



P-NAV, sistema de navegación para un robot móvil.

Daniel Aparicio García.

# MANUAL DEL USUARIO

# **1.- Introducción**

En este manual del usuario se describen todos los pasos que se deben seguir para la construcción de una carta de navegación para un robot regido por el sistema que en este proyecto se ha construido.

El sistema que en este proyecto se ha diseñado funciona mediante la lectura de mapas de habitaciones a partir de los cuales el software calcula en tiempo real la ruta que en cada momento el robot debe seguir. De esta manera es posible realizar una navegación sobre una zona definida en el sistema.

Por ello el usuario del sistema de navegación del que estamos tratando debe conocer con profundidad los pasos que se han de seguir para el correcto diseño de un mapa válido para que el robot que funcione con este sistema pueda cumplir la misión de seguir el camino apropiado en cada uno de sus viajes.

## **2.- Pasos a seguir para definir una zona de navegación**

Para construir un mapa completo de navegación debemos seguir varios pasos que en este apartado sólo se nombran pero a continuación se explican detalladamente.

Antes de especificar los pasos a seguir cabe destacar que es muy importante darle un valor a una variable de entorno del sistema cuyo nombre es “NAVEGACION”. La aplicación de instalación se encarga de ello en caso de tratarse el sistema operativo de Windows 95/98 (Para Windows NT/2000 ver manual de instalación). Ésta habrá ubicado el directorio de trabajo del sistema P-NAV en el directorio “P-NAV” de Saphira. Esta variable especifica el directorio donde se guardará la carta de navegación que debemos crear tal y como se indica en el apartado 5 de este documento y los mapas de obstáculos y distancias.

También podemos elegir otro directorio de trabajo. Para hacer esto en Windows 95/98 editaremos el archivo “C:\AUTOEXEC.BAT” y modificaremos la línea donde se especifica el valor de esta variable escribiendo:

“SET NAVEGACION=<directorio\_navegacion>”.

En Windows NT hay que acceder a Inicio → Configuración → Panel de Control → Sistema → Entorno y ahí modificar el valor de la variable “NAVEGACION”.

De esta manera las aplicaciones que se utilizarán para crear los mapas de distancias y de obstáculos lo harán en el directorio elegido, así como que el sistema buscará ahí dichos archivos. También se debe crear la carta de navegación en dicho directorio.

También es necesario darle valor a la variable de entorno “SAPHIRA”. De cualquiera de las maneras, si ésta no estuviese definida la aplicación de instalación no habría instalado P-NAV. Esta variable debe tener el valor del directorio donde se ha instalado el software Saphira, y usualmente suele ser “C:\SAPHIRA\VER62”. Para hacer esto hay que añadir de la misma manera una línea en el archivo AUTOEXEC.BAT especificando el valor de esta variable.

Una vez suministrados los valores de estas variables se deben seguir los pasos siguientes:

1. Definir un mapa de cada habitación, es decir, suministrar al sistema las coordenadas de todos los objetos fijos en cada habitación incluyendo puertas y paredes.
2. Creación a partir de los mapas definidos unos mapas binarios y unos mapas de distancias a los obstáculos que existen en cada habitación.
3. Creación de una carta de navegación que especifique cómo se comunican unas habitaciones con otras para poder navegar por todas ellas.

4. Y opcionalmente un archivo de órdenes que especifique todos los pasos (rutas) que el robot móvil debe seguir.

En los siguientes apartados se procede a explicar el modo de realizar cada uno de estos pasos por el orden fijado.

### 3.- Definiendo un mapa

A la hora de definir un mapa hay que hacerlo tal y como el autor del sistema Saphira explican en la documentación de dicho sistema. En el presente apartado se resume de una manera clara el modo de hacerlo.

La definición de los mapas en Saphira se realiza editando un tipo de archivo con extensión “.MAP”. A continuación se explica el formato de un archivo de mapas.

Un archivo “.MAP” tiene opcionalmente comentarios que irán precedidos de un “;”, y líneas especificando objetos en el mapa. Las coordenadas de dichos objetos son siempre globales, aunque se debe tener en cuenta que, como se explica más adelante, es posible cambiar de habitación y por tanto de mapa junto con las nuevas coordenadas, aunque esto es algo que no viene construido en el sistema estándar de Saphira, sino que forma parte del presente proyecto.

En estos mapas podemos definir pasillos, puertas, paredes y entradas a pasillos (que sería en la práctica similar a definir puertas).

La manera de definir cada uno es la siguiente:

- **PASILLOS:** Los definimos con la palabra “CORRIDOR” y, opcionalmente, con un identificador que debe ser un número entero positivo entre paréntesis. Lo siguiente son las coordenadas X e Y en las que se encuentra el centro del pasillo. La tercera coordenada es Th, la orientación en grados (su valor debe ser 0, 90, 180 ó 270) del pasillo y la cuarta y quinta coordenadas son su longitud y anchura (todos los valores se toman en milímetros). **NOTA:** Todas las coordenadas posicionales y de tamaño van separadas por una coma.
- **PUERTAS:** Las definimos como “DOOR” y con un identificador que coincidirá necesariamente con los identificadores de las puertas definidas en la carta de

navegación. Después se escriben como en el anterior caso las coordenadas del centro en el orden X, Y, Th y finalmente la longitud de la puerta. Es recomendable que el valor Th sea el ángulo (en grados) que el robot debe tomar para traspasar la puerta. Su valor debe ser 0, 90, 180 ó 270.

- ENTRADAS A PASILLOS: Están definidas como “JUNCTION” y su descripción es idéntica a la de las puertas.
- PAREDES: Definidas como “WALL” se especifican de la misma manera que las puertas.

Por ejemplo, el siguiente mapa representa el laboratorio de robótica de la Facultad de Ciencias de la Universidad de Salamanca:

;; Mapa del laboratorio de robótica.

```
WALL 0, 2400, 90, 4800
WALL 7500, 2400, 90, 4800
WALL 3750, 0, 0, 7500
WALL 3750, 4800, 0, 7500
DOOR (1) 900, 0, 90, 1200
WALL 450, 900, 0, 900
WALL 900, 1350, 90, 900
WALL 600, 1800, 0, 1200
WALL 1200, 2850, 90, 2100
WALL 1650, 3900, 0, 3300
WALL 3300, 4050, 90, 300
WALL 5100, 4350, 0, 3600
WALL 6600, 2400, 90, 4800
WALL 6150, 600, 0, 900
WALL 5700, 1650, 90, 3300
WALL 4650, 3300, 0, 2700
WALL 3300, 1650, 90, 3300
WALL 2550, 1200, 0, 1500
WALL 1800, 600, 90, 1200
```

**Figura 3.1:** Mapa del laboratorio de robótica en la Facultad de Ciencias de la Universidad de Salamanca.

NOTA: Véase que las coordenadas en los mapas están escritas con números enteros.

Pueden ser positivos o negativos, tanto Saphira como P-NAV están preparados para ello.

y en Saphira se traduce en lo siguiente:

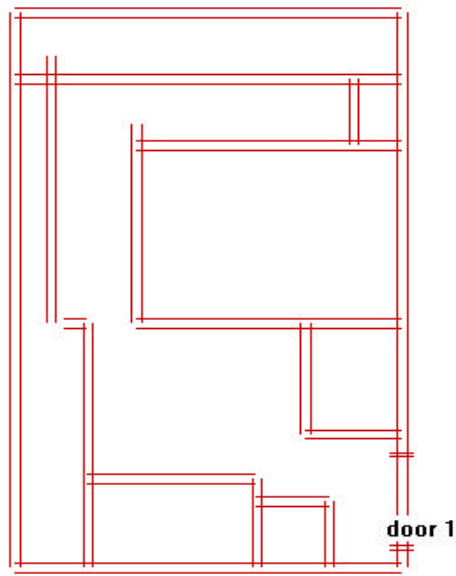


Figura 3.2.

## 4.- Creación de un bitmap y un mapa de distancias a obstáculos

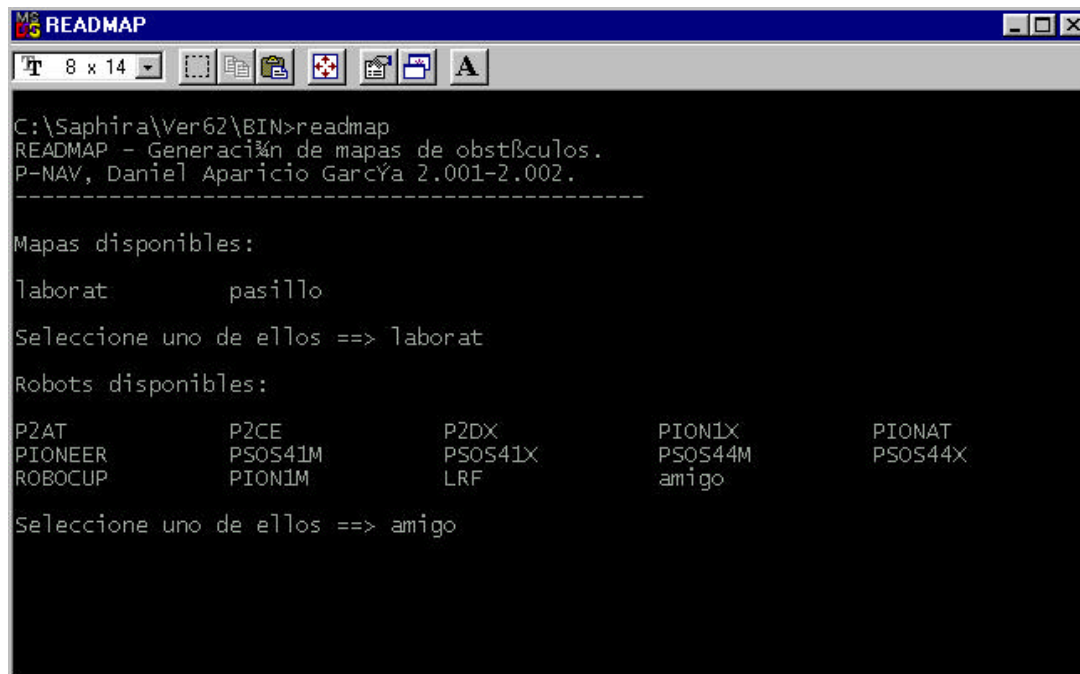
Una vez creado el mapa en el lenguaje del sistema Saphira se debe proceder a crear a partir de él los archivos de mapas que nuestro sistema de navegación utilizará para cumplir su misión.

El archivo “.MAP” sirve para poder visualizar por la interfaz del sistema Saphira una representación del lugar donde el robot se moverá además de permitirnos construir a partir de él dos archivos que contienen un mapa de bits que representa dónde hay obstáculos y dónde hay vía libre y un mapa de distancias a los obstáculos que permite realizar un cálculo mediante potenciales de las fuerzas repulsivas a dichos obstáculos. Su ubicación debe ser el directorio “MAPS” de Saphira.

Para crear los archivos de obstáculos y de distancias a ellos hay dos maneras de hacerlo:

- 1) Sólo hay que ejecutar en una interfaz de comandos (MS-DOS) la aplicación Readmap contenida en el directorio BIN de Saphira pasándole como argumentos el nombre SIN la extensión del archivo “.MAP” que contiene la definición del mapa a procesar y el nombre, también sin extensión, del archivo que contiene las características del robot que se va a utilizar. Tampoco hay que especificar la ubicación de éstos en el disco, las variable de entorno “SAPHIRA” se encarga de ello. Es decir, si el archivo “.MAP” se llama “PASILLO.MAP” y el archivo de características del robot se llama “AMIGO.P” hay que escribir “readmap pasillo amigo” y con ello se crearán tres archivos: “PASILLO.BIT”, “PASILLO.DIS” y “PASILLO.DES”, sirviendo éste último para que el usuario pueda editar mapas con coordenadas negativas. Estos 3 archivos que serán de vital importancia a la hora de realizar un viaje con el robot por la habitación “PASILLO”. En caso de no pasar una línea de comandos a la aplicación ésta le ofrecerá una lista de mapas y robots disponibles en modo texto y el usuario elegirá uno de cada.





```
C:\Saphira\Ver62\BIN>readmap
READMAP - Generaci3n de mapas de obst3culos.
P-NAV, Daniel Aparicio Garc3a 2.001-2.002.
-----
Mapas disponibles:
laborat      pasillo
Seleccione uno de ellos ==> laborat

Robots disponibles:
P2AT          P2CE          P2DX          PION1X        PIONAT
PIONEER       PSOS41M       PSOS41X       PSOS44M       PSOS44X
ROBOCUP       PION1M        LRF           amigo
Seleccione uno de ellos ==> amigo
```

Figura 4.1.

- 2) Otra manera m3s f3cil es acceder a la aplicaci3n Map2bin, ubicada en el directorio BIN de Saphira, con el rat3n pinchando sobre su icono. Esta aplicaci3n hace uso de los ejecutables Readmap.exe y Thinning.exe. Su interfaz gr3fica ofrecer3 una lista de mapas y de robots disponibles para generar los mapas de obst3culos y de distancias a ellos para que el usuario elija una de cada:

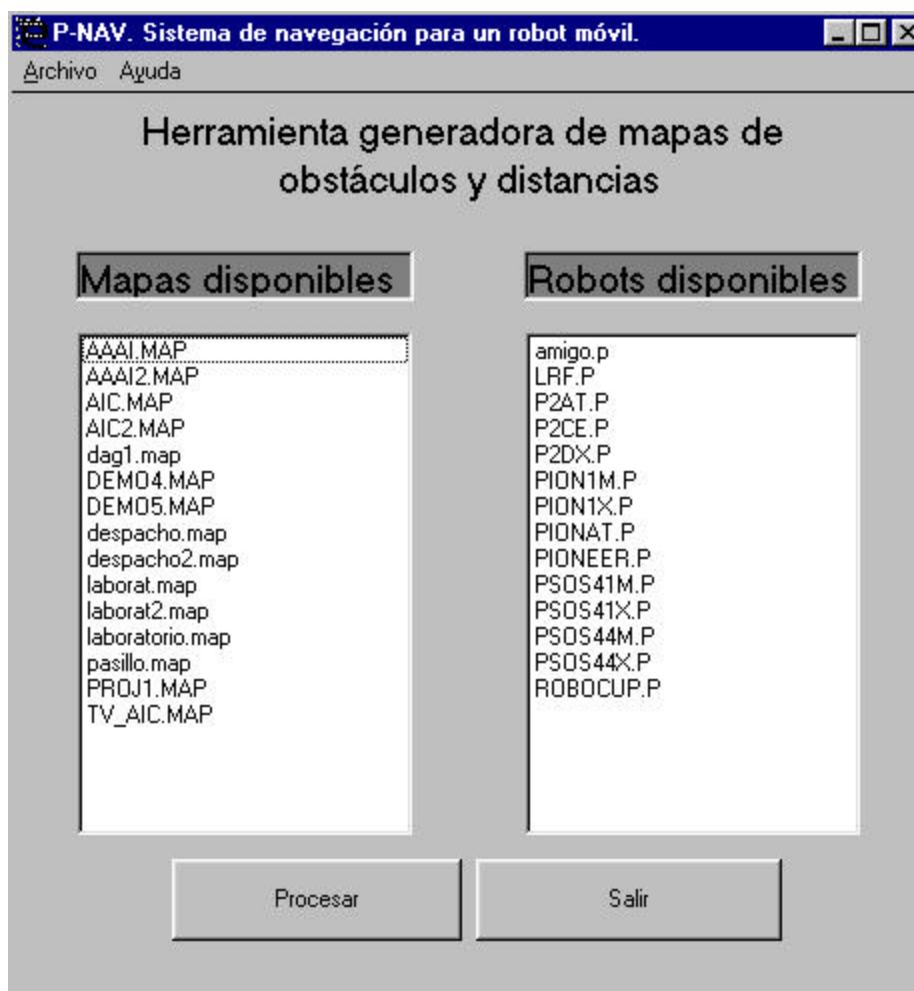


Figura 4.2.

Los archivos “.BIT” y “.DIS” obtenidos a partir de los cálculos son legibles con cualquier lector de archivos de texto y su contenido es una matriz de números que nuestro sistema deberá interpretar cuando realicemos la carga de la habitación en el momento de la conexión.

```
Command▶      *** Top of file ***      Options: h8kMpswTali
```

## 5.- Creación de la carta de navegación

Una vez que es conocido cómo crear mapas y sus correspondientes matrices de obstáculos y distancias es necesario especificar al sistema como se comunican unas habitaciones con otras para poder navegar por distintos mapas.

Para realizar esta tarea es necesario editar un archivo llamado “NAVEGACION.HYP” (HYP=Habitaciones Y Puertas) en el directorio que indica la variable de entorno “NAVEGACION”. Este tipo de archivo ha sido diseñado para este proyecto y posee una estructura muy sencilla. Es un fichero en modo texto y hace una declaración de las habitaciones en las que navegaremos y las puertas que las comunican.

Este paso se puede realizar con la aplicación “Cartapnav.exe”. Esta aplicación facilita al usuario mediante una interfaz gráfica la edición de la carta de navegación:

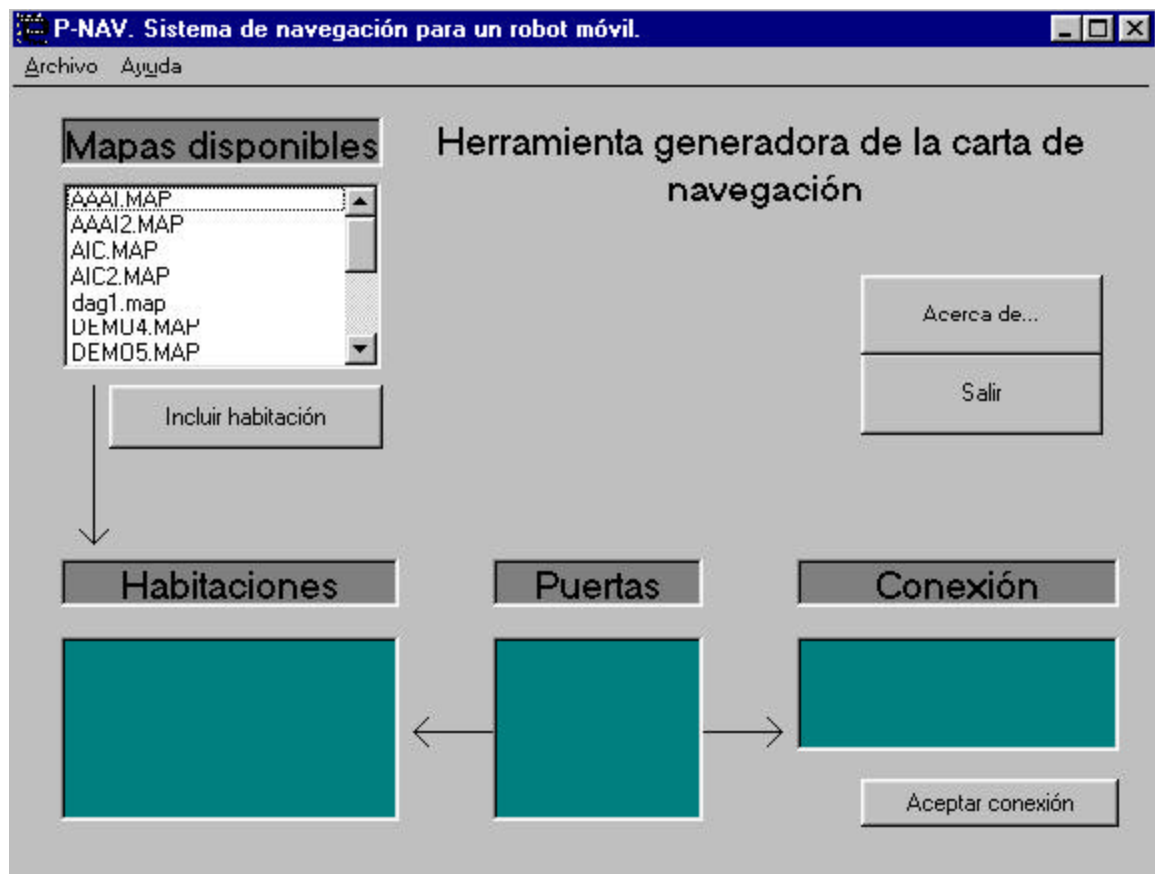


Figura 5.1.

La aplicación “Cartapnav.exe” está ubicada en el directorio “BIN” de Saphira, al igual que el resto de los ejecutables de este sistema de navegación.

Su uso es muy sencillo. Lo primero que hace al ejecutarse es leer la carta de navegación existente (en caso de que la haya) y aparecerán en el cuadro “Habitaciones” la lista de habitaciones disponibles en el sistema. Podemos añadir más habitaciones seleccionando un mapa disponible en la lista de mapas y pulsando el botón “Incluir habitación”. De este modo aparecerá la nueva habitación en el cuadro “Habitaciones”. Es importante conocer que la primera habitación que aparezca en el cuadro “Habitaciones” será en la que se inicie el sistema P-NAV cuando se conecte al robot, por tanto cuando se edite la carta de navegación por primera vez y se incluya en ella la primera habitación debe conocerse que será la habitación inicial del sistema. Después se pueden incluir las habitaciones en el orden que el usuario decida, no afectará en nada a la navegación.

Cuando seleccionemos una habitación del cuadro “Habitaciones” aparecerá en el cuadro “Puertas” las puertas disponibles en esta habitación. El sistema conoce esto mediante la lectura del archivo “.MAP” correspondiente. Por este motivo es necesario asignarle un identificador a las puertas en la edición de mapas.

Entonces lo siguiente es seleccionar la puerta que queremos definir para conectar la habitación con otra, y el sistema ofrecerá una lista de habitaciones que poseen un identificador de puerta idéntico para poder conectarlas.

Una vez hecho esto sólo queda pulsar el botón “Aceptar conexión” y la puerta habrá sido definida. Las puertas tienen un identificador único en el sistema, por lo que no podrá haber más de 2 habitaciones con una puerta que tenga el mismo identificador. La aplicación no permitirá conectar una puerta a más de 2 habitaciones (como es lógico).

Ejemplo:

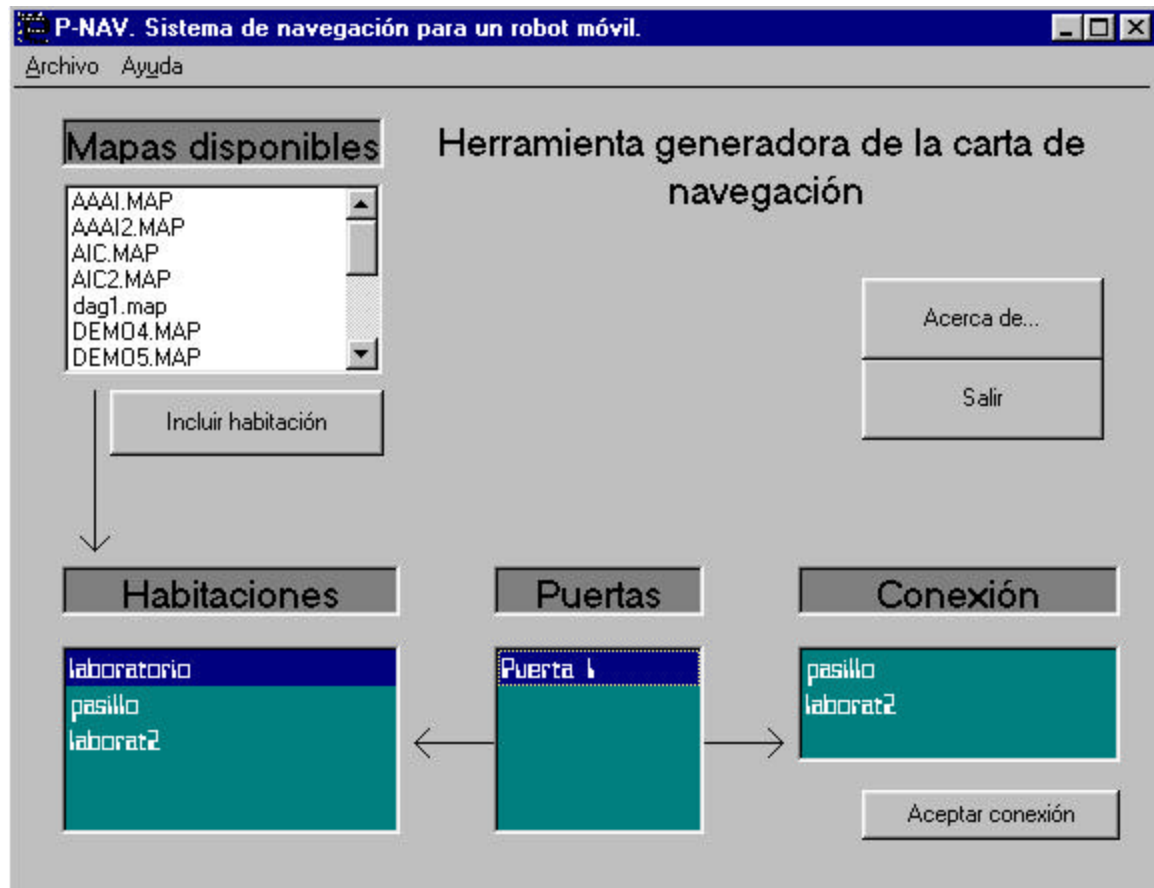


Figura 5.2.

Como se puede ver en la figura 5.2 el sistema nos da a elegir entre "pasillo" o "laborat2" como habitaciones para conectar a "laboratorio" por la puerta 1. Esto se debe a que "laboratorio.map", "pasillo.map" y "laborat2.map" contienen definida una puerta con identificador 1. En cuanto elijamos uno y aceptemos la conexión el sistema no nos dejará volver a definir una conexión mediante la puerta 1:

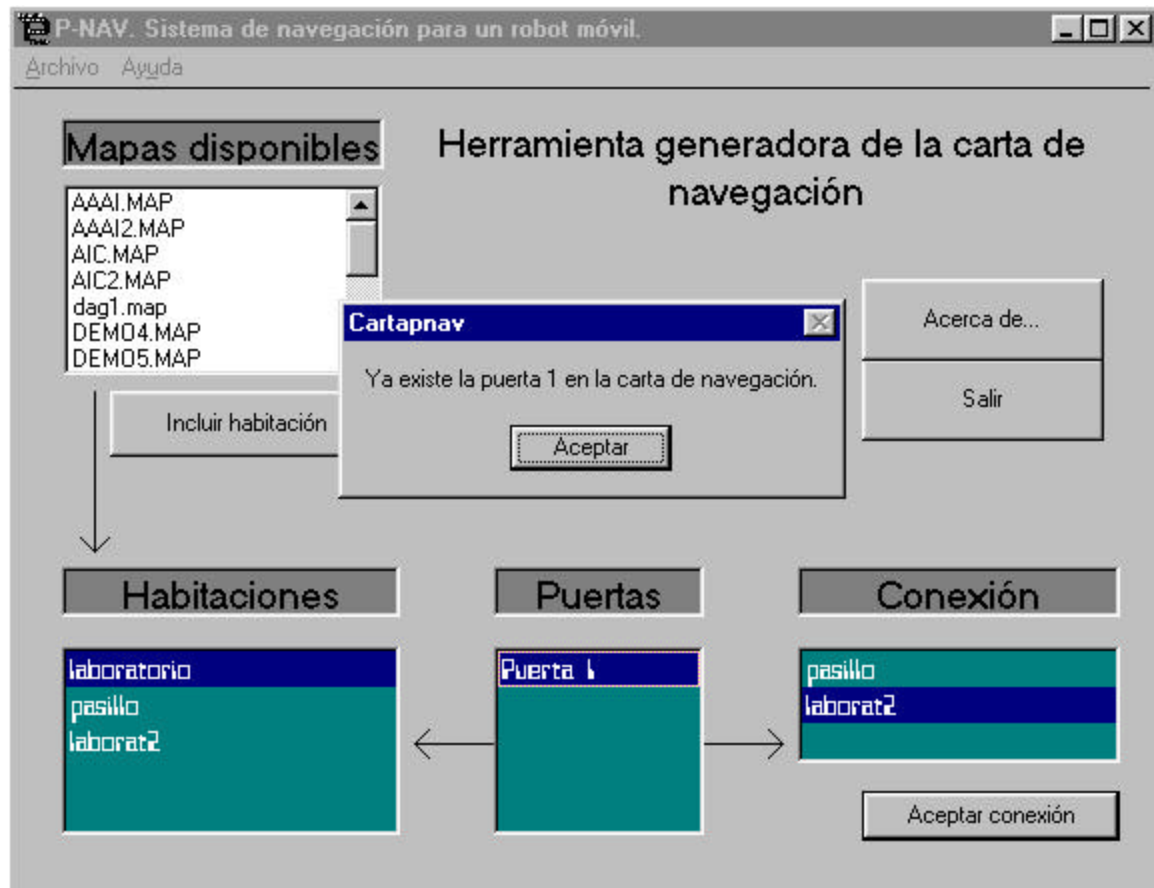


Figura 5.3.

Todo esto también lo puede hacer el usuario editando a mano el archivo “NAVEGACION.HYP” ubicado en el directorio que indica la variable de entorno “NAVEGACION”. Para ello conviene leer las siguientes líneas que indican el sencillo formato de este archivo.

Es de utilidad conocer que este archivo soporta comentarios que deben ir precedidos de un “;”.

Para declarar habitaciones hay que escribir en mayúsculas “HABITACIONES” y a continuación escribir línea por línea un número de identificación de cada habitación seguido de un nombre sin caracteres gramaticales como puede ser un punto o una coma, es decir, sólo números y letras. Este nombre se corresponderá con el campo del nombre del

archivo (sin extensión) del mapa de la habitación a tratar y por tanto con el mismo nombre de los archivos de matrices de distancias y de obstáculos. Es muy importante que así sea o de lo contrario el sistema de navegación no sabrá identificar una habitación con los archivos necesarios que debe interpretar para navegar por ella.

El número de identificación debe ser un entero positivo. Al menos debe haber especificada una habitación con identificador “1” que será la habitación en la que se iniciará la navegación.

A continuación se declararán las puertas. Para ello se debe escribir en mayúsculas la palabra “PUERTAS”. Después línea por línea editaremos los parámetros en números reales de cada puerta los cuales son:

Id: Identificador de la puerta, coincidirá con el identificador de la puerta en los

archivo “.MAP” que contienen citada la misma.

Hab1: Identificador de la primera habitación que comunica.

Posx\_1: Posición X de la puerta en la habitación 1.

Posy\_1: Posición Y de la puerta en la habitación 1.

Th\_1: Ángulo para salir de la habitación 1 por la puerta.

Posx\_2: Posición X de la puerta en la habitación 2.

Posy\_2: Posición Y de la puerta en la habitación 2.

Th\_2: Ángulo para salir de la habitación 2 por la puerta.

```

HABITACIONES
;Id      Nombre
1        LABORAT
2        PASILLO

PUERTAS
;Id      Hab1      Posx_1  Posy_1  Th_1  Hab2      Posx_2  Posy_2  Th_2
1        1        900.0   0.0    90.0  2        900.0   900.0   -90.0

```

Figura 5.4.



## 6.- Creación de un archivo de órdenes

Aunque el usuario puede disfrutar de una manera interactiva con la actividad del robot el funcionamiento recomendado de la navegación es mediante una lista de órdenes previamente programadas.

Esta tarea se hace mediante la edición de un archivo de actividades. Dicho archivo tiene extensión “.ACT” y se compone de una serie de llamadas a funciones.

Estos archivos son interpretados por el sistema Saphira y están compuestos por un lenguaje muy básico con la sintáxis de C, aunque es mucho más limitado y su función está destinada únicamente a hacer llamadas a funciones que realizarán ciertas actividades.

Para realizar las funciones de nuestro sistema de navegación se tienen 3 funciones y una variable que se deberá tener en cuenta constantemente. A continuación serán descritas:

- `void ir_a_pto(float x, float y, float th)`

Esta función se encarga de fijar un punto de destino dentro de la habitación actual y llevar al robot hacia él. Los parámetros x e y son las coordenadas en milímetros y deben ser números reales, es decir, debe suministrarse un campo de decimales aunque éstos sean nulos, por ejemplo, “1000.0”, y el parámetro th es el ángulo de orientación en grados del punto de destino. Este último parámetro sirve para que el robot al acercarse a su destino aproxime su orientación al valor especificado.

- `int ir_a_puerta(int id)`

Esta función se encarga de fijar como punto de destino las coordenadas del centro de la puerta en la habitación actual, aproximar el robot hasta él, cargar el mapa de la habitación con la que comunica y fijar las nuevas coordenadas. En caso de editar erróneamente el archivo de actividades e intentar llevar al robot a una puerta que no corresponde a la habitación actual la función devolverá un valor de “-1” en señal de error.

- `void pos_inicial(float x, float y, float th)`

Esta función, en caso de utilizarse, debería hacerse antes de llamar a cualquier otra función que provoque actividad en el robot o de lo contrario serán provocados errores de orientación y posicionamiento del robot dentro de la habitación. Su resultado es la colocación del robot en una posición en el mapa para partir de ese punto. Es decir, si esta función no es llamada antes de ejercer una actividad sobre el robot, el sistema colocará inicialmente el robot en la posición (300.0, 300.0, 0.0) respecto a la esquina inferior derecha dentro del mapa. Es decir, si dicha esquina se encuentra en el punto (-1000.0, -100.0) el robot se situará inicialmente dentro del mapa en la posición (-700.0, 200.0, 0.0).

- `int standby`

Esta variable será verdadera cuando el robot esté listo para recibir una nueva orden y será falsa mientras no haya terminado de ejecutar la anterior. Es muy importante por tanto escribir “waitfor(standby);” después de cada llamada a `ir_a_pto` o `ir_a_puerta`. Es una variable global y está predefinida en el sistema, está preparada para utilizarse como en este apartado se ha indicado.

Todas las órdenes deben integrarse dentro de una actividad definiendo a ésta con un nombre de la siguiente manera:

```
act nombre_actividad
{
    //órdenes
}
```

y será ejecutada escribiendo “start nombre\_actividad”.

```
act navega
{
    pos_inicial(4650.0,3750.0,180.0);
    sfSMessage("Vamos a cambiar de habitación...");
    ir_a_puerta(1);
    waitfor(standby);
    sfSMessage("Ahora nos damos una vueltecilla por aquí...");
    ir_a_pto(900.0,300.0,0.0);
    waitfor(standby);
    sfSMessage("Y volvemos de donde vinimos...");
    ir_a_puerta(1);
    waitfor(standby);
}

start navega;
```

Figura 6.1.

## 7.- Puesta en marcha

Una vez realizados con éxito todos los pasos para definir una zona de navegación ya se puede hacer viajar al robot por toda ella.

El modo de hacerlo es sencillo y se limita a la conexión con el robot mediante la comunicación previamente implementada en el sistema Saphira y, si así se desea, la carga de un archivo de actividades que especifique los pasos a seguir por el robot para llegar a donde interese al usuario.

Se comienza cargando el sistema Saphira que contiene la programación implementada para la navegación:

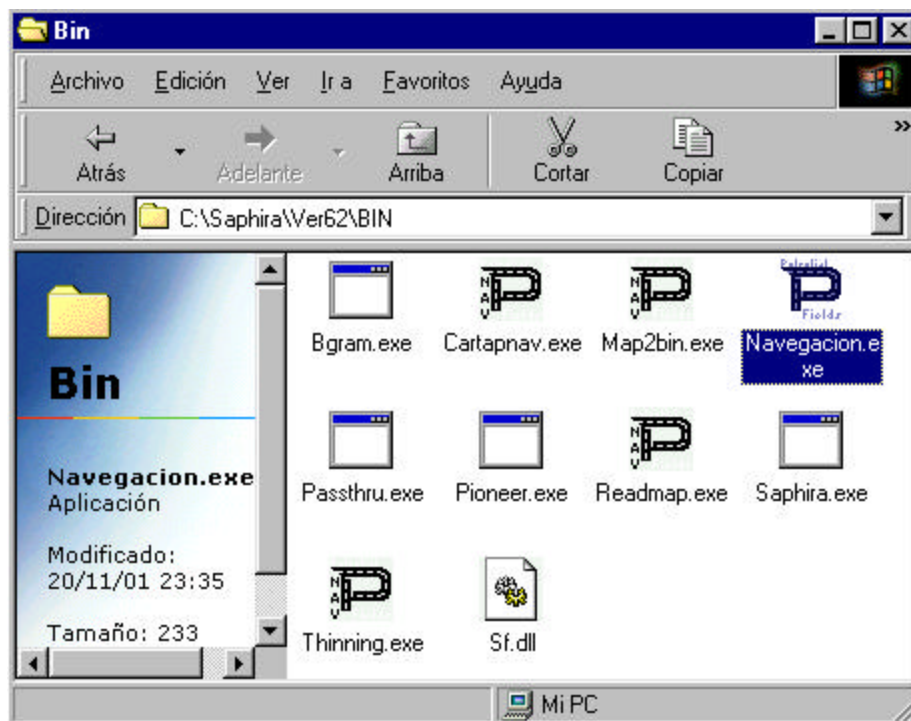


Figura 7.1.

Una vez cargado nuestro sistema debemos seleccionar en el menú principal el modo de conexión que necesitamos. Lo más habitual es que conectemos mediante el puerto serie como aparece en la figura 7.2:

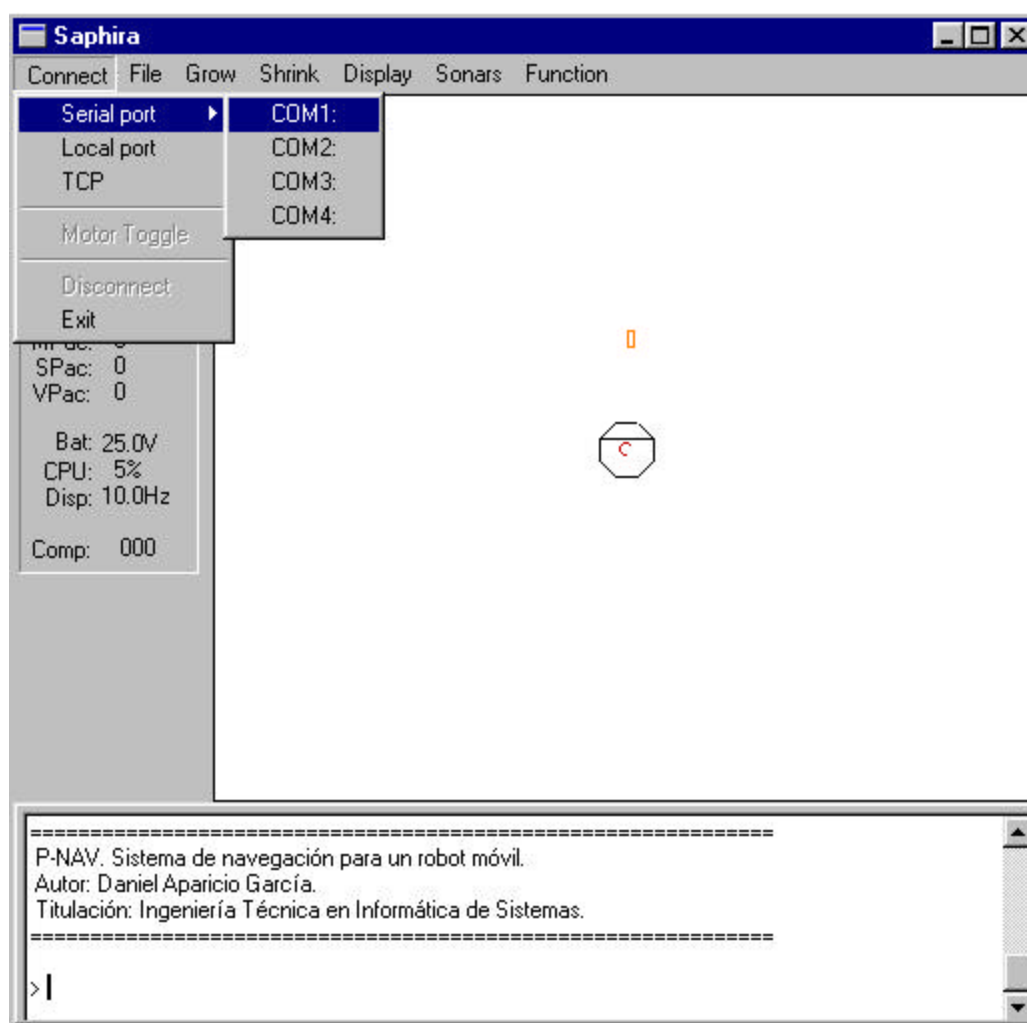


Figura 7.2.

Si la conexión se produce con éxito aparecerá en pantalla el mapa de la habitación que en la carta de navegación identificamos con el número “1” y cargará por omisión las características del robot denominado “AmigoBOT” contenidas en el archivo “AMIGO.P”. Si se desea cambiar de robot tendremos que cambiarlo en “File → Params”, pero antes se deberán crear los mapas de bits y de distancias adecuados a dicho robot como se indica en el apartado 4.

Si se desea aumentar o disminuir el zoom lo podemos hacer en el menú principal con las opciones “Grow” ó “Shrink”:

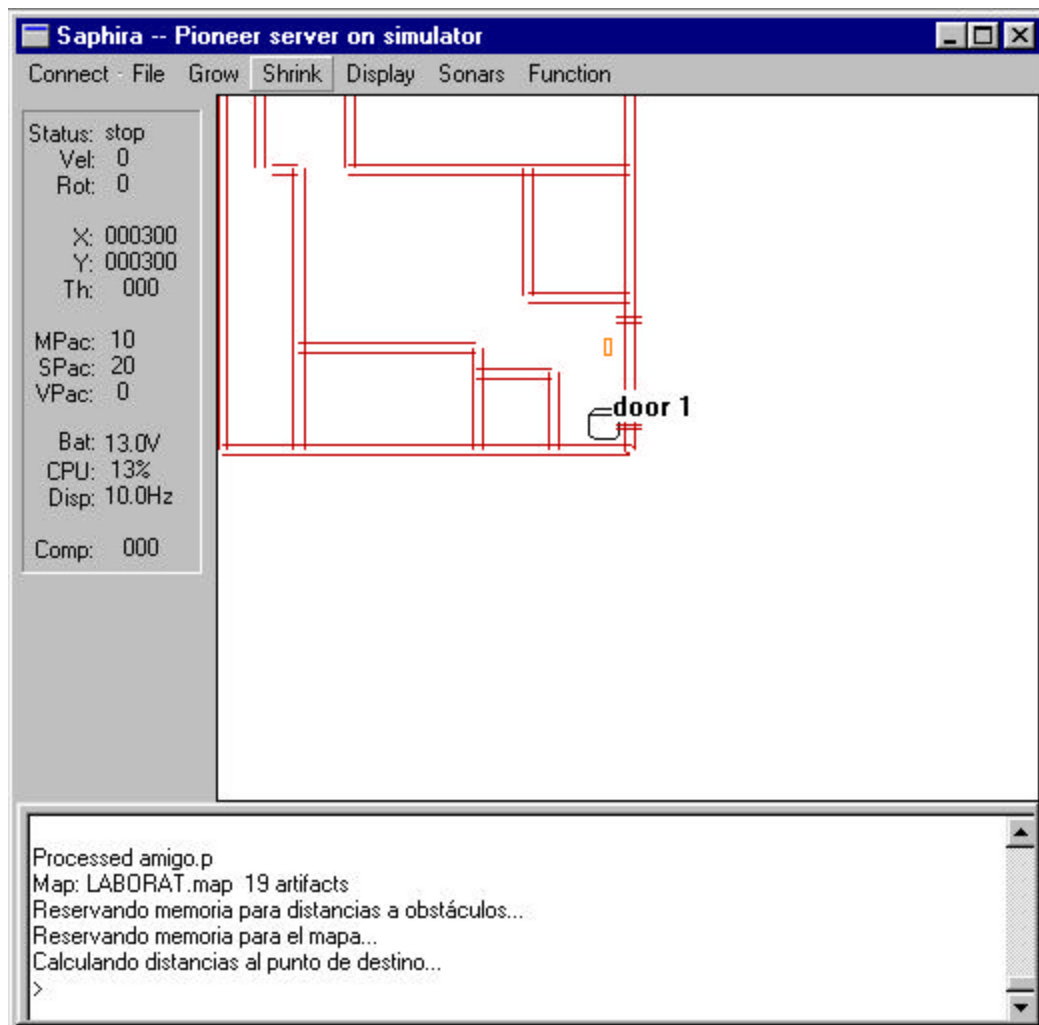


Figura 7.3.

NOTA: Véase que al no haber llamado a la función *pos\_inicial* el robot aparece en la posición (300.0, 300.0, 0.0) respecto a la esquina inferior derecha del mapa.

En estos momentos es posible navegar por la habitación de una manera interactiva pulsando con el puntero del ratón en el lugar de la sala al que queremos que se dirija el robot o bien escribiendo en la línea de comandos del sistema la orden que deseamos que el robot realice (ver funciones “pos\_inicial”, “ir\_a\_puerta” e “ir\_a\_pto” en el apartado 6):

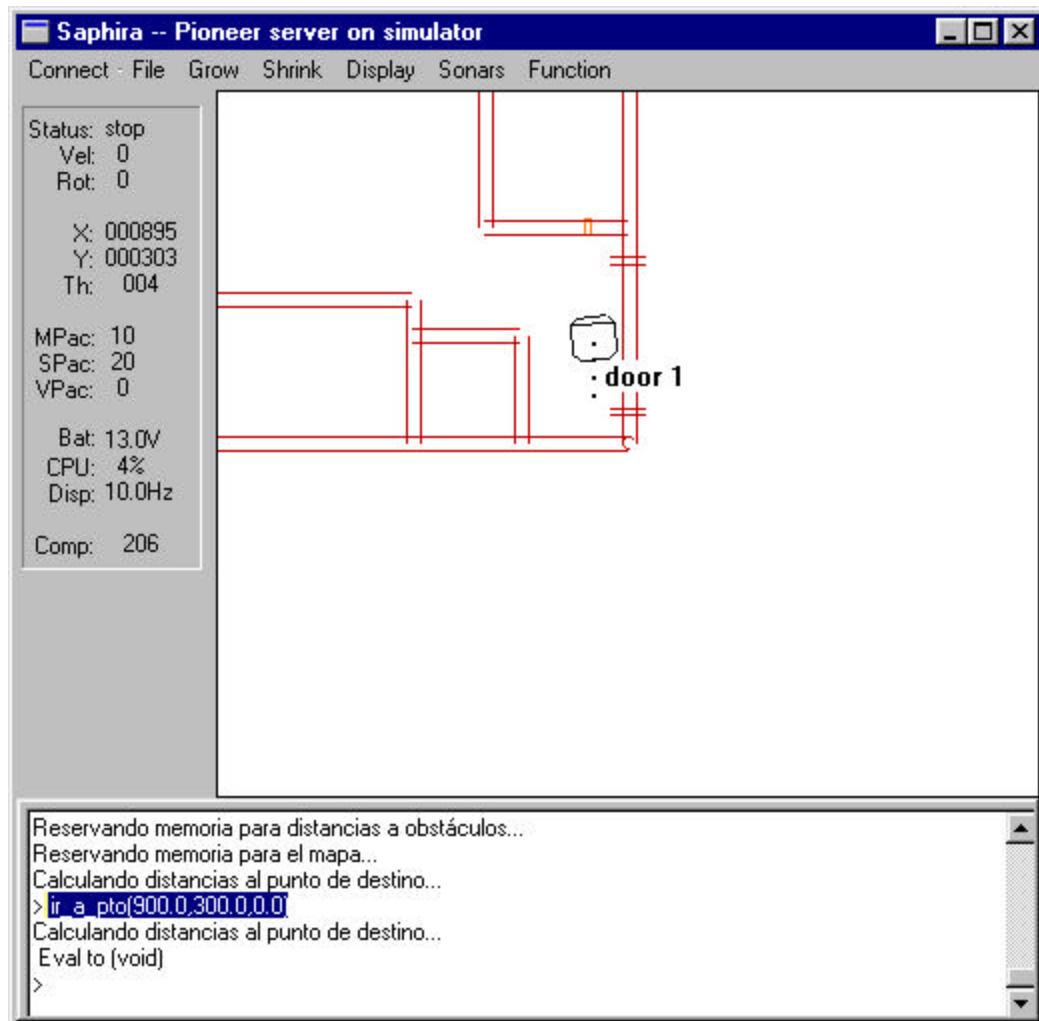


Figura 7.4.

De otro modo podemos editar tal y como se indica en el apartado 6 un archivo de órdenes y hacer que el sistema lo interprete.

Para proceder a la carga de este archivo podemos hacerlo de 2 maneras distintas:

- cargándolo mediante una línea de comandos desde el símbolo de sistema de Saphira mediante la instrucción “load”:

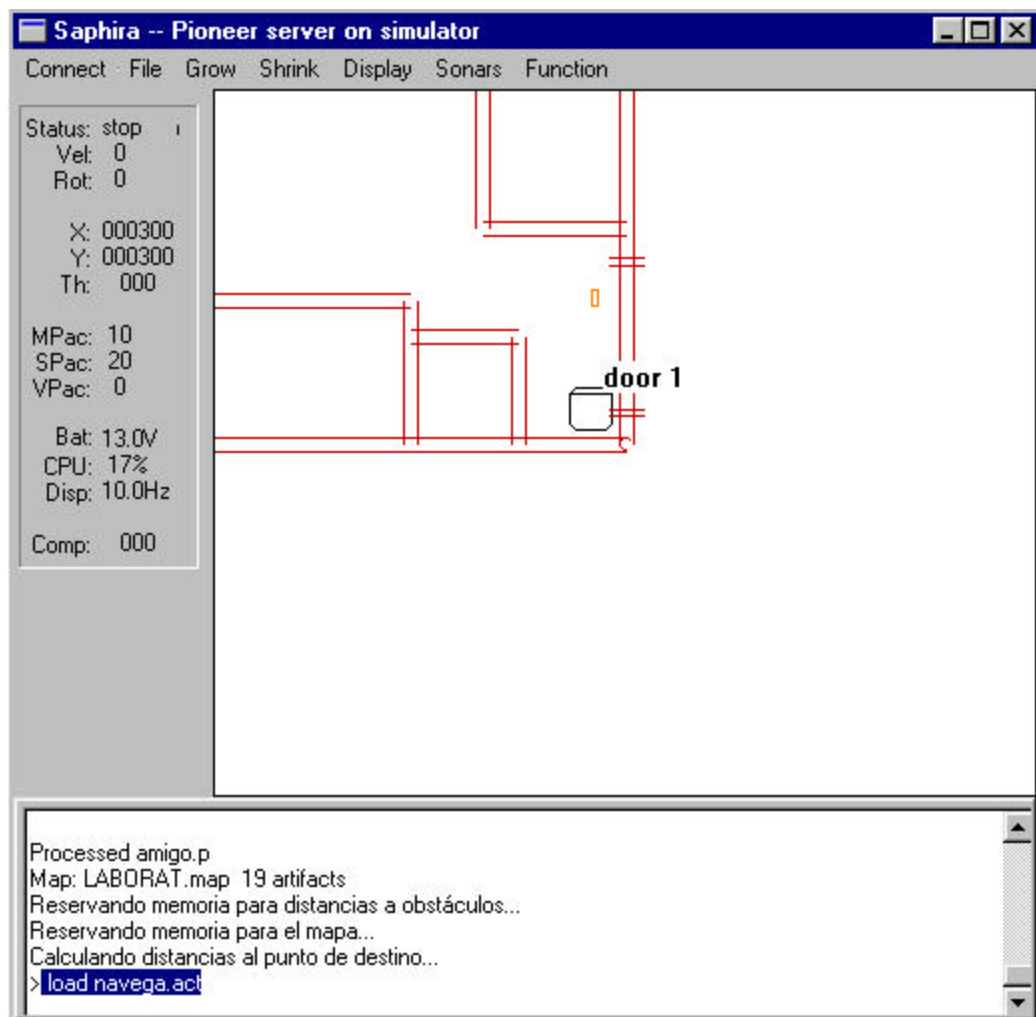


Figura 7.5.



- Seleccionando desde el menú principal la carga del archivo en File→Load File...:

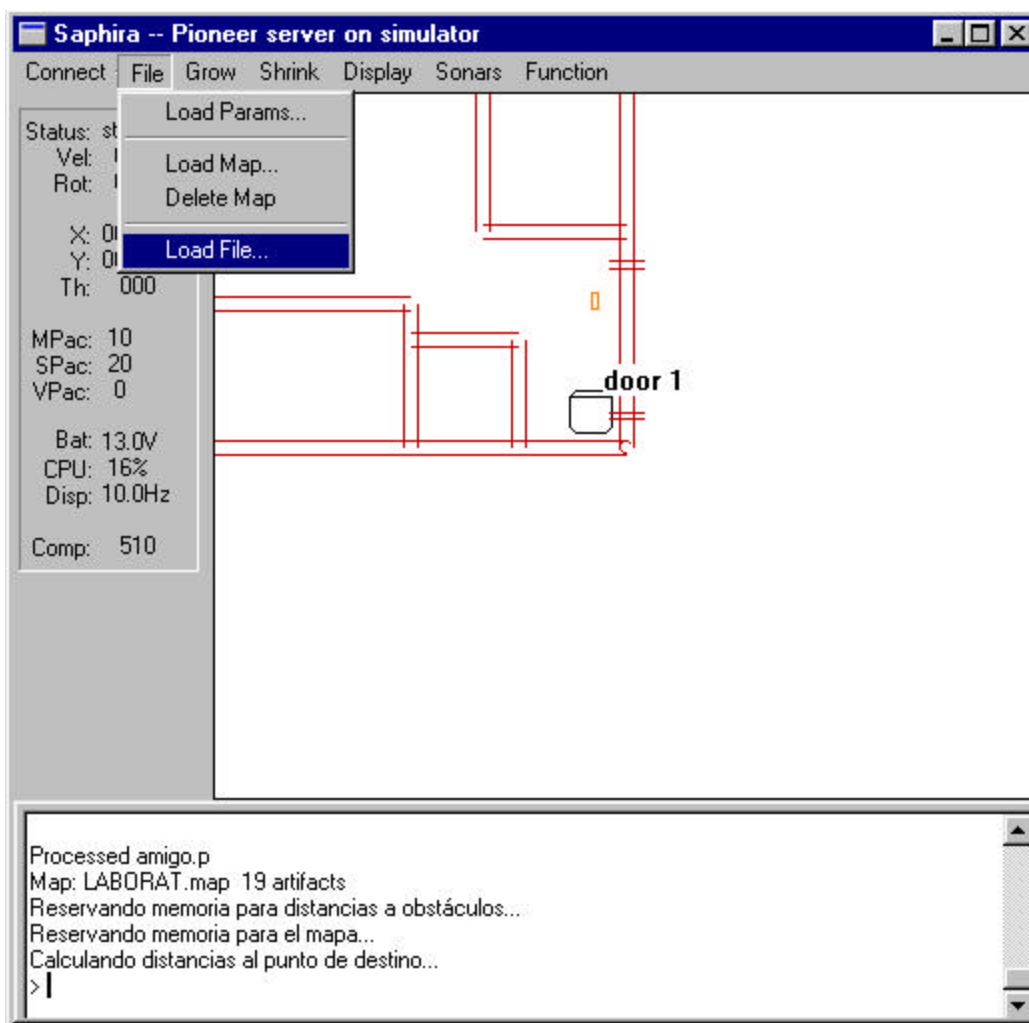


Figura 7.6.

Aparece así el cuadro de diálogo para abrir el archivo deseado:

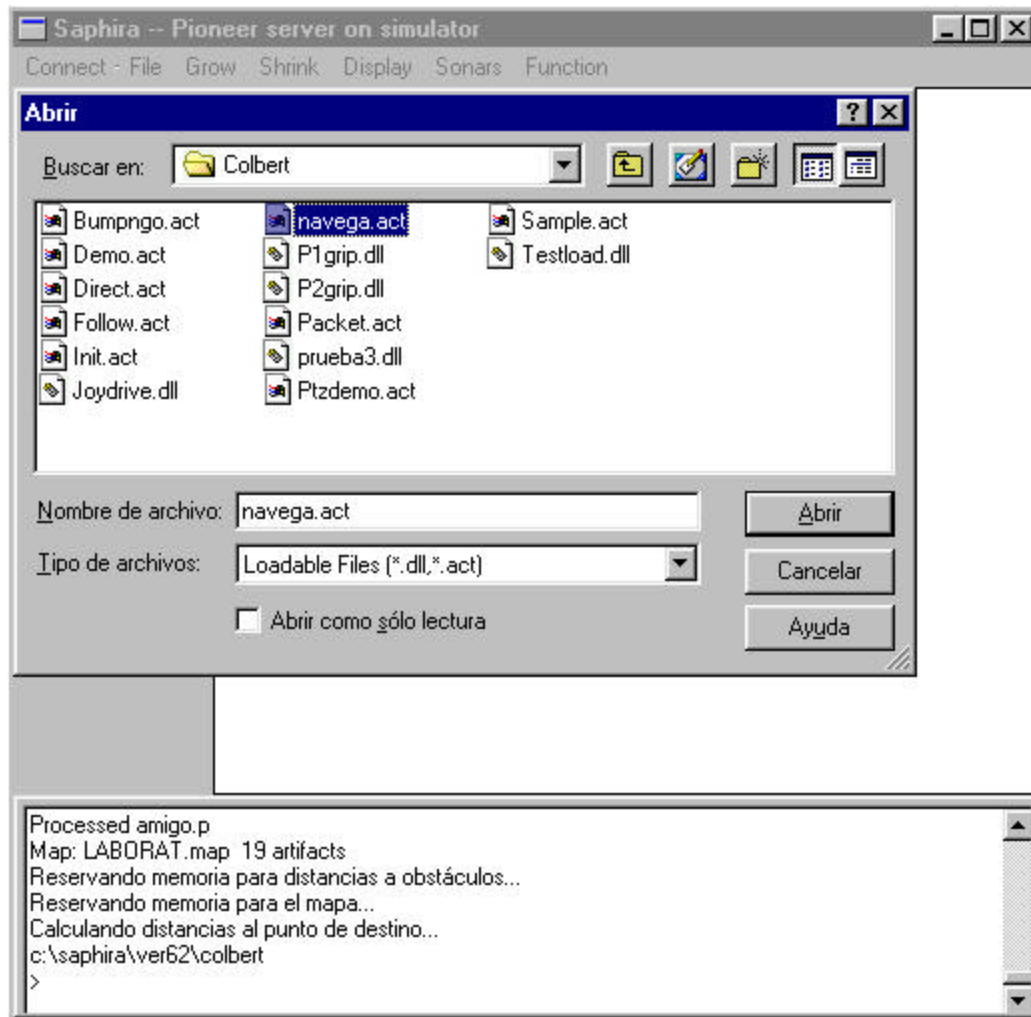


Figura 7.7.

A partir de aquí el robot se pondrá en funcionamiento y seguirá las órdenes programadas en el archivo de actividades. De esta manera se le puede hacer caminar hacia cualquier punto accesible del mapa y atravesar habitaciones (pasando por las puertas, claro está) para llegar al punto de destino deseado. Cabe recordar que cada habitación es tratada individualmente, por lo cual no se le puede decir al sistema que vaya a un punto de destino que en el momento actual no exista en el mapa, por lo que habrá que dar órdenes de cambiar de habitación hasta llegar a la que contiene el punto de destino deseado.

También es importante tener en cuenta que las coordenadas locales de una habitación no son las mismas que las de otra habitación, por ejemplo, si estamos en una habitación y queremos llegar al punto (1000.0, 1000.0 , 0.0) de otra habitación no se puede pedir directamente que se vaya al punto citado, sino que antes hay que llegar a esa habitación y luego dirigirse al punto deseado. Ver ejemplo del apartado 6.

## **8.- Utilizando el sistema Saphira...**

Saphira , visualizándolo a nivel de usuario, podría afirmarse que es una aplicación cliente que actúa de interfaz con el robot y está dotada de un protocolo de comunicación con el servidor (robot) y un intérprete de comandos que hace posible la interpretación de un programa de actividades escrito en un lenguaje parecido a C e interpretado por un intérprete que lleva integrado y se llama Colbert. También ofrece la posibilidad de la carga de bibliotecas dinámicas que pueden ampliar sus posibilidades añadiendo nuevos comportamientos y funciones.

A nivel visual es una aplicación con interfaz gráfica (GUI) que permite al usuario hacerse con el manejo del robot de una manera sencilla.

En los apartados anteriores se explica cómo establecer una comunicación con el robot y ponerlo a funcionar, pero aún se puede profundizar más en el conocimiento de este sistema de manera que el usuario de éste se dará cuenta que se trata de un entorno abierto a todo tipo de ampliaciones que se deseen implementar y con el cual el usuario tiene total control del robot que es regido por Saphira.

El usuario puede en todo momento conocer los comportamientos que se están llevando a cabo pinchando en el menú Function→Behaviors:



Figura 8.1.

De esta manera aparece una ventana que lista los comportamientos registrados en el sistema y su actividad:

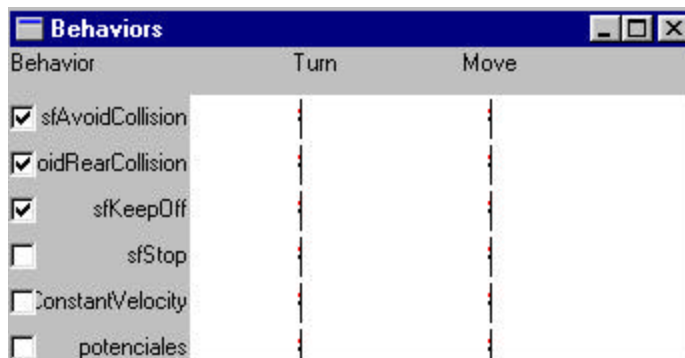


Figura 8.2.

Pinchando en las casillas de la izquierda conectaremos o desconectaremos los comportamientos elegidos, aunque se recomienda no hacerlo con “sfStop”, con “potenciales” ni con “sfAvoidCollision” ya que influiría negativamente en el funcionamiento de la navegación ya que estos tres comportamientos se activan y desactivan automáticamente para conseguir el objetivo de llegar al punto de destino.

Lo mismo se puede hacer con las actividades presentes en el sistema mediante el menú Function→Activities...



Figura 8.3.

...obteniendo así una ventana que presenta las actividades y su estado:

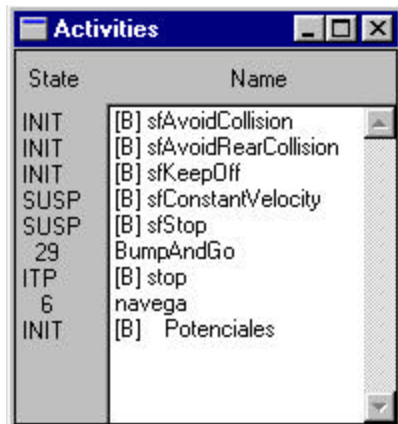


Figura 8.4.

El estado de las actividades lo podemos modificar pulsando con el ratón sobre su nombre, aunque de nuevo cabe decir que no es recomendable hacerlo en ninguna de las actividades que se relacionen directamente con la navegación para que ésta pueda tener un resultado favorable.