https://doi.org/10.22201/dgtic.30618096e.2025.3.3.115 **Vol. 3, Núm. 3.** julio-septiembre 2025, págs. 34 - 42

Carga dinámica en tiempo real de entornos tridimensionales para galerías virtuales 3D

Información del reporte:

Para citar este reporte técnico:

Cruz Lovera, T. M. (2025). Carga dinámica en tiempo real de entornos tridimensionales para Galerías Virtuales 3D. *Cuadernos Técnicos Universitarios de la DGTIC, 3* (3) páginas (34 - 42).

https://doi.org/10.22201/dgtic.30618096e.2025.3.3.115

Tayde Martín Cruz Lovera
Dirección General de Cómputo y de Tecnologías de Información y Comunicación
Universidad Nacional Autónoma de México
taydevr@comunidad.unam.mx
ORCID: 0009-0003-9519-8805

Resumen

Se documenta la solución implementada para la carga dinámica en tiempo real de entornos tridimensionales en galerías virtuales 3D, un servicio que permitirá crear exposiciones virtuales de obras digitales bidimensionales y tridimensionales.

Uno de los principales desafíos en el desarrollo de este tipo de sistemas es la optimización del rendimiento, ya que los entornos 3D suelen implicar un alto costo computacional y tiempos de carga prolongados, especialmente en plataformas web donde los recursos del usuario son muy variables. Para abordar este problema, se implementaron estrategias de optimización para mejorar los tiempos de carga. Además, se exploró el uso de almacenamiento estructurado para gestionar datos generados en tiempo de ejecución, lo que facilitó la reconstrucción de las galerías creadas y la transmisión de información entre el interactivo web y el servidor.

Palabras clave:

Unity, WebGL, WebGPU, JSON, gráficos 3D.

https://doi.org/10.22201/dgtic.30618096e.2025.3.3.115 **Vol. 3, Núm. 3.** julio-septiembre 2025, págs. 35 - 42

Abstract

The implemented solution for real-time dynamic loading of three-dimensional environments into 3D virtual galleries is documented. This service will allow the creation of virtual exhibitions of two- and three-dimensional digital artworks.

One of the main challenges in developing this type of system is performance optimization, as 3D environments typically involve high computational costs and long loading times, especially on web platforms where user resources are highly variable. To address this problem, optimization strategies were implemented to improve loading times. Additionally, the use of structured storage was explored to manage data generated at runtime, which facilitated the reconstruction of the created galleries and the transmission of information between the web interactive and the server.

Keywords:

Unity, WebGL, WebGPU, JSON, 3D graphics.

1. INTRODUCCIÓN

En 2022, el Coordinador del Programa de Posgrado en Artes y Diseño de la Universidad Nacional Autónoma de México se acercó a la Dirección General de Cómputo y de Tecnologías de Información y Comunicación con la necesidad de desarrollar una galería virtual para presentar los trabajos de titulación de cuatro estudiantes. Como respuesta, se diseñó un interactivo 3D navegable, que simulaba una galería con cuatro salas y permitía recorrerla en primera persona, interactuando con las obras.

El proyecto Galerías Virtuales 3D surgió de la necesidad de la comunidad universitaria de contar con un servicio gratuito para montar exposiciones digitales personalizadas, tanto de obras bidimensionales (fotografías, dibujos, pinturas y videos) como tridimensionales (esculturas y modelos). Aunque existen algunas soluciones en el mercado, éstas son de pago o tienen limitaciones en sus versiones gratuitas.

Desde el punto de vista de las tecnologías de información y comunicación, se abordó un problema relevante: la necesidad de desarrollar una experiencia tridimensional interactiva, personalizable y fluida en plataformas web, garantizando buen rendimiento incluso en equipos con recursos limitados. Los entornos virtuales en 3D con múltiples modelos, texturas y materiales presentan altos costos computacionales, lo que representa un desafío particular para navegadores web. Friston et al. (2017) señalan que una solución eficaz para la integración de recursos tridimensionales debe permitir la carga modular y la gestión eficiente de activos digitales en tiempo de ejecución, a fin de facilitar entornos interactivos y escalables en motores como Unity.

Para este proyecto, se eligieron tecnologías modernas y ampliamente utilizadas: Unity para la construcción del entorno 3D interactivo, React para la interfaz web, y un servidor construido en Node.js y Express con base de datos en SQLite. Unity destaca por su capacidad de exportación a WebGL, lo que facilita su ejecución en navegadores sin necesidad de instalar software adicional. Además, estudios como el de Zheng et al. (2023) destacan que motores como Unity ofrecen mayores capacidades de interacción y manejo de iluminación y física avanzada, por lo que superan a implementaciones directas en WebGL en términos de complejidad visual y experiencia de usuario.

https://doi.org/10.22201/dgtic.30618096e.2025.3.3.115 **Vol. 3, Núm. 3.** julio-septiembre 2025, págs. 36 - 42

El objetivo de este reporte técnico es documentar la solución técnica implementada para lograr la carga dinámica en tiempo real de entornos tridimensionales dentro del proyecto Galerías Virtuales 3D, para optimizar el rendimiento, la reducción de tiempos de carga y la implementación de un sistema flexible para la creación, almacenamiento y visualización de exposiciones digitales interactivas.

2. DESARROLLO TÉCNICO

El proyecto Galerías Virtuales 3D busca ofrecer una plataforma escalable y accesible para que cualquier miembro de la comunidad universitaria pueda crear y compartir su propia galería interactiva en web. Sin embargo, el desarrollo técnico enfrentó dos problemas principales.

Primero, reducir los tiempos de carga iniciales del interactivo, ya que los entornos tridimensionales contienen modelos complejos y texturas pesadas que afectan el rendimiento y aumentan el peso total del interactivo. Incluir todos los entornos disponibles dentro del interactivo implicaba una carga innecesaria de recursos, ya que cada usuario sólo requiere el entorno que haya seleccionado para su galería.

Segundo, diseñar un sistema eficiente para almacenar y recuperar dinámicamente las configuraciones de cada galería creada por los usuarios, con sus respectivas obras, fichas técnicas y disposición espacial, de forma que fuera fácil y rápido reconstruir las galerías creadas por los usuarios.

2.1 METODOLOGÍA

La metodología se estructuró en tres etapas: diseño, implementación y pruebas, lo que permitió ajustar decisiones técnicas de forma iterativa y progresiva.

2.2 DISEÑO

En esta etapa, se identificaron los principales requerimientos del sistema y se definieron las tecnologías para su implementación. Se consideraron los siguientes aspectos:

Se estableció un esquema de comunicación entre el interactivo y el servidor, mediante peticiones HTTP, para permitir la transferencia de datos de manera eficiente.

Se decidió emplear formatos ligeros para web, paquetes de recursos *(AssetBundles)* de Unity para la carga dinámica de los entornos, formato GLTF, ZIP (para GLTF con texturas y archivo binario) y GLB para los modelos, JPG y PNG para imágenes, MP3, WAV y OGG para audios, así como MP4 para videos.

Se decidió usar el formato JSON, un formato ligero optimizado para web, para almacenar de forma estructurada la información necesaria y poder reconstruir la galería al momento de abrirla.

Se definieron estrategias como la reducción de polígonos en modelos y precálculo de iluminación para mejorar el rendimiento.

- Uso de WebGPU y WebGL, aceleración gráfica en navegadores modernos para optimizar la experiencia.
- Las herramientas tecnológicas utilizadas se resumen a continuación en la Tabla 1.

https://doi.org/10.22201/dgtic.30618096e.2025.3.3.115 **Vol. 3, Núm. 3.** julio-septiembre 2025, págs. 37 - 42

Tabla 1

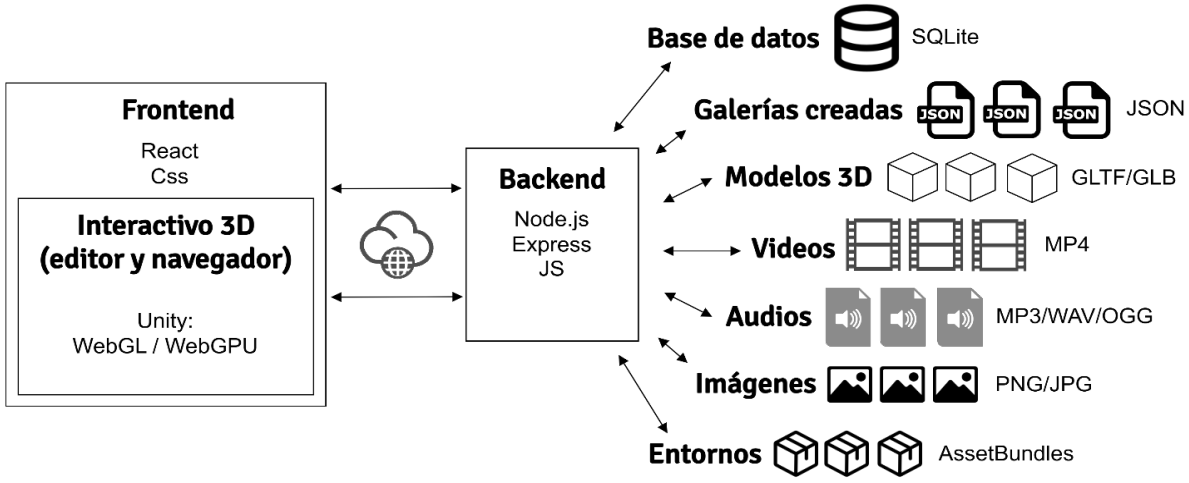
Tecnologías utilizadas en el desarrollo del sistema Galerías Virtuales 3D

Componente	Tecnología usada	Tecnología usada	Función principal
Motor 3D	*Unity*	*6.1* (6000.1.0f1)	*Desarrollo* del interactivo 3D y exportación WebGL y WebGPU
Interfaz web	*React*	*18.2.0*	*Plataforma* para incrustar el interactivo y gestionar el acceso
Servidor	*Node.js* + Express	*21.7.3* + 4.21.1	*Manejo* de peticiones HTTP, archivos y base de datos
Base de datos	*SQLite*	*5.1.7*	*Almacenamiento* de usuarios, galerías y obras

Como complemento a la Tabla 1, la Figura 1 ilustra la arquitectura del sistema Galerías Virtuales 3D, destacando cómo se integran los componentes tecnológicos descritos anteriormente.

Figura 1

Arquitectura de Galerías Virtuales 3D



2.3 IMPLEMENTACIÓN

La implementación del sistema se estructuró en tres componentes principales: el servidor, el interactivo 3D y la interfaz web. A continuación, se describe el proceso técnico seguido para su desarrollo e integración, las decisiones de arquitectura y los pasos necesarios para su reproducción.

- Escena base del interactivo en Unity: se utilizó Unity para construir la escena base del interactivo 3D, incluyendo únicamente los scripts de navegación, la interfaz gráfica y el sistema de carga dinámica.

https://doi.org/10.22201/dgtic.30618096e.2025.3.3.115 **Vol. 3, Núm. 3.** julio-septiembre 2025, págs. 38 - 42

No se incluyeron entornos ni modelos de obras en el paquete inicial, con el fin de mantenerlo ligero y optimizado para la web. Esta separación permite que los recursos pesados se carguen sólo cuando son necesarios y mejorar así el rendimiento desde el arranque.

- Exportación de entornos como paquetes de recursos (AssetBundles): para abordar el problema del almacenamiento de los distintos espacios o entornos disponibles, se optó por utilizar AssetBundles, una solución nativa de Unity que permite empaquetar modelos, texturas y otros recursos como archivos externos comprimidos, cargables en tiempo de ejecución. Cada entorno fue configurado como un objeto prefabricado independiente y exportado como un paquete de recursos. Esta estrategia no sólo reduce el peso del paquete base del interactivo, sino que garantiza que únicamente se descargue el entorno específico que el usuario selecciona, manteniendo su configuración original y todos sus componentes intactos. Este enfoque permite escalar fácilmente el número de entornos nuevos disponibles.
- Desarrollo del servidor: se implementó un servidor en Node.js con Express; este servidor expone puntos finales del API REST para:

  - Cargar y descargar los archivos JSON con los metadatos de cada galería.

  - Gestionar y servir los paquetes de recursos de entornos, así como imágenes, audios, videos y modelos 3D asociados a las obras.

  - Registrar y consultar datos desde una base de datos SQLite, incluyendo usuarios, galerías y obras.

- Creación de la interfaz web: la interfaz fue desarrollada en React, lo que permite a los usuarios:

  - Crear nuevas galerías a través de un formulario.

  - Visualizar la lista de las galerías existentes.

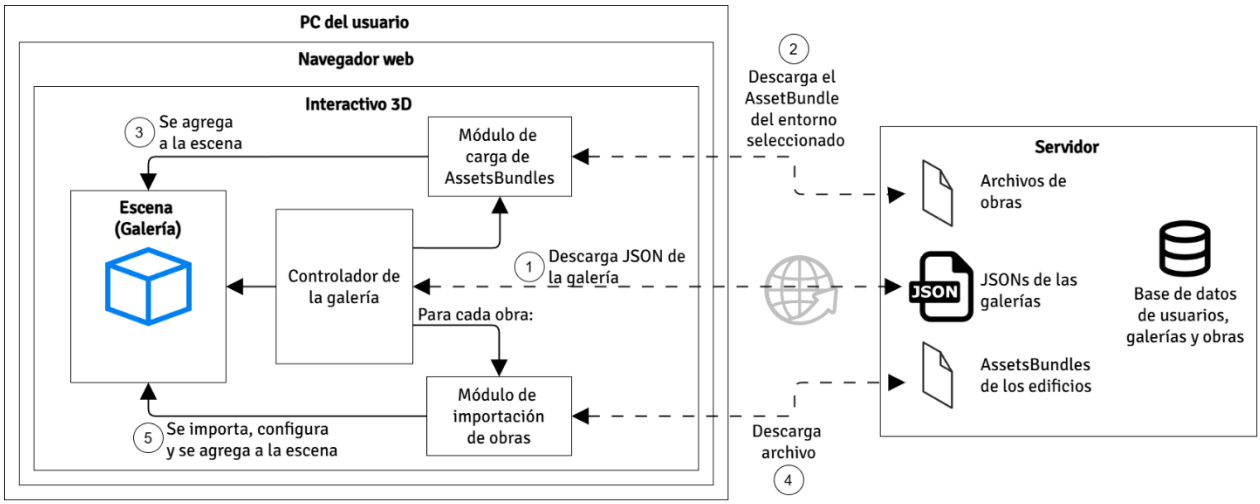
  - Acceder al interactivo en los diferentes modos (crear, editar o navegar) para una galería en específico.

Carga y deserialización en Unity: la reconstrucción de cada galería en tiempo real se logra mediante la descarga del archivo JSON desde el servidor y su deserialización en Unity. Este archivo contiene toda la información necesaria para ensamblar la escena: el entorno seleccionado, la lista de obras incluidas, sus posiciones, rotaciones, escalas, y fichas técnicas. A partir de esta información, el sistema descarga dinámicamente el paquete de recursos correspondiente al entorno seleccionado y lo integra en la escena; posteriormente descarga e instancia dinámicamente cada obra aplicando las propiedades definidas en el archivo JSON para su correcta visualización. Este proceso se ilustra en la Figura 2, que muestra la secuencia completa de ensamblado de una galería virtual en tiempo real.

https://doi.org/10.22201/dgtic.30618096e.2025.3.3.115 **Vol. 3, Núm. 3.** julio-septiembre 2025, págs. 39 - 42

Figura 2

Ensamblado de una galería en tiempo real



Nota. Los números de la figura corresponden a: 1) El interactivo 3D solicita al servidor el archivo JSON correspondiente; 2) El módulo de carga de paquetes de recursos descarga el entorno; 3) Se agrega el modelo del entorno correspondiente a la escena interactiva; 4) Para cada obra registrada en la galería, el interactivo 3D solicita su archivo correspondiente al servidor; 5) Una vez descargadas, las obras son configuradas según la información proporcionada en el JSON de la galería e incorporadas al entorno 3D, permitiendo su visualización dentro de la galería virtual.

2.4 ESTRUCTURA DEL ARCHIVO JSON

Para almacenar y recuperar las galerías de forma eficiente, se utiliza un archivo JSON por galería. Este formato ligero y optimizado para la web permite que cada galería se reconstruya incluso tras ser editada posteriormente. Su estructura incluye referencias clave: entorno seleccionado, obras incluidas, parámetros espaciales y metadatos.

La Figura 3 muestra un ejemplo de esta estructura jerárquica, similar al enfoque descrito por Friston et al. (2017), donde el uso de JSON permite reconstruir dinámicamente escenas 3D desde bases de datos estructuradas. Al dividir los modelos en componentes individuales y cargarlos de forma selectiva, se mejora notablemente el rendimiento, especialmente en entornos WebGL.

https://doi.org/10.22201/dgtic.30618096e.2025.3.3.115 **Vol. 3, Núm. 3.** julio-septiembre 2025, págs. 40 - 42

Figura 3

Estructura del archivo JSON para el almacenamiento de galerías con sus respectivas obras

```
{
  "fileName": "[nombre_archivo]",
  "id": "[identificador_único]",
  "name": "[nombre_galería]",
  "username": "[usuario_creador]",
  "enviromentId": [id_entorno_seleccionado],
  "description": "[descripción]",
  "startPosition": {
    "x": [posición_x],
    "y": [posición_y],
    "z": [posición_z]
  },
  "startRotation": {
    "x": [rotación_x],
    "y": [rotación_y],
    "z": [rotación_z],
    "w": [rotación_w]
  },
  "items": [
    {
      "id": "[identificador_único_item]",
      "name": "[nombre_item]",
      "type": "[tipo_item]",
      "position": {
        "x": [posición_x],
        "y": [posición_y],
        "z": [posición_z]
      },
      "rotation": {
        "x": [rotación_x],
        "y": [rotación_y],
        "z": [rotación_z],
        "w": [rotación_w]
      },
      "scale": {
        "x": [escala_x],
        "y": [escala_y],
        "z": [escala_z]
      },
      "metadata": {
        "source": "[fuente_dato]",
        "author": "[autor]",
        "date": "[fecha]"
      }
    }
  ]
}
```

https://doi.org/10.22201/dgtic.30618096e.2025.3.3.115 **Vol. 3, Núm. 3.** julio-septiembre 2025, págs. 41 - 42

2.5 PRUEBAS

Se realizaron pruebas funcionales y de rendimiento para validar la estabilidad y eficiencia del sistema:

- Verificación del funcionamiento en distintos navegadores.
- Validación del correcto almacenamiento y recuperación de las configuraciones de las galerías.
- Verificación de la correcta importación de modelos en el interactivo y su colocación.
- Validación de las interacciones en el interactivo.
- Evaluación de los cuadros por segundo durante la ejecución del interactivo.
- Evaluación de tiempos de carga con distintos tamaños de modelos y distintos entornos.

3. RESULTADOS

El enfoque modular permitió alcanzar mejoras tangibles en el rendimiento y flexibilidad del sistema. La carga dinámica de entornos mediante paquetes de recursos contribuyó directamente a la optimización del tamaño del interactivo y al rendimiento general, permitiendo cargar únicamente el entorno seleccionado por el usuario.

En pruebas realizadas durante el desarrollo, se comparó el peso del paquete completo, que incluía todos los entornos embebidos, contra una versión optimizada que excluye estos elementos y los distribuye como paquetes de recursos. La versión completa del interactivo alcanzaba tamaños superiores a 400 MB, mientras que el interactivo simplificado, que contiene sólo la lógica base e interfaz, se redujo a aproximadamente 80 MB, lo cual representa una disminución del 80% en el tamaño inicial de descarga. Esta diferencia se traduce directamente en menores tiempos de carga y mayor accesibilidad para los usuarios.

En cuanto al rendimiento en ejecución, se observó que, en equipos modernos con aceleración gráfica habilitada, especialmente aquellos con navegadores compatibles con WebGPU, el interactivo mantenía frecuencias de actualización superiores a los 60 FPS. Esta mejora se explica en parte porque "WebGPU proporciona un acceso más cercano al hardware moderno de la GPU [...] con mayor eficiencia y menor consumo de recursos del sistema" (Usta, 2024, p. 380), lo que permite aprovechar mejor la capacidad de los dispositivos actuales frente a WebGL. En contraste, en dispositivos con menor capacidad o navegadores con soporte limitado, la experiencia seguía siendo funcional, aunque con valores entre 25 y 45 FPS en escenas que incluían pocos modelos optimizados.

Sin embargo, se identificaron limitaciones cuando los usuarios cargan modelos 3D complejos sin compresión ni reducción de polígonos. Modelos de más de 10 MB o con geometría no optimizada provocaron caídas perceptibles en el rendimiento e incluso demoras en la carga. Esto resalta la importancia de contar en el futuro con un sistema de validación previa que alerte al usuario sobre el peso y complejidad de los modelos antes de integrarlos a una galería.

Por otro lado, se verificó la eficacia del esquema de ensamblado dinámico basado en archivos JSON, que permitió reconstruir correctamente las galerías en múltiples pruebas. El sistema logró esto utilizando los

https://doi.org/10.22201/dgtic.30618096e.2025.3.3.115 **Vol. 3, Núm. 3.** julio-septiembre 2025, págs. 42 - 42

archivos JSON generados, los cuales almacenan tanto la configuración del entorno como los datos de las obras, permitiendo su edición posterior sin pérdida de información o navegación.

En cuanto a la interoperabilidad, el interactivo se comportó de forma estable en navegadores modernos como Chrome, Edge y Firefox en Windows. No se identificaron errores críticos relacionados con la carga de recursos o la lectura de configuraciones, aunque falta continuar con pruebas en dispositivos móviles para validar la experiencia multiplataforma de forma completa.

4. CONCLUSIONES

La necesidad de reducir los tiempos de carga inicial y evitar paquetes pesados llevó a implementar una estrategia de carga dinámica en tiempo real para entornos tridimensionales en la plataforma de Galerías Virtuales 3D. Esta solución permitió excluir los entornos del núcleo del paquete base del interactivo y cargarlos según demanda del usuario, lo cual se tradujo en una experiencia más fluida. La solución permite escalar fácilmente la oferta de entornos virtuales disponibles sin impactar el rendimiento base del sistema.

La solución aplicada para tratar de optimizar los tiempos de carga en entornos interactivos, desarrollados en Unity para web, puede aplicarse en diversas situaciones para mejorar el rendimiento y la navegación fluida, así como el almacenamiento de datos en un archivo estructurado y ligero para almacenar información predefinida; no obstante, es aún más útil para almacenar información generada en tiempo de ejecución.

Un factor relevante es el número y características de las obras que agregue un usuario a su galería; queda pendiente agregar procesos que validen y limiten el peso, resolución de imágenes y número de vértices de los modelos 3D, con el objetivo de no sobrecargar la escena con imágenes o modelos tridimensionales muy pesados tanto en almacenamiento como en procesamiento gráfico.

REFERENCIAS

Friston, S., Fan, C., Doboš, J., Scully, T., & Steed, A. (2017). *3DRepo4Unity: Dynamic loading of version controlled 3D assets into the Unity game engine*. En *Proceedings of the 22nd International Conference on 3D Web Technology* (pp. 223–231). ACM. https://doi.org/10.1145/3055624.3075941

Usta, Z. (2024). WEBGPU: A NEW GRAPHIC API FOR 3D WEBGIS APPLICATIONS. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *48*(4/W9), 377–382. https://doi.org/10.5194/isprs-archives-XLVIII-4-W9-2024-377-2024

Zheng, Y., Merchant, A., Laninga, J., Xiang, Z. X., Alshaebi, K., Arellano, N., Romaniuk, H., Fai, S., & Sun, D. H. (2023). Comparison of characteristics of BIM visualization and interactive application based on WebGL and game engine. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLVIII-M-2-2023, 1671–1677. https://doi.org/10.5194/isprs-archives-XLVIII-M-2-2023-1671-2023