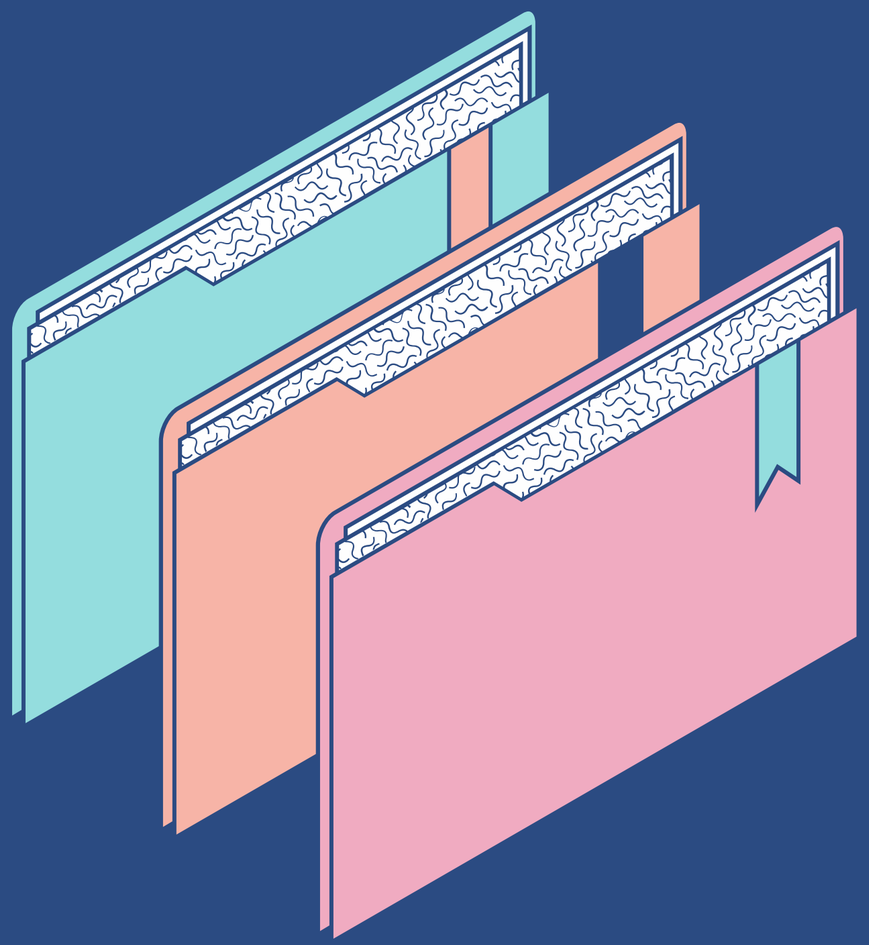
UNIVERSIDAD DE COSTA RICA
FACULTAD DE INGENIERÍAS
ESCUELA DE CIENCIAS DE LA COMPUTACIÓN E INFORMÁTICA

Grupo ZZZ:
Nathalie Alfaro Quesada - B90221
Diego Bolaños Villalobos - C21256
José Pablo Mora Cubillo - B75044

CI - 0124 Computabilidad y Complejidad
II Semestre, 2024



TAREA PROGRAMADA 1
AVANCE 3

Analizador sintáctico (Parser)

```
import ply.yacc as yacc
from lexer import tokens

def p_medi_plus_xml(t):
    'medi_plus_xml : header body TAG_HEALTH_TOPICS_CLOSURE'
    print(f'[p_medi_plus_xml]: {t[3]}')

def p_header(t):
    'header : header_version_encoding TAG_DOCTYPE tag_health_topics_nt'
    print(f'[p_header]: {t[2]}')

def p_header_version_encoding(t):
    'header_version_encoding : TAG_XML ATTRIBUTE_VERSION TEXT_OF_ATTRIBUTE ATTRIBUTE_ENCODING TEXT_OF_ATTRIBUTE XML_TAG_CLOSURE'
    print(f'[p_header_version_encoding]: {t[1]} + {t[2]} + {t[3]} + {t[4]} + {t[5]} + {t[6]}')

def p_tag_health_topics(t):
    'tag_health_topics_nt : TAG_HEALTH_TOPICS ATTRIBUTE_TOTAL NUMBER ATTRIBUTE_DATE_GENERATED DATE TIME TAG_CLOSURE'
    print(f'[p_tag_health_topics]: {t[1]} + {t[2]} + {t[3]} + {t[4]} + {t[5]} + {t[6]} + {t[7]}')

def p_body(t):
    '''body : tag_health_topic_nt tags_under_health_topic TAG_HEALTH_TOPIC_CLOSURE body
            | empty'''
    if(len(t) > 2):
        print(f'[p_body]: {t[3]}')
```

```
def p_health_topic_nt(t):
    'tag_health_topic_nt : TAG_HEALTH_TOPIC attributes_health_topic TAG_CLOSURE'
    print(f'[tag_health_topic_nt]: {t[1]} + {t[3]}')

def p_attributes_health_topic(t):
    '''attributes_health_topic : ATTRIBUTE_META_DESC TEXT_OF_ATTRIBUTE ATTRIBUTE_TITLE TEXT_OF_ATTRIBUTE ATTRIBUTE_URL URL ATTRIBUTE_ID NUMBER ATTRIBUTE_LANGUAGE TEXT_OF_ATTRIBUTE ATTRIBUTE_DATE_CREATED DATE
                               | ATTRIBUTE_URL URL ATTRIBUTE_ID NUMBER ATTRIBUTE_LANGUAGE TEXT_OF_ATTRIBUTE'''
    if(len(t) == 13):
        print(f'[p_attributes_health_topic]: {t[1]} + {t[2]} + {t[3]} + {t[4]} + {t[5]} + {t[6]} + {t[7]} + {t[8]} + {t[9]} + {t[10]} + {t[11]} + {t[12]}')
    else:
        print(f'[p_attributes_health_topic]: {t[1]} + {t[2]} + {t[3]} + {t[4]} + {t[5]} + {t[6]}')
```

```
def p_tags_under_health_topic(t):
    'tags_under_health_topic : tag_also_called_nt TAG_FULL_SUMMARY tag_group_nt tag_language_mapped_topic_nt tag_mesh_heading_nt tag_other_language_nt tag_primary_institute_nt tag_related_topic_nt tag_see_reference_nt tag_site_
    print(f'[p_tags_under_health_topic]: ***Full summary successfully captured***')
```

```
def p_tag_also_called_nt(t):
    '''tag_also_called_nt : TAG_ALSO_CALLED TEXT_OF_TAG TAG_ALSO_CALLED_CLOSURE tag_also_called_nt
                          | empty'''
    if (len(t) > 2):
        print(f'[p_tag_also_called_nt]: {t[1]} + {t[2]} + {t[3]}')

def p_tag_language_mapped_topic_nt(t):
    '''tag_language_mapped_topic_nt : TAG_LANGUAGE_MAPPED_TOPIC attributes_health_topic TAG_CLOSURE TEXT_OF_TAG TAG_LANGUAGE_MAPPED_TOPIC_CLOSURE
                                    | empty'''
    if (len(t) > 2):
        print(f'[p_tag_language_mapped_topic_nt]: {t[1]} + {t[3]} + {t[4]} + {t[5]}')

def p_tag_group_nt(t):
    '''tag_group_nt : TAG_GROUP ATTRIBUTE_URL URL ATTRIBUTE_ID NUMBER TAG_CLOSURE TEXT_OF_TAG TAG_GROUP_CLOSURE tag_group_nt
                    | TAG_GROUP ATTRIBUTE_URL URL ATTRIBUTE_ID NUMBER TAG_CLOSURE TEXT_OF_TAG TAG_GROUP_CLOSURE'''
    print(f'[p_tag_group_nt]: {t[1]} + {t[3]} + {t[4]} + {t[5]} + {t[6]}')
```

```
def p_tag_see_reference_nt(t):
    '''tag_see_reference_nt : TAG_SEE_REFERENCE TEXT_OF_TAG TAG_SEE_REFERENCE_CLOSURE tag_see_reference_nt
                            | empty'''
    if (len(t) > 2):
        print(f'[p_tag_see_reference_nt]: {t[1]} + {t[2]} + {t[3]}')

def p_tag_mesh_heading_nt(t):
    '''tag_mesh_heading_nt : TAG_MESH_HEADING tag_descriptor_nt TAG_MESH_HEADING_CLOSURE tag_mesh_heading_nt
                           | empty'''
    if (len(t)) > 2:
        print(f'[p_tag_mesh_heading_nt]: {t[1]} + {t[3]}')

def p_tag_descriptor_nt(t):
    '''tag_descriptor_nt : TAG_DESCRIPTOR ATTRIBUTE_ID TEXT_OF_ATTRIBUTE TAG_CLOSURE TEXT_OF_TAG TAG_DESCRIPTOR_CLOSURE tag_descriptor_nt
                         | TAG_DESCRIPTOR ATTRIBUTE_ID TEXT_OF_ATTRIBUTE TAG_CLOSURE TEXT_OF_TAG TAG_DESCRIPTOR_CLOSURE
                         | TAG_QUALIFIER ATTRIBUTE_ID TEXT_OF_ATTRIBUTE TAG_CLOSURE TEXT_OF_TAG TAG_QUALIFIER_CLOSURE tag_descriptor_nt
                         | TAG_QUALIFIER ATTRIBUTE_ID TEXT_OF_ATTRIBUTE TAG_CLOSURE TEXT_OF_TAG TAG_QUALIFIER_CLOSURE'''
    print(f'[p_tag_descriptor_nt]: {t[1]} + {t[2]} + {t[3]} + {t[4]} + {t[5]} + {t[6]}')
```

```
def p_tag_related_topic_nt(t):
    '''tag_related_topic_nt : TAG_RELATED_TOPIC ATTRIBUTE_URL URL ATTRIBUTE_ID NUMBER TAG_CLOSURE TEXT_OF_TAG TAG_RELATED_TOPIC_CLOSURE tag_related_topic_nt
                            | empty'''
    if(len(t) > 2):
        print(f'[p_attributes_health_topic]: {t[1]} + {t[2]} + {t[3]} + {t[4]} + {t[5]} + {t[6]} + {t[7]} + {t[8]}')


def p_tag_other_language_nt(t):
    '''tag_other_language_nt : TAG_OTHER_LANGUAGE ATTRIBUTE_VERNACULAR_NAME TEXT_OF_ATTRIBUTE ATTRIBUTE_URL URL TAG_CLOSURE TEXT_OF_TAG TAG_OTHER_LANGUAGE_CLOSURE tag_other_language_nt
                            |  TAG_OTHER_LANGUAGE ATTRIBUTE_VERNACULAR_NAME TEXT_OF_ATTRIBUTE ATTRIBUTE_URL TEXT_OF_ATTRIBUTE TAG_CLOSURE TEXT_OF_TAG TAG_OTHER_LANGUAGE_CLOSURE tag_other_language_nt
                            | empty'''
    if(len(t) > 2):
        print(f'[p_tag_other_language_nt]: {t[1]} + {t[2]} + {t[3]} + {t[4]} + {t[5]} + {t[6]} + {t[7]} + {t[8]}')

def p_tag_primary_institute_nt(t):
    '''tag_primary_institute_nt :  TAG_PRIMARY_INSTITUTE ATTRIBUTE_URL URL TAG_CLOSURE TEXT_OF_TAG TAG_PRIMARY_INSTITUTE_CLOSURE
                                | empty'''
    if(len(t) > 2):
        print(f'[p_tag_primary_institute_nt]: {t[1]} + {t[2]} + {t[3]} + {t[4]} + {t[5]} + {t[6]}')
```

```
def p_tag_site_nt(t):
    '''tag_site_nt : TAG_SITE ATTRIBUTE_TITLE TEXT_OF_ATTRIBUTE ATTRIBUTE_URL URL ATTRIBUTE_LANGUAGE_MAPPED_URL URL TAG_CLOSURE tags_under_site TAG_SITE_CLOSURE tag_site_nt
                   | TAG_SITE ATTRIBUTE_TITLE TEXT_OF_ATTRIBUTE ATTRIBUTE_URL URL ATTRIBUTE_LANGUAGE_MAPPED_URL URL TAG_CLOSURE tags_under_site TAG_SITE_CLOSURE
                   | TAG_SITE ATTRIBUTE_TITLE TEXT_OF_ATTRIBUTE ATTRIBUTE_URL URL TAG_CLOSURE tags_under_site TAG_SITE_CLOSURE tag_site_nt
                   | TAG_SITE ATTRIBUTE_TITLE TEXT_OF_ATTRIBUTE ATTRIBUTE_URL URL TAG_CLOSURE tags_under_site TAG_SITE_CLOSURE
                   | TAG_SITE ATTRIBUTE_TITLE TEXT_OF_ATTRIBUTE ATTRIBUTE_LANGUAGE_MAPPED_URL URL TAG_CLOSURE tags_under_site TAG_SITE_CLOSURE tag_site_nt
                   | TAG_SITE ATTRIBUTE_TITLE TEXT_OF_ATTRIBUTE ATTRIBUTE_LANGUAGE_MAPPED_URL URL TAG_CLOSURE tags_under_site TAG_SITE_CLOSURE'''
    if (len(t) >= 11 ):
        print(f'[p_tag_site_nt]: {t[1]} + {t[2]} + {t[3]} + {t[4]} + {t[5]} + {t[6]} + {t[7]} + {t[8]} + {t[10]}')
    else:
        print(f'[p_tag_site_nt]: {t[1]} + {t[2]} + {t[3]} + {t[4]} + {t[5]} + {t[6]} +  {t[8]}')


def p_tags_under_site(t):
    'tags_under_site : tag_information_category_nt tag_organization_nt tag_standard_description_nt'

def p_tag_information_category_nt(t):
    '''tag_information_category_nt : TAG_INFORMATION_CATEGORY TEXT_OF_TAG TAG_INFORMATION_CATEGORY_CLOSURE tag_information_category_nt
                                   | empty'''
    if(len(t) > 2):
        print(f'[p_tag_information_category_nt]: {t[1]} + {t[2]} + {t[3]}')
```

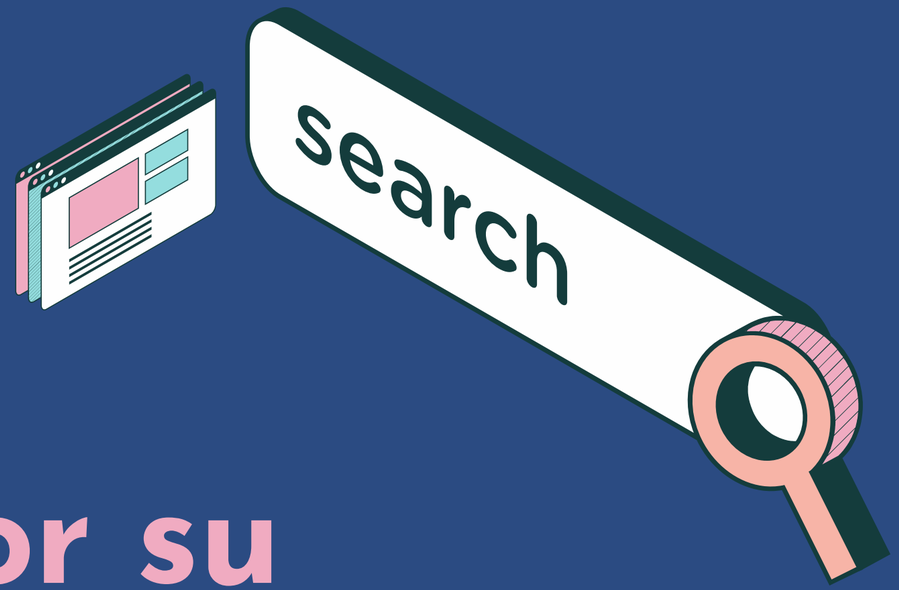
```
def p_tag_organization_nt(t):
    '''tag_organization_nt : TAG_ORGANIZATION TEXT_OF_TAG TAG_ORGANIZATION_CLOSURE tag_organization_nt
                           | empty'''
    if(len(t) > 2):
        print(f'[p_tag_organization_nt]: {t[1]} + {t[2]} + {t[3]}')

def p_tag_standard_description_nt(t):
    '''tag_standard_description_nt : TAG_STANDARD_DESCRIPTION TEXT_OF_TAG  TAG_STANDARD_DESCRIPTION_CLOSURE tag_standard_description_nt
                                   | empty'''
    if(len(t) > 2):
        print(f'[p_tag_standard_description_nt]: {t[1]} + {t[2]} + {t[3]}')

def p_empty(t):
    'empty :'

def p_error(t):
    print("Syntax error at '%s'" % t.value)


parser = yacc.yacc()
```



Gracias por su atención