



UNIVERSIDAD DE GRANADA

Ingeniería del conocimiento

Prácticas Grupo 2, Jueves 17:30-19:30

Práctica Final: Desarrollo de un sistema experto

Nombre: Adrián Jesús

Apellidos: Peña Rodríguez

Correo: adrianprodri@correo.ugr.es

DNI: 45604927-K

Fecha de entrega: 15/06/18

1.- Resumen	3
2.- Obtención del conocimiento	3
3.- Validación y verificación del sistema	3
4.- Descripción del sistema desarrollado	4

1.- Resumen

Comenzaremos esta memoria describiendo cómo funciona nuestro sistema experto. Nuestro SE esta basado en el manejo inteligente de una oficina mediante el uso de sensores, botones y demás hechos que nos ayudan a gestionar correctamente nuestra oficina inteligente.

El funcionamiento de nuestra oficina visto de manera rápida y resumida se compondría de un manejo inteligente de las solicitudes de administración de distintos usuarios así como medios para detectar anomalías en cuanto al trato a estos clientes, también tenemos control acerca de la situación de nuestros empleados, cuando entran, cuando salen, a fin de poder ofrecer al administrador un resumen de el comportamiento de estos durante la jornada de trabajo, entre otras muchas cosas que veremos en los siguientes apartados.

2.- Obtención del conocimiento

Para la obtención del conocimiento necesario para elaborar la práctica, hemos seguido la siguiente metodología:

- Primero leímos detenidamente la descripción de la práctica a fin de poder entenderla correctamente y posteriormente poder realizarla.
- Después de esto comenzamos a comprender lo que habíamos leído mediante la realización de croquis y demás escrituras en varios folios a fin de poder encontrar anomalías o dudas en nuestro conocimiento. Por ejemplo, determinamos el número máximo de usuarios que puede haber en según qué oficina.
- Por último realizamos una entrevista con un empleado de un banco el cual es el padre de un amigo con el fin de poder asegurar aún más el conocimiento, esto fue más que nada para resolver dudas ya que en cuanto al SE no nos podía ayudar pues su oficina como mucho tiene sensores de proximidad.

3.- Validación y verificación del sistema

Para la verificación de nuestro sistema hemos realizado múltiples pruebas con el sistema, además de un revisión del código y de la base del conocimiento con el fin de encontrar posibles errores. A medida que probamos casos y casos de prueba nos dimos cuenta de una cantidad considerable de fallos que fuimos corrigiendo poco a poco.

En cuanto a la validación, puesto que no había un experto como tal la hemos validado nosotros en función de nuestro criterio.

4.- Descripción del sistema desarrollado

A continuación vamos a proceder a hacer una descripción del sistema a fin de que el lector pueda comprenderlo mejor e incluso modificarlo según su criterio.

Comenzaremos hablando de los hechos de entrada que hemos creado, cabe decir que hay más hechos a parte de estos pero están creados dentro de reglas.

Los hechos Habitaciones son las habitaciones como tal, las instancias de la oficina

```
(defacts Habitaciones
(Habitacion Recepcion)
(Habitacion Pasillo)
(Habitacion Oficina1)
(Habitacion Oficina2)
(Habitacion Oficina3)
(Habitacion Oficina4)
(Habitacion Oficina5)
(Habitacion OficinaDoble)
(Habitacion Gerencia)
(Habitacion Papeleria)
(Habitacion Aseos)
(Habitacion AseoHombres)
(Habitacion AseoMujeres)
)
```

Estas son las puertas de nuestra oficina, el pasillo conecta con todas las instancias de hay que este señalado.

```
(defacts Puertas
(Puerta Recepcion Pasillo) ;; Hay una puerta que comunica Recepción con Pasillo
(Puerta Pasillo Oficina1)
(Puerta Pasillo Oficina2)
(Puerta Pasillo Oficina3)
(Puerta Pasillo Oficina4)
(Puerta Pasillo Oficina5)
(Puerta Pasillo Gerencia)
(Puerta Pasillo OficinaDoble)
(Puerta Pasillo Papeleria)
)
```

Los empleados, es decir, donde debe de trabajar cada uno de los empleados y el código de empleado.

```

(deffacts Empleados
  (Empleado G1 Oficina1)      ;;;; El empleado G1 atiende en la Oficina 1, iremos
modificando esto a medida que se mueva de sala
  (Empleado G2 Oficina2)
  (Empleado G3 Oficina3)
  (Empleado G4 Oficina4)
  (Empleado G5 Oficina5)
  (Empleado E1 OficinaDoble-1)
  (Empleado E2 OficinaDoble-2)
  (Empleado Recepcionista Recepcion)
  (Empleado Director Gerencia)
)

```

Codificación de la abreviaturas.

```

(deffacts Codificacion
  (Code TG "Tramites Generales")
  (Code TE "Tramites Especiales")
)

```

Tareas que puede llevar a cabo cada empleado

```

(deffacts Tareas
  (Tarea G1 TG)              ;;;; El empleado G1 atiende Trámites Generales
  (Tarea G2 TG)
  (Tarea G3 TG)
  (Tarea G4 TG)
  (Tarea G5 TG)
  (Tarea E1 TE)              ;;;; El empleado E1 atiende Trámites Especiales
  (Tarea E2 TE)
)

```

Con este template definimos el contenedor donde guardaremos las estadísticas de los empleados, cabe señalar que los nombres son descriptivos y por tanto no es necesaria la explicación de estos

```

(deftemplate estadoEmpleado ; Estadísticas
  (slot empleado (type SYMBOL) (default ?NONE))
  (slot estado (type SYMBOL) (default ?NONE))
  (slot numeroTramites)
  (slot tiempoTotal)
  (slot salidas)
)

```

Estos estaban inicialmente pero hemos añadido bastantes más, para saber que es cada uno hemos colocado una descripción al lado de cada uno de ellos:

(deffacts Inicializacion

(Personas 0) ;;; Inicialmente hay 0 personas en las oficinas

(Usuarios TG 0) ;;; Inicialmente hay 0 Usuarios de trámites generales

(UltimoUsuarioAtendido TG 0) ;;; Inicialmente se han atendido 0 usuarios de tramites generales

(Usuarios TE 0)

(UltimoUsuarioAtendido TE 0)

(Empleados 0) ;;; Inicialmente hay 0 empleados en las oficinas

(Empleados TG 0) ; Añadido para el ejercicio 2, esto es para controlar el número de empleados y poder avisar en caso de haber menos de los que debería

(Empleados TE 0) ; Añadido para el ejercicio 2, esto es para controlar el número de empleados y poder avisar en caso de haber menos de los que debería

(Ficho G1 0) ; Añadido para el ejercicio 3, con esto sabemos si un empleado ha fichado 1 o por el contrario no 0

(Ficho G2 0) ; Añadido para el ejercicio 3, con esto sabemos si un empleado ha fichado 1 o por el contrario no 0

(Ficho G3 0) ; Añadido para el ejercicio 3, con esto sabemos si un empleado ha fichado 1 o por el contrario no 0

(Ficho G4 0) ; Añadido para el ejercicio 3, con esto sabemos si un empleado ha fichado 1 o por el contrario no 0

(Ficho G5 0) ; Añadido para el ejercicio 3, con esto sabemos si un empleado ha fichado 1 o por el contrario no 0

(Ficho E1 0) ; Añadido para el ejercicio 3, con esto sabemos si un empleado ha fichado 1 o por el contrario no 0

(Ficho E2 0) ; Añadido para el ejercicio 3, con esto sabemos si un empleado ha fichado 1 o por el contrario no 0

(Ficho Director 0) ; Añadido para el ejercicio 3, con esto sabemos si un empleado ha fichado 1 o por el contrario no 0

(Esta G1 0) ; Añadido para el ejercicio 3, este hecho nos vale para saber que el empleado ya lleo (ficho) pero claro este puede salir a desayunar en cuyo caso sigo como fichado por lo que necesitamos saber que ha salido pero que el ya ficho

(Esta G2 0) ; Añadido para el ejercicio 3, este hecho nos vale para saber que el empleado ya lleo (ficho) pero claro este puede salir a desayunar en cuyo caso sigo como fichado por lo que necesitamos saber que ha salido pero que el ya ficho

(Esta G3 0) ; Añadido para el ejercicio 3, este hecho nos vale para saber que el empleado ya lleo (ficho) pero claro este puede salir a desayunar en cuyo caso sigo como fichado por lo que necesitamos saber que ha salido pero que el ya ficho

(Esta G4 0) ; Añadido para el ejercicio 3, este hecho nos vale para saber que el empleado ya lleo (ficho) pero claro este puede salir a desayunar en cuyo caso sigo como fichado por lo que necesitamos saber que ha salido pero que el ya ficho

(Esta G5 0) ; Añadido para el ejercicio 3, este hecho nos vale para saber que el empleado ya llevo (ficho) pero claro este puede salir a desayunar en cuyo caso sigo como fichado por lo que necesitamos saber que ha salido pero que el ya ficho

(Esta E1 0) ; Añadido para el ejercicio 3, este hecho nos vale para saber que el empleado ya llevo (ficho) pero claro este puede salir a desayunar en cuyo caso sigo como fichado por lo que necesitamos saber que ha salido pero que el ya ficho

(Esta E2 0) ; Añadido para el ejercicio 3, este hecho nos vale para saber que el empleado ya llevo (ficho) pero claro este puede salir a desayunar en cuyo caso sigo como fichado por lo que necesitamos saber que ha salido pero que el ya ficho

(Esta Director 0) ; Añadido para el ejercicio 3, este hecho nos vale para saber que el empleado ya llevo (ficho) pero claro este puede salir a desayunar en cuyo caso sigo como fichado por lo que necesitamos saber que ha salido pero que el ya ficho

(estadoEmpleado (empleado G1) (estado NOLLEGO) (numeroTramites 0) (tiempoTotal 0)) ; Añadido para el ejercicio 3, para el informe que se nos pide y para saber el estado del empleado el cual podemos preguntar gracias a la regla que se activa con (pregunta EMPLEADO), también sabremos el número de veces que salió (salidas) para en caso de ser más de una dar un aviso

(estadoEmpleado (empleado G2) (estado NOLLEGO) (numeroTramites 0) (tiempoTotal 0)) ; Añadido para el ejercicio 3, para el informe que se nos pide y para saber el estado del empleado el cual podemos preguntar gracias a la regla que se activa con (pregunta EMPLEADO), también sabremos el número de veces que salió (salidas) para en caso de ser más de una dar un aviso

(estadoEmpleado (empleado G3) (estado NOLLEGO) (numeroTramites 0) (tiempoTotal 0)) ; Añadido para el ejercicio 3, para el informe que se nos pide y para saber el estado del empleado el cual podemos preguntar gracias a la regla que se activa con (pregunta EMPLEADO), también sabremos el número de veces que salió (salidas) para en caso de ser más de una dar un aviso

(estadoEmpleado (empleado G4) (estado NOLLEGO) (numeroTramites 0) (tiempoTotal 0)) ; Añadido para el ejercicio 3, para el informe que se nos pide y para saber el estado del empleado el cual podemos preguntar gracias a la regla que se activa con (pregunta EMPLEADO), también sabremos el número de veces que salió (salidas) para en caso de ser más de una dar un aviso

(estadoEmpleado (empleado G5) (estado NOLLEGO) (numeroTramites 0) (tiempoTotal 0)) ; Añadido para el ejercicio 3, para el informe que se nos pide y para saber el estado del empleado el cual podemos preguntar gracias a la regla que se activa con (pregunta EMPLEADO), también sabremos el número de veces que salió (salidas) para en caso de ser más de una dar un aviso

(estadoEmpleado (empleado E1) (estado NOLLEGO) (numeroTramites 0) (tiempoTotal 0)) ; Añadido para el ejercicio 3, para el informe que se nos pide y para saber el estado del empleado el cual podemos preguntar gracias a la regla que se activa con (pregunta EMPLEADO), también sabremos el número de veces que salió (salidas) para en caso de ser más de una dar un aviso

(estadoEmpleado (empleado E2) (estado NOLLEGO) (numeroTramites 0) (tiempoTotal 0)) ;
Añadido para el ejercicio 3, para el informe que se nos pide y para saber el estado del empleado el cual podemos preguntar gracias a la regla que se activa con (pregunta EMPLEADO), también sabremos el número de veces que salió (salidas) para en caso de ser más de una dar un aviso

(estadoEmpleado (empleado Director) (estado NOLLEGO) (numeroTramites 0) (tiempoTotal 0)) ; Añadido para el ejercicio 3, para el informe que se nos pide y para saber el estado del empleado el cual podemos preguntar gracias a la regla que se activa con (pregunta EMPLEADO), también sabremos el número de veces que salió (salidas) para en caso de ser más de una dar un aviso

(tiempoMensaje 4) ; Tiempo para volver a sacar un mensaje por pantalla

(Ejecutar)

)

Control de luces, con esto sabremos si una determinada instancia de nuestra oficina tiene o no la luz encendida.

(deffacts Luces

(Luz Oficina1 OFF) ; Controlar las luces, es muy sencillo de entender, no es necesaria la explicación

(Luz Oficina2 OFF)

(Luz Oficina3 OFF)

(Luz Oficina4 OFF)

(Luz Oficina5 OFF)

(Luz OficinaDoble OFF)

(Luz Gerencia OFF)

(Luz Papeleria OFF)

(Luz Recepcion OFF)

(Luz Pasillo OFF)

(Luz Aseos OFF)

(Luz AseoHombres OFF)

(Luz AseoMujeres OFF)

)

Cantidad de personas en una oficina, el primer contador es las personas actuales en la oficina y el segundo es el máximo permitido en cada oficina según mi criterio.

(deffacts cantidadPersonas

(cantidad Oficina1 0 2) ; Cantidad de personas en una oficina, el 0 es ninguna en principio y el número de la derecha es el máximo permitido

(cantidad Oficina2 0 2)

(cantidad Oficina3 0 2)

(cantidad Oficina4 0 2)

(cantidad Oficina5 0 2)

(cantidad OficinaDoble 0 4)
)

Estos son los hechos iniciales, como podemos apreciar están los hechos necesarios para los ejercicios que se nos mandan.

En cuanto a los hechos de salida, tenemos que son informes y estadísticas las cuales están explicadas en el propio código y además son mostradas por pantalla mediante un mensaje. Lo siguiente a explicar es el esquema de razonamiento de nuestro sistema, lo haremos de la siguiente manera:

- El sistema está creado enteramente para satisfacer las necesidades que se nos han impuesto y de esta forma conseguir que la oficina sea inteligente y autónoma. Gracias a esto conseguimos que la oficina sea mucho más eficiente que cuando la manejan las propias personas y además así conseguimos agilizar el trabajo pues ningún empleado o usuario tiene que preocuparse de nada, todo es automático excepto el abrir y cerrar puertas.
- Nuestro sistema recibe a los trabajadores que llegan en primer lugar, fichan y comienzan a moverse por la oficina, movimientos los cuales tenemos controlados e iremos tomando estadísticas y nota de los sensores activados.
- Después de estos llegan los usuarios los cuales indican su tipo de trámite y son encolados para su posterior gestión, el usuario es informado de la oficina a la que tiene que ir y en caso de equivocación el sistema avisará. En caso de no haber más de un X trabajadores de un trámite trabajando se avisará a los usuarios al igual que si un usuario lleva más de X minutos esperando también será avisado, esto mismo ocurre cuando un trámite está durando más que el umbral establecido.
- En caso de no dar tiempo de atender a un nuevo usuario por falta de tiempo este será informado del percance.
- En cuanto a los movimientos de los empleados están controlados y sabemos en todo momento en que zona se encuentra un empleado, al igual que si sale a descansar o si ni siquiera a fichado. En caso de salir a descansar más de una vez se informará por pantalla de esto ya que no debería de ocurrir.
- Sabemos además si un usuario entra donde no debe o si se equivoca. También el número de personas en una oficina entre otras cosas.

En cuanto a los módulos realizados tenemos varias agrupaciones posibles y solo definiremos estas agrupaciones y en caso de querer una definición más específica siempre se podrá consultar el código el cual está comentado para que sea más fácil de comprender, sabemos que el comentar el código es una muy buena práctica a la hora del desarrollo de un sistema experto. No definiremos muy exhaustivamente cada uno para no extender demasiado esta guía.

Tenemos el módulo de gestión de usuarios en el cual meteremos las reglas EncolarUsuario, RegistrarCaso, AsignarEmpleado y NoposibleEncolarUsuario, este módulo se encarga de gestionar los usuarios que nos llegan, de manera que puedan ser atendidos, avisados o no atendidos según corresponda y sean avisados, en cuanto al conocimiento que usan es básicamente de horas, minutos, número de usuarios que llevamos atendiendo, el empleado asignado, el tiempo de gestión, el tiempo de cada caso, la media, etc. Deducimos que hay empleados para atender a los usuarios y debido a esto no comprobamos antes de encolar si hay empleados pues deducimos que debe de haber.

A continuación tenemos el módulo 2 que corresponde con el ejercicio 2, el cual se encarga de saber cuántos empleados hay y en caso de ser menos que un X avisar a los usuarios, podríamos haber introducido esta funcionalidad en las reglas del ejercicio 3 pero para poder diferenciarlos las hemos dejado así, la regla para generar informes nos sirve para el ejercicio 3 también.

Luego tenemos el módulo de control de empleados el cual como hemos indicado se encarga de controlar que hace cada empleado, cuando llega, cuando ficha, cuando se va, cuando descansa y luego nos dará un informe de su rendimiento por lo que también nos controla el tiempo de gestión, media, desviación además de las luces pues aquí identificamos cuando entra y cuando salen los empleados de un determinado sitio y por tanto podemos llevar un control de las luces.

El siguiente módulo es el de control de luces del pasillo y aseos pues las demás las controlamos en el módulo de arriba, en este módulo hacemos uso de los sensores de presencia y controlamos la luz de las zonas comunes.

Por último tenemos el módulo de control de incidencias donde como su nombre indica controlas incidencias y lo hacemos con hechos que o bien hemos definido al principio y hemos ido modificando o con flags que hemos sacado de hechos anteriores y que indican que algo va mal y lo sacamos por pantalla.

Por otro lado, tenemos los hechos que no son de entrada y salida y que utiliza el sistema para su funcionamiento estos hechos son sobre todo marcas en las reglas las cuales producen que no se vuelva a ejecutar la regla o que sepamos que un determinado evento a pasado y en base a este no podemos ejecutar alguna regla.

Algunos de estos hechos son marca y marca¹ los cuales nos ayudan a que no se produzca una situación en la que una regla no active otra y esta otra vuelva a activar esta y así infinitas veces. O auxiliares como (Ficho EMPLEADO 0/1) que nos sirve para tener un auxiliar a parte del propio fichar con el cual sabremos si un empleado fichó o no.

En cuanto a los informes que sacamos por pantalla al acabar la jornada tenemos que sacamos el número de trámites de cada tipo, la media de tiempo de cada tipo de trámite, la desviación típica de cada tipo de trámite así como esto mismo pero con cada uno de los empleados.

A parte de todo esto se puede ver más información en el propio código, información no colocada aquí para no extender de más ya que ya está en el código como se debe hacer en la creación de un SE.