

# **Práctica Final: Desarrollo de un SE**

Óscar David López Arcos  
75571640-B  
odlarcos@correo.ugr.es

Práctica Final:	1
Desarrollo de un SE	1
Resumen	3
Proceso de desarrollo	3
Representación y obtención del conocimiento	3
Procedimiento de validación y verificación	3
Descripción del sistema desarrollado	3
Control de Calidad de Servicios	3
Empleados atendiendo	3
Tiempo máximo de espera a ser atendido	4
Tiempo máximo gestión de trámite	4
Control de rendimiento	4
Retraso del empleado	5
Tiempo de descanso excedido	5
Mínimo número de trámites	5
Situación del empleado	5
Control Inteligente de las Luces	6
Luces Oficinas	6
Luces Aseos	6
Situaciones anómalas	7
Oficina vacía sin explicación	7
Más personas de lo esperado	7
Usuario no responde	7
Usuario se ha equivocado	7
Usuario en Gerencia o Papelería	7
Informes Generados	8
Informe 1	8
Informe 2	8
Escritura de informes	8

# Resumen

El Sistema Experto desarrollado es capaz de gestionar las tareas de administración de una oficina, desde asignar usuarios a empleados hasta controlar las luces que deben permanecer encendidas.

El sistema ha sido implementado en CLIPS y, mediante una serie de reglas y hechos con distinta prioridad, será capaz de llevar a cabo todas estas actividades. Según los hechos que aparezcan en su base de conocimiento, decidirá qué acción realizar. Muchos de estos hechos serán introducidos por sensores o por el propio personal de la oficina, de forma que el sistema sabrá lo que está ocurriendo en todo momento.

Aun así el conocimiento no siempre será perfecto, por lo que deberá aprender a trabajar con un ligero grado de incertidumbre.

## Proceso de desarrollo

### Representación y obtención del conocimiento

La representación del conocimiento debe ser lo más intuitiva posible, con hechos con nombres que expliquen claramente qué conocimiento representan. Se ha de intentar mantener por medio de estos hechos un conocimiento total de lo que ocurra en la oficina, y así poder actuar acorde a éste.

Para obtener el conocimiento, además de los datos proporcionados por sensores y personal, debemos observar la oficina e introducir manualmente aquello que creamos relevante para representar el conocimiento. Por ejemplo, será necesario definir los tipos de trámite que gestiona cada empleado, la oficina en la que trabaja, el estado inicial de las luces, etc. Más adelante explicaré detenidamente cada hecho que se utilice en el sistema.

### Procedimiento de validación y verificación

Para verificar el desarrollo he comprobado que la lógica seguida fuese completamente funcional y generase los resultados esperados. Para la validación, aunque debería someterse a un uso real de empleados, disponemos de la simulación de la oficina para comprobar que cumpla las necesidades y funcione adecuadamente.

## Descripción del sistema desarrollado

### Control de Calidad de Servicios

#### Empleados atendiendo

Para controlar cuántos empleados están atendiendo cada tipo de trámite, lo he representado con los hechos (**EmpleadosAtendiendo tipoTramite total**). Este hecho será inicializado a 0 en el deffacts tanto para el trámite general como el específico. Cada vez que a un empleado se le asigne un usuario (Regla **AsignarEmpleado**) el total de empleados atendiendo este tipo de trámite se aumentará en 1. Del mismo modo, en la

regla **RegistrarCaso** que se ejecuta cuando el trámite ha concluido, el contador se decrementará en 1.

Entonces, para comprobar el número de empleados atendiendo cada trámite (apartados **a** y **b**) disponemos de las dos siguientes reglas: **InformarMenosDeNEmpleadosTG** y **InformarNingunEmpleadoTE**.

Ambas comprobarán los hecho **EmpleadosAtendiendo** y en caso de de que se cumpla la condición\*, imprimirán un mensaje de alerta.

\* la condición para **InformarMenosDeNEmpleadosTG** es que el total sea menor que el indicado por el hecho **MinimoEmpleadosActivos**, por otro lado para el caso de la regla **InformarNingunEmpleadoTE** será suficiente con que el total sea igual a 0.

### Tiempo máximo de espera a ser atendido

Cada vez que se encola un usuario, se introduce el hecho (**TiempoLlegadaUsuario tipotramite id hora\_llegada**), la hora se representará en segundos para un mejor manejo de las operaciones.

Luego, tendremos la regla **ComprobarMaximoTiempoEspera** que se comprobará cada segundo (para ello mantendrá en sus precondiciones los hechos **hora\_actual**, **minutos\_actual** y **segundos\_actual**) y saltará si algún empleado lleva más tiempo esperando que el establecido en el hecho **MaximoEsperaParaSerAtendido**. El mensaje se mostrará sólo una vez, ya que se comprobará si ya ha sido emitido para no repetirlo (hecho **AvisoUsuarioEsperando tipoTramite id**).

Una vez el usuario sea asignado a un empleado (Regla **AsignarEmpleado**), se registrará cuánto tiempo ha estado esperando (regla **RegistrarEsperaUsuario**) bajo el hecho **EsperaUsuario tipoTramite tiempoTotal** y se eliminará **TiempoLlegadaUsuario**.

### Tiempo máximo gestión de trámite

El esquema para este aviso es muy similar al anterior. **TiempoInicioTramiteUsuario** almacenaría el tiempo de inicio en la regla **AsignarEmpleado**, y se comprobaría con la regla **ComprobarMaximoTiempoTramite** que no superase el límite, del mismo modo que en **ComprobarMaximoTiempoEspera**. Una vez el trámite acaba, se utilizará la regla **RegistrarTramiteUsuario** para guardar la información de tiempo relativa a éste.

## Control de rendimiento

Antes de comenzar a solventar las funciones que se nos proponen, es necesario crear las reglas necesarias para conocer el estado de un empleado según el hecho **Ficha empleado**. Estas reglas son las siguientes:

- **FichaAEntrar**: Se ejecutará la primera vez que un empleado fiche en el día y será la encargada de crear el hecho **EmpleadoTrabajando empleado estado hora**, donde el estado representa si está dentro (1) o fuera (0) de la oficina y la hora la última vez que

éste fichó. En este momento el estado se iniciará a 1, pues se entiende que el usuario ha llegado. También aumentará el total de empleados en 1.

- **Ficha:** Saltará el resto de veces que un empleado fiche. Se comprobará el estado de éste, si estaba fuera pasará a estar dentro y viceversa. El contador de empleados aumentará o decrementará según entre o salga.

### Retraso del empleado

Cuando un empleado llegue por primera vez (**FichaAlEntrar**) se lanzará la regla **ComprobarRetraso** que comprobará si el tiempo de llegada difiere en más de un **TiempoMaximoRetraso** de la hora del **ComienzoJornada**, emitiendo un mensaje indicando el retraso en caso afirmativo.

### Tiempo de descanso excedido

Cuando un empleado sale de la oficina, se introduce en la BC del sistema el hecho **ComprobacionDescanso empleado tiempo\_salida**, que se utilizará en la regla **TiempoDescansoSuperado** de forma muy similar a en las reglas **ComprobarMaximo-**del apartado anterior, donde se comprobará de forma reiterada si algún descanso supera el tiempo máximo de descanso.

Cuando el usuario vuelva, se invocará al método **EliminarComprobacionDescanso** que eliminará el hecho **ComprobacionDescanso**.

### Mínimo número de trámites

Este hecho se comprobará una vez haya acabado la jornada diaria. Por eso, el primer paso será crear la regla **acabarSistema** con una prioridad muy elevada que una vez superada la hora de cierre de la oficina, desencadenará la comprobación de mínimo de trámites por medio de la regla **MinimoTramites**. Esta regla únicamente mostrará por pantalla aquellos empleados cuyo hecho **TramitesGestionados empleado totalTramits totalTiempo** no supere el **MinimoTramitesPorDia tipoTramite**. Los trámites gestionados de cada empleado aumentarán dentro de la regla **RegistrarTramiteUsuario**.

### Situación del empleado

La BC siempre mantendrá información sobre la situación de cada uno de los empleados. El hecho **Estado empleado estado** será el encargado de almacenar esta información, y podrá almacenar 5 valores distintos. Al principio todos los empleados estarán inicializados a estado **NoHaLlegado**, y una vez que éste llegue (**FichaAlEntrar**) se cambiará el valor a **EnOficina**. Cuando se asigna un usuario al empleado (regla **AsignarEmpleado**) el estado se modificará a **Atendiendo**. Cuando éste indique su disponibilidad, (regla **RegistrarCaso**) se modificará a **Disponible** y cuando salga a descansar cambiará a **Descansando**. Por tanto la regla **Ficha** alternará el Estado del empleado entre los estados **EnOficina** y **Descansando**.

## Control Inteligente de las Luces

Para gestionar las luces de la oficina he tenido que definir unas reglas con prioridad muy elevada (más de 8000) ya que con una prioridad menor no siempre se ejecutaban (desconozco el motivo). De hecho, distintas ejecuciones sobre un mismo código provocaban resultados distintos.

### Luces Oficinas

Inicialmente, todas las luces estarán apagadas en el predicado (**Luz habitacion ON/OFF**) excepto la luz del pasillo y la de recepción. Además crearemos un hecho (**TotalPersonas habitacion numero**) con el que intentaremos estimar el número de personas dentro de una oficina.

En primer lugar, tenemos la regla **actualizarTiempoPasillo** que se ejecuta cada vez que el sensor de presencia de pasillo se active y actualiza el hecho (**TiempoUltimaVezPasillo hora**), que almacena la hora a la que se activó por última vez.

La regla **encenderLuzOficina** se ejecutará cuando se active el sensor de una puerta y la luz de la habitación esté apagada. Esto quiere decir que no había nadie dentro, por lo que acaba de entrar una persona y la luz debe encenderse.

La regla **aumentaPersonasHabitacion** ocurrirá cuando se active el sensor de una puerta y el del pasillo hace menos de 3 segundos que se activó. Normalmente esto significaría que una persona ha entrado, pero también cabe la posibilidad de que haya salido y el pasillo hubiese sido activado por una causa externa. Para curarnos en salud, aumentamos el contador de personas en la habitación (suponemos el primer caso).

La regla **saleOficina** se ejecutará cuando sea seguro que la persona ha salido de la oficina (se activa el sensor de la puerta y el pasillo hace más de 3 segundos que se activó). Decrementa el **TotalPersonas** en 1 y, si este se hace 0, apaga la luz de la habitación.

Por último, la regla **ForzarApagado**, que funciona sólo para las oficinas individuales (1-5), apagará la luz de la oficina cuando el empleado esté descansado (fuera de la oficina). Para la oficina doble habría que implementar una regla similar que comprobase si ambos están Descansando.

### Luces Aseos

Las luces de los aseos son gestionadas por dos reglas:

La primera **encenderLuzAseo** comprueba el sensor de presencia del aseo y si está activado enciende la luz (o la mantiene encendida) y almacena a qué hora se activó por última vez (**TiempoUltimaVez aseo hora**).

La segunda **apagarLuzAseo** comprueba si han pasado más de cinco minutos desde la última activación y apaga la luz en caso afirmativo.

## Situaciones anómalas

### Oficina vacía sin explicación

La primera vez que un empleado vuelve de Descansar, se supone que ha desayunado (sólo tienen un descanso al día). La regla **VuelveDeDesayunar** inserta el hecho **HaDesayunado empleado tiempo** cuando esto ocurre.

Posteriormente, la regla **OficinaVacíaSinJustificacion** comprobará si la luz de la habitación del empleado que volvió a desayunar hace más de 10 minutos está apagada y avisará, ya que no debería estar vacía.

### Más personas de lo esperado

La regla **MasPersonasEsperado** avisará si el **TotalPersonas** en una habitación es mayor a 2 personas, en caso afirmativo supondrá que hay más personas de la cuenta. Si el número de personas es exactamente 2, las reglas **ComprobarAsignado** y **ComprobarNoAsignado** comprobarán si el empleado está administrando un trámite y, por tanto, 2 personas es razonable o es más de lo esperado.

Esta regla sólo está pensada e implementada para las oficinas individuales, para la oficina doble habría que hacer más comprobaciones. Por ejemplo, más personas de lo esperado significaría más de 4. Si hay menos, habría que comprobar si alguno de los dos empleados está Descansando y cuál sería en consecuencia el número razonable de personas.

### Usuario no responde

Cuando el usuario es asignado a un empleado, la regla **UsuarioFueAsignado** añade el hecho **EsperandoAUsuario empleado tipoTramite usuario hora**. Cuando el empleado llegue a la oficina asignada, **UsuarioEntroEnOficina** eliminará esta espera. Si el usuario no responde a la llamada, cuando transcurran 4 minutos se informará de la situación y se pasará a comprobar si el usuario se ha equivocado de oficina.

### Usuario se ha equivocado

Con la información proporcionada por la regla anterior y los datos producidos por la regla **MasPersonasEsperado**, se podrá intuir si es que el usuario se había equivocado y entrado en una oficina equivocada. La regla **ComprobarEquivocacion** comprobará si esta equivocación ocurrió en el tiempo que se estaba esperando al usuario.

### Usuario en Gerencia o Papelería

Por último, si se activa el sensor de alguna de las puertas a estas habitaciones y las oficinas de los empleados que estén en trabajando no están vacías, se entiende que el que entró fue un usuario por error. Reglas: **EmpleadoEnGerenciaPapeleria**, **UsuarioEnGerenciaPapeleria** y **eliminarVisitaEmpleado**.

# Informes Generados

Tras acabar el sistema, se preparan los informes con estadísticas sobre la jornada laboral:

## Informe 1

Contiene el total de trámites y esperas ocurridas, clasificadas según el tipo de trámite y para los cuales se calculará la media y desviación típica del tiempo.

Una serie de reglas se encargarán de contar y sumar los hechos necesarios para calcular estos resultados (similar a las empleadas en la relación de ejercicios). Las reglas que se encargan de este proceso son:

- **contarEsperas/Tramites:** Cuenta el total de hechos EsperaUsuario / TramiteUsuario.
- **sumarEsperas/Tramites:** Suma los hechos contados antes y almacena el resultado en los hechos TotalTramites / TotalEsperas.
- **calcularDesviacionEsperas/Tramites:** Calcula la suma parcial de la fórmula de la desviación  $\sigma = \sqrt{\frac{1}{n-1} \sum (x_i - \bar{x})^2}$ .
- **sumarDesviacionEsperas/Tramites:** Realiza la sumatoria.

## Informe 2

Contiene las estadísticas de cada empleado, con el total de trámites gestionados, el tiempo total gestionando y la media de tiempo por trámite.

El hecho **TramitesGestionados empl total\_tramites total\_tiempo** que actualiza sus parámetros dentro de la regla **RegistrarTramiteUsuario** almacena esta información, por lo que sólo habrá que imprimirla en el informe.

## Escritura de informes

Con las reglas AbrirInforme, EscribirTiemposTotalesT/Esr, EscribirEstadisticasEmpleados, y CerrarInforme escribimos la información necesaria en cada informe. La escritura en ficheros se realiza tal y como viene explicada en los ficheros de apoyo de Decsai.

Para conocer más profundamente el funcionamiento, sugiero leer los comentarios incluidos en OficinalInteligente.clp.