

Universidad de Oviedo

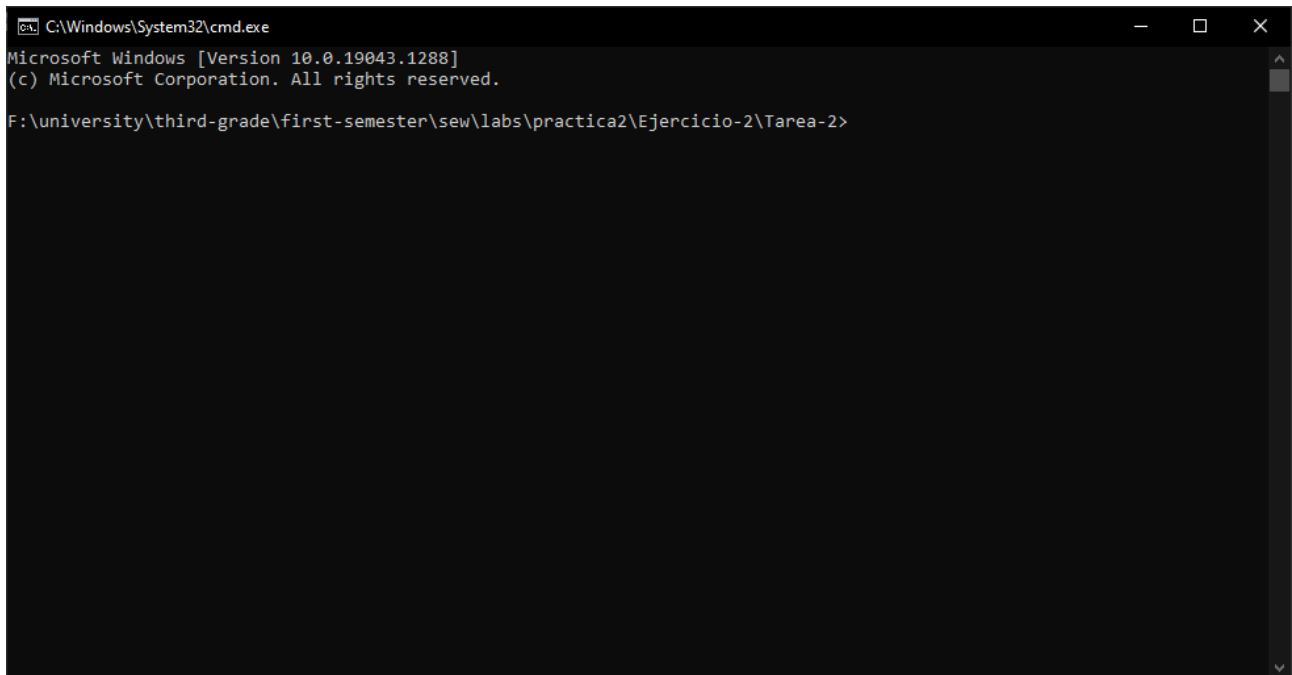
Escuela de Ingeniería de Software

KML Parser

Grado en Ingeniería de Software

Ángel Iglesias Préstamo

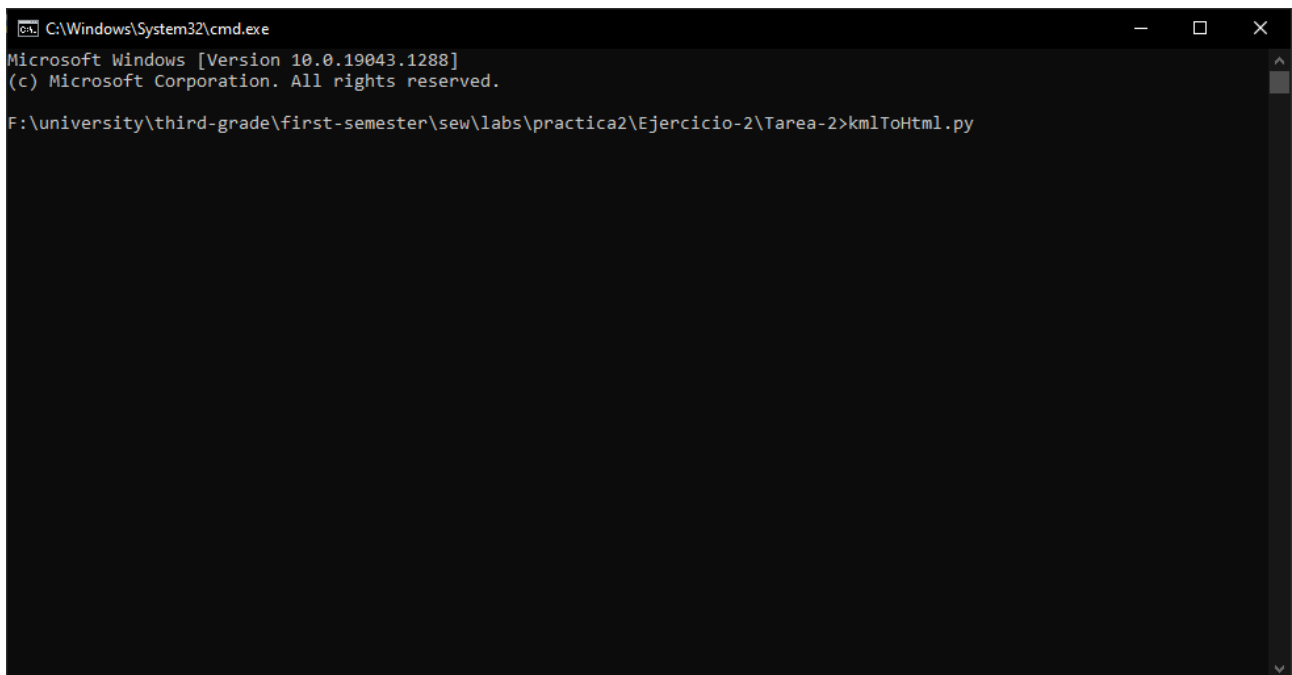
1 Forma de uso del programa.



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19043.1288]
(c) Microsoft Corporation. All rights reserved.

F:\university\third-grade\first-semester\sew\labs\practica2\Ejercicio-2\Tarea-2>
```

Paso 1: abrir la terminal en la carpeta donde está el archivo **.py*



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19043.1288]
(c) Microsoft Corporation. All rights reserved.

F:\university\third-grade\first-semester\sew\labs\practica2\Ejercicio-2\Tarea-2>kmlToHtml.py
```

Paso 2: introducir el nombre del archivo que queremos ejecutar. Si esto no sirve, será porque el intérprete de Python no ha sido incluido en el PATH de las variables de entorno del sistema. En ese caso escribe: `python3 nombre_del_archivo`

```
C:\Windows\System32\cmd.exe - kmlToHtml.py
Microsoft Windows [Version 10.0.19043.1288]
(c) Microsoft Corporation. All rights reserved.

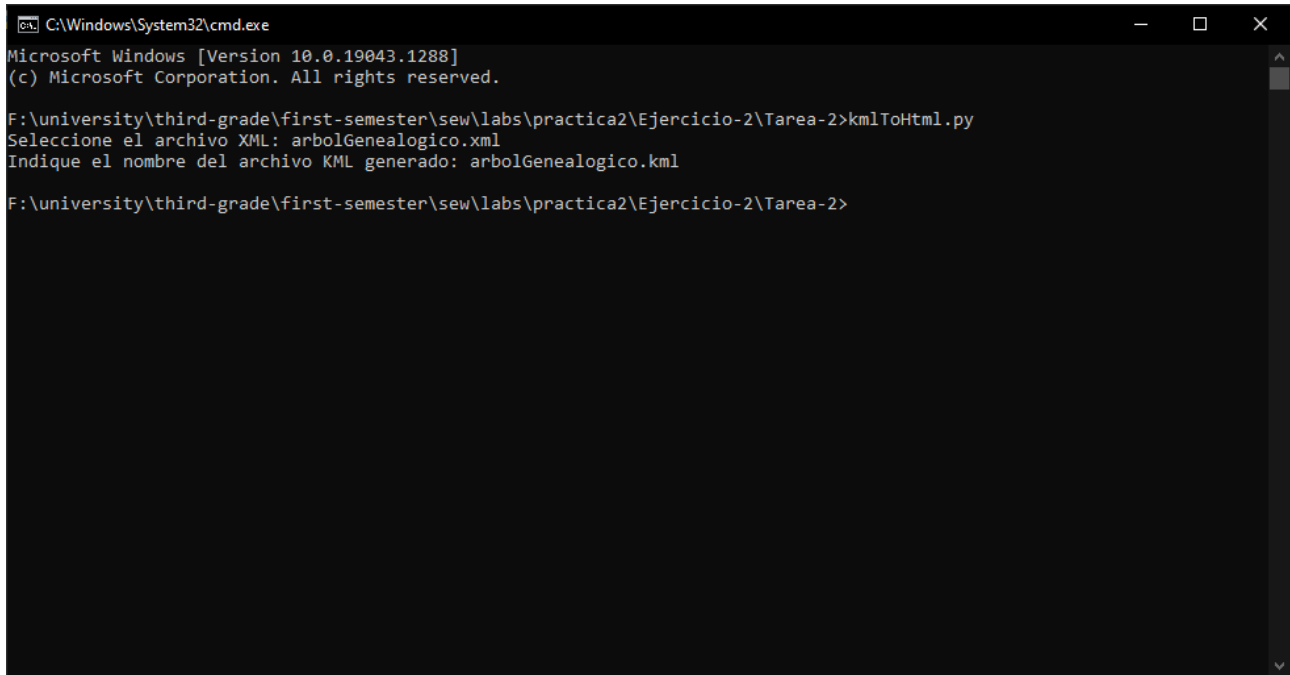
F:\university\third-grade\first-semester\sew\labs\practica2\Ejercicio-2\Tarea-2>kmlToHtml.py
Seleccione el archivo XML: arbolGenealogico.xml
```

Paso 3: introducimos el *path* al archivo XML del árbol genealógico

```
C:\Windows\System32\cmd.exe - kmlToHtml.py
Microsoft Windows [Version 10.0.19043.1288]
(c) Microsoft Corporation. All rights reserved.

F:\university\third-grade\first-semester\sew\labs\practica2\Ejercicio-2\Tarea-2>kmlToHtml.py
Seleccione el archivo XML: arbolGenealogico.xml
Indique el nombre del archivo KML generado: arbolGenealogico.kml
```

Paso 4: introducimos el nombre del archivo **.kml* que queremos generar como resultado



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19043.1288]
(c) Microsoft Corporation. All rights reserved.

F:\university\third-grade\first-semester\sew\labs\practica2\Ejercicio-2>kmlToHtml.py
Seleccione el archivo XML: arbolGenealogico.xml
Indique el nombre del archivo KML generado: arbolGenealogico.kml

F:\university\third-grade\first-semester\sew\labs\practica2\Ejercicio-2>
```

Paso 5: termina la ejecución, el archivo está esperando en el *path* indicado en el paso anterior

2 Consideraciones acerca del intérprete.

1. **Lenguaje de programación.** He utilizado Python3 para realizar toda la práctica. Ya que considero que:
 - Es interesante poder ejecutar el archivo en múltiples plataformas. Siempre y cuando tengas instalado el intérprete de Python.
 - Es un gran lenguaje de *scripting*.
 - Existen módulos - librerías - para casi cualquier cosa.
 - Es un lenguaje de programación que tiene implementado el paradigma de orientación a objetos, sin el cual sería más difícil la realización de la conversión de XML a cualquiera de las variantes pedidas: HTML, KML y SVG. Dado que la mayor parte de la aplicación se reutiliza. Sencillamente lo que hago es generar un *BinaryTree* - idea bastante evidente ya que un árbol genealógico es básicamente eso - y mediante un algoritmo recursivo recorro dicho árbol. Para mí, una solución bastante intuitiva.
2. **Versión.** Más en concreto se trata de la versión 3.10 de Python: la última estable. Aunque debería funcionar con versiones anteriores, no aseguro que lo haga. Por razones de seguridad siempre es recomendable utilizar la última versión - estable - disponible.
3. **Librería utilizada.** he utilizado la API *xml.etree.ElementTree*, una librería estándar de Python, es decir, viene pre-instalada; al menos en la versión 3.10 de Python. El resto de la implementación está hecha por mí.