



Ejercicio 1: Analizador léxico básico

## Ejercicio 1: Analizador léxico básico

### Conceptos a trabajar:

- Alfabeto
- Patrón y Lexema
- Analizador léxico

### Proyecto:

A través de GADUN, en la carpeta raíz de la herramienta busque en la sección de “GRAMÁTICAS\_EJERCICIOS” el proyecto “ejercicio1\_nombres\_apellidos.json”. Si no encuentra el directorio o el archivo por favor descargue los de [Repositorio GADUN](#).

### Introducción:

Como sabemos el analizador léxico de un compilador tiene la función de analizar el código fuente o entrante y transformarlo en una secuencia de componentes léxicos, siendo estas estructuras lógicas que le permiten al compilador entender los elementos que conforman el código entrante y de esta manera operar sobre el mismo. Para esto, como diseñador debes establecer el alfabeto que conforman las cadenas que integran el código fuente y que tu compilador deberá reconocer y esto se hace a través de patrones. En GADUN denominamos a estas reglas o patrones Tokens, siendo estas las clasificaciones o elementos con los que trabajara tu compilador. En el siguiente ejercicio se indica como GADUN define básicamente un analizador léxico.

Para este ejercicio crearemos un analizador léxico hipotético que permita reconocer nombres y apellidos de acuerdo a la lista dada a continuación

Nombres	Apellidos
Juan Carlos Nelson Carolina	Giménez Guerrero Campo Burbano

*Tabla de nombres y apellidos*

Observe que nuestro alfabeto inicial está conformado netamente por caracteres alfabéticos. Y en este caso debemos identificar dos grupos específicos, Nombres y Apellidos

## Proceso

1. Abre el proyecto indicado para esta guía y dirígete a la sección de definición “Analizador Léxico”, para este ejercicio solo nos centraremos en esta área.
2. Como podrá observar el proyecto ya contiene en Lista de Tokens, el Token de Nombres. Y este reconoce las cadenas de Nombres de la tabla de nombres y apellidos.

	Nombre	Expresión
1	Nombres	Juan,Carlos,Nelson,Carolina

3. Como ya tenemos la primera clasificación que conforma nuestro lenguaje solo debemos crear el token para Apellidos.

Para esto debe definir un Token de tipo conjunto que denominara Apellidos, y al que agregara las expresiones en la tabla de nombres y apellidos en la columna de Apellidos.

Token de Conjunto

Token de Expresión Regular

Nombre del Token:

Apellidos

Expresión:

Giménez

Agregar Expresión

Crear Token




Para verificar que este agregando las expresiones correctas al Token cada vez que presione Agregar Expresión verifique la lista lateral de “Elementos del Conjunto” donde podrá observar la lista de expresiones que reconocerá el Token.

Elementos del Conjunto
Giménez
Guerrero
Campo
Burbano

En el caso de agregar un elemento erróneo, simplemente eliminarlo con el botón (-) y vuelva a agregarlo. Cuando termine de agregar las expresiones presione Crear Token para Crear el Token de Apellidos.

4. Con los patrones definidos para el analizador léxico, ejecute el editor de GADUN utilizando el botón probar tokens.

	Nombre	Expresión
1	Nombres	Juan,Carlos,Nelson,Carolina
2	Apellidos	Giménez,Guerrero,Campo,Burbano

 Probar Tokens  

Con el editor abierto ingrese la secuencia:

Juan Carlos Giménez  
Nelson Burbano  
Carolina Campo

y presione el botón para ejecutar(play).

Observe que en la sección de resultado del editor aparecerá la secuencia:

Aceptado:

{ Nombres:Juan } { Nombres:Carlos } { Apellidos:Giménez }  
{ Nombres:Nelson } { Apellidos:Burbano }  
{ Nombres:Carolina } { Apellidos:Campo }

Aquí podrá visualizar la principal tarea del analizador léxico, utilizando una serie de reglas definidas como patrones, y en el caso de GADUN tokens, reconocer dentro de una secuencia de entrada elementos equivalentes definidos dentro de estas reglas, a estos elementos se les denomina Componentes léxicos.

En este caso Juan Carloz Giménez Nelson Burbano ... Corresponde a la secuencia de entrada o “Código fuente” del compilador, el analizador reconoce en orden los elementos de la secuencia ingresada y los vincula con el token o clasificación que ha reconocido al elemento y está definida en el analizador léxico por ejemplo: Juan → {Nombres:Juan} .De esta manera la

secuencia en resultado es el recurso que entiende el compilador y con la que trabajara en el análisis sintáctico.

5. Para observar mejor cómo funciona el analizador léxico dentro del editor con la misma secuencia de entrada presione la ejecución paso a paso.



En el área de resultado observe los elementos clave del analizador léxico.

Paso a paso	Resultado
<p>Elemento en Cabecera</p> <input type="text" value="Juan"/>	<p>Cadena Resultado</p> <input type="text" value="Nombres:Juan:Juan"/>
<p>Token Reconocido</p> <input type="text" value="Nombres"/>	

En un espacio se almacena la secuencia que ha reconocido el analizador o que coincide con uno de los patrones del mismo. GADUN denomina a este espacio la “Cabecera. En este caso la secuencia que se ha reconocido es la expresión ‘Juan’.

Elemento en Cabecera
<input type="text" value="Juan"/>

En el siguiente campo, “Token Reconocido” GADUN indica el patrón que ha permitido reconocer la primera expresión. En este caso Nombres contiene la expresión ‘Juan’ así que esta es la clasificación con la que conocerá el compilador a dicha secuencia.

Token Reconocido
<input type="text" value="Nombres"/>

En la lista de Cadena resultado se muestra el lexema o sub-token que se agregara a la cadena léxica, en este caso GADUN concatena el token reconocido (Nombres:Juan) y el valor de la secuencia (Juan).

Cadena Resultado

Nombres:Juan:Juan

Además, aunque esto no es visible el analizador está cumpliendo con otra tarea que obtiene información que puede ser usada en el análisis semántico o en la generación de código que es la “construcción” de la tabla de símbolos.

En esta tabla el analizador guarda, el valor de la secuencia, su posición, su clasificación y cualquier otra información de control que se considere necesaria. GADUN en este caso guarda esta información en la misma cadena léxica, solo que esta no es visible para el usuario ya que esta es usada directamente por el analizador.

6. Cambia el orden de los elementos de la secuencia de ingreso, pon primero los apellidos o solo pon nombres o los apellidos. Como podrás observar el analizador léxico identificará los elementos sin importar el orden en que aparezcan, ya que este no está en sus funciones. Pero si ingresas un nombre o apellido que no esté definido en los tokens, por ejemplo Pablo, el analizador marcará un error.

Secuencia ingresada

1	Juan Pablo Giménez
---	--------------------

Resultado:

Error léxico en fila 1 elemento 2:  
El elemento 'P' no pertenece a los tokens definidos.

Aquí puede identificar otra función del analizador léxico, si un elemento no está definido por las reglas o patrones, esto será identificado como un error léxico, ya que este elemento no pertenece a ninguna de las clasificaciones que puede manejar el compilador. Con esta función el analizador hace una primera validación de la secuencia ingresada, indicando la sección de error y el causante y no permitirá continuar el análisis hasta que este no sea resuelto, de manera que esta no genere errores más adelante.

7. Para cambiar el error léxico podría agregar a Pablo a la lista de Nombres, pero para permitir otras secuencias alfabéticas crearemos un nuevo Token de tipo expresión regular.

En la sección de nombre pondremos Otros y la expresión regular que permite identificar cadenas con caracteres alfabéticos. Para la profundización de los tokens expresión regular revisa el Ejercicio 2.

Token de Conjunto	Token de Expresión Regular
Nombre del Token:	Otros <span>✕</span>
Expresión Regular:	[l L+ <span>✕</span>
<input type="button" value="Crear Token"/>	

De esta manera podremos identificar cualquier otro nombre o apellido.

Tome por ejemplo:

Secuencia ingresada

1	Juan Paola Luisa Natalia
2	
3	etc
4	
5	
6	kuawdukawhdkbawdkj

Resultado

Aceptado:  
 { Nombres:Juan } { Otros } { Otros } { Otros }  
 { Otros }  
 { Otros }

Como observará todas las secuencias que no son reconocidas por 'Nombres' y 'Apellidos' son denominadas 'Otros', incluso aquellas que no tienen sentido, siendo esto un factor importante al tener en cuenta la definición de los Tokens, como recomendación gadun sugiere utilizar siempre que sea posible los Tokens Conjunto para identificar expresiones específicas y finitas, mientras que los tokens de expresión regular se usan para reconocer infinitas secuencias que cumplan con un patrón.

Para finalizar en el ejemplo puede visualizar la última finalidad del analizador léxico, que es eliminar espacios en blanco o separadores que no sean relevantes para el analizador.

Si bien el análisis de Nombres y Apellidos podría no ser útil en un espacio real para la construcción de un compilador, transfiera los conocimientos adquiridos en el reconocimiento de palabras reservadas, y secuencias específicas en un lenguaje de programación, siendo el mismo fundamento solo con distintas expresiones.

**Ejercicio de refuerzo:**

1. Defina los tokens necesarios para reconocer palabras reservadas y tipos de variables.
2. Consulte cómo definir una expresión regular en regex para reconocer número, variables y letras y cree los tokens correspondientes.