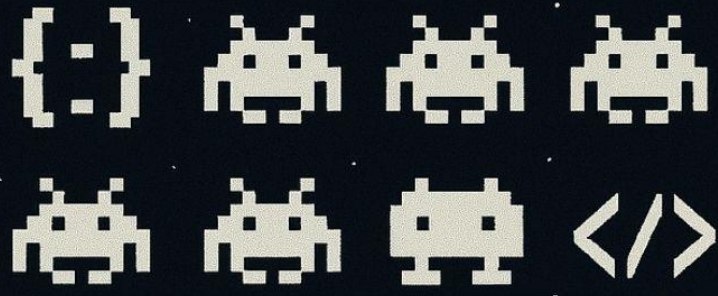
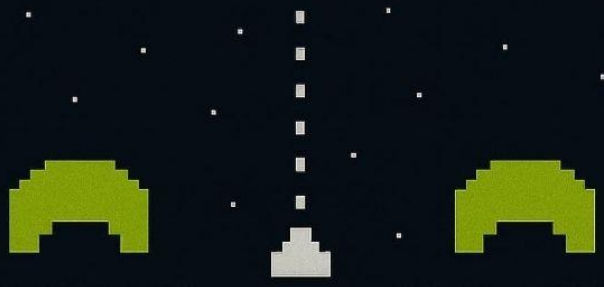
SYNTAX INVADERS





TFI DE SINTAXIS Y SEMANTICA DE LOS LENGUAJES

- BONGUAN Juliana
- CENTURION Constanza
- MAIDANA Paola
- TERUEL Axel
- SOLIS FA Matheo

INDICE DE CONTENIDO:

INTRODUCCION:

Este trabajo práctico tiene como objetivo construir analizadores léxico y sintáctico que permitan validar y transformar un archivo JSON con información. El sistema tomará un documento JSON, lo analizará para detectar posibles errores de estructura y, si es válido, generará automáticamente una versión en HTML del contenido.

Para lograr esto, se implementará un analizador léxico que identificará los tokens a partir del texto, y un analizador sintáctico que verificará la correcta construcción del documento según una gramática definida. Se utilizarán herramientas de programación, aplicando conceptos clave, trabajados durante la cursada.

Este proyecto permite aplicar los conocimientos teóricos de sintaxis y semántica en un entorno práctico, combinando programación y lenguajes formales.

Alianza y Presentación:

Nosotros bajo la identidad "Syntax Invaders", nos comprometemos a realizar este trabajo con la aptitud y actitud para trabajar en equipo que el proyecto demanda, de manera organizada, colaborativa y constante.

Nuestro objetivo principal es aplicar los conocimientos de gramáticas, léxicos y sintaxis en la creación de un lexer y parser funcional para finales de este cuatrimestre, combinando nuestras habilidades y aprendiendo mutuamente durante el proceso.

Modo de trabajo basado en la división y asignación de tareas adecuadas a las capacidades y habilidades de los integrantes del grupo.

Y queda definido entonces las reuniones semanales mediante la aplicación discord para llevar seguimiento del progreso sobre el proyecto.

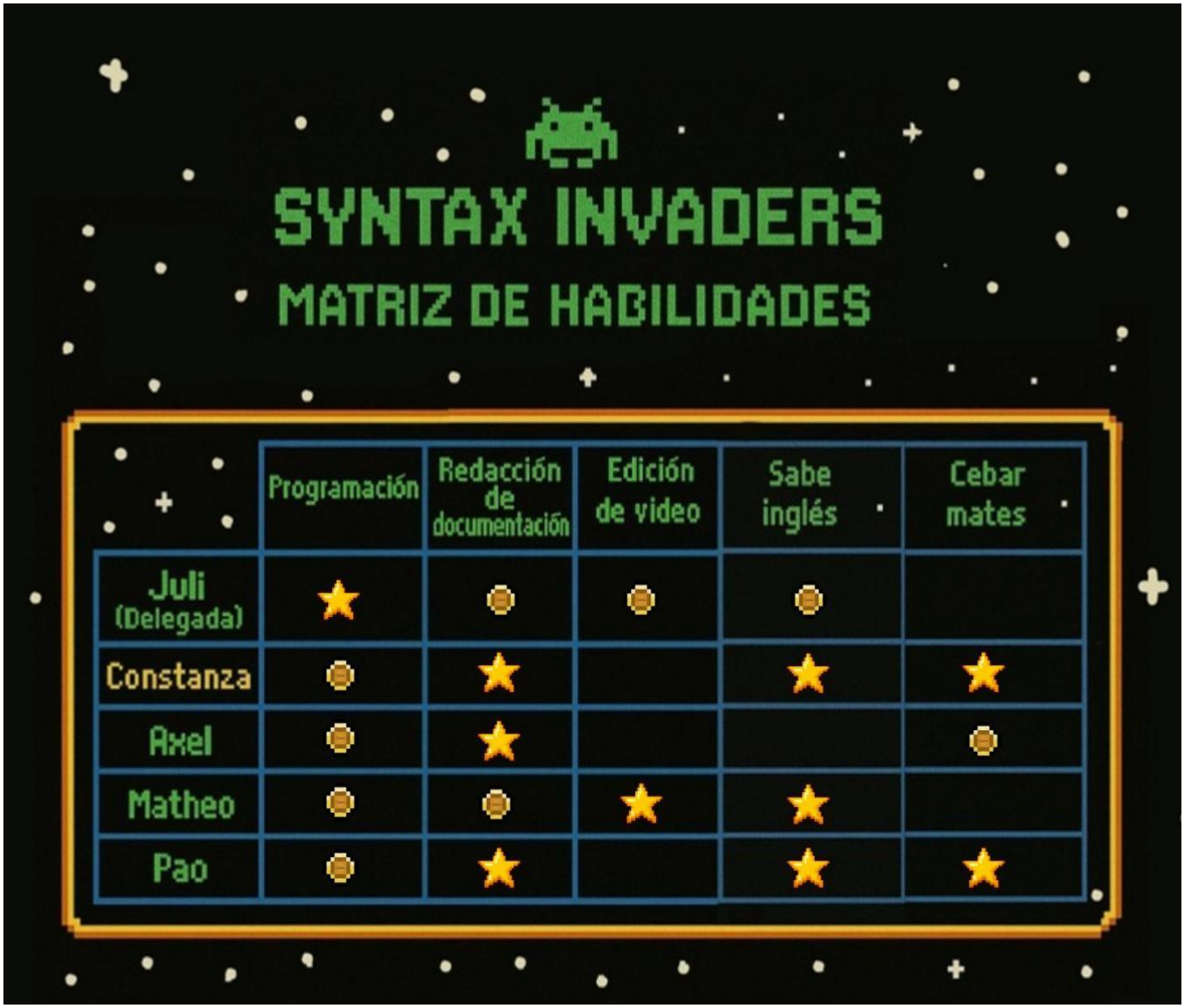
En caso de requerirse, se realizarán reuniones adicionales en casos como, organización, situaciones que requieran de distintos puntos de vista, solución de problemas en conjunto, entre otras.

El equipo esta conformado por:

- Bonguan Juliana
- Centurion Constanza
- Maidana Paola
- Solis Fa Matheo
- Teruel Axel

LOGO:

Matriz de Habilidades:



REFERENCIAS:

☆ : la persona es experta o conoce mucho del tema.
◍ : la persona no es experta pero podría desenvolverse con dicha competencia y esta interesado en aprender o desarrollar la habilidad

Simbolos de la Gramatica

Símbolo Inicial
Σ → JSON

Símbolos No Terminales
NO_TERMINALES = {
JSON, EQUIPOS, EQUIPO, DIR, DIR_DET, INTEGRANTES, INTEGRANTE,
NOMBRE_EQUIPO, IDENTIDAD_EQUIPO, CARRERA, ASIGNATURA, UNI_REGIONAL,
ALIANZA_EQUIPO, NOMBRE, EDAD, CARGO, FOTO, EMAIL, HABILIDADES, SALARIO, ACTIVO,
PROYECTOS, PROYECTO, NOMBRE_PROYECTO, ESTADO, RESUMEN, FECHA_INICIO,
FECHA_FIN,
VIDEO, CONCLUSION, VERSION, FIRMA_DIGITAL, TAREAS, TAREA
}

Símbolos Terminales (Tokens)
TOKENS = {
"equipos", "version", "firma_digital", "nombre_equipo", "identidad_equipo", "dirección",
"link",
"carrera", "asignatura", "universidad_regional", "alianza_equipo", "integrantes", "proyectos",
"nombre", "edad", "cargo", "foto", "email", "habilidades", "salario", "activo", "estado",
"resumen",
"tareas", "fecha_inicio", "fecha_fin", "video", "conclusion",
"To do", "In progress", "Canceled", "Done", "On hold",
"Product Analyst", "Project Manager", "UX designer", "Marketing", "Developer", "Devops",
"DB admin",
":", "{", "}", "[", "]", ",",
"STRING", "INTEGER", "FLOAT", "BOOL", "NULL", "URL", "EMAIL", "DATE"
}

Reglas de Producción
Estructura Principal del JSON
JSON → { EQUIPOS, VERSION, FIRMA_DIGITAL }
JSON → { EQUIPOS, FIRMA_DIGITAL, VERSION }
JSON → { VERSION, EQUIPOS, FIRMA_DIGITAL }
JSON → { VERSION, FIRMA_DIGITAL, EQUIPOS }
JSON → { FIRMA_DIGITAL, EQUIPOS, VERSION }
JSON → { FIRMA_DIGITAL, VERSION, EQUIPOS }

EQUIPOS→ (LISTA RECALCADA EN PAG.6)
VERSION “version” : STRING | “version” : {} | “version” : NULL |
FIRMA_DIGITAL “firma_digital” : STRING | “firma_digital” : {} | “firma_digital” : NULL |

Lista de Equipos

EQUIPOS → "equipos" : [LISTA_EQUIPOS]
LISTA_EQUIPOS → EQUIPO | EQUIPO , LISTA_EQUIPOS

Estructura de un Equipo

EQUIPO → { NOMBRE_EQUIPO, IDENTIDAD_EQUIPO, LINK?, ASIGNATURA, CARRERA, UNI_REGIONAL, DIR?, ALIANZA_EQUIPO, INTEGRANTES, PROYECTOS }
EQUIPO → { NOMBRE_EQUIPO, IDENTIDAD_EQUIPO, ASIGNATURA, CARRERA, UNI_REGIONAL, DIR?, ALIANZA_EQUIPO, INTEGRANTES, PROYECTOS }
EQUIPO → { NOMBRE_EQUIPO, IDENTIDAD_EQUIPO, LINK?, ASIGNATURA, CARRERA, UNI_REGIONAL, ALIANZA_EQUIPO, INTEGRANTES, PROYECTOS }
EQUIPO → { NOMBRE_EQUIPO, IDENTIDAD_EQUIPO, ASIGNATURA, CARRERA, UNI_REGIONAL, ALIANZA_EQUIPO, INTEGRANTES, PROYECTOS }

Atributos de un Equipo

NOMBRE_EQUIPO → "nombre_equipo" : STRING
IDENTIDAD_EQUIPO → "identidad_equipo" : URL
LINK → "link" : URL | "link" : NULL | "link" : {} |
ASIGNATURA → "asignatura" : STRING
CARRERA → "carrera" : STRING
UNI_REGIONAL → "universidad_regional" : STRING
ALIANZA_EQUIPO → "alianza_equipo" : STRING

Estructura de Dirección

DIR → "dirección" : { DIR_DET } | "dirección" : NULL | "dirección" : { } |
DIR_DET →
- "calle" : STRING , "ciudad" : STRING , "país" : STRING
- "calle" : STRING , "país" : STRING , "ciudad" : STRING
- "ciudad" : STRING , "calle" : STRING , "país" : STRING
- "ciudad" : STRING , "país" : STRING , "calle" : STRING
- "país" : STRING , "calle" : STRING , "ciudad" : STRING
- "país" : STRING , "ciudad" : STRING , "calle" : STRING

Lista de Integrantes

INTEGRANTES → "integrantes" : [LISTA_INTEGRANTES]
LISTA_INTEGRANTES → INTEGRANTE | INTEGRANTE , LISTA_INTEGRANTES

Estructura de un Integrante

INTEGRANTE → { NOMBRE, EDAD, CARGO, FOTO, EMAIL, HABILIDADES, SALARIO, ACTIVO }

Atributos de un Integrante

NOMBRE → "nombre" : STRING
EDAD → "edad" : INTEGER | "edad" : NULL | “edad” : {} |

CARGO → "cargo" : "Product Analyst" | "cargo" : "Project Manager" | "cargo" : "UX designer" | "cargo" :"Marketing" |"cargo" : "Developer" |"cargo" : "Devops" | "cargo" :"DB admin"
FOTO → "foto" : URL
EMAIL → "email" : EMAIL
HABILIDADES → "habilidades" : STRING
SALARIO → "salario" : FLOAT
ACTIVO → "activo" : BOOL

Lista de Proyectos
PROYECTOS → "proyectos" : [LISTA_PROYECTOS]
LISTA_PROYECTOS → PROYECTO | PROYECTO , LISTA_PROYECTOS

Estructura de un Proyecto
PROYECTO → { NOMBRE_PROYECTO, ESTADO, RESUMEN, TAREAS, FECHA_INICIO, FECHA_FIN, VIDEO, CONCLUSION }

Atributos de un Proyecto
NOMBRE_PROYECTO → "nombre" : STRING
ESTADO → "estado" : "To do" | "estado" : "In progress" | "estado" : "Canceled" | "estado" : "Done" | "estado" : "On hold"
RESUMEN → "resumen" : STRING
FECHA_INICIO → "fecha_inicio" : DATE | "fecha_inicio" : NULL
FECHA_FIN → "fecha_fin" : DATE | "fecha_fin" : NULL
VIDEO → "video" : URL | "video" : NULL
CONCLUSION → "conclusion" : STRING | "conclusion" : NULL

Lista de Tareas
TAREAS → "tareas" : [LISTA_TAREAS]
LISTA_TAREAS → TAREA | TAREA , LISTA_TAREAS

Estructura de una Tarea
TAREA → { NOMBRE, ESTADO, RESUMEN, FECHA_INICIO, FECHA_FIN }
TAREA → { ESTADO, NOMBRE, RESUMEN, FECHA_INICIO, FECHA_FIN }
TAREA → { NOMBRE, RESUMEN, ESTADO, FECHA_INICIO, FECHA_FIN }
TAREA → { RESUMEN, NOMBRE, ESTADO, FECHA_INICIO, FECHA_FIN }
TAREA → { NOMBRE, RESUMEN, ESTADO, FECHA_INICIO, FECHA_FIN }
TAREA → { ESTADO, RESUMEN, NOMBRE, FECHA_INICIO, FECHA_FIN }

Atributos de Tareas
NOMBRE → “Nombre”: STRING
ESTADO → “Estado”: STRING
RESUMEN → “Resumen”: STRING
FECHA_INICIO → “fecha_inicio : NULL | “fecha_inicio”: DATE | “fecha_inicio : {} |

FECHA_FIN → “fecha_fin : NULL | “fecha_fin”: DATE | “fecha_fin : {} |

Consideraciones Importantes

- Los objetos equipo, integrante y proyecto deben respetar el orden exacto de atributos como en la gramática.
- Los demás objetos (como dir) pueden tener el orden flexible.
- Los valores NULL pueden aparecer en campos opcionales o cuando se indica explícitamente.
- Los booleanos son true y false sin comillas.
- Las fechas deben estar entre comillas dobles y cumplir el formato "YYYY-MM-DD".
- Los campos URL y EMAIL deben cumplir con los formatos indicados y estar entre comillas dobles.
- Se debe respetar la coma después de cada elemento salvo antes de cierre de objeto o lista.
- Las claves son sensibles a mayúsculas y minúsculas y deben respetarse exactamente como se definen.

Intentamos hacer la gramática de forma modular, y además un análisis desglosándolo por estructura y atributos, teniendo en cuenta que si bien puede haber redundancia en algún no terminal, lo hicimos de esta forma para una mejor interpretación e implementación

Justificación

La gramática fue pensada y diseñada para facilitar la implementación del lexer y parser en **PLY**(Python) ya que es un lenguaje intuitivo, fácil de aprender y que maneja la mayoría del grupo.
Se prioriza la claridad, la validación precisa de tipos y valores, y la presentación formal, siguiendo el modelo de ejemplo.json presentado en el TPI de este año.
Asegurando la mayor detección de errores y la generación de salidas adicionales como “arch.HTML”.