

Superglús – Tutorial

La Torre

Autor Original: Uto

Conversion a Superglus: Manowar

Correcciones: Uto



CONSTRUYE TU PROPIA AVENTURA (I)

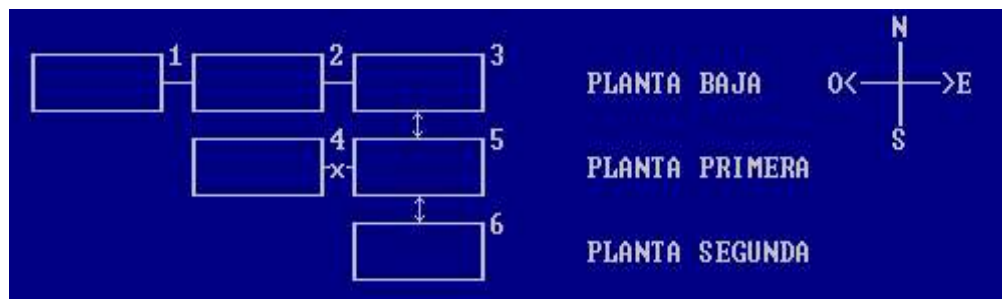
Esta serie de artículos esta dedicada a todos aquellos que nunca habéis programado vuestra propia aventura, y que pensáis que crear una incluso con un parser ha de ser difícil, por ello comienzo estos artículos en los que enseñaré, paso a paso, el modo de crear una aventura con SuperGlus.

Lo primero que lógicamente tendremos que hacer es tener un sencillo guión, sencillo porque lógicamente no voy a crear aquí una aventura de cientos de localidades. Por ello voy a realizar una mini-aventura en la que el único objetivo es salir de una torre en la que nos han encerrado.

Espero que esto os pueda servir como punto de inicio para crear vuestras propias aventuras, dejando claro que nada de lo que yo escriba aquí es dogma de fe y por tanto si no os gusta lo podéis cambiar en vuestras creaciones.

POR DONDE EMPEZAR

Pues quizá lo más normal es comenzar detallando las localidades, es decir, tener un pequeño mapa. Esto no es imprescindible pero sí lo más usual.



Nota: las líneas con flechas quieren decir subir/bajar y la que tiene una equis (x) no existe en un principio.

Ahora, y antes de comenzar a programar nada deberemos de crear la aventura en SuperGlus, para ello pulsaremos en el menu situado en la parte superior de la pantalla, elegimos la opción **Archivo->Nuevo** e introducimos entonces el nombre de la aventura que queremos crear y pulsamos en guardar. Con esto ya tendremos creados el ficheros correspondiente a nuestra aventura. Ahora es momento de comenzar a introducir la información de las localidades.

Nota: los usuarios que no usen el editor de Superglus deberán copiar el fichero START.TXP en otro fichero para empezar a editarlo, y usar su editor preferido.

Nuestro siguiente cometido es crear las descripciones, aunque si aún no las tenemos muy claras bastará con que pongamos un texto identificativo y las conexiones, de modo que al menos podamos movernos por el mapa. Como podéis ver todas las localidades tienen un número a su lado, que se asocia a ellas, esto nos será útil para referenciarlas en nuestro programa y también para describir las conexiones.

Para poner las descripciones en SuperGlus pulsamos sobre la pestaña de la parte superior identificada como **Localidades** y substituir los textos que aparecen por los siguientes:

/0
 /1
 Estás junto a la puerta principal. A su lado puedes una mesa de guardia y en la pared norte hay una chimenea.

/2
 Varios maltrechos catres se amontonan en esta habitación.

/3
 El viento ulula a través de la empinada escalera de caracol, una vieja armadura parece vigilar la escalera.

/4
 Una silenciosa estancia débilmente alumbrada por los rayos de luna que se filtran a través de un pequeño ventanuco. El suelo está lleno de paja, colgando de unos grilletes en la pared observas un esqueleto humano.

/5
 Los desgastados peldaños de piedra resbalan en ocasiones. A mitad de la escalera una antorcha en la pared impide que la oscuridad sea completa.

/6
 Una gran cama preside la estancia, los gruesos barrotes no permiten la salida por la ventana, aunque de todos modos estaría demasiado alta.

Ahora nos dirigimos a la pestaña **Conexiones** y sustituimos el texto existente por el que sigue a continuación:

/0
 /1
 ESTE 2
 /2
 OESTE 1
 ESTE 3
 /3
 SUBIR 5
 OESTE 2
 /4
 ESTE 5
 /5
 BAJAR 3
 SUBIR 6
 /6
 BAJAR 5

Veamos que es todo esto: la primera localidad (0) se deja en blanco dado que es una localidad especial que no debemos usar como otras, pero como el parser requiere que los números de localidades sean consecutivos y empiecen por 0 no podemos saltarnosla por las buenas. Supongo que no os será demasiado difícil ver como se describen las conexiones, baste decir que no hay más que poner cada una de las salidas de la localidad seguida de la localidad a la que se llega por ellas. Como podéis ver no he puesto la salida de la equis, pues en un principio no existe.

Bien, hecho esto si ejecutamos el test del juego (F5), podremos ya movernos por la torre, excepto por la localidad 4.

LA PUERTA SECRETA

Bien, ahora quizá debería empezar a hablar de los objetos, pero entonces no veríais nada de programación en esta entrega, por lo que lo que voy a hacer es las órdenes encargadas de permitirnos el acceso a la localidad 4. Para ello debéis saber

que (como muchos habréis ya supuesto), la antorcha que hay en mitad de la escalera sirve para abrir una puerta secreta, que nos da acceso a la localidad 4.

En primer lugar bastará con que empujemos la antorcha para que la puerta se abra, y tirar de ella para cerrarla, por ello es que hemos de añadir al vocabulario los verbos EMPUJAR Y TIRAR, para lo cual pulsaremos sobre la pestaña

Vocabulario y añadiremos lo siguiente:

EMPUJ	50	verb
EMPUJAR	50	verb
EMPUJA	50	verb
TIRAR	51	verb
TIRA	51	verb

Uso los números de verbo 50 y 51 porque están libres y me apetece, pero bien podría haber usado cualquier otro (157 y 215 por ejemplo). Si pongo tanto "tirar" como "tira" es para hacerlos sinónimos, de la misma forma "empuj", "empujar" y "empuja" serian tambien sinonimos entre si.

Bien, vamos allá: abramos la tabla de respuestas (Alt+R) y escribamos lo siguiente:

EMPUJAR	ANTORCHA	AT 5 GETEXIT OESTE 100 EQ 100 255 SETEXIT OESTE 4 MESSAGE 60 DONE
EMPUJAR	ANTORCHA	AT 5 GETEXIT OESTE 100 NOTZERO 100 MESSAGE 61 DONE

Añadamos los siguientes mensajes (ALT+M):

```
/60
Al empujar la antorcha una porción de pared se abre al oeste dando
acceso
a una estancia.
/61
La antorcha no cede más.
```

Explicaciones:

- Realizamos la entrada EMPUJAR ANTORCHA, por lo que las siguientes instrucciones sólo se ejecutarán si el jugador teclea eso.
- Luego comprobamos que estamos en la localidad adecuada (AT 5), si no es así tampoco se ejecutarán las siguientes ordenes.
- La siguiente instrucción (GETEXIT) nos devuelve el número de localidad al que nos lleva determinada salida de la localidad actual, o 255 si esa salida no existe, por tanto lo usaremos para saber si la entrada secreta está abierta. El valor será almacenado en el flag 100 (si no sabéis que son los

flags mirad la explicación del manual de SuperGlus, es demasiado extenso para explicarlo aquí).

- EQ 100 255: De nuevo una condición, que de no cumplirse impedirá la ejecución de las ordenes siguientes. Esta se cumple si el flag 100 contiene un valor de 255, lo cual será así si no hay salida al oeste (la puerta cerrada), pues GETEXIT habrá puesto un 255 en el flag 100.
- Ahora ya hemos comprobado que todo está en regla, por lo que no tenemos más que informar al jugador y abrir la puerta, para lo cual primero se usa SETEXIT, que crea una salida al OESTE en la localidad actual y que tiene por destino la localidad 4, y luego muestra el mensaje adecuado.
- La orden DONE final indica que ya se ha contestado al jugador, por lo que no sigue buscando más entradas que coincidan con lo que el jugador tecleo (EMPUJAR ANTORCHA), así que no llega a la siguiente entrada:
- La segunda entrada está pensada para que responda al jugador si este trata de empujar más aún la antorcha, su estructura es muy similar, excepto en que comprueba que el flag 100 *no* es 255 (es decir, que hay una salida al oeste) y pone el mensaje en ese caso.

Nota: en realidad en SETEXIT y GETEXIT deberían poner el número del verbo de dirección en lugar del propio verbo. Es decir, la sentencia GETEXIT OESTE 100 no funciona, y habría que poner GETEXIT x 100, donde x es el numero de 'oeste' en el vocabulario.

Cuidado: GETEXIT y SETEXIT son conductos plugin, lo cual quiere decir que no vienen de serie con Superglus. Debemos obtener los ficheros de la exitlib, que generalmente vendrán con el paquete de Superglus como fichero zip, tar, gzip, etc., y extraer de ellos los ficheros getexit.plu y setexit.plu y colocarlos en la carpeta *pgl* en el directorio de Superglús. Ya que estáis extraed tambien el fichero exits.plu, que hará falta en el capítulo II.

Bueno, pues por esta entrega nada más, en la siguiente trataremos acerca de los objetos y veremos como cerrar esta puerta secreta (cosa que ya podéis ir pensando por vuestra cuenta, no es muy diferente a la manera de abrirla).

CONSTRUYE TU PROPIA AVENTURA (II)

Seguimos con esta serie de artículos dedicados a enseñar a los que nunca habéis programado una aventura a hacerlo. En esta entrega hablaremos de los siempre necesarios objetos y de como cerrar aquella dichosa puerta de la que hablabamos en el número anterior.

LA PUERTA

En el anterior capítulo vimos como abrirla, y os dejaba a vosotros intentar el proceso de cerrarla. Pues bien, esta es mi solución, lo cual no quiere decir que no pueda haber otras tan buenas o mejores que esta:

Bien, en la tabla de respuestas y escribamos lo siguiente:

TIRAR	ANTORCHA	AT 5 GETEXIT 5 100 NOTEQ 100 255 CLEAREXIT 5 MESSAGE 62 DONE
TIRAR	ANTORCHA	AT 5 GETEXIT 5 100 EQ 100 255 MESSAGE 61 DONE

Añadamos el siguiente mensaje:

/62

Al tirar de la antorcha la porción de pared que se abrió vuelve a cerrarse.

Explicaciones:

- Añadimos la entrada TIRAR ANTORCHA, por lo que las siguientes instrucciones sólo se ejecutarán si el jugador teclea eso.
- Luego comprobamos que estamos en la localidad adecuada (AT 5), si no es así tampoco se ejecutarán las siguientes ordenes.
- Obtenemos con GETEXIT la localidad a la que iríamos al oeste.
- Comprobamos que no es 255 (en cuyo caso la puerta ya está cerrada).
- Ahora ya hemos comprobado que todo está en regla, por lo que no tenemos más que informar al jugador y cerrarla, para ello lo primero es usar CLEAREXIT, para destruir la salida.
- La orden DONE final indica que ya se ha contestado al jugador, por lo que no sigue buscando más entradas que coincidan con lo que el jugador tecleo (TIRAR ANTORCHA), así que no llega a la siguiente entrada:
- La segunda entrada está pensada para que responda al jugador si este trata de tirar aún más la antorcha, su estructura es muy similar, excepto en que comprueba que el flag 100 *si* es 255 (es decir, que no hay una salida al oeste) y pone el mensaje en ese caso (como podéis ver he aprovechado el

mismo mensaje, pues vale, pero esto no me supone un ahorro de memoria, aunque sí de espacio en disco.

Del mismo modo que en el ejemplo anterior no vale poner 'oeste', sino que hay que poner su número en el vocabulario; lo he puesto para mas claridad.

Código sobran te

En el código de abrir la puerta y en el de cerrarla hay en cada uno dos líneas de código innecesarias, el próximo número hablaré de ellas, pero vosotros podéis ir buscándolas por vuestra cuenta.

OBJETOS

Los objetos son fundamentales en casi cualquier aventura conversacional, y nos sirven como herramientas para hacer algo, como piezas para trueque, como armas e incluso algunos absolutamente para nada (los llamados *red herrings*). Vamos a ver los objetos que vamos a tener en nuestra aventurilla:

- una barra de hierro
- un trozo de carbón
- unas raidas sábanas
- un pequeño cuchillo

Por tanto su descripción en la pestaña **Objetos (ALT+O)** quedaría de la siguiente forma:

```

/0
una barra de hierro
/1
un trozo de carbón
/2
unas raidas sábanas
/3
un pequeño cuchillo
/4
unas fundas de catre

```

Y en la pestaña **Objetos [2] (ALT+b)** quedaria la siguiente forma:

[illegible]

Bien, ahora viene la explicación:

Para cada objeto ponemos primero su número y seguidamente su nombre todo ello en la tabla **Objetos**, despues, en la tabla **Objetos [2]** volvemos a teclear primeramente su numero, despues la localidad inicial en la cual se encontrara el objeto, seguimos colocando ahora el peso del objeto indicados en unidades de peso. Tras esto viene el nombre que identifica al objeto y que sera el que el jugador habra de teclear para referirse a el. Tambien podemos añadir un adjetivo como en el caso de la barra u omitirlo utilizando el caracter especial _. Una vez descrito todo eso solo nos quedan las 64 propiedades de objeto que estan repartidas en dos bloques de 32 para una mayor legibilidad.

Ademas de las localidades definidas por el programador hay tres localidades especiales que son las siguientes:

```
252: El objeto no está en ninguna parte (de momento no es visible).
253: El objeto lo llevamos puesto (es una prenda).
254: El objeto lo llevamos.
```

Como podéis ver todos los objetos de nuestra aventura son inaccesibles de primeras.

No vamos a entrar en este tutorial a ver las propiedades de los objetos por lo que de momento los dejaremos a 0 (64 ceros). Mas información en la Guía Tecnica de Superglús.

Por último, para que nuestros objetos funcionen bien, debemos poner las palabras que hemos dicho que los referirán en el vocabulario, como nombres:

BARRA	51	noun
CARBON	52	noun
SABANAS	53	noun
CUCHILLO	54	noun
FUNDA	55	noun
CHIMENEA	56	noun
HIERRO	10	adjective

No es necesario que los números de nombre sean exactamente los mismos que los números de objeto, pero sí es altamente recomendable, como veremos en breve.

Para finalizar este capítulo vamos a hacer aparecer el trozo de carbón, que está en la chimenea. Comencemos poniendo en la tabla de respuestas:

EXAMINAR	CHIMENEA	AT 1 MES 63 ISAT 1 252 MESSAGE 64 CREATE 1 DONE
EXAMINAR	CHIMENEA	AT 1 NEWLINE DONE

Y añadamos los nuevos mensajes:

```
/63
La chimenea está muy sucia y no hay fuego en ella.-
/64
Hay un gran trozo de carbón en ella.
```

Bien, ¿cómo funciona esto?:

Primero al examinar la chimenea comprobamos que estamos en la localidad correcta (AT 1), en cuyo caso ponemos la descripción general de la misma pero usando MES en lugar de MESSAGE, de modo que el cursor no pase a la siguiente línea.

Luego comprobamos que el objeto 1 está en la localidad 252, siguiendo sólo en ese caso. .

Bien, en caso de que el carbón estuviera inaccesible lo que haremos será poner el otro mensaje (que aparecerá justo tras el anterior, no en la siguiente línea), crear el objeto (CREATE 1), poniendolo de este modo en la localidad actual y dar por concluido (DONE).

Si el objeto no estaba en la localidad 252 nos saldremos de esa entrada para caer irremisiblemente en la segunda, en la que pasaremos el cursor a la línea siguiente (NEWLINE) y finalizaremos (DONE). Esta entrada es fundamental, pues de no existir nos encontraríamos con que si el objeto no está en la 252 nos pondría: "La chimenea está muy sucia y no hay fuego en ella. No puedes hacer eso. ".

Bien, pues aquí llega a su fin esta segunda entrega. En la próxima veremos como interactuar con otros objetos y desvelaré el secreto de las instrucciones de más. Hasta pronto.

CONSTRUYE TU PROPIA AVENTURA (III)

Pues bien, veíamos en el anterior capítulo que la forma de cerrar la puerta secreta, y decíamos que tanto en él como en el otro había dos instrucciones sobrantes. Os voy a decir cuales son las de la parte de cerrar la puerta y espero que deduzcais cuales son las de abrirla. Teníamos el siguiente código:

TIRAR	ANTORCHA	AT 5 GETEXIT 5 100 NOTEQ 100 255 CLEAREXIT 5 MESSAGE 62 DONE	
TIRAR	ANTORCHA	AT 5 GETEXIT 5 100 EQ 100 255 MESSAGE 61 DONE	---> Sobra. ---> Sobra.

Como véis he marcado las dos instrucciones sobrantes. ¿Y por qué sobran? Pues bien, dado que el código se va ejecutando de arriba a abajo en el caso de que estemos en la localidad 5 y el jugador teclee "TIRAR DE ANTORCHA" entraremos en la primera entrada, en la cual se guarda en el flag 100 la conexión al oeste y se comprueba que no es 255, en cuyo caso la matamos y se muestra el mensaje, pero lo fundamental es que la entrada acaba con DONE, por lo que el resto del proceso (o tabla de respuestas en este caso) es ignorado.

Por tanto si el programa está ejecutando la segunda entrada, y tras el AT 5, podemos estar seguros de que la conexión de la localidad esta a 255 (o sea, la puerta cerrada), porque de otro modo nunca habríamos llegado hasta allí. Es así que guardar de nuevo el valor de la conexión en el flag 100 y comprobar su valor carece de sentido y por eso sobran ambas ordenes. Lo cierto es que no es que esté mal, es que es redundante.

SIGAMOS CON NUESTRA FUGA

Bien, vimos en el capítulo anterior la forma de hacer aparecer el carbon, veamos ahora dónde está el cuchillo. El cuchillo resultará estar entre los huesos del esqueleto de la habitación oculta, para encontrarlos por supuesto que debemos examinarlo. Como veremos estas ordenes serán muy similares a las de encontrar el carbón.

Pongamos las siguientes ordenes en la tabla de respuestas:

EXAMINAR	ESQUELETO	AT 4 MES 65 ISAT 3 252 MESSAGE 66 DONE
EXAMINAR	ESQUELETO	AT 4 NEWLINE DONE

Los mensajes a añadir serán:

```
/65
Los huesos amarillentos, las cuencas vacías.
/66
Junto a él observas un viejo cuchillo.
```

Por último en el vocabulario añadiremos:

ESQUELETO	57	noun
-----------	----	------

de modo que la palabra esqueleto quede añadida al vocabulario.

VOLVIENDO A LA PUERTA

Vamos a continuar viendo la utilidad de un proceso que no es la tabla de respuestas, en concreto el proceso 1. Este proceso se ejecuta tras describir las localidades, por lo cual es ideal para describir cosas de las mismas que no sean

fijas, como por ejemplo el estado de la puerta que se abre con la antorcha.
Añadamos esto en el proceso 1 (Alt-1):

```
—      —      AT 5
          GETEXIT 5 100
          NOTEQ 100 255
          MESSAGE 67
          NOTDONE
```

Y añadamos el mensaje:

```
/67
Cierta porción de la pared se ha movido dando acceso a otra sala.
```

Bien, ¿qué es lo que hacemos aquí?. Pues de nuevo usamos la orden GETEXIT para saber a que localidad nos lleva la dirección OESTE en la localidad 5. En caso de que ésta no sea 255 quiere decir que la puerta secreta está abierta, en cuyo caso mostramos el mensaje 67, que aparecerá justo tras la descripción de la localidad tal que así:

"Los desgastados peldaños de piedra resbalan en ocasiones. A mitad de la escalera una antorcha en la pared impide que la oscuridad sea completa. Cierta porción de la pared se ha movido dando acceso a otra sala."

En caso de que la puerta esté cerrada, el mensaje 67 no aparece.

SALIDAS DE LOCALIDAD

Una de las cosas que tiene gran facilidad de hacerse en Superglue es una orden que nos de las salidas de la localidad actual, para ello nos valdremos del conducto EXITS. EXITS es un conducto plugin incluido en la exitlib.

Creemos los siguientes mensajes:

```
/68
Puedes ir-
/69
No veo ninguna salida.
/70
/71
/72
al norte
/73
al sur
/74
al este
/75
al oeste
/76
al noreeste
/77
al noroeste
/78
al sureste
/79
```

al suroeste
/80
arriba
/81
abajo
/82
dentro
/83
fuera

Bien, ¿por que hacemos esto?. Pues porque para la instruccion exits hay que preparar cierta cantidad de mensajes en este orden:

1. Uno para el inicio de la lista de salidas "Puedes ir ".
2. Otro para las localidades que no tengan salidas.
3. Un mensaje para cada dirección, en el orden en que estas salen en el vocabulario, es decir los verbos a los que corresponden.

Si nos fijamos en el vocabulario veremos que el primer verbo es el verbo numero 2, "norte", y es por eso por lo que el mensaje 70 y 71 están en blanco, porque son correspondientes al verbo 0 y 1 que no están definidos en el vocabulario, luego viene el 72 que corresponde al verbo 2 y así los demás. Aunque en nuestra aventura no necesitaríamos ninguno de los mensajes a partir del 82 (si nos fijamos en el mapa ni siquiera los de norte y sur), los hemos incluido porque más tarde les daremos una utilidad, pero eso será más adelante.

Bueno, ¿cómo hacemos para que se listen las salidas?, pues muy fácil, ponemos en la tabla de respuestas:

SALIDAS	—	EXITS @38 68
		DONE

Y en el vocabulario:

SALIDAS	55	verb
---------	----	------

Eso servirá para que se listen las conexiones de la localidad actual teniendo en cuenta que el mensaje 68 es a partir del cual tiene preparado todos los mensajes.

LOS FLAGS

En el ejemplo anterior hemos tenido la primera toma de contacto con un flag y con la indirección. Los flags nos servirán para controlar cosas en nuestra aventura, y algunos como el flag número uno contienen un valor dado por el parser.

En este caso tenemos el flag 38, que contiene siempre el número de localidad en que estamos. A su vez el signo @ es el signo de indirección, que hace que lo que le siga sea tomado como un número de flag y substituye el conjunto @número por el contenido del flag que tiene ese número. que lío no? Para que se vea más claro ahí va esto:

Supongamos que estamos en la localidad 2, por tanto el flag 38 contiene un dos, y entonces la orden EXITS @38 68 la traducirá el parser por:

```
EXITS 2 68
```

mostrando de este modo las salidas de la localidad 2

Supongamos que nos vamos a la localidad 4, por lo que el flag 1 pasa a valer 4, pues entonces la misma orden (EXITS @38 68) sería traducida por

EXITS 4 68, puesto que ahora el flag 38 vale 4, y listaría las salidas de la localidad 4.

CONSTRUYE TU PROPIA AVENTURA (y IV)

Bien, nuestra mini-aventura está casi terminada, al menos en su esqueleto principal, aunque aún nos falta saber que hemos de hacer para escapar. Los pasos que nos quedan serán los siguientes, que pasaremos a programar a continuación:

- Usar el cuchillo para romper las ataduras que sujetan la funda a la cama.
- Coger funda.
- Examinar los barrotes de la ventana (uno estará flojo).
- Coger/Tirar del barrote suelto.
- Atar la funda a otro barrote.
- Salir.

Como veréis he dejado de lado las sábanas, mas que nada por simplificar el código y que resulte más fácil de entender, así que las sabanas no aparecerán nunca.

Bien, vayamos por partes:

Para que el jugador sepa que hay una funda deberá examinar la cama, por lo que añadiremos las siguientes entradas en la tabla de respuestas:

EXAMINAR	CAMA	AT 6 ISAT 4 252 MESSAGE 84 DONE
EXAMINAR	CAMA	AT 6 ISNOTAT 4 252 MESSAGE 85 DONE

Y añadimos los siguientes mensajes:

```
/84
Una funda de tela cubre la cama.
/85
```

Solo restos de paja cubren la cama.

¿Y que hacemos con esto?

Pues si la funda esta en la localidad 252 (no creada, tal como es al principio) es que está puesta, y si está en cualquier otro sitio es que está quitada, por lo que ponemos uno u otro mensaje según donde esté. Si sois perspicaces os habréis dado cuenta de que de nuevo hay código sobrante: Sí, el ISNOTAT 4 252 sobra en la segunda entrada, pues no llegará a ella si, estando en la localidad 6, la funda está en la localidad 252, pues se comprueba en la entrada anterior.

Veamos ahora como impediremos que el jugador coja la funda cuando aún está atada.

Bien, primeramente, y ya sin otra salida posible, tendremos que usar un flag. Mi costumbre es empezar por el 255, que es el último, pero podríamos usar otro cualquiera. La utilidad del flag será la siguiente:

Si el flag vale 0 (que es el valor por defecto), querrá decir que las ataduras están intactas. Si vale 1 querrá decir que las ligaduras han sido cortadas.

Añadamos las siguientes entradas en la tabla de respuestas, colocando el vocabulario que fuera necesario (espero que ya sepáis hacerlo) en la tabla de vocabulario:

```
COGER          FUNDA          AT 6
                                ISAT 4 252      ; La funda está en la cama.
                                ZERO 255      ; Está atada!!!
                                MESSAGE 86
                                DONE
```

El DONE es *muy* importante, puesto que evita que se siga ejecutando la tabla de respuestas, impidiendo así que se llegue a la entrada "COGER _", que no nos interesa. Así hemos impedido que el jugador coja la colcha, a pesar de que con un EXAMINAR la ve. Nota: Estas entradas han de estar antes de la entrada que pone "**COGER _**" porque si no se ejecutaría esa en lugar de estas.

```
COGER          FUNDA          AT 6
                                ISAT 4 252
                                NOTZERO 255
                                CREATE 4      ; Creamos la funda
                                GET 4         ; La cogemos, a partir de
ahora la
                                ; funda es un objeto
                                DONE         ; normal.
```

¡Ah!. Añadamos los siguientes mensajes:

```
/86
La funda está atada a la cama por fuertes ligaduras.
/87
Cortas las ligaduras con el cuchillo.
/88
No puedes desatarlas.
```

Y ahora veamos como haremos para cortar las ligaduras. En la tabla de respuestas pongamos:

CORTAR	LIGADURAS	AT 6	
funda está en la cama		ISAT 4 252	; Comprobamos que
ligaduras están intactas		ZERO 255	; Comprobamos que las
llevamos el cuchillo.		CARRIED 3	; Comprobamos que
		MESSAGE 87	
del flag, ahora las ligaduras están		LET 255 1	; Cambiamos el valor
		DONE	; rotas.

Algunos autores de aventuras obligan al jugador a teclear la orden completa "CORTAR LIGADURAS CON CUCHILLO", pero yo considero que eso es restar jugabilidad, puesto que el jugador, si lleva el cuchillo, se da por supuesto que lo usará en lugar de usar las uñas o los dientes no?. En todo caso, si lo preferís, bastará con añadir las condiciones:

```
PREP CON
NOUN2 CUCHILLO
```

Añadiendo en cada caso las palabras del vocabulario necesarias.

Añadamos la entrada para el caso de que el jugador quisiera desatar la funda:

DESATAR	FUNDA	AT 6	
		MESSAGE 88	
		DONE	

Bien, a estas alturas podemos coger ya la funda, así que dejaremos el resto para la próxima entrega.

Nota: Esta serie de artículos se realizaron para el parser NMP y han sido adaptados a Superglús. Dado que la serie original quedó incompleta este tutorial tampoco finaliza. No obstante esperamos que lo visto hasta aquí os sirva de ayuda para entender Superglús y que vosotros mismos podáis finalizar esta aventura.