

# Desarrollo y Simulación de una Escena 2D

Luis Correa, Valeria Márquez, Maily Silva

## Resumen

Esta actividad se basa en la creación de una escena la cual será diseñada en 2D. Incluyendo una circunferencia y añadiendo un punto en cualquier ubicación. Para ello, se incluyeron funciones la cual nos permite visualizar de manera clara los diferentes cambios que esta puede tener, dichas funciones incluyen variables para crear la escena, cambiarla o incluir diferentes desplazamientos y simulaciones del punto.

**Palabras Claves:** escena 2D, animación, simulación, desplazamiento, circunferencia.

## 1 Introducción

El diseño de una escena en 2D es importante en los diferentes campos para detallar su estructura de manera que se pueda tener una visualización clara y eficiente mejorando la comprensión de los distintos sucesos. Su propósito como tal es aplicar los diferentes conceptos en base a la programación orientada a objetos y obtener una visualización mediante una gráfica que nos permita ejecutar una escena compuesta por una circunferencia y un punto con cualquier coordenada. La cual, incluimos funciones específicas para su proceso, tales como su creación, su modificación y la simulación de desplazamiento del punto dentro de la circunferencia.

## 2 Objetivos

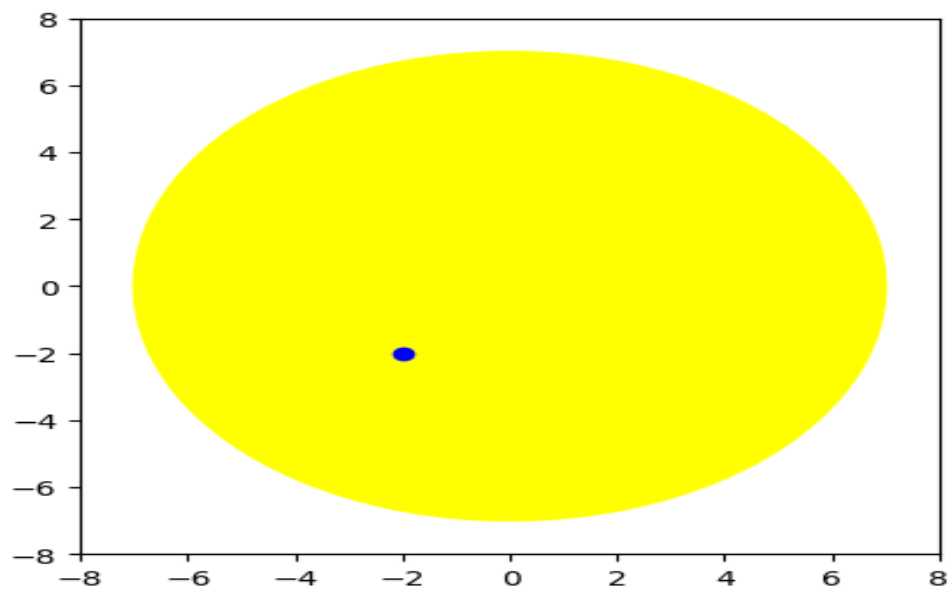
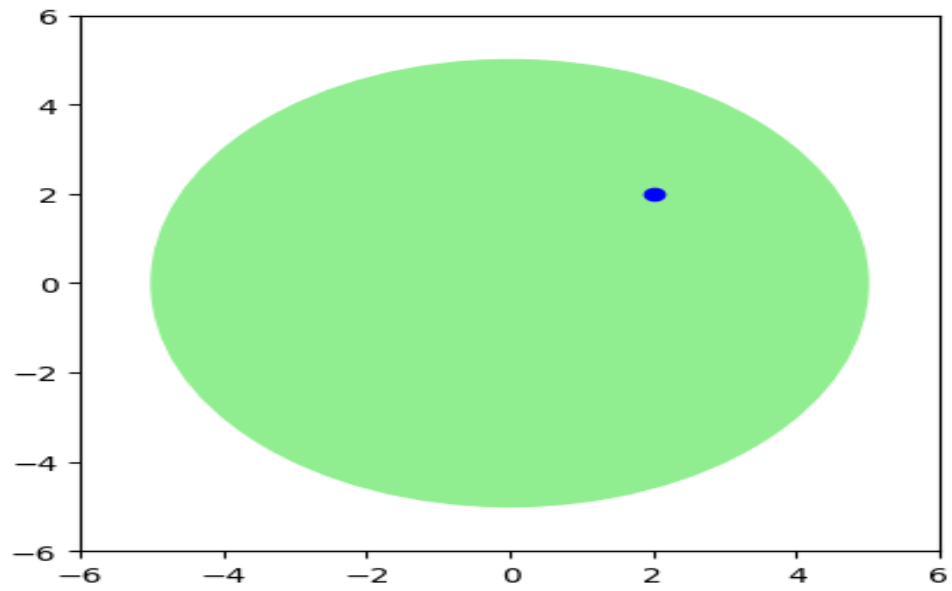
- Objetivo General
  - Desarrollar una clase en Python que permita la creación, modificación y simulación de una escena 2D, utilizando programación orientada a objetos y herramientas de visualización gráfica para representar y simular el punto dentro de la escena.
- Objetivos Específicos
  1. Implementar una clase Escena en Python que permita la creación de una escena 2D.
  2. Proporcionar métodos para modificar y visualizar la escena.
  3. Incorporar funcionalidad para mover el punto dentro de la escena.
  4. Implementar una simulación animada del movimiento del punto.

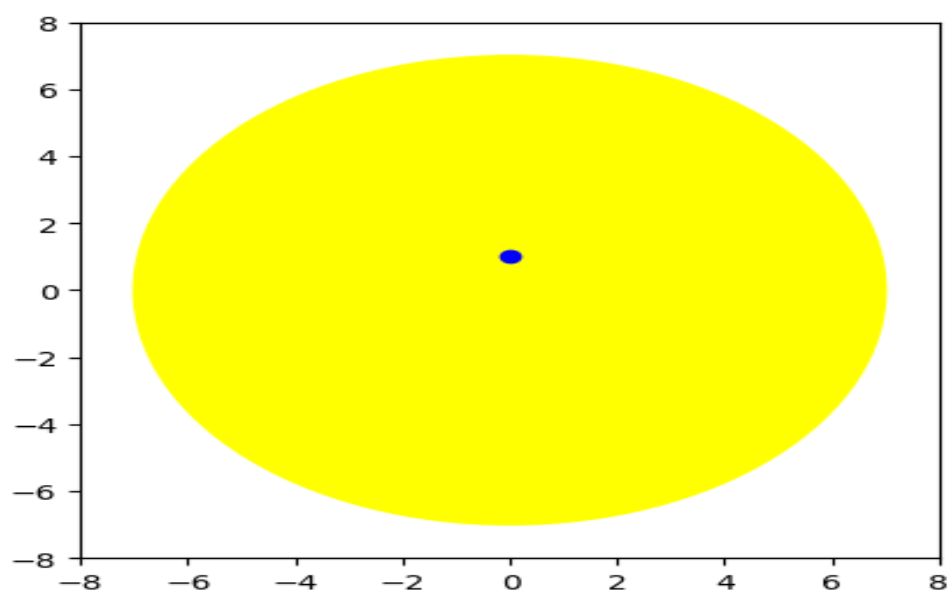
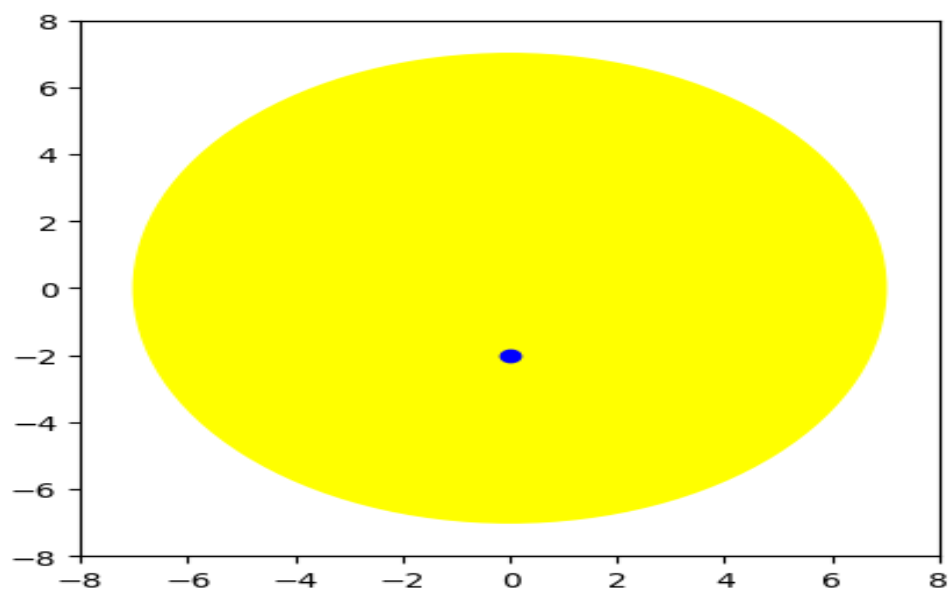
### 3 Descripción de la actividad

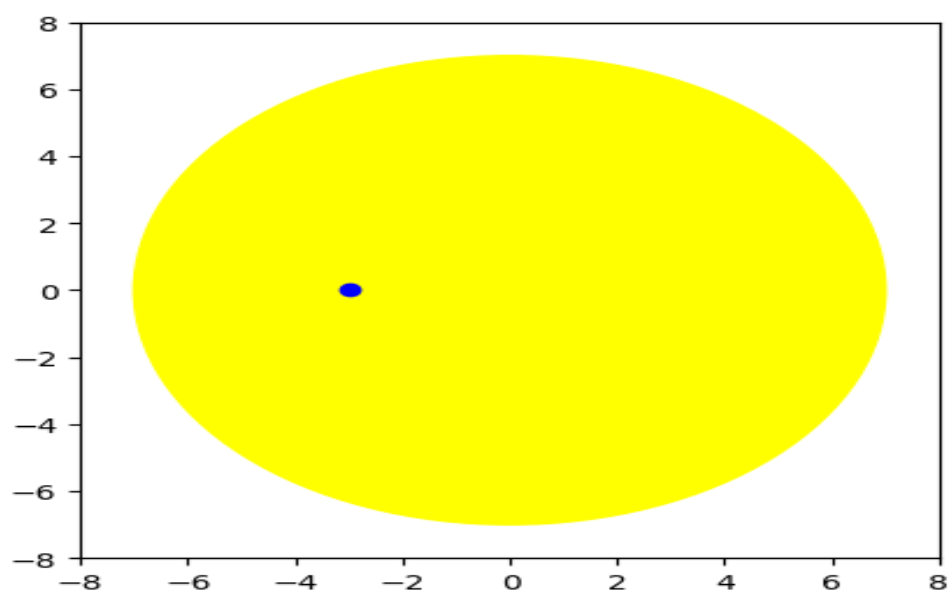
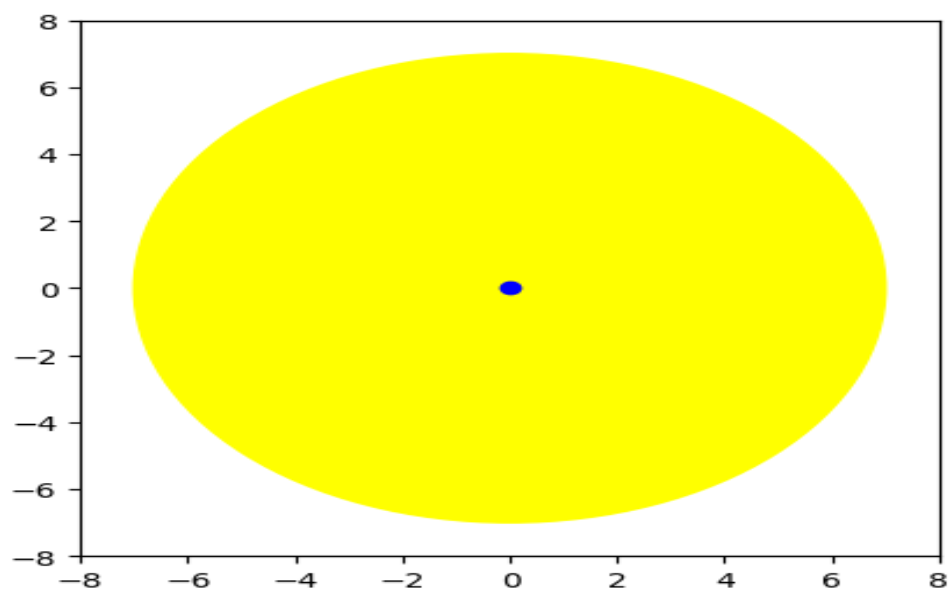
para realizar esta actividad lo primero que hicimos es crear una clase a la que llamaremos “Escena”, dentro del cual incluye una circunferencia y un punto añadido sobre ella. Después, se creo una variedad de funciones formuladas en operaciones vectoriales y que al final nos permita visualizar la escena y sus diferentes funciones de manera clara. Las funciones que añadimos son:

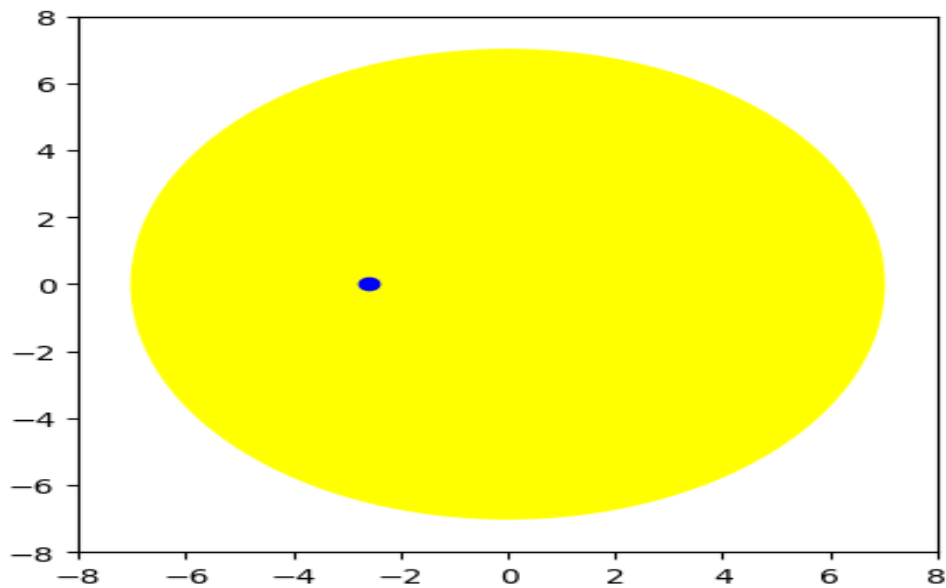
- Crear-escena: Permite instanciar un objeto de la clase Escena con una circunferencia de tamaño y color específicos. Se añadirá un punto superpuesto a la circunferencia.
- Cambiar-escena: Permite modificar los atributos de la escena, como el tamaño, el color de la circunferencia y la posición del punto.
- Dibujar-escena: Genera una representación gráfica de la escena mostrando la circunferencia y el punto en su posición actual.
- Desplazar puntos: Realiza el desplazamiento del punto en una dirección específica (derecha, izquierda, arriba o abajo) mediante operaciones vectoriales.
- Simular: Recibe como parámetros una velocidad y una dirección. También genera una animación que muestra el desplazamiento progresivo del punto dentro de la escena.

## 4 Gráficas









## 5 Análisis de cada gráfica

- Primera Gráfica

1. La circunferencia tiene un color verde claro y un radio aproximado de 5, por otro lado el punto azul está ubicado en la parte superior derecha específicamente en las coordenadas (2,2). Esta imagen representa la escena inicial creada con un radio definido y el punto en una posición distinta del centro. Esta escena ha sido generada con crear-escena con una configuración inicial.

- Segunda Gráfica

1. Se realizó un cambio de escena mediante cambiar-escena, modificando el color y el radio. El punto se encuentra más lejos del centro, lo que indica una configuración diferente con coordenadas (-2,-2). La circunferencia tiene un color amarillo y un radio mayor que en la primera gráfica y el punto azul está desplazado hacia la parte inferior izquierda.

- Tercera Gráfica

1. En esta gráfica podemos visualizar un desplazamiento a partir de la segunda gráfica con coordenadas (-2,-2) hacía las coordenadas (0,-2),

visualizando de esta manera un desplazamiento con 2 pasos hacia la derecha.

- Cuarta Gráfica

1. Esta escena tiene un desplazamiento 3 pasos hacia arriba a partir del tercer gráfico con coordenadas (0,-2), visualizando de manera clara el desplazamiento y quedando fijo en las coordenadas (0,1). su circunferencia es de color amarillo con un punto azul superpuesto y con el mismo radio que las demás gráficas a partir de la segunda gráfica.

- Quinta Gráfica

1. En esta gráfica también podemos observar un desplazamiento pero en sentido contrario que la cuarta gráfica con coordenadas (0,1) de este modo tiene un desplazamiento de 1 paso hacia abajo quedando así en las coordenadas(0,0).

- Sexta Gráfica

1. Podemos visualizar claramente un desplazamiento de 3 pasos hacia la izquierda a partir de las coordenadas de la quinta gráfica con coordenadas ((0,0) y quedando así en las coordenadas (-3,0).

- séptima - Gráfica de la Simulación

1. La imagen representa un estado intermedio de la simulación donde el punto azul se desplaza hacia la derecha dentro de la circunferencia amarilla. Según el código `escena.simular(velocidad=0.2, dirección='derecha', pasos=10)`, el punto debería moverse 10 unidades en total, el punto seguirá desplazándose en el eje (X) positivo hasta completar los 10 pasos, posiblemente llegando cerca del borde derecho de la circunferencia.

## 6 Conclusión

En este trabajo, hemos desarrollado una clase Escena que permite la representación y manipulación de una escena 2D con una circunferencia de fondo y un punto superpuesto. A través de funciones como crear-escena, cambiar-escena, dibujar-escena, desplazar-puntos y simular, hemos implementado operaciones vectoriales para modificar la posición del punto y generar animaciones dinámicas. El uso de operaciones vectoriales ha facilitado la manipulación del punto dentro de la escena, proporcionando una base matemática sólida para el desplazamiento y la simulación del movimiento. Además, la capacidad de cambiar los parámetros de la escena permite una flexibilidad en la