



Universidad
Carlos III de Madrid

Departamento de Informática

PROYECTO FIN DE CARRERA

DESARROLLO DE UNA
INTERFAZ HOMBRE-ROBOT
PARA UN ROBOT GUÍA EN
INTERIORES

Autor: Serrano Sánchez-Bravo, Laura

Tutor: Quintero Barrios, Ezequiel Antonio

Co-Director: Fernández Rebollo, Fernando

Leganés, julio de 2011

Título: Desarrollo de una Interfaz Hombre-Robot para un Robot Guía en Interiores.

Autor: Serrano Sánchez-Bravo, Laura.

Tutor: Quintero Barrios, Ezequiel Antonio.

Co-Director: Fernández Rebollo, Fernando.

EL TRIBUNAL

Presidente: _____

Vocal: _____

Secretario: _____

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día __ de _____ de 20__ en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE

A mis padres

Agradecimientos

En primer lugar quiero agradecer a mi tutor, Ezequiel Quintero, y al co-director de este proyecto, Fernando Fernández, por su gran dedicación e inestimable ayuda.

De forma muy especial quiero agradecer a mis padres, M^a Trinidad y Mariano, por su amor y por la educación que me han dado, sin ellos no sería la persona que soy. También agradecer a mi hermano, Mario, por su cariño, y a mis abuelos, que sin duda han desempeñado un papel muy importante en mi vida. Cómo no, debo dar las gracias a mi novio, Jorge, por su constante apoyo y gran ayuda durante todo el proyecto.

También debo dar las gracias a mis compañeros de carrera, por su confianza y por los momentos vividos, sin ellos estos años no habrían sido lo mismo.

Por último, debo agradecer a todas aquellas personas que han participado realizando la evaluación de este proyecto, muchas gracias por vuestra ayuda y colaboración.

Resumen

En la actualidad, los robots móviles han logrado un gran avance, siendo empleados para facilitar la vida de las personas, ayudándoles a realizar tareas repetitivas o que pueden resultar complicadas, tediosas o incluso peligrosas para el ser humano [1] [2] [3] [4].

Existen ciertos lugares como museos, bibliotecas o universidades que, debido al tipo de actividad que se realiza en ellos, son visitados por un gran número de personas todos los días. Estas personas, en muchas ocasiones no conocen el lugar y les resulta complicado guiarse por el mismo si es la primera vez que lo visitan. El desarrollo de sistemas robóticos guías puede resultar muy provechoso en este tipo de situaciones, en las que es posible contar con un robot móvil que acompañe al visitante durante su estancia en el edificio. Debido a las limitaciones de los robots, éstos no pueden percibir ciertos elementos de su entorno como carteles o números de despacho que ayuden en su ubicación. Además, debido al ruido en sensores y a problemas de precisión en técnicas como la odometría, la robótica móvil tiene dificultades en cuanto a la localización y existen determinadas limitaciones que podrían ser cubiertas por medio de los humanos. Una persona que visita por primera vez un edificio no conoce la distribución del mismo ni la forma de desplazarse por él, mientras que un robot guía sí puede tener esta información. Por otro lado, en algunas ocasiones un robot puede tener problemas con su localización, mientras que un humano puede percibir determinados elementos del entorno que posibiliten esta ubicación. Para solucionar el problema de desplazamiento del humano y el de localización del robot es posible una colaboración entre ambos para ayudarse mutuamente, guiando el robot al usuario a través del edificio y asistiendo el humano al robot en caso de requerir de su ayuda.

En este proyecto se realiza el desarrollo de una interfaz hombre-robot táctil que permite la interacción de un usuario con un robot Pioneer P3-DX que le guíe en el interior de la planta primera del edificio Sabatini de la Universidad Carlos III de Madrid. La interfaz gráfica construida permite además la asistencia al robot por parte del usuario en los casos en que éste requiera de su colaboración ya sea por encontrar un obstáculo que le impida continuar su desplazamiento o por no conocer el lugar exacto de su localización dentro del mapa del edificio. El robot cuenta además con una arquitectura de control con la que se deberá integrar la interfaz desarrollada.

Palabras clave: arquitectura de control robótico, interacción hombre-robot, odometría, planificación automática, robot guía.

Abstract

In recent years, mobile robots have received a great progress, and they are starting to make life easier for people, helping them to do repetitive, complicated, tedious or even dangerous tasks to humans [1][2][3][4].

There are many places such as museums, libraries or universities which are visited by a large number of people every day. That people often does not know the place and to guide themselves in their first visits is difficult. The development of guide robotic systems can be very helpful in these type of situations and mobile robots can accompany the visitor during his stay in the building. However, due to the limitations of the robots, they can not perceive certain elements of their environment as posters or office number which would support them in the localization task. In addition, due to noise in sensors and precision problems in techniques such as odometry, mobile robotics have difficulties in the location and there are certain limitations that might be covered through persons: on one hand, a person who visits a building for the first time does not know the distribution of the building or how to move inside it, but a guide robot can have this information by itself; on the other hand, sometimes a robot may have problems with its location, while a human can perceive certain environmental elements that help in his location. To solve the problem of human displacement and the robot location, it is possible a partnership between them to help each other: the robot guides the user through the building and, if necessary, the human assists the robot when it has location problems.

This project develops a tactile human-robot interface, that allows the interaction of a user with a Pioneer P3-DX robot, that helps him to guide in the first floor of Sabatini University Carlos III building (Madrid). This graphical interface also provides the robot with assistance by the user, in the cases where the robot requires his cooperation. This cooperation is required when an obstacle that prevents it to continue its displacement and when its location inside the plane of the building is unknown. In addition, the robot owns a control architecture which integrates the developed interface.

Keywords: robot control architecture, human-robot interaction, odometry, automated planning, guide robot.

Índice general

| | | |
|---|--|----|
| 1 | Introducción y objetivos | 1 |
| | 1.1 Objetivos | 3 |
| | 1.2 Fases del desarrollo..... | 4 |
| | 1.3 Estructura de la memoria | 4 |
| 2 | Estado del Arte..... | 7 |
| | 2.1 Robots Móviles..... | 7 |
| | 2.2 Arquitectura ROPEM..... | 10 |
| | 2.3 Trabajos Relacionados | 11 |
| 3 | Interfaz Hombre-Robot para un Robot Guía en Interiores | 17 |
| | 3.1 Análisis..... | 18 |
| | 3.1.1 Diagrama de Casos de Uso | 18 |
| | 3.1.2 Requisitos del Sistema | 19 |
| | 3.1.2.1 Interfaz Gráfica y Sistema de Voz | 20 |
| | 3.1.2.2 Comunicación Interfaz y Arquitectura ROPEM | 20 |
| | 3.2 Diseño | 21 |
| | 3.2.1 Diseño de la Arquitectura | 22 |
| | 3.2.2 Diseño de la Interfaz Gráfica y Sistema de Voz | 23 |
| | 3.2.2.1 Diseño Preliminar | 23 |
| | 3.2.2.2 Diseño Final | 25 |
| | 3.2.3 Comunicación Interfaz y Arquitectura ROPEM | 33 |
| | 3.3 Diseño Detallado | 39 |
| | 3.3.1 Arquitectura ROPEM | 39 |
| | 3.3.2 Interfaz Gráfica y Sistema de Voz..... | 44 |
| | 3.3.3 Comunicación Interfaz y Arquitectura ROPEM | 45 |
| 4 | Evaluación y Pruebas Preliminares | 50 |
| | 4.1 Pruebas Preliminares..... | 50 |
| | 4.1.1 Funcionamiento Correcto..... | 50 |
| | 4.1.2 Interrupción de STOP | 51 |
| | 4.1.3 Interrupción de Reubicar..... | 53 |
| | 4.1.4 Errores durante el Desplazamiento..... | 54 |
| | 4.1.5 Interrupciones y Errores durante el Desplazamiento | 55 |
| | 4.1.6 Búsqueda con el Directorio | 57 |
| | 4.2 Evaluación..... | 58 |
| 5 | Gestión del Proyecto | 64 |
| 6 | Conclusiones y Líneas Futuras | 70 |
| 7 | Bibliografía | 73 |

| | | |
|----|-------------------|----|
| 8 | Glosario | 76 |
| 9 | Referencias | 78 |
| 10 | Anexo A..... | 81 |
| 11 | Anexo B..... | 87 |
| 12 | Anexo C..... | 90 |
| 13 | Anexo D..... | 92 |
| 14 | Anexo E..... | 95 |

Índice de figuras

| | |
|--|----|
| Figura 1. Robots Móviles | 8 |
| Figura 2. Robots de Servicio | 9 |
| Figura 3. Roomba, robot aspiradora | 9 |
| Figura 4. Robot Pioneer P3-DX | 10 |
| Figura 5. Descripción del Entorno de la Aplicación..... | 11 |
| Figura 6. Robots de la Ciudad Banco Santander | 12 |
| Figura 7. Sacarino, robot botones | 13 |
| Figura 8. REEM, robot guía | 13 |
| Figura 9. COBOT, robot guía..... | 14 |
| Figura 10. SENA, silla robótica..... | 14 |
| Figura 11. Diagrama de Casos de Uso..... | 19 |
| Figura 12. Diseño de la Arquitectura Desarrollada | 22 |
| Figura 13. Primera Prueba de Interfaz Gráfica..... | 23 |
| Figura 14. Pantalla de Inicio del Prototipo de la Interfaz | 24 |
| Figura 15. Pantalla de Desplazamiento hacia Destino del Prototipo de la Interfaz | 24 |
| Figura 16. Pantallas de Ayuda al Robot del Prototipo de la Interfaz..... | 25 |
| Figura 17. Pantalla de Inicio de la Interfaz Desarrollada..... | 25 |
| Figura 18. Pantallas de Búsqueda de Destino de la Interfaz Desarrollada..... | 26 |
| Figura 19. Pantallas de Búsqueda en el Directorio de la Interfaz Desarrollada | 26 |
| Figura 20. Pantallas de Búsqueda del Origen de la Interfaz Desarrollada..... | 27 |
| Figura 21. Pantalla de Desplazamiento del Robot de la Interfaz Desarrollada | 27 |
| Figura 22. Pantallas de Ayuda al Robot de la Interfaz Desarrollada | 28 |
| Figura 23. Pantalla de Confirmación de Origen | 28 |
| Figura 24. Navegación de Pantallas | 29 |
| Figura 25. Plano Técnico de la Primera Planta del Edificio Sabatini de la Universidad Carlos III de Madrid..... | 30 |
| Figura 26. Plano Simplificado de la Primera Planta del Edificio Sabatini de la Universidad Carlos III de Madrid..... | 31 |
| Figura 27. Ampliación del Plano de la Zona B del Edificio Sabatini de la Universidad Carlos III de Madrid..... | 31 |
| Figura 28. Diagrama de Clases Simplificado | 32 |
| Figura 29. Intercambio de Mensajes entre Interfaz y Robot | 36 |
| Figura 30. Representación Jerárquica del Plano..... | 37 |
| Figura 31. Ejemplo de las Coordenadas de un Área del Plano | 38 |
| Figura 32. Ejemplo de la División del Plano..... | 38 |
| Figura 33. Ejemplo de Cálculo de Coordenadas de Desplazamiento | 41 |

| | |
|---|----|
| Figura 34. Ejemplo de Cálculo de Coordenadas de Desplazamiento al Recibir un Nuevo Problema..... | 41 |
| Figura 35. Ejemplo de Cálculo de Desplazamiento en el Eje y Negativo | 42 |
| Figura 36. Ejemplo de Cálculo de Desplazamiento en el Eje y Negativo al Recibir un Nuevo Problema | 43 |
| Figura 37. Ejemplo de Cálculo de Desplazamiento en el Eje y Positivo al Recibir un Nuevo Problema | 44 |
| Figura 38. Creación y uso del fichero PDDL del problema | 47 |
| Figura 39. Creación y uso de los ficheros PDDL parcial y completo..... | 48 |
| Figura 40. Prueba de Funcionamiento Correcto de la Aplicación Desarrollada | 51 |
| Figura 41. Prueba de Interrupción de STOP de la Aplicación Desarrollada..... | 52 |
| Figura 42. Prueba de Interrupción de Reubicar de la Aplicación Desarrollada | 53 |
| Figura 43. Prueba de Errores durante el Desplazamiento de la Aplicación Desarrollada | 55 |
| Figura 44. Prueba con Interrupción y Errores durante el Desplazamiento de la Aplicación Desarrollada..... | 56 |
| Figura 45. Prueba de Búsqueda de Destino por medio del Directorio de la Aplicación Desarrollada..... | 57 |
| Figura 46. Diagrama de Gantt del Proyecto | 66 |
| Figura 47. Diagrama de Clases | 88 |

Índice de tablas

| | |
|--|----|
| Tabla 1. Resultados de la Prueba de Funcionamiento Correcto de la Aplicación Desarrollada..... | 51 |
| Tabla 2. Resultados de la Prueba de Interrupción de STOP de la Aplicación Desarrollada..... | 52 |
| Tabla 3. Resultados de la Prueba de Interrupción de Reubicar de la Aplicación Desarrollada..... | 54 |
| Tabla 4. Resultados de la Prueba de Errores durante el Desplazamiento de la Aplicación Desarrollada..... | 55 |
| Tabla 5. Resultados de la Prueba con Interrupción y Errores durante el Desplazamiento de la Aplicación Desarrollada..... | 56 |
| Tabla 6. Resultados de la Prueba de Búsqueda de Destino por medio del Directorio de la Aplicación Desarrollada..... | 57 |
| Tabla 7. Resultados de las Mediciones de las Evaluaciones Realizadas | 61 |
| Tabla 8. División de Tareas del Proyecto | 65 |
| Tabla 9. Desglose Presupuestario del Personal | 67 |
| Tabla 10. Desglose Presupuestario de Equipos | 68 |
| Tabla 11. Resumen de costes del Proyecto | 68 |
| Tabla 12. Descripción Textual Caso de Uso Solicitar Ayuda para desplazarse a un destino por medio del plano | 81 |
| Tabla 13. Descripción Textual Caso de Uso Solicitar Ayuda para desplazarse a un destino por medio del directorio | 82 |
| Tabla 14. Descripción Textual Caso de Uso Corregir Localización Robot | 83 |
| Tabla 15. Descripción Textual Caso de Uso Orientar Robot | 84 |
| Tabla 16. Descripción Textual Caso de Uso Solicitar Ayuda | 84 |
| Tabla 17. Descripción Textual Caso de Uso Navegar | 85 |

Capítulo 1

Introducción y objetivos

El avance de la tecnología ha hecho posible la creación y el perfeccionamiento de nuevos sistemas capaces de complementar la vida de las personas mejorando ciertas actividades y la calidad de vida. Uno de estos sistemas, que está teniendo un gran avance hoy en día, son los sistemas de control robótico, que realizan gran cantidad de actividades en diversos campos relacionados con funciones que sirven para facilitar la vida de las personas. Los robots suelen llevar a cabo tareas repetitivas y que en muchas ocasiones podrían resultar tediosas o suponer un problema para el ser humano. De esta forma, el robot suplanta a la persona realizando estas acciones y ayudándole así en su vida diaria. Para que esta ayuda sea aún mayor y que los robots puedan realizar sus tareas dependiendo lo menos posible de la ayuda del hombre, surgió el desarrollo de los robots móviles. Al tratarse de máquinas capaces de desplazarse por sí solas y que no se encuentran sujetas a ningún tipo de anclaje ni ningún otro elemento que restrinja su operacidad, pueden maniobrar con libertad, aumentando su autonomía y minimizando la intervención humana.

Los robots móviles se desplazan en situaciones en las que los elementos cambian dinámicamente y en los que es necesaria alguna técnica que decida qué acciones deben realizar según la información percibida en cada momento. Gracias a la inteligencia artificial, se han podido desarrollar sistemas capaces de percibir el entorno y decidir qué acciones se han de ejecutar dependiendo de la información recibida. Para ello, se emplea la planificación automática [5], que permite decidir las acciones a realizar dependiendo del estado inicial, de la descripción del entorno y del objetivo final a ser alcanzado con independencia del dominio y del problema a ser resuelto.

Los robots móviles pueden ser empleados en multitud de campos y tareas [1]. Una de las posibles acciones a realizar es servir de guía en determinado tipo de edificios. Existen ciertos lugares como museos, bibliotecas o universidades que, debido al tipo de actividad que se realiza en ellos, son visitados por un gran número de personas todos los días. Estas personas, en muchas ocasiones no conocen el lugar, debido a que se trata de

edificios grandes y en los que resulta complicado ubicarse si es la primera vez que se visitan. Para ayudar a estas personas a desplazarse por el edificio, se suele contar con medios tales como planos o señales indicativas, pero en muchas ocasiones esto no es suficiente y es necesario acudir a personas que guíen al visitante hasta su destino, lo cual puede resultar una tarea complicada si el número de visitantes es elevado y que además requiere que existan personas dedicadas exclusivamente a esta actividad. En este ámbito, la robótica puede resultarnos muy útil, ya que un robot móvil podría suplantar a la persona guía, acompañando al usuario durante toda su visita por el edificio. De esta forma, el visitante sería guiado en todo momento por el edificio hasta el destino que haya seleccionado, sin problemas de localización y sin la necesidad de disponer de una persona que le acompañe durante todo el recorrido.

A pesar de que los robots han logrado una gran autonomía al dotarles de elementos que permiten su movilidad, no son completamente autosuficientes y tienen determinadas limitaciones que pueden ser cubiertas por medio de los humanos. La robótica móvil tiene problemas en cuanto a la localización, ya que el ruido en sensores limita la precisión del robot y la localización en interiores resulta difícil. Para la localización de robots en interiores suelen emplearse técnicas como cámaras, Wifi u odometría. La odometría es una de las técnicas más empleadas y extendidas y, en algunas ocasiones, puede tener problemas de precisión debido a la acumulación de errores al recorrer grandes distancias. Por este motivo, el hecho de que las personas colaboren con el robot en cuanto a la localización del mismo sería muy ventajoso, ya que podrían ubicarle en la posición correcta o ayudarlo a realizar tareas en las que el robot no puede valerse por sí mismo. De esta forma, obtenemos una ayuda recíproca por parte del robot al humano como del humano al robot. Por medio de la interacción entre el humano y el robot se pretende superar las restricciones que sufre el robot al mismo tiempo que se ayuda a los seres humanos en sus tareas diarias.

En este proyecto, el robot se orienta por medio de la odometría proporcionada por su base móvil. Este sistema genera error que ocasiona que la localización no sea correcta. La odometría proporcionada por su base móvil, junto con la información percibida por medio del *sonar* o de los *bumpers*, es todo lo que el robot puede percibir. Por el contrario, el humano puede percibir más información como carteles, números de despacho, zonas, imágenes del plano, etc. Con la ayuda del humano y, empleando esta información, es posible corregir los errores que genera el sistema de localización del robot.

Para que se dé esta ayuda, es necesario que la persona y el robot puedan comunicarse de una forma sencilla. Para ello se precisa de una interfaz que sirva de comunicación entre ambos y que el usuario pueda utilizar fácilmente. La interfaz debe ser amigable e intuitiva y que sirva para que el humano pueda comunicar cualquier hecho necesario al robot y para que este último informe al usuario de cualquier necesidad o hecho importante que ocurra.

En este proyecto, se desarrollará una interfaz que sirva como comunicación entre el usuario y el robot. Esta interfaz no sólo es necesaria para que el usuario le comunique al robot el lugar al que desea dirigirse, sino que también deberá ser usada para que el humano ayude al robot en caso de que éste tenga algún problema debido a sus limitaciones. Estas limitaciones surgen debido a que la odometría proporcionada por su

base móvil genera error y a que los movimientos que realiza el robot no son perfectos y en algunas ocasiones se choca, se pierde, encuentra algún obstáculo o no es capaz de moverse por determinados lugares sin ayuda. La interfaz construida, por lo tanto, debe servir para que el robot colabore con el humano mostrándole el camino a seguir para llegar a su destino y para que la persona ayude al robot en el caso en que éste requiera de su cooperación. En este proyecto se ha empleado un robot Pioneer P3-DX que cuenta con una arquitectura de control que permite enviar las órdenes de acción a realizar y obtener la información de los sensores del mismo.

1.1 Objetivos

El objetivo fundamental del proyecto es desarrollar una interfaz hombre-robot táctil que sirva para la interacción con un robot guía en el interior de un edificio.

La interfaz servirá para que el usuario y el robot puedan comunicarse y cooperen para alcanzar el objetivo final. El usuario indicará al robot, por medio de la interfaz, el lugar al que quiere desplazarse, bien a través del plano o bien a través de un directorio que contiene los datos del personal de la universidad, y el robot se desplazará hacia el lugar seleccionado. El movimiento del robot no es perfecto, sino que tiene una serie de errores que son tenidos en cuenta para disminuir su efecto por medio de la asistencia del usuario. De esta forma, cuando el robot se ha chocado o se ha encontrado con un obstáculo que le impide seguir avanzando, solicita ayuda al usuario que, por medio de la interfaz, le recolocará en el pasillo. La interfaz contará también con un sistema de voz para la bienvenida, la llegada al destino y los avisos de error.

En base a este objetivo general, se proponen los siguientes objetivos parciales:

- Diseñar e implementar una interfaz gráfica para la interacción del humano con el robot. Esta interfaz será amigable y con un diseño sencillo e intuitivo, para que el usuario pueda interactuar fácilmente con la misma y mostrará los avisos necesarios para que el usuario conozca el estado del robot en todo momento.
- Permitir que, por medio de esta interfaz, el usuario pueda seleccionar el destino al que desea llegar a través de un plano.
- Permitir que, por medio de la interfaz desarrollada, el usuario pueda seleccionar el destino al que desea llegar por medio de un buscador que accede al directorio de profesores de la universidad.
- Diseñar y elaborar un plano para la interfaz gráfica que resulte sencillo y aclarativo para el usuario.
- Diseñar y elaborar una estructura que represente el plano con sus diferentes despachos, divisiones y conexiones de forma que resulte intuitiva y entendible para futuros usuarios del programa y que resulte sencilla de modificar para futuras ampliaciones del proyecto.

- Permitir que, por medio de la interfaz desarrollada, el usuario pueda recolocar al robot cuando éste se choque o se pierda y colaborar en la localización del robot.
- Diseñar e implementar un sistema de voz que complemente la interfaz gráfica mostrada por pantalla para que el usuario esté informado en todo momento de lo que ocurre.
- Integrar la interfaz desarrollada con el robot Pioneer P3-DX conectando dicha interfaz con el sistema de control del robot, de manera que éste reciba correctamente los mensajes que quiera transmitirle el usuario y envíe la información correspondiente a su estado en cada caso.
- Diseñar y elaborar la estructura que deben seguir los mensajes enviados al sistema de control del robot y recibidos desde el mismo, de forma que resulten sencillos y fácilmente modificables y extensibles en el futuro.

1.2 Fases del desarrollo

El desarrollo de este proyecto ha consistido en la realización de un conjunto de tareas divididas en dos grandes etapas claramente diferenciadas en el proyecto.

Por una parte se ha desarrollado la interfaz necesaria para la comunicación entre humano y robot, que abarca el diseño e implementación de la interfaz gráfica y del sistema de voz, así como la conexión con LDAP, que permitirán al usuario seleccionar el lugar al que desea que el robot se desplace y la petición de ayuda del robot y asistencia por parte del humano en caso de que el robot precise de ella.

Por otro lado se ha realizado el diseño e implementación de todos los elementos necesarios para la conexión de la interfaz previamente desarrollada con el código controlador del robot, así como la adaptación del mismo para que éste reciba y envíe correctamente los mensajes necesarios en cada caso y realice las acciones de desplazamiento oportunas para alcanzar el destino seleccionado por el usuario.

A lo largo del desarrollo de este proyecto han sido empleados un tablet PC, en el que se muestra la interfaz gráfica desarrollada para que el usuario pueda interactuar con el robot, y un robot Pioneer P3-DX.

1.3 Estructura de la memoria

Para facilitar la lectura de la memoria, se incluye a continuación un breve resumen de cada capítulo.

Capítulo 1: Introducción y objetivos

En el capítulo 1 se realiza una introducción del proyecto junto con un análisis de los objetivos a cubrir, las fases de desarrollo realizadas, los medios empleados para llevar a cabo el proyecto y la estructura de la memoria.

En el capítulo 2 se expone el Estado del Arte de este proyecto en el que se realiza una introducción sobre la historia y evolución de los robots móviles, se explica la arquitectura de control robótico usada en este proyecto y se exponen trabajos de robots móviles relacionados con el desarrollado.

En el capítulo 3 se exponen las fases de análisis, diseño y diseño detallado realizadas para el desarrollo y construcción del presente proyecto. En el apartado de Análisis se tratan las funcionalidades y requisitos del sistema, distinguiéndose en este último apartado entre los requisitos de la interfaz gráfica y el sistema de voz y los requisitos de la comunicación de la interfaz desarrollada y el programa controlador del robot. En el apartado de diseño se tratan todos aquellos detalles que han sido necesarios para llevar a cabo el diseño de la arquitectura; de la interfaz gráfica y del sistema de voz, partiendo de las primeras aproximaciones y prototipo inicial y abarcando hasta el diseño del sistema final; los elementos de diseño necesarios para la comunicación entre la interfaz gráfica y el programa controlador del robot y el diseño detallado de todos los elementos anteriores.

En el capítulo 4 se exponen las pruebas que han sido llevadas a cabo para comprobar el correcto funcionamiento del sistema desarrollado, y la evaluación llevada a cabo con usuarios ajenos al proyecto para probar la funcionalidad y usabilidad del mismo, comprobando la facilidad de uso y la utilidad del sistema desarrollado.

En el capítulo 5 se exponen las actividades de gestión del proyecto, incluyendo la planificación de tareas y los costes directos, indirectos y totales del proyecto.

En el capítulo 6 se exponen las conclusiones del proyecto junto con el análisis de cómo se han cubierto los objetivos y sus líneas de desarrollo futuro.

Capítulo 2

Estado del Arte

En este capítulo se exponen los detalles referentes a la base de conocimiento sobre la que se encuentra desarrollado el proyecto. En primer lugar se realiza un breve resumen sobre la historia y avance de los robots móviles, desde la antigüedad hasta nuestros días, junto con la descripción y características del robot empleado. A continuación se describe la arquitectura ROPEM y el entorno sobre el que se encuentra integrado este proyecto. Por último se exponen una serie de trabajos relacionados que pueden servir como referencia y base de comparación con el proyecto desarrollado.

2.1 Robots Móviles

Desde la antigüedad el ser humano ha tratado de desarrollar sistemas y máquinas que imitan el comportamiento de los seres vivos. El término robot, tal y como lo conocemos actualmente, aparece por primera vez en 1921 en la obra teatral Rossum's Universal Robots, escrita por el novelista y dramaturgo Karel Capek, pero los griegos ya comenzaron a desarrollar autómatas que imitaban el movimiento de seres animados. Ya entonces, Arquitas de Tarento, hacia el año 400 a.C, construyó un autómata que consistía en una paloma de madera que intentaba imitar el vuelo de las aves.

La robótica móvil surge por la necesidad de construir sistemas capaces de ser autosuficientes minimizando la intervención humana para su funcionamiento. Uno de los orígenes de la robótica móvil fue la construcción de la tortuga de Walter en 1948, que era capaz de subir pendientes, detectar obstáculos e ir a un determinado punto a recargarse cuando la batería se acabara. En 1953 aparece ELSIE, que era capaz de seguir una fuente de luz y evitar los obstáculos durante su desplazamiento y en 1968 aparece Shakey, mostrado en la Figura 1.a, que cuenta con sensores y medios que le

permiten desplazarse por el suelo y ser controlado remotamente por radio. En los años setenta, la NASA desarrolla sistemas como el Mars-Rover, mostrado en la Figura 1.b, y la agencia espacial rusa consigue un alunizaje por control remoto pero, es en los años ochenta cuando, gracias al desarrollo de elementos como los sensores y sistemas de control, se consigue una mayor autonomía permitiendo que los robots realicen misiones complicadas para el ser humano tales como el rescate del avión de la Air Indian, siniestrado frente a las costas de Irlanda o el descubrimiento de los restos del Titanic.



a. Robot Shakey



b. Robot de exploración en Marte

Figura 1. Robots Móviles

En la actualidad, se han realizado gran cantidad de avances en la Inteligencia Artificial y en el campo de la robótica móvil, creando sistemas que permiten facilitar las tareas a las personas en su vida diaria con una gran autonomía. De esta forma nos encontramos con todo tipo de robots móviles implicados en gran cantidad de sectores entre los que nos podemos encontrar desde robots trepadores, subacuáticos, aéreos y espaciales hasta robots domésticos, de entretenimiento o guías.

En la última década han tomado especial relevancia los denominados robots de servicio, destinados a asistir a la sociedad, entre los que se incluyen los robots domésticos, como el que aparece en la Figura 2.a, de vigilancia, de entretenimiento, sanitarios y guías. En el año 2000 apareció ASIMO, mostrado en la Figura 2.b, un robot humanoide capaz de realizar movimientos complejos similares a los del ser humano y en el año 2002 salió al mercado Roomba, mostrado en la Figura 3, el robot aspiradora, que alcanzó una gran popularidad en muchos hogares.

No obstante, a pesar de estos grandes avances, todavía queda mucho camino por recorrer, ya que aun existen muchas limitaciones que hay que superar. Dentro de la Inteligencia Artificial, ramas como la planificación automática han demostrado ser útiles para lograr una mayor autonomía. Sin embargo, muchos robots, a pesar de contar con mayor maniobrabilidad al ser móviles, no son completamente autónomos y siguen

necesitando ayuda humana. Este campo también está teniendo numerosas investigaciones hoy en día, ya que con esta interacción entre hombre y robot, ambos pueden gozar de una ayuda mutua, facilitando así el robot las tareas al humano y el humano atendiendo al robot en situaciones en las que éste no pueda valerse por sí mismo.



a. Robot Doméstico



b. Robot ASIMO

Figura 2. Robots de Servicio



Figura 3. Roomba, robot aspiradora

Para este proyecto se ha empleado un robot Pioneer P3-DX, mostrado en la Figura 4. Se trata de un robot móvil; con cuerpo de aluminio de 44cmx38cmx22cm; con dos ruedas de 16,5cm de diámetro, cada una con un motor independiente; con ocho sensores ultrasónicos que dan una cobertura de 180 grados; y con parachoques delanteros y traseros, con sensores de contacto. Este robot sube pendientes de un 25% de inclinación, escalones de 2,5cm, carga hasta un peso de 23kg y puede alcanzar 18-24 horas de autonomía. No obstante, como ocurre con muchos otros robots, tiene una serie de limitaciones, ya que las ruedas se encuentran algo desviadas y acumula error en los

giros y desplazamientos, por lo que en determinadas ocasiones se pierde o se choca. Es en esos momentos en los que el robot precisa de la ayuda del humano.



Figura 4. Robot Pioneer P3-DX

2.2 Arquitectura ROPEM

Previamente a la explicación de los detalles de la interfaz gráfica construida, se expone el entorno donde se integrará dicha interfaz. En este apartado se introduce el entorno de trabajo para situar al lector, pero en la sección 3.3.1 se describen todos los detalles de su diseño. En dicho entorno, se distinguen dos bloques principales de implementación, consistentes en una capa de bajo nivel, que se encuentra comunicada directamente con el robot, y una capa de alto nivel que conecta el bajo nivel con la interfaz gráfica desarrollada.

La capa de bajo nivel se encarga de recibir peticiones de acción y de enviar las órdenes necesarias a los actuadores del robot. En esta capa se encuentra la plataforma de control robótico Player/Stage. Player es un servidor que se emplea para abstraer los detalles del hardware del robot y acceder de forma más sencilla a los sensores y a los actuadores del mismo. Player/Stage es una plataforma gratuita y de código abierto que además soporta varios lenguajes de programación. Anterior a este tipo de plataformas, se empleaban directamente los drivers para acceder a los dispositivos del robot, con lo que la tarea resultaba mucho más compleja. Tras la aparición de Player/Stage y de otras plataformas similares, el acceso a los sensores y a los actuadores del robot resultó mucho más sencillo, facilitando el control del robot.

Player actúa como un servidor para el control del robot. El programa cliente se conecta con Player mediante conexión por sockets TCP. Player envía a los actuadores la acción a ejecutar y recibe de los sensores la información que precisa.

A alto nivel, entre la plataforma de control robótico Player y la interfaz gráfica desarrollada, se encuentra ROPEM (*Robot Plan Execution with Monitoring*), la arquitectura de control robótico usada en este proyecto y basada en planificación automática. La arquitectura ROPEM recibe un PDDL (*Planning Domain Definition Language*), que es un lenguaje estándar de codificación para la planificación y que

consta de una serie de predicados que indican el estado inicial del problema, con las acciones que es posible llevar a cabo y con el destino final que queremos alcanzar. Esta capa de alto nivel emplea técnicas de planificación automática y, es este planificador, el que emplea el PDDL con la descripción del problema para hallar una solución a alto nivel. Tras la planificación, ejecuta las acciones de alto nivel que se hayan generado y comprueba el estado que ha resultado tras su realización y si los objetivos se han cumplido. El sistema permite la replanificación, por lo que si la primera solución hallada ha fallado, es posible volver a realizar una nueva planificación partiendo del nuevo estado al que hayamos llegado tras la ejecución de las acciones. Esta capa de alto nivel se comunica con la de bajo nivel solicitándole que ejecute las acciones de bajo nivel que se encuentran asociadas a una acción de alto nivel. En [5] se puede encontrar una explicación más detallada de la arquitectura ROPEM.

El entorno de desarrollo de este proyecto puede observarse en la Figura 5.

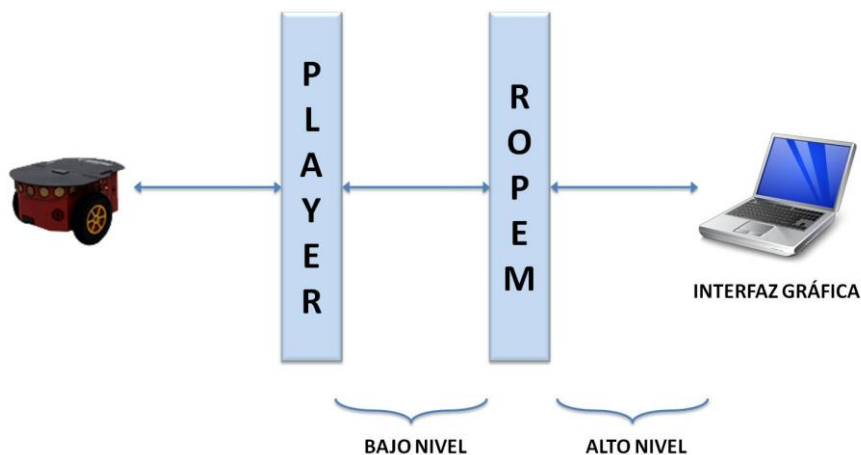


Figura 5. Descripción del Entorno de la Aplicación

El proyecto desarrollado ha consistido en la construcción completa de la interfaz gráfica y del sistema de voz y de la conexión de la misma con la arquitectura ROPEM. A pesar de que parte del código de dicha arquitectura ROPEM ya se encontraba realizado previamente al proyecto, ha sido necesario modificarlo y adaptarlo para que su funcionamiento en conjunto con la interfaz gráfica fuera el correcto.

2.3 Trabajos Relacionados

Desde la construcción de los primeros robots se ha mejorado la ejecución de determinadas actividades que suponían un gran esfuerzo para las personas. Desde entonces han surgido gran cantidad de sistemas que abarcan múltiples campos. Podemos encontrar desde robots industriales a robots móviles empleados en actividades submarinas y aéreas o espaciales, llegando incluso a los robots de servicio. Este último

tipo de sistemas ha tenido un gran avance en los últimos años, existiendo robots domésticos que realizan tareas del hogar, tareas de vigilancia o tareas de entretenimiento y educación. También han sido desarrollados robots para dar asistencia a personas discapacitadas o ancianas, robots utilizados en la sanidad o incluso robots que sirven como guías en museos, bibliotecas y todo tipo de edificios.

Existen gran cantidad de trabajos de investigación sobre la interacción entre humanos y robots móviles. En [2] se muestra la aparición de robots-guía en España, concretamente en la Ciudad Banco Santander. Estos robots pueden observarse en la Figura 6. El complejo de Ciudad Banco Santander es un lugar muy extenso, que recibe un gran número de visitantes a diario y en el que es muy fácil perderse. Debido a esta situación se ha procedido a la instalación de estos robots que son lo suficientemente inteligentes como para evitar a otros robots, personas u otros obstáculos y no chocar.



Figura 6. Robots de la Ciudad Banco Santander

En [3] se expone la aparición de “Bellbot” y “Sacarino” (este último mostrado en la Figura 7), dos robots móviles empleados en un hotel de Valladolid. Los robots desarrollan la función de botones del hotel y llevan a cabo tareas como el acompañamiento de los huéspedes a las habitaciones, el transporte de equipajes de mano o hacerse cargo del servicio de habitaciones. Los robots son capaces de desplazarse por las diferentes zonas del edificio, sortear obstáculos, emplear los ascensores y volver al punto donde se recargan sus baterías. El control de los robots puede realizarse o bien desde la recepción del hotel o desde una página web a la que se puede acceder desde la recepción o desde las habitaciones mediante la cual se les puede solicitar cualquier servicio que normalmente era llevado a cabo llamando a recepción.

En [4] se expone la construcción de REEM, mostrado en la Figura 8, un robot destinado a ofrecer servicio en lugares públicos como centros comerciales, aeropuertos, hospitales o museos. REEM cuenta con una pantalla táctil que sirve de comunicación con los usuarios y, al igual que en el presente proyecto, ofrece un mapa del edificio y da la posibilidad de guiar al usuario hasta el lugar que le sea indicado. El robot cuenta también con un sistema de voz que le permite la comunicación con el usuario en varios idiomas y un sistema de reconocimiento de rasgos faciales que le permite diferenciar a los usuarios. Esta sería una perfecta plataforma para comparar el resultado de este PFC.



Figura 7. Sacarino, robot botones



Figura 8. REEM, robot guía

En [6] se realizaron unas pruebas realizadas sobre Cobot, mostrado en la Figura 9, un robot móvil que ayuda a los humanos a moverse por un edificio y los guía por las distintas reuniones que tienen durante el día. Al igual que en este proyecto, se da una gran interacción entre el humano y el robot, ya que éste tiene una serie de limitaciones que han de ser superadas con la ayuda del usuario. El robot no puede abrir puertas, ya que no dispone de manos, y en algunos momentos se desubica, por lo que precisa de la asistencia del humano. Al igual que en este proyecto, el robot decide las acciones a realizar mediante la planificación automática, si bien se diferencia en que utiliza un sistema de localización basado en Wifi. Se realizaron pruebas con cinco usuarios distintos y se obtuvieron muy buenos resultados, ya que a cuatro de ellos les resultó completamente satisfactorio y volverían a utilizar a Cobot de nuevo. El usuario restante dijo que lo usaría de nuevo si se consiguiera disminuir el número de preguntas que el robot le realizó en los distintos momentos del experimento.



Figura 9. COBOT, robot guía

En [7] se expone un trabajo de investigación en el que se construyó un robot de asistencia para facilitar la movilidad de personas con problemas motrices. SENA, mostrado en la Figura 10, es una silla de ruedas robotizada que lleva al usuario a los diversos lugares que éste le solicite. Por medio de una serie de sensores y otros dispositivos controlados por un ordenador portátil, SENA es capaz de trasladar al humano a través del medio de forma automática, sin el control manual por parte del usuario. Al igual que en este proyecto, SENA tiene una serie de limitaciones ya que, por ejemplo, no puede abrir puertas ni llamar a un ascensor, por lo que dispone de una interfaz amigable a través de la cual se comunica con el humano para que éste la asista en los momentos en los que el robot no puede valerse por sí mismo. Se realiza una ayuda mutua, ya que el robot es capaz de facilitar la movilidad a la persona con dificultades motrices y esta última colabora con el robot en aquellas situaciones en las que él no puede ejecutar distintas acciones que sí pueden ser llevadas a cabo por el usuario.



Figura 10. SENA, silla robótica

En [8] se simula cómo un robot móvil Pioneer P3-DX puede ayudar a desplazarse a personas invidentes por un entorno interior como puede ser el de un hotel. A diferencia del proyecto presentado, no se empleó planificación automática, sino que, por medio de un grafo formado por un conjunto de nodos que contienen los diferentes lugares del entorno, se decide el camino a seguir por el robot por medio del algoritmo de Dijkstra.

En [9] se expone la construcción de un robot móvil que sirve como guía en un museo. La interacción con el usuario en este caso toma especial relevancia, ya que el robot no sólo guía a los usuarios y explica las diferentes salas, sino que debe ser capaz de responder correctamente a las preguntas que éstos le formulen. A diferencia del proyecto presentado, se empleó una técnica conocida como SLAM (Localización y Mapeo Simultáneos), que permite construir de forma incremental un mapa del entorno por medio de información sensorial y, utilizar el mapa creado para localización y navegación del robot.

En [10] se expone un trabajo de investigación en el que se realiza la construcción de Wheelesley, un sistema robótico para sillas de ruedas. En él se desarrolla una interfaz gráfica que permite al usuario comunicarse con la silla de ruedas. La sencillez de esta interfaz toma especial relevancia, ya que es probable que los usuarios que hagan uso de este sistema tengan dificultades para su manejo, por lo que incluso es posible personalizar esta interfaz según el usuario que la emplee. Al igual que en el proyecto desarrollado, el usuario emplea la interfaz gráfica para realizar órdenes de alto nivel como “Avanza” o “Retrocede” que son transmitidas al robot, el cual realiza la acción ayudándose de sus sensores para realizar el movimiento con seguridad. A diferencia del presente proyecto, la interfaz gráfica desarrollada no incluye mapas ni ningún sistema para comunicar el lugar de destino al robot, sino que consiste tan sólo en una serie de flechas y botones que permiten el avance, retroceso, giro a derecha e izquierda, parada y control de la velocidad de la silla, similar a la pantalla de movimiento para recolocar al robot desarrollada en este proyecto en caso de que éste encontrara un obstáculo o chocara contra algo que le impidiera continuar su desplazamiento.

En [11] se expone un trabajo de investigación sobre un robot guía en un museo. Al igual que en el proyecto desarrollado se trata de un robot guía en interiores que además es muy similar al empleado en este caso. De igual forma que en el presente proyecto, se usa un mapa local del edificio, que establece objetivos para alcanzar el destino. También utiliza odometría para calcular la posición del robot, aunque además utiliza otras técnicas que lo hacen más complejo y que le permiten no requerir de la ayuda del humano para su localización.

Capítulo 3

Interfaz Hombre-Robot para un Robot Guía en Interiores

El objetivo principal de este proyecto es el desarrollo de una interfaz hombre-robot táctil que permita la interacción de un usuario con un robot que le guiará en el interior de un edificio. La interfaz creada permite que el humano pueda indicar al robot a qué lugar desea que le acompañe dentro de la planta primera del edificio Sabatini de la Universidad Carlos III de Madrid y, que a su vez, el robot pueda solicitarle ayuda en caso de encontrarse con algún problema durante su desplazamiento.

La interfaz gráfica desarrollada se basa en un diseño sencillo, intuitivo y amigable. A través de ella, el usuario podrá indicar el destino al que desea llegar bien a través de un plano o a través de la introducción de los datos del profesor al que desea visitar. Esta última consulta se realiza por medio de la conexión con el directorio de la universidad donde se encuentran los datos del personal que trabaja en la misma. Todos los planos mostrados por pantalla son simples y fáciles de comprender e interpretar, lo que permite que el usuario conozca en todo momento su situación.

Durante el proyecto se ha creado una estructura que permite representar el plano para que sea entendible por el robot y que, debido a su sencillez, es fácilmente reutilizable, comprensible, modificable y extensible por parte de futuros usuarios.

Durante el desplazamiento del robot, el programa permite que el usuario le reubique en el mapa en caso de no encontrarse en el lugar indicado por pantalla. En caso de error, el robot solicitará asistencia al usuario y éste podrá recolocarle en el pasillo para que continúe su desplazamiento.

La interfaz gráfica desarrollada se encuentra complementada por un sistema de voz que permite al robot comunicar todos los mensajes necesarios al usuario en cada momento.

La interfaz creada se ha incluido en un tablet PC conectado con un robot Pioneer P3-DX. Los mensajes enviados por el programa controlador del robot y recibidos desde el mismo siguen una estructura sencilla de entender y fácilmente modificable por futuros usuarios.

En este capítulo se expondrán las fases de Análisis, Diseño y Diseño Detallado que han sido necesarias para el desarrollo de este proyecto.

3.1 Análisis

Para la realización de este proyecto se llevó a cabo un análisis exhaustivo de los diferentes requisitos que éste ha de tener y de las distintas funcionalidades que el proyecto abarcará. En esta sección se expone el diagrama de casos de uso que muestra las distintas funcionalidades del proyecto y, en segundo lugar, los requisitos del sistema, distinguiendo entre los requisitos de la interfaz gráfica y sistema de voz y los requisitos existentes en la comunicación de la interfaz desarrollada y el programa controlador del robot.

3.1.1 Diagrama de Casos de Uso

En el sistema se distinguen seis funcionalidades principales, cuatro de las cuales pueden ser realizadas por medio del usuario y dos por medio del robot.

La funcionalidad principal que puede llevar a cabo el usuario es **solicitar ayuda para desplazarse a un destino** determinado, dándose la posibilidad de seleccionar el destino **por medio del plano** o **por medio del directorio de la universidad**. Para ello el usuario debe seleccionar una de las dos opciones que le llevará a realizar cada una de las funciones correspondientes. A ambas peticiones el robot responde con una nueva funcionalidad, navegar hacia el destino seleccionado.

La funcionalidad **navegar** permite que la aplicación envíe al robot una serie de mensajes con las acciones que debe realizar para llegar al destino seleccionado previamente por el usuario. Tras la recepción de una acción, el robot pasa a ejecutarla pudiendo ocurrir dos situaciones: que el robot no tenga ningún problema y ejecute completamente la acción o que el robot se encuentre con un obstáculo, se choque o se pierda, en cuyo caso aparece una nueva funcionalidad que es la de **solicitar ayuda**.

En estas ocasiones en las que el robot se choca, se encuentra algún obstáculo que le impide su desplazamiento o se pierde, solicita ayuda al usuario. Por su parte, el humano puede **orientar al robot** por medio de su interacción con la interfaz, lo que implica una

nueva funcionalidad, y de esta forma ponerle correctamente ubicado en el pasillo y orientado hacia la dirección de destino hacia donde va a desplazarse.

En el caso de que el robot crea encontrarse en una posición distinta a la que se encuentra en realidad, el usuario podrá **corregir la localización del robot**. Por medio de esta funcionalidad el usuario podrá indicar al robot, por medio del plano, el lugar en el que se encuentra en realidad.

Todas estas funcionalidades, que pueden observarse en la Figura 11, tienen su descripción textual correspondiente en el Anexo A.

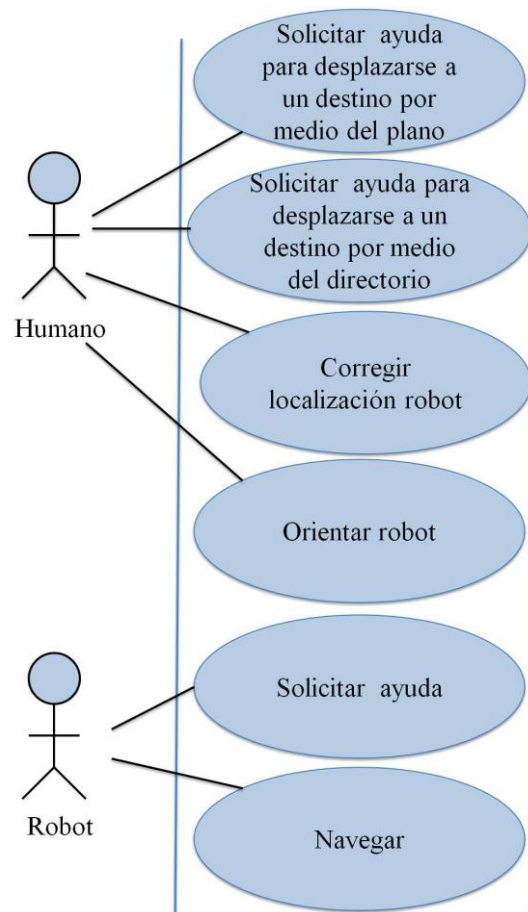


Figura 11. Diagrama de Casos de Uso

3.1.2 Requisitos del Sistema

En esta sección se exponen aquellos requisitos que debe cumplir el sistema desarrollado. Para ello se distinguen dos secciones: una primera en la que aparecen los requisitos que ha de seguir la interfaz gráfica desarrollada y el sistema de voz y una segunda que contiene los requisitos pertenecientes a la comunicación de dicha interfaz con la arquitectura ROPEM.

3.1.2.1 Interfaz Gráfica y Sistema de Voz

Para la consecución y realización de las funcionalidades planteadas se ha desarrollado una interfaz gráfica para la interacción del humano con el robot y un sistema de voz para que este último pueda informar al usuario de todo lo que ocurre en cada momento.

Como se ha comentado anteriormente, la selección del destino se podrá realizar o bien por medio de un plano o por medio de una búsqueda en el directorio de profesores de la Universidad.

Para el caso de la búsqueda a través del plano es necesario que éste sea fácil de entender y de interpretar. Debe ser sencillo, sin elementos que recarguen o compliquen su empleo.

En el caso de la búsqueda a través del directorio, el texto que se muestra debe ser claro y los mensajes fácilmente entendibles. Es necesaria la presencia de un teclado táctil en pantalla para que sea posible la introducción de los datos del profesor por medio del usuario. En caso en que la búsqueda no arroje resultados, se le comunicará al usuario este hecho para que pueda introducir nuevamente los términos de búsqueda o seleccione el destino al que desea dirigirse por medio del plano.

Una vez que el robot comience a desplazarse será necesario mostrar su situación en el plano por pantalla para que el usuario conozca su localización en todo momento y para que, en caso de que el robot crea que se encuentra en un lugar distinto al que realmente está localizado, pueda reubicarlo en el plano y comunicarle su verdadera posición.

El sistema también debe permitir que el usuario pare el movimiento del robot en cualquier momento pudiendo indicarle un nuevo lugar de destino.

En cuanto a los mensajes de aviso, estos deben ser comunicados al usuario a través de un cuadro de errores ubicado en pantalla y por medio de un sistema de voz que se activará en todos los momentos en que ocurra un evento que sea necesario comunicar al usuario.

3.1.2.2 Comunicación Interfaz y Arquitectura ROPEM

El sistema de control del robot y la interfaz deben estar comunicados para intercambiarse los distintos mensajes que sea necesario que el robot comunique al

humano por pantalla y la posición en que se encuentre en cada momento. Del mismo modo, la interfaz debe poder comunicar a la arquitectura de control del robot el lugar de destino seleccionado por el usuario, el lugar en el que el robot se encuentra en caso de que desconozca su posición inicialmente o en el caso de que se pierda, así como los distintos movimientos que el humano desea que el robot realice para ubicarlo centrado en el pasillo y orientado en la dirección de destino.

Para el intercambio de todos estos mensajes, se precisa de un sistema que permita la comunicación de la interfaz gráfica con el programa de control del robot y la comunicación de éste último con la interfaz desarrollada. Al tratarse de programas distintos y escritos en lenguajes de programación distintos, el canal de comunicación debe ser independiente de la plataforma y del lenguaje.

Del mismo modo, se precisa de la creación de una estructura de datos que recoja el mensaje a intercambiar en cada momento. Esta estructura debe ser fácilmente entendible, modificable y válida para todas las plataformas y lenguajes ya que los mensajes serán intercambiados entre programas distintos.

Para que el robot pueda desplazarse y alcanzar el objetivo seleccionado es necesario que éste conozca el estado de la situación en que se encuentra para que, de esta forma, pueda hallar una solución al problema. Para ello, es necesario definir el escenario en el que el robot se va a desplazar, el estado inicial en el que se encuentra, el objetivo al que debe desplazarse y las acciones que puede llevar a cabo para conseguirlo. Debido a la variabilidad de las condiciones del entorno esta descripción del escenario deberá crearse de forma automática cada vez que cambie alguno de los elementos de dicho escenario, tales como el lugar de origen o el objetivo a ser alcanzado. Este escenario se creará automáticamente a partir de una representación del plano en la que aparezcan sus distintos despachos y áreas en las que se divide junto con sus distintas conexiones. Esta representación debe ser sencilla y fácil de manejar para que otros usuarios puedan entenderla y modificarla para futuras ampliaciones o mejoras del presente proyecto. Inicialmente se pensó en dividir la imagen del plano en sub-imágenes que se correspondieran con cada una de las áreas que diferenciaría el robot para desplazarse, pero esta opción fue descartada ya que resultaba demasiado compleja y hacía difícil cualquier modificación o ampliación que quisiera llevarse a cabo. Por ello se optó por crear una estructura que fuera fácilmente entendible y que resultara sencilla de modificar.

3.2 Diseño

A continuación pasaremos a exponer los detalles de diseño realizados durante el proyecto. En la subsección 3.2.1 se describe el diseño de la arquitectura global del sistema, en la subsección 3.2.2 se detalla el diseño realizado de la interfaz gráfica y del sistema de voz, describiendo la evolución de dicha interfaz, distinguiendo las primeras aproximaciones realizadas del desarrollo de la interfaz definitiva, y el diseño de los elementos que aparecen en la misma y en la subsección 3.2.3 se describe el diseño realizado para comunicar la interfaz desarrollada con la arquitectura ROPEM.

3.2.1 Diseño de la Arquitectura

A continuación se describe el diseño de la arquitectura del sistema desarrollado, que puede observarse en la Figura 12. Dicha arquitectura consta de módulo ILHR (Interfaz Localización Hombre-Robot) que está compuesto por tres submódulos: la interfaz gráfica, con la que el usuario interactuará para indicar el destino del desplazamiento; el sistema de voz, empleado para avisar al usuario de los mensajes que el robot deba comunicar en cada momento de la navegación; y los mecanismos encargados de realizar la comunicación de la interfaz gráfica desarrollada con la arquitectura ROPEM. El módulo ILHR se conecta con LDAP, mecanismo que permite acceder a los datos almacenados en el directorio de la universidad. De esta forma, la interfaz realiza una consulta del profesor al que se desea visitar y LDAP devuelve el despacho asociado con dicho profesor. Por otra parte, el módulo ILHR se encuentra conectado con la arquitectura ROPEM por medio de un archivo PDDL, que consiste en un fichero con la información del problema a resolver y que es explicado con más detalle en la sección 3.2.3. Este archivo es generado por la interfaz a partir del mapa del edificio y de la información del destino y origen de la navegación, y contiene una descripción del entorno donde se desarrollará la acción junto con el estado inicial y objetivo a ser alcanzado. Dicho fichero PDDL, junto con el dominio procedente del exterior, son empleados por Sayphi, planificador de la arquitectura ROPEM, que decide qué acciones llevar a cabo para alcanzar el objetivo determinado en el problema a partir del estado inicial. Dichas acciones son descompuestas en acciones de bajo nivel cuya orden de ejecución es enviada al robot por medio de la plataforma de control Player. El módulo de monitorización es el encargado de coger la información de bajo nivel de los sensores del robot y transformarla a información de alto nivel, comprobando además la correcta realización de las acciones indicadas por el planificador.

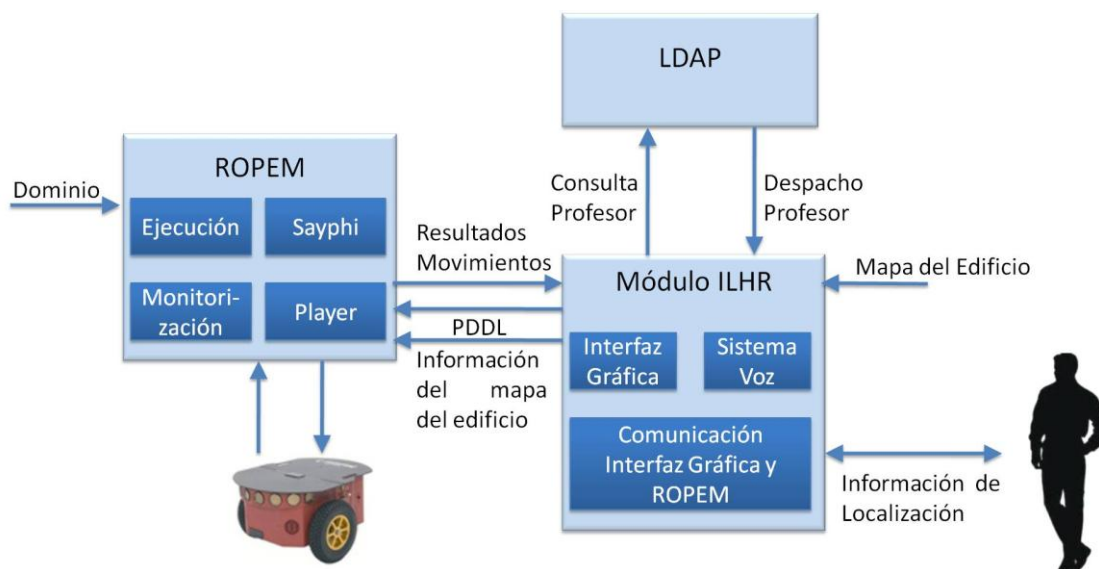


Figura 12. Diseño de la Arquitectura Desarrollada

3.2.2 Diseño de la Interfaz Gráfica y Sistema de Voz

Previamente al diseño de la interfaz definitiva, se realizó una aproximación sencilla para comprobar el funcionamiento de las interfaces de java con la pantalla táctil del ordenador. A continuación se describe el diseño preliminar realizado y posteriormente el diseño final de la interfaz desarrollada.

3.2.2.1 Diseño Preliminar

En un primer lugar, se realizó una aplicación, mostrada en la Figura 13, consistente en un botón de avance y otro de parada del robot que fuera capaz de capturar los eventos de pulsación que se produjeran sobre ellos para mandar esa información al programa controlador del robot.



Figura 13. Primera Prueba de Interfaz Gráfica

Una vez probado el funcionamiento de la interfaz desarrollada en la pantalla táctil se procedió a diseñar una primera aproximación de lo que se deseaba que fuera la interfaz definitiva de la aplicación. En este primer prototipo sólo existía la posibilidad de buscar un destino por medio del plano, ya que el que se pudiera realizar dicha búsqueda por medio del directorio de la universidad se decidió con posterioridad. Inicialmente, el usuario se encontraría con una pantalla de bienvenida, mostrada en la Figura 14, que mostraría un plano de la universidad en el que el usuario tendría que seleccionar el lugar al que desea dirigirse.



Figura 14. Pantalla de Inicio del Prototipo de la Interfaz

Posteriormente, cuando el robot comenzara su desplazamiento, se mostraría una segunda pantalla, que puede observarse en la Figura 15, en la que aparecería el plano junto con los botones de “STOP”, “Nuevo Destino” y “Volver a Casa”. En el diseño final, estos botones fueron modificados, conservándose un botón de STOP que incluye internamente la posibilidad de seleccionar un nuevo destino y habiéndose eliminado la opción de “Volver a Casa” ya que se consideró que sería preferible que el robot no tuviera un lugar fijo de origen, sino que la primera vez que se utilizara el sistema se solicitara dicha posición al usuario.



Figura 15. Pantalla de Desplazamiento hacia Destino del Prototipo de la Interfaz

En caso de que se produjera un error durante el desplazamiento debido a que el robot se chocara o encontrara un obstáculo que le impidiera continuar con su movimiento, se mostraría un botón de “Ayuda” (Figura 16.a) que tras su pulsación llevaría a una pantalla (Figura 16.b) en la que, por medio de una serie de flechas, se podría recolocar al robot en el pasillo. En este primer prototipo existía un botón de STOP, ya que la pulsación de las flechas ocasionaba el movimiento asociado a la misma y que había que finalizar con la pulsación de dicho botón de STOP. Este aspecto fue modificado en el diseño final ya que se consideró que era preferible que el robot realizara pequeños movimientos con la pulsación de cada una de las flechas para que no se volviera a chocar durante su recolocación, por lo que el botón STOP fue eliminado en esta pantalla y la pulsación de cada una de las flechas ocasiona un pequeño movimiento en lugar de uno ininterrumpido.



a. Pantalla de Solicitud de Ayuda

b. Pantalla de Recolocación del Robot

Figura 16. Pantallas de Ayuda al Robot del Prototipo de la Interfaz

3.2.2.2 Diseño Final

A continuación se detalla el diseño de la interfaz definitiva desarrollada a partir del prototipo anterior con una serie de modificaciones.

Inicialmente, el usuario se encontrará con una pantalla de bienvenida, mostrada en la Figura 17, en la que se le da la posibilidad de elegir el destino al que desea que el robot le guíe, bien por medio de un plano de la universidad o introduciendo los datos del profesor al que desea visitar.

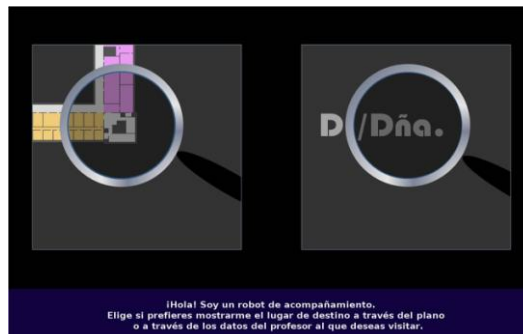


Figura 17. Pantalla de Inicio de la Interfaz Desarrollada

En caso de que el usuario haya seleccionado la opción de búsqueda del destino a través del plano le aparecerá una pantalla, como la mostrada en la Figura 18.a, con un mapa del edificio en la que deberá seleccionar la zona a la que desea dirigirse. Una vez seleccionada dicha zona, será mostrada por pantalla ampliada, como se muestra en la Figura 18.b, y el usuario deberá seleccionar el lugar de destino al que desea que le acompañe el robot.

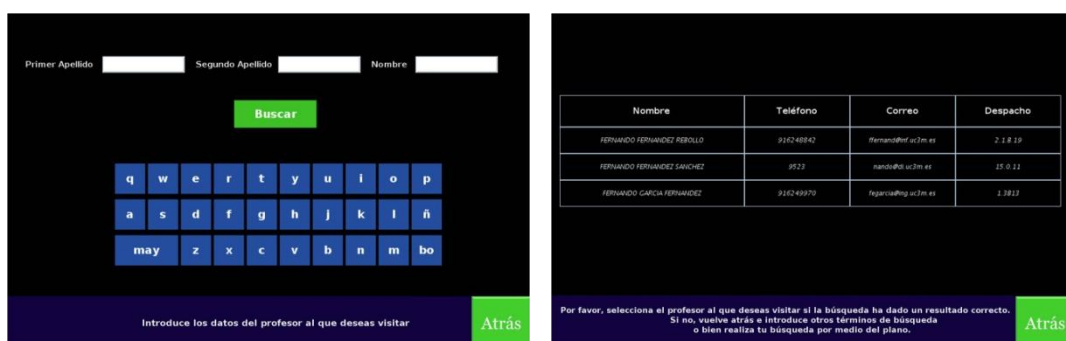


a. Plano General

b. Plano de Ampliación de Zona C

Figura 18. Pantallas de Búsqueda de Destino de la Interfaz Desarrollada

Si por el contrario, el usuario ha elegido la opción de buscar el lugar de destino a través de los datos del profesor al que desea visitar, se mostrará un buscador, como el mostrado en la Figura 19.a, en el que será posible introducir el nombre y los apellidos del profesor por medio de un teclado táctil que aparecerá por pantalla. Una vez que los datos hayan sido introducidos y el usuario haya pulsado en “Buscar”, se le informará de los resultados obtenidos, como puede observarse en la Figura 19.b. En caso de que la búsqueda haya dado resultados, se le mostrarán al usuario por pantalla para que él pueda seleccionar el profesor correcto.



a. Búsqueda Directorio

b. Resultados Búsqueda

Figura 19. Pantallas de Búsqueda en el Directorio de la Interfaz Desarrollada

Tras haber seleccionado el destino por medio de cualquiera de las dos opciones mostradas anteriormente, se solicitará al usuario el lugar donde se encuentra el robot en el caso de que éste lo desconozca (esta situación tendrá lugar cada vez que se inicie la aplicación). Para obtener el lugar de origen se mostrará al usuario un mapa del edificio para que seleccione la zona donde se encuentra localizado el robot y a continuación deberá seleccionar el lugar dentro de la zona escogida como se muestra en la Figura 20.



a. Plano General

b. Plano de Ampliación Zona C

Figura 20. Pantallas de Búsqueda del Origen de la Interfaz Desarrollada

Una vez que el robot conozca su ubicación y el lugar al que debe desplazarse, aparecerá por pantalla el mapa del edificio con una imagen del robot que irá mostrando al usuario dónde se encuentra situado en cada momento. Esta pantalla puede observarse en la Figura 21. Adicionalmente se incluyó que, aunque el usuario haya seleccionado la búsqueda por medio del plano, se realiza una búsqueda en el directorio para comprobar si el lugar al que se desplaza el robot corresponde con algún profesor y, en caso de ser así, se muestra por pantalla.

En caso de que el usuario observe que el robot está equivocado con respecto al lugar donde se encuentra, podrá reubicarlo en el mapa, en cuyo caso aparecerá por pantalla el plano para seleccionar la zona y, posteriormente, el lugar dentro de esa zona donde se encuentra el robot.

El usuario también podrá parar el robot pulsando el botón “Stop”, en cuyo caso el robot se detendrá y se mostrará la pantalla de Inicio.

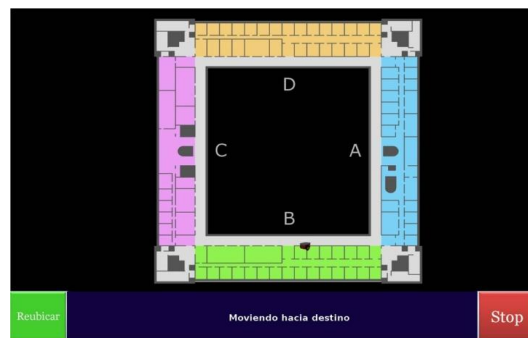
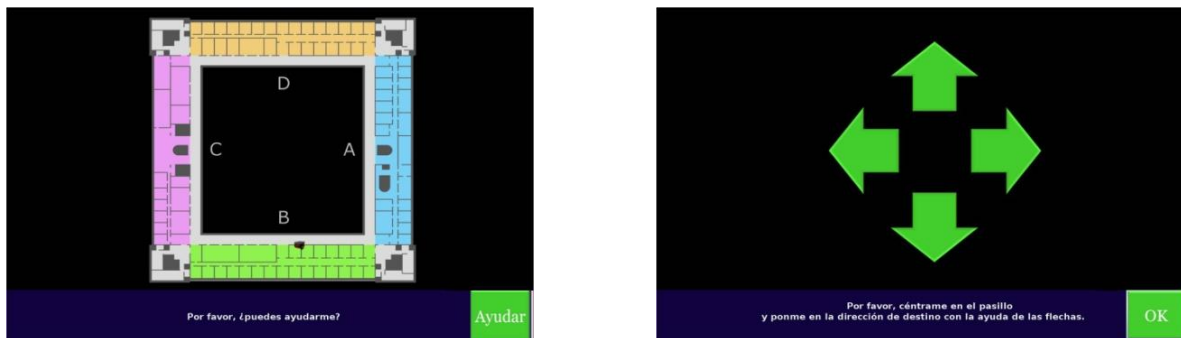


Figura 21. Pantalla de Desplazamiento del Robot de la Interfaz Desarrollada

En caso de que se produzca un error debido a que el robot se choque o detecte un obstáculo que le impida continuar su desplazamiento, se pedirá ayuda al usuario (Figura 22.a). Una vez que la persona haya pulsado en el botón “Ayudar” aparecerán por pantalla una serie de flechas (Figura 22.b) que permitirán colocar al robot en la posición correcta para que continúe con su desplazamiento. Estas flechas harán que el robot realice un pequeño movimiento cada vez que sean pulsadas de tal forma que, para que el

usuario coloque al robot en la posición correcta, tenga que pulsar varias veces sobre las flechas realizando varios pequeños movimientos. Una vez que el robot se encuentre en el lugar adecuado, el usuario pulsará en el botón “OK” y se le solicitará que compruebe si el robot se encuentra verdaderamente en el lugar indicado en el plano (Figura 23). En caso de no encontrarse en el lugar mostrado por el plano, el usuario deberá pulsar en la zona y a continuación en el lugar de la zona donde se encuentre ubicado. Una vez que el usuario haya confirmado la ubicación actual del robot, éste seguirá su desplazamiento.



a. Pantalla Solicitud Ayuda

b. Pantalla Recolocación

Figura 22. Pantallas de Ayuda al Robot de la Interfaz Desarrollada

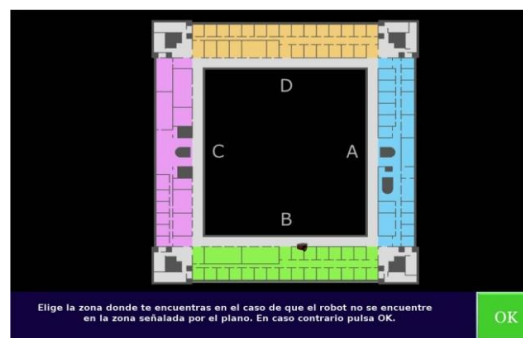


Figura 23. Pantalla de Confirmación de Origen

A continuación, en la Figura 24, se muestra un diagrama que ilustra cómo se realiza la navegación entre las distintas pantallas durante el empleo de la interfaz gráfica desarrollada.

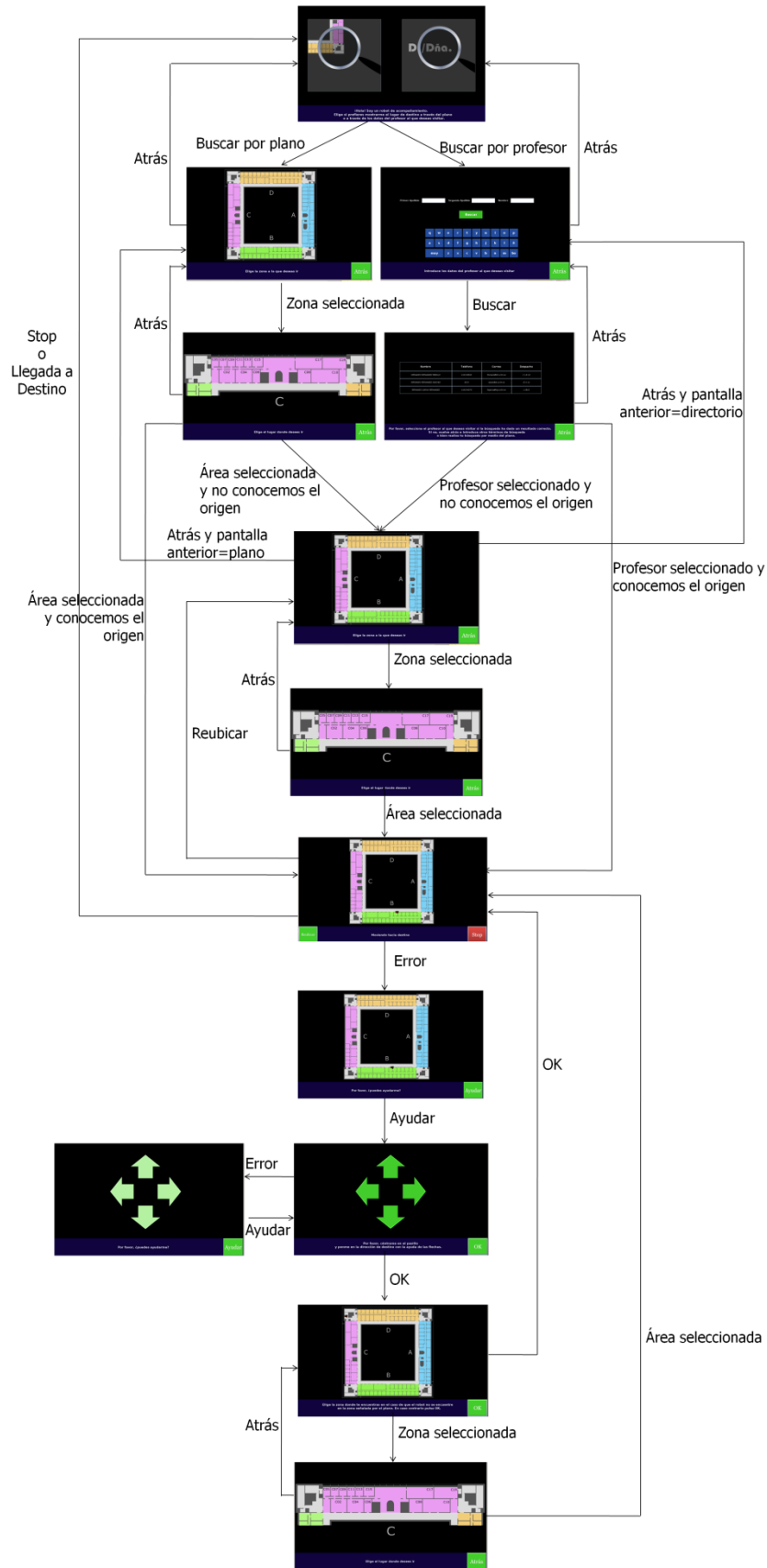


Figura 24. Navegación de Pantallas

Con respecto al diseño del plano mostrado por pantalla se decidió partir de un plano técnico de la Universidad y a partir de él realizar una serie de modificaciones para que resultara más fácil de interpretar. Como puede observarse en la Figura 25, el plano original tiene gran cantidad de detalles que hacen difícil su lectura. Además, las zonas no se encuentran señalizadas por su letra correspondiente. Por ello se decidió eliminar todos aquellos detalles que complicaran el plano, realizando una distinción de las zonas por colores a la vez que se indica su letra correspondiente. Los números de despacho son eliminados en el plano general, conservándose en los planos correspondientes a las zonas ampliadas y las líneas que delimitan el plano y los despachos se simplifican, a la vez que se modifica la representación de puertas, ascensores y escaleras haciéndola más sencilla. El diseño final del plano general y de una de las zonas ampliadas puede observarse en la Figura 26 y en la Figura 27.



Figura 25. Plano Técnico de la Primera Planta del Edificio Sabatini de la Universidad Carlos III de Madrid

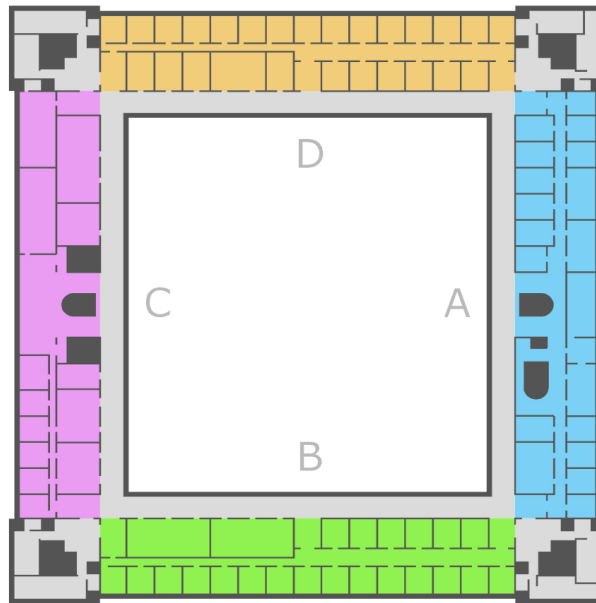


Figura 26. Plano Simplificado de la Primera Planta del Edificio Sabatini de la Universidad Carlos III de Madrid

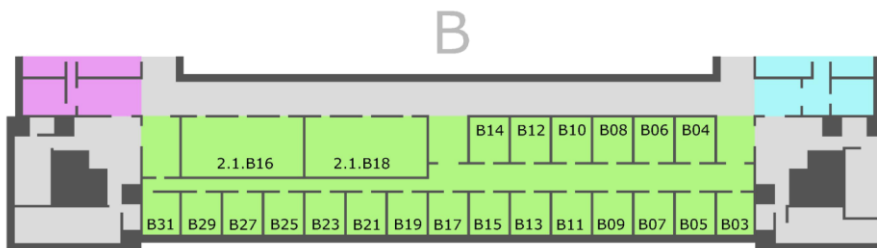


Figura 27. Ampliación del Plano de la Zona B del Edificio Sabatini de la Universidad Carlos III de Madrid

Para el desarrollo del programa de la interfaz ha sido necesaria la creación de diferentes clases, comunicadas entre sí de tal forma que posibilitara el desarrollo del programa y facilitara la utilización y ampliación del mismo por parte de futuros programadores. En la Figura 28 se muestra un diagrama de clases simplificado que muestra los distintos componentes del programa y su comunicación. Un diagrama de clases más detallado puede encontrarse en el Anexo B

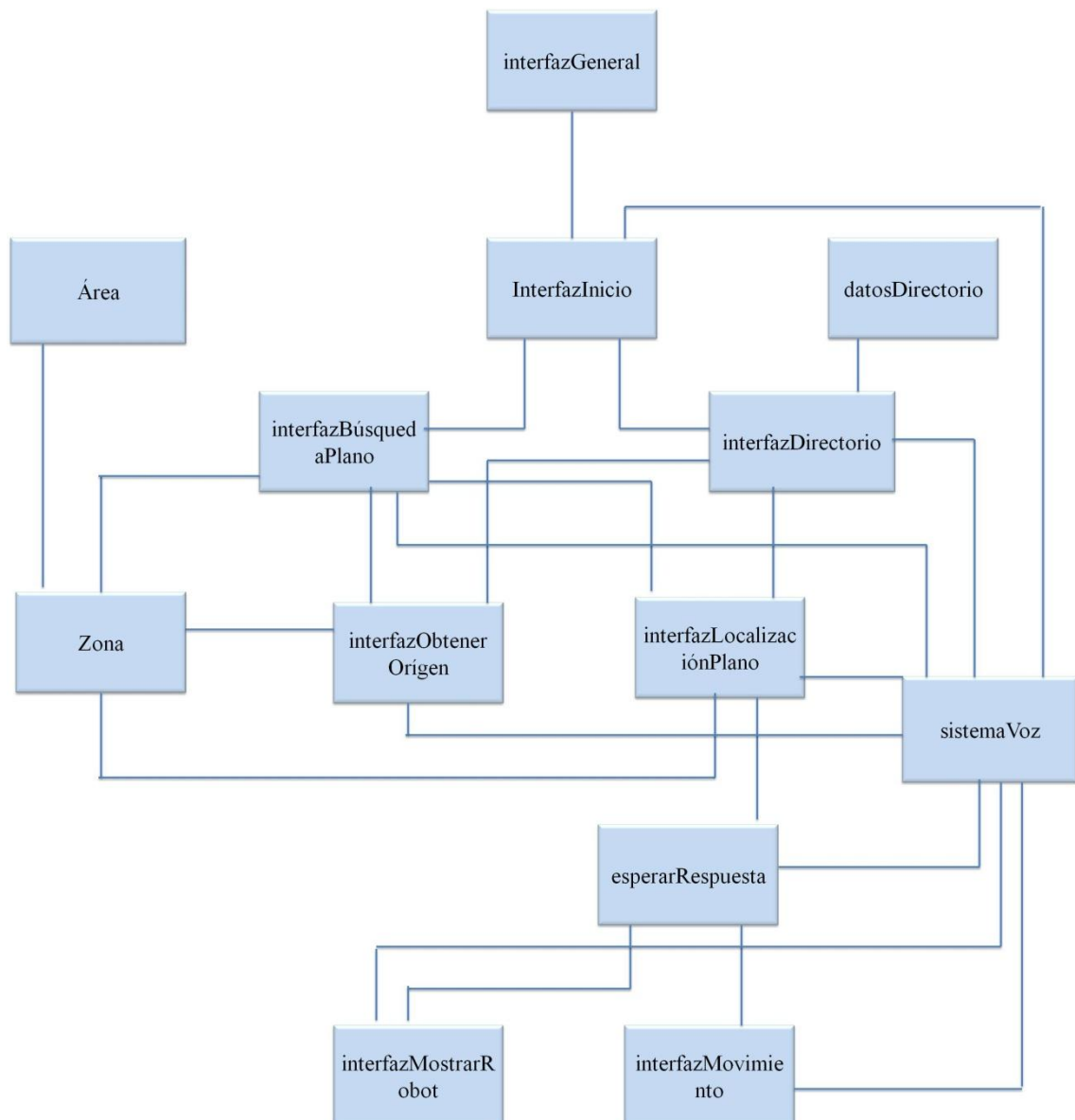


Figura 28. Diagrama de Clases Simplificado

A continuación se muestra una breve descripción de las clases mostradas en el diagrama anterior:

- **Zona:** clase que contiene los atributos de las zonas en las que se divide el plano (zonas A, B, C y D del edificio Sabatini). Los atributos de esta clase son la letra, las coordenadas y las áreas por las que está formada dicha zona.
- **Área:** clase que contiene los atributos de cada una de las divisiones de una zona. Los atributos de esta clase son el identificador de la división, las coordenadas y las áreas con las que está conectada dicha división.

- **datosDirectorio:** clase que contiene los atributos necesarios para almacenar los datos de un profesor buscado en el directorio. Los atributos de esta clase son el nombre completo del profesor, su teléfono, email y despacho.
- **interfazGeneral:** clase que crea una ventana inicial con un panel principal sobre el que se insertarán el resto de paneles y componentes del resto de clases y un cuadro de texto sobre el que se escribirán los avisos de la interfaz.
- **interfazInicio:** clase que muestra la pantalla de inicio en la que se pueden escoger las opciones de buscar por medio del plano o a través de la introducción de los datos del profesor al que se desea visitar.
- **interfazBúsquedaPlano:** clase que permite al usuario escoger el lugar de destino por medio del plano de la universidad a través de una imagen general e imágenes ampliadas de cada una de las zonas.
- **interfazDirectorio:** clase que permite al usuario escoger el lugar de destino por medio de la búsqueda en un directorio de los datos del profesor al que desea visitar.
- **interfazObtenerOrigen:** clase que solicita al usuario el lugar de origen donde se encuentra el robot por medio de una imagen general del plano y de imágenes ampliadas de cada una de las zonas.
- **interfazLocalizaciónPlano:** clase que muestra por pantalla el plano sobre el que se insertará la imagen del robot según vaya avanzando.
- **esperarRespuesta:** clase que sirve de comunicación con el programa controlador del robot esperando la respuesta de éste que le indique si la acción se ha producido correctamente o si ha habido algún error, todo ello junto con el lugar de desplazamiento.
- **interfazMostrarRobot:** clase que muestra al robot por pantalla colocado en la posición que el controlador del robot ha enviado en la clase esperarRespuesta.
- **interfazMovimiento:** clase que permite al usuario recolocar al robot en el pasillo cuando éste ha encontrado un obstáculo o se ha chocado contra algo que le impidiera continuar con el desplazamiento.
- **sistemaVoz:** clase encargada de reproducir el mensaje de voz correspondiente según se vaya realizando la navegación con la interfaz gráfica.

3.2.3 Comunicación Interfaz y Arquitectura ROPEM

Tal y como se comentó anteriormente, la arquitectura ROPEM y la interfaz deben intercambiarse determinados mensajes para que el robot actúe de forma adecuada según las indicaciones del usuario y para que la interfaz muestre el estado en que se encuentra el robot en cada momento.

Inicialmente la arquitectura ROPEM se encontrará a la espera de una orden de desplazamiento. Esta orden le llegará por medio del envío de la ruta de un fichero que

contendrá toda la información relativa al estado actual en el que se encuentra el robot, el destino al que debe desplazarse y las acciones que puede llevar a cabo para lograr el objetivo, así como una descripción del escenario donde se desplazará. En un principio se trató de incluir toda la información relativa al plano en el que se desplazará el robot en un único fichero pero, debido a la gran cantidad de información que éste tenía y a que el número de elementos que el planificador puede manejar es limitado, fue necesario realizar una separación de la información del problema de la información del mapa de navegación. Por lo tanto, en un primer lugar, se envía el fichero con la información completa del mapa de navegación y en segundo lugar, el fichero con la información necesaria para describir el problema y tomar las decisiones oportunas para llegar al destino seleccionado por el usuario.

El mapa de navegación se encuentra dividido en una serie de áreas o waypoints, cuya partición se explicará más adelante, La información referente a dicho mapa de navegación contendrá, por lo tanto, la descripción (coordenadas y conexiones) de los distintos waypoints así como el estado inicial y el objetivo a ser alcanzado del problema. Por otro lado, la información referente al problema a resolver contiene tan solo las conexiones entre los distintos waypoints y el estado inicial y objetivo del problema, no incluyendo por lo tanto las coordenadas de las distintas áreas en las que se encuentra dividido el mapa de navegación.

Una vez que el robot haya recibido la ruta de dichos ficheros comenzará a desplazarse. Cada vez que el robot avance a un waypoint distinto se enviará un mensaje a la interfaz informándole del lugar donde se encuentra para así poder mostrar al usuario por pantalla el lugar donde se encuentra el robot en cada momento.

En caso de que el robot se choque contra algo o detecte que existe un obstáculo que le impide seguir desplazándose, el sistema de control mandará un mensaje de error a la interfaz para que el usuario pueda ayudar al robot recolocándole en el pasillo. Para recolocarlo, la persona empleará una serie de flechas que aparecerán por pantalla. Cada vez que el usuario pulse en una de estas flechas, la interfaz enviará al programa de control del robot un mensaje con el movimiento correspondiente que debe realizar y éste responderá con un mensaje que indique si el movimiento realizado ha sido o no correcto. En caso de que durante el movimiento del robot por medio de las flechas, éste se choque o detecte un obstáculo que impida su movimiento se enviará un mensaje de error a la interfaz para que el usuario pueda asistirle de nuevo.

Tras el envío de cada uno de los mensajes de confirmación del desplazamiento a los waypoints, el programa correspondiente con el robot quedará a la espera de un nuevo mensaje. En caso de que durante el desplazamiento del robot hacia el último waypoint, el usuario haya pulsado en “Reubicar” o en “Stop” la interfaz enviará al programa de control del robot un mensaje indicando que el robot debe reubicarse o detenerse respectivamente. En un primer lugar se pensó en esta comunicación como una serie de interrupciones durante el desplazamiento pero, debido a la complejidad de desarrollo de esta alternativa y a que se sale de los objetivos del proyecto actual, se optó por permitir que estos mensajes sólo pudieran ser mandados una vez alcanzado cada uno de los waypoints del pasillo. Debido a esto, cuando el usuario pulsa “Stop” o “Reubicar”, el robot continuará con su desplazamiento hasta que llegue al siguiente waypoint, por lo

que se incluyó un mensaje al usuario en el cuadro de avisos y por el sistema de voz que indica que la acción de STOP o Reubicar puede tardar unos minutos.

Si, tal y como se ha indicado, el usuario observa que el lugar indicado por el plano no corresponde con la localización real del robot, podrá reubicarlo. En ese caso, la interfaz enviará al sistema de control del robot un mensaje de STOP para que el robot se detenga y a continuación una nueva ruta del fichero con la nueva descripción del problema a resolver (variará el estado inicial donde se encuentra el robot). Este fichero corresponderá con el fichero parcial explicado anteriormente que tan sólo contendrá la mínima información necesaria para una correcta planificación del problema.

Del mismo modo, durante el desplazamiento del robot, el usuario puede decidir detener al robot, por lo que la interfaz enviará un mensaje de STOP y el programa de control del robot quedará a la espera de la nueva ruta del fichero parcial con la nueva descripción mínima necesaria del problema a resolver.

El intercambio de todos estos datos puede observarse en la Figura 29.

Tal y como se ha comentado, para la navegación del robot por el edificio, es necesario enviar al programa de control del robot una descripción del problema a resolver. Esta descripción es enviada a través de rutas en dos partes: la primera contendrá la información del mapa de navegación y la segunda la información del problema para realizar la planificación. Esta representación es generada automáticamente por la interfaz a partir de una estructura creada previamente al programa y que pueda ser modificada, interpretada y ampliada de forma sencilla por parte de posibles usuarios futuros.

El plano del edificio se encuentra representado en un fichero XML con una estructura jerárquica en la que el nivel más alto es campus, al que le sigue el nivel hijo edificio y a éste último el nivel planta. El nivel planta tiene como hijos a cada una de las zonas en las que se encuentra dividido el edificio y a éstas le siguen los elementos hijos “áreas” que representan cada una de las divisiones o waypoints que realizamos del plano y que el robot distinguirá para su avance. Cada una de las áreas tiene una serie de conexiones, que son las áreas con las que se encuentra comunicado directamente. Esta jerarquía podemos observarla en la Figura 30.

Las zonas y las áreas poseen una serie de atributos que completan la representación del plano. Estos atributos son un identificador de la letra o número de la zona o área y las coordenadas x e y de inicio y de fin de la zona o área correspondiente. Un ejemplo de estos atributos puede observarse en la Figura 31, en la que puede observarse cómo la x e y de inicio se encuentran en el vértice superior izquierdo, mientras que la x e y de fin corresponden con el vértice inferior derecho del área.

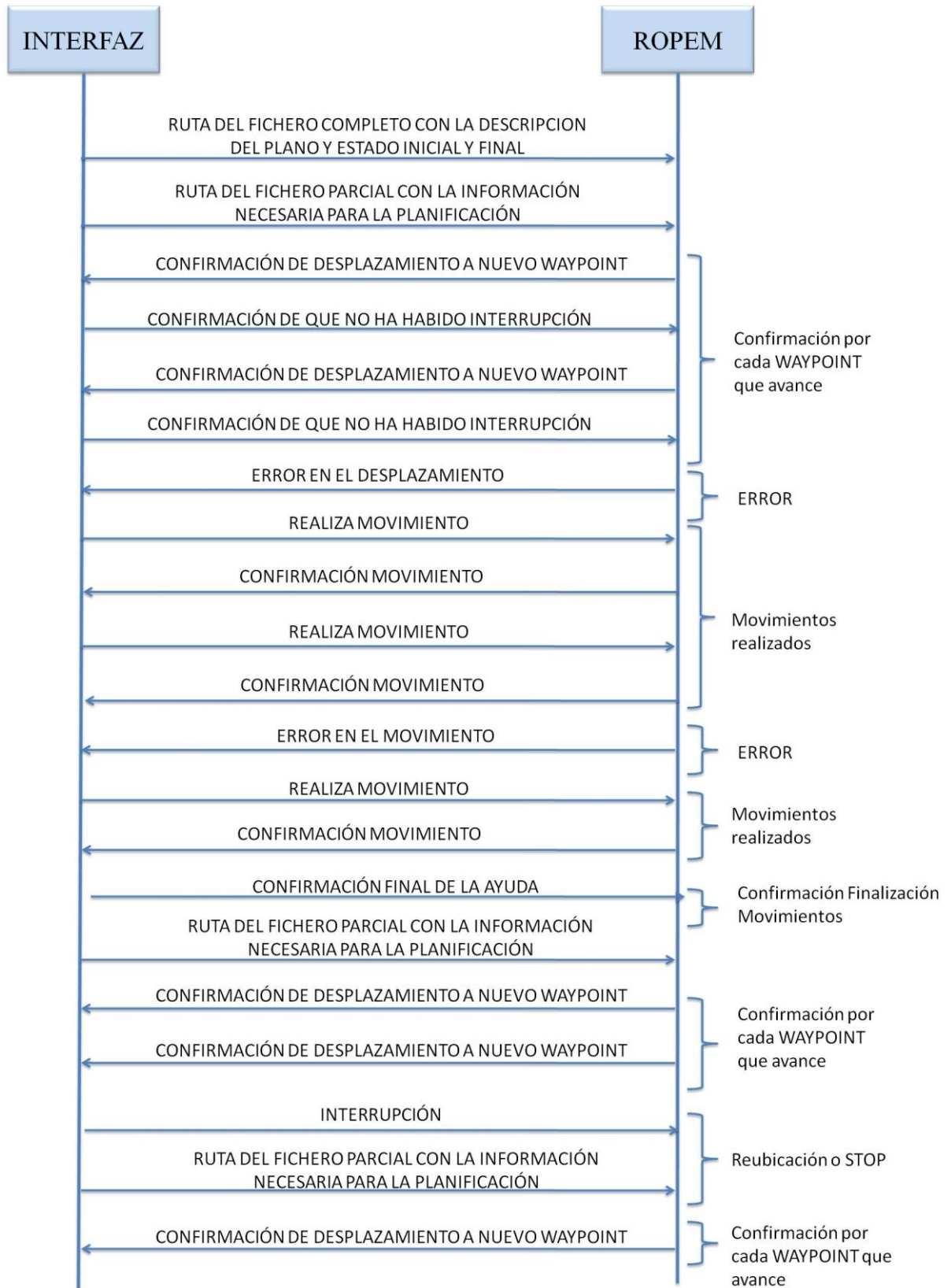


Figura 29. Intercambio de Mensajes entre Interfaz y Robot

El criterio empleado para realizar la partición de las distintas áreas o waypoints es realizar una división por cada puerta de despacho. De esta forma, un despacho con dos puertas se dividiría en dos áreas y el fragmento de pasillo correspondiente con dicho despacho igualmente se dividiría en dos partes. Un ejemplo de esta división puede observarse en la Figura 32, en la que se realizan dos divisiones en el despacho debido a la existencia de dos puertas y otras dos en el lugar del pasillo correspondiente. Adicionalmente, fue añadida una línea de waypoints en la parte interior del plano en la zona correspondiente con el patio del edificio para contemplar todos los posibles casos y para tener un margen de distancia en el caso de que el robot considere que se encuentra en dicho lugar por la acumulación de error en el cálculo de su posicionamiento.

El plano del edificio se ha escalado a 1000x1000 y, a partir de esta escala, han sido calculadas las distintas coordenadas de todas las zonas y áreas. Un ejemplo de una porción del fichero XML con la descripción del plano se presenta en el Anexo C.

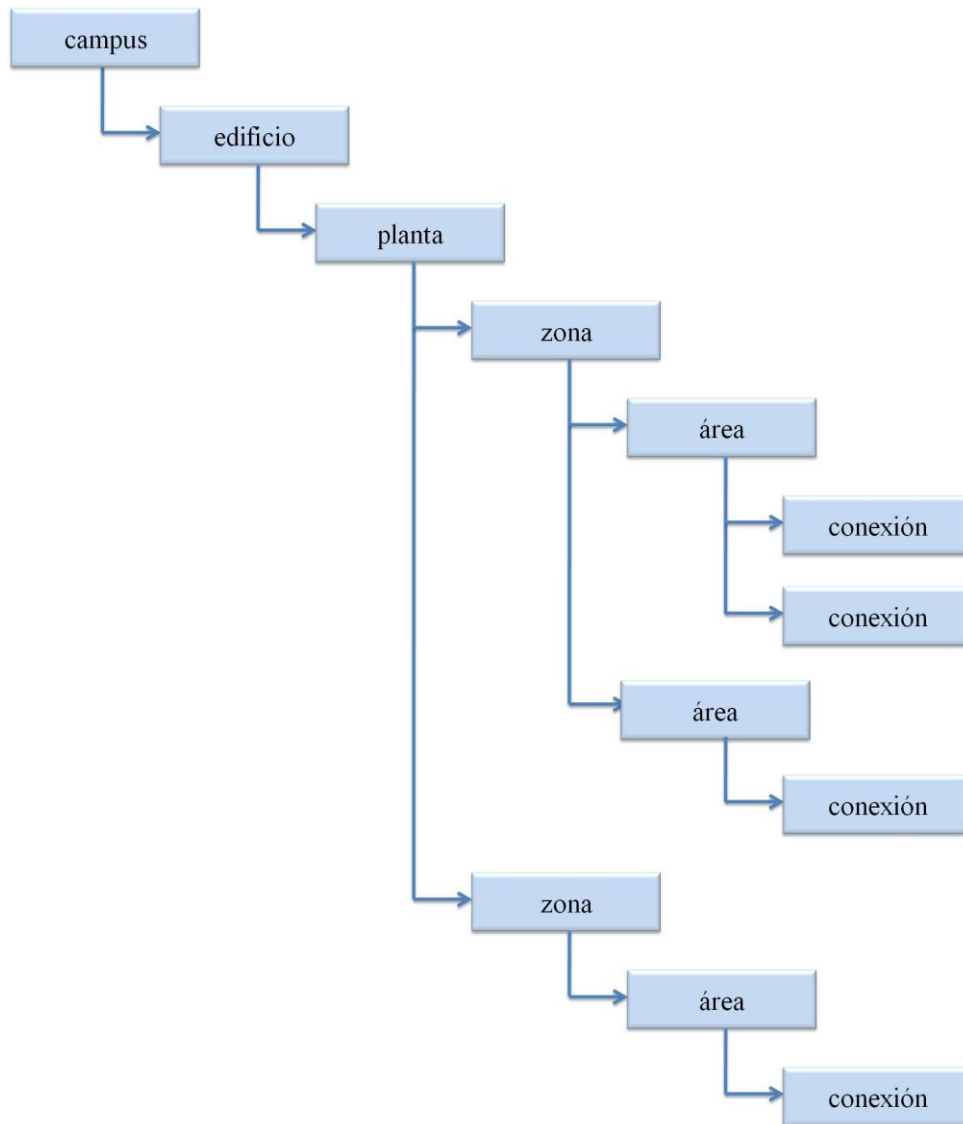


Figura 30. Representación Jerárquica del Plano

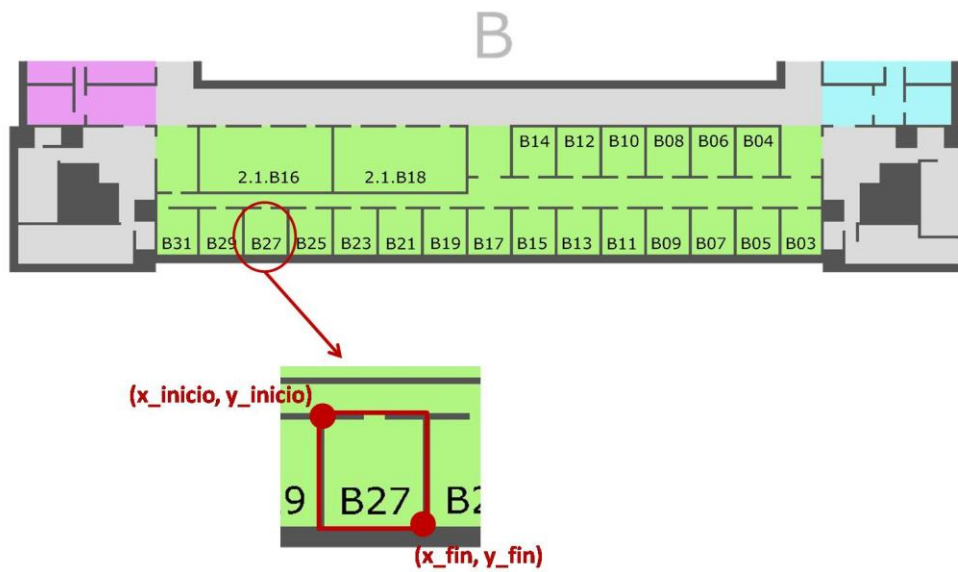


Figura 31. Ejemplo de las Coordenadas de un Área del Plano

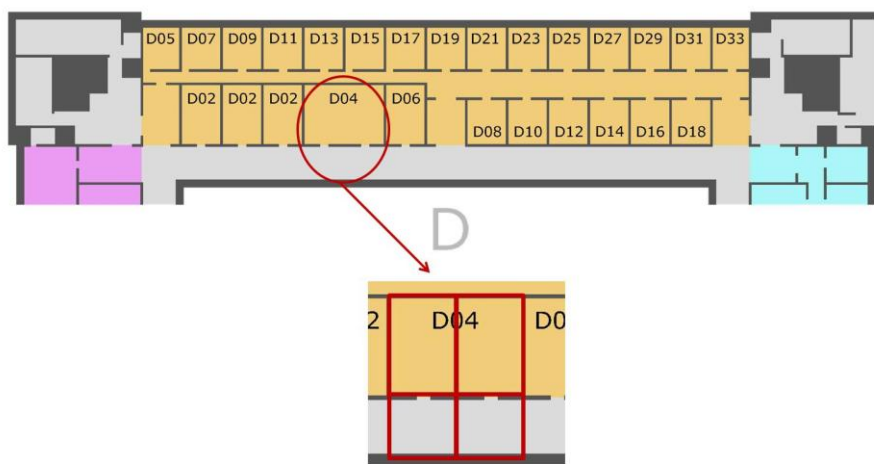


Figura 32. Ejemplo de la División del Plano

Toda esta información será empleada por el programa al que pertenece la interfaz para generar los ficheros que contengan la descripción del problema a resolver. Para ello se ha diseñado una estructura que permite al robot conocer el estado inicial en el que se encuentra, el destino a ser alcanzado, las características del entorno en el que se desplaza y las acciones que puede ejecutar para alcanzar el objetivo marcado. Todos estos datos se almacenarán en un fichero PDDL, que es un lenguaje estándar de codificación para la planificación, que será creado por la interfaz para que el robot pueda emplearlo para decidir las acciones a realizar para llegar al destino. Dicho fichero seguirá la siguiente estructura: en primer lugar aparecerá la declaración de todos los

elementos tratados en el fichero. A continuación se encontrará una descripción del entorno en la que aparecerán las distintas áreas existentes con sus coordenadas y tamaños y los distintos waypoints con los que se encuentran comunicadas cada una de las áreas. Finalmente aparecerá el estado inicial donde se encuentra el robot y el destino o meta a ser alcanzado. Un ejemplo de dominio y problema PDDL junto con la solución generada por el planificador puede observarse en el Anexo D.

3.3 Diseño Detallado

A continuación se exponen los detalles pertenecientes al diseño detallado de los apartados anteriores. En primer lugar se describe el funcionamiento de la arquitectura ROPEM empleada en este proyecto y a continuación se exponen el diseño detallado de la interfaz gráfica y de los mecanismos de conexión entre la interfaz y el programa controlador del robot.

3.3.1 Arquitectura ROPEM

A continuación se realiza una breve descripción del funcionamiento de ROPEM, arquitectura empleada para el control del robot y que se encuentra comunicado a alto nivel con la interfaz gráfica, para recibir y enviar el problema a resolver y los datos que deben ser mostrados al usuario respectivamente, y a bajo nivel con la plataforma de control robótico Player, para enviar las órdenes de movimiento que deben ser realizadas por el robot y obtener información de los sensores del mismo.

En primer lugar se establece la conexión con la interfaz gráfica, tras lo que se recibe la ruta del fichero pddl que contiene la información del mapa de navegación. Una vez recibida dicha ruta, se almacena el mapa de navegación en un formato interno y se obtiene el estado de bajo nivel del robot, traduciéndolo a continuación a alto nivel.

Cada vez que se pulsa en “Reubicar” o “STOP” o cada vez que se produce un error durante el desplazamiento o se inicia una nueva navegación, se necesita recibir un nuevo problema. Los pasos anteriores se realizan una única vez al iniciar el programa controlador del robot pero, cada vez que se recibe un problema nuevo, la arquitectura ROPEM espera en primer lugar la ruta del fichero pddl que contiene la descripción de dicho problema, tras lo cual carga el lugar de origen y obtiene el estado de bajo nivel del robot a través de la capa de control traduciéndolo a continuación a alto nivel. Una vez realizados los pasos anteriores se ejecuta Sayphi, que se trata de un planificador que establece las acciones a realizar para alcanzar el objetivo del problema. Por cada una de las acciones establecidas por el planificador, se envía a la plataforma Player el movimiento correspondiente para alcanzar el objetivo determinado por dicha acción, enviando el resultado que provoca dicha acción a la interfaz gráfica, pudiendo ser este resultado éxito o fracaso.

Una vez realizada cada una de las acciones establecidas por el planificador, la arquitectura ROPEM queda a la espera de que la interfaz gráfica envíe un mensaje que indique si durante el desarrollo de esa acción se ha producido un error o una interrupción por parte del usuario. En caso de haberse producido una interrupción o de haber finalizado el total de acciones a realizar, se procederá a esperar la ruta del fichero pddl con la descripción de un nuevo problema a resolver, tras lo que se repetirán los pasos anteriores. En caso de haberse producido un error, se recibirán las cadenas xml enviadas desde la interfaz gráfica y cuya orden será enviada al código de bajo nivel, tras lo que será enviada a la interfaz gráfica un mensaje con el resultado de si el movimiento se ha realizado o no correctamente. Una vez acabada la recolocación del robot por medio de estos movimientos se pasará a esperar la ruta del fichero pddl con la descripción de un nuevo problema a resolver, tras lo que se repetirán los pasos anteriores.

El programa controlador del robot emplea dos tipos de medidas para localizar su posición actual y calcular las distancias a recorrer en sus desplazamientos. En primer lugar se distinguen las medidas reales del robot, que almacenan la distancia recorrida por el robot asumiendo un inicio en (0,0). Por otra parte, se distinguen las medidas respecto al plano, que tienen en cuenta los metros reales del pasillo, de las que se almacenan las coordenadas iniciales de cada problema a resolver. Adicionalmente, fue necesario almacenar la distancia recorrida por el robot cada vez que se recibe un nuevo problema. Este último almacenamiento es preciso para calcular la distancia que se ha de recorrer, ya que este valor ha de ser añadido o sustraído al calcular la distancia a recorrer y el lugar donde se encuentra el robot respectivamente. El cálculo de la distancia a recorrer es distinto si el desplazamiento se realiza en el eje x o en el eje y . Para el eje x el desplazamiento que debe ser realizado por el robot es calculado restando las coordenadas de destino con las de origen y sumándole a dicha diferencia el valor de la distancia que ha recorrido al recibir el nuevo problema. Un ejemplo de este cálculo puede observarse en la Figura 33 y Figura 34. En ellas, se puede observar que el robot tiene inicialmente las coordenadas (0,0) aunque se encuentre en las coordenadas respecto del plano (23'78,74'76). Al desplazarse del Waypoint 66 al 70 el robot resta las coordenadas de destino menos las del origen y suma los 0 metros que lleva acumulados en el desplazamiento al recibir este primer problema. En la Figura 34, observamos cómo, al recibir un nuevo problema, el robot llevaba recorridos 4'21 metros en el eje x del desplazamiento anterior y, para calcular la distancia a recorrer, realiza la diferencia entre las coordenadas de destino y las del origen a las que suma esos 4'21 metros que ya llevaba recorridos. De esta forma, el robot debe desplazarse hasta los 8'41 metros y , como ya había recorrido 4'21 se desplazará 4'2 metros alcanzando así el Waypoint 82.

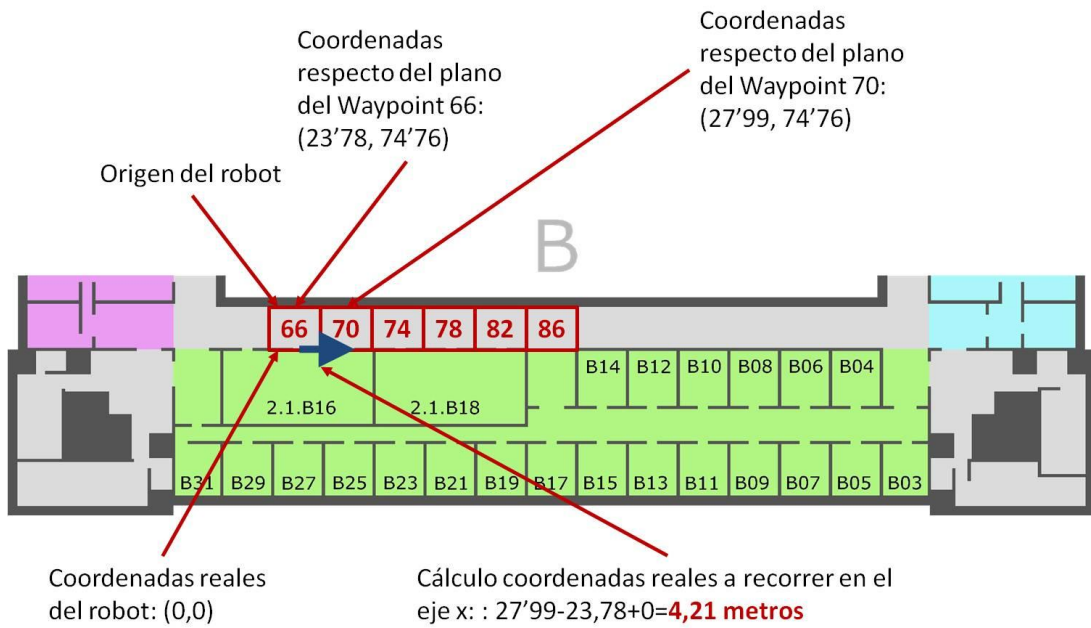


Figura 33. Ejemplo de Cálculo de Coordenadas de Desplazamiento

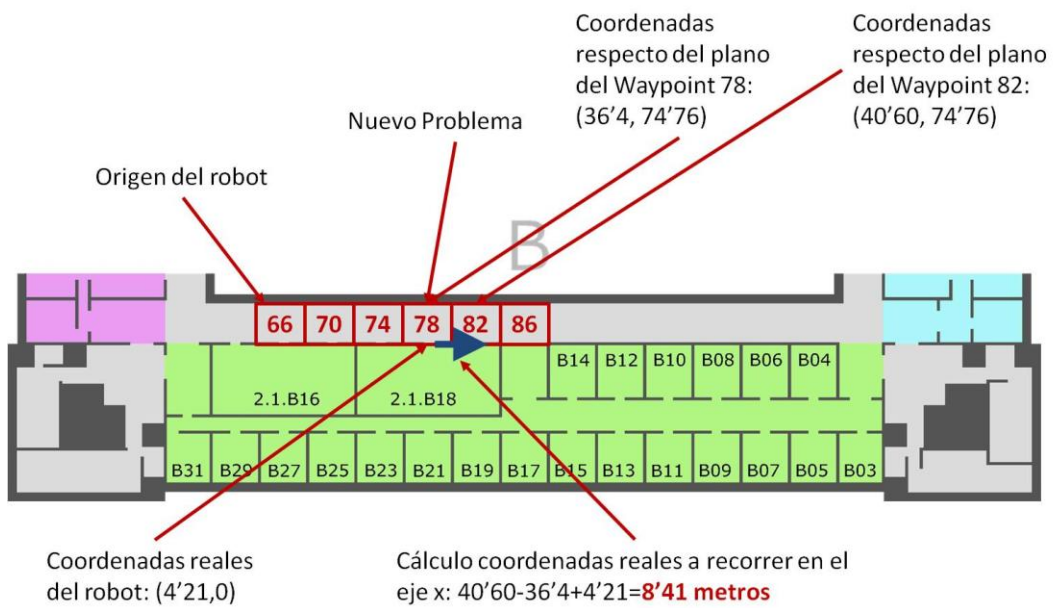


Figura 34. Ejemplo de Cálculo de Coordenadas de Desplazamiento al Recibir un Nuevo Problema

En el eje y el cálculo se realiza de forma diferente según nos desplazemos hacia las coordenadas y positivas o negativas. Para las coordenadas y positivas el cálculo es realizado restando las coordenadas de destino con las de origen y sumándole al valor absoluto de dicha diferencia el valor de la distancia que ha recorrido al recibir el nuevo problema. Por el contrario, para las coordenadas y negativas el cálculo es realizado restando las coordenadas de destino con las de origen, restándole al valor absoluto de dicha diferencia el valor de la distancia que ha recorrido al recibir el nuevo problema y cambiándole el signo al resultado obtenido. Un ejemplo de estos cálculos puede observarse en la Figura 35, Figura 36 y Figura 37. En ellas se puede observar que el robot tiene inicialmente las coordenadas (0,0) aunque se encuentre en las coordenadas respecto del plano (15'51,63'31). Al desplazarse del Waypoint 40 al 44 nos estamos desplazando en el eje y negativo, por lo que se restan las coordenadas de destino menos las del origen y se restan los 0 metros que lleva acumulados en el desplazamiento al recibir este primer problema, tras lo que se cambia el signo al resultado obtenido. En la Figura 36, observamos cómo, al recibir un nuevo problema el robot llevaba recorridos 3,89 metros en el eje y negativo del desplazamiento anterior, lo que hace que la posición en la que se encuentre sea (0, -3'89). Para calcular la distancia a recorrer, se realiza la diferencia entre las coordenadas de destino y las del origen a las que se restan esos -3'89 metros resultantes de lo que ya llevaba recorrido, tras lo que se cambia el signo al resultado obtenido. De esta forma el robot debe desplazarse hasta los -7'6 metros y, como ya se encontraba en -3'89 se desplazará 3'71 metros en el eje y negativo, alcanzando así el Waypoint 52.

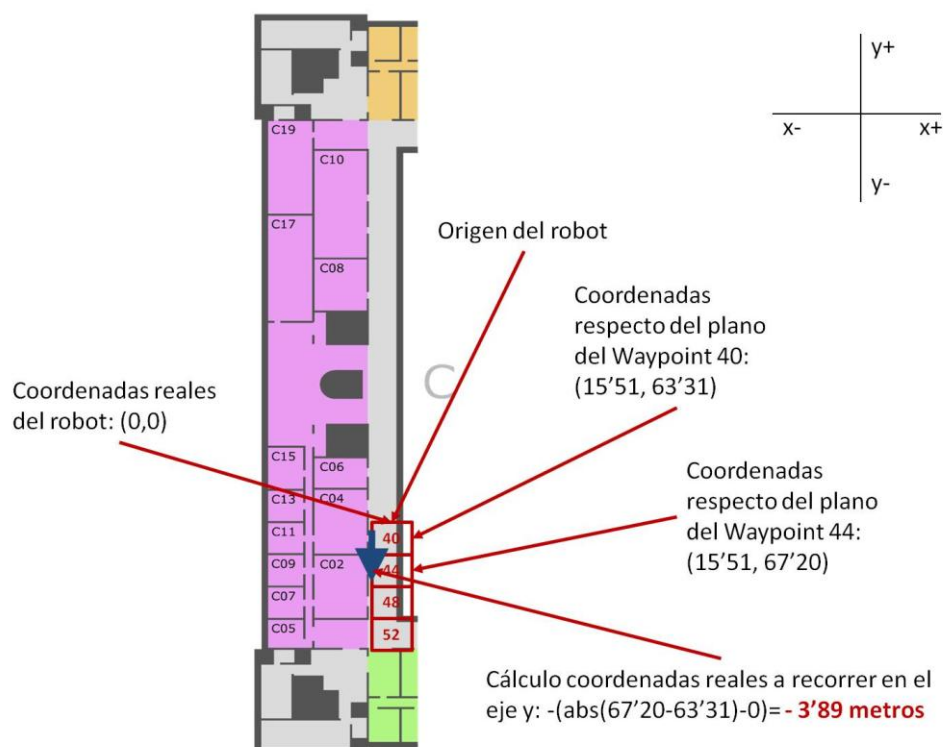


Figura 35. Ejemplo de Cálculo de Desplazamiento en el Eje y Negativo

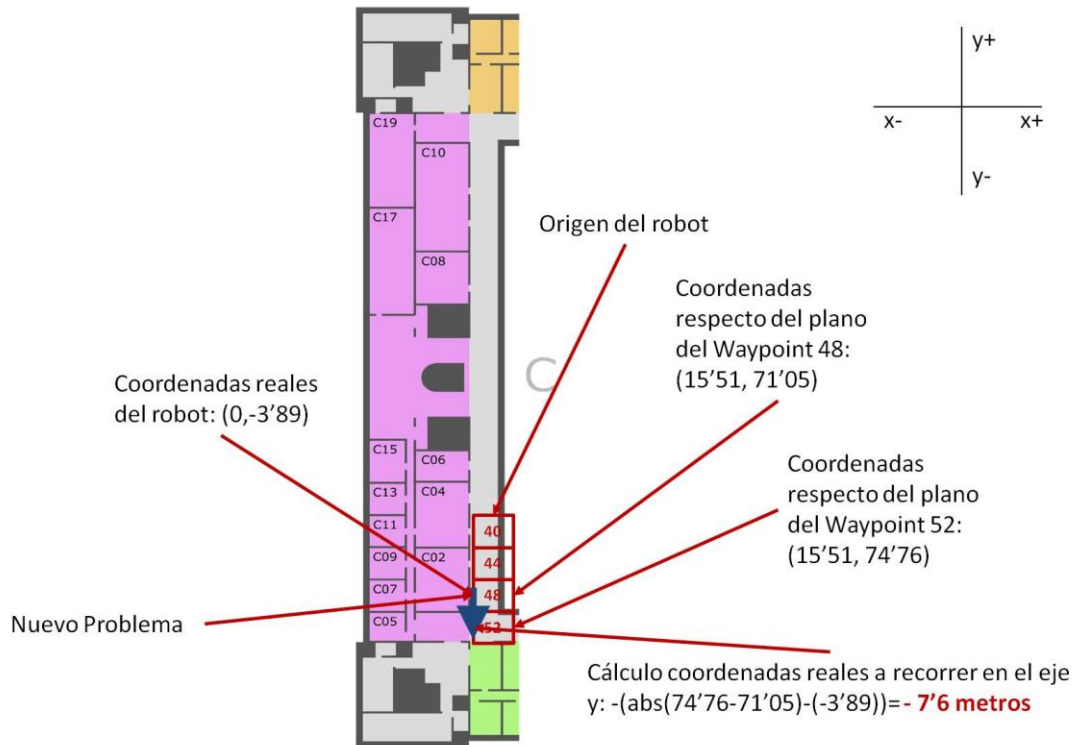


Figura 36. Ejemplo de Cálculo de Desplazamiento en el Eje y Negativo al Recibir un Nuevo Problema

En la Figura 37 observamos el desplazamiento en el eje y positivo al recibir un nuevo problema partiendo del caso anterior. Hasta este punto, el robot llevaba recorridos 7'6 metros en el eje y negativo, lo que hace que la posición en la que se encuentre sea (0, -7'6). Para calcular la distancia a recorrer, se realiza la diferencia entre las coordenadas de destino y las del origen, a cuyo resultado se suman esos -7'6 metros consecuencia de lo que ya llevaba recorrido el robot. De esta forma el robot debe desplazarse hasta los -3'89 metros y, como se encontraba en -7'6, se desplazará 3'71 metros en el eje y positivo, alcanzando así el Waypoint 48.

Con respecto a los movimientos realizados por el robot, ya sea en la navegación por el edificio como en su colocación por medio de las flechas de la interfaz, su velocidad es de 0'2 m/s para el avance y retroceso y de 0'2 rad/s para los giros realizados a derecha e izquierda. Esta velocidad fue decidida tras realizar unos estudios previos al proyecto en los que se tuvo en cuenta que, con mayores velocidades, el error acumulado era mayor.

En cuanto a los sensores del robot, se estableció una distancia de alcance de 0'5 metros, por lo que si el robot encuentra un obstáculo a esa distancia se detendrá. Debido a que alguno de los pasillos por los que debe moverse son estrechos, el robot detectaba el obstáculo de la pared al desplazarse al lado de ella, por lo que se desactivó la comprobación de esa distancia en los cuatro sensores laterales del robot, dos de cada lado, quedando activa la comprobación en los cuatro sensores centrales.

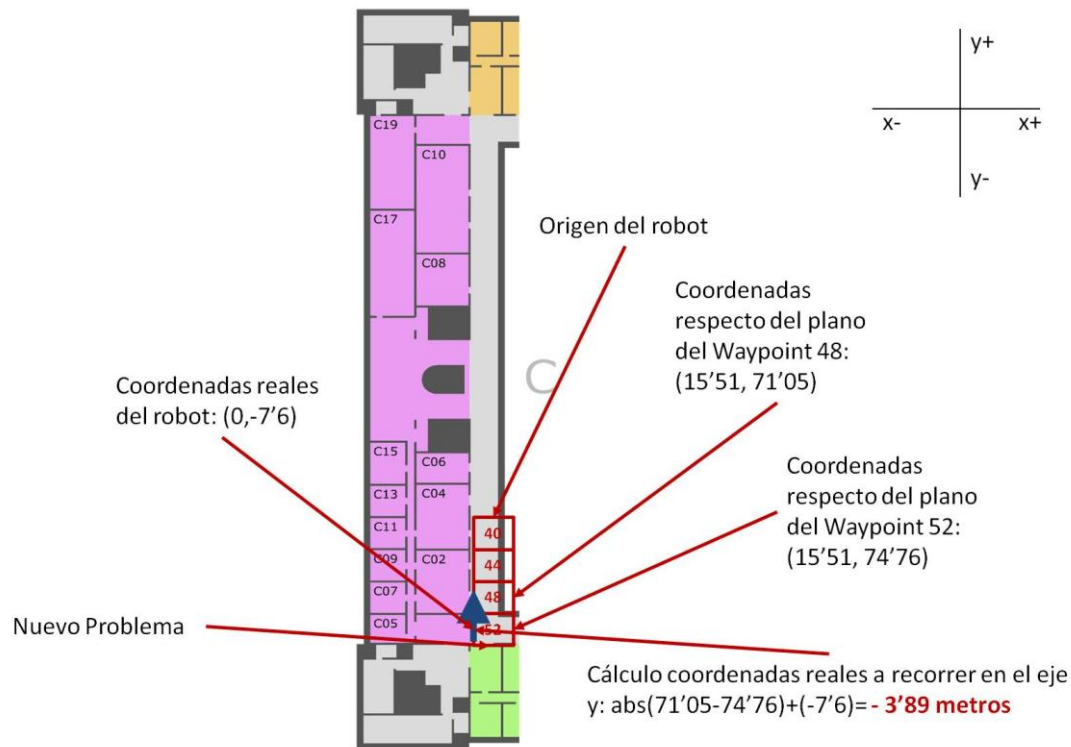


Figura 37. Ejemplo de Cálculo de Desplazamiento en el Eje y Positivo al Recibir un Nuevo Problema

3.3.2 Interfaz Gráfica y Sistema de Voz

La interfaz creada para la comunicación entre Hombre y Robot ha sido desarrollada por medio del lenguaje de programación Java empleando para ello Netbeans IDE 6.9.1. En el programa desarrollado han sido implementadas las clases anteriormente detalladas en el apartado de Diseño.

Tanto en la clase interfazBusquedaPlano como en interfazObtenerOrigen, el usuario escoge el lugar al que desea desplazarse o donde se encuentra por medio de un plano. Para la obtención de esta información a partir de la selección por parte del usuario se ha empleado un fichero con extensión .xml. Cada vez que el usuario pulsa sobre un lugar del plano, el programa lanza un evento de pulsación sobre un botón transparente que se encuentra encima de ese lugar. La creación de estos botones la realiza la interfaz a partir de la lectura del fichero .xml. En este fichero se encuentra almacenada toda la información correspondiente con la descripción de las distintas zonas y áreas del plano. Este fichero sigue la estructura jerárquica expuesta en el apartado de Diseño y es creado manualmente por el usuario previamente a la ejecución del programa. XML es un metalenguaje que, por medio de etiquetas, crea una estructura fácil de manejar y modificar o ampliar en el futuro. Debido a estas características fue elegido como forma de representación del plano ya que, para futuras ampliaciones del proyecto, resultaría muy sencillo y cómodo añadir nuevas etiquetas con nuevos campos necesarios además de resultar muy fácil de interpretar por terceros usuarios. Este fichero es empleado tanto

para la implementación de la interfaz desarrollada como para la creación del fichero que contiene la descripción del escenario que, como se explicará más adelante, será empleado en la comunicación con el robot. Es por ambas cosas, que es necesario el tratamiento y manipulación de ficheros .xml por medio del lenguaje Java, para lo que empleamos JDOM. JDOM es una librería de Java que permite el procesamiento de ficheros con extensión .xml. Esta librería fue escogida entre otras existentes debido a que fue creada específicamente para Java, lo que hace que sea más eficiente y cómoda de usar que otras similares. JDOM genera un árbol de nodos a partir del fichero .xml, lo que permite que la búsqueda de sus elementos sea rápida y sencilla.

En la clase `interfazDirectorio` el usuario escoge el lugar al que desea desplazarse por medio de la introducción de los datos del profesor. Para ello, ha sido necesario realizar una conexión con el directorio de la Universidad. Para acceder al directorio en el que se encuentran almacenados los datos del personal de la Universidad se emplea un mecanismo denominado LDAP. LDAP es un protocolo que permite acceder a los datos almacenados en un directorio y de esta forma realizar búsquedas en él. El directorio se encuentra almacenado de forma jerárquica y por medio de LDAP se realiza una búsqueda filtrando los resultados según los términos introducidos. Para integrar estas búsquedas en el código Java de nuestro programa, ha sido preciso emplear la librería JNDI que permite realizar búsquedas en el directorio de una forma sencilla.

3.3.3 Comunicación Interfaz y Arquitectura ROPEM

La comunicación entre la interfaz y el programa controlador del robot es necesaria para todo el intercambio de mensajes. Gracias a ella, el usuario podrá comunicar al robot el lugar al que desea moverse o en el cual se encuentra y éste último podrá solicitar la ayuda del usuario en caso de necesitarla por algún motivo. La interfaz y el programa controlador deberán intercambiarse los mensajes mostrados para que el robot actúe conforme a las indicaciones dadas por el usuario y para que la interfaz muestre los mensajes que el robot debe comunicar al usuario en cada momento. Este intercambio de mensajes se ha realizado por medio de sockets. Un socket es una forma de comunicación entre dos programas, en el que uno actúa de cliente y otro de servidor. Un socket queda definido por un número de puerto, un protocolo de comunicación y una dirección IP. El protocolo de comunicación empleado en este caso ha sido TCP ya que garantiza que todos los mensajes serán recibidos correctamente, sin omisiones o errores, y en el mismo orden en que fueron enviados. Tal y como se ha comentado anteriormente, uno de los programas comunicados a través de sockets actúa de cliente, mientras que el otro actúa de servidor. En este caso, la interfaz desarrollada toma el papel de cliente, mientras que el programa que controla el robot actúa como servidor, ya que es el que permanece abierto a la espera de que el otro desee establecer la comunicación.

Una vez que la conexión entre ambos programas se encuentra establecida, se procede a enviar los mensajes de comunicación necesarios. Para que dichos mensajes

sean comprendidos tanto por el robot como por la interfaz y para que sean fácilmente entendibles y modificables por futuros usuarios del programa, se ha empleado una estructura xml. La elección de que los mensajes estén escritos en xml se debe a que se trata de un metalenguaje de comunicación fácilmente entendible y extensible por terceros y que aumenta la compatibilidad entre el robot y la interfaz ya que existen librerías en los lenguajes en que se encuentran codificados ambos programas para su procesamiento. La interfaz desarrollada se encuentra programada en lenguaje Java por lo que también es empleada la librería JDOM para el procesamiento de los mensajes recibidos. En el caso del programa que controla al robot, también emplea una librería, con funciones en C que permite manejar, procesar y crear archivos .xml fácilmente. Los mensajes XML intercambiados se encuentran en el Anexo E. Por medio de esta comunicación establecida entre la interfaz y el programa controlador del robot, éstos se enviarán los mensajes que se detallaron en el apartado de 3.2.3. Los mensajes intercambiados pueden resumirse en:

- 1 La interfaz envía al programa controlador del robot la ruta de los ficheros necesarios para que este último realice la planificación de las acciones a llevar a cabo.
- 2 Una vez recibida esta información el programa controlador del robot realiza la planificación de las acciones que el robot debe realizar para alcanzar el objetivo seleccionado. Cada una de estas acciones es enviada secuencialmente al robot. En caso de que la acción sea realizada sin problemas, el programa controlador envía a la interfaz un mensaje de confirmación junto con la información del waypoint en el que se encuentra. En caso de que durante la realización de la acción los sensores o parachoques del robot hayan detectado algún obstáculo, el programa controlador envía a la interfaz un mensaje de error junto con la información del waypoint en el que cree que se encuentra.
- 3 La interfaz recibe el mensaje enviado por el programa controlador del robot. En caso de que se trate de un mensaje de error se procederá a recolocar al robot por medio de una serie de flechas, enviando para ello los mensajes correspondientes en cada caso.

Como se ha comentado, el programa controlador realiza una planificación de las acciones necesarias a llevar a cabo para alcanzar el objetivo seleccionado. Este proceso se realiza mediante planificación automática en la que, a partir de un fichero que contiene una descripción del problema a resolver y del dominio del mismo, el programa controlador decide qué acciones llevar a cabo para alcanzar el objetivo partiendo desde el estado inicial. Para la representación de este problema, es preciso crear una estructura que defina el escenario donde se producirá el movimiento. Esta estructura se almacenará en un fichero codificado en lenguaje PDDL. El lenguaje PDDL permite la descripción de problemas por medio de una serie de predicados que muestran el estado inicial del problema, las acciones que pueden llevarse a cabo, una descripción del dominio del problema y el objetivo a alcanzar. Debido a la variabilidad del entorno, este fichero es creado automáticamente por el programa correspondiente a la interfaz desarrollada. Para ello, emplea el fichero .xml de la representación del plano antes comentado que es creado manualmente por el usuario. De él extrae cada una de las áreas en las que se divide el plano, con sus coordenadas y las diferentes conexiones con el resto de waypoints. Esta información, junto con el lugar inicial donde se encuentra el robot y el

destino a ser alcanzado, conforma el fichero .pddl creado. La ruta de este fichero es enviada por medio de sockets al programa controlador del robot, que lo empleará para planificar la ruta a seguir para llegar al destino seleccionado y generará las órdenes de las acciones necesarias a llevar a cabo. Cada vez que varíe alguno de los elementos que conforman el PDDL, ya sea el lugar de inicio o el de fin, se creará un fichero .pddl nuevo y se enviará su ruta por sockets para que se generen las acciones a realizar.

Las acciones a realizar establecidas por el planificador son acciones de alto nivel que indican qué desplazamientos entre waypoints hay que realizar para alcanzar el objetivo. Estas acciones de alto nivel son traducidas posteriormente a acciones de bajo nivel, de forma que una acción de alto nivel que implica “navegar de waypoint i a waypoint j ” puede estar formada por varias acciones de bajo nivel como son “avanzar n metros en el eje x ”, “avanzar n metros en el eje y ”, “girar n grados hacia la derecha” o “girar n grados hacia la izquierda”.

El ciclo de creación del problema PDDL, junto con el envío de su ruta y la planificación de las acciones a realizar puede observarse en la Figura 38.

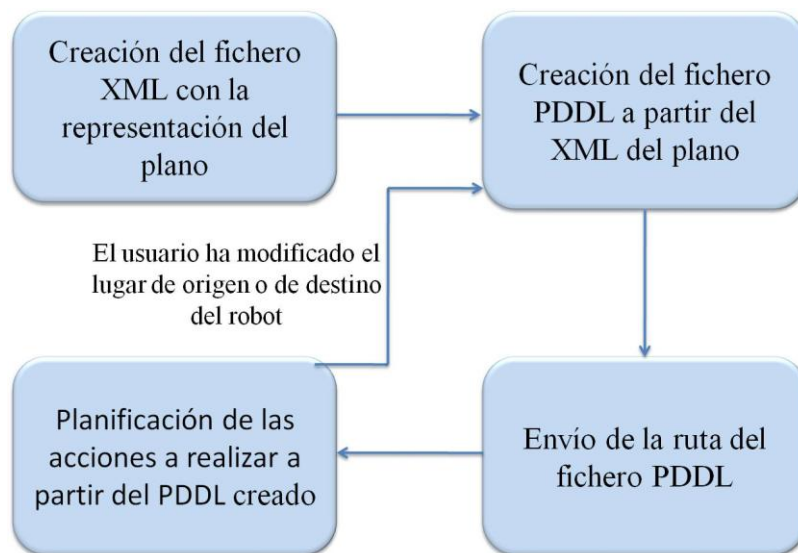


Figura 38. Creación y uso del fichero PDDL del problema

Como se comentó en el apartado de Diseño, en un principio se intentó que el fichero PDDL contuviera toda la información referente a todos los waypoints con sus coordenadas, tamaños y conexiones. Debido a la gran cantidad de información a almacenar, el fichero PDDL creado resultaba extremadamente extenso y el número de elementos imposibilitaba la planificación de las acciones a realizar a partir del fichero generado. Por este motivo se pasó a la creación de dos ficheros PDDL separados: uno parcial que contuviera la información mínima necesaria para la planificación de la ruta (definición de los waypoints, conexiones entre los mismos, estado inicial y objetivo a alcanzar) y otro completo que contuviera la información anterior junto con la

descripción detallada del plano en la que aparecieran las coordenadas y tamaños de cada waypoint. Ambos ficheros son creados automáticamente por el programa de la interfaz a partir del fichero XML con la descripción del plano creado manualmente por el usuario y sus rutas son enviadas por medio de sockets al programa controlador del robot. El PDDL completo tan sólo se envía una vez durante todo el transcurso del programa, ya que la información utilizada del mismo no varía (el estado inicial y objetivo sí pueden variar pero aunque se encuentran almacenados en este fichero no son utilizados, sino que esta información es obtenida del fichero .pddl parcial). El PDDL parcial, por lo tanto, es creado de nuevo y enviada su ruta por sockets cada vez que varíe algunos de los elementos que conforman dicho fichero, ya sea el lugar de inicio o de fin. El ciclo que se genera debido a esta división puede observarse en la Figura 39.

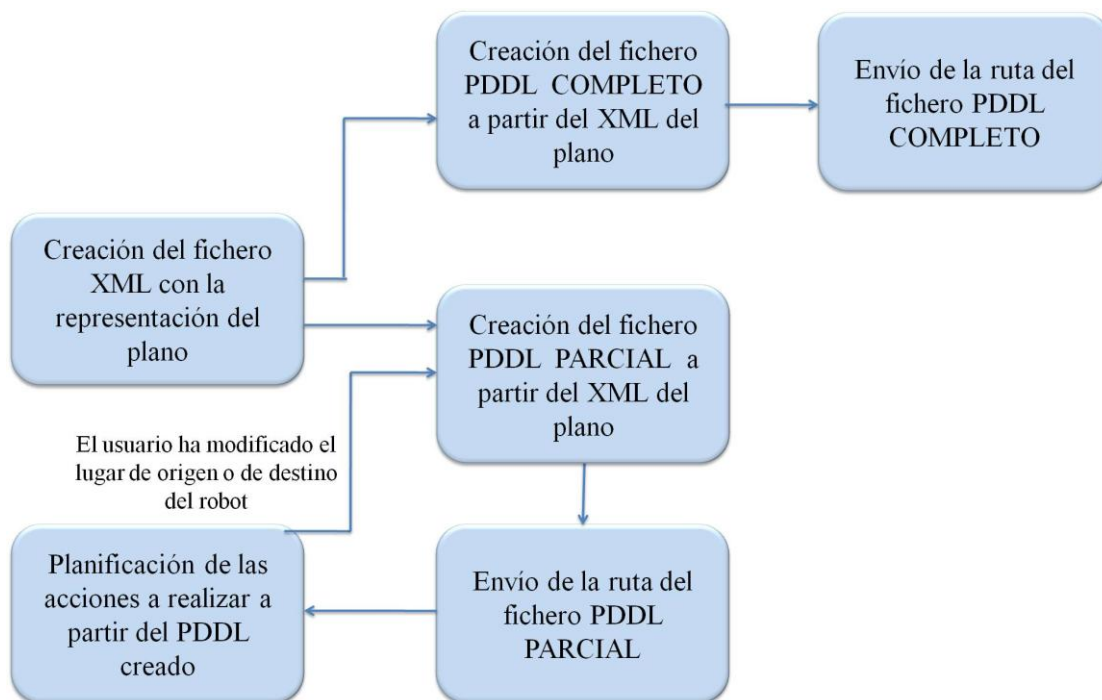


Figura 39. Creación y uso de los ficheros PDDL parcial y completo

Capítulo 4

Evaluación y Pruebas Preliminares

Previamente a la realización de la evaluación de la aplicación construida se realizaron una serie de pruebas preliminares para comprobar que todas las funcionalidades establecidas previamente se desarrollaban de forma adecuada y sin problemas.

4.1 Pruebas Preliminares

A continuación se detallan los objetivos de cada una de las pruebas realizadas y el resultado de salida que debía comprobarse en las mismas.

4.1.1 Funcionamiento Correcto

En primer lugar se probó el caso perfecto en el que el robot realiza un desplazamiento con llegada al destino sin interrupciones por parte del usuario ni errores durante la navegación. En él, son probados los casos de uso de “Solicitar Ayuda para desplazarse a un destino por medio del plano” y “Navegar”. Para ello, se estableció una ruta corta definida por el usuario en la que eran suficientes dos Waypoints en el desplazamiento. La prueba realizada sobre el plano puede observarse en la Figura 40. En ella, el robot debería desplazarse sin ningún problema desde el origen en el que se encuentra hasta el destino establecido ya que es una distancia muy corta en la que el robot no debería chocarse ni encontrar obstáculos, mostrando por medio de la interfaz la imagen del robot en el plano según va avanzando y emitiendo a su llegada un mensaje de que se ha llegado al destino y dando la posibilidad de volver a la pantalla de Inicio para realizar una nueva navegación. Una vez realizada la prueba se obtienen los resultados mostrados en la Tabla 1.

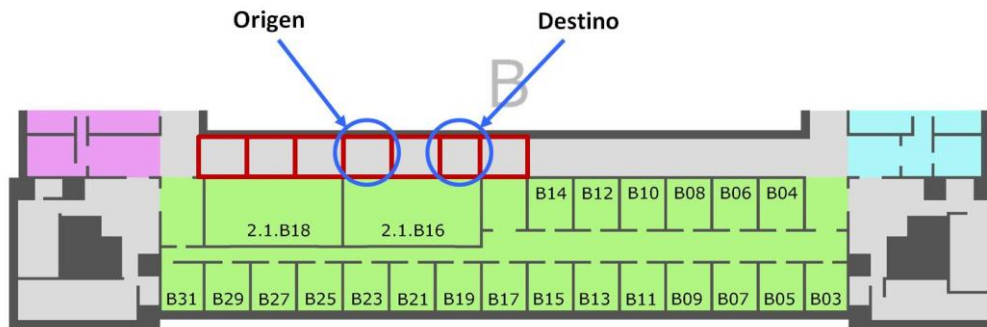


Figura 40. Prueba de Funcionamiento Correcto de la Aplicación Desarrollada

| Aspecto a probar | Resultado |
|--|----------------|
| Distancia teórica recorrida | 8'3 metros |
| Distancia real recorrida | 8'37 metros |
| Tiempo total | 44'39 segundos |
| Número de veces que ha sido necesario Reubicar | 0 |
| Número de errores durante la navegación | 0 |
| El robot llega al destino | Correcto |
| Se admite la realización de una nueva búsqueda | Correcto |

Tabla 1. Resultados de la Prueba de Funcionamiento Correcto de la Aplicación Desarrollada

4.1.2 Interrupción de STOP

Una vez realizadas las pruebas de llegada al destino sin interrupciones por parte del usuario ni errores durante la navegación, se pasó a realizar una segunda prueba en la que se realizaba un desplazamiento corto para llegar al destino sin errores durante la navegación pero en el que se realizaba una interrupción de STOP por parte del usuario durante el desplazamiento del robot. En ella, son probados los casos de uso “Solicitar ayuda para desplazarse a un destino por medio del plano” y “Navegar” con una interrupción durante su realización. Para ello se estableció una ruta larga en la primera selección y que, tras la interrupción, fue modificada quedando la misma ruta que en los casos anteriores pero en la que ha sido pulsado el botón de STOP en el transcurso del desplazamiento del primer Waypoint al segundo. Esta prueba fue realizada para garantizar el correcto funcionamiento de la interrupción de STOP junto con su posterior recuperación, cambio de destino y llegada al mismo en situaciones en las que no se producen errores durante el desplazamiento. La prueba realizada sobre el plano puede observarse en la Figura 41. En ella, el robot debía comenzar a desplazarse y, tras la interrupción de STOP realizada por el usuario, emitir un mensaje de que la interrupción

puede tardar unos minutos y, al llegar al siguiente Waypoint detenerse, dando la posibilidad al usuario de seleccionar un nuevo destino a través del plano o de la búsqueda en el directorio de la universidad. Tras la selección de un nuevo destino por parte del usuario, el robot debía desplazarse hasta el mismo mostrando por medio de la interfaz la imagen del robot en el plano según va avanzando y emitiendo a su llegada un mensaje de que se ha llegado al destino y dando la posibilidad de volver a la pantalla de Inicio para realizar una nueva navegación. Durante el desplazamiento no deben ocurrir errores, ya que la distancia que tiene que recorrer el robot es muy corta y el robot no debería chocarse ni encontrar obstáculos durante la navegación. Una vez realizada la prueba se obtienen los resultados mostrados en la Tabla 2.

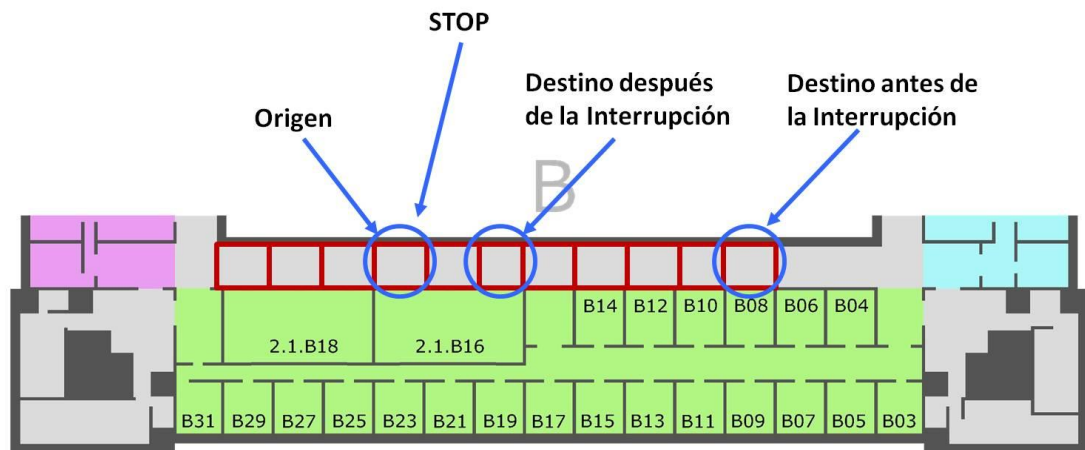


Figura 41. Prueba de Interrupción de STOP de la Aplicación Desarrollada

| Aspecto a probar | Resultado |
|--|----------------|
| Distancia teórica recorrida | 8'3 metros |
| Distancia real recorrida | 8'37 metros |
| Tiempo tratamiento interrupción | 10'05 segundos |
| Tiempo total | 54'57 segundos |
| Número de veces que ha sido necesario Reubicar | 0 |
| Número de errores durante la navegación | 0 |
| Tratamiento interrupción de STOP | Correcto |
| El robot llega al destino | Correcto |

Tabla 2. Resultados de la Prueba de Interrupción de STOP de la Aplicación Desarrollada

4.1.3 Interrupción de Reubicar

La siguiente prueba consistió en un desplazamiento corto sin errores, al igual que en los casos anteriores, pero en el que se produce una interrupción para Reubicar al robot. En ella se probaron los casos de uso “Solicitar ayuda para desplazarse a un destino por medio del plano”, “Corregir localización robot” y “Navegar” con una interrupción de Reubicar durante la misma. En esta prueba se seleccionó un Waypoint distinto al que inicialmente se encuentra el robot, hecho que se corrige nada más empezar el desplazamiento seleccionando el lugar donde se encuentra el robot realmente. Esta prueba se realizó para comprobar el correcto funcionamiento de la interrupción de Reubicar junto con su posterior recuperación, cambio de origen y llegada correcta al destino. La prueba realizada sobre el plano puede observarse en la Figura 42. En ella, el robot debía comenzar a desplazarse y, tras la interrupción de Reubicar realizada por el usuario, emitir un mensaje de que la interrupción puede tardar unos minutos y, al llegar al siguiente Waypoint detenerse, dando la posibilidad al usuario de seleccionar el lugar donde se encuentra a través del plano. Tras la selección del lugar de ubicación por parte del usuario, el robot debía desplazarse hasta el destino mostrando, por medio de la interfaz, la imagen del robot en el plano según va avanzando y emitiendo a su llegada un mensaje de que se ha llegado al destino y dando la posibilidad de volver a la pantalla de Inicio para realizar una nueva navegación. Durante el desplazamiento no deben ocurrir errores, ya que la distancia total que tiene que recorrer el robot es muy corta y el robot no debería chocarse ni encontrar obstáculos durante la navegación. Una vez realizada la prueba se obtienen los resultados mostrados en la Tabla 3. El número de veces que ha sido necesario reubicar no es significativo en este caso, ya que se forzó a su recolocación indicando que el robot se encontraba en un lugar distinto al que se encontraba realmente para probar su correcto funcionamiento tras la reubicación.

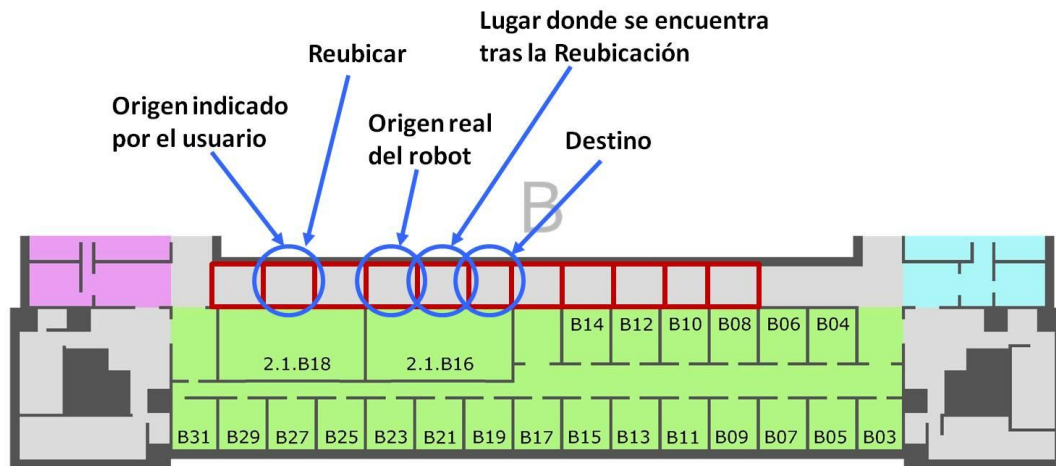


Figura 42. Prueba de Interrupción de Reubicar de la Aplicación Desarrollada

| Aspecto a probar | Resultado |
|--|------------------|
| Distancia teórica recorrida | 8'3 metros |
| Distancia real recorrida | 8'37 metros |
| Tiempo tratamiento interrupción | 12'08 segundos |
| Tiempo total | 56'47 segundos |
| Número de veces que ha sido necesario Reubicar | 1 |
| Número de errores durante la navegación | 0 |
| Tratamiento interrupción de Reubicar | Correcto |
| El robot llega al destino | Correcto |

Tabla 3. Resultados de la Prueba de Interrupción de Reubicar de la Aplicación Desarrollada

4.1.4 Errores durante el Desplazamiento

La siguiente prueba consistió en el desplazamiento del robot en una ruta más larga que las anteriores en la que se da un error pero no existen interrupciones por parte del usuario. En ella son probados los casos de uso “Solicitar ayuda para desplazarse a un destino por medio del plano”, “Solicitar Ayuda” por parte del robot, “Orientar robot” y “Navegar”. Para ello se estableció una ruta en la que el robot tuviera que desplazarse por cinco waypoints, lugar donde se produce un error por ser una distancia larga en la que el robot se termina chocando con alguna pared, tras lo que se produciría la recuperación del error por parte del usuario y posteriormente la correcta llegada al destino. Esta prueba fue realizada para comprobar la correcta gestión de los errores y posterior recuperación de los mismos con finalización en el destino seleccionado. La descripción de esta prueba puede observarse en la Figura 43. En ella, el robot debía comenzar a desplazarse y, tras la navegación por tres Waypoints, se prevé que dará un error al encontrarse con el obstáculo de la pared, momento en el cual indicará esta situación al usuario. Tras la recolocación del robot, la interfaz desarrollada debería solicitar el lugar donde se encuentra el robot al usuario y tras la introducción del mismo, desplazarse hasta el destino mostrando por medio de la interfaz la imagen del robot en el plano según va avanzando y emitiendo a su llegada un mensaje de que se ha llegado al destino y dando la posibilidad de volver a la pantalla de Inicio para realizar una nueva navegación. Una vez realizada la prueba se obtienen los resultados mostrados en la Tabla 4.

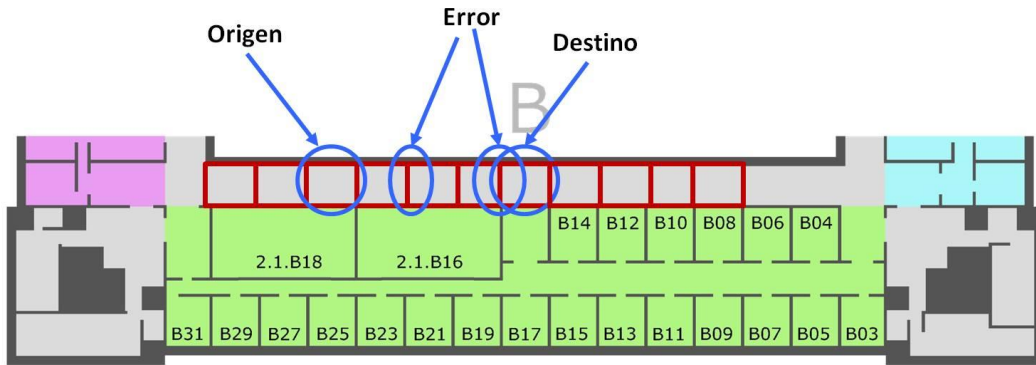


Figura 43. Prueba de Errores durante el Desplazamiento de la Aplicación Desarrollada

| Aspecto a probar | Resultado |
|--|----------------------------|
| Distancia teórica recorrida | 16'66 metros |
| Distancia real recorrida | 19'28 metros |
| Tiempo total tratamiento errores | 1 minuto y 10'4 segundos |
| Tiempo total | 2 minutos y 41'08 segundos |
| Número de veces que ha sido necesario Reubicar | 1 |
| Número de errores durante la navegación | 2 |
| Recuperación del error | Correcto |
| El robot llega al destino | Correcto |

Tabla 4. Resultados de la Prueba de Errores durante el Desplazamiento de la Aplicación Desarrollada

4.1.5 Interrupciones y Errores durante el Desplazamiento

Una vez probadas las interrupciones y errores de forma independiente se realizó una prueba en la que se dieran ambos casos en el mismo problema. En ella son probados los casos de uso “Solicitar ayuda para desplazarse a un destino por medio del plano”, “Solicitar Ayuda” por parte del robot, “Orientar robot” y “Navegar” con errores e interrupciones durante la misma. Para ello se estableció una ruta larga en la que se produce en primer lugar un error, a continuación una interrupción de STOP con cambio de destino y un error de nuevo. Esta prueba fue realizada para comprobar la correcta gestión de interrupciones y errores de forma conjunta con recuperación de ambas y posterior llegada al destino seleccionado. La prueba realizada puede observarse en la Figura 44. En ella, el robot debía comenzar a desplazarse y, tras la navegación por tres

Waypoints, se prevé que dará un error al encontrarse con el obstáculo de la pared, momento en el cual indicará esta situación al usuario. Tras la recolocación del robot, la interfaz desarrollada debería solicitar el lugar donde se encuentra el robot al usuario y tras la introducción del mismo, comenzar a desplazarse hacia el destino. Tras esto, el usuario realizará una interrupción de STOP, momento en el que la interfaz debe dar la posibilidad de elegir un nuevo destino por medio del plano o del directorio de la universidad. Tras la selección del nuevo destino por parte del usuario, el robot debería seguir desplazándose hacia el destino, dando un nuevo error tras el desplazamiento de tres Waypoints, dando la posibilidad al usuario de reubicarlo de nuevo en el pasillo y solicitándole su actual ubicación. Tras su recolocación, el robot se debería desplazar hacia el último destino seleccionado, mostrando por medio de la interfaz la imagen del robot en el plano según va avanzando y emitiendo a su llegada un mensaje de que se ha llegado al destino y dando la posibilidad de volver a la pantalla de Inicio para realizar una nueva navegación. Una vez realizada la prueba se obtienen los resultados mostrados en la Tabla 5.

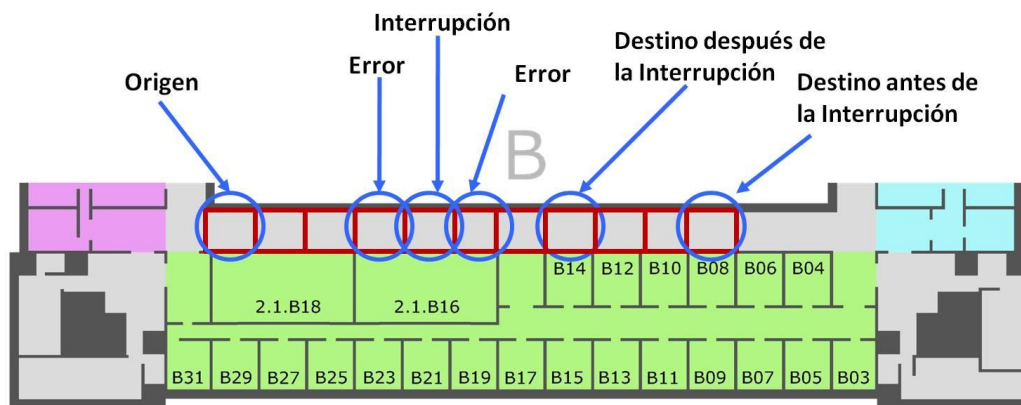


Figura 44. Prueba con Interrupción y Errores durante el Desplazamiento de la Aplicación Desarrollada

| Aspecto a probar | Resultado |
|--|---------------------------|
| Distancia teórica recorrida | 20'8 metros |
| Distancia real recorrida | 26'57 metros |
| Tiempo tratamiento interrupción | 3'6 segundos |
| Tiempo total tratamiento errores | 56'18 segundos |
| Tiempo total | 4 minutos y 8'18 segundos |
| Número de veces que ha sido necesario Reubicar | 1 |
| Número de errores durante la navegación | 2 |
| Tratamiento interrupciones | Correcto |
| Recuperación del error | Correcto |
| El robot llega al destino | Correcto |

Tabla 5. Resultados de la Prueba con Interrupción y Errores durante el Desplazamiento de la Aplicación Desarrollada

4.1.6 Búsqueda con el Directorio

Hasta el momento se han realizado las pruebas empleando la selección de destino a través del plano de la universidad. En esta prueba se realizó la selección de la ruta por medio del directorio para lo cual se llevó a cabo una búsqueda del profesor D. Fernando Fernández Rebollo, cuyo despacho es el 2.1.B19, para comprobar la correcta navegación del robot al mismo. Además, en la ruta seleccionada existen pasillos más estrechos, giros que debe realizar el robot para alcanzar el destino y puertas que dificultan su desplazamiento, por lo que esta prueba servirá tanto para probar el correcto funcionamiento de búsqueda por medio del directorio de la universidad como la navegación del robot en situaciones más complejas. En ella son probados los casos de uso “Solicitar ayuda para desplazarse a un destino por medio del directorio”, “Solicitar Ayuda” por parte del robot, “Orientar robot”, “Corregir localización robot” y “Navegar”. La prueba realizada puede observarse en la Figura 45. En ella, la interfaz debería mostrar los datos de los profesores cuyos datos coinciden con los introducidos y, tras la selección de uno de ellos por parte del usuario, iniciar la navegación hacia el despacho seleccionado. Durante esta navegación se prevé que el robot dará una serie de errores en los que el usuario deberá asistirle para alcanzar el destino seleccionado. Una vez realizada la prueba se obtienen los resultados mostrados en la Tabla 6.

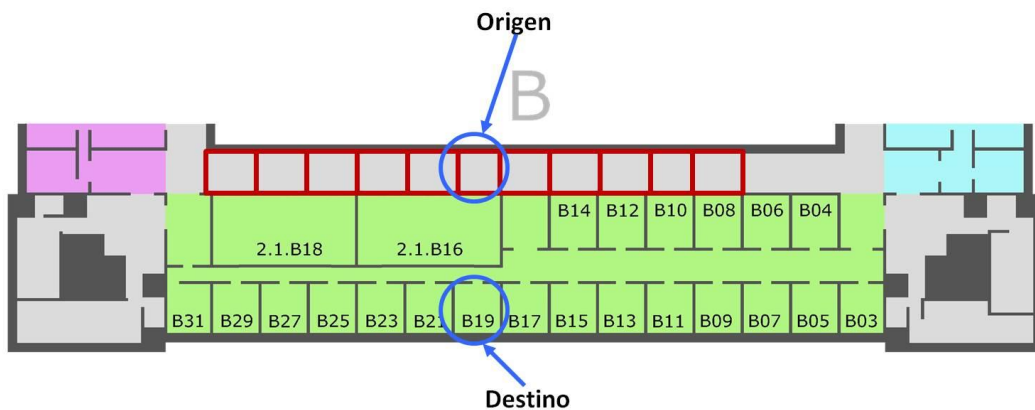


Figura 45. Prueba de Búsqueda de Destino por medio del Directorio de la Aplicación Desarrollada

| Aspecto a probar | Resultado |
|--|---------------------------|
| Distancia teórica recorrida | 16'7 metros |
| Distancia real recorrida | 20'32 metros |
| Tiempo total tratamiento interrupciones | 0 segundos |
| Tiempo total tratamiento errores | 1 minuto y 24'36 segundos |
| Tiempo total | 4 minutos y 57'6 segundos |
| Número de veces que ha sido necesario Reubicar | 1 |
| Número de errores durante la navegación | 3 |
| El robot llega al destino | Correcto |

Tabla 6. Resultados de la Prueba de Búsqueda de Destino por medio del Directorio de la Aplicación Desarrollada

4.2 Evaluación

A continuación se expone la evaluación realizada de la aplicación desarrollada, para lo que se escogieron seis usuarios que tuvieron que realizar una serie de tareas típicas establecidas previamente sobre la interfaz gráfica, tras lo que se les realizaron unas preguntas sobre su opinión sobre la misma. Además, durante el desarrollo de la evaluación se tomaron otras medidas para comprobar la efectividad del sistema. Los objetivos de la evaluación son:

- Facilidad de interacción con la aplicación. Se comprobará si el usuario sabe desenvolverse correctamente con la interfaz, encontrando rápidamente cada una de las opciones de búsqueda, sabiendo introducir los datos en el directorio y colocando al robot en el pasillo por medio de las flechas.
- Comprobar si el usuario sabe interpretar de forma correcta los planos mostrados por pantalla y ubicarse fácilmente en ellos.
- Averiguar si el usuario tarda un tiempo adecuado en realizar las tareas.
- Comprobar si el usuario es capaz de alcanzar el objetivo marcado en la tarea.
- Evaluar si el usuario está satisfecho con el sistema y si lo considera útil como guía en el edificio.
- Obtener comentarios y sugerencias por parte de los evaluadores que puedan servir para ampliaciones futuras del proyecto.

La evaluación se realizó de forma individual por cada usuario, explicándole previamente el objetivo de dicha evaluación y las tareas a realizar.

En primer lugar, los usuarios debían seleccionar desplazarse por medio del plano desde el origen donde se encontraban, que se estableció en el pasillo a la altura del despacho 2.1.B16, hasta el pasillo a la altura del despacho 2.1.C06. Con la realización de esta tarea se comprueba si el usuario sabe elegir la opción de búsqueda por medio del plano y cómo seleccionar la zona y lugar de destino de forma correcta. Asimismo, se comprueba si el usuario sabe asistir al robot cuando éste precise de su ayuda llegando a alcanzar el destino seleccionado. En esta prueba se espera la ocurrencia de dos errores: uno al recorrer la primera parte del pasillo por ser una distancia larga y otro al realizar el giro para desplazarse por el pasillo de la zona C, ya que los giros realizados por el robot acumulan algo de error.

En segundo lugar, partiendo de la puerta del despacho 2.1.B16, los usuarios debían seleccionar el destino por medio de una búsqueda en el directorio de la universidad del profesor D. Daniel Borrajo Millán. Con esta tarea se comprueba si el usuario sabe seleccionar la opción de búsqueda por medio del directorio e introducir de forma correcta los datos del profesor, así como asistir al robot cuando éste precise de su ayuda. En esta prueba se esperan entre cuatro y seis errores, al tratarse de una distancia larga, que implica dos giros y en la que hay que desplazarse en un pasillo con columnas en el centro.

Capítulo 4: Evaluación y Pruebas Preliminares

En último lugar, los usuarios debían seleccionar desplazarse por medio del plano desde el origen donde se encontraban, que se estableció en el pasillo a la altura del despacho 2.1.C06, hasta el final del pasillo de la zona B. Esta prueba fue realizada para comprobar el comportamiento del usuario ante el desplazamiento del robot en una distancia larga, así como para comprobar si el usuario sabe elegir la opción de búsqueda por medio del plano, selección de la zona y lugar de destino de forma correcta y asistencia al robot cuando éste precise de su ayuda. En esta prueba se espera la ocurrencia de entre 6 y 8 errores, por tratarse de un recorrido muy largo con un giro para dirigirse hacia el pasillo de la zona B.

Una vez finalizadas las tareas se hizo entrega de un cuestionario a cada uno de los usuarios para comprobar la evolución en la realización de las tareas y posibles problemas u otros aspectos observados. A continuación se incluyen las preguntas entregadas a los usuarios para la evaluación de la aplicación.

Tarea 1: partiendo de la puerta del despacho 2.1.B16, realizar una petición de desplazamiento al robot hasta la zona del pasillo ubicada a la altura del despacho 2.1.C06.

¿Ha hallado fácilmente la opción de búsqueda por medio del plano? Sí No

¿Los planos presentados han sido intuitivos o ha tenido algún problema en la localización del lugar de desplazamiento? En caso de haber tenido algún problema indique el motivo. Sí No

¿Ha sido necesaria su asistencia al robot en algún momento del desplazamiento?

Sí No

En caso de haber sido necesaria su asistencia al robot, ¿ha podido asistirle con facilidad o por el contrario ha encontrado algún problema durante la asistencia?

Sí, he podido asistirle con facilidad

No, he encontrado problemas durante la asistencia

¿Ha logrado completar la tarea indicada? En caso negativo indique qué problemas ha tenido. Sí No

Tarea 2: partiendo de la puerta del despacho 2.1.B16, realizar una petición de desplazamiento al robot hasta el despacho del profesor D. Daniel Borrajo Millán.

¿Ha hallado fácilmente la opción de búsqueda por medio del directorio? Sí No

La búsqueda del profesor indicado ¿ha dado un resultado satisfactorio? Sí No

¿Ha sido necesaria su asistencia al robot en algún momento del desplazamiento?

Sí No

En caso de haber sido necesaria su asistencia al robot, ¿ha podido asistirle con facilidad o por el contrario ha encontrado algún problema durante la asistencia?

Sí, he podido asistirle con facilidad

No, he encontrado problemas durante la asistencia

¿Ha logrado completar la tarea indicada? En caso negativo indique qué problemas ha tenido. Sí No

Tarea 3: partiendo de la puerta del despacho 2.1.C06, realizar una petición de desplazamiento al robot hasta el final del pasillo de la zona B.

¿Ha hallado fácilmente la opción de búsqueda por medio del plano? Sí No

¿Los planos presentados han sido intuitivos o ha tenido algún problema en la localización del lugar de desplazamiento? En caso de haber tenido algún problema indique el motivo. Sí No

¿Ha sido necesaria su asistencia al robot en algún momento del desplazamiento?

Sí No

En caso de haber sido necesaria su asistencia al robot, ¿ha podido asistirle con facilidad o por el contrario ha encontrado algún problema durante la asistencia?

Sí, he podido asistirle con facilidad

No, he encontrado problemas durante la asistencia

¿Ha logrado completar la tarea indicada? En caso negativo indique qué problemas ha tenido. Sí No

Preguntas generales:

¿Le ha parecido adecuada la interfaz gráfica presentada? En caso negativo indique qué cosas modificaría de la misma. Sí No

¿Considera útil la existencia de un robot guía para personas no familiarizadas con el edificio? En caso negativo, indique el motivo. Sí No

¿Volvería a utilizar aplicación? En caso negativo, indique el motivo. Sí No

¿Qué aspectos modificaría o mejoraría de la aplicación?

Otras observaciones.

Capítulo 4: Evaluación y Pruebas Preliminares

La evaluación fue realizada por seis usuarios que posteriormente realizaron la encuesta. Además, durante el desarrollo de la evaluación se tomaron otras medidas como son el tiempo total para completar la tarea, el número de errores que se produjeron durante la navegación y el número de veces que fue necesario reubicar la posición del robot en el plano. A continuación se muestran los resultados tomados de las tareas realizadas por todos ellos:

| Evaluación | Tarea Realizada | Distancia Recorrida | Tiempo total | Errores durante la navegación | Veces que fue necesario Reubicar | Compleitud de la tarea |
|------------|-----------------|---------------------|------------------|-------------------------------|----------------------------------|------------------------|
| 1 | 1 | 17'37 metros | 05:05'64 minutos | 2 | 0 | Correcto |
| 2 | 1 | 17'37 metros | 06:53'22 minutos | 3 | 0 | Correcto |
| 3 | 2 | 28'55 metros | 07:34'58 minutos | 4 | 2 | Correcto |
| 4 | 2 | 28'55 metros | 08:05'64 minutos | 5 | 2 | Correcto |
| 5 | 3 | 61'92 metros | 14:31'97 minutos | 6 | 3 | Correcto |
| 6 | 3 | 61'92 metros | 16:03'07 minutos | 8 | 3 | Correcto |

Tabla 7. Resultados de las Mediciones de las Evaluaciones Realizadas

A continuación y, tras la realización del cuestionario por todos ellos, se concluyen los siguientes resultados:

- La totalidad de los usuarios ratificaron que habían hallado fácilmente las opciones de búsqueda por medio del plano o por medio del directorio, según la tarea que llevaron a cabo.
- Todos los usuarios que realizaron las tareas 1 y 3 de búsqueda de destino por medio del plano estuvieron de acuerdo en que los planos son intuitivos y ninguno de ellos tuvo ningún problema en su utilización.
- Todos los usuarios que realizaron la tarea 2 de búsqueda de destino por medio del directorio indicaron que la búsqueda del profesor había dado un resultado satisfactorio.
- En la realización de las tres tareas, la totalidad de los usuarios necesitaron asistir al robot durante algún momento de su desplazamiento.
- Durante la asistencia de los usuarios al robot, todos pudieron asistirle con facilidad, aunque uno de ellos remarcó que sería preferible que los movimientos a la hora de centrar al robot en el pasillo tuvieran mayor precisión.
- La totalidad de los usuarios lograron completar la tarea realizada.
- A la totalidad de los usuarios evaluadores les pareció adecuada la interfaz gráfica, recalcando en observaciones que era elegante, amigable, intuitiva y usable.

- De los seis usuarios evaluadores, cinco consideraron que la existencia de un robot guía para personas no familiarizadas con el edificio sería útil, mientras que uno de ellos pensó que no lo era, indicando en este caso que es más útil un mapa o el uso de GPS.
- De los seis usuarios evaluadores, cinco volverían a utilizar la aplicación. La persona que no volvería a usarla indicó que esto se debía a que el robot era muy lento en sus desplazamientos. Debido a los resultados de tiempo obtenidos, se concluye que los usuarios se basaron en la utilidad del sistema como criterio para volver a emplear la aplicación.
- En cuanto a los aspectos a mejorar un usuario indicó que el robot podría esquivar los obstáculos en lugar de pedir ayuda y otro de los evaluadores consideró que se podría buscar un término medio entre la asistencia del usuario y la autosuficiencia haciendo que el robot pudiera proponer por sí mismo las soluciones cuando se encuentra con un obstáculo.

Capítulo 5

Gestión del Proyecto

A continuación se expone la planificación de las diferentes fases y tareas que se han llevado a cabo en el presente proyecto, junto con su correspondiente diagrama de Gantt y el presupuesto del proyecto distinguiendo entre costes de personal, costes de material y costes totales.

El desarrollo del presente proyecto ha supuesto la planificación de cada una de las tareas a realizar y la agrupación de las mismas en fases en las que distinguimos Análisis, Diseño, Implementación, Pruebas y Documentación. Cada una de dichas fases está compuesta por un conjunto de tareas que se agrupan a su vez, en el caso de las fases de Análisis, Diseño e Implementación, en dos grupos principales: Interfaz Localización Hombre-Robot (ILHR), que contiene todas aquellas tareas relativas al desarrollo y construcción de la interfaz gráfica desarrollada, e Interfaz ILHR-ROBOT, que contiene las tareas relativas a la conexión de la interfaz gráfica desarrollada con el sistema de control del robot. De esta forma distinguimos las tareas de la Tabla 8 cuya planificación puede observarse en el diagrama de Gantt de la Figura 46.

| | |
|----|---|
| 1 | Inicio del Proyecto |
| 2 | Análisis |
| 3 | Interfaz Hombre-Robot (ILHR) |
| 4 | Analizar los requisitos de la Interfaz del Plano |
| 5 | Analizar los requisitos de la Interfaz Movimiento Robot |
| 6 | Definir requisitos sistema de voz |
| 7 | Interfaz ILHR-ROBOT |
| 8 | Analizar requisitos de las cadenas XML |
| 9 | Analizar requisitos archivo PDDL |
| 10 | Analizar comunicación por sockets |
| 11 | Analizar requisitos pruebas |

| | |
|----|---|
| 12 | Diseño |
| 13 | Interfaz Hombre-Robot (ILHR) |
| 14 | Diseñar Borrador Interfaz General |
| 15 | Diseño botones Start y Stop |
| 16 | Diseño y simplificación del plano |
| 17 | Interfaz ILHR-ROBOT |
| 18 | Diseño estructura XML para movimiento robot |
| 19 | Diseño estructura XML para navegación robot |
| 20 | Diseño estructura XML para el plano |
| 21 | Diseño estructura PDDL |
| 22 | Diseño pruebas |
| 23 | Implementación |
| 24 | Interfaz Hombre-Robot (ILHR) |
| 25 | Primera versión Interfaz Botones Start y Stop |
| 26 | Primera versión Interfaz Movimiento Robot |
| 27 | Primera versión Interfaz Plano |
| 28 | Versión final Interfaz Movimiento Robot |
| 29 | Versión final Interfaz Plano |
| 30 | Sistema de voz |
| 31 | Interfaz ILHR-ROBOT |
| 32 | Primera versión Sockets |
| 33 | Prototipo Movimiento Robot |
| 34 | Paso cadena con xml de movimiento por sockets |
| 35 | Genero y Recibo ACK y NACK |
| 36 | Versión final Movimiento Robot |
| 37 | Prototipo Localización Robot |
| 38 | XML para las zonas del plano |
| 39 | Codificación generación botones automática a partir de XML |
| 40 | Codificación generación automática de PDDL a partir de XML |
| 41 | Codificación paso de cadena xml de NAVEGACIÓN robot por sockets |
| 42 | Versión final Localización Robot |
| 43 | Pruebas |
| 44 | Realización de pruebas |
| 45 | Informe pruebas |
| 46 | Documentación |
| 47 | Elaboración memoria |
| 48 | Elaboración y preparación presentación |
| 49 | Fin Proyecto |

Tabla 8. División de Tareas del Proyecto

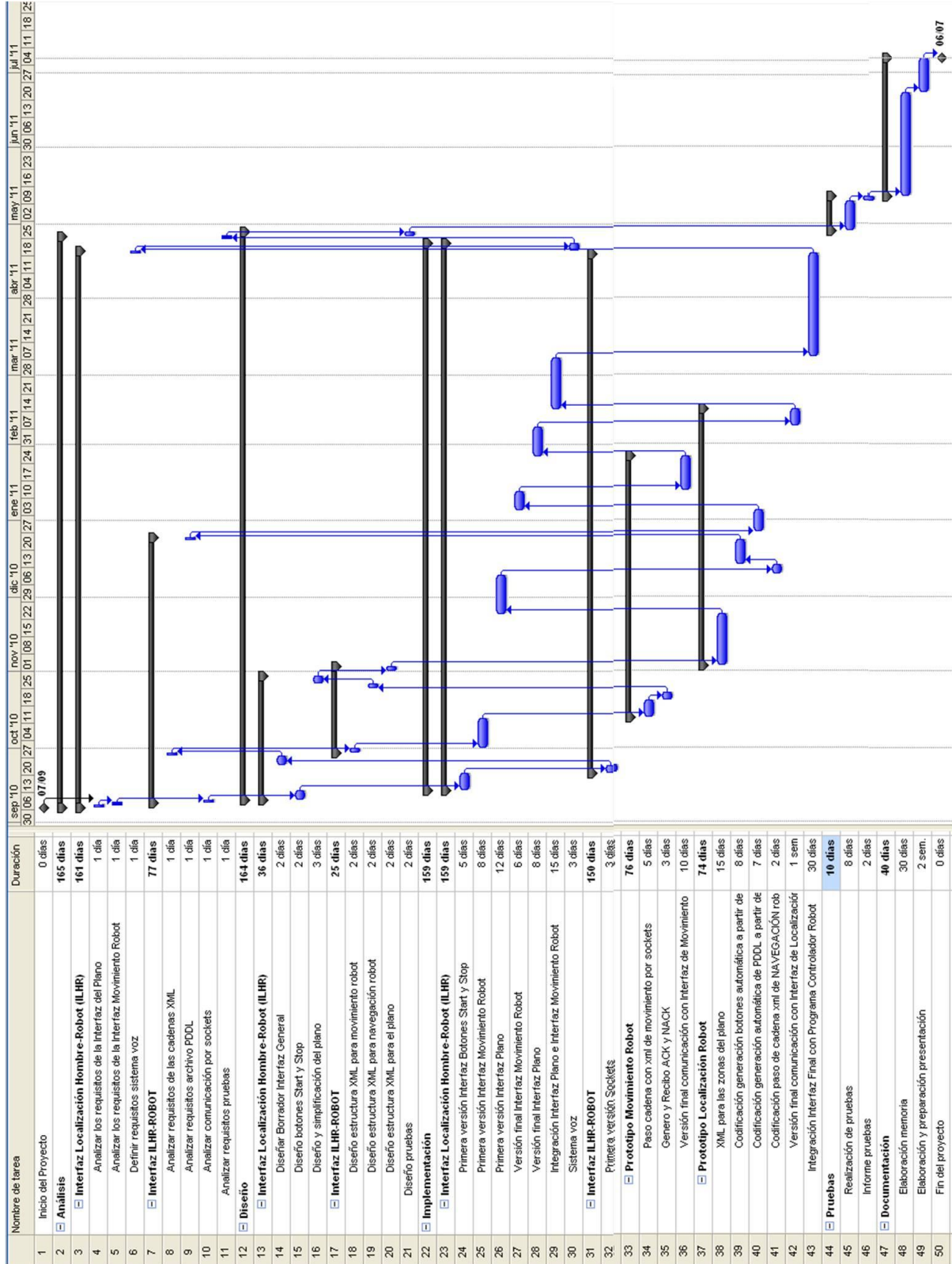


Figura 46. Diagrama de Gantt del Proyecto

Capítulo 5: Gestión del Proyecto

A continuación se presenta el presupuesto del presente proyecto que incluye el desglose de costes directos de personal y materiales y de costes indirectos, así como el coste total del proyecto.

1.- Autor:

Serrano Sánchez-Bravo, Laura

2.- Departamento:

Informática

3.- Descripción del Proyecto:

- Título Desarrollo de una Interfaz Hombre-Robot para un Robot Guía en Interiores

- Duración (meses) 10 meses

Tasa de costes Indirectos: **20%**

4.- Presupuesto total del Proyecto (valores en Euros):

30.783 Euros

5.- Desglose presupuestario (costes directos)

PERSONAL

| Apellidos y Nombre | Categoría | Dedicación (hombres mes) ^{a)} | Coste hombre mes | Coste (Euro) |
|-------------------------------|-----------------------------|--|------------------|------------------|
| Serrano Sánchez-Bravo, Laura | Analista y Programador | 5,76 | 3.806,25 | 21.924,00 |
| Quintero Barrios, Ezequiel A. | Jefe de Proyecto y Analista | 0,4 | 5.250,00 | 2.100,00 |
| Fernández Rebollo, Fernando | Jefe de Proyecto | 0,2 | 5.250,00 | 1.050,00 |
| Hombres mes 9,51 | | | Total | 25.074,00 |

Tabla 9. Desglose Presupuestario del Personal

^{a)} 1 Hombre mes = 131,25 horas. Máximo anual de dedicación de 12 hombres mes (1575 horas)

Máximo anual para PDI de la Universidad Carlos III de Madrid de 8,8 hombres mes (1.155 horas)

EQUIPOS

| Descripción | Coste (Euro) | % Uso dedicado proyecto | Dedicación (meses) | Periodo de depreciación | Coste imputable ^{d)} |
|---------------------|--------------|-------------------------|--------------------|-------------------------|-------------------------------|
| Tablet PC | 900 € | 100 | 9 | 60 | 135,00 € |
| Robot Pioneer P3-DX | 2.954 € | 100 | 9 | 60 | 443,10 € |
| Total | | | | | 578,10 € |

Tabla 10. Desglose Presupuestario de Equipos

^{d)} Fórmula de cálculo de la Amortización:

$$\frac{A}{B} \times C \times D$$

A = nº de meses desde la fecha de facturación en que el equipo es utilizado

B = periodo de depreciación (60 meses)

C = coste del equipo (sin IVA)

D = % del uso que se dedica al proyecto (habitualmente 100%)

6.- Resumen de costes

| Presupuesto Costes Totales | Presupuesto Costes Totales |
|----------------------------|----------------------------|
| Personal | 25.074 € |
| Amortización | 578 € |
| Subcontratación de tareas | 0 € |
| Costes de funcionamiento | 0 € |
| Costes Indirectos | 5.130 € |
| Total | 30.783 € |

Tabla 11. Resumen de costes del Proyecto

El presupuesto total de este proyecto asciende a la cantidad de Treinta Mil Setecientos Ochenta y Tres EUROS.

Leganés a 07 de julio de 2011

El ingeniero proyectista

Fdo. Laura Serrano Sánchez-Bravo

Capítulo 6

Conclusiones y Líneas Futuras

En este apartado se describe cómo se han cubierto los objetivos del proyecto y las posibles líneas de trabajo futuro que podrían realizarse a partir del mismo.

El trabajo presentado en este proyecto ha sido la realización de una interfaz gráfica para la interacción con un robot guía en la planta primera del edificio Sabatini de la Universidad Carlos III de Madrid. Para ello se alcanzaron las siguientes submetas presentadas a continuación.

En primer lugar ha sido necesaria la elaboración de una interfaz gráfica que permitiera al usuario seleccionar las opciones de búsqueda de destino, el lugar de destino en sí mismo y las acciones de movimiento requeridas para la recolocación del robot. Para ello, se han debido de tratar todos los detalles de diseño de la interfaz, la elaboración de un plano para ser mostrado al usuario y la construcción de una estructura que representara las zonas y áreas del plano en el que se desplaza el robot.

En segundo lugar se ha realizado el diseño e implementación de los elementos que permiten la comunicación de la interfaz gráfica desarrollada con el sistema de control basado en planificación automática usado por el robot, siendo necesario el establecimiento de un esquema de comunicación entre ambos y la elaboración de las estructuras de los mensajes intercambiados.

Por último, se realizó la adaptación del código del sistema de control del robot para permitir la comunicación con la interfaz gráfica y para que el comportamiento del robot siguiera las indicaciones dadas por el usuario, así como la gestión de coordenadas y la función de transformación entre el espacio de coordenadas reales y el espacio de coordenadas del mapa realizados para la correcta navegación del robot por las distintas áreas en las que se encuentra dividido el plano.

Con todo esto se ha conseguido que el humano y el robot puedan colaborar con una ayuda mutua que permite al robot guiar al usuario durante su desplazamiento por el edificio, al disponer éste de la información del plano del mismo, y al humano asistir al robot en caso de que éste encuentre un obstáculo o se pierda durante su desplazamiento, al poder percibir elementos del entorno como son los números de despacho. Además, tras la obtención de los resultados obtenidos en la evaluación del proyecto, se concluye que el sistema desarrollado cumple con los requisitos de los usuarios y resulta útil para guiar a las personas en el interior del edificio. Una vez realizados los puntos anteriores puede concluirse que el proyecto ha alcanzado con éxito los objetivos marcados. Como líneas de trabajo futuro pueden ser considerados los siguientes puntos:

- Adaptación del proyecto para que el robot pueda desplazarse por el edificio Sabatini al completo, para lo que sería necesario ampliar la estructura que representa el mapa con todas las plantas siendo comunicadas éstas por medio de los ascensores. También sería necesario adaptar la interfaz gráfica para permitir la selección de las distintas plantas del edificio.
- Adaptación del proyecto para que el robot pueda desplazarse por la completitud del campus, para lo que sería necesario otro robot preparado para poder desplazarse por otro tipo de superficies y con un sistema de localización válido para exteriores.
- Modificación del sistema de voz para que, si el usuario interacciona con la interfaz pulsando en los botones con mayor rapidez que la evolución del mensaje de voz, se permitan interrupciones en el mismo, reproduciendo la siguiente grabación y deteniendo el mensaje de voz anterior.
- Modificación del fichero con la descripción del plano con waypoints más pequeños para obtener una mayor precisión en los desplazamientos.
- Ampliación del proyecto para que sea posible seleccionar la orientación del robot. Para ello sería necesario adaptar la interfaz gráfica para permitir dicha selección y realizar el diseño de los mensajes a enviar al programa controlador del robot con la orientación, así como la implementación de una función que llevara el cálculo de la orientación en la que se encuentra el robot en cada momento.
- Mejora de la interfaz gráfica para reducir los retardos durante su funcionamiento ya que, actualmente, tarda unos segundos en pasar de una pantalla a otra.
- Realización de una aplicación que, por medio de una interfaz gráfica en la que se pudieran introducir planos, se puedan realizar las divisiones del mismo gráficamente y retorne el fichero XML con la información de las zonas y áreas en las que se ha dividido con sus respectivas coordenadas y conexiones. Esta aplicación sería de gran utilidad ya que en este proyecto ha sido necesaria la construcción del fichero XML “a mano” lo cual puede resultar un trabajo largo y tedioso para entornos o edificios más grandes.

Con este proyecto se presenta una forma de interacción humano-robot que supone un avance en la comunicación entre ambos y que se espera siga evolucionando en el futuro.

Bibliografía

Schildt, Herbert. *Java 2: manual de referencia*. (Osborne/McGraw-Hill, 2001, 4ª ed.)

Ceballos Sierra, Francisco Javier. *Java 2: Interfaces gráficas y aplicaciones para Internet*. (Rama, 2008, 3ª ed.)

Colomina, Otto y Cazorla, Miguel Ángel. *Introducción a Player/Stage*. Disponible [Internet]: <<http://rua.ua.es/dspace/bitstream/10045/11427/1/intro-ps.pdf>> [12 de noviembre de 2010]

Perez Paina, Gonzalo F. *Entorno de desarrollo de robots Player/Stage/Gazebo*.

Quintero, Ezequiel, García, Ángel, Borrajo, Daniel y Fernández, Fernando. *Control architecture for autonomous mobile robots with automated planning techniques*. Disponible [Internet]: <<http://scalab.uc3m.es/~ffernand/papers/waf2010.pdf>> [17 de noviembre de 2010]

Stephanie Rosenthal, Joydeep Biswas y Manuela Veloso. *An Effective Personal Mobile Robot Agent Through*. Disponible [Internet]: <<http://www.cs.cmu.edu/~mmv/papers/10aamas-cobot.pdf>> [31 de enero de 2011]

Ollero Baturone, Aníbal. *Robótica: manipuladores y robots móviles*. (MARCOMBO S.A, 2001, 1st ed.), pp. 1-13.

Sánchez Martín FM, Millán Rodríguez F, Salvador Bayarri J, Palou Redorta J, Rodríguez Escovar F, Esquena Fernández S, Villavicencio Mavrich H. *Historia de la robótica: de Arquitas de Tarento al Robot da Vinci. (Parte I)*. Disponible [Internet]: <<http://scielo.isciii.es/pdf/aue/v31n2/original1.pdf>>

Sánchez-Martín FM, Jiménez Schlegl P*, Millán Rodríguez F, Salvador-Bayarri J, Monllau Font V, Palou Redorta J, Villavicencio Mavrich H. *Historia de la robótica: de Arquitas de Tarento al Robot da Vinci. (Parte II)*. Disponible [Internet]: <<http://scielo.isciii.es/pdf/aue/v31n3/v31n3a02.pdf>>

J. González, C. Galindo, J.A. Fernández, J.L. Blanco, A. Muñoz y V. Arévalo. *La silla robótica SENA. Un enfoque basado en la interacción hombre-máquina*. Disponible [Internet]: <<http://recyt.fecyt.es/index.php/RIAI/article/viewArticle/826>>

Rafael Aracil, Carlos Balaguer y Manuel Armada. *Robots de servicio*. Disponible [Internet]: <http://e-archivo.uc3m.es/bitstream/10016/9855/1/robots_balaguer_riai_2008.pdf>

Berrio, Paula. *Diseño y simulación de un robot asistente para personas invidentes*. Universidad Carlos III de Madrid, 2009. Disponible [Internet]: <http://e-archivo.uc3m.es/bitstream/10016/6696/1/MemoriaPFC_Paula_Berrio_Martinez.pdf> [31 de enero de 2011]

Juan Manuel Lucas Cuesta, Rosario Alcázar Prior, Juan Manuel Montero Martínez, Fernando Fernández Martínez, Roberto Barra-Chicote, Luis Fernando D'Haro Enríquez, Javier Ferreiros López, Ricardo de Córdoba Herralde, Javier Macías Guarasa, Rubén San Segundo Hernández y José Manuel Pardo Muñoz. *Desarrollo de un Robot-Guía con Integración de un Sistema de Diálogo y Expresión de Emociones: Proyecto ROBINT*. Disponible [Internet]: <<http://www.sepln.org/revistaSEPLN/revista/40/09p12.pdf>> [31 de enero de 2011]

Disponible [Internet]: <<http://www.gizmodo.es/2010/04/08/robots-guia-en-ciudad-banco-santander-porque-espana-tambien-innova.html>> [15 de marzo de 2011]

Disponible [Internet]: <http://www.fayerwayer.com/2011/02/robots-botones-comienzan-a-trabajar-en-hotel-espanol/?utm_source=feedburner&utm_medium=feed&utm_campaign=Feed:+fayerwayer+%28FayerWayer%29> [18 de marzo de 2011]

Infantes, Guillaume, Ghallab, Malik e Ingrand, Félix. *Learning the behavior model of a robot*. Disponible [Internet]: <<http://www.autonomousrobotsblog.com/learning-behavioral-models/>> [18 de marzo de 2011]

Disponible [Internet]: <http://actualidad.rt.com/ciencia_y_tecnica/electronica_tecnologia/issue_21912.html> [25 de abril de 2011]

Holly A. Yanco. *Wheelesley, a Robotic Wheelchair System: Indoor Navigation and User Interface*. Disponible [Internet]: <http://www.cs.uml.edu/~holly/papers/ai-at-yanco.pdf> [25 de abril de 2011]

Infantes, Guillaume, Ghallab, Malik e Ingrand, Félix. *Learning the behavior model of a robot*. Disponible [Internet]: <<http://www.autonomousrobotsblog.com/learning-behavioral-models/>> [30 de junio de 2011]

Bibliografía

Glosario

| | |
|-------|--|
| IDE | <i>Integrated Development Environment</i> |
| IP | <i>Internet Protocol</i> |
| JNDI | <i>Java Naming and Directory Interface</i> |
| LDAP | <i>Lightweight Directory Access Protocol</i> |
| PDDL | <i>Planning Domain Definition Language</i> |
| ROPEM | <i>RObot Plan Execution with Monitoring</i> |
| TCP | <i>Transmission Control Protocol</i> |
| XML | <i>eXtensible Markup Language</i> |

Referencias

- [1] Rafael Aracil, Carlos Balaguer y Manuel Armada. *Robots de servicio*. Disponible [Internet]: http://e-archivo.uc3m.es/bitstream/10016/9855/1/robots_balaguer_riai_2008.pdf
- [2] Disponible [Internet]: <http://www.gizmodo.es/2010/04/08/robots-guia-en-ciudad-banco-santander-porque-espana-tambien-innova.html> [15 de marzo de 2011]
- [3] Disponible [Internet]: http://www.fayerwayer.com/2011/02/robots-botones-comienzan-a-trabajar-en-hotel-espanol/?utm_source=feedburner&utm_medium=feed&utm_campaign=Feed:+fayerwayer+%28FayerWayer%29 [18 de marzo de 2011]
- [4] Disponible [Internet]: http://actualidad.rt.com/ciencia_y_tecnica/electronica_tecnologia/issue_21912.html [25 de abril de 2011]
- [5] Quintero, Ezequiel, García, Ángel, Borrajo, Daniel y Fernández, Fernando. *Control architecture for autonomous mobile robots with automated planning techniques*. Disponible [Internet]: <http://scalab.uc3m.es/~ffernand/papers/waf2010.pdf> [17 de noviembre de 2010]
- [6] Stephanie Rosenthal, Joydeep Biswas y Manuela Veloso. *An Effective Personal Mobile Robot Agent Through*. Disponible [Internet]: <http://www.cs.cmu.edu/~mmv/papers/10aamas-cobot.pdf> [31 de enero de 2011]
- [7] J. González, C. Galindo, J.A. Fernández, J.L. Blanco, A. Muñoz y V. Arévalo. *La silla robótica SENA. Un enfoque basado en la interacción hombre-máquina*. Disponible [Internet]: <http://recyt.fecyt.es/index.php/RIAI/article/view/826/562> [31 de enero de 2011]
- [8] Berrio, Paula. *Diseño y simulación de un robot asistente para personas invidentes*. 2009. Disponible [Internet]: http://e-archivo.uc3m.es/bitstream/10016/6696/1/MemoriaPFC_Paula_Berrio_Martinez.pdf [31 de enero de 2011]
- [9] Juan Manuel Lucas Cuesta, Rosario Alcázar Prior, Juan Manuel Montero Martínez, Fernando Fernández Martínez, Roberto Barra-Chicote, Luis Fernando D'Haro Enríquez,

Referencias

Javier Ferreiros López, Ricardo de Córdoba Herralde, Javier Macías Guarasa, Rubén San Segundo Hernández y José Manuel Pardo Muñoz. *Desarrollo de un Robot-Guía con Integración de un Sistema de Diálogo y Expresión de Emociones: Proyecto ROBINT*. Disponible [Internet]: <http://www.sepln.org/revistaSEPLN/revista/40/09p12.pdf> [31 de enero de 2011]

[10] Holly A. Yanco. *Wheelesley, a Robotic Wheelchair System: Indoor Navigation and User Interface*. Disponible [Internet]: <http://www.cs.uml.edu/~holly/papers/ai-at-yanco.pdf> [25 de abril de 2011]

[11] Infantes, Guillaume, Ghallab, Malik e Ingrand, Félix. *Learning the behavior model of a robot*. Disponible [Internet]: <http://www.autonomousrobotsblog.com/learning-behavioral-models/> [30 de junio de 2011]

Anexo A

Descripción Textual Casos Uso

| | |
|-------------------------|---|
| Nombre | Solicitar ayuda para desplazarse a un destino por medio del plano |
| Actores | Humano |
| Objetivo | Solicitar al robot ayuda para desplazarse a un lugar determinado indicándole el destino por medio de un plano |
| Precondiciones | ---- |
| Postcondiciones | Comunicación al programa controlador del robot de desplazamiento hacia destino |
| Escenario básico | <ol style="list-style-type: none">1. La aplicación muestra un plano del edificio completo y solicita al usuario que seleccione la zona a la que desea desplazarse2. El actor selecciona la zona del plano a la que desea desplazarse3. La aplicación muestra la zona seleccionada ampliada y solicita al usuario que elija el lugar de esa zona al que desea desplazarse4. El actor selecciona el lugar de la zona a la que desea desplazarse5. La aplicación comunica al programa controlador del robot el destino al que debe desplazarse |

Tabla 12. Descripción Textual Caso de Uso Solicitar Ayuda para desplazarse a un destino por medio del plano

| | |
|------------------------------|--|
| Nombre | Solicitar ayuda para desplazarse a un destino por medio del directorio |
| Actores | Humano |
| Objetivo | Solicitar al robot ayuda para desplazarse a un lugar determinado indicándole el destino por medio de los datos del profesor al que se desea visitar |
| Precondiciones | ---- |
| Postcondiciones | Comunicación al programa controlador del robot de desplazamiento hacia destino |
| Escenario básico | <ol style="list-style-type: none"> 1. La aplicación muestra un buscador en el que introducir el nombre y/o apellidos del profesor a visitar. 2. El actor introduce los datos del profesor al que desea visitar y pulsa en el botón buscar. 3. La aplicación muestra el nombre y apellidos del profesor o profesores que coinciden con los datos introducidos por el actor, junto con su teléfono, email y despacho 4. El usuario selecciona, entre los mostrados, el profesor al que desea visitar 5. La aplicación comunica al programa controlador del robot el destino al que debe desplazarse |
| Escenario alternativo | <ol style="list-style-type: none"> 3. Los datos introducidos por el actor no coinciden con ningún profesor <ol style="list-style-type: none"> 3.1 La aplicación indica que los datos introducidos no coinciden con los de ningún profesor |
| Escenario alternativo | <ol style="list-style-type: none"> 3. Los datos introducidos por el actor arrojan demasiados resultados <ol style="list-style-type: none"> 3.1 La aplicación indica que los datos introducidos no acotan lo suficiente la búsqueda |

Tabla 13. Descripción Textual Caso de Uso Solicitar Ayuda para desplazarse a un destino por medio del directorio

Anexo A: Descripción Textual Casos Uso

| | |
|-------------------------|--|
| Nombre | Corregir localización robot |
| Actores | Humano |
| Objetivo | Indicar el lugar de ubicación real del robot en caso de que éste se encuentre equivocado en cuanto a su localización |
| Precondiciones | ---- |
| Postcondiciones | El programa controlador del robot corrige su ubicación |
| Escenario básico | <ol style="list-style-type: none"> 1. El actor pulsa el botón Reubicar 2. La aplicación muestra el plano completo de la planta del edificio y solicita al usuario que seleccione la zona donde se encuentra 3. El actor selecciona la zona del plano donde se encuentra 4. La aplicación muestra la zona del plano seleccionada ampliada y solicita al usuario que seleccione el lugar donde se encuentra 5. El actor selecciona el lugar del plano donde se encuentra el robot |

Tabla 14. Descripción Textual Caso de Uso Corregir Localización Robot

| | |
|-------------------------|--|
| Nombre | Orientar robot |
| Actores | Humano |
| Objetivo | Orientar al robot centrado en el pasillo y encaminado hacia la dirección de destino |
| Precondiciones | El robot ha solicitado ayuda debido a que se ha chocado, se ha perdido o ha encontrado un obstáculo durante su desplazamiento |
| Postcondiciones | El programa controlador del robot corrige la orientación del robot en el pasillo |
| Escenario básico | <ol style="list-style-type: none"> 1. La aplicación solicita ayuda al usuario 2. El actor pulsa el botón Ayudar 3. La aplicación muestra por pantalla cuatro flechas para la orientación del robot y solicita que se centre al robot en el pasillo y orientado en la dirección de destino 4. El actor pulsa en las flechas para orientar al robot 5. El actor pulsa el botón OK |

Tabla 15. Descripción Textual Caso de Uso Orientar Robot

| | |
|-------------------------|--|
| Nombre | Solicitar ayuda |
| Actores | Robot |
| Objetivo | El robot solicita ayuda porque se ha perdido, se ha chocado o ha encontrado un obstáculo durante su desplazamiento |
| Precondiciones | El robot se ha perdido, se ha chocado o ha encontrado un obstáculo durante su desplazamiento |
| Postcondiciones | El robot ha solicitado ayuda |
| Escenario básico | <ol style="list-style-type: none"> 1. El programa controlador del robot envía a la aplicación un mensaje indicando que necesita ayuda 2. La aplicación muestra por pantalla e indica por voz que el robot necesita ayuda |

Tabla 16. Descripción Textual Caso de Uso Solicitar Ayuda

Anexo A: Descripción Textual Casos Uso

| | |
|-------------------------|--|
| Nombre | Navegar |
| Actores | Robot |
| Objetivo | Desplazamiento del robot hacia el destino solicitado |
| Precondiciones | Destino seleccionado por el usuario |
| Postcondiciones | El robot se desplaza hacia el destino |
| Escenario básico | <ol style="list-style-type: none">1. La aplicación envía al programa controlador del robot un mensaje indicando la acción que debe llevar a cabo el robot2. El actor realiza el movimiento indicado por la aplicación |

Tabla 17. Descripción Textual Caso de Uso Navegar

Anexo B

Diagrama de Clases

A continuación se muestra el diagrama de clases completo del sistema en el que aparecen todas las clases implementadas en el presente proyecto junto con sus atributos, conexiones y cardinalidades con el resto de clases.

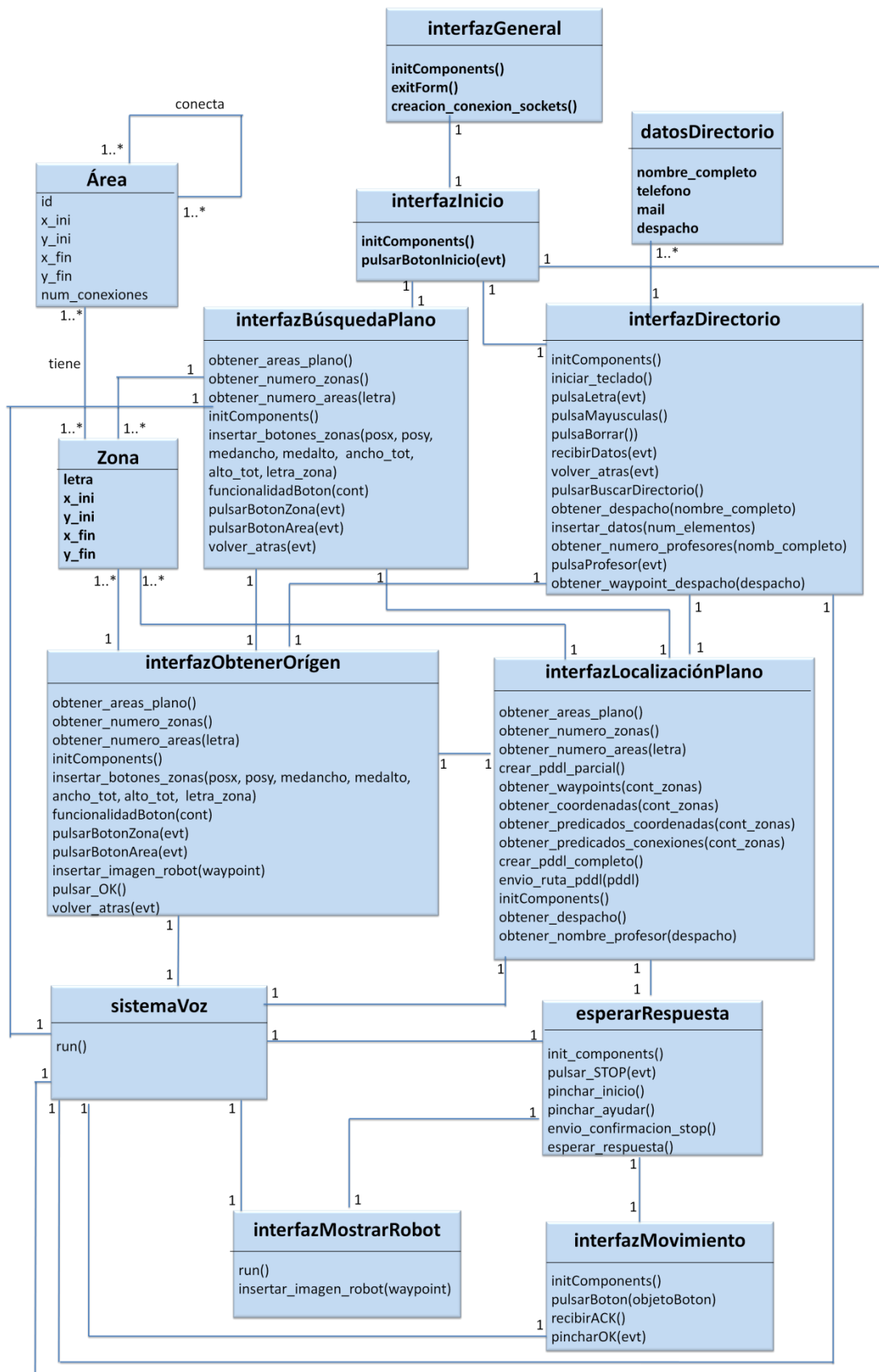


Figura 47. Diagrama de Clases

Anexo B: Diagrama de Clases

Anexo C

Ejemplo XML con la Descripción del Plano

```
<?xml version="1.0"?>
<campus>
  <edificio numero="2">
    <zona letra="D" x_ini="0" y_ini="0" x_fin="806" y_fin="184">
      <area id="0" x_ini="0" y_ini="0" x_fin="151" y_fin="49"
        descripcion="oficinas">
        <conexion>2</conexion>
      </area>
      <area id="1" x_ini="0" y_ini="49" x_fin="50" y_fin="125"
        descripcion="secretaria">
        <conexion>2</conexion>
      </area>
      <area id="2" x_ini="49" y_ini="49" x_fin="151" y_fin="142"
        descripcion="pasillo">
        <conexion>0</conexion>
        <conexion>1</conexion>
        <conexion>5</conexion>
        <conexion>245</conexion>
      </area>
    </zona>
  </edificio>
</campus>
```

Anexo C: Ejemplo XML con la Descripción del Plano

Anexo D

Ejemplo Dominio, Problema y Solución PDDL

Ejemplo de Dominio PDDL:

```
(define (domain SimpleNavigation)
  (:requirements :typing)
  (:types robot waypoint store camera mode lander objective coord)

  (:predicates (at ?x - robot ?y - waypoint)
               (coordinates ?w - waypoint ?x - int ?y - int ?tamx - int ?tamy - int)
               (can_traverse ?r - robot ?x - waypoint ?y - waypoint)
  )

  (:action navigate
   :parameters (?x - robot ?y - waypoint ?z - waypoint)
   :precondition (and (can_traverse ?x ?y ?z) (at ?x ?y)
  )
   :effect (and (not (at ?x ?y)) (at ?x ?z)
  )
  )
)
```

Anexo D: Ejemplo Dominio, Problema y Solución PDDL

Ejemplo de Problema PDDL:

```
(define (problem simple) (:domain SimpleNavigation)
 (:objects
  pela - robot
  waypoint1 waypoint2 waypoint3 waypoint4 - Waypoint
  1551 7476 402 357 1963 420 2378 411 2799 429 - coord
 )
 (:init
  (coordinates waypoint1 1551 7476 402 357)
  (coordinates waypoint2 1963 7476 420 357)
  (coordinates waypoint3 2378 7476 411 357)
  (coordinates waypoint4 2799 7476 429 357)
  (can_traverse pela waypoint1 waypoint2)
  (can_traverse pela waypoint2 waypoint1)
  (can_traverse pela waypoint2 waypoint3)
  (can_traverse pela waypoint3 waypoint1)
  (can_traverse pela waypoint3 waypoint4)
  (can_traverse pela waypoint4 waypoint3)
 (at pela waypoint1)
 )
 (:goal (at pela waypoint4))
 )
```

Solución del problema dada por el planificador:

```
domain loaded!
problema.pddl problem loaded!
Solution:
0: (NAVIGATE PELA WAYPOINT1 WAYPOINT2) [1.000]
1: (NAVIGATE PELA WAYPOINT2 WAYPOINT3) [1.000]
2: (NAVIGATE PELA WAYPOINT3 WAYPOINT4) [1.000]

#<SOLUTION: T Length: 3 Nodes: 3 Evaluated: 2 Depth: 1 >
Total Time      : 0.004
Instantiating Time: 0.0
Search Time     : 0.004

Metric         : NIL
Total Cost     : 3
```


Anexo E

Mensajes XML intercambiados

- Confirmación desplazamiento a Nuevo Waypoint:

```
<atom name="ACK">  
  <term name="WAYPOINTXX"></term>  
</atom>
```

- Error en el desplazamiento:

```
<atom name="NACK">  
  <term name="WAYPOINTXX"></term>  
</atom>
```

- Realiza Movimiento Avanzar:

```
<action-plan name="GO" index="1">  
  <term name="ROBOTX"></term>  
</action-plan>
```

- Confirmación Movimiento Avanzar:

```
<atom name="ACK">  
  <accion name="GO"/>  
</atom>
```

- Error en el Movimiento Avanzar:

```
<atom name="NACK">  
  <accion name="GO"/>  
</atom>
```

- Realiza Movimiento Retroceder:

```
<action-plan name="BACK" index="1">  
  <term name="ROBOTX"></term>  
</action-plan>
```

- Confirmación Movimiento Retroceder:

```
<atom name="ACK">  
  <accion name="BACK"/>  
</atom>
```

- Error en el Movimiento Retroceder:

```
<atom name="NACK">  
  <accion name="BACK"/>  
</atom>
```

- Realiza Movimiento Girar hacia la Derecha:

```
<action-plan name="RIGHT" index="1">  
  <term name="ROBOTX"></term>  
</action-plan>
```


Anexo E: Mensajes XML intercambiados

- Confirmación Movimiento Girar hacia la Derecha:

```
<atom name="ACK">  
  <accion name="RIGTH"/>  
</atom>
```

- Error en el Movimiento Girar hacia la Derecha:

```
<atom name="NACK">  
  <accion name="RIGHT"/>  
</atom>
```

- Realiza Movimiento Girar hacia la Izquierda:

```
<action-plan name="LEFT" index="1">  
  <term name="ROBOTX"></term>  
</action-plan>
```

- Confirmación Movimiento Girar hacia la Izquierda:

```
<atom name="ACK">  
  <accion name="LEFT"/>  
</atom>
```

- Error en el Movimiento Girar hacia la Izquierda:

```
<atom name="NACK">  
  <accion name="LEFT"/>  
</atom>
```

- Confirmación Final de la Ayuda:

```
<action-plan name="OK" index="1">  
  <term name="ROBOTX"></term>  
</action-plan>
```

- Confirmación de que no ha habido interrupción:

```
<action-plan name="OK" index="1">  
    <term name="ROBOTX"></term>  
  
</action-plan>
```

- Interrupción:

```
<action-plan name="STOP" index="1">  
    <term name="ROBOTX"> </term>  
  
</action-plan>
```