



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

SISTEMA DE APOYO PARA DESPLAZAMIENTOS COMPLEJOS EN
AMBIENTES CERRADOS

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN
COMPUTACIÓN

FRANCISCO JAVIER VÁSQUEZ CISTERNA

PROFESOR GUÍA:

LUIS ALBERTO GUERRERO BLANCO

MIEMBROS DE LA COMISIÓN:

MARÍA CECILIA RIVARA ZÚÑIGA

JAVIER ALEJANDRO BUSTOS JIMÉNEZ

SANTIAGO DE CHILE
ENERO 2011

Resumen

El desplazamiento en entornos cerrados y complejos se puede convertir en una tarea de gran dificultad para las personas con discapacidad visual. El trabajo de título presentado a continuación se centra en la creación de un sistema que ayude a las personas con discapacidad visual a orientarse en entornos cerrados.

El sistema se basa en la utilización de herramientas existentes, que permiten generar una solución que esté al alcance de los usuarios finales. En particular, se usó el control de la consola Nintendo Wii para obtener los datos asociados a la posición del usuario. Estos datos fueron comparados con la información del entorno, permitiendo entregar al usuario información apropiada a través de audio.

Los datos de audio pueden ser entregados al usuario en dos modalidades. A través de parlantes conectados a un computador y a través de un dispositivo móvil, el cual a través de sus parlantes entrega la información al usuario. Para seleccionar el modo de entrega de información se desarrolló una interfaz gráfica que permite a un administrador realizar estos cambios. La interfaz también permite realizar modificaciones a otros parámetros de ejecución, de acuerdo a las necesidades del usuario.

Además, se dotó de una nueva funcionalidad al bastón guía que usan las personas con discapacidad visual para ubicarse, de tal forma que a través de un botón el usuario pueda entregar automáticamente los datos de su posición al sistema, permitiendo así a éste determinar en qué momento desea recibir información de parte del sistema.

Por último, se realizaron pruebas que permitieron determinar el correcto funcionamiento de la aplicación, con lo que se comprobó la utilidad de la herramienta.

Agradecimientos

“Teddy no manifestó jamás el más mínimo agradecimiento por lo que recibía. Claro que nadie esperaba que lo hiciera, pues, al fin y al cabo, no era más que un perro.”

Normalmente esperamos que los seres humanos muestren más gratitud que los animales. Lamentablemente, en ocasiones nos sentimos decepcionados al ver que no se cumplen nuestras expectativas. Que grato, por tanto, es poder manifestar gratitud, especialmente cuando se ve en ello una muestra sincera de agradecimiento por aquellas personas que nos han acompañado en esos momentos decisivos de la vida, que quedarán grabados en el recuerdo.

Es por eso que en esta página quiero tratar de dejar ese sentimiento impreso, el cual, aunque el tiempo borre de estas páginas, perdurará en mi corazón.

Agradezco en primer lugar a Dios, por brindarme la oportunidad de este momento, de la vida y de conocerlo. En segundo lugar a mis padres, de los cuales podría haber escrito páginas y páginas, pero que debo resumirlas a un solo concepto, al amor que siento por ellos. A mis hermanos, con quienes he compartido toda una vida y que ahora extraño a la distancia y, siendo sincero, también en silencio.

Agradezco también a mis queridos amigos, con quienes he compartido este momento y también con aquellos que el tiempo y la distancia han alejado, pero que no han borrado de mi corazón. En particular agradezco a Felipe, por esa idea tan inspiradora que merece ser recordada. A su querida vidita, a los ñoñis y a Marcelo, por haber mostrado esa disposición tan noble conmigo. A Natan, que estuvo presto a ayudarme y me hizo ver que el tiempo y la distancia se reducen a nada si hay amistad de por medio. A Cristhian, por su permanente preocupación por mi bienestar y a su esposa, por la ayuda brindada para hacer de esta ensalada de ideas algo entendible.

Agradezco por supuesto, al profesor Luis Guerrero, quien estuvo siempre dispuesto a ayudar y a mostrar consideración en esos momentos de duda.

ESTE TRABAJO FUE PARCIALMENTE APOYADO POR EL FONDO NACIONAL DE
DESARROLLO CIENTÍFICO Y TECNOLÓGICO (FONDECYT), PROYECTO N° 1090352

Índice General

1. Introducción	1
1.1. Conceptos Básicos	2
2. Especificación del Problema	3
2.1. Motivación	3
2.2. Descripción del problema	4
2.3. Requisitos de la solución	5
2.3.1. Requerimientos generales de usuario	5
2.3.2. Requerimientos del modelo genérico de navegación en entornos cerrados	7
2.3.3. Consideraciones a los requerimientos presentados	8
2.4. Criterios de Aceptación de la solución	8
3. Antecedentes	11
3.1. Trabajo relacionado	11
3.2. Navegación en ambientes cerrados	12
3.3. Modelo de Diseño para navegación en ambientes cerrados	13
3.3.1. Entorno de trabajo	14
3.4. El control Wiimote	15
4. Descripción de la Solución	18
4.1. Descripción general de la solución	18
4.2. Librería WiimoteLib	19
4.2.1. Adaptación de la librería	20
4.3. Interfaz usuaria	21
4.3.1. Interfaz para la interacción de la persona con discapacidad visual y el sistema	22
4.3.2. Interfaz de administración y pruebas	25

4.4.	Generación de la información del entorno	28
4.5.	Reconocimiento de la posición del usuario	29
4.5.1.	El proceso de triangulación	30
4.5.2.	Restricciones del algoritmo presentado	33
4.6.	Comunicación	34
4.6.1.	Transmisión de la información	34
4.6.2.	Comunicación entre los componentes del sistema	35
4.7.	Diseño de clases y programación	35
4.7.1.	Comentarios generales sobre la programación	35
4.7.2.	Diagrama de Clases	36
5.	Validación de la Solución	40
5.1.	Datos generales del grupo de pruebas	40
5.1.1.	Comentarios generales del entorno de pruebas	40
5.2.	Resultados	41
5.2.1.	Resultados sin hacer uso del dispositivo móvil	41
5.2.2.	Resultados haciendo uso del dispositivo móvil	44
6.	Conclusiones	46
	Bibliografía	48
	Apéndices	50
A .	Interfaz de Administración	50
B .	Documentos XML para representar entornos cerrados	52
B .1.	Esquema para los documentos XML	52
B .2.	Ejemplo de documento XML	54
C .	Códigos Fuente	58
C .1.	Cálculo de la posición del usuario	58
C .2.	Método para actualizar posición del usuario	59
D .	Resultados de las pruebas realizadas	62
D .1.	Sistema transmitiendo datos a través de parlantes	62
D .2.	Transmisión de datos usando dispositivo móvil	66

Índice de figuras

3.1. Entorno de trabajo para un sistema de navegación indoor.	15
3.2. (Arriba) Vista superior del Wiimote. (Abajo) Vista frontal con cámara infrarroja del Wiimote en primer plano.	16
3.3. Representación de dos puntos a los que el Wiimote les asocia el mismo valor (pertenecen al mismo ángulo) pero que no tienen el mismo valor para otro tipo de eje genérico.	17
4.1. Representación del área visible y relación con programa de prueba.	19
4.2. Programa de prueba en el instante inicial	20
4.3. Programa de prueba después de mover a la esquina inferior izquierda el LED	21
4.4. Vista superior de la adaptación realizada al bastón. Se aprecia que la posición de los emisores infrarrojos permiten emitir luz infrarroja en todas direcciones.	25
4.5. Vista lateral de la adaptación realizada al bastón. Se observa el switch de color rojo que permite al usuario enviar su posición al sistema cuando este lo requiera.	26
4.6. Vista de general de la interfaz de administración.	27
4.7. Relación entre los objetos definidos en la estructura XML.	29
4.8. Representación gráfica del proceso de triangulación.	32
4.9. Interacción del sistema sin usar el dispositivo móvil.	35
4.10. Interacción del sistema usando el dispositivo móvil.	36
4.11. Representación gráfica del proceso de triangulación.	39
5.1. Imagen de las pruebas realizadas.	41
5.2. Descripción de resultados generales sin uso de un dispositivo móvil.	42
5.3. Resumen de los resultados sin hacer uso de un dispositivo móvil.	43

5.4. Descripción de resultados generales haciendo uso de un dispositivo móvil.	45
5.5. Resumen de los resultados haciendo uso de un dispositivo móvil.	45
6.1. Vista de la sección de entrega de parámetros de la interfaz de administración.	50
6.2. Mapa de habitación desplegado en la interfaz de administración.	51

Capítulo 1

Introducción

El desplazamiento en entornos cerrados es una tarea cotidiana para la mayoría de la gente, sin embargo, para personas con discapacidad visual se puede convertir en un verdadero reto. Este trabajo de título expone una solución que ayuda a las personas con discapacidad visual a desplazarse en entornos cerrados.

En un mundo agitado, a menudo se muestra desconsideración por las personas que tienen circunstancias especiales, por lo que es fundamental dotarlos de herramientas que faciliten su integración en la sociedad. A pesar de que se han construido soluciones que abordan el problema del desplazamiento en entornos cerrados, en el caso de las personas con discapacidad visual es fundamental considerar dentro de cualquier solución el factor económico y social. Económico porque, en muchos casos, las soluciones generadas son tan costosas de implementar que, finalmente, no llegan a los usuarios finales. Social, puesto que también se debe tomar en cuenta el entorno de la persona, el cual en muchos limita los accesos a nuevas tecnologías. Es por esta razón que aquí se abordará el problema usando herramientas conocidas, accesibles y de bajo costo.

La solución contempla el uso de los controles de la consola Nintendo Wii, conocidos como Wiimote, los cuales permiten capturar la información asociada a la posición del usuario. Teniendo esto, es posible generar información precisa de la relación espacial del usuario con su entorno, la cual puede ser comunicada al usuario a través de un sistema de audio, en el que se transmite con palabras la información generada.

Para la recolección de los datos del entorno se creó un modelo de documento XML que permite almacenar la información necesaria asociada al entorno. Este modelo fue encapsu-

lado en un esquema que permite al sistema corroborar la correctitud de la forma en la que los datos son almacenados en el documento XML.

Además de presentar estos datos se construyó una interfaz no convencional, dotando de nuevas funcionalidades al bastón guía usado por las personas con discapacidad visual, para que, de esa forma, puedan interactuar con el sistema creado.

El sistema fue probado de dos formas. En la primera se transmite la información a través de parlantes conectados al computador, mientras que en la segunda se transmite la información a un dispositivo móvil, el cual se encarga de entregar la información al usuario a través de audio también. Ambas pruebas fueron realizadas con el mismo grupo de usuarios, de forma que se puedan contrastar ambas alternativas propuestas.

Las pruebas realizadas con la aplicación ponen de manifiesto su utilidad, permitiendo a las personas con discapacidad visual orientarse y reconocer su entorno de forma más rápida y segura.

1.1. Conceptos Básicos

Se presenta a continuación una serie de conceptos que permitirán seguir la lectura del documento.

Desplazamientos Complejos: Se refiere a aquellos desplazamientos realizados por personas ciegas o con visibilidad reducida, en los que se requiere de una ayuda externa. Ejemplos: sortear obstáculos, subir escaleras, entre otros.

Wiimote: Control Remoto creado por Nintendo y utilizado en su consola Wii. Contiene una cámara infrarroja que permite detectar la posición de emisores infrarrojos, los cuales, ubicados convenientemente, ayudan a determinar la posición de otros objetos o personas inclusive.

Dispositivo Móvil: Celular, PDA u otro dispositivo que permita interactuar con el usuario, sin que éste vea limitadas sus posibilidades de desplazamiento de una forma considerable.

Capítulo 2

Especificación del Problema

Antes de determinar una solución adecuada asociada a un problema existente, es fundamental ser capaces de definir de forma correcta el problema. A continuación se presenta una descripción detallada del problema a resolver, donde se incluye: la motivación que permite estimar el problema como relevante e interesante de solucionar, una descripción detallada de éste, la descripción de los requisitos básicos que deben ser cumplidos por cualquier solución que resuelva el problema descrito y, finalmente, los criterios mínimos que servirán de base para aceptar cualquier solución propuesta al problema.

2.1. Motivación

La población con problemas visuales ocupa una importante proporción dentro de las personas con discapacidades en Chile. Tan sólo el número de personas con ceguera total alcanza en el país la cifra de 42.931 personas [4], sumado a otra importante cantidad con distintos grados de discapacidad visual. Considerando estos datos, se puede concluir que no es posible ignorar los problemas a los que estas personas se enfrentan, por lo que es esencial encontrar formas de ayudarles a integrarse de manera exitosa en los entornos en los que se deben desenvolver.

Dentro de las dificultades constantes a las que se deben enfrentar las personas con discapacidad visual, se encuentra el hecho de tener que lidiar con los frecuentes problemas que se presentan al llegar a lugares que no están preparados para su presencia. En estos casos, muchos de los objetos que son cotidianos para la mayoría de las personas se convierten en verdaderos obstáculos para ellos, poniendo en riesgo incluso su integridad física. Además, se presenta la dificultad que conlleva la adaptación al entorno, teniendo que dedicar mucha

atención, y una importante carga cognitiva a éste, lo que disminuye el esmero que pueden dar a la tarea que efectivamente quieren desarrollar allí. Incluso el hecho de tener que acudir por ayuda de otras personas en estas situaciones puede afectar de forma considerable su sentido de independencia, mermando las opciones de decisión que una persona con esta clase de discapacidad pudiera tomar bajo circunstancias idóneas. Estos problemas persisten en el caso particular de los entornos cerrados, donde simples objetos como sillas, mesas y escaleras pueden convertirse en una molestia innecesaria e incluso ser causantes de graves accidentes.

Es precisamente en esta clase de entornos en los cuales descansa el problema a presentar, el que puede ser introducido con la siguiente pregunta: ¿Cómo ayudar a personas con discapacidad visual a poder orientarse en entornos cerrados y complejos?

2.2. Descripción del problema

Con entornos cerrados y complejos se hace referencia a todos aquellos lugares cerrados que son desconocidos para la persona con discapacidad visual, por lo que orientarse en ellos no es una tarea trivial. Tal como lo indicaba la pregunta anterior, el problema a resolver consiste en ayudar a las personas con discapacidad visual a orientarse en estos lugares cerrados que son complejos para ellos. Esto implica a lo menos abordar el problema considerando tres áreas:

1. reconocer la posición del usuario y calcular su desplazamiento en tiempo real.
2. obtener los datos vinculados al entorno y asociarlos con los datos referentes a la posición del usuario.
3. comunicar la información generada al usuario.

Otro problema que debe ser abordado está asociado a la forma en la que el sistema y el usuario deben interactuar, considerando las circunstancias especiales de este último, de forma que la solución se acomode a sus necesidades sin bloquear el resto de sus sentidos, lo que le impediría realizar otras actividades que son las que efectivamente el usuario desea desarrollar en el lugar en el que se encuentra. Siempre se debe entender que, en este tipo de soluciones, éstas no son el fin, sino el medio para lograr otro fin. Esto implica que la

interacción del usuario con el sistema debe realizarse de la forma más sencilla posible, para así facilitar su relación con el resto del entorno y las actividades que el usuario efectúa.

Además de lo ya expuesto, otro factor que debe ser abordado tiene que ver con los costos asociados a la solución que se pretende desarrollar e incluso la disponibilidad de las herramientas requeridas por el sistema. De esta forma se hace posible que cualquier solución desarrollada pueda llegar a la mayor cantidad de personas posible, convirtiéndose en una ayuda real en el proceso de integración de este grupo de personas con necesidades especiales.

Teniendo en cuenta las aristas del problema que han sido presentadas, es necesario reconocer cuáles son los requisitos necesarios que cualquier solución debiese considerar.

2.3. Requisitos de la solución

A continuación se presentan los requisitos de usuario y funcionales que debiesen ser incluidos en el proceso de resolución de este problema. La determinación de los requerimientos presentados en esta sección fue realizada a partir de una investigación elaborada por Wu et al. [15], en la que se procuró determinar de forma general un modelo de navegación en entornos cerrados para personas con discapacidad visual.

En primer lugar se presentarán de forma explícita todos los requerimientos presentados por Wu et al. en su trabajo, para luego exponer ciertas consideraciones a estos requerimientos de acuerdo a las necesidades particulares del problema que ya ha sido presentado.

2.3.1. Requerimientos generales de usuario

La captura de requerimientos de usuario fue realizada con un conjunto de personas visualmente impedidas a lo largo de Europa, en el proyecto ENABLED [1]. Se recogió una muestra de cincuenta personas representativas de esta comunidad, incluyendo tanto a personas con visión limitada como a personas totalmente ciegas, de distintas edades y con condiciones adquiridas o heredadas.

A pesar de que la investigación fue conducida principalmente para reconocer y establecer las necesidades de los usuarios en lo que respecta a los desplazamientos en entornos cerrados, existen requerimientos expuestos que son aplicables a los sistemas de navegación en entornos

abiertos. A pesar de esto, esta clase de requerimientos fue incluida para reflejar de forma íntegra la investigación presentada.

Los requerimientos de esta comunidad se han resumido en tres grandes áreas.

1. Requerimientos del sistema de navegación

- Administración y planificación inteligente de ruta
- Indicación adaptable de ruta
- Traspaso suave de ambientes, incluyendo medios de transporte
- Retroalimentación precisa del progreso
- Guía actualizada para evitar y advertir peligros
- Información general del entorno

2. Interoperabilidad y Usabilidad

- Interoperabilidad con otras herramientas de asistencia (ej: bastón, ayuda auditiva)
- No bloqueante de otros sentidos
- Tasa y densidad de información personalizable
- Privilegiar la independencia, pero que permita la asistencia de una persona con vista normal
- Interfaz de usuario personalizable
- Niveles graduados de funcionalidad (i.e. principiante, experto)

3. Requerimientos basados en la comunidad

- Compartir información, bases de datos y mapas
- Petición de asistencia en caso de emergencia
- Posición relativa de otros usuarios o acompañantes

2.3.2. Requerimientos del modelo genérico de navegación en entornos cerrados

Los requerimientos asociados al diseño de un sistema de navegación en entornos cerrados pueden ser categorizados en tres partes:

1. Requerimientos de alcance del modelo:

- **Generalidad:** El modelo tiene que ser capaz de acomodarse a distintos tipos de entornos cerrados. Además debe tener la capacidad de ajustarse a los sensores y tecnologías que han sido desarrolladas por sistemas de navegación y posicionamiento.
- **Cobertura local:** El modelo debe tener la capacidad potencial de administrar, controlar y guardar los servicios y elementos del entorno que han sido abstraídos del mundo real en los ambientes cerrados que han sido cubiertos.

2. Requerimientos operacionales:

- **Centrado en el usuario:** los servicios en el modelo deben considerar el perfil del usuario, sus preferencias y su nivel de visión.
- **Disponibilidad:** el modelo debe compensar la deficiencia y falta de comunicación de los sistemas de navegación y posicionamiento.
- **Escalabilidad:** el modelo debe resolver de forma escalable, tanto en sentido espacial como temporal, la información asociada a la ubicación y los servicios entregados.
- **Rendimiento:** baja latencia, alto rendimiento e información de confianza son deseables.

3. Requerimientos asociados al modelo de negocios:

- **Registro de usuarios:** los usuarios tienen que registrarse con un proveedor de servicios de localización con el fin de realizar un seguimiento o recibir información de su ubicación. El registro es necesario con el fin de relacionar los perfiles de usuario con la implementación individual del modelo.

- Uso de estándares bien conocidos: las interfaces y protocolos usados deben ser publicados y estandarizados. Esto reduce la complejidad de la comunicación entre las aplicaciones basadas en la localización y los proveedores de servicios.
- Mantenimiento y costo: El costo de mantenimiento e instalación por unidad de área es un buen indicador del costo general de un sistema específico. El costo de cualquier hardware y software específico suele ser un costo único. Esquemas de suscripción a los proveedores de servicios y actualizaciones de software pueden también ser considerables durante la vida útil del sistema.

2.3.3. Consideraciones a los requerimientos presentados

Ahora bien, como se comentó anteriormente, los requerimientos descritos han sido establecidos para una solución genérica. Considerando esto, para el problema que se aborda en este trabajo, se incluirá un subconjunto de estos requerimientos, dentro de los que se encuentran aquellos considerados más relevantes y abordables, teniendo en consideración el tiempo disponible para el desarrollo y pruebas de la aplicación.

Dentro de los requerimientos de usuarios no se considerarán aquellos que están relacionados con la planificación de ruta, ni el cambio de ambientes, puesto que no aplican al tipo de entorno bajo el cual se está trabajando. Por otro lado, los requerimientos basados en la comunidad tampoco serán considerados, para poder concentrarse principalmente en las funcionalidades que debe tener la solución.

En lo que respecta a los requerimientos asociados al diseño de la solución, se obviará la necesidad de registrar al usuario para poner a su disposición el sistema. De esta forma se puede poner a disposición la solución de la forma más simple posible al usuario, sin imponer trabas innecesarias. Además, dado el contexto de la solución, no se construirán perfiles asociados a cada persona, de forma que el registro de usuarios carece de sentido práctico.

2.4. Criterios de Aceptación de la solución

Por último, es necesario establecer ciertos criterios apropiados para la aceptación de la solución propuesta. Considerando las características de la solución, se estima conveniente

establecer estos criterios basándose en pruebas efectuadas con personas que no conozcan la aplicación, idealmente, usuarios finales.

Para establecer de manera objetiva el criterio de aceptación en función de la percepción de la correcta funcionalidad del sistema por parte del grupo que efectúa las pruebas, se han seleccionado un set de preguntas que, a manera de encuesta, serán respondidas por las personas luego de probar el sistema. Estas preguntas se enumeran a continuación:

1. De una escala de 1 a 10, donde 1 corresponde a *muy compleja* y 10 corresponde a *muy sencilla*, ¿cómo evaluaría la complejidad de la interacción con el sistema mientras lo usó?
2. De una escala de 1 a 10, donde 1 corresponde a *incomprensible* y 10 corresponde a *claro*, ¿cuán comprensible le pareció la calidad de la entrega de información del sistema?
3. De una escala de 1 a 10, donde 1 corresponde a *inútil* y 10 corresponde a *muy útil*, ¿cuán útil le pareció la información entregada por el sistema para desplazarse?
4. De una escala de 1 a 10, donde 1 corresponde a *muy retardada* y 10 corresponde a *inmediata*, ¿cuán rápida le pareció la entrega de información del sistema?
5. De una escala de 1 a 10, donde 1 corresponde a *muy imprecisa* y 10 corresponde a *exacta*, ¿cuán precisa le pareció la información del entorno entregada por el sistema?

Además de proveer este set de preguntas a los usuarios que prueben el sistema, se les brindará la oportunidad de realizar comentarios que permitan justificar sus respuestas, de tal forma de obtener retroalimentación de parte de los usuarios. Por último, podrán realizar comentarios generales de aspectos que no consideren incluidos dentro de la encuesta y que permitan mejorar el funcionamiento de la aplicación.

Dadas las preguntas y ponderaciones indicadas por los usuarios se debe definir un criterio estándar para considerar aceptable cada uno de los componentes del sistema evaluados a través de las preguntas. Los criterios adoptados son presentados a continuación, junto con las medidas a tomar si es que cierto componente no satisface los criterios presentados:

-
- Se considerará aceptable un componente si su evaluación es igual o mayor a 6, dada la escala de 1 a 10, lo que corresponde a un 60% de satisfacción de parte del usuario con la funcionalidad evaluada.
 - En caso de encontrar algún componente con una evaluación promedio menor a 6, dada la escala de 1 a 10, se debe revisar su funcionamiento, para determinar las causas del incumplimiento, dados los comentarios de los usuarios.
 - Aquellos componentes que incumplan el criterio de aceptación deben ser corregidos en la medida que las características del sistema desarrollado permitan hacerlo.
 - En caso, de que no sea posible corregir una funcionalidad, se deben especificar claramente las causas que motivaron esto y que medidas se pueden tomar a futuro para corregir este error.

Capítulo 3

Antecedentes

En este capítulo se presentan los antecedentes necesarios que permiten comprender el contexto bajo el cual se trabajó en la búsqueda de la solución apropiada al problema y que permiten entender de mejor forma las razones por las que se tomaron ciertas decisiones durante el desarrollo de la solución.

3.1. Trabajo relacionado

Existen una serie de trabajos que han procurado cubrir distintos aspectos del problema abordado en este documento, ya sea de forma parcial o de forma general.

En primer lugar, cabe destacar que en el área de asistencia en la navegación se han hecho investigaciones en los siguientes campos, según indican Nicholas A. Bradley y Mark D. Dunlop [2]:

1. Ambientes cerrados (ej: [11, 8, 7])
2. Ambientes abiertos (ej: [12, 13, 6])
3. Ambientes Mixtos, una combinación de los anteriores (ej: [3])

Para ambientes abiertos, se distinguen las tecnologías de “micro-navegación”, que proveen asistencia en cuanto al ambiente inmediato, y de “macro-navegación”, que proveen asistencia con respecto al ambiente distante [2]. En estos casos ha resultado bastante conveniente el uso de sistemas de posicionamiento global (GPS) para reconocer la posición de la persona.

Por otro lado, en ambientes cerrados, área en la cual se centrará la memoria, existen una serie de tecnologías [2] que han debido desarrollarse para la entrega de información útil, debido a la incapacidad del GPS de entregar información de posicionamiento en entornos cerrados. A continuación se presentan algunas de estas soluciones.

3.2. Navegación en ambientes cerrados

Sonnenblick [11] implementó un sistema de navegación en ambientes cerrados basado en la utilización de guías infrarrojas, ubicadas convenientemente a lo largo de los lugares de desplazamiento de las personas con discapacidad. Luego, a través de un dispositivo, se interpretan esas guías y se entrega información adecuada para que la persona pueda ubicarse espacialmente. El problema que se presenta con esta solución gravita en el hecho de que para captar las guías infrarrojas, el dispositivo encargado de detectar la existencia de las guías infrarrojas debe apuntar en dirección de éstas para captarlas, perdiéndose la naturalidad en la integración del dispositivo con el ambiente. Una opción que podría solucionar este problema consistiría en construir un dispositivo que pueda captar estas guías en todas direcciones, de tal forma que el usuario no tenga que apuntar hacia un sector en particular. El problema que surge en ese caso, es que se deben interpretar muchos datos a la vez, dada la distribución de las guías, si es que esto es posible. Además, se deben considerar los costos y disponibilidad de un dispositivo de esta clase.

Por otro lado Hub, Diepstraten y Ertl [8] desarrollaron un sistema para determinar la posición de objetos y personas en un ambiente cerrado, utilizando cámaras para detectar los objetos y sensores de dirección para reconocer la dirección en la cuál la persona se está moviendo. El problema que se presenta en este caso es similar al caso anterior, puesto que se requiere de un dispositivo especializado para que el usuario interactúe con el ambiente. En particular, en su investigación, Hub, Diepstraten y Ertl desarrollaron un dispositivo que cumple la función deseada. De esta solución no se conocen los costos, sin embargo, el hecho de que no sea una herramienta masiva ya existente, puede dificultar su llegada a los usuarios finales. Además, por la ubicación sugerida para las cámaras, nuevamente se presenta una falta de naturalidad en la interacción, puesto que el usuario tendría que apuntar su bastón en la dirección en la cual se encuentran los objetos. Esto podría resultar natural para objetos que se encuentren en el suelo, sin embargo no lo sería para objetos elevados, puesto que no

serían identificables debido a la posición de las cámaras¹.

En un trabajo posterior, Hub, Hartter y Ertl [7] fueron un poco más allá en el concepto presentado en el párrafo anterior e incluyeron la capacidad de realizar un seguimiento de varios tipos de objetos móviles. Luego, a través de un algoritmo similar a la percepción humana, procuraron identificar dichos objetos en base a comparaciones de color y forma con objetos de entrenamiento. Puesto que este trabajo es una extensión del presentado en el párrafo anterior, también hereda los mismos problemas.

A pesar de que en esta ocasión se efectuó un análisis crítico de las soluciones de forma de detectar ciertos vacíos que puedan ser llenados por la solución a implementar, cabe destacar que estos trabajos permitieron obtener ideas interesantes con respecto a la forma en la que es posible solucionar el problema. En particular, en el caso del trabajo de Hub, Diepstraten y Ertl, llama la atención la adaptación de herramientas presentes en la vida cotidiana de una persona con discapacidad visual para generar la interacción con el sistema. En particular se hace referencia al bastón guía usado por los ciegos. Llama la atención este hecho, puesto que para las personas con discapacidad visual es natural utilizar esta herramienta para reconocer su entorno, por lo que esta nueva funcionalidad no debiese parecerles extraña para el objeto seleccionado.

Por otro lado, el trabajo desarrollado por Sonnenblick, llamó la atención debido a lo ingenioso de la utilización de guías infrarrojas para la determinación de la posición del usuario. El costo de los emisores infrarrojos es bastante bajo, por lo que su uso se considerará en la implementación.

3.3. Modelo de Diseño para navegación en ambientes cerrados

Además del desarrollo de herramientas específicas para la navegación en ambientes cerrados, se han diseñado modelos que de forma genérica establecen las componentes que debieran tener este tipo de sistemas. Aquí se considerará el modelo [15] propuesto por Wu et al.

¹Las cámaras, según este modelo, apuntan en la misma dirección que el bastón, lo que naturalmente será hacia abajo.

Dentro de lo cubierto en el desarrollo de este modelo, se presenta una serie de requerimientos asociados a la implementación del modelo propuesto. Estos ya fueron abordados en la descripción del problema, por lo que ahora se dará énfasis al entorno de trabajo que se presenta, el cual será considerado en el desarrollo de la solución.

3.3.1. Entorno de trabajo

El entorno de trabajo propuesto cuenta con tres capas, tal como se puede apreciar en la figura 3.1.

La primera capa corresponde al trabajo de detección de la posición del usuario. Esta labor debe ser realizada utilizando alguna tecnología apropiada (ej. RFID, Wi-Fi, WLAN). Se destaca en este caso que la capa no está encargada de efectuar el trabajo de la lógica de la aplicación, por lo que su objetivo es sólo determinar la posición del usuario en base a algún punto de referencia determinado previamente. Considerando esto, se repara en el hecho de que esta capa es esencialmente hardware desde el punto de vista del desarrollo, puesto que la porción de software que genera el valor asociado a la posición, dada la entrada capturada, generalmente viene integrada dentro del mismo sistema, por lo que no debiera existir a priori mayor intervención de parte de la aplicación a construir, más que recibir los datos.

La segunda capa corresponde a una capa de datos demonio, que incluye la planificación de la ruta, servicios de seguimiento, mapas inteligentes y el registro de usuarios y perfiles. Esta capa es llamada “Modelo de Navegación Indoor”. Se destaca que es en esta capa donde se propone que se encapsule toda la inteligencia del modelo, por lo que esta capa es liberada de otras responsabilidades, como la comunicación con el usuario. Se debe recordar además que, para el problema presentado, las responsabilidades de esta capa disminuyen, al no tener que incluir la planificación de la ruta ni el registro de usuarios y perfiles.

La tercera capa corresponde a una interfaz interactiva centrada en el usuario. Esta capa recibe los requerimientos de usuario y crea los perfiles individuales. Puesto que, como ya se ha mencionado en la definición de los requerimientos del sistema, no se considerará la construcción de perfiles de usuario como requerimiento, para efectos prácticos, esta capa se

convierte en una interfaz que permite al usuario enviar los datos necesarios al sistema, para que este último genere los datos relevantes en la relación del usuario con el entorno.

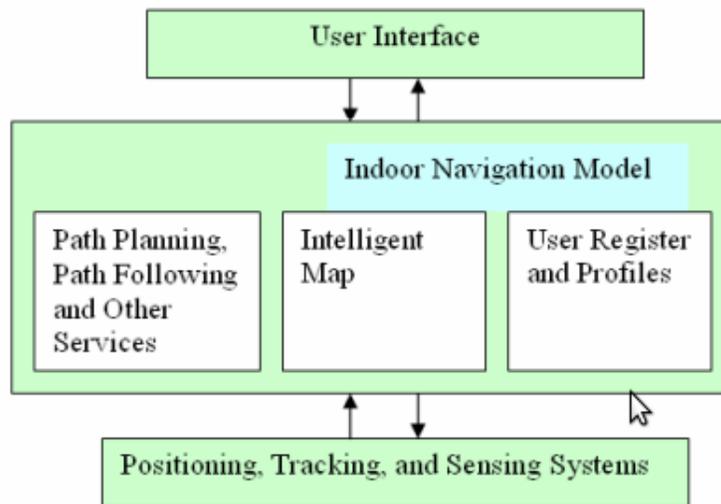


Figura 3.1: Entorno de trabajo para un sistema de navegación indoor.

3.4. El control Wiimote

Se conoce como control “Wiimote” o “Wii Remote” al control utilizado por la consola Nintendo Wii, reconocida por introducir al mundo de los videojuegos una interacción con los usuarios desconocida anteriormente.

Este control cuenta con una serie de interesantes características. Sin embargo, en esta ocasión se destacará una en particular, la cámara infrarroja, que se encuentra ubicada en la parte frontal del control (ver figura 3.2).

Esta cámara es capaz de reconocer objetos emisores de luz infrarroja, lo que le permite al “Wiimote” calcular y entregar una posición relativa a su propia posición, de la fuente que emite dicha luz. De este modo, fijando el Wiimote en un entorno cerrado, y conociendo su posición exacta, es posible reconocer de forma precisa la posición de los objetos que emitan luz infrarroja en ese entorno. Esta particularidad es bastante interesante bajo el contexto del problema presentado, ya que combinada con la idea de Sonnenblick de utilizar guías infrarrojas puede suministrar ayuda importante en la solución del problema.



Figura 3.2: (Arriba) Vista superior del Wiimote. (Abajo) Vista frontal con cámara infrarroja del Wiimote en primer plano.

Para ser un poco más precisos en este sentido, se debe establecer el significado de los datos que entrega el Wiimote. La cámara infrarroja del control es capaz de entregar un punto en dos dimensiones, asociado al plano cuya normal puede ser representada por el vector de visualización del Wiimote (plano perpendicular al Wiimote). Este vector indica hacia dónde está observando la cámara infrarroja del control. Ahora bien, matemáticamente existen infinitos planos que cumplen con la condición de ser perpendiculares al vector de visualización del Wiimote. En términos prácticos, esto indica que la cámara infrarroja es incapaz de entregar datos asociados a la profundidad en la que se encuentra un emisor infrarrojo. Además, puesto que la cámara infrarroja tiene un ángulo de visualización no recto, es decir, su rango de visibilidad es más amplio a medida que se encuentra más distante el emisor infrarrojo, el valor de cada eje entregado por el Wiimote no puede ser interpretado como un desplazamiento en un solo eje, sino que debe entenderse como un ángulo. La figura 3.3 representa esta idea.

Cabe destacar que el Wiimote entrega valores normalizados para cada eje, es decir, valores dentro del rango de $[0,1]$. Para convertir estos datos a valores en radianes, basta con una simple transformación lineal.

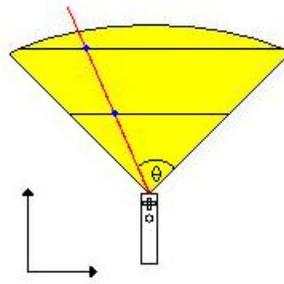


Figura 3.3: Representación de dos puntos a los que el Wiimote les asocia el mismo valor (pertenecen al mismo ángulo) pero que no tienen el mismo valor para otro tipo de eje genérico.

Se hace notar también, que el ángulo de apertura de visualización del Wiimote corresponde a aproximadamente 22° . Este valor ha sido obtenido de forma experimental al evaluar los rangos de visualización del Wiimote ante diferentes distancias del objeto emisor infrarrojo y el Wiimote.

Ahora bien, una vez que se puede obtener la “posición” de los objetos que emiten luz infrarroja con el Wiimote, el siguiente paso será transmitir esta información a un computador. Para poder comunicarse con un computador, el Wiimote cuenta con un dispositivo bluetooth que le permite transmitir cierta información, la cual es utilizada para comunicar la consola con el control. Esta característica fue capturada a través de una librería [14] creada para estos efectos, la cual se encarga de recibir e interpretar la información enviada por el Wiimote y desplegarla en una forma entendible por otros sistemas. Esta librería está disponible de forma gratuita en la Web (ver <http://www.codeplex.com/WiimoteLib>).

Para un correcto funcionamiento de la comunicación, el creador de la librería recomienda la utilización de BlueSoleil como software Bluetooth en el computador que recibirá la información, debido a que la librería utilizada fue creada y probada utilizando ese software. Esto, según el creador, garantizaría un correcto funcionamiento de la aplicación. De todas formas, pruebas preliminares indican que también funciona correctamente con otros controladores Bluetooth.

Capítulo 4

Descripción de la Solución

Previamente, al describir el problema y las características que debía cumplir su solución, se comentó que ésta debía considerar al menos 3 áreas: reconocer la posición del usuario a tiempo real, obtener la información asociada al entorno y ser capaz de interpretar y comunicar esta información al usuario. A continuación se explicará cómo se abarcaron estas 3 áreas en el desarrollo de la solución propuesta.

4.1. Descripción general de la solución

La solución se conforma de la siguiente forma. Para que el usuario pueda entregarle su posición al sistema se creó una interfaz no convencional, que consiste en la adaptación de un bastón para ciegos, de tal forma que sea capaz de emitir luz infrarroja cuando el usuario lo requiera. Esta luz infrarroja emitida por el bastón debe ser captada por al menos dos Wiimotes. A través de estos se obtiene un ángulo, tal como se explicó en la sección 3.4, que determina la relación entre la posición del usuario y la posición del Wiimote, la cual es enviada a la porción del sistema que se encarga de la lógica. En esta última se calcula la posición del usuario, en relación a su entorno, y se compara con la posición de los objetos que se encuentran en la habitación. Si la distancia entre los objetos y el usuario es menor a un cierto valor, el sistema le dará aviso al usuario a través de audio, utilizando una librería TTS (Text-To-Speech). Estos avisos pueden ser generados directamente por los parlantes conectados al computador o a través de un dispositivo móvil adaptado para esa labor. La cantidad de información puede ser filtrada de dos formas: modificando el valor límite con el cual se debe comenzar a dar aviso al usuario y limitando los datos entregados, para que el usuario solamente reciba advertencias acerca de los objetos que se encuentran en la dirección en la que se está moviendo. La forma en la que esto puede ser realizado se explica en la

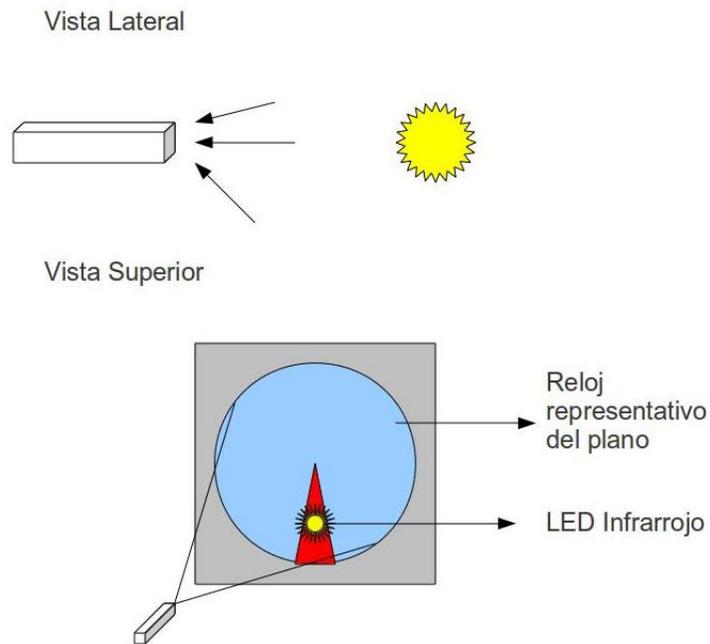


Figura 4.1: Representación del área visible y relación con programa de prueba.

sección 4.3.2.

4.2. Librería WiimoteLib

Lo primero que se debe hacer es establecer ciertas certezas que permitan determinar la viabilidad del proyecto. En particular, se desea evaluar la posibilidad de conectar el Wiimote y recibir los datos apropiados. Esto es importante, puesto que de funcionar de manera correcta, permitiría contar de forma inmediata con la primera capa expuesta en el modelo presentado en la sección 3.3.

Para comprobar esto se debe realizar una prueba básica que permita demostrar el correcto funcionamiento de la librería.

La idea consiste en la generación de un círculo dividido en segmentos de colores, tipo reloj, que representará el área visible para el Wiimote en un plano. Cuando un sector del área visible del Wiimote es apuntado por un LED infrarrojo, el segmento asociado en el

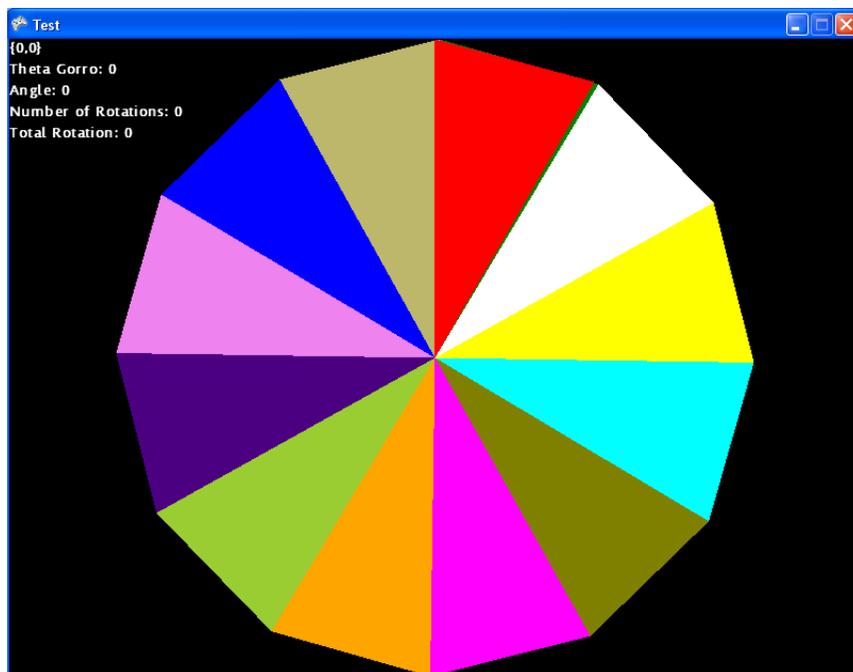


Figura 4.2: Programa de prueba en el instante inicial

círculo debe cambiar a color rojo, para así determinar que se están recibiendo de forma correcta los datos capturados por el Wiimote. Esta idea puede ser visualizada en la figura 4.1. La prueba pudo ser realizada con éxito, tal como se aprecia en las figuras 4.2 y 4.3.

4.2.1. Adaptación de la librería

Luego de comprobar la exactitud de la librería para captar la posición de emisores infrarrojos en un plano visible, fue necesario chequear en más detalle la librería para ver si cumplía con todos los requerimientos necesarios para el desarrollo de la solución. Al realizar la revisión de los componentes de la librería se notó que la librería genera un identificador asociado al Wiimote que no era consistente a través de distintas instancias de ejecución, es decir, el identificador era persistente durante el periodo de ejecución, mas no entre distintas ejecuciones de la librería.

Para solucionar este problema se debió adaptar la librería para obtener un identificador único para el Wiimote. Esto se logró realizar tomando los identificadores capturados por el controlador Bluetooth al leer un dispositivo. Estos fueron capturados y entregados a la hora de inicializar una conexión con un nuevo dispositivo.

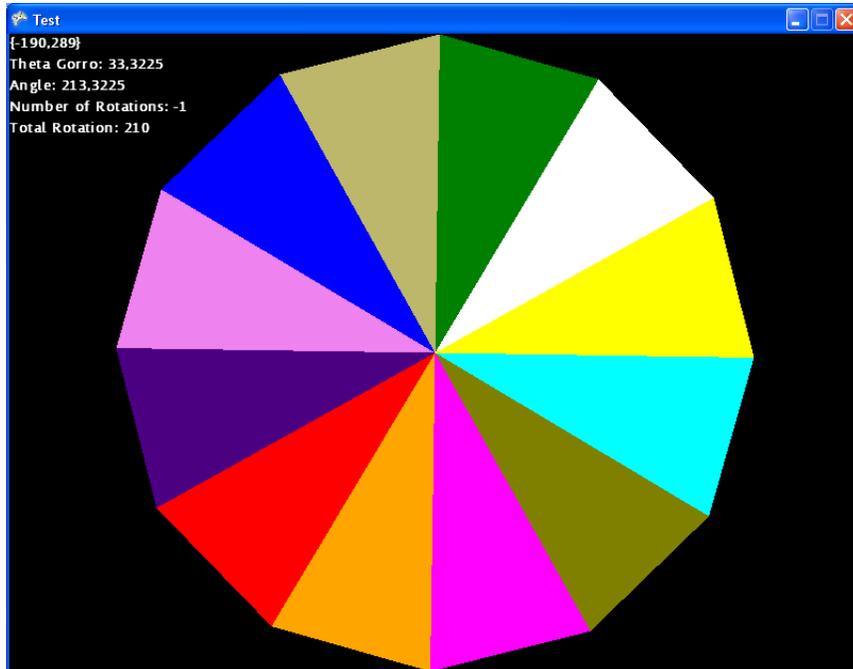


Figura 4.3: Programa de prueba después de mover a la esquina inferior izquierda el LED

A pesar de que esto soluciona el problema, impone la restricción de que una vez implantado el sistema, no se puede cambiar de controlador Bluetooth, ya que otro controlador generaría un identificador en formato diferente. Sin embargo, se consideró que cambiar un controlador no es una acción que se realice a menudo. De todas formas se destaca que esto hace referencia sólo a cambios de controlador, por lo que reinstalar el mismo o actualizarlo no debiera generar problemas.

4.3. Interfaz usuaria

Al definir la interacción necesaria entre el o los usuarios y el sistema, se encontró que ésta podía ser separada en dos grandes requerimientos: otorgar al usuario la capacidad de entregarle al sistema su posición y permitir la interacción de un administrador con el sistema para hacerle entrega a este último de la definición de los parámetros necesarios para una correcta ejecución de la aplicación.

4.3.1. Interfaz para la interacción de la persona con discapacidad visual y el sistema

La interacción entre el usuario con discapacidad visual y el sistema es uno de los grandes desafíos presentes en esta solución, puesto que exige generar una interfaz adaptada a las circunstancias especiales del usuario. En el modelo propuesto por Wu et al. corresponde a la tercera capa mencionada en la sección 3.3. Teniendo en cuenta este hecho se requiere diseñar una interfaz sencilla que permita al usuario interactuar rápidamente con el sistema. Por otro lado, considerando que para el usuario es fundamental tener a disposición el resto de sus sentidos, se considera prudente integrar la solución con objetos ya presentes en la cotidianidad del usuario, de tal forma de no incomodarlo con nuevos objetos.

Teniendo en cuenta las restricciones mencionadas, se estimó adecuado desarrollar un objeto aumentado [9], utilizando alguno de los objetos disponibles para el usuario de forma cotidiana. Ahora bien, además de no sobrecargar al usuario de forma física, tampoco es apropiado cargarlo cognitivamente. Por esta razón es necesario realizar una selección apropiada del objeto a aumentar, de tal forma que la nueva funcionalidad otorgada le parezca lo más natural posible. Para conseguir esto se tomó en cuenta una investigación realizada por Guerrero, Ochoa y Horta [5] en la que se plantea un proceso de desarrollo de objetos aumentados.

En la investigación se plantea que el proceso de desarrollo de objetos aumentados cuenta con las siguientes etapas:

- **Definición del problema:** identificar las limitaciones a superar y las oportunidades que permiten aprovechar el uso de objetos aumentados.
- **Contexto de uso del objeto aumentado:** analizar el contexto en el que será usado el objeto. Permite definir una serie de requerimientos no funcionales.
- **Definición de requerimientos:** usando los datos de los dos puntos anteriores, formalizar los requerimientos del objeto aumentado.
- **Selección del objeto a ser aumentado:** dados los requerimientos, identificar el mejor objeto real para ser aumentado. Se debe analizar el modelo sintáctico y semántico

del objeto, es decir, cómo será manipulado y cómo debe ser interpretado su uso. En este análisis debe considerarse la naturaleza física del objeto real.

- **Desarrollo del objeto aumentado:** construir el objeto aumentado a través de sucesivos prototipos.
- **Pruebas con usuarios:** Probar cada prototipo, de forma de obtener retroalimentación del usuario y mejorar el objeto aumentado en el siguiente incremento, hasta obtener un objeto que satisfaga los requerimientos planteados.

Teniendo en cuenta el proceso propuesto, se puede ver cómo fue utilizado en el diseño del objeto aumentado.

En primer lugar es necesario *definir el problema*. De forma general, se puede reconocer que el objetivo final es ayudar al usuario a desplazarse en entornos cerrados. Sin embargo, en particular, para el caso del objeto aumentado, su objetivo se puede limitar a darle al usuario la capacidad de emitir, de una forma entendible, información que permita al sistema determinar su posición en relación a su entorno. Esto se logra, dadas las características del sistema, emitiendo luz infrarroja en la posición en la que se encuentra el usuario. De todas formas, esto no tiene necesariamente que ser conocido por el usuario, pero permite definir el problema en su más bajo nivel, esto es, emitir luz infrarroja en todas direcciones cuando el usuario lo requiera.

Luego, dentro del *contexto de uso del objeto aumentado*, se debe tomar en cuenta las circunstancias especiales que rodean el uso del objeto. En particular se consideran las circunstancias asociadas al usuario, las que impiden el uso de objetos en los que la visión sea un actor relevante. Además, se toma en cuenta que el objeto será usado en ambientes cerrados, los que generalmente cuentan con un espacio limitado de desenvolvimiento.

Dadas las consideraciones expuestas en las etapas anteriores ahora es posible realizar la *definición de requerimientos*. Dentro de los requerimientos funcionales se consideró:

- Emitir la posición del usuario en todas direcciones.
- Permitir que el usuario decida en que momento desea emitir dicha información.

Por otro lado, como requerimientos no funcionales se consideró:

- objeto aumentado de tamaño razonable para no incomodar, dadas las dimensiones del entorno.
- cualquier interacción con el usuario no debe depender el uso de la visión para su funcionamiento.
- si la interacción con el usuario depende de un objeto, por ejemplo, un botón, éste debe ser lo suficientemente grande para ser reconocido por el usuario.

Considerando los requerimientos generados en la etapa anterior se trabajó en la *selección del objeto a ser aumentado*. Tal como se comentaba previamente, en este caso es necesario procurar que la carga cognitiva requerida para asimilar la nueva funcionalidad del objeto sea mínima. Considerando que la interpretación del uso que se le dará a esta nueva funcionalidad será la de reconocer objetos que se encuentran en el entorno, se consideró que el objeto que representa una menor carga cognitiva para el usuario es el bastón, puesto que éste ya es usado de forma similar para reconocer los obstáculos presentes en el entorno. Además, puesto que para usarlo se requieren las manos, su uso con esta nueva funcionalidad permitirá que, rápidamente, a través de un botón, el usuario pueda activar la nueva funcionalidad.

Ahora que se realizó la elección del objeto a aumentar se debía *construir el objeto aumentado*. Para esto se desarrolló un prototipo, utilizando un bastón, el cual fue dotado en el extremo superior de una serie de emisores infrarrojos apuntando en distintas direcciones, tal como se aprecia en la figura 4.4.

Por otro lado, para encender estos emisores, se dotó al objeto con un switch de tamaño adecuado, a la altura de la mano del usuario, para permitir de esta forma una rápida interacción entre el objeto y el usuario. Esto puede ser visto en la figura 4.5.

En lo que respecta a las *pruebas realizadas* sobre este prototipo, éstas serán comentadas en el capítulo 5.



Figura 4.4: Vista superior de la adaptación realizada al bastón. Se aprecia que la posición de los emisores infrarrojos permiten emitir luz infrarroja en todas direcciones.

4.3.2. Interfaz de administración y pruebas

Para poder interactuar de forma correcta con el sistema, se encontró pertinente desarrollar una interfaz que permita entregar y/o modificar los parámetros básicos necesarios para el correcto funcionamiento del sistema. Las características de la interfaz se presentan en el siguiente resumen, mientras que una imagen de la interfaz generada se puede ver en la figura 4.6¹ :

- Permite cargar la representación digital de la habitación que se desea usar². Para eso se debe acceder a: Archivo → Load → Room, con lo que se desplegará un Menú adecuado para cargar el archivo solicitado.
- Permite definir si la información será desplegada por un dispositivo móvil o por parlantes conectados al computador. Al seleccionar la opción del dispositivo móvil se debe ingresar la IP asociada al dispositivo en la red (Sección “Deployer device” en la interfaz.).
- Permite definir qué tipo de información será desplegada. Puede desplegarse toda la información o sólo la asociada a los objetos que se encuentran en la trayectoria del

¹Para ver imágenes ampliadas de la interfaz consultar el apéndice A .

²La forma en la que se genera esta representación digital se explica en la sección 4.4.



Figura 4.5: Vista lateral de la adaptación realizada al bastón. Se observa el switch de color rojo que permite al usuario enviar su posición al sistema cuando este lo requiera.

usuario (Sección “Display Object Information” en la interfaz.).

- Permite definir cuál es la distancia mínima entre un objeto y el usuario para la cual el sistema comenzará a advertir al usuario la presencia del objeto (Sección “Deployer device” en la interfaz. En particular ver menú de selección “Warning Distance”).
- Presenta datos que indican lo que está haciendo el sistema en el momento para, de esta forma, revisar su comportamiento (Sección “What’s Happening” en la interfaz.).
- Genera un mapa sencillo que representa al entorno.

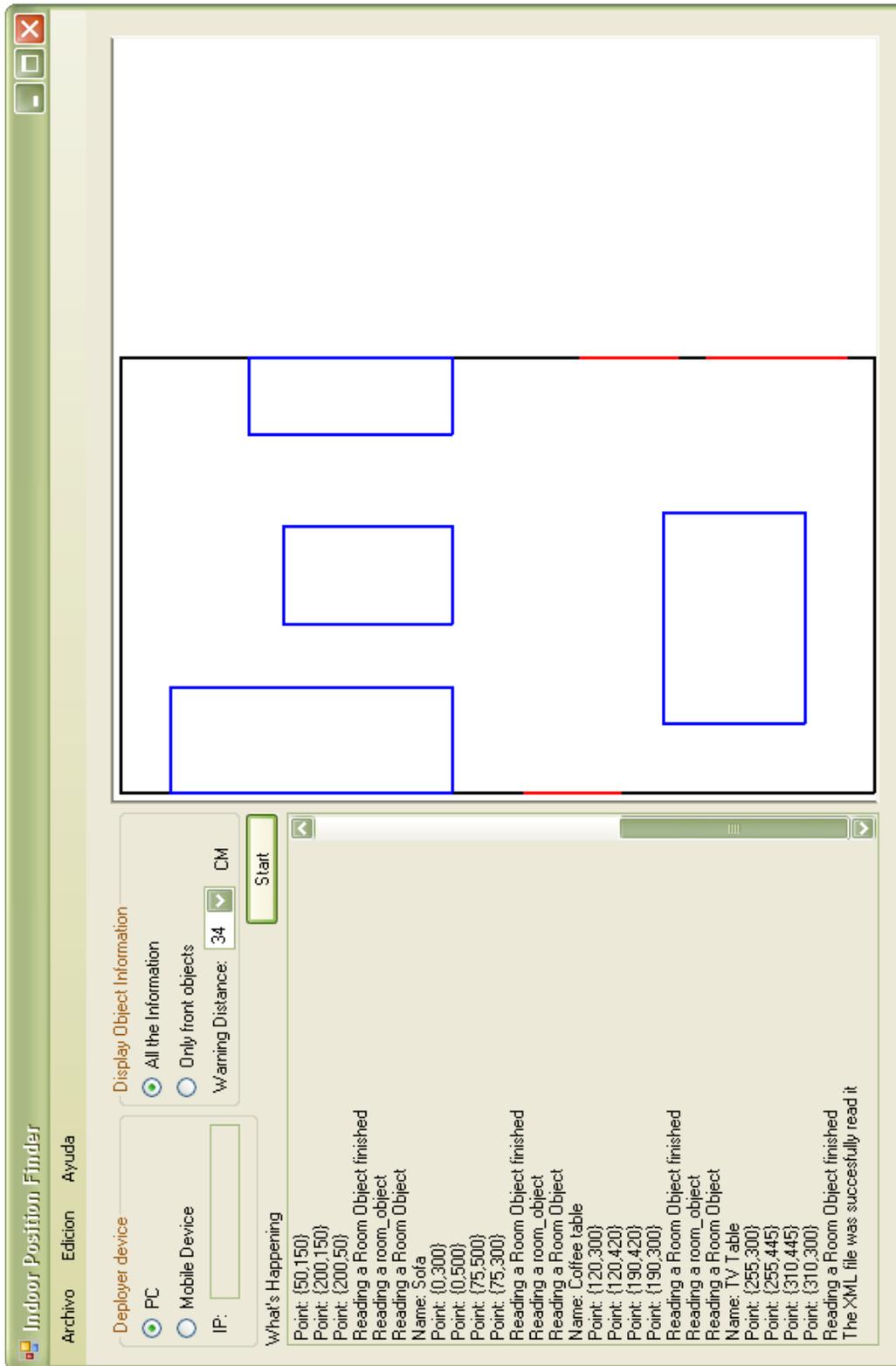


Figura 4.6: Vista de general de la interfaz de administración.

4.4. Generación de la información del entorno

Para poder entrar en funcionamiento, el sistema requiere obtener algunos parámetros iniciales, asociados a la habitación o entorno cerrado, y a la posición de los Wiimotes, que le permitirán asociar los datos obtenidos a través de los controles y que permitirán ubicar a la persona espacialmente en la habitación. Estos datos deben ser obtenidos de forma preliminar y forman parte del proceso de implantación del sistema, ya que, el fijarlos como parte de la implementación, limitaría considerablemente la generalidad de la solución planteada en este sistema.

Los datos del espacio o habitación son guardados en un archivo bajo un formato XML estándar. El formato elegido permite representar todos los datos de una forma comprensible y ordenada. De esta manera se genera una estandarización para el manejo de los datos del sistema, lo que facilita la comunicación con futuros sistemas que quieran interactuar con éste.

Con respecto al esquema³ utilizado, se generó a través del lenguaje de esquema XML Schema, con el cual se definió la estructura válida y las restricciones de los contenidos incluidos en cualquier archivo XML que quiera ser usado como representación de los datos relevantes del espacio cerrado. De esta forma, cada vez que un nuevo archivo XML sea cargado por el sistema, será posible verificar el cumplimiento de los estándares propuestos. Esto puede ser realizado a través de un pequeño programa, que es parte de la solución propuesta, el que usa este documento de esquema como pauta de verificación de la correctitud de los archivos. De esa forma se evita que un archivo malformado sea introducido al sistema e induzca un fallo al tratar de obtener la información necesaria, al no ser reconocido a tiempo por la aplicación. Algunos de los datos requeridos en el archivo XML son los siguientes:

- largo y ancho de la habitación.
- posición de cada uno de los Wiimotes en relación a la habitación.
- ángulo en el cual fueron fijados los controles en relación a los muros y dirección en la que miran.

³La estructura de este esquema puede revisarse en el apéndice B .1

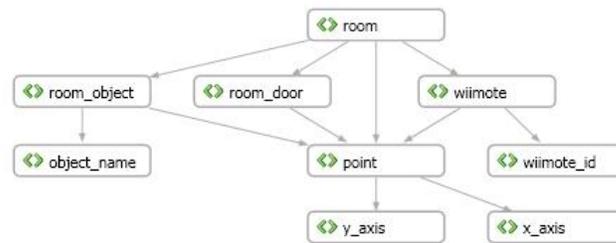


Figura 4.7: Relación entre los objetos definidos en la estructura XML.

- posición en relación a la habitación de cada componente o posible obstáculo que hay en ésta (escaleras, muebles, puertas, etc.).

Al registrar cada uno de estos datos se genera un mapa completo de la habitación, con el cual se puede reconocer el entorno en el que se encuentra el usuario. En la Figura 4.7 se presentan las posibles relaciones existentes entre los objetos del archivo XML. Utilizando esta estructura, es posible representar cualquier espacio o habitación y todos los objetos contenidos en ella. El siguiente paso será determinar la exacta posición del usuario y mapearla en esta estructura.

4.5. Reconocimiento de la posición del usuario

El reconocimiento de la posición del usuario es uno de los puntos centrales de la aplicación desarrollada. Es en esta porción donde se concentra la lógica del sistema, lo que corresponde a la segunda capa en el modelo presentado en la sección 3.3. La posición exacta del usuario se obtiene a través de los datos obtenidos con los Wiimotes y se asocia (mapea) espacialmente con el resto de la habitación. De esta forma es posible entregarle información valiosa al usuario con respecto a los obstáculos cercanos.

Cuando el Wiimote detecta la presencia de un emisor infrarrojo, es capaz de determinar la posición (o ángulo, tal como se explicó en la sección 3.4) en la que éste se ubica en un plano perpendicular a la dirección en la cual apunta la cámara infrarroja. Sin embargo, este dato, a pesar de ser necesario, no es suficiente para determinar la posición del usuario relativa a la

habitación. Para lograr esto, se debe utilizar un segundo Wiimote que determine la posición del emisor infrarrojo desde otra perspectiva. Teniendo estas dos vistas es posible determinar la ubicación del emisor y, por tanto, del usuario, a través de un proceso de triangulación.

4.5.1. El proceso de triangulación

El proceso de triangulación consiste en tomar los datos de cada Wiimote y generar dos rectas. Estas rectas deben pasar por el punto en el cual el Wiimote está ubicado, punto que es conocido gracias a los parámetros iniciales que son almacenados en el documento XML, y por el punto en el cual se ubica el usuario, el cual se desconoce en principio. Sin embargo, como es posible, gracias al Wiimote, conocer el ángulo generado por esta recta, se puede calcular de forma precisa la ubicación de la proyección de esta recta en el muro opuesto de la habitación. Con este segundo punto es posible calcular la ecuación de la recta asociada. Como esto se realiza para ambos Wiimotes, se puede calcular la intersección de las dos rectas generadas, que corresponde a la posición exacta del usuario, que es precisamente lo que se desea obtener.

En primera instancia, esto es útil sólo para habitaciones rectangulares, debido a que se debe proyectar la recta en un “muro” opuesto al muro en el que se encuentra el Wiimote. Además, es necesario que para realizar el cálculo el ángulo formado entre estos dos muros sea recto. Por tanto, ¿es posible generalizar esta solución para cualquier tipo de habitación? La respuesta es afirmativa. Esto es posible puesto que los muros descritos anteriormente no deben ser reales. Para una solución general, el sistema es capaz de crear muros virtuales, los que corresponden a los lados del cuadrilátero que circunscribe a la habitación. De esta forma es posible proyectar la recta sobre los lados de este cuadrilátero y así obtener los mismos resultados explicados anteriormente, sin importar la forma de la habitación.

El hecho de que la proyección generada en el caso de la solución general traspase los límites de la habitación no debiese convertirse en un problema, puesto que aunque esto ocurra, la intersección de las dos rectas jamás debiese estar fuera de los límites de la habitación.

En el caso general, cabe destacar que la posición del punto de origen debe corresponder a la esquina inferior izquierda del cuadrilátero que circunscribe a la habitación. Esta convención permite simplificar los cálculos realizados por el algoritmo. Ahora bien, se nota

que esta restricción depende solamente de la forma en la cual se mire la habitación, por lo que cualquier esquina del cuadrilátero indicado puede corresponder a la esquina inferior izquierda. Por tanto, al fijar este punto, se debe respetar la convención para la definición de todas las otras posiciones descritas en el documento XML que representa la habitación.

Este resultado general es bastante interesante, puesto que permite modelar muchas habitaciones de forma muy similar, describiendo los muros como objetos en la habitación, de tal forma que se informe al usuario de su presencia. Así, el concepto de habitación expresado hasta ahora puede generalizarse a entornos cerrados, abarcando en un solo documento XML a toda una casa, por ejemplo. De todas formas, en esos casos se debieran utilizar más Wiimotes, puesto que estos no son capaces de reconocer emisores infrarrojos a través de los muros. En estos casos debiera usarse al menos un par de Wiimotes por habitación.

Las ecuaciones utilizadas para realizar el cálculo de las rectas y el proceso de triangulación se presentan a continuación.

En primer lugar el sistema de ecuaciones 4.1 describe el caso general para dos rectas generadas en el proceso de triangulación.

$$Y - Y_{w1} = m_1(X - X_{w1})$$

$$Y - Y_{w2} = m_2(X - X_{w2})$$

donde,

$$m_i = \frac{Y_{pi} - Y_{wi}}{X_{pi} - X_{wi}} \quad (4.1)$$

(X_{pi}, Y_{pi}) : Proyección de la recta asociada al Wiimote i

(X_{wi}, Y_{wi}) : Posición asociada al Wiimote i

Por otro lado, en la ecuación 4.2 se presenta la solución del sistema de ecuaciones 4.1.

$$X = \frac{(m_2 X_{w2} - m_1 Y_{w1}) - (Y_{w2} - Y_{w1})}{(m_2 - m_1)} \quad (4.2)$$

$$Y = m_2 \left[\frac{(m_2 X_{w2} - m_1 Y_{w1}) - (Y_{w2} - Y_{w1})}{(m_2 - m_1)} \right] - m_2 X_{w2} + Y_{w2}$$

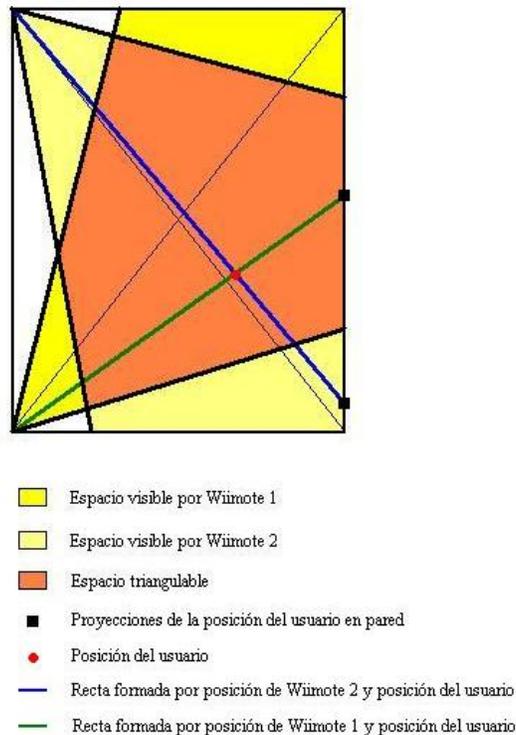


Figura 4.8: Representación gráfica del proceso de triangulación.

Una explicación gráfica de la forma en que estas ecuaciones son utilizadas para realizar el cálculo del proceso de triangulación, se muestra en la Figura 4.8. El entorno o habitación se ve como un sistema de coordenadas en dos dimensiones, donde el punto de origen del plano ($X=0$, $Y=0$) corresponde a la esquina inferior izquierda. En este plano, los dos Wiimotes (colocados uno en la esquina inferior izquierda y el otro en la esquina superior izquierda) permiten definir un espacio “triangulable”, de modo que cualquier objeto dentro de este espacio puede ser mapeado con exactitud.

Realizando esta iteración para el cálculo de la triangulación de los objetos en el entorno de forma constante, es posible reconocer la dirección en la que la persona se está desplazando. Teniendo estos dos datos, es decir, la dirección en la que se dirige y su posición, es posible determinar si se está acercando a alguno de los obstáculos que han sido registrados en el mapa XML del espacio o habitación.

Para el cálculo de la dirección se toman las dos últimas posiciones calculadas por el algoritmo y se restan vectorialmente. De esta forma se crea un vector de desplazamiento

que corresponde a la dirección en la que el usuario se mueve. Ahora bien, puesto que al sostener el bastón es inevitable que se produzcan pequeños movimientos, el algoritmo desestima aquellos vectores de norma muy pequeña. De esta forma se logra que el sistema entregue información que esté asociada a los desplazamientos que efectivamente realizan los usuarios del sistema.

Esto permite indicarle al usuario la ubicación de los objetos de distintas formas. Por ejemplo, podría indicarle la ubicación de un objeto cercano sólo si se dirige hacia él. También podría ser capaz de describirle todos los objetos que se encuentran a su alrededor, a una cierta distancia, indicándole el ángulo en el que se encuentran los objetos, dada la dirección en la que se está moviendo el usuario. Este paradigma de orientación es bien conocido por las personas con discapacidad visual. En éste, se agrupan los ángulos en horas y se le indica a los usuarios la “hora” en donde se encuentra algún objeto, donde las 12 siempre se refiere a la posición frontal del usuario. En particular, para el caso del sistema, se entrega la opción de transmitir la información con esta convención. Sin embargo, puesto que el algoritmo no puede determinar hacia donde está mirando el usuario, asume que la dirección en la que se desplaza el usuario es hacia donde está mirando.

4.5.2. Restricciones del algoritmo presentado

La única limitación encontrada en el sistema se presenta en el caso de una habitación cuadrada. En este caso particular, el algoritmo impide que dos Wiimotes sean ubicados en las esquinas opuestas de la habitación y en el mismo ángulo con respecto a los muros. Si ese fuera el caso, el algoritmo no sería capaz de triangular la posición del usuario. ¿La razón? Al estar ubicados de forma perfectamente simétrica, los Wiimotes miran el mismo plano, por lo que entregan la misma información al algoritmo, la que es insuficiente para poder calcular la posición del usuario. ¿Cómo se puede lidiar con esta limitación? La solución en este caso se limita sencillamente a ubicar los controles de tal forma que compartan un muro o que sean ubicados con ángulos distintos con respecto a los muros aunque permanezcan en esquinas opuestas. De esa forma se elimina la simetría y cada Wiimote sería capaz de presentar datos asociados a distintos planos para calcular la posición del usuario, permitiendo realizar la triangulación sin ningún problema.

4.6. Comunicación

Finalmente, se comenta la forma en la cual se comunica el sistema y también cómo se transmite la información al usuario. Se comenzará por esta última.

4.6.1. Transmisión de la información

Puesto que se tiene la posición del usuario y se reconoce si se dirige a algún obstáculo, es necesario que el sistema sea capaz de transmitirle esta información a la persona que se desplaza. Para lograr esto, se implementó una segunda aplicación que es capaz de transmitir dicha información al usuario de forma auditiva, del tipo “text-to-speech”. De esta manera, un usuario con discapacidad visual puede recibir la información, sin mayor problema, como un mensaje de alerta (de voz) emitido por el computador a través de un parlante simple o a través de un dispositivo móvil.

El tipo de información entregado a los usuarios por el sistema no tiene por objetivo indicarle los movimientos precisos que debe realizar, sino que le entrega una guía que le permite decidir cuál es la siguiente acción de desplazamiento a tomar. Por ejemplo, si la persona se dirige a una mesa, el sistema no puede indicarle si debe esquivarla, puesto que no sabe si se dirige a ella a buscar algo. Por esta razón, el sistema se limita a indicarle al usuario si se dirige a un objeto y le indica qué objeto es. Esta información le debiera permitir al usuario decidir qué acción tomar.

Para transmitir el audio se requiere un parlante conectado a un computador, ya sea un laptop o un PC de escritorio. Por otro lado, para la conexión con el dispositivo móvil se generó otra aplicación pequeña compuesta de un cliente y un servidor, que se comunica a través de una WLAN. El servidor funciona como servicio en el dispositivo móvil, crea un socket y espera conexiones de parte del cliente. El cliente se conecta a este socket creado y envía un stream de audio a través del socket. Ese socket es recogido por el servidor y transmitido a través de los parlantes del dispositivo móvil.

El objetivo principal de habilitar esta característica es simplificar el sistema de tal forma que pueda ser adaptado para acoger a múltiples usuarios a la vez, cosa que no se podría hacer si por un solo parlante se emitiera la información para todos los usuarios. De todas formas,

la capacidad de funcionar con múltiples usuarios no fue considerada en este desarrollo.

4.6.2. Comunicación entre los componentes del sistema

Para comunicar los distintos componentes del sistema se utilizaron las tecnologías disponibles según los participantes de la comunicación. A continuación se presenta un resumen de los componentes usados y la forma en la cual se comunican:

- **Bastón** → **Wiimote**: Señal infrarroja continua.
- **Wiimote** → **Computador**: Bluetooth.
- **Computador** → **Dispositivo infrarrojo**: Socket en WLAN.

Las figura 4.9 muestra la interacción del sistema sin la participación de un dispositivo móvil, mientras que la figura 4.10 presenta la misma interacción con la participación del dispositivo móvil.

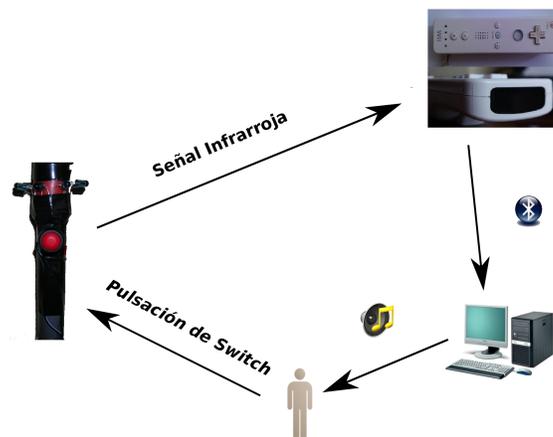


Figura 4.9: Interacción del sistema sin usar el dispositivo móvil.

4.7. Diseño de clases y programación

4.7.1. Comentarios generales sobre la programación

Con respecto al lenguaje de programación utilizado se destaca el uso del lenguaje C# para el desarrollo. Esta elección se efectuó considerando que la librería creada para la comunicación del computador con el Wiimote estaba escrita en este lenguaje. Por esta razón la interacción resultaba natural.

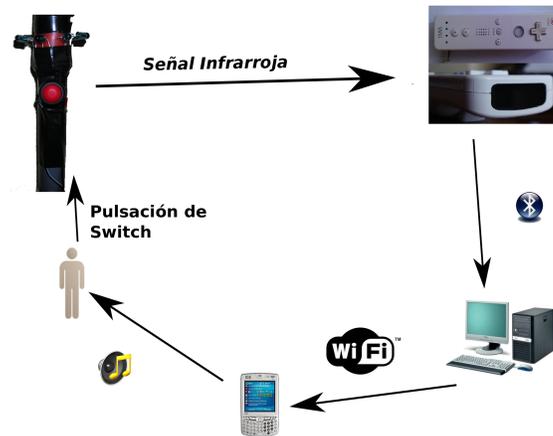


Figura 4.10: Interacción del sistema usando el dispositivo móvil.

Para la creación de la interfaz y la lógica del sistema se utilizó el IDE Visual Studio 2010. Además, en este ambiente se creó el cliente del socket utilizado para transmitir la información del sistema al dispositivo móvil.

Por otro lado, el servidor del socket se debió desarrollar en Visual Studio 2008. Esto se debió a limitaciones asociadas al dispositivo móvil usado en las pruebas, el cual, dada su versión, obligaba a utilizar esta versión del IDE.

Con respecto al código implementado no se realizan comentarios más específicos, debido a que corresponde, en términos generales, a la traducción de los conceptos matemáticos mencionados en la sección 4.5. La especificación de los métodos principales asociados a la lógica del sistema se incluyen en el apéndice C .

4.7.2. Diagrama de Clases

A continuación se presenta el diagrama de clases básico para el sistema construido. Se explicará de forma general el objetivo que se cumple en cada clase. Se incluyeron todas las clases que se consideraron relevantes para comprender el problema. El diagrama asociado se presenta en la figura 4.11.

- **Package IndoorPositionFinder:** Almacena la lógica del sistema.
- **RoomWiimote:** Almacena los datos asociados a los Wiimotes que se registran en una habitación.

- **RoomObject:** Almacena la información asociada a los objetos pertenecientes a una habitación.
- **RoomDoor:** Almacena los datos asociados a una puerta en el sistema.
- **Room:** Objeto que representa una habitación. Contiene colecciones de elementos que representan a los distintos componentes presentes en la habitación así como otros datos generales.
- **XmlProcessor:** Interpreta el archivo XML que representa a la habitación y genera las clases asociadas.
- **WiimoteRecognizer:** Se encarga de la lógica del sistema. Calcula la posición del usuario, las distancias del usuario con otros objetos y genera los mensajes que serán enviados al usuario.
- **Librería WiimoteLib:** Se encarga de comunicar al sistema con el Wiimote.
 - **WiimoteCollection:** Conecta al sistema con una colección de Wiimotes.
 - **Wiimote:** Conecta al sistema con un Wiimote. Es usado por WiimoteCollection.
- **Package RoomDisplay:** Genera la interfaz gráfica del sistema.
 - **Form1:** Interfaz gráfica para la administración del sistema.
 - **Program:** Gatilla la interfaz gráfica.
- **Package TextToSpeech:** Permite comunicar la información vía audio.
 - **TTS:** Interfaz que determina la estructura genérica que debiera tener una clase del tipo Text-To-Speech.
 - **SapiTTS:** Implementación de la interfaz TTS usando la librería Text-To-Speech de Microsoft.
- **Package Socket:** Genera un socket de comunicación para enviar datos a un dispositivo móvil.
 - **Cliente:** Genera un cliente para un socket y envía un stream de audio con la información requerida para el cliente a través del socket.
 - **Servidor:** Genera un servidor en el dispositivo móvil que recibe el stream de audio con los datos y lo reproduce.

Al asociar esta estructura con el modelo presentado en la sección 3.3 se pueden identificar las siguientes relaciones:

- **Primera Capa. Trabajo de detección de la posición del usuario:** Esta capa es fundamentalmente hardware, por lo tanto corresponde al trabajo realizado por los Wiimotes. Sin embargo, aquí es donde participa la librería Wiimotelib, encargada de la traducción de los datos enviados por el Wiimote.
- **Segunda Capa. Modelo de navegación Indoor:** Esta capa es la encargada de la lógica del sistema. Esta capa corresponde al paquete IndoorPositionFinder, el que se encarga de toda la lógica del sistema.
- **Tercera Capa. Interfaz centrada en el usuario:** Esta capa se encarga de recibir los datos del sistema entregados por el usuario. En esta capa aparece el bastón aumentado como componente hardware. Por otro lado, en términos de software, el paquete asociado a esta capa corresponde a RoomDisplay.
- **Comunicación:** Dentro del modelo no se hablaba nunca de la capa de comunicación. Sin embargo para la programación se separó esta interacción del resto del sistema. Este trabajo es realizado por los paquetes TextToSpeech, encargado de la generación de audio, y Socket, encargado de la comunicación del sistema con el dispositivo móvil.

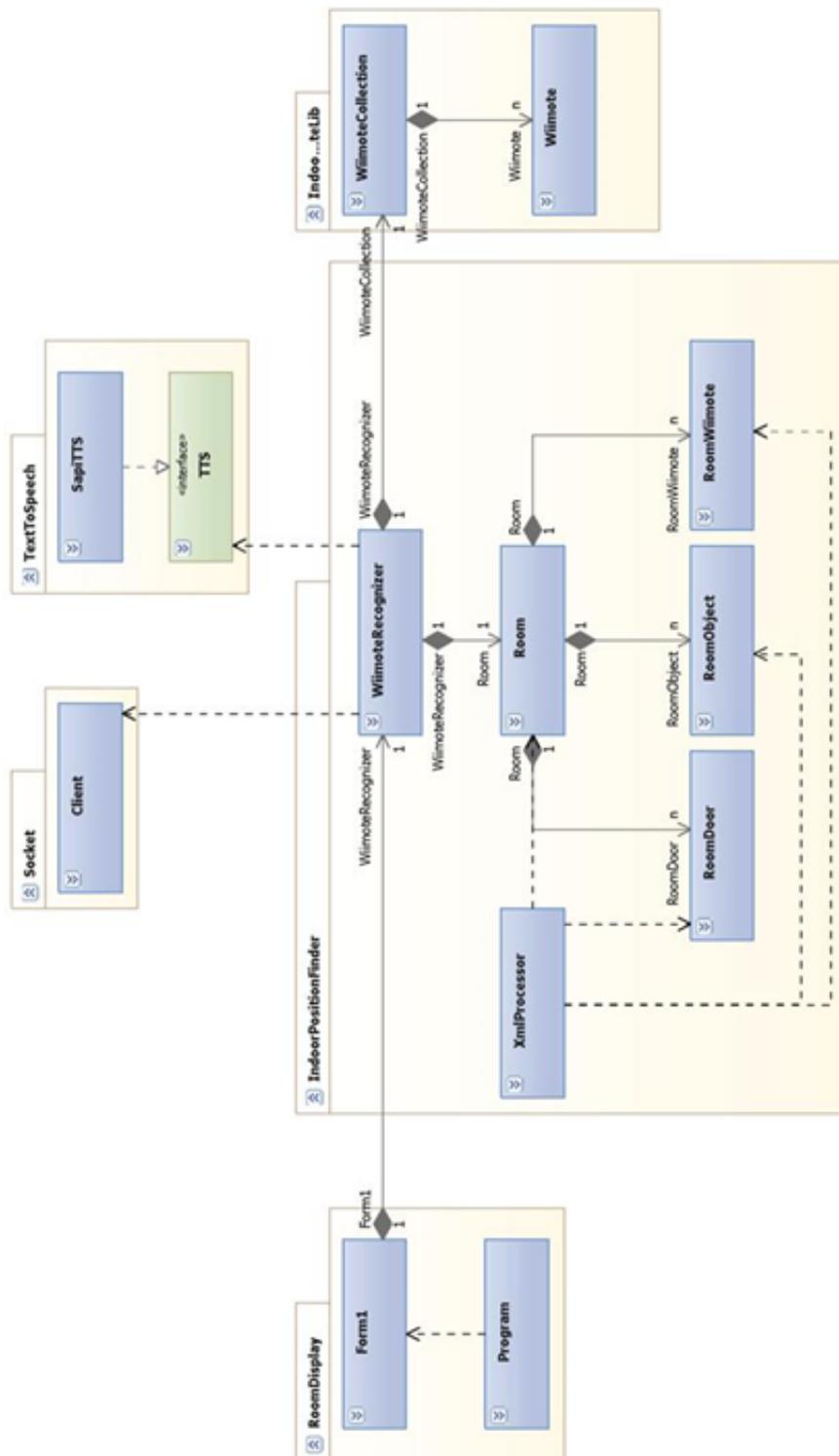


Figura 4.11: Representación gráfica del proceso de triangulación.

Capítulo 5

Validación de la Solución

Tal como se presentó en la sección 2.4, para poder validar la solución presentada es necesario evaluar sus distintos componentes utilizando las preguntas definidas. A continuación se presenta un resumen obtenido de esta evaluación.

5.1. Datos generales del grupo de pruebas

- **Cardinalidad del grupo:** 5 personas.
- **Hombres:** 3.
- **Mujeres:** 2.
- **Usuarios Finales:** No.
- **Edad:** Entre 24 y 28 años.

5.1.1. Comentarios generales del entorno de pruebas

Tal como se puede notar a partir de los datos presentados arriba, el grupo de pruebas es pequeño para ser considerado relevante en términos estadísticos. Sin embargo, esto no quiere decir que la encuesta en si misma no tenga ninguna utilidad. Por el contrario, es parte de un proceso de experimentación más largo, donde, usando un diseño cuasiexperimental de investigación [10], se permite mantener un control sobre lo que se está haciendo. Lamentablemente, el terminar la etapa de experimentación se salía del alcance del trabajo aquí presentado, especialmente por motivos de tiempo, por lo que se propone terminar este proceso como trabajo futuro.

Para generar un entorno de pruebas motivante para el grupo de pruebas se planteó el siguiente desafío. Debían ser capaces de cruzar una habitación habilitada para estos efectos sin utilizar la visión, sólo contaban con la ayuda de la aplicación. En la ruta se le puso obstáculos a los participantes, de tal forma de hacer interesante la prueba, tanto para ellos como para efectuar las pruebas. La figura 5.1 muestra el momento en que estas pruebas fueron realizadas. En la figura se puede percibir la emisión de la luz infrarroja, puesto que las cámaras digitales son capaces de representar este tipo de luz.



Figura 5.1: Imagen de las pruebas realizadas.

5.2. Resultados

A continuación se presenta un resumen de los resultados obtenidos a partir de las pruebas realizadas. Los resultados se dividieron en dos grupos, con y sin dispositivo móvil. Los resultados sin resumir, con comentarios incluidos, se pueden ver en el apéndice D .

5.2.1. Resultados sin hacer uso del dispositivo móvil

Para el caso de pruebas sin hacer uso del dispositivo móvil se recibió una descripción favorable en términos generales de la aplicación. Es posible ver, en la figura 5.3, que en todas las áreas se presentan resultados aceptables. Sin embargo, tanto para la pregunta

2 como para la pregunta 4 existieron participantes que evaluaron de forma deficiente la aplicación, tal como se aprecia en la figura 5.2.

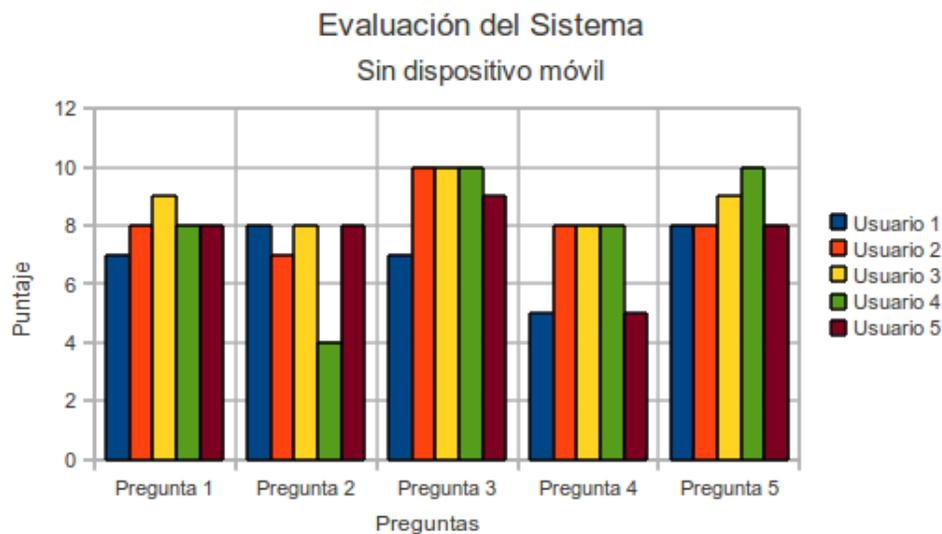


Figura 5.2: Descripción de resultados generales sin uso de un dispositivo móvil.

Para comprender la razón de esta deficiencia se analizaron los comentarios emitidos por las personas que dieron esta evaluación insuficiente, encontrando lo siguiente:

Pregunta 2: Esta pregunta hace referencia a la claridad de la información presentada.

Según los comentarios expuestos, la calidad del medio usado para entregar la información pareció baja. Esto puede deberse a que se trata de un sistema Text-To-Speech, cuyo sonido artificial puede incomodar a algunos usuarios. Por otro lado se comenta que la aplicación hablaba en inglés, que a pesar de ser un idioma dominado por todos los participantes de la muestra, no corresponde a su idioma nativo, lo que puede afectar la muestra.

Pregunta 4: Esta pregunta pretende recoger la percepción acerca de la rapidez con que se entrega la información. Los usuarios que calificaron de forma insuficiente al sistema en esta área, notaban que a veces el sistema presentaba información en un orden no conveniente. Esto ocurre particularmente cuando se despliega toda la información, no sólo la de los objetos que se encuentran enfrente. Por esta razón, la solución que se propone en este caso es, en primer lugar, filtrar la información, de forma de dar aviso sólo de aquellos objetos que se encuentren en la dirección de desplazamiento

del usuario. Esto ya está implementado en el sistema. Lamentablemente, debido a la limitación de tiempo no pudo rehacerse la experiencia con este cambio, por lo que se deja como trabajo futuro. Otra sugerencia, que fue planteada por un usuario, para el caso en el que se despliega toda la información del entorno, consiste en cambiar el orden en el que se presentan los datos. Hasta ahora el sistema funciona como una cola FIFO. Tomando en cuenta el comentario hecho por el usuario, se propone cambiar la distribución de los mensajes. Esto se puede hacer asignándole una prioridad a cada mensaje. En particular, los mensajes referidos a objetos que se encuentren en la dirección de desplazamiento deben ser mencionados primero. Se estudió la factibilidad de hacer esto con la librería utilizada, encontrando que esto se puede hacer solamente si se le asigna otra voz al mensaje urgente. La ejecución de esta solución queda propuesta. Por otro lado, los usuarios plantearon la idea de usar otros medios de comunicación más rápidos, como vibraciones. La limitación que esto presenta es que se perdería información importante, que no puede ser transmitida a través de vibraciones, como los nombres de los objetos.

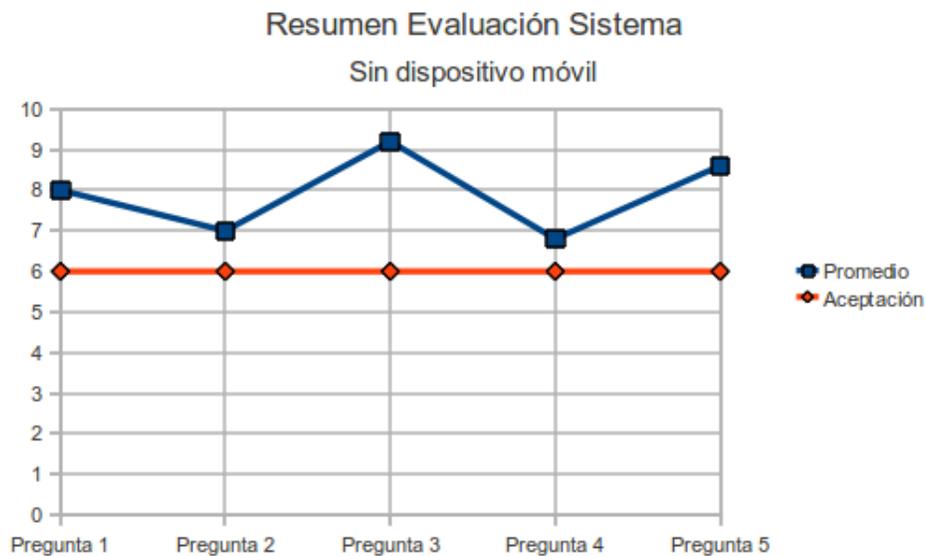


Figura 5.3: Resumen de los resultados sin hacer uso de un dispositivo móvil.

5.2.2. Resultados haciendo uso del dispositivo móvil

Los resultados haciendo uso del dispositivo móvil no variaron mucho en cuanto a la evaluación general del sistema, como se puede apreciar al comparar las gráficas 5.4 y 5.5 con las de la sección anterior. Aún así, llaman la atención nuevamente los resultados de las preguntas 2 y 4. Mientras que en la pregunta 2 la evaluación bajó considerablemente, llegando a un nivel inaceptable, en la pregunta 4 se logró una pequeña alza.

Pregunta 2: Esta pregunta sufrió una caída en la evaluación, con respecto al caso sin dispositivo móvil. Al revisar los comentarios se notó una particularidad que ocurrió al desplegar los datos desde el dispositivo. Cuando los datos eran entregados por el dispositivo móvil, estos eran reproducidos de tal forma que el dispositivo no esperaba que terminara de darse una instrucción para dar la otra; esto afectó considerablemente el desempeño de aquellos usuarios que optaban por presionar de forma constante el botón de interacción puesto en el bastón. Para reparar esto se modificó la clase servidor, de tal forma que ahora reproduzca un mensaje a la vez.

Pregunta 4: En este caso, la evaluación sufrió una mejora, curiosamente por la misma razón que perjudicó a la pregunta 2. En conversaciones informales con los usuarios se notó que percibieron que al entregarse la información de forma asíncrona, todo se entregaba más rápido, aunque les costaba mucho más entender la información. A pesar de esta pequeña mejora en la evaluación, la opción de reproducir de forma asíncrona la información fue descartada, debido a los problemas que genera en los otros aspectos evaluados.

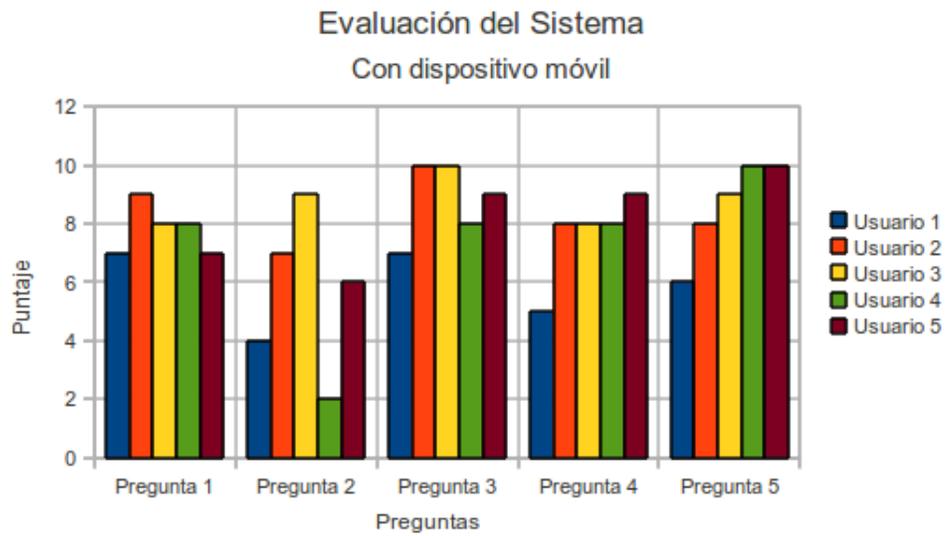


Figura 5.4: Descripción de resultados generales haciendo uso de un dispositivo móvil.

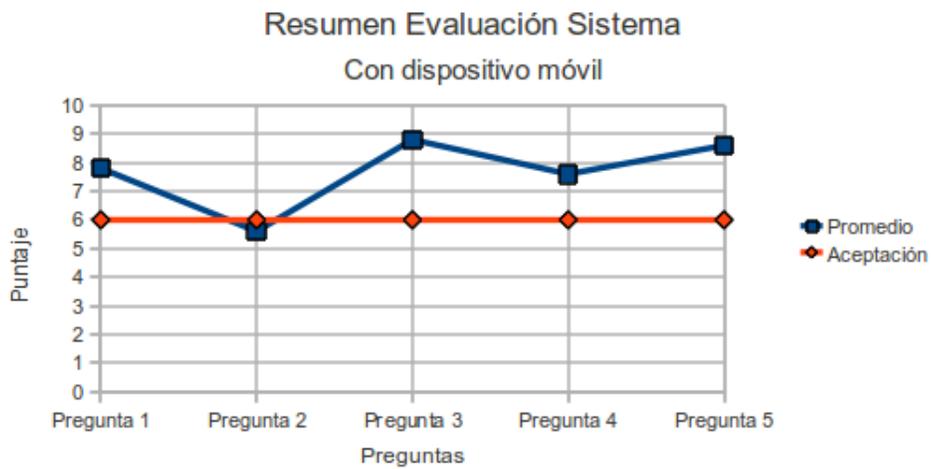


Figura 5.5: Resumen de los resultados haciendo uso de un dispositivo móvil.

Capítulo 6

Conclusiones

Las pruebas preliminares realizadas al sistema son bastante alentadoras, puesto que muestran la correcta funcionalidad de éste. Sin embargo, es importante continuar realizando pruebas en entornos reales, con grupos cada vez más grandes y representativos de las personas con discapacidad visual para, de esta forma, no sólo mostrar la correctitud de la aplicación, sino también la generalidad y la utilidad de ésta. Experimentos más rigurosos deben ser diseñados y realizados. Sin embargo, el prototipo desarrollado cumplió con las expectativas en términos generales.

Un aspecto que se debe mejorar es el de la transmisión de los datos al usuario. Es en esta parte donde se recibieron las mayores críticas. Aunque esta parte no es directamente administrada por el sistema, puesto que se utilizaron librerías de text-to-speech, se debe continuar la búsqueda de una que satisfaga a los usuarios, o en último caso desarrollar una aplicación de este tipo que satisfaga los requerimientos de los usuarios.

Por otro lado, es importante notar que se ha desarrollado un sistema accesible en cuanto a las herramientas y tecnología utilizadas, de forma que pueda ser efectivamente implementado con un costo muy bajo. El prototipo desarrollado muestra que es posible realizar un buen aporte a la integración e independencia de las personas con discapacidad visual, que deben movilizarse en entornos cerrados.

Un aspecto importante a destacar, es que la solución propuesta no solamente le permite a un usuario con discapacidad visual moverse en un entorno cerrado evitando obstáculos. También le podría ayudar a interactuar con el entorno, dado que se tienen mapeados todos los objetos que allí se encuentran.

Se reconoce que, al ser un primer prototipo, aún existen mejoras y desafíos que deben ser considerados para robustecer la solución propuesta. Las mejoras propuestas se resumen a continuación:

- Crear una aplicación gráfica para la generación de los documentos que representan los entornos cerrados.
- Cambiar el orden de entrega de información de un sistema FIFO a un sistema FIFO con prioridad.
- Separar de forma clara, en la interfaz gráfica y en el sistema, las opciones de reconocer obstáculos y de presentar la información del entorno, de forma tal que, en caso de percibir los objetos como obstáculos, se pueda cambiar las alertas de un sistema con voz, a otro, como por ejemplo, vibración, tal como fue propuesto durante las pruebas.
- Integrar más funcionalidades a la interfaz asociada al objeto aumentado. En particular se desea permitir que el usuario de forma directa modifique los parámetros del sistema según su conveniencia, sin la intervención de terceros.

Para finalizar, se destaca que esta solución, aplicada a la ubicación de personas, tiene otras aplicaciones. Durante las pruebas se notó que la herramienta puede ser utilizada incluso como pizarra digital, con una resolución aceptable y una respuesta rápida. Las posibles aplicaciones extra de esta solución, junto con su conveniencia frente a otras soluciones quedan propuestas como trabajo futuro.

Bibliografía

- [1] ENABLED – Enhanced Network Accessibility for the Blind and Visually Impaired. <http://www.enabledweb.org>. European Project Number FP6-2003-IST-2-004778.
- [2] Nicholas A. Bradley and Mark D. Dunlop. An experimental investigation into wayfinding directions for visually impaired people. *Comp*, 9:395–403, 2005.
- [3] Vlad Coroama. The chatty environment – a world explorer for the visually impaired. In *Adjunct Proceedings of Ubicomp 2003*, pages 221–222, Seattle, Washington, USA, October 2003.
- [4] Instituto Nacional de Estadísticas. Censo 2002, población con discapacidades, por sexo y tipo de discapacidad, según grupos de edad. <http://www.ine.cl/cd2002/index.php>.
- [5] Ochoa S. Guerrero L. A. and Horta H. Developing augmented objects: A process perspective. *Journal of Universal Computer Science*, 16(12):1612–1632, 2010.
- [6] Simon Holland, David R. Morse, and Henrik Gedenryd. Audio GPS: Spatial audio in a minimal attention interface. In *3rd International Workshop on HCI with Mobile Devices*, pages 28–33, 2001.
- [7] A. Hub, T. Hartter, and T. Ertl. Interactive Localization and Recognition of Objects for the Blind. In *Proceedings of the California State University, Northridge Center on Disabilities’ 21st Annual International Technology and Persons with Disabilities Conference (CSUN 2006); March 20-25; Los Angeles, CA, USA*, 2006.
- [8] Andreas Hub, Joachim Diepstraten, and Thomas Ertl. Design and development of an indoor navigation and object identification system for the blind. In Julie A. Jacko and Andrew Sears, editors, *Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility, ASSETS 2004, Atlanta, GA, USA, October 18-20, 2004*, pages 147–152. ACM, 2004.

-
- [9] Hiroshi Ishii and Brygg Ullmer. Tangible bits: Towards seamless interfaces between people, bits and atoms. In Steven Pemberton NYNEX Science and Technology, editors, *CHI '97 Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 234–241. ACM New York, NY, USA, 1997.
- [10] William R. Shadish, Thomas D. Cook, and Donald T. Campbell. *Experimental and Quasi-Experimental Designs for Generalized Causal Inference*. Houghton Mifflin Company, 2 edition, July 2001.
- [11] Y Sonnenblick. An indoor navigation system for blind individuals. *Proceedings of the 13th Annual Conference on Technology and Persons with Disabilities*, 1998.
- [12] J. Sánchez and C. Oyarzún. Asistencia móvil basada en audio para la movilización por medio de microbús de personas ciegas. In Sánchez J., editor, *Nuevas Ideas en Informática Educativa*, pages 377–396. Lom Ediciones S.A., 2007.
- [13] J. Sánchez and M. Sáenz. Orientación y movilidad en espacios exteriores para aprendices ciegos con el uso de dispositivos móviles. *Revista Anales de la Universidad metropolitana*, 8(2):47–66, 2008. Venezuela.
- [14] Amanda Williams and Daniela Karin Rosner. Wiimote hackery studio proposal. In Marcelo Coelho, Jamie Zigelbaum, Hiroshi Ishii, Robert J. K. Jacob, Pattie Maes, Thomas Pederson, Orit Shaer, and Ron Wakkary, editors, *Proceedings of the 4th International Conference on Tangible and Embedded Interaction 2010, Cambridge, MA, USA, January 24-27, 2010*, pages 365–368. ACM, 2010.
- [15] Hua Wu, Alan Marshall, Wai Yu, and Yun-Maw Cheng. Applying HTA method to the design of context-aware indoor navigation for the visually-impaired. In *Mobility Conference'07*, pages 632–635, Singapore, 2007.

Apéndices

A . Interfaz de Administración

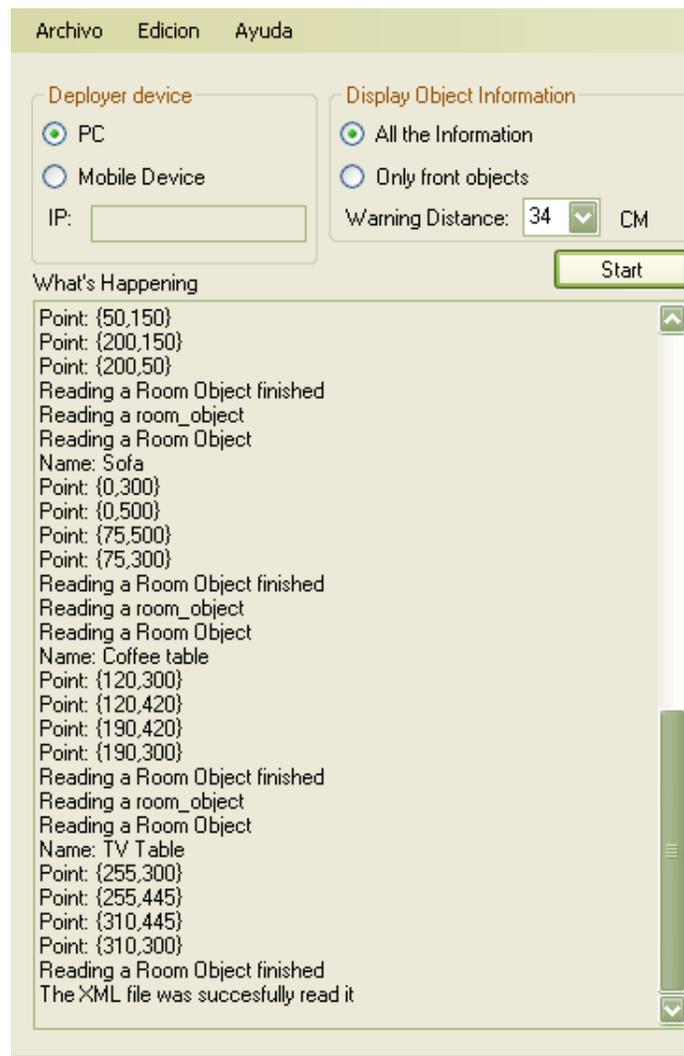


Figura 6.1: Vista de la sección de entrega de parámetros de la interfaz de administración.

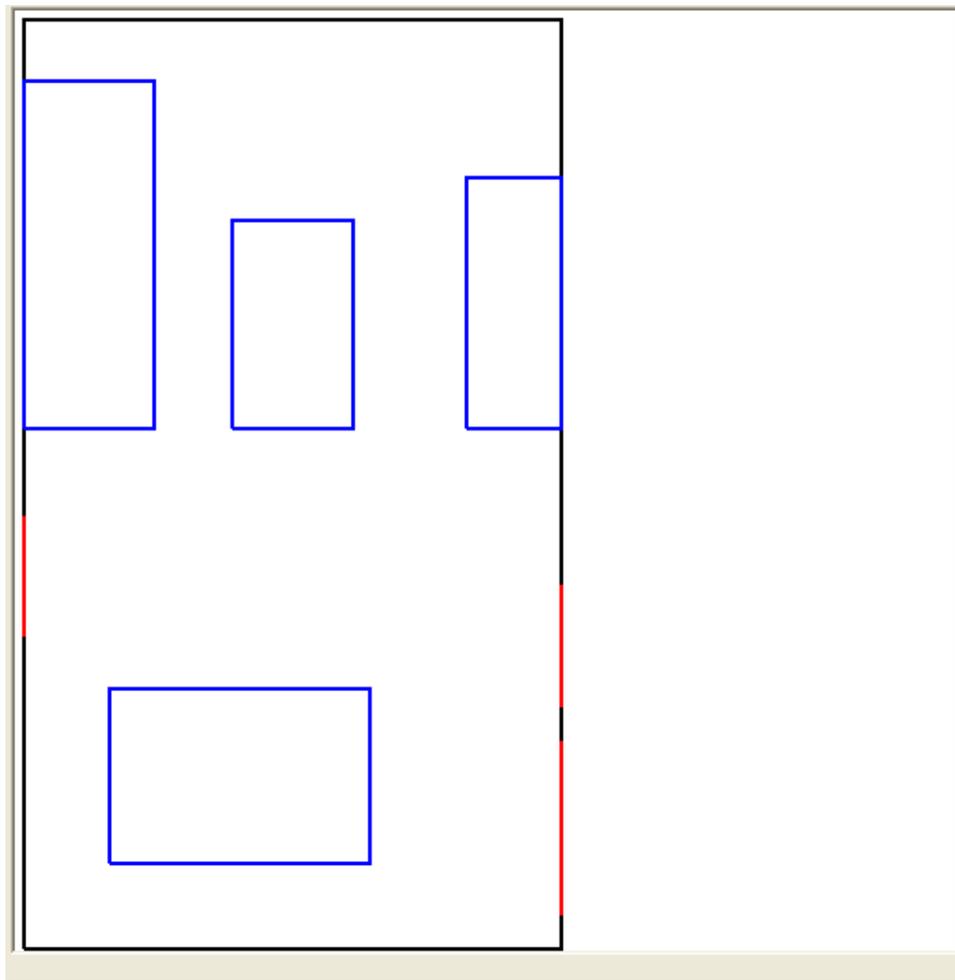


Figura 6.2: Mapa de habitación desplegado en la interfaz de administración.

B . Documentos XML para representar entornos cerrados

B .1. Esquema para los documentos XML

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <xs:schema id="indoorMaps"
3   elementFormDefault="qualified"
4   xmlns:xs="http://www.w3.org/2001/XMLSchema"
5 >
6   <!-- definition of simple elements -->
7
8   <xs:element name="x_axis" type="xs:float"/>
9   <xs:element name="y_axis" type="xs:float"/>
10  <xs:element name="wiimote_id" type="xs:string"/>
11  <xs:element name="object_name" type="xs:string"/>
12  <xs:element name="angle" type="xs:float"/>
13  <xs:annotation>
14    <xs:appinfo>
15      IndoorPositionFinder direction
16    </xs:appinfo>
17    <xs:documentation>
18      Defines a kind of direction to the wiimotes. The meaning are the
19      following:
20      1: The wiimote is pointing in a positive direction in the X and Y axis.
21      2: The wiimote is pointing in a positive direction in the X axis and in
22      a negative direction in the Y axis.
23      3: The wiimote is pointing in a negative direction in the X and Y axis.
24      4: The wiimote is pointing in a negative direction in the X axis and in
25      a positive direction in the Y axis.
26      All the values are relative to the axis associated with the room.
27    </xs:documentation>
28  </xs:annotation>
29  <xs:element name="direction">
30    <xs:simpleType>
31      <xs:restriction base="xs:integer">
32        <xs:pattern value="1|2|3|4"/>
33      </xs:restriction>
34    </xs:simpleType>
35  </xs:element>
36
37  <!-- definition of atributtes -->
38
39  <!-- definition of complex elements -->
40
41  <xs:annotation>
42    <xs:appinfo>
43      IndoorPositionFinder point
44    </xs:appinfo>
45    <xs:documentation xml:lang="en">
46      Defines a point in two dimensions
47    </xs:documentation>
48  </xs:annotation>
49  <xs:element name="point">
50    <xs:complexType>
51      <xs:sequence>
52        <xs:element ref="x_axis"/>
53        <xs:element ref="y_axis"/>

```

```
51     </xs:sequence>
52   </xs:complexType>
53 </xs:element>
54
55 <xs:annotation>
56   <xs:appinfo>
57     IndoorPositionFinder Wiimote
58   </xs:appinfo>
59   <xs:documentation xml:lang="en">
60     Defines a Wiimote. Requires an Id, an angle and a location point.
61   </xs:documentation>
62 </xs:annotation>
63 <xs:element name="wiimote">
64   <xs:complexType>
65     <xs:all>
66       <xs:element ref="wiimote_id" />
67       <xs:element ref="point" />
68       <xs:element ref="angle" />
69       <xs:element ref="direction" />
70     </xs:all>
71   </xs:complexType>
72 </xs:element>
73
74 <xs:annotation>
75   <xs:appinfo>
76     IndoorPositionFinder Room Object
77   </xs:appinfo>
78   <xs:documentation xml:lang="en">
79     Defines an object. Requires a name and four points minimum to know its
80     position. The point's definition starts in the left bottom point and
81     continues clockwise.
82   </xs:documentation>
83 </xs:annotation>
84 <xs:element name="room_object">
85   <xs:complexType>
86     <xs:sequence>
87       <xs:element ref="object_name" />
88       <xs:element ref="point" minOccurs="4" maxOccurs="unbounded" />
89     </xs:sequence>
90   </xs:complexType>
91 </xs:element>
92
93 <xs:annotation>
94   <xs:appinfo>
95     IndoorPositionFinder Door
96   </xs:appinfo>
97   <xs:documentation xml:lang="en">
98     Defines a door. Requires two points to define its position.
99   </xs:documentation>
100 </xs:annotation>
101 <xs:element name="room_door">
102   <xs:complexType>
103     <xs:sequence>
104       <xs:element ref="point" minOccurs="2" maxOccurs="2" />
105     </xs:sequence>
106   </xs:complexType>
107 </xs:element>
108
109 <xs:annotation>
```

```

110 <xs:appinfo>
111   IndoorPositionFinder Room
112 </xs:appinfo>
113 <xs:documentation xml:lang="en">
114   Defines a room. The room is the root object in the XML. Contains all
115   the other elements (doors, objects and wiimotes). Requires four points
116   and two wiimotes minimum for its definition. The other elements (doors
117   and objects) are optional but they are strongly recommended.
118   The points need to be defined clockwise starting with the left bottom
119   point.
120 </xs:documentation>
121 </xs:annotation>
122 <xs:element name="room">
123   <xs:complexType>
124     <xs:sequence>
125       <xs:element ref="point" minOccurs="4" maxOccurs="unbounded"/>
126       <xs:element ref="wiimote" minOccurs="2" maxOccurs="unbounded"/>
127       <xs:element ref="room_door" minOccurs="1" maxOccurs="unbounded"/>
128       <xs:element ref="room_object" minOccurs="1" maxOccurs="unbounded" />
129     </xs:sequence>
130   </xs:complexType>
131 </xs:element>
</xs:schema>

```

B .2. Ejemplo de documento XML

```

1 <?xml version="1.0" encoding="utf-8" ?>
2
3
4 <room xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5 xsi:noNamespaceSchemaLocation="indoorMaps.xsd">
6   <!-- Defining the points for the room -->
7   <point>
8     <x_axis>0</x_axis>
9     <y_axis>0</y_axis>
10  </point>
11  <point>
12    <x_axis>0</x_axis>
13    <y_axis>535</y_axis>
14  </point>
15  <point>
16    <x_axis>310</x_axis>
17    <y_axis>535</y_axis>
18  </point>
19  <point>
20    <x_axis>310</x_axis>
21    <y_axis>0</y_axis>
22  </point>
23  <!-- Defining the existing wiimotes in the livingroom -->
24  <wiimote>
25    <wiimote_id>\\?\hid#vid_057e&pid_0306#2&203dff11&0&0000#{4
26      d1e55b2-f16f-11cf-88cb-001111000030}</wiimote_id>
27    <point>
28      <x_axis>0</x_axis>
29      <y_axis>0</y_axis>
30    </point>

```

```
30     <angle>60</angle>
31     <direction>1</direction>
32 </wiimote>
33 <wiimote>
34     <wiimote_id>\\?\hid#vid_057e&pid_0306#2&32f7afb6&0&0000#{4
        d1e55b2-f16f-11cf-88cb-001111000030}</wiimote_id>
35     <point>
36         <x_axis>0</x_axis>
37         <y_axis>535</y_axis>
38     </point>
39     <angle>30</angle>
40     <direction>2</direction>
41 </wiimote>
42 <!-- Defining doors and objects -->
43 <!-- A la pieza-->
44 <room_door>
45     <point>
46         <x_axis>0</x_axis>
47         <y_axis>180</y_axis>
48     </point>
49     <point>
50         <x_axis>0</x_axis>
51         <y_axis>250</y_axis>
52     </point>
53 </room_door>
54 <!-- Al pasillo-->
55 <room_door>
56     <point>
57         <x_axis>310</x_axis>
58         <y_axis>20</y_axis>
59     </point>
60     <point>
61         <x_axis>310</x_axis>
62         <y_axis>120</y_axis>
63     </point>
64 </room_door>
65 <!-- A la cocina-->
66 <room_door>
67     <point>
68         <x_axis>310</x_axis>
69         <y_axis>140</y_axis>
70     </point>
71     <point>
72         <x_axis>310</x_axis>
73         <y_axis>210</y_axis>
74     </point>
75 </room_door>
76 <!-- Comedor-->
77 <room_object>
78     <object_name>Eating table</object_name>
79     <point>
80         <x_axis>50</x_axis>
81         <y_axis>50</y_axis>
82     </point>
83     <point>
84         <x_axis>50</x_axis>
85         <y_axis>150</y_axis>
86     </point>
87     <point>
```

```
88     <x_axis>200</x_axis>
89     <y_axis>150</y_axis>
90 </point>
91 <point>
92     <x_axis>200</x_axis>
93     <y_axis>50</y_axis>
94 </point>
95 </room_object>
96 <!--Sofa-->
97 <room_object>
98     <object_name>Sofa</object_name>
99     <point>
100     <x_axis>0</x_axis>
101     <y_axis>300</y_axis>
102 </point>
103 <point>
104     <x_axis>0</x_axis>
105     <y_axis>500</y_axis>
106 </point>
107 <point>
108     <x_axis>75</x_axis>
109     <y_axis>500</y_axis>
110 </point>
111 <point>
112     <x_axis>75</x_axis>
113     <y_axis>300</y_axis>
114 </point>
115 </room_object>
116 <!--Mesa de Centro-->
117 <room_object>
118     <object_name>Coffee table</object_name>
119     <point>
120     <x_axis>120</x_axis>
121     <y_axis>300</y_axis>
122 </point>
123 <point>
124     <x_axis>120</x_axis>
125     <y_axis>420</y_axis>
126 </point>
127 <point>
128     <x_axis>190</x_axis>
129     <y_axis>420</y_axis>
130 </point>
131 <point>
132     <x_axis>190</x_axis>
133     <y_axis>300</y_axis>
134 </point>
135 </room_object>
136 <!--Mueble-->
137 <room_object>
138     <object_name>TV Table</object_name>
139     <point>
140     <x_axis>255</x_axis>
141     <y_axis>300</y_axis>
142 </point>
143 <point>
144     <x_axis>255</x_axis>
145     <y_axis>445</y_axis>
146 </point>
```

```
147     <point>
148         <x_axis>310</x_axis>
149         <y_axis>445</y_axis>
150     </point>
151     <point>
152         <x_axis>310</x_axis>
153         <y_axis>300</y_axis>
154     </point>
155 </room_object>
156 </room>
```

C . Códigos Fuente

C .1. Cálculo de la posición del usuario

```

1  /// <summary>
2  /// Return the position associated to the user relative to the room axis
3  /// </summary>
4  /// <param name="firstWiimoteId">NON-Permanent ID associated with the first
   wiimote</param>
5  /// <param name="secondWiimoteId">NON-Permanent ID associated with the second
   wiimote</param>
6  /// <param name="firstPosition">Position captured by the first wiimote</param>
7  /// <param name="secondPosition">Position captured by the second wiimote</
   param>
8  /// <returns></returns>
9  public PointF userPosition(Guid firstWiimoteId , Guid secondWiimoteId , PointF
   firstPosition , PointF secondPosition)
10 {
11     // Identifying the RoomWiimote associate with the Wiimotelib.Wiimote
12     RoomWiimote firstRoomWiimote = roomWiimotes[wiimotesIDs[firstWiimoteId]];
13     RoomWiimote secondRoomWiimote = roomWiimotes[wiimotesIDs[secondWiimoteId]];
14     // First , we use the coordinates from the different wiimotes to obtain
15     // the angle between the wiimote's visibility border vector and the
   position vector.
16     float firstPointAngle = positionToAngle(firstPosition.X);
17     float secondPointAngle = positionToAngle(secondPosition.X);
18     // Calculating the projection points for each Wiimote
19     PointF firstProjectionPoint = firstRoomWiimote.projectionPoint(roomPoints ,
   firstPointAngle);
20     PointF secondProjectionPoint = secondRoomWiimote.projectionPoint(roomPoints
   , secondPointAngle);
21     // IN THIS FINAL PART THE ALGORITHM CALCULATE THE USER POSITION
   INTERSECTING THE STRAIGHT LINES
22     // GENERATED BETWEEN THE WIIMOTE POINT AND ITS RESPECTIVE PROJECTION
23     // calculating the slopes (Y2-Y1)/(X2-X1)
24     float firstSlope = (firstProjectionPoint.Y - firstRoomWiimote.Position.Y) /
   (firstProjectionPoint.X - firstRoomWiimote.Position.X);
25     float secondSlope = (secondProjectionPoint.Y - secondRoomWiimote.Position.Y
   ) /
26         (secondProjectionPoint.X - secondRoomWiimote.Position.X);
27
28
29     // Position Calculation
30     PointF userPosition = new PointF();
31     userPosition.X = (secondSlope * secondRoomWiimote.Position.X -
   firstSlope * firstRoomWiimote.Position.X +
32         firstRoomWiimote.Position.Y - secondRoomWiimote.Position.Y) /
   (secondSlope - firstSlope);
33
34     userPosition.Y = secondRoomWiimote.Position.Y + secondSlope *
   (userPosition.X - secondRoomWiimote.Position.X);
35
36     return userPosition;
37
38 }
39

```

C .2. Método para actualizar posición del usuario

```

1  public System.Drawing.PointF changeChecker(bool onlyFront, bool device, int
2      warningDistance = 20, String ip = "EOF")
3  {
4      int wiimoteChanged = 0;
5      PointF [] position = new PointF [2];
6      Guid [] wiimoteId = new Guid [2];
7      bool updatePosition = false;
8      foreach (KeyValuePair<Guid, WiimoteState> dictionary in wiimoteStates)
9      {
10         Guid tempGuid = dictionary.Key;
11         WiimoteState ws = dictionary.Value;
12         IRSensor irSensor = ws.IRState.IRSensors [0];
13         if (irSensor.Found)
14         {
15             wiimoteId [wiimoteChanged] = tempGuid;
16             position [wiimoteChanged++] = irSensor.Position;
17             // The update in the user position is produced only if
18             // at least two wiimotes can capture the user position
19             if (wiimoteChanged >= 2)
20             {
21                 updatePosition = true;
22                 break;
23             }
24         }
25     }
26     // Here we put the logic required to warning to the user about any change
27     // in his relation with the environment
28     if (updatePosition)
29     {
30         Debug.WriteLine("Updating position");
31         PointF oldPosition = this.lastUserPosition;
32         // calculating the new user position
33         this.lastUserPosition = this.userPosition (wiimoteId [0], wiimoteId [1],
34             position [0], position [1]);
35         // calculating the movement vector
36         PointF movementVector = new PointF ();
37         movementVector.X = this.lastUserPosition.X - oldPosition.X;
38         movementVector.Y = this.lastUserPosition.Y - oldPosition.Y;
39         Debug.WriteLine(movementVector);
40         if (!this.lastUserPosition.Equals (oldPosition) &&
41             movementVector.X > 0.3 &&
42             movementVector.Y > 0.3)
43         {
44             movementVector.X = this.lastUserPosition.X - oldPosition.X;
45             movementVector.Y = this.lastUserPosition.Y - oldPosition.Y;
46             Debug.WriteLine(movementVector.ToString());
47             // calculating the centroid of the object and checking the distance
48             // between the user an
49             // the object's centroid. If the distance is smaller than
50             // warningDistance the program alert
51             // the user about the proximity
52             foreach (RoomObject roomObject in roomObjects)
53             {
54                 // float distance = roomObject.distance (lastUserPosition);
55                 // Truncate the number into a two decimals number
56                 // If the object is too close

```

```
53     bool toClose = roomObject.toClose(warningDistance, this.  
54         lastUserPosition);  
55     if (toClose)  
56     {  
57         String objectName = roomObject.Name;  
58         float angle = roomObject.relativeAngle(lastUserPosition,  
59             movementVector);  
60         if (onlyFront)  
61         {  
62             if (angle == 12)  
63             {  
64                 if (device)  
65                     notifier.speak("You have a " + objectName + " in front  
66                         ");  
67                 else  
68                     this.sendInfoToDevice(ip, "You have a " + objectName + "  
69                         " in front");  
70             }  
71         }  
72         else  
73         {  
74             if (device)  
75                 notifier.speak("You have a " + objectName + " at " +  
76                     angle);  
77             else  
78                 this.sendInfoToDevice(ip, "You have a " + objectName + "  
79                     at " + angle);  
80         }  
81     }  
82 }  
83 // now the same with the doors  
84 foreach (RoomDoor roomDoor in roomDoors)  
85 {  
86     float distance = roomDoor.distance(lastUserPosition);  
87     // Truncate the number into a two decimals number  
88     distance = (float)Math.Truncate((double)distance * 100.0) / 100f;  
89     // If the door is too close  
90     if (distance < warningDistance)  
91     {  
92         float angle = roomDoor.relativeAngle(lastUserPosition,  
93             movementVector);  
94         if (onlyFront)  
95         {  
96             if (angle == 12)  
97             {  
98                 if (device)  
99                     notifier.speak("You have a door at " + distance + "  
100                         centimeters");  
101                 else  
102                     this.sendInfoToDevice(ip, "You have a door at " +  
103                         distance + " centimeters");  
104             }  
105         }  
106         else  
107         {  
108             if (device)  
109                 notifier.speak("You have at " + angle + " a door at " +  
110                     distance + " centimeters");  
111             else
```

```
102         this.sendInfoToDevice(ip, "You have at " + angle + " a  
103             door at " + distance + " centimeters");  
104     }  
105 }  
106     return new System.Drawing.PointF(lastUserPosition.X, lastUserPosition  
107         .Y);  
108 }  
109     else  
110         return new System.Drawing.PointF(-1f, -1f);  
111 }  
112     else  
113         return new System.Drawing.PointF(-1f, -1f);  
114 }
```

D . Resultados de las pruebas realizadas

D .1. Sistema transmitiendo datos a través de parlantes

Usuario 1

- **Sexo:** Femenino.

- **Edad:** 28 años.

1. De una escala de 1 a 10, donde 1 corresponde a muy compleja y 10 corresponde a muy sencilla, ¿cómo evaluaría la complejidad de la interacción con el sistema mientras lo usó?

R: Nivel 7

2. De una escala de 1 a 10, donde 1 corresponde a incomprendible y 10 corresponde a claro, ¿cuán comprensible le pareció la calidad de la entrega de información del sistema?

R: Nivel 8. La información es concreta y breve.

3. De una escala de 1 a 10, donde 1 corresponde a inútil y 10 corresponde a muy útil, ¿cuán útil le pareció la información entregada por el sistema para desplazarse?

R: Nivel 7.

4. De una escala de 1 a 10, donde 1 corresponde a muy retardada y 10 corresponde a inmediata, ¿cuán rápida le pareció la entrega de información del sistema?

R: Nivel 5. A Menudo el sistema entregaba la información cuando el usuario ya había chocado con la silla.

5. De una escala de 1 a 10, donde 1 corresponde a muy imprecisa y 10 corresponde a exacta, ¿cuán precisa le pareció la información del entorno entregada por el sistema?

R: Nivel 8

Usuario 2

- **Sexo:** Masculino.

- **Edad:** 24 años.

1. De una escala de 1 a 10, donde 1 corresponde a muy compleja y 10 corresponde a muy sencilla, ¿cómo evaluaría la complejidad de la interacción con el sistema mientras lo usó?

R:8, ya que a veces el sistema no respondía en forma inmediata, pero esto ocurrió pocas veces, cuando no llegaba la señal al sensor. En general no es complejo usar el sistema, una vez familiarizado con él las respuestas son muy útiles para ubicarse en el espacio.

2. De una escala de 1 a 10, donde 1 corresponde a incomprensible y 10 corresponde a claro, ¿cuán comprensible le pareció la calidad de la entrega de información del sistema?

R:7, la calidad de la información fue buena, pero en algunos momentos si presionaba muy seguido el botón, llegaba mucha información a la vez.

3. De una escala de 1 a 10, donde 1 corresponde a inútil y 10 corresponde a muy útil, ¿cuán útil le pareció la información entregada por el sistema para desplazarse?

R:10, la información entregada por el sistema fue bastante útil ya que indicaba qué objeto y en qué lugar estaba ubicado, permitiendo al usuario encontrar un camino sin obstáculos.

4. De una escala de 1 a 10, donde 1 corresponde a muy retardada y 10 corresponde a inmediata, ¿cuán rápida le pareció la entrega de información del sistema?

R: 8, la información en general llega a tiempo.

5. De una escala de 1 a 10, donde 1 corresponde a muy imprecisa y 10 corresponde a exacta, ¿cuán precisa le pareció la información del entorno entregada por el sistema?

R:8, la información en general es bastante precisa, sin embargo existen algunos “puntos ciegos” donde el sistema no entrega información.

Usuario 3

- **Sexo:** Masculino.
- **Edad:** 28 años.

1. De una escala de 1 a 10, donde 1 corresponde a muy compleja y 10 corresponde a muy sencilla, ¿cómo evaluaría la complejidad de la interacción con el sistema mientras lo usó?

R: 9

2. De una escala de 1 a 10, donde 1 corresponde a incomprendible y 10 corresponde a claro, ¿cuán comprensible le pareció la calidad de la entrega de información del sistema?

R: 9

3. De una escala de 1 a 10, donde 1 corresponde a inútil y 10 corresponde a muy útil, ¿cuán útil le pareció la información entregada por el sistema para desplazarse?

R: 10

4. De una escala de 1 a 10, donde 1 corresponde a muy retardada y 10 corresponde a inmediata, ¿cuán rápida le pareció la entrega de información del sistema?

R: 8

5. De una escala de 1 a 10, donde 1 corresponde a muy imprecisa y 10 corresponde a exacta, ¿cuán precisa le pareció la información del entorno entregada por el sistema?

R: 9

Usuario 4

- **Sexo:** Masculino.
- **Edad:** 28 años.

1. De una escala de 1 a 10, donde 1 corresponde a muy compleja y 10 corresponde a muy sencilla, ¿cómo evaluaría la complejidad de la interacción con el sistema mientras lo usó?

R: 8. El sistema no es complejo, pero tampoco resulta intuitivo. Falta mejorar usabilidad con otros tipos de pruebas.

2. De una escala de 1 a 10, donde 1 corresponde a incomprendible y 10 corresponde a claro, ¿cuán comprensible le pareció la calidad de la entrega de información del sistema?

R: 4. Calidad de entrega baja, falta estudio de alternativas para proporcionar la información. El delay o la forma de entregarla no es la mejor por lo que dificulta la comprensión.

3. De una escala de 1 a 10, donde 1 corresponde a inútil y 10 corresponde a muy útil, ¿cuán útil le pareció la información entregada por el sistema para desplazarse?

R: 10. La información que se entrega es muy útil si se lograra identificar objetos prioritarios y evitar entregar exceso de información. El mapa mental que se generaba con la información resultaba intuitivo.

4. De una escala de 1 a 10, donde 1 corresponde a muy retardada y 10 corresponde a inmediata, ¿cuán rápida le pareció la entrega de información del sistema?

R: 8. Información debe ser identificada de alguna manera como prioritaria frente a otra. La forma de audio no es la mejor para entregar la información.

5. De una escala de 1 a 10, donde 1 corresponde a muy imprecisa y 10 corresponde a exacta, ¿cuán precisa le pareció la información del entorno entregada por el sistema?

R:10. Muy exacta respecto a la orientación y espacio físico representado.

Usuario 5

- **Sexo:** Femenino.
- **Edad:** 27 años.

1. De una escala de 1 a 10, donde 1 corresponde a muy compleja y 10 corresponde a muy sencilla, ¿cómo evaluaría la complejidad de la interacción con el sistema mientras lo usó?

R: 8

2. De una escala de 1 a 10, donde 1 corresponde a incomprensible y 10 corresponde a claro, ¿cuán comprensible le pareció la calidad de la entrega de información del sistema?

R: 8

3. De una escala de 1 a 10, donde 1 corresponde a inútil y 10 corresponde a muy útil, ¿cuán útil le pareció la información entregada por el sistema para desplazarse?

R: 9

4. De una escala de 1 a 10, donde 1 corresponde a muy retardada y 10 corresponde a inmediata, ¿cuán rápida le pareció la entrega de información del sistema?

R: 5

5. De una escala de 1 a 10, donde 1 corresponde a muy imprecisa y 10 corresponde a exacta, ¿cuán precisa le pareció la información del entorno entregada por el sistema?

R: 8

D .2. Transmisión de datos usando dispositivo móvil

Usuario 1

- **Sexo:** Femenino.
- **Edad:** 28 años.

1. De una escala de 1 a 10, donde 1 corresponde a muy compleja y 10 corresponde a muy sencilla, ¿cómo evaluaría la complejidad de la interacción con el sistema mientras lo usó?

R: Nivel 7

2. De una escala de 1 a 10, donde 1 corresponde a incomprendible y 10 corresponde a claro, ¿cuán comprensible le pareció la calidad de la entrega de información del sistema?

R: Nivel 4. Las respuestas se acoplaban y costaba entender que decía.

3. De una escala de 1 a 10, donde 1 corresponde a inútil y 10 corresponde a muy útil, ¿cuán útil le pareció la información entregada por el sistema para desplazarse?

R: Nivel 7

4. De una escala de 1 a 10, donde 1 corresponde a muy retardada y 10 corresponde a inmediata, ¿cuán rápida le pareció la entrega de información del sistema?

R: Nivel 5

5. De una escala de 1 a 10, donde 1 corresponde a muy imprecisa y 10 corresponde a exacta, ¿cuán precisa le pareció la información del entorno entregada por el sistema?

R: Nivel 6

Usuario 2

- **Sexo:** Masculino.
- **Edad:** 24 años.

1. De una escala de 1 a 10, donde 1 corresponde a muy compleja y 10 corresponde a muy sencilla, ¿cómo evaluaría la complejidad de la interacción con el sistema mientras lo usó?

R:9, la interacción con el sistema fue buena.

2. De una escala de 1 a 10, donde 1 corresponde a incomprendible y 10 corresponde a claro, ¿cuán comprensible le pareció la calidad de la entrega de información del sistema?

R: 7, en ocasiones el sistema entregaba mucha información a la vez, lo cual puede confundir.

3. De una escala de 1 a 10, donde 1 corresponde a inútil y 10 corresponde a muy útil, ¿cuán útil le pareció la información entregada por el sistema para desplazarse?

R: 10, ya que sin el sistema no hubiese sido posible desplazarse sin tropezar.

4. De una escala de 1 a 10, donde 1 corresponde a muy retardada y 10 corresponde a inmediata, ¿cuán rápida le pareció la entrega de información del sistema?

R: 8, la información llegó a tiempo.

5. De una escala de 1 a 10, donde 1 corresponde a muy imprecisa y 10 corresponde a exacta, ¿cuán precisa le pareció la información del entorno entregada por el sistema?

R:8, la información entregada es precisa y útil, aunque en pocas ocasiones era confuso ubicarse según la posición de las manecillas del reloj.

Usuario 3

- **Sexo:** Masculino.

- **Edad:** 28 años.

1. De una escala de 1 a 10, donde 1 corresponde a muy compleja y 10 corresponde a muy sencilla, ¿cómo evaluaría la complejidad de la interacción con el sistema mientras lo usó?

R: 8

2. De una escala de 1 a 10, donde 1 corresponde a incomprendible y 10 corresponde a claro, ¿cuán comprensible le pareció la calidad de la entrega de información del sistema?

R: 9

3. De una escala de 1 a 10, donde 1 corresponde a inútil y 10 corresponde a muy útil, ¿cuán útil le pareció la información entregada por el sistema para desplazarse?

R: 10

4. De una escala de 1 a 10, donde 1 corresponde a muy retardada y 10 corresponde a inmediata, ¿cuán rápida le pareció la entrega de información del sistema?

R: 8

5. De una escala de 1 a 10, donde 1 corresponde a muy imprecisa y 10 corresponde a exacta, ¿cuán precisa le pareció la información del entorno entregada por el sistema?

R: 9

Usuario 4

- **Sexo:** Masculino.

- **Edad:** 28 años.

1. De una escala de 1 a 10, donde 1 corresponde a muy compleja y 10 corresponde a muy sencilla, ¿cómo evaluaría la complejidad de la interacción con el sistema mientras lo usó?

R: 8

2. De una escala de 1 a 10, donde 1 corresponde a incomprensible y 10 corresponde a claro, ¿cuán comprensible le pareció la calidad de la entrega de información del sistema?

R: 2. Demasiada información mal entregada. Principalmente por falta de iteraciones en el prototipo.

3. De una escala de 1 a 10, donde 1 corresponde a inútil y 10 corresponde a muy útil, ¿cuán útil le pareció la información entregada por el sistema para desplazarse?

R: 8

4. De una escala de 1 a 10, donde 1 corresponde a muy retardada y 10 corresponde a inmediata, ¿cuán rápida le pareció la entrega de información del sistema?

R: 8

5. De una escala de 1 a 10, donde 1 corresponde a muy imprecisa y 10 corresponde a exacta, ¿cuán precisa le pareció la información del entorno entregada por el sistema?

R: 10

Usuario 5

- **Sexo:** Femenino.
- **Edad:** 27 años.

1. De una escala de 1 a 10, donde 1 corresponde a muy compleja y 10 corresponde a muy sencilla, ¿cómo evaluaría la complejidad de la interacción con el sistema mientras lo usó?

R: 7.

2. De una escala de 1 a 10, donde 1 corresponde a incomprensible y 10 corresponde a claro, ¿cuán comprensible le pareció la calidad de la entrega de información del sistema?

R: 6.

3. De una escala de 1 a 10, donde 1 corresponde a inútil y 10 corresponde a muy útil, ¿cuán útil le pareció la información entregada por el sistema para desplazarse?

R: 9, a pesar de que por celular fue menos comprensible, la información sí fue mas precisa que por el PC.

4. De una escala de 1 a 10, donde 1 corresponde a muy retardada y 10 corresponde a inmediata, ¿cuán rápida le pareció la entrega de información del sistema?

R: 9

5. De una escala de 1 a 10, donde 1 corresponde a muy imprecisa y 10 corresponde a exacta, ¿cuán precisa le pareció la información del entorno entregada por el sistema?

R: 10, fue mucho mas exacta y menos tardia.