

Creando un CAPTCHA Inteligente

Cómo un Generador (GAN) y un Lector (CNN) colaboran para crear desafíos visuales únicos



Zapatilla



Vestido



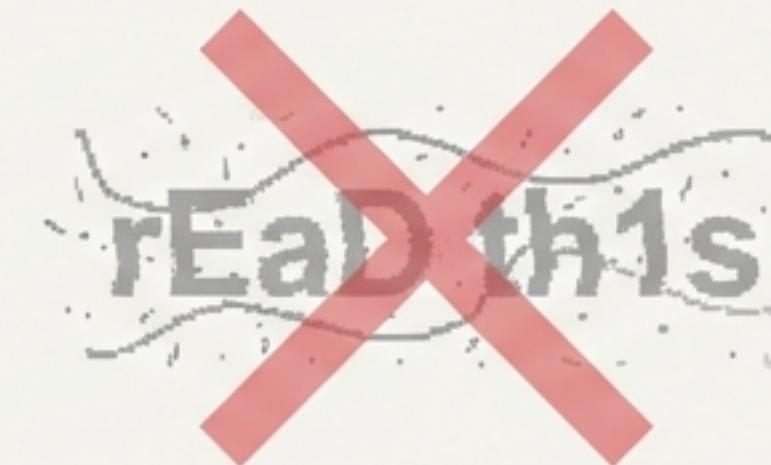
Bolso



Camisa

Más Allá de las Letras Distorsionadas

Los CAPTCHAs tradicionales son a menudo repetitivos y vulnerables a sistemas de OCR avanzados. Nuestra propuesta: un sistema dinámico que no solo verifica a los humanos, sino que también los involucra con un desafío visual único en cada ocasión.



Un Dúo de Redes Neuronales

El corazón de nuestro sistema son dos especialistas de IA, cada uno con una misión distinta pero complementaria. Los entrenamos por separado para que luego puedan colaborar.



El Lector: El Identificador

Una Red Neuronal Convolucional (CNN) experta en clasificar prendas de vestir. Su trabajo es identificar qué es cada imagen.



El Generador: El Creador

Una Red Generativa Antagónica (GAN) entrenada para crear imágenes de ropa nuevas y realistas a partir de la nada.

Conozcamos al Lector: Un Experto en Moda

Misión

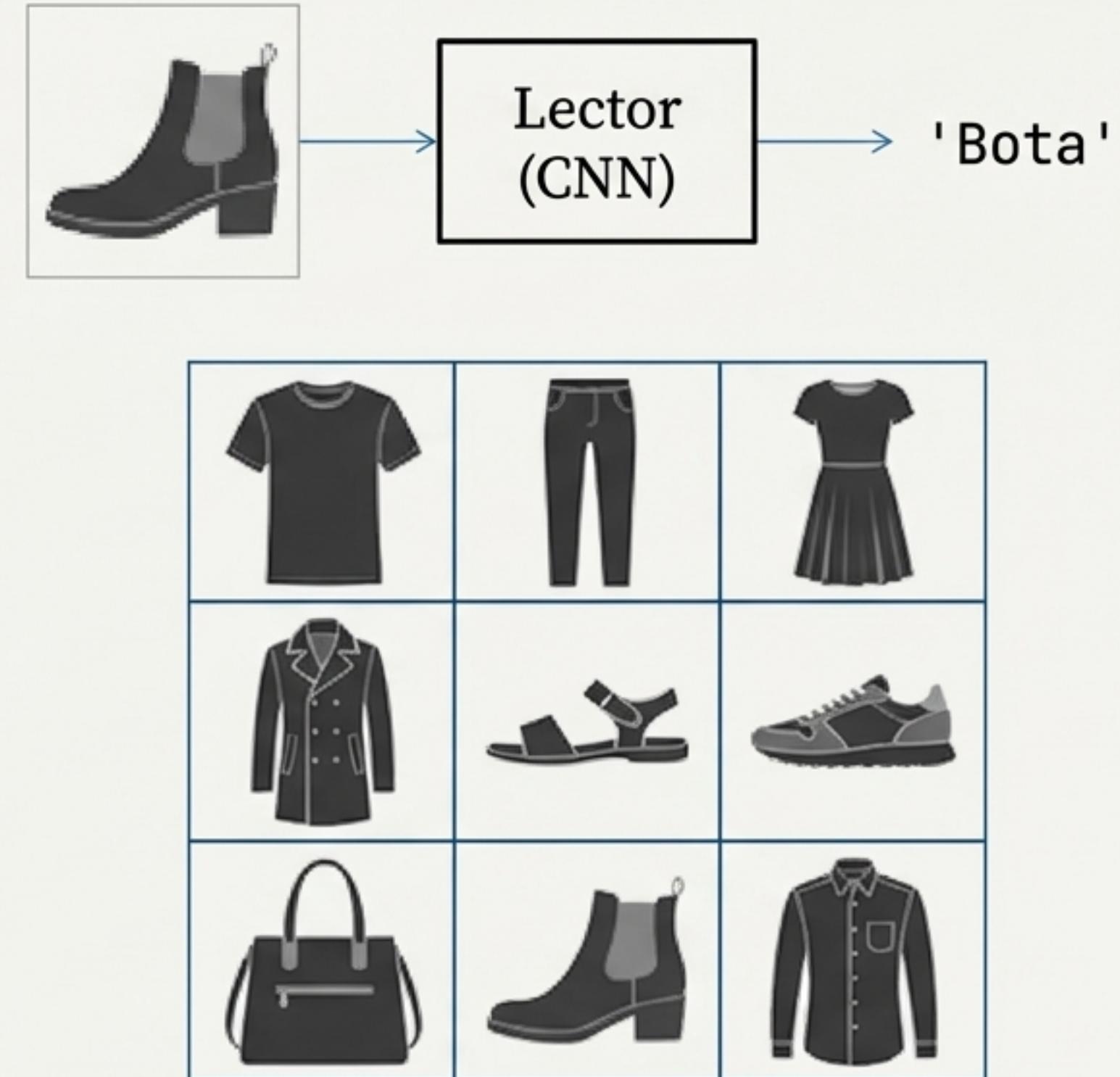
Su único propósito es observar una imagen de 28x28 píxeles y determinar a cuál de las 10 categorías de ropa pertenece.

Tecnología

Es una Red Neuronal Convolucional (CNN) clásica, una arquitectura probada para la clasificación de imágenes.

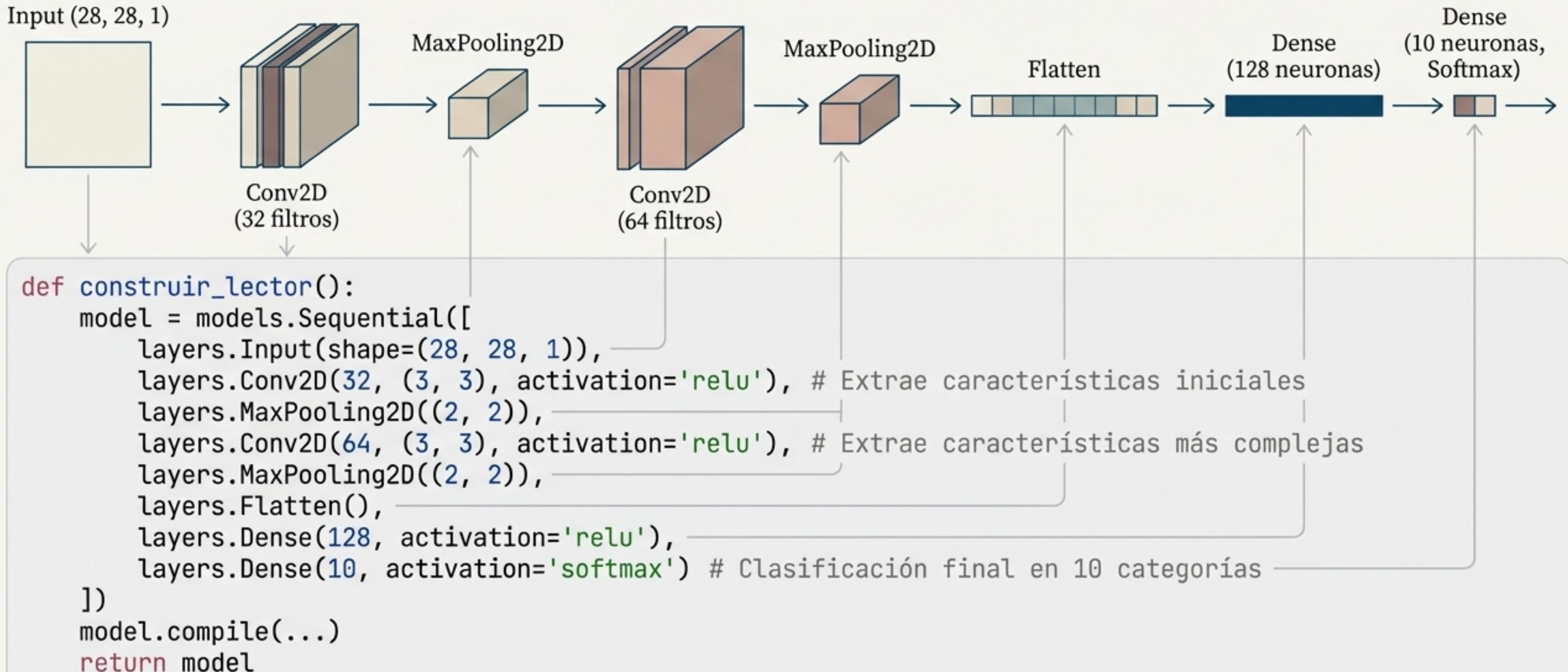
Dataset de Entrenamiento

Fashion MNIST, una colección de 60,000 imágenes de entrenamiento que le enseña a distinguir entre Camisetas, Pantalones, Bolsos, etc.



La Arquitectura del Lector

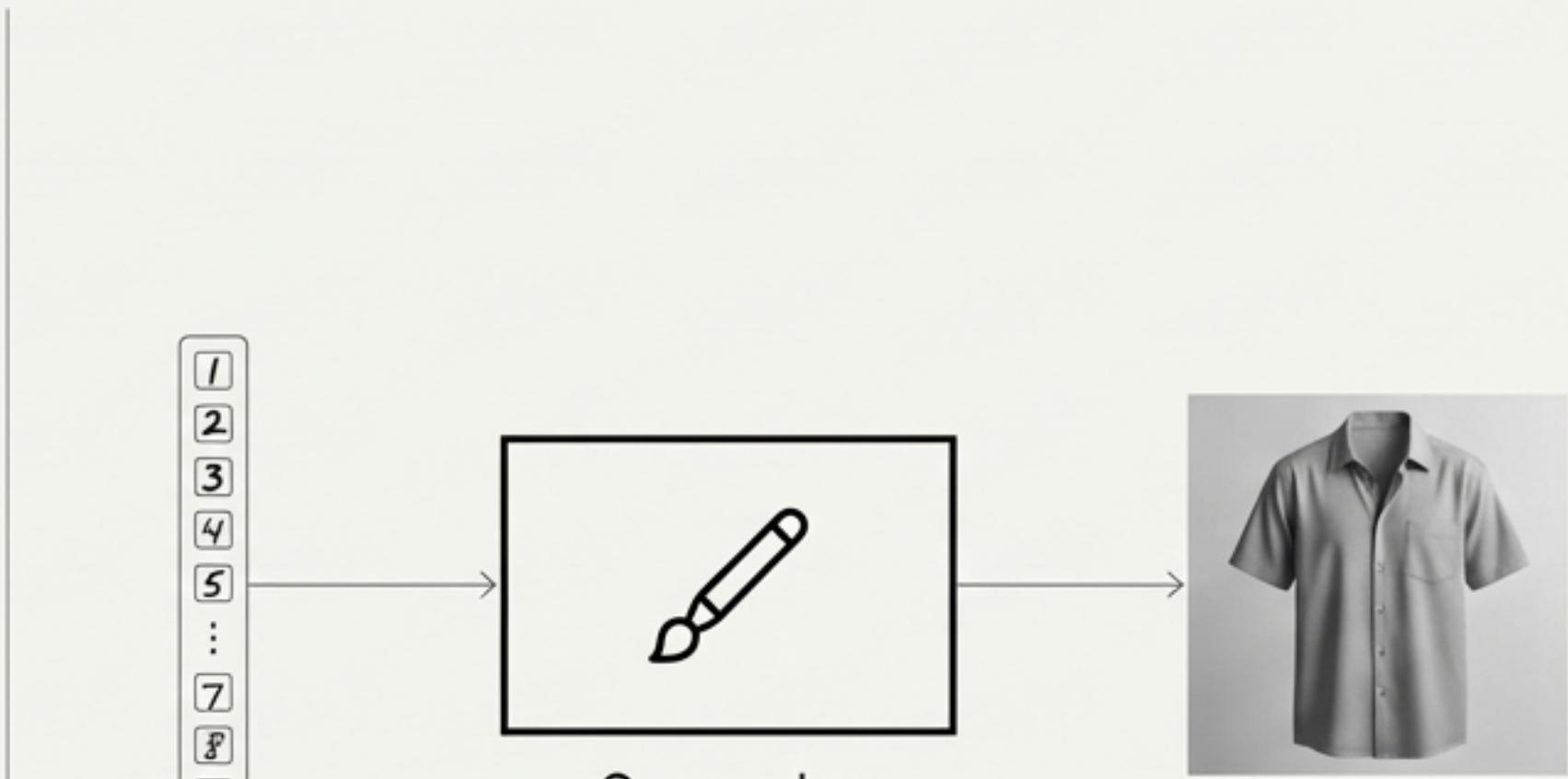
La estructura del Lector es una secuencia de capas que extraen características de la imagen (bordes, texturas) y luego las clasifican.



El Generador: Mador: Un Artista Digital

Misión: Su objetivo es crear una imagen convincente de una prenda de vestir a partir de un vector de ruido aleatorio ($Z_SIZE = 100$). Esencialmente, aprende a dibujar.

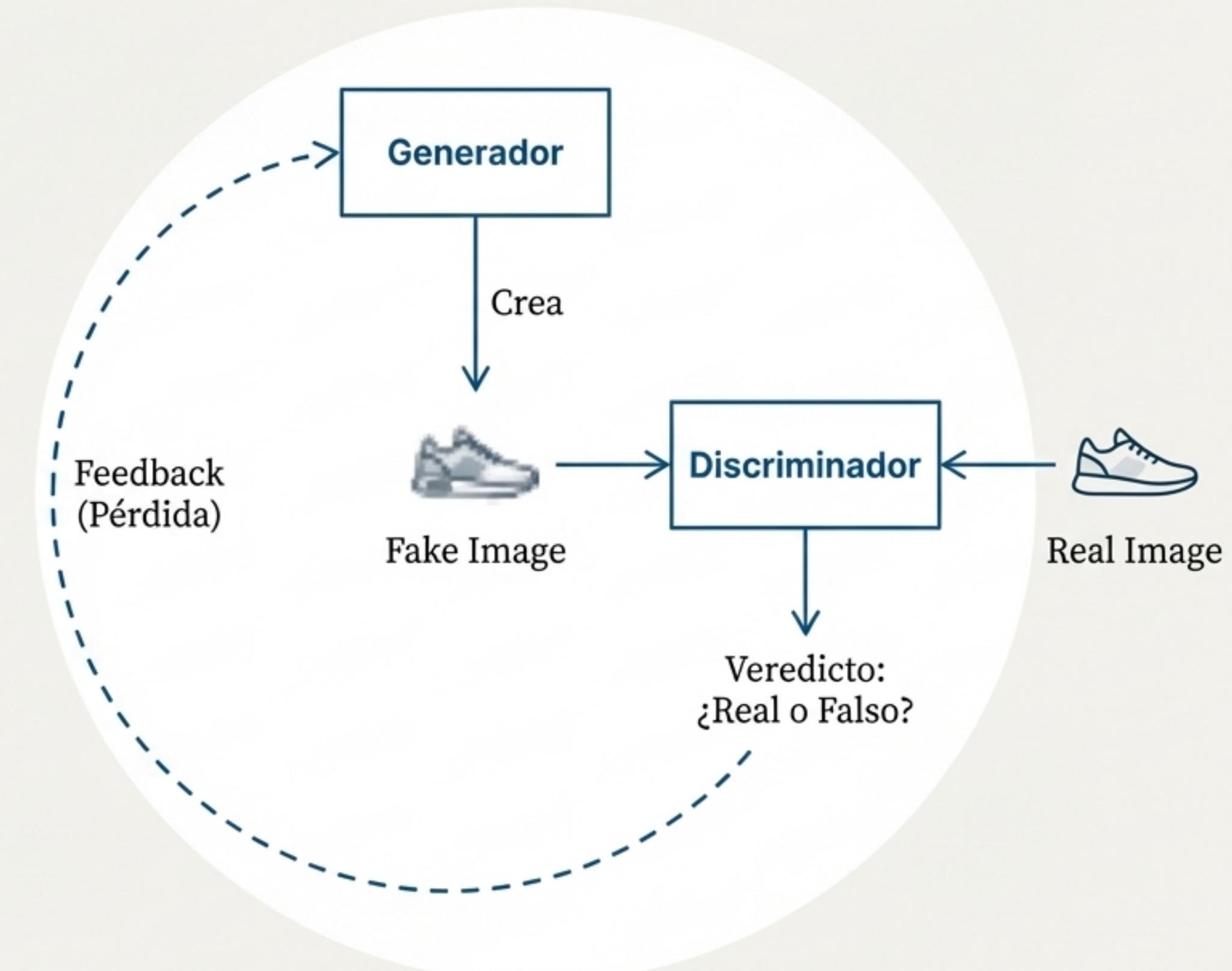
Tecnología: Es una Red Generativa Antagónica (GAN), específicamente una arquitectura tipo DCGAN (Deep Convolutional GAN). No trabaja solo; se entrena en un duelo constante.



El Duelo Interno: Generador vs. Discriminador

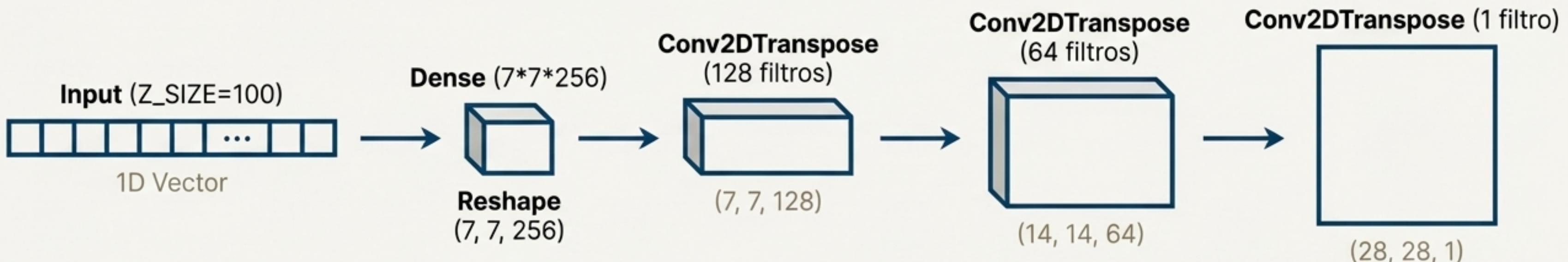
Para que el Generador aprenda, necesita un rival: el Discriminador. Su entrenamiento es una competencia:

- 1. El Generador** crea una imagen "falsa" y se la muestra al Discriminador.
- 2. El Discriminador** recibe tanto imágenes falsas como imágenes reales del dataset y debe aprender a distinguirlas.
3. El Generador usa el feedback del Discriminador para mejorar, intentando engañarlo en el siguiente intento.



La Arquitectura del Generador

A diferencia del Lector que comprime la información, el Generador la expande. Comienza con un vector de 100 dimensiones y lo ‘descomprime’ progresivamente hasta formar una imagen de 28x28.



```
def construir_generador():
    model = tf.keras.Sequential([
        layers.Dense(7 * 7 * 256, ..., input_shape=(Z_SIZE,)),
        layers.BatchNormalization(),
        layers.LeakyReLU(),
        layers.Reshape((7, 7, 256)),
        # Capas que 'des-convolucionan' el vector a una imagen
        layers.Conv2DTranspose(128, (5, 5), strides=(1, 1), ...),
        layers.Conv2DTranspose(64, (5, 5), strides=(2, 2), ...),
        layers.Conv2DTranspose(1, (5, 5), strides=(2, 2), ..., activation='tanh')
    ])
    return model
```

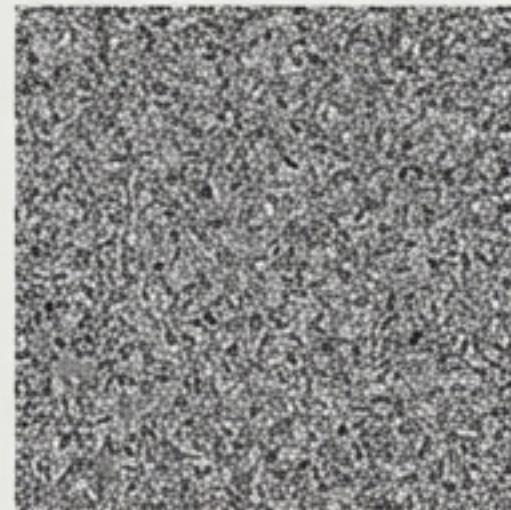
Clave para 'dibujar' y
expandir la imagen

El Proceso de Entrenamiento

Ambos modelos son entrenados de forma independiente antes de poder colaborar. Cada uno tiene sus propios requisitos.

Lector (Clasificador)

- **Épocas:** 10
- **Batch Size:** 64
- **Normalización de Imagen:** 0 a 1
- **Objetivo:** Maximizar la precisión en la clasificación.



Época 1



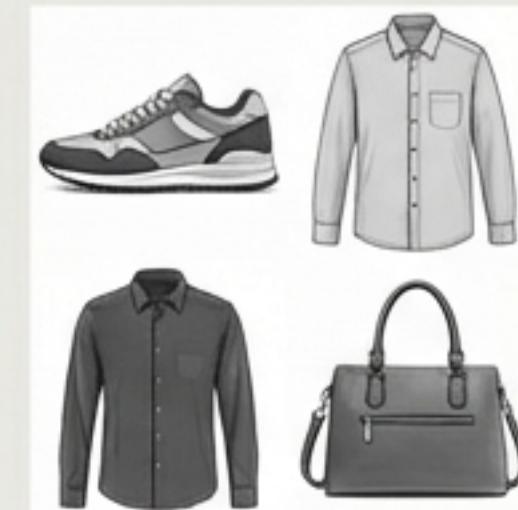
Época 20

Generador (GAN)

- **Épocas:** 100
- **Batch Size:** 64
- **Normalización de Imagen:** -1 a 1
(requerido por la activación tanh)
- **Objetivo:** Engañar al Discriminador de forma consistente.



Época 50



Época 100

La Colaboración: Generando el CAPTCHA

Una vez entrenados, los modelos trabajan en secuencia para crear el desafío final. El Generador crea las imágenes y el Lector, que conoce la 'verdad', proporciona las etiquetas correctas.

Paso 1: Ruido

Se crea un tensor de ruido aleatorio.

```
tf.random.normal([4, Z_SIZE])
```



1

2

3

4

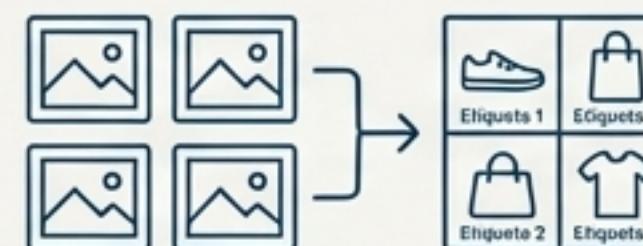
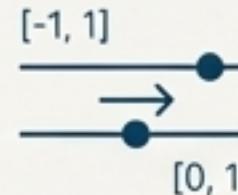
5



Paso 3: Normalización

Las imágenes se re-escalan al rango '0 a 1' para que el Lector pueda entenderlas.

```
(imagenes_falsas + 1) / 2.0
```



Paso 2: Creación

El 'Generador' recibe el ruido y produce 4 imágenes falsas (rango de píxeles: -1 a 1).

Paso 4: Identificación

El 'Lector' predice la clase de cada una de las 4 imágenes generadas.



Paso 5: Ensamblaje

Las 4 imágenes se componen en una cuadrícula de 2x2 y se asocian con sus etiquetas predichas.

El Resultado Final



El sistema genera esta cuadrícula y, simultáneamente, produce la lista de respuestas correctas.

Salida del Lector: `['Zapatilla', 'Vestido', 'Bolso', 'Camisa']`

Nota al Margen: Cada vez que se solicita un CAPTCHA, se genera un conjunto de imágenes y etiquetas completamente nuevo.

El Plano Completo del Sistema

Desde los datos de origen hasta el desafío final, este es el flujo completo de información y lógica que da vida al CAPTCHA inteligente.

