

**Grupo Jueves 15:00-17:00 semanas A**  
**– Práctica 2 –**  
**Autor: Patricia Briones Yus**

## Ejercicio 1:

### 1. Resumen

En el ejercicio 1.1 he decidido crear cuatro condiciones de arranque (inclusivas) para contar la repetición de cada letra de los nucleótidos. Cada vez que detecte una de las letras que forman el nucleótido, activará la condición de arranque correspondiente a esta letra, y si hay más de una las contará y sustituirá por el número de repeticiones. Y por último he declarado la regla ADN para reconocer las secuencias de este.

En el ejercicio 1.2 he creado en el apartado de reglas: una condición de arranque de tipo inclusiva llamada REPETIR que se activa cada vez que detecta una letra del nucleótido, una variable de tipo char (letra) para guardar la letra detectada y la regla ADN como en el apartado anterior. Si está activada la condición de arranque REPETIR, reconocerá el número que sigue a la letra y la escribirá tantas veces como indique el número.

## Ejercicio 2:

### 1. Resumen

He declarado una variable de tipo int para cada uno de los apartados que se piden, y dos condiciones de arranque de tipo inclusiva, COMEN, que detecta cuando ha empezado un comentario, y FUNC, que reconoce cuando se ha empezado una función.

## Ejercicio 3:

### 1. Resumen

Primero he creado el autómata de la imagen, he sacado su expresión regular y la he pasado al lenguaje de flex. En el apartado de reglas he creado una regla para cada caso, y que cada vez que se reconozca alguno de ellos, se imprima por pantalla. El resto de las palabras que no cumplan las reglas se ignorarán.

Expresión regular:  $(o = \text{todo carácter distinto de salto de línea, ' '})$

$casa+cas(\epsilon+o)+casao+(ca(\epsilon+o)+caso+(c(\epsilon+o)+cao+(\epsilon+o+co+oc)a)s)a+(ca(\epsilon+o)+caso+(c(\epsilon+o)+cao+(\epsilon+o+co+oc)a)s)(\epsilon+o)+(cas(\epsilon+o)+casao+(ca(\epsilon+o)+caso+(c(\epsilon+o)+cao+(\epsilon+o+co+oc)a)s)a)o+((c(\epsilon+o)+cao+(\epsilon+o+co+oc)a)(\epsilon+o)+(ca(\epsilon+o)+caso+(c(\epsilon+o)+cao+(\epsilon+o+co+oc)a)s)o+((\epsilon+o+co+oc)(\epsilon+o)+(c(\epsilon+o)+cao+(\epsilon+o+co+oc)a)o+(o(\epsilon+o)+(\epsilon+o+co+oc)o+ooc)a)s)a.$

