

# PRÁCTICAS

2016-2017

## SESIÓN 3

**PLMAN**

Descarga / Instalación/  
Autenticarse  
Uso básico

**PLMAN-SE MUEVE**

Ciclo de  
ejecución

Acciones:  
Moverse,  
coger, dejar, y usar  
objetos

Sensor de  
visión normal

Ejemplos

**MÁS SOBRE REGLAS**

Múltiples  
cláusulas

Orden de  
Ejecución





Comentarios:	% Esto es un comentario
Predicados:	pelea(bartolo, palo).
Átomos (constantes):	carlos, 'Carlos López'
Números:	10, -212, 12.75
Variables:	X, Pepe, _valor, _
Hechos:	alumn(carlos). legusta(carlos,pilar).
Reglas:	buentipo(X) :- alum(X), moreno(X).



## Empezamos con Plman

## Web de prácticas

<http://logica.i3a.ua.es>

- Descargar **pl-man.zip**
- Descomprimir (por ej. en ~/Escritorio)    **\$ unzip pl-man.zip**
- Abrir terminal y Entrar carpeta plman /:  
**usuario@home/usuario: ~\$ cd Escritorio/plman**
- Comprobar que está todo: ls
  - docs/**                      - Tutorial de uso y otros.
  - maps/**                      - Mapas de ejemplo (para practicar pero no entregar).
  - pl-man-game/**           - Código fuente de Plman (comandos para resolver mapas).
  - plman**                      - Script de lanzamiento para compilar y ejecutar programas.

➤ *Puedes cambiar contraseña...*



## PASOS PARA **RESOLVER** UN MAPA de Plman

**1º** Obtener el mapa:

→ Carpeta: **plman/maps/...** (ejemplos de todas las fases: no evaluables).

→ **<http://logica.i3a.ua.es>** → Entrega de prácticas → Descargar

**2º** Incluir el mapa en carpeta plman.

**3º** Editar un fichero para escribir la solución al mapa **\$ gedit solucion.pl &**

**4º** En solucion.pl escribir:

-- en la 1ª línea: **`:- use_module('pl-man-game/main').`**

-- en líneas siguientes: **reglas** que resuelvan el mapa.

**5º** Ejecutar plman: **\$ ./plman mapa.pl solucion.pl regla-principal**

regla-principal: cabeza de la regla que resuelve el mapa.

*Golpe del espaciador: Plman ejecuta el código escrito en solución.pl.*

*Dentro de Plman: Esc + X se sale del modo ejecución y se vuelve al terminal.*



## EJEMPLO: Resolvemos mapaej1.pl

Descargar en la carpeta plman: </maps/ejemplos/mapaej1.pl>

Editar fichero solución: `$ gedit solej.pl &`

Escribir

```
:- use_module('pl-man-game/main').  
  
rej1.
```

Ejecutar solución `$ ./plman mapaej1.pl solej.pl rej1`

*! Plman se mueve !...??*



## ACCIONES PARA QUE PLMAN SE **MUEVA**

Plman puede moverse en cuatro posibles direcciones:

$D = \{ \text{right, left, up, down} \}$

predicado predefinido: **doAction(ACCION).**

ACCION : **move(X)**,  $X \in D$

doAction( **move(right)**)

doAction( **move(down)**)

doAction( **move(left)**)

doAction( **move(up)**)

Acción para **NO** MOVERSE,

doAction( **move(none)**)



Añade reglas para que Plman se mueva:

```
:- use_module('pl-man-game/main').  
rej1:- doAction(move(right)).  
rej1:- doAction(move(down)).  
rej1:- doAction(move(left)).
```

Ejecuta de nuevo : **\$ ./plman mapaej1.pl solej.pl rej1**

*¿ Qué sucede ...?*



Plman puede realizar acciones según lo que “vea” a su alrededor.

Predicado predefinido: **see/3**

**see(normal, DIR, OBJETO)**

**normal:** visión de la posición siguiente a la ubicación de Plman

**DIR** = { right, left, down, up, here, down-right, down-left, up-right, up-left }

**OBJETO:** objeto que ve en la dirección DIR

Ej. de lo que vee Plman en este mapa

**see(normal, right, ' ')** --> Éxito

**see(normal, right, '.')** --> Fracaso

**see(normal, left, 0)** --> Éxito

**see(normal, down, X)** --> instancia la variable X = 'o'







Añade predicados de **visión**

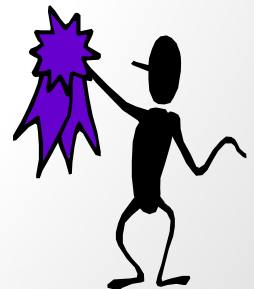
```
:- use_module('pl-man-game/main').  
  
rej1:- see(normal, right, '.'), doAction(move(right)).  
  
rej1:- see(normal, down, '.'), doAction(move(down)).  
  
rej1:- doAction(move(left)).
```

Ejecuta : **\$ ./plman mapaej1.pl solej.pl rej1**

**Ojo:** en el cuerpo de una regla se pueden añadir varios predicados see/3  
pero sólo un predicado doAction/1

*¿Plman se mueve tal como habíamos previsto?*

*¿Por qué en la última regla no se pone “see...”?*





- *Descargarlo en carpeta plman.*
- *Con fichero solución anterior (solej.pl) abrir este mapa para visualizarlo.*
- \$ ./plman mapaej8.pl solej.pl rej1**
- *Aparecen objetos: pistola: "I", enemigo: "E". (ver objetos abriendo: mapaej8.pl).*
- *Posible estrategia: **coger** la pistola y "**fulminar**" al enemigo usando pistola.*
- *Se necesita saber cómo coger, usar y dejar objetos*

**COGER** OBJETO: doAction(**get**(DIR))

**DEJAR** OBJETO: doAction(**drop**(DIR))

**USAR** OBJETO: doAction(**use**(DIR))

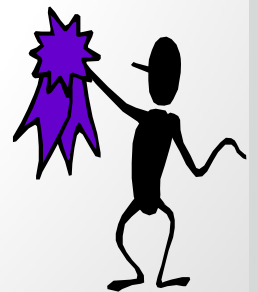
- *En cada turno Plman sólo puede realizar **una acción** (o moverse, o coger, o dejar...)*
- *Sólo puede llevar **un objeto***
- *Si necesita **usar** varios OBJETOS debe dejar el que lleva y coger el que necesite.*



## Reglas para solucionar mapaej8.pl

```
:- use_module('pl-man-game/main').  
rej8:- see(normal,down,'I'), doAction(get(down)).  
rej8:- see(normal,up,'.'), doAction(move(up)).  
rej8:- see(normal,down,'.'), doAction(move(down)).  
rej8:- see(normal,right,'.'), doAction(move(right)).  
rej8:- see(normal,up,' '), doAction(move(up)).  
rej8:- see(normal,right,'E'), doAction(use(right)).
```

*Puedes añadir la solución de este mapa al fichero solej.pl ya que al usar otras reglas (por ej rej8) no hay conflicto*





**1.** Para controlar las acciones de Plman usaremos n-reglas.

>> Pueden coincidir en la cabeza y tener distintos hechos en el cuerpo.

>> El cuerpo de cada regla controlará una sola acción de Plman.

>> Para cada turno tendremos una acción diferente.

>> *Escribimos tantas reglas como **condiciones** puedan pasarle a Plman en el transcurso de la resolución del mapa.*

### Ejemplo

% Si veo un coco a la izquierda, me muevo hacia la izquierda

regla\_pepito :- see(normal,left,'.'), doAction(move(left)).

% Si veo un espacio arriba, me muevo hacia arriba

regla\_pepito :- see(normal,up,' '), doAction(move(up)).

% Si nada de lo anterior ha funcionado, me muevo a la derecha

regla\_pepito :- doAction(move(right)).



**2.** Por la “forma” en que Prolog lleva a cabo la **ejecución** de un programa es muy **IMPORTANTE EL ORDEN** en el que se escriben las reglas.

En el fichero solej.pl **cambiar** el orden de algunas reglas de la regla rej8.

```
:- use_module('pl-man-game/main').  
rej8:- see(normal,down,'I'), doAction(get(down)).  
rej8:- see(normal,up,'.'), doAction(move(up)).  
rej8:- see(normal,down,'.'), doAction(move(down)).  
rej8:- see(normal,right,'.'), doAction(move(right)).  
rej8:- see(normal,up,' '), doAction(move(up)).  
rej8:- see(normal,right,'E'), doAction(use(right)).  
rej8:- see(normal,down,'I'), doAction(get(down)).  
rej8:- see(normal,up,'.'), doAction(move(up)).
```



Una misma regla puede tener varias cláusulas o, lo que es lo mismo, distintas formas de llegar a la misma conclusión

En PROLOG, cláusulas y objetivos se comprueban y ejecutan de arriba a abajo y de izquierda a derecha.

Usaremos una única regla para controlar a Plman, que puede cumplirse de distintas formas. En cada turno, se cumplirá de una forma distinta.



## NORMAS FUNDAMENTALES: para resolver mapas

- **No construir** la solución entera antes de probar.
- **Añadir** reglas de una en una e ir probando.
- Siempre **analizar** qué ha ocurrido antes de seguir.

### REGLA DE ORO

**Que Plman se coma siempre todos los cocos cuando pase por un sitio**

- >> **No dejes cocos al pasar** esperando que Plman vuelva luego a comérselos.
- >> Algunas veces, hacer que Plman vuelva, resulta **imposible**.



**Construir soluciones** para los  
mapas de **plman/maps/ejemplos**