

# Sistema Basado en Reglas: JESS

## Parte 1: Razonamiento backward

Índice

Diseño de la base de conocimiento.....3

Evaluación.....6

Conclusiones.....7

## Diseño de la base de conocimiento

En esta práctica vamos a trabajar con encadenamiento inferencial backward (hacia atrás) pero como ya sabemos jess no tiene un mecanismo nativo implementado para trabajar con este tipo de encadenamiento pero nos permite simular su comportamiento mediante la implementación pseudo-backward-chaining. El encadenamiento inferencial backward va dirigido por los objetivos y parte del conocimiento que necesita (ya que no puede ser deducido a partir de las reglas) lo requiere haciendo preguntas al usuario. En esta parte de la práctica vamos a trabajar con hechos ordenados, y vamos a crear un sistema que simule un encadenamiento inferencial backward (pseudo-backward) que nos permita adivinar que animal estamos pensando, para ello vamos a describir como esta estructurado nuestro programa.

```
/*
  AUTOR: Pascual Andres Carrasco Gomez
  Practica: Jess (Parte 1)
  Descripción: Programa para adivinar que animal estamos pensando
  Encadenamiento inferencial: backward
*/

(clear)

(do-backward-chaining cubre)
(do-backward-chaining reproduce)
(do-backward-chaining pico)
(do-backward-chaining vuela)
(do-backward-chaining nada)
(do-backward-chaining color)
(do-backward-chaining rayas)
(do-backward-chaining piel)
(do-backward-chaining come)
(do-backward-chaining garras)
(do-backward-chaining da_leche)
```

Observamos que la declaración “do-backward-chaining <fact>” se hace siempre después de las definiciones de templates (no es nuestro caso porque trabajamos con hechos ordenados) y antes de la definición de las reglas del sistema. Estos hechos son aquellos que el sistema no puede deducir de forma automática y tiene que preguntar al usuario.

Vamos a ver ahora las reglas que tenemos en el sistema, tenemos tres tipos de reglas:

- **Regla inicial:** Permite iniciar el sistema.

```
; REGLA INICIAL
⊖ (defrule inicio
  (nombre ?nombre)
  =>
  (printout t "El nombre del animal es: " ?nombre crlf)
  (halt) ; cuando encuentra el primer objetivo finaliza
)
```

- **Reglas para el conocimiento:** son aquellas reglas que permiten al sistema inferir el conocimiento.

```
; REGLAS CONOCIMIENTO
+ (defrule es_tipo_mamifero_1[])
+ (defrule es_tipo_ave_1[])
+ (defrule alimenta_carnivoro[])
+ (defrule es_un_guepardo[])
+ (defrule es_un_tigre[])
+ (defrule es_un_pinguino[])
+ (defrule es_una_gaviota[])
+ (defrule es_tipo_mamifero_2[])
+ (defrule es_tipo_ave_2[])
```

- **Reglas para preguntar al usuario:** Estas reglas son las que permiten al sistema comunicarse con el usuario pero tienen una característica especial y es que para cada hecho definido como “do-backward-chaining” existe una regla que pregunta al usuario el valor de dicho hecho.

```
; PREGUNTAS AL USUARIO
+ (defrule preguntar-cubre[])
+ (defrule preguntar-reproduce[])
+ (defrule preguntar-pico[])
+ (defrule preguntar-vuela[])
+ (defrule preguntar-nada[])
+ (defrule preguntar-color[])
+ (defrule preguntar-rayas[])
+ (defrule preguntar-piel[])
+ (defrule preguntar-come[])
+ (defrule preguntar-garras[])
+ (defrule preguntar-da_leche[])
```

Consideraciones que hemos tenido en cuenta para desarrollar el sistema:

- La regla inicial tiene un valor muy importante en el sistema ya que es la que inicia el encadenamiento hacia atrás del sistema, en nuestro caso lo que queremos averiguar es el nombre del animal que queremos adivinar.

- En las reglas de conocimiento es importante el orden en el que definimos las condiciones necesarias que constituyen la parte izquierda de la regla (LHS), ya que si ponemos los hechos que son deducibles por el sistema (no observables) antes que los hechos no deducibles por el sistema (observables) se lanzan menos preguntas al usuario de las que son necesarias ya que si el hecho deducible no es cierto el sistema no lanzará dichas preguntas al usuario. Ejemplo:

```
(defrule alimenta_carnivoro
  (tipo mamifero)
  (come ?come)
  (test (eq ?come carne))
  (garras ?garras)
  (test (eq ?garras si))
=>
  (assert (alimenta carnivoro))
)
```

NOTA: En esta regla si el sistema no infiere que el animal es de tipo mamífero no lanzará las reglas “preguntar-come” ni “preguntar-garras” que necesitan información del usuario.

- Las reglas para preguntar al usuario tienen la siguiente estructura:

```
(defrule preguntar-come
  (need-come ?)
=>
  (printout t "Come: (carne|vacio)" crlf)
  (assert (come (read))))
)
```

NOTA: Cuando se declara un hecho <fact> como backward chaining reactivo (do-backward-chaining come) y el interprete de jess encuentra una regla con dicho hecho en su parte LHS, y necesita ser evaluado, automáticamente creará un hecho <need-fact> (need-come), de esta forma asociamos los hechos observables (no deducibles por el sistema) a las preguntas realizadas al usuario.

NOTA: Un lenguaje con encadenamiento inferencial backward nativo no necesita realizar los pasos que hemos comentado antes pero jess al simular un encadenamiento backward (pseudo-backward) necesita de dichas instrucciones.

- Como jess de forma nativa trabaja con encadenamiento forward es importante utilizar la instrucción (halt) en la parte derecha de la regla que sea un objetivo, en nuestro caso es la regla inicial, para que cuando encuentre el objetivo deseado (nombre del animal) no siga activando y ejecutando reglas que no son necesarias.

## Evaluación

En este apartado vamos a evaluar el sistema mediante una serie de ejecuciones que demuestran que el sistema funciona correctamente, también vamos a observar que el sistema no siempre encuentra una respuesta que proporcionar al usuario.

- El animal es una gaviota:

```
Jess, the Rule Engine for the Java Platform
Copyright (C) 2008 Sandia Corporation
Jess Version 7.1p2 11/5/2008

This copy of Jess will expire in 1692 day(s).
Pico: (si|no)
si
Reproduce: (oviparo|vacio)
oviparo
Vuela: (mal|muy_bien|vacio)
muy_bien
El nombre del animal es: gaviota
```

- El animal es un pingüino:

```
Jess, the Rule Engine for the Java Platform
Copyright (C) 2008 Sandia Corporation
Jess Version 7.1p2 11/5/2008

This copy of Jess will expire in 1692 day(s).
Pico: (si|no)
si
Reproduce: (oviparo|vacio)
oviparo
Vuela: (mal|muy_bien|vacio)
mal
Nada: (bien|vacio)
bien
El nombre del animal es: pinguino
```

- El animal es un tigre:

```
Jess, the Rule Engine for the Java Platform
Copyright (C) 2008 Sandia Corporation
Jess Version 7.1p2 11/5/2008

This copy of Jess will expire in 1692 day(s).
Pico: (si|no)
no
Da leche: (si|no)
si
Come: (carne|vacio)
carne
Garras: (si|no)
si
Color: (pardo|vacio)
pardo
Rayas: (negras|vacio)
negras
El nombre del animal es: tigre
```

- El animal es un guepardo:

```
Jess, the Rule Engine for the Java Platform
Copyright (C) 2008 Sandia Corporation
Jess Version 7.1p2 11/5/2008

This copy of Jess will expire in 1692 day(s).
Pico: (si|no)
no
Da leche: (si|no)
si
Come: (carne|vacio)
carne
Garras: (si|no)
si
Color: (pardo|vacio)
pardo
Rayas: (negras|vacio)
vacio
Piel: (manchas|vacio)
manchas
El nombre del animal es: guepardo
```

---

- Ejecución del sistema sin respuesta:

```
Jess, the Rule Engine for the Java Platform
Copyright (C) 2008 Sandia Corporation
Jess Version 7.1p2 11/5/2008

This copy of Jess will expire in 1692 day(s).
Pico: (si|no)
no
Da leche: (si|no)
no
Cubre: (plumas|pelo|vacio)
pelo
Come: (carne|vacio)
carne
Garras: (si|no)
no
|
```

NOTA: Si el sistema no tiene respuesta el usuario debe finalizar la ejecución de forma manual.

## Conclusiones

El sistema es un programa de juguete que solo infiere conocimiento de cuatro animales pero es interesante conocer como funciona el encadenamiento backward para ver que es mas eficiente que el forward por la simple razón de que como hemos visto va guiado por los objetivos mientras que el encadenamiento forward va guiado por los datos y deduce todo lo que puede deducir. Como ya sabemos jess se utiliza bastante en empresas actualmente pero si queremos trabajar con sistemas que funcionen con un encadenamiento backward hemos de buscar otras opciones que por desgracia suelen ser de pago ya que jess simula un encadenamiento backward como hemos visto en esta parte de la practica pero no tiene este tipo de encadenamiento de forma nativa implementado.

# Sistema Basado en Reglas: JESS

Parte 2: Razonamiento forward



# Índice

Descripción del Problema.....10

Diseño de la base de conocimiento.....11

Evaluación del sistema.....13

Conclusiones.....17

## Descripción del Problema

Se trata de un sistema experto que se encargue de diagnosticar problemas que presenta un ordenador doméstico, la descripción es la que muestro a continuación:

- Si el S.O es Windows y no hay antivirus instalado entonces entran virus.
- Si el arranque del S.O tarda entonces hay errores de búsqueda.
- Si entran virus entonces ficheros defectuosos.
- Si hay errores de búsqueda y ficheros defectuosos entonces el disco duro va lento.
- Si el PC tiene mas de dos años entonces el pc es viejo.
- Si el PC tiene dos años o menos entonces el pc es nuevo.
- Si el disco duro va lento y el S.O es Windows y las aplicaciones van lentas entonces el disco duro esta fragmentado.
- Si el PC es viejo y los ventiladores hacen ruido entonces obstrucción de suciedad en el PC.
- Si hay cortes de luz y hay programas en segundo plano entonces aparecen ficheros defectuosos.
- Si las aplicaciones son lentas entonces hay programas ejecutándose en segundo plano.
- Si programas en segundo plano y reinicios constantes entonces se calienta la CPU.
- Si se calienta la CPU y hay obstrucción de suciedad entonces se produce un fallo en la refrigeración del disipador.
- Si PC es viejo y programas ejecutándose en segundo plano entonces memoria RAM saturada.
- Si las aplicaciones se ejecutan rápido y el disco duro va lento entonces es recomendable que instales un antivirus.
- Si el disco duro esta fragmentado entonces consejo de desfragmentar el disco duro.
- Si RAM saturada y se calienta la CPU y S.O es Windows entonces consejo formatear el PC.
- Si el disco duro va lento y el S.O es Linux entonces consejo formatear el PC.
- Si fallo en la refrigeración del disipador entonces consejo cambiar la pasta térmica del disipador y limpiar por dentro el PC.

## Diseño de la base de conocimiento

Hemos de tener en cuenta que cuando el dominio de un problema que queremos representar es complejo se requiere de representaciones estructurales y flexibles, esto nos lleva a realizar un estudio previo del dominio y establecer categorías que identifiquen y organicen la información.

Nuestro sistema no tiene un dominio excesivamente complejo pero es interesante trabajar con una ontología que nos permita representar el conocimiento y aplicarla de forma practica en jess.

Una ontología esta formada por clases, relaciones, axiomas e instancias las cuales vamos a describir para nuestro sistema a continuación:

- Clases: Nuestro sistema esta representado por tres clases “pc” que describe las características del ordenador que permiten realizarle un diagnostico, “sistema operativo” que describe las características de dos sistemas operativos (linux y windows) y por último “usuario” que contiene el nombre del propietario del PC.
- Relaciones: En nuestro sistema no aparecen relaciones entre clases porque trabajamos solo con un usuario y un ordenador pero si trabajásemos con mas usuarios tendríamos que establecer clases del tipo un usuario tiene un ordenador (tiene(id\_user,id\_pc), instalado(id\_pc,sistema\_operativo)).
- Axiomas: Los axiomas determinan afirmaciones que son siempre ciertas en la ontología y se representan mediante la herencia, jerarquía, tipo de valores, cardinalidad, ...
- Instancias: En este caso como he comentado antes solo tendremos tres instancias en nuestro sistema, una que hace referencia a un PC, otra que hace referencia al sistema operativo que tiene ese PC instalado y un usuario propietario de dicho PC.

Las clases de nuestro sistema presentan la siguiente estructura:

Clase (templates)	Atributos (slots y facetas)
pc	estado, disco_duro, cpu, ram, ruido_ventiladores, fuente_cortes, disipador, disco_fragmentado, obstruccion_suciedad, mas_2_anos
sistema_operativo	id, antivirus, virus, arranque, ficheros, aplicaciones, errores_busqueda, programas_2plano, reinicio_constante
usuario	nombre

NOTA: En el código están comentados los valores que puede tomar cada atributo en el programa.

No se ha especificado las facetas de los slots en el código porque trabajamos siempre con “modify” y en ese caso el sistema tiene plena confianza en el programador.

Las reglas en nuestro sistema se dividen en tres grupos:

- Reglas de conocimiento del sistema: permiten mediante una inferencia obtener respuestas y son las que representan el conocimiento experto del sistema. Estas reglas tienen nombres

concretos.

- Reglas de interacción con el usuario: permiten que el sistema obtenga información del usuario para rellenar los atributos (slots y facetas) de las clases (templates) del sistema. Estas reglas tienen la siguiente nomenclatura: q\_nombreDeRegla.
- Reglas de solución: Estas reglas proporcionan una respuesta al usuario (siempre y cuando el conocimiento del sistema pueda inferir una respuesta mediante los datos proporcionados por el usuario) y finalizan la ejecución del sistema con la instrucción “halt”. Estas reglas tienen la siguiente nomenclatura: sol\_nombreDeRegla.

Tipo de regla	Regla
Reglas de conocimiento	entran_virus, errores_busqueda, ficheros_defectuosos_virus, disco_luro_lento, ...
Reglas de interacción	q_nombre, q_sistema_operativo, q_arranque, ...
Reglas de solución	sol_antivirus, sol_desfragmentar, ...

He creado una función “inicio()” que pregunta al usuario antes de lanzar la orden (run) por el control inferencial y si queremos realizar la ejecución en modo verbose.

Por último comentar que se ha especificado en las definiciones de templates la opción (declare (slot-specific TRUE)) que impide que las reglas se disparen varias veces sobre el mismo hecho.

```
; TEMPLATES
(deftemplate pc

(deftemplate sistema operativo

(deftemplate usuario


; INSTANCIAS (VACIAS)
(assert (pc))
(assert (sistema_operativo))
(assert (usuario))


; REGLAS
(defrule entran_virus

(defrule errores_busqueda

(defrule ficheros_defectuosos_virus

(defrule disco_duro_lento

(defrule pc_viejo

(defrule pc_nuevo

(defrule disco_duro_fragmentado

(defrule obstruccion_suciedad

(defrule ficheros_defectuosos_cortes

(defrule programas_2_plano

(defrule calienta_cpu

(defrule fallo_disipador

(defrule ram_saturada

(defrule q_nombre

(defrule q_sistema_operativo

(defrule q_arranque

(defrule q_edad_pc

(defrule q_antivirus

(defrule q_cortes_luz

(defrule q_aplicaciones_lentas
```

## Evaluación del sistema

En este apartado vamos a explicar como funciona el sistema mediante ejecuciones. Hemos de tener en cuenta que nuestro sistema trabaja con un encadenamiento inferencial de tipo forward (reglas tratadas como expresiones if-then).

A continuación vamos a realizar diferentes ejecuciones del sistema que nos proporcionarán una respuesta según el conocimiento descrito en la descripción del problema, pero hemos de tener en cuenta que el sistema no siempre nos va a dar una respuesta como mostraré mas adelante.

- Desfragmentar el disco duro:

```
Jess, the Rule Engine for the Java Platform
Copyright (C) 2008 Sandia Corporation
Jess Version 7.1p2 11/5/2008
```

```
This copy of Jess will expire in 1693 day(s).
```

```
¿Indica el control inferencial? (depth | breadth)
```

```
depth
```

```
¿Quieres activar el modo verbose? (si | no)
```

```
no
```

```
¿Me dice su nombre para poder dirigirme a usted?
```

```
Pascu
```

```
Pascu: ¿Que sistema operativo tiene instalado? (windows | linux)
```

```
windows
```

```
Pascu: ¿Como son las aplicaciones al ejecutarse? (lentas | rapidas)
```

```
lentas
```

```
Pascu: Una vez iniciada la sesion, ¿el ordenador se reinicia constantemente? (si | no)
```

```
no
```

```
Pascu: ¿Cómo es el arranque del ordenador? (lento | normal)
```

```
lento
```

```
Pascu: ¿Tienes algun antivirus instalado en el sistema operativo? (si | no)
```

```
no
```

```
Pascu: ¿Tu PC tiene mas de 2 años? (si | no)
```

```
no
```

```
Pascu: El disco duro esta fragmentado, se recomienda que utilices una aplicacion para desfragmentarlo.
```

```
(defrule q_ruido_ventiladores
(defrule q_reinicio_constante
(defrule sol_antivirus
(defrule sol_desfragmentar
(defrule sol_formateo_windows
(defrule sol_formatear_linux
(defrule sol_cambiar_pasta_termica
(deffunction inicio ()
(inicio)
(run)
```

- Formatear Windows:

```
Jess, the Rule Engine for the Java Platform
Copyright (C) 2008 Sandia Corporation
Jess Version 7.1p2 11/5/2008
```

```
This copy of Jess will expire in 1693 day(s).
```

```
¿Indica el control inferencial? (depth | breadth)
```

```
depth
```

```
¿Quieres activar el modo verbose? (si | no)
```

```
no
```

```
¿Me dice su nombre para poder dirigirme a usted?
```

```
Pascu
```

```
Pascu: ¿Que sistema operativo tiene instalado? (windows | linux)
```

```
windows
```

```
Pascu: ¿Como son las aplicaciones al ejecutarse? (lentas | rapidas)
```

```
lentas
```

```
Pascu: Una vez iniciada la sesion, ¿el ordenador se reinicia constantemente? (si | no)
```

```
si
```

```
Pascu: ¿Cómo es el arranque del ordenador? (lento | normal)
```

```
normal
```

```
Pascu: ¿Tienes algun antivirus instalado en el sistema operativo? (si | no)
```

```
si
```

```
Pascu: ¿Se va la luz con frecuencia en su domicilio? (si |no)
```

```
no
```

```
Pascu: ¿Los ventiladores hacen mas ruido de lo normal? (si | no)
```

```
no
```

```
Pascu: ¿Tu PC tiene mas de 2 años? (si | no)
```

```
si
```

```
Pascu: Se recomienda que el S.O Windows sea formateado.
```

- Formatear Linux:

```
Jess, the Rule Engine for the Java Platform
Copyright (C) 2008 Sandia Corporation
Jess Version 7.1p2 11/5/2008

This copy of Jess will expire in 1693 day(s).
¿Indica el control inferencial? (depth | breadth)
depth
¿Quieres activar el modo verbose? (si | no)
no
¿Me dice su nombre para poder dirigirme a usted?
Pascu
Pascu: ¿Que sistema operativo tiene instalado? (windows | linux)
linux
Pascu: ¿Como son las aplicaciones al ejecutarse? (lentas | rapidas)
lentas
Pascu: Una vez iniciada la sesion, ¿el ordenador se reinicia constantemente? (si | no)
no
Pascu: ¿Cómo es el arranque del ordenador? (lento | normal)
lento
Pascu: ¿Tu PC tiene mas de 2 años? (si | no)
no
Pascu: ¿Los ventiladores hacen mas ruido de lo normal? (si | no)
no
Pascu: ¿Se va la luz con frecuencia en su domicilio? (si |no)
si
Pascu: Se recomienda que el S.O Linux sea formateado.
```

- Cambiar pasta térmica y limpiar el PC:

```
Jess, the Rule Engine for the Java Platform
Copyright (C) 2008 Sandia Corporation
Jess Version 7.1p2 11/5/2008

This copy of Jess will expire in 1693 day(s).
¿Indica el control inferencial? (depth | breadth)
depth
¿Quieres activar el modo verbose? (si | no)
no
¿Me dice su nombre para poder dirigirme a usted?
Pascu
Pascu: ¿Que sistema operativo tiene instalado? (windows | linux)
linux
Pascu: ¿Como son las aplicaciones al ejecutarse? (lentas | rapidas)
lentas
Pascu: Una vez iniciada la sesion, ¿el ordenador se reinicia constantemente? (si | no)
si
Pascu: ¿Cómo es el arranque del ordenador? (lento | normal)
normal
Pascu: ¿Se va la luz con frecuencia en su domicilio? (si |no)
no
Pascu: ¿Los ventiladores hacen mas ruido de lo normal? (si | no)
si
Pascu: ¿Tu PC tiene mas de 2 años? (si | no)
si
Pascu: Se recomienda que se cambie la pasta termica del disipador del procesador y que se limpie la suciedad interna del PC
```

- Instalar un antivirus:

```
Jess, the Rule Engine for the Java Platform
Copyright (C) 2008 Sandia Corporation
Jess Version 7.1p2 11/5/2008

This copy of Jess will expire in 1693 day(s).
¿Indica el control inferencial? (depth | breadth)
depth
¿Quieres activar el modo verbose? (si | no)
no
¿Me dice su nombre para poder dirigirme a usted?
Pascu
Pascu: ¿Que sistema operativo tiene instalado? (windows | linux)
windows
Pascu: ¿Como son las aplicaciones al ejecutarse? (lentas | rapidas)
rapidas
Pascu: Una vez iniciada la sesion, ¿el ordenador se reinicia constantemente? (si | no)
no
Pascu: ¿Cómo es el arranque del ordenador? (lento | normal)
lento
Pascu: ¿Tienes algun antivirus instalado en el sistema operativo? (si | no)
no
Pascu: El sistema operativo tiene virus, es recomendable que instales un antivirus.
```

Hemos de tener en cuenta que las combinaciones en las ejecuciones anteriores no son las únicas que existen para que el sistema nos proporcione la respuesta. El sistema va activando reglas y ejecutándolas hasta que ejecuta una regla de tipo `sol_nombreDeRegla` que contiene la instrucción `(halt)` y detiene la ejecución, pero como he comentado antes el sistema puede que no encuentre una respuesta, lo vemos por ejemplo en la siguiente ejecución:

```
Jess, the Rule Engine for the Java Platform
Copyright (C) 2008 Sandia Corporation
Jess Version 7.1p2 11/5/2008

This copy of Jess will expire in 1693 day(s).
¿Indica el control inferencial? (depth | breadth)
depth
¿Quieres activar el modo verbose? (si | no)
no
¿Me dice su nombre para poder dirigirme a usted?
Pascu
Pascu: ¿Que sistema operativo tiene instalado? (windows | linux)
linux
Pascu: ¿Como son las aplicaciones al ejecutarse? (lentas | rapidas)
rapidas
Pascu: Una vez iniciada la sesion, ¿el ordenador se reinicia constantemente? (si | no)
no
Pascu: ¿Cómo es el arranque del ordenador? (lento | normal)
lento
Pascu: ¿Los ventiladores hacen mas ruido de lo normal? (si | no)
si
Pascu: ¿Se va la luz con frecuencia en su domicilio? (si |no)
no
Pascu: ¿Tu PC tiene mas de 2 años? (si | no)
no
```

NOTA: Si el sistema no proporciona una respuesta no se ejecuta la instrucción `(halt)` por lo tanto el usuario debe detener la ejecución manualmente.

Es interesante también comentar que todas las ejecuciones que he realizado han sido con control inferencial en profundidad (`depth`), hemos de tener en cuenta que nuestro sistema puede obtener diferentes resultados según el control inferencial indicado (`depth` o `breadth`), esto se debe a que la ejecución de las reglas activadas es en distinto orden según el control inferencial indicado y al utilizar una orden externa como es `(halt)` cuando encontramos una respuesta el sistema detiene la ejecución. El sistema lo he implementado de este modo para comprender mejor que no siempre se ejecutan las mismas reglas o el mismo número de reglas cuando trabajamos con un control inferencial diferente. Lo explico con detalle en las siguientes ejecuciones:

- Control inferencial: breadth

```
Jess, the Rule Engine for the Java Platform
Copyright (C) 2008 Sandia Corporation
Jess Version 7.1p2 11/5/2008
```

```
This copy of Jess will expire in 1693 day(s).
```

```
¿Indica el control inferencial? (depth | breadth)
```

```
breadth
```

```
¿Quieres activar el modo verbose? (si | no)
```

```
no
```

```
¿Me dice su nombre para poder dirigirme a usted?
```

```
Pascu
```

```
Pascu: ¿Se va la luz con frecuencia en su domicilio? (si | no)
```

```
si
```

```
Pascu: ¿Tu PC tiene mas de 2 años? (si | no)
```

```
si
```

```
Pascu: ¿Que sistema operativo tiene instalado? (windows | linux)
```

```
windows
```

```
Pascu: ¿Como son las aplicaciones al ejecutarse? (lentas | rapidas)
```

```
lentas
```

```
Pascu: ¿Cómo es el arranque del ordenador? (lento | normal)
```

```
normal
```

```
Pascu: Una vez iniciada la sesion, ¿el ordenador se reinicia constantemente? (si | no)
```

```
si
```

```
Pascu: ¿Los ventiladores hacen mas ruido de lo normal? (si | no)
```

```
si
```

```
Pascu: ¿Tienes algun antivirus instalado en el sistema operativo? (si | no)
```

```
si
```

```
Pascu: Se recomienda que se cambie la pasta termica del disipador del procesador y que se
```

- Control inferencial: depth

```
Jess, the Rule Engine for the Java Platform
Copyright (C) 2008 Sandia Corporation
Jess Version 7.1p2 11/5/2008
```

```
This copy of Jess will expire in 1693 day(s).
```

```
¿Indica el control inferencial? (depth | breadth)
```

```
depth
```

```
¿Quieres activar el modo verbose? (si | no)
```

```
no
```

```
¿Me dice su nombre para poder dirigirme a usted?
```

```
Pascu
```

```
Pascu: ¿Que sistema operativo tiene instalado? (windows | linux)
```

```
windows
```

```
Pascu: ¿Como son las aplicaciones al ejecutarse? (lentas | rapidas)
```

```
lentas
```

```
Pascu: Una vez iniciada la sesion, ¿el ordenador se reinicia constantemente? (si | no)
```

```
si
```

```
Pascu: ¿Cómo es el arranque del ordenador? (lento | normal)
```

```
normal
```

```
Pascu: ¿Tienes algun antivirus instalado en el sistema operativo? (si | no)
```

```
si
```

```
Pascu: ¿Se va la luz con frecuencia en su domicilio? (si | no)
```

```
si
```

```
Pascu: ¿Los ventiladores hacen mas ruido de lo normal? (si | no)
```

```
si
```

```
Pascu: ¿Tu PC tiene mas de 2 años? (si | no)
```

```
si
```

```
Pascu: Se recomienda que el S.O Windows sea formateado.
```



En ambos casos se activan en algún momento de la ejecución las reglas que el sistema operativo sea Windows, que la CPU se calienta, que la RAM esta saturada y que hay obstrucción de suciedad.

En el caso de control inferencial breadth se ejecuta la regla sol\_cambiar\_pasta\_termica ya que el sistema infiere antes a partir del conocimiento que la CPU se calienta y que hay obstrucción de suciedad.

En el caso de control inferencial depth se ejecuta la regla sol\_formateo\_windows ya que el sistema infiere el conocimiento antes de que el sistema operativo es Windows, la CPU se calienta y la RAM esta saturada.

NOTA: Si no utilizásemos la instrucción (halt) en las reglas de solución el sistema en este caso devolvería las dos respuestas intercambiando el orden según la aparición en la inferencia, con esto observamos que si utilizamos el encadenamiento inferencial forward y no hay retractaciones ni acciones externas el control inferencial aplicado no es relevante.

## Conclusiones

Hemos observado que cuando trabajamos con un encadenamiento hacia delante (forward) el sistema deduce todo lo que se puede deducir por eso este tipo de encadenamiento se conoce como un encadenamiento guiado por los datos, por otra parte observamos que la utilidad de este tipo de encadenamiento en un SBC resulta útil cuando trabajamos con sistemas que no precisan delimitar una respuesta y al mismo tiempo pueden proporcionar una o varias respuestas. Por último comentar que como hemos visto antes el control de inferencia es irrelevante cuando no hay hecho retractados ni acciones externas.