

PRÁCTICA DESARROLLO DE SISTEMAS INTELIGENTE



PARTICIPANTES:
MIGUEL SALINAS EGEA (PORTAVOZ)
JOSÉ LUIS GALIANO GOMEZ
JOSÉ IGNACIO MARTÍNEZ-LOZANO PÉREZ
CURSO 2024/25

Índice

Estructura general.....	2
Mito de Perseo y Andrómeda	5
Mito de Teseo y el Minotauro	6
Caso Especial Heracles	7
Mito personalizado.....	8
Manual de usuario	9
Explicación escenarios fase 3.....	9

Estructura general

El código Java de este proyecto tiene por objetivo crear un Sistema Inteligente capaz de responder ciertas preguntas sobre ciertos mitos de la mitología griega, haciendo uso de diversas clases para ello.

Nuestro código está compuesto mayormente por 5 partes. La primera de ellas es el paquete *dsi*. Dentro de este paquete podremos encontrar varias clases de gran interés, como lo son:

- **BaseHechos:** en esta clase, creamos e inicializamos todos los personajes, objetos y relaciones que estarán al inicio de cada ejecución del programa. Usaremos varios mapas para poder estructurar y localizar de una forma más eficiente a los diversos personajes y a sus objetos. Además, es en esta clase en la que insertamos todos estos personajes y objetos en la propia base de hechos.
- **CondicionParada:** nuestro sistema inteligente debe ser capaz de responder distintos tipos de preguntas, en las que cada una querrá saber si somos capaces de realizar unas cosas u otras. Cada una de estas preguntas tendrá que llegar hasta cierto punto en la ejecución del programa para saber si puede ser satisfecha o no. Es por esto por lo que decidimos crear esta clase. Con esta clase, somos capaces de indicarle al programa la condición que queremos que se cumpla, y una vez cumplida que se detenga el programa. Para ello introduciremos el personaje que queremos que realice cierta acción, el objeto/personaje/capacidad objetivo de la pregunta, y la acción en si que provocará la parada (esta acción de parada será explicada en la clase TipoParada).
- **Lanzador:** con esta clase, seleccionamos la carpeta en la que estarán todos nuestros ficheros de entrada para los diversos escenarios de nuestros mitos, así como donde queremos que se guarden las respuestas para cada uno de estos escenarios. También es en esta clase en la que lanzamos todas las reglas para que se ejecuten en su respectivo orden mediante el comando *fireAllRules()*.
- **LectorFichero:** con esta clase, conseguimos procesar las diversas preguntas y condiciones de cada uno de los escenarios que se le planteen a nuestro sistema inteligente. En función del tipo de pregunta y de los personajes involucrados, así como de las diversas condiciones que se nos planteen, introduciremos ciertos objetos en nuestra base de hechos.
- **TipoParada:** esta clase es un enumerado, en el cual introducimos las diversas acciones que en el caso de cumplirse podamos acabar la ejecución de nuestro programa. Las condiciones son: que cierto personaje libere a otro cierto personaje, que cierto personaje obtenga un objeto en concreto, y por el último que cierto personaje obtenga una capacidad en específico.

El siguiente paquete del que está compuesto nuestro proyecto es el llamado *objeto*. En este paquete incluimos las clases que representan los diversos objetos que son usados en nuestros escenarios. Las clases de las que está compuesto este paquete son:

- **Objeto:** es una clase abstracta de la cual van a heredar todos los demás objetos. Sirve para representar simplemente la idea de objeto para nuestro sistema inteligente, y es por esto por lo que solamente tiene como parámetro su nombre.
- **Arma:** esta clase sirve para representar las diversas armas que tendrán nuestros personajes a lo largo de los diversos escenarios de nuestros mitos.
- **ObjetoNormal:** sirve para representar los objetos corrientes, que son aquellos que no conceden ninguna capacidad especial a su portador.
- **ObjetoEspecial:** esta clase es la más interesante de este paquete. Los objetos especiales son aquellos que conceden ciertas capacidades especiales a sus portadores. Algunos ejemplos de objetos de este estilo son: Sandalias Aladas, Escudo de Bronce, Casco de

Hades, etc. Las propiedades que concede cada objeto a su portador son representadas en la clase Propiedad.

- **Propiedad:** esta clase es un enumerado que nos sirve para mostrar todas las propiedades que pueden conceder los diversos objetos especiales. Estas propiedades son: invisibilidad (Casco de Hades), petrificar (Cabeza de Medusa), vuelo (Sandalias Aladas), reflejo (Escudo Espejo) y superfuerza (esta propiedad es especial y será explicada cuando hablemos de Heracles).

También tenemos el paquete *relación*, el cual nos ayuda a representar todas las posibles relaciones que pueden tener los personajes entre sí, así como las relaciones entre personajes y objetos. Las clases que conforman este paquete son:

- **Apresa:** esta clase sirve para representar la acción en la que un ser, el apresador, apresa a otro ser, que es el apresado. Con los objetos de esta clase, podremos representar diversas acciones en nuestro sistema, como por ejemplo cuando Minos apresa a Dédalo e Ícaro, o por ejemplo cuando Andrómeda cae presa por cierta Criatura Marina.
- **Capacidades:** esta clase sirve para representar si un ser en concreto posee cierta capacidad, o bien por poseer cierto objeto especial o por tenerla de forma intrínseca.
- **Enojo:** esta clase sirve para representar el hecho de que un ser en concreto, el enojador, enoja a otro personaje, el enojado, por diversos motivos.
- **Familia:** esta clase sirve para representar la relación padre-hijo que puede existir entre varios personajes de nuestra base de hechos. Esta relación nos será de especial utilidad sobre todo en el mito de Teseo y el Minotauro.
- **Favor:** esta clase sirve para representa si cierto Dios (el favorecedor), decide dar su favor a un Mortal (favorecido). Que un Mortal tenga el favor de un Dios le concede ciertos privilegios, como por ejemplo el hecho de que el Dios le dará sus objetos especiales a dicho mortal.
- **Libera:** esta clase nos sirve para representar el hecho de que cierto Ser (liberador) consigue liberar a otro Ser (liberado) de su apresamiento. Esta acción podrá ser accionada en diversas situaciones, como por ejemplo cuando nuestro héroe consigue matar a la criatura marina que apresa a Andrómeda.
- **Mata:** esta clase nos sirve para representar la acción en la que cierto Ser (matador) mata a otro Ser (matado). Como hemos dicho anteriormente, esto sucede cuando nuestro héroe consigue matar a la Criatura Marina que tiene apresada a Andrómeda, o por ejemplo cuando algún héroe atrapado en el laberinto consigue matar al Minotauro.
- **Posee:** esta clase es de gran interés, ya que nos sirve para representar el hecho de que cierto Ser (poseedor) tiene en su posesión un objeto. Esta relación nos es de gran utilidad ya que un ser tiene las capacidades que tienen los Objetos Especiales que posee. Además, nos sirve para saber si un Ser tiene los objetos necesarios para poder realizar ciertas acciones, como por ejemplo tener un Arma para poder matar al Minotauro.

También tenemos el paquete *ser*, en el cual se encuentran las clases que representan a todos los seres que usaremos en nuestros escenarios. Las clases de las cuales está compuesto este paquete son:

- **Ser:** esta es una clase abstracta que será usada como el padre de todas las demás clases de este paquete. Esta clase tiene 2 atributos, el nombre del ser y un atributo booleano que usaremos para representar si dicho ser está vivo o no.
- **Mortal:** esta clase también es abstracta, y nos sirve para representar a todos aquellos seres que pueden morir, es decir, que son mortales. En esta clase, a parte de los atributos de nombre y vivo heredados de la clase Ser, tenemos dos nuevos atributos que son `tiene_enojo_de` y `tiene_favor_de`. Estos atributos son listas que representan tal y como el

propio nombre indica, los seres los cuales están enojados y los seres que favorecen a dicho mortal.

- **Humano:** esta clase sirve para representar a los personajes que son humanos en nuestras historias. Los humanos son aquellos personajes cuyos padres, en caso de tenerlos en cuenta para la historia, no son ningún tipo de deidad. Esta clase hereda directamente de la clase Mortal, por lo tanto el único atributo de esta clase será el nombre del humano.
- **Semidios:** esta clase sirve para representar a los semidioses/héroes de nuestra base de hechos. Los semidioses son personajes especiales, ya que uno de sus padres es una deidad. Además, en nuestro caso son los semidioses los protagonistas de nuestras historias siendo los encargados de rescatar a Andrómeda. De esta forma, podemos decir que para nosotros los semidioses son el equivalente a los héroes. El único atributo que tendrá esta clase será el propio nombre del semidios.
- **Criatura:** esta clase sirve para representar a las criaturas que formarán parte de nuestra historia. Para ser más específicos y poder distinguir entre las criaturas marinas del mito de Perseo y al Minotauro, por separado tenemos la clase Criatura Marina, de la cual hablaremos más adelante. El único atributo de esta clase es el nombre de la criatura.
- **CriaturaMarina:** esta clase sirve para representar a las diversas criaturas marinas que pueden aparecer en el mito de Perseo (Ceto y el Kraken). Al igual que la clase anterior, esta clase solamente tiene un atributo que es el nombre de la criatura.
- **Dios:** esta clase sirve para representar a los diversos Dioses que aparecerán en nuestra historia. Esta clase hereda directamente de la clase Ser, sin embargo, también tiene 2 nuevos atributos. Tenemos las listas enojado_por y favorece_a, las cuales representan qué seres han hecho enojar al Dios y a qué seres favorece dicho Dios respectivamente. Esta segunda lista es de gran utilidad, ya que para el desarrollo del programa, los seres que son favorecidos por ciertos dioses reciben de estos ciertos objetos clave para el desarrollo de la historia. Los dioses tienen una cosa especial, y es que siempre estarán vivos, por lo tanto su atributo booleano de vivo nunca será modificado en ninguna de las reglas.
- **DeidadMayor:** esta clase sirve para representar a los dioses más importantes de la historia (Zeus, Poseidón, Hades o Atenea entre otros). Algunos de estos dioses tienen ciertos objetos especiales, los cuales darán a los seres a los cuales favorezcan.
- **DeidadMenor:** con esta clase representamos a las deidades menores, los cuales a nivel práctico son lo mismo que las deidades mayores, pero en la historia tienen un papel no tan crucial como las deidades mayores. Esta clase, al igual que las deidades mayores, tan solo tendrán como atributo su propio nombre.
- **Ninfa:** esta clase sirve para representar a las Ninfas, que son un tipo específico de deidad menor. Dentro de estas Ninfas encontramos algunas de gran importancia, como por ejemplo Doris o las Nereidas.
- **Gorgona:** esta clase sirve para representar a las Gorgonas, que son un tipo especial de Deidad Menor. Dentro de las Gorgonas, cabe destacar a Medusa la cual es una de las protagonistas de nuestra historia. Para las gorgonas, el único atributo que tendremos será de nuevo su nombre.

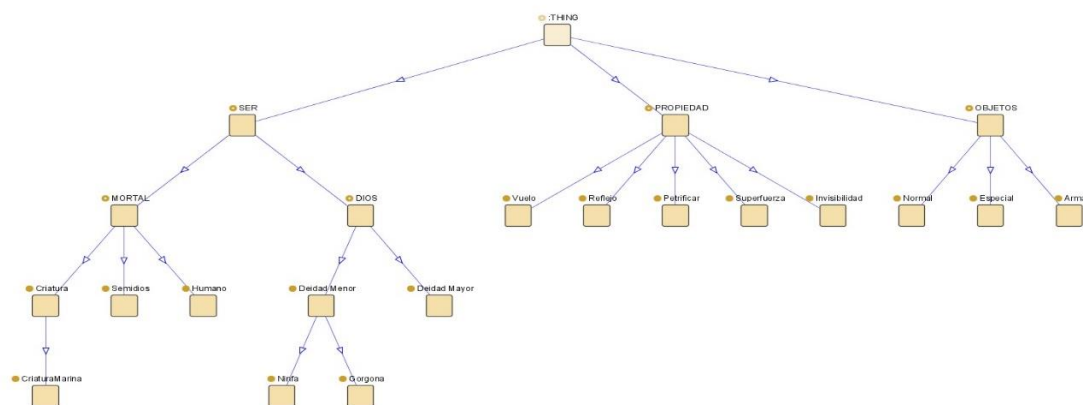


Ilustración 1 Esquema resumen de las clases

La última parte importante de nuestro código es la clase *ReglasDSI.drl*. En esta clase, es donde hemos programado todas las reglas que usa nuestro sistema inteligente. Para hacerlo más sencillo de leer, las reglas las hemos dividido en varios grupos en función de cuál sea el mito al que representan. Es por esto por lo que vamos a hablar de cada uno de estos conjuntos de reglas en su apartado correspondiente. Sin embargo, tenemos una serie de reglas genéricas que son utilizadas en cualquiera de los mitos y escenarios que se nos puedan plantear. Estas reglas genéricas son:

- 1.1 Mortal obtiene los objetos de los dioses que le favorecen: en esta regla, hacemos que aquellos mortales que son favorecidos por cierto Dios reciban los objetos de este Dios. Hay que tener en cuenta que una de las posibles condiciones de parada era que cierto personaje tuviese un objeto en concreto. Por lo tanto, dentro de esta regla introducimos una de las condiciones de parada.
- 1.2 Ser obtiene las capacidades de sus objetos especiales: con esta regla hacemos que el ser adquiera todas las propiedades que tengan cualquiera de sus objetos. Además, volvemos a introducir de nuevo una condición de parada ya que de estas condiciones de parada de nuestro programa era que un personaje obtuviera cierta capacidad.
- 1.3 Ser obtiene los objetos del Ser al que mata: con esta regla representamos que cuando un Ser mata a otro Ser, los objetos del Ser matado en vez de quedar olvidados pasan a ser posesión del Ser matador. De nuevo, lanzamos la condición de parada en la que un cierto Ser posee cierto objeto.
- 1.4 Ser libera al apresado de quien haya matado: mediante esta regla, hacemos que cuando cierto Ser es matado por otro Ser, y el Ser matado tenían algún prisionero estos quedan liberados. Con esta regla, vuelve a aparecer una condición de parada, ya que una de estas condiciones de parada era que cierto personaje fuese liberado por otro personaje en concreto.

Mito de Perseo y Andrómeda

En la primera parte del proyecto se nos planteó el mito de Perseo y Andrómeda, en el cual Andrómeda cae presa por la Criatura Marina Ceto, y Perseo tiene que rescatarla. Para la elaboración de esta parte del proyecto, creamos una serie de reglas en el fichero *ReglasDSI.drl*. Estas reglas son:

- 2.1 Poseidón libera a Criatura Marina: esta regla representa cuando Poseidón se enfada con Casiopea, y a raíz de este enojo libera a una Criatura Marina. Decimos Criatura Marina para que el sistema pueda escoger aleatoriamente entre Ceto y el Kraken.
- 2.2 Criatura marina apresada a Andrómeda: una vez la criatura marina es liberada, esta apresada a Andrómeda como castigo a Casiopea por haber enfadado a Poseidón.

- 2.3 Semidios obtiene el favor de las Grayas: en esta regla representamos todo lo que tiene que ver con las Grayas. Primero hacemos que el semidios pueda localizar a las Grayas gracias a tener el favor de Hermes y Atenea. Una vez localizadas las Grayas, si nuestro semidios tiene la propiedad de Invisibilidad, este le puede robar el Ojo a las Grayas. Por último, el semidios le devuelve el Ojo a las Grayas, ganándose de esta forma su favor.
- 2.4 Semidios obtiene favor Ninfas: las Grayas saben dónde se encuentran las Ninfas, por lo que si el semidios tiene el favor de las Grayas entonces podrá localizar a las Ninfas y podrán obtener sus objetos (en caso de que las Ninfas tengan algún objeto).
- 2.5 Semidios mata a Medusa: en el caso en el que el semidios tenga las propiedades de Vuelo, Reflejo e Invisibilidad, además de un Arma, entonces podrá ir a matar a Medusa.
- 2.6 Semidios muere si mata a Medusa sin el Zurrón Mágico: una vez el semidios ha conseguido matar a Medusa, es muy importante que este cuente con el Zurrón Mágico, el cual es un objeto que le deberían de haber otorgado las Ninfas. En el caso de que el semidios mate a Medusa pero no tenga el Zurrón Mágico, entonces morirá ya que sin él el semidios no puede tocar la Cabeza de Medusa.
- 2.7 Semidios mata a Criatura Marina: una vez el semidios ha conseguido matar a Medusa y tiene su cabeza, entonces puede ir a matar a la Criatura Marina con ella, ya que para matar a esta Criatura Marina hay que tener la capacidad de Petrificar que concede la Cabeza de Medusa.
- 2.8 Semidios obtiene a Pegaso: si nuestro semidios tenía el favor del Dios Hefestos, entonces este le proporciona al semidios las Bidas de Oro. Con estas Bidas de Oro, el semidios es capaz de capturar a Pegaso, siendo esta otra forma de conseguir la propiedad de Vuelo.

Con todas estas reglas conseguimos llevar a cabo y responder todos los posibles escenarios que se nos plantean para este primer mito. Un apunte importante que queremos destacar en estas reglas es el que nos encontramos con las reglas 2.3, 2.4, 2.5, 2.7 y 2.8. En estas reglas, hemos añadido una condición a la regla de que el héroe (en nuestro caso el semidios) que tuviese como objetivo salvar a Andrómeda, tenía que estar libre de cualquier tipo de apresamiento para poder realizar las distintas acciones explicadas en estas reglas. Esto es debido a que en la siguiente parte de la práctica, se nos introdujo el hecho de que este héroe podía caer preso en el laberinto y por lo tanto, no podría realizar ninguna de estas acciones.

Mito de Teseo y el Minotauro

En la última fase del proyecto también se nos introdujo el mito de Teseo y el Minotauro. Este mito nos hizo tener que introducir nuevos personajes a nuestra base de hechos. Y no solamente eso, sino que además nos hizo tener que generalizar todas las reglas. Esto se debe a que en los escenarios de la fase 3 se nos indica que debemos permitir que diferentes personajes puedan realizar las acciones principales de los dos mitos.

Las reglas que introdujimos para poder llevar a cabo la historia de este mito son:

- 3.1 Teseo recibe hilo: cuando Teseo es encerrado en el laberinto, él y Ariadna se enamoran y por tanto, Ariadna le da su hilo para así poder escapar del laberinto.
- 3.2 Semidios mata a Minotauro: en el caso de que haya un héroe (en nuestro caso un semidios) dentro del laberinto y este tenga un arma, entonces dicho semidios matará al Minotauro. Para poder matar al Minotauro, este debe estar vivo. Además, una vez matas al Minotauro, el semidios podrá recoger el Cuerno del Minotauro y guardarlo en su inventario.
- 3.3 Semidios escapa del laberinto con el hilo: una vez está muerto el Minotauro y el semidios posee el hilo de Ariadna, entonces dicho semidios puede escapar del laberinto

por su cuenta. De esta forma, y debido a que hacemos uso de la clase Libera, podemos decir que el semidios se libera a sí mismo del laberinto.

- 3.4 Minos se enfada con Dédalo: cuando el semidios consigue escapar del laberinto y matar al Minotauro, Minos se enfada mucho ya que supuestamente su laberinto era tan grande y complejo que nadie podía escapar de él. Por lo tanto, al conseguir escapar nuestro semidios, Minos se enfada mucho con Dédalo, que fue el arquitecto que diseñó el laberinto.
- 3.5 Minos apresa a sus enemigos y sus familias: cuando Minos se apresa con algún mortal, como castigo este le apresa a él y a todos sus hijos en el laberinto. De esta forma, podemos representar el hecho de que Minos apresa a Dédalo y a su hijo Ícaro cuando el semidios consigue escapar del laberinto.
- 3.6 Ser se encuentra las alas en el laberinto: cuando hay algún Ser apresado dentro del laberinto, pero el Minotauro está muerto, entonces este semidios consigue construir las alas, para lo cual necesitará reunir la cera y las plumas. Para simplificar la función, simplemente hacemos que el Ser consigue las alas, no que tenga que construirlas como tal.
- 3.7 Ser escapa del laberinto volando: en el caso de que haya algún Ser atrapado dentro del laberinto y este tenga la Capacidad de Vuelo a su disposición, entonces podrá escapar volando del laberinto. Originalmente esta regla fue diseñada para que solo se pudiese escapar volando del laberinto en el caso de que ese ser hubiese construido unas alas. Sin embargo, para poder generalizar con todos los escenarios de la fase 3, con tener la Capacidad de vuelo, sea de la forma que sea, el Ser podrá escapar del laberinto.
- 3.8 Ícaro vuela demasiado cerca del Sol: esta regla fue creada para poder representar correctamente la historia. Una vez Ícaro consigue construir sus alas y escapar volando del laberinto, este siente una gran alegría por haber ser libre y entonces empieza a volar cada vez más alto. Al ir tan alto, el calor emitido por el Sol hizo que la cera de sus alas se derritiera y por tanto, que sus alas se rompieran. Al romperse sus alas, Ícaro cae al suelo y muere.

En este conjunto de reglas, vamos a destacar una serie de cosas que son de interés. En la regla 3.1, hemos decidido que sea solamente Teseo quien pueda recibir el hilo de Ariadna. Esto es así ya que Ariadna se enamora solamente de Teseo, y por tanto solo tendrá la intención de ayudarlo a él a escapar del laberinto. Por otro lado, también es importante resaltar qué pasa con las alas cuando Ícaro cae y muere. Cuando muere Ícaro, consideramos que sus alas se quedan sin dueño tiradas en el suelo. Es por esto por lo que decidimos crear la regla 4.1, en la que un héroe (en nuestro caso semidios) que tenga la intención de salvar a Andrómeda (esto lo representamos con el hecho de que dicho semidios tiene en su inventario el Zurrón Mágico para poder guardar la Cabeza de Medusa) podrá recoger dichas alas para así poder adquirir la Capacidad de Vuelo y poder llegar hasta donde se encuentra Medusa.

Caso Especial Heracles

En la fase 3 también se nos introduce un nuevo héroe para nuestras historias, Heracles. Heracles es hijo de Zeus, lo cual le concede una gran cantidad de ventajas. La más importante de todas es que puede pedirle a su padre cualquier objeto, que él se lo concederá sin ningún problema. Para representar esto en nuestro sistema, hemos introducido a Heracles directamente con todas las Propiedades disponibles, representando que puede hacer cualquier acción gracias a los objetos que le concedería su padre. Por otro lado, también es importante hablar de otra característica importante de Heracles, que es su superfuerza. En la práctica se nos dice que Heracles cuenta con una fuerza sobrehumana, la cual le permite hacer entre otras muchas cosas matar al Minotauro sin la necesidad de tener un Arma. Para ello, en la clase Propiedad creamos una nueva propiedad llamada SUPERFUERZA, la cual solamente será poseída por Heracles. En la regla 3.2 podemos

ver reflejada esta idea, en la que el semidios en cuestión debe tener o bien un arma, o bien la Propiedad SUPERFUERZA a su disposición para poder matar al Minotauro. Para las Criaturas Marinas no hemos contemplado la posibilidad de que Heracles pudiese matarlas con su superfuerza ya que hemos supuesto que es necesario petrificar a dichas criaturas para poder matarlas.

Para mostrar que Heracles obtiene sus capacidades, creamos la regla 4.2. En esta regla volvemos a introducir de nuevo una condición de parada, ya que en el caso de que la condición de parada sea que Heracles obtiene cierta Capacidad, la respuesta debería ser verdadera y por tanto se debería detener el programa.

Mito personalizado

En la fase 3, en concreto en el escenario 8, se nos pide que creamos nuestra propia historia para probar las reglas que ya teníamos creadas, y en el caso de ser necesario crear nuevas reglas. La historia que creamos fue la siguiente:

- Siguiendo el mito de Teseo, Ariadna se enamora de Teseo y recibe su hilo antes de ser apresado en el laberinto, pero esta vez Teseo no consigue meter su arma a escondidas.
- Mientras tanto, en la comida de Dioses, Casiopea, en vez de decirle a Poseidón que Andrómeda era más guapa que sus hijas, se lo dijo a Minos, hablando de Ariadna. Minos se enfada con Casiopea y encierra en el laberinto del Minotauro a ella y a su hija, Andrómeda.
- Minos se acuerda de Perseo y, para evitar que intente salvar a Andrómeda, lo encierra también.
- Cefeo, el marido de Casiopea, le pide ayuda a Zeus. Zeus se apiada y le dice a su hijo, Heracles, que las salve a las dos. Para tratar de ayudarles a escapar, Heracles enoja a Minos apostando para ser encerrado en el laberinto y así poder ayudarles desde dentro.
- Heracles entra en el laberinto y mata al minotauro, haciendo así que Teseo pueda escapar del laberinto sin problemas. Teseo jura a Heracles que le devolverá el favor.
- La muerte del Minotauro enfada al Toro Blanco, padre del Minotauro. Como castigo, el Toro Blanco apresa a Heracles como venganza por haber matado a su hijo.
- Teseo le pide ayuda a Ariadna para salvar a Heracles, como agradecimiento por salvarle a él del laberinto. Ariadna propone pedirle ayuda a Perseo también. Perseo acepta la propuesta, y piensa en usar la Cabeza de Medusa para petrificar al Toro Blanco y así poder liberar a Heracles.
- Perseo obtiene la capacidad de petrificar matando a Medusa y guardando su cabeza en el Zurrón Mágico.
- Perseo, Teseo Y Ariadna colaboran y matan al Toro Blanco, consiguiendo liberar finalmente a Heracles.

Para poder llevar a cabo esta historia, tuvimos que introducir nuevos personajes en la base de hechos, como es el Toro Blanco. Además, también tuvimos que crear algunas reglas nuevas. Estas reglas son:

- 5.1 Heracles enoja a Minos para ser encerrado: en el caso de que Casiopea esté apresada por Minos en el laberinto, Heracles decide enojar a propósito a Minos para que este le encierre a él también en el laberinto.
- 5.2 Toro Blanco apresa a Heracles: en el caso de que el Minotauro esté muerto, el Toro Blanco apresa a aquel Ser que haya matado a su hijo. Para realizar correctamente esta regla, comprobamos que ni el Ser que ha matado al Minotauro ya esté preso ni que el propio Toro Blanco este apresado por otro ser.

- 5.3 Perseo y Teseo matan al Toro Blanco: en el caso en el que el Toro Blanco tenga apresado a Heracles, el resto de los héroes van a colaborar para salvarle. Comprobaremos si alguno de ellos (Perseo, Teseo o Ariadna) disponen de la Capacidad de Petrificar. En el caso de que si la tenga, juntos irá y matarán al Toro Blanco, consiguiendo liberar así a Heracles.

Manual de usuario

Para poder simular los escenarios, simplemente tendremos que darle al botón de ejecutar a la clase Lanzador. Todos los ficheros de entrada para todos los casos, tanto de la fase 2 como de la fase 3, se encuentran en una carpeta dentro del proyecto llamada *escenarios*. Por otro lado, las soluciones a cada uno de estos ficheros de entrada serán creadas en sus respectivos ficheros de salida en la carpeta *soluciones*, que también encontramos dentro de nuestro proyecto.

Explicación escenarios fase 3

En la fase 3 se nos planteaban diversos escenarios que nuestro sistema inteligente debía de ser capaz de superar. Estos escenarios fueron descritos mediante una serie de condiciones en los ficheros de entrada correspondientes a cada uno de estos escenarios de la siguiente forma:

- **Escenario 1:** este escenario era simplemente conseguir que funcionase correctamente el mito descrito en la fase 2, en el que Perseo tenía que liberar a Andrómeda ya que esta estaba apresada por Ceto.
- **Escenario 2:** este escenario era para comprobar que funcionaba correctamente el mito descrito en la fase 3, en el que Minos se enfadaba con Dédalo e Ícaro y los apresaba en el laberinto, y posteriormente estos conseguían escapar construyendo unas alas.
- **Escenario 3:** este escenario es muy similar al descrito en el escenario 1, pero en este caso será Teseo quien tenga que salvar a Andrómeda. Para poder conseguir esto, simplemente tuvimos que generalizar todas las reglas relacionadas con este primer mito para que no se refiriesen directamente a Perseo, sino a cualquier semidios.
- **Escenario 4:** este escenario es igual que el descrito en el escenario 1, con la única diferencia de cómo consigue Perseo la Capacidad de Vuelo. En este escenario, las Ninfas no tendrán las Sandalias Aladas, y Hefesto no tendrá las Bridas de Oro, por lo que es imposible que Perseo consiga volar sin las Alas de Ícaro. Por lo tanto, Dédalo e Ícaro tendrán que escapar del laberinto, y una vez muera Ícaro entonces Perseo se quedará con sus alas y podrá ir a por Medusa para matarla y así proseguir con el mito.
- **Escenario 5:** para este escenario, decidimos seguir las líneas ya explicadas para el escenario 1 con una única diferencia, que Perseo era apresado por Minos. Por lo tanto, antes de poder salvar a Andrómeda tendrá que escapar del laberinto. Para poder escapar del laberinto, también hicimos que Minos apresase a Teseo para que fuese él quien matase al Minotauro.
- **Escenario 6:** en este escenario debía ser un personaje femenino quien libere a algún personaje masculino. Por esto, decidimos que sería Ariadna quien liberase a Perseo de Ceto.
- **Escenario 7:** en este escenario, los 3 héroes principales de la historia (Perseo, Teseo y Heracles) tienen la misma misión, que es salvar a Andrómeda. Para ello, haremos que todos los héroes tengan el favor de los mismos dioses, y que todos caigan presos en el laberinto de Minos. De esta forma, podremos como van consiguiendo escapar y conseguir los diversos objetos y capacidades necesarias para poder salvar a Andrómeda.
- **Escenario 8:** en este escenario se nos pide crear nuestra propia historia, la cual ya ha sido explicada y detallada en su respectivo apartado de este documento.