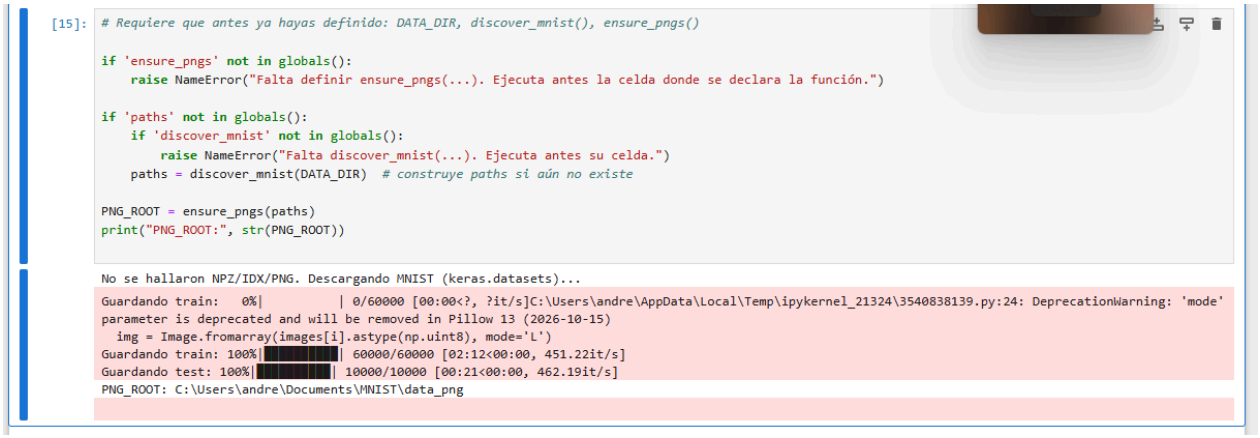
Taller NoSQL: MongoDB Compass + MNIST

Bryan Stiven Gomez Taborda

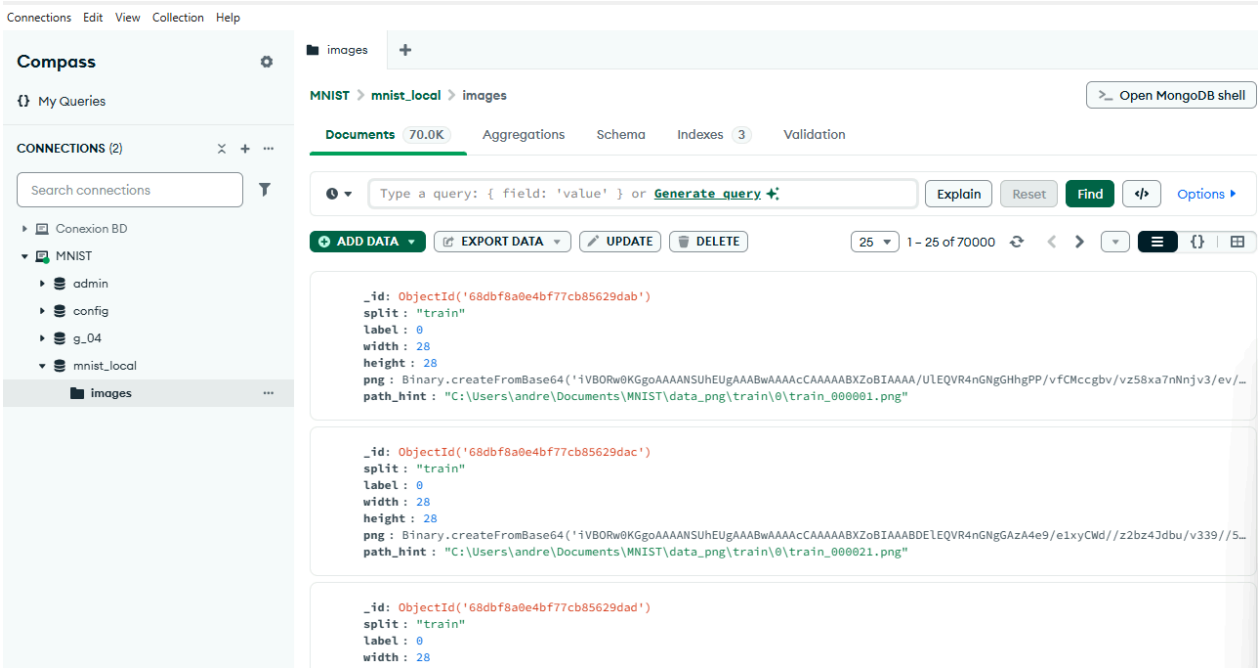
conversión a PNG e inserción a Compass desde Júpiter Notebook de Colección MNIST y Colección Images:



```
[15]: # Requiere que antes ya hayas definido: DATA_DIR, discover_mnist(), ensure_pngs()

if 'ensure_pngs' not in globals():
    raise NameError("Falta definir ensure_pngs(...). Ejecuta antes la celda donde se declara la función.")

if 'paths' not in globals():
    if 'discover_mnist' not in globals():
        raise NameError("Falta discover_mnist(...). Ejecuta antes su celda.")
    paths = discover_mnist(DATA_DIR)  # construye paths si aún no existe

PNG_ROOT = ensure_pngs(paths)
print("PNG_ROOT:", str(PNG_ROOT))
```

```
No se hallaron NPZ/IDX/PNG. Descargando MNIST (keras.datasets)...
Guardando train:   0%|          | 0/60000 [00:00<?, ?it/s]C:\Users\andre\AppData\Local\Temp\ipykernel_21324\3540838139.py:24: DeprecationWarning: 'mode'
parameter is deprecated and will be removed in Pillow 13 (2026-10-15)
  img = Image.fromarray(images[i].astype(np.uint8), mode='L')
Guardando train: 100%|██████████| 60000/60000 [02:12<00:00, 451.22it/s]
Guardando test: 100%|██████████| 10000/10000 [00:21<00:00, 462.19it/s]
PNG_ROOT: C:\Users\andre\Documents\MNIST\data_png
```

```
Inserción train: 60000it [00:32, 1858.81it/s]
Insertados 60000 documentos en split='train'.
Inserción test: 10000it [00:02, 3433.42it/s]
Insertados 10000 documentos en split='test'.
{'ns': 'mnist_local.images',
 'size': 30279875,
 'count': 70000,
 'avgObjSize': 432,
 'numOrphanDocs': 0,
 'storageSize': 38694912,
 'freeStorageSize': 22306816,
 'capped': False,
 'wiredTiger': {'metadata': {'formatVersion': 1},
  'creationString': 'access_pattern_hint=none,allocation_size=4KB,app_metadata=(formatVersion=1),assert=(commit_timestamp=none,durable_timestamp=none,read_timestamp=none,write_timestamp=off),block_allocation=best,block_compressor=snappy,cache_resident=false,checksum=on,colgroups=,collator=,columns=,dictionary=0,encryption=(keyid=,name=),exclusive=false,extractor=,format=btree,huffman_key=,huffman_value=,ignore_in_memory_cache_size=false,immutable=false,import=(compare_timestamp=oldest_timestamp,enabled=false,file_metadata=,metadata_file=,panic_corrupt=true,repair=false),internal_item_max=0,internal_key_max=0,internal_key_truncate=true,internal_page_max=4KB,key_format=q,key_gap=10,leaf_item_max=0,leaf_key_max=0,leaf_page_max=32KB,leaf_value_max=64MB,log=(enabled=true),lsm=(auto_throttle=true,bloom=true,bloom_bit_count=16,bloom_config=,bloom_hash_count=8,bloom_oldest=false,chunk_count_limit=0,chunk_max=5GB,chunk_size=10MB,merge_custom=(prefix=,start_generation=0,suffix=),merge_max=15,merge_min=0),memory_page_image_max=0,memory_page_max=10m,os_cache_dirty_max=0,os_cache_max=0,prefix_compression=false,prefix_compression_min=4,source=,split_deepen_min_child=0,split_deepen_per_child=0,split_pct=90,tiered_storage=(auth_token=,bucket=,bucket_prefix=,cache_directory=,local_retention=300,name=,object_target_size=0),type=file,value_format=u,verbose=[],write_timestamp_usage=none',
  'type': 'file',
```



Visualización Rápida en NoteBook:

Modelo keras y Entrenamiento:

```
model.save("models/mnist_model.keras")

# (opcional) formato HDF5 legado (.h5) — mostrará un WARNING, pero funciona
model.save("models/mnist_model.h5")

# Para exportar a SavedModel (carpeta) para TF Serving / TFLite:
model.export("models/mnist_savedmodel")   # <-- ESTA es la API correcta en Keras 3
```

```
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file format is considered legacy. W
e recommend using instead the native Keras format, e.g. `model.save('my_model.keras')` or `keras.saving.save_model(model, 'my_model.keras')`.
Saved artifact at 'models/mnist_savedmodel'. The following endpoints are available:

* Endpoint 'serve'
  args_0 (POSITIONAL_ONLY): TensorSpec(shape=(None, 28, 28, 1), dtype=tf.float32, name='keras_tensor')
Output Type:
  TensorSpec(shape=(None, 10), dtype=tf.float32, name=None)
Captures:
  1795916101584: TensorSpec(shape=(), dtype=tf.resource, name=None)
  1795914436048: TensorSpec(shape=(), dtype=tf.resource, name=None)
  1795919242128: TensorSpec(shape=(), dtype=tf.resource, name=None)
  1795916089488: TensorSpec(shape=(), dtype=tf.resource, name=None)
  1795916101776: TensorSpec(shape=(), dtype=tf.resource, name=None)
  1795919240400: TensorSpec(shape=(), dtype=tf.resource, name=None)
  1795717346128: TensorSpec(shape=(), dtype=tf.resource, name=None)
  1795717347856: TensorSpec(shape=(), dtype=tf.resource, name=None)
```

```
[22]: %gui qt
```

Funcionamiento: