



UNIVERSIDAD DE BUENOS AIRES  
FACULTAD DE CIENCIAS EXACTAS Y NATURALES  
DEPARTAMENTO DE COMPUTACIÓN

# Síntesis audiorrítmica en tiempo real con StyleGAN 3

Tomás Jaratz, LU 59/15

Buenos Aires, 2025

## RESUMEN

En este trabajo práctico se desarrolló un sistema capaz de traducir información sonora en tiempo real a vectores latentes compatibles con StyleGAN3, permitiendo una síntesis visual coherente con la dinámica auditiva. A través del uso de JACK y librosa, se capturó audio estéreo desde un sintetizador y se lo procesó mediante una red de filtros mel que extrae envolventes por banda. Estas envolventes se utilizaron para modificar de forma continua un vector latente de 512 dimensiones, el cual es enviado vía TCP a un visualizador externo que lo interpreta como semilla para generar imágenes.

El sistema cuenta con una puerta de silencio, suavizado dinámico, y mecanismos de ataque/decay por banda, permitiendo transiciones suaves y respuesta estable incluso ante variaciones abruptas en la señal. La modulación se complementa con un random walk que aporta variabilidad continua al espacio latente. Se diseñó también un servidor que reenvía los vectores al visualizador sin bloquear la ejecución en caso de que este no esté disponible.

El resultado es una herramienta funcional para sintetizar visuales reactivas al audio en tiempo real, útil tanto en contextos de performance como en instalaciones audiovisuales experimentales.

## Índice general

1.	Componentes del sistema . . . . .	1
2.	Pipeline de procesado de audio . . . . .	1
3.	Servidor de reenvío . . . . .	2
4.	Integración con StyleGAN 3 . . . . .	2
4.1.	Funciones clave del widget . . . . .	2
5.	Repositorio . . . . .	3

<sup>1</sup> El tamaño de bloque se eligió como compromiso entre latencia y resolución espectral.

9. **Suavizado (opcional):**  $z \leftarrow 0,85 z_{\text{prev}} + 0,15 z$ .
10. **Serialización y envío:**  $z \Rightarrow 512 \times \text{float32} = 2048$  bytes, little-endian, usando `sendall()` a 65431.

### 3. Servidor de reenvío

- Escucha en 65431. Acepta múltiples clientes de audio si fuera necesario.
- Mantiene un único socket persistente con el widget (puerto 65437).
- Si llega un paquete mientras hay otro pendiente, descarta el viejo y reenvía el más reciente, evitando latencia acumulada.
- Implementado con hilos `daemon` y `queue.Queue(maxsize=1)` para no bloquear.

### 4. Integración con StyleGAN 3

El widget recibe cada desplazamiento  $\mathbf{z}_{\text{off}} \in R^{512}$  y lo suma al vector semilla antes de pasar por el mapeo:  $\mathbf{z}' = \mathbf{z} + \mathbf{z}_{\text{off}}$ . De esta forma se logra un *morphing* suave entre imágenes sin recalcular la red.

#### 4.1. Funciones clave del widget

Si bien el archivo a ejecutar para correr la GAN es `visualizer.py`, los principales cambios para renderizar y modificar el espacio latente son `renderer.py` y `latent_widget.py`

*Manejo de la cola de latentes*

- `_reader_thread()`: hilo *daemon*:
  1. acepta la conexión entrante en 65437;
  2. acumula bytes en un buffer circular;
  3. extrae de a 2048 bytes y mantiene sólo el último vector en una `queue.Queue(maxsize=1)`.
- `fetch_translation_from_server()`: se invoca en cada frame GUI; si hay un vector nuevo lo filtra con un IIR de primer orden y lo pasa a StyleGAN vía `viz.args.z_offset`.

*Pipeline de síntesis (Renderer)*

1. `get_network()`: carga el `.pk1` de StyleGAN 3 con una red ya entrenada
2. `_render_impl()`:
  - a) genera/recibe semillas y construye  $\mathbf{z}$ ;
  - b) suma el desplazamiento `z_offset` proveniente del audio;
  - c) pasa por el mapeo ( $\mathbf{z} \rightarrow \mathbf{w}$ ) y compone  $\mathbf{w}$  final;
  - d) ejecuta la síntesis
  - e) normaliza, escala y convierte la imagen a `uint8`.

3. `run_synthesis_net()`: encapsula la síntesis y permite capturar capas internas.

En conjunto, estas funciones permiten actualizar de forma fluida los mapas de características de StyleGAN 3 cada vez que llega un nuevo vector latente del audio, produciendo imágenes sincronizadas con la señal.

## 5. Repositorio

El código completo, se encuentran en el siguiente repositorio:  
[https://github.com/khabalghoul/audio\\_synthesis](https://github.com/khabalghoul/audio_synthesis)