



# PROCESADORES DE LENGUAJE

Ingeniería Informática  
Especialidad de Computación  
Tercer curso  
Segundo cuatrimestre



Departamento de Informática y Análisis Numérico  
Escuela Politécnica Superior de Córdoba  
Universidad de Córdoba

Curso académico 2019 - 2020

---

## Hoja de ejercicios de FLEX

### 1. Convertir números enteros en reales

- Codifica un programa en flex que
  - reciba el nombre de un fichero que puede contener números enteros
  - y genere otro fichero en el que todos los números enteros se han convertido en reales

- Ejemplo:

***./convertir.exe entrada.txt salida.txt***

#### ***entrada.txt***

*El precio es de 100 euros sin iva y de 121 con iva.*

*El espejo costó 37.5 euros*

*La temperatura era de 20 grados.*

#### ***salida.txt***

*El precio es de 100.0 euros sin iva y de 121.0 con iva.*

*El espejo costó 37.5 euros*

*La temperatura era de 20.0 grados*

- Observación
  - Se recomienda usar la función `atoi()` de C que transforma una cadena de caracteres en su valor entero.

### 2. Contar palabras

- Codifica un analizador léxico que permita:
  - Contar el número de palabras de un fichero
    - Ejemplo  
*contar\_palabras.exe entrada.txt*  
*entrada.txt contiene 20 palabras*

- Contar el número de veces que aparece una palabra
  - Ejemplo  
*contar\_palabras.exe entrada.txt de*  
*La palabra <de> aparece 3 veces en el fichero entrada.txt*
  
  - contar\_palabras.exe entrada.txt costó*  
*La palabra <costó> aparece 1 vez en el fichero entrada.txt*

- **Observación**

- Se considerarán solamente las palabras compuestas por letras, mayúsculas o minúsculas, incluyendo las letras “ñ” y “Ñ” y las vocales acentuadas: á, é, í, ó, ú, Á, É, Í, Ó, Ú.

### 3. Analizador léxico de pseudocódigo

- Codifica un analizador léxico que permita reconocer los componentes léxicos de un programa escrito en pseudocódigo.
- **Palabras reservadas**
  - *inicio, fin, leer, escribir, si, entonces, si\_no, fin\_si, mientras, hacer, fin\_mientras, repetir, hasta\_que, para, desde, hasta, paso, fin\_para*
  - No se distinguirá entre mayúsculas ni minúsculas.
  - Las palabras reservadas no se podrán utilizar como identificadores.
- **Identificador**
  - Características
    - Estarán compuestos por una serie de letras, dígitos y el subrayado;
    - Deben comenzar por una letra,
    - No podrán acabar con el símbolo de subrayado, ni tener dos subrayados consecutivos.
    - No se distinguirá entre mayúsculas ni minúsculas.
  - Ejemplos
    - Identificadores válidos:  
*dato, dato\_1, dato\_1\_a*
    - Identificadores **no** válidos:  
*\_dato, dato\_, dato\_\_1*
- **Número**
  - Se utilizarán números enteros, reales de punto fijo y reales con notación científica.
  - Todos ellos serán tratados conjuntamente como números.

- **Cadena**
  - Estará compuesta por una serie de caracteres delimitados por comillas simples:  
*‘Ejemplo de cadena’*
  - Deberá permitir la inclusión de la comilla simple utilizando la barra (\):  
*‘Ejemplo de cadena con \' comillas\' simples’.*
  - **Nota:**
    - Las comillas exteriores no formarán parte de la cadena.
- **Operador de asignación**
  - ASIGNACIÓN: :=
- **Operadores aritméticos:**
  - SUMA: +
  - RESTA: -
  - PRODUCTO: \*
  - DIVISIÓN: /
  - MÓDULO: \_\_mod
  - POTENCIA: \*\*
- **Operador alfanumérico:**
  - CONCATENACIÓN: ||
- **Operadores relacionales de números y cadenas:**
  - MENOR\_QUE: <
  - MENOR\_IGUAL\_QUE: <=
  - MAYOR\_QUE: >
  - MAYOR\_IGUAL\_QUE: >=
  - IGUAL: ==
  - DISTINTO: <>
  - Por ejemplo:
    - Si *A* es una variable numérica y *control* una variable alfanumérica, se pueden generar las siguientes expresiones relacionales:  
*(A >= 0)*  
*(control <> ‘stop’)*
- **Operadores lógicos:**
  - DISYUNCIÓN\_LÓGICA: .o.
  - CONJUNCIÓN\_LÓGICA: .y.
  - NEGACIÓN\_LÓGICA: .no.
  - Por ejemplo:  
*(A >= 0) .y. .no. (control <> ‘stop’)*

- **Comentarios**

- De varias líneas: delimitados por (\* y \*)

*(\* ejemplo maravilloso  
de comentario  
de tres líneas \*)*

- De una línea:
  - Todo lo que siga al carácter # hasta el final de la línea.

*# ejemplo espectacular de comentario de una línea*

- **Otro componentes léxicos**

- FIN\_SENTENCIA: ;
- Paréntesis
  - Izquierdo: (
  - Derecho: )

- **Control de errores**

- El intérprete deberá controlar toda clase de errores:
  - Identificador mal escrito.
  - Números mal escritos.
  - Utilización de símbolos no permitidos.
  - Etc.

- **Prueba**

- Se deberá comprobar el funcionamiento del analizador léxico usando tres ficheros:
  - Fichero denominado Newton.txt
  - ejemplo\_1.txt: fichero original **sin** errores.
  - ejemplo\_2.txt: fichero original **con** errores.

#### 4. Código IBAN

- Considérese un fichero que contiene el código IBAN de números de cuentas corrientes

Código de país				Entidad Bancaria				Oficina				D.C.		Número de cuenta									
E	S	X	Y	C <sub>1</sub>	C <sub>2</sub>	C <sub>3</sub>	C <sub>4</sub>	C <sub>5</sub>	C <sub>6</sub>	C <sub>7</sub>	C <sub>8</sub>	D <sub>1</sub>	D <sub>2</sub>	N <sub>1</sub>	N <sub>2</sub>	N <sub>3</sub>	N <sub>4</sub>	N <sub>5</sub>	N <sub>6</sub>	N <sub>7</sub>	N <sub>8</sub>	N <sub>9</sub>	N <sub>10</sub>

- Por ejemplo:  
cuentas.txt

ES55 2400 2000 87 1234567899  
 ES69 2400 2000 86 1234567890  
 ES68 2400 2001 20 1234567891  
 ES22 2400 2001 25 1234567892

ES68 1234 1500 21 1234567893

IT20 2034 3050 67 1234567894

UK20 3212 1700 78 1234567895

Son números ficticios y, seguramente, incorrectos.

- Codifica un analizador léxico que permita:
  - Contar el número de cuentas corrientes del fichero *cuentas.exe fichero.txt*  
*Número total de cuentas: 7*
  - Contar el número de cuentas corrientes de un país  
*cuentas.exe fichero.txt ES*  
*Número total de cuentas de ES: 5*
  - Contar el número de cuentas corrientes de una entidad bancaria de un país  
*cuentas.exe fichero.txt ES 2400*  
*Número total de cuentas de ES: 4*
  - Contar el número de cuentas corrientes de una oficina una entidad bancaria de un país  
*cuentas.exe fichero.txt ES 2400 2000*  
*Número total de cuentas de ES: 2*
  - Opcional:
    - Filtra las cuentas corrientes de España que son correctas  
*cuentas\_correctas.exe cuentas.txt correctas.txt*
- Observaciones
  - Se recomienda el uso de “Estados de Flex”.
  - Véase el **Anexo** para comprobar si es correcto un código IBAN.

## ANEXO. Cálculo del código IBAN

- **Formato**

- Código del país: 4 caracteres
  - 2 dígitos, por ejemplo ES
  - 2 dígitos de control
- Código del banco al que pertenece la cuenta: 4 dígitos
- Código de la sucursal u oficina: 4 dígitos
- Dos dígitos de control
- Número de cuenta: 10 dígitos

Código de país				Entidad Bancaria				Oficina				D.C.		Número de cuenta									
E	S	X	Y	C <sub>1</sub>	C <sub>2</sub>	C <sub>3</sub>	C <sub>4</sub>	C <sub>5</sub>	C <sub>6</sub>	C <sub>7</sub>	C <sub>8</sub>	D <sub>1</sub>	D <sub>2</sub>	N <sub>1</sub>	N <sub>2</sub>	N <sub>3</sub>	N <sub>4</sub>	N <sub>5</sub>	N <sub>6</sub>	N <sub>7</sub>	N <sub>8</sub>	N <sub>9</sub>	N <sub>10</sub>

- Ejemplo

Calcular\_IBAN 2010 0001 1234567899  
ES15 2100 0001 07 1234567899

- **Primer dígito de control: D<sub>1</sub>**

- Procesamiento de las cifras de la entidad bancaria
  - Sean C<sub>1</sub> C<sub>2</sub> C<sub>3</sub> C<sub>4</sub> las cuatro cifras de la entidad bancaria.
  - Se calcula
    - ✓  $A = 4 C_1 + 8 C_2 + 5 C_3 + 10 C_4$
- Procesamiento de las cifras de la oficina
  - Sean C<sub>5</sub> C<sub>6</sub> C<sub>7</sub> C<sub>8</sub> las cuatro cifras de la oficina
  - Se calcula
    - ✓  $B = 9 C_5 + 7 C_6 + 3 C_7 + 6 C_8$
- Cálculo del resto de la suma de A y B módulo 11
  - $C = (A + B) \text{ módulo } 11$
- Primer dígito de control D<sub>1</sub>:
  - Si  $11 - C < 10$  entonces  $D_1 = 11 - C$
  - Si  $11 - C = 10$  entonces  $D_1 = 1$
  - Si  $11 - C = 11$  entonces  $D_1 = 0$
- Nota<sup>1</sup>
  - Se ha usado aritmética módulo 11 al elegir los factores por los que se han multiplicado las 8 primeras cifras de CCC, es decir (4, 8, 5, 10, 9, 7, 3, 6)
  - Estos 8 factores son los valores de  $2^2, 2^3, 2^4, 2^5, 2^6, 2^7, 2^8, 2^9$  y  $2^{10}$  (módulo 11).
  - Esto se hace para detectar una posible trasposición de números, por si se escribe 38 en lugar de 83, por ejemplo.

<sup>1</sup> [https://www.lasexta.com/tecnologia-tecnoplora/ciencia/divulgacion/iban-asi-calculan-numeros-cuenta-bancaria\\_2014020957fca03d0cf2fd8cc6b0e1a2.html](https://www.lasexta.com/tecnologia-tecnoplora/ciencia/divulgacion/iban-asi-calculan-numeros-cuenta-bancaria_2014020957fca03d0cf2fd8cc6b0e1a2.html)

- **Segundo dígito de control:  $D_2$** 
  - Procesamiento de las cifras de la cuenta corriente personal
    - Sean  $N_1 N_2 N_3 N_4 N_5 N_6 N_7 N_8 N_9 N_{10}$  las diez cifras de la cuenta corriente personal
    - $D = 1 N_1 + 2 N_2 + 4 N_3 + 8 N_4 + 5 N_5 + 10 N_6 + 9 N_7 + 7 N_8 + 3 N_9 + 6 N_{10}$
  - Cálculo del resto de  $D$  módulo 11
    - $E = D \text{ módulo } 11$
  - Segundo dígito de control  $D_2$ :
    - Si  $11 - E < 10$  entonces  $D_2 = 11 - E$
    - Si  $11 - E = 10$  entonces  $D_2 = 1$
    - Si  $11 - E = 11$  entonces  $D_2 = 0$
  - Nota<sup>2</sup>
    - Los factores 1, 2, 4, 8, 5, 10, 9, 7, 3, 6 se corresponden con los valores de  $2^0, 2^1, 2^2, 2^3, 2^4, 2^5, 2^6, 2^7, 2^8, 2^9$  y  $2^{10}$  (módulo 11) para controlar los errores de transposición.
- **Dígitos de IBAN: X e Y**
  - Se forma el siguiente número de 26 dígitos  
 $N = C_1 C_2 C_3 C_4 C_5 C_6 C_7 C_8 D_1 D_2 N_1 N_2 N_3 N_4 N_5 N_6 N_7 N_8 N_9 N_{10} \underline{14} \underline{28} \underline{00}$ 
    - Donde el número 14 corresponde a la letra E y el 28 a la S
  - Cálculo de los dígitos de IBAN
    - $R = N \text{ módulo } 97$
    - Si  $98 - R$  tiene dos cifras entonces esas son las que forman el código XY del IBAN
    - Si  $98 - R$  solamente tiene una cifra, se pone un 0 delante.
      - ✓ Por ejemplo:  $98 - R = 4$ , el código IBAN comienza con ES04

<sup>2</sup> [https://www.lasexta.com/tecnologia-tecnologia/ciencia/divulgacion/iban-asi-calculan-numeros-cuenta-bancaria\\_2014020957fca03d0cf2fd8cc6b0e1a2.html](https://www.lasexta.com/tecnologia-tecnologia/ciencia/divulgacion/iban-asi-calculan-numeros-cuenta-bancaria_2014020957fca03d0cf2fd8cc6b0e1a2.html)