

Universidad San Carlos de Guatemala
Facultad de ingeniería.
Ingeniería en ciencias y sistemas



Proyecto 1

TourneyJS: Analizador Léxico para Gestión de Torneos Deportivos

PONDERACIÓN: 25

Contenido

1. Resumen Ejecutivo	2
2. Competencia que desarrollaremos	3
3. Objetivos del Aprendizaje	4
3.1 Objetivo General	4
3.2 Objetivos Específicos	4
4. Enunciado del Proyecto	5
4.1 Estructura del Lenguaje	6
4.2 Entregables	12
5. Metodología	13
6. Desarrollo de Habilidades Blandas	14
7. Cronograma	14
8. Rúbrica de Calificación	15
8.1 Requisitos para optar a la calificación	15
8.4 Valores	16
9. Entrega	17
10. Restricciones	18

1. Resumen Ejecutivo

TourneyJS es un proyecto de desarrollo de software cuyo objetivo principal es construir una herramienta educativa que permita analizar, interpretar y representar estructuras de torneos deportivos utilizando JavaScript como lenguaje de implementación. Este proyecto se enfoca en el diseño e implementación de un analizador léxico en JavaScript, capaz de reconocer sintácticamente la definición de equipos, jugadores, partidos y resultados, partiendo de archivos de texto estructurados con reglas específicas.

El problema que se aborda es la necesidad de contar con una solución manual, controlada y didáctica para procesar lenguajes formales orientados a gestión deportiva, sin recurrir a herramientas automáticas como ANTLR o parsers generados automáticamente. Esto permite al estudiante comprender en profundidad la lógica detrás del análisis léxico y la transición entre modelos teóricos y aplicaciones prácticas.

La solución propuesta incluye:

- La implementación de un autómata finito determinista (AFD) que analiza el archivo de entrada carácter por carácter.
- La generación de tokens válidos y la detección de errores léxicos en una tabla estructurada HTML.
- El cálculo automático de tablas de posiciones y estadísticas de torneos.
- La producción de diagramas de brackets de eliminación y árboles de torneos utilizando Graphviz.
- Una interfaz web interactiva desarrollada en HTML/CSS/JavaScript que permite al usuario cargar, editar, analizar y exportar sus torneos.

Este proyecto no solo fortalece las competencias del estudiante en análisis léxico, sino que también integra conocimientos de estructuras de datos, algoritmos de ordenamiento, representación visual de competencias deportivas, y desarrollo de interfaces web modernas, todo bajo un enfoque riguroso y sin el uso de librerías externas para el análisis del lenguaje.

2. Competencia que desarrollaremos

- El estudiante aplica el análisis léxico mediante el uso de expresiones regulares para solucionar problemas que requieran el reconocimiento de lenguajes.
- El estudiante aplica técnicas de minimización de autómatas finitos deterministas (AFD) mediante la identificación de estados equivalentes y particiones sucesivas para reducir el número de estados sin alterar el lenguaje reconocido.
- El estudiante construye aplicaciones mediante el uso del análisis léxico y sintáctico utilizando herramientas de procesamiento de lenguajes formales para dar solución a enunciados que requieren el reconocimiento de lenguajes.

3. Objetivos del Aprendizaje

3.1 Objetivo General

El estudiante será capaz de diseñar y desarrollar una herramienta de análisis léxico en JavaScript que interprete archivos estructurados para la definición de torneos deportivos, generando automáticamente tablas de posiciones, estadísticas y diagramas de brackets mediante Graphviz. La solución integrará conocimientos de teoría de autómatas, programación orientada a objetos, interfaces web con HTML/CSS/JavaScript, y buenas prácticas de desarrollo de software, entregando un prototipo funcional y documentado.

3.2 Objetivos Específicos

Al finalizar el proyecto, los estudiantes deberán ser capaces de:

1. **Implementar un analizador léxico funcional en JavaScript**
 - a. Diseñar y desarrollar un escáner léxico basado en un autómata finito determinista (AFD) que procese archivos estructurados, reconozca tokens válidos del lenguaje definido para torneos deportivos y detecte errores léxicos con precisión.
2. **Integrar el cálculo automático de estadísticas y generación de diagramas**
 - a. Construir un módulo que, a partir de los tokens analizados, genere tablas de posiciones automáticas, calcule estadísticas deportivas y cree diagramas de brackets de eliminación utilizando Graphviz.
3. **Desarrollar una interfaz web interactiva**
 - a. Diseñar una UI moderna que permita a los usuarios cargar archivos de torneos, visualizar los tokens o errores detectados, generar las tablas y brackets correspondientes, y exportar los resultados fácilmente en un entorno web intuitivo

4. Enunciado del Proyecto

Se solicita el desarrollo de una herramienta de análisis léxico en JavaScript que permita interpretar archivos de texto que describan un torneo deportivo basado en equipos, jugadores y resultados de partidos. La aplicación deberá realizar un análisis léxico del archivo de entrada, extraer los tokens correspondientes y generar dos tipos de salida:

1. Un reporte HTML con las tablas de posiciones finales, estadísticas individuales y de equipos, ordenadas según el sistema de puntuación del torneo.
2. Diagramas de brackets de eliminación en formato gráfico, generados con Graphviz, representando visualmente la estructura del torneo y el avance de los equipos.

El sistema contará con una interfaz web que permitirá a los usuarios cargar archivos, visualizar el análisis léxico, generar las salidas y exportarlas.

Interfaz Propuesta

TourneyJS - Analizador de Torneos

Cargar Archivo Analizar Torneo Generar Reporte Mostrar Bracket

Definición del Torneo

Cargue un archivo de torneo...

4.1 Estructura del Lenguaje

El archivo de entrada sigue una estructura jerárquica donde los elementos principales son:

1. **TORNEO**: Sección donde se define la información general del torneo de eliminación.
2. **EQUIPOS**: Sección donde se definen todos los equipos participantes y sus jugadores.
3. **ELIMINACION**: Sección donde se definen los partidos del torneo de eliminación directa organizados por fases.

Reglas de Sintaxis del Lenguaje

Elemento	Descripción	Estructura y Valores Permitidos	Ejemplo
TORNEO	Define la información general del torneo de eliminación	Debe comenzar con TORNEO { y terminar con }. Debe incluir nombre y cantidad de equipos	TORNEO { nombre: "Copa Mundial", equipos: 8 }
EQUIPOS	Define la sección donde se listarán los equipos participantes	Debe comenzar con EQUIPOS { y terminar con }	EQUIPOS { equipo: ... }
equipo	Define un equipo dentro de la sección EQUIPOS	equipo: "Nombre" [jugadores: ...] donde se listan todos los jugadores	equipo: "Brasil" [jugador: "Pelé", jugador: "Ronaldo"]
jugador	Define un jugador dentro de un equipo	jugador: "Nombre" [posicion: "POSICION", numero: NUMERO, edad: EDAD]	jugador: "Messi" [posicion: "DELANTERO", numero: 10, edad: 35]
ELIMINACION	Define la estructura del torneo de eliminación directa	Debe comenzar con ELIMINACION { y contener las fases en orden	ELIMINACION { cuartos: {...}, semifinal: {...}, final: {...} }
fase	Define una fase específica del torneo	fase: [partidos de esa fase]	cuartos: [partido: ..., partido: ...]
partido	Define un partido entre dos equipos con resultado	partido: "Equipo1" vs "Equipo2" [resultado: "X-Y", goleadores: {...}]	partido: "Brasil" vs "Argentina" [resultado: "2-1", goleadores: ["Pelé", "Messi"]]
goleador	Define quién anotó goles en un partido	goleador: "NombreJugador" [minuto: X]	goleador: "Pelé" [minuto: 23]
Posiciones	Valores permitidos para posiciones de jugadores	"PORTERO", "DEFENSA", "MEDIOCAMPO", "DELANTERO"	posicion: "DELANTERO"

Sección de Errores Léxicos

Cuando se detectan errores en la sintaxis del archivo de entrada, estos deben reportarse en una tabla HTML.

No.	Lexema	Tipo de Error	Descripción	Línea	Columna
1	"equip0"	Token inválido	Palabra clave mal escrita	8	5
2	"["	Falta de símbolo esperado	Corchete de cierre faltante	12	15
3	"3-"	Formato incorrecto	Resultado de partido incompleto	25	45

Formato de la Tabla de Errores

Cada error léxico debe contener la siguiente información:

- **Número del error:** Identificador del error en la lista.
- **Lexema:** El valor que causó el error.
- **Tipo de Error:** Puede ser:
 - **Token inválido**
 - **Falta de símbolo esperado**
 - **Uso incorrecto de comillas o paréntesis**
- **Descripción:** Breve explicación del error.
- **Línea y columna:** Ubicación exacta del error en el archivo de entrada.

Sección de Tokens Extraídos

Cuando el código de entrada es válido, el analizador debe extraer y mostrar los tokens reconocidos en una tabla HTML. Estos tokens representan los elementos válidos del lenguaje.

Formato de la Tabla de Tokens

No.	Lexema	Tipo	Línea	Columna
1	"TORNEO"	Palabra Reservada	1	1
2	"{"	Llave Izquierda	1	8
3	"nombre"	Atributo	2	5
4	"Copa Mundial Universitaria"	Cadena	2	13
5	"equipo"	Palabra Reservada	6	5
6	"Leones FC"	Identificador	6	

Reportes Generados por el Sistema

El sistema debe generar automáticamente cuatro reportes principales que aprovechen toda la información procesada, estos reportes deben realizarse en uno o varios reportes individuales, a discreción del estudiante, en formato **HTML**:

Reporte de Bracket de Eliminación

Muestra el avance de todos los equipos a través de las fases del torneo, incluyendo resultados y próximos enfrentamientos.

Fase	Partido	Resultado	Ganador
Cuartos de Final	Leones FC vs Cóndores FC	3-1	Leones FC
Cuartos de Final	Águilas United vs Tigres Academy	2-0	Águilas United
Semifinal	Leones FC vs Águilas United	1-0	Leones FC
Final	Leones FC vs TBD	Pendiente	-

Reporte de Estadísticas por Equipo

Análisis detallado del desempeño de cada equipo en el torneo.

Equipo	Partidos Jugados	Ganados	Perdidos	Goles Favor	Goles Contra	Diferencia	Fase Alcanzada
Leones FC	2	2	0	4	1	+3	Final
Águilas United	2	1	1	2	1	+1	Semifinal
Cóndores FC	1	0	1	1	3	-2	Cuartos
Tigres Academy	1	0	1	0	2	-2	Cuartos

Reporte de Goleadores

Ranking de los jugadores que más goles han anotado en el torneo.

Posición	Jugador	Equipo	Goles	Minutos de Gol
1	Pedro Martínez	Leones FC	2	15', 45'
2	Diego Ramírez	Águilas United	1	22'
2	Sofia Hernández	Águilas United	1	75'
2	Mario López	Leones FC	1	34'

2	Luis García	Leones FC	1	88'
2	Valeria Cruz	Cóndores FC	1	67'

Reporte de Información General del Torneo

Resumen completo con datos estadísticos del torneo.

Estadística	Valor
Nombre del Torneo	Copa Mundial Universitaria
Sede	Guatemala
Equipos Participantes	4
Total de Partidos Programados	3
Partidos Completados	2
Total de Goles	6
Promedio de Goles por Partido	3.0
Edad Promedio de Jugadores	22.06 años
Fase Actual	Semifinal

Ejemplo Código Entrada

```
TORNEO {
  nombre: "Copa Mundial Universitaria",
  equipos: 8,
  sede: "Guatemala"
};

EQUIPOS {
  equipo: "Leones FC" [
    jugador: "Carlos Ruiz" [posicion: "PORTERO", numero: 1, edad: 22],
    jugador: "Luis García" [posicion: "DEFENSA", numero: 4, edad: 21],
  ];

  equipo: "Águilas United" [
    jugador: "Juan Pérez" [posicion: "PORTERO", numero: 1, edad: 24],
    jugador: "Ana Morales" [posicion: "DEFENSA", numero: 2, edad: 19],
  ];

  equipo: "Tigres Academy" [
    jugador: "Roberto Silva" [posicion: "PORTERO", numero: 12, edad: 21],
    jugador: "Carmen Vargas" [posicion: "DEFENSA", numero: 5, edad: 20],
  ];

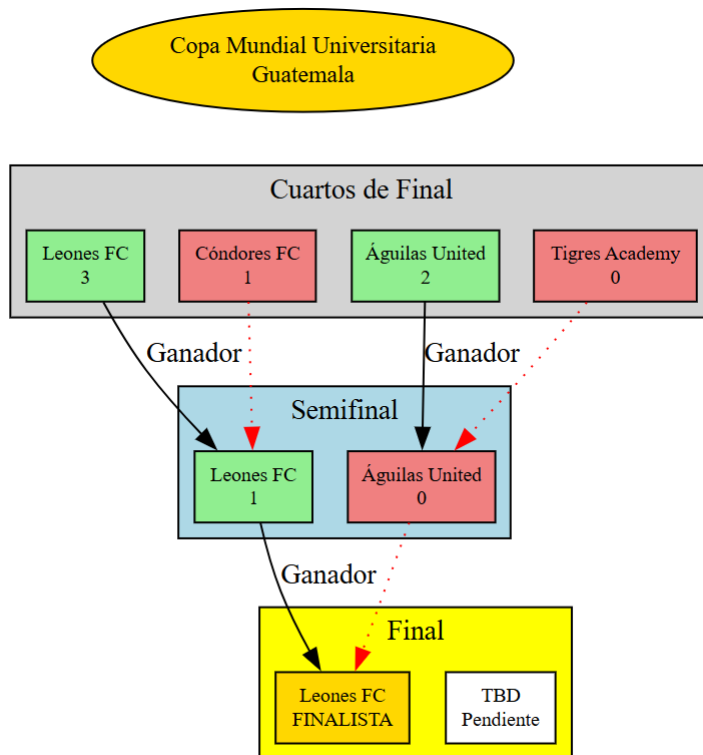
  equipo: "Cóndores FC" [
    jugador: "Fernando Díaz" [posicion: "PORTERO", numero: 1, edad: 23],
    jugador: "Patricia Luna" [posicion: "DEFENSA", numero: 3, edad: 21],
  ];
};

ELIMINACION {
  cuartos: [
    partido: "Leones FC" vs "Cóndores FC" [
      resultado: "3-1",
      goleadores: [
        goleador: "Pedro Martínez" [minuto: 15],
      ]
    ],
    partido: "Águilas United" vs "Tigres Academy" [
      resultado: "2-0",
      goleadores: [
        goleador: "Diego Ramírez" [minuto: 22],
      ]
    ]
  ];

  semifinal: [
    partido: "Leones FC" vs "Águilas United" [
      resultado: "1-0",
      goleadores: [
        goleador: "Pedro Martínez" [minuto: 45]
      ]
    ]
  ];
};
```

Graphviz Generado:

Adicional a esto deberán generar un reporte en Graphviz que represente los resultados y juegos del torneo de la siguiente manera:



4.2 Entregables

Ejemplo:

Tipo	Descripción
Manual de Usuario	Documento que explica paso a paso cómo utilizar la herramienta TourneyJS. Incluye cómo cargar archivos de torneos, interpretar los resultados léxicos, generar tablas de posiciones, visualizar brackets de eliminación, y exportar los resultados. Contiene capturas de pantalla, ejemplos prácticos y solución a errores comunes.
Manual Técnico	Informe que documenta el proceso completo de desarrollo. Describe la arquitectura del sistema, el funcionamiento del analizador léxico, el diseño del AFD, los algoritmos utilizados para

	la detección de tokens y errores, el cálculo de estadísticas deportivas, la integración con Graphviz y el diseño de la interfaz web. Incluye diagramas de clases, flujo de datos y decisiones de diseño
Código Fuente	El código completo del proyecto, con una estructura clara y comentada, que refleje las buenas prácticas de programación web y uso de control de versiones

5. Metodología

Para el desarrollo del proyecto TourneyJS, se propone una metodología en fases basada en el enfoque cascada, adaptado a proyectos individuales y académicos:

1. Análisis del Lenguaje y Requisitos

Estudio de la estructura sintáctica del lenguaje definido para describir torneos deportivos, identificando palabras clave, símbolos, reglas de agrupación y posibles errores léxicos.

2. Diseño del AFD y Diagrama del Sistema

Diseño del autómata finito determinista que servirá como base del analizador léxico. Se desarrollarán diagramas de flujo para representar el procesamiento de tokens y errores.

3. Implementación del Analizador Léxico

Programación manual del escáner léxico en JavaScript, caracterizado por el uso explícito de estados. Se desarrollará también la interfaz web y se integrará con Graphviz.

4. Generación de Salidas

Desarrollo de módulos para calcular automáticamente las tablas de posiciones y estadísticas, y exportar los diagramas de brackets. Integración de estas salidas en la interfaz web para que el usuario pueda visualizarlas y exportarlas.

5. Pruebas, Corrección y Documentación

Ejecución de casos de prueba válidos e inválidos. Registro de errores léxicos y verificación de resultados. Redacción del manual técnico y del manual de usuario.

6. Desarrollo de Habilidades Blandas

6.1 Proyectos Individuales

El trabajo individual en TourneyJS promueve la **autonomía y la toma de decisiones técnicas** exigiendo al estudiante asumir la totalidad del ciclo de vida del proyecto.

6.2.1 Autogestión del Tiempo

El estudiante planifica y organiza su cronograma personal para cumplir con las fases del desarrollo, entrega y calificación, lo que fortalece su capacidad de organización y responsabilidad.

6.2.2 Responsabilidad y Compromiso

Al ser el único responsable de su producto, el estudiante asume compromisos técnicos y éticos. Esto lo obliga a cumplir estándares de calidad, puntualidad y claridad.

6.2.3 Resolución de Problemas

La implementación manual del analizador léxico, sin herramientas externas, fomenta la búsqueda de soluciones lógicas, estructuradas y autónomas ante desafíos de diseño, sintaxis o integración.

6.2.4 Reflexión Personal

Al finalizar el proyecto, el estudiante evalúa su desempeño, identifica fortalezas y áreas de mejora. Este proceso de reflexión fortalece su capacidad de autoaprendizaje continuo.

7. Cronograma

El siguiente cronograma define las etapas principales del proyecto, junto con los plazos asignados para su desarrollo. Cada fase incluye fechas de inicio y finalización estimadas, permitiendo que el estudiante planifique, ejecute y entregue su proyecto de forma organizada.

Tipo	Fecha Inicio	Fecha Fin
Asignación de Proyecto	25/08/2025	21/09/2025
Calificación	26/09/2025	27/09/2025

8. Rúbrica de Calificación

8.1 Requisitos para optar a la calificación

Antes de la evaluación del proyecto, los estudiantes deben cumplir con los requisitos que se indiquen en esta sección.

Tema	Descripción	Cumple (Sí/No)
Cumplimiento de la tecnología establecida	El desarrollo debe haberse realizado en el lenguaje de programación JavaScript , sin uso de herramientas automáticas para el análisis léxico.	
Uso de herramientas requeridas	Se utilizó interfaz. No se emplearon frameworks externos como JFlex, ANTLR, etc.	
Gestión y entregas del proyecto	Todas las fases deben estar presentes en el repositorio de GitHub (incluyendo commits, versiones y entregables en carpeta Proyecto 1).	
Documentación obligatoria	Se entrega manual de usuario y manual técnico, junto con diagramas de arquitectura del sistema.	
Pruebas y funcionalidad mínima	El sistema debe funcionar correctamente: cargar archivo, extraer tokens, generar reportes, etc. Se deben presentar evidencias de pruebas.	
Otros	El código debe estar comentado, con estructura modular, y no debe haberse compartido entre estudiantes (originalidad garantizada).	

8.4 Valores

Durante el desarrollo del proyecto **TourneyJS**, se espera que cada estudiante actúe con integridad, compromiso y profesionalismo. Para garantizar una evaluación justa y una experiencia de aprendizaje significativa, se establecen los siguientes principios fundamentales:

1. Originalidad del Trabajo

- Cada estudiante debe desarrollar su propio proyecto, partiendo de cero y aplicando los conocimientos adquiridos durante el curso.
- Tanto el código fuente como la documentación entregada deben ser resultado del trabajo personal e individual, conforme a los lineamientos establecidos por el curso.

2. Prohibición de Copias y Plagio

- **Se prohíbe estrictamente** la copia total o parcial del código, documentación o cualquier otro entregable de:
 - Compañeros del curso.
 - Semestres anteriores.
 - Fuentes externas no referenciadas.
- Cualquier evidencia de plagio resultará en una **calificación de 0 puntos**, sin excepción.

3. Uso Responsable de Recursos Externos

- Está permitido utilizar bibliotecas o recursos externos únicamente cuando:
 - Se mencionan claramente sus fuentes.
 - Se comprende completamente su funcionamiento.
 - **No se usan herramientas automáticas** para generar analizadores léxicos (como ANTLR, JFlex, Lex o similares), conforme a lo indicado en las restricciones del proyecto.
- Se recomienda consultar con el catedrático si existe duda sobre la validez del uso de algún recurso.

4. Revisión y Detección de Plagio

- El proyecto podrá ser sometido a herramientas automáticas de detección de plagio y a revisiones manuales por parte del equipo docente.
- En caso de sospecha:

- El estudiante deberá explicar y defender el funcionamiento de su código y demostrar su autoría.
 - Si no logra justificar su trabajo adecuadamente, se **anulará su proyecto y se asignará una calificación de 0 puntos**.
- Cualquier irregularidad será reportada al catedrático para que determine las acciones correspondientes, las cuales pueden incluir **notificación a la Escuela de Ciencias y Sistemas**.

9. Entrega

- Link al repositorio de GitHub en el apartado de UEDI.
- En el repositorio de GitHub debe de encontrarse lo siguiente:
 - Manual de Usuario
 - Manual Técnico
 - Código Fuente
- Se debe agregar como colaborador al auxiliar de la sección correspondiente.
 - A+ bryan967132
 - A- ATPCOXBAU
 - B+ kevinmp2
 - B- riverroman

10. Restricciones

- El lenguaje de programación a utilizar es JavaScript.
- La aplicación se debe desarrollar en consola.
- El manual técnico deberá contener por lo menos un diagrama de flujo y diagrama de clases para la solución.
- La práctica debe realizarse de forma individual.
- El análisis de la entrada de texto debe de realizarse manipulando la cadena únicamente con funciones nativas del lenguaje.
- Queda prohibida cualquier librería que facilite el análisis de la entrada de texto.
- El nombre del repositorio debe tener el siguiente formato **LFP_2S2025_#Carnet**, dentro del repositorio deberá crear una carpeta llamada **Proyecto 1**.
- Se debe de agregar como colaborador al auxiliar de la sección correspondiente.
- La calificación se realizará de forma **PRESENCIAL**.
- Cualquier indicio de copia será motivo para realizar una revisión exhaustiva.
- **EN CASO DE ENCONTRAR COPIAS PARCIALES O TOTALES AUTOMÁTICAMENTE SE ANULARÁ LA PRÁCTICA, SE NOTIFICARÁ AL CATEDRÁTICO DEL CURSO Y SE REPORTARÁ A ESCUELA PARA APLICAR LAS SANSIONES CORRESPONDIENTES.**

Fecha de entrega: 21 de septiembre de 2025 antes de las 23:59, no se recibirán entregas después de la fecha y hora límite establecidas.