

---

# Clase 160 — Stable Diffusion XL + ControlNet en profundidad

Parte: 2 — Deep Learning · Fuente: Rombach et al. (2022) SD + Podell et al. (2023) SDXL + Zhang et al. (2023) ControlNet. Duración estimada: 90 min.

## Clase 160 — Stable Diffusion XL + ControlNet en profundidad

Parte: 2 — Deep Learning · Fuente: Rombach et al. (2022) SD + Podell et al. (2023) SDXL + Zhang et al. (2023) ControlNet. Duración estimada: 90 min.

### Objetivo

Dominar Stable Diffusion XL (Stability AI 2023) en producción: pipeline completo (text encoder dual, U-Net, VAE), schedulers modernos (DPM-Solver++, Euler ancestral, UniPC), CFG (Classifier-Free Guidance), prompt weighting. Combinar con ControlNet para condicionamiento espacial (Canny, depth, pose, segmentation, lineart). Conocer Flux (Black Forest Labs 2024) como sucesor open-source.

### Resultados de aprendizaje

Al finalizar, el estudiante podrá:

- Cargar SDXL: `StableDiffusionXLPipeline.from_pretrained('stabilityai/stable-diffusion-xl-base-1.0')`.
- Aplicar refiner opcional para detalle final.
- Usar schedulers distintos y entender trade-off speed/quality.
- Agregar ControlNet (canny, depth, openpose) sobre SDXL.
- Aplicar LoRA para estilo custom (`pipe.load_lora_weights('path')`).
- Reconocer Flux y SD3 como sucesores.

### Temas

- Pipeline SDXL: `text_encoder_1` (CLIP-L), `text_encoder_2` (CLIP-G), U-Net 2.6B params, VAE.
- Schedulers: DDIM (50 steps), DPM-Solver++ (20 steps), Euler ancestral (creative), UniPC (10-20 steps).
- CFG: `guidance_scale=7.5` default; > 12 over-fitting al prompt.
- Negative prompts.
- Refiner (paso opcional adicional).
- ControlNet variantes: Canny, Depth, OpenPose, Scribble, Lineart, MLSD, Tile.

### Definiciones y características

- SDXL: 2.6B params U-Net, latent space  $128 \times 128 \times 4$  desde imagen  $1024 \times 1024$ .
- Dual text encoder: CLIP-L (768d) + CLIP-G (1280d) concatenados.
- CFG:  $\text{output} = \text{uncond} + \text{scale} \cdot (\text{cond} - \text{uncond})$ .
- Negative prompt: lo que NO querés (ugly, blurry, low quality).
- ControlNet: red adicional, freezeada o trained, que inyecta condicionamiento espacial.
- LoRA: rank-low adapter, ~50-200 MB, para estilos.
- Flux 1: sucesor open-source SDXL (2024), mejor calidad y prompt following.

### Dataset / recursos

- HuggingFace: `stabilityai/stable-diffusion-xl-base-1.0`, ControlNets en `diffusers/controlnet-*-sdxl-1.0`.

- Librerías: diffusers, transformers, accelerate, controlnet\_aux.

## Ejercicios

1. SDXL básico: prompt → imagen 1024<sup>2</sup>. Comparar schedulers (DPM-Solver++, Euler a, UniPC).
2. CFG sweep: guidance\_scale {3, 7.5, 12, 18}. Ver trade-off creativity/fidelity.
3. Negative prompt: agregar "blurry, watermark, low quality" y comparar.
4. ControlNet Canny: extraer Canny de una foto, generar variante manteniendo estructura.
5. LoRA estilo: cargar un LoRA de estilo (CivitAI), aplicar. Variar cross\_attention\_kwargs={'scale': 0.8}.

## Homework verificable

Pipeline visual: SDXL + ControlNet Canny + LoRA estilo:

1. Foto base → Canny.
2. SDXL + ControlNet, prompt + LoRA estilo.
3. Generar 4 variantes; comparar.
4. Reportar tiempo y memoria.

Criterio de aceptación: estructura preservada por Canny; estilo del LoRA visible; outputs de buena calidad.

## Errores comunes

Síntoma / mensaje	Causa y cómo arreglar
OOM en GPU 8 GB	Modelo grande. Fix: pipe.enable_attention_
Outputs raros, deformados	CFG demasiado alto. Fix: guidance_scale=5-
ControlNet no obedece estructura	controlnet_conditioning_scale muy bajo. Fi
Manos deformadas	Bug clásico de difusión. Fix: LoRA especif
LoRA no se siente	Scale muy bajo. Fix: cross_attention_kwarg

## Preguntas frecuentes

SDXL o Flux?

Flux (2024) mejor calidad y prompt following. SDXL más maduro, más LoRAs/ControlNets disponibles. Ambos viables.

SD3 dónde está?

Stability AI lo publicó pero con licencia restrictiva → comunidad migró a Flux. SD3.5 medium open license.

Refiner vale la pena?

A veces. Toma más tiempo. Probar con vs sin para tu caso.

Cómo entreno LoRA para SDXL?

diffusers examples + dataset de 10-30 imágenes + script de training. 1-2 horas GPU.

Inpainting?

StableDiffusionXLInpaintPipeline. Modificar regiones específicas con máscara.

## Referencias

- Podell et al. (2023), SDXL.
- Zhang et al. (2023), ControlNet, ICCV.
- diffusers docs.
- Flux (Black Forest Labs).
- CivitAI — comunidad LoRA/checkpoints.

## Siguiente clase

Clase 161 — RL: aprendizaje por recompensa, Gymnasium (Farama)

## Apéndice: notebook (primer bloque)

Fallback CPU-friendly: implementamos forward diffusion sobre imagen 32×32 sintética, un denoiser MLP, y simulamos ControlNet con Canny edges.

```
import numpy as np
import matplotlib.pyplot as plt
rng = np.random.default_rng(42)

DIFFUSERS_OK = False
try:
    import diffusers # noqa
    DIFFUSERS_OK = True
    print('diffusers disponible (pero requiere GPU para SDXL real)')
except Exception:
    print('diffusers no disponible → usando fallback DDPM toy')
```

## Archivos complementarios

- notebook.ipynb