



<https://creativecommons.org/licenses/by/4.0/>

# AUTOMATIZACIÓN DEL PROCESO PARA LA GENERACIÓN DE MODELOS 3D A PARTIR DE UNA INTELIGENCIA ARTIFICIAL

## *Automation of the process for the generation of 3D models from artificial intelligence*

CAROLINA ANDREA GARCÍA SARMIENTO<sup>1</sup>, JUAN ESTEBAN INFANTE ANDRADE<sup>2</sup>,  
RICARDO ANDRÉS SANTA QUINTERO<sup>3</sup>

Recibido:30 de diciembre de 2023. Aceptado:17 de enero de 2024

DOI: <http://dx.doi.org/10.21017/rimci.2024.v11.n21.a158>

### RESUMEN

El objetivo del proyecto es describir los pasos para automatizar el proceso de generar modelos 3D, a partir del uso de la inteligencia artificial Point-E, del refinamiento con la aplicación Blender y finalmente la inserción de estos, en la aplicación Unity, donde se realizó un Showroom, esta descripción se realizó después de la investigación de documentación especializada y revisión bibliográfica, donde se encontró que dicha automatización sería posible, si se siguen los pasos para la generación de los Scripts necesarios, en Google Colab mediante el lenguaje de programación Python, dentro de Blender se realiza con el uso de la consola para realizar scripts con Python y en la aplicación Unity creando scripts en el lenguaje C#. Se concluye que es posible realizar la automatización del proceso descrito, además, al hacer uso de esta automatización se ahorra gran cantidad de tiempo en comparación con el proceso hecho manualmente.

**Palabras clave:** Automatización; lenguaje de programación; script; modelos 3D; inteligencia; artificial.

### ABSTRACT

The objective of the project is to describe the steps to automate the process of generating 3D models, from the use of artificial intelligence Point-E, the refinement with the Blender application and finally the insertion of these, in the Unity application, where a Showroom was made, this description was made after the research of specialized documentation and literature review, where it was found that such automation would be possible, if the steps for the generation of the necessary scripts are followed, in Google colab using the Python programming language, inside Blender by using the console to script with Python and in the Unity application creating scripts in the C# language. It is concluded that it is possible to perform the automation of the process described, in addition, by making use of this automation a great amount of time is saved compared to the process done manually.

**Keywords:** Automation; programming language; script; 3D models; intelligence; artificial.

## I. INTRODUCCIÓN

CON BASE en el proyecto “Uso de la inteligencia artificial para el modelado 3D aplicados a un ambiente de realidad virtual” [1], se planteó la opción de realizar los scripts para la automatización

del proceso que allí se mencionó, donde se consideró en primera instancia, hacer uso de Google Colab, para implementar la inteligencia artificial (IA) Point - E, en la cual se generaron modelos en nubes de puntos, que es un paso inicial para generar los modelos 3D, seguido a ello, dentro de la

1 Estudiante de ingeniería de sistemas en la Universidad Libre, perteneciente al semillero de investigación Sensorama. ORCID: <https://orcid.org/0009-0004-8599-0230>. Correo electrónico: carolinaa-garcias@unilibre.edu.co

2 Estudiante de ingeniería de sistemas en la Universidad Libre, perteneciente al semillero de investigación Sensorama. ORCID: <https://orcid.org/0009-0007-2522-1984>. Correo electrónico: juane-infantea@unilibre.edu.co

3 Docente Ingeniería de Sistemas, coordinador semillero Sensorama de la Universidad Libre sede El Bosque, desde febrero 2022. ORCID: <https://orcid.org/0000-0002-8399-2425>. Correo electrónico: ricardo.a.santaq@unilibre.edu.co

aplicación de Blender, se refinaron cada uno de estos modelos mejorando su forma y textura, para pasarlos finalmente a Unity, donde se realizó un Showroom con la apariencia de un museo, permitiendo ver cada uno de los modelos y permitiendo interactuar con algunos de ellos en este espacio.

La IA es una de las tecnologías que más se está usando recientemente, por su gran capacidad para la resolución de problemas, además, no solamente en ámbitos centrados en el uso de la tecnología, sino en muchos otros campos de aplicación[2]; en este caso, se hizo una mejora de un proyecto ya existente, donde la IA fue usada para la generación de modelos 3D, a través de Google Colab con el lenguaje de programación Python, fue refinada para que estos modelos solicitados se pudieran guardar de forma automática, sean 2 o 1000, según como se requieran, ya que, es posible solicitar la cantidad deseada.

Asimismo, en cada una de las aplicaciones usadas, se hizo uso de los lenguajes de programación para generar los Scripts de automatización. En la aplicación de Blender, se tuvo en cuenta que existe una API con la que se pueden hacer Scripts dentro de la aplicación, esta usa el lenguaje de programación Python, el cual, hizo que realizar los códigos necesarios, fuese mucho más sencillo y se tuvieran mejoras notables al hacerlo de esta forma, comparado con el tiempo al realizarlo de forma manual.

Otra herramienta muy importante que se utilizó en ese proceso fue Unity, en la cual, usando el lenguaje de programación C# y la herramienta para generar Scripts, por medio de Unity editor, una API brindada por la aplicación.

Principalmente, el proceso inicial mencionado, fue hecho todo de forma manual, por esta razón, en este proyecto, se hizo una automatización preliminar y básica del proceso, donde se incluyó la creación de Scripts, con ayuda de distintos lenguajes de programación, que son aceptados y leídos correctamente por cada una de las aplicaciones que se utilizaron para dicho proceso.

Este proyecto fue principalmente hecho como una investigación detallada, para comprobar la posibilidad de hacer la automatización de cada uno de los componentes, basando los resultados obtenidos en Scripts hechos a partir de contenidos y

conocimientos netamente teóricos, para mostrar los pasos que se deben seguir.

Es importante resaltar que se busca realizar la automatización del proceso mencionado, ya que, el objetivo de seleccionar un proyecto ya realizado y mejorarlo, es que sea más práctico su uso para cualquier persona que necesite realizar replicar este o realizar un proyecto similar y que no necesite de mucho tiempo al realizar todo el proceso, ya que, este puede ser usado muchas áreas de estudio como una herramienta de aprendizaje.

## II. PROBLEMA A RESOLVER

El problema que aborda este proyecto radica en la necesidad de mejorar la eficiencia de los procesos, que surge al generar modelos 3D con IA, mediante un paso a paso hecho de forma manual.

Luego de notar esta gran problemática, se pensó en una automatización que optimizara el uso de todos los recursos destinados al proceso, con base en esto, se identifican tres momentos clave que pueden ser automatizados, iniciando con la creación de los modelos dentro de Colab, seguido del refinamiento o revisión inicial dentro de Blender y por último, la ubicación de los modelos finales dentro del Showroom en Unity, ya que, estos procesos resultaban ser los más tardados haciendo que tareas simples se convirtiesen en complejas rápidamente, por el desperdicio de tiempo que tiene.

Teniendo en cuenta el proceso hecho en las 3 aplicaciones, tiene una tendencia exponencial haciendo que al desarrollar un modelo se consumieron 10 segundos, este tiempo aumentara al desarrollar 10 modelos, complicando el desarrollo de una gran cantidad de modelos, por lo anterior, se propone una solución que funcione para x cantidad de modelos haciendo que no se desperdicien ni el tiempo ni los recursos disponibles.

## III. MARCO DE REFERENCIA

La IA se puede definir como la capacidad que llegan a tener las computadoras, para hacer uso de algoritmos, que les permita adquirir aprendizaje de distintos datos dados, que podrían ser de distintos temas, dependiendo del ejemplo[3]. En el

amplio saber de la IA, es posible hacer desde predicciones hasta toma de decisiones, como lo haría una persona en su vida cotidiana. Esta se comprende como la habilidad de una computadora para replicar actividades con inteligencia y decisión propia, siendo propio de un ser humano, no de una máquina[4].

La IA, entre sus funciones, tiene principalmente la búsqueda de soluciones a diferentes problemas existentes, que muchas veces son triviales para los seres humanos. Esta tecnología ha avanzado tanto al pasar los años, que actualmente soluciona problemas de todo tipo, desde temas tecnológicos y con ayudas para los programadores, hasta temas más complejos como el hecho de que un auto pueda “ver” y de esta forma, no necesitar de un conductor, logrando conducir de forma autónoma[5].

Para usar y crear una IA, generalmente se debe manejar el lenguaje de programación Python, con las librerías necesarias para ello, ya que, permiten su creación de una forma sencilla y accesible[6]. Estas librerías y lenguaje de programación se pueden usar en Google Colaboratory, una herramienta creada para el desarrollo de programas en una plataforma virtual de forma gratuita, donde se tiene un espacio en la nube de Google para almacenar todos los trabajos realizados y sin la necesidad de usar los componentes propios de la computadora personal[7].

Junto a la IA, también se ha aplicado la automatización a distintos procesos que se pueden realizar en el día a día, al ser esto tan importante, se ha convertido en un tema muy polémico dentro de la sociedad, debido a las controversiales opiniones que se tienen de estos temas. La automatización, espera mejorar la eficiencia de muchos de los procesos realizados en las diferentes máquinas o programas existentes hasta el momento.

A lo largo de la historia, se ha visto como la automatización ha reemplazado los trabajos manuales, siendo más rápido y con menor cantidad de errores. Junto a esto, la mejora de softwares se ha vuelto de vital importancia, ya que, esto da solución a problemas, haciendo distintos cambios, que han logrado dar soluciones cada vez más eficientes[8], por lo que, se ha convertido en algo muy apetecido, gracias a su eficacia en cada proceso

realizado por máquinas y/o softwares[9]. Debido a esto, es importante automatizar cada uno de los procesos que pueden ser de gran impacto, como los realizados en la aplicación de Blender.

Para realizar correctamente una automatización en la aplicación de Blender, es necesario saber que en esta se trabaja con una API para Python que se encuentra dentro de la aplicación misma[10], para poder tener mayor contexto, conocimiento y formas de hacer bien la programación, se debe hacer uso de la documentación propia de Blender donde tiene información específica y completa de la API con la que se puede trabajar[11]. También, se debe tener en cuenta lo que se desea hacer en dicha aplicación y la automatización que se desea realizar, para ello se debe tener apoyo de documentación propia que muestra paso a paso como programar con Python dentro de la aplicación[12].

Otro impacto importante y de gran valor es automatizar cada uno de los procesos que se pueden en la aplicación de Unity, pudiendo hacer Showrooms con las especificaciones dadas inicialmente sin tener que seguir muchos pasos manualmente, para ello es necesario seguir la documentación que brinda la página de Unity[13], donde se tienen explicaciones y contenidos relacionados al lenguaje de programación C#, haciendo uso de una API especial para esta aplicación, teniendo como guía documentos que presentan las bases para crear códigos en esta aplicación[14].

## IV. DESARROLLO

Para este proyecto, es esencial automatizar el proceso en la creación y modelado de objetos 3D, con base a una metodología cuantitativa[15], ya que, en este caso, lo único que se hizo fue plantear la posibilidad de automatizar los procesos, usando diferentes lenguajes de programación, según los softwares que se usan, por tal motivo, la perfección y refinamiento total de todo el proceso, no fue algo que se tuvo presente ni esencial al momento de realizar el proyecto.

En este punto, lo que se necesitó para lograr el objetivo, fue tener conocimientos de los softwares implementados y de los lenguajes de programación a usar, estos datos se recogen con la búsqueda de información contenida en trabajos de grado,

artículos de investigación y demás contenido verídico y certificado, encontrado en la herramienta Google académico, donde se toma la mayor cantidad de información posible, que pueda ser útil para la automatización del proceso.

Uno de los obstáculos a vencer en esta investigación y lograr la entrega de resultados, fue la falta de conocimientos empíricos de la programación en los entornos y lenguajes de Blender y de Unity, debido a que no se tenía total certeza de que los resultados esperados, basados en la teoría, fuesen los deseados. Esta investigación logró identificar los Scripts en cada uno de los entornos y lenguajes que se manejan.

Con ayuda de esta investigación y la creación de los Scripts, se tuvieron en cuenta distintos factores para tomar las medidas de los tiempos aproximados tanto para el proceso manual como automático para 10 modelos, mostrado en la tabla I. En Google Colab, fue con base a la creación de los modelos en nubes de puntos, a partir de los prompts dados; en Blender, el tiempo se tomó teniendo en cuenta el traslado de los modelos a esta aplicación y el suavizado; en Unity, se tuvo en cuenta el traslado de los modelos y el posicionamiento de ellos en todo el espacio propuesto allí.

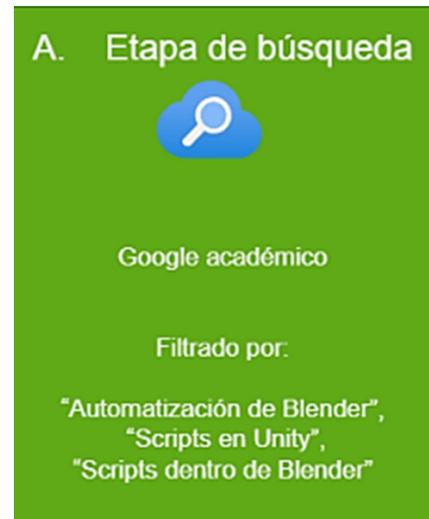
**Tabla I.** comparación de los tiempos.

Proceso	Manual	Automático
Google Colab	1 hora y 20 minutos	15 minutos
Blender	20 minutos	2 minutos
Unity	20 minutos	1 minuto
<b>Total</b>	<b>2 horas</b>	<b>18 minutos</b>

## V. MÉTODOS

### A. Etapa de búsqueda

Se hizo la búsqueda de información necesaria, por medio del buscador especializado Google académico, donde se filtró por palabras clave como "Automatización de Blender", "Scripts en Unity", "Scripts dentro de Blender" para obtener mejores resultados de las búsquedas y asegurarse que la información era lo más pertinente posible. Fig. 1.



**Fig. 1.** Búsqueda de información [Elaboración propia método A].

### B. Etapa de revision

Se reviso la información obtenida donde se encontraron diferentes documentos. Se tomó dicha información y se seleccionó lo que se considera relevante para esta investigación, ya que, se puede tomar como ayuda, debido a que contiene temas como automatización de procesos, scripts dentro de las herramientas Google, Blender y Unity, así como también, formas de entender los lenguajes a usar, esta información encontrada es verificada y certificada. Fig. 2.



**Fig. 2.** Revisión de la información [Elaboración propia método B].

### C. Etapa de síntesis

Se hizo una síntesis de la información final, esto mediante Scripts, ofreciendo buenos resultados, donde se evidenció la forma en que desarrollan los Scripts de automatización para Google, Blender y Unity. Fig. 3.

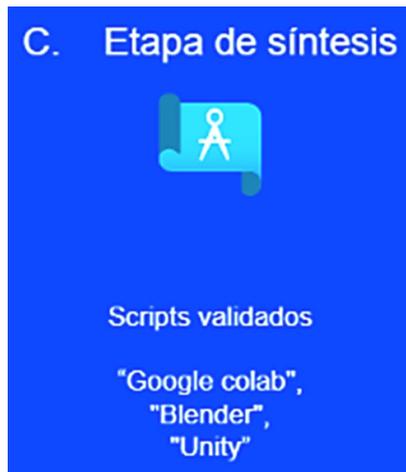


Fig. 3. Sintetizar la información  
(Elaboración propia método C).

## VI. RESULTADOS

### A. Automatización en Google Colab

Inicialmente, se cuenta con el Script de Google Colab, hecho para automatizar el proceso de creación de modelos 3D en nubes de puntos, haciendo uso de un control de flujos en el código, realizado mediante ciclos repetitivos, para crear varios modelos y guardarlos dentro del ambiente de Colab.

Primero, se crearon 2 listas encargadas de contener las peticiones de los modelos deseados que se realizaran, una llamada «prompts» y la otra llamada «names»; la cual contiene los nombres con los que se guardan los modelos, luego se hizo el control de flujos, mediante un ciclo for, recorriendo la lista prompts según su tamaño.

Se seleccionó la petición, usando un iterador que recorre la lista, seguido a esto, se pasa al ciclo while, donde se establece que se haga la iteración, mientras este sea «True» y que realice el mismo proceso que se hacía antes de forma normal; el cual se basó en la creación del modelo 3D con la Figura seleccionada.

```

1  from traitlets.traitlets import Undefined
2  import time
3
4  prompts=[«a big brown dog», «an airplane with
5  thicker wings»]
6  names=[«animal1.ply»,»airplane1.ply»]
7
8  start_time = time.time()
9
10 for i in range(len(prompts)):
11     # Set a prompt to condition on.
12     prompt = prompts[i]
13     while True:
14         # Produce a sample from the model.
15         samples = None
16         for x in tqdm(sampler.sample_batch_
17             progressive(batch_size=1, model_kwargs=
18                 dict(texts=[prompt]))):
19             samples = x
20
21         pc = sampler.output_to_point_clouds
22             (samples)[0]
23
24         device = torch.device('cuda' if torch.cuda.is_
25             available() else 'cpu')
26
27         print('creating SDF model...')
28         name = 'sdf'
29         model = model_from_config(MODEL_
30             CONFIGS[name], device)
31         model.eval()
32
33         print('loading SDF model...')
34         model.load_state_dict(load_checkpoint
35             (name, device))
36
37         # Produce a mesh (with vertex colors)
38         mesh = marching_cubes_mesh(
39             pc=pc,
40             model=model,
41             batch_size=4096,
42             grid_size=128, # increase to 128 for resolution
43                 used in evals
44             progress=True,
45             )
46         # Write the mesh to a PLY file to import into
47             some other program.
48         with open(names[i], 'wb') as f:
49             mesh.write_ply(f)
50         break
51         elapsed_time = time.time() - start_time
52         print(f«Elapsed time for execution: {elapsed_
53             time} seconds»)

```

Esto se guarda donde se tienen dos diferencias; la primera es que se usó el mismo iterador para seleccionar el nombre correspondiente, a la misma posición de la petición inicial, luego se agrega un “break” al guardar el modelo, para completar el ciclo y crear el siguiente modelo.

Volviendo a el for inicial, también es una parte del código agregada, pero no se encarga de automatizar el proceso, esta es la medición del tiempo usado para la creación de los modelos, de modo que, antes de entrar en el primer ciclo for, se inicia un contador, que al final del código se pausa e imprime, para poder saber la duración de la ejecución.

Para realizar código presentado anteriormente, se debe tener conocimiento del manejo de Google Colab, así como también, del lenguaje de programación Python, por esta razón, este es el único código hecho sin búsqueda de información y del cual se tiene total seguridad de su funcionamiento.

Es importante mencionar que, este código, debe ser compilado justo después de compilar el “Sampler”, y darle los valores necesarios, que, a su vez, salta el proceso de visualización de los modelos dentro de Colab, que no fue considerado importante para el proceso de automatización.

## B. Automatización en Blender

Este código, se hizo uso de la interfaz de programación de aplicaciones (API) de Blender por medio de Python, donde se pudo aprovechar la posibilidad de usar todas las herramientas de Blender, mediante código y, en este caso, se pensó que los pasos de importar los modelos 3D y del suavizado de cada uno.

```

1  import bpy
2  import os
3
4  path = «C:/Users/juane/Downloads/Modelos/
    PLY»
5
6  files =[f for f in os.listdir(path) if f.ends
    with('.ply')]
7
8  for f in files:
9      filepath = os.path.join(path, f)
10
```

```

11  bpy.ops.import_mesh.ply(filepath=filepath)
12
13  obj = bpy.context.selected_objects[0]
14
15  bpy.context.view_layer.objects.active = obj
16
17  bpy.ops.object.shade_smooth()
```

Lo mostrado fue más rápido, ya que, se realizó con todos los modelos al mismo tiempo, haciendo que, mientras se importa cada modelo, se le va aplicando el suavizado, para que la textura de los modelos sea menos rugosa y, de esta forma, facilitar el trabajo del modelado, que deberá seguir siendo manual.

Teniendo en cuenta la información encontrada en documentos de Google Académico, donde se obtuvieron explicaciones del uso de la API para Python dentro de Blender[4], de este modo, se supo que, si es posible automatizar uno de los pasos dentro del proceso realizado en Blender, para esto también se usó como fundamenta la página oficial de Blender con toda la documentación de la API[5]. Asimismo, se usó toda la documentación e información encontrada en el buscador especializado, lo cual ayudó mucho a obtener buenos resultados y Scripts sin errores.

## C. Automatización en Unity

En el software de Unity, se encontró una API, fue una herramienta importante para la automatización en la aplicación, llamada “Editor Scripting”, con esto se logró tener el código propuesto para la automatización de los pasos, que son importar los modelos a la escena; es probable que se puedan hacer muchas mejoras para la automatización de otros pasos dentro de este software, pero por falta de información y conocimientos, no fue posible proponer más Scripts para automatizaciones.

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEditor;
4  using UnityEngine;
5
6  public class ScriptInicial : MonoBehaviour
7  {
8      private void Start()
9      {
```

```
10 string carpetaDeModelos = «Assets/Modelos»;
11
12 string[] guids = AssetDatabase.FindAssets
  («GameObject», new string[] { carpetaDe
  Modelos });
13
14 Debug.Log(«Encontrados « + guids.Length
  + « assets en la carpeta»);
15
16 foreach (string guid in guids)
17 {
18 string path = AssetDatabase.GUIDTo
  AssetPath(guid);
19 GameObject modeloPrefab = AssetData
  base.LoadAssetAtPath <GameObject>
  (path);
20 if(modeloPrefab != null)
21 {
22 GameObject modeloInstanciado = Instan
  tiate(modeloPrefab);
23 modeloInstanciado.transform.position =
  new Vector3(1.0f, 1.0f, 1.0f);
24 }
25 Debug.Log(«Instanciado prefab « + modelo
  Prefab.name);
26 }
27 Debug.Log(«Terminando la ejecución del
  script»);
28 }
29 }
```

Para poder hacer este código, con base a lo teórico, se hizo uso de documentación propia de la página de Unity[8], donde se explica como con el lenguaje C# se puede hacer uso de la API, generando los scripts de automatización para el software. Así mismo, se hizo lectura de un proyecto de grado[9], donde se dan bases para la creación de código en Unity y se explican distintas formas de hacer un juego allí, con el lenguaje C#.

## VII. CONCLUSIONES

Se confirmó que es posible hacer una automatización de los procesos, en todos los software usados para el proyecto, esto fue posible gracias a la lectura de la documentación de dichos programas y de trabajos especializados.

Las soluciones propuestas no pueden ser dadas como las finales, pero son un primer paso hacia la

automatización del proyecto “Uso de la inteligencia artificial para la creación de modelos 3D”, y se considera que, con el uso de estas soluciones iniciales, se pueden obtener soluciones definitivas y totalmente funcionales.

La información para la realización de estos scripts es suficiente y pertinente, con respecto a lo que se desea hacer, además, existen muchos documentos de investigación que detallan el paso a paso para hacer las automatizaciones dentro de las aplicaciones, haciendo que la etapa de investigación fuese más simple.

## VIII. REFERENCIAS

- [1] J. Infante y C. García, *Uso de la ineligencia artificial para la creación de modelos 3D (en prensa)*, Bogotá: Ciencia y poder aéreo, 2023.
- [2] A. Erazo, F. Ramos, P. Galarza y M. Boné, «La inteligencia artificial aplicada a la optimización de programas informáticos,» *Journal of Economic and Social Science Research*, vol. III, n° 1, pp. 48-63, 2023.
- [3] L. Rouhiainen, *Inteligencia artificial - 101 cosas que debes saber hoy sobre nuestro futuro*, Barcelona: Editorial Planeta S.A., 2018.
- [4] D. Ignacio, S. Jesús, V. Víctor y S. Pedro, «Aplicaciones de la inteligencia artificial en cardiología: el futuro ya está aquí,» *Revista Española de Cardiología*, pp. 1065-1075, Diciembre 2019.
- [5] IBM, «Youtube - Discovery Latinoamérica | Inteligencia Artificial - IBM,» 26 Septiembre 2018. [En línea]. Available: <https://www.youtube.com/watch?v=5rvZBsueMoc>. [Último acceso: 2023].
- [6] C. Monroy, «EL LENGUAJE PYTHON Y SU POTENCIAL EN EL DESARROLLO DE SOFTWARE DE INTELIGENCIA ARTIFICIAL,» *Revista Científica de la Universidad Salvadoreña Alberto Masferrer*, pp. 18-41, Marzo 2022.
- [7] C. E., M. R. y M. J., «Más allá de Jupyter: usando Google Colab para la programación de robots,» *Repositorio Universidad de Coruña*, pp. 662-669, 2022.
- [8] D. Acemoglu y P. Restrepo, «Automation and New Tasks: How Technology Displaces and Reinstates Labor,» *Journal of Economic Perspectives*, vol. 33, n° 2, pp. 3-30, 2019.
- [9] D. Acemoglu y P. Restrepo, «Artificial Intelligence, Automation, and Work,» de *The Economics of Artificial Intelligence: An Agenda*, Chicago, University of Chicago Press, 2019, pp. 197-236.

- [10] F. Blender, «Blender 2.68.0 r58372 - API documentation,»[En línea]. Available: [https://docs.blender.org/api/blender\\_python\\_api\\_2\\_68a\\_release/#](https://docs.blender.org/api/blender_python_api_2_68a_release/#). [Último acceso: 20 Agosto 2023].
- [11] F. Blender, «Blender 2.68.0 r58372 - API documentation - Operators (bpy.ops),»[En línea]. Available: [https://docs.blender.org/api/blender\\_python\\_api\\_2\\_68a\\_release/bpy.ops.action.html?highlight=bpy.ops#bpy.ops.action](https://docs.blender.org/api/blender_python_api_2_68a_release/bpy.ops.action.html?highlight=bpy.ops#bpy.ops.action). [Último acceso: 20 Agosto 2023].
- [12] Anónimo, *Programación con Python sobre Blender*, Creative Commons, 2021.
- [13] U. Technologies, «Unity Documentation,» 11 Agosto 2023.[En línea]. Available: <https://docs.unity3d.com/ScriptReference/Editor.html>. [Último acceso: 20 Agosto 2023].
- [14] D. G. García, *Sistema para el desarrollo y gestión de videojuegos en Unity3D*, 2017.
- [15] R. Hernandez-Sampieri y C. Mendoza, «Las rutas cuantitativa, cualitativa y mixta,» de *Metodología de la investigación*, Ciudad de México, McGRAW-HILL INTERAMERICANA EDITORES, S.A. de C.V., pp. 1-7.