

UNIVERSIDAD AUTÓNOMA DE MADRID



ESCUELA POLITÉCNICA SUPERIOR

PROYECTO FIN DE CARRERA

**CARTOGRAFÍA ASISTIDA PARA PERSONAS CIEGAS, EN
TELÉFONOS O TABLETAS APPLE**

Andrés Romero Sánchez

Septiembre, 2014

**CARTOGRAFÍA ASISTIDA PARA PERSONAS CIEGAS, EN
TELÉFONOS O TABLETAS APPLE**

AUTOR: Andrés Romero Sánchez

TUTOR: Carlos Minchola

PONENTE: Eduardo Boemo

Digital System Lab

Departamento de Tecnología Electrónica y de Comunicaciones

Escuela Politécnica Superior

Universidad Autónoma de Madrid

2014

Agradecimientos Personales

Es de justicia comenzar dando las gracias a mi tutor, Carlos Minchola, por haberme dado la oportunidad de llevar a cabo este proyecto; y, por medio de él, a todos los profesores de la Escuela que han sido, en su conjunto, los que, con sus enseñanzas, han hecho posible que esté a punto de culminar, junto a otros compañeros, mi proyecto formativo de Ingeniería de Telecomunicación.

Gratitud particular merecen mis padres, Pedro y Ana, sin cuyo permanente apoyo y comprensión no hubiera podido llegar hasta aquí; gracias a ellos por su actitud alineada en todo momento hacia mi progreso formativo, pasando por alto los malos momentos (que haberlos los ha habido) y potenciando aquéllos otros en los mi recorrido avanzaba adecuadamente en pro de la adquisición de una buena educación académica y humana; gracias por haberme hecho sentir que, pasara lo que pasara, contaba siempre con su apoyo y su consejo.

A mi hermano, Pedro, Ingeniero Industrial, le tengo que agradecer su afecto, su respaldo y su amparo fraternal. Lástima que, durante los últimos años, no haya podido contar con su cercanía física, por estar realizando en Alemania su actividad profesional. Tan lejos, pero siempre tan cerca.

De mis abuelos, Agapito y Antonia, he recibido constantemente su incondicional aliento y cariño. Gracias, yayos.

Me gustaría igualmente manifestar mi agradecimiento a todos aquellos amigos y compañeros, cuya compañía y camaradería ha sido esencial a lo largo de mi vida, especialmente de mi vida académica. A todos los tendré siempre en mi recuerdo; en representación de todos, entre los más cercanos, cito a Jorge, Eduardo, Diego y Héctor. Muchas gracias amigos.

Por último también me gustaría agradecer, a la ciudad de Trento, donde conocí a mi novia Alicia y a la Università degli Studi di Trento, por la buena acogida y el año tan excelente que me brindaron.

Muchas gracias a todos.

Resumen

El proyecto consiste en la realización de una aplicación iOS. El objetivo esencial de este proyecto es, pues, hacer uso del potencial de dichos dispositivos móviles para diseñar y desarrollar una aplicación capaz de generar un sistema de “cartografía asistida” personalizada, manejable y desarrollable con relativa facilidad por una persona ciega. La aplicación permitirá al invidente construir y almacenar sus propios itinerarios para poder recrearlos en el momento preciso, consiguiendo así una amplia capacidad de desplazamiento, con el menor riesgo posible; se trata, pues, de un sistema guía a la carta, basado en los sentidos de la audición y el tacto.

Palabras Clave

iOS

Iphone

Ipad

Guía

GPS

3G

Aplicaciones Móviles

Abstract

The project is an IOS tracker application. The main goal is to develop an "Assisted Tracking System" using the full potential of mobile devices. The tracker has been thought to be personalized, manageable and relatively easy to be used by a blind person. The application allows to build and store itineraries and to use them later with the lowest possible risk. The project is, therefore an auto-guidance system based on the senses of hearing and touching.

Key Words

iOS

Iphone

Ipad

Guide

GPS

3G

Mobile Applications

Índice de contenido

1. INTRODUCCIÓN	1
1.1 Motivación	1
1.3 Organización de la Memoria.....	5
1.4 Sentido del Proyecto	7
2. ESTADO DEL ARTE	9
2.1 Dispositivos Móviles Inteligentes	10
2.2 Avances tecnológicos para personas con ceguera.....	15
2.3 Sistemas operativos para móviles	20
2.3.1 Android de Google.....	22
2.3.2 iOS de Apple	23
2.3.3 Windows Phone de Microsoft	24
2.3.4 Comentario final sobre los sistemas operativos.....	25
2.4 Localización geográfica	27
2.4.1 Concepto y técnica de la geolocalización	27
2.4.2 Métodos de Localización en iOS	30
2.5 Aplicaciones Móviles con Guiado GPS.....	37
3. DISEÑO	41
3.1 Introducción.....	41
3.2 Entrevista con una persona con discapacidad visual.....	42
3.3 Crear un Camino	43
3.3.1 Ayudas al sistema de guiado a la hora de crear un camino.....	43
3.3.2 Ayudas para creadores de caminos (personas sin discapacidad) ..	44
3.3.3 Ayudas a problemas de localización GPS	44
3.4 Guiado por una persona con discapacidad visual.....	45
3.4.1 Ayudas para el guiado al ciego.....	45
3.5 Recursos Tecnológicos mínimos necesarios para el Usuario	47
3.6 Software necesario para ser desarrollado el proyecto	49

3.6.1	Software necesario para Programación y Simulación.....	49
3.6.2	Software necesario para imágenes y tonos de audio	51
4.	DESAROLLO.....	55
4.1	Introducción.....	55
4.2	Estructura de una aplicación en iOS.....	56
4.3	Estructura de la aplicación.....	59
4.3.1	Pantalla inicial	60
4.3.2	Pantalla de lista de caminos	61
4.3.3	Pantalla de guiado.....	63
4.3.4	Pantalla de menú principal	67
5.	PRUEBAS, RESULTADOS Y CONCLUSIONES.....	73
5.1	Introducción.....	73
5.2	Prueba de Error de Localización (CoreLocation VS Google Maps) ..	74
5.3	Prueba de creación de un camino	76
5.4	Prueba de guiado (leer el camino).....	83
5.5	Prueba masiva de pulsación.....	89
6.	CONCLUSIÓN, APRENDIZAJE Y TRABAJOS FUTUROS	91
6.1	Conclusión.....	91
6.2	Aprendizaje	92
6.3	Trabajo futuros	93
7.	ANEXOS: INFORMACIÓN ADICIONAL.....	95
Anexo I	Referencias Bibliográficas	95
Anexo II	Tutoriales mostrados en el menú principal	98
Anexo III	Presupuesto	114
Anexo IV	Pliego de condiciones	121

Índice de ilustraciones

Ilustración 1. Teléfono para Invidentes, Pág 13.

Ilustración 2. Google Maps, Pág. 34.

Ilustración 3. iCoche, Pág.36.

Ilustración 4. Wilkiloc, Pág.37.

Ilustración 5. iPhone 4s, Pág.44.

Ilustración 6. Model View Controller, Pág.52.

Ilustración 7. Estructura del proyecto, Pág.54.

Ilustración 8. Pantalla inicial, Pág.55.

Ilustración 9. Pantalla de navegación, Pág.60.

Ilustración 10. Pantalla de menú principal, Pág.63.

Ilustración 11. Pantalla de escritura del nombre del camino, Pág.64.

Ilustración 12. Pantalla de creación del camino, Pág.64.

Ilustración 13. Pruebas de GPS, Pág.68.

Ilustración 14. Rendimiento en la pantalla inicial, Pág.70.

Ilustración 15. Rendimiento en el menú principal, Pág.71.

Ilustración 16. Rendimiento al pulsar el botón de crear camino, Pág.72.

Ilustración 17. Rendimiento al escribir el nombre de origen en crear camino, Pág.73.

Ilustración 18. Rendimiento en la zona de comenzar a crear camino, Pág.73.

Ilustración 19. Pantalla de error por falta de precisión, Pág.74.

Ilustración 20. Rendimiento en la pantalla de lista de caminos, Pág.77.

Ilustración 21. Rendimiento en la pantalla de guiado, Pág.78.

Ilustración 22. Dirección apuntada por CLHeading aceptada, Pág.79.

Ilustración 24. Rendimiento en la pantalla de guiado, Pág.80.

Ilustración 25. Pantalla de guiado, avisando de punto peligroso, Pág.80.

Ilustración 26. Rendimiento tras una prueba masiva de pulsación, Pág.83.

Índice de tablas

Tabla 1. Sistemas Operativos y Marcas que los utilizan, Pág. 18.

Tabla 2. Sistemas Operativos más usados, Pág. 19.

Tabla 3. Pruebas de GPS, Pág.69.

Tabla 4. Presupuesto Económico del proyecto, Pág.89.

1. INTRODUCCIÓN

1.1 Motivación

Constantemente recibimos información del ambiente que nos rodea a través de los receptores que se encuentran ubicados en los órganos de los sentidos; a saber, por los ojos (sentido de la visión), los oídos (sentido de la audición), la lengua (sentido del gusto), la nariz (sentido del olfato) y la piel (sentido del tacto).

En cada órgano hay receptores específicos que captan determinados estímulos. Las señales enviadas por los receptores son analizadas en el cerebro en un proceso muy complejo que, finalmente, da “sentido” a la información que percibimos.

El problema para las personas surge cuando alguno de los órganos de los sentidos sufre una agresión tan grave (congénita o adquirida) que inutiliza su funcionamiento total o en un grado muy elevado.

Centrándonos en el sentido de la visión (el más relevante de los sentidos del ser humano), la **ceguera** es la más grave de las agresiones que el órgano de la visión (los ojos) puede recibir, provocándole una discapacidad física que técnicamente no es sino la pérdida total o en grado muy elevado de dicho.

Pero, pese a los indudables cada vez más y mejores medios puestos a su disposición, siguen siendo evidentes las limitaciones a las que se ven sometidas las personas ciegas a la hora de moverse en su entorno vital.

Por ello, cualesquiera mejoras de los medios al uso o nuevas ayudas con las que pueda contar el colectivo de personas ciegas, en pro de facilitarle el movimiento individual de la manera más autónoma posible y ampliarle el territorio en que pueda desenvolverse con relativa normalidad, serán sin duda bien recibidas. El importante nivel de avance que ha alcanzado ya la telefonía móvil, y las telecomunicaciones en general, permite que ésta pueda ser

aplicada en este orden de cosas como un medio altamente útil y eficaz, con riesgo limitado.

Los teléfonos inteligentes, las tabletas, los sistemas operativos móviles y las plataformas de distribución de aplicaciones móviles para los mismos son dispositivos que poseen una gran cantidad de funcionalidades que, si se activan adecuadamente, pueden ayudar a las personas con ceguera a ampliar su capacidad de movimiento en un entorno cada vez más extenso.

Por tanto el proyecto se basará en usar dichos teléfonos móviles o tabletas inteligentes y conseguir con ellos una ayuda más para las personas con discapacidad visual.

Mediante el uso de otros órganos de los sentidos, concretamente, el tacto y el oído, conseguiremos ayudar al ciego y añadiremos una ayuda que con ayuda del bastón no tendrá problemas en realizar una ruta, incluso siendo desconocida para este.

Servirá para realizar una función similar a la del perro guía, pero a un precio claramente inferior, ya que el coste de un terminal Apple más la aplicación, ni por asomo se acercará a la cantidad de coste de entrenamiento que es de al menos treinta mil euros.

1.2 Objetivos

El principal objetivo es realizar una aplicación que será capaz de crear y posteriormente leerá un camino, con las características de grabado de fácil uso y un sistema de guiado destinado para ciegos con la inclusión de avisos por voz y manejo por tacto.

Por tanto se deben añadir una serie de funciones que permitan mantener al ciego en la senda o vector de trayectorias, sin caer entrar en zonas peligrosas.

Los pasos a dar para la ejecución del proyecto serán los siguientes:

- 1) Fijación de los recursos técnicos adecuados, es decir, los sistemas operativos y dispositivos móviles con y en los que poder desarrollar la aplicación. Por la propias características del proyecto, habrá que establecer una doble consideración:
 - a) Recursos para el diseño y desarrollo del prototipo: los existentes en el “Digital System Lab” del Departamento de Tecnología Electrónica y de Comunicaciones de la Escuela Politécnica Superior de la Universidad Autónoma de Madrid.
 - b) Recursos para los usuarios potenciales de la aplicación: podrán coincidir o no con los anteriores, por lo que habrá que determinar las alternativas que ofrece el mercado de teléfonos móviles y tabletas.
- 2) Planteamiento del prototipo de la aplicación, con un desarrollo que tendrá carácter universal en lo que se refiere al sistema, pero individual en lo relativo a su uso.
 - a) La idea principal es conseguir desarrollar una aplicación capaz de generar un sistema de “cartografía asistida” personalizada, manejable y desarrollable con relativa facilidad por una persona ciega.
 - b) La aplicación ha de permitir al invidente construir y almacenar sus propios itinerarios para poder recrearlos en el momento preciso. Se

trata, pues, de crear un sistema guía a la carta, basado en los sentidos de la audición y el tacto.

- c) La aplicación disponga la mayor cantidad de información de forma auditiva y táctil, en sustitución a los estímulos visuales. Además, los controles de manejo han de ser de fácil manejo para minimizar los márgenes de error.
- 3) Instrucciones de uso. Habrá que dotar a la aplicación con un adecuado capítulo explicativo de voz y sonido, con indicaciones sobre su uso para que los usuarios puedan aprender a usarla ante la imposibilidad de consultar un manual escrito.
- 4) Realización de las correspondientes pruebas para comprobar su viabilidad.

1.3 Organización de la Memoria

Este escrito se inicia con las referencias usuales del contexto en que realiza el Proyecto (Fin de Carrera), así como con el apartado de agradecimientos a cuantos han tenido que ver con el autor del mismo, a lo largo de su vida académica y personal, haciendo posible el momento actual.

El Informe en sí queda dividido en cinco capítulos, correspondiendo cada uno de ellos a una de las fases generales de desarrollo del proyecto, a saber:

- En el “**Capítulo 1.- INTRODUCCIÓN**”, se explica brevemente la razón de ser (del presente Proyecto (justificación), los pasos a dar para la ejecución del mismo objetivo) y la organización de este Informe (escrito de presentación formal).
- En el “**Capítulo 2.- ESTADO DEL ARTE**” se recoge el estado del arte y los medios a utilizar. Así pues haremos un balance de las tecnologías que se pueden usar y elegiremos la más adecuada. Además veremos qué trabajo se ha realizado previamente con respecto a los temas que abarcamos, es decir aplicaciones móviles y sistemas de guiado dirigidas a personas con discapacidad visual.
- En el “**Capítulo 3.- DISEÑO**”, se plantea a fondo el problema al que nos enfrentamos y trataremos de obtener soluciones para realizar el prototipo, dando al final una idea general esquemática de la aplicación y sus apartados. Nos familiarizaremos primero con el entorno de trabajo, contando brevemente las pautas a seguir para el desarrollo de una aplicación en la plataforma escogida.
- En el “**Capítulo 4.- DESARROLLO**”, se explicará con detalle los pasos que se han seguido a la hora de desarrollar el código de la aplicación.
- En el “**Capítulo 5.- PRUEBAS, RESULTADOS Y CONCLUSIONES**”, se analizan en las pruebas de campo realizadas. Se tratará de comprobar la funcionalidad de la aplicación previa a un uso real en personas con discapacidad. Se estudiarán asuntos como precisión, viabilidad, exactitud

en los resultados y otros apartados de importancia. Se hará una prueba con un usuario con discapacidad. Finalmente, se expondrán algunas conclusiones tras haber probado el prototipo y posteriormente se hablará sobre el trabajo futuro a realizar y la relación del proyecto con otros que también se está realizando sobre el tema, que complementan el uso de la aplicación.

- En el “**Capítulo 6.- CONCLUSIÓN, APRENDIZAJE Y TRABAJOS FUTUROS**” se comentan las conclusiones finales del diseño, desarrollo y pruebas de la aplicación, también se explica los conocimientos adquiridos en el tiempo de realización del proyecto y por último, los trabajos de mejora de este.
- En el “**Capítulo 7.- ANEXOS: INFORMACIÓN ADICIONAL**” se recoge determinada información económica y jurídica de importancia para los aspectos industriales del proyecto, los tutoriales mostrados en el menú principal, así como reseñas sobre la base documental que ha servido de ayuda para la elaboración del mismo.

1.4 Sentido del Proyecto

Según datos de la OMS, en el mundo hay aproximadamente 285 millones de personas con discapacidad visual, de las cuales 39 millones son ciegos y 246 millones presentan baja visión. Además, aproximadamente un 90% de la carga mundial de discapacidad visual se concentra en los países en desarrollo.

En España ya existían en 2008, según elmundo.es, casi 70.000 personas ciegas. El porcentaje de hombres y mujeres es prácticamente igual. Las comunidades autónomas con mayor número de ciegos son, en este orden, Andalucía, Cataluña y Madrid. El coste de un perro lazarillo supera los 30.000 euros (el monto íntegro lo asume la Organización Nacional de Ciegos Españoles). Hace 30 años se consideraba que su único trabajo posible era ser vendedor de cupones de la ONCE. En 2008, sólo la tercera parte se dedica a eso. Muchos son directivos, abogados, ingenieros... Más de 1.000 canes guía están actualmente operativos. El coste de un ordenador adaptado para el uso de los invidentes (con sintetizador de voz, "monitor" en braille...) supera los 5.000 euros. Con patente española hay bastones con GPS, robots de cocina que hablan, clasificadores de ropa por colores, etc.

Existe un alto costo económico y social de la ceguera que contribuye a empeorar unas economías de por sí débiles.

Como indica el Director General de la OMS: "Los costos que suponen la educación, la rehabilitación y la pérdida de productividad por causa de la ceguera tienen importantes repercusiones en las personas, las familias, las comunidades y las naciones, en particular en las zonas más pobres del mundo". La ceguera y la debilidad visual son consideradas padecimientos con repercusiones de "gran envergadura para el desarrollo humano, social y económico del individuo, así como en su calidad de vida".

A la hora de realizar las tareas cotidianas, es claro que personas ciegas o con visibilidad reducida no tienen las mismas posibilidades que las personas que sí ven a la hora de realizar tareas cotidianas. Según la ONCE "el 80% de la información necesaria para nuestra vida cotidiana implica el órgano de la visión. Esto supone que la mayoría de las habilidades que poseemos, de los

conocimientos que adquirimos y de las actividades que desarrollamos las aprendemos o ejecutamos basándonos en información visual”.

Asimismo, la ONCE resalta que “la visión representa, de esta forma, un papel central en la autonomía y desenvolvimiento de cualquier persona. Las diferentes patologías y alteraciones oculares pueden reducir en diversos grados o anular la entrada de esta información visual imprescindible para nuestro desempeño diario y bienestar. En este sentido, cuando hablamos en general de ceguera o deficiencia visual nos estamos refiriendo a condiciones caracterizadas por una limitación total o muy seria de la función visual. Es decir, nos estamos refiriendo a personas que, o bien no ven absolutamente nada, o bien, en el mejor de los casos, incluso llevando gafas o utilizando otras ayudas ópticas, ven mucho menos de lo normal y realizando un gran esfuerzo.

El presente proyecto va encaminado a ayudar a este colectivo, es decir a las personas que tienen prácticamente perdido el sentido de la visión; personas con **ceguera completa**, que no pueden ver nada, ni siquiera la luz. La mayoría de las personas, cuando emplean el término "ceguera", se refieren a ceguera completa.

Ceguera completa sí, pero no se ha de padecer al mismo tiempo alteración significativa alguna en los sentidos de la audición y del tacto, pues uno y otro son imprescindibles para la instrumentalización del presente proyecto. Voz, sonido y tacto son los elementos sensoriales necesarios para la utilización de los recursos técnicos del proyecto (teléfono o tableta iPhone, sistema operativo iOS y sistema de localización y dirección GPS incorporados).

Es sabido, por otra parte, que por compensación de la pérdida del sentido de la visión, los sentidos de la audición y tacto tienden a incrementar su funcionalidad.

2. ESTADO DEL ARTE

Hablaremos de las tecnologías en tanto que:

- Conjunto de los recursos tecnológicos en el mercado, que pueden estar a disposición del usuario para explotar adecuadamente de dicho “Sistema de Cartografía Asistida” elaborado; es decir, de los teléfonos móviles inteligentes y de las tabletas, así como de los sistemas operativos y de los sistemas de localización y dirección incorporables a los móviles personales.
- Conjunto de avances tecnológicos destinados para personas con discapacidad visual.
- Estudio de los diferentes sistemas operativos.
- Estudio de los diferentes métodos de capturar una coordenada en iOS.
- Evaluación de las aplicaciones con sistema de guiado GPS y 3G parecidas a la realizada en el proyecto.

2.1 Dispositivos Móviles Inteligentes

La telefonía móvil consiste en la combinación de una red de estaciones transmisoras o receptoras de radio (repetidores, estaciones base o BTS) y una serie de centrales telefónicas de conmutación de 1.er y 5º nivel (MSC y BSC respectivamente), que posibilita la comunicación entre terminales telefónicos portátiles (teléfonos móviles) o entre terminales portátiles y teléfonos de la red fija tradicional.

En su operación, el teléfono móvil establece comunicación con una estación base y, a medida que se traslada, los sistemas computacionales que administran la red van transmitiendo la llamada a la siguiente estación base de forma transparente para el usuario. Es por eso que se dice que las estaciones base forman una red de celdas, cual panal de abeja, sirviendo cada estación base a los equipos móviles que se encuentran en su celda.

La evolución del teléfono móvil ha permitido disminuir su tamaño y peso, desde el Motorola DynaTAC, el primer teléfono móvil en 1983 que pesaba 800 gramos, a los actuales más compactos y con mayores prestaciones de servicio. El desarrollo de baterías más pequeñas y de mayor duración, pantallas más nítidas y de colores, la incorporación de software más amigable, hacen del teléfono móvil un elemento muy apreciado en la vida moderna.

El avance de la tecnología ha hecho que estos aparatos incorporen funciones que no hace mucho parecían futuristas, como juegos, reproducción de música MP3 y otros formatos, reloj despertados, calculadora, correo electrónico, SMS, agenda electrónica PDA, fotografía digital y video digital, videollamada, navegación por Internet, GPS, y hasta Televisión digital. Las compañías de telefonía móvil ya están pensando nuevas aplicaciones para este pequeño aparato que nos acompaña a todas partes. Algunas de esas ideas son: medio de pago, localizador e identificador de personas.

A este tipo de evolución del teléfono móvil se le conoce como teléfono inteligente (smartphone), pudiendo realizar con él una gran cantidad de acciones, lo que explica que prácticamente todas las personas de los países

desarrollados disponga de uno de estos elementos. Las prestaciones que ofrecen los dispositivos móviles han ido aumentando de una manera incesante, así como la capacidad de información que se puede intercambiar entre ellos a través de la redes de telecomunicación.

Pese a lo que pueda parecer, se trata de un mercado relativamente nuevo que presenta un futuro prometedor para quienes destinen sus esfuerzos a la mejora continua de la telefonía móvil y de sus prestaciones. Y encima cuenta ya con un gran número de usuarios a los que satisfacer. Por ello el desarrollo y la distribución de aplicaciones móviles han experimentado un crecimiento excepcional en los últimos años.

En este proyecto se van a aprovechar las tan valiosas prestaciones que ofrecen los teléfonos inteligentes (smartphones), como soporte físico (hardware) de la aplicación, en tanto que se trata de una herramienta sumamente adecuada para ello, al contar de antemano con un gran número de potenciales usuarios, ser sobradamente conocida y a cuya producción y distribución se dedican empresas líderes del ámbito tecnológico.

Afortunadamente, en la actualidad, el mercado de los teléfonos inteligentes se caracteriza por el amplio abanico de marcas y la infinidad de modelos entre los que podemos elegir dependiendo de nuestras propias necesidades y preferencias.

De entre los teléfonos inteligentes, ¿cuál elegir para cargar y explotar el “Sistema de Cartografía Asistida” aquí presentado? La respuesta no es tan simple como pudiera parecer y tan ni siquiera hay una única respuesta.

Lo primero que hay que tener en cuenta es que el teléfono forma parte de un todo, integrado por el soporte físico (hardware) y el soporte lógico (software), lo que puede implicar que la elección de un soporte condicione la elección necesaria del otro, sea el primero elegido el físico o el lógico. Los elementos esenciales de soporte lógico son el sistema operativo y el navegador, de los que se glosará en los siguientes apartados.

Se conoce como software al equipamiento lógico o soporte lógico de un sistema informático, que comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas; en contraposición a los componentes físicos, que son llamados hardware, es decir, todas las partes tangibles de un sistema informático, cuyos componentes son eléctricos, electrónicos, electromecánicos y mecánicos. Componen el hardware los cables, gabinetes o cajas, periféricos de todo tipo y cualquier otro elemento físico involucrado; contrariamente, el soporte lógico es intangible

Por otra parte, dada la gran expansión alcanzada entre la población de los países desarrollados, es más que probable que el potencial usuario del “sistema” disponga ya de uno de estos teléfonos inteligentes. Entonces la pregunta sería: ¿sirve su teléfono para tal finalidad?

En la actualidad, elegir un teléfono móvil que cumpla con nuestros requisitos y necesidades más elementales puede convertirse en una tarea complicada, lo es más aún si consideramos el factor precio como elemento determinante de compra. El mercado ofrece tantas y tan variadas propuestas que la elección, a veces, no es nada fácil.

Se puede seguir teorizando mucho más al respecto sobre la influencia en la elección del teléfono de factores tales como los costes del mercado, el acceso al producto, el diseño del aparato, la diversidad funcional, etc.; pero, aquí y ahora, la cruda realidad se impone: el “Sistema de Cartografía Asistida” ha sido diseñado y desarrollado, como ha quedado dicho, con tecnología Apple, utilizando el sistema operativo iOS de Apple, en un iPhone de Apple.

¿Condiciona este hecho? Sí y mucho, aquí y ahora; nada en el futuro, pues en absoluto existen impedimentos para su posterior diseño y desarrollo con otros sistemas operativos y otros modelos telefónicos.

Los smartphones de hoy en día nos aportan innumerables ventajas con las que no contábamos hace no tanto tiempo; sin embargo todavía hay colectivos que no pueden disfrutar de ellas, al menos no como a ellos les gustaría. Uno de esos colectivos es el de los ciegos y demás gente que padece problemas serios de visión.

La llegada de los smartphones con pantallas táctiles no ha ayudado demasiado a las personas ciegas, ya que, aun contando con aplicaciones software para mejorar la accesibilidad de estos modelos, son en principio mucho menos adecuados que los terminales antiguos con teclados y botones físicos.

Así y todo, resulta evidente que las personas con ceguera se están viendo beneficiadas de la nueva tecnología, sencillamente porque, con todas las dificultades que encuentran, la están usando. No hay, pues, lugar para los más pesimistas, sino para creer en el futuro.

Precisamente para resolver esta carencia un laboratorio de investigación en India ha desarrollado el primer smartphone totalmente pensado para ciegos, ya que su pantalla con tecnología Shape Memory Alloy permite mostrar todo tipo de caracteres Braille y formas para ayudar a sacar mucho más provecho de estos dispositivos.



Ilustración 1. Teléfono para Invidentes.

Esta tecnología utiliza el concepto de que los metales recuerdan su forma original, y se expanden y contraen para llegar a ella tras su uso. La pantalla de este teléfono tiene una matriz de pequeñas clavijas que se mueven hacia arriba y hacia abajo según las necesidades del momento.

Así, la matriz permite crear caracteres Braille, pero también es posible mostrar formas que ayudarían a los invidentes a obtener información mucho más valiosa para ellos en todo tipo de situaciones.

Sumit Dagar, cuya empresa está desarrollando esta solución en el Centro de Innovación, Incubación y Emprendimiento en el campus de la Universidad de Delhi, ha indicado que “este producto está basado en una innovadora pantalla táctil que es capaz de elevar y deprimir el contenido que recibe para transformarlo en patrones táctiles”.

2.2 Avances tecnológicos para personas con ceguera

Las pantallas táctiles han iniciado un viaje de ida y vuelta para adaptarse a las necesidades de determinados colectivos, por ejemplo los ciegos. Así, la compañía californiana Tactus ha patentado una tecnología de microfluidos que transforma la superficie plana de una tableta o un smartphone en un teclado en relieve, lo cual permite que los invidentes los empleen sin tanto esfuerzo como hasta ahora.

Por debajo de la “pantalla dinámica” hay diversos canales por los que circula un fluido que, cuando procede, eleva la membrana deformable de la cobertura. Posteriormente ésta recobra su aspecto original. No es el primer experimento en esta línea, sin embargo, según los analistas, sí se trata de uno de los más prometedores.

También lo fue el que se puso en marcha desde la Escuela de Ingeniería de la Universidad de Stanford: una tableta táctil para que los ciegos puedan escribir y gestionar sus datos con ella gracias a un método de reconocimiento del usuario por las yemas de los dedos. Aunque sólo sirven para escribir en braille y son poco versátiles, los ordenadores portátiles especializados y ciertas herramientas para enviar correos electrónicos todavía resultan caros.

Con la finalidad de facilitar igualmente la navegación por internet de los ciegos, un equipo de la Universidad Estatal de Carolina del Norte desarrolló en 2010 un mecanismo que posibilitaba representar en braille lo aparecido en un ordenador, incluso las imágenes. Este dispositivo, que traducía píxeles en puntos con volumen, contaba con las ventajas de ser barato, resistente y altamente sensible a las órdenes de quien lo manejase.

La Universitat Autònoma de Barcelona y la ONCE crearon ya en 2006 un teclado de ordenador en braille muy útil para ciegos con problemas motores en las manos. Los inventores sustituyeron las teclas alfanuméricas convencionales por las ocho características del lenguaje para invidentes, pero, además, las combinaron con las de desplazamientos y funciones.

Un paso importante se dio cuando se perfeccionó una aplicación basada en Android para smartphones que podía utilizarse al margen de la visión del consumidor. Era ideal para los ciegos porque operaba a través de procesos táctiles y auditivos, de manera que, entre otras tareas, podían localizar un contacto en la agenda y llamarle a continuación sin problemas.

Mucho antes, concretamente en 2004, nuevamente la ONCE y Telefónica habían presentado el Owasys 22C, un móvil producido para invidentes. Y una década atrás los técnicos de Microsoft ya trabajaban en interfaces que eran independientes de la visión en distintos grados.

La Federación Nacional de Ciegos de Estados Unidos y la empresa Knfb Reading Technology elaboraron un software para ir más lejos que las tradicionales máquinas de lectura de textos. Con este programa los invidentes podían escuchar lo impreso sobre cualquier superficie mediante un objeto de poco más de 100 gramos: un Nokia N82. Todo con la finalidad de otorgarles a los afectados independencia y autonomía.

Unas gafas de realidad virtual ideadas recientemente por un grupo de expertos de la Universidad Carlos III de Madrid ayudan a las personas con una moderada discapacidad visual a desplazarse. El sistema interactúa con el sujeto que lo lleva puesto tras saber qué forma tienen los objetos cercanos y dónde están situados. La lista de gadgets es extensísima en este campo, aunque muchos no pasan de ser prototipos sin implantación en el mercado.

A través de un sistema de ultrasonidos, una computadora instalada en el calzado reconocerá los baches y una vez que se detecte el obstáculo, el zapato vibrará con una intensidad según la distancia al objeto. Las gafas utilizan el mismo sistema de orientación que los murciélagos para determinar la textura, la distancia y el tamaño de los obstáculos. Así lo explicó en Hong Kong el director del proyecto científico que pretende revolucionar el mundo de los invidentes con unos zapatos "que ven" y unos anteojos que utilizan ultrasonidos para orientar a su portador, según el diario South China Morning Post.

A través de un sistema de ultrasonidos, una computadora instalada en el calzado reconocerá los baches: "una vez que se detecte el obstáculo, el zapato vibrará, probablemente con más intensidad a medida que se acerque al objeto", explicó Leung Woon-fong, Director del Instituto del Productos y Tecnologías Innovadoras.

Los científicos contemplan la posibilidad de que distintos tipos de vibración identifiquen si el obstáculo es una elevación o una depresión, y se plantean acompañar el dispositivo con un cinturón que también emita vibraciones.

A estos artefactos se sumarían las gafas de ultrasonidos diseñadas por el profesor He Jufang, Director del proyecto, que el año pasado ganaron un segundo premio en el Salón Internacional de Invenciones Técnicas y Productos Nuevos de Ginebra.

Las gafas, denominadas "Electronic Bat Ears" (oídos de murciélago electrónicos) utilizan el mismo sistema de orientación que estos pequeños mamíferos para determinar la textura, la distancia y el tamaño de los obstáculos.

Los anteojos utilizan unos auriculares que recogen el tono, la frecuencia y el volumen de las ondas para alertar a su portador de lo que le rodea.

He Jufang también estudia conectar las gafas a través de un sistema de navegación GPS para indicar al invidente dónde se encuentra en cada momento.

"Activando un interruptor una voz les dirá cómo llegar exactamente de un sitio a otro", explicó el profesor, "el portador sabrá hacia qué dirección mira y dónde está. Queremos hacer el sistema tan pequeño que se pueda esconder en la montura, con las pilas y el sistema de sonar".

Los ciegos también podrán ver con sus oídos

World Acces for the Blind es la organización que está liderando una campaña para que los ciegos logren encontrar una manera más de "ver". Esta técnica es

conocida como ecolocación, y consta del mismo método que usan los murciélagos para poder ver en la noche.

Científicos norteamericanos descubrieron que en el cerebro de los ciegos, la corteza visual no deja de funcionar como antes se creía, sino que de hecho, esta se activa cuando los ciegos utilizan otro sentido como el del audio.

Por eso, ahora Ben Underwood, de 14 años de edad, nacido en Sacramento (California), es una de las pocas personas del mundo que siendo ciegas, utilizan la ecolocación, y con el sonido de un clic pueden saber, por medio del eco, la ubicación de los objetos y personas alrededor suyo.

El pequeño Underwood afirma, que además de escuchar la radio y el ventilador, él puede también escuchar las paredes, el piso y las escaleras de su casa. Según los científicos de California, la ecolocación crea imágenes en el cerebro de los ciegos, justo de la misma manera como lo hacen los murciélagos y los delfines.

Los nuevos inventos tecnológicos podrían convertirse así en una fuente de integración para los ciegos, que elevarían su calidad de vida y podrían trabajar y desarrollar actividades que hoy en día no pueden realizar y, además, convertirse en una suculenta fuente de ingresos para los que desarrollen este tipo de tecnologías.

Una empresa tecnológica de Pereira (Colombia) ha desarrollado un dispositivo que permite que niños con discapacidad visual reconozcan formas y colores en una pantalla a través del tacto. Gracias a este sistema, llamado Proyecto Iris, los alumnos identificarán formas geométricas y conceptos visuales complejos y aprenderán en las mismas condiciones que sus compañeros.

La aplicación contiene un sistema de códigos que transforma los colores en vibraciones. "Por ejemplo, el negro va a una velocidad mayor, el blanco es la ausencia de vibración y el rojo es intermedio, y así con cada color", declara María Fernanda Zúñiga Zavala, una de las integrantes del equipo. Las imágenes se representan en una cuadrícula en la que cada cajetín vibra de forma independiente, permitiendo así que un estudiante pueda distinguir formas

táctiles siguiendo con el dedo un trayecto que posea la misma vibración. De esta manera, un estudiante en situación de discapacidad visual puede identificar formas y colores al desplazar el dedo por la cuadrícula y según la velocidad de las vibraciones.

Las últimas innovaciones en tecnología pueden haber cambiado la vida de los usuarios normales, pero hay muy pocos aparatos para ayudar o entretener a los usuarios con discapacidad visual; por ello, toda aplicación encaminada a mejorar las condiciones de vida de este colectivo será siempre bienvenida.

2.3 Sistemas operativos para móviles

Definido el tema del teléfono inteligente (smartphone), se pasa al análisis de los sistemas operativos. En este tema no hay tanta variedad para elegir como en el caso de la telefonía, lo que quiere decir que muchos modelos de teléfono comparten sistema operativo.

Un sistema operativo para móviles es un sistema operativo que controla un dispositivo móvil al igual que los PCs utilizan Windows o Linux entre otros. Sin embargo, los sistemas operativos para móviles son mucho más simples y están más orientados a la conectividad inalámbrica, los formatos multimedia para móviles y las diferentes maneras de introducir información en ellos

A continuación, se recogen algunos datos orientativos para la mejor comprensión de la situación del mercado de sistemas operativos para móviles.

Cuadro sobre los sistemas operativos que usan marcas de teléfonos móviles (México)	
S.O.	Marcas de Teléfonos Móviles
Android	Alcatel, Azumi, BLU, Huawei, HTC, Lanix, LG, Motorola, Movistar, Samsung, Sony, ZTE
iOS	Apple
Windows Phone	Asus, BenQ, Huawei, HTC, LG, Motorola, Movistar, Nokia, Palm, Samsung, Toshiba
Blackberry OS	Blackberry
Linus	Samsung
Symbian OS	Nokia, Panasonic
Firefox OS	Motorola
Otros	Palm, Samsung, Siemens, Treo

Tabla 1. Sistemas Operativos y Marcas que los utilizan.

Top Smartphone Operating Systems, Shipments, and Market Share, 2013 Q3 (Units in Millions)

Operating System	2Q13 Unit Shipments	2Q13 Market Share	2Q12 Unit Shipments	2Q12 Market Share	Year-over-Year Change
Android	187.4	79.3%	108	69.1%	73.5%
iOS	31.2	13.2%	26	16.6%	20.0%
Windows Phone	8.7	3.7%	4.9	3.1%	77.6%
BlackBerry OS	6.8	2.9%	7.7	4.9%	-11.7%
Linux	1.8	0.8%	2.8	1.8%	-35.7%
Symbian	0.5	0.2%	6.5	4.2%	-92.3%
Others	N/A	0.0%	0.3	0.2%	-100.0%
Total	236.4	100.0%	156.2	100.0%	51.3%

Source: IDC Worldwide Mobile Phone Tracker, August 7, 2013

Tabla 2. Sistemas Operativos más usados.

A la vista de los dos cuadros anteriores, se sacan tres contundentes conclusiones:

- 1) Los cuatro sistemas operativos más usados (Android, iOS, Windows Phone y Blackberry OS) copan el 90% del mercado.
- 2) De los cuatro sistemas operativos más usados,
 - Dos son exclusivos de marca: iOS de Apple y Blackberry OS de Blackberry.
 - Los otros dos están abiertos a una pluralidad de marcas, algunas de las cuales usan indistintamente uno u otro (Android o Windows Phone: Huawei, HTC, LG, Motorola, Movistar y Samsung).
- 3) Competir razonablemente en este mercado con nuevas incorporaciones de sistemas operativos es hartamente difícil, lo que no es óbice para que algunas otras firmas lo sigan intentando. La industria de la telefonía móvil sigue apoyando nuevos proyectos de sistemas operativos basados en el software libre. Este año y el próximo, se espera que salgan al mercado estos nuevos sistemas para móviles y tabletas. Tanto para desarrolladores como para usuarios, una de sus ventajas es que permitirán ejecutar aplicaciones realizadas en el estándar web HTML5, lo que hará que una misma aplicación sea compatible con varios sistemas operativos. En cuanto a estrategia, en un principio, el objetivo es dirigirse a usuarios que busquen

un terminal de gama media-baja y también a dispositivos móviles para países y mercados en desarrollo como Rusia, la India o Brasil.

El hecho de que este proyecto haya sido diseñado y desarrollado, como ya ha quedado dicho, utilizando hardware y software de Apple, lleva a decantarse por iOS de Apple como sistema operativo y por un Smartphone igualmente de Apple, si bien más pronto que tarde los sistemas operativos quedarán liberados y abiertos a todas las posibilidades del mercado, pues la propia evolución tecnológica y la normativa internacional al respecto lo facilitarán.

Así las cosas, en la situación actual, parece lógico centrarse en el análisis y exposición pormenorizada de los cuatro S.O. más usados en el mercado de la telefonía móvil, a saber:

- Android
- iOS
- Windows Phone
- BlackBerry OS

2.3.1 Android de Google

El sistema operativo Android es sin duda el líder del mercado móvil en sistemas operativos, está basado en Linux. El desarrollador de este S.O. es Google.

La gran ventaja de este sistema operativo es su carácter abierto. Android se distribuye bajo dos tipos de licencias, una que abarca todo el código del Kernel y que es GNU GPLv2, que implica que su código se debe poner al alcance de todos y que todos podremos hacer con este código lo que nos parezca oportuno, modificarlo, ampliarlo, recortarlo, pero siempre estaremos en la obligación de volver a licenciarlo con la misma licencia.

Google también tiene otra licencia para el resto de componentes del sistema bajo el nombre APACHE v2, que implica que el código se pueda distribuir para ser modificado y usado a antojo del que lo utilice, pero a diferencia del primer

caso, las modificaciones y el código resultante no es obligatorio el licenciarlo bajo las mismas condiciones en las que se encontraba.

Como desarrollador, las principales ventajas que encontramos son el enorme número de usuarios y la facilidad a la hora de empezar a desarrollar y posteriormente probar y distribuir la aplicación. Además a día de hoy algunos dispositivos Android son igual de potentes o más que los iPhone, que era una de las cosas que siempre tenía a favor la marca de la manzana.

Sin embargo, las principales virtudes también suponen ciertos problemas. El hecho de que haya evolucionado tan rápido hace que existan muchos dispositivos distintos, con cualidades hardware muy distintas. Esto supone que las diferencias en rendimiento o en resolución de pantalla entre dispositivos sean muy grandes. Resulta una tarea complicada para el desarrollador poder garantizar el rendimiento óptimo y/o una visualización correcta de las aplicaciones en varios dispositivos y asegurar la compatibilidad con modelos más antiguos.

Con respecto a la accesibilidad nos encontramos con los mismos problemas que acabamos de mencionar, ya que la amplia mayoría de modelos ofrecen opciones distintas. En las últimas versiones Android ofrece un lector de pantalla llamado TalkBack que permite interactuar con el teléfono a personas que no vean la pantalla. Además ofrece las opciones más habituales como zoom o cambiar el tamaño de la letra. Como punto fuerte se puede destacar un sistema reconocedor de voz de Google, que puede resultar muy útil para sustituir la entrada de texto por teclado virtual.

2.3.2 iOS de Apple

iOS, anteriormente denominado iPhone OS, es el sistema operativo que da vida a dispositivos como el iPhone, el iPad, el iPod Touch o el Apple TV. Su simplicidad y optimización son sus pilares para que millones de usuarios se decanten por iOS en lugar de escoger otras plataformas que necesitan más hardware para mover con fluidez el sistema.

Creado por Apple originalmente para el iPhone, es un derivado de Mac OS X.

A finales de 2013 Apple lanzó para todos los usuarios de dispositivos iOS una nueva versión de su sistema operativo llegando así a la iOS 7.1 como actualización importante, en la que hay novedades centradas en la interface de usuario y mejoras generales en animaciones y transiciones que harán que funcionen de una manera mucho más rápida gastando así menos recursos y batería.

Apple iOS ha sido sin duda la referencia durante mucho tiempo, hasta la aparición de Android. Apple siempre ha ofrecido la mejor experiencia al usuario de la industria cuidando los detalles al máximo y haciendo las cosas de manera sencilla.

En cuanto a la accesibilidad, iOS dispone de más alternativas que el resto. Ofrece un sistema de lectura de pantalla muy completo llamado VoiceOver (usado en el proyecto), que una vez se aprende a usar resulta de gran utilidad para usar el teléfono sin ver la. Dispone también de las funcionalidades enumeradas en los otros sistemas como la posibilidad de ampliar la pantalla mediante un zoom, se pueden ampliar el tamaño de las letras, cambiar los colores a los opuestos para mejorar el contraste o configurar el acceso directo a alguna opción mediante el botón de inicio. Además las últimas versiones de iOS incorporan un reconocedor de voz con el que se puede interactuar, y es muy útil para un acceso rápido a muchas tareas, además de dar la posibilidad de introducir texto por voz en lugar de por teclado pantalla y por tanto es muy importante en una aplicación para ciegos, todo esto tendrá en peso importante en la aplicación.

2.3.3 Windows Phone de Microsoft

Windows Phone (abreviado WP) es un sistema operativo móvil desarrollado por Microsoft como sucesor de Windows Mobile. Compite directamente contra Android de Google e iOS de Apple. Se basa en el núcleo del sistema operativo Windows CE y cuenta con un conjunto de aplicaciones básicas.

Ha sido el último en llegar al mercado de los dispositivos móviles. La primera versión llamada Windows Phone 7 salió al mercado en 2010. La versión actual es la reciente Windows 8.

Se trata de un sistema cerrado, robusto, basado en la sencillez y ofrece una interfaz fluida y simple. En general se puede decir que los teléfonos con Windows Phone tienen un rendimiento muy bueno.

Como principales puntos a favor desde el punto de vista de desarrollo podemos señalar que se ha creado un estándar en la resolución del sistema que supondrá un beneficio a la hora de realizar sólo una la interfaz gráfica. Otra ventaja es que Microsoft provee de un entorno de trabajo a la altura de los mejores, con una gran cantidad de aplicaciones que hacen que la programación resulte lo más sencilla posible.

La mayor desventaja es que su tardía llegada al mercado ha hecho que le sea muy difícil competir con otros sistemas como Android o iOS, que ya disponían de una amplia cuota de mercado, y al no ofrecer grandes saltos de calidad con respecto a estas alternativas es difícil convencer al usuario para que cambie de plataforma. Por ello el número de aplicaciones existentes es mucho menor que en el de los sistemas citados anteriormente.

Con respecto a la accesibilidad, ofrece opciones como cambiar el tamaño del texto, utilizar temas con alto contraste, usar lupa para ampliar partes de pantalla o comandos de voz para realizar un determinado número de tareas.

2.3.4 Comentario final sobre los sistemas operativos

De entre los dos O.S mayoritariamente implantados en el mercado nos decantamos por iOS por dos motivos:

- Por ser el utilizado en el proyecto.
- Por la gran aceptación que tiene entre los usuarios a los que va destinado el proyecto (personas ciegas o con baja visión)

Por su relevancia en relación con este proyecto, resaltamos algunos párrafos de la “Información útil para el manejo de dispositivos Apple por personas ciegas o con baja visión” publicada por Centro de Investigación, Desarrollo y Aplicación Tiflotécnica (CIDAT) de la ONCE:

- “Dada la gran aceptación que tienen entre nuestros usuarios los dispositivos táctiles de Apple en general, y el iPhone y el iPad en particular, al incorporar soluciones de accesibilidad, ofrecemos información que puede facilitar a las personas ciegas o con baja visión el manejo de estos equipos”.
- “Por un lado, hemos elaborado unos vídeos demostrativos en los que se muestra el manejo de un iPhone utilizando el revisor de pantalla VoiceOver, así como el manejo de éste con la línea Braille Focus:
- “También hemos recopilado información de fácil consulta relativa a los gestos y combinaciones de teclas de VoiceOver en iOS, sistema operativo para dispositivos de Apple como el iPhone, iPad, etc.”
- “Por otro lado, y orientado especialmente a los desarrolladores en iOS, hemos elaborado un documento relativo a la programación accesible en este entorno. Puede descargarse desde el siguiente enlace: [Programación accesible en iOS.](#)”
- “Aunque en cuanto a la accesibilidad la filosofía de Apple es la misma en todos sus dispositivos mediante el software VoiceOver, la utilización de sus ordenadores requiere una mención aparte por las obvias diferencias de manejo con los dispositivos mencionados anteriormente.”

2.4 Localización geográfica

2.4.1 Concepto y técnica de la geolocalización

Un proyecto como éste (“Cartografía Asistida para Personas Ciegas, en Teléfonos o Tabletas Apple”) está basado, por definición en el concepto “geolocalización”.

La “geolocalización” es un término que hace referencia a conocer nuestra ubicación geográfica automáticamente. Hay varias maneras de que esto suceda, y como es natural, los dispositivos móviles son los que más fácilmente permiten la actualización de nuestra posición, por su portabilidad.

Las posibilidades de la geolocalización son muy diversas. Según el Instituto Cartográfico Nacional, el 80% de la actividad humana requiere conocer su posición sobre un mapa. Unos 500 millones de personas en todo el mundo acceden hoy como usuarios habituales a servicios de geolocalización. La previsión es de crecimiento, puesto que faltan por incorporarse a esa cifra muchos países en desarrollo.

Se puede localizar cualquier objeto sobre un mapa. A veces, ni siquiera es necesaria la localización por satélite GPS; basta con detectar a qué antenas están conectados los teléfonos. Los usos son muy variados en todos los campos de la actividad humana; lo que hace sólo unos años estaba sólo al alcance de técnicos, ahora se ha popularizado.

Con la localización por mapas se puede obtener y manejar muchísima información territorial. La colaboración entre los sectores públicos y privados ha sido esencial en el desarrollo de soluciones tecnológicas de este sector.

Viene bien para el proyecto conocer algo sobre las tecnologías que se usan en este campo de las telecomunicaciones, y más concretamente sobre aquellas de las que dispone el teléfono inteligente; como es bueno tener alguna referencia sobre el funcionamiento de estas tecnologías, así como sobre las funcionalidades que el dispositivo puede obtener de ellas a partir del hardware instalado.

Particularmente importantes para el proyecto han sido los servicios de localización y los mapas proporcionados en el SDK del iPhone. Una de las características especiales del iPhone es que “sabe dónde estás”: utiliza mapas y GPS de una manera sencilla como la que ofrece CoreLocation y MapKit.

El iPhone SDK es un Kit de desarrollo de software para la plataforma iOS de Apple, con el objetivo de permitir a terceros desarrollar aplicaciones nativas para el iOS. El lenguaje de programación principal para iPhone OS, al igual que en Mac OS, es Objective-C.

El GPS integrado en el iPhone ofrece al desarrollador unas funcionalidades que combinadas con otro tipo de información (fotos, sonidos, etc.) permiten crear una enorme cantidad de aplicaciones y servicios para el usuario; también ofrece a los desarrolladores la posibilidad de incluir en sus propias aplicaciones los mapas de Google para dar al usuario final una representación de su posición, por lo que un programador puede sacar el máximo provecho.

El iPhone utiliza actualmente tres métodos para calcular la posición geográfica:

A. Hace uso de la red GPS. De hecho, hay un conjunto de satélites que se mueven a la misma velocidad que la tierra (geoestacionarios). De esta manera, la posición de cada uno de estos satélites se mantiene siempre fija respecto a la superficie terrestre. A través de una antena receptora integrada, con el iPhone se puede obtener la posición de estos satélites.

En teoría, es suficiente que la antena detecte por lo menos a tres de los satélites para que pueda calcular exactamente su posición. Pero en el mundo real, el número de satélites necesarios es mayor, porque la señal de cada satélite que llega a la antena no tiene la misma potencia debido a diferentes causas (atmosféricas, electromagnéticas, interferencias de radio frecuencias, etc.) y al entorno físico en el cual se encuentra el móvil durante su intento de estimar su posición.

B. En lugares cerrados (un túnel o una habitación), la señal GPS es demasiado baja y se convierte en algo inútil. Por eso Apple decidió añadir dos maneras más para que las probabilidades de servicio fueran más altas.

De hecho, el iPhone original ya utilizaba estos canales, sin tener receptor GPS.

Uno de ellos utiliza la red móvil. Dado que las posiciones de las antenas de la red de telefonía móviles (GSM, GPRS, UMTS, etc.) son conocidas, Apple ha pensado utilizar esta información que se encuentra almacenada en tres bases de datos: SkyHook WiFi, Google Maps y US DoD.

A través del canal telefónico, el iPhone recibe la posición geográfica de las antenas más cercanas (en teoría como en el caso del GPS es suficiente conocer la posición de tres antenas) para calcular su posición.

C. Si la red telefónica también fallara, el tercer método sería utilizar las antenas Wi-Fi que se puedan encontrar alrededor. Este caso es muy dependiente del hecho de que la antena Wi-Fi esté registrada en las bases de datos mencionadas anteriormente. Si quieres participar activamente en este servicio con tu AirPort mejorando el sistema de localización, es suficiente que vayas al primer enlace (SkyHook) y proporciones las coordenadas de tu Wi-Fi de manera completamente anónima. De esta manera, tu antena Wi-Fi entrará a formar parte de un conjunto mundial de antenas y sus datos podrán ser utilizados para que un iPod Touch o un iPhone pueda calcular más exactamente su posición.

Los tres métodos no son excluyentes los unos con los otros. De hecho, el iPhone en realidad combina de manera inteligente los tres para ofrecerte tu posición con una probabilidad de servicio lo más alta posible.

El SDK del iPhone ofrece al programador diferentes opciones, para que las coordenadas geográficas calculadas con los métodos presentados anteriormente puedan ser utilizadas para diferentes fines. Apple ha incluido dos framework: CoreLocation y el nuevo MapKit. Todo esto explicado al detalle en el apartado 2.6.3

CoreLocation es el servicio que se encuentra disponible en el SDK para proporcionar la ubicación geográfica del usuario. Es la capa del SDK que se comunica con el GPS.

Las tecnologías usadas en el proyecto, en resumen, son:

- GPS
- 3G
- Wi-Fi

2.4.2 Métodos de Localización en iOS

La localización GPS se basa en el framework de CoreLocation sirve para determina la localización actual y el heading asociado al dispositivo. El framework usa el hardware disponible para determinar la posición del usuario y del heading. Se usan clases y protocolos en el framework para configurar y dirigir los eventos de localización y heading.

Este framework también permite definir regiones y monitorizarlas cuando el usuario lo desee.

La ruta del framework es la siguiente:

```
/System/Library/Frameworks/CoreLocation.framework
```

La ruta de los ficheros de cabecera:

```
/System/Library/Frameworks/CoreLocation.framework/Headers
```

TIPOS DE CLASES:

CLBeacon

Su inicio se data en el 2007 cuando Nokia creo el Bluetooth 4.0, que tiene la capacidad de estar continuamente enviando información y poder leerla desde otro dispositivo.

Su alcance es de 50-70m, pero hasta el iOS 7 no existía una api para trabajar con beacons incluyendo en el Framework CoreLocation.

En el bluetooth LE hay dos grupos, los peripheral que informan de servicios y exponen sus propiedades y el grupo central, que buscan servicios y leen-escriben sus propiedades.

iBeacon es un peripheral. Para poder interactuar con iBeacons en iOS7 se han añadido algunos métodos al protocolo CLLocationManagerDelegate:

- `locationManager:didDetermineState:forRegion:` el CLLocationManager llama a este método cuando se pasa la frontera de una región.
CLRegionState: (CLRegionStateUnknown, CLRegionStateInside, CLRegionStateOutside).
- `locationManager:didRangeBeacons:inRegion:` el CLLocationManager llama a este método cuando detecta beacons o dejar de detectarlos. Los iBeacons detectados se reciben en un array ordenados por proximidad. El más cercano el primero.
- `locationManager:rangingBeaconsDidFailForRegion:withError:` si una CLBeaconRegion contiene datos no válidos o falla al registrarse.
- `requestStateForRegion:` nos permite preguntar por estado del dispositivo en relación con una región.

iBeacons queda representado en CLBeaconRegion. Que provee de las siguientes propiedades:

- `proximityUUID:` un ID global, por ejemplo para identificar una compañía.
- `major` y `minor:` permite crear áreas o secciones de beacons
- `proximity:` distancia relativa a un iBeacon (Immediate, Near, Far, Unknown)
- `accuracy:` distancia en metros. No posición exacta, determinada en función de rssi.
- `rssi:` fuerza de la señal recibida medida en dBm.

CLBeaconRegion hereda de CLRegion, define una región basada en la proximidad de un dispositivo a un CLBeacon. Nos ofrece 3 inicializadores:

- `initWithProximityUUID:identifier:`
- `initWithProximityUUID:major:identifier:`
- `initWithProximityUUID:major:minor:identifier:`

En función de ellos podemos crear regiones que contengan uno o varios beacons, si no identificamos el menor, puede haber más de un beacon con el mismo `proximityUUID` y `major`.

El “`identifier`” es una cadena de texto que nos permitirá identificar la región en la que está el dispositivo si lo necesitamos, por ejemplo cuando hay varias regiones.

Otro método importante incluido en iOS7:

- `peripheralDataWithMeasuredPower:` devuelve un diccionario con la información que identifica al beacon codificada además de la información a transmitir. Este diccionario se debe pasar a una instancia de `CBPeripheralManager` al método `startAdvertising`. Este es uno de los pasos a realizar para convertir nuestro dispositivo en un `iBeacon`. Lo veremos después en un ejemplo.

El parámetro `measuredPower` es opcional, representa la medida de la fuerza de la señal a un metro de distancia del beacon. Propiedades:

- `notifyEntryStayOnDisplay:` si su valor es `YES`, el sistema avisará a nuestra app mediante el método del protocolo de `CLLocationManager`.
- `locationManager:didDetermineState:forRegion:` cuando el usuario enciende la pantalla del dispositivo y está dentro de la región, incluso si la app no está corriendo, en este caso la app se lanza en `background`. Ojo que la app no esté corriendo se refiere a que esté inactiva porque la ha parado el SO, si la apara el usuario, la app no recibirá notificaciones. Su valor por defecto es `NO`.

Además con en IOS7, podemos indicar si queremos que se nos informe al salir o entrar de una región, es nuevo en CLRegion, no es específico de CLBeaconRegion:

- `notifyOnEntry`: llama a `locationManager:didEnterRegion:` cuando se entre en una región .Default Yes
- `notifyOnExit`: llama a `locationManager:didExitRegion:` cuando se salga en una región .Default Yes

Para monitorizar CLBeacons, primero creamos la región, en este caso nos interesan todos los beacons con un determinado UUID, además indicamos que queremos que se nos informe si estamos dentro de la región cuando el usuario active la pantalla del teléfono.

Una vez estamos dentro de la región nuestra aplicación será informada de ello a través del método del protocolo, entonces es cuando empezaremos a tratar de determinar e identificar los beacons cercanos. Si salimos de la región lo que hacemos es parar el “ranging” de los beacons.

Una vez determinados los beacons cercanos, los recibiremos en un array en el método del protocolo “.....didRangeBeacons:inRegion:” además también recibiremos la región. Una característica importante es que en el array en el que recibimos los beacons estos vienen ordenados por proximidad de mayor a menor, es decir el primer elemento del array corresponderá al beacon más cercano al dispositivo.

CLGeocoder:

La geolocalización inversa, es decir, la obtención de los datos referentes al país, ciudad, calle... a partir de las coordenadas de latitud y longitud se llevaba a cabo mediante las clases `MKReverseGeocoder` y `MKReverseGeocoderDelegate` de framework MapKit. A partir de iOS 5 se utiliza la clase `CLGeocoder` de Core Location.

Se puede seguir usando `MKReverseGeocoder` pero Apple recomienda el uso de `CLGeocoder`.

- Para obtener la posición actual del usuario, para lo cual haremos uso de la clase `CLLocationManager` y del protocolo `CLLocationManagerDelegate`.
- Una vez obtenida la posición del usuario, haremos uso de las coordenadas obtenidas para la geolocalización inversa a través de la clase `CLGeocoder`.

Esta tarea es realmente sencilla. Lo primero será declarar un objeto del tipo `CLLocationManager` en nuestro archivo de cabecera e inicializarlo en el método oportuno —por ejemplo, en `viewDidLoad`.

- `locationManager:didUpdateToLocation:fromLocation:` Es el método donde se ejecutara lo que necesitemos cuando tengamos las coordenadas.
- `distanceFilter`: Con este parámetro indicamos el número de metros que deben realizarse para actualizar la coordenada. Es un typedef del tipo primitivo `double`.
- `desiredAccuracy`. En este punto es importante destacar que cuanto mayor sea la exactitud que exijamos, el dispositivo sufrirá un mayor gasto de batería y el tiempo de respuesta empeorará. Por tanto es interesante ajustar esta propiedad a nuestras necesidades reales.
- `startUpdatingLocation`: para que nuestro `CLLocationManager` comience a trabajar y obtener la localización actual del usuario.

Cuando obtengamos la dirección actual, esta vendrá empaquetada en un objeto del tipo `CLLocation`. La propiedad que más nos va a interesar es `coordinate`. Esta propiedad es un struct de tipo `CLLocationCoordinate2D`.

- `Latitude`
- `Longitude`

Geolocalización inversa:

Como decía anteriormente, para obtener el nombre del país y demás datos a partir de unas coordenadas determinadas, haremos uso de la clase

CLGeocoder. Debes tener en cuenta que esta clase necesita de una conexión a Internet para obtener los datos que le vamos a exigir. Su implementación es muy sencilla. En nuestro caso, la vamos a llevar a cabo en el método `locationManager:didUpdateToLocation:fromLocation:`

El método `reverseGeocodeLocation:completionHandler:` es el que hace uso de un bloque para gestionar la respuesta. El bloque viene definido dentro de la constante `CLGeocodeCompletionHandler`.

El bloque recibirá como parámetros un `NSArray` llamado `placemark` y un objeto del tipo `NSError` con información sobre posibles errores ocurridos. El array contendrá objetos del tipo `CLPlacemark` y lo normal es que tan solo recibamos un objeto dentro de dicho array. El objeto del tipo `CLPlaceMark` encapsula las propiedades que nos interesan para obtener el país, ciudad, calle... para las coordenadas actuales.

CLHeading:

Los objetos de la clase `CLHeading` contienen los datos de orientación que ha generado un objeto de la clase `CLLocationManager`.

Esta clase ha sido usada en el proyecto para fijar el vector de dirección entre dos puntos.

Propiedades:

- `CLLocationDirection magneticHeading;`
- `CLLocationDirection trueHeading;`
- `CLLocationDirection headingAccuracy;`
- `CLHeadingComponentValue x;`
- `CLHeadingComponentValue y;`
- `CLHeadingComponentValue z;`
- `NSDate * timestamp;`

CLLocation:

CLLocationManager que será la encargada de facilitarnos los métodos de acceso al posicionamiento.

Al indicar que nuestra clase será delegada para el CLLocationManagerDelegate, debemos implementar dos métodos:

- `didUpdateToLocation`: que se ejecutará cada vez que se modifique la posición de nuestro dispositivo indicando la posición anterior y la nueva en cada llamada.
- `didFailWithError`: que se ejecutará en caso de que se produzca algún tipo de error.

Básicamente estamos implementando 4 métodos: `init` y `dealloc` que nos permite inicializar y liberar todo aquello que utiliza nuestra clase, y los dos métodos que hemos visto anteriormente y que son requeridos al actuar como delegado.

El objeto `newLocation` contiene toda la información relativa a nuestra posición actual, aunque quizás la información más utilizada es la disponible dentro de la propiedad `newLocation.coordinates.latitude` y `newLocation.coordinates.altitude` que son la latitud y altitud detectada por el dispositivo.

Por último cuando queramos utilizar en nuestro `ViewController` lo implementado hasta este punto, tan solo deberemos crear un objeto de tipo `PosicionamientoController` en nuestro `ViewController.h`, y a continuación inicializar el mismo mediante:

```
posicionamientoController = [[MyCLController alloc] init];
```

```
[posicionamientoController.locationManager startUpdatingLocation];
```

La primera línea simplemente inicializa el objeto, y la segunda y más importante es la que hace de lanzador para que cada vez que se actualice la posición se invoque el método `didUpdateToLocation` que hemos definido previamente.

2.5 Aplicaciones Móviles con Guiado GPS

Se enumerarán las principales aplicaciones de guiado existentes en el sistema iOS.

Google Maps



Ilustración 2. Google Maps.

Probablemente sea la aplicación de guiado más conocida del mundo, desbancando aplicaciones como TomTom o Systel.

Se basa en el uso de la señal GPS y 3G de un terminal móvil, también puede usar la Wi-Fi para localizar.

Entre sus puntos fuertes cabe destacar, que es el dueño del paquete de mapas usado en iOS y Android.

Concretamente hablamos del framework de MapKit en iOS. Donde le mandaríamos la coordenada de CLLocation (CoreLocation Framework) explicada en el apartado anterior y nos la mostraría en el mapa.

Puntos a favor:

- Tiene un motor muy fuerte, con infinidad de coordenadas en todo el mundo, donde por ejemplo, es capaz de predecir el camino más rápido haciendo un análisis de flujo de tráfico en las carreteras a tiempo real.
- Gran renombre y está extendido por todo el mundo.

- Gratuita.

Aspectos diferentes a nuestra app:

- No es una aplicación destinada para ciegos.
- No es posible trazar caminos por todas las zonas peatonales, tiende a hacer el recorrido de mayor longitud.
- No avisa de puntos peligrosos.

iCoche

Se trata de una aplicación que guía hasta el punto donde se ha dejado el coche.

Se basa en los paquetes de Google Maps para posicionar la coordenada, obtenida por CoreLocation, por tanto también usa MapKit.



Ilustración 3. iCoche.

Puntos a favor:

- Permite enviar direcciones a otros usuarios.
- Tiene todas las ventajas de Google Maps, ya que usa los mismos métodos.

Aspectos diferentes a nuestra app:

- No es una aplicación destinada para ciegos.
- No es posible trazar caminos por todas las zonas peatonales, tiende a hacer el recorrido de mayor longitud.
- No avisa de puntos peligrosos.

Wikiloc

Aplicación que permite guardar rutas y posteriormente permite leerlas.

Inicialmente ha sido creada para el mundo del senderismo y de la bicicleta.

Basado como las anteriores en dos frameworks Mapkit, donde se usa mapas de google para enseñar el camino a seguir y también CoreLocation, que usa para capturar las coordenadas.



Ilustración 4. Wikiloc.

Puntos a favor:

- Permite grabar caminos personalizados.
- Tiene todas las ventajas de Google Maps, ya que usa los mismos métodos.

Aspectos diferentes a nuestra app:

- No es una aplicación destinada para ciegos.
- No avisa de puntos peligrosos.

3. DISEÑO

3.1 Introducción

En este apartado del proyecto, procederemos a introducir un prototipo de la aplicación, así como el motivo del diseño de la misma.

Se partirá del estudio previo de los principales problemas que encuentran en su movilidad las personas con discapacidad visual, tanto en el ámbito del uso sencillo de la interfaz gráfica del terminal, como en las distintas ayudas introducidas para evitar los riesgos más habituales detectados y a los que tienen que enfrentarse cada día.

En primer lugar, hablaremos de los principales problemas que se les plantean a las personas invidentes en los trayectos a pie, tratando con el diseño de minimizar en su mayor parte este tipo de riesgos.

En segundo lugar, trataremos de explicar lo que debe ser capaz de realizar la aplicación, de los principales problemas que pueda tener el terminal para localizar de la manera más óptima posible el itinerario, donde se ofrecerá una solución a los problemas.

En tercer lugar, hablaremos del sistema de guiado y de la dificultad que entraña el uso del terminal móvil para una persona ciega, así como las soluciones ofrecidas, para que el uso y el guiado sea lo más confortable posible para un persona que carece de visión.

3.2 Entrevista con una persona con discapacidad visual

Después de mantener conversaciones con personas con problemas de visión, hay que manifestar que los principales problemas detectados en el tránsito por una vía pública son:

- Obstáculos temporales o inesperados, como por ejemplo:
 - la existencia coyuntural de una zanja para cambiar una tubería o introducir fibra óptica.
 - un vehículo mal aparcado en la acera.
 - un bache en el camino.
 - un cartel de publicidad.
 - suelo resbaladizo (agua, nieve, aceite, etc.).
 - un ciclista imprudente por la acera.
- Semáforos: problema importante, sobre todo cuando éstos no dispongan de aviso por voz.
- Obstáculos peligrosos: desde un tramo de escaleras, un terraplén a los lados o una acera estrecha, hasta un paso de cebra sin semáforo o una calle sin pasos señalados.

Por otro lado nos indica la gran dificultad que dispone para ellos el uso de un terminal telefónico.

3.3 Crear un Camino

La aplicación debe poder grabar un camino de la manera más precisa posible, teniendo en cuenta que este camino debe ser creado por una persona que no tenga ceguera.

Como todos sabemos no todo el mundo tiene la misma capacidad de manipular un Smartphone. De ahí nuestro principal problema, la zona de creación y manuales debe ser lo más sencilla posible, pero dentro de la sencillez, que podamos incluir un gran número de avisos para que posteriormente la persona guiada, que será una persona ciega o con gran pérdida de visión, pueda tener un mejor guiado a lo que actualmente se ofrece en el mercado.

Pero existe otro gran problema y es la toma de una coordenada usando GPS, donde sabemos que depende el lugar donde se tome y la manera pueden llegar a ser de tres metros en adelante, los valores más normales suelen ser de cinco a doce metros.

3.3.1 Ayudas al sistema de guiado a la hora de crear un camino

Con el fin de reducir, en la mayor medida posible, este tipo de peligros, hemos creado en el menú de creación del camino, una serie de botones en los cuales se puedan guardar, por ejemplo, vicisitudes tales como tramo de escaleras ascendente, tramo de escaleras descendente, peligro a izquierdas y derechas; y, sobretodo, el aviso de paso de peatones sin semáforos, parte muy peligrosa.

Cada botón guardará una coordenada y el peligro que será avisado posteriormente en la guía (según se explicará en el posterior Apartado de Desarrollo con más detenimiento).

Hay que aclarar que, en ningún momento el terminal, sustituirá al bastón, dado que tiene por error cuatro o cinco metros, pero sí podrá avisar al invidente con la mayor brevedad posible de los peligros, con el fin de mantenerle en alerta.

3.3.2 Ayudas para creadores de caminos (personas sin discapacidad)

Con el fin de simplificar la parte de creación y así ser más fácil de producir caminos por cualquier persona sin influir sus conocimientos informáticos, se ha procedido a la creación de botones táctiles con imagen incorporada para facilitar tanto escrita como visualmente el uso de la aplicación.

Y dado que esta parte debe ser realizada por personas sin discapacidad, procedemos a realizar una serie de tutoriales para realizar el camino con la mayor precisión. Estos podrán verse en los anexos finales de este escrito.

3.3.3 Ayudas a problemas de localización GPS

Otro problema grande como hemos comentado anteriormente, es el GPS para localizar, dado que el error de localización suele ser grande; por ello se diseñan funciones con el fin de minimizar el baile de posiciones y que la coordenada sea lo más próxima posible. Se bloqueará la coordenada que no disponga de la precisión necesaria y será mostrada mediante una alerta.

Se generará una vibración siempre que el terminal tenga una coordenada, dentro de unos márgenes que hemos decidido.

También elegimos los métodos más precisos de captura que nos facilita iOS, lo cual nos induce en un consumo elevado de la batería.

3.4 Guiado por una persona con discapacidad visual

También hemos usado un sistema de alertas con vibración por el cual se avisara al guiado de posibles pérdidas de señal o de los peligros citados anteriormente.

Otro problema grave se producía en la propia guía, ya que la minusvalía que tienen estas personas es total o parcialmente visual.

Los terminales móviles en los cuales se basa la aplicación son un gran problema para personas con deficiencia visual.

3.4.1 Ayudas para el guiado al ciego

Hemos procedido a usar varias funciones que tiene el sistema iOS como VoiceOver con el fin de facilitar el manejo de dicho terminal bajo la manipulación de una persona sin visión.

Pero la parte más importante en cuanto a la interfaz gráfica ha sido una simplificación de la misma, dividiendo la pantalla inicial en dos botones que redirigen a dos submenús uno para elegir el camino (menú para invidentes) y otro para proceder a la creación e información (para personas sin discapacidad).

Se opta por una primera interfaz que contendrá un botón aproximado del 85% de la pantalla que será para entrar en modo de guiado y otro para videntes que será del 15% restante.

En el guiado, se dividirá la pantalla en dos partes, una con información, que cada vez que desee el ciego podrá pulsar y así será avisado de los metros hasta el siguiente punto y de si es peligroso ese punto (a diez metros del peligro se avisa). Y el segundo es un botón para resituarse en la senda o acciones como empezar o finalizar.

También, se procederá a implementar y generar un conjunto sonidos con el fin de ayudar al ciego en la guía.

3.5 Recursos Tecnológicos mínimos necesarios para el Usuario

La elección de los recursos tecnológicos a ser utilizados por el usuario estará condicionada por las disponibilidades del mercado y los costes que pueda asumir, pero le serán necesarios:

- 1 Un **iPhone 4S** (o una tableta ipad mini) de Apple.
- 2 Un **monoauricular** para poder escuchar las órdenes del terminal móvil.
- 3 **S.O. iOS 7.1 o superior.**

Todo ello, para poder cargar y explotar el “**Sistema de Cartografía Asistida**”.

Pero antes, teniendo siempre presente que este proyecto va dirigido a ayudar a potenciales usuarios con ceguera completa, se completará el actual con una exposición pormenorizada de cada uno de los cuatro instrumentos básicos (de hardware y software) a disponer por el usuario, sobre las alternativas que ofrece el mercado actual, el potencial de viabilidad de cada uno y su pertinente análisis de idoneidad, en aras del mejor aprovechamiento del Proyecto.



Ilustración 5. iPhone 4s.

Características Técnicas iPhone 4S:

- Chipset ARM Apple A5
- Procesador 1 GHz
- 64 Bit
- Memoria 64 GB
- Sistema Operativo iOS 7.1
- AGPS
- UMTS/HSDPA/HSUPA (850, 900, 1900, 2100 MHz); GSM/EDGE (850, 900, 1800,1900MHz); CDMA EV-DO Rev. A (800, 1900 MHz),Wi-Fi 802.11b/g/n (sólo 802.11n de 2.4 GHz)
- Max Download Speed 14.4 Mbps
- Max Upload Speed 5.8 Mbps
- Multitouch
- Audio: respuesta en frecuencia de 20-20K Hz
- Autonomía conversación 480 minutos
- Autonomía en standby 200 horas
- Amperaje 1420 Mah

El precio oscila desde 225€ a 400€, teniendo en cuenta que este terminal es el mínimo necesario pero también sería más recomendable el último modelo de Apple, el iPhone 5S.

3.6 Software necesario para ser desarrollado el proyecto

3.6.1 Software necesario para Programación y Simulación.

Xcode

Es el software principal que se va a utilizar para la realización de la fase de programación del proyecto.

Xcode es el entorno de desarrollo integrado de Apple; es gratuito y actualmente para iOS7 se encuentra en la versión 5.1, es capaz de compilar código en Objective-C, que es el lenguaje en el que se ha realizado este proyecto.

La última versión, citada anteriormente, tiene una serie de mejoras relevantes respecto de las otras versiones anteriores; como ejemplo, ya no se carga tanto peso en la interfaz y sí en el contenido, lo que se ve reflejado en la disminución de los tamaños de los elementos en la interfaz, como la de la cabecera.

Otro elemento característico de esta versión es la introducción de un medidor de consumo de la memoria RAM (Random Access Memory) y uso de los núcleos de los microprocesadores del terminal; lo cual es tremendamente útil para medir el rendimiento del terminal según se van implementando más funciones.

Desde nuestro punto de vista, se trata de una aplicación bastante útil y agradable para el programador con un gran sistema de debug y un buen sistema de manejo de interfaces.

OBJECTIVE-C

El Objective-C es un lenguaje de programación a objetos creado como un superconjunto de C. Actualmente es el lenguaje de programación utilizado por Mac OS y iOS.

Este lenguaje es bastante dinámico aunque a primera vista no lo parezca; basa su filosofía en el concepto de envío de mensajes entre objetos. A diferencia de C++, este no invoca un método a un objeto, sino que envía mensajes a un objeto. Permite hacer estrategias al programador más liberal.

Tiene las siguientes características:

- Las clases son objetos pudiendo crear métodos y clases en tiempo de ejecución.
- Type introspection y Reflection
- Uso de bloques de código.

Objective-C ofrece al programador un sistema sencillo para declarar las propiedades y atributos que definen a un objeto.

El único punto negativo es el sistema de administración de memoria, que es algo más complejo que el de otros lenguajes de programación, como puede ser Java, así como, el proceso de iniciación y el de construcción de objetos.

Como la plataforma elegida para la realización del proyecto ha sido iOS, para poder lanzar la aplicación en tabletas y teléfonos móviles Apple, optamos por elegir Objective-C sobre el compilador de Xcode 5.1 para la versión de iOS 7.1.

CocoaTouch

Es una API (Application Programming Interface), es decir, un conjunto de funciones y métodos que ofrece la biblioteca para ser usado por otro software como una capa de abstracción.

Cocoa Touch proporciona una capa de abstracción al sistema operativo iOS.

Cocoa Touch se basa en el set de herramientas que proporciona el API de Cocoa para crear programas sobre la plataforma de Mac OS.

Nos ha servido de gran ayuda este sistema, haciendo uso de funcionalidad y evitando programar todas las funciones desde el principio; como consecuencia

de esto se produce un gran ahorro temporal en la programación, así como, una mayor facilidad de ésta.

Para poder usar esta API es necesario disponer de un montante elevado inicial. Debido a que los equipos de la compañía Apple Inc. adolecen de un precio elevado, serían necesarios los siguientes productos:

- Un equipo Mac, ya sea MacMinis (disponibles en nuestro laboratorio), iMAC o MacBook.
- Tener descargado e instalado Xcode
- Tener habilitada la cuenta de desarrollo, dado que es necesario abonar una determinada cantidad anual por el servicio.

En este proyecto se procederá al uso de Cocoa Touch, debido a que es el utilizado por iOS para el desarrollo de sus aplicaciones.

3.6.2 Software necesario para imágenes y tonos de audio

CorelDRAW X6

Corel DRAW GRAPHICS SUITE X6 es una herramienta de diseño de comunicaciones visuales, ofrece una solución completa y fiable. Tanto si eres diseñador principiante como experimentado, su flujo de trabajo es intuitivo y completo. Por ello, hemos elegido esta herramienta como principal sustento de imágenes en la aplicación.

Esta aplicación se ha desarrollado con compatibilidad nativa con procesadores de 64bits, por tanto se requiere un ordenador con sistema operativo de 64bits.

Las imágenes que hemos creado con este programa son las siguientes:

- Botones del menú principal.
- Botones submenús.
- Botones de los distintos riesgos que nos podemos encontrar.

Cabe recordar que en Xcode es posible poner cada imagen creada como imagen botón, es decir, que el nuestra imagen creada será el botón pero

siempre respetando las medidas iniciales de ese botón. Este último aspecto es muy importante porque no es necesario crear la imagen de determinado tamaño, menos el icono de la aplicación que debe ser de ciento veinte por ciento veinte píxeles.

Easy Synth

Easy Synth es un instrumento de teclado con la característica de ser sintetizador. El teclado es la parte frontal provista de teclas en la cual hay un botón de seleccionador de la octava a usar.

Tiene un desplegable para seleccionar el instrumento a utilizar donde se encuentran pianos, guitarras, trombones etcétera, o incluso efectos de sonido como windsnake, steamPepes, mice etcétera.

La aplicación se compone de ocho pestañas:

- Sound: pestaña en la que se selecciona el instrumento o efecto, la octava y se encuentra el teclado.
- Harmonics: pestaña que cuenta con 32 fadders para variar cada harmónico.
- Filter: pestaña para seleccionar el tipo de filtro (si es paso banda, paso alto, fade u otros definidos por nosotros).
- Amp: pestaña para variar un fadder de attack, decay, sustain, reléase o overdrive.
- Mod: pestaña para seleccionar Pitch Filter o Amp y el tipo de shape.
- Echo: pestaña para seleccionar la cantidad, el delay, el feedback y flange Amoun-rate
- Arpeggio: 17 fadders para controlarlo.

- Record: para poder comenzar la grabación y guardar dicha grabación (parte de pago), pero si te permite reproducir los que has grabado, por tanto usamos esta opción para luego ajustarla con el Virtual DJ.

Con el fin de no aumentar los costes del proyecto se procede a lanzar la aplicación sobre una tableta Android Bq Edison de diez pulgadas y su salida de audio enviarla por un cable minijack de tres coma cinco a minijack de tres coma cinco ambos machos, a la entrada de micrófono de la CPU, concretamente a una tarjeta de sonido Asus Xonar DS.

Mediante el software de grabación Virtual Dj, que comentaremos a continuación, se ha producido la grabación y el posterior recorte y selección de formato.

Esto no es más que un sistema barato de realizar los sonidos, el caso óptimo sería siempre dentro de una sala acondicionada para el sonido, mediante el uso de un piano midi sintetizador y con un software de audio profesional. Todo esto supondría un montante económico muy grande, ya que solo la sala hablamos de una gran cantidad de gasto y el software de audio también es excesivamente caro.

Con estos sistemas conseguiríamos que el audio no rompiese, es decir no saturase y conseguiríamos una calidad de audio mucho mayor.

Aun así, hemos usado un cable específico de audio, con conectores minijack de uso profesional, con lo cual hemos reducido en gran medida los problemas iniciales que nos encontramos.

VirtualDJ

El VirtualDj es un programa de mezclas de música en los formatos de audio (Mp3, Wav, OGG, AAC...), creado por la marca de origen francés Atomix Productions.

La principal característica de esta aplicación es que muestra dos platinas o tocadiscos, asociados a una mesa de mezclas; e incluye controles de PITCH,

ecualizadores de altos, medios y graves; te muestra una lista de todos los clips disponibles y tiene un potente gestor listas de audio y efectos de estos mismos.

El programa se puede conseguir con una versión de prueba que te mantiene todas las opciones completas por un tiempo limitado de veinte días.

Entre las diversas opciones con que contábamos, lo hemos elegido como nuestro sistema de grabación de audio por:

- Su sencillez de interfaz y su fácil manejo.
- Permitir, de manera sencilla, ajustar las calidades del formato wav o mp3 que son los que usaremos; y además, permitir recortar en milisegundos las pistas con el fin ajustar el tamaño y la duración a nuestra manera.

4. DESAROLLO

4.1 Introducción

En este apartado centro del proyecto, desarrollaremos en profundidad cada view controller de la aplicación. Concretamente seguiremos el siguiente orden:

En primer lugar, hablaremos sobre cómo es una estructura de una aplicación en iOS, donde cabe destacar el famoso model-view-controller.

En segundo lugar, hablaremos de la parte de creación de los métodos utilizados, desde un texto hasta una coordenada, mostrando claramente las distintas ayudas, tales como la vibración o las coordenadas con peligro.

En tercer lugar, se comentará la parte de la guía donde hay que destacar varias situaciones, ya que se involucran varios views:

- TableviewController: encargada de mostrar los caminos, indicando el principio y el fin; también se indicarán las distintas ayudas en ese view.
- UINavigationController: que es el encargado en sí de la guía.

4.2 Estructura de una aplicación en iOS

Sigue el modelo MVC: Model View Controller

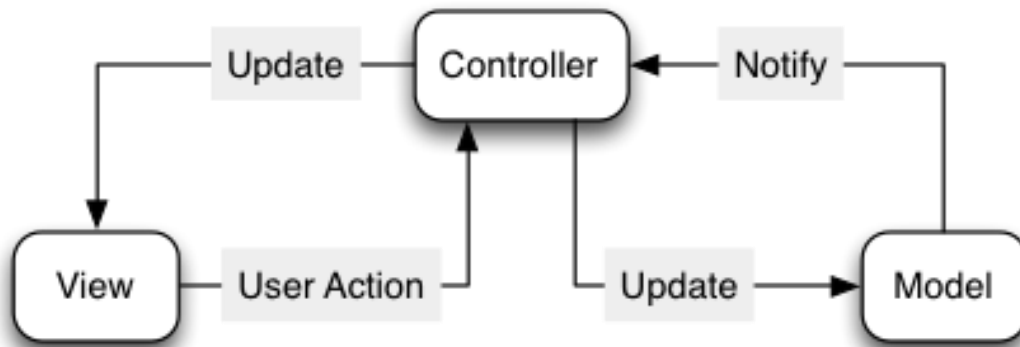


Ilustración 6. Model View Controller.

- **View:** cualquier elemento visual mediante el cual un usuario de la aplicación pueda ver e interactuar. Ejemplos de esto pueden ser los botones táctiles, un label de texto no editable o un text field con texto editable, que son clase de UIView.
- **Model:** incluye objetos que permiten tratar con los datos, tanto guardarlos como manipularlos. Esto no tiene nada que ver con la interfaz visual. Es visualmente cerrado para el usuario de la aplicación.
- **Controller:** es el master de la aplicación, manda y recibe órdenes de del View y del Model. Es el encargado de, por ejemplo, hacer diferentes cambios en la interfaz o el encargado de realizar acciones cuando se pulsa un botón táctil.

El Model View Controller tiene una serie de restricciones que hay que tener presentes:

- El Controller puede comunicarse con View y Model a la vez.

- Model y View no pueden comunicarse entre sí.
- Los Controllers pueden comunicarse con otros Controllers.

En la aplicación encontramos archivos de este tipo:

- **UIViewController:** sirve para presentar y gestionar los Views, gestiona la relación entre los Views y los Models.
- Existen dos tipos de ficheros .h y .m, donde podremos crear referencias, métodos o incluso señalar a qué otro UIViewController ir en caso de realizar la acción deseada.
- **Storyboard:** es una de las clases más características de esta programación, ya que en ésta es donde controlaremos el tipo View. En ella, por ejemplo, se pueden crear UIViewControllers y luego definirlos, como por ejemplo, una tabla y asociarla a un fichero nuestro, con lo que obtendrá sus métodos y características.
- **AppDelegate:** de extensiones .h y .m, se generan cada vez que se crea un proyecto y su función es la de controlar entre otras cosas el comienzo o el fin de la aplicación. Un ejemplo de su uso puede ser el de poner una imagen inicial y esconderla a los 3 segundos.
- **Data:** de extensión .xcdatamodel, según el modelo de la imagen anterior corresponderían a la parte de Model en el modelo ModelViewController.
- Se trata de una base de datos interna en la cual se pueden guardar como ocurre en este proyecto las distintas coordenadas. Es necesario añadir un paquete de Framework llamado CoreData.
- **UITableView:** son objetos basados en celdas, usadas en el proyecto para ofrecer los distintos caminos, que ofrecen multitud de opciones como cambiar fondos, letras, añadir imágenes, etcétera. También pueden tener la característica de Controller (UIViewController); en esta última opción también serían las encargadas de controlar la aplicación.

Protocolos: los de extensión .m y .h, son protocolos creados con el fin de manejar un determinado método en un cierto momento. Pueden manejar distintas APIs predefinidas por iOS, como puede ser lanzar un clip de música o leer un fichero.

4.3 Estructura de la aplicación

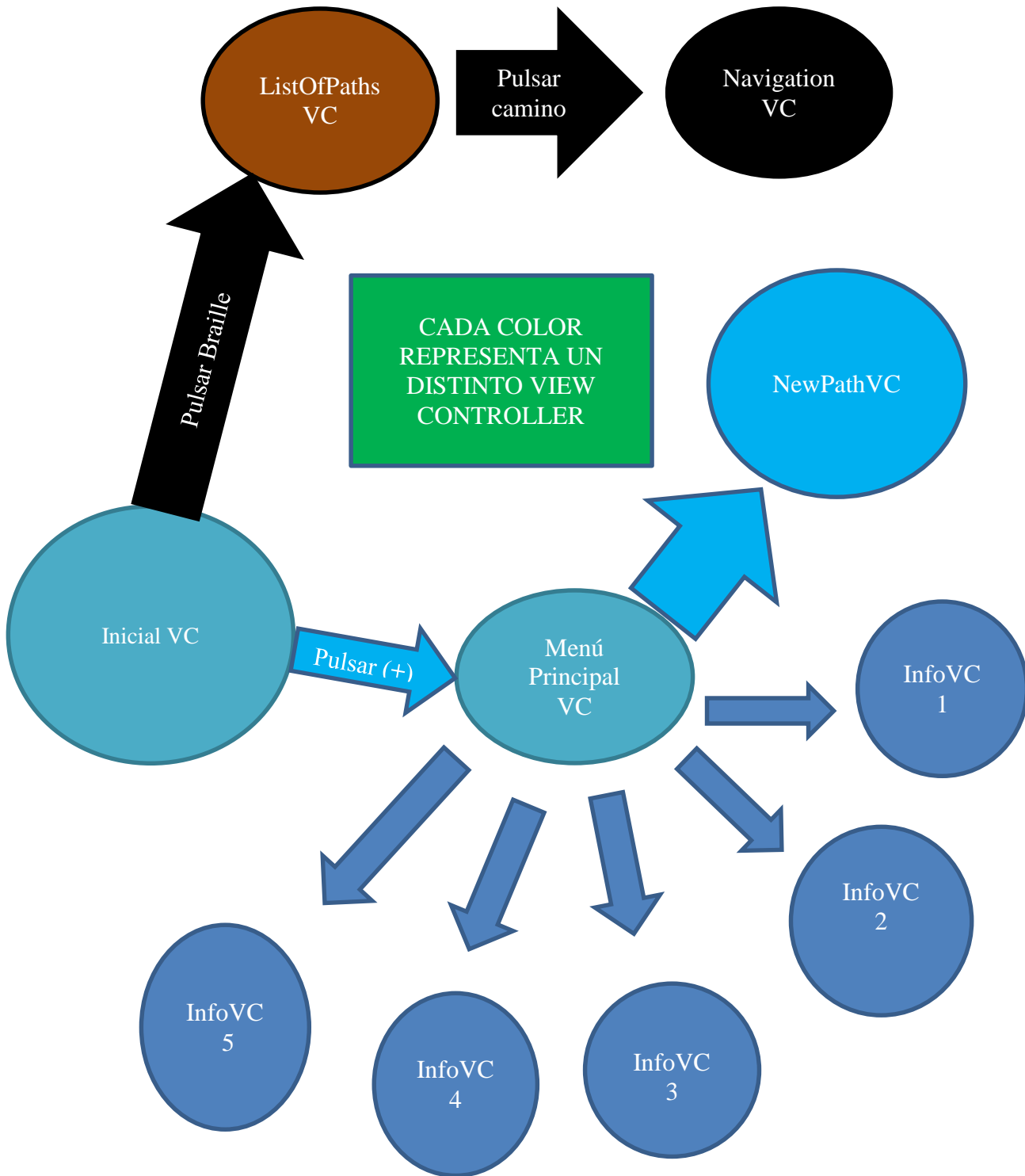


Ilustración 7. Estructura del proyecto.

4.3.1 Pantalla inicial

La pantalla principal está prácticamente ocupada en su totalidad para pulsar sobre ella, de modo y manera que la persona invidente pueda, con suma facilidad, iniciar el proceso de guiado (selección del trayecto y realización del mismo). Al fin y al cabo, éste es el objetivo de la aplicación.

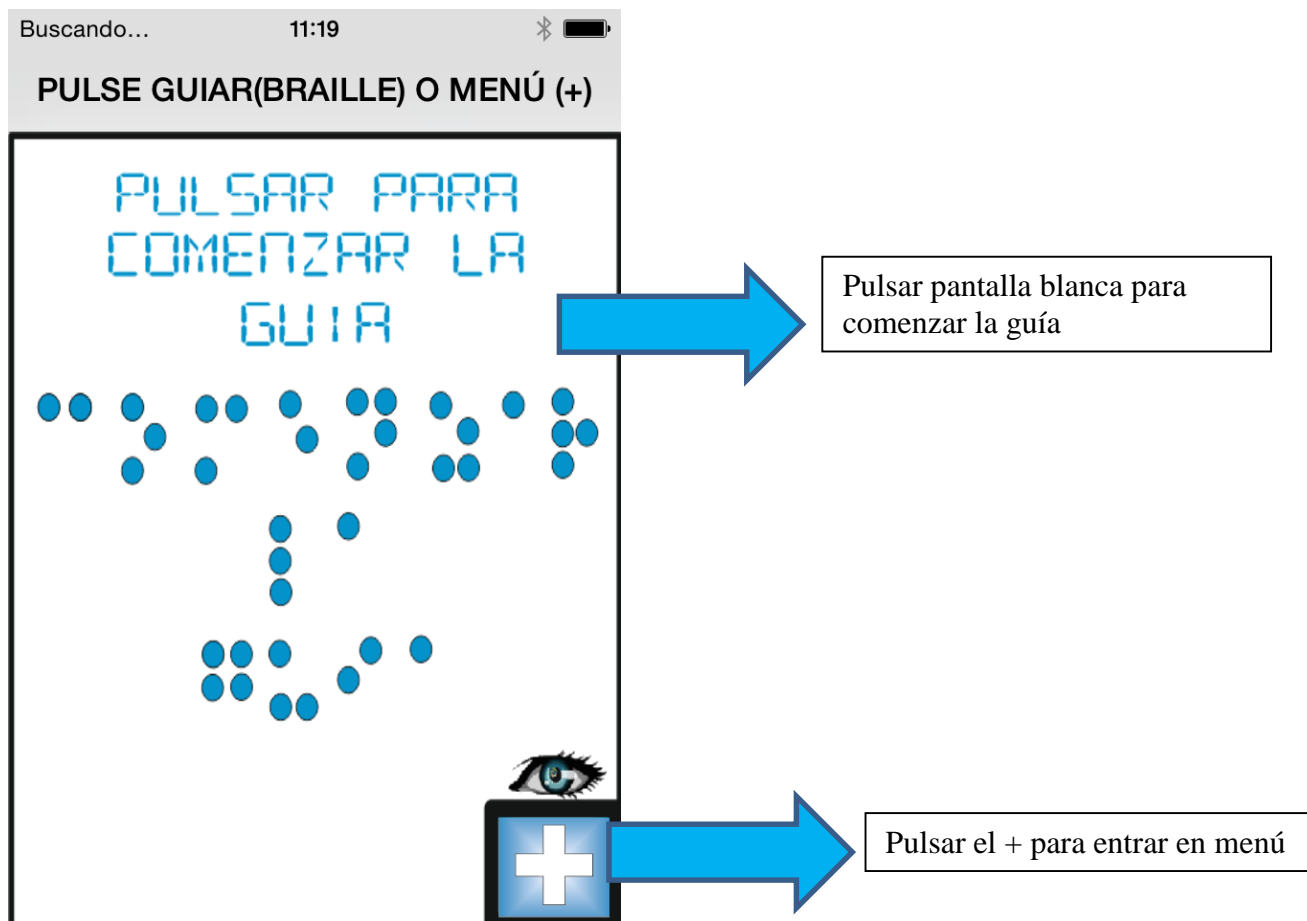


Ilustración 8. Pantalla inicial.

Este view controller carece de nombre específico, puesto que ha sido creado como modelo estándar y posee, por defecto, las mismas características que un view controller .

Se compone de dos botones que sirven para separar las áreas más características de esta aplicación: la guía y la creación.

Como se puede observar en la figura anterior, el botón de guía es mucho mayor que el botón de menú de creación. Esto es debido a que, de esta manera, se facilita la aplicación para la manipulación por un ciego.

En cuanto a las ayudas al ciego, destinatario de la aplicación, cabe resaltar el aviso por voz de VoiceOver en esta pantalla que, al iniciarse, reproducirá un “Pulse Guiar o Menú, ir a botón, Menú Creación Botón”. De esta manera hacemos saber al ciego que existe un botón con tal nombre y cuando pulse el botón de guía, en caso de ser un único toque, se reproducirá “ir a botón”; y en caso de pulsación doble entrará en el siguiente view controller.

Al pulsar en el + reproducirá “Menú Creación Botón”, por lo que sabrá que no es el botón que necesita.

4.3.2 Pantalla de lista de caminos

Como ya ha quedado indicado con anterioridad, una vez se inicia la aplicación, aparece la pantalla principal que contiene una parte destinada exclusivamente al ciego, la guía.

Al pulsar sobre el botón de guiado, la aplicación pasará a ser controlada por otro view controller, llamado ListOfPathsViewController.

ListOfPathsViewController

Interfaz gráfica:

Está compuesta por una tabla; en cada fila llamada celda se pueden divisar varios rasgos que no se encuentran en una tabla estándar.

Contiene dos líneas de texto donde se indica el origen del camino y el destino.

La cabecera debe ser personalizada para que el ciego, al entrar y describir la pantalla con el VoiceOver sepa en qué view controller está.

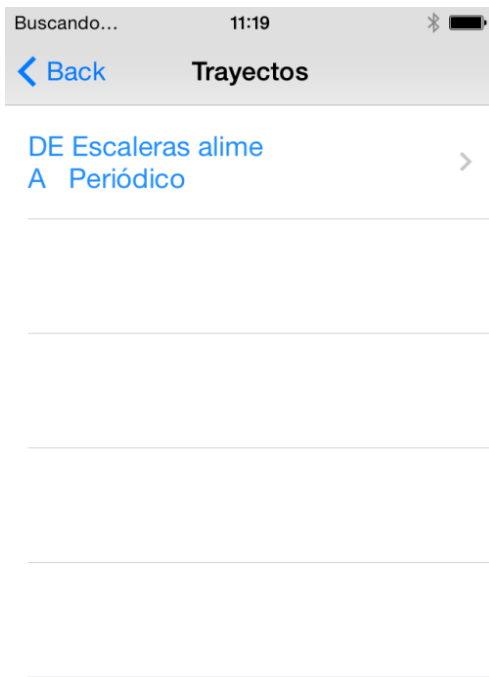


Ilustración 8. Lista de caminos.

Funciones principales:

- `(void)viewDidLoad`: existente en todos los views controllers, es la encargada, en este caso, de leer la base de datos (`datastore.sqlite`) de todos los paths, que es la estructura creada para cada camino, según se explicara más tarde.
- `(NSInteger)numberOfSectionsInTableView:(UITableView *)tableView`: hay que poner el número de secciones o grupos de celdas que queremos; en nuestro caso, como se puede ver en la figura de arriba, es uno.
- `(NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section`: se encarga de decir el número de celdas existentes; en este caso llamamos a otra función secundaria que se llama `existingPaths` en la cual se calcula dicho número.
- `(UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath`: trabaja con celdas indexadas y tratar con ellas posteriormente, como por ejemplo poner dos textos como hemos citado anteriormente.

- (void)tableView:(UITableView *)tableView willDisplayCell:(UITableViewCell *)cell forRowAtIndexPath:(NSIndexPath *)indexPath: usada para cambiar colores, por ejemplo al dejar pulsado, solo cambia contenido visual.
- (void) prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender: es la encargada de pasar a otro viewcontroller
- (void)notification: parte clave en este proyecto, es la encargada de leer cada vez que se inicia la aplicación un texto, informando sobre todas las acciones disponibles.

Accesibilidad:

Como hemos citado anteriormente, al pulsar una vez lee los textos de la celda y al pulsar dos veces pasa a guiar ese camino.

Al tener activo el reconocedor de gestos, al deslizar dos dedos para abajo, nos narrará todos los caminos existentes.

También hemos añadido un comentario al iniciar el view controller donde reproducirá el siguiente texto al ciego.

“La pantalla actual consta de una lista con celdas con trayectos guardados. Deslice dos dedos hacia abajo para oír todos los trayectos. Pulse dos veces una celda para seleccionar ese trayecto y acceder a navegación. Actualmente hay %d trayectos" %d es el resultado de una llamada a la función de existingPaths que hemos explicado anteriormente y que devuelve un integer.

4.3.3 Pantalla de guiado

Para acceder a este view controller es necesario pulsar sobre el botón de guía en la pantalla inicial y posteriormente seleccionar un camino pulsando sobre la celda elegida.

Este view controller es muy especial llamado **NavigationViewController**, sin lugar a dudas es el núcleo de la aplicación, ya que requiere de **NavigationManager**.

Hablaremos inicialmente de este último y posteriormente de NavigationVC que es el encargado de utilizar todas las funciones que le aporta el manager y también es el portador de la interfaz de guiado.

NavigationManager

No es un view controller simplemente, ya que añade una serie de funciones a las que podrá llamar UINavigationController y serán de vital importancia para la toma de decisiones.

Funciones Principales:

- `-(NSMutableArray *)orderArrayByDistanceToPoint:(CLLocation *)startPoint fromArray:(NSMutableArray *)desorderedArray:` función usada internamente entre funciones, calcula distancias entre un punto y los restantes.
- `-(NSMutableArray *)getShortestPathFromPoint:(CLLocation *)startPoint toPoint:(CLLocation *)finishPoint inArray:(NSMutableArray *)arrayOfPoints:` usado exclusivamente para arreglar errores de una mala utilización del terminal a la hora de guardar, concretamente arregla errores del tipo multipunto, es decir pulsar varias veces el guardar punto a distancias muy cortas.
- `(CLLocationDistance) calculateDistanceInPath:(NSArray*)path:` calcula la distancia total a recorrer en un camino, se realiza punto a punto.
- `(NSMutableArray*)calculateWholePathDirections:(NSArray*)path:` usada para el sistema de guiado concretamente, en la parte de Heading, es decir es una de las encargadas para guiarte de un punto a otro, apuntando con el terminal. Si el estamos andando en dirección equivocada el terminal nos avisara de que apuntemos antes de seguir la guía.

NavigationViewController

Es el view controller sobre el que se apoya la aplicación para el guiado. Requiere de NavigationManager, que le aporta una serie de funciones que hemos citado anteriormente.

Interfaz gráfica:

Se compone de dos partes claramente diferenciadas, un panel de indicaciones donde nos avisa del recorrido elegido, de las distancias a los puntos y sobretodo de los puntos peligrosos.

La otra parte es el botón de acción.

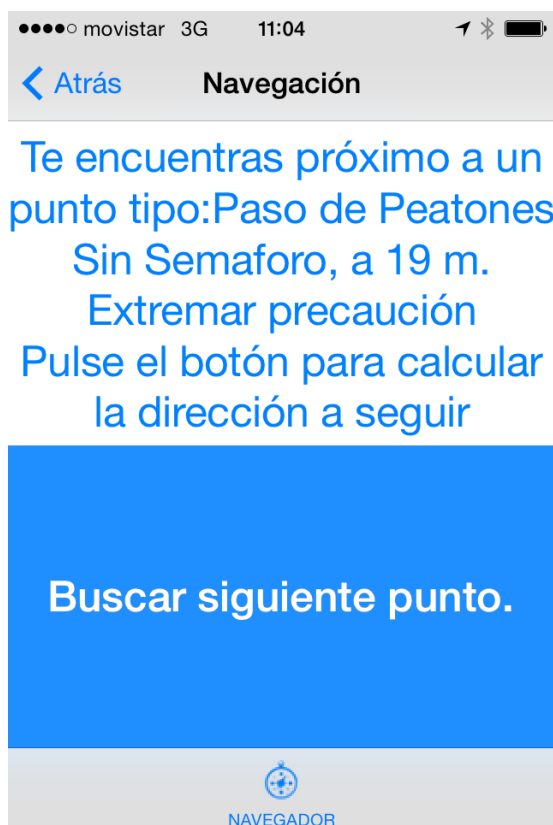


Ilustración 9. Pantalla de navegación.

Funciones principales:

- (void)viewDidLoad: existente en todos los viewcontrolles, en este caso se carga de calcular diversos aspectos del camino y también inicializa

varios booleans usados en las otras funciones para las tomas de decisiones. Otro aspecto importante es la inicialización del proceso de lectura de coordenadas y de la calidad de la misma, más información ver apartado 2.4.2.

- (IBAction)navigate:(id)sender: función creada por nosotros, además refleja una acción en la interfaz gráfica, el contenido de este botón se ejecuta cuando se pulsa el botón de acción (el botón de azul en la figura de la Pág.62). Su función es recalcular la dirección al punto, por si nos hemos desviado o para calcular la ruta al siguiente.
- (void)locationManager:(CLLocationManager *)manager
didUpdateLocations:(NSArray *)locations: usada para el guiado, calcula si se ha llegado al punto para actualizar las localizaciones.
- (void)locationManager:(CLLocationManager *)manager
didUpdateHeading:(CLHeading *)newHeading: similar a la anterior pero en términos de cabeza del terminal, es decir a que dirección apuntamos.
- (BOOL)didFindNextPoint:(MyAnnotation *)nextPoint: es la encargada de manejar las direcciones a seguir entre puntos. Da la orden para reproducir el tono de dirección encontrada y así avisara al ciego de que la dirección es óptima.
- (BOOL)isLastPoint: función para reconocer si se trata del último punto y si es afirmativo, se encarga de lanzar el aviso al ciego y dará la orden para reproducir la melodía de destino.
- (BOOL)didGetToNextPoint: es posiblemente la función de mayor importancia para el ciego, ya que es la encargada de lanzar los avisos por pantalla y da la orden para reproducir un sonido de cercanía y que el terminal vibre, cuando existen puntos peligrosos. La distancia de avisos es de diez metros.
- (void)vibrationOfDevice: es llamada para que el terminal vibre.
- (void)playSound:(NSString*)param: es la encargada de reproducir los tonos mp3.
- (void)showAlertView:(NSString*)param: es la encargada de realizar las alertas al ciego.

Accesibilidad:

Suponiendo que tiene VoiceOver activado, la aplicación avisará, como en casos anteriores, de la disposición y composición de los botones.

Incluyendo en todos los casos el paquete reconocedor de gestos que nos proporciona Apple en VoiceOver.

Será de vital importancia pulsar el botón de indicaciones, para conocer los metros al siguiente punto y ver así que seguimos el camino correcto al no incrementar la cantidad.

También se le pedirá al ciego, en caso de perder la conexión, que se pulse el botón inferior para recalcular direcciones.

Cada vez que el ciego se aproxima a un nuevo punto, a una distancia de cinco metros, el terminal empezará a vibrar y parará una vez tome el punto.

Hemos diseñado y desarrollado, haciendo uso del piano sintetizador y del programa VirtualDJ, tres tonos con el fin de ayudar a saber la situación actual al ciego:

- Cerca: cuando nos acercamos al punto.
- Dirección: cuando encontramos la dirección óptima al punto.
- Destino: cuando encontramos la dirección óptima.

4.3.4 Pantalla de menú principal

Para acceder a él, es necesario pulsar en **+** en la pantalla inicial. Se trata como ocurría en la pantalla inicial, de un view controller estándar. No requiere de ninguna función auxiliar.

Eso sí, se compone de seis botones, entre los cuales cabe destacar, el que nos conducirá a otro de los núcleos de la aplicación, **NewPathViewController** del que hablaremos posteriormente.

Los botones 1, 2, 3, 4 y 5 conducen a **Informacionesviewcontroller** que no tiene función alguna, solo realiza una acción de lectura de pdfs o web según se llame el identificador del botón.

Interfaz gráfica:



Ilustración 10. Pantalla de menú principal.

El no numerado (entre el 1 y el 5) es el hay que pulsar para iniciar el proceso de “crear caminos”, siguiendo los pasos establecidos que va indicando el sistema.

Mediante el uso de los botones 1, 2 y 3, se obtiene información, en modelo PDF, de los pasos de los procesos de creación (1) y guiado (2); asimismo, se visualiza información general (3) sobre la herramienta VoiceOver y los menús de la aplicación.

Los botones 4 y 5 conectan con el “Digital System Lab”, para realizar posibles consultas y recabar asistencia técnica (4), así como para acceder a la página Web y obtener conocimiento de otras aplicaciones y servicios disponibles (5).

NewPathViewController

Es sin duda uno de los view controllers más completos de la aplicación. Es la creación del camino en sí.

Para acceder al él es necesario pulsar la tecla + en la pantalla de inicio y luego el botón de crear camino.

Interfaz gráfica:

Se encarga en una misma pantalla de pedir nombre de inicio y final del camino.

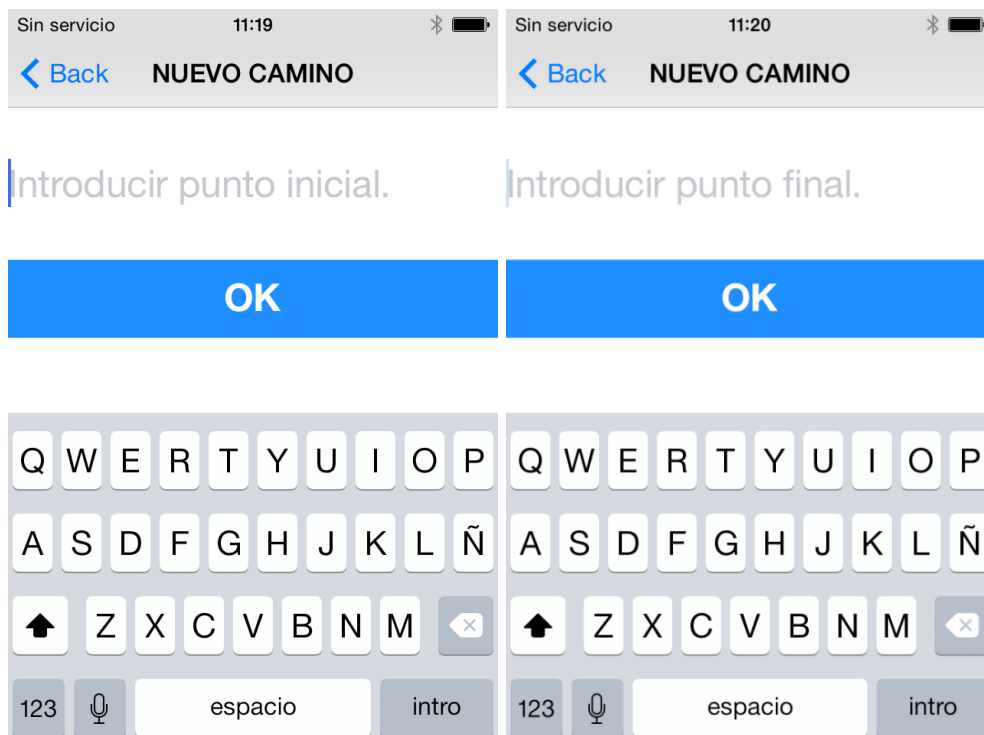
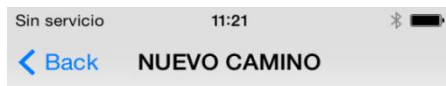


Ilustración 11. Pantalla de escritura del nombre del camino.

Posteriormente esconde ambos text fields y sus respectivos botones. Acto seguido activa los botones de la creación del camino.



Origen: Portal casa
Destino:Portal casa sierra
Puntos añadidos:0



Ilustración 12. Pantalla de creación del camino.

Compuesto de un cuadro de texto y ocho botones.

Funciones principales:

- (void)viewDidLoad: genera un nuevo objeto de la estructura path. También se encarga de la inicialización del proceso de lectura de coordenadas y de la calidad de la misma, más información ver apartado 2.4.2. Y por último se encarga de esconder los botones y mostrar el textfield de inicio del camino.
- (IBAction)addNewPath:(id)sender: se entra al pulsar el botón de ok del inicio del camino y cambia para guardar ahora el destino, que es un String, Al finalizar la escritura del destino, muestra los ocho botones y el cuadro de texto.
- (IBAction)startUpdating:(id)sender: se entra al pulsar el botón de comenzar, si no se ha comenzado no se podrá guardar puntos y si se ha comenzado cambiara el botón de comenzar a parar. Y mostrar otro botón para guardar el camino
- (IBAction)saveX(del 1 al 7):(id)sender: son los siete botones que aparecen en pantalla para guardar una coordenada, añade también una

etiqueta de tipo a la coordenada para después ser avisada en la guía al ciego.

- (void)locationManager:(CLLocationManager*)manager
didUpdateLocations:(NSArray *)locations: se encarga de gestionar la precisión horizontal mínima, si es correcta dará una orden para el terminal vibre y puedan guardarse puntos.

Accesibilidad

Todas las ayudas en este apartado, han sido enfocadas para personas sin discapacidad visual. Y son las siguientes:

- Sin pulsar en comenzar el terminal no guardará puntos.
- Si la señal GPS no es la óptima, mostrará un mensaje en pantalla advirtiéndole que no se dispone de la precisión mínima necesaria para que funcione, y no permitirá añadir puntos.
- Cuando la señal GPS sea buena como para tomar un punto el terminal vibrará.
- Se dispone de seis botones para puntos peligrosos, indicando con una imagen de que se trata. Posteriormente cuando se lea el camino avisar al ciego a una distancia de diez metros al peligro.
- Si no se está satisfecho con el camino guardado, cuando se le pulsa en parar, no se guardará el camino en la base de datos hasta que se pulse en el botón de guardar.

Estructura path

Esta estructura es el contenido que se guarda en la memoria y posteriormente será leída. Se compone de lo siguiente:

- NSString *beginning;
- NSString *end;
- NSArray *points;
- NSArray *typeOfPoint;
- NSArray *route;
- NSDate *time;

5. PRUEBAS, RESULTADOS Y CONCLUSIONES

5.1 Introducción

En este apartado hablaremos principalmente, como su propio nombre indica, de las pruebas realizadas. Empezaremos con una prueba de error de localización y después se llevarán a cabo otras dos pruebas más.

La primera de ellas consistirá en la creación de un camino, en el que cabe destacar la captura de un punto o baliza, que sólo será posible en determinadas zonas (con buena calidad en la recepción) y también, la correcta señalización de los puntos peligrosos, ya que serán de gran ayuda para la posterior guía.

Ya sabemos que, en la creación del camino, un 80% de los problemas futuros a la hora de guiarlo serán una mala manipulación al crear el propio camino; pero se presupone que el creador ha leído la información de creación y maneja con conocimiento el terminal.

La segunda prueba será la de guiado, en la cual se probará el teléfono en buenas y malas zonas de recepción, dando especial importancia al aviso de las zonas peligrosas y a que esté a una distancia óptima, así como al correcto uso de las funciones para la elección de la trayectoria vectorial para el guiado correcto.

Y, por último, se hará un test de pulsación masiva para evaluar si la aplicación se bloquea o se disparan los consumos.

5.2 Prueba de Error de Localización (CoreLocation VS Google Maps)

Se trata de uno de los mayores problemas que se presentan en la aplicación, al tener que usar el sistema GPS y observar cómo obtenemos bastantes variaciones respecto a sus valores en un mismo punto, llegando a ser en su punto más crítico de 12 metros y, en su punto más óptimo, de 3 metros.

Este comportamiento no es aleatorio, se trata de la calidad que tú le exijas a la aplicación. En el sistema iOS existe una serie de frameworks y métodos por los cuales puedes cambiar la calidad de la precisión; en concreto, se trata de CoreLocation. Disponemos de una serie de funciones facilitadas por Apple en las cuales podemos indicar cómo de preciso queremos que sea la coordenada.

Nos situamos en la siguiente zona y mostramos la coordenada de google.

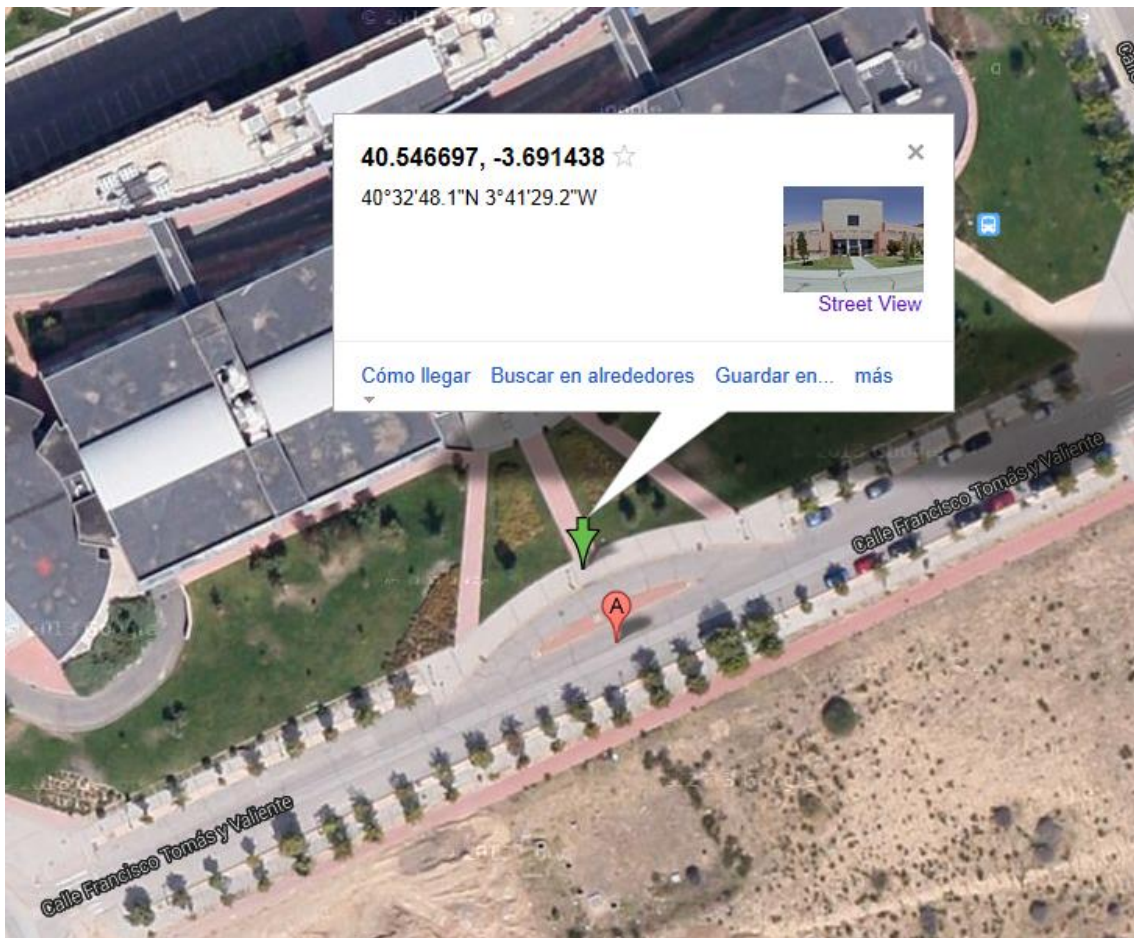


Ilustración 13. Pruebas de GPS

Hemos tomado varias coordenadas; la primera indicada en la tabla es el valor que nos da google y las siguientes son las medias de los valores que hemos tomado.

Coordenada bestacc, corresponde con el modo:

```
locationManager.desiredAccuracy = kCLLocationAccuracyBest;
```

Coordenada tenmet, corresponde con el modo:

```
locationManager.desiredAccuracy = kCLLocationAccuracyNearestTenMeters;
```

Ambos modos han sido creados por Apple y se nos facilitan con el paquete CoreLocation. Podemos observar cómo en el primero el error medio se sitúa en torno a los 4-6 metros; y en el segundo, en todos los datos que hemos ido tomando, en torno a las 11-13 metros.

COORDENADA GOOGLE	40.546697, -3691438
COORDENADA BESTACC	40.546731, -3.691398
COORDENADA TENMET	40.546590, -3.691365

Tabla 3. Pruebas de GPS

El principal problema es el consumo de la batería pues, con el modo CLLocationAccuracyBest, el nivel de la batería baja rápidamente.

Debido a que nuestra aplicación requiere la mejor localización posible, nos vemos obligados a usar este modo. De ahí que se recomiende, si los trayectos van a ser largos, el uso de una carcasa con batería incluida. En el mercado el valor de estas carcasas varía desde los diez hasta los cuarenta euros.

5.3 Prueba de creación de un camino

Iniciamos el terminal, cargamos la aplicación con VoiceOver apagado.

Vemos cómo nos carga el logotipo del laboratorio y lo mantiene durante 1.5 segundos.

Nos aparece el menú principal, a continuación mostramos los gráficos de consumo de memoria RAM y uso de los threads.

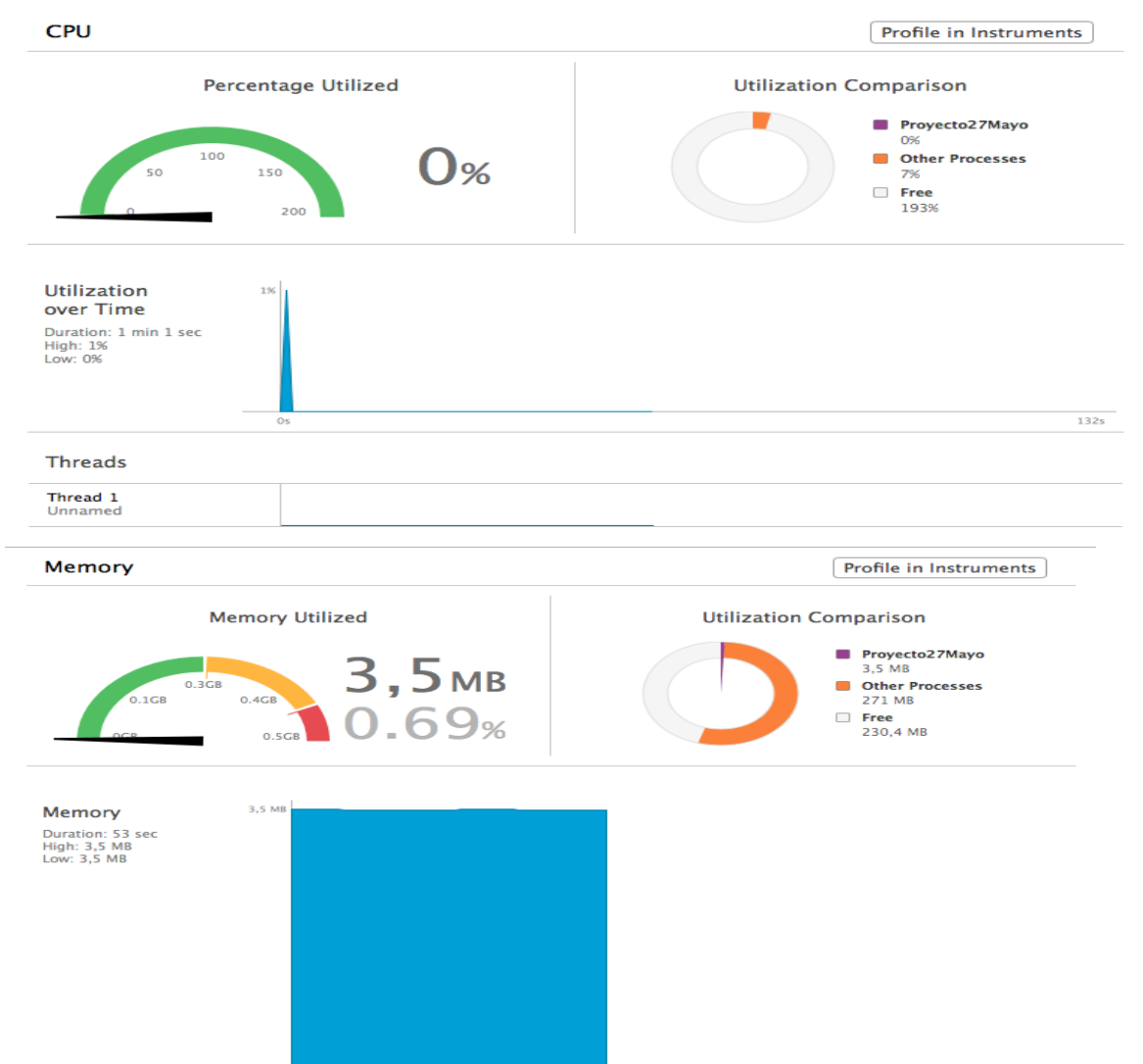


Ilustración 14. Rendimiento en la pantalla inicial.

Pulsamos en el botón + y accedemos al punto secundario de un menú de seis botones.

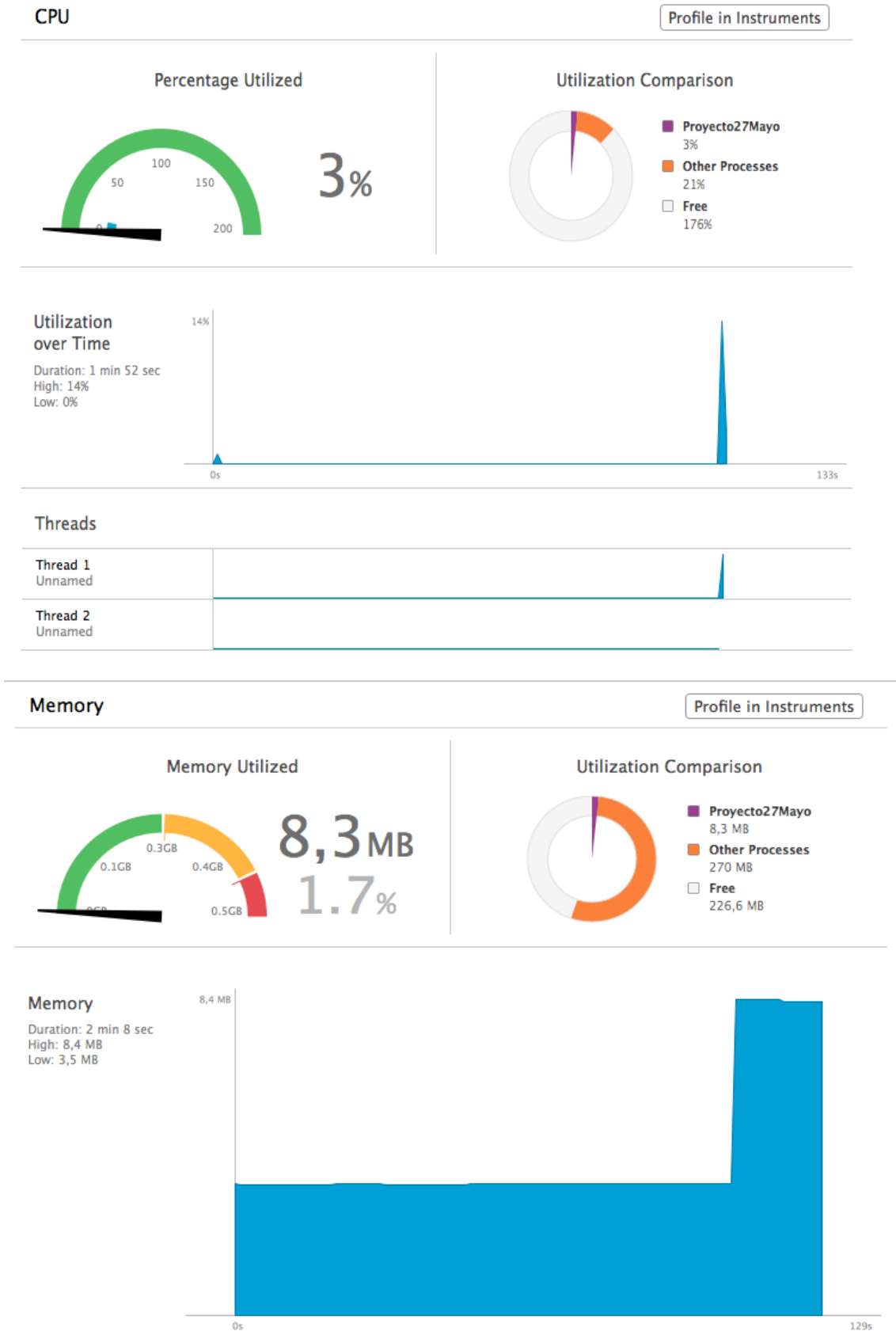


Ilustración 15. Rendimiento en el menú principal.

Pulsamos en crear camino y dejamos la aplicación en reposo.

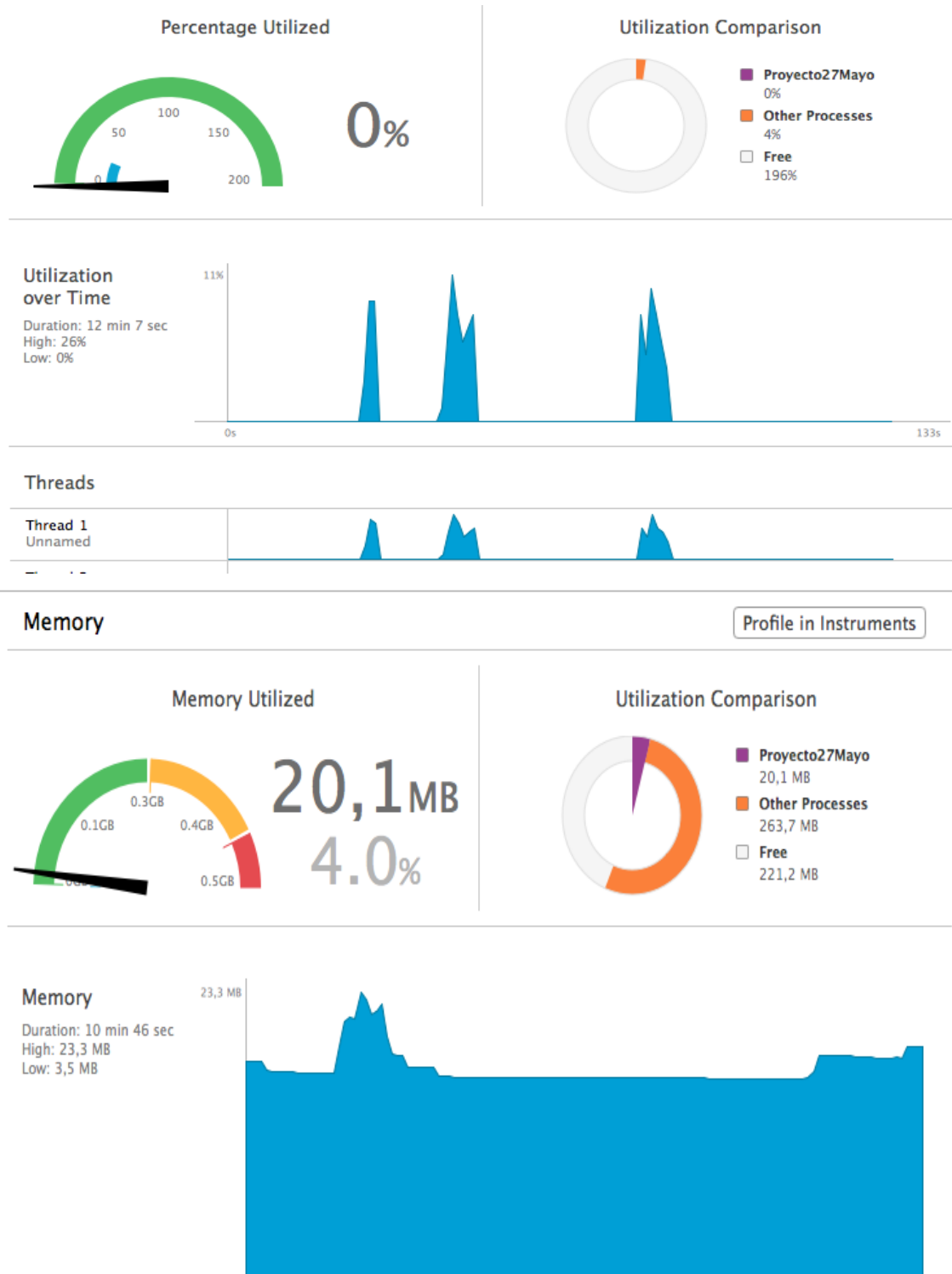


Ilustración 16. Rendimiento al pulsar el botón de crear camino.

Escribimos el nombre de inicio y comprobamos los niveles en el momento de la pulsación.

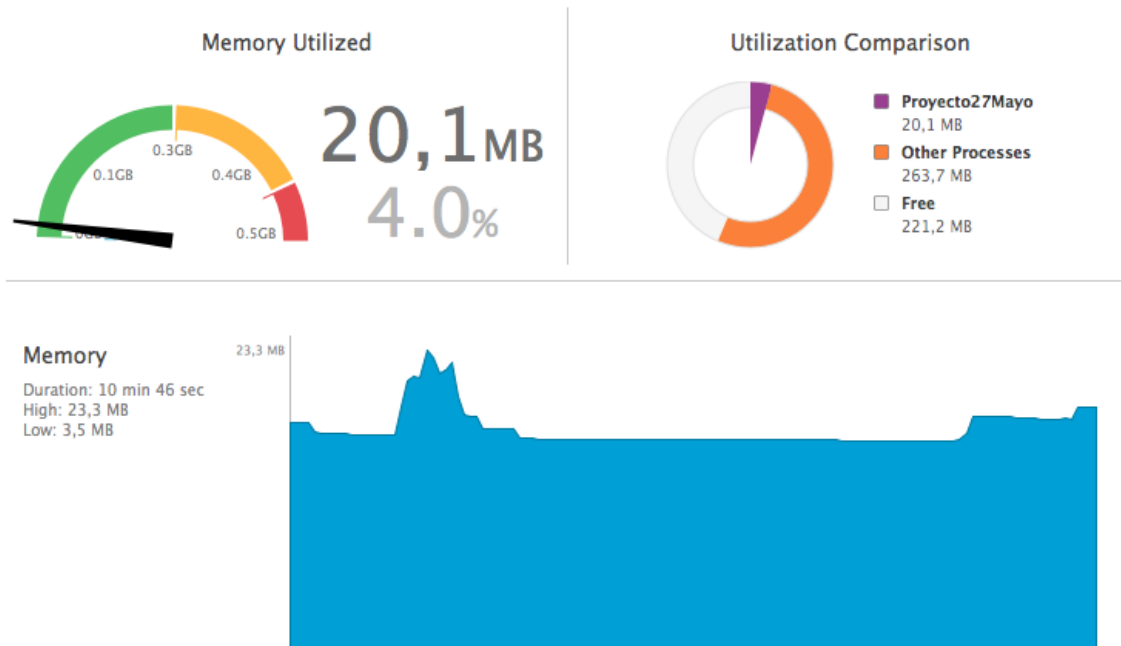


Ilustración 17. Rendimiento al escribir el nombre de origen en crear camino.

Vemos el rendimiento al cambiar a la zona de creación en sí.

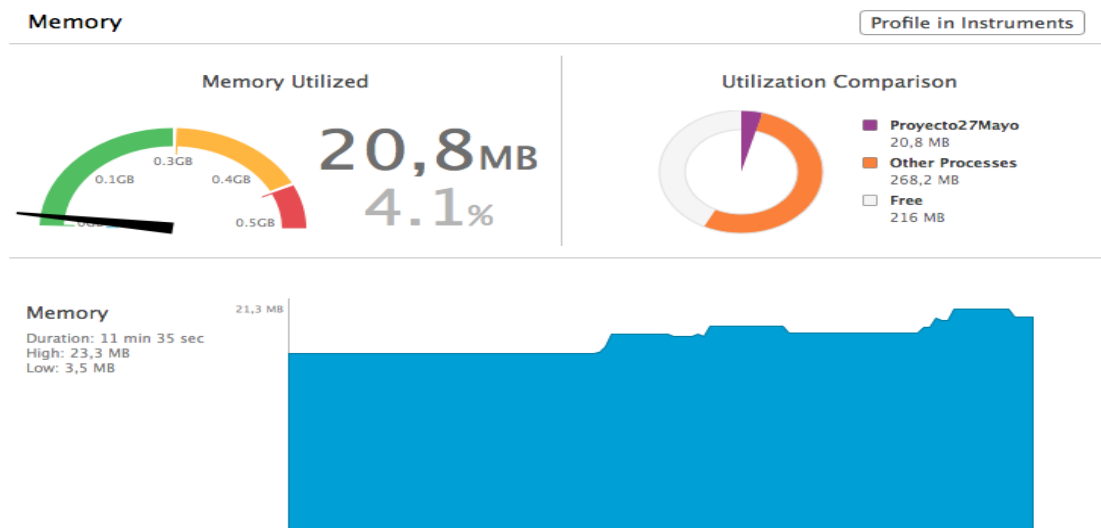


Ilustración 18. Rendimiento en la zona de comenzar a crear camino

Pulsamos en Guardar punto y observamos cómo no captura el punto al no estar iniciada la aplicación.

Pulsamos en comenzar y en guardar un punto en zona sin señal, nos aparece un aviso “No se dispone de la precisión requerida para añadir un punto, inténtelo cuando se disponga de ella (el teléfono vibrará)”.

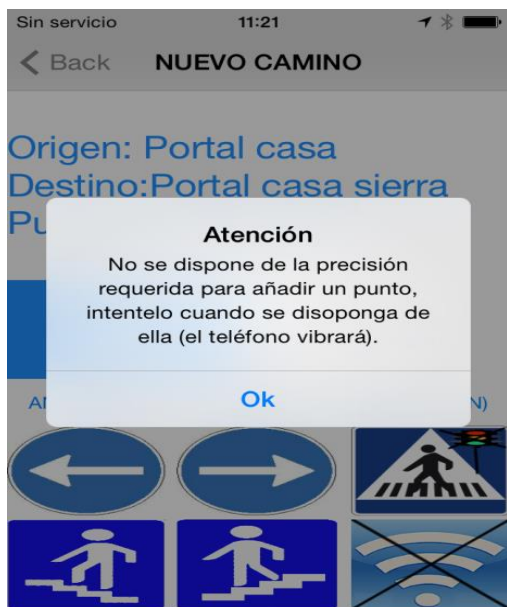


Ilustración 19. Pantalla de error por falta de precisión.

Posteriormente, accedemos a una zona con buena cobertura y observamos cómo podemos registrar los puntos sin problema.

Pulsamos en acabar y Guardar.

Se comprueba cómo efectivamente aparece el camino.

A título de conclusiones:

- Hemos observado cómo en zonas con poca señal, el terminal no vibra constantemente o directamente no vibra; por lo tanto, el punto puede llegar incluso a siete metros de error, tratándose del modo de máxima precisión. Por ello hemos añadido en la explicación de creación un manual de modo de empleo, ya que, será la parte más importante de la aplicación.
- Por este motivo, es probable que nos tengamos que alejar unos pasos del portal al iniciar, por ejemplo, un camino.

- Hemos preferido que el dato sea lo más preciso posible para poder avisar de los peligros y no ser simplemente una aplicación de guía sin precisión alguna; por lo tanto se mantiene el modo estricto en la adquisición de puntos. Creemos más importante mantener al invidente en la acera que la propia dirección en sí.
- El resto de los puntos y partes para la creación de un camino siguen su secuencia sin problema alguno; y el sistema de vibración explicado anteriormente no falla en situaciones favorables.
- En cuanto al rendimiento, se observan puntos relevantes a resaltar: por ejemplo, en el uso de los threads detectamos el pico de máximo uso cuando pulsamos el botón de iniciar un camino, pico que es normal, ya que en ese view, llamado Newpathviewcontroller, existen varios botones, etiquetas de texto editable y se guardan caminos en memoria.
- En cualquier caso, en el terminal simulado, iPhone 4s de dos núcleos, ni siquiera llegamos a usar el segundo, pues no le ha hecho falta más potencia.
- En cuanto al uso de la memoria RAM, el cambio más brusco se produce al entrar en el menú principal debido a que, al ser un Navigationcontroller, debe saber y poder ir tanto para adelante como para atrás, por lo que debe guardar muchos datos en memoria RAM y se dispara el consumo. Aunque no existe problema alguno, porque como se muestra en el gráfico, nos encontramos en la zona verde de uso.
Ilustración 13.
- En definitiva, nuestros datos de rendimiento son óptimos; pero, por poner alguna pega y debido siempre al uso del GPS, el terminal consumirá mucha batería y tenderá a incrementar su temperatura.

Prueba con persona anónima en la creación de un camino.

En esta parte, se procederá a mostrar las impresiones vistas por una persona externa a la aplicación. Ilustración 13. Rendimiento en la pantalla inicial. Se

trata de una mujer de 57 años, no muy habilidosa en el uso de Smartphones. Lo primero que hace al iniciar la aplicación es leer la cabecera, en la cual indica cuál es el menú para personas sin minusvalía.

Lo primero que hace al entrar en el menú es leer atentamente las informaciones de la aplicación. Nos comenta que se agradece el uso de imágenes y explicaciones, y que en su opinión, facilita el aprendizaje.

Después de dos lecturas de la información sobre cómo crear un camino, se arma de valor y se decide a crear un camino.

El camino se realiza en la localidad Tres Cantos y el viaje elegido fue ir desde la casa de nuestra tester hasta el quiosco de prensa, situado a unos 200 metros.

Escribe sin problema aparente el nombre y el final del camino.

Posteriormente aparece en la pantalla de guardar puntos, lo primero que hace es guardar un punto inicial, pero la aplicación no la deja. Tras varios segundos se da cuenta que no ha pulsado el botón de comenzar, tras pulsarlo guarda el primer punto.

Y coloca otro punto a unos 25 metros de este primero. Justo antes de llegar a un doble punto peligroso (hablamos de un giro a la derecha posterior a un paso de peatones), pone ambos puntos a unos tres metros de dicho peligro y posteriormente puntos normales hasta el quiosco.

Una vez terminado, guarda el camino y se sitúa en la pantalla inicial. Pulsa en la zona para invidentes y comprueba satisfactoriamente cómo tenemos el camino guardado.

En una entrevista final se muestra satisfecha, nos añade que cualquier persona podría manipular y entender nuestra aplicación sin ningún problema.

Por último, se le pregunta si le gusta visualmente la aplicación y nos contesta que sí, que el color azul le da un toque elegante y a la par es un color bonito.

5.4 Prueba de guiado (leer el camino)

Conectamos el VoiceOver, abrimos la aplicación y comprobamos cómo nos narra todo lo que se encuentra en el menú inicial.

Pinchamos en guiado y nos aparecen las celdas de selección del camino.

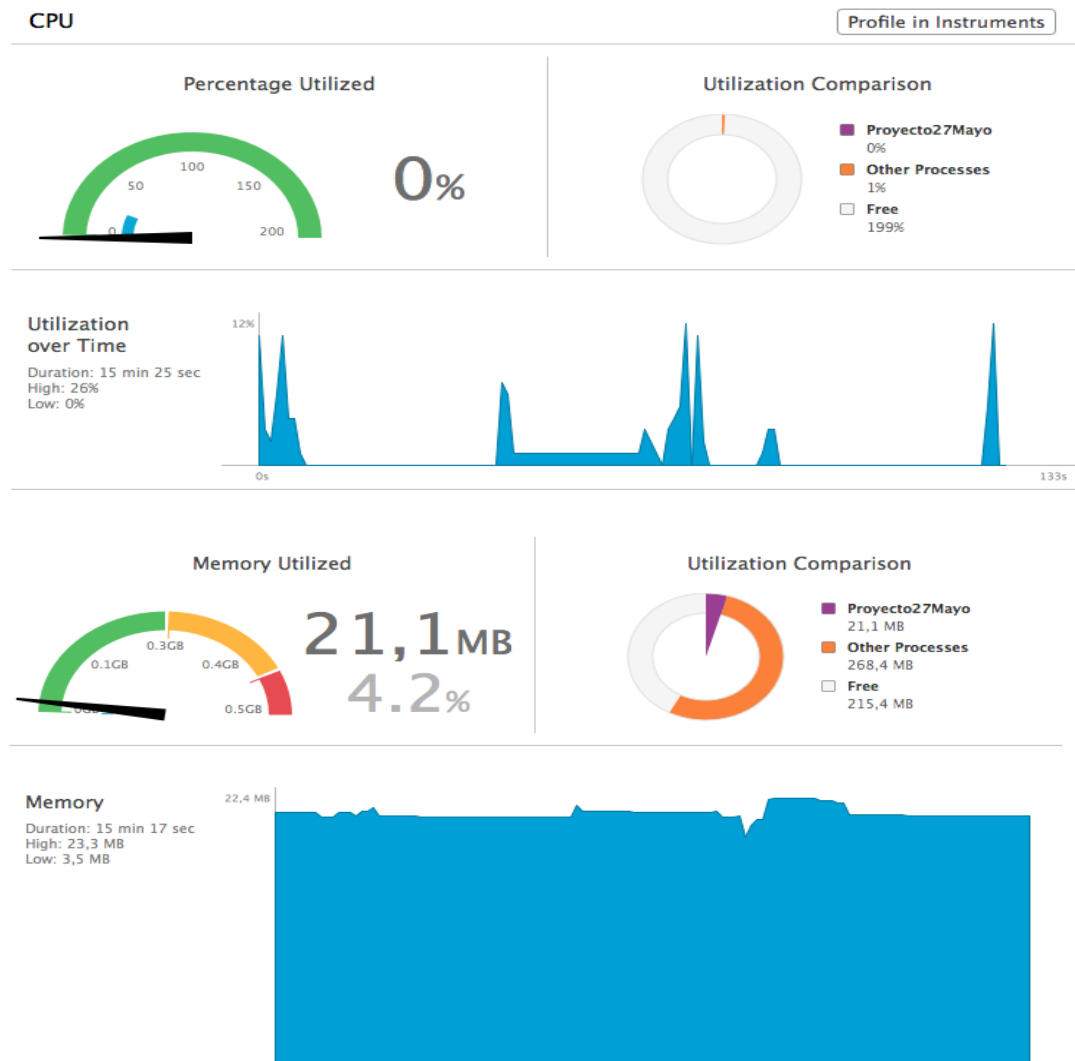


Ilustración 20. Rendimiento en la pantalla de lista de caminos

Pulsamos comenzar en la zona indicada como inicial, se presume que la persona guiado conoce el punto de salida, ya que, en el interior de un edificio será imposible.

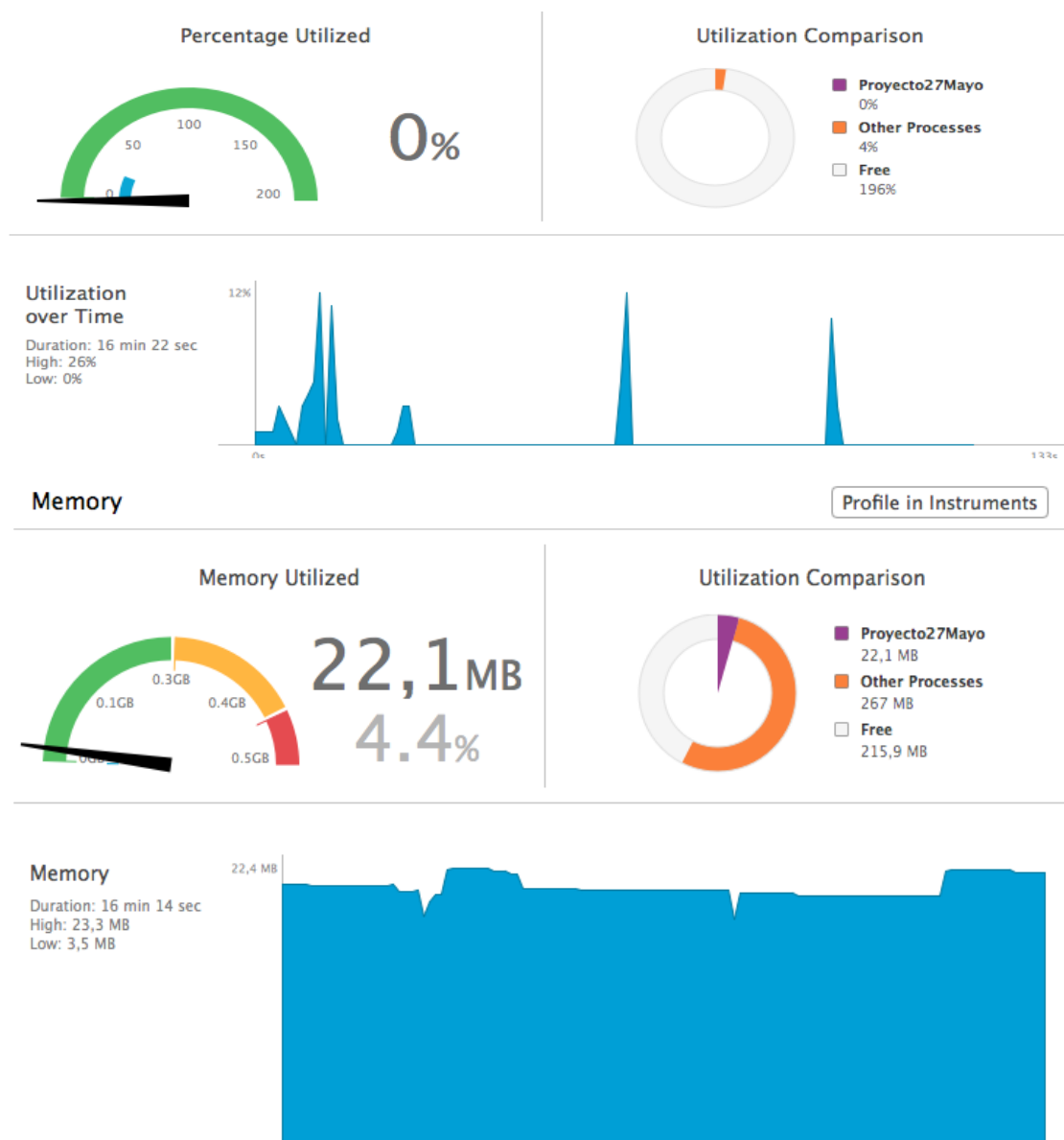


Ilustración 21. Rendimiento en la pantalla de guiado.

Nos avisa de haber llegado al siguiente punto.

Recordamos que para ayudar al ciego, reproduciremos una alerta mediante un sonido llamado dirección.mp3.

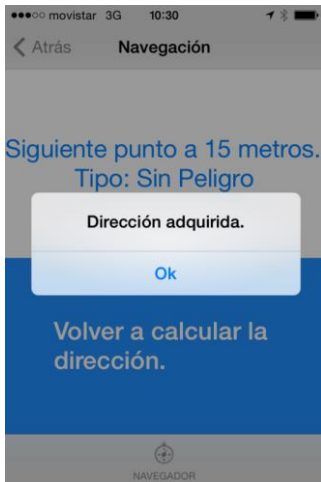


Ilustración 22. Dirección apuntada por CLHeading aceptada.

Pinchamos en volver a calcular la dirección para evaluar el comportamiento del terminal.

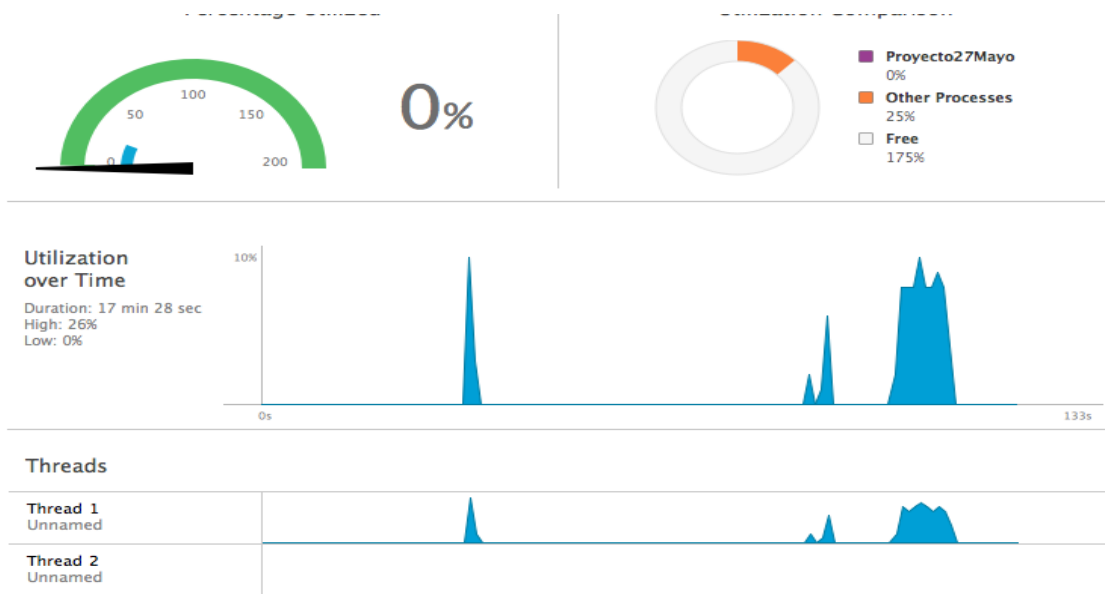


Ilustración 23. Rendimiento de la CPU en la pantalla de guiado al capturar una dirección.

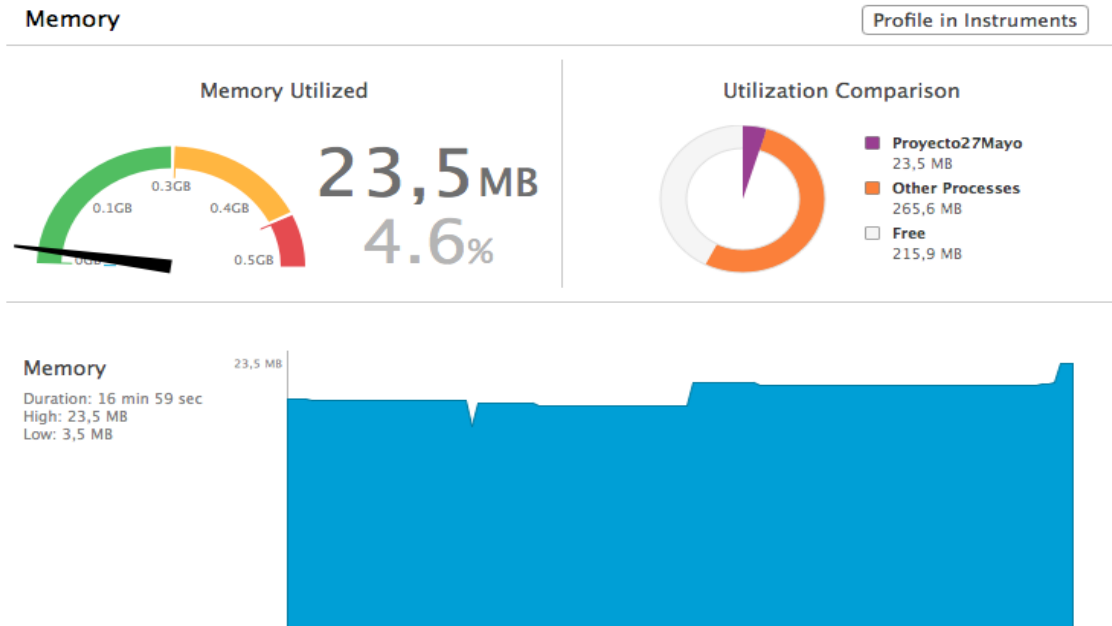


Ilustración 24. Rendimiento en la pantalla de guiado

Nos avisa de punto peligroso.

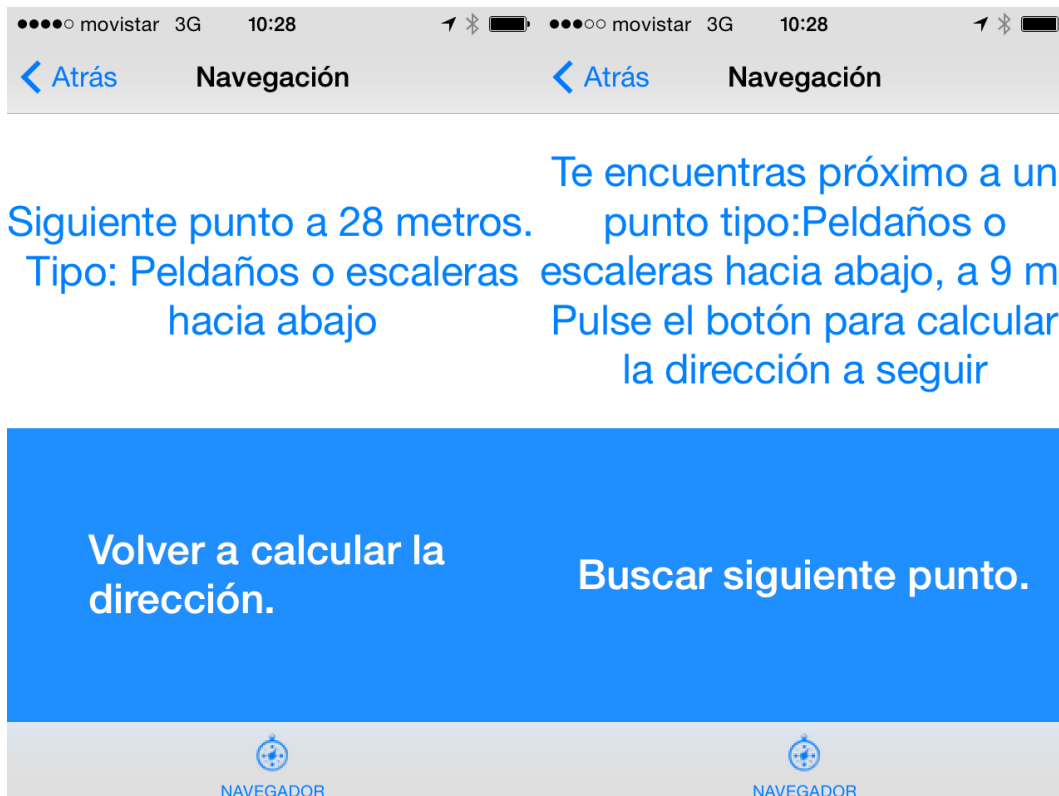


Ilustración 25. Pantalla de guiado, avisando de punto peligroso.

Probamos a ir en dirección contraria y vemos como efectivamente se incrementa la distancia y al volver a pulsar en calcular dirección nos devuelve a la senda necesaria.

Y confirmamos cómo la sintonía de fin y el aviso de haber llegado a su destino funcionan correctamente.

Conclusiones del guiado:

- Hemos observado claramente cómo en zonas con poca señal, la aplicación ha mareado un poco las distancias, podría ser perjudicial si por ejemplo se introdujese un punto en una zona como esa. Los síntomas serían los siguientes, bailarían las distancias en doce metros y sería muy difícil encontrar el punto, incluyendo la posibilidad de atravesar y tener que dar la vuelta.
- Por eso se pide al creador conocer el camino previamente y así evitar estos errores de navegación.
- En cuanto a la guía, comprobamos cómo a diez metros de un punto peligroso, la aplicación nos avisa para que el invidente esté en alerta.
- En relación con el rendimiento, se comprueba cómo se producen varios picos en el uso de la CPU, cómo siempre sus puntos de mayor auge son los cambios entre view controllers; pero, como pasaba en el caso de la creación, la aplicación no requiere ni en esos mismos puntos el uso del segundo núcleo del terminal.
- En cuanto al uso de la memoria RAM, como nos pasó en el caso de la creación al pasar del view controller inicial a ListOfPathsViewVontroller, sufrimos una subida grande memoria RAM, que siempre es debido al NavController para poder facilitar el paso entre estos. También se producen pequeñas variaciones debidas a las pulsaciones de botones, pero siempre dentro de lo normal.

- En consecuencia, tanto el sistema de creación como este de guía se encuentran dentro de los niveles de rendimiento óptimos fijados por el sistema iOS.
- Cabe recordar que el terminal usado para estas pruebas es un iPhone 4s y que tanto su memoria RAM como la frecuencia de los núcleos de la CPU son inferiores a los modelos superiores, iPhone 5, iPhone 5c e iPhone 5s.

5.5 Prueba masiva de pulsación

Con carácter extraordinario, se ha realizado una última prueba consistente en el paso entre view controllers de la manera más rápida posible, para identificar posibles bugs o lentitud de la aplicación.

El resultado de esta prueba ha sido exitoso, pues en ningún momento la aplicación se ha mostrado lenta o se ha cerrado automáticamente. El consumo se ha incrementado en un cincuenta por ciento, pero queda siempre dentro de la zona verde, que son los valores óptimos en cuanto a consumo de RAM y porcentaje de uso de la CPU.

Podemos observar en las siguientes imágenes cómo el uso de la CPU sube hasta un veintiséis por ciento y el consumo de RAM llega hasta 30Mb. Pero siempre usando un solo núcleo y en zona óptima de RAM (sólo un cuatro con seis por ciento de uso).

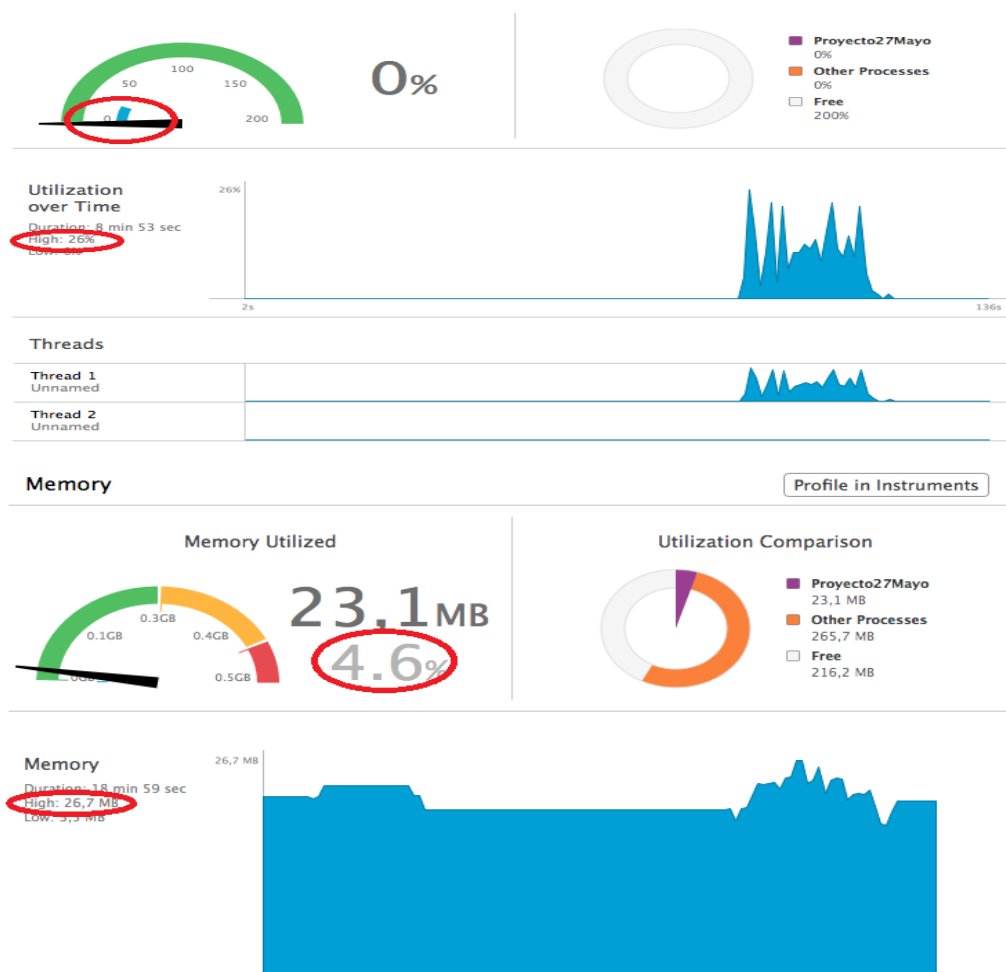


Ilustración 26. Rendimiento tras una prueba masiva de pulsación.

6. CONCLUSIÓN, APRENDIZAJE Y TRABAJOS FUTUROS

6.1 Conclusión

Una vez se ha concluido el diseño, desarrollo y pruebas de la aplicación las conclusiones son las siguientes:

- La aplicación es capaz de crear un camino sin problema alguno aparente, tras el periodo de prueba observamos como la aplicación se maneja en los niveles óptimos marcados por Apple para poder introducirla en su mercado de AppleStore.
- La aplicación es capaz de leer caminos, exceptuando los problemas típicos causados por baja señal de GPS o los causados por la pérdida de datos, bien sea wi-fi o 3G, la aplicación funciona correctamente, avisando al ciego en sus puntos peligroso y reproduciendo todo tipo de sonidos y alertas para ayudar a este en su guiado.
- La aplicación se muestra muy accesible para su uso, tanto por una persona sin mucha habilidad a la hora de manejar smartphones en la creación, como para el ciego, principal manipulador de la aplicación. Todos los programas como VoiceOver y todas las funciones añadidas por nosotros, con la finalidad de ayudar al ciego, funcionan correctamente y es más, no producen ningún gesto de lentitud en la aplicación.

6.2 Aprendizaje

El proceso de aprendizaje ha sido largo y duro, ya que no se conocían los conocimientos necesarios para la programación orientada a objetos. Si bien, todas las programaciones enseñadas en la facultad, tales como C, nos han servido de ayuda, aun así, nos ha hecho falta un trabajo previo de unos pocos meses.

También ha sido importante el apoyo de Tomás Merino, el elaboró la primera parte de este proyecto y muchas de sus funciones han sido utilizadas con algunas ligeras modificaciones. Todo esto ha hecho que después de ese aprendizaje del lenguaje, nos ha hecho falta un estudio previo del programa diseñado por Tomás.

Por tanto se ha aprendido el uso de los siguientes programas o habilidades:

- Conocimientos del mercado actual a nivel programador.
- Objective-C.
- Xcode.
- Geolocalización GPS.
- Localización Angular mediante 3G.
- Diseño y producción de iconos con CorelDraw x6.
- Diseño y producción de tonos de telefonía.
- Autogestión.

6.3 Trabajo futuros

Existen varios trabajos que se podrían mejorar la aplicación:

- Sin lugar a dudas el cinturón detector de obstáculos que actualmente se encuentra en proceso de desarrollo en el laboratorio, sería un gran añadido, ya que podría enviarnos vía bluetooth un aviso y después nosotros reproducir la alerta con el terminal.
- Sería óptimo cambiarse al nuevo sistema de localización introducido en iOS 8, así como usar las nuevas ventajas del nuevo terminal iPhone 6. Con el nuevo terminal y haciendo uso de su procesador m7, mejora sustancialmente la coordenada GPS en espacios interiores, lo que supondría una mejora grande en nuestro proyecto.
- La coordenada actualmente es en 2D, podría estudiarse el uso de la coordenada 3D para por ejemplo indicar desniveles u otros tipos de obstáculos, Su uso sería interesante por ejemplo podríamos medir la longitud de un tramos de escaleras y así ver cuantos peldaños tiene, para luego poder avisarlo al ciego.
- El sistema de guiado actual se basa en ajustar ángulo y la distancia entre puntos, sería una opción probar a usar solo el punto y trazar una dirección sobre los últimos puntos para ajustar el ángulo, aunque podría ser inviable.
- Actualmente el único aviso de dirección equivocada es la distancia en metros en el panel informativo superior, en la pantalla de guiado. En un primer momento se realizó una función encargada de gestionar una mala dirección, pero resulto causar muchos problemas en zonas con mala cobertura, provocando graves problemas en el funcionamiento de la aplicación. Actualmente se dispone de dicho panel y botón para recalcular la ruta. Pero posiblemente con los nuevos cambios en el sistema de guiado en iOS 8 pueda retomarse e intentar recuperar dicha función.
- Podría crearse un servidor que guardase todos los caminos creados por todas las personas y así poderlos descargar y posteriormente guiar cualquier ciego. Esto supondría una gran mejora en la aplicación, ya que solo hay una aplicación conocida de guiado capaz de enviar direcciones,

no es punto a punto y mucho menos para personas con discapacidad visual.

- Aumentar el número de peligros posibles, poniendo un slider o algún tipo de bar creado por nosotros, con el fin de aumentar el mayor número de puntos peligrosos, pero podría hacer la parte de creación pesada e incluso podría dificultarla, si la persona que lo realiza no está muy acostumbrada al uso de terminales inteligentes.

7. ANEXOS: INFORMACIÓN ADICIONAL

Anexo I Referencias Bibliográficas

Agradecimientos especiales, por la información obtenida de sus escritos y publicaciones, a:

- Fundación ONCE
- Tomás Merino Mateo: por su Proyecto Fin de Carrera “Desarrollo en iOS de Aplicaciones de Guiado en GPS para Discapacitados Visuales”, de marzo de 2013.

Libros:

IOS 7 programming fundamentals Objective-C, Xcode, and Cocoa Basics, Neuburg,
Editor: O'Reilly.

Desarrollo en IOS de aplicaciones de guiado GPS para discapacitados visuales, Tomás Merino.

Learning Cocoa with Objective-C 2^aed, Davidson, James Duncan, *Editor*:
O'Reilly.

iPhone SDK programming developing mobile applications for Apple iPhone and iPod Touch, Ali, Maher, Editor: John Wiley & Sons.

Enlaces internet:

Utunes: Curso Online Gratuito de iOS Universidad de Stanford:
itunes.stanford.edu

http://es.wikipedia.org/wiki/Tecnolog%C3%ADas_de_la_informaci%C3%B3n_y_la_comunicaci%C3%B3n

<http://es.wikipedia.org/wiki/Tiflotecnolog%C3%ADa>

http://www2.gobiernodecanarias.org/educacion/17/WebC/eltanque/lossentidos/organosdelossentidos_p.html

<http://www.monografias.com/trabajos16/los-cinco-sentidos/los-cinco-sentidos.shtml>

<http://www.scientificpsychic.com/workbook/sentidos-humanos.html>

<http://www.banrepcultural.org/blaavirtual/ninos/web5sentidos/index.htm>

<http://www.once.es/new>

<http://www.fundaciononce.es/>

<http://es.wikipedia.org/wiki/Ceguera>

<http://ceguera.net/>

<http://www.salud180.com/salud-z/ceguera>

http://html.rincondelvago.com/ceguera_1.html

<http://www.who.int/mediacentre/factsheets/fs282/es/>

http://es.wikipedia.org/wiki/Ceguera#Perros_gu.C3.ADa

<http://perrosguia.once.es/home.cfm?id=2&nivel=1>

<http://www.perrosguiadeandalucia.org/historia>

<http://docum.x10.mx/soc/invidentes.htm>

http://es.wikipedia.org/wiki/Sistema_de_posicionamiento_global

<http://redesnani.weebly.com/sistema-android-ios-blackberry-os-symbian-windowsphone.html>

<http://www.appsmag.es/805/la-telefonía-móvil-al-servicio-de-la-discapacidad-visual/>

<http://software.computadora-discapacidad.org/>

<http://somosiphone.com/como-ayuda-el-iphone-a-usuarios-ciegos-o-con-deficiencia-visual>

<http://www.bloggueros.es/2013/08/ios-7-brujula.html>

<http://es.gizmodo.com/ios-7-a-fondo-estas-son-todas-las-novedades-que-veras-1337341897>

<http://www.esferaiphone.com/iphone/los-usuarios-prefieren-el-diseno-de-ios-7-frente-al-ofrecido-por-ios-6/>

<http://www.20minutos.es/noticia/1947635/0/apps-android/windows-ios-mitad/muertas/>

http://es.wikipedia.org/wiki/Recursos_humanos

http://www.sepe.es/contenido/empleo_formacion/empresas/contratos_trabajo/a_sistente/pdf/guia_contratos.pdf

http://www.sepe.es/contenido/empleo_formacion/empresas/contratos_trabajo/a_sistente/pdf/indefinido/Indefinido.pdf

http://www.sepe.es/contenido/empleo_formacion/empresas/contratos_trabajo/a_sistente/pdf/temporal/Temporal.pdf

http://arantxa.ii.uam.es/~euroform_dslab/

http://arantxa.ii.uam.es/~euroform_dslab/android.htm

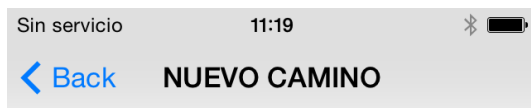
<http://www.cocoaosx.com/2014/01/30/ibeacons-ios7/>

<http://www.punteroavoid.com/blog/2012/03/27/donde-estoy-geolocalizacion-inversa/>

Anexo II Tutoriales mostrados en el menú principal

INFORMACIÓN MOSTRADA POR EL BOTÓN 1: “CÓMO CREAR UN CAMINO”

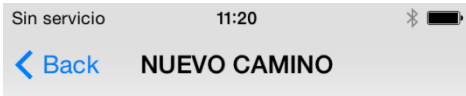
En el momento en que se pulse el botón 1, de crear camino, nos aparecerá una línea para escribir el nombre del punto inicial del camino. Se recomienda indicar de la manera más precisa posible en qué sitio se inicia, indicando por ejemplo cuál de las salidas de portal queremos elegir, en el caso de haber varias. Un ejemplo sería “Portal Casa Panadería”.



Introducir punto inicial.



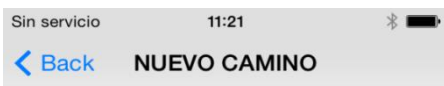
Una vez se pulse el ok, nos aparecerá otro cuadro de texto para poder indicar cuál será el nombre del destino. Una vez más será de vital importancia definir con la máxima precisión posible cuál será el nombre a dale, indicando por ejemplo el nombre de la calle y número. Un ejemplo sería “casa Alicia calle Índico 3”.



Introducir punto final.



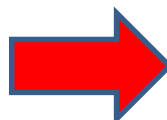
Posteriormente, aparecerá una pantalla en la cual se nos informará del nombre de inicio y destino, así como, el número de puntos añadidos. Dispondremos, a continuación, de varios botones: uno para comenzar el camino y los otros para guardar los puntos normales del recorrido, así como los seis puntos con peligro. La aplicación se basa en guardar una serie de puntos o balizas que, posteriormente, se juntarán de uno en uno creando una conexión punto a punto; y se usará para el guiado de forma vectorial.



Origen: Portal casa
Destino: Portal casa sierra
Puntos añadidos: 0



AÑADIR PUNTO PELIGROSO(PULSAR IMAGEN)



BOTONES DE PUNTOS PELIGROSOS

El sistema de guiado posee un modo en el cual, si la señal no es la óptima, no nos permitirá guardar el punto; para ello, avisará con una vibración cuando así sea posible.

Cabe destacar los puntos sin peligro, poniendo uno cada treinta metros, y los puntos con peligro, respecto de los cuales se avisará unos veinte metros antes, con el fin de advertir, por ejemplo, de un tramo de escaleras descendente o ascendente, giros de noventa grados a izquierda o derecha, de una zona con pérdida de señal (cuando no se dispone de vibración durante un período mayor a tres segundos) o de un paso de peatones sin semáforo.

Los puntos deben ser puestos de la siguiente manera:

- 1. Poner puntos sin peligro cada treinta metros, unos treinta y cinco pasos.**
- 2. Los puntos con peligro situarlos a cinco metros antes de dicho peligro, debido a que el error del GPS es de tres metros.**

Ejemplo:



Un primer punto sin peligro sería el punto naranja; posteriormente, a unos treinta metros, pondríamos un punto peligroso, que sería el rojo avisando, en este caso, de un paso de peatones sin semáforo; pero también podría ser un giro peligroso a derechas, debido a que, si no se avisa, podría el invidente entrar en la carretera.

Tras haber terminado el camino, se pulsará la tecla de parar y posteriormente, en caso de que queramos guardarlo y que aparezca en la lista de caminos para la guía, pulsaremos en guardar.

Una vez se ha guardado el camino, lo encontraremos en el apartado de guía y mostrado por la etiqueta escrita como inicio y final del camino dentro de una celda de una tabla.

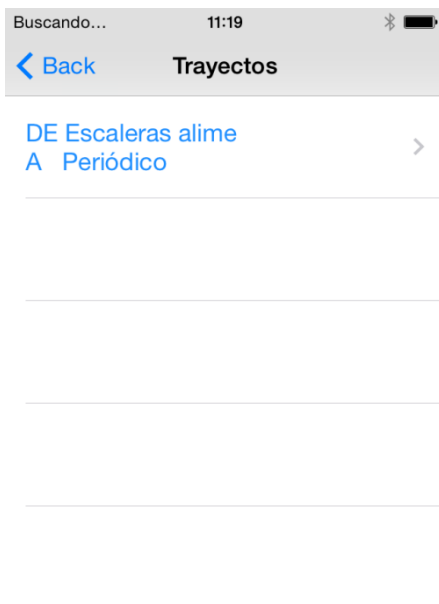
INFORMACIÓN MOSTRADA POR EL BOTÓN 2: “GUIADO”

ES IMPRESCINDIBLE TENER VOICEOVER ACTIVADO PARA EL SISTEMA DE GUIADO (más detalles en información general)

Recordar que esta parte debe ser explicada de manera detallada, en caso de ser invidente y que, en ningún momento, esta aplicación sustituiría el uso del bastón, sino que, serviría de complemento y ayuda.

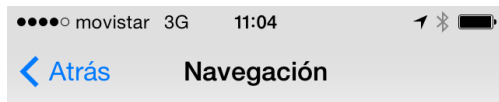
El teléfono en la guía debe estar en posición horizontal con el altavoz mirando al frente, ya que usamos un “sistema brújula” para identificar trayectorias.

Lo primero que se encontrará, al pulsar este botón, será una tabla con los distintos caminos que han sido grabados previamente por una persona sin discapacidad visual.

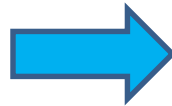


Se nos avisará mediante VoiceOver (más detalles en información general) de todos los caminos disponibles. Recordar que cuando se pulse una vez nos leerá el nombre y tras pulsar dos veces seleccionaremos el camino.

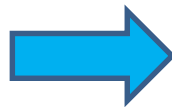
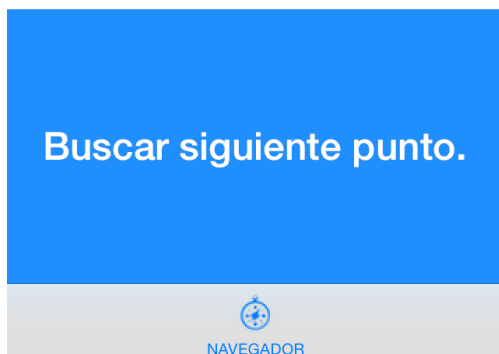
Después, tras seleccionar un camino tendremos una primera ventana en la cual debemos pulsar en el botón de actuación y empezará a leer el camino, indicándonos la distancia al primer punto guardado. Hay que resaltar y recordar que el camino debe empezar en zonas con señal estable; caso contrario, existe una probabilidad muy alta de mal funcionamiento de la aplicación.



Te encuentras próximo al siguiente punto, a 9 m.
Extremar precaución
Pulse el botón para calcular la dirección a seguir



Línea de información donde nos avisa de proximidad y si debemos actuar



Botón de actuación

En la figura anterior, podemos distinguir las dos zonas citadas anteriormente: la zona blanca es la zona de información y la azul es el botón a pulsar.

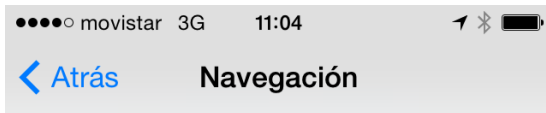
Como tenemos el VoiceOver activado, con pulsar encima de la parte blanca nos indicará el contenido del mismo. En este caso, nos dirá: "te encuentras próximo al siguiente punto, a 9m extremar precaución, pulse el botón para calcular la dirección a seguir".

Existen puntos con peligro, esos puntos son extremadamente importantes, de ahí que se nos avisen de con mucha antelación concretamente veinte metros.

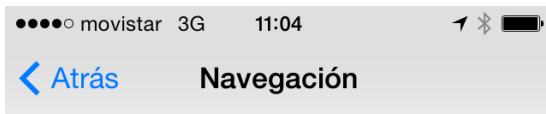
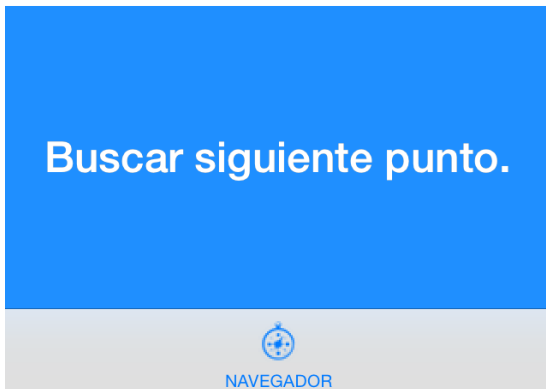
Estos puntos son los siguientes:

- Giro a izquierdas
- Giro a derechas
- Tramo de escalones ascendente
- Tramo de escalones descendentes
- Paso de peatones sin semáforo
- Zona con pérdida de señal

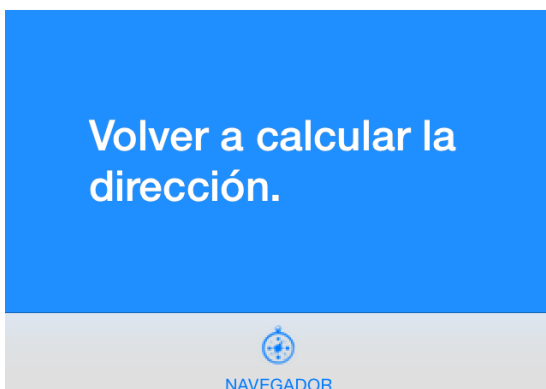
Todos estos puntos peligrosos serán avisados de la siguiente manera: a veinte metros saltará una alerta avisándonos del peligro; y, además, mostrará en la parte de indicaciones un aviso, que podrá ser pulsada para consultar la distancia y el tipo, cuantas veces se quiera.



Te encuentras próximo a un punto tipo:Paso de Peatones Sin Semaforo, a 19 m.
Extremar precaución
Pulse el botón para calcular la dirección a seguir



Siguiente punto a 18 metros.
Tipo: Giro a Izquierdas



Cuando se llegue al punto final del camino, se lanzará un clip de música y se comunicará el haber alcanzado el destino deseado.

INFORMACIÓN MOSTRADA POR EL BOTÓN 3: “INFORMACIÓN GENERAL”

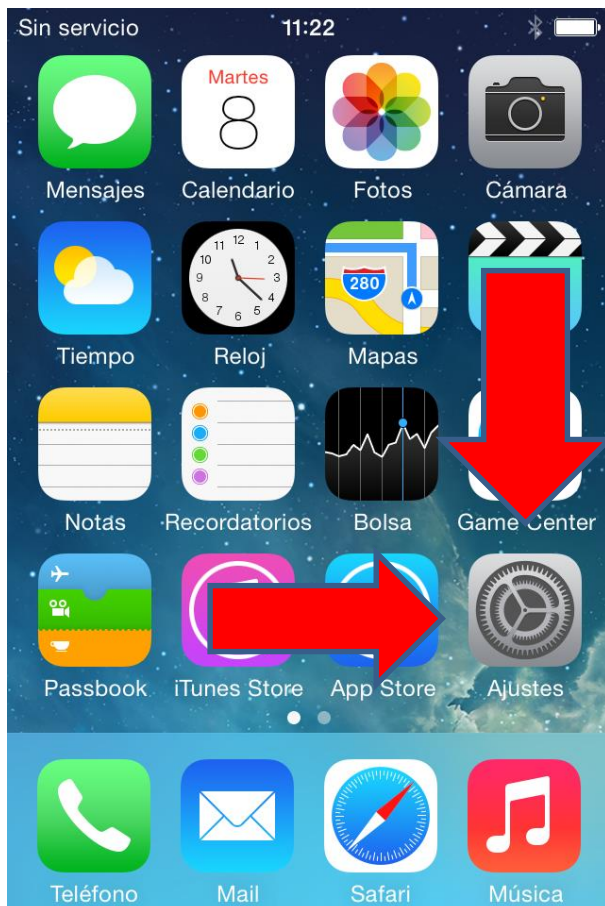
La información general está estructurada en dos partes: la primera relativa a la herramienta VoiceOver; y la segunda a los menús de la aplicación.

Parte 1: VoiceOver.

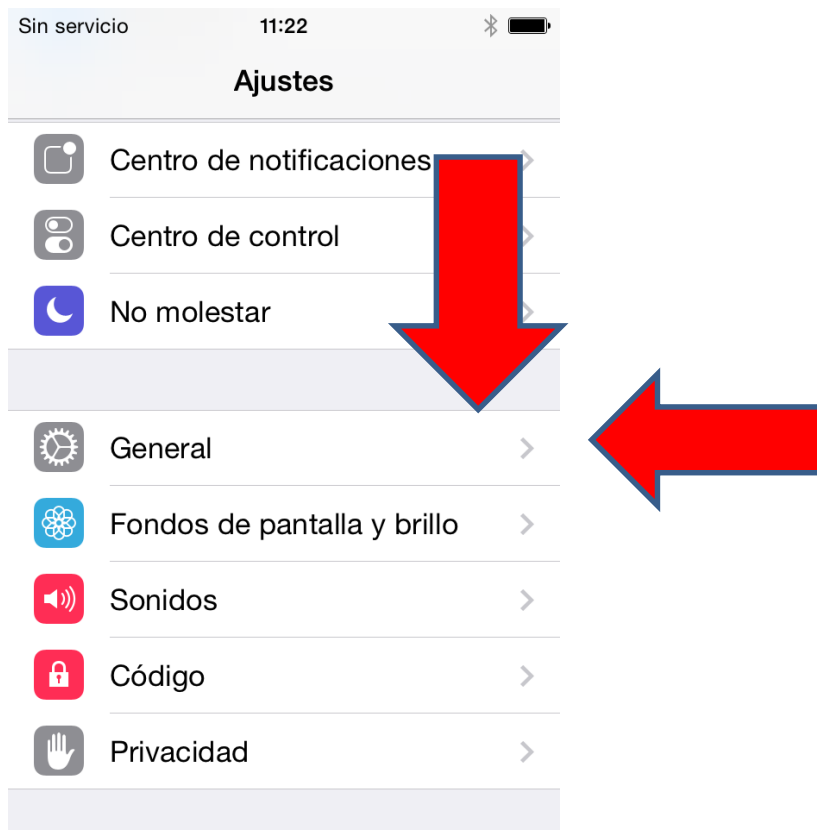
Tiene que estar en modo activo, ya que es la utilidad encargada de leer lo que pulsamos y será, de vital importancia, si como es el caso tratamos con personas con discapacidad visual.

Para activar es necesario seguir la siguiente ruta:

- Entrar en ajustes



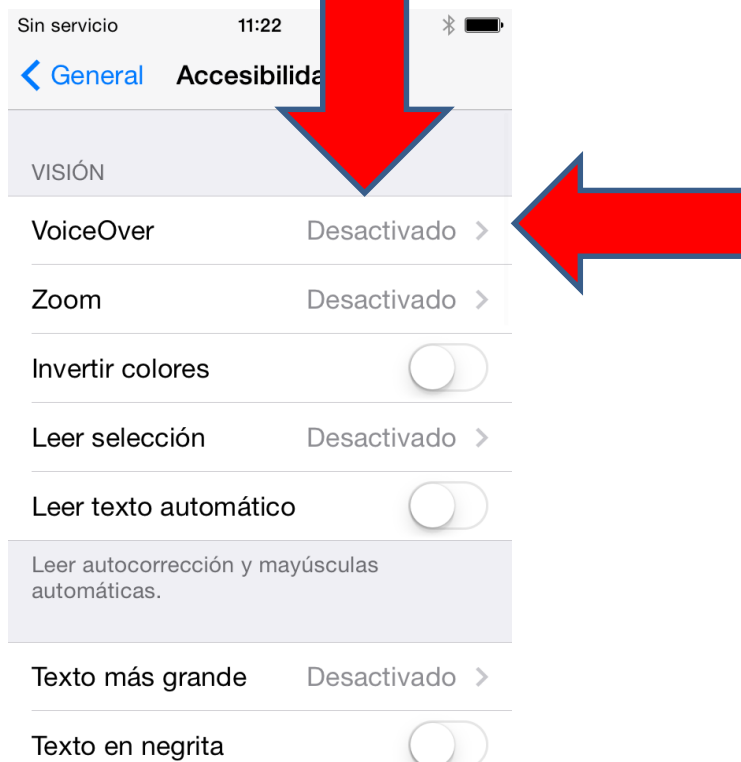
- Posteriormente seleccionar General:



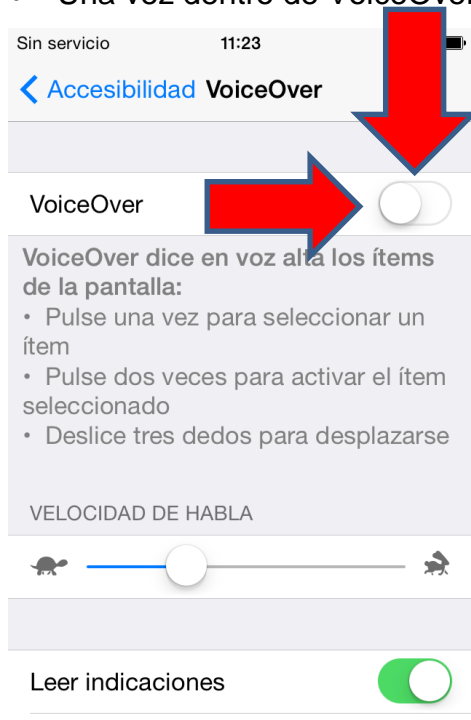
- Luego Accesibilidad:



- Después seleccionar VoiceOver:



- Una vez dentro de VoiceOver, activarlo:



En este menú de VoiceOver podemos seleccionar la velocidad del habla a nuestro gusto.

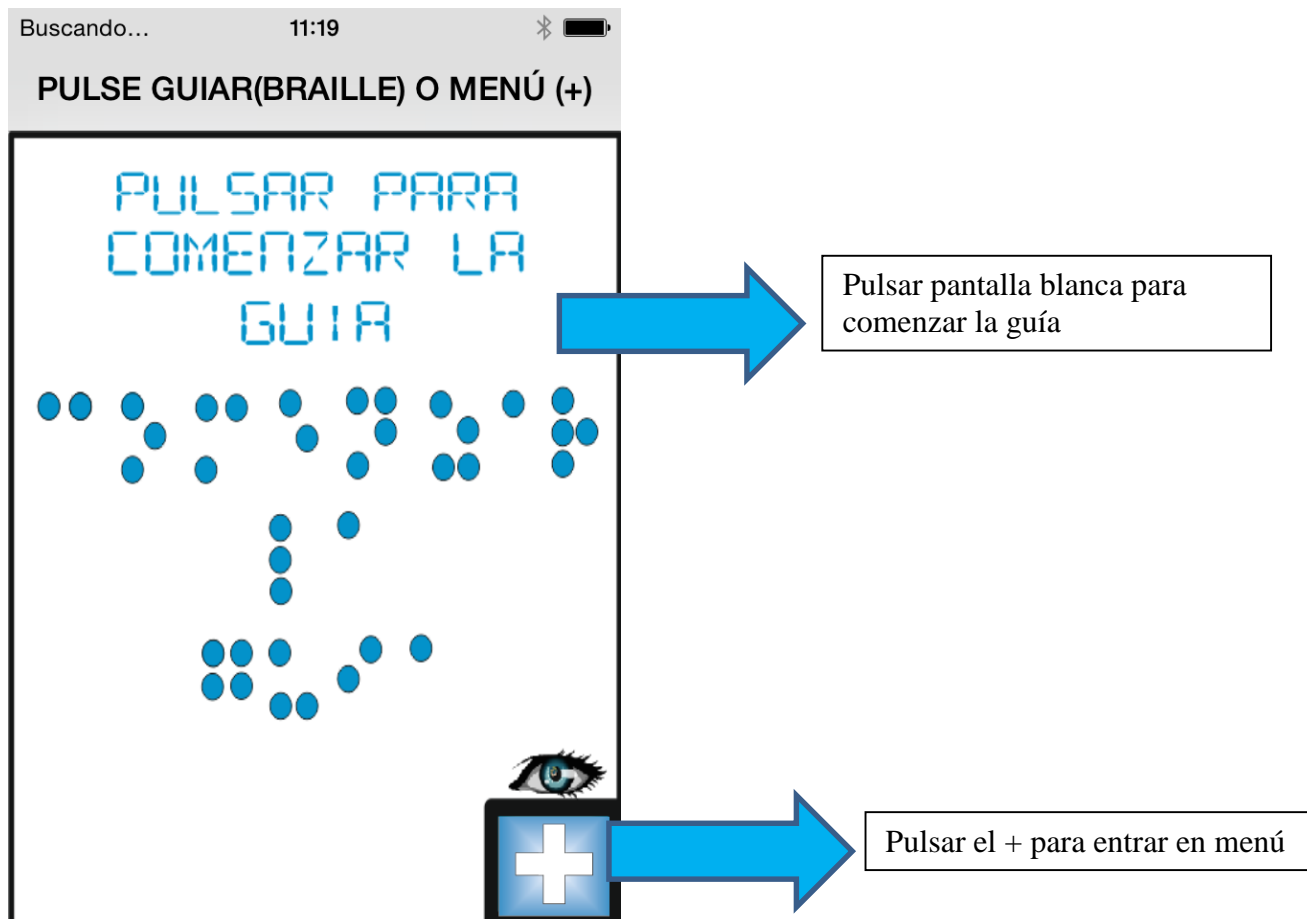
RECORDAR QUE VOICEOVER DEBE ESTAR SIEMPRE ACTIVO PARA LA GUÍA, ASÍ COMO TODAS LAS OPCIONES QUE SE REFLEJAN EN ESTE MENÚ COMO LEER INDICACIONES, ETC...

Sin embargo, a la hora de crear un camino, recomendamos su desactivación.

Más información: <http://www.apple.com/es/accessibility/ios/voiceover/>

Parte 2: Menú de la Aplicación.

Como ya ha quedado indicado con anterioridad, una vez se inicia la aplicación, aparece la pantalla principal contiene partes: la guía y el menú +.



Pulsando la parte blanca de la pantalla (la más grande), accedemos a la sección de “**guiado**”, siendo indispensable tener activado VoiceOver.

En cambio, si elegimos la “opción +” (cuadrícula inferior derecha de la pantalla), entraremos en una segunda zona, en la cual dispondremos de un “**menú principal**” ordenado en seis botones con imagen.



El primer botón, empezando de izquierda a derecha y de arriba abajo, es el que nos llevará a “**crear camino**”.

Guiado y camino son, obviamente, las utilidades más relevantes del sistema, pues los restantes accesos son ayudas, bien para obtener la información sobre

los dos procesos anteriores, bien para obtener asistencia o información general del “Digital System Lab”.

Para más información sobre guiado y camino, ver los apartados “3.3.1 INFORMACIÓN MOSTRADA POR EL BOTÓN 1: CÓMO CREAR UN CAMINO” y “3.3.2 INFORMACIÓN MOSTRADA POR EL BOTÓN 2: GUIADO”.

En este punto estamos hablando del contenido de botón “INFO GENERAL”, que lleva sobreimpreso un 3.

El botón “CONTACTO”, que lleva sobreimpreso un 4, como su propio nombre indica, sirve para contactar con el “Digital System Lab” y consultar las dudas que puedan plantear a los usuarios en relación con el sistema desarrollado.

El botón colocado en último lugar (“OTRAS APPS”), que lleva sobreimpreso un 5, informa sobre otras aplicaciones diseñadas por DSlab de la Universidad Autónoma de Madrid, tanto en la plataforma de iOS como en Android. En principio, estos dos apartados sobre información adicional al proyecto quedarán por el momento prácticamente vacíos de contenido, no tanto por no disponer de datos para cumplimentarlos, sino por dejar a propósito el espacio a disposición de la dirección del DSlab, laboratorio de investigación de la universidad situada en la Facultad de Ingeniería (EPS) de la Universidad Autónoma de Madrid (UAM) España. Lo contrario no parecería razonable, aquí y ahora.

INFORMACIÓN MOSTRADA POR EL BOTÓN 4: “CONTACTO”

- **Dirección de contacto:**

Escuela Politécnica Superior C-115
c/ Francisco Tomás y Valiente
Universidad Autónoma de Madrid
Ctra. Colmenar Km.15,
28039 Madrid, España.

- **Punto de Encuentro:**

TBD

- **Coordenadas de la Escuela Politécnica Superior:**

Latitud: ----- 40° 32' 51.23" N

Longitud: ----- 03° 41' 31.49" O

- **WEB:** http://arantxa.ii.uam.es/~euroform_dslab/



INFORMACIÓN MOSTRADA POR EL BOTÓN 4: “OTRAS APPs”

El DSLab es un laboratorio de investigación de la universidad situada en la Facultad de Ingeniería (EPS) de la Universidad Autónoma de Madrid (UAM) España. Principales líneas de I + D se centran en:

- Metodología de diseño de FPGA.
- Diseño de bajo consumo y de análisis térmico de los Circuitos Integrados.
- Auto-sincronización temporizada.
- Aplicaciones Industriales de FPGAs.
- Smartphones como componente electrónico estándar.
- Aplicación para Apple iOS y Android.
- Las tecnologías de asistencia para personas con discapacidades.

Además, el laboratorio promueve actividades académicas fuertemente relacionadas con la tecnología FPGA:

- Formación Profesional en Tecnología FPGA y técnicos a través de Seminarios EUROFORM Español Polo
- Organización de eventos técnicos y las conferencias sobre FPGAs
- Participación en la Junta Editorial de Revistas Científicas: ACM TRETs y JOLPE
- Los documentos técnicos y Ph.D. Tesis

http://arantxa.ii.uam.es/~euroform_dslab/android.htm

Anexo III Presupuesto

1) Ejecución Material

- Compra de ordenador personal (Software incluido)..... 3225 €
- Alquiler de impresora láser durante 6 meses 50 €
- Material de oficina 700 €
- Total de ejecución material..... 3975 €

2) Gastos generales

- 16 % sobre Ejecución Material 636 €

3) Beneficio Industrial

- 6 % sobre Ejecución Material 238.5 €

4) Honorarios Proyecto

- 600 horas a 20 € / hora x2..... 24000 €
- Gastos varios 10.077,26€

5) Material fungible

- Gastos de impresión..... 60 €
- Encuadernación 200 €

6) Subtotal del presupuesto

- Subtotal Presupuesto 38312,26 €

7) I.V.A. aplicable

- 16% Subtotal Presupuesto 6129.97 €

8) Total presupuesto

- Total Presupuesto 44442,23 €

Madrid, Septiembre de 2014

El Ingeniero Jefe de Proyecto

Fdo.: Andrés Romero Sánchez
Ingeniero de Telecomunicación

Justificación del Presupuesto

Costes de Hardware y Software:

- **2 PCs Mac Mini:** los nuevos vienen de serie con versiones actualizadas de Xcode.
- Se trata del equipo base para llevar a cabo las tareas de programación y creación, tanto de la música, como de las imágenes. Su coste en el mercado es unos 600 euros unidad, por lo que el gasto ascenderá a 1.200 euros.
- **Tableta Bq Edison:** está disponible en 2 versiones, una sólo con conexión Wi-Fi por 200 € (a adquirir) y otra con Wi-Fi y 3G por 250 € (no necesario).
- **Corel DRAW:** para realizar todo tipo de tareas de diseño, de dibujo y de retoque de imágenes, con un precio en torno a 725 euros.
- **Virtual DJ:** software que permite mezclar audio y video directamente desde el ordenador; se trata de un programa simple e intuitivo, con muchas características avanzadas para explorar y personalizar, cuyo precio está en torno a 300 euros.
- **iOS:** sistema operativo móvil de la empresa Apple, originalmente desarrollado para el iPhone (iPhone OS), siendo después usado en otros dispositivos como el iPod. Se adquiere por licencia cuyo valor es de unos 90 euros anuales.
- **2 Monitores de apoyo:**
- **2 Cables de Mac-HDMI:**
- **2 Ratones y 2 teclados:**

En consecuencia, la inversión económica en este apartado se situará en el entorno de los 3.225 euros.

Costes de Personal:

Recursos Humanos: sería necesario contratar a dos jóvenes ingenieros de telecomunicaciones, con dedicación plena y contrato temporal, modalidad de obra o servicio determinado, por ser de entre los cuatro tipos generales de contratos existentes (indefinido, temporal, formación-aprendizaje y prácticas) y de las distintas modalidades de temporal, lo más ajustado jurídicamente al trabajo a realizar. Seis meses de duración, con un salario bruto mensual de 2.000 euros, lo que supondría un coste inicial de 24.000 euros, a los que habría que añadir los gastos derivados de:

- Gestión de contratos de trabajo y nóminas (incluida la del IRPF), a razón de unos 70 euros mensuales, lo que supondría para todo el período un montante de 420 euros.
- Seguridad Social a cargo de la Empresa: 5.964 euros, por aplicación de un porcentaje del 24,85 (23,6% por contingencias comunes y 1,25% por accidente de trabajo y enfermedad profesional) sobre los 24.000 euros globales de salarios.
- Comida y transporte: 2.560 euros, a razón de 10 euros por 128 días hábiles y por dos personas.
- Formación en Prevención de Riesgos Laborales: realización de un curso de Prevención en Riesgos Laborales para Técnicos Audiovisuales, por un importe global aproximado de 300 euros, a razón de 150 euros por persona.
- Reconocimiento médico: unos 100 euros, a razón de 50 euros por persona.
- Indemnización por extinción: 733,26 euros, a razón de 66,66 euros, por 5,5 días y por dos personas.

Con la advertencia de que tan sólo hemos procedido a valorar los costes empresariales por gastos de personal, indicamos que hemos pasado por alto el análisis de las deducciones o retenciones a practicar directamente a los trabajadores por IRPF, seguridad social, etc.

En consecuencia, la inversión económica en este apartado alcanzaría a 34.077,26 euros.

Costes Generales de Oficina:

Calculamos que a este tipo de necesidades (impresión, encuadernación, fotocopia, material de oficina y a herramientas ofimáticas en general) debemos presupuestar en torno a los 1010 euros, a razón de 168.33 euros mes.

Total Gastos Generales del Proyecto:

En resumen, pues, los gastos generales del proyecto ascenderían a 38.052,26 euros, según el siguiente desglose:

RESUMEN ECONÓMICO DEL PROYECTO	
Costes de Hardware y Software	3.225,00 euros
Costes de Personal	34.077,26 euros
Costes Generales de Oficina	1010,00 euros
Total (sin iva)	38.312,26 euros

Tabla 4. Presupuesto Económico del proyecto

Es evidente que éste no sería el coste real de un producto desarrollado con las características del aquí descrito, sino que sería notablemente superior, pues faltan por valorar, entre otros, detalles tan importantes como la Dirección del Proyecto desde y en el DSLab o la utilización de las instalaciones y equipamiento del propio DSLab, además de otros tales como los preparativos de asistencia técnica y asesoramiento general, en aras de una posible promoción y venta.

CONTRATO TEMPORAL DE OBRA O SERVICIO DETERMINADO

FORMALIZACIÓN, DURACIÓN Y JORNADA

- Este contrato tiene por objeto la realización de obras o servicios con autonomía y sustantividad propias dentro de la actividad de la empresa y cuya ejecución, aunque limitada en el tiempo, es en principio de duración incierta, no pudiendo tener una duración superior a tres años.
- Los Convenios Colectivos podrán identificar aquellos trabajos o tareas con sustantividad propia dentro de la actividad normal de la empresa que puedan cubrirse con contratos de esta naturaleza.

Jornada y duración:

- Podrá concertarse a tiempo completo o a tiempo parcial.
- Será la del tiempo exigido para la realización de la obra o servicio. Si el contrato fijara una duración o un término, éstos deberán considerarse de carácter orientativo en función de lo establecido anteriormente, no pudiendo superar los tres años de duración, ampliable en doce meses más por Convenio Colectivo de ámbito sectorial estatal o en su defecto por convenios sectoriales de ámbito inferior. Transcurridos estos plazos, los trabajadores adquirirán la condición de trabajadores fijos de la empresa.

Formalización:

- El contrato deberá formalizarse siempre por escrito y deberá especificar con precisión y claridad el carácter de la contratación e identificar suficientemente la obra o el servicio que constituya su objeto, la duración del contrato, así como el trabajo a desarrollar.

EXTINCIÓN

- El contrato se extinguirá, previa denuncia de cualquiera de las partes, cuando finalice la obra o servicio objeto del contrato.
- Cuando la duración del contrato sea superior a un año, la parte que formule la denuncia está obligada a notificar a la otra la terminación del contrato con una antelación mínima de quince días. El incumplimiento por parte del empresario del plazo mencionado anteriormente le obligará al abono de una indemnización equivalente al salario correspondiente a los días en que dicho 54 plazo se haya incumplido.
- Ejecutada la obra o servicio, si no hubiera denuncia expresa y el trabajador continuara prestando sus servicios, el contrato se considerará prorrogado tácitamente por tiempo indefinido, salvo prueba en contrario que acredite la naturaleza temporal de la prestación.
- A la finalización del contrato, el trabajador tendrá derecho a recibir una indemnización, hasta el 31 de diciembre de 2013, de cuantía equivalente a la parte proporcional de la cantidad que resultaría de abonar diez días de salario por cada año de servicio. Esta cantidad se incrementará anualmente en un día, hasta el 1 de enero de 2015, en que quedara fijada en doce días, o la establecida, en su caso, en la normativa específica que sea de aplicación.

OTRAS CARACTERÍSTICAS

Se transforma en indefinido, salvo prueba en contrario que acredite la naturaleza temporal de la prestación:

- Por falta de forma escrita. En el supuesto de contrato a tiempo parcial, la falta de forma escrita determinará asimismo que el contrato se presuma celebrado a jornada completa, salvo prueba en contrario que acredite el carácter a tiempo parcial de los servicios.
- Por falta de alta en la Seguridad Social si hubiera transcurrido un período superior al período de prueba.

- Si llegado el término no se hubiera producido denuncia de alguna de las partes y se continuara realizando la prestación laboral.
- También se presumirán por tiempo indefinido los celebrados en fraude de ley.

NORMATIVA

- Artículo 15 del Estatuto de los Trabajadores, modificado por la Ley 35/2010, de 17 de septiembre.
- Ley 35/2010, de 17 de septiembre.
- R. D. 2720/98, de 18 de diciembre.
- Ley 3/2012, de 6 de julio.

Anexo IV Pliego de condiciones

Este documento contiene las condiciones legales que guiarán la realización, en este proyecto, de un “CARTOGRAFÍA ASISTIDA PARA PERSONAS CIEGAS, EN TELÉFONOS O TABLETAS APPLE”. En lo que sigue, se supondrá que el proyecto ha sido encargado por una empresa cliente a una empresa consultora con la finalidad de realizar dicho sistema. Dicha empresa ha debido desarrollar una línea de investigación con objeto de elaborar el proyecto. Esta línea de investigación, junto con el posterior desarrollo de los programas está amparada por las condiciones particulares del siguiente pliego.

Supuesto que la utilización industrial de los métodos recogidos en el presente proyecto ha sido decidida por parte de la empresa cliente o de otras, la obra a realizar se regulará por las siguientes:

Condiciones generales

1. La modalidad de contratación será el concurso. La adjudicación se hará, por tanto, a la proposición más favorable sin atender exclusivamente al valor económico, dependiendo de las mayores garantías ofrecidas. La empresa que somete el proyecto a concurso se reserva el derecho a declararlo desierto.
2. El montaje y mecanización completa de los equipos que intervengan será realizado totalmente por la empresa licitadora.
3. En la oferta, se hará constar el precio total por el que se compromete a realizar la obra y el tanto por ciento de baja que supone este precio en relación con un importe límite si este se hubiera fijado.
4. La obra se realizará bajo la dirección técnica de un Ingeniero Superior de Telecomunicación, auxiliado por el número de Ingenieros Técnicos y Programadores que se estime preciso para el desarrollo de la misma.
5. Aparte del Ingeniero Director, el contratista tendrá derecho a contratar al resto del personal, pudiendo ceder esta prerrogativa a favor del Ingeniero Director, quien no estará obligado a aceptarla.

6. El contratista tiene derecho a sacar copias a su costa de los planos, pliego de condiciones y presupuestos. El Ingeniero autor del proyecto autorizará con su firma las copias solicitadas por el contratista después de confrontarlas.

7. Se abonará al contratista la obra que realmente ejecute con sujeción al proyecto que sirvió de base para la contratación, a las modificaciones autorizadas por la superioridad o a las órdenes que con arreglo a sus facultades le hayan comunicado por escrito al Ingeniero Director de obras siempre que dicha obra se haya ajustado a los preceptos de los pliegos de condiciones, con arreglo a los cuales, se harán las modificaciones y la valoración de las diversas unidades sin que el importe total pueda exceder de los presupuestos aprobados. Por consiguiente, el número de unidades que se consignan en el proyecto o en el presupuesto, no podrá servirle de fundamento para entablar reclamaciones de ninguna clase, salvo en los casos de rescisión.

8. Tanto en las certificaciones de obras como en la liquidación final, se abonarán los trabajos realizados por el contratista a los precios de ejecución material que figuran en el presupuesto para cada unidad de la obra.

9. Si excepcionalmente se hubiera ejecutado algún trabajo que no se ajustase a las condiciones de la contrata pero que, sin embargo, es admisible a juicio del Ingeniero Director de obras, se dará conocimiento a la Dirección, proponiendo a la vez la rebaja de precios que el Ingeniero estime justa y si la Dirección resolviera aceptar la obra, quedará el contratista obligado a conformarse con la rebaja acordada.

10. Cuando se juzgue necesario emplear materiales o ejecutar obras que no figuren en el presupuesto de la contrata, se evaluará su importe a los precios asignados a otras obras o materiales análogos si los hubiere y cuando no, se discutirán entre el Ingeniero Director y el contratista, sometiéndolos a la aprobación de la Dirección. Los nuevos precios convenidos por uno u otro procedimiento, se sujetarán siempre al establecido en el punto anterior.

11. Cuando el contratista, con autorización del Ingeniero Director de obras, emplee materiales de calidad más elevada o de mayores dimensiones de lo

estipulado en el proyecto, o sustituya una clase de fabricación por otra que tenga asignado mayor precio o ejecute con mayores dimensiones cualquier otra parte de las obras, o en general, introduzca en ellas cualquier modificación que sea beneficiosa a juicio del Ingeniero Director de obras, no tendrá derecho sin embargo, sino a lo que le correspondería si hubiera realizado la obra con estricta sujeción a lo proyectado y contratado.

12. Las cantidades calculadas para obras accesorias, aunque figuren por partida alzada en el presupuesto final (general), no serán abonadas sino a los precios de la contrata, según las condiciones de la misma y los proyectos particulares que para ellas se formen, o en su defecto, por lo que resulte de su medición final.

13. El contratista queda obligado a abonar al Ingeniero autor del proyecto y director de obras así como a los Ingenieros Técnicos, el importe de sus respectivos honorarios facultativos por formación del proyecto, dirección técnica y administración en su caso, con arreglo a las tarifas y honorarios vigentes.

14. Concluida la ejecución de la obra, será reconocida por el Ingeniero Director que a tal efecto designe la empresa.

15. La garantía definitiva será del 4% del presupuesto y la provisional del 2%.

16. La forma de pago será por certificaciones mensuales de la obra ejecutada, de acuerdo con los precios del presupuesto, deducida la baja si la hubiera.

17. La fecha de comienzo de las obras será a partir de los 15 días naturales del replanteo oficial de las mismas y la definitiva, al año de haber ejecutado la provisional, procediéndose si no existe reclamación alguna, a la reclamación de la fianza.

18. Si el contratista al efectuar el replanteo, observase algún error en el proyecto, deberá comunicarlo en el plazo de quince días al Ingeniero Director de obras, pues transcurrido ese plazo será responsable de la exactitud del proyecto.

19. El contratista está obligado a designar una persona responsable que se entenderá con el Ingeniero Director de obras, o con el delegado que éste designe, para todo relacionado con ella. Al ser el Ingeniero Director de obras el que interpreta el proyecto, el contratista deberá consultarle cualquier duda que surja en su realización.

20. Durante la realización de la obra, se girarán visitas de inspección por personal facultativo de la empresa cliente, para hacer las comprobaciones que se crean oportunas. Es obligación del contratista, la conservación de la obra ya ejecutada hasta la recepción de la misma, por lo que el deterioro parcial o total de ella, aunque sea por agentes atmosféricos u otras causas, deberá ser reparado o reconstruido por su cuenta.

21. El contratista, deberá realizar la obra en el plazo mencionado a partir de la fecha del contrato, incurriendo en multa, por retraso de la ejecución siempre que éste no sea debido a causas de fuerza mayor. A la terminación de la obra, se hará una recepción provisional previo reconocimiento y examen por la dirección técnica, el depositario de efectos, el interventor y el jefe de servicio o un representante, estampando su conformidad el contratista.

22. Hecha la recepción provisional, se certificará al contratista el resto de la obra, reservándose la administración el importe de los gastos de conservación de la misma hasta su recepción definitiva y la fianza durante el tiempo señalado como plazo de garantía. La recepción definitiva se hará en las mismas condiciones que la provisional, extendiéndose el acta correspondiente. El Director Técnico propondrá a la Junta Económica la devolución de la fianza al contratista de acuerdo con las condiciones económicas legales establecidas.

23. Las tarifas para la determinación de honorarios, reguladas por orden de la Presidencia del Gobierno el 19 de Octubre de 1961, se aplicarán sobre el denominado en la actualidad "Presupuesto de Ejecución de Contrata" y anteriormente llamado "Presupuesto de Ejecución Material" que hoy designa otro concepto.

Condiciones particulares

La empresa consultora, que ha desarrollado el presente proyecto, lo entregará a la empresa cliente bajo las condiciones generales ya formuladas, debiendo añadirse las siguientes condiciones particulares:

1. La propiedad intelectual de los procesos descritos y analizados en el presente trabajo, pertenece por entero a la empresa consultora representada por el Ingeniero Director del Proyecto.
2. La empresa consultora se reserva el derecho a la utilización total o parcial de los resultados de la investigación realizada para desarrollar el siguiente proyecto, bien para su publicación o bien para su uso en trabajos o proyectos posteriores, para la misma empresa cliente o para otra.
3. Cualquier tipo de reproducción aparte de las reseñadas en las condiciones generales, bien sea para uso particular de la empresa cliente, o para cualquier otra aplicación, contará con autorización expresa y por escrito del Ingeniero Director del Proyecto, que actuará en representación de la empresa consultora.
4. En la autorización se ha de hacer constar la aplicación a que se destinan sus reproducciones así como su cantidad.
5. En todas las reproducciones se indicará su procedencia, explicitando el nombre del proyecto, nombre del Ingeniero Director y de la empresa consultora.
6. Si el proyecto pasa la etapa de desarrollo, cualquier modificación que se realice sobre él, deberá ser notificada al Ingeniero Director del Proyecto y a criterio de éste, la empresa consultora decidirá aceptar o no la modificación propuesta.
7. Si la modificación se acepta, la empresa consultora se hará responsable al mismo nivel que el proyecto inicial del que resulta el añadirla.
8. Si la modificación no es aceptada, por el contrario, la empresa consultora declinará toda responsabilidad que se derive de la aplicación o influencia de la misma.

9. Si la empresa cliente decide desarrollar industrialmente uno o varios productos en los que resulte parcial o totalmente aplicable el estudio de este proyecto, deberá comunicarlo a la empresa consultora.
10. La empresa consultora no se responsabiliza de los efectos laterales que se puedan producir en el momento en que se utilice la herramienta objeto del presente proyecto para la realización de otras aplicaciones.
11. La empresa consultora tendrá prioridad respecto a otras en la elaboración de los proyectos auxiliares que fuese necesario desarrollar para dicha aplicación industrial, siempre que no haga explícita renuncia a este hecho. En este caso, deberá autorizar expresamente los proyectos presentados por otros.
12. El Ingeniero Director del presente proyecto, será el responsable de la dirección de la aplicación industrial siempre que la empresa consultora lo estime oportuno. En caso contrario, la persona designada deberá contar con la autorización del mismo, quien delegará en él las responsabilidades que ostente.