

Práctica final Ingeniería de Conocimiento



Universidad de Granada

Antonio Manuel Fresneda Rodríguez

antoniomfr@correo.ugr.es

15 de Junio de 2018

Índice

1	Ejercicio 1	3
2	Ejercicio 2	3
3	Ejercicio 3	4
4	Ejercicio4	4

1 Ejercicio 1

Este ejercicio se realizó en la práctica 1.

2 Ejercicio 2

- **Menos de N empleados TG**

En este ejercicio nos pide que si hay menos de N empleados atendiendo TG. Esto quiere decir que el sistema avise cuando haya menos de N empleados fichados de un tipo.

Para realizar esto, he definido la regla libres, que captura la hora actualizada, el minimo de empleados y el hecho *ComprobacionMinEmpleados*. Este hecho se retracta en esta regla mientras que se aserta cada vez que un empleado ficha. Así conseguimos que cada vez que un empleado fiche (ya sea para irse, descansar o empezar a trabajar) se dispare la regla y se comunique si hay menos del minimo o si hay suficientes.

- **No hay empleados TE**

Este ejercicio es lo mismo que el anterior si cambiamos TG por una variable. Lo que he hecho ha sido modificar el fichero de constantes.

- **Usuario que lleva más de un tiempo máximo de espera**

Lo que he hecho ha sido que a la hora de que un usuario entre a la cola de espera, calculo el tiempo actual del sistema con la siguiente orden:

(bind ?t (HoraActualizada))

Se añade un hecho con el identificador del usuario y su tiempo.

Luego tengo una regla que captura este hecho (identificando aquellos usuarios cuyo identificador sea mayor al del ultimo usuario atendido de dicho tramite) y si se da la condición:

(if (> (- (HoraActualizada) ?tiempo) ?tiempoMax) se imprime por pantalla que dicho usuario ya lleva demasiado tiempo esperando. Esta regla se llama: **ComprobarTiempo**

- **Usuario cuyo trámite se retrasa**

Para este ejercicio lo que he hecho ha sido añadir un nuevo hecho cada vez que a una persona se le ha asignado una oficina. Este hecho guarda el tiempo en que se le asigno la oficina.

Luego, tengo otra regla que captura dos hechos: El que guarda el tiempo en el que se le asignó la oficina y *(HoraActualizada ?n)* con el que consigo que esta regla se dispare siempre y en el caso de que la resta del tiempo actual menos el tiempo en el que el usuario entro sea mayor que el limite hago la notificación. He añadido tambien un hecho de control para que no cicle.

Para la parte de guardar los datos lo que he hecho ha sido en la regla RegistrarCaso he añadido que capture tanto el hecho con la hora de inicio del tramite como la hora en la que empezó a estar en cola.

Una vez hecho esto he calculado el tiempo de tramite como: *(- (HoraActualizada) tiempoInicioTramite)* y el tiempo de espera en cola como: *(- tiempoInicialTramite tiempoInicialCola)*. Ambos datos para todos los usuarios se han guardado en los ficheros datosE.txt (datos espera) y datosT.txt (datos tiempo trámite).

Una vez los datos estan en el fichero, he realizado un script en Python llamado Informe.py que leera los datos generados y realizará el informe.

3 Ejercicio 3

- **Empleado llega a la oficina tarde**

Para este apartado he modificado la regla de fichar al inicio de la jornada y compruebo la hora actual y obtengo el tiempo máximo de retraso.

Una vez hecho esto, resto la hora actual y la hora de comienzo de jornada y en el caso de que la diferencia sea mayor que ese tiempo máximo, se imprime el mensaje.

- **Empleado lleva mas de un tiempo máximo de descanso**

Para resolver este apartado, he creado la regla *EmpleadoSeVa*. En esta regla se captura el hecho de ficha cuando el estado del empleado es *No fichado*. Si la hora en segundos en la que ficha es menor que la hora en segundos de la hora de cierre, significa que el empleado se va a descansar mientras que si no es así, el empleado se va de la oficina. Cuando se va a descansar se guarda la hora en la que fichó. Una vez hecho esto, cuando el estado del empleado es *Descansando*, y este ficha, la regla captura la hora en la que fichó y la hora actual y en el caso de que la resta entre hora actual y hora en la que fichó sea mayor que el tiempo máximo de descanso, se imprime el mensaje.

- **Ha gestionado un número mínimo de trámites**

Cuando se hace un *assert* del hecho (*fin*) se dispara esta regla.

Cada vez que un empleado tramita un trámite, tiene un contador asociado que se incrementa. La regla captura este hecho y el hecho donde esta el mínimo de trámites para un tipo. Si el número de trámites que ha tramitado es menor que el este, se notifica.

- **Pregunta por la situación de los empleados**

Aquí si he cambiado bastante. Primero he creado un hecho para cada empleado llamado *EstadoEmpleado* que registra el estado del empleado de la siguiente forma:

- 0 si no está fichado.
- 1 si está fichado.
- 2 si está disponible.
- 3 si está atendiendo-
- 4 si está descansando.

Cada vez que el empleado cambia de situación actualizo el campo según lo que vaya a hacer. Finalmente tengo una regla que captura el hecho *Consulta ?empl* y el hecho *EstadoEmpleado* e imprime por pantalla el estado actual del empleado.

Por ultimo he adaptado algunas reglas para que usen estos hechos y así tener menos hechos de control. Para el informe de este apartado, este se va a generar cuando introduzcamos el hecho *fin*, se van a imprimir por pantalla y se van a guardar en el archivo "DatosEmpleados.txt".

El hecho *fin* se introduce cuando ficha el director y la hora actual es mayor o igual a la hora del cierre de la oficina.

4 Ejercicio4

Para apagar las luces de las oficinas, he considerado que si un trabajador está en el edificio (alomejor ha salido al servicio o ha salido a hablar con algún compañero) voy a dejar la luz de su oficina encendida. Solo se va a apagar la luz una vez que el empleado se va a descansar o se va de la oficina, en resumen, cuando fíche una vez que ya haya fichado.

Para encender la luz de una habitación he considerado que si se enciende el sensor de la puerta de la habitación (y en el caso de los aseos que se encienda el sensor de presencia) y la luz está apagada la enciendo.

En cuanto al pasillo, la luz se va a encender cuando se active el sensor de la puerta del pasillo y se va a apagar si el sensor de presencia esta apagado, si el sensor de la puerta esta apagado y si tras 10 segundos desde que se encendió el sensor de la puerta que provocó que se encendiese la luz. Así lo que consigo es que si ha entrado alguien puedo dejar la luz encendida un tiempo para que no se apague y así en el caso de que en ese intervalo de tiempo llegue alguien, no haya que encender la luz de nuevo.

Si alguien entra en el pasillo en esos 10 segundos, la luz sigue encendida y justo cuando salga del pasillo la

luz se apaga.

Comento que intenté usar un sistema algo más reactivo, pero no funcionaba nada bien (siempre apagaba las luces en algunos casos donde había gente dentro de la habitación)