

# Teoría de la Computación

## Grado en Ingeniería Informática

- Prácticas de Laboratorio - \*

*Elvira Mayordomo Cámara*

email: elvira@unizar.es

*José Manuel Colom Piazuolo*

email: jm@unizar.es

Dpto. de Informática e Ingeniería de Sistemas

Escuela de Ingeniería y Arquitectura

Universidad de Zaragoza

<http://webdiis.unizar.es/assignaturas/TC/>

Curso 2014-2015

\*Material elaborado a partir de los guiones de prácticas mantenidos por los profesores Gregorio de Miguel Casado (Teoría de la Computación curso 2011-2012), Pedro Álvarez, Rubén Béjar y Jorge Júlvez para la asignatura *Lenguajes Gramáticas y Autómatas* de la titulación *Ingeniería de Informática* (122) del plan de estudios BOE 1-2-1995.



# Introducción a las Prácticas de la Asignatura

## *Entorno de Trabajo y Entrega de Prácticas*

Las prácticas de la asignatura Teoría de la Computación abordan aspectos de implementación de reconocedores para Expresiones Regulares (análisis léxico) y reconocedores de lenguajes caracterizados con gramáticas (análisis sintáctico) mediante las herramientas Unix *Flex* y *Bison*, respectivamente. El aprendizaje de estas herramientas es de interés general, ya que permite abordar, con posterioridad, la construcción de compiladores, programas para la traducción/migración entre formatos de ficheros y similares.

### Entorno de Trabajo

Las prácticas de Teoría de la Computación se realizan en Hendrix, cuya dirección es [hendrix-ssh.cps.unizar.es](http://hendrix-ssh.cps.unizar.es). Podeis abrir sesión en cualquier terminal de Linux y utilizar el editor *nedit* para editar vuestros ficheros fuente. Para compilar los programas, hay que compilar los programas usando los ejecutables correspondientes a *Flex* y *Bison*. En las siguientes URLs podeis encontrar todo lo relacionado a estas herramientas.

<http://flex.sourceforge.net/>

<http://www.gnu.org/software/bison/>

### Entrega de Prácticas

Las prácticas se realizarán de forma **individual**.

Para cada una de las cuatro prácticas se deberá entregar un paquete *.zip* que contenga una memoria en formato *PDF* y los ficheros fuente y de prueba para cada ejercicio planteado. Los siguientes apartados detallan los contenidos de la memoria, el procedimiento para empaquetar en un fichero *.zip* los ficheros para la entrega y, finalmente el mecanismo de entrega mediante el comando *someter* en *Hendrix*.

---

**El incumplimiento de las normas establecidas en este apartado para el formato de la memoria y/o ficheros se reflejará en la calificación de la práctica.**

**Las copias o plagios que se detecten en las memorias y/o programas supondrán un suspenso directo de la parte práctica de la asignatura.**

## Formato de la Memoria

- **Portada:** Número de Práctica, Grupo y Autor. Ejemplo:

<p><b>Grupo COM1M</b> – <b>Práctica 1</b> – <b>Autor:</b> Al Anturing</p>
---

- **Una sección para cada ejercicio resuelto.** Razona todas las decisiones de implementación que has tomado en la elaboración de tu código e incluye las pruebas de ejecución realizadas. Ejemplo:

<p><b>Ejercicio 1:</b></p>
----------------------------

<p><b>1. Resumen</b></p>
--------------------------

<p>He creado el patrón X para poder reconocer Y</p>
---

<p>...</p>
------------

<p><b>2. Pruebas</b></p>
--------------------------

<p>Para la entrada Z he obtenido la salida W</p>
--

<p>...</p>
------------

**Nota:** el formato del fichero de la memoria deberá ser *PDF*.

## Empaquetado de los Ficheros

- Accede a tu cuenta en *Hendrix* con un terminal de *Linux*:

<pre>ssh -X usuario@hendrix-ssh.cps.unizar.es</pre>
---

- 
- Verifica que todos tus ficheros fuente (*.l* de *Flex* y *.y* de *Bison*) contienen en sus primeras líneas número de práctica y ejercicio así como el NIP y nombre del autor. Todos los programas deberán estar debidamente documentados.
  - Crea un directorio que contenga exclusivamente el fichero con la memoria en formato *PDF*, los ficheros fuente con tu código (*.l* de *Flex* y *.y* de *Bison*) y los de prueba (*.txt* de texto). No usar subdirectorios.
  - Accede al directorio con tus ficheros y ejecuta el comando

```
zip nipPrX.zip *.*
```

donde *nip* es el identificador personal y *X* es el número de práctica (1,2,3 ó 4).

- En caso de que el fichero resultante tenga un tamaño mayor de 512 KB deberás repetir la creación del directorio dividiéndolo en varios ficheros *.zip* de como máximo 512 KB cada uno, en ese caso llama a los ficheros resultantes *nipPrX1.zip*, *nipPrX2.zip*, etc.

## Entrega con *someter* en *Hendrix*

- Una vez que ya tengas el fichero *.zip*, en tu cuenta de *Hendrix* ejecuta el comando:

```
someter -v tc_14 nipPrX.zip
```

- La fecha tope de entrega para cada una de las prácticas será hasta el día anterior a la sesión en la que comience la siguiente práctica. Para la Práctica 4 se concretará una fecha específica.



# Práctica 1A

## Introducción al manejo de *Flex*

### Tareas

1. Aprende a acceder a la página web de la asignatura, a los entornos BlackBoard Learn y Moodle y a trabajar con tu cuenta en Hendrix.
2. Descarga el documento `Intro_Flex_Bison.pdf` elaborado por el profesor Rubén Béjar Hernández y lee detenidamente el capítulo '*Introducción a Flex*'.
3. Lee la introducción de esta práctica y realiza los ejercicios 1 a 4 propuestos.
4. Elabora la memoria de la práctica y entrégala junto con los ficheros fuente según el Procedimiento de Entrega de Prácticas explicado en la Introducción a las Prácticas de la Asignatura (página 3 de este documento). La fecha tope de entrega será hasta el día anterior al comienzo de la Práctica 1B.

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

### Introducción

El objetivo principal de esta práctica de la asignatura es familiarizarse con la herramienta de creación de analizadores léxicos *Flex*. Para ello, se propone la creación con dicha herramienta de una serie de pequeños procesadores de texto.

Las prácticas de Teoría de la Computación se realizan en **hendrix-ssh.cps.unizar.es**. Para compilar los programas, hay que compilar los programas usando los ejecutables correspondientes a *Flex* y *Bison*. Ejemplo (*Flex*):

```
flex nombre_fichero_fuente.l
gcc lex.yy.c -lfl -o nombre_ejecutable
./nombre_ejecutable <fichentrada >fichsalida
```

El alfabeto  $\Sigma$  que maneja *Flex* está formado por los caracteres manejables por el sistema (p. ej. todos los símbolos del código ASCII). En las prácticas se trabaja con letras (distinguiendo entre mayúsculas y minúsculas), números y signos de puntuación.

---

## Ejercicio 1

Escribe un programa con *Flex* de nombre *ej1.l* que elimine todas las apariciones de la cadena *unizar*

## Ejercicio 2

Escribe un programa con *Flex* de nombre *ej2.l* que sustituya dos saltos de línea seguidos por uno solo y que sustituya cada tabulador por un espacio en blanco. Ejemplo:

**Entrada:**

```
Este es un texto
```

```
<EOL><EOL><EOL><EOL>
```

```
con 2 saltos de linea <EOL>
```

```
<EOF>
```

**Salida:**

```
Este es un texto <EOL><EOL>
```

```
con 2 saltos de linea <EOL>
```

```
<EOF>
```

## Ejercicio 3

Elabora un programa en *Flex* de nombre *ej3.l* que cuente el número de veces que aparece la cadena *TC*

## Ejercicio 4

Construye un programa en *Flex* de nombre *ej4.l* que cambie cada letra mayúscula por la correspondiente minúscula.