



Asignación de TFGs mediante CLIPS:

Pablo Tarrío Otero

Conocimiento y Razonamiento Automático

Ingeniería Informática

Escuela Técnica Superior de Ingeniería

Universidad de Santiago de Compostela

Diciembre de 2021

ÍNDICE

1. Introducción	1
2. Cómo cargar el programa utilizando CLIPS	1
3. Plantillas definidas	1
4. Funcionamiento de las reglas.....	2
5. Base de hechos inicial	8
6. Funcionalidades extra	9
7. Experimentación y Análisis de resultados.....	11
8. Conclusiones.....	14
<hr/>	
Referencias.....	15

1. Introducción

Como todos los años, en la ETSE se asignan una gran cantidad de TFGs a los alumnos que lo solicitan. Es por ello que se ha pedido implementar un sistema basado en reglas en CLIPS para la asignación automática de los TFGs a los alumnos de la escuela.

Para la asignación de un TFG a un alumno, se tendrán en cuenta varios aspectos:

- Nota del expediente académico.
- Prioridad obtenida en una entrevista.
- Orden de propuestas elegidas por el alumno (que debe escoger tres de las opciones propuestas por la escuela).

En caso de producirse un empate en el expediente o en la prioridad de entrevista, se asignará el TFG al alumno que haya realizado la propuesta en primer lugar. Si todas las opciones escogidas por el alumno están ocupadas, se le asignará un TFG por descarte de los que se encuentren disponibles. Por último, si no quedan TFGs disponibles para un alumno, este se quedará sin tener un TFG asignado.

2. Cómo cargar el programa utilizando CLIPS

Para hacer funcionar el programa, simplemente habrá que cargar el archivo con formato *.bat* creado para esta práctica: *start_all.bat*. Este hará cargar el archivo que contiene las funciones, reglas y plantillas implementadas para el funcionamiento del programa (*sheets_rules.clp*), así como el archivo con los hechos del problema (*facts.clp*). Por último, cargará los hechos utilizando (*reset*), y ejecutará las reglas correspondientes con (*run*).

3. Plantillas definidas

Las plantillas las podemos encontrar definidas en el archivo de funciones, reglas y plantillas: *sheets_rules.clp*. En este archivo encontramos dos plantillas:

- **Alumno**
 - **Id:** DNI del alumno.
 - **Nombre:** Nombre del alumno.
 - **Expediente:** Nota media del alumno durante la carrera. Por defecto 0.
 - **Entrevista:** Prioridad obtenida durante la entrevista. Por defecto 0.
 - **Asignacion:** Propuesta asignada al alumno.
- **TFG**
 - **Id:** Número identificativo del TFG.
 - **NumeroAsignaciones:** Plazas disponibles para participar en el TFG (por defecto 1).
 - **Título:** Título del TFG.
 - **Tutoria:** Profesores encargados de dirigir el TFG.
 - **Descripcion:** Breve descripción del contenido del TFG.
 - **Requisitos:** Requisitos para poder realizar el TFG.
 - **Tipo:** Tipo del TFG: Puede ser A, B o C (por defecto A).

- **MetodoAsignacion:** Método de asignación del TFG; Puede ser Expediente o Entrevista (por defecto Expediente).
- **PlazasUsadas:** Plazas que han sido asignadas hasta el momento de este TFG.

Se muestra a continuación la implementación de ambas plantillas:

```
; PLANTILLAS
; Plantilla de Alumno
(deftemplate Alumno
  (field Id)
  (multifield Nombre)
  (field Expediente (default 0))
  (field Entrevista (default 0))
  (field Asignacion)
)

; Plantilla de TFG
(deftemplate TFG
  (field Id)
  (field NumeroAsignaciones (default 1))
  (multifield Titulo)
  (multifield Tutoria)
  (multifield Descripcion)
  (multifield Requisitos)
  (field Tipo (type SYMBOL) (allowed-symbols A B C) (default A))
  (field MetodoAsignacion (type SYMBOL) (allowed-symbols Expediente
Entrevista) (default Expediente))
  (field PlazasUsadas (default 0))
)
```

Código 1. Plantillas.

4. Funcionamiento de las reglas

Las reglas las podemos encontrar definidas en el archivo de funciones, reglas y plantillas: *sheets_rules.clp*. Si se desea observar el funcionamiento del código en detalle, se puede consultar en dicho archivo, mientras que en esta memoria se mostrará un resumen de las 12 reglas existentes. Todas estas tienen declarada una prioridad para que CLIPS las ejecute en el orden correcto, pues en caso contrario el programa puede llegar a fallar en alguna ocasión.

Las 9 primeras reglas se pueden agrupar en 3 conjuntos: primera opción, segunda opción y tercera opción (según la preferencia en las elecciones de propuestas del alumno). En cada uno de los conjuntos encontramos tres tipos de reglas diferentes, las cuales podemos resumir de la siguiente forma:

- **Asignación de TFG según expediente.** En caso de que el TFG ya hubiese sido asignado previamente, habrá que comprobar la nota de expediente del alumno al que se le había asignado con el alumno actual. En caso de tener una nota mayor, se ha de asignar el TFG al nuevo alumno, y buscar un nuevo TFG para el alumno anterior.

- **Asignación de TFG según entrevista.** En caso de que el TFG ya hubiese sido asignado previamente, habrá que comprobar la prioridad de la entrevista del alumno al que se le había asignado con la prioridad del alumno actual. En caso de tener una prioridad mayor, se ha de asignar el TFG al nuevo alumno, y buscar un nuevo TFG para el alumno anterior.
- **Asignación de TFG de forma libre.** En caso de que el TFG no haya sido asignado todavía a nadie, se asignará al alumno directamente.

Las nueve reglas mencionadas anteriormente reciben los siguientes nombres en el documento de reglas: *Asignacion_Opcion_1_Expediente*, *Asignacion_Opcion_1_Entrevista*, *Asignacion_Opcion_1_Sin_Criterio*, *Asignacion_Opcion_2_Expediente*, *Asignacion_Opcion_2_Entrevista*, *Asignacion_Opcion_2_Sin_Criterio*, *Asignacion_Opcion_3_Expediente*, *Asignacion_Opcion_3_Entrevista*, *Asignacion_Opcion_3_Sin_Criterio*.

A continuación de las nuevas reglas, que conforman el funcionamiento principal del programa, encontramos otras tres “secundarias”. La primera de ellas (*Asignacion_Ninguna_Opcion_Disponible*) se encarga de asignar el TFG a los alumnos cuyas tres opciones ya hayan sido escogidas por otros alumnos con mejor expediente o mayor prioridad en la entrevista. Son asignaciones “por descarte”.

La siguiente regla (*Sin_TFG*) se encarga de indicar que un alumno se ha quedado sin TFG debido a que no quedan propuestas disponibles (todas las propuestas han sido ya asignadas).

Por último, una regla que elimina todos aquellos hechos de asignación auxiliares utilizados durante el programa (*Eliminar_Asignacion*).

Se muestra a continuación la implementación en CLIPS de las reglas existentes:

```
; REGLAS
; Asignacion de TFG pero el alumno tiene mas nota de expediente que
; otro (con opcion 1)
(defrule Asignacion_Opcion_1_Expediente
  (declare (salience 7))
  ?a <- (Alumno (Id ?dni) (Expediente ?exp) (Asignacion nil))
  (Propuesta ?dni ?op1 ?op2 ?op3)
  ?t <- (TFG (Id ?op1) (NumeroAsignaciones ?plazas) (PlazasUsadas
?usadas) (MetodoAsignacion Expediente))
  (test (eq ?plazas ?usadas))
  ?asig <- (Asignacion ?otroAlumno ?opOtro)
  (test (eq ?opOtro ?op1))
  ?otro <- (Alumno (Id ?otroAlumno) (Expediente ?expOtroAlumno))
  (test (> ?exp ?expOtroAlumno))
  =>
  (retract ?asig)
  (assert (Asignacion ?dni ?op1))
  (modify ?a (Asignacion ?op1))
  (modify ?otro (Asignacion nil))
  (printout t "(-) Retirado TFG " ?op1 " a " ?otroAlumno ". Motivo:
Expediente de otro alumno mejor." crlf)
  (printout t "(+) Asignado TFG " ?op1 " a " ?dni ". Es su opcion 1."
crlf crlf)
```

```

)

; Asignacion de TFG pero el alumno tiene mas nota de entrevista que
otro (con opcion 1)
(defrule Asignacion_Opcion_1_Entrevista
  (declare (salience 7))
  ?a <- (Alumno (Id ?dni) (Entrevista ?ent) (Asignacion nil))
  (Propuesta ?dni ?op1 ?op2 ?op3)
  ?t <- (TFG (Id ?op1) (NumeroAsignaciones ?plazas) (PlazasUsadas
?usadas) (MetodoAsignacion Entrevista))
  (test (eq ?plazas ?usadas))
  ?asig <- (Asignacion ?otroAlumno ?opOtro)
  (test (eq ?opOtro ?op1))
  ?otro <- (Alumno (Id ?otroAlumno) (Entrevista ?entOtroAlumno))
  (test (> ?ent ?entOtroAlumno))
  =>
  (retract ?asig)
  (assert (Asignacion ?dni ?op1))
  (modify ?a (Asignacion ?op1))
  (modify ?otro (Asignacion nil))
  (printout t "(-) Retirado TFG " ?op1 " a " ?otroAlumno ". Motivo:
Entrevista de otro alumno mejor." crlf)
  (printout t "(+) Asignado TFG " ?op1 " a " ?dni ". Es su opcion 1."
crlf crlf)
)

; Asignacion de TFG opcion 1 al alumno sin tener en cuenta notas
(defrule Asignacion_Opcion_1_Sin_Criterio
  (declare (salience 6))
  ?a <- (Alumno (Id ?dni) (Asignacion nil))
  (Propuesta ?dni ?op1 ?op2 ?op3)
  ?t <- (TFG (Id ?op1) (NumeroAsignaciones ?plazas) (PlazasUsadas
?usadas))
  (test (neq ?plazas usadas))
  =>
  (assert (Asignacion ?dni ?op1))
  (modify ?a (Asignacion ?op1))
  (modify ?t (PlazasUsadas (+ ?usadas 1)))
  (printout t "(+) Asignado TFG " ?op1 " a " ?dni ". Es su opcion 1."
crlf crlf)
)

; Asignacion de TFG pero el alumno tiene mas nota de expediente que
otro (con opcion 2)
(defrule Asignacion_Opcion_2_Expediente
  (declare (salience 5))
  ?a <- (Alumno (Id ?dni) (Expediente ?exp) (Asignacion nil))
  (Propuesta ?dni ?op1 ?op2 ?op3)

```

```

    ?t <- (TFG (Id ?op2) (NumeroAsignaciones ?plazas) (PlazasUsadas
?usadas) (MetodoAsignacion Expediente))
    (test (eq ?plazas ?usadas))
    ?asig <- (Asignacion ?otroAlumno ?opOtro)
    (test (eq ?opOtro ?op2))
    ?otro <- (Alumno (Id ?otroAlumno) (Expediente ?expOtroAlumno))
    (test (> ?exp ?expOtroAlumno))
    =>
    (retract ?asig)
    (assert (Asignacion ?dni ?op2))
    (modify ?a (Asignacion ?op2))
    (modify ?otro (Asignacion nil))
    (printout t "(-) Retirado TFG " ?op2 " a " ?otroAlumno ". Motivo:
Expediente de otro alumno mejor." crlf)
    (printout t "(+) Asignado TFG " ?op2 " a " ?dni ". Es su opcion 1."
crlf crlf)
)

; Asignacion de TFG pero el alumno tiene mas nota de entrevista que
otro (con opcion 2)
(defrule Asignacion_Opcion_2_Entrevista
  (declare (salience 5))
  ?a <- (Alumno (Id ?dni) (Entrevista ?ent) (Asignacion nil))
  (Propuesta ?dni ?op1 ?op2 ?op3)
  ?t <- (TFG (Id ?op2) (NumeroAsignaciones ?plazas) (PlazasUsadas
?usadas) (MetodoAsignacion Entrevista))
  (test (eq ?plazas ?usadas))
  ?asig <- (Asignacion ?otroAlumno ?opOtro)
  (test (eq ?opOtro ?op2))
  ?otro <- (Alumno (Id ?otroAlumno) (Entrevista ?entOtroAlumno))
  (test (> ?ent ?entOtroAlumno))
  =>
  (retract ?asig)
  (assert (Asignacion ?dni ?op1))
  (modify ?a (Asignacion ?op2))
  (modify ?otro (Asignacion nil))
  (printout t "(-) Retirado TFG " ?op2 " a " ?otroAlumno ". Motivo:
Entrevista de otro alumno mejor." crlf)
  (printout t "(+) Asignado TFG " ?op2 " a " ?dni ". Es su opcion 1."
crlf crlf)
)

; Asignacion de TFG opcion 2 al alumno sin tener en cuenta notas
(defrule Asignacion_Opcion_2_Sin_Criterio
  (declare (salience 4))
  ?a <- (Alumno (Id ?dni) (Asignacion nil))
  (Propuesta ?dni ?op1 ?op2 ?op3)

```

```

    ?t <- (TFG (Id ?op2) (NumeroAsignaciones ?plazas) (PlazasUsadas
?usadas))
    (test (neq ?plazas usadas))
    =>
    (assert (Asignacion ?dni ?op2))
    (modify ?a (Asignacion ?op2))
    (modify ?t (PlazasUsadas (+ ?usadas 1)))
    (printout t "(+) Asignado TFG " ?op2 " a " ?dni ". Es su opcion 1."
crlf crlf)
)

; Asignacion de TFG pero el alumno tiene mas nota de expediente que
otro (con opcion 3)
(defrule Asignacion_Opcion_3_Expediente
  (declare (salience 3))
  ?a <- (Alumno (Id ?dni) (Expediente ?exp) (Asignacion nil))
  (Propuesta ?dni ?op1 ?op2 ?op3)
  ?t <- (TFG (Id ?op3) (NumeroAsignaciones ?plazas) (PlazasUsadas
?usadas) (MetodoAsignacion Expediente))
  (test (eq ?plazas ?usadas))
  ?asig <- (Asignacion ?otroAlumno ?opOtro)
  (test (eq ?opOtro ?op3))
  ?otro <- (Alumno (Id ?otroAlumno) (Expediente ?expOtroAlumno))
  (test (> ?exp ?expOtroAlumno))
  =>
  (retract ?asig)
  (assert (Asignacion ?dni ?op3))
  (modify ?a (Asignacion ?op3))
  (modify ?otro (Asignacion nil))
  (printout t "(-) Retirado TFG " ?op3 " a " ?otroAlumno ". Motivo:
Expediente de otro alumno mejor." crlf)
  (printout t "(+) Asignado TFG " ?op3 " a " ?dni ". Es su opcion 3."
crlf crlf)
)

; Asignacion de TFG pero el alumno tiene mas nota de entrevista que
otro (con opcion 3)
(defrule Asignacion_Opcion_3_Entrevista
  (declare (salience 3))
  ?a <- (Alumno (Id ?dni) (Entrevista ?ent) (Asignacion nil))
  (Propuesta ?dni ?op1 ?op2 ?op3)
  ?t <- (TFG (Id ?op3) (NumeroAsignaciones ?plazas) (PlazasUsadas
?usadas) (MetodoAsignacion Entrevista))
  (test (eq ?plazas ?usadas))
  ?asig <- (Asignacion ?otroAlumno ?opOtro)
  (test (eq ?opOtro ?op3))
  ?otro <- (Alumno (Id ?otroAlumno) (Entrevista ?entOtroAlumno))
  (test (> ?ent ?entOtroAlumno))

```



```

=>
(retract ?asig)
(assert (Asignacion ?dni ?op3))
(modify ?a (Asignacion ?op3))
(modify ?otro (Asignacion nil))
(printout t "(-) Retirado TFG " ?op3 " a " ?otroAlumno ". Motivo:
Entrevista de otro alumno mejor." crlf)
(printout t "(+) Asignado TFG " ?op3 " a " ?dni ". Es su opcion 3."
crlf crlf)
)

; Asignacion de TFG opcion 3 al alumno sin tener en cuenta notas
(defrule Asignacion_Opcion_3_Sin_Criterio
  (declare (salience 2))
  ?a <- (Alumno (Id ?dni) (Asignacion nil))
  (Propuesta ?dni ?op1 ?op2 ?op3)
  ?t <- (TFG (Id ?op3) (NumeroAsignaciones ?plazas) (PlazasUsadas
?usadas))
  (test (neq ?plazas ?usadas))
  =>
  (assert (Asignacion ?dni ?op3))
  (modify ?a (Asignacion ?op3))
  (modify ?t (PlazasUsadas (+ ?usadas 1)))
  (printout t "(+) Asignado TFG " ?op3 " a " ?dni ". Es su opcion 3."
crlf crlf)
)

; Asignacion de TFG a aquellos alumnos cuya propuesta no puede ser
aceptada porque no cumplen suficiente nota de expediente o entrevista
; En este caso se asigna por orden de llegada de la propuesta (no se
tienen en cuenta las notas)
(defrule Asignacion_Ninguna_Opcion_Disponible
  (declare (salience 1))
  ?a <- (Alumno (Id ?dni) (Asignacion nil))
  ?t <- (TFG (Id ?op) (NumeroAsignaciones ?plazas) (PlazasUsadas
?usadas))
  (test (neq ?plazas ?usadas))
  =>
  (assert (Asignacion ?dni ?op))
  (modify ?a (Asignacion ?op))
  (modify ?t (PlazasUsadas (+ ?usadas 1)))
  (printout t "(+) Asignado TFG " ?op " a " ?dni " por descarte."
crlf crlf)
)

; Si no hay suficientes propuestas, habrá alumnos que queden sin TFG
asignado
(defrule Sin_TFG

```

```

(declare (salience 0))
?a <- (Alumno (Id ?dni) (Asignacion nil))
=>
(printout t "## El alumno " ?dni " se ha quedado sin TFG por no
existir suficientes propuestas. ##" crlf crlf)
)

; Cuando no existan mas alumnos a los que asignar TFGs, eliminamos los
hechos auxiliares utilizados
(defrule Eliminar_Asignacion
(declare (salience -1))
?a <- (Asignacion ? ?)
=>
(retract ?a)
)

```

Código 2. Reglas.

5. Base de hechos inicial

La base de hecho inicial la podemos encontrar en el archivo de datos: *facts.clp*. En primer lugar, se encuentran definidos los alumnos, a continuación, los TFGs disponibles, y, por último, las propuestas realizadas por los alumnos existentes.

Los hechos correspondientes a alumnos y TFGs, se corresponden con las plantillas previamente definidas para estos. En cuanto a las propuestas, su primer parámetro es el identificador del alumno (DNI), y los siguientes tres parámetros son las propuestas (ids de los TFGs) escogidas por el alumno por orden de preferencia.

Se muestra a continuación la base de hechos utilizada durante las pruebas:

```

; BASE DE HECHOS INICIAL
(deffacts Datos
; Alumnos
(Alumno (Id 00000000A)(Nombre Pablo)(Expediente 6)(Entrevista 4))
(Alumno (Id 00000000B)(Nombre Carlos)(Expediente 2)(Entrevista 4))
(Alumno (Id 00000000C)(Nombre Laura)(Expediente 7)(Entrevista 4))
(Alumno (Id 00000000D)(Nombre Carla)(Expediente 3)(Entrevista 4))
(Alumno (Id 00000000E)(Nombre Pedro)(Expediente 1)(Entrevista 4))
(Alumno (Id 00000000F)(Nombre Daniel)(Expediente 9)(Entrevista 4))
(Alumno (Id 00000000G)(Nombre Celia)(Expediente 9)(Entrevista 4))
(Alumno (Id 00000000H)(Nombre Valeria)(Expediente 9)(Entrevista 4))
(Alumno (Id 00000000I)(Nombre Marcos)(Expediente 9)(Entrevista 10))
(Alumno (Id 00000000L)(Nombre Natalia)(Expediente 9)(Entrevista 4))

; TFGs
(TFG (Id 1) (NumeroAsignaciones 1))
(TFG (Id 2) (NumeroAsignaciones 2))
(TFG (Id 3) (NumeroAsignaciones 1))
(TFG (Id 4) (NumeroAsignaciones 1))

```

```
(TFG (Id 5) (NumeroAsignaciones 1) (MetodoAsignacion Entrevista))
(TFG (Id 6) (NumeroAsignaciones 1) (MetodoAsignacion Entrevista))
(TFG (Id 7) (NumeroAsignaciones 1) (MetodoAsignacion Entrevista))
```

```
; Propuestas
```

```
(Propuesta 00000000A 1 2 3)
(Propuesta 00000000B 1 2 3)
(Propuesta 00000000C 1 2 3)
(Propuesta 00000000D 1 2 3)
(Propuesta 00000000E 1 2 3)
(Propuesta 00000000F 1 2 3)
(Propuesta 00000000G 5 6 1)
(Propuesta 00000000H 5 4 6)
(Propuesta 00000000I 5 6 4)
(Propuesta 00000000L 1 2 3)
```

```
)
```

Código 3. Base de hechos.

Esta base de hechos cuenta con un número suficiente, pero no escaso, de alumnos, TFG y propuestas de alumnos que nos permitirán probar un gran número de casos que se pueden dar en la asignación real de TFGs. Existen coincidencias entre propuestas que nos forzarán a hacer reasignaciones y elegir segundas o terceras opciones, buscas asignaciones por descarte, e incluso algún alumno quedará sin asignación. Lo veremos en el siguiente apartado.

6. Funcionalidades extra

Se ha realizado la implementación de dos funciones que permiten la introducción de hechos mediante la lectura de teclado. La primera de ellas se encarga de añadir nuevos alumnos, junto con las propuestas en su orden de preferencia. Para poder ejecutar esta función, es necesario ejecutar *(alumno-propuesta)* en la ventana de diálogo de CLIPS. Lo mismo se ha implementado para la inserción por teclado de nuevas propuestas de TFG, en cuyo caso se debe hacer un llamamiento al comando *(TFG)*. Se muestra el código de las funciones a continuación:

```
; FUNCIONES
; Funcion que permite introducir un alumno a la vez que su preferencia
de propuestas en la base de hechos
(defun alumno-propuesta ()
  ; Solicitud y almacenamiento de los datos en variables (lectura de
  teclado)
  (printout t "Introduzca el id del alumno: ")
  (bind ?id (read))
  (printout t "Introduzca el nombre del alumno: ")
  (bind ?nombre (read))
  (printout t "Introduzca el expediente del alumno: ")
  (bind ?expediente (read))
  (printout t "Introduzca la prioridad en entrevista del alumno: ")
  (bind ?entrevista (read))
```

```

    (printout t "Introduzca la primera opcion de propuesta del alumno:
")
    (bind ?op1 (read))
    (printout t "Introduzca la segunda opcion de propuesta del alumno:
")
    (bind ?op2 (read))
    (printout t "Introduzca la tercera opcion de propuesta del alumno:
")
    (bind ?op3 (read))

    ; Introduccion de los hechos en la base de hechos
    (assert (Alumno (Id ?id) (Nombre ?nombre) (Expediente ?expediente)
(Entrevista ?entrevista)))
    (assert (Propuesta ?id ?op1 ?op2 ?op3))
    (return TRUE)
)

; Funcion que permite introducir nuevos TFGs en la base de hechos
(defun TFG ()
    ; Solicitud y almacenamiento de los datos en variables (lectura de
teclado)
    (printout t "Introduzca el id del TFG: ")
    (bind ?id (read))
    (printout t "Introduzca el numero de plazas disponibles para el
TFG: ")
    (bind ?numeroPlazas (read))
    (printout t "Introduzca el titulo del TFG: ")
    (bind ?titulo (read))
    (printout t "Introduzca la tutoria del TFG: ")
    (bind ?tutoria (read))
    (printout t "Introduzca la descripcion del TFG: ")
    (bind ?descripcion (read))
    (printout t "Introduzca los requisitos para poder realizar el TFG:
")
    (bind ?requisitos (read))
    (printout t "Introduzca el tipo de TFG (A, B, C): ")
    (bind ?tipo (read))
    (printout t "Introduzca el metodo de asignacion del TFG
(Expediente, Entrevista): ")
    (bind ?metodoAsignacion (read))

    ; Introduccion del hecho en la base de hechos
    (assert
        (TFG (Id ?id) (NumeroAsignaciones ?numeroPlazas) (Titulo
?titulo) (Tutoria ?tutoria) (Descripcion ?descripcion) (Requisitos
?requisitos) (Tipo ?tipo) (MetodoAsignacion ?metodoAsignacion)
        )
    )
)

```

```
(return TRUE)
)
```

Código 4. Funciones que permiten la inserción de nuevos alumnos, propuestas y TFGs.

7. Experimentación y Análisis de resultados

En primer lugar, encontramos una base de hechos inicial con todas las propuestas, alumnos sin asignación y TFGs:

```
f-0 (initial-fact)
f-1 (Alumno (Id 00000000A) (Nombre Pablo) (Expediente 6) (Entrevista 4) (Asignacion nil))
f-2 (Alumno (Id 00000000B) (Nombre Carlos) (Expediente 2) (Entrevista 8) (Asignacion nil))
f-3 (Alumno (Id 00000000C) (Nombre Laura) (Expediente 7) (Entrevista 4) (Asignacion nil))
f-4 (Alumno (Id 00000000D) (Nombre Carla) (Expediente 3) (Entrevista 4) (Asignacion nil))
f-5 (Alumno (Id 00000000E) (Nombre Pedro) (Expediente 1) (Entrevista 4) (Asignacion nil))
f-6 (Alumno (Id 00000000F) (Nombre Daniel) (Expediente 9) (Entrevista 10) (Asignacion nil))
f-7 (Alumno (Id 00000000G) (Nombre Celia) (Expediente 9) (Entrevista 4) (Asignacion nil))
f-8 (Alumno (Id 00000000H) (Nombre Valeria) (Expediente 9) (Entrevista 4) (Asignacion nil))
f-9 (Alumno (Id 00000000I) (Nombre Marcos) (Expediente 9) (Entrevista 10) (Asignacion nil))
f-10 (Alumno (Id 00000000L) (Nombre Natalia) (Expediente 9) (Entrevista 4) (Asignacion nil))
f-11 (TFG (Id 1) (NumeroAsignaciones 1) (Titulo) (Tutoria) (Descripcion) (Requisitos) (Tipo A) (MetodoAsignacion Expediente) (PlazasUsadas 0))
f-12 (TFG (Id 2) (NumeroAsignaciones 2) (Titulo) (Tutoria) (Descripcion) (Requisitos) (Tipo A) (MetodoAsignacion Expediente) (PlazasUsadas 0))
f-13 (TFG (Id 3) (NumeroAsignaciones 1) (Titulo) (Tutoria) (Descripcion) (Requisitos) (Tipo A) (MetodoAsignacion Expediente) (PlazasUsadas 0))
f-14 (TFG (Id 4) (NumeroAsignaciones 1) (Titulo) (Tutoria) (Descripcion) (Requisitos) (Tipo A) (MetodoAsignacion Expediente) (PlazasUsadas 0))
f-15 (TFG (Id 5) (NumeroAsignaciones 1) (Titulo) (Tutoria) (Descripcion) (Requisitos) (Tipo A) (MetodoAsignacion Entrevista) (PlazasUsadas 0))
f-16 (TFG (Id 6) (NumeroAsignaciones 1) (Titulo) (Tutoria) (Descripcion) (Requisitos) (Tipo A) (MetodoAsignacion Entrevista) (PlazasUsadas 0))
f-17 (TFG (Id 7) (NumeroAsignaciones 1) (Titulo) (Tutoria) (Descripcion) (Requisitos) (Tipo A) (MetodoAsignacion Entrevista) (PlazasUsadas 0))
f-18 (TFG (Id 8) (NumeroAsignaciones 1) (Titulo) (Tutoria) (Descripcion) (Requisitos) (Tipo A) (MetodoAsignacion Entrevista) (PlazasUsadas 0))
f-19 (Propuesta 00000000A 1 2 3)
f-20 (Propuesta 00000000B 8 1 3)
f-21 (Propuesta 00000000C 1 2 3)
f-22 (Propuesta 00000000D 1 2 3)
f-23 (Propuesta 00000000E 8 2 3)
f-24 (Propuesta 00000000F 8 2 3)
f-25 (Propuesta 00000000G 5 6 1)
f-26 (Propuesta 00000000H 8 4 6)
f-27 (Propuesta 00000000I 5 6 4)
f-28 (Propuesta 00000000L 1 2 3)
```

Ilustración 1. Base de hechos inicial.

A continuación, se comienzan a asignar los TFGs a los alumnos. El primer alumno en tener un TFG asignado ha sido 00000000L. Como no se le había asignado a nadie ningún TFG todavía, se le podrá asignar su primera opción sin problemas:

(+) Asignado TFG 1 a 00000000L. Es su opcion 1.

El siguiente TFG que se asignará será al alumno 00000000I. En este caso el TFG asignado ha sido el 5, y, al igual que en el caso anterior, se asigna la primera opción directamente:

(+) Asignado TFG 5 a 00000000I. Es su opcion 1.

Lo mismo ocurre en el caso del alumno 00000000H y el TFG 8:

(+) Asignado TFG 8 a 00000000H. Es su opcion 1.

Seguido de esto nos encontramos un proceso de reasignación de TFG. Este es debido a que la prioridad en la entrevista de 00000000B es mayor que la de 00000000H. En este caso, debemos asignar el TFG al alumno 00000000B con su primera opción (TFG 8).

(-) Retirado TFG 8 a 00000000H. Motivo: Entrevista de otro alumno mejor.

(+) Asignado TFG 8 a 00000000B. Es su opcion 1.

El siguiente TFG que se debe asignar debe ser al alumno 00000000F. La situación que ocurre es la misma que en el caso anterior, pero esta vez le ocurre al alumno 00000000B.

(-) Retirado TFG 8 a 00000000B. Motivo: Entrevista de otro alumno mejor.

(+) Asignado TFG 8 a 00000000F. Es su opcion 1.

Los siguientes casos asignan segundas opciones a los alumnos, debido a que o bien la nota de entrevista es menor, o bien el expediente es menor (con respecto al alumno que ya tiene asignado el TFG correspondiente a su primera opción). En el caso del TFG 2 se asigna en dos ocasiones porque tiene 2 plazas disponibles.

(+) Asignado TFG 4 a 00000000H. Es su opcion 2.

(+) Asignado TFG 6 a 00000000G. Es su opcion 2.

(+) Asignado TFG 2 a 00000000E. Es su opcion 2.

(+) Asignado TFG 2 a 00000000A. Es su opcion 2.

El siguiente caso que se produce es semejante al caso de una mayor prioridad de entrevista, pero en este caso se aplica a un mayor expediente. El expediente de 00000000C es mayor que el de 00000000A, con lo cual se debe reasignar el TFG.

(-) Retirado TFG 2 a 00000000A. Motivo: Expediente de otro alumno mejor.

(+) Asignado TFG 2 a 00000000C. Es su opcion 2.

Y lo mismo ocurre ahora entre 00000000A y 00000000E (mayor expediente de 00000000A con respecto a 00000000E).

(-) Retirado TFG 2 a 00000000E. Motivo: Expediente de otro alumno mejor.

(+) Asignado TFG 2 a 00000000A. Es su opcion 2.

Asignamos ahora un TFG a 00000000E, el cual es su tercera propuesta elegida.

(+) Asignado TFG 3 a 00000000E. Es su opcion 3.

Debido a que el resto de las propuestas ya han sido asignadas, y que el expediente o prioridad de entrevista de 00000000B no le permite escoger una propuesta según sus preferencias, a este se le ha asignado un TFG por descarte.

(+) Asignado TFG 7 a 00000000B por descarte.

Por último, nos encontramos un alumno cuya nota en el expediente y la entrevista es muy bajas. Debido a esto, deberá esperar a que se publiquen nuevas propuestas de TFG, pues de momento todas las propuestas ya han sido asignadas a alumnos con mejor expediente o entrevista.

El alumno 00000000D se ha quedado sin TFG por no existir suficientes propuestas.

Por último, se insertará por medio de teclado un nuevo alumno junto con su propuesta, así como un nuevo TFG. En primer lugar, se insertará al nuevo alumno. Para ello, debemos ejecutar (*alumno-propuesta*) en CLIPS, del cual se nos solicitará su DNI (01234567A), el nombre (Pepe), su expediente (2.5), su prioridad en entrevista (1), y las tres opciones de propuesta que este desea (1, 2, 3). Podemos verlo en la siguiente imagen:

```
CLIPS> (alumno-propuesta)
Introduzca el DNI del alumno: 01234567A
Introduzca el nombre del alumno: Pepe
Introduzca el expediente del alumno: 2.5
Introduzca la prioridad en entrevista del alumno: 1
Introduzca la primera opcion de propuesta del alumno: 1
Introduzca la segunda opcion de propuesta del alumno: 2
Introduzca la tercera opcion de propuesta del alumno: 3
TRUE
```

Ilustración 2. Inserción de un alumno por teclado.

La introducción de este hecho produce la activación de la regla *Sin_TFG* (recordemos que nos habíamos quedado sin propuestas de TFG disponibles en el paso anterior. Con lo cual, se nos indica.

El alumno 01234567A se ha quedado sin TFG por no existir suficientes propuestas.

Ahora insertaremos un nuevo TFG, también por medio de teclado, y veremos qué ocurre. En este caso para el TFG se ha introducido un id (10), el número de plazas disponibles (1), el título del TFG (Titulo de TFG), la tutoría del TFG (Profesor), la descripción (Descripcion de TFG), requisitos (-), el tipo de TFG (B) y el método de asignación (Expediente).

```
CLIPS> (TFG)
Introduzca el id del TFG: 10
Introduzca el numero de plazas disponibles para el TFG: 1
Introduzca el titulo del TFG: Titulo de TFG
Introduzca la tutoria del TFG: Profesor
Introduzca la descripcion del TFG: Descripcion de TFG
Introduzca los requisitos para poder realizar el TFG: -
Introduzca el tipo de TFG (A, B, C): B
Introduzca el metodo de asignacion del TFG (Expediente, Entrevista): Expediente
TRUE
```

Ilustración 3. Inserción de un TFG por teclado.

Hasta el momento tenemos dos alumnos sin un TFG asignado, que son el alumno *01234567A* y *00000000D*. Estos dos alumnos competirán ahora por la nueva propuesta de TFG introducida, pero el que se llevará en este caso el TFG es *00000000D*, pues tiene una mayor nota de expediente (3 vs 2.5) y, además, ha presentado su solicitud antes que *01234567A*. Por tanto:

(+) Asignado TFG 10 a 00000000D por descarte.

Tras la ejecución de todas las reglas, obtenemos la base de hechos final, con todos los alumnos que tienen respuesta sobre su asignación al TFG (la cual aparece marcada en el atributo *Asignacion* del *Alumno*), así como el número de plazas que han sido cubiertas para cada TFG (en este caso todos los TFGs están completos).

En la base de hechos final podremos ver que existe un alumno que se ha quedado sin asignación (*01234567A*), las diferentes propuestas realizadas por los alumnos, así como los TFGs propuestos y los alumnos que tienen una asignación presente. Todo ello lo podemos observar en la siguiente ilustración:

```
f-0 (initial-fact)
f-19 (Propuesta 00000000A 1 2 3)
f-20 (Propuesta 00000000B 8 1 3)
f-21 (Propuesta 00000000C 1 2 3)
f-22 (Propuesta 00000000D 1 2 3)
f-23 (Propuesta 00000000E 8 2 3)
f-24 (Propuesta 00000000F 8 2 3)
f-25 (Propuesta 00000000G 5 6 1)
f-26 (Propuesta 00000000H 8 4 6)
f-27 (Propuesta 00000000I 5 6 4)
f-28 (Propuesta 00000000L 1 2 3)
f-30 (Alumno (Id 00000000L) (Nombre Natalia) (Expediente 9) (Entrevista 4) (Asignacion 1))
f-31 (TFG (Id 1) (NumeroAsignaciones 1) (Titulo) (Tutoria) (Descripcion) (Requisitos) (Tipo A) (MetodoAsignacion Expediente) (PlazasUsadas 1))
f-33 (Alumno (Id 00000000I) (Nombre Marcos) (Expediente 9) (Entrevista 10) (Asignacion 5))
f-34 (TFG (Id 3) (NumeroAsignaciones 1) (Titulo) (Tutoria) (Descripcion) (Requisitos) (Tipo A) (MetodoAsignacion Entrevista) (PlazasUsadas 1))
f-37 (TFG (Id 5) (NumeroAsignaciones 1) (Titulo) (Tutoria) (Descripcion) (Requisitos) (Tipo A) (MetodoAsignacion Entrevista) (PlazasUsadas 1))
f-42 (Alumno (Id 00000000F) (Nombre Daniel) (Expediente 5) (Entrevista 10) (Asignacion 8))
f-45 (Alumno (Id 00000000H) (Nombre Valeria) (Expediente 9) (Entrevista 4) (Asignacion 4))
f-46 (TFG (Id 4) (NumeroAsignaciones 1) (Titulo) (Tutoria) (Descripcion) (Requisitos) (Tipo A) (MetodoAsignacion Expediente) (PlazasUsadas 1))
f-48 (Alumno (Id 00000000G) (Nombre Celia) (Expediente 9) (Entrevista 4) (Asignacion 6))
f-49 (TFG (Id 6) (NumeroAsignaciones 1) (Titulo) (Tutoria) (Descripcion) (Requisitos) (Tipo A) (MetodoAsignacion Entrevista) (PlazasUsadas 1))
f-55 (TFG (Id 2) (NumeroAsignaciones 2) (Titulo) (Tutoria) (Descripcion) (Requisitos) (Tipo A) (MetodoAsignacion Expediente) (PlazasUsadas 2))
f-57 (Alumno (Id 00000000C) (Nombre Laura) (Expediente 7) (Entrevista 4) (Asignacion 2))
f-60 (Alumno (Id 00000000A) (Nombre Pablo) (Expediente 6) (Entrevista 4) (Asignacion 3))
f-63 (Alumno (Id 00000000E) (Nombre Pedro) (Expediente 3) (Entrevista 4) (Asignacion 3))
f-64 (TFG (Id 3) (NumeroAsignaciones 1) (Titulo) (Tutoria) (Descripcion) (Requisitos) (Tipo A) (MetodoAsignacion Expediente) (PlazasUsadas 1))
f-66 (Alumno (Id 00000000B) (Nombre Carlos) (Expediente 3) (Entrevista 8) (Asignacion 7))
f-67 (TFG (Id 7) (NumeroAsignaciones 1) (Titulo) (Tutoria) (Descripcion) (Requisitos) (Tipo A) (MetodoAsignacion Entrevista) (PlazasUsadas 1))
f-68 (Alumno (Id 01234567A) (Nombre Pepe) (Expediente 2.5) (Entrevista 1) (Asignacion nil))
f-69 (Propuesta 01234567A 1 2 3)
f-72 (Alumno (Id 00000000D) (Nombre Carla) (Expediente 3) (Entrevista 4) (Asignacion 10))
f-73 (TFG (Id 10) (NumeroAsignaciones 1) (Titulo Titulo) (Tutoria Profesor) (Descripcion Descripcion) (Requisitos -) (Tipo B) (MetodoAsignacion Expediente) (PlazasUsadas 1))
```

Ilustración 4. Base de hechos final.

8. Conclusiones

De las pruebas realizadas en el previo apartado de *Experimentación y Análisis de resultados* podemos extraer una serie de conclusiones.

Si bien las asignaciones se van realizando de forma ordenada según los hechos encontrados en la base de hechos, la generación de estas es correcta, y si es necesario asignar de nuevo un TFG a un alumno, se hará correctamente.

Si un alumno tiene un mejor expediente que otro, la asignación del TFG se hará al alumno de mejor expediente. Lo mismo ocurre con las entrevistas. En caso de encontrar un alumno con una mayor prioridad de entrevista, el TFG se asigna al alumno con mayor prioridad. En caso de empate, se asigna a aquel que se encuentre primero en la base de hechos (orden de solicitud). Inclusive, si el TFG acepta más de una plaza, no existe inconveniente en que se les asigne este TFG a varios alumnos.

En el caso de que un alumno no pueda elegir un TFG según sus preferencias, se tratará de buscar un TFG libre por descarte para que este no quede sin TFG. Además, en caso de que no existan propuestas suficientes para todos los alumnos, se notifica a aquellos sin asignación.

Además, se permite la introducción por teclado de forma correcta de nuevas propuestas de TFG por parte de nuevos alumnos, así como de nuevas propuestas de TFG por parte de la escuela para que los alumnos que hayan quedado sin TFG asignado puedan hacer un nuevo TFG.

Referencias

- [1] 2015. *CLIPS Reference Manual. Volume I. Basic Programming Guide*. En: *CLIPS: A Tool for Building Expert Systems* [en línea]. Disponible en: <http://clipsrules.sourceforge.net/documentation/v630/bpg.pdf> [consulta: 04/12/2021].
- [2] *Reglamento do Traballo Fin de Grao Enxeñaría Informática*. En: *Universidade de Santiago de Compostela* [en línea]. Disponible en: https://pro-assets-usc.azureedge.net/cdn/ff/JMrlGW4kDoN0PO0c5yvDbggv3mXUIQtWmULcwYUhbso/1635401133/public/paragraphs/links/2021-05/20200604_RegulamentoTFG_GrEI_vf_0.pdf [consulta: 04/12/2021].