



UNIVERSIDAD DE CÓRDOBA

ESCUELA POLITÉCNICA SUPERIOR

DEPARTAMENTO DE
INFORMÁTICA Y ANÁLISIS NUMÉRICO



LENGUAJES

DE INTELIGENCIA ARTIFICIAL

INGENIERÍA TÉCNICA EN INFORMÁTICA DE GESTIÓN

INGENIERÍA TÉCNICA EN INFORMÁTICA DE SISTEMAS

SEGUNDO CURSO

PRIMER CUATRIMESTRE

CURSO ACADÉMICO 2009 - 2010



Primera
parte:
Scheme

Tema 1.- Introducción al Lenguaje Scheme

Tema 2.- Expresiones y Funciones

Tema 3.- Predicados y sentencias condicionales

Tema 4.- Iteración y Recursión

Tema 5.- Tipos de Datos Compuestos

Tema 6.- Abstracción de Datos

Tema 7.- **Lectura y Escritura**

Segunda
parte: Prolog

Tema 8.- Introducción al Lenguaje Prolog

Tema 9.- Elementos Básicos de Prolog

Tema 10.- Listas

Tema 11.- Reevaluación y el "corte"

Tema 12.- Entrada y Salida

Primera parte: **Scheme**

Tema 1.- Introducción al Lenguaje Scheme

Tema 2.- Expresiones y Funciones

Tema 3.- Predicados y sentencias condicionales

Tema 4.- Iteración y Recursión

Tema 5.- Tipos de Datos Compuestos

Tema 6.- Abstracción de Datos

Tema 7.- **Lectura y Escritura**

5. Interacción entre funciones y ficheros

- ✓ Procedimientos que cambian los puertos de entrada o salida actuales:
 - with-input-from-file:
 - Cambia el puerto de entrada actual → *current-input-port*
 - with-output-to-file:
 - Cambia el puerto de salida actual → *current-output-port*

- ✓ Procedimientos que operan directamente con los ficheros:
 - call-with-input-file: utiliza un fichero de entrada
 - call-with-output-file: utiliza un fichero de salida

Forma especial:

with-input-from-file

Sintaxis:

(with-input-from-file f p)

f: nombre de un fichero existente

p: procedimiento sin argumentos

Significado:

1. Abre el fichero *f* y lo asocia al *puerto de entrada actual* → *current-input-port*
2. Llama al procedimiento *p*, que realiza sus operaciones de *lectura* desde el fichero asociado al *puerto de entrada actual*

Ejemplo:

```
(define (contar-palabras)
  (cond
    ((eof-object? (read)) 0)
    (else (+ 1 (contar-palabras))))
)
```

:: Llamada

```
(with-input-from-file "datos.txt" contar-palabras)
```

Forma especial:

with-output-to-file

Sintaxis:

(with-output-to-file f p)

f: nombre de un fichero que se va a crear

p: procedimiento sin argumentos

Significado:

1. Abre el fichero *f* y lo asocia al *puerto de salida actual* → *current-output-port*
2. Llama al procedimiento *p*, que realiza sus operaciones de *escritura* en el fichero asociado al *puerto de salida actual*

Ejemplo:

```
(with-output-to-file  
  "salida.txt"  
  (lambda () (load "hanoi.s")))
```

Observación: el procedimiento *p* de *with-output-to-file* es

```
(lambda () (load "hanoi.s"))
```

Donde *"hanoi.s"* es el fichero que contiene el procedimiento que resuelve el problema de la Torres de Hanoi.


```

;; Fichero "hanoi.s"
;; Definición de la función "hanoi"
(define (hanoi n a b c)
  (define (cambio a b)
    (display a)
    (display " → ")
    (display b)
    (newline)
    1)
  (cond ((= n 1) (cambio a b))
        (else (+
                (hanoi (- n 1) a c b)
                (cambio a b)
                (hanoi (- n 1) c b a)
                )
        )
  )
)

```

```

;; Llamada a la función "hanoi"
(hanoi 3 "a" "b" "c")

```

Forma especial:

call-with-input-file

Sintaxis:

(call-with-input-file f p)

f: nombre de un fichero existente

p: procedimiento que recibe como argumento un puerto asociado a un fichero de entrada.

Significado:

- 1. Abre el fichero f y lo asocia a un puerto de entrada*
- 2. Llama al procedimiento p, que realiza sus operaciones de lectura desde el fichero asociado al puerto de entrada abierto en el paso anterior*

Ejemplo:

```
(define (sumar puerto)
  (let
    ;; variables de let
    (
      (n (read puerto))
    )
    ;; cuerpo de let
    (cond
      ((eof-object? n) 0)
      (else (+
              n
              (sumar puerto)
            )
      )
    )
  )
)

;; Llamada
(call-with-input-file "datos.txt" sumar)
```

Equivalencia

:: Llamada
(call-with-input-file "datos.txt" sumar)

Es equivalente a

:: Se abre el fichero de entrada y se asocia al puerto de entrada
(define puerto (open-input-file "datos.txt"))

:: Se llama a la función o procedimiento
(sumar puerto)

:: Se cierra el puerto de entrada
(close-input-port puerto)

Forma especial:

call-with-output-file

Sintaxis:

(call-with-output-file f p)

f: nombre de un fichero que se va a crear

p: procedimiento que recibe como argumento un puerto asociado a un fichero de salida.

Significado:

- 1. Abre el fichero f y lo asocia a un puerto de salida*
- 2. Llama al procedimiento p, que realiza sus operaciones de escritura en el fichero asociado al puerto de salida abierto en el paso anterior*

Ejemplo:

```
(define (escribir-carta puerto)
  (do
    (
      (a (read) (read))
    )
  )
```

```
;; condición de salida
((eof-object? a) (newline puerto))
```

```
;; cuerpo de salida
(display a puerto)
(display " " puerto)
)
)
```

```
;; Llamada
(call-with-output-file "nota.txt" escribir-carta)
```

Equivalencia

:: Llamada

(call-with-output-file "nota.txt" escribir-carta)

Es equivalente a

:: Se abre el fichero de salida y se asocia al puerto de salida

(define puerto (open-output-file "nota.txt"))

:: Se llama a la función o procedimiento

(escribir-carta puerto)

:: Se cierra el puerto de salida

(close-output-port puerto)



UNIVERSIDAD DE CÓRDOBA

ESCUELA POLITÉCNICA SUPERIOR

DEPARTAMENTO DE
INFORMÁTICA Y ANÁLISIS NUMÉRICO



LENGUAJES

DE INTELIGENCIA ARTIFICIAL

INGENIERÍA TÉCNICA EN INFORMÁTICA DE GESTIÓN

INGENIERÍA TÉCNICA EN INFORMÁTICA DE SISTEMAS

SEGUNDO CURSO

PRIMER CUATRIMESTRE

CURSO ACADÉMICO 2009 - 2010

