

Ingeniería del Conocimiento:  
Práctica Final: Desarrollo de un Sistema Experto para  
ayudar a un inversor en bolsa.

Anabel Gómez Ríos.

DNI: 75929914Z.

E-mail: [anabelgrios@correo.ugr.es](mailto:anabelgrios@correo.ugr.es)

30 de junio de 2016

Curso 2015-2016

Grupo de prácticas: Lunes 17:30-19:30

Quinto curso del Doble Grado en Ingeniería Informática y Matemáticas.

# Índice

<b>1. Resumen sobre cómo funciona el sistema experto</b>	<b>3</b>
<b>2. Descripción del proceso seguido para el desarrollo</b>	<b>3</b>
2.1. Sesión 1: Entrevista informal general . . . . .	3
2.2. Sesión 2: Entrevista formal general . . . . .	4
2.3. Sesión 3: Entrevista formal sobre el módulo de detectar valores peligrosos y el módulo detectar valores inestables . . . . .	4
2.4. Sesión 4: Entrevista formal sobre los módulos de detección de valores infravalorados y sobrevalorados . . . . .	5
2.5. Sesión 5: Entrevista formal sobre el módulo de obtener propuestas para el usuario	5
2.6. Procedimiento de validación y verificación del sistema seguido . . . . .	6
<b>3. Descripción del sistema desarrollado</b>	<b>6</b>
3.1. Variables de entrada del problema . . . . .	6
3.2. Variables de salida del programa . . . . .	8
3.3. Conocimiento global del sistema . . . . .	9
3.4. Especificación de los módulos que se han desarrollado . . . . .	10
3.4.1. Módulo inicial: leer datos de los ficheros . . . . .	10
3.4.2. Módulo Cero . . . . .	10
3.4.3. Módulo Uno . . . . .	10
3.4.4. Módulo Dos . . . . .	11
3.4.5. Módulo Tres . . . . .	11
3.4.6. Módulo Cuatro . . . . .	11
3.4.7. Módulo Cinco . . . . .	11
3.4.8. Módulo Seis: Salir del programa . . . . .	12
3.5. Estructura de funcionamiento del esquema de razonamiento del sistema . . . . .	12
3.6. Lista de hechos que utiliza el sistema durante su funcionamiento y la forma de representarlos . . . . .	12
3.7. Hechos y reglas de cada módulo . . . . .	13
3.7.1. Módulo inicial . . . . .	13
3.7.2. Módulo Cero . . . . .	14
3.7.3. Módulo Uno . . . . .	14
3.7.4. Módulo Dos . . . . .	14
3.7.5. Módulo Tres . . . . .	15
3.7.6. Módulo Cuatro . . . . .	15
3.7.7. Módulo Cinco . . . . .	16
3.7.8. Módulo Seis (Salir) . . . . .	17
<b>4. Breve manual de uso del sistema</b>	<b>17</b>

## 1. Resumen sobre cómo funciona el sistema experto

El sistema experto se compone de siete módulos funcionales: módulo inicial, encargado de leer los datos iniciales de la cartera, módulo cero, encargado de detectar valores inestables, módulo uno, encargado de detectar valores peligrosos, módulo dos, encargado de detectar valores sobrevalorados, módulo tres, encargado de detectar valores infravalorados, módulo cuatro, encargado de hacer las propuestas, módulo cinco, encargado de la interacción con el usuario sobre las propuestas y módulo seis, encargado de guardar la cartera y salir del programa.

En el módulo de interacción con el usuario sobre las propuestas se podrá elegir entre salir del programa o eliminar las propuestas existentes y volver al módulo uno para generar unas nuevas según la nueva cartera del usuario, ya que ésta habrá cambiado al llevarse a cabo la acción que pida el usuario, en caso de ser posible llevarla a cabo.

## 2. Descripción del proceso seguido para el desarrollo

Paso a detallar lo que he sacado en claro de cada entrevista con el experto y el directivo y los usuarios cuando estuvieron presentes.

### 2.1. Sesión 1: Entrevista informal general

En esta entrevista han estado presentes directivo, experto y usuarios. Ha sido la primera entrevista, ha sido informal y ha sido principalmente para tener una idea general sobre qué se desea que haga el sistema experto.

El **directivo** nos pide un sistema que ayude a los novatos sobre cómo actuar. Estos agentes tienen una cartera de dinero y tienen que decidir cómo mover las acciones para obtener rentabilidad. Los agentes utilizan muchos datos, tales como el tipo de fondo que se maneja, situación económica actual y situación económica de la cartera de acciones. Quiere que el novato, con darle a un botón, obtenga una proposición de venta o compra de acciones y el por qué de ese razonamiento. Se opera con los datos del Ibex 35 y el proceso suele ser analizar la situación, analizar valores de caída, valores prometedores y situación política, y en base a eso decidir si se invierte o no. Hay también unos valores más estables que otros. Nos pide también que como mucho aparezcan 5 valores en los que invertir o vender (de más importante a menos importante) y que informe también del riesgo de la acción y el beneficio esperado. Además los valores que tengamos invertidos y que preveamos que van a caer hay que venderlos. Una de las entradas será la cartera actual, que se leerá de un fichero al iniciar el programa. Una vez que se toma una decisión el sistema deberá actualizar la cartera.

El **experto** nos dice que los principales datos de los que se dispone es de la evolución de los valores a 6 meses y a un año, y que los mismos se cogen de la bolsa de Madrid. Además se suele hacer un filtro de los valores prometedores, los que van a tener una caída o los que van a tener una subida. Una vez hecho esto se distinguen los valores que están en caída real o no y los que están en subida real o no.

Le pedimos al **usuario** que sea él el que introduzca la situación política general y particular de cada valor, que acabarán siendo noticias sobre la economía. Nos pide que el programa especifique explícitamente por qué toma cada decisión.

El **esquema conceptual** sacado en claro de esta sesión, es decir, lo que hace el experto a alto nivel, es lo siguiente:

1. Ver si hay que vender algo urgentemente.
2. Detectar valores prometedores.
3. Clasificar subidas y bajadas en falsos positivos o positivos.
4. Detectar valores estables.

## **2.2. Sesión 2: Entrevista formal general**

Esta segunda entrevista se ha llevado a cabo con el experto y ha sido de tipo formal. Los objetivos de esta sesión han sido obtener el esquema general de funcionamiento del sistema: de qué módulos se compone, el funcionamiento de cada módulo y la interacción entre los módulos.

Fases del proceso de cálculo:

1. Detección de valores inestables
2. Detección de valores peligrosos
3. Detección de valores infravalorados
4. Detección de valores sobrevalorados

Además, tendrá que haber un módulo previo que se encargue de recoger los datos y un módulo posterior que se encargue de proponer las mejores acciones al usuario. El orden de los módulos debe ser este ya que el primer módulo de recogida de datos es necesario para todos los demás, y los valores inestables serán necesarios para calcular los peligrosos, de la misma forma que los peligrosos serán necesarios para calcular los infravalorados y los sobrevalorados. La cartera de acciones son aquellas acciones que en ese momento posee y se introduce de entrada en el sistema. La cartera consta del dinero que tienen libre y el que tiene en acciones y el número de acciones que tiene en cada valor y cuánto dinero sería.

La detección de valores peligrosos buscará aquellos valores que se prevean en caída libre a corto plazo. Los valores prometedores son los valores infravalorados. Para comprar es necesario que haya suficiente dinero en la cartera o vender previamente. Se utilizarán los datos de la bolsa a la hora del cierre del día anterior.

## **2.3. Sesión 3: Entrevista formal sobre el módulo de detectar valores peligrosos y el módulo detectar valores inestables**

Para el módulo de valores inestables se tendrá en cuenta si ha habido noticias malas sobre un valor concreto o sobre un sector (si es sobre el sector se extenderá a todos los valores de ese

sector), ya que una noticia mala pondrá un valor inestable durante dos días. Del mismo modo, si un valor era inestables, una noticia buena pondrá dicho valor estable durante dos días. Si hay una noticia sobre un valor y sobre un sector prevalece la noticia sobre el valor. Si hay una noticia buena y una mala sobre un mismo valor, prevalece la buena. Además, los valores del sector de la construcción serán inestables por defecto y si la economía está ajando, los valores del sector servicios serán inestables por defecto.

Es importante tener en cuenta que para el módulo de detectar valores peligrosos es necesario que ya se haya hecho la actualización de valores inestables.

Si un valor no es inestable, se considera peligroso si el valor está cayendo durante cinco días y la diferencia entre la variación media del sector y la variación del valor sea menor del -5 %. Por otro lado, si el valor es inestable y está cayendo durante 3 días, el valor se considera peligroso.

## **2.4. Sesión 4: Entrevista formal sobre los módulos de detección de valores infravalorados y sobrevalorados**

El objetivo de esta sesión ha sido sacar en claro cuándo un valor está infravalorado o sobrevalorado y lo obtenido ha sido lo siguiente:

En cuanto al módulo de detección de valores infravalorados, se considerará que un valor lo es si su PER es bajo y el RPD es alto, si la empresa ha caído más de un 30 % en los últimos 3, 6 o 12 meses y ha subido entre un 5 % y un 10 % en el último mes y el PER es bajo o si la empresa es grande, tiene el RPD alto, el PER mediano, no ha bajado en cinco días y la variación con respecto a su sector es mayor que 5.

En cuanto al módulo de detección de valore sobrevalorados, un valor lo estará si su PER es alto y el RPD bajo, si la empresa es pequeña y tiene el PER alto o si la empresa es pequeña y tiene el PER mediano pero el RPD bajo. Además, si la empresa es grande, tiene el RPD bajo, el PER mediano o alto, o si la empresa es grande, tiene el RPD medio y el PER alto, también se considerarán valores sobrevalorados.

## **2.5. Sesión 5: Entrevista formal sobre el módulo de obtener propuestas para el usuario**

El objetivo de esta sesión es encontrar todas las posibles propuestas que se podrán dar al usuario, en base a los valores que se han obtenido en los módulos anteriores. Por esto, es necesario que este módulo se ejecute después de los cuatro anteriores.

Lo obtenido ha sido que hay cuatro posibles casos en los que conviene comprar o vender valores, que son los siguientes:

1. Si la empresa es peligrosa, ha bajado en el último mes y ha bajado más de un 3 % con respecto a su sector en ese último mes, proponer vender las acciones que se tengan de dicha empresa, ya que habrá posibilidad de que esta empresa caiga al cabo de un año un 20 %.

2. Si la empresa está infravalorada y se dispone de dinero, comprar acciones de dicha empresa.
3. Si una empresa está sobrevalorada y el rendimiento por año es menor que 5 más el precio del dinero, proponer vender las acciones que se tengan de dicha empresa.
4. Si una empresa no está sobrevalorada y su RPD es mayor que la revalorización por año esperado más su RPD más 1 de una empresa de mi cartera que no está infravalorada y no poseo ya acciones de la primera empresa, proponer cambiar las acciones de las empresas.

Además, a la hora de comprar y vender valores de una empresa es necesario tener en cuenta que hay que pagar un 0.05 % de lo que se compra y/o vende.

## 2.6. Procedimiento de validación y verificación del sistema seguido

Con respecto a la verificación, a lo largo del desarrollo del sistema experto ha habido varios fallos que se han corregido después, como entrar en un bucle infinito con las noticias, que un valor aparezca como sobrevalorado dos veces o que una regla no llegase a ejecutarse nunca. Estos fallos ya están corregidos en la versión final y ahora el SE funciona correctamente. No presenta inconsistencias estructurales, lógicas ni semánticas que se hayan detectado.

En cuanto a la validación, creo que la interfaz es comprensible para el usuario, la explicación del razonamiento es suficiente y cumple los requisitos de ejecución en tiempo real. Además, la comunicación entre módulos es la apropiada. Se han hecho varios casos de prueba con el experto y los resultados han sido los esperados, con lo que se considera que el sistema está validado.

## 3. Descripción del sistema desarrollado

### 3.1. Variables de entrada del problema

Las variables de entrada necesarias serán un fichero llamado *Analisis.txt*, que contendrá los valores de todas las variables necesarias (explicadas a continuación) para cada valor de la bolsa, un fichero llamado *AnalisisSectores.txt* con el valor de las variables para los sectores (también explicadas a continuación), un fichero llamado *Cartera.txt* que contendrá las acciones que tiene el usuario en su cartera, junto con el saldo disponible y por último un fichero llamado *Noticias.txt* que contendrá las noticias que se han dado en los dos últimos días. Además, se le solicitará al usuario si quiere añadir más noticias.

En todos los casos el formato de los ficheros será el valor de las variables que se detallan a continuación para cada fichero donde los valores de cada valor con el siguiente se separan por un salto de línea.

Las variables en *Analisis.txt* son las siguientes:

1. *Nombre*: nombre de la empresa.
2. *Precio*: precio al cierre de la sesión anterior.

3. *VariacionDiaAnterior*: % variación del precio con respecto al día anterior.
4. *Capitalizacion*: valor total de la empresa.
5. *PER*: Capitalización dividido por beneficios anuales obtenidos por la empresa.
6. *RPD*: repartido a los accionistas por dividendos (anual).
7. *Tamano*: toma los valores Pequenio (si el porcentaje del ibex es menor a 2), Mediano (entre 2 y 5) y Grande (mayor de 5).
8. *PorcentajeIbex*: capitalización con respecto a la capitalización total del ibex.
9. *EtiquetaPER*: toma los valores Alto si es mayor de 18, Medio si está entre 12 y 18 y Bajo si es menor que 12.
10. *EtiquetaRPD*: toma los valores Alto si es mayor que el 5 %, Medio si está entre 2 y 5 y Bajo si es menor que el 2 %.
11. *Sector*: nombre del sector al que pertenece la empresa.
12. *Variacion5Dias*: % variación del precio respecto al de hace 5 días.
13. *Bajada3Dias*: toma los valores true o false según haya estado bajando los últimos 3 días.
14. *Bajada5Dias*: toma los valores true o false según haya estado bajando los últimos 5 días.
15. *Variacion5DiasSector*: % Var 5 dias - % VarSector 5 dias.
16. *Variacion5DiasSectorMenor5*: toma los valores true o false según % variación con respecto al sector en los últimos 5 días sea menor que -5.
17. *VariacionPrecio1Mes*: % variación del precio respecto al de hace 1 mes.
18. *VariacionPrecio3Meses*: % variación del precio respecto al de hace 3 meses.
19. *VariacionPrecio6Meses*: % variación del precio respecto al de hace 6 meses.
20. *VariacionPrecio1Ano*: % variación del precio respecto al de hace 12 meses.

Las variables en *AnalisisSectores.txt* son las siguientes:

1. *Nombre*: nombre del sector.
2. *VariacionDia*: media del % de variación del día en las empresas del sector.
3. *Capitalizacion*: suma de la capitalización de las empresas del sector.
4. *PER*: media del PER de las empresas del sector.
5. *RPD*: % reparido por dividendos del último año en media de las empresas del sector.

6. *PorcentajeIbex*: % capitalización con respecto a la capitalización total de ibex.
7. *Variacion5Dias*: media % variación 5 días en las empresas del sector.
8. *Bajada3Dias*: toma los valores true o false según el sector haya estado bajando los tres últimos días o no.
9. *Bajada5Dias*: toma los valores true o false según el sector haya estado bajando los cinco últimos días o no.
10. *Variacion1Mes*: media % variación del último mes en las empresas del sector.
11. *Variacion3Meses*: media % variación del último trimestre en las empresas del sector.
12. *Variacion6Meses*: media % variación del último semestre en las empresas del sector.
13. *Variacion1Ano*: media % variación de los últimos 12 meses en las empresas del sector.

Las variables en *Cartera.txt* son las siguientes:

1. *Nombre*: nombre de la empresa a la que pertenecen las acciones o *Disponible* para el saldo disponible.
2. *Acciones*: cantidad de acciones poseídas de la empresa.
3. *ValorActual*: precio actual de las acciones que hay de esa empresa.

Las variables en *Noticia.txt* son las siguientes:

1. *Nombre*: Nombre de la empresa a la que hace referencia la noticia.
2. *Tipo*: toma los valores Buena o Mala según sea la noticia.
3. *Dias*: días desde que la noticia se emitió (1 ó 2).

### 3.2. Variables de salida del programa

La salida del programa será reescribir el fichero *Cartera.txt* actualizando con los cambios que el usuario haya decidido hacer (o con lo que ya había si no ha querido realizar ninguna acción).



### 3.3. Conocimiento global del sistema

Los hechos y relaciones que se cargan inicialmente son los siguientes:

1. Un hecho por cada valor del Ibex 35, conteniendo las variables presentes en el fichero *Analisis.txt* comentadas en el apartado 1 de esta sección.
2. Un hecho por cada distinto sector a los que pertenecen los valores del Ibex 35, conteniendo las variables presentes en el fichero *AnalisisSectores.txt* comentadas en el apartado 1 de esta sección.
3. Un hecho por cada noticia presente en el fichero *Noticias.txt* conteniendo el valor al que hace referencia, si es buena o mala y la cantidad de días (1 o 2) que hace que se emitió dicha noticia.
4. Un hecho por cada valor que el usuario tenga en la cartera que se haya leído del fichero *Cartera.txt* conteniendo el nombre del valor, la cantidad de acciones que se poseen junto con su valor actual y la cantidad de dinero disponible en la cartera.
5. Un hecho con el precio del dinero actual, que se puede cambiar al inicio del fichero *practica.clp* si alguna vez cambia, ya que está definido en un **def facts**.
6. Un hecho por cada valor deducido como inestable.
7. Un hecho por cada valor deducido como peligroso junto con la explicación de por qué se considera peligroso.
8. Un hecho por cada valor deducido como infravalorado junto con la explicación de por qué se considera infravalorado.
9. Un hecho por cada valor deducido como sobrevalorado, junto con la explicación de por qué se considera sobrevalorado.
10. Un hecho por cada propuesta que se puede hacer al usuario conteniendo el tipo de la propuesta (si es para comprar, vender o cambiar unas acciones por otras) el valor o los valores involucrados, el rendimiento que se sacaría de dicha acción y una explicación sobre por qué se da esa propuesta.

Una vez se han cargado todos estos hechos se le presentan al usuario las cinco mejores propuestas que se pueden hacer (aquellas cinco que tengan un mejor rendimiento) y se le pregunta si desea hacer cualquiera de estas acciones o por el contrario otra cosa distinta o si desea salir. En caso de que quiera llevar a cabo una acción se le pregunta cuál y qué empresa quiere vender y/o comprar y cuántas acciones de cada una. A continuación se llevan a cabo dichas acciones en caso de que se disponga de dinero suficiente para comprar las acciones pedidas en caso de que sea comprar y se disponga de acciones suficientes para vender si es vender. En caso de que no haya dinero o acciones suficientes se muestra un mensaje por pantalla indicándolo y no se realiza dicha acción. Después se le pregunta si quiere hacer más acciones o quiere salir del programa. En caso de querer realizar más acciones se eliminan las propuestas y se recalculan unas nuevas en función de los nuevos valores que se tienen en cartera y se vuelven a presentar las cinco mejores. Si quiere salir lo que se hace es guardar la cartera tal como esté en ese momento y salir del programa.

### 3.4. Especificación de los módulos que se han desarrollado

#### 3.4.1. Módulo inicial: leer datos de los ficheros

**OBJETIVO:** Recoger los datos iniciales necesarios para el correcto funcionamiento del SE. Se leen desde fichero en este módulo los valores junto con sus variables, los sectores junto con sus variables, la cartera del usuario y las noticias, según se ha indicado en apartados anteriores.

**CONOCIMIENTO QUE UTILIZA:** Este módulo no necesita conocimiento previo para ejecutarse, sólo necesita que los ficheros *Analisis.txt*, *AnalisisSectores.txt*, *Noticias.txt* y *Cartera.txt* estén en la misma carpeta en la que está el fichero *practica.clp*.

**CONOCIMIENTO QUE SE DEDUCE:** Este módulo no deduce conocimiento, simplemente inserta en la base de hechos un hecho por cada valor, un hecho por cada sector, un hecho por cada noticia y un hecho por cada valor del que se tengan acciones en la cartera, de la forma en la que se explica en la sección 3.3.

#### 3.4.2. Módulo Cero

**OBJETIVO:** Este módulo tiene como objetivo deducir los valores inestables. Por defecto, todo aquel valor que no aparezca en un hecho del tipo (Inestable (Nombre *NombreValor*)) será estable.

**CONOCIMIENTO QUE UTILIZA:** Este módulo necesita que el módulo inicial descrito en el apartado anterior ya haya terminado de ejecutarse, ya que son necesarios todos los valores, todos los sectores y todas las noticias. Las noticias son lo más relevante aquí ya que serán las que pongan un valor a estable (es decir, que quiten que es inestable) o inestable según si hay noticias sobre ese valor o sobre su sector y según sean buenas o malas. En caso de que haya más de una noticia sobre un valor y sean contradictorias prevalecerá la buena, mientras que si hay una noticia sobre un valor concreto y otra sobre un sector y son contradictorias prevalecerá la noticia sobre el valor concreto. Si un sector es inestable todos los valores de ese sector serán inestables por defecto.

**CONOCIMIENTO QUE SE DEDUCE:** El conocimiento que se deduce son aquellos valores que son inestables. Todo valor que no se deduzca que es inestable (o que deje de serlo en algún momento) será estable.

#### 3.4.3. Módulo Uno

**OBJETIVO:** El objetivo de este módulo es detectar los valores peligrosos que el usuario tiene en la cartera. Un valor puede ser peligroso porque esté en bajada durante 5 días y caiga más de un 5 % de su sector o porque sea inestable y esté en bajada durante 3 días.

**CONOCIMIENTO QUE UTILIZA:** Por tanto, dicho módulo necesita como conocimiento previo aquel que da el módulo cero (valores inestables) y por tanto el módulo cero deberá ejecutarse antes que este. Además necesita el conocimiento que ya hay en la base de hechos sobre los

valores y los sectores.

CONOCIMIENTO QUE SE DEDUCE: El conocimiento deducido aquí son aquellos valores pertenecientes a la cartera del usuario que son peligrosos. Para aquellos que lo sean se insertará en la base de hechos un hecho del tipo (Peligroso (Nombre *NombreValor* Explicacion *explicación*)).

#### 3.4.4. Módulo Dos

OBJETIVO: El objetivo de este módulo es detectar los valores sobrevalorados.

CONOCIMIENTO QUE UTILIZA: El conocimiento que utiliza este módulo es el de los valores y los sectores, por lo que no es necesario que se ejecute después de los módulos anteriores, aunque se ha elegido hacerlo así para que quede más claro el procedimiento.

CONOCIMIENTO QUE SE DEDUCE: Se deducen aquellos valores que están sobrevalorados, es decir, que se prevee que van a dar menos rendimiento del que parece a priori.

#### 3.4.5. Módulo Tres

OBJETIVO: El objetivo de este módulo es detectar los valores infravalorados.

CONOCIMIENTO QUE UTILIZA: El conocimiento que utiliza este módulo es el de los valores y los sectores, por lo que no es necesario que se ejecute después de los módulos anteriores, aunque se ha elegido hacerlo así para que quede más claro el procedimiento.

CONOCIMIENTO QUE SE DEDUCE: El conocimiento que se deduce son aquellos valores que están infravalorados, es decir, aquellos que se prevee que van a dar más rendimiento del que puede parecer.

#### 3.4.6. Módulo Cuatro

OBJETIVO: El objetivo de este módulo es encontrar todas las posibles propuestas que se le puedan dar al usuario porque vayan a tener un mejor rendimiento de lo que actualmente se tiene en cartera (bien sea cambiando valores como vendiendo o comprando valores de los que no se tengan previamente acciones o de los que sí).

CONOCIMIENTO QUE UTILIZA: El conocimiento que necesita este módulo son los valores sobrevalorados e infravalorados, por lo que los módulos dos y tres deben ejecutarse antes que este.

CONOCIMIENTO QUE SE DEDUCE: Se deducen las propuestas que se le pueden hacer al usuario y se insertan en la base de hechos con hechos del tipo (Propuesta (Tipo Valor Rendimiento Explicacion)).

#### 3.4.7. Módulo Cinco

OBJETIVO: El objetivo de este módulo es encontrar las cinco mejores propuestas y mostrarlas al usuario por pantalla para posteriormente pedirle una acción entre *Comprar*, *Vender*,

*Cambiar y Salir*. Además, si la acción está entre las tres primeras, las lleva a cabo preguntado valor a comprar/vender y cantidad de acciones a comprar/vender. Es el que maneja la interacción con el usuario. También tiene la función de que, si no se desea salir del programa, volver al módulo uno para recalcular valores inestables, peligrosos, sobrevalorados e infravalorados (en caso de que cambiara alguno) y volver a calcular las propuestas en base a la cartera nueva.

**CONOCIMIENTO QUE UTILIZA:** Este módulo utiliza el conocimiento resultante del módulo cuatro, es decir, las posibles propuestas que se le van a hacer al usuario, además del conocimiento sobre los valores y los sectores.

**CONOCIMIENTO QUE SE DEDUCE:** Los nuevos valores que se compran o venden y la cantidad de acciones de los mismos, es decir, se llevan a cabo las acciones que indica el usuario y se cambian en la base de hechos.

### 3.4.8. Módulo Seis: Salir del programa

**OBJETIVO:** Guardar la cartera con las modificaciones que se hayan hecho y cerrar el programa.

**CONOCIMIENTO QUE UTILIZA:** Los hechos del tipo (Cartera (Nombre *NombreValor* Acciones *NumAcciones* ValorActual *ValorAcciones*)) que son los referentes a la cartera para guardarlos en el archivo *Cartera.txt*

**CONOCIMIENTO QUE SE DEDUCE:** La cartera final con el efecto de las acciones que ha elegido el usuario sobrescribiendo el archivo *Cartera.txt*.

## 3.5. Estructura de funcionamiento del esquema de razonamiento del sistema

Como se ha ido comentando durante el apartado anterior, hay módulos que tienen que ser secuenciales por dependencias unos con otros, como el módulo inicial de lectura, que tiene que ejecutarse antes que cualquier otro módulo, o el de detección de valores inestables, que tiene que ejecutarse antes que el de detección de valores peligrosos. En concreto, los únicos dos módulos que no han de ser secuenciales son los módulos tres y cuatro, que es suficiente con que se ejecuten después del módulo inicial y antes del módulo cinco. Sin embargo, la estructura seguida es secuencial para facilitar la lectura del programa y porque realmente no había diferencia en que esos dos módulos se ejecutaran de forma paralela en cuanto a tiempos de ejecución. Por tanto, el orden en el que actúan los módulos es módulo inicial - módulo cero - módulo uno - módulo dos - módulo tres - módulo cuatro - módulo cinco - módulo seis (salir).

## 3.6. Lista de hechos que utiliza el sistema durante su funcionamiento y la forma de representarlos

Para representar los hechos he utilizado `deftemplate` en la mayoría de los casos (en todos excepto para el precio del dinero y para hechos que iba a borrar seguidamente, como para pasar de leer un archivo a cerrarlo). Por tanto, hay un `deftemplate` para Valor, otro para

Sector, otro para Cartera, otro para Noticia, otro para Inestable, otro para Peligroso, otro para Sobrevalorado, otro para Infravalorado, otro para Propuesta y otro para Modulo, ya que la gestión de los módulos la hago condicionando a que haya en la base de hechos un hecho del tipo (Modulo (Numero *numeroDelModulo*)). Cada **deftemplate** tiene tantos **fields** como variables hubiera que guardar de cada tipo en cuanto a Valor, Sector, Noticia y Cartera. Para Inestable sólo guardo el nombre del valor, para Peligroso, Sobrevalorado e Infravalorado el nombre y la explicación de por qué los valores son de ese tipo y para Propuesta el tipo de la propuesta, el nombre del valor al que hace referencia, el rendimiento que conllevaría comprar o vender dicho valor y la explicación de por qué es conveniente hacerlo.

## 3.7. Hechos y reglas de cada módulo

### 3.7.1. Módulo inicial

Como se ha comentado previamente, el objetivo de este módulo es leer los datos de los ficheros. Por tanto, las reglas pertenecientes a este módulo son para abrir, leer y cerrar los cuatro ficheros. Hay una regla para abrir cada fichero, una para fichero para seguir leyendo y una para cada fichero para cerrar. En la regla para leer es en la que se van creando los hechos guardando para cada **field** su valor, que será el que aparezca en el fichero correspondiente. Por ejemplo, la regla para leer el fichero de sectores es la siguiente:

```
(defrule LeerSectoresFromFile
  (declare (salience 46))
  ?f <- (SeguirLeyendoSectores)
=>
  (bind ?NombreSector (read datosSectores))
  (retract ?f)
  (if (neq ?NombreSector EOF) then
    (assert (Sector
      (Nombre ?NombreSector)
      (VariacionDia (read datosSectores))
      (Capitalizacion (read datosSectores))
      (PER (read datosSectores))
      (RPD (read datosSectores))
      (PorcentajeIbex (read datosSectores))
      (Variacion5Dias (read datosSectores))
      (Bajada3Dias (read datosSectores))
      (Bajada5Dias (read datosSectores))
      (Variacion1Mes (read datosSectores))
      (Variacion3Meses (read datosSectores))
      (Variacion6Meses (read datosSectores))
      (Variacion1Año (read datosSectores))))
    (assert (SeguirLeyendoSectores)))
  )
```

Como vemos, se rellena cada hecho con todos sus campos y se pasa al siguiente hecho (al siguiente sector en este caso) hasta que se encuentra un final de fichero.

### 3.7.2. Módulo Cero

El objetivo de este módulo es detectar los valores inestables (cualquier valor que no tenga asociado un hecho del tipo (Inestable (Nombre *nombreValor*)) será estable por defecto). Consta de 8 reglas, además de las reglas para entrar y salir de dicho módulo.

1. *InestablePorSectorConst*: es la regla para insertar el hecho de que el sector construcción es inestable.
2. *InestableServiciosPorEconomia*: es la regla para insertar el hecho de que el sector servicios es inestable si la economía lleva en bajada 5 días, es decir, si el sector Ibex lleva en bajada 5 días.
3. *InestablePorSector*: es la regla para insertar un hecho inestable por cada valor dentro de un sector marcado como inestable.
4. *EstablePorNoticiaSector*: es la regla para borrar que un hecho es inestable si hay una noticia buena sobre su sector y no hay una mala sobre el propio valor.
5. *EstableSectorPorNoticiaSector*: es la regla para borrar que un sector es inestable si hay una noticia buena sobre él.
6. *EstablePorNoticiaValor*: es la regla para borrar que un valor es inestable si hay una noticia buena sobre él.
7. *InestablePorNoticia*: es la regla para insertar un hecho inestable sobre un valor si hay una noticia mala sobre el mismo.
8. *InestablePorEconomia*: es la regla para insertar un hecho inestable sobre un valor si hay una noticia mala sobre la economía general, es decir, sobre el sector Ibex.

### 3.7.3. Módulo Uno

El objetivo de este módulo es detectar los valores peligrosos. Consta de dos reglas:

1. *PeligrosoInestable*: regla para insertar un hecho del tipo (Peligroso (Nombre *nombreValor* Explicacion *explicacion*)) si un valor de la cartera del usuario es inestable y ha estado bajando durante 3 días consecutivos.
2. *PeligrosoBajada5*: regla para insertar un hecho para un valor peligroso si pertenece a la cartera del usuario, ha estado bajando durante cinco días y la variación con respecto a su sector ha sido menor que -5.

### 3.7.4. Módulo Dos

El objetivo de este módulo es detectar valores sobrevalorados. Consta de cinco reglas:

1. *SobrevaloradoGeneral*: inserta un hecho del tipo (Sobrevalorado (Nombre *nombreValor* Explicacion *explicacion*)) si el PER de ese valor es alto y el RPD es bajo. Además se comprueba que no exista ya un hecho (Sobrevalorado ...) con el mismo nombre de valor y otra explicación, ya que una misma empresa puede activar distintas reglas porque no son excluyentes.

2. *SobrevaloradoEmpPeq*: inserta un hecho para un valor sobrevalorado si la empresa es pequeña, tiene el PER alto y no está sobrevalorada ya por otra causa.
3. *SobrevaloradoEmpPeq2*: inserta un hecho para un valor sobrevalorado si la empresa es pequeña, tiene el PER medio, el RPD bajo y no está ya sobrevalorada por otra causa.
4. *SobrevaloradoEmpGra*: inserta un hecho para un valor sobrevalorado si la empresa es grande, tiene el RPD bajo, el PER medio o alto y no está ya sobrevalorada por otra causa.
5. *SobrevaloradoEmpGra2*: inserta un hecho para un valor sobrevalorado si la empresa es grande, tiene el RPD medio, el PER alto y no está ya sobrevalorada por otra causa.

### 3.7.5. Módulo Tres

El objetivo de este módulo es detectar valores infravalorados. Consta de tres reglas:

1. *InfravaloradoGeneral*: inserta un hecho del tipo (Infravalorado (Nombre *nombreValor* Explicacion *explicacion*)) si dicho valor tiene el PER bajo y el RPD alto.
2. *InfravaloradoCaida*: inserta un hecho para un valor infravalorado si dicho valor tiene el PER bajo, ha caído más de un 30 por ciento en los últimos 3, 6 o 12 meses y ha subido entre un 5 y un 10 % en el último mes.
3. *InfravaloradoGra*: inserta un hecho para un valor infravalorado si la empresa es grande, tiene el RPD alto, el PER mediano, no ha bajado consecutivamente durante los últimos cinco días y la variación con respecto a su sector no es menor que -5 %.

### 3.7.6. Módulo Cuatro

El objetivo de este módulo es obtener todas las posibles propuestas que se le pueden hacer al usuario. Consta de cuatro reglas:

1. *PropuestaPeligrosa*: es la regla que se encarga de insertar propuestas de tipo vender acciones de empresas peligrosas. Inserta dicha propuesta si el usuario tiene un valor en la cartera que sea peligroso cuya variación del precio en el mes anterior sea negativa y la diferencia de dicha variación con la variación de su sector sea menor que -3. El rendimiento de dicha propuesta es  $20 - 100 * RPD$ .
2. *PropuestaInfravalorada*: es la regla que se encarga de proponer invertir en empresas infravaloradas. Inserta dicha propuesta si hay una empresa infravalorada y hay dinero disponible en la cartera del usuario. El rendimiento de dicha propuesta es  $20 * (PER_{sector} - PER) / (100 * RPD)$ .
3. *PropuestaVenderSobrevalorada*: es la regla que se encarga de insertar una propuesta para vender una empresa sobrevalorada. Dicha regla se inserta si el usuario tiene un valor sobrevalorado tal que su variación en un año más  $100 * RPD$  sea menor que 5 más el precio del dinero. El rendimiento de esta propuesta es  $(-100 * RPD) + (100 * (PER - PER_{sector})) / (5 * PER)$ .

4. *PropuestaCambio*: es la regla que se encarga de proponer intercambiar un valor de la cartera del usuario por otro. Se inserta esta propuesta si un valor de la cartera del usuario no está infravalorado y hay otro valor que no tiene ya el usuario en la cartera y no está sobrevalorado de forma que  $100 * RPD\_nuevaEmp > 100 * RPD + variacion\_1Ano + 1$ . El rendimiento de dicha propuesta es  $100 * RPD\_nuevaEmp - (variacion\_1Ano + 1 + 100 * RPD)$ .

### 3.7.7. Módulo Cinco

Este módulo tiene como objetivo encontrar las cinco mejores propuestas y mostrárselas al usuario para seguidamente pedirle una acción (entre las que se recomiendan u otra a su elección). Es el módulo de interacción con el usuario. Se compone de 14 reglas:

1. *FindMax*: Encuentra la propuesta con mayor rendimiento si se han mostrado menos de cinco propuestas e inserta un hecho del tipo (Max *nombreValor rendimiento*).
2. *MostrarMax*: Muestra por pantalla la propuesta asociada al máximo que coincide con el hecho (Max *nombreValor rendimiento*) que se acaba de insertar y borra la propuesta asociada con este rendimiento máximo para poder volver a buscar el máximo y que no sea el mismo. Además, aumenta el contador de propuestas mostradas en 1.
3. *PedirAccion*: Pide una acción al usuario entre Comprar, Vender, Cambio y Salir y guarda la respuesta del usuario.
4. *PedirAccionVender*: Si la respuesta del usuario ha sido Vender, pide que indique qué valor desea vender y cuántas acciones quiere vender del mismo y se inserta un hecho indicando que se quiere vender. Después, pregunta si quiere que se vuelvan a generar propuestas o desea salir y en caso de querer salir se guarda el hecho (RespuestaUsuario Salir).
5. *PedirAccionComprar*: Si la respuesta del usuario ha sido comprar, pide que indique qué valor desea comprar y cuántas acciones del mismo desea comprar y se inserta un hecho indicando que se desea comprar. Después, pregunta si quiere que se vuelvan a generar propuestas o desea salir y en caso de querer salir se guarda el hecho (RespuestaUsuario Salir).
6. *PedirAccionCambio*: Si la respuesta del usuario ha sido cambio, pide que indique qué valor desea vender y el número de acciones a vender y el valor que desea comprar y el número de acciones a comprar y se insertan hechos tanto para vender como para comprar. Después, pregunta si quiere que se vuelvan a generar propuestas o desea salir y en caso de querer salir se guarda el hecho (RespuestaUsuario Salir).
7. *CambiarCarteraVender*: Si se ha insertado un hecho con que el usuario ha elegido vender un valor se comprueba que ese valor esté en la cartera y que la cantidad de acciones actual sea mayor o igual que la cantidad de acciones que se quieren vender. De ser esto así, se elimina el hecho de querer vender, el hecho de la cartera que había sobre ese valor y el hecho de la cartera con el dinero disponible y se inserta un hecho en la cartera con el valor disponible después de la operación, y si después de la venta siguen quedando acciones, se inserta un hecho de la cartera con las acciones que queden.



8. *ErrorVender*: Comprueba si no se disponen de tantas acciones como se desean vender y avisa al usuario mediante un mensaje en pantalla.
9. *CambiarCarteraComprar*: Si se ha insertado un hecho indicando que el usuario quiere comprar un valor, se comprueba que queda suficiente dinero en la cartera para la compra, que no existiera ya ese valor en la cartera y se actualiza el valor que queda disponible en la cartera y se inserta un hecho con el nuevo valor adquirido.
10. *CambiarCarteraComprarExistente*: variante del anterior en el que se compra una acción que ya existe en la cartera y se modifica, además de todo lo anterior, el nuevo valor con la suma de las acciones que se tenían antes más las que se compran ahora.
11. *ErrorComprar*: Comprueba si no se dispone de dinero suficiente para realizar la compra de las acciones y de ser así se avisa al usuario con un mensaje por pantalla.
12. *EliminarPropuestas*: Si el usuario no ha introducido en ningún momento que quiere salir del programa es porque quiere que se vuelvan a recalcular las propuestas. Por tanto, se comprueba si se ha introducido la opción de salir y de no ser así se eliminan todas las propuestas hechas y se introduce la opción de volver al módulo uno. Esto es así porque al cambiar la cartera cambiarán las propuestas y se volverán a calcular cuando vuelva a pasar por el módulo cuatro.
13. *SalirModulo5*: Si se ha introducido la opción de volver al módulo 1, se inserta el hecho (Modulo (Numero Uno)) y se borra el hecho (Modulo (Numero Cinco)).
14. *SalirPrograma*: Si se ha introducido la opción Salir, salta al módulo salir insertando el hecho (Modulo (Numero Salir)) y borrando (Modulo (Numero Cinco)).

### 3.7.8. Módulo Seis (Salir)

El objetivo de este módulo es guardar la cartera del usuario sobrescribiendo el archivo *Cartera.txt* con la cartera que se encuentre en ese momento en la base de hechos del programa, es decir, guardar todos los hechos del tipo (Cartera (...)) y salir del programa. Consta de cuatro reglas, una para abrir el archivo, otra para escribirlo, otra para cerrarlo y una última que se ejecuta cuando ya se ha hecho todo esto y sólo tiene la acción (exit).

## 4. Breve manual de uso del sistema

Para el correcto funcionamiento es necesario que el archivo *practica.clp* se encuentre en el mismo directorio que los archivos *Analisis.txt*, *AnalisisSectores.txt*, *Cartera.txt* y *Noticias.txt*. Alrededor de las 140 primeras ejecuciones son la lectura de datos de los ficheros, la generación de los valores inestables, peligrosos, sobrevalorados, etc. y la generación de propuestas junto con su impresión por pantalla. Después se pide al usuario que introduzca una acción (se dan las posibles opciones por pantalla) y se pide que indique el valor y la cantidad de acciones a mover y si se desea salir o continuar con el programa. Si seguimos haciendo **run** se cambiará la cartera con los nuevos movimientos hechos y se continuará hasta que el usuario indique que quiere salir, momento en el que se guardará la cartera y se cerrará el programa. Para todo

esto es necesario hacer `run`, y es aconsejable hacerlo poco a poco. Como he comentado después del primer (`reset`) hacer (`run 140`) e ir de 10 en 10 a partir de ahí, introduciendo los datos cuando se pidan.