PRÁCTICAS DE MATEMÁTICAS 1

2019/2020

PRESENTACIÓN

Ângelo Araujo Costa

angelogoncalo.costa@ua.com

Tutorías:

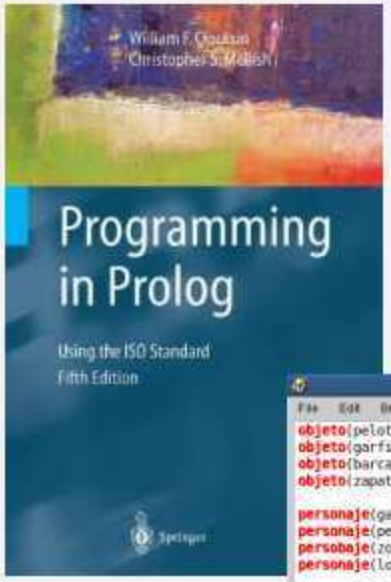
Lunes y Jueves → 11:00-13:00

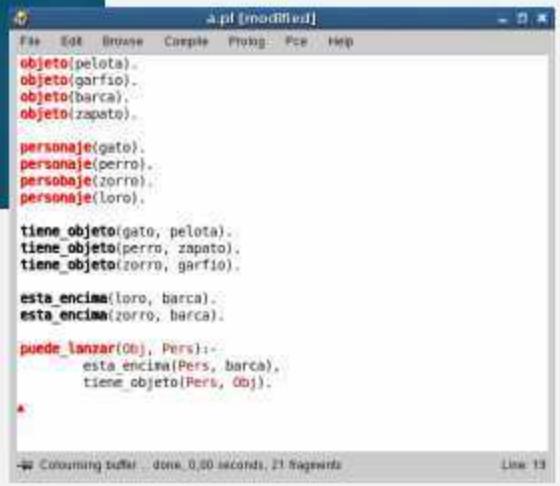
SUPORTE A LAS CLASES

http://bit.ly/Matematicas1

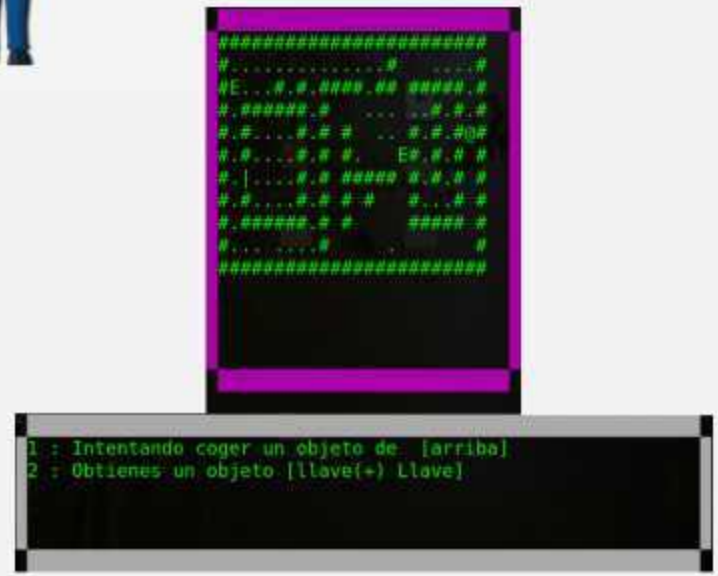
QUÉ VAMOS A USAR

Prolog





PLMAN



EVALUACIÓN

- Nota final de Matemáticas 1 es Teoría + Prática
- Teoría son 0 a 5 puntos, del examen final de teoría.
- Práctica son 0 a 5 puntos, de resolver mapas de PLMan.

- Práticas: **50%** de la nota final
- [Muy Importante] Para aprobar la práctica:
- ($P \geq 2$): Alcanzar el 40% de nota de prácticas (2 de los 5 puntos).

Se obtiene puntos por cada solución a un mapa. La suma total es P, la nota de prácticas.

Todas las prácticas son individuales.

Hay 2 tipos principales de mapas,

Presenciales: deben ser resueltos durante una sesión de prácticas.

Para casa: se dispone de unos días para resolverlos y entregarlos.

Durante el curso habrá 7 sesiones de mapas presenciales y 4 mapas para casa.

EVALUACIÓN

- Cada uno puntúa diferente según su clasificación, según 2 criterios Fase y Dificultad:
  - Fase: grupo de mapas que necesitan los mismos conocimientos para resolverse.
  - Dificultad: nivel de habilidad para resolver un mapa, con los mismos conocimientos.
- Cuanto más avanzada la fase y mayor la dificultad, más puntos

		DIFICULTAD					PUNTOS MAX	PUNTOS EXTRA
		1	2	3	4	5		
FASE	0			0,10			**0,50**	0,00
	1	0,10	0,20	0,25	0,40	0,60	**1,50**	0,05
	2	0,35	0,45	0,50	0,90	1,15	**2,75**	0,60
	3	0,60	0,90	1,00	1,70	2,15	**4,25**	2,10
	4		1,25	1,75	2,25		**1,00**	1,25
							10,00	4,00

EVALUACIÓN

Cada fase tiene un máximo de mapas que se puede resolver:

- Fase 0: máximo 5 mapas de la misma dificultad (3).
- Fases 1-3: máximo 5 mapas, sin repetir dificultad.
- Fase 4: máximo 1 mapa.

Cada fase tiene una puntuación máxima que se puede obtener (columna amarilla).

- Si en una fase se supera la puntuación máxima, el excedente se acumula como Nota Extra.
- La Nota Extra no cuenta en prácticas, pero puede sumar a la Nota Final de la asignatura (ver condiciones en la guía docente).

EVALUACIÓN

Algunos mapas a resolver se asignarán automáticamente y otros podrán ser elegidos:

- Fase 0: 5 mapas presenciales asignados automáticamente.
- Fases 1-3:
  - Casa: se asigna automáticamente de dificultad 3.
  - Presenciales: se elige dificultad en clase.
- Fase 4: se elige dificultad y se obtiene un mapa para casa.

No se puede rechazar un mapa tras ser asignado. Una vez obtenido, debe ser resuelto.

No es necesario resolver el 100% de un mapa.

La puntuación que se obtiene es proporcional al porcentaje que se haya resuelto del mapa.

Por ejemplo, en un mapa de Fase 2, Dificultad 4, podremos obtener 1,70 puntos si lo resolvemos al 100%.

Pero, si lo resolvermos al 50% obtendremos 1,70 x 50% = 0,85 puntos.

Y si lo resolvemos al 10% obtendremos 1,70 x 10% = 0,17 puntos.

EVALUACIÓN

Durante el curso habrá 7 sesiones presenciales destinadas a resolver mapas. Funcionarán así:

- Se elige una fase y dificultad y se obtiene un mapa.
- Se elaboran y entregan soluciones.
- No hay límite al entregar soluciones.
- Sólo contará la mejor solución entregada.
- Se puede realizar tantos mapas como se quiera en una misma sesión.
- Al terminar la sesión:
  - Todos los mapas presenciales quedarán bloqueados.
  - Se sumará la puntuación obtenida a Nota de Prácticas (P) y Nota Extra (E).
  - No se podrá volver a entregar más soluciones a estos mapas.

En cada sesión, los mapas disponibles para elegir dependerán de:

- Las fases que estén abiertas (se van abriendo durante el curso).
- Las dificultades que aún no se hayan realizado.
- Un retraso mínimo de 5 minutos entre elegir un mapa y poder elegir otro.

EVALUACIÓN

Las sesiones presenciales de prácticas son NO-RECUPERABLES. Si en una sesión hay 2 horas para resolver mapas y no asistes, no es posible volver atrás en el tiempo: esas 2 horas las habrás perdido.

Para evitar el problema que esto supone, sigue estos consejos:

- Presta atención a la planificación de tu grupo para saber cuando son las sesiones presenciales de resolución de mapas.
- Si una semana no puedes asistir a tu grupo, notifícalo, justifícalo y habla con profesores de otros grupos:
  - Puedes asistir provisionalmente a otro grupo para disponer del tiempo de resolución de mapas.
- Las sesiones presenciales sólo pueden realizarse en los grupos y aulas oficiales de prácticas.

QUÉ ES PROLOG?

Lenguaje de programación lógico

VERDAD o FALSO

Base de conocimiento + Interpretador

SWI Prolog

Base Conocimiento:

```
coche(ford).
coche(bmw).
```

Interpretador:

```
coche(ford).
  true.
coche(mercedes).
  false.
coche(X).
  X = ford ;
  X = bmw.
```

PLMAN

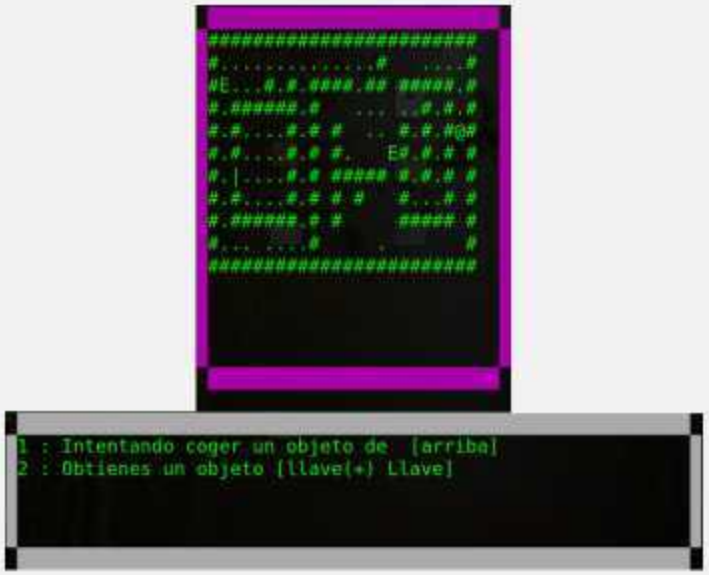
Juego de programación lógica, para el entorno SWI-Prolog

Se programa el comportamiento del protagonista, PLMan, utilizando para ello reglas lógicas de Prolog

PLMan se encuentra dentro de un mapa, del que debe salir, generalmente encontrando la llave de ese nivel y usándola para abrir la puerta.

Malo: trampas, laberintos, pistas falsas y terribles enemigos

Bueno: teleportadores, muros de cobertura y escudos



OBJETIVO

Fase 0	5 mapas en clase
Fase 1	1 mapa casa + 2 mapas clase
Fase 2	1 mapa casa + 2 mapas clase
Fase 3	1 mapa casa + 2 mapas clase
Fase 4	1 mapa casa

Mapas casa: 10 días para resolver (desde asignación)

PUNTUACION (sobre 5)
Mapas Clase: 3,75p (75%)
Mapas Casa: 1,25p (25%)

OBJETIVO: RESUMEN

- MAPAS CLASE
  - Se puede elegir dificultad
  - Al final de clase queda bloqueado (no se puede entregar más)
  - Se pueden hacer varios mapas en una sesión, pero no se puede continuar entre sesiones.
  - IMPORTANTE! No se pueden resolver los trabajos de los compañeros.
  - NO RECUPERABLE
    - Si no se asiste una semana, la pierde.
    - Si no puede en un turno, puede ir a otro esa semana (preguntar disponibilidad a los profesores)
  - Se puede superar la nota de la fase
    - Se acumula como nota extra, no como nota principal.

- MAPAS CASA
  - Se asignan automáticamente: Salvo en fase 4 (dificultad 2-4)
  - 10 días para resolverlos desde que se asignan.
  - Cada grupo tendrá su asignación y su deadline.

¿CÓMO?

Objetivo:

Aplicación para vuestro ordenador que os asignará los mapas y que os permitirá subir las soluciones.

Puede ser que en las primeras semanas no sea posible.

EL TERMINAL

Prompt
nombre_de_usuario@
nombre_de_máquina:

Linea de comandos
Aquí escribimos comandos o órdenes para que sean ejecutados. Los espacios separan la orden de los datos que le damos para que la orden pueda ser ejecutada.

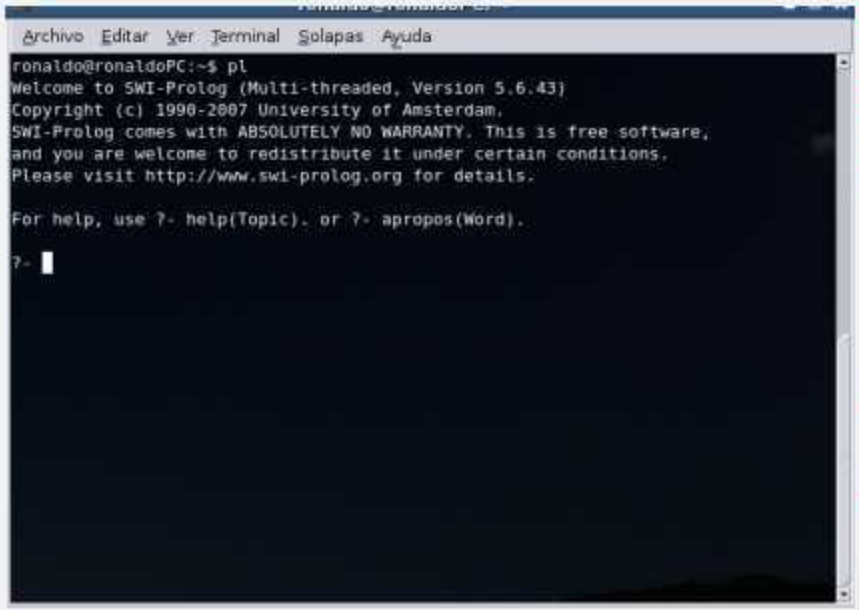
En este caso “ls” lista los contenidos de la carpeta.

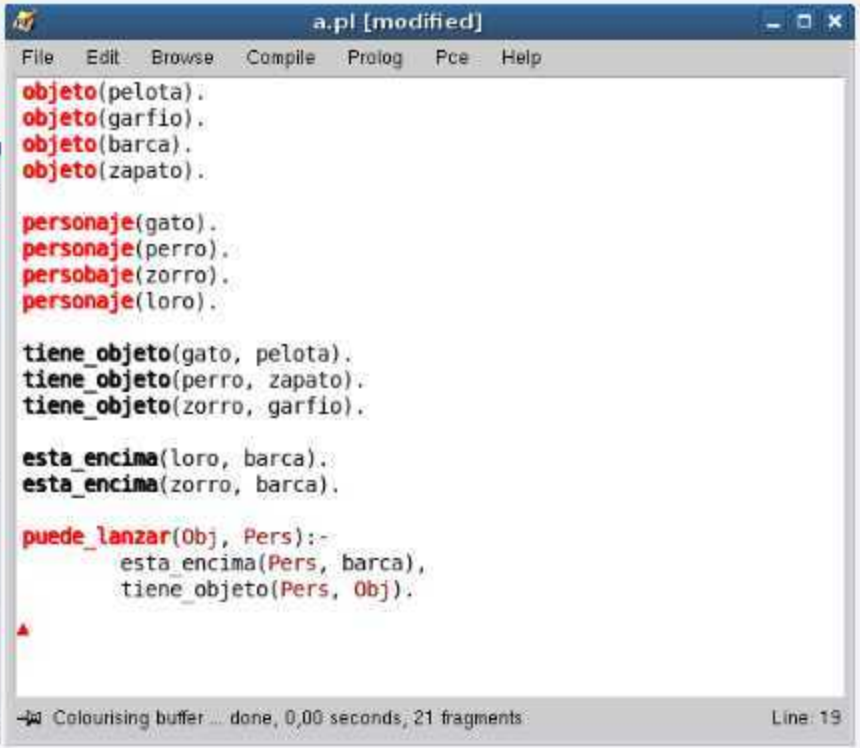
EL TERMINAL

cd	change directory (cambiar directorio)
cd Desktop	Entra en directorio “Desktop”
cd ..	Sube al directorio superior
cd /	Va al directorio raíz
cd ~	Va al directorio del usuario
ls	Lista contenidos del directorio
ls Desktop	Lista contenidos del directorio “Desktop”
swipl	Ejecuta el intérprete de swi-prolog

SWI-PROLOG

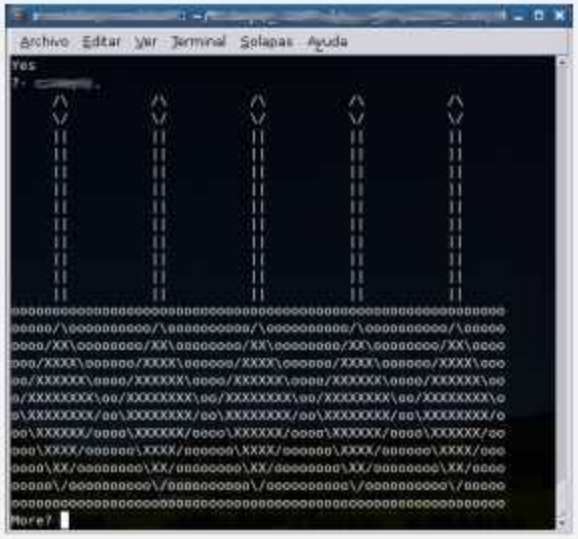
- http://www.swi-prolog.org
- Compilar/Interpretar/Ejecutar

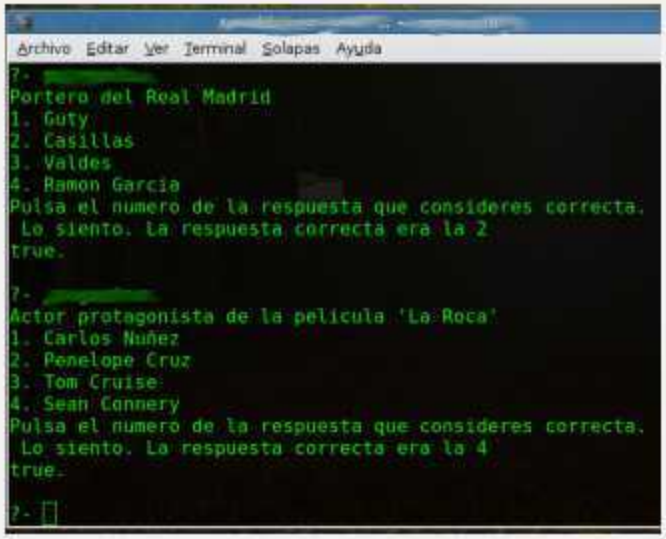




SWI-PROLOG: EJEMPLOS DE PRUEBA

- Lanzar consola SWI-Prolog
- Editar código con gedit
- Lanzar ejemplos
- Entender y modificar





SWI-PROLOG: ESTRUCTURA

Memoria

REGLAS DE INFERENCIA (Lógica)

BASE DE CONOCIMIENTO (datos)

Interfaz de usuario

```
[                    ~]$ swipl
Welcome to SWI-Prolog (Multi-threaded, 64 bits, Version 7.2.3)
Copyright (c) 1990-2015 University of Amsterdam, VU Amsterdam
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to redistribute it under certain conditions.
Please visit http://www.swi-prolog.org for details.

For help, use ?- help(Topic). or ?- apropos(Word).

?-
```

SWI-PROLOG: PROGRAMA

Existen dos tipos de cláusulas: Hechos y Reglas

```
Cabeza :- Cuerpo.
```

o

```
Resultado :- Condiciones.
```

si cuerpo entonces cabeza

cuerpo → cabeza

SWI-PROLOG: PROGRAMA

Fichero:

```
humano (mario).
persona (X) :- humano(X).
```

Interpretador:

```
humano(mario).
True.

humano(X).
X = mario.

persona(X).
X = mario.
```

SWI-PROLOG: PROGRAMA

- Comentários:

```
% Comentario
% Otro comentario
... código ....

% Más comentarios
... código ....
/* Comentario
de 2 líneas */
```

SWI-PROLOG: PROGRAMA

- ATENCIÓN:

- func (Var) ≠ func (var)

- Var → Variable
- var → Términos, sujetos

SWI-PROLOG: PROGRAMA

- Multiplas condiciones en el cuerpo:

```
padre(juan, maria).
padre(matias, juan).
```

SWI-PROLOG: PROGRAMA

- Multiplas condiciones en el cuerpo:

```
padre(juan, maria).
padre(matias, juan).

hijo (A,B) :- padre(B,A).
```

SWI-PROLOG: PROGRAMA

- Multiplas condiciones en el cuerpo:

```
padre(juan, maria).
padre(matias, juan).

hijo (A,B) :- padre(B,A).
abuelo (A,B) :- padre(A,C), padre(C,B).
```

INTÉRPRETE DE SWI-PROLOG

Ejecutar el intérprete de SWI-PROLOG

```
swipl
```

Ejecutarlo cargando una base de conocimientos

```
swipl -f base.pl
```

Es posible crear ejecutables así:

```
swipl -o ejecutable -c base.pl
```

El ejecutable creado incorpora un intérprete completo de swi-prolog, y funciona igual.

INTÉRPRETE DE SWI-PROLOG

- Atención!
- Hay diferencias entre el código y el interpretador:
  - ; → dice al interpretador que se busca más resultados
    - padre(X,Y).
    - X= juan
    - Y = maria
    - ;
    - X = matias
    - Y = juan
  - ; →en el código significa la lógica “o”

INTÉRPRETE DE SWI-PROLOG

Cargar y compilar un fichero prolog en el intérprete

```
consult('fichero.pl').
```

Ejecutar una pregunta en SWI-PROLOG

```
objetivo.
```

Salir del intérprete

```
halt.
```

FUENTES

Apuntes de Prolog y material (castellano)
Campus virtual /Materiales
http://www.dccia.ua.es/logica/prolog/docs/prolog.pdf
http://www.dccia.ua.es/logica/prolog/material.htm

Adventure in prolog (inglés)
http://www.amzi.com/AdventureInProlog/advfrtop.htm

LIBROS
Buscar en Biblioteca y Archivo, web UA http://www.ua.es
The Art of Prolog
http://gaudi.ua.es/uhtbin/cgisirsi/AkB95t6saS/0/253110069/9
Programación en Prolog
http://gaudi.ua.es/uhtbin/cgisirsi/LO4Ho5QWQ5/0/253110069/9

EJERCICIOS

Volviendo al ejemplo de familia:

1. Añadir hechos relativos a primos:
    1. Maria es prima de Luis
    2. Jose es primo de Maria
    3. Bea es prima de Luis
2. Crear regla primo(A,B) tal que primo(maria, luis) sea igual a primo(luis,maria)
3. Verificar cuantos primos tiene maria
4. Crear la regla tio(A,B) tal que tio(juan, luis) sea verdad
5. Verificar cuantos sobrinos juan es tío.