

Ingeniero	Auxiliar
Luis Espino	Robin Rolando Navarro Dionicio

## Proyecto 2 (Análisis Sintáctico)

### Objetivos

- Realizar la gramática respectiva para el análisis Sintáctico
- Retornar los errores durante el análisis sintáctico.
- Realizar la recuperación de errores sintácticos, durante la ejecución.

### Descripción

La empresa CompiuGames ha quedado complacido con la primera entrega de dicho software, por lo que ahora se le ha confiado la elaboración de la fase 2 de dicho proyecto, en donde se basara en la entrega de la fase 1, en donde ahora se le ha pedido que realice un minicompilador que pueda reconocer la estructura definida por los archivos de entrada utilizando una gramática definida por su persona, en donde la estructura se compone de elementos esenciales que se desglosa de la siguiente manera:

- Uno o varios módulos principales con la etiqueta de **configuracion**, dentro del cual se espera al menos un submódulo.
- 3 tipos de submódulos, con las etiquetas de **Juego, Usuario, Sonido y Carta**, donde dichos módulos son los encargados de la funcionalidad del juego, por lo que estos no pueden cambiar.
- Dentro del submódulo de **Juego** pueden venir los submódulos **Niveles y Tiempo**, donde cada uno de ellos son los encargados de las funcionalidades básicas del proyecto, por lo que estos no pueden cambiar.

La estructura del archivo de entrada se especifica dentro de la tabla No. 1, en donde dicho orden puede variar (esto aplica a los submódulos y el contenido de cada submódulo), además de ello dicho minicompilador no se considera la diferencia de mayúsculas y minúsculas, ya que los archivos de entrada que especifica el usuario no son CaseSensitive, por lo que las palabras reservadas (palabras únicas dentro del lenguaje) pueden encontrarse en mayúsculas y minúsculas.

Por lo cual se solicita que para dicha fase el estudiante, pueda realizar una gramática (Tipo 3) que se adecue a los archivos de entrada, para realizar el análisis sintáctico, ya que de no venir con la estructura definida, dicho sistema debe de tener la capacidad de poder reconocer dichos errores y almacenarlos en la tabla descrita mas adelante. Por lo que es de suma importancia dicho orden para que el Juego pueda ser ejecutado.

## Estructura del archivo de entrada:

```
[Configuracion]
  [Juego]
    [Nivel]
      [Facil] -> [2.2*2.6 X 3.5-0.5]
      [Dificil] -> [3/3*6 X 6.0]
      [Intermedio] -> [3.5 * 8 X 6+3]
    [End-Nivel]
  [Tiempo]
    [Facil] -> [3*3]
    [Dificil] -> [6/2*6.3]
    [Intermedio] -> [9+1.6]
  [End-Tiempo]
[End-Juego]
[Usuarios]
  [Nombre] -> ["Juan"]
  [Nombre] -> ["Carlos"]
  [Nombre] -> ["Steven"]
  [Nombre] -> ["Mario"]
[End-Usuarios]
[Sonido]
  [Track_1] -> ["C:\Usuario\Juan\Documentos\cancion1"]
  [Track_2] -> ["C:\Usuario\Juan\Documentos\cancion2"]
  [Track_3] -> ["C:\Usuario\Juan\Documentos\cancion3"]
[End-Sonido]
[Carta]
  [Nombre] -> ["Carta_1"]
  [Imagen] -> ["C:\Usuario\Juan\Documentos\Imagen_1"]
  [Idioma] -> ["Ingles"]
  [Palabra] -> ["Tiger"]
[End-Carta]
[Carta]
  [Nombre] -> ["Carta_2"]
  [Imagen] -> ["C:\Usuario\Juan\Documentos\Imagen_2"]
  [Idioma] -> ["Aleman"]
  [Palabra] -> ["Katze"]
[End-Carta]
[End-Configuracion]
```

Tabla No. 1, estructura del archivo de entrada

Para lo cual se tiene en cuenta que ya se ha realizado el análisis léxico, se restringe la utilización del **método LL(1)** para realizar el análisis sintáctico correspondiente. A continuación, se especifica el orden de cada submódulo para que sea tomado en cuenta, al momento de la elaboración de la gramática.

## 1. Definición de la estructura de los Módulos y Submódulos

### 1.1. Módulo Configuración

Dicho modulo, es el modulo principal del Juego, sin dicho modulo no se podría construir todo el juego por lo que es de suma importancia que el mismo pueda venir dentro del archivo de entrada.

Dicho modulo debe de tener al menos un submódulo (que serán especificado más adelante), ya que no puede venir vacío para que sea válido, además dicho modulo debe de venir dentro de la etiqueta apertura [ y finalizar con la etiqueta de cierre ].

Además, el módulo principal puede venir varias veces dentro del archivo, si se desea agregar mas palabras, canciones o usuarios dentro del juego. La palabra **configuracion** es una palabra reservada por lo que no puede cambiar.

### Modulo Configuracion

[Configuracion	Etiqueta de apertura para el módulo de configuración
End-Configuracion]	Etiqueta de cierre para el módulo de configuración

## 1.2. Submódulo Juego

Dicho submódulo es el encargado de darle la funcionalidad total del juego sin dicho modulo el juego no podría funcionar en su totalidad. Dicho modulo **no es obligatorio** por lo que puede o no venir dentro del archivo de entrada, y solamente venir los submódulos de **Usuario**, **Sonido** y **Carta** dentro del módulo de **configuracion**.

Dentro de dicho submódulo, deben de venir al menos un submódulo, como lo son Niveles y Tiempo, pero no puede venir vacío.

La estructura de dicho modulo se especifica a continuación:

### Submódulo Juego

[Juego	Etiqueta de apertura para el módulo de Juego
End-Juego]	Etiqueta de cierre para el módulo Juego

### 1.2.1. Submódulo Niveles

Dicho submódulo tiene la funcionalidad de poder asignarle un nivel al juego, dependiendo del que el archivo de entrada pueda especificar. Además, cada nivel es especificado por un número o una operación aritmética, el cual es el indicador de la matriz de imágenes que serán escogidas, para luego ser traducidas por el usuario.

Dicho modulo contienen las palabras reservadas **Niveles**, **Facil**, **Intermedio**, **Difícil**, los cuales son palabras reservadas y no podrán cambiar ni ser modificadas.

Dichas palabras reservadas vienen dentro de las etiquetas de [ , ], por lo que dichas etiquetas deben de ser consideradas dentro de la gramática.

Dichos niveles pueden venir en diferente orden dentro del submódulo y además pueden venir en otros módulos **configuracion** por lo que se deben de considerar los últimos submódulos niveles encontrados. Se debe de considerar además que dicho submódulo no puede venir vacío.

La estructura de dicho modulo se especifica a continuación:

## Submódulo Niveles

[Niveles]	Etiqueta de apertura para el submódulo de Niveles
End-Niveles]	Etiqueta de cierre para el módulo Niveles
<b>Niveles</b>	
[Facil] -> [2.2*2.6 X 3.5-0.5]	Etiqueta del nivel fácil
[Intermedio] -> [3.5 * 8 X 6+3]	Etiqueta de nivel intermedio
[Dificil] -> [3/3*6 X 6.0]	Etiqueta de nivel difícil

**Nota:** los números pueden ser enteros o decimales, por lo que dentro del juego se deben de considerar que los números a tomar es la parte entera de los mismos.

## 1.2.2. Submódulo Tiempo

Dicho submódulo es el encargado de colocarle tiempo a cada nivel dentro del juego, dicho tiempo estará especificado en segundos, ya que esto es un plus de velocidad dentro del juego, ya que el juego estará en funcionamiento mientras no se ha cumplido el tiempo establecido, luego de eso no será posible ingresar ninguna traducción. Dicho modulo no puede ser vacío, por lo que debe de venir al menos un tiempo.

Además, dicho modulo se establece dependiendo del nivel escogido por el usuario, por lo que se debe de verificar que el nivel exista para que se le asigne el tiempo, caso contrario, se establecerá un tiempo por defecto.

Dicho modulo contienen las palabras reservadas **Tiempo**, **End-Tiempo**, **Facil**, **Intermedio**, **Dificil**, los cuales son palabras reservadas y no podrán cambiar ni ser modificadas.

Dichas palabras reservadas vienen dentro de las etiquetas de [, por lo que dichas etiquetas deben de ser consideradas dentro de la gramática. Además, dichos tiempos pueden venir en cualquier orden y/o en cualquier módulo de **configuracion**, por lo que deben de considerarse los últimos submódulos de tiempo encontrados.

La estructura de dicho submódulo se especifica a continuación:

## Submódulo Tiempo

[Tiempo]	Etiqueta de apertura para el submódulo de Tiempo
End-Tiempo]	Etiqueta de cierre para el módulo Tiempo
<b>Tiempos por nivel</b>	
[Facil] -> [3*3]	Etiqueta del tiempo en nivel fácil
[Intermedio] -> [9+1.6]	Etiqueta del tiempo en nivel intermedio
[Dificil] -> [6/2*6.3]	Etiqueta del tiempo en nivel difícil

**Nota:** los números pueden ser enteros o decimales, por lo que dentro del juego se deben de considerar que los números a tomar es la parte entera de los mismos.

### 1.2.3. Submódulo Sonido

Dicho submódulo es el encargado de amenizar el ambiente del juego, esto con el objetivo de que mientras el usuario pueda jugar, se reproduzca una melodía (dicha melodía queda a discreción del estudiante), dicha melodía se debe de reproducir una luego de otra (*en caso de que el tiempo fuera relativamente grande o que la melodía sea corta*), considerando que existan más submódulos de sonido.

Cada submódulo contendrá las etiquetas de nombre de la melodía (Track\_1), y su respectiva ruta. Dichas etiquetas vienen dentro de las etiquetas [..].

El orden de las etiquetas se comporta de manera aleatoria, por lo que se debe de considerar dentro de la gramática realizada. El modulo de sonido no puede venir vacío (tomarlo en cuenta que debe de existir al menos una melodía).

La estructura será especificada a continuación:

#### Submódulo Sonido

[Sonido	Etiqueta de apertura para el módulo de Sonido
End-Sonido]	Etiqueta de cierre para el módulo Sonido
[Track_1] -> ["C:\Usuario\Juan\Documentos\cancion1"]	Etiqueta de la ruta de la melodía

### 1.3. Submódulo Usuario

Dicho submódulo es el encargado de cargar a los usuarios y los cuales deben de aparecer dentro de la interfaz inicial de juego en donde se escogen los niveles, por lo cual se deben de considerar todos los usuarios que exista dentro del archivo de entrada y dentro de los módulos de configuracion que puedan existir.

De igual manera dichos submódulo contiene las palabras reservadas Usuario y Nombre, los cuales no podrán cambiar ni ser modificadas. Dicho submódulo no podrá venir vacío, ya que al menos debe de existir algún usuario cargado. Dichas etiquetas vienen dentro de las etiquetas [..].

La estructura de dicho submódulo se especifica a continuación:

#### Submódulo Usuario

[Usuario	Etiqueta de apertura para el módulo de Usuario
End-Usuario]	Etiqueta de cierre para el módulo Usuario
[Nombre] -> ["Juan"]	Etiqueta del nombre del Usuario.

## 1.4. Submódulo Carta

Dicho submódulo es el corazón del juego, sin dicho modulo, el juego no se podrá ejecutar, por lo que dicho modulo no puede venir vacío, debe de venir al menos una palabra.

Dicho submódulo es el encargado de almacenar todas las palabras que serán de utilidad dentro del juego, por lo que es de suma importancia, dentro de dicho submódulo se encuentran las etiquetas de: Nombre, Imagen, Idioma, Palabra las cuales son palabras reservadas y no podrán ser modificadas o cambiadas.

Las etiquetas serán especificadas a continuación:

- **Nombre:** dicha etiqueta es la encargada de especificar el de la carta a colocar.
- **Imagen:** dicha etiqueta especifica la ruta de la imagen a colocar dentro de la matriz.
- **Idioma:** dicha etiqueta especifica el idioma que será traducida una palabra, que representa la imagen que se colocará dentro de la matriz.
- **Palabra:** dicha etiqueta es la encargada de contener la traducción de la palabra relacionada a la imagen y en el idioma que fue escogido anteriormente.

El orden de dichas etiquetas, se presentan en forma ordenada, por lo cual se debe de tomar en cuenta dentro de la gramática.

La estructura de dicho submódulo se especifica a continuación:

### Submódulo Vocabulario

[Vocabulario	Etiqueta de apertura para el módulo de Vocabulario.
End-Vocabulario]	Etiqueta de cierre para el módulo Vocabulario.
[Nombre] -> ["Carta_1"]	Etiqueta del Nombre.
[Imagen] -> ["C:\Usuario\Juan\Documentos\Imagen_1"]	Etiqueta de Imagen.
[Idioma] -> ["Ingles"]	Etiqueta del Idioma.
[Palabra] -> ["Tiger"]	Etiqueta de la Palabra.

## 2. Detección de errores Léxicos y Sintácticos

### 2.1. Tabla de símbolos

Dicha tabla es la que fue generada dentro del Proyecto No. 1, por lo cual mantiene la misma estructura, dicha tabla será lanzada si no existe errores dentro del archivo de entrada. Y se genera en un archivo con formato HTML.

La estructura de la tabla se especifica a continuación:

No.	lexema	Token	Fila	Columna
1	Configuracion	Token_Reservada	3	5
2	[	Token_	8	10

## 2.2. Tabla de Errores Sintácticos

Dicha tabla es la que es generada luego de realizar el análisis sintáctico con el Método LL(1), para lo cual se debe de determinar que no existan errores léxicos para que se pueda generar el análisis sintáctico, si existiera errores léxicos no se deberá de generar el análisis sintáctico. Y dicha tabla se mostrar en un archivo con formato HTML.

La estructura para dicho archivo se especifica a continuación.

No.	lexema	Fila	Columna	Tipo de error	Descripción
1	Var1	3	5	Error Sintáctico.	Se esperaba =
2	.jpg	8	10	Error Sintáctico.	Se esperaba .jpg

## 3. Interfaz de Usuario

En esta sección será especificada la interfaz de Usuario, la cual es la encargada de analizar el archivo de entrada de manera Léxica y Sintácticamente, por lo que se sugiere una interfaz al final del enunciado (**Ver anexos**).

Dicha interfaz contiene las siguientes opciones:

### 3.1. Área de texto:

Dicha área es en la que se cargan y modifican los archivos de entrada, generados por el usuario, para posteriormente ser analizados.

### 3.2. Botón de abrir y Analizar Archivo

Dicho botón tiene la funcionalidad de facilitar la carga de los archivos utilizados para realizar el respectivo análisis. Dicho botón no es obligatorio por lo que queda a criterio del estudiante el poder implementarlo. Los archivos que se cargaran tendrán la extensión “:LFP”, además es el utilizado en la fase anterior, por lo que es el encargado de evaluar el archivo de entrada de manera léxica, y lanzar posteriormente la tabla de símbolos si no existe ningún error y caso contrario se lanzara la tabla de errores léxicos.

### 3.3. Botón de Análisis Sintáctico

Dicho botón es el encargado de realizar el análisis sintáctico (LL(1)), al archivo de entrada, con los datos generados en la tabla de símbolos, por lo que posteriormente lanzara la tabla de errores sintácticos en caso que se encontraran errores.

Se debe de tomar en cuenta que, dentro del análisis, se necesita que dicho análisis se recupere de errores, por lo que dentro de la tabla de errores sintácticos se deben de mostrar cada uno de los errores encontrados durante los análisis sintácticos.

### 3.4. Botón de inicio del juego

Dicho botón es el encargado de iniciar el juego. Dicho botón será activado luego de realizar los dos análisis y de no haberse encontrado ningún error en ambos. Por lo que este botón lanzara la interfaz de inicio del Juego.

## 4. Funcionalidad

### 4.1. Interfaz de Inicio

Dicha interfaz es la que se ejecuta luego de presionar el botón de inicio de juego. Dicha interfaz contendrá los datos obtenidos del archivo de entrada, tanto de los usuarios como de los niveles establecidos.

Teniendo en cuenta que los usuarios que se visualizan serán los que se encuentren dentro del archivo de entrada, así como los niveles encontrados.

Se propone una interfaz de inicio, para su realización:

Proyecto Unico (Fase 2)

Ingrese / escoja un Nickname

▼  
Marcos  
Mario

Seleccione un Nivel  ▼  
Intermedio  
Dificil

Seleccione el Idioma  ▼  
Ingles  
Aleman

### 4.2. Interfaz del Juego y carga de los datos de la interfaz de inicio


Dicha interfaz es en donde se presentarán los datos cargados de la interfaz de inicio y de las palabras contenidas dentro del archivo de entrada. Con la matriz especificada para cada nivel, por lo que se deben presentar las cartas aleatoriamente.

Se debe de mostrar las cartas colocadas al frente con su respectivo espacio para poder ingresar la palabra correcta, así como el tiempo especificado, dependiendo del nivel escogido. Además, se debe de colocar el idioma en que ha sido especificado, así como una imagen referente a dicho idioma, así como colocar el título de la canción que se encuentra sonando en ese momento.

Se sugiere una interfaz para la formación de dicho juego:



Proyecto 2



Frances

Nivel

Intermedio




Usuario









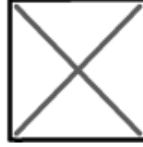
Pedro

Tiempo

45 Seg

musica.mp3






 <input type="text"/>	 <input type="text"/>	 <input type="text"/>
 <input type="text"/>	 <input type="text"/>	 <input type="text"/>
 <input type="text"/>	 <input type="text"/>	 <input type="text"/>

Finalizar

El juego terminara cuando se haya presionado el botón de finalizar o bien se haya terminado el tiempo, para luego calificar cada una de las palabras dentro del juego, y colocarle un identificativo de la palabra para saber si esta correcta o no. Se sugiere la siguiente interfaz.

Proyecto 2



Frances

Nivel

Intermedio




Usuario










Pedro

Tiempo

45 Seg

musica.mp3

 <input type="text"/>	 <input type="text"/>	 <input type="text"/>
 <input type="text"/>	 <input type="text"/>	 <input type="text"/>
 <input type="text"/>	 <input type="text"/>	 <input type="text"/>

Finalizar

### 4.3. Funcionalidad de Sonido

Dicha funcionalidad es la encargada de reproducir las melodías que se encuentren dentro del archivo de entrada, por lo que se debe verificar que dichas melodías existan durante el análisis sintáctico. Se debe tomar en cuenta que dentro del archivo de entrada pueden encontrarse varios archivos de música, por lo que el juego debe poder tener la capacidad de reproducir una lista de melodías.

### 4.4. Reportes

Como parte del proyecto se necesita que luego de haber finalizado dicho juego se despliegue un reporte, donde se muestran las estadísticas de los usuarios que han jugado recientemente, dicho reporte contendrá la estructura perteneciente a los archivos con formato JSON, se muestra a continuación un ejemplo de dicho reporte.

```
[Reporte]
  [Juego]
    [Usuario] = ["Pedro"]
    [Puntuación] = [55.55]
    [Nivel] = [Intermedio]
    [Tiempo] = [40]
    [Idioma] = ["Frances"]
    [Palabras Correctas] = [5]
    [Palabras Erróneas] = [9]
  [Fin-Juego]
  [Juego]
    [Usuario] = ["Marcos"]
    [Puntuación] = [60]
    [Nivel] = [Fácil]
    [Tiempo] = [20]
    [Idioma] = ["español"]
    [Palabras Correctas] = [3]
    [Palabras Erróneas] = [1]
  [Fin-Juego]
[Reporte]
```

## 5. Requisitos mínimos

Dentro de los requisitos mínimos se establecen los siguiente:

- Implementación del método LL(1)
- Archivos donde se verifique la implementación del LL(1)
- Tablas de reportes en HTML
- Funcionalidad (inciso 4)

## 6. Entregables

Los entregables serán los que se especifican a continuación:

- Archivos de verificación del método LL(1).
- Manuales de Usuario y Técnico.
- Ejecutable de la Aplicación.
- Código Fuente de la aplicación.

## 7. Notas importantes

- El proyecto se debe realizar de manera individual.
- Esta práctica se deberá desarrollar utilizando Java con Netbeans o Eclipse.
- El analizador léxico y sintáctico debe ser implementado por el estudiante, no está permitido el uso de ninguna herramienta.
- **Si se detecta copia parcial, total, o código de internet, los estudiantes involucrados tendrán una nota de 0 puntos y se notificará a la escuela 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.
- El método LL(1) debe de ser implementado, caso contrario no se calificará dicho proyecto

## 8. Fecha de entrega

**Fecha de entrega 4 Julio del 2018 antes de las 7:00 am**

La forma de entrega será como el proyecto 1 y se les proporcionará el link para que suban sus proyectos con todos los entregables antes mencionados.

## 9. Nexos

