# 1.Escenario tratamiento de datos con Pyhton

## 1.1.Descripción de los datos

Los datos que tratará el scripts corresponden con una serie de hojas de la BTN25 del IGN descargadas directamente del Centro de descarga del IGN. Para el ejercicio se incluyen 9 hojas para no alargar en demasía el procesamiento. Se proporcionan 9 carpetas, cada una de ellas correspondiente a una hoja de la BTN25. Dentro de cada carpeta se encuentran un conjunto de clases de entidad en formato shapefile cuya extensión se restringe a la de la hoja. Si el alumno examina las capas comprobará que las capas toman los mismos nombres. Así por ejemplo, en todas las carpetas existe la capa con información de curvas de nivel con el nombre:

BTN25\_ETRS\_BCN0201L\_CUR\_NIV\_line

No obstante, hay capas que no están presentes en todas las hojas. Así por ejemplo, la capa de zonas protegidas solo está presente en las hojas 32044, 33044, 34041 y 34042. Esta capa se denomina:

BTN25\_ETRS\_BCN0107S\_ZON\_PRO\_polygon

## 1.2.Planteamiento

El problema que se desea resuelva el script es el siguiente. Se desea crear un continuo de datos en una GDB de fichero para poder aplicar todas las funcionalidades que proporciona la geodatabase. Para ello el script ha de recorrer las carpetas e ir fusionando (herramienta Merge) las capas que tengan el mismo nombre en una única clase de entidad con ese mismo nombre que almacenará en una GDB a su elección.

Como ejercicio opcional, se propone dotar al script de parámetros y añadirlo a una toolbox en la que se configurarán estos parámetros

# 2. Creación del Script

Para la creación del script se planteó dividirlo en diversos apartado que se fueron resolviendo de menor a mayor dificultad.

## 2.1.Merge

Empezamos creando un script de Phyton con la herramienta que se utilizará al final del proceso que nos piden que en este caso es el Merge para así saber de antemano que elementos necesito tener antes de ejecutarlo.

import arcpy

import os

arcpy.env.overwriteOutput=True

tf=r"C:\Student\DatosPracticaPhyton\32044\BTN25\_ETRS\_BCN0201L\_CUR\_NIV\_line.shp"

jf=r"C:\Student\DatosPracticaPhyton\32045\BTN25\_ETRS\_BCN0201L\_CUR\_NIV\_line.shp"

gdb=r"C:\Student\Proyectos\Phyton\Phyton.gdb"

of=r"BTN25\_ETRS\_BCN0201L\_CUR\_NIV\_line\_Merge"

arcpy.Merge\_management([tf,jf],os.path.join(gdb,of))

De este paso deducimos que es importante tener una lista con la ruta de todos los elementos a hacer la unión.

## 2.2.Rutas

Continuamos desarrollando un script para obtener las rutas de cada archivo contenido en una carpeta

import arcpy

field=r"C:\Student\DatosPracticaPhyton\32044"

des=arcpy.Describe(field)

for children in des.children:

print(children.catalogPath)

Con el des.children obtenemos un lista describe de los subelementos del archivo describe y con catalogPath pedimos la ruta de esos archivos.

## 2.3.Rutas Rutas

Ahora nos proponemos a obtener todas las rutas de los archivos contenido en una carpeta incluidos los contenidos en sus subcarpetas.

import arcpy

dir=r"C:\Student\DatosPracticaPhyton"

desDir=arcpy.Describe(dir)

for hijosDir in desDir.children:

print(hijosDir.catalogPath)

for nietosDir in hijosDir.children:

print(nietosDir.catalogPath)

## 2.4.Lista .shp

Como la herramienta Merge solo nos pide la ruta de los archivos .shp nos disponemos a crear una lista solo con esos archivos utilizando un if.

import arcpy

dir=r"C:\Student\DatosPracticaPhyton"

desDir=arcpy.Describe(dir)

listaSHPD=[]

for hijosDir in desDir.children:

for nietosDir in hijosDir.children:

if nietosDir.name[-4:]==r".shp":

listaSHPD.append(nietosDir)

for D in listaSHPD:

print(D.catalogPath)

## 2.5.Lista Sin Elementos Repetidos

Como el criterio para unir los archivos es que tengan el mismo nombre se necesita una lista con todos los nombres sin elementos repetidos para poder comparar el nombre de la lista de los describe de los archivos .shp con esta nueva lista y así a la hora de crear una lista de rutas por nombre el proceso se hará sin repeticiones innecesarias.

Se utiliza la función set que lo que hace es eliminar los elementos repetidos de una lista.

import arcpy

arcpy.env.overwriteOutput=True

dir=r"C:\Student\DatosPracticaPhyton"

desDir=arcpy.Describe(dir)

listaSHPD=[]

for hijosDir in desDir.children:

for nietosDir in hijosDir.children:

if nietosDir.name[-4:]==r".shp":

listaSHPD.append(nietosDir)

listaNombres=[]

for shpDN in listaSHPD:

listaNombres.append(shpDN.baseName)

ListaNombresUnicos=set(listaNombres)

print(ListaNombresUnicos)

print(len(listaNombres))

print(len(ListaNombresUnicos))

## 2.6.Unión Por Nombre

Es el paso final y lo que se hace utilizar lo anteriormente mencionado. Lo que hace este script es hacer la unión de los elementos con igual nombre. Para ello, se recorre la lista de los describe de los archivos .shp y los compara con los elementos de la lista que tiene un elemento por nombre y los va añadiendo a una nueva lista que es la que se usara a la hora de ejecutar el Merge.

import arcpy

import os

arcpy.env.overwriteOutput=True

gdb=r"C:\Student\Proyectos\Phyton\Phyton.gdb"

dir=r"C:\Student\DatosPracticaPhyton"

desDir=arcpy.Describe(dir)

listaSHPD=[]

listaNombres=[]

for hijosDir in desDir.children:

for nietosDir in hijosDir.children:

if nietosDir.name[-4:]==r".shp":

listaSHPD.append(nietosDir)

listaNombres.append(nietosDir.baseName)

ListaNombresUnicos=set(listaNombres)

for nombreUnico in ListaNombresUnicos:

listaIgualNombre = []

for shpD in listaSHPD:

if shpD.baseName==nombreUnico:

listaIgualNombre.append(shpD.catalogPath)

arcpy.Merge\_management(listaIgualNombre,os.path.join(gdb,nombreUnico))

# 

# 3.ToolBox

Se genera una funcion llamada Unir\_Por\_Nombre y se parametriza con arcpy.GetParametersAsText

import arcpy

import os

arcpy.env.overwriteOutput=True

def Unir\_Por\_Nombre(gdb, dir):

desDir=arcpy.Describe(dir)

listaSHPD=[]

listaNombres=[]

for hijosDir in desDir.children:

for nietosDir in hijosDir.children:

if nietosDir.name[-4:]==r".shp":

listaSHPD.append(nietosDir)

listaNombres.append(nietosDir.baseName)

ListaNombresUnicos=set(listaNombres)

for nombreUnico in ListaNombresUnicos:

listaIgualNombre = []

for shpD in listaSHPD:

if shpD.baseName==nombreUnico:

listaIgualNombre.append(shpD.catalogPath)

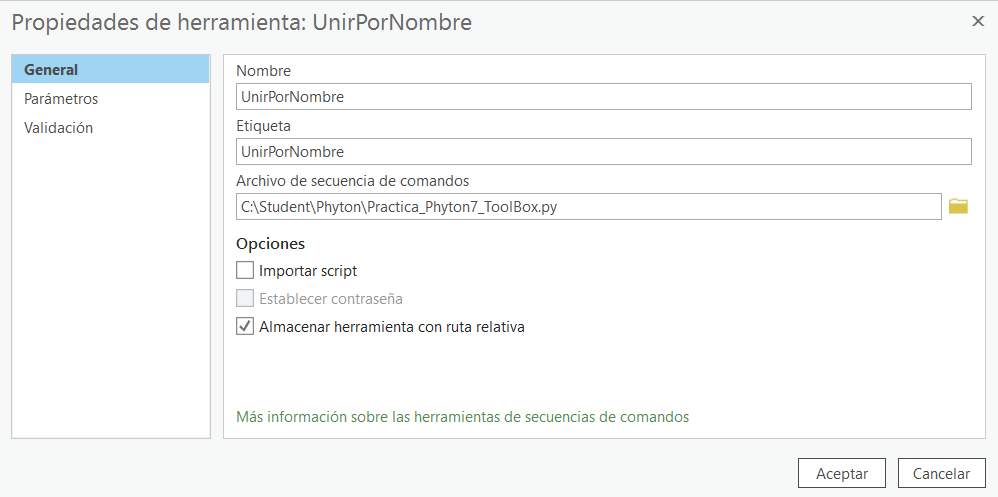
arcpy.Merge\_management(listaIgualNombre,os.path.join(gdb,nombreUnico))

if \_\_name\_\_ == "\_\_main\_\_":

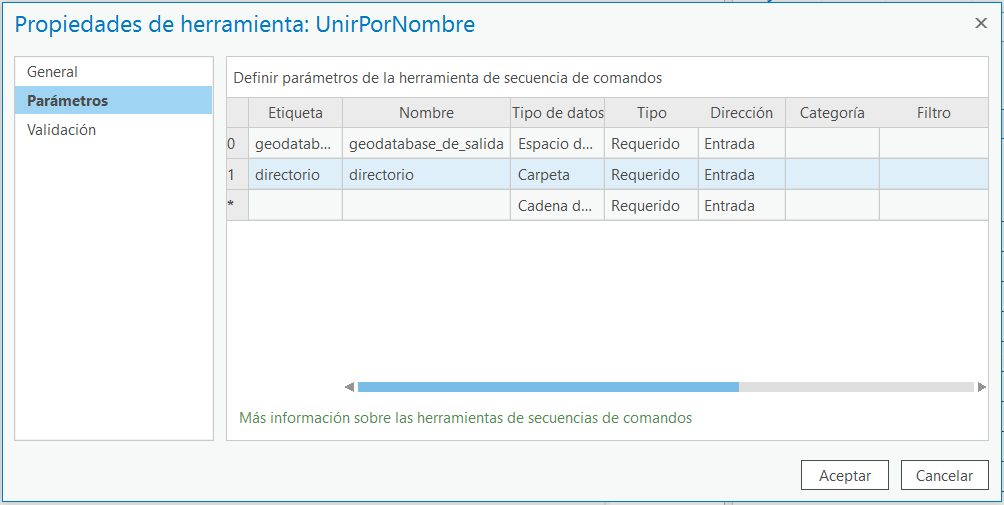
Unir\_Por\_Nombre(arcpy.GetParameterAsText(0),arcpy.GetParameterAsText(1))

luego se crea una secuencia de Comandos y se importa el código.





Se establecen los parámetros.



y por último se completan los metadatos.

