# Generative Neural Networks

# Equipo

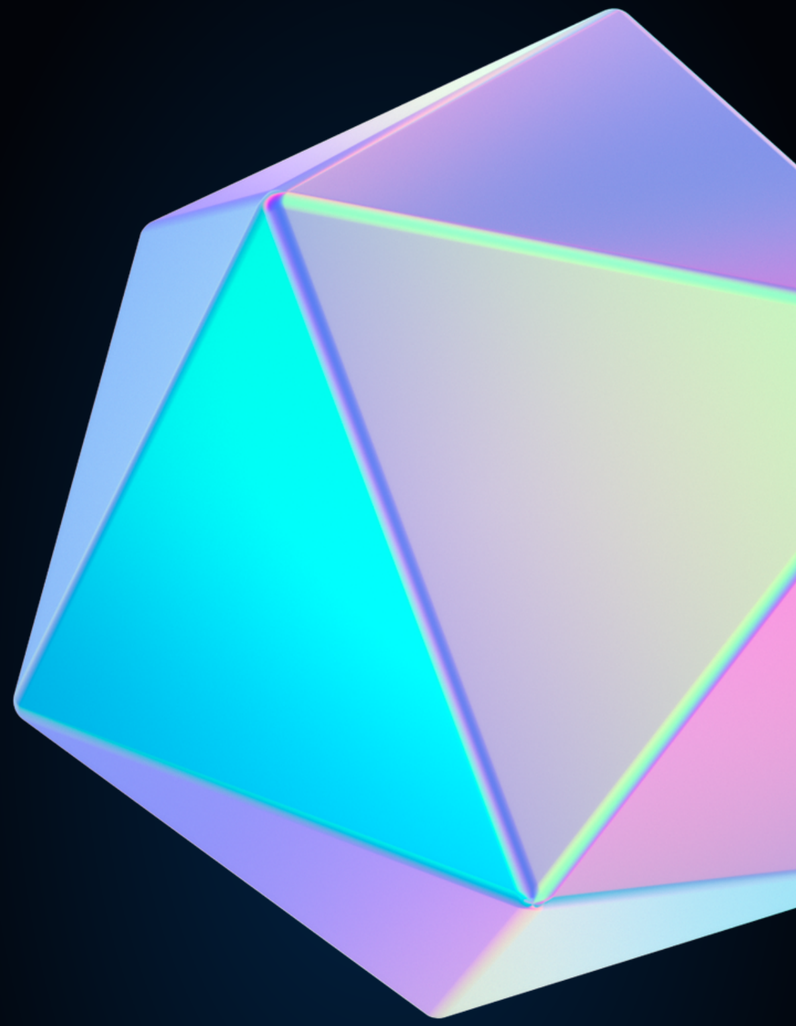
**Gabriel Balderrama**

**Angy Encinas**

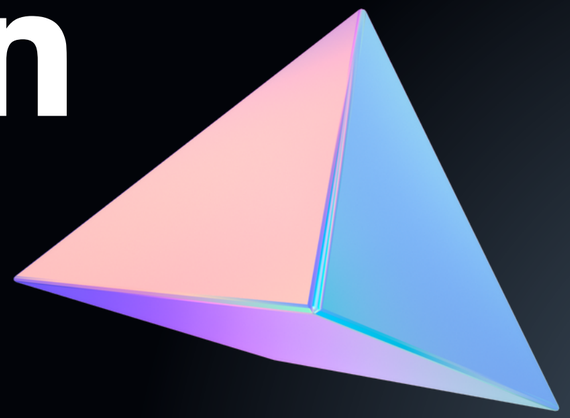
**Rodrigo Guardia**

# Que son las GNN's

Las Redes Neuronales Generativas son sistemas artificiales que aprenden a generar nuevos datos con características similares a los datos originales. Los GNN se pueden usar para generar imágenes, texto e incluso música. Se basan en redes neuronales profundas y han logrado resultados impresionantes en los últimos años.



# Como funcionan las GNN's



Las redes neuronales generativas utilizan un modelo generativo para aprender la distribución de probabilidad de los datos de entrada.

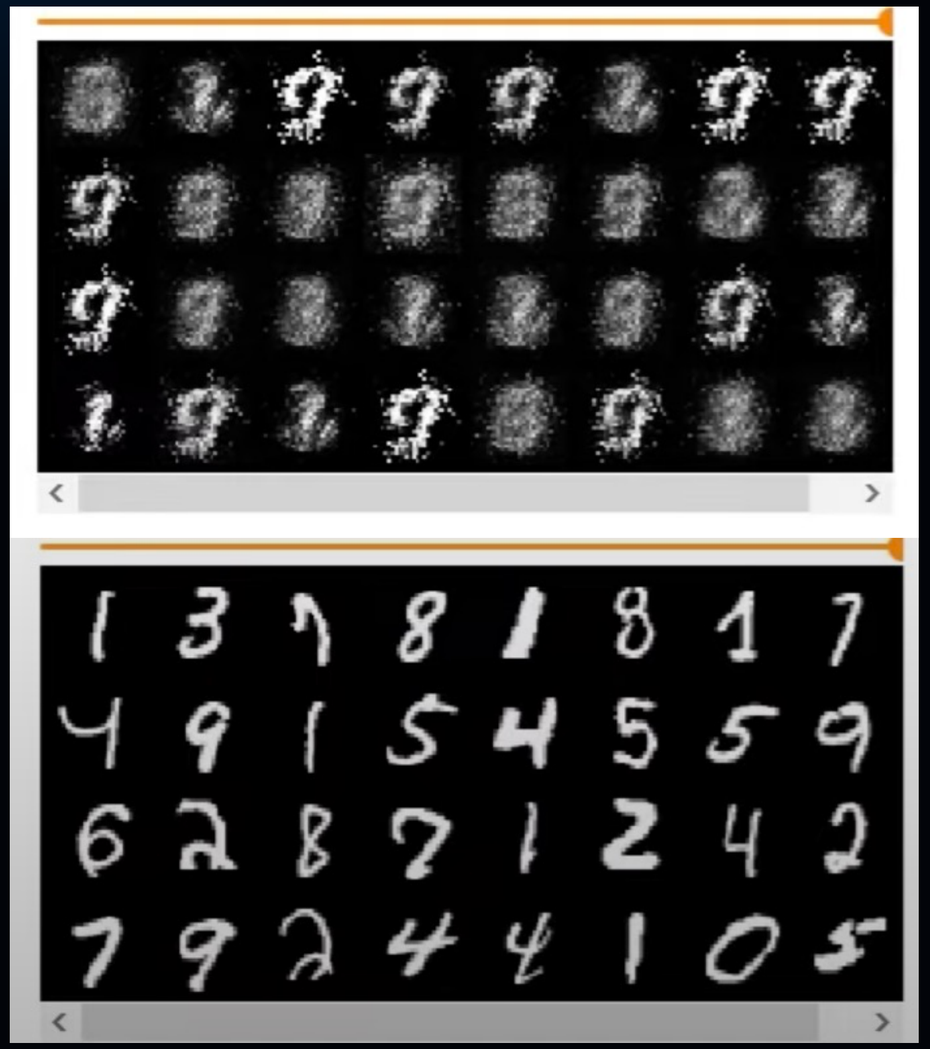
Una vez que el modelo ha aprendido la distribución, puede generar nuevos datos muestreándolos.

Los GNN más populares son Variational Autoencoders y las Generative Adversarial Networks

# AVANCES Y DESAFIOS DE LAS GNN

Los avances recientes de las GNNs incluyen el progreso y crecimiento de GANs, para generar imágenes de alta resolución y StyleGAN, que permite controlar características de las imágenes generadas. Otros avances son Text-2-Image, Video generation, y Conditional GANs.

# AVANCES Y DESAFIOS DE LAS GNN



Uno de los principales desafíos es el colapso de modo, donde GNN genera solo un conjunto limitado de muestras.

Otro desafío es la inestabilidad, donde el GNN no logra converger o produce muestras de mala calidad. Abordar estos desafíos es un área activa de investigación en el campo.

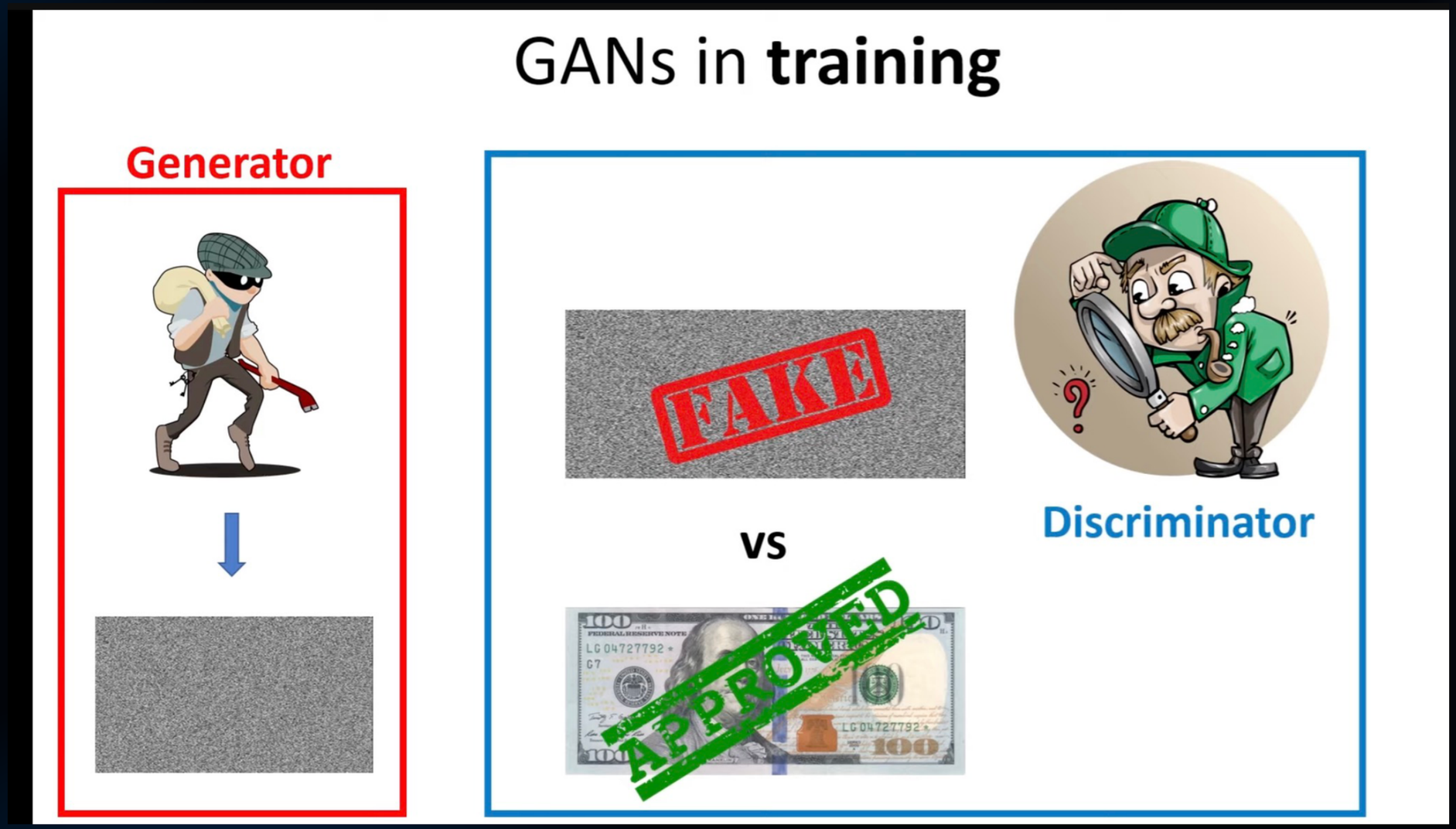
# GENERATIVE ADVERSARIAL NETWORKS (GAN)

Las redes adversarias generativas (GAN) son un tipo de algoritmo de aprendizaje profundo que puede generar nuevas muestras de datos similares a los datos de entrenamiento.

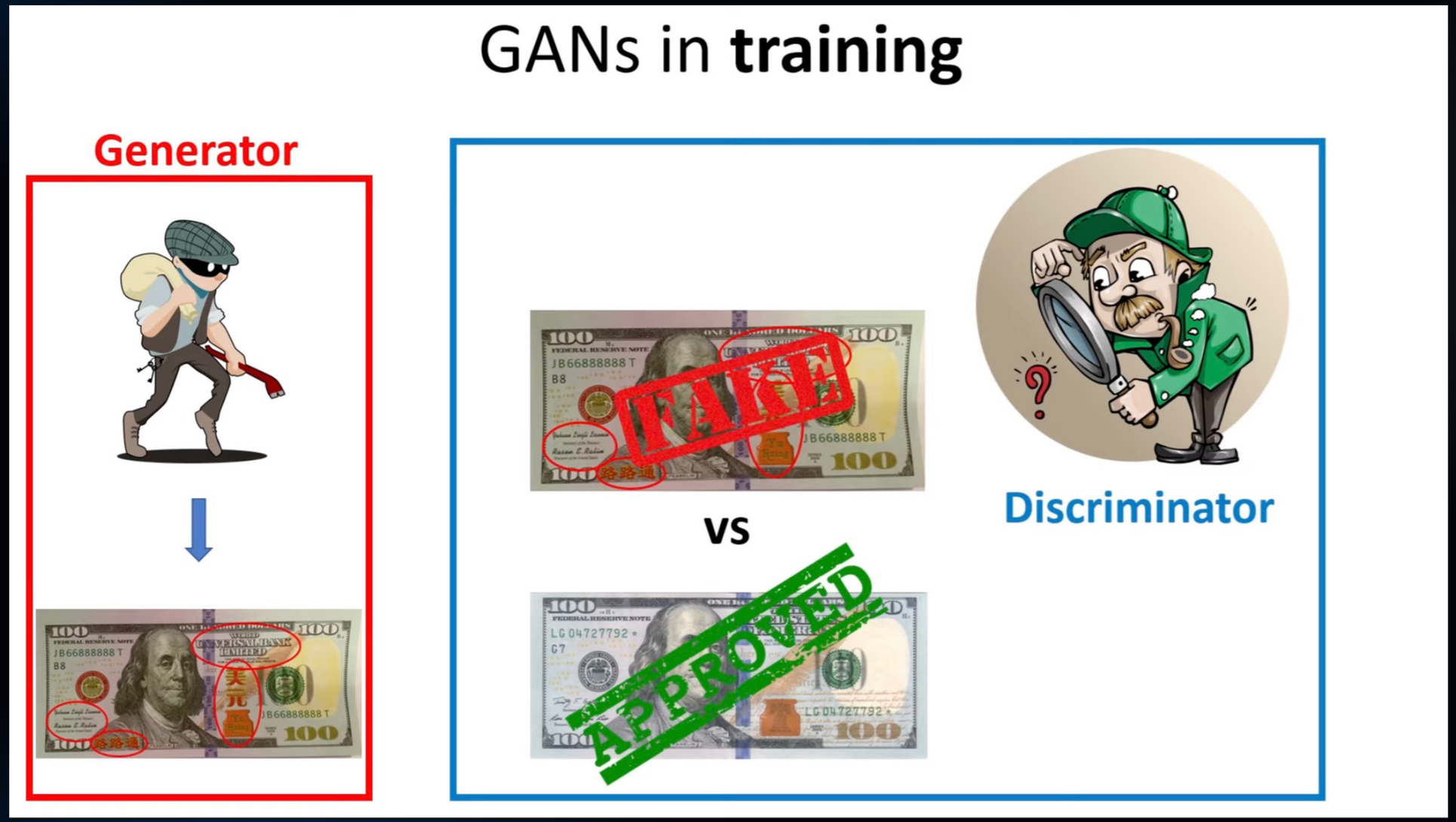
Las GAN constan de dos redes neuronales: un generador y un discriminador.

El generador crea muestras de datos falsos, mientras que el discriminador evalúa si las muestras son reales o falsas.

# GENERATIVE ADVERSARIAL NETWORKS (GAN)



# GENERATIVE ADVERSARIAL NETWORKS (GAN)

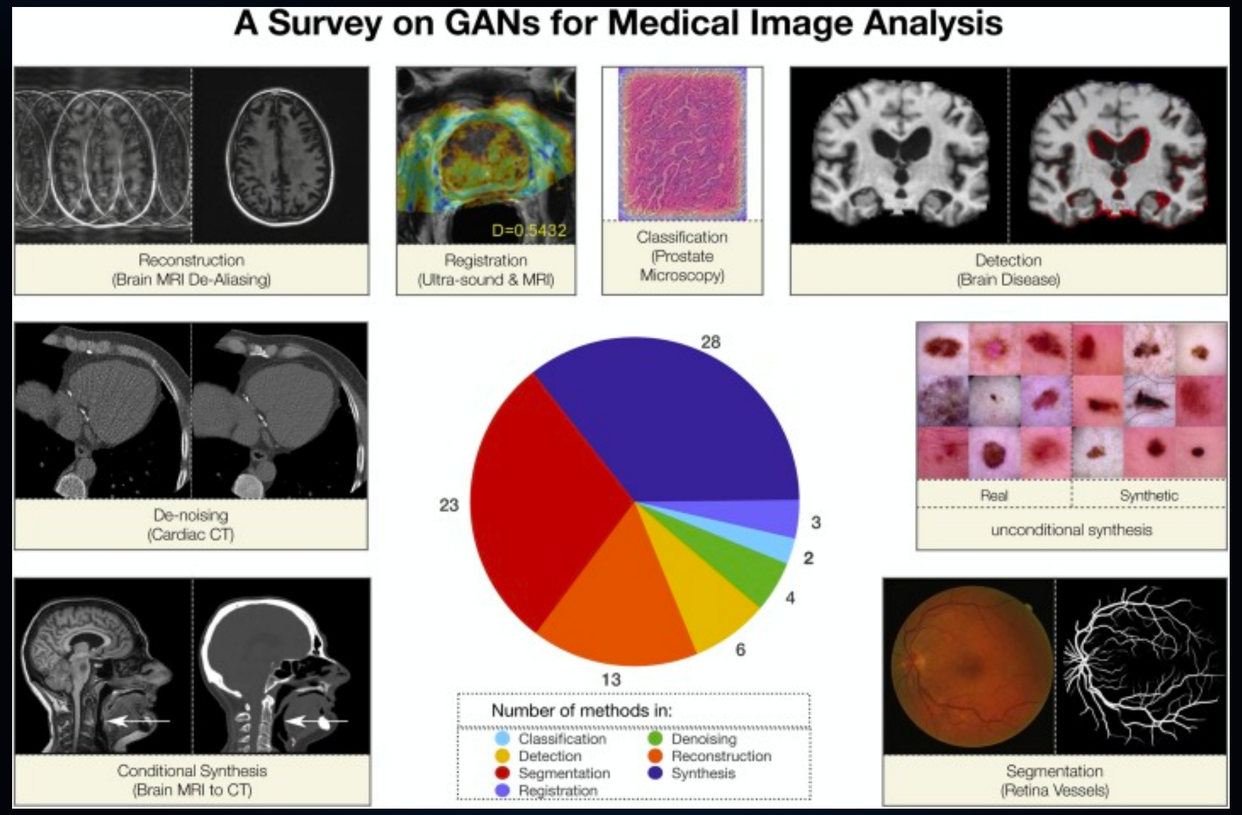


# GENERATIVE ADVERSARIAL NETWORKS (GAN)

El discriminador busca maximizar el error

El generador busca minimizar el error

# MAS POSIBLES APLICACION ES DE LAS (GAN)




- Data Augmentation (Media)
- Aplicaciones en el Campo de la medicina (privacidad)
- Semi-Supervised Learning

# Music Generation with GANs

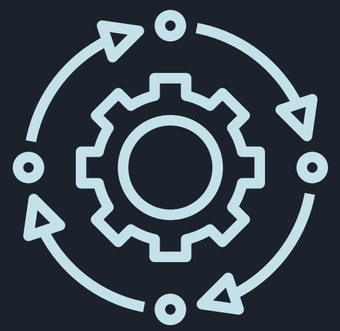
- Para generar musica con GANs se debe entrenar la red generadora con un dataset de archivos MIDI. Una vez finalizado el entrenamiento se devolvera un archivo MIDI que puede ser reproducido con sintetizadores digitales.
- El discriminador evaluara el archivo generado por los archivos MIDI basados en las similaridades que tenga con los datos de entrenamientos.
- Un gran desafio de la generacion de datos con redes GANs y que estos mantengan una estructura coherente comparando con los datos originales. Para solucionar este problema se desarrollaron GANs recurrentes y GANs tradicionales

# Music Generation with GANs

- Python tiene varia librerias para implementar GANs como TensorFlow, Keras y PyTorch.Estas librerias proveeran funciones ya existenes para la creacion de redes generadoras y redes discriminadoras y tambien modelos para el entrenameinto GAN.
- Adicionalmente, exsiten modelos GAN "open-source" en linea que pueden ser modificados para generacion de musica u otros datos.
- Los datos tienen que ser preprocesados extrayendo datos importantes de cada archivo, como ser pitch, duration y velocidad. existe gran cantidad de datasets online como ser Lakh MIDI Dataset o MuseScore.

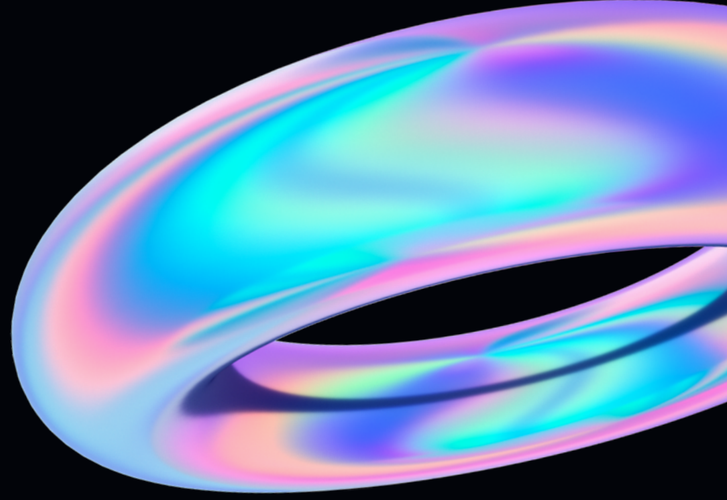
# Implementacion de GANs con Python

## Preprocesamiento

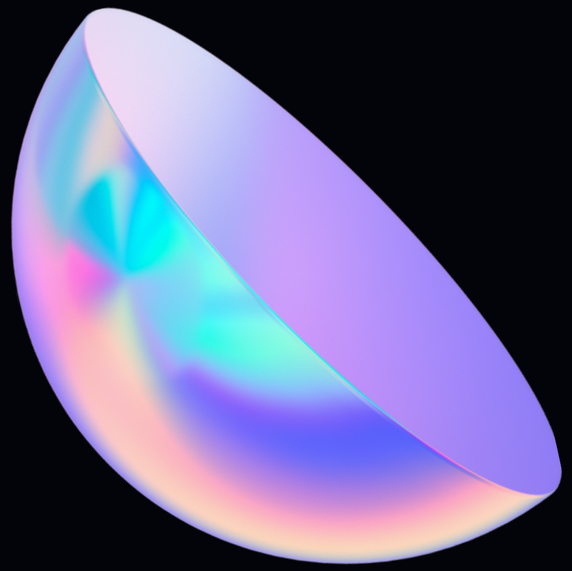


## Train

## Evaluate

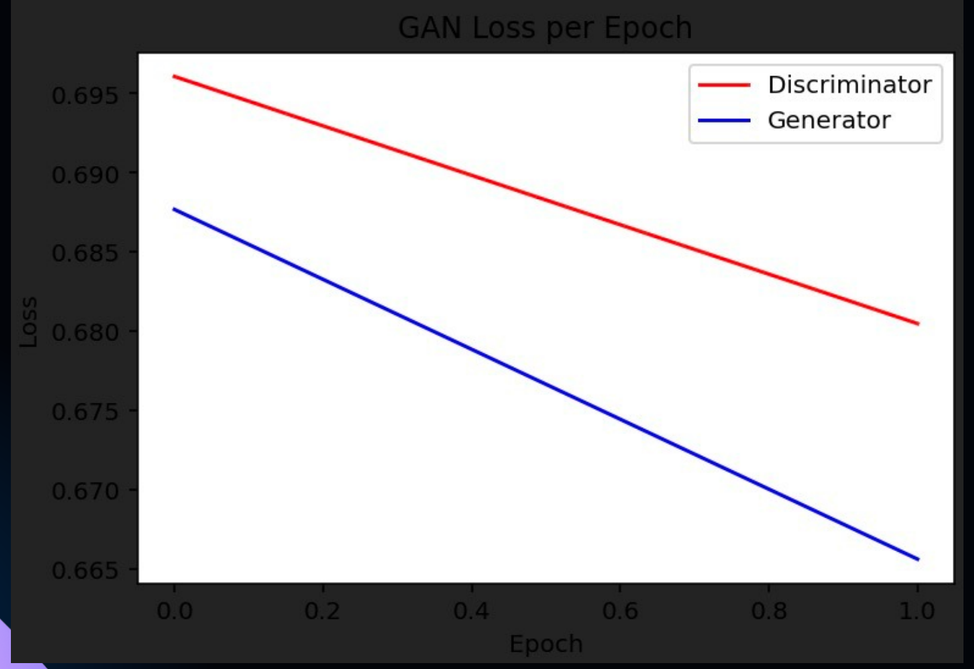


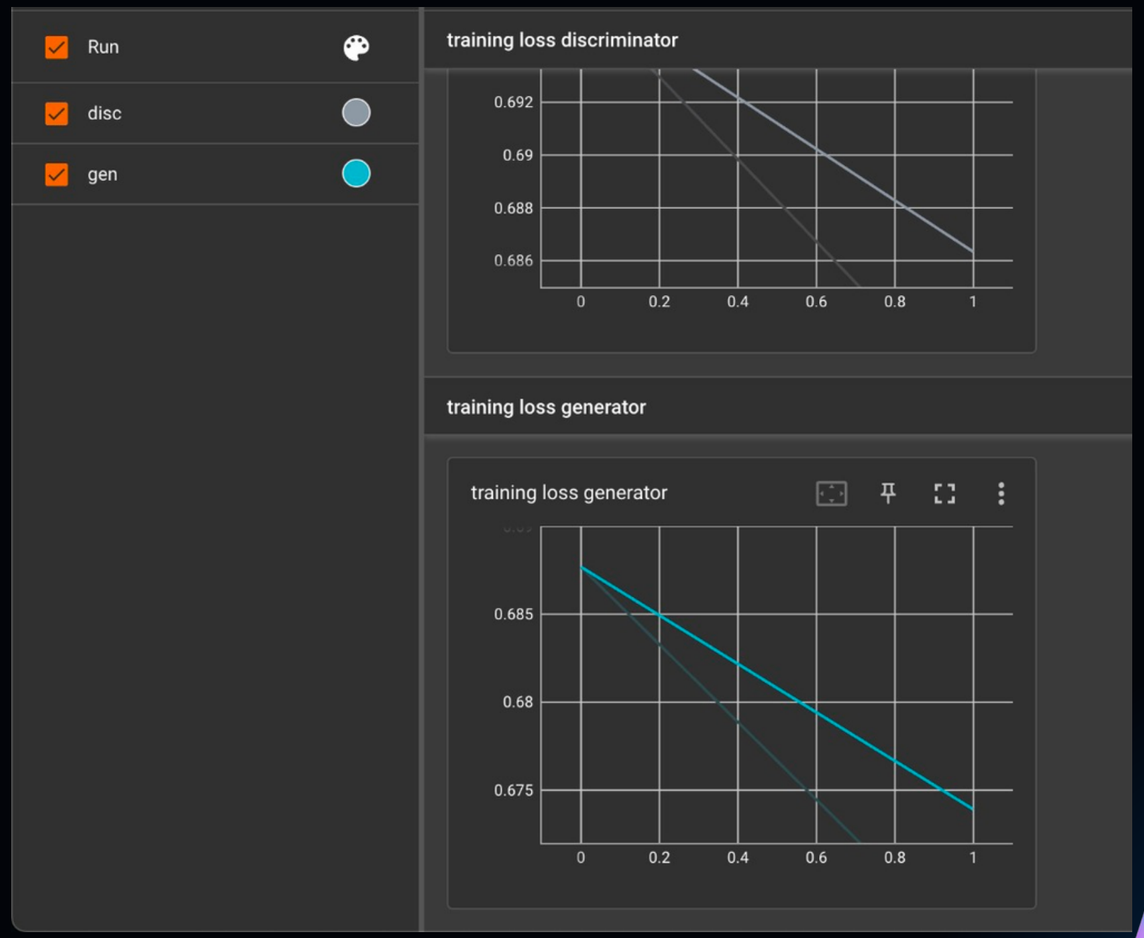
# EJEMPLOS APLICACIÓN



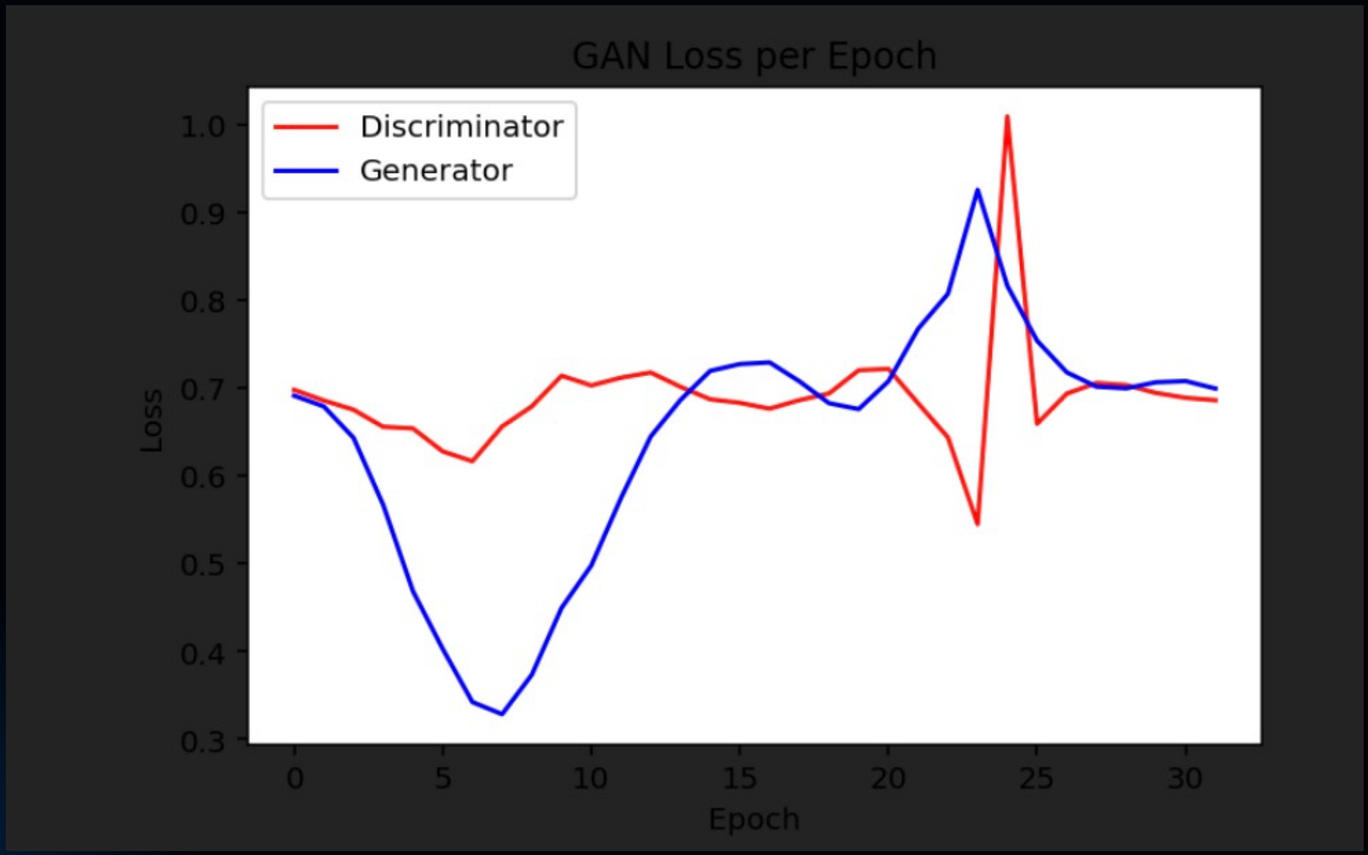
# 2 epoch

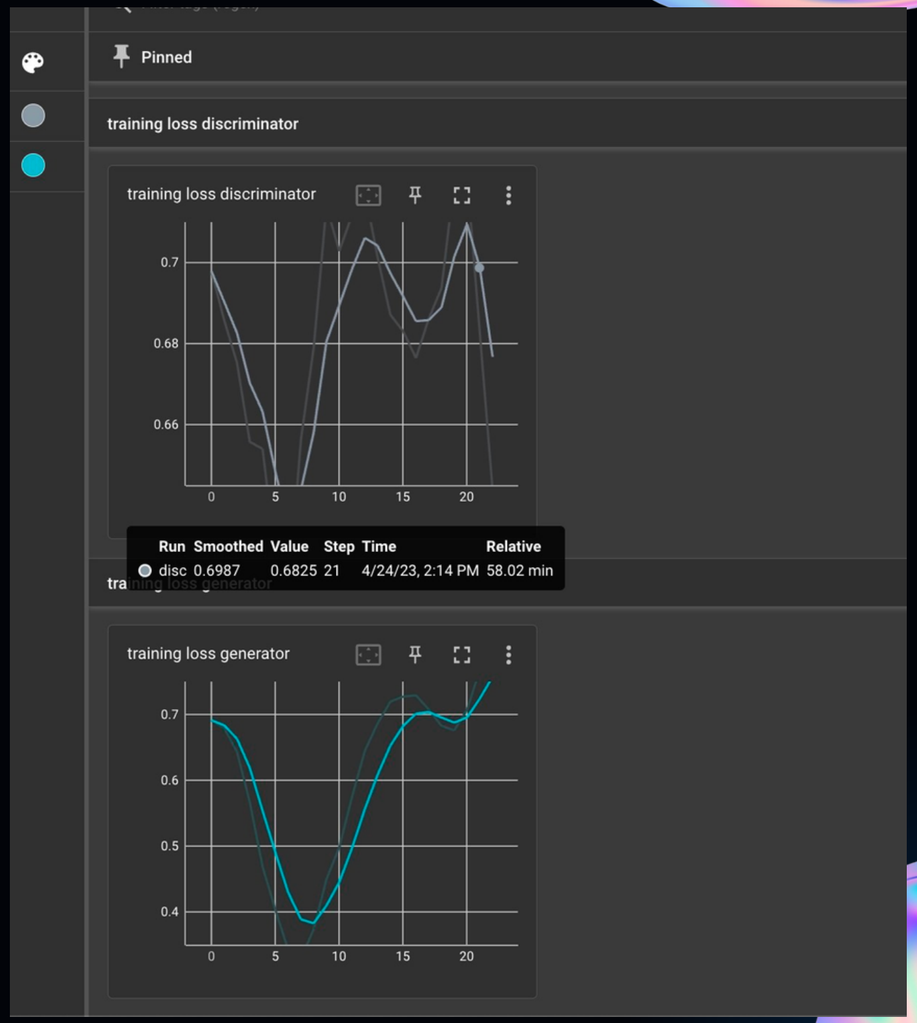
```
model.plot_loss()
```



# 32 epoch


GAN Loss per Epoch
Discriminator
Generator
Loss
Epoch
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
5
10
15
20
25
30

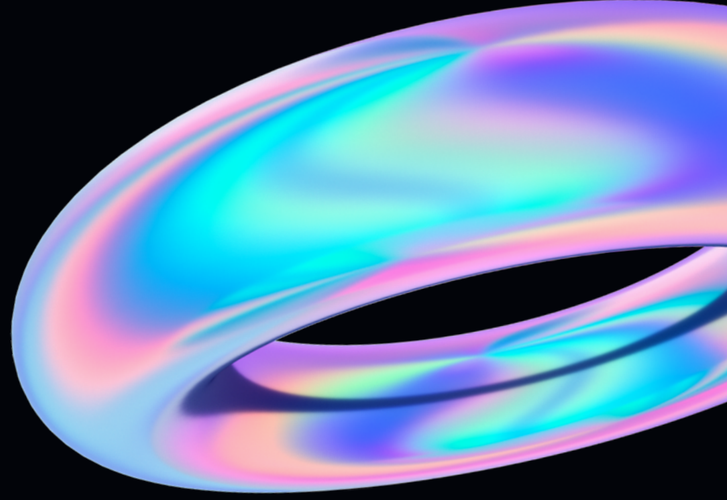

Pinned
training loss discriminator
training loss discriminator
0.7
0.68
0.66
0
5
10
15
20
Run Smoothed Value Step Time Relative
disc 0.6987 0.6825 21 4/24/23, 2:14 PM 58.02 min
training loss generator
0.7
0.6
0.5
0.4
0
5
10
15
20

# RESULTADOS

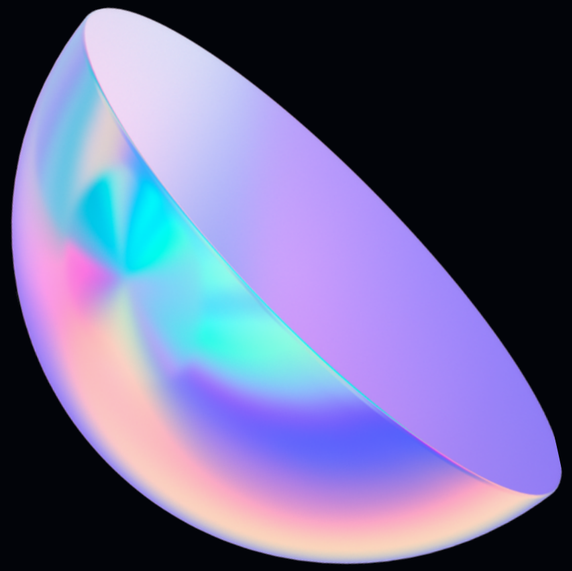
# 32 epochs

Piano roll - LABS
1
2
3
4
5
6
7
8
9
10
C7
C6
C5
C7
B6
A#6
A#6
A6
A6
G#6
G6
G6
G6
G6
F6
F6
F6
E6
E6
E6
D6
D6
C#6
C#6
C#6
C#6
C6
C6
B5
B5
A#5
A#5
A#5
A#5
A5
A5
A5
A5
A5
A5
G#5
G#5
G5
G5
G5
G5
G5
G5
G5
F5
F5
F5
F5
F5
F5
E5
E5
D#5
D5
D5
D5
D5
D5
C#5
C#5
C#5
C#5
C#5
C#5
C5
C5
C5
C5
B4
B4
B4
B4
A4
A4
A4
A4
A4
A4
A4
A4
G4
G4
G4
F#4
F#4
F4

# 32 epochs

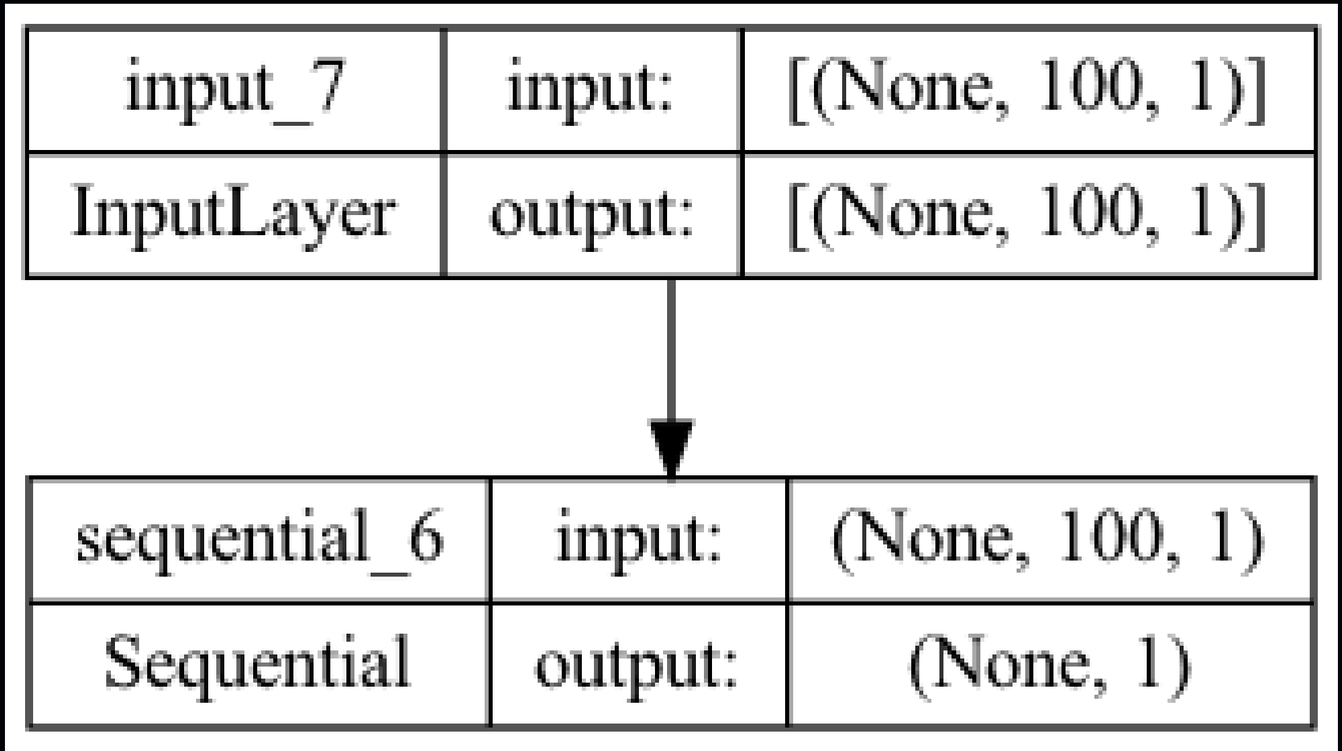


# EJEMPLOS APLICACIÓN



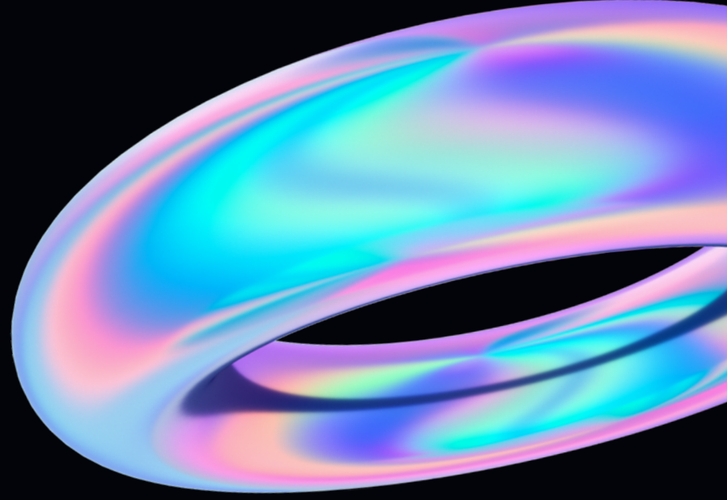
# arquitectura de al red

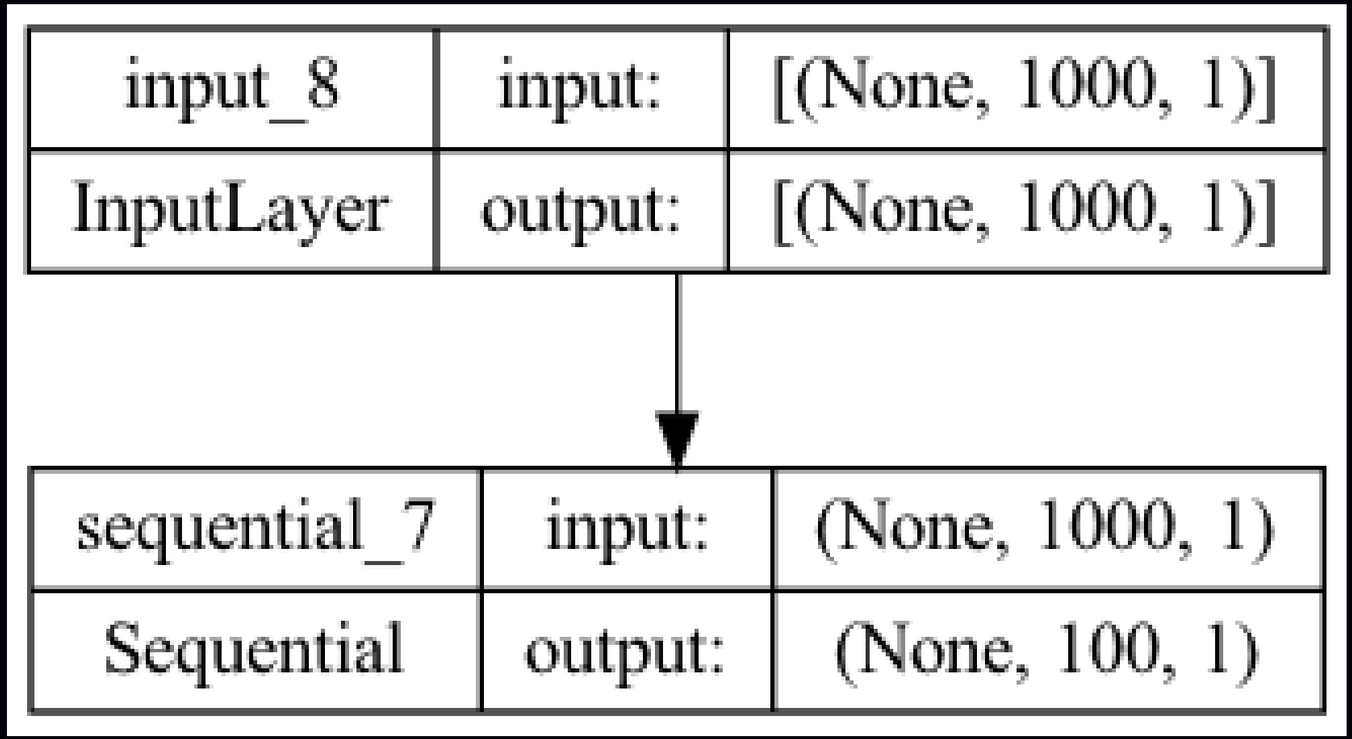
## discriminador



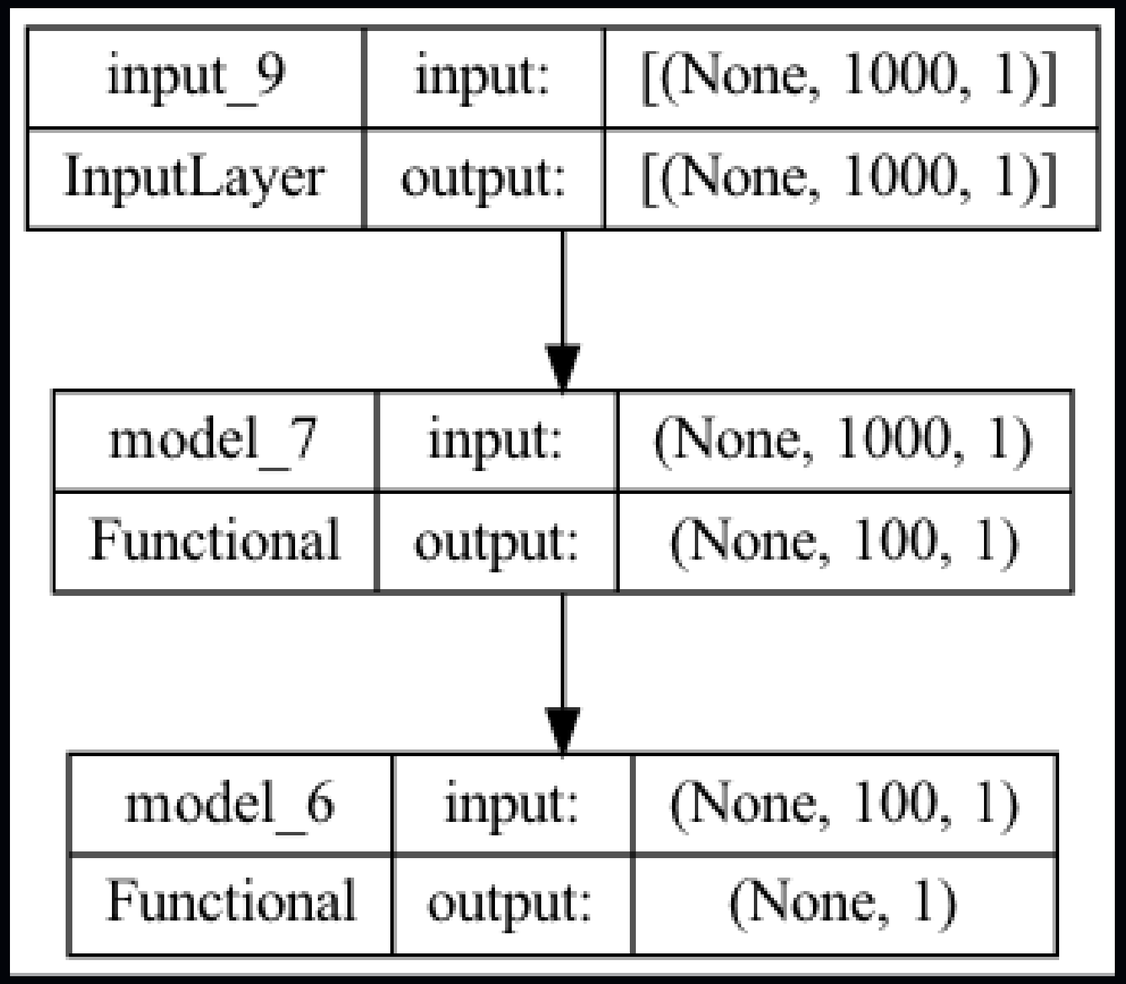
# arquitectura de al red

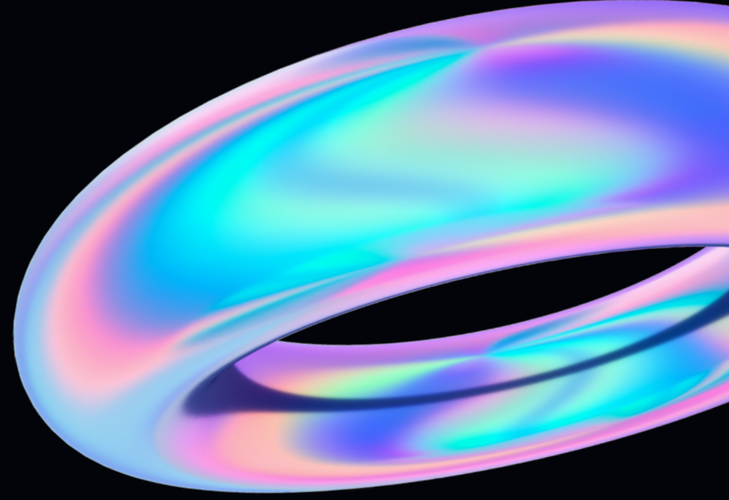
## generador



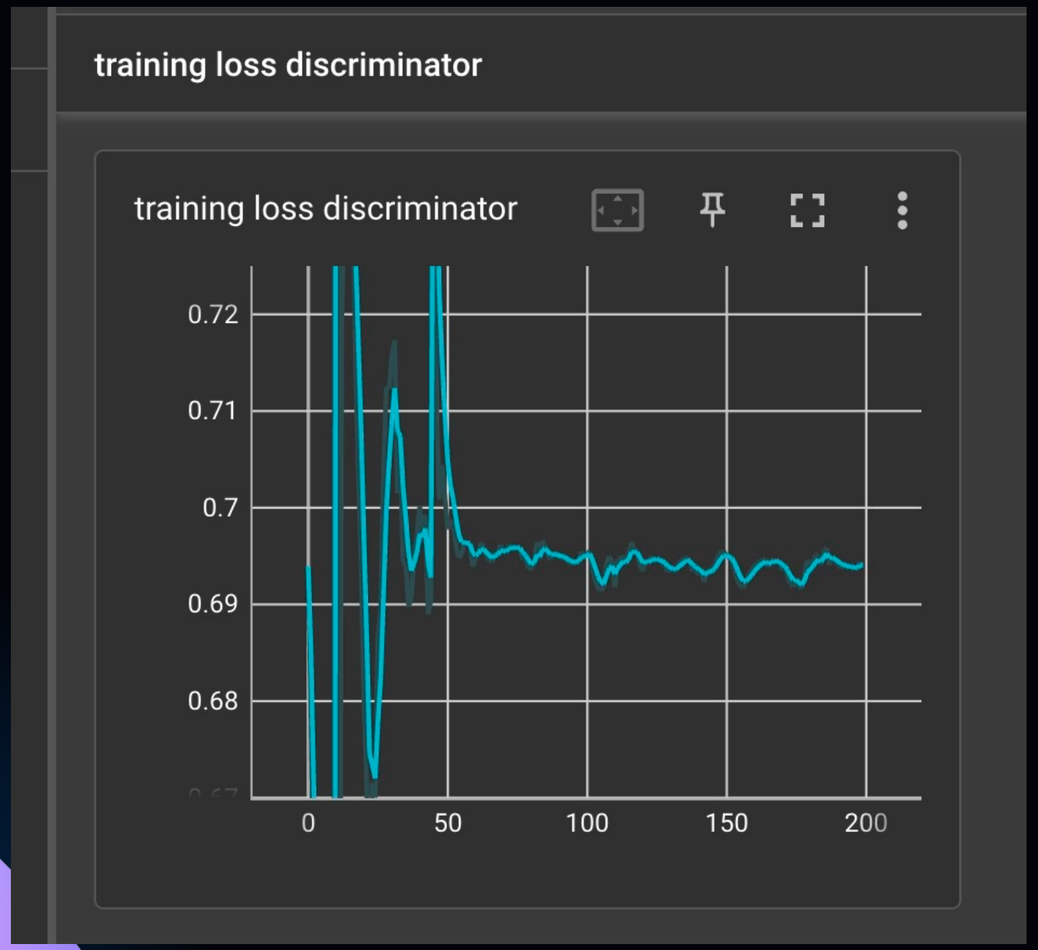


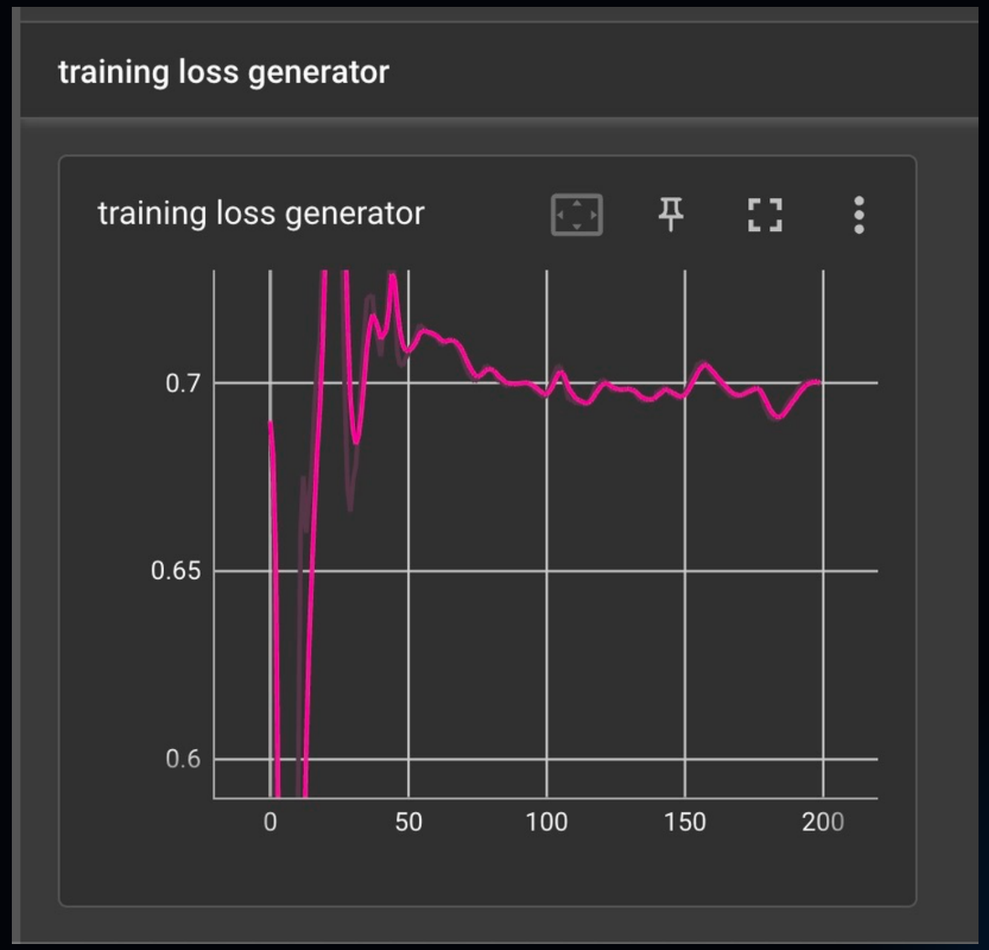
# red combinada



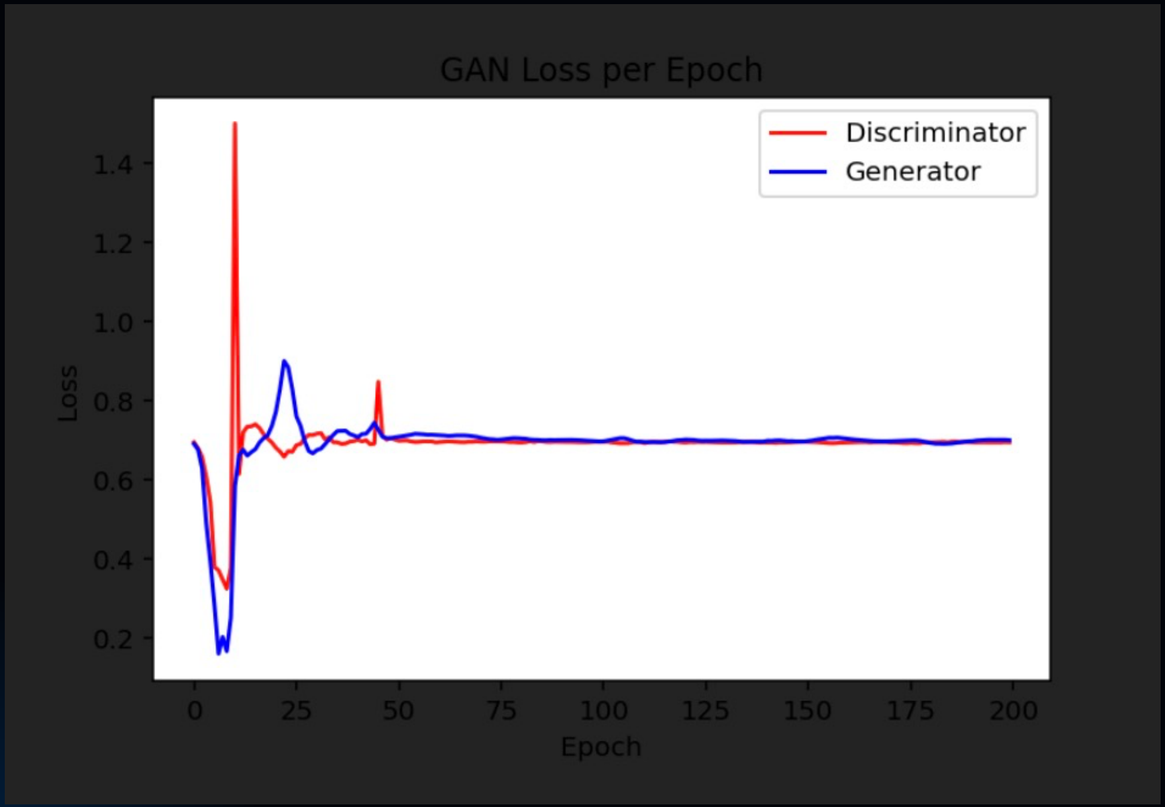



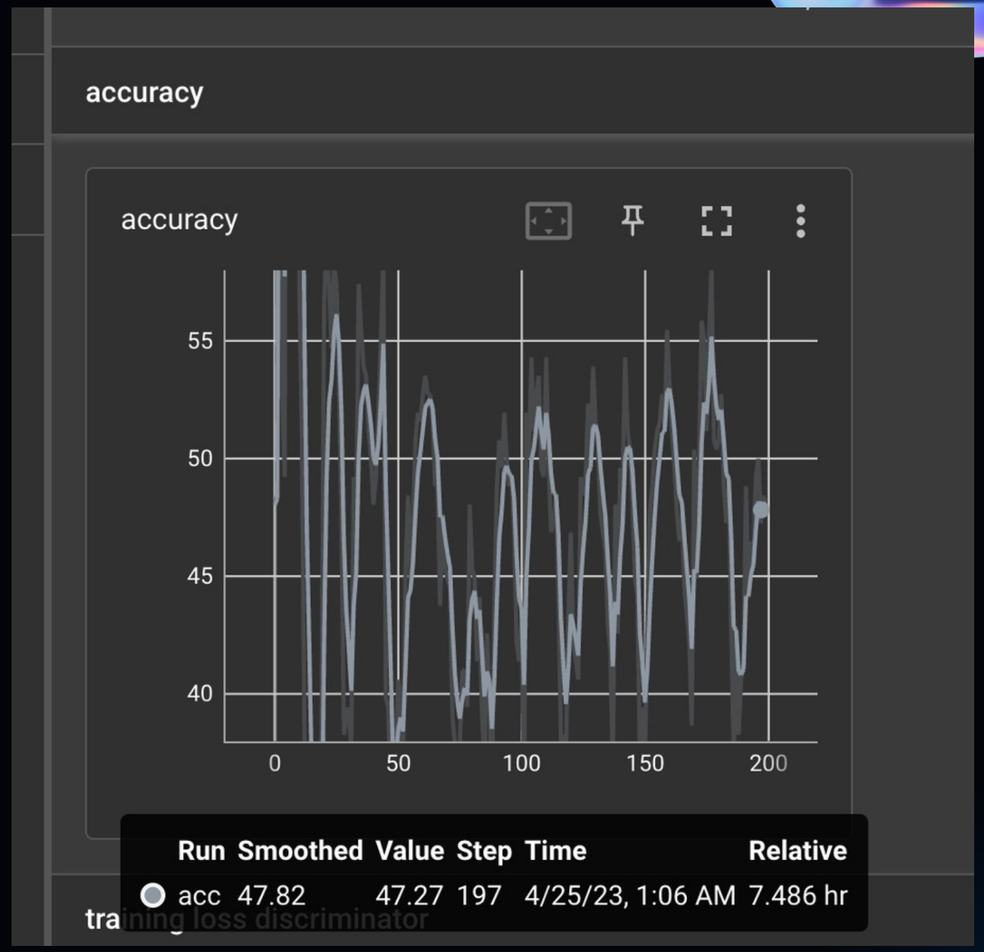
# 200 epoch


training loss discriminator
training loss discriminator
0.72
0.71
0.7
0.69
0.68
0
50
100
150
200


training loss generator
training loss generator
0.7
0.65
0.6
0
50
100
150
200

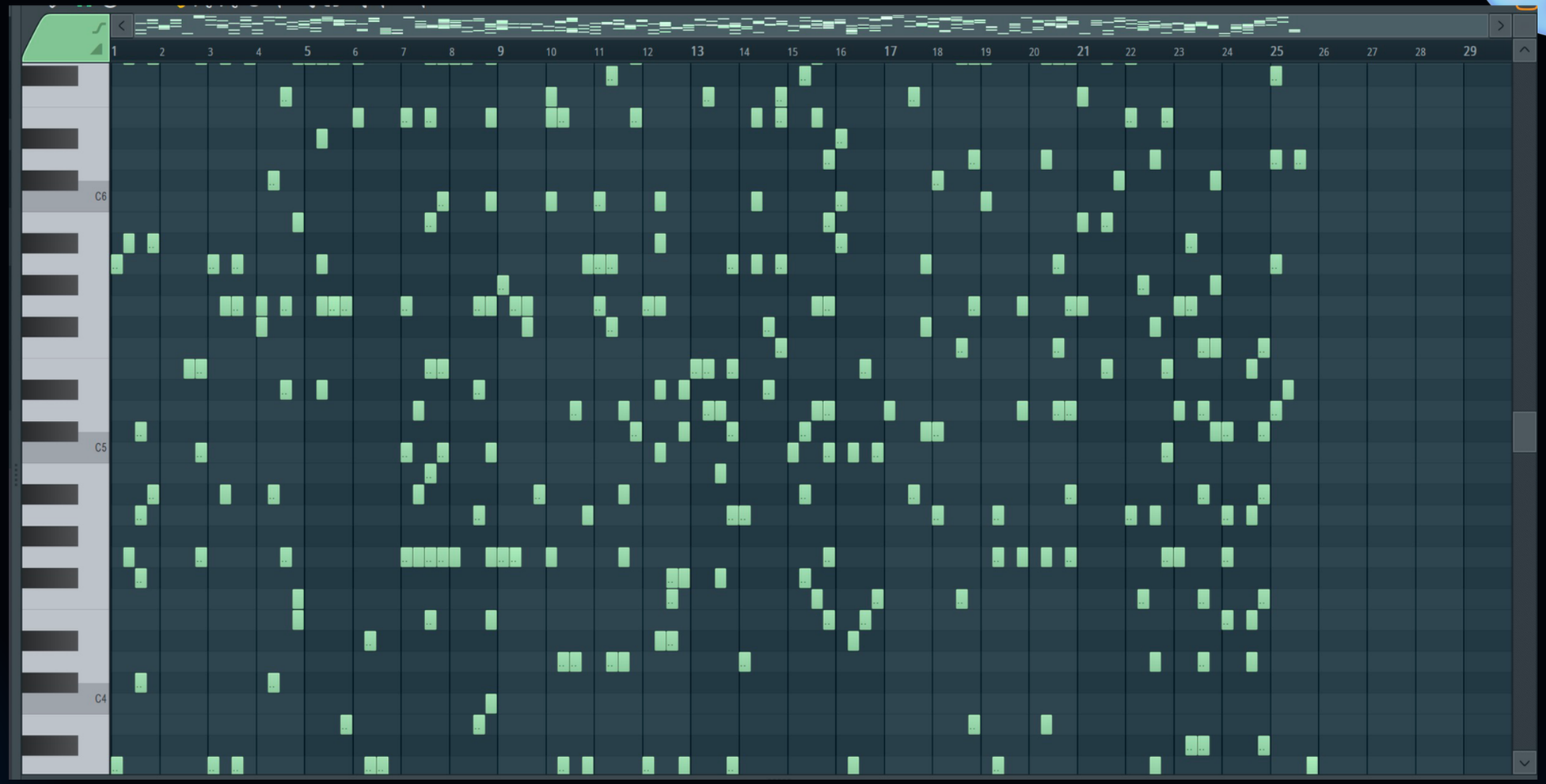
# 200 epoch


GAN Loss per Epoch
Discriminator
Generator
Loss
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0
25
50
75
100
125
150
175
200
Epoch


accuracy
accuracy
55
50
45
40
0
50
100
150
200
Run Smoothed Value Step Time Relative
acc 47.82 47.27 197 4/25/23, 1:06 AM 7.486 hr

# RESULTADOS

# Resultados 200 epoch



# Resultados 200 epoch

# Discusión

Se podria haber mejorado el modelo con las siguientes recomendaciones:

- Cambiar la función de pérdida utilizada en la GAN
- Cambiar los parámetros  de entrenamiento del modelo, optimizador, learning rate, momentum, entre otros.
- Añadir capas adicionales

El problema con la ultima recomendacion es el aumento drastico de tiempo para el entrenamiento del modelo.

Los resultados fueron buenos pero existe espacio para la mejora

# Discusión

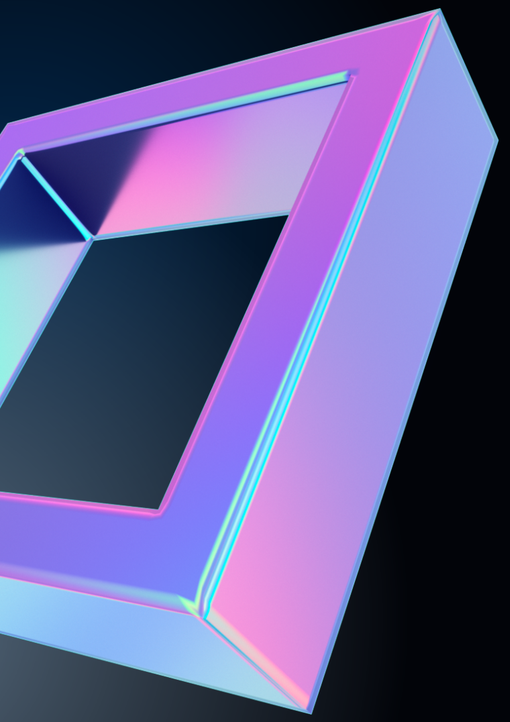
Se aprendió nuevos conceptos ya que antes solo conociamos que se pasaba información de manera unidireccional. La utilización de una LSTM bidireccional fue interesante porque permitió que la red tenga acceso tanto a la información pasada como a la futura en la secuencia de entrada.

la red neuronal generativa está  generando música que si es coherente pero no tan atractiva para los oyentes. La red aprendió algunas características de la música, pero aún no tiene la capacidad de crear piezas musicales completas y satisfactorias.

# Conclusion

El entrenamiento del modelo fue satisfactorio, pero existe un gran espacio a la mejora.
Los datos obtenidos de la red generadora mantienen una buena coherencia con la estructura de los datos originales y las predicciones de la red discriminadora suelen acertar si el input analizado es verdadero o falso.

La conclusión del entrenamiento del modelo es prometedora, ya que la red generadora pudo generar datos coherentes con la estructura de los datos originales y la red discriminadora pudo distinguir correctamente entre datos verdaderos y falsos. Sin embargo, hay espacio para mejorar el modelo en términos de calidad de los datos generados y la capacidad de la red para capturar patrones complejos de la música. Además, sería interesante experimentar con diferentes arquitecturas de red neuronal generativa y discriminativa para comparar los resultados y determinar cuál es la más efectiva en términos de generación de música de alta calidad.



# Gracias por la atención