

# Universidad Mayor Real y Pontificia de San Francisco Xavier de Chuquisaca

FACULTAD DE TECNOLOGIA



ING. DISEÑO Y ANIMACION DIGITAL

Inteligencia Artificial

**COM 300**

**NOMBRE:** Betsabel Abigail

**APELLIDO:** Alvarado Choque

**DOCENTE:** Pacheco Lora Carlos Walter

**FECHA:** 23/06/2025

SUCRE – BOLIVIA

# Generador de Paisajes con Agrupamiento no Supervisado en Blender

## 1. Introducción

En este proyecto final se presenta un recurso multimedial interactivo y visual generado mediante Blender, que simula un paisaje natural con diferentes regiones: pradera, montaña, lago y bosque. El paisaje es creado de manera automática utilizando técnicas de aprendizaje no supervisado, específicamente el algoritmo KMeans, aplicado sobre la malla de un plano subdividido.

El propósito principal del proyecto es demostrar cómo es posible aplicar modelos de inteligencia artificial dentro de un entorno creativo y visual como Blender, permitiendo automatizar la generación de escenas complejas y coherentes sin intervención directa en cada componente individual.

## 2. Objetivos

### Objetivo general:

- Implementar un sistema procedural en Blender que genere un paisaje tridimensional utilizando técnicas de aprendizaje no supervisado.

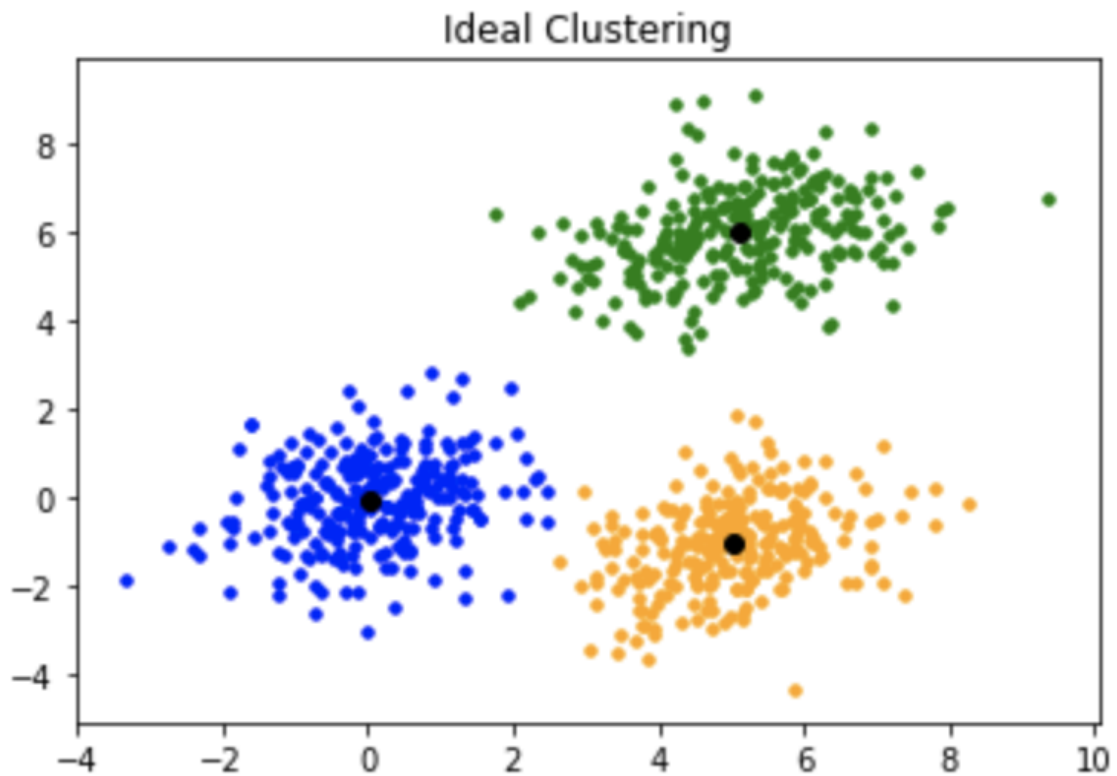
### Objetivos específicos:

- Utilizar el algoritmo KMeans para agrupar regiones del terreno 3D en zonas naturales diferenciadas.
- Aplicar variaciones de altura y materiales a cada zona en función del cluster asignado.
- Insertar elementos naturales como árboles y rocas de forma lógica según el tipo de terreno.
- Exportar y documentar todo el proceso para demostrar su integración con Blender y Python.

## 3. Fundamentación Técnica

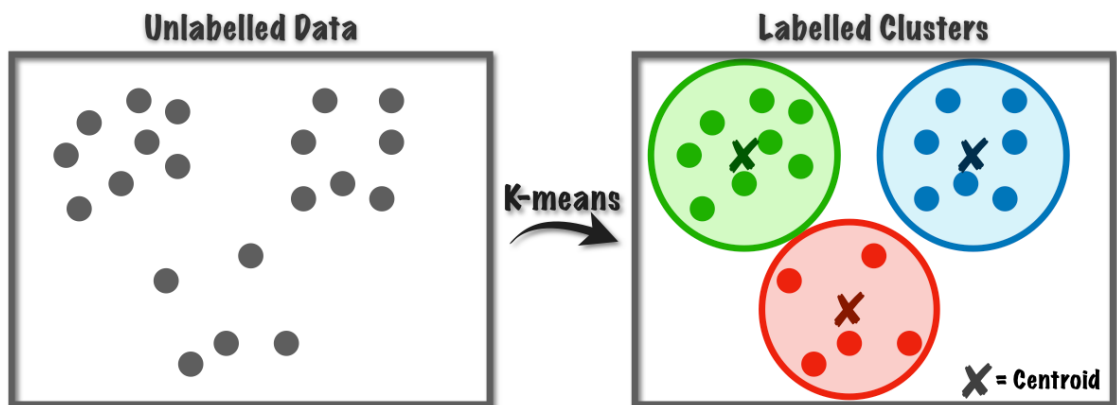
### ¿Qué es k-means?

**KMeans** es un algoritmo de **aprendizaje no supervisado** que se utiliza para **agrupar datos** en un número específico de grupos o *clusters*. El objetivo del algoritmo es organizar los datos en grupos donde los elementos de cada grupo sean lo más parecidos posible entre sí, y lo más distintos posible de los elementos de los otros grupos.



El proceso funciona así:

1. Se elige un número de grupos ( $k$ ).
2. El algoritmo selecciona aleatoriamente  $k$  centros iniciales (llamados centroides).
3. Cada dato se asigna al centro más cercano.
4. Se recalculan los centros promedios de cada grupo.



para dividir el plano base del terreno (subdividido en vértices) en 4 clusters distintos basados en la proximidad de sus coordenadas  $X$  y  $Y$ . Cada grupo generado es tratado como una zona geográfica distinta:

- Cluster 0: Pradera
- Cluster 1: Montaña
- Cluster 2: Lago
- Cluster 3: Bosque

A partir de esa clasificación, se manipulan las alturas (coordenada Z), los materiales y los objetos (como árboles y rocas) de manera diferente en cada región.

El ruido procedural (tipo Perlin simulado) se genera con funciones trigonométricas para dar variación natural al relieve. Finalmente, todos los elementos son creados dentro de Blender utilizando su API de scripting con Python.

#### **4. Desarrollo del Recurso**

El proceso de creación del paisaje incluyó las siguientes etapas:

##### a. Generación del terreno base

- Se creó un plano subdividido de tamaño configurable (30x30 unidades con 70 subdivisiones).
- Se extrajeron las coordenadas XY de cada vértice como datos de entrada para el modelo de clustering.

##### b. Aplicación de KMeans

- Se clasificaron los puntos en 4 clusters utilizando KMeans de la librería sklearn.
- Cada vértice del plano recibió una etiqueta correspondiente a una región natural.

##### c. Asignación de altura y materiales

- A cada región se le aplicó una fórmula diferente de altura usando ruido procedural.
- Se generaron materiales únicos para cada zona, incluyendo transparencia para el lago.

##### d. Inserción de objetos naturales

- Árboles (tronco + hojas en forma de conos apilados) fueron colocados aleatoriamente sobre zonas de pradera y bosque.
- Rocas esféricas fueron ubicadas en regiones montañosas.
- Un plano con material semitransparente simula el agua del lago.

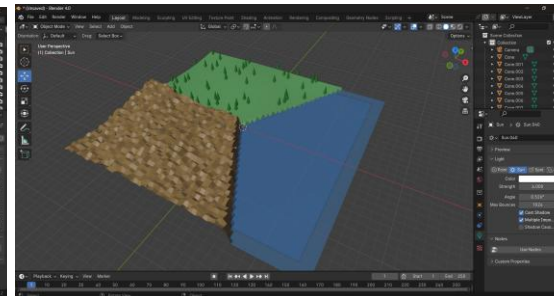
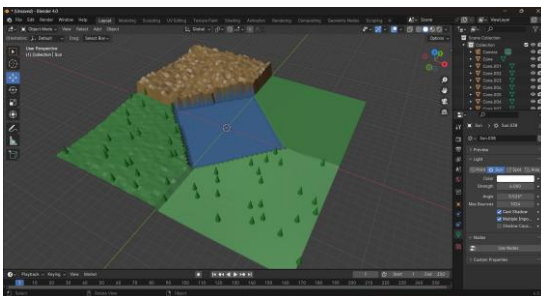
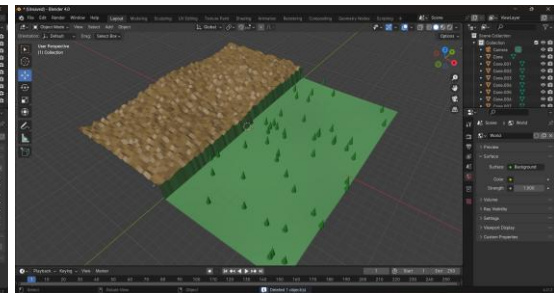
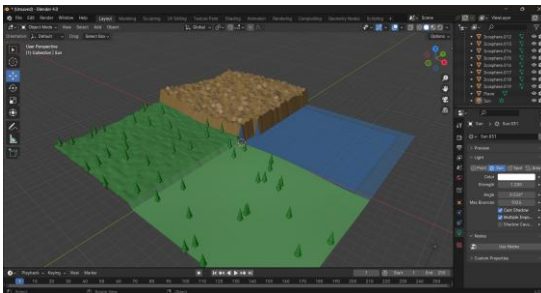
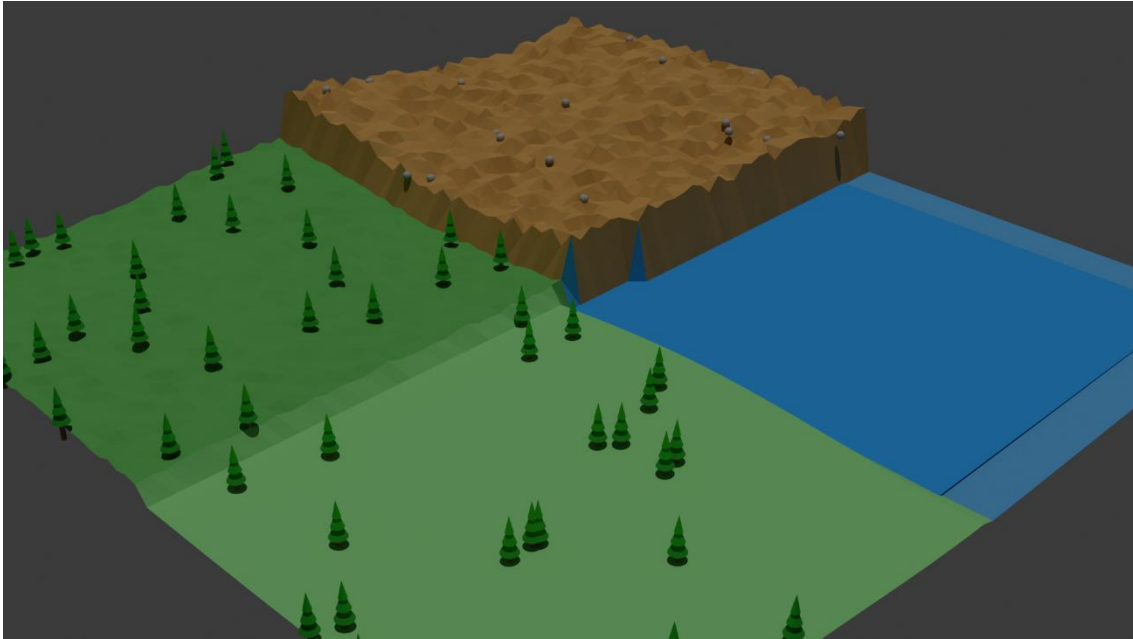
##### e. Configuración de escena

- Se añadieron una cámara y una fuente de luz solar para permitir renders de alta calidad.

## 5. Resultados

El sistema produce un paisaje 3D completamente procedural donde:

- La segmentación territorial se hace de forma automática.
- Cada tipo de terreno tiene su propio comportamiento visual y funcional.
- Los elementos como árboles y rocas aparecen solo en las zonas lógicas.



## 6. Conclusiones

Este proyecto demostró que es posible integrar técnicas de aprendizaje automático con herramientas visuales como Blender, generando recursos multimedia complejos de forma automatizada y coherente.

El uso de KMeans permitió dividir de manera eficiente el terreno en regiones diferenciadas, sin necesidad de etiquetar los datos previamente. El modelo fue interpretado visualmente mediante manipulación de altura, color y objetos, dando vida a un entorno natural.

## 7. Referencias

- Blender Python API: <https://docs.blender.org/api/current/>
- Scikit-learn KMeans: <https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html>
- Procedural Terrain Generation: [https://en.wikipedia.org/wiki/Procedural\\_generation](https://en.wikipedia.org/wiki/Procedural_generation)
- Python Math Library: <https://docs.python.org/3/library/math.html>