UNIVERSIDAD DE CÓRDOBA

ESCUELA POLITÉCNICA SUPERIOR DE CÓRDOBA

DEPARTAMENTO DE
INFORMÁTICA Y ANÁLISIS NUMÉRICO

PROGRAMACIÓN DECLARATIVA

INGENIGERÍA INFORMÁTICA

CUARTO CURSO

PRIMER CUATRIMESTRE

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

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

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

Índice

Índice

1. Lectura y escritura de términos

- Escritura
- Lectura

1. Lectura y escritura de términos

- **Escritura**
- Lectura

1. Lectura y escritura de términos

- **Escritura**
  - *write y display*
  - Ejemplos

1. Lectura y escritura de términos

- **Escritura**
  - ***write y display***
  - Ejemplos

1. Lectura y escritura de términos

- **Escritura**
  - ***write y display***
    - **Sintaxis**

write(*argumento*)

display(*argumento*)

❑ Argumento:

✓ número,

✓ átomo,

✓ estructura,

✓ lista,

✓ etc.

1. Lectura y escritura de términos

- **Escritura**
  - ***write y display***
    - **Descripción**
      - Escriben el valor del argumento en el dispositivo de salida actual (*current output device*) que, por defecto es la pantalla.
      - ***display*** muestra la representación interna de las estructuras, considerando como tales a las listas y las expresiones aritméticas.

1. Lectura y escritura de términos

- **Escritura**
  - ***write y display***
    - **Ejemplos**
      - Números, átomos y estructuras.

*?-**write**(12).* *12*	*?-**display**(12).* *12*
*?-**write**(luz).* *luz*	*?-**display**(luz).* *luz*
*?-**write**(autor('Juan', 'Varela')).* *autor(Juan, Varela).*	*?-**display**(autor('Juan', 'Varela')).* *autor(Juan, Varela).*

1. Lectura y escritura de términos

- **Escritura**
  - ***write y display***
    - **Ejemplos**
      - Variables (1/3)
        - ✓ Si una variable no está instanciada, muestra su dirección de memoria

          ?- write(X), display(X).

          _G2062_G2062

          true.

1. Lectura y escritura de términos

- **Escritura**
  - ***write y display***
    - **Ejemplos**
      - Variables (2/3)
        - ✓ Si una variable está instanciada, muestra su valor

?- factorial(3,R), write(R),tab(1), display(R).

6 6

R = 6

true.

1. Lectura y escritura de términos

- **Escritura**
  - ***write y display***
    - **Ejemplos**
      - Variables (3/3)
        - ✓ Si dos variables "comparten" memoria y no están instanciadas, se muestra la misma dirección de memoria.

?- X=Y, write(X), display(X), write(Y),display(Y).

_G2868_G2868_G2868_G2868

X = Y

1. Lectura y escritura de términos

- **Escritura**
  - ***write y display***
    - **Ejemplos**
      - Expresiones aritméticas
        - ✓ ***write*** no evalúa la expresión, pero muestra cada uno de los argumentos.
        - ✓ ***Display*** no evalúa la expresión, pero muestra su representación como estructura.

*?-**write**(2+3).* *2+3*	*?-**display**(2+3).* *+(2,3)*
*?-X is 2, **write**(X+3).* *2+3*	*?-X is 2, **display**(X+3).* *+(2,3)*

1. Lectura y escritura de términos

- **Escritura**
  - ***write y display***
    - **Ejemplos**
      - Listas
        - ✓ ***write*** muestra cada uno de los argumentos.
        - ✓ ***display*** muestra la representación interna.

*?-**write**([1,2,3]).* *[1,2,3].*	*?- **display**([1,2,3]).* *.(1,.(2,.(3,[])))*
*?- X is 2, **write**([1,X,Y]).* *[1,2,_G326]* *X = 2*	*?- X is 2, **display**([1,X,Y]).* *.(1,.(2,.(_G338,[])))* *X = 2*

1. Lectura y escritura de términos

- **Escritura**
  - ***write y display***
    - **Ejemplos**
      - Cadenas de caracteres
        - ✓ ***write*** muestra una lista de códigos ASCII
        - ✓ ***display*** muestra la representación interna de dicha lista.

*?- **write**("Hola").* *[72,111,108,97]* *true.*	*?- **display**("Hola").* *.(72,.(111,.(108,.(97,[]))))* *true.*

1. Lectura y escritura de términos

- **Escritura**
  - *write y display*
  - **Ejemplos**

1. Lectura y escritura de términos

- **Escritura**
  - **Ejemplos**
    - Torres de Hanoi
    - Escritura de listas

1. Lectura y escritura de términos

- **Escritura**
  - **Ejemplos**
    - **Torres de Hanoi**
    - Escritura de listas

1. Lectura y escritura de términos

- **Escritura**
  - **Ejemplos**
    - **Torres de Hanoi**
      - Primera parte

```
hanoi(N):- mover(N,izquierda,centro,derecha).

mover(1,A,_,C):- escribir_movimiento(A,C), !.

mover(N,A,B,C):- N1 is N-1,
                 mover(N1,A,C,B),
                 escribir_movimiento(A,C),
                 mover(N1,B,A,C).
```

1. Lectura y escritura de términos

- **Escritura**
  - **Ejemplos**
    - **Torres de Hanoi**
      - Segunda parte

```
escribir_movimiento(Origen, Destino):-
                    nl,
                    write(Origen),
                    write(' --> '),
                    write(Destino).
```

1. Lectura y escritura de términos

- **Escritura**
  - **Ejemplos**
    - **Torres de Hanoi**

?- hanoi(3).

izquierda --> derecha

izquierda --> centro

derecha --> centro

izquierda --> derecha

centro --> izquierda

centro --> derecha

izquierda --> derecha

true.

1. Lectura y escritura de términos

- **Escritura**
  - **Ejemplos**
    - Torres de Hanoi
    - **Escritura de listas**

1. Lectura y escritura de términos

- **Escritura**
  - **Ejemplos**
    - **Escritura de listas**
      - ❑ Escritura de una lista en una fila
      - ❑ Escritura de una lista en una columna
      - ❑ Escritura sangrada de una lista con sublistas

1. Lectura y escritura de términos

- **Escritura**
  - **Ejemplos**
    - **Escritura de listas**
      - Escritura de una lista en una **fila**

```
escribir_fila([]).

escribir_fila([Cabeza|Cola]):-
        write(Cabeza),
        tab(1),
        escribir_fila(Cola).

?- escribir_fila([1,2,3,4]).
1 2 3 4
true.
```

1. Lectura y escritura de términos

- **Escritura**
  - **Ejemplos**
    - **Escritura de listas**
      - Escritura de una lista en una **columna**

```
escribir_columna([]).
escribir_columna([Cabeza|Cola]):-
        write(Cabeza),
        nl,
        escribir_columna(Cola).
```

1. Lectura y escritura de términos

- **Escritura**
  - **Ejemplos**
    - **Escritura de listas**
      - Escritura de una lista en una **columna**

        ?- escribir_columna([a,b,c,d]).

        a

        b

        c

        d

        true

1. Lectura y escritura de términos

- **Escritura**
  - **Ejemplos**
    - **Escritura de listas**
      - Escritura **sangrada** de una lista con sublistas
        - ✓ Versión con "not" (1/6)

```
/* El argumento no es una lista */
escribir_lista(X,Columna):-
                    not(es_lista(X)),
                    tab(Columna),
                    write(X),
                    nl.
```

1. Lectura y escritura de términos

- **Escritura**
  - **Ejemplos**
    - **Escritura de listas**
      - ❑ Escritura **sangrada** de una lista con sublistas
        - ✓ Versión con "not" (2/6)

```
/* El argumento es una lista con cabeza y cola */
escribir_lista([Cabeza|Cola],Columna):-
                    Lugar is Columna + 3,
                    escribir_lista(Cabeza,Lugar),
                    escribir_sublista(Cola,Lugar).
```

1. Lectura y escritura de términos

- **Escritura**
  - **Ejemplos**
    - **Escritura de listas**
      - ❑ Escritura **sangrada** de una lista con sublistas
        - ✓ Versión con "not" (3/6)

```
/* Si la sublista es vacía, no escribe nada */
escribir_sublista([],_).
```

1. Lectura y escritura de términos

- **Escritura**
  - **Ejemplos**
    - **Escritura de listas**
      - ❑ Escritura **sangrada** de una lista con sublistas
        - ✓ Versión con "not" (4/6)

```
/* Si la sublista no es vacía, se escribe la cabeza
   y la cola */

escribir_sublista([Cabeza|Cola],Columna):-
                    escribir_lista(Cabeza,Columna),
                    escribir_sublista(Cola,Columna).
```

1. Lectura y escritura de términos

- **Escritura**
  - **Ejemplos**
    - **Escritura de listas**
      - ❑ Escritura **sangrada** de una lista con sublistas
        - ✓ Versión con "not" (5/6)

```
/* Se comprueba si el argumento es una lista */
/* Es la lista vacía */
es_lista([]).
/* Es una lista que posee cabeza y cola */
es_lista([_|Cola]):- es_lista(Cola).
```

1. Lectura y escritura de términos

- **Escritura**
  - **Ejemplos**
    - **Escritura de listas**
      - ❑ Escritura **sangrada** de una lista con sublistas
        - ✓ Versión con "not" (6/6)

?- escribir_lista([a,[b,c],d,[e]],10).

a

b

c

d

e

true.

1. Lectura y escritura de términos

- **Escritura**
  - **Ejemplos**
    - **Escritura de listas**
      - ❑ Escritura **sangrada** de una lista con sublistas
        - ✓ Versión con "el corte" (1/4)

```
/* El argumento es una Lista que posee Cabeza y Cola */
escribir_lista([Cabeza|Cola],Columna):-
                    !,
                    Lugar is Columna + 3,
                    escribir_lista(Cabeza,Lugar),
                    escribir_sublista(Cola,Lugar).
```

1. Lectura y escritura de términos

- **Escritura**
  - **Ejemplos**
    - **Escritura de listas**
      - ❑ Escritura **sangrada** de una lista con sublistas
        - ✓ Versión con "el corte" (2/4)

```
/* El argumento es un elemento */
escribir_lista(X,Columna):-
                    tab(Columna),
                    write(X),
                    nl.
```

1. Lectura y escritura de términos

- **Escritura**
  - **Ejemplos**
    - **Escritura de listas**
      - ❑ Escritura **sangrada** de una lista con sublistas
        - ✓ Versión con "el corte" (3/4)

```
/* Si es la sublista está vacía, no escribe nada */
        escribir_sublista([],_).
/* El argumento es una Sublista con Cabeza y Cola */
        escribir_sublista([Cabeza|Cola],Columna):-
                escribir_lista(Cabeza,Columna),
                escribir_sublista(Cola,Columna).
```

1. Lectura y escritura de términos

- **Escritura**
  - **Ejemplos**
    - **Escritura de listas**
      - ❑ Escritura **sangrada** de una lista con sublistas
        - ✓ Versión con "el corte" (4/4)

?- escribir_lista([a,[b,c],d,[e]],10).

a

b

c

d

e

true.

1. Lectura y escritura de términos

- Escritura
- **Lectura**

1. Lectura y escritura de términos

- Lectura
  - ***read***

1. Lectura y escritura de términos

- **Lectura**
  - ***read***
    - **Sintaxis**

      read*(Variable)*

1. Lectura y escritura de términos

- **Lectura**
  - ***read***
    - **Descripción**
      - Lee el siguiente **término,**
        - ✓ que debe terminar en punto “.”,
        - ✓ que esté disponible en el dispositivo de entrada actual (***current input device***)
        - ✓ que, por defecto, es el **teclado**.
      - La variable quedará instanciada con el valor leído.
      - Si la variable estuviera instanciada antes de la lectura, se comprobará si el término leído es igual al valor de la variable.

1. Lectura y escritura de términos

- **Lectura**
  - ***read***
    - **Ejemplos**

?- read(X), write(X).

|: 1.

1

X = 1.

?- read(X), write(X).

|: agua.

agua

X = agua.

Se escribe el punto "." para finalizar

1. Lectura y escritura de términos

- Lectura
  - ***read***
    - **Ejemplos**

?- read(X), write(X).

|: 2+3.

2+3

X = 2+3.

?- read(X), write(X).

|: autor('Juan','Varela').

autor(Juan,Varela)

X = autor('Juan', 'Varela').

1. Lectura y escritura de términos

- **Lectura**
  - ***read***
    - **Ejemplos**

?- X is 2, read(X).

|: 2.

X = 2.

?- X is 2, read(X).

|: 3.

false

1. Lectura y escritura de términos

- **Lectura**
  - ***read***
    - **Ejemplos**

?- read(X), write(X).

|: Dato. ⟵ El Dato leído es una variable

_G287

true.

1. Lectura y escritura de términos

- **Lectura**
  - ***read***
    - **Ejemplos**

?- read(X), write(X).

|: [a,b,c].

[a,b,c]

X = [a, b, c].

?- read(X), write(X).

|: [a,B,c].

[a,_G411,c]

X = [a, _G411, c].

1. Lectura y escritura de términos

- **Lectura**
  - ***read***
    - **Ejemplos**

```
padre(juan,miguel).
padre(marta,miguel).
padre(carmen,miguel).
buscar_padre:- write('Nombre --> '),
               read(X),
               write('Padre de '), write(X), write(' es '),
               padre(X,Y),
               write(Y).
```

1. Lectura y escritura de términos

- **Lectura**
  - ***read***
    - **Ejemplos**

?- buscar_padre.

Nombre --> marta.

Padre de marta es miguel

true.

Índice

2. Lectura y escritura de caracteres

- Escritura
- Lectura

2. Lectura y escritura de caracteres

- **Escritura**
- Lectura

2. Lectura y escritura de caracteres

- **Escritura**
  - *nl*
  - *put*
  - *tab*
  - Escritura de cadenas de caracteres

2. Lectura y escritura de caracteres

- **Escritura**
  - **Observación**
    - Todas las sentencias de escritura
      - ✓ se realizan en el dispositivo de salida actual (***current output device***)
      - ✓ que, por defecto, es la **pantalla.**

2. Lectura y escritura de caracteres

- **Escritura**
  - ***nl***
  - *put*
  - *tab*
  - Escritura de cadenas de caracteres

2. Lectura y escritura de caracteres

- **Escritura**
  - ***nl***
    - **Descripción**
      - ❑ Escribe un salto de línea (*new line*).
      - ❑ Solamente se satisface una vez.
    - **Ejemplo**

      *?- write(1), **nl**, write(2).*

      1

      2

      true

2. Lectura y escritura de caracteres

- **Escritura**
  - *nl*
  - ***put***
  - *tab*
  - Escritura de cadenas de caracteres

2. Lectura y escritura de caracteres

- **Escritura**
  - ***put***
    - **Sintaxis**

      put(argumento)

    - **Descripción**
      - ❑ El argumento debe ser
        - ✓ un átomo con un carácter
        - ✓ o un valor numérico que se corresponda con un carácter,
        - ✓ o una cadena de caracteres con un carácter
      - ❑ Solamente se satisface una vez.

2. Lectura y escritura de caracteres

- **Escritura**
  - ***put***
    - **Ejemplo**

*?- **put**('L').*

L

true.

*?- **put**(76).*

L

true.

*?- **put**("L").*

L

true.

2. Lectura y escritura de caracteres

- **Escritura**
  - ***put***
    - **Ejemplo**

      ?- put(104), put(111), put(108), put(97).

      hola

      true.

2. Lectura y escritura de caracteres

- **Escritura**
  - *nl*
  - *put*
  - ***tab***
  - Escritura de cadenas de caracteres

2. Lectura y escritura de caracteres

- **Escritura**
  - ***tab***
    - **Sintaxis**

      tab(argumento)

    - **Descripción**
      - ❑ El argumento debe contener un valor numérico
      - ❑ Escribe el número de espacios en blanco indicados por el argumento.
      - ❑ Solamente se satisface una vez.

2. Lectura y escritura de caracteres

- **Escritura**
  - ***tab***
    - **Equivalencia**

```
tab(0):- !.
tab(N):- put(32),
         M is N-1,
         tab(M).
```

2. Lectura y escritura de caracteres

- **Escritura**
  - ***tab***
    - **Ejemplo**

?- write(uno), tab(1),write(diez), tab(10),write(fin).

uno diez fin

true.

2. Lectura y escritura de caracteres

- **Escritura**
  - *nl*
  - *put*
  - *tab*
  - **Escritura de cadenas de caracteres**

2. Lectura y escritura de caracteres

- **Escritura**
  - **Escritura de cadenas de caracteres**
    - **Definición**

```
escribir_cadena([]).
escribir_cadena([Cabeza|Cola]):-
                    put(Cabeza),
                    escribir_cadena(Cola).
```

2. Lectura y escritura de caracteres

- **Escritura**
  - **Escritura de cadenas de caracteres**
    - **Ejemplo**

?- escribir_cadena("Cadena maravillosa").

Cadena maravillosa

true.

2. Lectura y escritura de caracteres

- Escritura
- **Lectura**

2. Lectura y escritura de caracteres

- **Lectura**
  - *get0*
  - *get*
  - Lectura de una frase y transformación en átomos

2. Lectura y escritura de caracteres

- **Lectura**
  - **Observación**
    - Todas las sentencias de lectura se realiza
      - ✓ en el dispositivo de entrada actual (***current output device***)
      - ✓ que, por defecto, es el **teclado**, también denominado "***user***".

2. Lectura y escritura de caracteres

- **Lectura**
  - ***get0***
  - *get*
  - Lectura de una frase y transformación en átomos

2. Lectura y escritura de caracteres

- **Lectura**
  - ***get0***
    - **Sintaxis**

      get0*(Variable)*
    - **Descripción**
      - ❑ Lee el siguiente **carácter** que se teclee.
      - ❑ La lectura finaliza al pulsar la tecla de "***enter***".

2. Lectura y escritura de caracteres

- **Lectura**
  - ***get0***
    - **Ejemplos**

?- get0(X), put(X).

|: a

a

X = 97.

?- get0(X), get0(Y), put(X), put(Y).

|: ab

ab

X = 97,

Y = 98

Se pulsa la tecla de "*enter*" para finalizar

2. Lectura y escritura de caracteres

- **Lectura**
  - ***get0***
    - **Ejemplos**

?- get0(X), put(X).

|:

X = 10.

Se pulsa la tecla de "*enter*" para finalizar

2. Lectura y escritura de caracteres

- **Lectura**
  - *get0*
  - ***get***
  - Lectura de una frase y transformación en átomos

2. Lectura y escritura de caracteres

- **Lectura**
  - ***get***
    - **Sintaxis**

      get*(Variable)*
    - **Descripción**
      - Lee el siguiente **carácter** **imprimible** que se teclee.
      - La lectura finaliza al pulsar la tecla de "*enter*".

2. Lectura y escritura de caracteres

- **Lectura**
  - ***get***
    - **Ejemplos**

?- get(X), put(X).
|:
|: b
X = 98.
?- get(X), get(Y), put(X), put(Y).
|:
|: a b
ab
X = 97,
Y = 98.

Se pulsa la tecla de "*enter*" para finalizar

2. Lectura y escritura de caracteres

- **Lectura**
  - *get0*
  - *get*
  - ***Lectura de una frase y transformación en átomos***

2. Lectura y escritura de caracteres

- **Lectura**
  - **Lectura de una frase y transformación en átomos**
    - Primera parte

```
leer_frase(Palabras):-
                    get0(Caracter),
                    leer_resto(Caracter,Palabras).
```

2. Lectura y escritura de caracteres

- **Lectura**
  - **Lectura de una frase y transformación en átomos**
    - Segunda parte

```
/* El punto "." (carácter 46) indica el fin de la frase */
   leer_resto(46,[]):- !.
/* Se omite el espacio en blanco (carácter 32) */
  leer_resto(32,Palabras):- !,
                            leer_frase(Palabras).

/* Lee los caracteres de la palabra actual */
leer_resto(Caracter,[Palabra|Palabras]):-
    leer_caracteres(Caracter,Caracteres,Siguiente_caracter),
    name(Palabra,Caracteres),
    leer_resto(Siguiente_caracter,Palabras).
```

2. Lectura y escritura de caracteres

- **Lectura**
  - **Lectura de una frase y transformación en átomos**
    - **Observación**

      name(Palabra,Caracteres)

      ❑ Hace la conversión entre un átomo y una cadena de caracteres.

    - **Ejemplos**

      ?- name(Palabra,"Cadena").
      Palabra = 'Cadena'.

      ?- name(cadena,Caracteres).
      Caracteres = [99, 97, 100, 101, 110, 97]

2. Lectura y escritura de caracteres

- **Lectura**
  - **Lectura de una frase y transformación en átomos**
    - Tercera parte

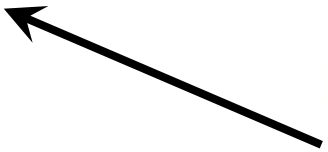
```
/* Fin de palabra: 46 = punto "." */
leer_caracteres(46,[],46):- !.
/* Fin de palabra: 32 = espacio en blanco */
leer_caracteres(32,[],32):- !.

leer_caracteres(Caracter,
                [Caracter|Caracteres],
                Siguiente_caracter):-
   get0(Nuevo_Caracter),
   leer_caracteres(Nuevo_Caracter,Caracteres,
   Siguiente_caracter).
```

2. Lectura y escritura de caracteres

- **Lectura**
  - **Lectura de una frase y transformación en átomos**
    - **Ejemplo**

?- leer_frase(X).
| Esta frase va a ser transformada.

X = [‘Esta’, frase, va, a, ser, transformada]



Se utilizan las comillas simples para que no sea una variable, sino un átomo.

Índice

3. Lectura y escritura usando ficheros

- Introducción
- Lectura desde un fichero
- Escritura en un fichero
- Ejemplos finales

3. Lectura y escritura usando ficheros

- **Introducción**
- Lectura desde un fichero
- Escritura en un fichero
- Ejemplos finales

3. Lectura y escritura usando ficheros

• Introducción

- **Nombres de los ficheros**
  - Se representan como átomos de Prolog, escribiéndolos entre comillas simples.
  - Ejemplos

    '/home/usuario/fichero.txt'

    'salida.txt'
- Fichero de entrada por defecto
  - El teclado y se denomina "*user*"
- Fichero de salida por defecto
  - La pantalla y se denomina "*user*"

3. Lectura y escritura usando ficheros

- Introducción
- **Lectura desde un fichero**
- Escritura en un fichero
- Ejemplos finales

3. Lectura y escritura usando ficheros

- **Lectura desde un fichero**
  - *see*
  - *seeing*
  - *seen*

3. Lectura y escritura usando ficheros

- Lectura desde un fichero
  - ***see***
  - *seeing*
  - *seen*

3. Lectura y escritura usando ficheros

- ## Lectura desde un fichero

  - ***see***

    - **Sintaxis**

      see*(argumento)*

    - **Descripción**

      - Abre para lectura el fichero indicado por el argumento
      - El fichero pasa a ser el dispositivo de lectura actual de ***read***, ***get0*** y ***get***.

3. Lectura y escritura usando ficheros

- ## Lectura desde un fichero

  - ***see***

    - **Descripción (1/2)**

      - El argumento indica el dispositivo de entrada actual que será utilizado por ***read***, ***get0*** y ***get***.
      - Si el argumento es el átomo ***user*** entonces la lectura se realizará desde el teclado.

3. Lectura y escritura usando ficheros

• Lectura desde un fichero

○ *see*

▪ Descripción (2/2)

- ❑ Si el argumento indica un fichero entonces
  - ✓ si no estaba abierto, la lectura empieza desde el principio del fichero.
  - ✓ si ya estaba abierto, la lectura continúa desde el punto inmediatamente posterior a la de la última lectura.

3. Lectura y escritura usando ficheros

- ## Lectura desde un fichero

  - ***see***

    - **Ejemplos**

      - Apertura del fichero entrada.txt del directorio /home/usuario

        *?- **see**('/home/usuario/entada.txt').*

      - Apertura del fichero indicado por la variable X

        *?- read(X), **see**(X).*

3. Lectura y escritura usando ficheros

- **Lectura desde un fichero**
  - ***see***
    - **Ejemplos**

```
/* Se numeran los elementos leídos a partir de N */
    contar(N):-
        read(Termino),
        mostrar(Termino,N).

    mostrar(end_of_file,_):- !.
    mostrar(Termino,N):- write(N),
                         tab(2),
                         write(Termino),
                        nl,
                        N1 is N + 1,
                        contar(N1).
```

3. Lectura y escritura usando ficheros

- ## Lectura desde un fichero

  - ### *see*

    - **Ejemplos**

```
/* Contenido del fichero entrada.txt */
agua.
fuego.
tierra.
aire.

?- see('entrada.txt'), contar(1).
1  agua
2  fuego
3  tierra
4  aire
true.
```

3. Lectura y escritura usando ficheros

- Lectura desde un fichero
  - *see*
  - ***seeing***
  - *seen*

3. Lectura y escritura usando ficheros

- **Lectura desde un fichero**
  - ***seeing***
    - **Sintaxis**

      seeing*(argumento)*

    - **Descripción**
      - Si argumento es una variable no instanciada entonces le asocia el nombre del dispositivo de entrada actual.
      - Si argumento es una variable instanciada o una constante entonces se comprueba si es el nombre del dispositivo de entada actual.

3. Lectura y escritura usando ficheros

- ## Lectura desde un fichero

  - ***seeing***

    - **Ejemplos**

      *?-**seeing**(‘datos’).*

      - Es cierto si ***datos*** es el dispositivo de lectura actual.

      *?- **seeing**(X).*

      - Si X no tiene un valor entonces le asigna a X el valor del fichero de lectura actual.
      - Si X tiene un valor, se comprueba si coincide con el valor del fichero de lectura actual.

3. Lectura y escritura usando ficheros

- Lectura desde un fichero
  - *see*
  - *seeing*
  - ***seen***

3. Lectura y escritura usando ficheros

- **Lectura desde un fichero**
  - ***seen***
    - **Sintaxis**

      seen

    - **Descripción**
      - Cierra el fichero de lectura actual, volviendo el teclado (*user*) a ser el dispositivo de lectura actual.

3. Lectura y escritura usando ficheros

- Introducción
- Lectura desde un fichero
- **Escritura en un fichero**
- Ejemplos finales

3. Lectura y escritura usando ficheros

- **Escritura en un fichero**
  - *tell*
  - *telling*
  - *told*

3. Lectura y escritura usando ficheros

• Escritura en un fichero

○ Observación

- Todas las sentencias de escritura se realizan en el dispositivo de salida actual (***current output device***) que, por defecto, es la **pantalla**.

3. Lectura y escritura usando ficheros

- **Escritura en un fichero**
  - ***tell***
  - *telling*
  - *told*

3. Lectura y escritura usando ficheros

- ## Escritura en un fichero
  - ***tell***
    - **Sintaxis**

tell*(argumento)*

3. Lectura y escritura usando ficheros

• Escritura en un fichero

○ *tell*

▪ Descripción (1/2)

- ❑ El argumento indica el dispositivo de salida actual que será utilizado por ***write***, ***display***, ***tab***, ***nl*** y ***put***.
- ❑ Si el argumento es el átomo ***user*** entonces la escritura se realizará en la pantalla.

3. Lectura y escritura usando ficheros

- ## Escritura en un fichero

  - ### *tell*

    - **Descripción (2/2)**

      - Si el argumento indica un fichero entonces
        - ✓ si no estaba abierto, se abre para escritura.
        - ✓ si ya estaba abierto, la escritura continúa desde el punto inmediatamente posterior al último carácter escrito previamente.

3. Lectura y escritura usando ficheros

- **Escritura en un fichero**
  - ***tell***
    - **Ejemplos**

?- ***tell****(‘salida.txt’).*

❑ Abre para escritura el fichero salida.txt

?- ***tell****(X).*

❑ Abre para escritura el fichero indicado por X

3. Lectura y escritura usando ficheros

- **Escritura en un fichero**
  - *tell*
  - ***telling***
  - *told*

3. Lectura y escritura usando ficheros

- **Escritura en un fichero**
  - ***telling***
    - **Sintaxis**

      telling*(argumento)*

    - **Descripción**
      - Si argumento es una variable no instanciada entonces le asocia el nombre del dispositivo de salida actual.
      - Si argumento es una variable instanciada o una constante entonces se comprueba si es el nombre del dispositivo de salida actual.

3. Lectura y escritura usando ficheros

- ## Escritura en un fichero
  - ***telling***
    - **Ejemplos**

      *?- **telling**('datos').*

      ❑ Es cierto si ***datos*** es el dispositivo de salida actual.

      *?- **telling**(X).*

      ❑ Si X no tiene un valor entonces le asigna a X el valor del fichero de escritura actual.

      ❑ Si X tiene un valor, se comprueba si coincide con el valor del fichero de escritura actual.

3. Lectura y escritura usando ficheros

- **Escritura en un fichero**
  - *tell*
  - *telling*
  - ***told***

3. Lectura y escritura usando ficheros

- ## Escritura en un fichero
  - ***told***
    - **Sintaxis**

      told

    - **Descripción**
      - Cierra el fichero asociado al dispositivo de salida actual, volviendo la pantalla (*user*) a ser el dispositivo de salida actual.

3. Lectura y escritura usando ficheros

- ## Escritura en un fichero

  - ### *told*

    - **Ejemplo**

```
/* Contenido del fichero entrada.txt */
agua.
fuego.
tierra.
aire.
/* Fin del contenido del fichero */
?-see('entrada.txt'),tell('salida.txt'),contar(1),told,seen.
true
/* Contenido del fichero salida.txt */
1  agua
2  fuego
3  tierra
4  aire
```

3. Lectura y escritura usando ficheros

- Introducción
- Lectura desde un fichero
- Escritura en un fichero
- **Ejemplos finales**

3. Lectura y escritura usando ficheros

- **Ejemplos finales**
  - Leer un fichero y escribirlo por pantalla
  - Pedir el nombre de un fichero, leerlo y escribirlo por pantalla
  - Fuente:

    http://www.cpp.edu/~jrfisher/www/prolog_tutorial/contents.html

3. Lectura y escritura usando ficheros

- ## Ejemplos finales
  - **Leer un fichero y escribirlo por pantalla**
  - Pedir el nombre de un fichero, leerlo y escribirlo por pantalla

3. Lectura y escritura usando ficheros

- **Ejemplos finales**

  - **Leer un fichero y escribirlo por pantalla**

```
browse(File) :-
        seeing(Old),        /* save for later */
        see(File),          /* open this file */
        repeat,
        read(Data),         /* read from File */
        process(Data),
        seen,               /* close File */
        see(Old),           /*  previous read source */
        !.                  /* stop now */

process(end_of_file) :- !.
process(Data) :- write(Data), nl, fail.
```

3. Lectura y escritura usando ficheros

- **Ejemplos finales**
  - **Leer un fichero y escribirlo por pantalla**

?- [browser].
% browser compiled 0.00 sec, 1 clauses
true.

?- browse('entrada.txt').
agua.
fuego.
tierra.
aire.
true.

3. Lectura y escritura usando ficheros

- ## Ejemplos finales
  - Leer un fichero y escribirlo por pantalla
  - **Pedir el nombre de un fichero, leerlo y escribirlo por pantalla**

3. Lectura y escritura usando ficheros

- **Ejemplos finales**
  - **Pedir el nombre de un fichero, leerlo y escribirlo por pantalla**

```
browse :-  seeing(Old),      /* save for later */
           see(user),
           write('Enter name of file to browse: '), read(File),
           see(File),        /* open this file */
           repeat,
           read(Data),       /* read from File */
           process(Data),
           seen,             /* close File */
           see(Old),         /*  previous read source */
           !.                /* stop now */
process(end_of_file) :- !.
process(Data):-  write(Data), nl, fail.
```

3. Lectura y escritura usando ficheros

- ## Ejemplos finales

  - **Pedir el nombre de un fichero, leerlo y escribirlo por pantalla**

    ?- [browser_interactivo].
    % browser_interactivo compiled 0.00 sec, 1 clauses
    true .

    ?- browse.
    Enter name of file to browse: 'entrada.txt'.
    agua
    fuego
    tierra
    aire
    ***true**.*

UNIVERSIDAD DE CÓRDOBA

ESCUELA POLITÉCNICA SUPERIOR DE CÓRDOBA

DEPARTAMENTO DE
INFORMÁTICA Y ANÁLISIS NUMÉRICO

PROGRAMACIÓN DECLARATIVA

INGENIGERÍA INFORMÁTICA

CUARTO CURSO

PRIMER CUATRIMESTRE

Tema 12.- Entrada y salida