

Tarea 1

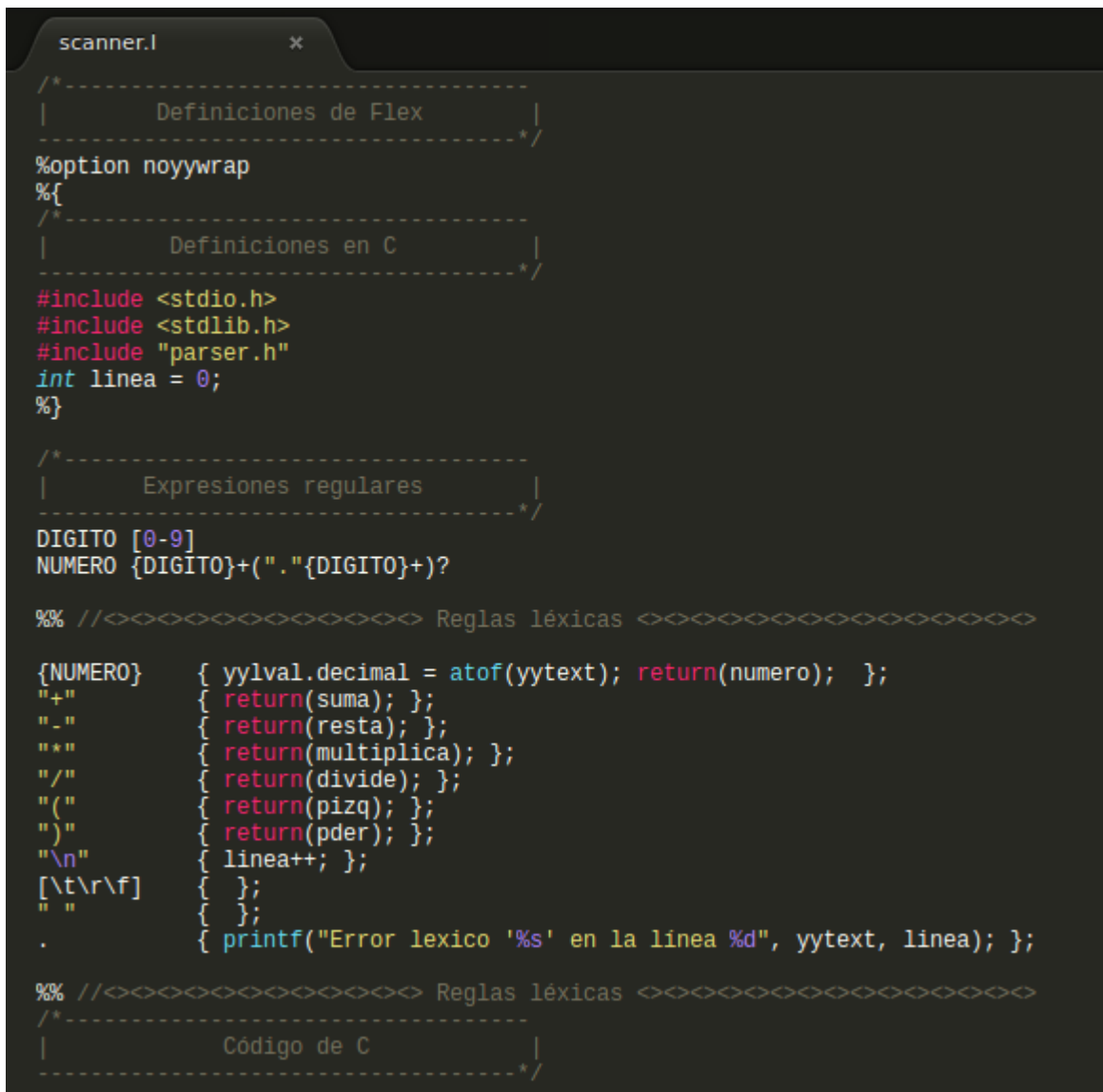
Calculadora con Bison, Flex y C++

Integración de Bison, Flex y C++ con el IDE Qt creator (bajo sistema operativo Linux) para realizar una calculadora con diversas gramáticas (una ambigua y otra no ambigua).

Instalación de Flex:

```
sudo apt-get install flex
```

Archivo para definir el analizador léxico, **scanner.l**

A screenshot of a code editor showing the content of a file named 'scanner.l'. The code is written in Flex (lexer) syntax. It includes comments for 'Definiciones de Flex' and 'Definiciones en C'. It uses '%option noyywrap' and includes standard C headers (<stdio.h>, <stdlib.h>) and 'parser.h'. It declares a variable 'linea' as an integer and initializes it to 0. It defines regular expressions for 'DIGITO' (digits 0-9) and 'NUMERO' (numbers with optional decimal points). It then lists lexical rules for various tokens: numbers, '+', '-', '*', '/', '(', ')', '\n', whitespace, and an error rule. Each rule has an associated action block. The error rule prints a message indicating a lexical error at a specific line. The code ends with a comment for 'Código de C'.

Instalación de Bison:

```
sudo apt-get install bison
```

Archivo para definir el analizador sintáctico, **parser.y**

Sección de definiciones en C. Las sentencias **QString resultado; #include <QString>** son para la integración con Qt y C++.

```
%{  
  
/*-----  
|           Declaraciones en C           |  
-----*/  
  
#include <stdio.h>  
#include <stdlib.h>  
#include <QString>  
#include <math.h>  
#include "scanner.h"  
extern int yylex(void);  
extern int linea;  
void yyerror(char *s);  
QString resultado;  
  
%}
```

Sección de declaraciones propias de Bison. Se definen los terminales y no terminales; la estructura de datos (union) que contendrá los atributos de cada componente gramatical; además de la precedencia y asociatividad para dichos componentes.

```
/*-----  
|           Declaraciones de Bison           |  
-----*/  
  
//Atributos  
%union{  
    double decimal;  
    int entero;  
    struct{  
        char cadena[255];  
        double valor;  
    } atributos;  
}  
  
//Terminales  
%token <decimal> numero  
%token suma resta multiplica divide pizq pder  
  
//No terminales  
%type <atributos> INICIO EXPRESION  
  
//Asociatividad y precedencia  
//(implicita en el orden de declaración)  
%left suma resta  
%left multiplica divide
```

Seguido de la sección de declaraciones de bison se colocan %% para indicar que se empezará con la definición de la gramática, con acciones asociadas a sus nodos. A continuación se presenta la gramática ambigua.

```

/*-----
      Gramática ambigua
-----*/
S -> E
E -> E + E
    | E - E
    | E * E
    | E / E
    | ( E )
    | num
*/

EXPRESSION
: EXPRESSION suma EXPRESSION
{
    $$valor = $1.valor + $3.valor;
    sprintf($$.cadena, "%s+%s", $1.cadena, $3.cadena);
}
| EXPRESSION resta EXPRESSION
{
    $$valor = $1.valor - $3.valor;
    sprintf($$.cadena, "%s-%s", $1.cadena, $3.cadena);
}
| EXPRESSION multiplica EXPRESSION
{
    $$valor = $1.valor * $3.valor;
    sprintf($$.cadena, "%s*s", $1.cadena, $3.cadena);
}
| EXPRESSION divide EXPRESSION
{
    $$valor = $1.valor / $3.valor;
    sprintf($$.cadena, "%s/%s", $1.cadena, $3.cadena);
}
| pizq EXPRESSION pder
{
    $$valor = $2.valor;
    sprintf($$.cadena, "(%s)", $2.cadena);
}
| numero
{
    $$valor = $1;
    sprintf($$.cadena, "%.f", $1);
};

```

Por último se declara la sección de código C propio del desarrollador.

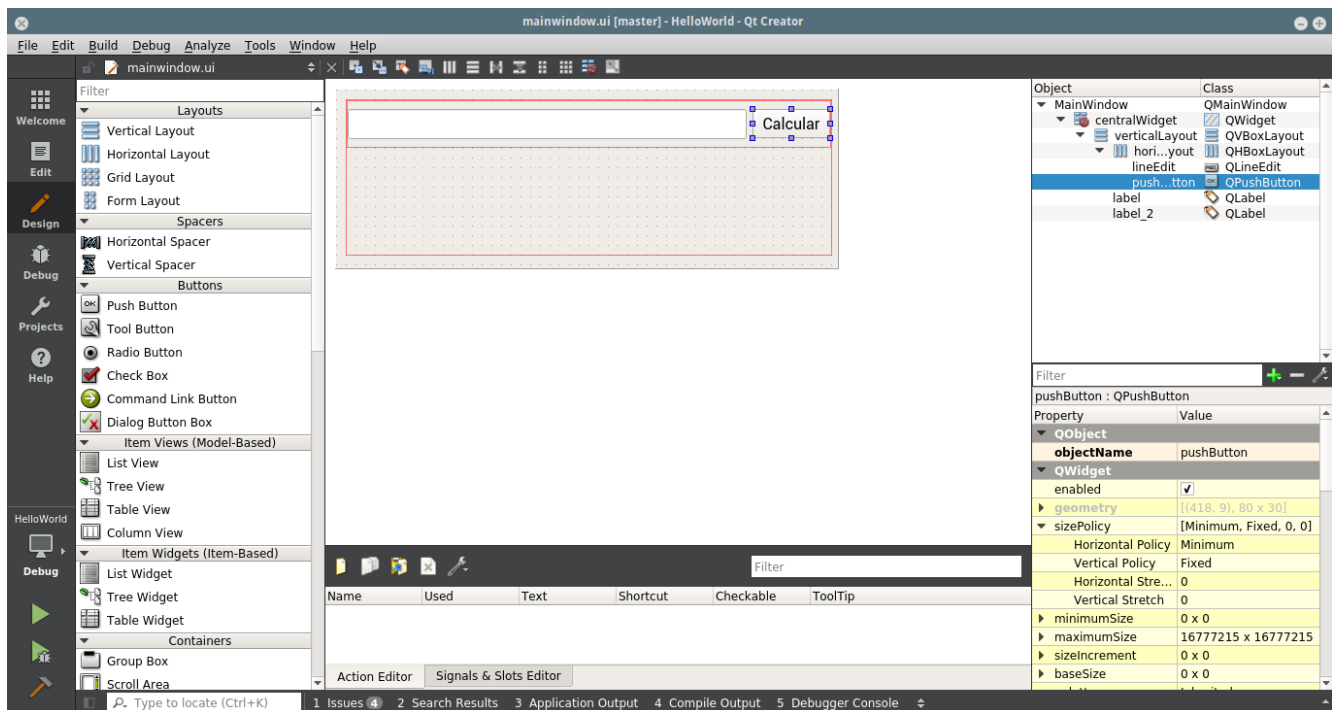
```
%//<><><><><><><><><><><><><><><><><><>
/*-----
|          Código C adicional          |
|-----*/
void yyerror(char *s)
{
    printf("Error sintactico: %s",s);
}
```

Para compilar los archivos scanner.l y parser.y se utiliza un archivo con extensión .sh con permisos de ejecución y que contenga los siguientes comandos.

```
echo "Compilando BISON..."
bison -o parser.cpp --defines=parser.h parser.y
#Se generan los archivos parser.cpp y parser.h
echo "Compilando FLEX..."
flex -o scanner.cpp --header-file=scanner.h scanner.l
#Se generan los archivos scanner.cpp y scanner.h
echo "Generación del compilador terminada..."
```

En [este enlace](#) se detalla como se instala el IDE Qt creator. Con el IDE listo para trabajar se crea una nueva Qt widgets application; dentro del formulario se crean los componentes mínimos para interactuar con el usuario.

- Un área de texto (QLineEdit) para recibir la entrada.
- Un botón (QPushButton) para realizar la compilación de la entrada.
- Un label (QLabel) para mostrar la salida del proceso de compilación.



Al botón se le debe asignar un nuevo slot; click derecho sobre el botón y se selecciona Go to slot, luego se selecciona el evento con el que se desea activar el botón (para este caso, clicked()). Esto creará un nuevo método y nos dirigirá a esa sección del código.

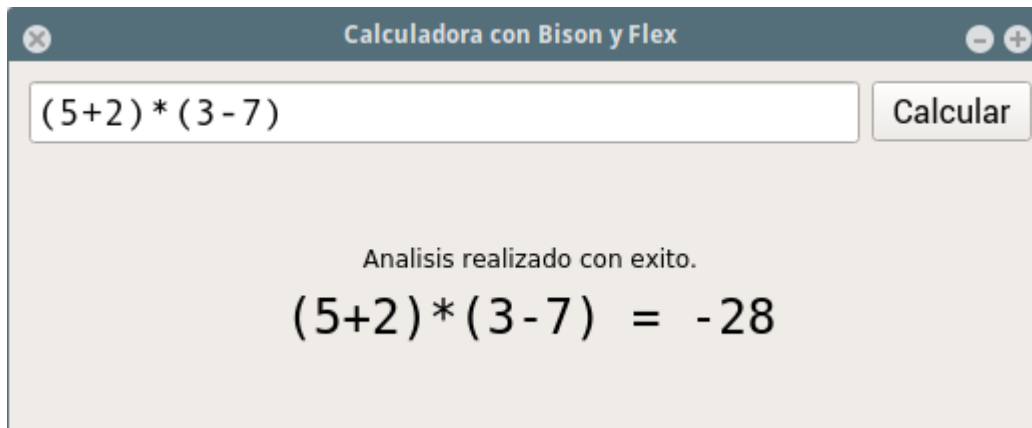
En esta nueva sección debemos agregar las siguientes líneas (de la 2 a la 7; la línea 5 es opcional).

```
< > mainwindow.cpp | X | MainWindow::MainWindow(QWidget *)
1 #include "mainwindow.h"
2 #include "ui_mainwindow.h"
3 #include "parser.h"
4 #include "scanner.h"
5 #include <QMessageBox>
6 extern int yyparse();
7 extern QString resultado;
```

Por último en el slot asignado al dar click al botón se debe colocar el siguiente código.

```
< > mainwindow.cpp | X | MainWindow::on_pushButton_clicked(): void
30
31 YY_BUFFER_STATE bufferState = yy_scan_string(str.toUtf8().constData());
32
33 if(yyparse()==0){
34     ui->label->setText(resultado);
35 }else{
36     ui->label->setText("");
37 }
38
39 yy_delete_buffer(bufferState);
40 }
```

Al final de todos los pasos, con Ctrl + R pondremos a andar nuestro programa y estará funcionando.



Referencias consultadas:

- Introducción a Bison
<http://ccia.ei.uvigo.es/docencia/PL/doc/BISON.html>
- El Blog del Inge Hobbit: Flex y Bison con Qt
<http://inghobbit.blogspot.com/2011/04/flex-y-bison-con-qt.html>
- C++ - Using Flex and Bison with Qt – Stack Overflow
<http://stackoverflow.com/questions/33622275/using-flex-and-bison-with-qt>
- Compiler construction – Multiple attributes in bison – Stack Overflow
<http://stackoverflow.com/questions/15699911/multiple-attributes-in-bison>



Código fuente disponible [aquí](#):