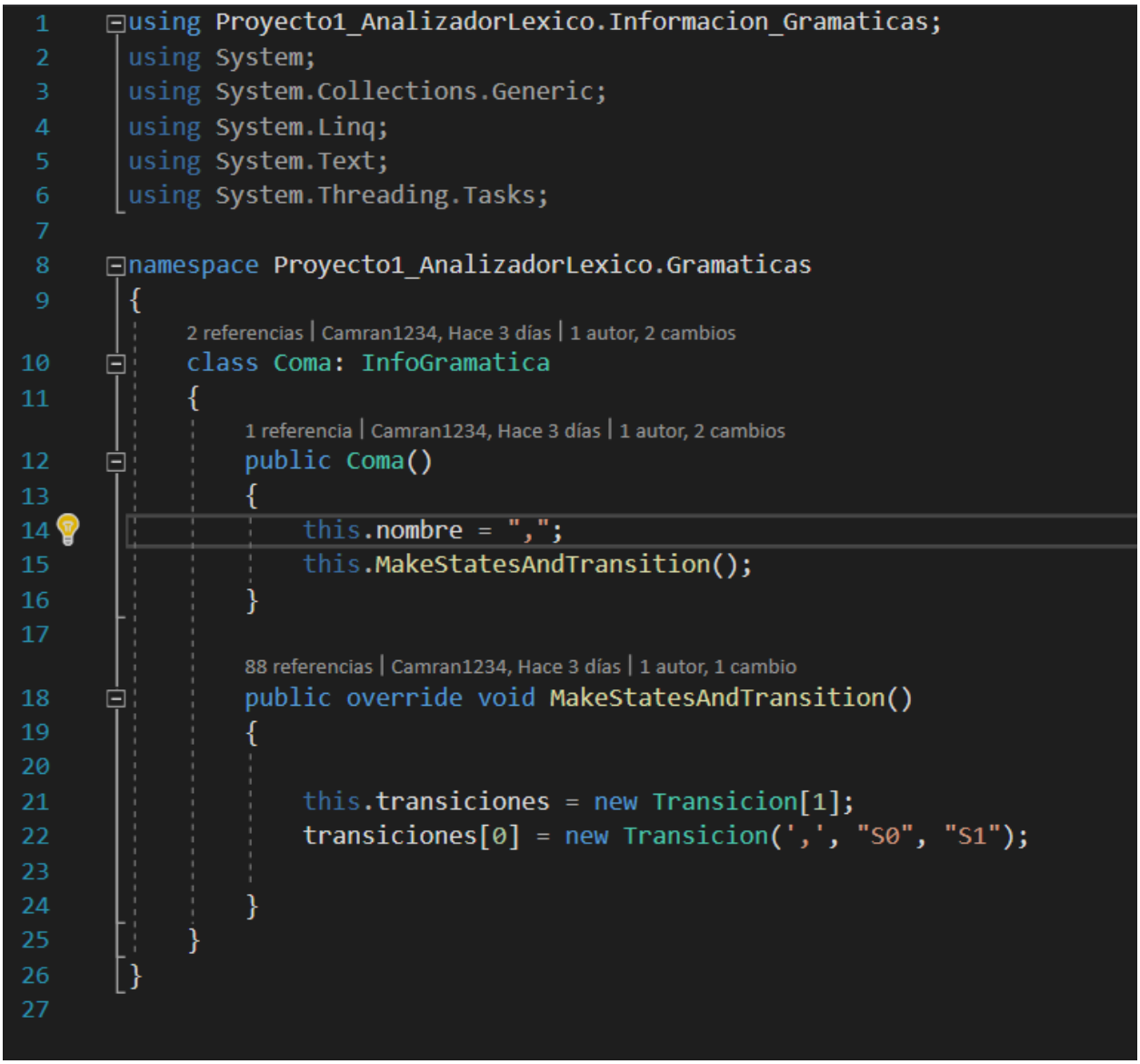
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
CENTRO UNIVERSITARIO DE OCCIDENTE

MANUAL TÉCNICO

Agregar Nuevo Elemento al Analizador Léxico

Para agregar un nuevo elemento al analizador léxico, debemos de crear una nueva clase y que esta heredé de la clase InfoGramatica, asignamos los valores de la siguiente manera:



```
using Proyecto1_AnalizadorLexico.Informacion_Gramaticas;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Proyecto1_AnalizadorLexico.Gramaticas
{
    2 referencias | Camran1234, Hace 3 días | 1 autor, 2 cambios
    class Coma: InfoGramatica
    {
        1 referencia | Camran1234, Hace 3 días | 1 autor, 2 cambios
        public Coma()
        {
            this.nombre = ",";
            this.MakeStatesAndTransition();
        }

        88 referencias | Camran1234, Hace 3 días | 1 autor, 1 cambio
        public override void MakeStatesAndTransition()
        {

            this.transiciones = new Transicion[1];
            transiciones[0] = new Transicion(',', "S0", "S1");

        }
    }
}
```

Luego agregamos la nueva clase a nuestro lenguaje.

Agregar Nuevo Elemento al Analizador Sintáctico

Si queremos agregar un nuevo elemento a nuestro analizador sintáctico, de primero tenemos que revisar la documentación del analizador sintáctico, basado en gramática LL1, por lo que tomamos la gramática y cambiamos tanto los cálculos de primeros, cálculos de siguientes y la tabla de transiciones (parte importante), por lo que es

necesario después de agregar el análisis agregarlo a nuestra clase AnalizadorSintactico, de la siguiente manera:


```
public AnalizadorSintactico()
{
    expresiones[0] = new Expresion("E");
    expresiones[0].AddTokensApunta("Palabra", "F G");
    expresiones[1] = new Expresion("F");
    expresiones[1].AddTokensApunta("Palabra", "Palabra Pa Pc Y");
    expresiones[2] = new Expresion("Y");
    expresiones[2].AddTokensApunta("{", "{");
```

Estructura del Código

La estructura del código es de la siguiente manera:

Crear Función

```
principal ( ) {

}
```

Estructura DESDE

```
DESDE _i = 0 HASTA _i < 20 INCREMENTO 1 {
_c = 5;
_a = 159.01 ;
Imprimir (”datos”);
Entero _var = 15;
}
```

Crear Variables

```
Decimal _b , _c , _d ;
Entero _a = 0;
```

Estructura Mientras

```
MIENTRAS ( _a < 5 ) {
                Imprimir ( _a + ”resultado \n” ) ;
                _a ++ ;
}
```

Incrementar Numero

```
_a ++ ;
```

Disminuir Numero

```
_a ++ ;
```

Sentencia SI

```
SI (! False ){
_a = 770;
}
SINO_SI (_c > _b && True){
_b = 50;
}
SINO_SI ( True ){
//sentencias
}
SINO {
//sentencias
}
```

Sentencia Hacer

```
HACER {
_c = 5;
_a = 159.01;
Imprimir (”datos”);
Entero _var = 15;
} MIENTRAS ( _datos == 2 )
```

Sentencia Imprimir

```
Imprimir (  _var + ”Hola Mundo”  + 4) ;
```

Sentencia Leer

```
Leer ( _var ) ;
```

Más Información:

Para verificar cómo esta programado el programa se puede ver la carpeta Clases.