

Asignatura: **Fundamentos de Inteligencia Artificial.**

Prueba de Evaluación Continua 2: **Construcción en Prolog de un sistema basado en reglas.**

rsanchez628@alumno.uned.es

1.- Descripción del conocimiento del dominio.

El tema sobre el que se desarrolla la actividad es: *Dragones en el núcleo solar*.

La población de dragones que habita el núcleo solar es estable en ese ecosistema. El simulador permite al usuario ver la evolución que tendrían las poblaciones de dragones en distintas estrellas.

El usuario puede seleccionar una estrella e ir analizando ciclo a ciclo como la población crece o, llegado el caso, se extingue memorizando el ciclo en que se produjo la extinción.

El objetivo del simulador es demostrar al usuario como pequeñas diferencias en las variables que definen el ecosistema pueden ser causa de grandes perturbaciones en el largo plazo.

Como documentación se han utilizado simuladores más complejos de ecosistemas terrestres.

2.- Metodología de desarrollo.

La **elección de tema** se basó en la voluntad de recrear un mundo virtual, a partir de ese deseo sólo hubo que acotar marco y protagonistas.

El **aprendizaje de Prolog** se hizo a través del material suministrado el Equipo Docente, tutorías y tutoriales de internet.

Para la **declaración de hechos y reglas** en primer lugar se describió en lenguaje natural cómo sería la actuación del usuario con el simulador y qué información del mundo virtual se le suministraría.

Una vez decidido que el usuario elegiría una estrella distinta al sol donde simular cómo evolucionaría la población de las distintas poblaciones de dragones que pueblan su núcleo, se establecen las reglas que rigen el comportamiento de las especies de los dragones y el entorno que proporcionan las estrellas. Al definir las estrellas se procede a la **creación de distintos escenarios**.

El usuario elige la estrella en la que realizar la simulación y el simulador ofrece la población de las especies ciclo a ciclo. De esta forma, las reglas biológicas que dirigen el comportamiento de los dragones en un ecosistema son las reglas que aplica el simulador.

Las **pruebas de aplicación** han consistido en ofrecer diferentes entornos para que las especies de dragones tengan distintos comportamientos en distintas situaciones. Así, habrá estrellas en las que se extingan en poco tiempo, otras en las que se produzca la extinción tras un periodo de crecimiento y por último estrellas en las que el crecimiento tienda a la superpoblación.

Por último, el entorno de desarrollo para la realización del simulador ha sido:

- Distribución SWI – Prolog.
- Editor Visual Studio Code

3.- Descripción de la estructura de la base de reglas.

La base de conocimiento está dividida en varios hechos y reglas agrupados según su función en el simulador.

Siguiendo el orden en que aparecen en el código:

1. Contantes.
2. Hechos sin variable.
3. Predicados dinámicos.
4. Reglas de uso general.
 1. Inicio de la simulación.
 2. Reloj.
 3. Informar al usuario.
 4. Extinción de especies.
5. Reglas para la alimentación de dragones.
6. Reglas para la reproducción de dragones.
7. Bucles – Recursividad.
8. Ejecución del programa.
9. Auxiliares a la ejecución. Impresión de menú.

4.- Descripción de la estructura de la base de reglas.

1. constantes.

Especies de dragones:

- Dragones Rojos.
- Dragones Amarillos.

Estrellas en las que realizar la simulación:

- Estrella 1.
- Estrella 2.
- Estrella 3.

2. Hechos sin variable.

- Fertilidad de cada especie de dragón. Cuánto se reproduce cada especie.
- Frecuencia de cada especie de dragón. Con qué frecuencia se reproducen las especies.
- Población inicial de dragones en cada estrella.
- Descripción de cada estrella. Datos que definen a las estrellas.

3. Predicados dinámicos.

- Inicio. Auxiliar.
- Temperatura. Auxiliar.
- Contador. Almacena los datos de la simulación que se le mostrarán al usuario.
 - Población de dragones rojos.
 - Población de dragones amarillos.
 - Número de ciclos consumidos en la simulación.
 - Ciclo de extinción de la especie de dragones rojos si se produce.
 - Ciclo de extinción de la especie de dragones amarillos si se produce.

4. Reglas de uso general.

- Estado0. Objetivo. Establece el procedimiento para inicializar el contador.

- Contar ciclo. Objetivo. Actualiza el contador.
- Informar al usuario.
 - Regla. Estado. Da la información al usuario sobre el estado de la simulación.
 - Objetivo. Proporcionar información por consola.
 - Estado pleno. Formato para informar sin poblaciones de dragones extintas.
 - Estado extinción total. Formato para informar con todos los dragones extintos.
 - Estado extinción roja. Formato para informar de la extinción de la especie de dragones rojos.
 - Estado extinción amarilla. Formato para informar de la extinción de la especie de dragones amarillos.
- Extinción de especies.
 - Extinción total. Si se cumple, las especies se han extinguido.
 - Objetivo. Actualiza el contador con el ciclo de extinción de las especies.
 - Extinción rojo. Si se cumple, la especie roja se ha extinguido.
 - Objetivo. Actualiza el contador con el ciclo de extinción de la especie.
 - Extinción total. Si se cumple, la especie amarilla se ha extinguido.
 - Objetivo. Actualiza el contador con el ciclo de extinción de la especie.

5. Reglas para alimentación de los dragones.

Objetivo. Descontar dragones de la población de cada especie.

Los dragones rojos se alimentan de la energía solar y minerales en su entorno de forma automática, como las plantas terrestres.

Los dragones amarillos se alimentan de dragones rojos. En ausencia de dragones rojos, practican el canibalismo hasta su extinción.

- Reglas
 - Comer. Se come si hay comida, en caso contrario se practica el canibalismo o los dragones se extinguieron.
 - Amarillo come amarillo. Regla que determina según la población de dragones la cantidad de dragones que comen.
 - Amarillo come rojo. Regla que determina según la cantidad de dragones amarillos, cuántos dragones rojos son comidos.
 - Comer (_, _ dragon). En función de la extinción de alguna de las especies, determina que especie de dragones será el alimento de los dragones amarillos.
- Objetivo
 - Descontar (_, Dragón). Descontará de la población la especie de dragones que se le indique.

6. Reglas para la reproducción de los dragones.

Objetivo. Aumentar la población de cada especie de dragones según la fertilidad y frecuencia reproductora de la especie.

La fertilidad de la especie nos dará el número de dragones por camada, la frecuencia nos dice con qué frecuencia aparecerán las camadas.

- Reglas
 - Condición reproducción. Si se cumple, en ese ciclo habrá nueva camada.
 - Condición reproducción naranja. Si se cumple, se reproducen todas las especies de dragones.
 - Condición reproducción roja. Si se cumple, se reproducen los dragones rojos.
 - Condición reproducción amarilla. Si se cumple, se reproducen los dragones amarillos.
- Objetivo
 - Reproducir (Dragón). Aumenta la población de dragones en el número que indica la fertilidad de la especie.

7. Bucles – Recursividad.

Reglas que usan la recursividad para repetir acciones el número de veces indicado.

- Reglas.
 - Tiempo.
 - Tic. Regla auxiliar de tiempo.
 - Comer AR cantidad.
 - Comer AA cantidad.

8. Ejecución del programa.

Regla que rige el comportamiento del simulador. Desencadena el resto de reglas hasta cumplir los objetivos.

Para ejecutar el programa el usuario debe:

1. Cargar el fichero con el código fuente en el intérprete de Prolog.
 2. Escribir el comando `main` seguido de un punto (.)
 3. Seguir las instrucciones que el programa ofrezca por pantalla.
- `Main`. Comportamiento del simulador.
 - `Set inicio`. Desencadena la simulación según la opción seleccionada por el usuario. Auxiliar
 - `Continuar`. Facilita la lectura de los mensajes por pantalla y el uso del simulador por parte del usuario. Auxiliar.
 - `Ciclo5`. Simula 5 ciclos más o termina la simulación según indicación del usuario. Auxiliar.
 - `Clear`. Limpiar la consola facilitando la comunicación con el usuario. Auxiliar.

9. Auxiliares a la ejecución. Impresión del menú.

- Mensaje presentación.
- Mensaje opciones.
 - Chequeo opción. Evalúa la validez de la opción introducida por el usuario. Auxiliar a mensaje opciones.

5.- Casos de prueba.

- Opción 1.

Frecuencia 3				Estado 0 10 1							
Frecuencia 5											
Ciclo		Rojos	Amarillos	Ciclo		Rojos	Amarillos	Ciclo		Rojos	Amarillos
1	comer	9	1	13	comer	32	4	25	comer	0	4
	reproducirse	9	1		reproducirse	32	4		reproducirse	0	8
	total	9	1		total	32	4		total	0	8
2	comer	8	1	14	comer	28	4	26	comer	0	4
	reproducirse	8	1		reproducirse	28	4		reproducirse	0	4
	total	8	1		total	28	4		total	0	4
3	comer	7	1	15	comer	24	4	27	comer	0	2
	reproducirse	14	1		reproducirse	48	8		reproducirse	0	2
	total	14	1		total	48	8		total	0	2
4	comer	13	1	16	comer	40	8	28	comer	0	1
	reproducirse	13	1		reproducirse	40	8		reproducirse	0	1
	total	13	1		total	40	8		total	0	1
5	comer	12	1	17	comer	32	8	29	comer	0	0
	reproducirse	12	2		reproducirse	32	8		reproducirse	0	0
	total	12	2		total	32	8		total	0	0
6	comer	10	2	18	comer	24	8				
	reproducirse	20	2		reproducirse	48	8				
	total	20	2		total	48	8				
7	comer	18	2	19	comer	40	8				
	reproducirse	18	2		reproducirse	40	8				
	total	18	2		total	40	8				
8	comer	16	2	20	comer	32	8				
	reproducirse	16	2		reproducirse	32	16				
	total	16	2		total	32	16				
9	comer	14	2	21	comer	16	16				
	reproducirse	28	2		reproducirse	32	16				
	total	28	2		total	32	16				
10	comer	26	2	22	comer	16	16				
	reproducirse	26	4		reproducirse	16	16				
	total	26	4		total	16	16				
11	comer	22	4	23	comer	0	16				
	reproducirse	22	4		reproducirse	0	16				
	total	22	4		total	0	16				
12	comer	18	4	24	comer	0	8				
	reproducirse	36	4		reproducirse	0	8				
	total	36	4		total	0	8				

```
SWI-Prolog (AMD64, Multi-threaded, version 8.4.2)
File Edit Settings Run Debug Help
Seleccione la opcion: 1

Si desea simular 5 ciclos del tiempo solar,
Introduzca '5.' y pulse intro.
Si introduce otro numero, la simulacion finaliza.
|: 5.

Dragones ROJOS: 10
Dragones AMARILLOS: 1
Numero de CICLOS: 0

Dragones ROJOS: 9
Dragones AMARILLOS: 1
Numero de CICLOS: 1

Dragones ROJOS: 8
Dragones AMARILLOS: 1
Numero de CICLOS: 2

Dragones ROJOS: 7
Dragones AMARILLOS: 1
Numero de CICLOS: 3

Dragones ROJOS: 13
Dragones AMARILLOS: 1
Numero de CICLOS: 4

Dragones ROJOS: 13
Dragones AMARILLOS: 1
Numero de CICLOS: 5

Si desea simular 5 ciclos del tiempo solar,
Introduzca '5.' y pulse intro.
Si introduce otro numero, la simulacion finaliza.
|: 5.

Dragones ROJOS: 12
Dragones AMARILLOS: 1
Numero de CICLOS: 5

Dragones ROJOS: 10
Dragones AMARILLOS: 2
Numero de CICLOS: 6

Dragones ROJOS: 18
Dragones AMARILLOS: 2
Numero de CICLOS: 7

Dragones ROJOS: 16
Dragones AMARILLOS: 2
Numero de CICLOS: 8

Dragones ROJOS: 14
Dragones AMARILLOS: 2
Numero de CICLOS: 9

Dragones ROJOS: 28
Dragones AMARILLOS: 2
Numero de CICLOS: 10

Si desea simular 5 ciclos del tiempo solar,
Introduzca '5.' y pulse intro.
Si introduce otro numero, la simulacion finaliza.
|:
```

```
SWI-Prolog (AMD64, Multi-threaded, version 8.4.2)
File Edit Settings Run Debug Help

Dragones ROJOS: 16
Dragones AMARILLOS: 2
Numero de CICLOS: 8

Dragones ROJOS: 14
Dragones AMARILLOS: 2
Numero de CICLOS: 9

Dragones ROJOS: 28
Dragones AMARILLOS: 2
Numero de CICLOS: 10

Si desea simular 5 ciclos del tiempo solar,
Introduzca '5.' y pulse intro.
Si introduce otro numero, la simulacion finaliza.
|: 5.

Dragones ROJOS: 26
Dragones AMARILLOS: 2
Numero de CICLOS: 10

Dragones ROJOS: 22
Dragones AMARILLOS: 4
Numero de CICLOS: 11

Dragones ROJOS: 18
Dragones AMARILLOS: 4
Numero de CICLOS: 12

Dragones ROJOS: 32
Dragones AMARILLOS: 4
Numero de CICLOS: 13

Dragones ROJOS: 28
Dragones AMARILLOS: 4
Numero de CICLOS: 14

Dragones ROJOS: 28
Dragones AMARILLOS: 4
Numero de CICLOS: 15

Si desea simular 5 ciclos del tiempo solar,
Introduzca '5.' y pulse intro.
Si introduce otro numero, la simulacion finaliza.
|: 5.

Dragones ROJOS: 24
Dragones AMARILLOS: 4
Numero de CICLOS: 15

Dragones ROJOS: 40
Dragones AMARILLOS: 8
Numero de CICLOS: 16

Dragones ROJOS: 32
Dragones AMARILLOS: 8
Numero de CICLOS: 17

Dragones ROJOS: 24
Dragones AMARILLOS: 8
Numero de CICLOS: 18

Dragones ROJOS: 40
Dragones AMARILLOS: 8
Numero de CICLOS: 19

Dragones ROJOS: 40
Dragones AMARILLOS: 8
Numero de CICLOS: 20

Si desea simular 5 ciclos del tiempo solar,
Introduzca '5.' y pulse intro.
Si introduce otro numero, la simulacion finaliza.
|: █
```



```
SWI-Prolog (AMD64, Multi-threaded, version 8.4.2)
File Edit Settings Run Debug Help

Dragones ROJOS: 24
Dragones AMARILLOS: 8
Numero de CICLOS: 18

Dragones ROJOS: 40
Dragones AMARILLOS: 8
Numero de CICLOS: 19

Dragones ROJOS: 40
Dragones AMARILLOS: 8
Numero de CICLOS: 20

Si desea simular 5 ciclos del tiempo solar,
Introduzca '5.' y pulse intro.
Si introduce otro numero, la simulacion finaliza.
|: 5.

Dragones ROJOS: 32
Dragones AMARILLOS: 8
Numero de CICLOS: 20

Dragones ROJOS: 16
Dragones AMARILLOS: 16
Numero de CICLOS: 21

Dragones ROJOS: 16
Dragones AMARILLOS: 26
Numero de CICLOS: 22

Dragones ROJOS EXTINTOS en ciclo: 23
Dragones AMARILLOS: 16
Numero de CICLOS: 23

Dragones ROJOS EXTINTOS en ciclo: 23
Dragones AMARILLOS: 8
Numero de CICLOS: 24

Dragones ROJOS EXTINTOS en ciclo: 23
Dragones AMARILLOS: 8
Numero de CICLOS: 25

Si desea simular 5 ciclos del tiempo solar,
Introduzca '5.' y pulse intro.
Si introduce otro numero, la simulacion finaliza.
|: 5.

Dragones ROJOS EXTINTOS en ciclo: 23
Dragones AMARILLOS: 4
Numero de CICLOS: 25

Dragones ROJOS EXTINTOS en ciclo: 23
Dragones AMARILLOS: 4
Numero de CICLOS: 26

Dragones ROJOS EXTINTOS en ciclo: 23
Dragones AMARILLOS: 2
Numero de CICLOS: 27

Dragones ROJOS EXTINTOS en ciclo: 23
Dragones AMARILLOS: 1
Numero de CICLOS: 28

Dragones ROJOS EXTINTOS en ciclo: 23
Dragones AMARILLOS EXTINTOS en ciclo: 29
Numero de CICLOS: 29

Dragones ROJOS EXTINTOS en ciclo: 23
Dragones AMARILLOS EXTINTOS en ciclo: 29
Numero de CICLOS: 30

Si desea simular 5 ciclos del tiempo solar,
Introduzca '5.' y pulse intro.
Si introduce otro numero, la simulacion finaliza.
|: ■
```

- Opción 2

Frecuencia	3			Estado 0	4	1	
Frecuencia	5						
Ciclo		Rojos	Amarillos	Ciclo		Rojos	Amarillos
1	comer	3	1	13	comer	0	0
	reproducirse	3	1		reproducirse	0	0
	total	3	1		total	0	0
2	comer	2	1	14	comer	0	0
	reproducirse	2	1		reproducirse	0	0
	total	2	1		total	0	0
3	comer	1	1	15	comer	0	0
	reproducirse	2	1		reproducirse	0	0
	total	2	1		total	0	0
4	comer	1	1	16	comer	0	0
	reproducirse	1	1		reproducirse	0	0
	total	1	1		total	0	0
5	comer	0	1	17	comer	0	0
	reproducirse	0	2		reproducirse	0	0
	total	0	2		total	0	0
6	comer	0	1	18	comer	0	0
	reproducirse	0	1		reproducirse	0	0
	total	0	1		total	0	0
7	comer	0	0	19	comer	0	0
	reproducirse	0	0		reproducirse	0	0
	total	0	0		total	0	0

```
SWI-Prolog (AMD64, Multi-threaded, version 8.4.2)
File Edit Settings Run Debug Help
Seleccione la opcion: 2

Si desea simular 5 ciclos del tiempo solar,
Introduzca '5.' y pulse intro.
Si introduce otro numero, la simulacion finaliza.
|: 5.

Dragones ROJOS: 4
Dragones AMARILLOS: 1
Numero de CICLOS: 0

Dragones ROJOS: 3
Dragones AMARILLOS: 1
Numero de CICLOS: 1

Dragones ROJOS: 2
Dragones AMARILLOS: 1
Numero de CICLOS: 2

Dragones ROJOS: 1
Dragones AMARILLOS: 1
Numero de CICLOS: 3

Dragones ROJOS: 1
Dragones AMARILLOS: 1
Numero de CICLOS: 4

Dragones ROJOS: 1
Dragones AMARILLOS: 1
Numero de CICLOS: 5

Si desea simular 5 ciclos del tiempo solar,
Introduzca '5.' y pulse intro.
Si introduce otro numero, la simulacion finaliza.
|: 5.

Dragones ROJOS EXTINTOS en ciclo: 5
Dragones AMARILLOS: 1
Numero de CICLOS: 5

Dragones ROJOS EXTINTOS en ciclo: 5
Dragones AMARILLOS: 1
Numero de CICLOS: 6

Dragones ROJOS EXTINTOS en ciclo: 5
Dragones AMARILLOS EXTINTOS en ciclo: 7
Numero de CICLOS: 7

Dragones ROJOS EXTINTOS en ciclo: 5
Dragones AMARILLOS EXTINTOS en ciclo: 7
Numero de CICLOS: 8

Dragones ROJOS EXTINTOS en ciclo: 5
Dragones AMARILLOS EXTINTOS en ciclo: 7
Numero de CICLOS: 9

Dragones ROJOS EXTINTOS en ciclo: 5
Dragones AMARILLOS EXTINTOS en ciclo: 7
Numero de CICLOS: 10

Si desea simular 5 ciclos del tiempo solar,
Introduzca '5.' y pulse intro.
Si introduce otro numero, la simulacion finaliza.
|:
```

- Opción 3

Frecuencia	3			Estado 0	14	1					
Frecuencia	5										
Ciclo		Rojos	Amarillos	Ciclo		Rojos	Amarillos	Ciclo		Rojos	Amarillos
1	comer	13	1	13	comer	96	4	25	comer	976	16
	reproducirse	13	1		reproducirse	96	4		reproducirse	976	32
	total	13	1		total	96	4		total	976	32
2	comer	12	1	14	comer	92	4	26	comer	944	32
	reproducirse	12	1		reproducirse	92	4		reproducirse	944	32
	total	12	1		total	92	4		total	944	32
3	comer	11	1	15	comer	88	4	27	comer	912	32
	reproducirse	22	1		reproducirse	176	8		reproducirse	1824	32
	total	22	1		total	176	8		total	1824	32
4	comer	21	1	16	comer	168	8	28	comer	1792	32
	reproducirse	21	1		reproducirse	168	8		reproducirse	1792	32
	total	21	1		total	168	8		total	1792	32
5	comer	20	1	17	comer	160	8	29	comer	1760	32
	reproducirse	20	2		reproducirse	160	8		reproducirse	1760	32
	total	20	2		total	160	8		total	1760	32
6	comer	18	2	18	comer	152	8	30	comer	1728	32
	reproducirse	36	2		reproducirse	304	8		reproducirse	3456	64
	total	36	2		total	304	8		total	3456	64
7	comer	34	2	19	comer	296	8	31	comer	3392	64
	reproducirse	34	2		reproducirse	296	8		reproducirse	3392	64
	total	34	2		total	296	8		total	3392	64
8	comer	32	2	20	comer	288	8	32	comer	3328	64
	reproducirse	32	2		reproducirse	288	16		reproducirse	3328	64
	total	32	2		total	288	16		total	3328	64
9	comer	30	2	21	comer	272	16	33	comer	3264	64
	reproducirse	60	2		reproducirse	544	16		reproducirse	6528	64
	total	60	2		total	544	16		total	6528	64
10	comer	58	2	22	comer	528	16	34	comer	6464	64
	reproducirse	58	4		reproducirse	528	16		reproducirse	6464	64
	total	58	4		total	528	16		total	6464	64
11	comer	54	4	23	comer	512	16	35	comer	6400	64
	reproducirse	54	4		reproducirse	512	16		reproducirse	6400	128
	total	54	4		total	512	16		total	6400	128
12	comer	50	4	24	comer	496	16	36	comer	6272	128
	reproducirse	100	4		reproducirse	992	16		reproducirse	12544	128
	total	100	4		total	992	16		total	12544	128

```
SWI-Prolog (AMD64, Multi-threaded, version 8.4.2)
File Edit Settings Run Debug Help
Seleccione la opcion: 3

Si desea simular 5 ciclos del tiempo solar,
Introduzca '5.' y pulse intro.
Si introduce otro numero, la simulacion finaliza.
|: 5.

Dragones ROJOS: 14
Dragones AMARILLOS: 1
Numero de CICLOS: 0

Dragones ROJOS: 13
Dragones AMARILLOS: 1
Numero de CICLOS: 1

Dragones ROJOS: 12
Dragones AMARILLOS: 1
Numero de CICLOS: 2

Dragones ROJOS: 11
Dragones AMARILLOS: 1
Numero de CICLOS: 3

Dragones ROJOS: 21
Dragones AMARILLOS: 1
Numero de CICLOS: 4

Dragones ROJOS: 21
Dragones AMARILLOS: 1
Numero de CICLOS: 5

Si desea simular 5 ciclos del tiempo solar,
Introduzca '5.' y pulse intro.
Si introduce otro numero, la simulacion finaliza.
|: 5.

Dragones ROJOS: 20
Dragones AMARILLOS: 1
Numero de CICLOS: 5

Dragones ROJOS: 18
Dragones AMARILLOS: 2
Numero de CICLOS: 6

Dragones ROJOS: 34
Dragones AMARILLOS: 2
Numero de CICLOS: 7

Dragones ROJOS: 32
Dragones AMARILLOS: 2
Numero de CICLOS: 8

Dragones ROJOS: 30
Dragones AMARILLOS: 2
Numero de CICLOS: 9

Dragones ROJOS: 60
Dragones AMARILLOS: 2
Numero de CICLOS: 10

Si desea simular 5 ciclos del tiempo solar,
Introduzca '5.' y pulse intro.
Si introduce otro numero, la simulacion finaliza.
|: █
```

```
SWI-Prolog (AMD64, Multi-threaded, version 8.4.2)
File Edit Settings Run Debug Help

Dragones ROJOS: 32
Dragones AMARILLOS: 2
Numero de CICLOS: 8

Dragones ROJOS: 30
Dragones AMARILLOS: 2
Numero de CICLOS: 9

Dragones ROJOS: 60
Dragones AMARILLOS: 2
Numero de CICLOS: 10

Si desea simular 5 ciclos del tiempo solar,
Introduzca '5.' y pulse intro.
Si introduce otro numero, la simulacion finaliza.
|: 5.

Dragones ROJOS: 58
Dragones AMARILLOS: 2
Numero de CICLOS: 10

Dragones ROJOS: 54
Dragones AMARILLOS: 4
Numero de CICLOS: 11

Dragones ROJOS: 50
Dragones AMARILLOS: 4
Numero de CICLOS: 12

Dragones ROJOS: 96
Dragones AMARILLOS: 4
Numero de CICLOS: 13

Dragones ROJOS: 92
Dragones AMARILLOS: 4
Numero de CICLOS: 14

Dragones ROJOS: 92
Dragones AMARILLOS: 4
Numero de CICLOS: 15

Si desea simular 5 ciclos del tiempo solar,
Introduzca '5.' y pulse intro.
Si introduce otro numero, la simulacion finaliza.
|: 5.

Dragones ROJOS: 88
Dragones AMARILLOS: 4
Numero de CICLOS: 15

Dragones ROJOS: 168
Dragones AMARILLOS: 8
Numero de CICLOS: 16

Dragones ROJOS: 160
Dragones AMARILLOS: 8
Numero de CICLOS: 17

Dragones ROJOS: 152
Dragones AMARILLOS: 8
Numero de CICLOS: 18

Dragones ROJOS: 296
Dragones AMARILLOS: 8
Numero de CICLOS: 19

Dragones ROJOS: 296
Dragones AMARILLOS: 8
Numero de CICLOS: 20

Si desea simular 5 ciclos del tiempo solar,
Introduzca '5.' y pulse intro.
Si introduce otro numero, la simulacion finaliza.
|:
```

```
SWI-Prolog (AMD64, Multi-threaded, version 8.4.2)
File Edit Settings Run Debug Help

Dragones ROJOS: 152
Dragones AMARILLOS: 8
Numero de CICLOS: 18

Dragones ROJOS: 296
Dragones AMARILLOS: 8
Numero de CICLOS: 19

Dragones ROJOS: 296
Dragones AMARILLOS: 8
Numero de CICLOS: 20

Si desea simular 5 ciclos del tiempo solar,
Introduzca '5.' y pulse intro.
Si introduce otro numero, la simulacion finaliza.
|: 5.

Dragones ROJOS: 288
Dragones AMARILLOS: 8
Numero de CICLOS: 20

Dragones ROJOS: 272
Dragones AMARILLOS: 16
Numero de CICLOS: 21

Dragones ROJOS: 528
Dragones AMARILLOS: 16
Numero de CICLOS: 22

Dragones ROJOS: 512
Dragones AMARILLOS: 16
Numero de CICLOS: 23

Dragones ROJOS: 496
Dragones AMARILLOS: 16
Numero de CICLOS: 24

Dragones ROJOS: 992
Dragones AMARILLOS: 16
Numero de CICLOS: 25

Si desea simular 5 ciclos del tiempo solar,
Introduzca '5.' y pulse intro.
Si introduce otro numero, la simulacion finaliza.
|: 5.

Dragones ROJOS: 976
Dragones AMARILLOS: 16
Numero de CICLOS: 25

Dragones ROJOS: 944
Dragones AMARILLOS: 32
Numero de CICLOS: 26

Dragones ROJOS: 912
Dragones AMARILLOS: 32
Numero de CICLOS: 27

Dragones ROJOS: 1792
Dragones AMARILLOS: 32
Numero de CICLOS: 28

Dragones ROJOS: 1760
Dragones AMARILLOS: 32
Numero de CICLOS: 29

Dragones ROJOS: 1760
Dragones AMARILLOS: 32
Numero de CICLOS: 30

Si desea simular 5 ciclos del tiempo solar,
Introduzca '5.' y pulse intro.
Si introduce otro numero, la simulacion finaliza.
|: █
```

```
SWI-Prolog (AMD64, Multi-threaded, version 8.4.2)
File Edit Settings Run Debug Help

Dragones ROJOS: 512
Dragones AMARILLOS: 16
Numero de CICLOS: 23

Dragones ROJOS: 496
Dragones AMARILLOS: 16
Numero de CICLOS: 24

Dragones ROJOS: 992
Dragones AMARILLOS: 16
Numero de CICLOS: 25

Si desea simular 5 ciclos del tiempo solar,
Introduzca '5.' y pulse intro.
Si introduce otro numero, la simulacion finaliza.
|: 5.

Dragones ROJOS: 976
Dragones AMARILLOS: 16
Numero de CICLOS: 25

Dragones ROJOS: 944
Dragones AMARILLOS: 32
Numero de CICLOS: 26

Dragones ROJOS: 912
Dragones AMARILLOS: 32
Numero de CICLOS: 27

Dragones ROJOS: 1792
Dragones AMARILLOS: 32
Numero de CICLOS: 28

Dragones ROJOS: 1760
Dragones AMARILLOS: 32
Numero de CICLOS: 29

Dragones ROJOS: 1760
Dragones AMARILLOS: 32
Numero de CICLOS: 30

Si desea simular 5 ciclos del tiempo solar,
Introduzca '5.' y pulse intro.
Si introduce otro numero, la simulacion finaliza.
|: 5.

Dragones ROJOS: 1728
Dragones AMARILLOS: 32
Numero de CICLOS: 30

Dragones ROJOS: 3392
Dragones AMARILLOS: 64
Numero de CICLOS: 31

Dragones ROJOS: 3328
Dragones AMARILLOS: 64
Numero de CICLOS: 32

Dragones ROJOS: 3264
Dragones AMARILLOS: 64
Numero de CICLOS: 33

Dragones ROJOS: 6464
Dragones AMARILLOS: 64
Numero de CICLOS: 34

Dragones ROJOS: 6464
Dragones AMARILLOS: 64
Numero de CICLOS: 35

Si desea simular 5 ciclos del tiempo solar,
Introduzca '5.' y pulse intro.
Si introduce otro numero, la simulacion finaliza.
|: █
```

6.- Informe y valoración.

La primera dificultad y quizás la mayor, la que produce más incomodidad, es la falta de concreción en el enunciado de la práctica.

La elección de tema puede resultar agónica. El estudiante de ciencias no suele ser una persona creativa, sino alguien al que se le da un problema a resolver con lo que partir de una hoja en blanco resulta novedoso y produce agorafobia.

Resolviendo un problema, el estudiante siempre sabe si ha alcanzado el éxito. Con una práctica de estilo creativo nunca puede estar seguro de la calidad de su trabajo.

Sin estar seguro de la calidad de su trabajo, ni tener una forma objetiva de valorarlo y ante las innumerables ramificaciones que puede tener un proyecto de este estilo, la práctica puede terminar convirtiéndose en un sumidero de tiempo.

La siguiente dificultad consiste en habituarse a programar en un nuevo paradigma de programación cuando en las asignaturas anteriores se adquirieron automatismos propios de otros paradigmas.

Por el contrario, resulta interesante descubrir nuevos lenguajes de programación y encontrar utilidades prácticas a la lógica que se estudió en asignaturas anteriores. Ver como algo sumamente teórico termina teniendo aplicación siempre es grato y motivador para el estudiante que en ocasiones, abrumado por la teoría y el estrés de los plazos, se puede sentir haciendo esfuerzos que sólo se justifican en el aprobado cuando en realidad sí tiene utilidad en fuera del mundo académico.

ANEXO: CÓDIGO FUENTE

```

1 %Dragones - Constantes
2 dragon_rojo.
3 dragon_amarillo.
4
5 %fertilidad - Fertilidad(Temperatura, Dragon)
6 fertilidad(_, dragon_rojo, 2).
7 fertilidad(_, dragon_amarillo, 2).
8
9 %frecuencia - Frecuencia(Temperatura, Dragon)
10 frecuencia(_, dragon_rojo, 3).
11 frecuencia(_, dragon_amarillo, 5).
12
13 %gula - Gula(Temperatura)
14 gula(_, 1).
15
16 %Estrellas - Constantes
17 estrella1.
18 estrella2.
19 estrella3.
20
21 %reset - Reset(Estrella)
22 reset(estrella1, (11, 1)).
23 reset(estrella2, (5, 1)).
24 reset(estrella3, (15, 1)).
25
26 %descripcion - Descripcion(Estrella)
27 descripcion(estrella1, ("Luhman 16", "enana marron", 1210)).
28 descripcion(estrella2, ("Kraz", "Gigante luminosa", 5100)).
29 descripcion(estrella3, ("HW Virginis", "Subenana caliente", 30000)).
30
31 :- dynamic temp/1. %Predicado dinamico - Auxiliar
32 temp(_).
33
34 :- dynamic inicio/2. %Predicado dinamico - Auxiliar
35 inicio(_, _).
36
37 %Predicado dinamico - BBDD
38 %Cuenta dragones rojos, amarillos y ciclos.
39 %Ciclos de extincion de dragones rojos y amarillos.
40 :- dynamic contador/5.
41 contador(_, _, _, _, _).
42
43 %----- Reglas uso general -----
44 %Uso del contador
45 %Establece origen segun estrella - Opcion menu
46 estado0 :-
47     inicio(Rojo0, Amarillo0),
48     Rojo is Rojo0,
49     Amarillo is Amarillo0,
50     Ciclo is 0,
51     Fin_rojo is 0,
52     Fin_amarillo is 0,
53     retractall(contador(_, _, _, _, _)),
54     assert(contador(Rojo, Amarillo, Ciclo, Fin_rojo, Fin_amarillo)).
55
56 %Contar Ciclo -> Ciclo = Ciclo + 1.
57 contar_ciclo :-
58     contador(Rojo, Amarillo, Ciclo, Fin_rojo, Fin_amarillo),
59     U is Ciclo + 1,

```

```

60     retractall(contador(Rojo, Amarillo, _, Fin_rojo, Fin_amarillo)),
61     assert(contador(Rojo, Amarillo, U, Fin_rojo, Fin_amarillo)).
62
63 %----- Informacion por consola -----
64 %Informacion por consola del estado del ecosistema
65 estado:-
66     not(extincion_total), not(extincion_rojo), not(extincion_amarillo),
estado_pleno;
67     not(extincion_total), extincion_rojo, estado_extincion_roja;
68     not(extincion_total), extincion_amarillo, estado_extincion_amarilla;
69     extincion_total, estado_extincion_total.
70 %Distintos mensajes segun especie de dragon extinta
71 estado_pleno :-
72     contador(Rojo, Amarillo, Ciclo, _, _),
73     nl,
74     write('Dragones ROJOS: '), write(Rojo), nl,
75     write('Dragones AMARILLOS: '), write(Amarillo), nl,
76     write('Numero de CICLOS: '), write(Ciclo), nl.
77 estado_extincion_total :-
78     contador(_, _, Ciclo, Fin_rojo, Fin_amarillo),
79     nl,
80     write('Dragones ROJOS EXTINTOS en ciclo: '), write(Fin_rojo), nl,
81     write('Dragones AMARILLOS EXTINTOS en ciclo: '), write(Fin_amarillo), nl,
82     write('Numero de CICLOS: '), write(Ciclo), nl.
83 estado_extincion_roja :-
84     contador(_, Amarillo, Ciclo, Fin_rojo, _),
85     nl,
86     write('Dragones ROJOS EXTINTOS en ciclo: '), write(Fin_rojo), nl,
87     write('Dragones AMARILLOS: '), write(Amarillo), nl,
88     write('Numero de CICLOS: '), write(Ciclo), nl.
89 estado_extincion_amarilla :-
90     contador(Rojo, _, Ciclo, _, Fin_amarillo),
91     nl,
92     write('Dragones ROJOS: '), write(Rojo), nl,
93     write('Dragones AMARILLOS EXTINTOS en ciclo: '), write(Fin_amarillo), nl,
94     write('Numero de CICLOS: '), write(Ciclo), nl.
95 %----- Fin informacion por consola -----
96
97 %----- Condicion extincion especies -----
98 extincion_total :-
99     contador(Rojo, Amarillo, Ciclo, Fin_rojo, Fin_amarillo),
100     0 is Rojo,
101     0 is Fin_rojo,
102     0 is Amarillo,
103     0 is Fin_amarillo,
104     retractall(contador(Rojo, Amarillo, Ciclo, _, _)),
105     assert(contador(Rojo, Amarillo, Ciclo, Ciclo, Ciclo));
106     contador(Rojo, Amarillo, Ciclo, Fin_rojo, Fin_amarillo),
107     0 is Rojo,
108     0 is Fin_rojo,
109     retractall(contador(Rojo, Amarillo, Ciclo, _, Fin_amarillo)),
110     assert(contador(Rojo, Amarillo, Ciclo, Ciclo, Fin_amarillo));
111     contador(Rojo, Amarillo, Ciclo, Fin_rojo, Fin_amarillo),
112     0 is Amarillo,
113     0 is Fin_amarillo,
114     retractall(contador(Rojo, Amarillo, Ciclo, Fin_rojo, _)),
115     assert(contador(Rojo, Amarillo, Ciclo, Fin_rojo, Ciclo));
116     contador(Rojo, Amarillo, _, _, _),
117     0 is Rojo,
118     0 is Amarillo.

```

```

119 extincion_rojo :-
120     contador(Rojo, Amarillo, Ciclo, Fin_rojo, Fin_amarillo),
121     0 is Rojo,
122     0 is Fin_rojo,
123     retractall(contador(Rojo, Amarillo, Ciclo, _, Fin_amarillo)),
124     assert(contador(Rojo, Amarillo, Ciclo, Ciclo, Fin_amarillo));
125     contador(Rojo, _, _, _, _),
126     0 is Rojo.
127 extincion_amarillo :-
128     contador(Rojo, Amarillo, Ciclo, Fin_rojo, Fin_amarillo),
129     0 is Amarillo,
130     0 is Fin_amarillo,
131     retractall(contador(Rojo, Amarillo, Ciclo, Fin_rojo, _)),
132     assert(contador(Rojo, Amarillo, Ciclo, Fin_rojo, Ciclo));
133     contador(_, Amarillo, _, _, _),
134     0 is Amarillo.
135 %----- Fin Condicion extincion especies -----
136 %----- FIN Reglas uso general -----
137
138 %----- dragones comiendo-----
139 %Los dragones amarillos comen dragones rojos si los hay
140 %En ausencia de dragones rojos, son canibales
141 comer(Temperatura) :-
142     not(extincion_rojo), amarillo_come_rojo(Temperatura);
143     not(extincion_amarillo), amarillo_come_amarillo(Temperatura);
144     extincion_rojo, extincion_amarillo.
145
146 %El numero de veces que comen los dragones depende de la poblacion de
147 %la poblacion de cada especie
148 amarillo_come_amarillo(Temperatura) :-
149     contador(_, Amarillo, _, _, _),
150     1 is Amarillo,
151     comerAA_cantidad(Temperatura, 1);
152     contador(_, Amarillo, _, _, _),
153     Y is Amarillo // 2,
154     comerAA_cantidad(Temperatura, Y).
155
156 amarillo_come_rojo(Temperatura) :-
157     contador(_, Amarillo, _, _, _),
158     comerAR_cantidad(Temperatura, Amarillo).
159
160 %Los dragones comidos se descuentan de las poblaciones
161 comer(Temperatura, dragon_amarillo, Dragon) :-
162     Dragon==dragon_rojo →
163     (
164         extincion_rojo;
165         descontar(Temperatura, Dragon)
166     );
167     extincion_amarillo;
168     descontar(Temperatura, Dragon).
169
170 %descontar(Temperatura, Dragon) → Dragon = Dragon - 1.
171 descontar(Temperatura, Dragon) :-
172     Dragon==dragon_amarillo →
173     (
174         contador(Rojo, Amarillo, Ciclo, Fin_rojo, Fin_amarillo),
175         gula(Temperatura, Gula),
176         U is Amarillo - Gula,
177         retractall(contador(Rojo, _, Ciclo, Fin_rojo, Fin_amarillo)),
178         assert(contador(Rojo, U, Ciclo, Fin_rojo, Fin_amarillo))

```

```

179 );
180 contador(Rojo, Amarillo, Ciclo, Fin_rojo, Fin_amarillo),
181 gula(Temperatura, Gula),
182 U is Rojo - Gula,
183 retractall(contador(_, Amarillo, Ciclo, Fin_rojo, Fin_amarillo)),
184 assert(contador(U, Amarillo, Ciclo, Fin_rojo, Fin_amarillo)).
185 %----- FIN dragones comiendo-----
186
187 %-----Los dragones se reproducen-----
188 %Reproduccion - Reproduccion(Temperatura, Dragon)
189 reproducir(Temperatura, Dragon):-
190     Dragon==dragon_amarillo ->
191     (
192         contador(Rojo, Amarillo, Ciclo, Fin_rojo, Fin_amarillo),
193         fertilidad(Temperatura, Dragon, Fertilidad),
194         U is Amarillo * Fertilidad,
195         retractall(contador(Rojo, _, Ciclo, Fin_rojo, Fin_amarillo)),
196         assert(contador(Rojo, U, Ciclo, Fin_rojo, Fin_amarillo))
197     );
198     contador(Rojo, Amarillo, Ciclo, Fin_rojo, Fin_amarillo),
199     fertilidad(Temperatura, Dragon, Fertilidad),
200     V is Rojo * Fertilidad,
201     retractall(contador(_, Amarillo, Ciclo, Fin_rojo, Fin_amarillo)),
202     assert(contador(V, Amarillo, Ciclo, Fin_rojo, Fin_amarillo)).
203
204 %Analisis de las condiciones de reproduccion
205 %Cada especie se reproduce con distinta frecuencia
206 condicion_reproduccion(Temperatura):-
207     condicion_naranja(Temperatura);
208     condicion_roja(Temperatura);
209     condicion_amarillo(Temperatura).
210
211 condicion_naranja(Temperatura) :-
212     contador(_, _, Ciclo, _, _),
213     frecuencia(Temperatura, dragon_rojo, Frecuencia_roja),
214     frecuencia(Temperatura, dragon_amarillo, Frecuencia_amarilla),
215     Frecuencia_naranja is Frecuencia_roja * Frecuencia_amarilla,
216     not(0 is Ciclo),
217     0 is mod(Ciclo, Frecuencia_naranja),
218     reproducir(Temperatura, dragon_rojo),
219     reproducir(Temperatura, dragon_amarillo),
220     contar_ciclo.
221
222 condicion_roja(Temperatura) :-
223     contador(_, _, Ciclo, _, _),
224     frecuencia(Temperatura, dragon_rojo, Frecuencia_roja),
225     not(0 is Ciclo),
226     0 is mod(Ciclo, Frecuencia_roja),
227     reproducir(Temperatura, dragon_rojo),
228     contar_ciclo.
229
230 condicion_amarillo(Temperatura) :-
231     contador(_, _, Ciclo, _, _),
232     frecuencia(Temperatura, dragon_amarillo, Frecuencia_amarilla),
233     not(0 is Ciclo),
234     0 is mod(Ciclo, Frecuencia_amarilla),
235     reproducir(Temperatura, dragon_amarillo),
236     contar_ciclo.
237 %-----FIN Los dragones se reproducen-----
238

```

```

239 %Bucles - Recursividad - simula el paso del tiempo
240 tiempo(_, 0):- estado.
241 tiempo(Temperatura, X) :-
242     comer(Temperatura),
243     tic(Temperatura),
244     Y is X - 1,
245     tiempo(Temperatura, Y).
246
247 %Auxiliar
248 tic(Temperatura) :-
249     estado,
250     condicion_reproduccion(Temperatura);
251     contar_ciclo.
252
253 %Bucles - Recursividad - Cantidad comida por especie / Ciclo
254 comerAR_cantidad(_, 0).
255 comerAR_cantidad(Temperatura, X) :-
256     comer(Temperatura, dragon_amarillo, dragon_rojo),
257     Y is X - 1,
258     comerAR_cantidad(Temperatura, Y).
259
260 comerAA_cantidad(_, 0).
261 comerAA_cantidad(Temperatura, X) :-
262     comer(Temperatura, dragon_amarillo, dragon_amarillo),
263     Y is X - 1,
264     comerAA_cantidad(Temperatura, Y).
265
266 %----- Main -----
267 %Comportamiento del simulador
268 main:-
269     clear,
270     mensaje_presentacion,
271     continuar,
272     mensaje_opciones,
273     estado0,
274     temp(Temp),
275     Temperatura is Temp,
276     ciclo5(Temperatura).
277
278 % Auxiliares main
279 % Desencadena la simulacion segun la opcion elegida por el usuario
280 set_inicio(Opcion) :-
281     Opcion==1 →
282     (
283         inicio(_, _),
284         reset(estrella1, (R1, A1)),
285         retractall(inicio(_, _)),
286         assert(inicio(R1, A1)),
287         temp(_),
288         descripcion(estrella1, (_, _, Temp1)),
289         retractall(temp(_)),
290         assert(temp(Temp1))
291     );
292     Opcion==2 →
293     (
294         inicio(_, _),
295         reset(estrella2, (R2, A2)),
296         retractall(inicio(_, _)),
297         assert(inicio(R2, A2)),
298         temp(_),

```

```

299     descripcion(estrella2, (_, _, Temp2)),
300     retractall(temp(_)),
301     assert(temp(Temp2))
302 );
303 Opcion==3 →
304 (
305     inicio(_, _),
306     reset(estrella3, (R3, A3)),
307     retractall(inicio(_, _)),
308     assert(inicio(R3, A3)),
309     temp(_),
310     descripcion(estrella3, (_, _, Temp3)),
311     retractall(temp(_)),
312     assert(temp(Temp3))
313 ).
314
315 % Facilita la lectura de los mensajes por consola- Auxiliar
316 continuar :-
317     write("Escriba 'C.' y pulse intro para continuar: "),
318     read(X),
319     write(X).
320
321 % Solicita al usuario si simular 5 ciclos más o terminar
322 % la simulaciom - Auxiliar
323 ciclo5(Temperatura) :-
324     nl,nl,
325     write("Si desea simular 5 ciclos del tiempo solar,"),nl,
326     write("Introduzca '5.' y pulse intro,"), nl,
327     write("Si introduce otro numero, la simulacion finaliza."), nl,
328     read(X),
329     (
330         X==5 → tiempo(Temperatura, 5), ciclo5(Temperatura);
331         nl, nl, write("** LA SIMULACION HA TERMINADO **"), nl, nl
332     )
333 .
334
335 % Facilita la lectura de los mensajes por consola- Auxiliar
336 %Limpia consola
337 clear :- write('\e[2J').
338
339 %----- Menu. Mensajes. -----
340 mensaje_presentacion :-
341     write("
342     ====="), nl,
343     write(" ** SIMULADOR DEL ECOSISTEMA SOLAR EN DISTINTOS TIPOS DE ESTRELLAS
344     **"), nl,
345     write("
346     ====="), nl, nl,
347     write("El nucleo solar se encuentra habitado por dos tipos de dragones."),
348     nl,
349     write("Dragones Rojos"), nl,
350     write("Dragones Amarillos. Depredadores."), nl, nl,
351     write("A continuacion, podra simular la evolucion de la poblacion de
352     dragones"), nl,
353     write("en los ecosistemas que proporcionan distintos tipos de estrellas."),
354     nl, nl.
355
356 mensaje_opciones :-
357     clear,

```



```

352     write("El sol es una estrella 'enana amarilla' con temperatura: 5800 K"),
nl,
353     write("La poblacion de dragones que habita su nucleo es estable."), nl, nl,
354     write("Elija la opcion en la que desea simular la evolucion de la poblacion
de dragones."), nl, nl,
355     write("*OPCION 1*"), nl,
356     write("-----"), nl,
357     descripcion(estrella1, (Nombre1, Tipo1, T1)),
358     write("Estrella: "), write(Nombre1), write(". Tipo: "), write(Tipo1),
write(". Temperatura: "), write(T1), write(" k."), nl,
359     write("Pulse '1.' e intro."), nl, nl,
360     write("*OPCION 2*"), nl,
361     write("-----"), nl,
362     descripcion(estrella2, (Nombre2, Tipo2, T2)),
363     write("Estrella: "), write(Nombre2), write(". Tipo: "), write(Tipo2),
write(". Temperatura: "), write(T2), write(" k."), nl,
364     write("Pulse '2.' e intro."), nl, nl,
365     write("*OPCION 3*"), nl,
366     write("-----"), nl,
367     descripcion(estrella3, (Nombre3, Tipo3, T3)),
368     write("Estrella: "), write(Nombre3), write(". Tipo: "), write(Tipo3),
write(". Temperatura: "), write(T3), write(" k."), nl,
369     write("Pulse '3.' e intro."), nl, nl,
370     read(Opcion),
371     clear,
372     write("Selecciono la opcion: "), write(Opcion), nl,
373     (
374         opcion_correcta(Opcion) → set_inicio(Opcion);
375         mensaje_opciones
376     ).
377
378 %Chequeo opcion
379 opcion_correcta(Opcion) :-
380     Opcion==1;Opcion==2;Opcion==3.

```