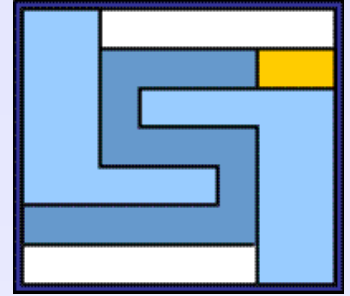




UNIVERSIDAD DE SEVILLA
E. T. S. INGENIERÍA INFORMÁTICA
LENGUAJES Y SISTEMAS INFORMÁTICOS



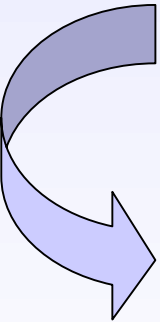
PRÁCTICAS DE LABORATORIO
JFLEX

LENGUAJES FORMALES Y AUTÓMATAS
CURSO 2005/2006

¿Qué es el análisis léxico?

El **análisis léxico** consiste en identificar en un texto aquellas cadenas que se ajustan (encajan en) a un determinado patrón lingüístico

Aunque la inauguración de la Feria Internacional del Turismo y el Ocio (FITO'04) se pensaba realizar el 25/06/2004, la falta de 250.000.000 euros ha determinado que los consejeros D. José Fernández Díaz y Dña. María Pérez Gómez pospongan dicho acontecimiento hasta el 25 de octubre en el que se espera recibir una subvención de 75 millones de euros procedentes de la UE y de 150 de las arcas del estado.



Patrón	Cadenas que encajan en el patrón
FECHA	25/06/2004 25 de octubre
SIGLAS	FITO'04 UE
CANTIDADES	250.000.000 euros 75 millones de euros 150

Las **expresiones regulares** son uno de los medios más habituales para representar patrones léxicos

¿Qué son las Expresiones Regulares (ExpReg)?

Una **expresión regular** (ExpReg) representa un conjunto de cadenas mediante una expresión en la que se mezclan símbolos y operadores

ExpReg	Cadenas representadas
"casa"	casa
a e i o u	a e i o u
(0 1)(0 1)	00 01 10 11
be+	be bee beee beeee

SÍMBOLOS BÁSICOS

Caracteres: Incluye los recogidos en la norma UNICODE. Los caracteres:

| () { } [] < > \ . * + ? ^ \$ / . " ~ !

deben ser precedidos por la barra (\) o encerrados entre comillas dobles ya que se usan también como operadores dentro de las expresiones regulares

Secuencias de escape: Donde se recogen los símbolos no imprimibles o especiales

\n fin de línea \r retorno de carro

\t tabulador \f alimentación de hoja

¿Qué es la herramienta JFLEX?

JFLEX es una herramienta JAVA que permite actuar sobre aquellas cadenas de un fichero de texto que encajan en una expresión regular

Zona de código de usuario:

Reservada para incluir las instrucciones *import*

```
/* Código incluido fuera de la clase a generar */
```

```
%%
```

Zona de opciones y declaraciones:

Las opciones comienzan con un % al inicio de línea

```
%class Lexer /* Se generara un fichero Lexer.java que
              incluire la clase Lexer */
%standalone  /* La clase Diego incluirá la función main
              Esta función recibirá como argumento el
              fichero de texto a analizar */
```

```
%%
```

Zona de reglas léxicas:

Incluyen las expreg junto con sus acciones

```
/* Cambia en un texto digo por Diego */
"digo"  { System.out.print("Diego"); }
```

Toda parte de la entrada que no encaja en el patrón es devuelta tal cual por defecto

Donde dije digo,
digo Diego

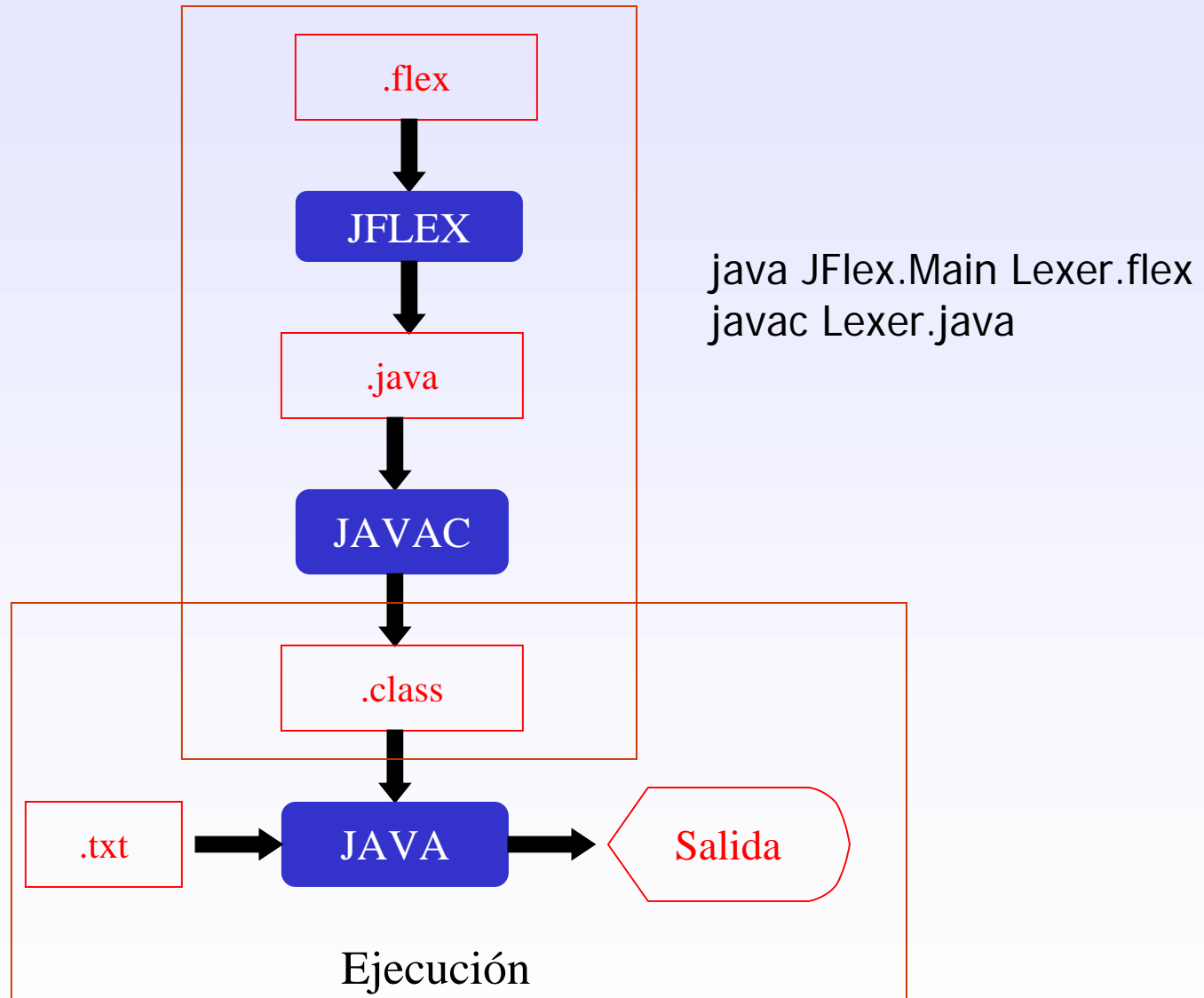
Entrada



Donde dije Diego,
Diego Diego

Salida

¿Cómo se ejecuta JFLEX?



Operadores básicos

Nombre	Patrón	Descripción
Unión	$\alpha \beta$	Cadenas que encajen en α o en β
Concatenación	$\alpha\beta$	Cadenas que encajen en α seguidas de cadenas que encajen en β
Cierre estrella	α^*	Cero o más repeticiones de cadenas que encajen en α
Cierre positivo	α^+	Una o más repeticiones de cadenas que encajen en α
Opción	$\alpha?$	Una o ninguna aparición de una cadena que encaje en α
Comodín	.	Cualquier carácter salvo el salto de línea
Paréntesis	(α)	Cadenas que encajen en α

`("+"|"-"?(0|1|2|3|4|5|6|7|8|9)+` representa un número entero

(BinarioPar) Escriba un analizador léxico que sustituya las apariciones de un número par escrito en notación binaria por la cadena "BINARIO_PAR".

(Ca_sa) Escriba un analizador léxico que sustituya las cadenas casa, camisa y carcasa que figuran en un texto por la cadena "CA_SA".

(Blancos) Escriba un analizador léxico que reduzca a un único espacio en blanco todas las secuencias de espacios en blanco y tabuladores de un texto.

(ComentarioLinea) Escriba un analizador léxico que suprima los comentarios de línea de un texto (desde un # hasta el fin de la línea).

Métodos públicos

String yytext()	devuelve la cadena que encajó en el patrón
char yycharat(int i)	devuelve el i-ésimo carácter de la cadena que encajó
int yylength()	devuelve la longitud de la cadena que encajó

```
/* Imprime el ultimo bit de cada numero binario */
%%

%class UltimoBit
%standalone

%

(0|1)+    { int ultimo=yylength()-1;
           System.out.print("[Binario "+yytext());
           System.out.print(" Ultimo Bit: "+yycharat(ultimo)+"]");
           }
```

(Asteriscos) Escriba un analizador léxico que inserte un * delante y detrás de cada carácter + que figura en un texto

(Puntos) Escriba un analizador léxico que enmarca entre corchetes el carácter que precede a cada uno de los puntos que figuran en un texto.

(DiaSemana) Escriba un analizador léxico que sustituya las apariciones de un número de la semana (de 1 a 7) por su correspondiente nombre de día.

Conjuntos

El patrón conjunto (operador []) representa un carácter de los incluidos entre los corchetes (se admiten rangos de caracteres usando el símbolo guión). Si el primer carácter del conjunto es el ángulo, entonces el patrón representa cualquier carácter no incluido en el conjunto

[aeiou] representa una de las vocales

[^A-Za-z0-9] representa un carácter no alfanumérico

(Operador) Escriba un analizador léxico que sustituya los operadores de suma +, resta -, producto * y división / y potencia ^ por la cadena "OPERADOR".

(Hexadecimal) Escriba un analizador léxico que sustituya las apariciones de un número escrito en base hexadecimal por la cadena "HEXADECIMAL".

(A_Mayusculas) Escriba en mayúsculas todas las palabras de un texto que comienzan por mayúsculas. (Considere que una palabra es una secuencia constituida por letras minúsculas o mayúsculas).

(Cadenas) Escriba un analizador léxico que sustituya todas las cadenas de un texto por el contenido de la cadena sin las comillas (Por ejemplo: "hola" sería sustituido por hola). Se define como cadena cualquier texto enmarcado entre comillas dobles con la condición de que en su contenido no figuran ni comillas dobles ni saltos de línea.

Macros

Una **macro** asocia un identificador a una expreg. Su utilidad es doble, por un lado se favorece la reutilización, y por otra se aumenta la legibilidad del analizador léxico

Las macros son declaraciones no recursivas que se incluyen en la zona de declaraciones y opciones.

Digito = [0-9]

Para expandir la macro se encierre el nombre de la macro entre llaves.

{Digito}

```
/* Muestra los identificadores de un texto */

%%

%class Lexer
%standalone

Digito      = [0-9]
Letra       = [a-zA-Z]
Identificador = {Letra}({Letra}|{Digito})*

%%
{Identificador} {System.out.print("IDENT");}
```

(Real.flex) Escriba un analizador léxico que sustituya las apariciones de número reales en notación científica por la cadena "REAL".

Se admiten como cadenas (lexemas): -3.4 .4 3.E10 .6e-2

No se admiten las siguientes: 45 4.6E .

(Horas.flex) Escriba un analizador léxico que sustituya las expresiones horarias que encajan o bien con el patrón HH:MM o bien con el patrón H:MM por la cadena "HORA". Tenga en cuenta que las horas deben ser correctas. Por ejemplo: 29:80 no sería una hora correcta.

Análisis de varios patrones

Esquema para el
análisis léxico
de varios
patrones



```
/* Identifica enteros y palabras de un texto*/  
%%  
  
%class Lexer  
%standalone  
  
Blanco = [\r\n \t\f]  
Entero = ("+" | "-")?[0-9]+  
Palabra = [a-zA-Z]+  
%%
```

Reglas léxicas para los caracteres blancos

```
{Blanco}+ {}
```

Reglas léxicas para los patrones a analizar

```
{Entero} {System.out.print("INT: "+yytext()+"\n");}  
{Palabra} {System.out.print("WORD: "+yytext()+"\n");}
```

Regla léxica cuando no encaja ningún patrón

```
. {System.out.print("OTHER: "+yytext()+"\n");}
```

(May_Min.flex) Sustituye las palabras de un texto que empiezan por minúsculas por la cadena MINUSCULA y las palabras que empiecen por mayúsculas por la cadena MAYUSCULA.

(Fechas.flex) Para representar fechas existen dos formatos: el americano (AAAA/DD/MM) y el europeo (DD/MM/AAAA). Escriba un analizador léxico que imprima todas las fechas de un texto que se ajusten a uno de dichos formatos indicando además qué formato fue usado (Ignore la problemática relacionada con los años bisiestos).

Ambigüedad: Reglas de desambiguación

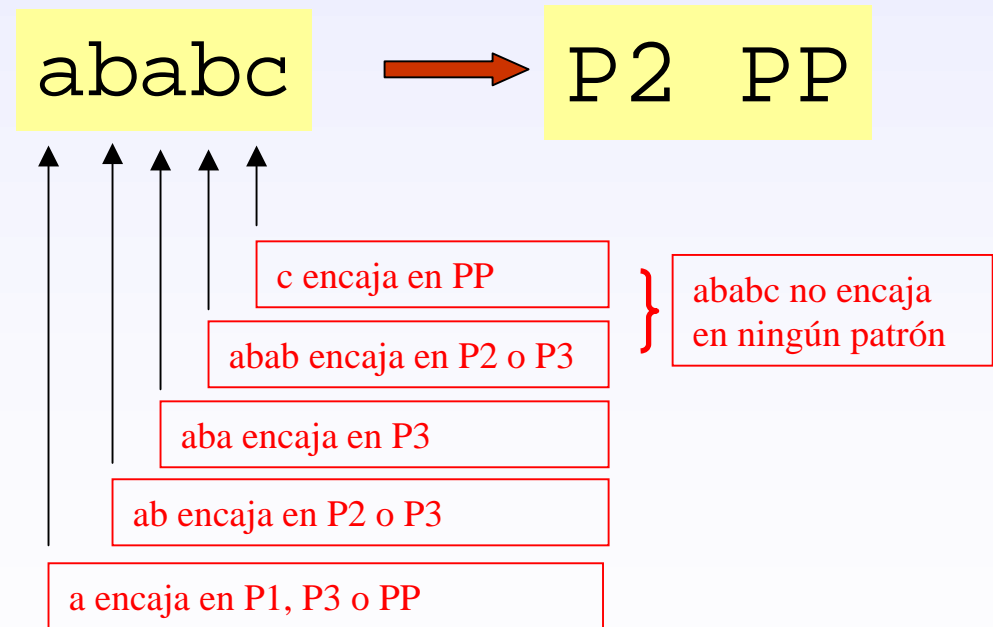
El orden de las reglas léxicas es relevante cuando los patrones son ambiguos. Dos patrones son ambiguos si existe una cadena que encaja en ambos.

Regla de desambiguación:

1. El analizador lee la entrada hasta encontrar la cadena más larga que encaje con algún patrón.
2. En caso de empate se escoge la regla léxica escrita en primer lugar

```
/* Patrones ambiguos */
%%
%class Lexer
%standalone
Blanco = [\r\n \t\f]

%%
{Blanco}+ {}
a|b      {System.out.print("P1 ");}
(ab)*    {System.out.print("P2 ");}
(a|b)*   {System.out.print("P3 ");}
.        {System.out.print("PP ");}
```



¿Cuál es la salida del analizador para las entradas "a", "aaaa", "abab" y "ababb"?

Ambigüedad: Ejemplo de prioridad según el orden

```
/* Identifica los número en base 3 y 2.
   Prioridad a la base 3
*/
%%

%class Lexer
%standalone
Blanco = [\r\n \t\f]

%%
{Blanco}+ {;}
[012]+ {System.out.print("BASE 3\n");}
[01]+ {System.out.print("BASE 2\n");}
. {}
```

es de BASE 3

0001111

es de BASE 2

```
/* Identifica los número en base 2 y 3.
   Prioridad a la base 2
*/
%%

%class Lexer
%standalone
Blanco = [\r\n \t\f]

%%
{Blanco}+ {;}
[01]+ {System.out.print("BASE 2\n");}
[012]+ {System.out.print("BASE 3\n");}
. {}
```

(If_then_else.flex) Escriba un analizador léxico que sustituya cada identificador de un texto por la cadena IDENT salvo que el identificador sea una de las palabras reservadas: If, Then o Else. En este último caso deberá imprimirse respectivamente las cadenas IF, THEN y ELSE. Las anteriores palabras reservadas no son sensibles a la capitalización, es decir, podrán aparecer mezcladas arbitrariamente letras minúsculas y mayúsculas. Por ejemplo, para la palabra reservada Else se admiten también las formas ELSE o eLSE.

(Numeros.flex) Escriba un analizador léxico que distinga entre los siguientes tipos de números: entero (INT) entero largo (LONG) y real (FLOAT). Los enteros largos son enteros terminados con una letra l mayúscula o minúscula. Los reales deben incluir de forma obligatoria parte entera y parte decimal separadas por un punto.