TEC

Compiladores e Intérpretes
Proyecto #2: Analizador Léxico
Profesor: Francisco Torres

Dennisse Rojas Casanova
Treicy Sánchez Gutiérrez

25 de Mayo, 2016

Análisis Léxico y Flex

El Análisis Léxico consiste en descomponer un fuente de entrada en categorías léxicas mínimas llamadas tokens.Un programa en Flex consiste básicamente en una lista de expresiones regulares que definen acciones a ejecutar cuando ocurre un match.

```
23 ifndef  yyHEADER_ H 1
1
23 line  6
23 line  8
ALIGNED  short   int
```

```
_ SCANNER_ MAJOR_ VERSION  2
_ SCANNER_ MINOR_ VERSION  5
_ SCANNER_ SUBMINOR_ VERSION  35
23 if   INT_ SCANNER_ SUBMINOR_ VERSION  >   0
_ BETA 23 endif
```

23 include 23 include
23 ifndef FLEXINT_ H

```
23 if  defined ( __STDC_VERSION__ ) && __STDC_
VERSION__ >= 199901L
23 ifndef __STDC_LIMIT_MACROS LIMITSTDCMACROS 1
23 endif typedef int8STDCt flexSTDCint8STDCt;
typedef uint8STDCt flexSTDCuint8STDCt;
typedef int16STDCt flexSTDCint16STDCt;
```

```
typedef  uint16STDCt  flexSTDCuint16STDCt;
typedef  int32STDCt  flexSTDCint32STDCt;
typedef  uint32STDCt  flexSTDCuint32STDCt;
23 else
typedef  signed  char  flexSTDCint8STDCt;
typedef  short  int  flexSTDCint16STDCt;
typedef  int  flexSTDCint32STDCt;
typedef  unsigned  char  flexSTDCuint8STDCt;
typedef  unsigned  short  int  flexSTDCuint16STDCt;
typedef  unsigned  int  flexSTDCuint32STDCt;
```

```
23 ifndef  INT8STDCMIN                    ( - 128 )
23 endif 23 ifndef  INT16STDCMIN                ( - 32767 - 1 )
23 endif 23 ifndef  INT32STDCMIN                  ( -
2147483647 - 1 )
```

```
23 endif 23 ifndef  INT8STDCMAX                    ( 127 )
23 endif 23 ifndef  INT16STDCMAX                  ( 32767 )
23 endif 23 ifndef  INT32STDCMAX                  (
2147483647 )
23 endif
```

```
23 ifndef  UINT8STDCMAX                    ( 255 U)
23 endif 23 ifndef  UINT16STDCMAX                  ( 65535 U)
23 endif 23 ifndef  UINT32STDCMAX                  (
4294967295 U)
23 endif
```

```
23 endif
23 endif
23 ifdef  STDCSTDCcplusplus
STDCUSESTDCCONST
23 else
```

```
23 if  defined  ( STDCSTDCSTDCSTDCSTDC)
STDCUSESTDCCONST
23 endif
23 endif
23 ifdef  INTSTDCUSESTDCCONST
```

```
23 else
23 endif
23 ifndef  INTSTDCBUFSTDCSIZE 23 ifdef
STDCSTDCia64STDCSTDC
STDCBUFSTDCSIZE  32768
23 else
```

```
STDCBUFSTDCSIZE 16384
23 endif
23 endif
23 ifndef
INTSTDCTYPEDEFSTDCINTSTDCBUFFERSTDCSTATE
STDCTYPEDEFSTDCINTSTDCBUFFERSTDCSTATE typedef
struct yySTDCbufferSTDCstate *
INTSTDCBUFFERSTDCSTATE;
23 endif
extern int yyleng;
```

```
extern    FILE   * yyin,    * yyout;
23 ifndef  INTSTDCTYPEDEFSTDCINTSTDCSIZESTDCT
STDCTYPEDEFSTDCINTSTDCSIZESTDCT typedef    sizeSTDCt
yySTDCsizeSTDCt;
23 endif
23 ifndef
INTSTDCSTRUCTSTDCINTSTDCBUFFERSTDCSTATE
STDCSTRUCTSTDCINTSTDCBUFFERSTDCSTATE
```

```
struct   yySTDCbufferSTDCstate   {
  FILE  * yySTDCinputSTDCfile;
  char   * yySTDCchSTDCbuf;
  char   * yySTDCbufSTDCpos;

  yySTDCsizeSTDCt  yySTDCbufSTDCsize;
```

```
int   yySTDCnSTDCchars;

int   yySTDCisSTDCourSTDCbuffer;

int   yySTDCisSTDCinteractive;
```

```
int   yySTDCatSTDCbol;
      int   yySTDCbsSTDClineno;
      int   yySTDCbsSTDCcolumn;


int   yySTDCfillSTDCbuffer;
int   yySTDCbufferSTDCstatus;
```

```
  } ;
23 endif
void   yyrestart  ( FILE  * inputSTDCfile    ) ;
void   yySTDCswitchSTDCtoSTDCbuffer  (
INTSTDCBUFFERSTDCSTATE  newSTDCbuffer    ) ;
INTSTDCBUFFERSTDCSTATE  yySTDCcreateSTDCbuffer  (
FILE  * file, int   size    ) ;
void   yySTDCdeleteSTDCbuffer  (
INTSTDCBUFFERSTDCSTATE  b    ) ;
void   yySTDCflushSTDCbuffer  (
INTSTDCBUFFERSTDCSTATE  b    ) ;
void   yypushSTDCbufferSTDCstate  (
INTSTDCBUFFERSTDCSTATE  newSTDCbuffer    ) ;
void   yypopSTDCbufferSTDCstate  ( void   ) ;
```

```
INTSTDCBUFFERSTDCSTATE yySTDCscanSTDCbuffer ( char
* base, yySTDCsizeSTDCt size ) ;
INTSTDCBUFFERSTDCSTATE yySTDCscanSTDCstring ( const
 char * yySTDCstr ) ;
INTSTDCBUFFERSTDCSTATE yySTDCscanSTDCbytes ( const
 char * bytes, int len ) ;
void * yyalloc ( yySTDCsizeSTDCt ) ;
void * yyrealloc ( void * , yySTDCsizeSTDCt ) ;
void yyfree ( void * ) ;
```

```
  1
STDCSKIPSTDCYYWRAP
extern    int    yylineno;
extern    char    * yytext;
23 ifdef  INTSTDCHEADERSTDCEXPORTSTDCSTARTSTDC-
CONDITIONS
```

```
23 endif
23 ifndef  INTSTDCNOSTDCUNISTDSTDCH 23 include
23 endif
23 ifndef  INTSTDCEXTRASTDCTYPE
STDCEXTRASTDCTYPE  void   *
23 endif
```

```
int    yylexSTDCdestroy  ( void    ) ;
int    yygetSTDCdebug  ( void    ) ;
void    yysetSTDCdebug  ( int    debugSTDCflag     ) ;
INTSTDCEXTRASTDCTYPE  yygetSTDCextra  ( void    ) ;
```

```
void   yysetSTDCextra  ( INTSTDCEXTRASTDCTYPE
userSTDCdefined    ) ;
FILE  * yygetSTDCin  ( void   ) ;
void   yysetSTDCin    ( FILE  *   inSTDCstr    ) ;
FILE  * yygetSTDCout  ( void   ) ;
void   yysetSTDCout    ( FILE  *   outSTDCstr    ) ;
```

```
int    yygetSTDCleng  ( void    ) ;
char    * yygetSTDCtext  ( void    ) ;
int    yygetSTDClineno  ( void    ) ;
void    yysetSTDClineno  ( int    lineSTDCnumber     ) ;
```

```
23 ifndef INTSTDCSKIPSTDCYYWRAP 23 ifdef
STDCSTDCcplusplus extern C int n ( void ) ;
23 else
extern int n ( void ) ;
23 endif 23 endif
23 ifndef yytext static void yySTDCflexSTDCstrncpy ( char
* , const char * , int ) ;
```

```
23 endif
23 ifdef  INTSTDCNEEDSTDCSTRLEN static   int
yySTDCflexSTDCstrlen  ( const   char   *   ) ;
23 endif
23 ifndef  INTSTDCNOSTDCINPUT
23 endif
```

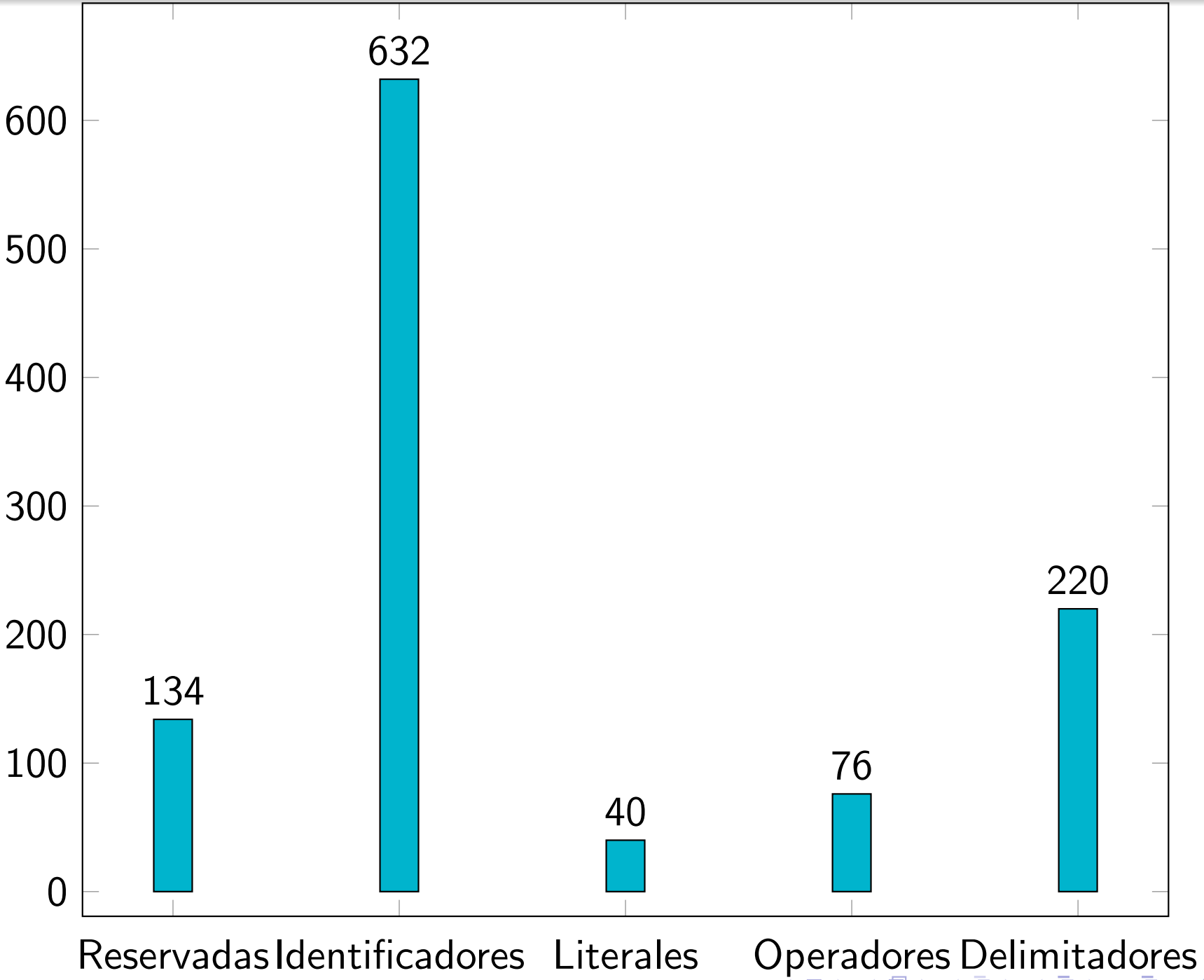
```
23 ifndef  INTSTDCREADSTDCBUFSTDCSIZE 23 ifdef
STDCSTDCia64STDCSTDC
STDCREADSTDCBUFSTDCSIZE  16384
23 else
STDCREADSTDCBUFSTDCSIZE  8192
23 endif
23 endif
```

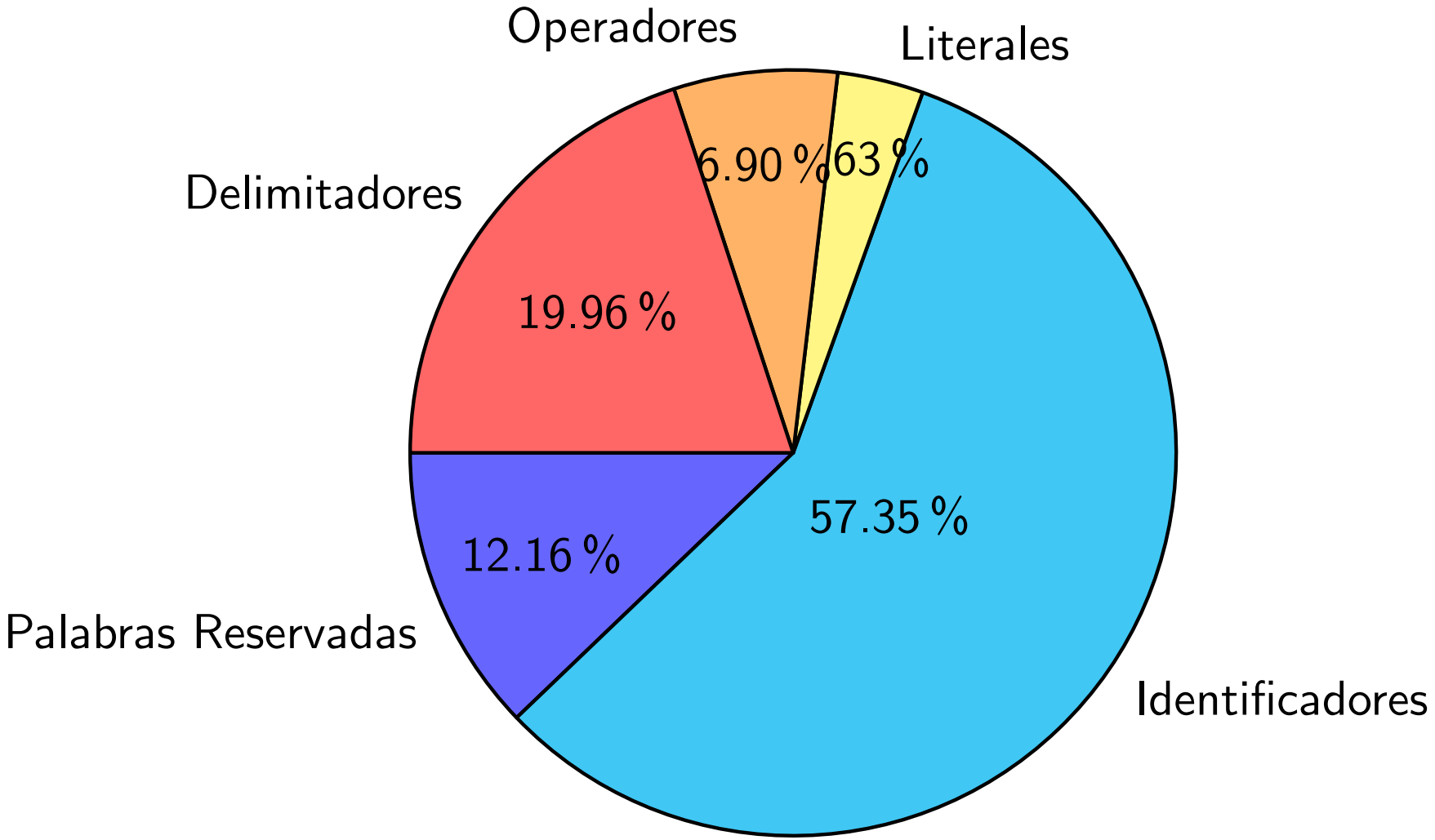
```
23 ifndef  INTSTDCSTARTSTDCSTACKSTDCINCR
STDCSTARTSTDCSTACKSTDCINCR  25
23 endif
23 ifndef  INTSTDCDECL STDCDECLSTDCISSTDCOURS  1
extern   int   yylex  ( void ) ;
```

```
STDCDECL int yylex ( void )
23 endif
23 undef INTSTDCNEWSTDCFILE 23 undef
INTSTDCFLUSHSTDCBUFFER 23 undef yySTDCsetSTDCbol
23 undef yySTDCnewSTDCbuffer
```

23 undef yySTDCsetSTDCinteractive 23 undef
INTSTDCDOSTDCBEFORESTDCACTION
23 ifdef INTSTDCDECLSTDCISSTDCOURS 23 undef
INTSTDCDECLSTDCISSTDCOURS 23 undef INTSTDCDECL 23
endif
23 line 159

23 line 334
23 undef HEADERSTDCHEADER 23 endif


600
500
400
300
200
100
0
632
134
40
76
220
Reservadas
Identificadores
Literales
Operadores
Delimitadores


Operadores
Literales
Delimitadores
6.90 %
3 %
19.96 %
12.16 %
57.35 %
Palabras Reservadas
Identificadores