

# Tarea programada I

## Computabilidad y Complejidad

### Avance 1

#### Integrantes

- Diego Bolaños Villalobos
- Nathalie Alfaro Quesada
- José Pablo Mora Cubillo

#### Descripción de la aplicación

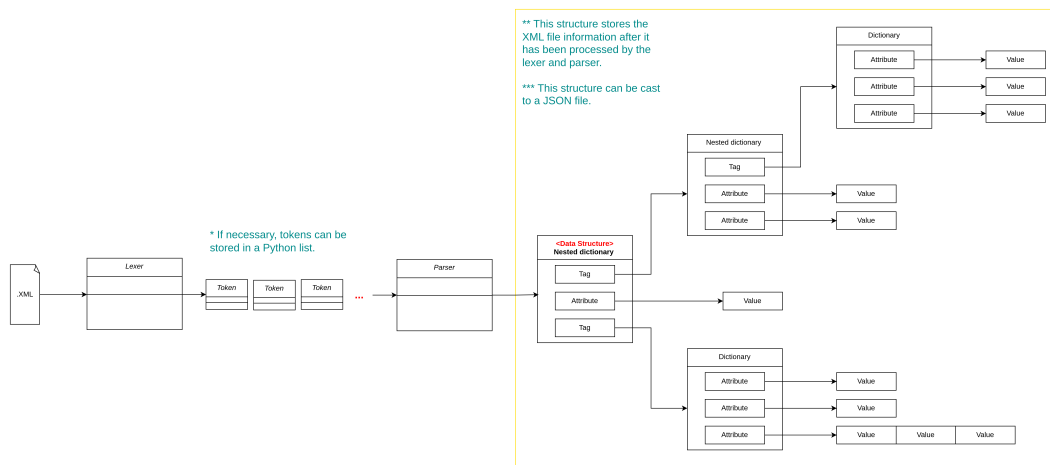
Programa escrito en Python dirigido a procesar un archivo XML con información médica relevante para pacientes, utilizando un analizador léxico y un analizador sintáctico. El objetivo es mostrar los datos de manera legible y agradable para los usuarios. Este primer avance incluye una aplicación web prototipo con las principales funcionalidades gráficas, el procesamiento del archivo XML se realizará en futuras entregas.

#### Funcionalidades

La aplicación prototipo incluye tres funcionalidades básicas enfocadas en mostrar de manera legible los datos que se extraen del archivo XML. Las funcionalidades son las siguientes:

- 1) Diagrama de árbol que muestra de manera jerárquica la información contenida en el archivo XML. En el diagrama se puede colapsar o expandir las diferentes etiquetas, así como copiar toda la información en formato JSON.
- 2) Un gráfico de barras que muestra las diez categorías de información que más se han usado en las distintas entradas de temas de salud (*Health topics*).
- 3) Una tabla que muestra la información de todos los temas de salud contenidos en el archivo XML, incluye información contenida en las etiquetas `<@title>`, `<also-called>`, `<full-summary>`, `<group>` y `<site>`.

#### Principales estructuras de datos



La imagen anterior describe un flujo de procesamiento para un archivo XML. El archivo es almacenado en memoria y posteriormente procesado por un analizador léxico (*lexer*) que divide el texto en *tokens*, estos son consumidos por el analizador sintáctico (*parser*) y la información resultante es almacenada en una estructura de datos de tipo diccionario anidado o diccionario de diccionarios.

Es importante notar que, si se desea, es posible almacenar los *tokens* en una lista o en alguna estructura más compleja como diccionarios o lista de listas. Sin embargo, la implementación de PLY en la mayoría de los casos no requiere de estructuras de datos intermedias para los *tokens*. La estructura del diccionario anidado es una representación simplificada, pero es importante considerar que una clave puede hacer referencia a otro diccionario, a una lista o a una cadena de caracteres.

### **Guía de usuario**

La explicación de cómo ejecutar la aplicación puede ser encontrada en el archivo *README.md*.