

# Práctica 2

## Manejo Avanzado de *Flex*

Elvira Mayordomo y Jorge Álvarez

### Tareas

1. Estudia la sección sobre las *condiciones de arranque* en *Intro\_Flex\_Bison.pdf* (páginas 7 a 9) y luego el capítulo 9 del manual de *Flex* (*flex-es-2.5.pdf*).
2. Lee la introducción de esta práctica y realiza los ejercicios propuestos.
3. Elabora la memoria de la práctica y entrégala junto con el fichero fuente según el Procedimiento de Entrega de Prácticas explicado en la Introducción a las Prácticas de la Asignatura. La fecha tope de entrega será hasta el día anterior al comienzo de la Práctica 3.

**Nota: El incumplimiento de las normas de entrega se reflejará en la calificación de la práctica.**

Se recuerda especialmente lo siguiente:

- Debes crear un directorio que contenga **exclusivamente** el fichero con la memoria en formato *PDF*, el fichero fuente con tu código (*.l* de *Flex*) y aquellos de prueba que hayas generado. No utilices subdirectorios.
- En caso de que el fichero *.zip* resultante tenga un tamaño mayor de 512 KB deberás repetir la creación del directorio dividiéndolo en varios ficheros *.zip* de como máximo 512KB cada uno, en ese caso llama a los ficheros resultantes *nipPrX1.zip*, *nipPrX2.zip*, etc.

### Introducción

El objetivo principal de esta práctica es aprender a desarrollar analizadores léxicos en *Flex* más sofisticados, profundizando en el manejo de lo que se conoce como *condiciones de arranque*. Las *condiciones de arranque* no son imprescindibles, ya que siempre se pueden emular utilizando código *C* en las acciones de los patrones. No obstante, su

---

uso facilita mucho el desarrollo de los programas en *Flex*, ya que ayudan a estructurar conjuntos de patrones/acciones en función de un contexto determinado o condiciones previas.

## Ejercicio 1

Implementa un programa en *Flex* que tome como entrada un fichero fuente en *Java* y devuelva la siguiente información:

- Número de líneas y caracteres (no blancos) del programa fuente.
- Número de comentarios (*/\* ... \*/*, */\*\* ... \*/*, *// ... <EOL>*) y longitud media expresada en forma de número de caracteres.
- Número de instrucciones *import*.
- Número de instrucciones de comparación del programa (operador *==*).
- Número de instrucciones de composición condicional (*if*, *else*, *switch*) y de composición iterativa (*while*, *for*, *do*, *while*).

**Ayuda:** En este ejercicio, las condiciones de arranque pueden ser útiles para, por ejemplo, no contabilizar palabras reservadas que estén dentro de comentarios o de constantes de tipo cadena que se utilicen en el programa. En este sentido, se podría definir una condición de arranque específica dentro de un comentario (de tipo estándar o especiales para el generador automático de documentación *Javadoc*) y otra distinta para las cadenas. Así, deberíamos poder contar las palabras clave sólo cuando estas condiciones de arranque no estén activas.

Nota: Podéis descargar un fichero de prueba en la página web de la asignatura.

## Ejercicio 2

El código genético convierte cada cadena de ADN (representada como una cadena sobre el alfabeto de 4 letras o bases  $\{A, C, G, T\}$ ) en una proteína (representada como una cadena sobre el alfabeto de 21 caracteres o aminoácidos

$$\{A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y, *\}.$$

Implementa un programa en *Flex* que traduzca el código genético, es decir, que convierta cadenas de ADN en proteínas. Debe tomar como entrada un fichero con formato FASTA que contiene el ADN y devolver la proteína resultado de hacer corresponder con cada 3 bases (codón) un aminoácido en el mismo formato. El formato FASTA consiste en representar las cadenas de ADN o proteínas de forma secuencial, empezando siempre cada

---

secuencia en una nueva línea, con el carácter “>” seguido del identificador de cada secuencia (que se asume único en el fichero) incluyendo, en algunos casos, una descripción tras el identificador (separada por un espacio en blanco) hasta el final de línea. En las siguientes líneas se encuentra la secuencia correspondiente de ADN que se deberá traducir o la secuencia de la proteína resultante (ver [http://es.wikipedia.org/wiki/Formato\\_FASTA](http://es.wikipedia.org/wiki/Formato_FASTA) para más información). En esta práctica se asumirá que cada línea de la secuencia (excluyendo la línea del identificador) tiene como máximo 60 caracteres, teniendo que introducir un salto de línea cada vez que se alcance esa cifra.

La traducción se debe hacer siguiendo la tabla encontrada en <http://www.cs.au.dk/~chili/CSS/Exercises/codontable.jpg>, donde las primeras tres letras corresponden al codón, la segunda columna contiene el nombre reducido del aminoácido y en la última columna se encuentra el símbolo por el que se debe substituir cada codón. (También puedes encontrar la tabla anterior en texto en [http://en.wikipedia.org/wiki/DNA\\_codon\\_table](http://en.wikipedia.org/wiki/DNA_codon_table)). Un ejemplo se muestra a continuación:

Secuencias ADN:	Secuencias proteínas:
>Seq1	>Seq1
ACCTGT	TC
>Seq2 Otra	>Seq2 Otra
GCACTATTT	ALF
>Seq3 Secuencia larga	>Seq3 Secuencia larga
GGATGTTGCT	GCC*
AA	