

# ING1103 - Programación

## Tarea M3-V3

Fecha de Entrega: 23 de agosto de 2018. Hasta las 23:59 hrs.



Universidad de  
**los Andes**

Facultad de  
**Ingeniería  
y Ciencias Aplicadas**

## 1. Problema a resolver

Los subtítulos de películas suelen venir en archivos llamados `nombre_archivo.srt`, los cuales tienen una estructura especial que puede ser reconocida por reproductores multimedia. A continuación se muestra un ejemplo de la estructura de este tipo de archivos:

```
1
0:01:30,98 --> 0:01:35,99
Historias del pasado

2
0:01:36,52 --> 0:01:41,50
Me crie en una pequeña granja
al norte de la Muralla María.

3
0:01:42,16 --> 0:01:45,45
Una granja
dentro de las tierras de los Reiss.
```

Notar que cada subtítulo tiene una primera línea con el número del subtítulo, una segunda línea con el intervalo de tiempo y luego una **cantidad variable** de líneas (la cual es  $\geq 1$ ) con los subtítulos desplegados durante ese intervalo. La separación entre subtítulos consiste en una línea en blanco.

Desarrollando esta tarea será capaz de leer este tipo de archivos, cargarlos en una lista y poder trabajar sobre ella. Para esto debe implementar las siguientes funciones

### 1.1. Funciones a desarrollar

- No solo el contenido de un archivo es útil, muchas veces basta analizar sólo su nombre para obtener información. Para el presente problema asuma que el string con el nombre del archivo de subtítulos tiene la siguiente estructura:  
`nombre_de_serie-SNumeroTemporadaENúmeroEpisodio-título_del_episodioIdioma.srt`  
Crear una función llamada `informacion_nombre_archivo(nombre_archivo)` que, a partir del nombre del archivo, retorna una lista con 5 elementos: `nombre_serie`, `numero_temporada`, `numero_episodio`, `título_episodio`, `idioma`. Guíese por el ejemplo:

```
In [1]: informacion_nombre_archivo("Shingeki_no_Kyojin-S03E03-Historias_del_pasado-ES.srt")
Out[1]: ['Shingeki no Kyojin', 3, 3, 'Historias del pasado', 'ES']
In [2]: informacion_nombre_archivo("Shingeki_no_Kyojin-S03E03-Old_Story-EN.srt")
Out[2]: ['Shingeki no Kyojin', 3, 3, 'Old Story', 'EN']
```

**Puntaje FUN1:** 1,0 punto por obtener el resultado correctamente, respetando tipos de datos. 0,5 si se obtiene el resultado correcto, sin respetar los tipos o si no separa correctamente.

- Desarrolle la función `leer_inicio_y_fin(tiempo)` que al recibir una línea con la duración de un subtítulo, retorne una lista separando el tiempo de inicio y de fin. Guíese por el ejemplo:

```
In [1]: leer_inicio_y_fin("0:23:42,52 --> 0:23:44,52")
Out[1]: ['0:23:42,52', '0:23:44,52']
In [2]: leer_inicio_y_fin("0:10:48,94 --> 0:10:50,98")
Out[2]: ['0:10:48,94', '0:10:50,98']
```

Importante: Fíjese que los tiempos están separados por el string ' -->', el cual incluye dos espacios en blanco, dos guiones y el signo de mayor.

**Puntaje FUN2:** 1,0 punto por obtener el resultado correctamente, respetando tipos de datos. 0,5 si se obtiene el resultado correcto, sin respetar los tipos o si no separa correctamente.

- Desarrolle la función `convertir_hora_a_lista_de_enteros(hora_completa)` que recibe una string con la hora en formato hh:mm:ss,ms. La función retorna una lista de enteros [hora, minutos, segundos, milisegundos]. Guíese por los ejemplos:

```
In [1]: convertir_hora_a_lista_de_enteros("0:23:42,52")
Out[1]: [0, 23, 42, 52]
In [2]: convertir_hora_a_lista_de_enteros("0:23:44,52")
Out[2]: [0, 23, 44, 52]
```

**Puntaje FUN3:** 1,0 punto por obtener el resultado correctamente, respetando tipos de datos. 0,5 si se obtiene el resultado correcto, sin respetar los tipos.

- Desarrolle la función `leer_subtitulos(nombre_archivo)` que permita procesar el archivo de subtítulos, cargándolo en una lista de listas. Por cada subtítulo en el archivo se añadirá como elemento una lista con la siguiente estructura: `[numero_subtitulo, [inicio,fin],[subtitulo1,subtitulo2,...]]`. Notar que el inicio y fin son listas de enteros (podría usar la función anterior...). Guíese por el ejemplo:

```
In [1]: leer_subtitulos("Shingeki_no_Kyojin-S03E03-Historias_del_pasado-ES.srt")
Out[1]: [[1, [[0, 1, 30, 98], [0, 1, 35, 99]], ['Historias del pasado']],
[2, [[0, 1, 36, 52], [0, 1, 41, 50]], ['Me crie en una pequeña granja',
'al norte de la Muralla María.']],
[3, [[0, 1, 42, 16], [0, 1, 45, 45]], ['Una granja',
'dentro de las tierras de los Reiss.']],
...]
In [2]: leer_subtitulos("Shingeki_no_Kyojin-S03E03-Old_Story-EN.srt")
Out[2]: [[1, [[0, 1, 30, 980], [0, 1, 35, 990]], ['"Old Story"']],
[2, [[0, 1, 36, 530], [0, 1, 42, 160]], ['I was born on a small farm in',
'the northern region of Wall Rose.']],
[3, [[0, 1, 42, 160], [0, 1, 45, 620]], ['It was on land owned by',
'the noble family of Reiss.']],
...]
```

**Puntaje FUN4:** 2,0 puntos por obtener el resultado correcto, respetando la estructura de la lista y los tipos de datos. 1,0 si se obtiene una lista de listas con la información, sin respetar a cabalidad la estructura y tipos de datos.

**Importante:** Los puntos suspensivos indican que las listas pueden tener más elementos.

- Desarrolle la función `buscar_subtitulo(numero,nombre_archivo)` la cual, a partir de un entero con el número de subtítulo y el archivo de subtítulos retorna una lista con la información del subtítulo con el formato `[numero_subtitulo, [inicio,fin],[subtitulo1,subtitulo2,...]]`. Si el número del subtítulo no se encuentra, entonces retorna una lista vacía. Guíese por el ejemplo:

```

In [1]: buscar_subtitulo(289,"Shingeki_no_Kyojin-S03E03-Historias_del_pasado-ES.srt")
Out[1]: [289, [[0, 23, 42, 52], [0, 23, 44, 52]], ['Gente que confía en la justicia.']]
In [2]: buscar_subtitulo(289,"Shingeki_no_Kyojin-S03E03-Old_Story-EN.srt")
Out[2]: [288, [[0, 23, 42, 350], [0, 23, 44, 480]], ['Those who believe in justice...']]
In [3]: buscar_subtitulo(2,"Shingeki_no_Kyojin-S03E03-Historias_del_pasado-ES.srt")
Out[3]: [2, [[0, 1, 36, 52], [0, 1, 41, 50]], ['Me crie en una pequeña granja',
'al norte de la Muralla María.']]
In [4]: buscar_subtitulo(2,"Shingeki_no_Kyojin-S03E03-Old_Story-EN.srt")
Out[4]: [2, [[0, 1, 36, 530], [0, 1, 42, 160]], ['I was born on a small farm in',
'the northern region of Wall Rose.']]
In [5]: buscar_subtitulo(9999,"Shingeki_no_Kyojin-S03E03-Old_Story-EN.srt")
Out[5]: []

```

**Puntaje FUN5:** 1,0 por obtener el subtítulo y su información correctamente, respetando el formato.  
0,5 si obtiene el subtítulo y su información, sin respetar el formato.

## 2. Nota

La nota final se calcula con el punto base, siendo finalmente:

$$Nota = 1,0 + \sum_1^5 FUNi$$

## 3. Evaluación y Entrega

**Fecha de entrega:** 23 de agosto de 2018 hasta las 23:59, según la hora del sistema canvasUANDES, oficial en toda la universidad. Por ningún motivo se aceptarán tareas atrasadas.

**Formato de entrega:** Subir **un solo archivo comprimido en formato zip** que contenga el programa Python del problema evaluado (archivo `t3.py`), al buzón en la página de la tarea en el sitio web del curso en canvasUANDES <https://uandes.instructures.com>. El nombre de archivo debe ser “TareaM3-RUT.zip”, reemplazando “RUT” por su número de RUT sin puntos ni guión. Por ejemplo, si su RUT es 18.123.122-K, el nombre de archivo debe ser “TareaM3-18123122k.zip”.

Aquellas tareas que no funcionen correctamente a tal punto que no sea posible ejecutarlas, o produzcan errores que impidan la ejecución, serán calificadas con nota máxima 3,9.

Debe asegurarse de que su tarea sea descargable de forma íntegra desde canvasUANDES (el archivo `.zip` con la `t3.py` en su interior).

## 4. Consideraciones

El trabajo en las tareas es estrictamente individual, por lo que cuide su tarea para que no sea copiada **parcial o íntegramente** por otros. Cualquier copia será penalizada, recibiendo el mismo castigo tanto quien copia como quien permite que le copien. También es considerada copia cualquier ayuda externa recibida directamente en la tarea, sin importar si proviene de un alumno del curso, de la universidad, o de otro lugar. El castigo será establecido por el Consejo de la Facultad, siendo como mínimo un 1,0 de promedio en el curso.