



Vacaciones de junio 2025

Sección	Catedrático	Tutor Académico
P	Inga. Asuncion Mariana Sic Sor	Elian Saúl Estrada Urbina

ENUNCIADO DE LA PRÁCTICA

Objetivos:

- Implementar un analizador léxico básico para poder obtener información de un archivo de entrada.
- Entender e implementar el modelo matemático de AFD para el reconocimiento de patrones.
- Comprender la diferencia entre Patrón, Token y Lexema.
- Familiarizar al estudiante con el lenguaje de programación TypeScript

Descripción:

Durante los últimos años la gran compañía de videojuegos “Game Freak” conocida mayormente por ser la creadora del mundialmente conocido juego de Pokémon, ha estado buscando nuevas formas de ayudar a la comunidad de jugadores competitivos, creando para ello un programa que según un listado de Pokémons y sus estadísticas poder escoger de ella los mejores 6 teniendo en cuenta ciertas consideraciones.

A pesar de todos sus esfuerzos, aun no han logrado avanzar con el proyecto debido que todo su equipo de desarrollo esta enfocado en el lanzamiento de su nuevo juego, por lo que, Game Freak lo ha contratado a usted para que desarrolle un programa que analice un lenguaje específico (que se define más adelante) con el cual se analizarán todos los Pokémons de un listado y se escogerán los seis candidatos más aptos. La aplicación deberá ser desarrollada en el lenguaje de programación TypeScript y consumir la PokeApi para obtener los Sprites de los pokémons a escoger (Este API se detalla más adelante).

La elección de los Pokémons se realizará a partir del contenido del editor, este editor puede ser llenado directamente, o a través de la carga de un archivo con extensión: .pk1fp.

La aplicación debe proveer la opción de analizar el contenido del editor, y deberá generar como salida un gráfico con el nombre del jugador y sus seis Pokémons seleccionados y sus tipos, así como también, una tabla de tokens indicando el tipo, su valor, fila y columna; en caso de que el archivo contenga errores, mostrará una página HTML con el detalle de estos.

Definición del lenguaje

Dentro del enunciado se usarán las siguientes notaciones para explicar de mejor manera la composición del archivo de entrada.

Color	Token
Azul	Palabras Reservadas
Naranja	Cadenas de Texto
Morado	Números
Negro	Otros

El lenguaje contara con los siguientes bloques:

```
Jugador: <texto> {  
  <bloque_pokemons>  
}
```

El bloque mas alto será el Jugador, que recibirá el nombre de este.

```
<text>[<tipo>] := (  
  <bloque_stats>  
)
```

Este bloque recibirá el nombre del Pokémon seguido por su tipo.

```
[salud]=numero;  
[ataque]=numero;  
[defensa]=numero;
```

En este bloque se indican las estadísticas del Pokémon

Dentro de un bloque Jugador pueden venir uno o más Pokémons.

La etiqueta <tipo> engloba palabras reservadas para los tipos de los Pokémons, los cuales estarán limitados y serán los siguientes:

- agua
- dragon
- planta
- psiquico
- fuego
- normal

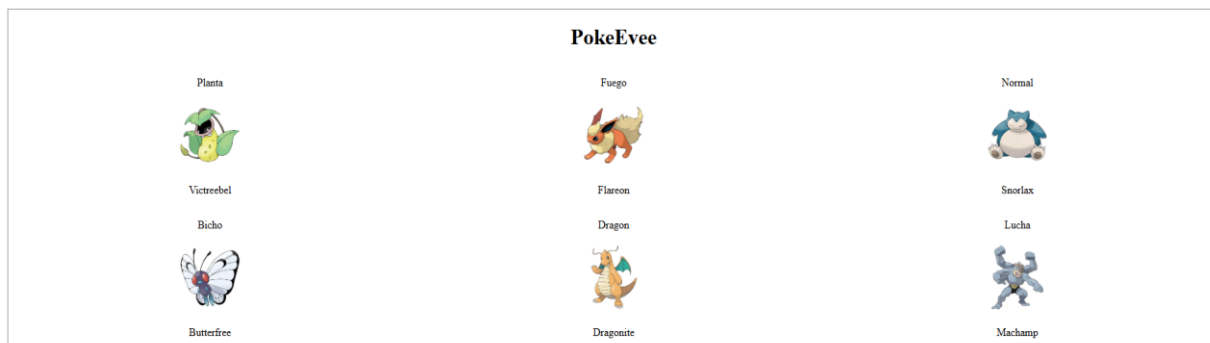
Estas palabras reservadas deberán agregarse a su AFD tal y como están aquí, en minúsculas y sin tildes.

A Continuación, se muestra un archivo de entrada, el cual contiene un jugador con 7 pokémons:

Editor de texto con resaltado de sintaxis

```
Jugador: "PokeEvee" {  
    "venusaur"[planta] := (  
        [salud]=12;  
        [ataque]=11;  
        [defensa]=15;  
    )  
    "dragonite"[dragon] := (  
        [salud]=10;  
        [ataque]=15;  
        [defensa]=14;  
    )  
    "butterfree"[bicho] := (  
        [salud]=14;  
        [ataque]=10;  
        [defensa]=15;  
    )  
    "snorlax"[normal] := (  
        [salud]=15;  
        [ataque]=12;  
        [defensa]=14;  
    )  
    "machop"[lucha] := (  
        [salud]=14;  
        [ataque]=13;  
        [defensa]=11;  
    )  
    "victreebel"[planta] := (  
        [salud]=15;  
        [ataque]=14;  
        [defensa]=14;  
    )  
    "flareon"[fuego] := (  
        [salud]=12;  
        [ataque]=15;  
        [defensa]=14;  
    )  
}
```

Al ejecutar las instrucciones anteriores en el editor, se generaría una página HTML por Jugador que mostrará algo como lo siguiente:



Como se puede observar el Jugador “PokeEvee” de su lista de siete Pokémons solamente escogimos seis, a los cuales mediante la PokeApi conseguimos sus sprites.

Calculo de IV's y Consumo de la PokeAPI:

Para el calculo de los IV's que son los valores individuales de cada Pokémon para ello se utiliza la siguiente fórmula:

$$IV's = \frac{salud+ataque+defensa}{45} * 100$$

Donde:

- **IV's:** Es el indicador de “Calidad” del Pokémon representado como un porcentaje (%)
- **Salud:** Es una de las estadísticas del Pokémon que leímos en el archivo.
- **Ataque:** Es una de las estadísticas del Pokémon que leímos en el archivo.
- **Defensa:** Es una de las estadísticas del Pokémon que leímos en el archivo.

Una vez obtenidos los IV's de cada Pokémon debemos escoger a los seis mejores, procurar asegurarse de que el tipo no se repita, si el tipo se repite se tomará al Pokémon del tipo con mejor IV's.

Para el consumo de la PokeAPI utilizaremos el siguiente enlace:

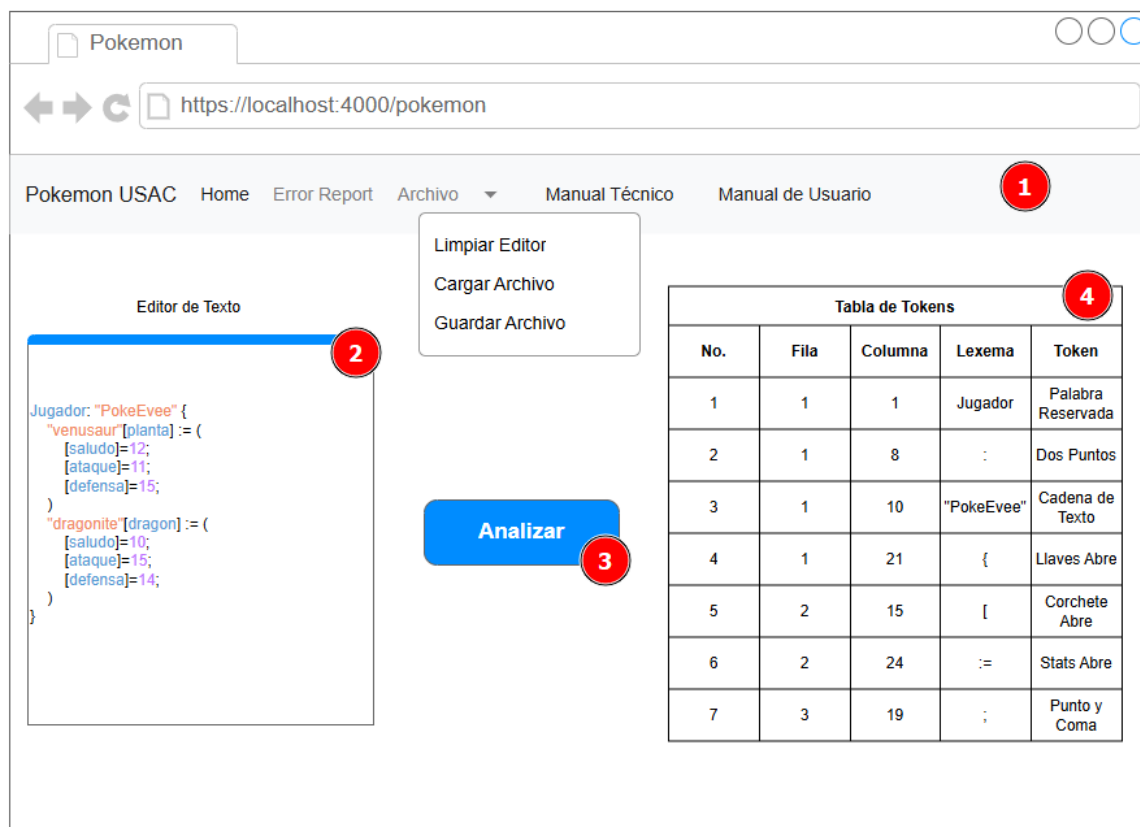
<https://pokeapi.co/api/v2/pokemon/>

En esta url, debermos agregar al final el nombre del Pokémon para obtener toda su data, ejemplo:

<https://pokeapi.co/api/v2/pokemon/venusaur>

La obtención del Sprite se explicará en el laboratorio.

Interfaz Gráfica:



Se muestra una propuesta de interfaz gráfica:

- Barra de Navegación:**
 - Home:** Recargará la página
 - Error Report:** Esta opción abrirá la tabla de errores léxicos si los hubiera.
 - Archivo:** Este menú tiene varias opciones para el manejo de los archivos:
 - Limpiar Editor:** Dejará el editor en blanco
 - Cargar Archivo:** Permitirá al usuario cargar un archivo de extensión .pklfp
 - Guardar Archivo:** Permitirá al usuario guardar el contenido actual del editor en un archivo de extensión: .pklfp.
- Editor de Texto:** En esta área es posible editar el texto que será analizado para cargar los Pokémons, es importante mencionar que se debe resaltar, con los colores mencionados al inicio de la definición del lenguaje, los diferentes tokens reconocidos.
- Botón de Analizar:** Al presionar el botón "Analizar" se procederá a realizar el análisis léxico del texto en el editor para posteriormente crear las páginas para mostrar el equipo de seis Pokémons que se deberá de abrir automáticamente, así como la página de errores léxicos si los hubiera, y también la tabla de Tokens se debe limpiar y llenar con los tokens reconocidos en este análisis.

4. **Tabla de Tokens:** En esta tabla se deben listar todos los tokens reconocidos durante el análisis, cabe destacar que en cada ejecución se deberá limpiar la tabla y mostrar el listado del input actual.

Reportes:

El reporte de errores léxicos, si existieran, se debe mostrar en una página HTML con la información en una tabla como se muestra a continuación:

#	Fila	Columna	Carácter	Descripción
1	05	10		Desconocido
2	08	30	¬	Desconocido
3	10	05	~	Desconocido

El reporte de Tokens como se mostro en la interfaz se debe mostrar en una tabla como la siguiente:

#	Fila	Columna	Lexema	Token
1	01	01	Jugador	Palabra Reservada
2	02	05	“venusaur”	Cadena de Texto
3	04	18	11	Número Entero

Entregables:

- **Manual de Usuario:** Explicación de cómo utilizar la aplicación.
- **Manual Técnico:** Debe explicar la lógica que usó para construir su analizador y demás aspectos, en este documento debe incluir el Autómata Finito Determinista que utilizó para la implementación de su Analizador Léxico.
- **Código Fuente:** Link del repositorio

Notas Importantes:

- El proyecto se deberá desarrollar de forma individual.
- Este proyecto se deberá desarrollar utilizando TypeScript
- La entrega se realizará en Uedi
- El proceso de obtener tokens, se debe hacer a través de la implementación del autómata finito determinista desarrollado por el propio estudiante.
- **Copia parcial o total del proyecto tendrá una nota de 0 puntos, y se notificará a la escuela de sistemas para que se apliquen las sanciones correspondientes.**
- **En el caso de no cumplir con alguna de las indicaciones antes mencionadas, no se calificará el proyecto; por lo cual, se tendrá una nota de cero puntos.**

Fecha de entrega: 08 de junio de 2025 a las 23:59