

Práctica Final: Desarrollo de un S.E.

Gustavo Rivas Gervilla DNI: 75570417F

Índice

1	Resumen de cómo funciona el sistema experto	3
2	Descripción del proceso seguido para el desarrollo	3
2.1	Sesiones con el experto, indicando información obtenida en cada una de ellas	3
2.2	Procedimiento de validación y verificación del sistema seguido . .	5
3	Descripción del sistema desarrollado	6
3.1	Variables de entrada del problema	6
3.2	Variables de salida del problema	6
3.3	Conocimiento global del sistema (hechos y relaciones que se cargan inicialmente)	6
3.4	Especificación de los módulos que se han desarrollado (incluyendo para cada módulo el objetivo, conocimiento que utiliza y conocimiento que se deduce)	8
3.5	Estructura de funcionamiento del esquema de razonamiento del sistema (cuándo actuará cada módulo)	10
3.6	La lista de hechos que utiliza el sistema durante su funcionamiento y la forma de representarlos	10
3.7	Los hechos y las reglas de cada módulo	13
3.7.1	Módulo menú	13
3.7.2	Módulo 0	14
3.7.3	Módulo 1	15
3.7.4	Módulo 2	16
3.7.5	Módulo 3	16
3.7.6	Módulo 4.1	17
3.7.7	Módulo 4.2	17
3.7.8	Módulo Venta	18
3.7.9	Módulo Compra	19
3.7.10	Módulo Intercambio	20
4	Breve manual de uso del sistema	20

1 Resumen de cómo funciona el sistema experto

El sistema carga inicialmente todos los datos que necesita y que no puede deducir, estos datos se enumerarán en la sección correspondiente. Una vez que se han cargado los datos el sistema presenta un menú en el que podemos: pasar al sistema asesor (el SE propiamente dicho), mostrar la cartera actual del usuario o cerrar el programa.

Si se pasa al sistema asesor este deducirá en base a los datos que tiene qué valores del Ibex35 son inestables, peligrosos, infravalorados o sobrevalorados. Una vez a deducido esto verá qué recomendaciones son adecuadas para hacer al usuario y que este las pueda aceptar y actuar así sobre su cartera: vendiendo, comprando o intercambiando acciones. Entonces el sistema le mostrará al usuario una serie de acciones a realizar y llevará acabo aquella que seleccione el usuario, si selecciona alguna.

El usuario puede decidir si seguir viendo recomendaciones o volver al menú principal.

2 Descripción del proceso seguido para el desarrollo

2.1 Sesiones con el experto, indicando información obtenida en cada una de ellas

En la primera sesión se establecieron en primer lugar los módulos que iban a componer el SE, los cuáles son: detención de valores infravalorados, detención de valores sobrevalorados, detención de valores infravalorados, detención de valores peligrosos y el gestor de decisiones del usuario. Los tres primeros módulos se comunicarán con el último que será el que se encargue de proponerle al usuario las distintas acciones que puede realizar y de gestionarlas; será el más complejo de todos.

Para detectar los valores peligrosos se recibirá como entrada los datos de la bolsa de los cinco días durante los últimos cinco días, distintas noticias que pueden llevar una fecha y la cartera del usuario.

En esta misma sesión se estableció que un valor se considera peligroso si está cayendo durante 5 días seguidos y si la variación con respecto a su sector es mayor del 5%, cada día habrá que comprobar qué ha pasado en los últimos cinco días y si algún día ha habido una diferencia mayor del 5% entonces ese valor ya será peligroso. Por tanto para esto se hace patente la necesidad de tener también datos sobre los distintos sectores a los que pertenecen las distintas empresas del Ibex35.

También va a influir en que sea peligroso que haya una noticia mala sobre ese valor o sobre el sector. Puede haber noticias (positivas o negativas) sobre un valor, sobre un sector o sobre toda la economía. Las noticias sobre un sector se

propagan a todos los valores del sector y las noticias sobre la economía a todos los valores del mercado. Las noticias sobre un valor concreto van a tener más fuerza que las del sector y estas a su vez más fuerza que las de toda la economía, sobre todo en caso de que hayas unas positivas y otras negativas.

Si un sector está cayendo mucho todos los valores de ese sector serán peligrosos, que esté cayendo mucho significa que el sector está cayendo más que la economía general (más que la media general) y se mantiene así durante al menos cuatro o cinco días. Hay sectores que son más estables que otros (y por tanto unos que son menos propensos a ser peligrosos que otros). Habrá que guardar los sectores que sean menos peligrosos. La banca será estable, servicios y construcción inestables y el resto normales (ni estables ni inestables). Hay una diferencia entre el sector construcción y el sector servicios. La inestabilidad del sector servicios va muy ligada a la inestabilidad de la economía general (si la economía general está inestable, el sector servicios será inestable), mientras que la construcción será por defecto un sector inestable. Ya veremos como toda esta información adquirida en la primera sesión luego se precisa mucho más y cambio mucho a medida que aumentan las sesiones con el experto.

Un valor también se considerará peligroso si está cayendo durante tres días (ya veremos más adelante que en este caso no dependemos también de la variación con respecto al sector).

Por otro lado en esta sesión también se fijó cuándo un valor era considerado como inestable: un valor es inestable si pertenece a un sector inestable o hay una noticia negativa sobre él, las noticias tienen relevancia hasta dos días después de su publicación. Las noticias buenas cancelan a las malas, teniendo también efecto durante dos días. Así se decide darle al sistema solamente las noticias que tienen relevancia en lugar de darle una base de noticias con fechas y que sea el sistema el que tome las que le interesan.

En una sesión posterior se concretó con el experto que efectivamente las noticias buenas tenían mayor peso que las malas a la hora de decidir si un valor era inestable o no.

En las siguientes sesiones el experto ha especificado más concretamente las distintas reglas o razonamientos que le llevan a etiquetar un valor como inestable, sobrevalorado, infravalorado o peligroso, así se distinguen cuatro módulos distintos, en lugar de cuatro en lo que se refiere a etiquetar los distintos valores del Ibex35 para la continuación realizar sus propuestas.

En una sesión se concretó con el experto cuándo consideraba que un valor era inestable dando lugar a las reglas R0.13, R0.14, R0.15, R0.16, R0.17 y R0.18. En esta misma sesión se estableció cuándo un valor era considerado peligroso, además el experto, dado que no lo necesita para su razonamiento posterior, sólo revisa si los valores de la cartera del cliente son peligrosos o no, no prestando atención al resto de valores del Ibex35. Así el conocimiento del experto nos lleva a extraer las siguientes reglas para etiquetar un valor como peligroso: R1.1 y R1.2. (estas reglas ya habían sido señaladas en sesiones previas con el experto como hemos comentado antes).

En otra sesión decidimos centrarnos en los valores sobrevalorados de los que también había hablado el experto en la primera sesión. En esta sesión extrajimos las reglas R2.1, R2.2, R2.3, R2.4 y R2.5. En esta sesión fue necesario fijar qué se consideraba con alto, medio o bajo para el RPD y el PEr así como en base a qué se definía el tamaño de una empresa, obteniendo la siguiente información:

Una empresa es pequeña si su capitalización supone menos del 2% de la capitalización total del Ibex, mediana si está entre el 2% y el 5% y grande si es superior.

El PER de una empresa es bajo si menor que 12, medio si está entre 12 y 18 y alto si es superior. Del mismo modo el RPD es bajo si es menor que 2, medio si está entre 2 y 5 y alto si es mayor.

Por último en otra sesión aclaramos cuándo el experto consideraba que un valor se consideraba infravalorado dando lugar a las reglas R3.1, R3.2 y R3.3.

En la última sesión con el experto nos enfocamos en qué recomendaciones realizaba en base a todos los datos, los de partida y los deducidos y en cómo calificaba una acción con respecto a otras. El experto evalúa las acciones según un rendimiento esperado que se calcula según unas fórmulas para cada acción a recomendar, estas recomendaciones que realiza el experto dan lugar a las reglas: R41.1, R41.2, R41.3 y R41.4.

2.2 Procedimiento de validación y verificación del sistema seguido

Para verificar el sistema lo que hemos hecho ha sido probar que cada una de las reglas se activase cuando debía, probando tanto con datos que las debían activar como con datos que no debían hacerlo, y así hemos visto que el sistema hace lo que hemos programado. Así hemos visto que el sistema se comporta como queremos tanto en lo referente al razonamiento que ha de realizar con el orden procedural que ha de seguir. Con lo cuál aquí hemos visto que las conclusiones del sistema son correctas, completas y congruentes ya que vemos que las reglas se activan cuando, y solo cuándo han de activarse.

El sistema cuenta con mecanismos de seguridad en caso en que el usuario dé respuestas que no sean correctas al interactuar con el sistema.

El código de nuestro sistema es bastante legible ya que CLIPS es un lenguaje muy simple en cuanto a sintaxis además de estar correctamente comentado.

Por último hemos podido comprobar como el sistema es rápido en su ejecución.

Por otra parte para la validación hemos actuado como un usuario y hemos visto que la interfaz responde correctamente a cualquier eventualidad que pueda ocurrir antes el comportamiento de un usuario, hemos realizado así multitud de pruebas, poniendo a prueba entre otras cosas los mecanismo de seguridad.

Aceptando acciones, dando respuestas incorrectas como usuarios, realizando diversos patrones de decisión, para ver que el sistema se comporta según lo que espera el usuario y el empresario encontrar.

Actuando como usuario vemos que la interfaz es intuitiva, además de ser estar muy restringidas las acciones que puede llevar a cabo el usuario. El razonamiento está bien explicado pues hemos ido combinando las distintas explicaciones de todas las deducciones que llevan a la recomendación de una acción, con lo que la explicación es lo más completa posible.

No podemos saber si los resultados son de utilidad pues no estamos testeando el sistema con un usuario real.

3 Descripción del sistema desarrollado

3.1 Variables de entrada del problema

El sistema recibe la información actual de las empresas del Ibex35 en la sección correspondiente veremos la estructura de los hechos que representan y almacenan tal información, del mismo modo tenemos la información sobre los distintos sectores a los que pertenecen las empresas así como la información de la economía en general. También se carga el precio actual del dinero.

También recibe la cartera del usuario indicando los valores de los que posee acciones el usuario así como la cantidad de dinero de la que dispone para invertir.

Por último el sistema ha de conocer las noticias, que se consideren relevantes, que afecten a su razonamiento indicando el ámbito de tal noticia (a qué afecta) y si es buena o mala.

3.2 Variables de salida del problema

Como salida el sistema recomienda al usuario una serie de acciones (hasta 5) que puede realizar con las acciones y el dinero del que dispone, además el sistema producirá actualizaciones en la cartera del usuario según las acciones que tome.

3.3 Conocimiento global del sistema (hechos y relaciones que se cargan inicialmente)

Como hemos comentado anteriormente el sistema recibe como entrada la información de las distintas empresas del Ibex35 así como de los distintos sectores a los que pertenecen (entre los que está el propio Ibex para recoger la información de la economía general), estos datos se cargan en hechos que son templates de tipo **DatosEmpresa** y **DatosSector**. Aquí me he decantado por usar templates ya que como vemos a continuación, donde se muestra la declaración de ambas plantillas, estos hechos tienen gran multitud de campos y

con lo cual por medio de una plantilla es mucho más claro ver qué representa cada dato sin necesidad de ver qué orden ocupa, además es mucho más cómodo trabajar con plantillas en las reglas, que permiten no tener que especificar todos los campos, que con un hecho básico con un número tan elevado de campos.

```
(deftemplate DatosEmpresa
  (field Nombre)
  (field Precio (type NUMBER))
  (field VarDia (type NUMBER))
  (field Capital (type NUMBER))
  (field PER (type NUMBER))
  (field RPD (type NUMBER))
  (field Tamanio (type SYMBOL) (allowed-symbols
    ↪ PEQUENIA MEDIANA GRANDE))
  (field Ibex (type NUMBER))
  (field EtPER (type SYMBOL) (allowed-symbols Alto
    ↪ Medio Bajo))
  (field EtRPD (type SYMBOL) (allowed-symbols Alto
    ↪ Medio Bajo))
  (field Sector)
  (field Var5 (type NUMBER))
  (field Bajando3 (type SYMBOL) (allowed-symbols
    ↪ true false))
  (field Bajando5 (type SYMBOL) (allowed-symbols
    ↪ true false))
  (field VarRespSector5 (type NUMBER))
  (field VRS5 (type SYMBOL) (allowed-symbols true
    ↪ false))
  (field VarMes (type NUMBER))
  (field VarTri (type NUMBER))
  (field VarSem (type NUMBER))
  (field Var12mes (type NUMBER))
)

(deftemplate DatosSector
  (field Nombre)
  (field VarDia (type NUMBER))
  (field Capital (type NUMBER))
  (field PER (type NUMBER))
  (field RPD (type NUMBER))
  (field Ibex (type NUMBER))
  (field Var5 (type NUMBER))
  (field Bajando3 (type SYMBOL) (allowed-symbols
    ↪ true false))
  (field Bajando5 (type SYMBOL) (allowed-symbols
    ↪ true false))
  (field VarMes (type NUMBER))
  (field VarTri (type NUMBER))
  (field VarSem (type NUMBER))
  (field Var12mes (type NUMBER))
)
```

|)

En cambio dado que tanto los datos pertenecientes a la cartera del usuario como las noticias almacenadas no tienen tantos datos he decidido usar hechos básicos que siguen la siguiente estructura:

```
( Cartera NombreValor | DISPONIBLE Acciones | Cash
  ↳ ValorAcciones | Cash )
( Noticia NombreEmpresa | NombreSector | Ibex buena | mala )
```

y estos son los datos, junto con el precio del dinero que se almacena en un hecho simple, que carga de inicio el sistema para poder iniciar su razonamiento.

3.4 Especificación de los módulos que se han desarrollado (incluyendo para cada módulo el objetivo, conocimiento que utiliza y conocimiento que se deduce)

En primer lugar tenemos un módulo que podríamos llamar el **Módulo Menú** pero realmente este módulo no forma parte del razonamiento del sistema, este módulo sólo lleva a cabo funciones de interacción con el usuario: mostrarle el menú, la cartera del usuario, salir del sistema almacenando el estado actual de la cartera del usuario y dar paso al núcleo del SE, que son los módulos donde se produce el razonamiento para el objetivo para el que está destinado el sistema realmente.

El **Módulo 0** es el módulo que inicia el funcionamiento del SE, en él se cargan los datos que hemos dicho que recibe como entrada el SE, en consecuencia este módulo no emplea ningún conocimiento a priori. Ahora bien, el otro objetivo del que se encarga este módulo es el de deducir cuáles son los valores inestables, para ello utiliza el conocimiento que previamente ha cargado, en concreto los datos sobre los sectores (la economía), los datos de las empresas y las noticias que se han cargado. Así el conocimiento deducido por este módulo son los valores inestables que hay en ese momento.

El **Módulo 1** parte del conocimiento de los valores que han sido marcados como Inestables, de la cartera del cliente y los datos de las empresas. Tiene como objetivo marcar qué valores de la cartera del cliente están sobrevalorados, así que el conocimiento deducido por este módulo es saber qué valores están sobrevalorados de entre aquellos que tiene el usuario en su cartera.

El **Módulo 2** parte sólo de los datos que se han cargado sobre cada una de las empresa y su objetivo es deducir el conocimiento de qué valores (de la cartera del usuario o no) están sobrevalorados.

El **Módulo 3** al igual que el anterior parte sólo de la información cargada en el Módulo 0 sobre las empresas y su objetivo es de deducir el conocimiento

de qué valores están infravalorados.

El **Módulo 4.1** emplea el conocimiento deducido por los módulos anteriores además de los datos de las empresas, los sectores, la cartera del usuario y el precio del dinero. Este módulo tiene como objetivo hacer distintas recomendaciones sobre los valores del Ibex35 en base a la información obtenida, así el conocimiento deducido son las distintas acciones que pueden llevarse a cabo junto con el rendimiento expertado de estas acciones (y una explicación de por qué se recomienda tal acción).

El **Módulo 4.2** se encarga de manejar las recomendaciones deducidas por el razonamiento de los módulos anteriores, mostrando al usuario las mejores que ha encontrado (hasta un máximo de cinco), solicitándole que elija una o rechace todas y en caso de aceptar alguna pasar el control a los distintos módulos que gestionan cada una de las posibles operaciones. Cuando acaba la ejecución del módulo correspondiente elimina los hechos que no son necesarios y pasa el Módulo 1 para actualizar la base de hechos y por ende las recomendaciones según haya cambiado la cartera del usuario tras la acción. En este módulo no se genera nuevo conocimiento.

El **Módulo de ventas** se encarga de gestionar la venta de las acciones de un cierto valor (de la cartera del cliente) que ha elegido el usuario, para ello emplea la información de la empresa involucrada en la operación, de la cartera del cliente y la recomendación que el usuario ha aceptado. El conocimiento que genera este módulo es cómo queda la cartera del cliente tras la transacción así como eliminar aquella información que ya deja de tener sentido en el SE: si el usuario se queda sin acciones del valor vendido entonces las recomendaciones de vender acciones de esa empresa (que no sean la que está en curso) y de hacer un intercambio con acciones de esa empresa ya no tienen sentido, además los valores peligrosos son valores de la cartera del usuario, en caso de desaparecer de la cartera ya no se considerarán peligrosos.

El **Módulo de compras** se encarga de gestionar la compra de acciones de un cierto valor que ha elegido el usuario, para ello emplea la información de la empresa involucrada en la operación, de la cartera del cliente y la recomendación que el usuario ha aceptado. El conocimiento que genera este módulo es cómo queda la cartera del cliente tras la transacción así como eliminar aquella información que ya deja de tener sentido en el SE: una vez se ha realizado la compra se comprueba que el dinero disponible que le queda al usuario es suficiente para poder llevar a cabo el resto de operaciones de compra que se le han recomendado al usuario.

Por último el **Módulo de intercambios** se encarga de gestionar el intercambio de acciones de un cierto valor (de la cartera del usuario) que ha elegido el usuario por otro recomendado, para ello emplea la información de las empresas involucradas en la operación, de la cartera del cliente y la recomendación que el usuario ha aceptado. El conocimiento que genera este módulo es cómo queda la cartera del cliente tras la transacción así como eliminar aquella información que ya deja de tener sentido en el SE: si el usuario se queda sin acciones del valor intercambiado entonces las recomendaciones de vender acciones de esa empresa

y de hacer un intercambio con acciones de esa empresa (que no sean la que está en curso) ya no tienen sentido, además los valores peligrosos son valores de la cartera del usuario, en caso de desaparecer de la cartera ya no se considerarán peligrosos.

3.5 Estructura de funcionamiento del esquema de razonamiento del sistema (cuándo actuará cada módulo)

El esquema de funcionamiento del sistema experto diseñado responde al siguiente grafo:

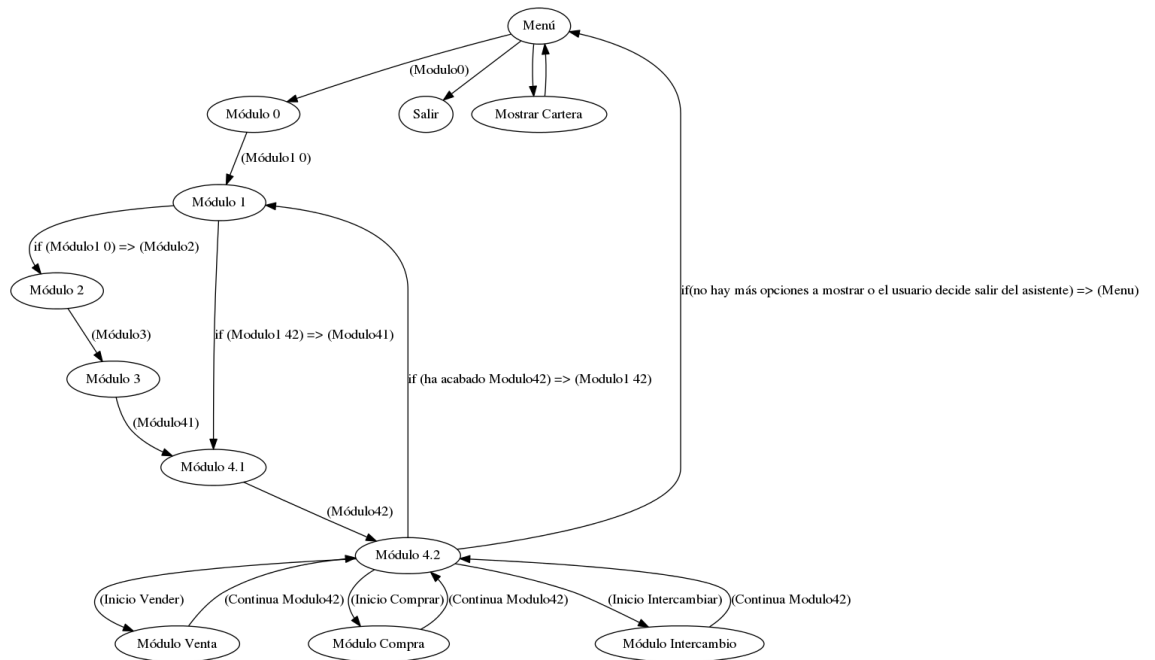


Figure 1: En las aristas mostramos los hechos relevantes que se lanzan para pasar al módulo correspondiente

3.6 La lista de hechos que utiliza el sistema durante su funcionamiento y la forma de representarlos

Los hechos que se emplean en el SE son los siguientes:

H1 **DatosEmpresa** donde se representan los distintos datos que tenemos de las empresas, se representan por medio de la plantilla DatosEmpresa que hemos mostrado anteriormente.

H2 **DatosSector** donde se representan los distintos datos que tenemos de los sectores a los que pertenecen las distintas empresas del Ibex35 así como

los datos de la economía en general representada por el sector Ibex. Se almacenan en la forma del template DatosSector que hemos mostrado en un apartado anterior.

- H3 (**Noticia ?Afectado buena|mala**) almacenan las distintas noticias que conocemos donde tenemos tanto al que afecta la noticia (que puede ser una empresa, un sector o la economía general) como si la noticia es buena o mala. Como vemos no almacenamos ni la antigüedad ni la fecha de la noticia ya que suponemos que partimos, al igual que dijo el experto, de un listado de noticias que son relevantes (es decir con una antigüedad de un día o dos a lo sumo).
- H4 (**Cartera ?Valor ?Nº Acciones ?ValorAcciones**) donde se almacenan los datos de la cartera del usuario incluyendo un Valor especial que es el DISPONIBLE, el dinero del que dispone el usuario.
- H5 (**SeguirLeyendo**
DatosEmpresa|DatosSector|Cartera|Noticias hechos que se usan en las distintas instrucciones que se encargan de cargar los distintos datos iniciales del sistema, con estos hechos realizamos los bucles que leen cada uno de los ficheros de datos hasta el final.
- H6 (**Inestable ?Valor ?Explicacion**) hechos en los que representamos el conocimiento deducido por el Módulo 0, así se almacenan qué valores son Inestables junto con una breve explicación de por qué ese valor ha sido etiquetado como inestable.
- H7 (**Modulo1 ?ModuloInvocador**) este tipo de hecho lo empleamos para pasar el control al Modulo1, la variable ?ModuloInvocador se debe a que dicho módulo puede ser invocado tanto por el módulo 0 como por el 4.1. como hemos explicado en la sección anterior y necesitamos distinguir quién pasa el control a este módulo para ver qué acciones acometer tras la finalización de dicho módulo.
- H8 (**Peligroso ?Valor ?Explicacion**) hechos en los que representamos el conocimiento deducido por el Módulo 1, es decir, almacenamos qué valores son inestables junto con una explicación de por qué se deduce que ese valor es inestable.
- H9 (**Modulo2**) hecho con el que pasamos el control al Módulo 2 tras completarse la ejecución del Módulo 1 si procede.
- H10 (**Modulo41**) hecho con el que pasamos el control al Módulo 4.1 tras completarse el razonamiento del Módulo 1 si procede.
- H11 (**Sobrevalorado ?Valor ?Explicacion**) hechos en los que representamos el conocimiento deducido por el Módulo 2, es decir, almacenamos qué valores están sobrevalorados junto con una explicación de por qué se deduce tal cosa.
- H12 (**Modulo3**) hecho con el que pasamos el control al Módulo 3 tras completarse el razonamiento del Módulo 2.

- H13 (**Infravalorado ?Valor ?Exp**) hechos en los que representamos el conocimiento deducido por el Módulo 3, es decir, almacenamos qué valores están infravalorados junto con una explicación de por qué se deduce tal cosa.
- H14 (**QuitadosPorBuenas**) hecho que empleamos en el Módulo 0 para evitar un bucle infinito al quitar un valor de Inestable si hay una noticia buena sobre él o su sector y a la vez no producir dos hechos Inestables para el mismo valor por distintas razones.
- H15 **Recomendación** estos tipos de hechos se emplean para almacenar el conocimiento deducido por el Módulo 4.1, las distintas recomendaciones que realiza el SE en base al conocimiento deducido por los módulos anterior y que carga el SE de inicio. La información que se deduce se almacena en forma de un template definido del siguiente modo:

```
(deftemplate Recomendacion
  (field ID (type NUMBER))
  (field Accion (type SYMBOL) (allowed-symbols Vender
    ↪ Comprar Intercambiar))
  (field RE (type NUMBER))
  (field Empresa1)
  (field Empresa2 (default NULL))
  (field Texto)
)
```

en las operaciones Comprar y Vender sólo tenemos la Empresa1, ahora para la operación intercambiar la Empresa1 será la empresa de la que adquiriremos las acciones y la Empresa2 es la empresa de nuestra cartera cuyas acciones empleamos para el intercambio.

- H16 (**PrecioDinero ?Precio**) este hecho es precargado por el sistema al inicio he indica qué precio tiene el dinero en ese momento, información que emplea el Módulo 4.1 para su razonamiento.
- H17 (**Modulo 42**) hecho que empleamos para pasar el control al Módulo 4.2.
- H18 (**NumPropuestas ?Num**) con este hecho llevamos un contador de cuántas propuestas han sido seleccionadas (por orden decreciente de su RE) para mostrarle al usuario, para que no sean más de 5.
- H19 (**Propuesta ?Id**) hechos en los que guardamos, por su ID, las recomendaciones que han sido seleccionadas para ser mostradas al usuario.
- H20 (**SolicitarAccion**) con este hecho indicamos que hemos de solicitarle al usuario qué propuesta desea aceptar, permitiéndonos realizar un bucle de comprobación para ver si la respuesta del usuario es correcta, es decir, si es una de las propuestas presentadas o -1.
- H21 (**Aceptada ?Id**) aquí almacenamos la respuesta que da el usuario a la consulta anterior.

- H22 (**Inicio ?Accion**) si el usuario acepta alguna de las propuestas con este hecho damos paso al módulo que se encargue de llevar acabo la acción requerida: comprar, vender o intercambiar. Estos hechos también están involucrados en mantener un bucle para comprobar si la respuesta del usuario es correcta y en caso contrario volver a solicitar las acciones que desea comprar, vender o intercambiar.
- H23 (**Continua Modulo42**) este hecho es empleado por los módulos de manejo de acción una vez han acabado su trabajo para indicarle al Módulo 4.2 que puede seguir con sus acciones posteriores al manejo de la acción.
- H24 (**QuiereVender ?Respuesta**) en estos hechos se almacena el número de acciones que el usuario desea vender del valor considerado, el cuál se le solicitará.
- H25 (**Limpiar Venta**) hecho que se emplea para, una vez se ha completado la venta, eliminar aquella información de la base de conocimiento del SE que deja de tener sentido, la cual hemos indicado anteriormente.
- H26 (**QuiereComprar ?Respuesta**) en estos hechos se almacena el número de acciones que el usuario desea comprar del valor considerado, el cuál se le solicitará.
- H27 (**Limpiar Compra**) hecho que se emplea para, una vez se ha completado la compra, eliminar aquella información de la base de conocimiento del SE que deja de tener sentido, la cual hemos indicado anteriormente.
- H28 (**Menu**) hecho con el que mostramos el Menú del sistema.
- H29 (**Opc ?Respuesta**) hecho en que recogemos la opción del menú que ha elegido el usuario.
- H30 (**Modulo0**) hecho con el que pasamos al razonamiento del Módulo 0, detectar los inestables, el Módulo 0 también se encarga de precargar los datos de entrada del sistema pero estas actividades se realizan antes de mostrar el menú, antes de comenzar la ejecución del sistema por así decirlo.
- H31 (**Ids ?Contador**) hecho contador para ir asignando distintos Ids a las recomendaciones que se van deduciendo a lo largo de la ejecución del sistema.
- H32 (**OpcElegida ?Opc**) hecho donde almacenamos la decisión del usuario tras rechazar todas las propuestas que se le presentan: si quiere ver otras nuevas o salir del asistente al menú principal.

3.7 Los hechos y las reglas de cada módulo

3.7.1 Módulo menú

Los hechos que se emplean o generan en este módulo son los hechos H4, H28, H29 y H30. Y las reglas que en el aparecen son:

- RM.1 **mostrarMenu** regla con la que mostramos al usuario el menú del sistema y le pedimos al usuario que elija entre las opciones disponibles.

- RM.2 **rechazarOpc** regla con la que rechazamos, en caso de no ser correcta, la opción elegida por el usuario, volviendo a mostrar el menú.
- RM.3 **iniciarAsistente** regla con la que pasamos el control al Módulo 0 para que comience el razonamiento del SE si así lo requiere el usuario.
- RM.4 **mostrarCartera1**
- RM.5 **mostrarCartera2** con estas dos reglas mostramos el estado actual de la cartera al usuario si así lo requiere.
- RM.6 **volverMenuTrasMostrarCartera** con esta regla mostramos nuevamente el menú una vez se le ha mostrado la cartera al usuario.
- RM.7 **abrirFicheroCarteraAntesDeSalir** con esta regla abrimos el fichero en que se almacenará el estado actual de la cartera del cliente si decide salir.
- RM.8 **copiarDatosCartera** con esta regla copiamos los datos de la cartera del cliente al fichero abierto.
- RM.9 **salir** con esta regla cerramos el fichero anterior y detenemos la ejecución del sistema.

3.7.2 Módulo 0

Los hechos que se emplean o generan en este módulo son los hechos H1, H2, H3, H4, H5, H6, H7 ,H14, H16, H18, H30 y H31. Y las reglas que en el aparecen son:

- R0.1 **abrirficheroemp** regla con la que abrimos el fichero Analisis.txt que contiene los datos de las distintas empresas del Ibex35.
- R0.2 **leervalorescierreemp** regla con la que se van leyendo las sucesivas líneas del fichero anterior para ir produciendo hechos del tipo DatosEmpresa con la información de cada empresa.
- R0.3 **cerrarficheroemp** regla que cierra el fichero abierto con los datos de las empresas al concluir su lectura.
- R0.4 **abrirficherosec** regla con la que abrimos el fichero AnalisisSectores.txt que contiene los datos de los distintos sectores (así como de la economía que es el sector Ibex) de las empresas del Ibex35.
- R0.5 **leervalorescierresec** regla con la que se van leyendo las sucesivas líneas del fichero anterior para ir produciendo hechos del tipo DatosSector con la información de cada sector.
- R0.6 **cerrarficherosec** regla que cierra el fichero abierto con los datos de los sectores al concluir su lectura.
- R0.7 **abrirficheroartera** regla con la que abrimos el fichero Cartera.txt que contiene los datos de la cartera del usuario.

- R0.8 **leercartera** regla con la que se van leyendo las sucesivas líneas del fichero anterior para ir produciendo hechos del tipo Cartera con la información de cada valor de la cartera del usuario.
- R0.9 **cerrarficheroocar** regla que cierra el fichero abierto con los datos de la cartera al concluir su lectura.
- R0.10 **abrirficheronoticias** regla con la que abrimos el fichero Noticias.txt que contiene las noticias relevantes para el razonamiento del SE.
- R0.11 **leernoticias** regla con la que van leyendo las sucesivas líneas del fichero anterior para ir produciendo hechos del tipo Noticia con las distintas noticias.
- R0.12 **cerrarficheronoticias** regla que cierra el fichero abierto con las noticias al concluir su lectura.
- R0.13 **contruccionInestableDefecto** regla que se encarga de marcar como inestables, a priori, todos aquellos valores que pertenezcan al sector de la Construcción.
- R0.14 **siEconomiaBajaServiciosInestablesDefecto** regla que en caso de que la economía (el sector Ibex) esté bajando marca como Inestables todos los valores del sector Servicios. Marcar un hecho como Inestable es incluir un hecho del tipo 6 de la lista de hechos de la sección anterior.
- R0.15 **inestablePorNoticiasMalas1** si hay una noticia mala sobre un valor concreto esta regla lo marca como inestable.
- R0.16 **inestablePorNoticiasMalas2** si hay una noticia mala sobre el sector de un valor éste es marcado como Inestable.
- R0.17 **inestablePorNoticiasMalas3** regla que marca cada uno de los valores del Ibex35 como Inestables si hay una noticia mala sobre la economía general.
- R0.18 **establePorNoticiasBuenas** si hay una noticia buena sobre un valor o su sector este ya no es Inestable, eliminado el hecho (Inestable ?Valor ?) en caso de haber sido antes etiquetado como Inestable. Eso sí, una noticia buena sobre el sector de un valor no prima más que una noticia mala sobre el propio valor; con lo que en ese caso el valor seguiría siendo Inestable.
- R0.19 **pasarAModulo1** regla para pasar el control al Módulo 1.

3.7.3 Módulo 1

Los hechos que se intervienen en el funcionamiento de este módulo junto con los que se generan son los hechos H1, H4, H6, H7, H8, H9 y H10. Las reglas que componen este módulo son sólo tres:

- R1.1 **deducirPeligrosos1** esta regla marca como Peligrosos aquellos valores de la cartera del cliente que hayan sido marcados como Inestables en el Módulo 0 y que lleven tres días bajando de forma continuada. Marcar un valor como Peligroso es generar un hecho de tipo H8.

- R1.2 **deducirPeligrosos2** esta regla marca como Peligroso aquellos valores de la cartera del usuario que lleven cinco días bajando de forma continuada y que la variación con respecto al sector de ese valor en los últimos cinco días haya sido mayor que un -5%.
- R1.3 **pasarAModulo2o41** esta regla da el control al Módulo 2 o al Módulo 4.1 del sistema dependiendo de si lo ha invocado el Módulo 0 o el Módulo 4.2 atendiendo así al flujo de funcionamiento del SE.

3.7.4 Módulo 2

Los hechos que intervienen en el razonamiento de este módulo junto con los que se generan en él son los hechos H1, H9, H11 y H12. Las reglas que componen este módulo son:

- R2.1 **detectarSobrevalorados1** regla que etiqueta un valor como sobrevalorado si su PER es alto y su RPD bajo. Marcar un valor como sobrevalorado es generar un hecho del tipo H11.
- R2.2 **detectarSobrevalorados2** regla que etiqueta una empresa pequeña (valor) como sobrevalorada si su PER es alto.
- R2.3 **detectarSobrevalorados3** regla que etiqueta una empresa pequeña como sobrevalorada si su PER es mediano y su RPD bajo.
- R2.4 **detectarSobrevalorados4** regla que etiqueta una empresa grande como sobrevalorada si su PER es alto o mediano y su RPD bajo.
- R2.5 **detectarSobrevalorados5** regla que etiqueta una empresa grande como sobrevalorada si su PER es alto y su RPD mediano.
- R2.6 **pasarAModulo3** regla que da el control al Módulo 3 cuando finaliza el razonamiento del Módulo 2.

3.7.5 Módulo 3

Los hechos que intervienen en el razonamiento de este módulo junto con los que se generan en él son los hechos H1, H10, H12 y H13. Las reglas que componen este módulo son:

- R3.1 **detectarInfravalorados1** regla que etiqueta un valor como infravalorado si su PER es bajo y su RPD es alto. Marcar un valor como infravalorado es generar un hecho del tipo H13.
- R3.2 **detectarInfravalorados2** regla que etiqueta un valor como infravalorado si en el último año, o en los últimos 6 o 3 meses ha caído más de un 30%, ha subido un poco (menos del 10%) en el último mes y su PER es bajo.
- R3.3 **detectarInfravalorados3** regla que etique una empresa grande como infravalorada si su RPD es alto, su PER mediano, no está bajando y se comporta mejor que su sector (considerando para tal estudio la variación en la semana tanto de la empresa como del sector).
- R3.4 **pasarAModulo41** regla que pasa el control al Módulo 4.1 cuando finaliza el razonamiento del Módulo 3.

3.7.6 Módulo 4.1

Los hechos que intervienen en el razonamiento de este módulo junto con los que se generan en él son los hechos H1, H2, H4, H8, H10, H11, H13, H15 y H16. Las reglas que componen este módulo son:

- R41.1 **proponerVenderPeligrosos** regla que propone vender las acciones de la cartera del cliente para un valor que sea peligroso, que haya bajado en el último mes y que además esta bajada sea un 3% más que la de su sector. Cuando decimos proponer algo es que se crea un dato de tipo H15 con la acción correspondiente, la(s) empresa(s) implicadas y la justificación de dicha recomendación junto con el rendimiento esperado de tal acción.
- R42.2 **proponerInvertirInfravalorados** regla que propone invertir en acciones de un valor que esté infravalorada, siempre que el usuario tenga dinero para ello.
- R43.3 **proponerVenderSobrevalorados** regla que propone vender las acciones de un valor de la cartera del usuario que esté sobrevalorada y cuyo rendimiento por año sea menor que 5 más el precio del dinero.
- R44.4 **proponerIntercambiarAcciones** regla que propone cambiar las acciones de un cierto valor del usuario que no esté infravalorado por las de otra empresa que no esté sobrevalorada cuyo RPD sea mayor que la suma del RPD de la empresa de la cartera más su rendimiento esperado por año más 1. Propondremos este intercambio siempre que el dinero que nos dé las acciones de nuestra cartera que intercambiamos nos valgan para obtener al menos una acción de otra empresa.
- R45.5 **pasarAModulo42** regla que pasa el control al Módulo 4.2 cuando finaliza el razonamiento del Módulo 4.1.

3.7.7 Módulo 4.2

Los hechos que intervienen en la actividad de este módulo junto con los que se generan en él son los hechos H7, H15, H17, H18, H19, H20, H21, H22, H23, H28 y H32. Las reglas que componen este módulo son:

- R42.1 **buscarMejorRecomendacion** esta regla busca de entre todas las recomendaciones posibles que no hayan sido ya seleccionadas cuál tiene mayor RE (y siempre que no se hayan elegido ya cinco acciones) generando un hecho de tipo H19 y actualizando el contador de propuestas escogidas.
- R42.2 **mostrarLas5Propuestas** esta regla muestra por terminal las recomendaciones que han sido seleccionadas por la regla anterior, mostrando el ID de la acción para que el usuario la pueda seleccionar, el RE de la acción y el motivo de la recomendación.
- R42.3 **solicitarAccionARealizar** con esta regla le pedimos al usuario que elija una de las acciones mostradas por su ID o -1 si no le interesa ninguna.

- R42.4 **rechazarAceptada** regla con la que hacemos que se le vuelva a solicitar al usuario qué acción desea realizar en caso de que la respuesta no se correcta.
- R42.5 **aceptarAccion** regla con la que, en caso de que el usuario haya escogido una de las acciones propuestas, le pasamos el control al módulo correspondiente de manejo de la acción.
- R42.6 **limpiarPropuestas** regla con la que, tras finalizar la ejecución del módulo de manejo de acción correspondiente se eliminan los hechos de tipo H19 para que se pueda realizar una nueva selección de recomendaciones si procede.
- R42.7 **limpiarRecomendacion** con esta regla eliminamos la recomendación, el hecho de tipo H15, que ha aceptado el usuario y que ha sido llevada a cabo. También elimina el hecho de tipo H21 que almacena la respuesta del usuario, sobre qué propuesta escoge.
- R42.8 **volverAModulo1** con esta regla regresamos al Módulo 1 para ver si hay nuevos valores peligrosos tras realizar la acción seleccionada, también reiniciamos aquí el contador de recomendaciones seleccionadas para futuras selecciones.
- R42.9 **manejarRechazo** regla con la que, una vez el usuario a rechazado todas las opciones que se le presentan, se le pregunta qué quiere hacer.
- R42.10 **rechazarOpcion** regla con la que rechazamos la respuesta del usuario a la cuestión anterior si no es correcta.
- R42.11 **salirDelAsistente** regla con la que, si el usuario lo decide, salimos del asistente al menú principal.
- R42.12 **limpiarParaMostrarOtrasPropuestas** regla con la que eliminamos todas las recomendaciones que el usuario a rechazado (en caso de rechazar todas) junto con el hecho que marcaba que habían sido escogidas para mostrárselas al usuario.
- R42.13 **mostrarNuevasPropuestas** regla con la que pasamos a seleccionar otro grupo de hasta cinco propuestas para presentárselas al usuario si no acepta ninguna de las mostradas y quiere ver más.
- R42.14 **noHayMasPropuestas** regla con la que pasamos al menú principal si el usuario ha solicitado nuevas propuestas pero el sistema no tiene más.

3.7.8 Módulo Venta

Los hechos que intervienen en el razonamiento de este módulo junto con los que se generan en él son los hechos H1, H4, H8, H15, H21, H22, H24 y H25. Las reglas que componen este módulo son:

- RV.1 **solicitarAccionesAVender** regla con la que se le pide al usuario el número de acciones que desea vender del valor indicado.

RV.2 **rechazarVenta** regla para rechazar la respuesta del usuario, es decir, si el usuario indica un número de acciones negativo o por encima del número de acciones del valor que tiene en su cartera, entonces la respuesta se rechazará y se solicitará de nuevo el número de acciones a vender.

RV.3 **realizarVenta** esta regla se encarga de realizar la venta cuando la respuesta del usuario es correcta. Así en esta regla se quitan de la cartera aquellas acciones que se hayan vendido y se aumenta el DISPONIBLE del cliente con la venta realizada quitándole la comisión.

RV.4 **limpiezaTrasVenta1**

RV.5 **limpiezaTrasVenta2**

RV.6 **limpiezaTrasVenta3** esta regla junto con las dos anteriores son las que se emplean para eliminar de la base del conocimiento del SE la información que ya no tiene sentido tras la venta: recomendaciones de venta de ese valor, de intercambio de ese valor y la información que marca ese valor como peligroso, respectivamente.

3.7.9 Módulo Compra

Los hechos que intervienen en el razonamiento de este módulo junto con los que se generan en él son los hechos H1, H4, H15, H21, H22, H26 y H27. Las reglas que componen este módulo son:

RC.1 **solicitarAccionesAComprar** regla con la que se le pide al usuario el número de acciones que desea comprar del valor indicado.

RC.2 **rechazarCompra** regla para rechazar la respuesta del usuario, es decir, si el usuario indica un número de acciones negativo o por encima de lo que puede comprar con el dinero del que dispone, entonces la respuesta se rechazará y se solicitará de nuevo el número de acciones a vender.

RC.3 **realizarCompra1** esta regla se encarga de realizar la compra cuando la respuesta del usuario es correcta y el usuario no disponía anteriormente de acciones del valor a comprar. Así en esta regla se añaden a la cartera aquellas acciones que se hayan comprado y se disminuye el DISPONIBLE según la compra que se haya realizado contando la comisión de la operación.

RC.4 **realizarCompra2** esta regla se encarga de realizar la compra cuando la respuesta del usuario es correcta y el usuario ya dispone de acciones del valor a comprar. Así en esta regla se añaden a la cartera aquellas acciones que se hayan comprado y se disminuye el DISPONIBLE según la compra que se haya realizado contando la comisión de la operación.

RC.5 **limpiezaTrasCompra** esta regla es la que se emplean para eliminar de la base del conocimiento del SE la información que ya no tiene sentido tras la compra y que hemos mencionado anteriormente.

3.7.10 Módulo Intercambio

Los hechos que intervienen en el razonamiento de este módulo junto con los que se generan en él son los hechos H1, H4, H15, H21, H22, H28 y H29. Las reglas que componen este módulo son:

- RI.1 **solicitarAccionesAIntercambiar** regla con la que se le pide al usuario el número de acciones que desea intercambiar del valor indicado.
- RI.2 **rechazarIntercambio** regla para rechazar la respuesta del usuario, es decir, si el usuario indica un número de acciones negativo o por encima del número de acciones del que dispone o bien el número de acciones no permiten al usuario adquirir una sola acción de la otra empresa (teniendo en cuenta las comisiones), entonces la respuesta se rechazará y se solicitará de nuevo el número de acciones a intercambiar.
- RI.3 **realizarIntercambio1** esta regla se encarga de realizar el intercambio cuando la respuesta del usuario es correcta y el usuario no disponía anteriormente de acciones del valor del que se van a obtener acciones con el intercambio. Así en esta regla se añaden a la cartera aquellas acciones que se hayan obtenido, disminuye el número de acciones de la empresa que se han intercambiado y se aumenta el DISPONIBLE según lo que sobre con el intercambio.
- RI.4 **realizarIntercambio2** esta regla se encarga de realizar el intercambio cuando la respuesta del usuario es correcta y el usuario ya dispone de acciones del valor del que se van a obtener acciones con el intercambio. Así en esta regla se añaden a la cartera aquellas acciones que se hayan obtenido, disminuye el número de acciones de la empresa que se han intercambiado y se aumenta el DISPONIBLE según lo que sobre con el intercambio.
- RI.5 **limpiezaTrasIntercambio1**
- RI.6 **limpiezaTrasIntercambio2**
- RI.7 **limpiezaTrasIntercambio3** esta regla junto con las dos anteriores son las que se emplean para eliminar de la base del conocimiento del SE la información que ya no tiene sentido tras el intercambio: recomendaciones de venta de ese valor, de intercambio de ese valor y la información que marca ese valor como peligroso, respectivamente.

4 Breve manual de uso del sistema

Para emplear el sistema necesitamos tener los ficheros de código que se adjunta en la carpeta SE en la misma carpeta, podemos simplemente copiar esta carpeta donde prefiramos. Además de los ficheros con el código del sistema en la carpeta incluimos los ficheros: Analisis.txt, AnalisisSectores.txt, Noticias.txt y Cartera.txt los cuales son ejemplos de conocimiento inicial para el sistema. Estos ficheros deberán estar también en la misma carpeta que los ficheros de código.

Entonces si arrancamos CLIPS desde esta carpeta (en Windows podemos hacer esto copiando el ejecutable de CLIPS en la carpeta) lo único que tendremos que hacer es cargar el contenido del fichero `run.clp` y una vez se haya cargado ejecutaremos la función `Run` que en el se define, la cual se encarga de cargar el resto de archivos de código, realizar un reset para cargar los datos definidos y hacer run para que el sistema comience a funcionar. Para ejecutar la función no tenemos más que dar la orden (`Run`) en la línea de comandos.

Una vez que hagamos esto el sistema comenzará a funcionar normalmente y no tendremos más que seguir las instrucciones de la interfaz.