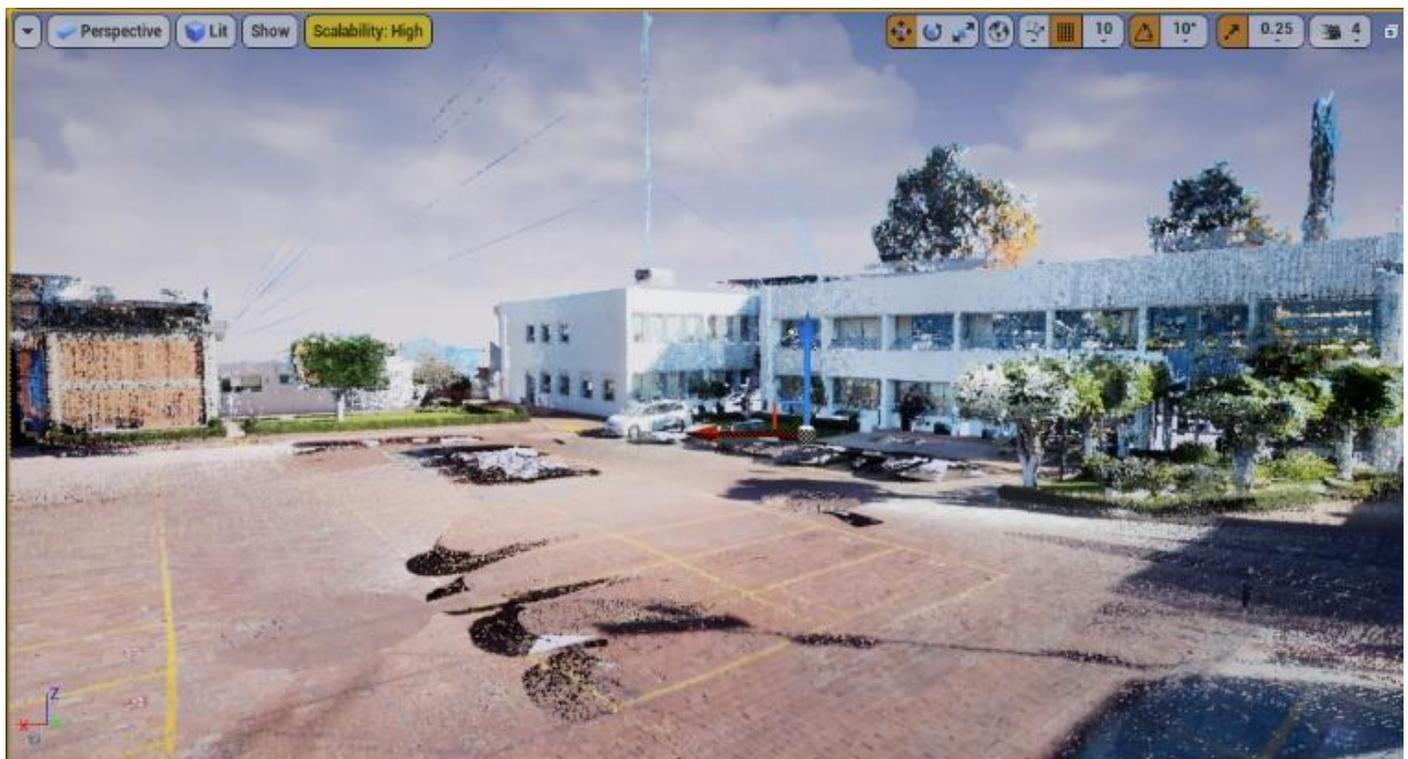


Figura 15: Nube de puntos formato E57 importada en Unreal Engine.

La herramienta gratuita y de código abierto "PointCloudPlugin" se utilizó para importar los tres formatos de nube de puntos en Unreal Engine. Al importar los archivos de nube de puntos, se creó un objeto de nube de puntos que contiene todos los puntos extraídos. Este objeto puede ser manipulado en el editor de Unreal Engine de la misma manera que cualquier otro objeto. En las Figuras 16 y 17 se puede observar el resultado de la importación. Los datos obtenidos se muestran en la Tabla 6.

Tabla 6: Información importada de nube de puntos en Unreal Engine

| Formato | Total de puntos | Tamaño de datos |
|---------|-----------------|-----------------|
| E57     | 52,094,651      | 996 MB          |
| LAS     | 52,093,724      | 996 MB          |
| PTS     | 52,093,215      | 996 MB          |

La variabilidad en la cantidad y tamaño de los puntos importados es prácticamente inexistente, esto sugiere que Unreal Engine lee los distintos formatos de manera similar y no almacena su estructura, sino solo la interpretación de las coordenadas y los colores. Cada nube de puntos se optimizó y se empaquetó de manera individual para comparar el tamaño final de cada una en el programa de Unreal Engine, los resultados se muestran en la Tabla 7.

Tabla 7: Comparativa de archivos ejecutables creados con Unreal Engine

|                 | Tamaño empaquetado | Diferencia de tamaño |
|-----------------|--------------------|----------------------|
| Modelo sin nube | 329 MB             | N/A                  |
| Modelo con E57  | 778 MB             | 449 MB               |
| Modelo con LAS  | 735 MB             | 406 MB               |
| Modelo con PTS  | 736 MB             | 407 MB               |

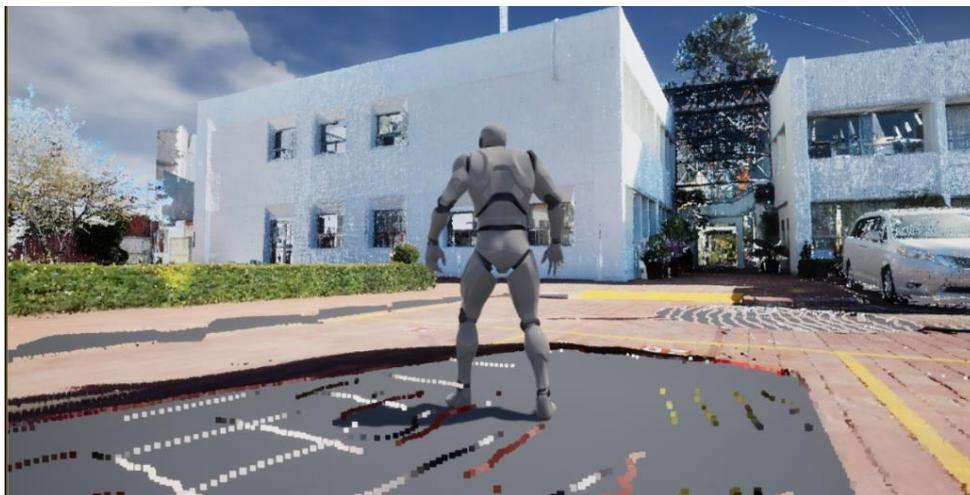
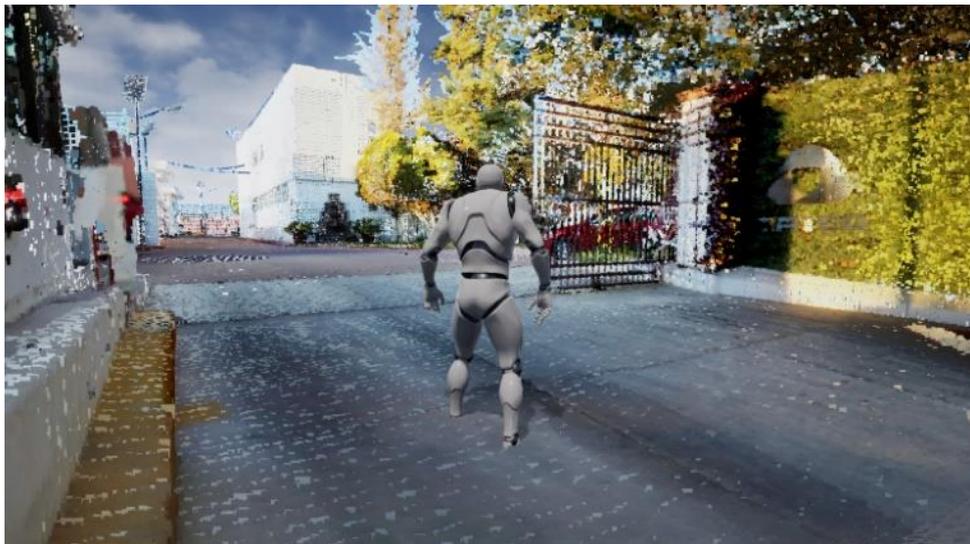


Figura 16 y 17: Archivo ejecutable mostrando a un personaje en tercera persona recorriendo la nube de puntos.

### 3. Conclusiones

En este trabajo se realizó investigación y se efectuó un experimento para determinar si la elección de los formatos de exportación e importación de nubes de puntos es determinante para garantizar su uso efectivo en motores gráficos de Realidad Virtual como Unreal Engine, a la vez que permita enriquecer la experiencia del usuario. En el experimento se consideraron factores importantes como: tamaño del archivo, el rendimiento del motor gráfico y la compresión de información, para elegir el formato adecuado de la nube de puntos. A continuación, se resumen las principales observaciones de la investigación:

- Entre los formatos utilizados, se encuentran los archivos .Pts, .E57 y .Las, los cuales contienen la información necesaria para importar y exportar la nube de puntos con información de color.
- Al exportar desde el software Cyclone, se encontró que el formato .E57 comprimió mejor la información, siendo un 57% y 42% más ligero que los formatos .Las y .Pts, respectivamente.
- Una vez importados a Unreal Engine, los tres formatos contenían la misma información y el programa los interpreta de manera equivalente, independientemente del formato en que introdujeron.
- Durante la creación de los proyectos ejecutables, se encontró que el formato .E57 resultó ser el más pesado en comparación con los formatos .Las y .Pts, siendo un 90.42% y un 90.64% más pequeños respectivamente. A pesar de esto, se pudo verificar que el desplazamiento dentro del proyecto fue de una manera fluida, no importando el formato, lo que permitió recorrer toda la nube de puntos sin problemas de rendimiento.

### Referencias

- Bitharis, S., Karolos, I. A., Tsioukas, V., Pikridas, C., Bantis, V., Gkamas, T., y Kontogiannis, S. (2022). Industrial area 3D geometric documentation using Terrestrial Laser Scanners. *FIG CONGRESS*.
- Heinrich-Wild-Strasse (2018). Leica Cyclone Basic Comprehensive laser scan software for field & office.
- Huber, D. (2011). The ASTM E57 file format for 3d imaging data exchange. En *Three-Dimensional Imaging, Interaction, and Measurement*, volumen 7864, pp. 88–96. SPIE.
- IDEA (2019). ¿Qué es una nube de puntos? IDEA Ingeniería. <https://ideaingenieria.es/nube-de-puntos/que-es-nube-de-puntos/>
- Maar, H. y Hans M., Z. (2021). WFD – Wave Form Digitizer Technology. <https://leica-geosystems.com/about-us/content-features/wave-form-digitizer-technology-white-paper>
- Mosquera G, B. C. (2017). Cartografía De Actividades Extractivas A Partir De Datos Lidar. Tesis de máster. Universidad Politécnica De Catalunya.
- Ramos, D. (2016). Cómo convertir ficheros LiDAR .LAZ a .Las – MappingGIS. [http://mappinggis.com/2016/08/como-convertir-ficheros-lidar-laz-a-las/#Como\\_convertir\\_de\\_LAZ\\_a\\_LAS\\_con\\_LAStools\\_en\\_QGIS](http://mappinggis.com/2016/08/como-convertir-ficheros-lidar-laz-a-las/#Como_convertir_de_LAZ_a_LAS_con_LAStools_en_QGIS)
- Ramos, L., Marchamalo, M., Rojas, J., y Martínez, R. (2015). Aplicación del Láser Escáner Terrestre (TLS) a la modelización de estructuras: precisión, exactitud y diseño de la adquisición de datos en casos reales. *Informes de la Construcción*, 67(538): e074. DOI: <http://dx.doi.org/10.3989/ic.13.103>
- Reshetyuk, Y. (2006). Investigation and calibration of pulsed time-of-flight terrestrial laser scanners. Tesis doctoral, KTH.
- Romero G., J. Adan y Cuellar V., José Luis. (2015). Estudio comparativo entre las tecnologías de escaneo laser terrestre. *CIATEQ A.C.* <http://ciateq.repositorioinstitucional.mx/jspui/handle/1020/149>
- Sun, R., Zhang, E., Mu, D., Ji, S., Zhang, Z., Liu, H., y Fu, Z.(2023). Optimization of the 3D Point Cloud Registration Algorithm Based on FPFH Features. *Applied Sciences*, 13(5):3096. DOI: <https://doi.org/10.3390/app13053096>
- Ugarte Goicurúa, I. (2019). Combinación de nube de puntos proveniente de un drone fotogramétrico de ala fija con la generada por una estación Trimble SX10 (Escáner+ Estación total+ Estación fotogramétrica). <https://gredos.usal.es/handle/10366/140245>
- Unreal Engine (2021). Build Operations: Cook, Package, Deploy, and Run. <https://docs.unrealengine.com/4.27/enUS/SharingAndReleasing/Deployment/BuildOperations/>
- Unreal Engine (2023). Características <https://www.unrealengine.com/es-ES/features>
- Xu, G., Pang, Y., Bai, Z., Wang, Y., y Lu, Z. (2021). A Fast Point Clouds Registration Algorithm for Laser Scanners. *Applied Sciences*, 11(8):3426. DOI: <https://doi.org/10.3390/app11083426>