Notebook 1: Preparación de Datos

¿Qué hace este notebook?

Este notebook organiza las imágenes del dataset **Animals5** en carpetas separadas para:

- **Entrenamiento** (train): Las imágenes que el modelo usará para aprender
- **Validación** (val): Imágenes para verificar el progreso durante el entrenamiento

- **Prueba** (test): Imágenes nuevas para evaluar el rendimiento final

¿Por qué es importante?

Separar los datos evita que el modelo "haga trampa" memorizando las imágenes de prueba. Así podemos medir qué tan bien generaliza a imágenes que nunca ha visto.

Distribución de datos

- Cada categoría tiene **1,446 imágenes** (balanceado)
- Train: 723 | Validación: 361 | Test: 362 por clase

```
import os
import shutil
import random

# Configuración
SOURCE_DIR = 'Animals5'
BASE_DIR = 'dataset_split'
CATEGORIES = ['cavallo', 'elefante', 'gallina', 'mucca', 'pecora']
# El conteo mínimo encontrado fue 1446 (elefante), limitamos todas las
clases a este número para simetría.
TARGET_COUNT = 1446

TRAIN_SPLIT = 0.5
VAL_SPLIT = 0.25
TEST_SPLIT = 0.25

def prepare_data():
    # Limpiar directorio de destino si existe
    if os.path.exists(BASE_DIR):
        shutil.rmtree(BASE_DIR)

    for category in CATEGORIES:
        print(f"Procesando {category}...")

        # Ruta fuente
        src_path = os.path.join(SOURCE_DIR, category)
```

```
        all_files = [f for f in os.listdir(src_path) if
f.lower().endswith(('.png', '.jpg', '.jpeg'))]

        # Aleatorizar
        random.shuffle(all_files)

        # Truncar al conteo objetivo
        selected_files = all_files[:TARGET_COUNT]
        print(f"  Seleccionadas {len(selected_files)} imágenes de
{len(all_files)}")

        # Calcular índices de división
        n_train = int(TARGET_COUNT * TRAIN_SPLIT) # 723
        n_val = int(TARGET_COUNT * VAL_SPLIT)     # 361
        # Test toma el resto para asegurar que el total coincida

        train_files = selected_files[:n_train]
        val_files = selected_files[n_train:n_train+n_val]
        test_files = selected_files[n_train+n_val:]

        # Crear directorios y copiar
        for subset, files in [('train', train_files), ('val',
val_files), ('test', test_files)]:
            dest_dir = os.path.join(BASE_DIR, subset, category)
            os.makedirs(dest_dir, exist_ok=True)

            for f in files:
                shutil.copy2(os.path.join(src_path, f),
os.path.join(dest_dir, f))

        print(f"  Split: Train={len(train_files)},
Val={len(val_files)}, Test={len(test_files)}")

prepare_data()
```