



TRABAJO DE FIN DE GRADO

Sistema de guiado en exteriores para personas con necesidades especiales mediante Smartwatches

Por:

Roberto de Castro Muñoz

Tutor: Javier Gómez Escribano
Ponente: Germán Montoro Manrique

GRADO EN INGENIERÍA INFORMÁTICA
Departamento de Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid

Julio 2016

Resumen

Sistema de guiado en exteriores para personas con necesidades especiales mediante Smartwatches

por Roberto de Castro Muñoz

El día a día de una persona con necesidades especiales está plagado de desafíos. Tareas como prepararse el desayuno o ir a la compra son tareas que para la mayoría de personas no suponen demasiado esfuerzo pero para ciertas personas pueden suponer verdaderos problemas. Desde hace años se están realizando proyectos para conseguir una integración de este tipo de personas en la sociedad actual, de manera que destaquen lo menos posible en el entorno. Sin embargo, las ayudas actuales no son suficientes y aún queda mucho trabajo por hacer. Este proyecto se centra en concreto en la autonomía de una persona con discapacidad intelectual de cara a realizar un desplazamiento. El objetivo es que la persona sea capaz de viajar de un origen a un destino de la mejor manera posible, con la menor ayuda humana posible y con el mayor aprendizaje posible.

Estos desplazamientos pueden producirse en entornos conocidos por el usuario, en los cuales puede usar su experiencia previa para realizar el trayecto, o bien se realizan en lugares desconocidos para la persona, en cuyo caso deberá de usar alguna herramienta de orientación para lograr el objetivo. Desde **AMILab** se ha diseñado una herramienta llamada **Assist-Out-Android-Wear** que consiste en un sistema de ayuda para este tipo de personas, permitiéndoles realizar un trayecto de manera sencilla y que a su vez vayan interiorizando el proceso. El trabajo ha sido diseñado en unos dispositivos muy recientes e innovadores, los *Smartwatches*. Éstos dispositivos proveen una gran facilidad de uso y ofrecen la ventaja de ir continuamente pegados al cuerpo además de una gran accesibilidad.

En este documento aborda en primera instancia el estado actual de la discapacidad mental y su papel dentro de la tecnología de asistencia, centrándose sobre todo en el uso de *Smartwatches*. Se habla también de otros tipos de proyectos de asistencia existentes actualmente y las ideas que aportan. Posteriormente se presenta el diseño y justificación del proyecto así como el desarrollo de la implementación. Por último se analizan las pruebas realizadas y se muestran las conclusiones obtenidas.

Palabras clave - Navegación en exteriores, Smartwatches, Discapacidad intelectual, Cálculo de direccionamiento

Abstract

SmartWatch outdoor guidance system for people with special needs

by Roberto de Castro Muñoz

The day-to-day life of a person with special needs has plenty of challenges. Tasks like preparing breakfast or going shopping are not hard for most people but for these people an extra effort is needed. In recent years a lot of projects have been realised in order to integrate these people into society, however there is a lot of work to do yet. This project is focused mainly on people with mental disabilities' autonomy when they getting around. The objective is that the user can get from an origin to a destination using the best form, the least human support and the maximum learning.

The journeys can be made in a known environment, using one's previous experiences or they can be to new places, where the user needs an orientation support system. The tool called AssisT-Out-Android-Wear has been developed in AMILab, which is for people with mental disabilities, providing a guidance system that helps them to make these journeys easily and at the same time learning the way. The project was implemented using very recent and innovative technology; Smartwatches. These devices have a very high grade of usability and accesibility and they are always worn.

This document contains, first of all, the current state of mental disability and its role inside support technology, especially using Smartwatches. Other projects related to this theme and their ideas also appear. Following that is the Design and Implementation section and at the end there is the Test Analysis and the Conclusions.

Key words - Outdoor guidance, Smartwatches, Mental disability, Route calculation

Agradecimientos

En primer lugar quiero agradecer a Germán Montoro Manrique y a Javier Gómez Escribano por haberme dado la oportunidad de realizar este proyecto y haberme guiado durante él, así como a mis compañeros de laboratorio AMILab Juan Carlos y Javier. Agradecer especialmente a Marta y a su marido por haber colaborado conmigo en las pruebas de este proyecto.

A Samuel y a Víctor por todas las risas y buenos ratos jugando, o simplemente haciendo tonterías. A Cristina y Guido por tenerles siempre ahí cuando les he necesitado. A Irene por todos los momentos inolvidables que hemos vivido. A Álvaro por haberme sacado de los malos momentos y por haberme enseñado tanto sobre la vida. Y sobretodo a mi abuela y mi madre a las que debo todo lo bueno que tengo.

”Siempre hacia delante”

Glosario

- **TFG:** Trabajo de Fin de Grado
- **CIF:** Clasificación Internacional del Funcionamiento de la Discapacidad y de la Salud
- **OMS:** Organización Mundial de la Salud
- **Etiología:** Parte de la medicina que estudia el origen o las causas de las enfermedades
- **Hipotonía:** Disminución de la tensión o del tono muscular
- **CI:** Coeficiente Intelectual
- **Micro-Prompting:** Dispositivos que proveen navegación paso a paso para realizar tareas
- **GPS:** Global Positioning System
- **USB:** Universal Serial Bus

Índice general

| | |
|---|-----------|
| Resumen | III |
| Abstract | v |
| Glosario de términos | IX |
| Índice | XII |
| 1. Introducción | 1 |
| 1.1. Motivación | 1 |
| 1.2. Objetivo | 2 |
| 2. Personas con necesidades especiales | 3 |
| 2.1. Introducción | 3 |
| 2.2. Discapacidad intelectual | 3 |
| 2.2.1. Causas | 4 |
| 2.2.2. Grados de discapacidad mental | 5 |
| 3. Estado del arte | 7 |
| 4. Diseño | 11 |
| 4.1. Introducción | 11 |
| 4.2. ¿Por qué <i>SmartWatches</i> ? | 11 |
| 4.2.1. Concepto <i>pervasive</i> | 11 |
| 4.2.2. Dispositivos comerciales y no comerciales | 12 |
| 4.2.2.1. ¿Qué tipo de dispositivo usar? | 12 |
| 4.2.3. Acceptance o aceptación: Componente normalizador | 13 |
| 4.3. Requisitos | 13 |
| 4.3.1. Requisitos Funcionales | 13 |
| 4.3.2. Requisitos No Funcionales | 14 |
| 4.4. AssisT-Out y AssisT-Out-Android-Wear | 15 |
| 4.5. Representación de ruta | 17 |
| 4.5.1. Cálculo del ángulo entre dos puntos de la ruta | 17 |
| 4.5.2. Cálculo del ángulo de la orientación actual | 19 |
| 4.5.2.1. Primer cuadrante | 20 |

| | |
|--|-----------|
| 4.5.2.2. Resto de circunferencia | 21 |
| 4.5.3. Cálculo de dirección final | 22 |
| 5. Desarrollo | 25 |
| 5.1. Introducción | 25 |
| 5.2. Módulo del Smartphone | 25 |
| 5.3. Módulo del Smartwatch | 26 |
| 5.3.1. Selección de destino | 26 |
| 5.3.2. Sistema de guiado | 27 |
| 5.3.2.1. Obtención de la orientación actual | 27 |
| 5.3.2.2. Cálculo del ángulo de dirección | 29 |
| 5.3.2.3. Cálculo del ángulo final | 31 |
| 5.3.2.4. Cálculo y representación del progreso | 32 |
| 5.3.2.5. Diagrama de clases | 33 |
| 6. Pruebas | 35 |
| 6.1. Pruebas de componentes | 35 |
| 6.1.1. GPS | 35 |
| 6.1.2. Brújula | 36 |
| 6.1.3. Obtención de coordenadas | 36 |
| 6.1.4. Enrutado | 36 |
| 6.2. Pruebas con usuarios | 36 |
| 6.2.1. Usuarios sin discapacidad | 36 |
| 6.2.2. Usuario con discapacidad | 37 |
| 7. Conclusiones | 39 |
| 7.1. Conclusiones | 39 |
| 7.2. Trabajo Futuro | 40 |
| Bibliografía | 41 |

Índice de figuras

| | |
|---|----|
| 4.1. Diagrama general. | 15 |
| 4.2. Ejemplo ruta general. | 16 |
| 4.3. Rellenado flecha | 16 |
| 4.4. Cálculo dirección primer cuadrante. | 18 |
| 4.5. Cálculo dirección segundo cuadrante. | 18 |
| 4.6. Cálculo dirección tercer cuadrante. | 19 |
| 4.7. Cálculo dirección cuarto cuadrante. | 19 |
| 4.8. Conversión Acimut-Cartesiano. | 20 |
| 4.9. Ejemplo 90 grados | 22 |
| 4.10. Ejemplo 180 grados | 22 |
| 4.11. Ejemplo 270 grados | 23 |
| 5.1. Ejemplo menú casa | 26 |
| 5.2. Ejemplo menú escuela | 27 |
| 5.3. Ejemplo menú metro | 27 |
| 5.4. Imagen próximo punto de la ruta | 30 |
| 5.5. Imagen final ruta | 31 |
| 5.6. Diagrama de clases | 34 |
| 6.1. Ruta calentamiento | 38 |
| 6.2. Ruta de prueba del sistema | 38 |

Índice de tablas

| | |
|---|----|
| 4.1. Tabla de cuadrantes. | 18 |
| 4.2. Casos de prueba en el primer cuadrante. | 20 |
| 4.3. Casos de prueba en el segundo cuadrante. | 21 |
| 4.4. Casos de prueba en el tercer cuadrante. | 21 |
| 4.5. Casos de prueba en el cuarto cuadrante. | 21 |



1 Introducción

1.1. Motivación

La integración de las personas con alguna discapacidad es un objetivo que afortunadamente se persigue cada vez más a lo largo de los años. Se ven numerosos proyectos llevados a cabo por colectivos y asociaciones para este sector de la población, entre los que se encuentran numerosas campañas de concienciación o ayudas del estado. Aun así, queda mucho trabajo por hacer para que todo ser humano pueda disfrutar de una vida cómoda independientemente de sus capacidades. La sociedad actual avanza sin pausa hacia una sociedad cada vez más tecnológica, conectada e informatizada y como profesionales del sector es nuestro deber poner esta tecnología al servicio de quien lo necesita.

Este trabajo en concreto se centra en las personas con discapacidad intelectual y en su dificultad a la hora de orientarse o conseguir realizar un trayecto con éxito. Por ello se ha desarrollado una nueva versión de la aplicación **Assist-Out** para que sea funcional en los nuevos relojes inteligentes (*Smartwatches*) respetando los criterios que dicha aplicación ya traía consigo como es el uso de una web para almacenar los datos de la ruta o la forma en que se obtienen los datos de dicha ruta. Esta idea surgió en base a experiencias en proyectos previos en el **AMILab**, donde se hizo la sugerencia de realizar la nueva versión en una plataforma emergente y más manejable.

1.2. Objetivo

El proyecto **AssisT-Out-Android-Wear**, desarrollado en el Laboratorio de Inteligencia Ambiental (**AMILab**) de la Escuela Politécnica Superior de la Universidad Autónoma de Madrid, pretende ver las posibilidades que ofrecería la herramienta **AssisT-Out** en una plataforma *Smartwatch*. Para este fin se han establecido los siguientes objetivos:

1. **Fiabilidad de guiado:** Para el cálculo de la ruta debe usarse un sistema que proporcione una o más rutas que no sólo sean correctas, si no también eficientes y que tengan en cuenta factores que afectan al enrutado, como pueden ser obras en el terreno o el tráfico.
2. **Claridad del guiado:** El sistema de guiado debe ser lo más claro posible para evitar confusiones a la hora de escoger un camino o para evitar proporcionar un direccionamiento erróneo, así como una representación lógica de la distancia hacia el destino.
3. **Tiempo de respuesta:** La aplicación debe responder en un tiempo lo suficientemente corto como para que el usuario sienta que el ritmo de acciones es fluido y que no se estanca en ningún momento.
4. **Facilidad de uso:** El sistema debe ser lo más intuitivo y trivial posible y ajustado a las capacidades del usuario, de manera que cada acción tenga un sentido y objetivo totalmente claro y definido.
5. **Robustez:** Se requiere un sistema que minimice los errores.
6. **No dependencia del sistema:** La idea subyacente es que el usuario vaya reduciendo su dependencia con el sistema, de manera que vaya aprendiendo conforme utiliza la aplicación y solo tenga que usarla para recordar alguna ruta en algún momento puntual.



2 Personas con necesidades especiales

2.1. Introducción

La Clasificación Internacional del Funcionamiento, de la Discapacidad y de la Salud (CIF), adoptada como marco conceptual, define la discapacidad como un término genérico que engloba deficiencias, limitaciones de actividad y restricciones para la participación. La discapacidad denota los aspectos negativos de la interacción entre personas con un problema de salud (como parálisis cerebral, síndrome de Down o depresión) y factores personales y ambientales como actitudes negativas, transporte y edificios públicos inaccesibles, y falta de apoyo social.

OMS

2.2. Discapacidad intelectual

¿Qué es la discapacidad intelectual: una enfermedad, un conjunto de ellas, o un déficit asociado una enfermedad? La discapacidad intelectual es un conjunto de enfermedades y trastornos de origen genético y ambiental que provoca un déficit de funciones cerebrales de una magnitud tal, que produce interferencias significativas en el funcionamiento normal de un individuo. A diferencia por ejemplo de otros problemas mentales como la demencia, la discapacidad intelectual es una deficiencia que se puede adquirir incluso antes del nacimiento.

2.2.1. Causas

La etiología se fija mayoritariamente por varios factores y la mezcla de esos factores condiciona la diversidad de manifestaciones clínicas. Se estima que en entre un 30 % y un 40 % de los casos examinados a nivel de consulta externa, la etiología específica no se puede determinar a pesar de someterse a complejas evaluaciones. En el resto de los casos, los factores se comportan de la siguiente manera: entre un 15 % y un 20 % se encuentran influencias del entorno y trastornos mentales, en un 30 % alteraciones tempranas en el desarrollo embrionario, en un 50 % trastornos somáticos de la niñez y en otro 50 % se detectan factores hereditarios o genéticos [1].

Actualmente varios especialistas afirman la falta de consistencia y de un marco conceptual común para clasificar los diferentes tipos de necesidades que presentan las personas con discapacidad intelectual, sin embargo, en el marco clínico existen ciertas pautas para diagnosticarlo:

1. **Limitaciones significativas en el funcionamiento intelectual:** Nivel intelectual significativamente inferior a la media (inferior a 69-75 según los criterios).
2. **Limitaciones significativas en la conducta adaptativa:** Comunicación, Autocuidado, Vida en el Hogar, Habilidades Sociales, Uso de la Comunidad, Autodirección, Salud y Seguridad, Habilidades Académicas Funcionales, Ocio, y Trabajo [2].
3. **Rasgos físicos característicos:** Hipotonía muscular ¹, estatura baja, cráneo ancho y redondeado, pliegues de epicanto en la esquina interna del ojo y cuello corto entre numerosas otras [3].
4. **Edad de inicio:** Comienzo antes de los 18 años (Luckasson et al., 2002/2004).

¹Disminución de la tensión o del tono muscular

2.2.2. Grados de discapacidad mental

Las distintas clasificaciones existentes se basan única y exclusivamente en el grado de inteligencia, representado por el cociente intelectual (CI) obtenido mediante pruebas de inteligencia aplicadas al sujeto. Lo fundamental no recae en categorizar al sujeto en base a su CI, si no en sus puntos fuertes dentro de su grupo (Organización Mundial de la Salud; 2004 en Espacio Logopédico; 2007).

A continuación se detallan las distintas categorías [4]:

- **Discapacidad Intelectual límite** [5]: Coeficiente Intelectual entre 68-85. Este grupo es difícil de catalogar como discapacitado mental puesto que sólo muestran retraso en el aprendizaje o alguna dificultad concreta en el mismo.
- **Discapacidad Intelectual ligera:** Coeficiente Intelectual entre 50-69. Presentan un retraso leve en las áreas perceptivas y motoras aunque son capaces de socializar y comunicarse y tienen capacidad de adaptación e integración en el mundo laboral.
- **Discapacidad Intelectual moderada o media:** Coeficiente Intelectual entre 35-49. Presentan numerosas dificultades en la expresión oral aunque pueden adquirir hábitos de autonomía personal y social.
- **Discapacidad Intelectual severa:** Coeficiente Intelectual entre 20-34. Suelen presentar un notable deterioro psicomotor y necesitan de ayuda diaria debido a que su nivel de autonomía es muy pobre.
- **Discapacidad Intelectual profunda:** Coeficiente Intelectual inferior a 20. Presentan un grave deterioro en los aspectos sensorio-motrices y de comunicación con el medio.

¹Disminución de la tensión o del tono muscular



3 Estado del arte

En esta sección se muestran algunos proyectos desarrollados en los últimos años que pertenecen al ámbito tecnológico del que trata este TFG.

Desde casi su nacimiento, los *Wearables* se han usado como sistema de medición de datos, ya sean datos biométricos, estilos de vida, comportamientos o mejora de la salud [6]. Gillespie [7] estudió los productos que existían para personas con discapacidades cognitivas y su relación con su clasificación internacional de capacidades, discapacidades y salud [8], dándose cuenta de que los problemas de atención se pueden combatir con tecnologías de alarmas, el control emocional con tecnologías de distracción, sistemas de micro-prompting² proporcionan organización y planificación, tecnologías de almacenamiento y reproducción se usan para problemas de memoria y los recordatorios para control del tiempo.

Debido a la masificación del uso de dispositivos como *Smartphones* o *Tablets*, han sido numerosos los proyectos desarrollados para estos dispositivos, sin embargo Lancioni [9] señala que es fundamental desarrollar aplicaciones para personas con problemas cognitivos antes de que esta tecnología se extienda y se haga famosa.

Los *Smartwatches* son dispositivos portátiles, para las muñecas, que ofrecen características parecidas a los *Smartphones*. La característica más importante es su equipo de sensores: Acelerómetro, medidor de pulso, GPS, sensor de luz, brújula, o Wi-Fi, además de interacciones mediante pantalla táctil, sistema de vibración o micrófono. Son éstas características las que hacen de los *Smartwatches* una plataforma viable en el desarrollo de investigaciones para personas con problemas cognitivos.

En 2014 Kearns (Kearns et al., 2013) desarrolló su propio *Smartwatch* integrado en una casa inteligente, la cual ayudaba a personas con trastornos cognitivos en su día

a día además de proveer de un sistema de planificación y recordatorio de tareas. Los investigadores concluyeron que el aspecto más importante del sistema fue el sistema recordatorio, alertando del consumo de medicamentos puntualmente cada día.

Bieber (Bieber et al., 2012) usó el *Smartwatch* como set de sensores para aplicaciones de asistencia en entornos vivos, centrándose en la capacidad de estos dispositivos para ser un sensor permanente en nuestra propia muñeca.

El **AMILab** también está incluido en este grupo diseñando un sistema que proporciona ayuda a las personas con *desorden del espectro autista*³ mediante las mediciones de los sensores del *Smartwatch* sin necesidad de ayuda externa u otros dispositivos [10]. De este conjunto de proyectos se destaca la capacidad de los *Smartwatches* para proporcionar ayuda en situaciones que requieren auto-control, como pueden ser estados de nerviosismo, fobias o situaciones socialmente estresantes. Independientemente de la situación, el *Smartwatch* está siempre midiendo las emociones del sujeto en su muñeca y no sólo eso, los *Smartwatches* pueden detectar estas situaciones y registrarlas así como proporcionar medidas para evitarlas.

Por otro lado existen casos en los que se aplican las tecnologías emergentes en sistemas de direccionamiento. En 2015 se implementó un sistema de guiado en *Smartphone* que utilizaba puntos emblemáticos o significativos del lugar (monumentos, edificios religiosos o culturalmente relevantes) como puntos de referencia para el guiado [11]. Los sistemas basados en puntos de referencia llevan siendo utilizados desde hace muchos años y utilizan estos puntos de referencia como localizaciones geográficas que ayudan al usuario a identificar el camino actual así como a orientarse. A la hora de establecer estos puntos de referencia se necesitan tener en cuenta aspectos como su aspecto visual, estructural y dificultad semántica. Estos tres atributos conforman la importancia de un punto de referencia, sin embargo, esta importancia no es intrínseca del propio punto, si no que se le otorga en función del entorno. Por ejemplo, el color de un edificio no es relevante a no ser que destaque frente al resto de edificios de alrededor. Por supuesto este criterio es totalmente subjetivo y varía en función de cada persona, dejando así un amplio abanico de opiniones acerca de los puntos de referencia de un área en concreto.

David Dobbstein [12] diseñó una aplicación en *Smartwatches* que da una vuelta de tuerca al sistema de direccionamiento, guiando al usuario a través de un sistema de vibración, lo que origina que el usuario no tenga que mirar a la pantalla y pueda centrarse más en disfrutar del trayecto.

Actualmente existen también proyectos para personas con problemas de movilidad como AXSMAP[13] o handimap[14]. AXSMAP fue creada por un hombre minusválido que se encontraba cada día con el problema de saber si el sitio al que se dirigía poseía medidas de ayuda para personas con problemas de movilidad. Por ello creó la aplicación AXSMAP con el objetivo de que la gente colaborase enviando feedback informando de en qué manera estaban preparados los lugares que dicha gente visitaba para con las

personas con movilidad reducida.

²Dispositivos que proveen navegación paso a paso para realizar tareas

³Grupo de discapacidades del desarrollo que pueden causar problemas significativos de socialización, comunicación y conducta.



4 Diseño

4.1. Introducción

Este trabajo se engloba dentro de las líneas de trabajo actuales del **AMILab** y más concretamente dentro del apartado de sistema de asistencia en exteriores para personas con necesidades especiales. El sistema previo se desarrolló en la plataforma de *Smartphones* y para esta nueva aplicación se ha decidido realizarlo en la plataforma de *Smartwatches*.

4.2. ¿Por qué *SmartWatches*?

4.2.1. Concepto *pervasive*

Los *Smartwatches* son dispositivos *wearables*, es decir, el usuario siempre lo lleva encima y pegado al cuerpo. Además, para realizar cualquier acción que requiera su uso, el usuario simplemente tiene que girar la muñeca, lo cual facilita significativamente su utilización. Por ejemplo, a la hora de consultar una notificación, en el caso de tener un *Smartphone* habría que sacar éste del bolsillo, bolso o lugar donde el usuario lo guardase para poder consultarla. En el caso de un *Smartwatch*, el usuario simplemente mirando a su muñeca podría consultar dicha información y además en cualquier lugar.

4.2.2. Dispositivos comerciales y no comerciales

A la hora de hablar de *Smartwatches* es necesario distinguirlos en función si se comercializan en el mercado con propósito general o si son diseñados en algún laboratorio/empresa con un propósito específico, generalmente de investigación.

Los *Smartwatches* no comerciales son dispositivos diseñados, desarrollados y utilizados por empresas para investigaciones o estudios. Estos dispositivos suelen disponer de características únicas que los diferencian del resto de *Smartwatches*, como por ejemplo la forma, tamaño, el tipo de sensores o su número. Estas características están pensadas para otorgar el máximo rendimiento en la investigación para la cual han sido creados además de que no suelen prestar atención al componente visual ni al diseño.

Sin embargo, los *Smartwatches* comerciales son dispositivos creados por una empresa con la idea de conseguir un uso *masivo* de estos por la sociedad. Como cualquier producto del mercado, poseen unas características concretas que lo definen, pero al ser construido en grandes cantidades se pueden conseguir con muchísima más facilidad en el mercado.

4.2.2.1. ¿Qué tipo de dispositivo usar?

Lo que se desea es que el usuario que utilice la aplicación de *Assist-Out-Android-Wear* destaque lo menos posible en la sociedad actual, por lo tanto, se le debe proporcionar un dispositivo que esté lo más aceptado por la comunidad posible. Debido a esto se ha decidido usar un *Smartwatch* comercial, puesto que usando este tipo de dispositivo es más probable que las personas que se encuentren alrededor del usuario vean como normal su uso, lo cual se traduce en un *componente normalizador*.

Este hecho queda más explicado en estos dos casos:

En un primer caso el usuario final utiliza un *Smartwatch* no comercial y lo que generalmente ocurre es que o bien el propio usuario no lo acepta por ser diferente al resto de dispositivos, o bien es la sociedad la que se percata de ello, haciendo sentir al usuario señalado o diferente a los demás.

En el otro caso estaría el usuario que usa un *Smartwatch* comercial, y al ser en apariencia y funcionalidad igual o prácticamente igual al resto de dispositivos en el mercado, el usuario no se sentiría desplazado y la sociedad vería su uso como algo normal. Es este último caso el que se desea para el usuario final de la aplicación.

4.2.3. Acceptance o aceptación: Componente normalizador

La *normalización* se puede definir como la aceptación natural de un dispositivo en la vida de un usuario. La aceptación no sólo se aplica al usuario sino también al entorno y no sólo se ve definida por su componente comercial, sino también por su usabilidad y mantenibilidad. Por ejemplo, un dispositivo que fuese difícil de actualizar o de conseguirle un recambio sería un producto con baja aceptación, y como tal, produciría un componente *estigmatizador*.

Uno de los aspectos más importantes a la hora de realizar aplicaciones para personas con necesidades especiales es el *componente normalizador*, que queda definido como en qué medida la asistencia que se le proporciona a un usuario lo diferencia de los demás. Teniendo esto en cuenta se ha decidido que un *Smartwatch* aporta una gran *componente normalizador* puesto que el usuario al utilizarlo está usando un dispositivo que llevan muchas personas, por lo que el resto de la gente verá a una persona que mira a un reloj, un acto completamente natural.

4.3. Requisitos

4.3.1. Requisitos Funcionales

RF 1. Sistema de obtención de posición actual: La aplicación deberá ser capaz de obtener la localización actual del usuario ya sea mediante el sistema *GPS* en un tiempo máximo de 10 segundos en condiciones normales.

RF 2. Sistema de cálculo de ruta y parseado en pasos: Se deberá proporcionar un mecanismo de cálculo de la ruta a realizar y una segmentación de los mismos en pasos en un tiempo estimado de menos de 10 segundos.

RF 3. Sistema de recuento de los pasos realizados: Se creará un sistema de trackeado de pasos para poder obtener información de las rutas que se van realizando.

RF 4. Sistema de cálculo y representación de dirección de la flecha: La aplicación tendrá un sistema de representación mediante una flecha y un sistema de cálculo del ángulo de direccionamiento.

RF 5. Sistema de cálculo y representación de la distancia al siguiente punto: La aplicación además agregará al sistema anterior un sistema de cálculo y representación de la distancia restante hasta el siguiente punto.

RF 6. Sistema de selección de destinos: El sistema deberá proporcionar un sistema de elección de destino para el usuario que le permita disponer de varias opciones a la hora de querer fijar su trayecto.

RF 7. Requisitos de dispositivos: Para poder usar esta aplicación es necesario un *Smartphone* con sistema de GPS,Bluetooth y que soporte como mínimo la API 21 de Android (Android 5.0). De igual manera se necesita un *Smartwatch* con sistema de brújula.

4.3.2. Requisitos No Funcionales

RNF 1. Sistema de suavizado para la flecha: El dispositivo de brújula de Android es muy sensible al movimiento, llegando a estar variando en varios grados aunque el dispositivo esté prácticamente quieto, por lo tanto es conveniente diseñar un sistema de suavizado para proporcionar un direccionamiento más claro.

RNF 2. Sistema de personalización de colores para personas con daltonismo: Aprovechando que se está realizando una aplicación para personas con necesidades especiales, no está de más pensar también en las personas con daltonismo, de manera que el código de colores elegido no les suponga ningún inconveniente.

RNF 3. Conversión de aplicación a stand-alone: Como en un principio se pensó y de cara al futuro próximo, sería una gran ventaja el poder usar la aplicación sin necesidad de la ayuda de un *Smartphone*. Por lo tanto, en cuanto la tecnología lo permita, la aplicación se convertirá a stand-alone.

RF 4. Configuración de destinos: El sistema deberá proporcionar un sistema de configuración de destinos de manera que un usuario disponga de un conjunto de trayectos programable y alojado en un servidor.

4.4. AssisT-Out y AssisT-Out-Android-Wear

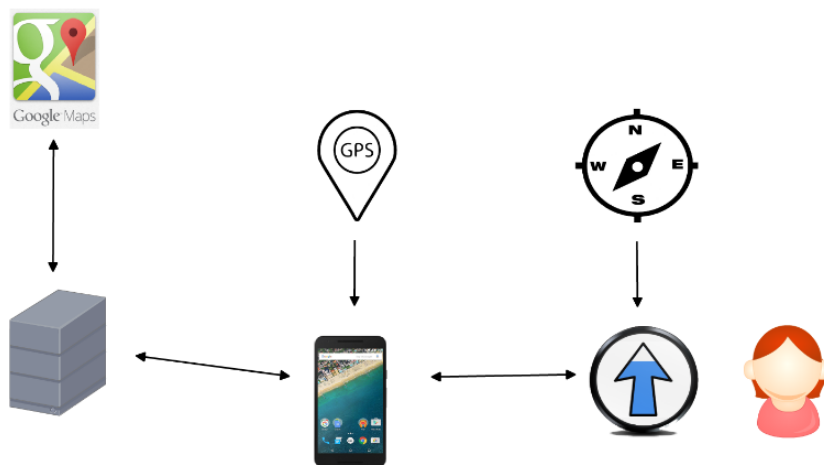


Figura 4.1. Diagrama general de la aplicación.

AssisT-Out es una aplicación de navegación orientada al uso de imágenes, las cuales van indicando el recorrido. Los puntos de la ruta se dividen en "puntos de decisión", que consisten en acciones como girar a derecha o izquierda. Esto unido al uso de una barra de progreso entre puntos permite al usuario realizar el trayecto entre el origen y destino.

AssisT-Out-Android-Wear parte de la misma idea pero con otra idea, desarrollándose para *Smartwatch*. La aplicación se basa en la división de la ruta en los segmentos delimitados por cada par de *puntos* pertenecientes al trayecto. Las rutas se componen de:

- Un punto inicial
- Un punto final
- Un conjunto de puntos intermedios

La función que realiza *AssisT-Out-Android-Wear* es la de calcular los puntos de una ruta especificada y posteriormente guiar a través de ellos.

Se ha demostrado que la mejor manera de representar algo es mediante una imagen, pero en casos de no conseguir una resolución deseada y más aún cuando el usuario final padece discapacidades cognitivas, el uso de imágenes como sistema de guía puede resultar arriesgado.

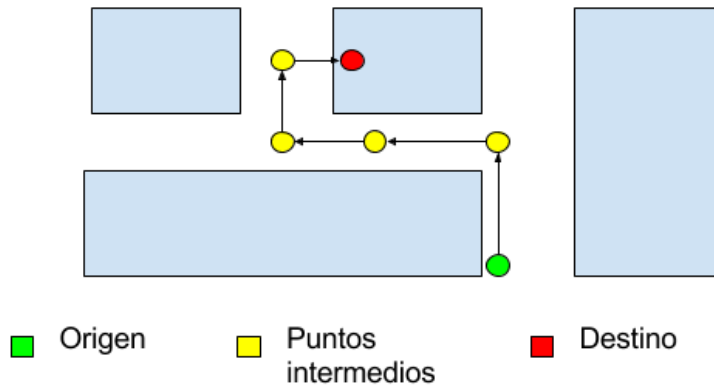


Figura 4.2. Diagrama del concepto de una ruta.

En *AssisT-Out-Android-Wear* se ha proporcionado una interfaz distinta. Debido al tamaño bastante reducido de la pantalla de un *Smartwatch*, se ha decidido usar un modo de navegación basado en una flecha. La flecha es un símbolo universal que consta de un segmento con un triángulo en un extremo, usado para indicar una trayectoria o dirección y que se utiliza en numerosos contextos de la vida diaria como por ejemplo señales de tráfico, señales de aeropuertos, señales de emergencia, guías o esquemas, lo que lo convierten en un símbolo muy común y fácil de entender. A su vez se ha querido representar de alguna manera el progreso en la distancia hasta el destino y para ello se ha pensado en que sea el propio interior de la flecha el que indique el progreso.

Uniando estas dos ideas se ha diseñado una interfaz basada en una flecha cuyo interior se va rellenando progresivamente.

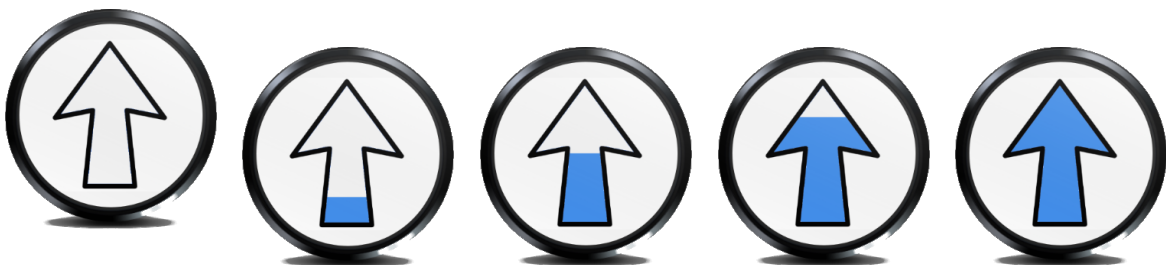


Figura 4.3. Ejemplo del rellenado de la flecha

4.5. Representación de ruta

Para poder representar la ruta *AssisT-Out* descarga las fotos de los sucesivos puntos. *AssisT-Out-Android-Wear* utiliza la localización actual y el siguiente punto de la ruta, unido la orientación actual del usuario para calcular la dirección a la que debe apuntar la flecha. Como base para la representación se ha utilizado el *sistema cartesiano de coordenadas* ya que se va a hacer uso de ángulos. La flecha indica la dirección que debe seguir el usuario para llegar al siguiente punto, de manera que la flecha se moverá en el abanico de los 360^0 cartesianos, apuntando hacia arriba (90^0) cuando el usuario se encuentra mirando al punto destino.

La tarea del cálculo del ángulo de la flecha se divide en tres partes:

1. Cálculo del ángulo entre dos puntos de la ruta.
2. Cálculo del ángulo de la orientación actual.
3. Cálculo de dirección final.

4.5.1. Cálculo del ángulo entre dos puntos de la ruta

Para realizar este cálculo se ha utilizado el siguiente método matemático:

1. Se calcula la distancia entre los dos puntos.
2. Se traza una paralela al eje x desde el punto origen.
3. Se calcula la diferencia en el eje y entre el punto origen y el destino.
4. Se calcula el ángulo que forma el segmento que une los dos puntos y la paralela al eje x.

En este método hay que tener en cuenta en qué cuadrante se realiza la dirección para corregir el ángulo calculado. Esto se realiza midiendo el incremento o decremento del valor x e y desde el punto origen al destino.

| Longitud | Latitud | Cuadrante |
|-----------|-----------|-----------|
| aumenta | aumenta | 1 |
| disminuye | aumenta | 2 |
| disminuye | disminuye | 3 |
| aumenta | disminuye | 4 |

Tabla 4.1. Tabla de cuadrantes.

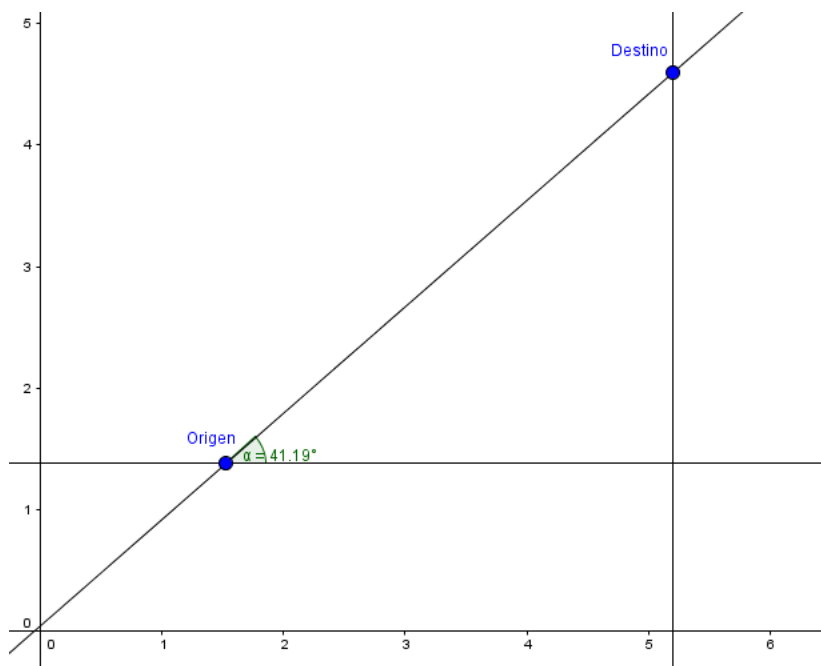


Figura 4.4. Ejemplo del cálculo de dirección en el primer cuadrante.

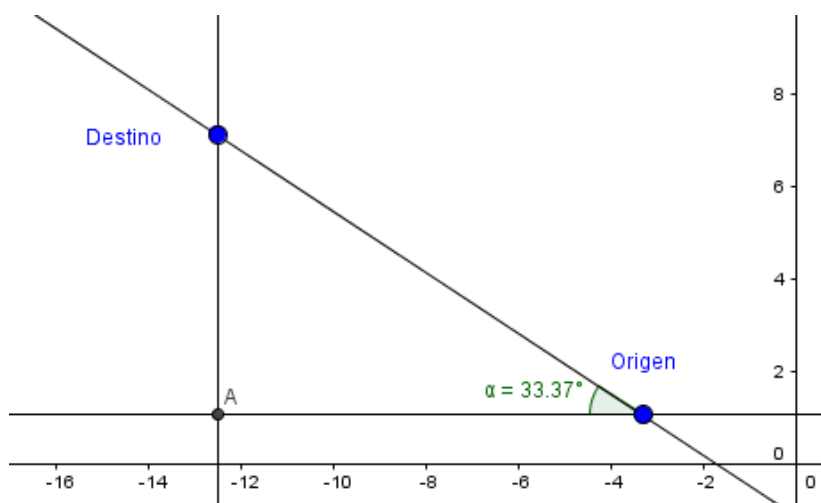


Figura 4.5. Ejemplo del cálculo de dirección en el segundo cuadrante.

4.5. REPRESENTACIÓN DE RUTA

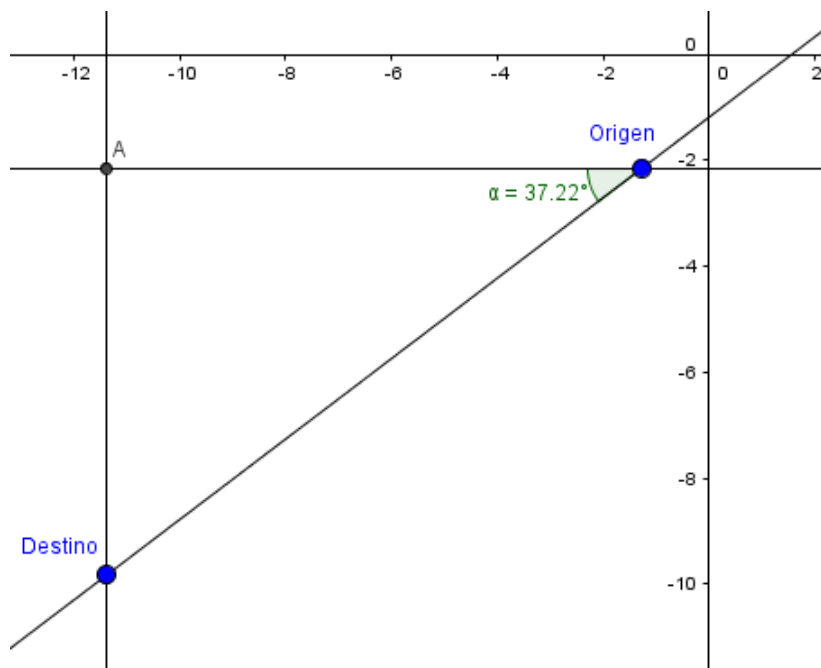


Figura 4.6. Ejemplo del cálculo de dirección en el tercer cuadrante.

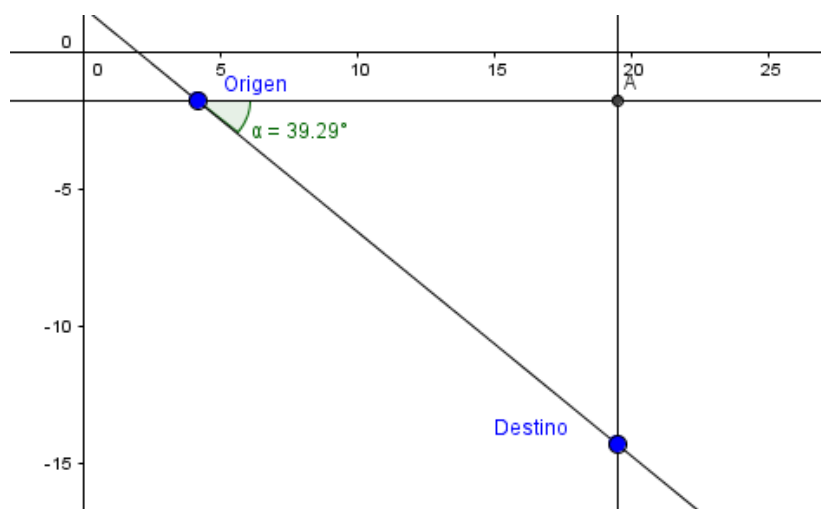


Figura 4.7. Ejemplo del cálculo de dirección en el cuarto cuadrante.

4.5.2. Cálculo del ángulo de la orientación actual

Android dispone de un sistema de brújula que permite al *Smartphone* obtener los grados acimut en la orientación actual. Los grados acimut se definen como los grados en sentido horario respecto al polo norte magnético. El sistema implementado sigue

la representación de ángulos cartesianos, por lo tanto, debido al sentido inverso de los mismos, se debe realizar una conversión de ángulos entre acimut y cartesiano.

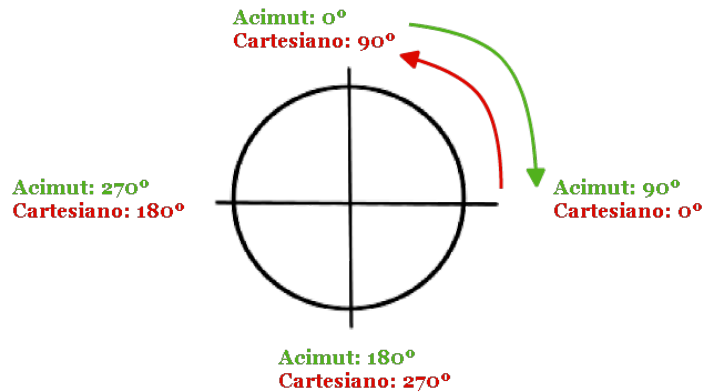


Figura 4.8. Esquema de comparación entre ángulo acimut y ángulo cartesiano.

Para dicha conversión, se han generado 2 casos, conversión en el primer cuadrante y conversión en el resto.

4.5.2.1. Primer cuadrante

En este caso se puede comprobar mediante casos de ejemplo, que, mientras los grados acimut aumentan, los grados cartesianos disminuyen y además en la misma medida:

| Grados acimut | Grados cartesianos |
|---------------|--------------------|
| 0° | 90° |
| 30° | 60° |
| 60° | 30° |
| 90° | 0° |

Tabla 4.2. Casos de prueba en el primer cuadrante.

Viendo estos resultados y teniendo en cuenta que de lo que se dispone en un principio es de los grados acimut, el cálculo del respectivo ángulo cartesiano es tan sencillo como:

$$\begin{array}{l} \text{si } \text{anguloAcimut} < 90 \\ \text{entonces } \text{anguloCartesiano} = 90 - \text{anguloAcimut} \end{array}$$

4.5.2.2. Resto de circunferencia

Bien, ya se dispone de la conversión de ángulos para el caso de que el ángulo calculado se encuentre en el primer cuadrante, pero, ¿qué pasa en el resto de cuadrantes? Pues bien, mediante el método seguido anteriormente, dando casos de prueba, se puede obtener la solución.

| Grados acimut | Grados cartesianos |
|------------------|--------------------|
| 270 ^o | 180 ^o |
| 300 ^o | 150 ^o |
| 315 ^o | 135 ^o |
| 330 ^o | 120 ^o |
| 360 ^o | 90 ^o |

Tabla 4.3. Casos de prueba en el segundo cuadrante.

| Grados acimut | Grados cartesianos |
|------------------|--------------------|
| 180 ^o | 270 ^o |
| 210 ^o | 240 ^o |
| 225 ^o | 225 ^o |
| 240 ^o | 210 ^o |
| 270 ^o | 180 ^o |

Tabla 4.4. Casos de prueba en el tercer cuadrante.

| Grados acimut | Grados cartesianos |
|------------------|--------------------|
| 90 ^o | 360 ^o |
| 120 ^o | 330 ^o |
| 135 ^o | 315 ^o |
| 150 ^o | 300 ^o |
| 180 ^o | 270 ^o |

Tabla 4.5. Casos de prueba en el cuarto cuadrante.

Si se estudian por un momento los datos, se puede detectar un detalle: la suma de los 2 ángulos da como resultado 450. Por lo tanto, de nuevo, teniendo los grados acimut podemos calcular su respectivo ángulo cartesiano realizando la siguiente operación:

si $\text{anguloAcimut} > 90$
 entonces $\text{anguloCartesiano} = 450 - \text{anguloAcimut}$.

4.5.3. Cálculo de dirección final

Una vez ambos ángulos son calculados simplemente se calcula la diferencia entre ellos y éste será el valor representado por la flecha. A continuación se muestran ejemplos de cómo se representarían los ángulos calculados:

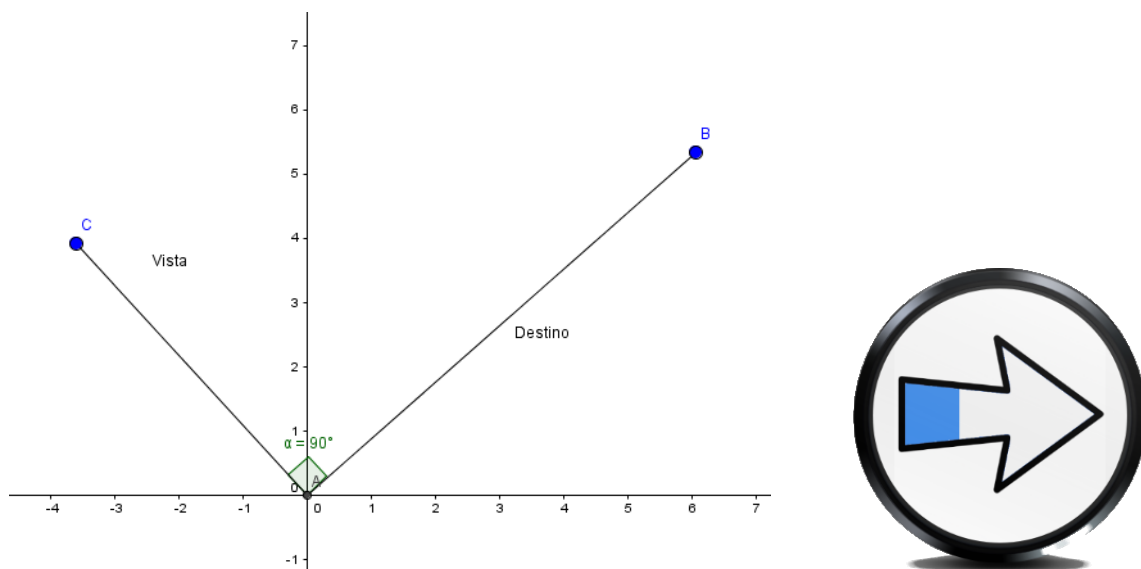


Figura 4.9. Ejemplo de cálculo y direccionamiento de 90 grados

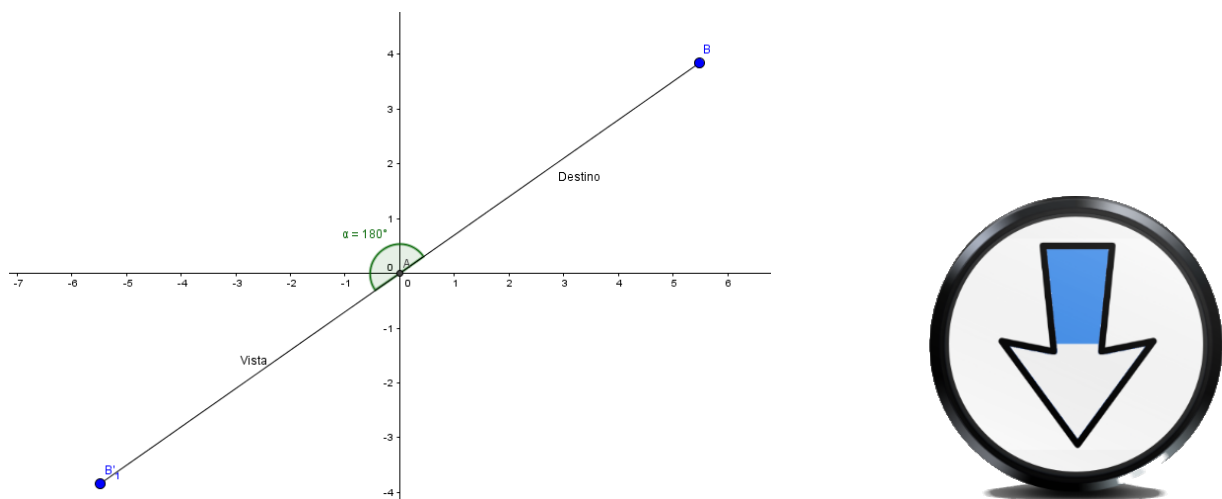


Figura 4.10. Ejemplo de cálculo y direccionamiento de 180 grados

4.5. REPRESENTACIÓN DE RUTA

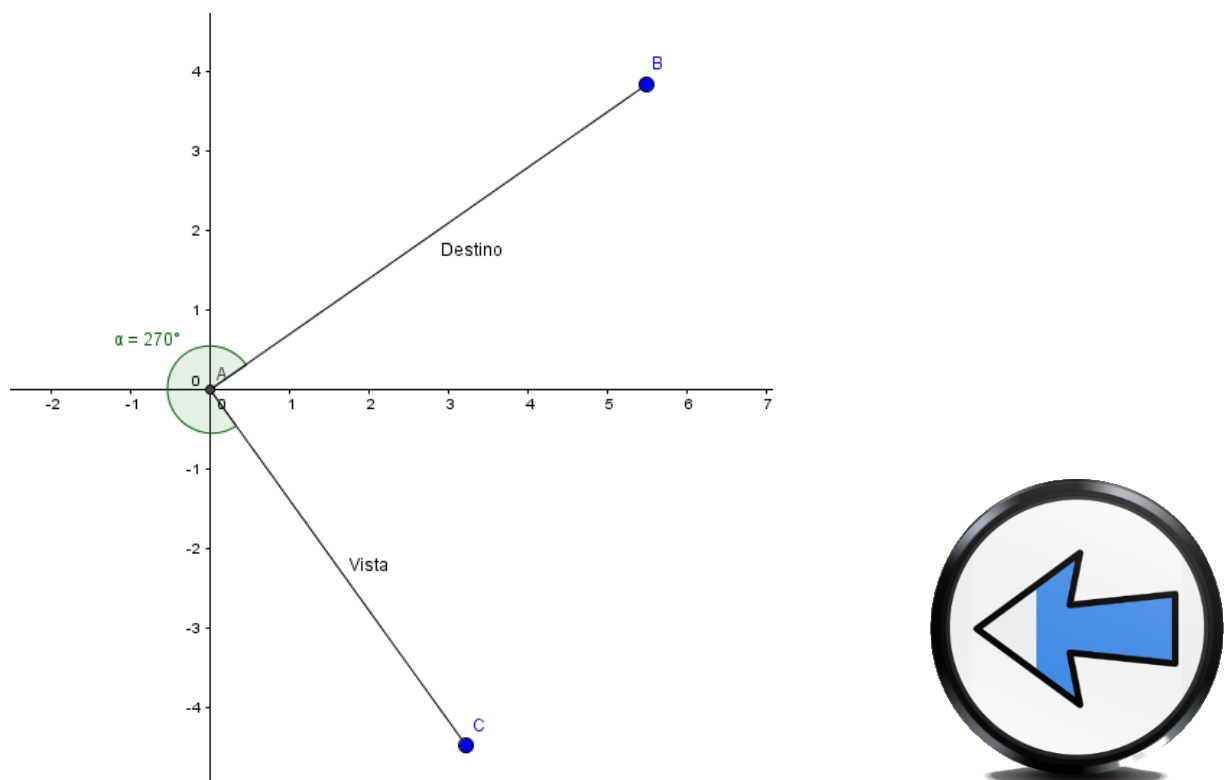


Figura 4.11. Ejemplo de cálculo y direccionamiento de 270 grados



5 Desarrollo

5.1. Introducción

Desde un inicio la aplicación se pensó que fuese *stand-alone*, es decir, que sólo se necesitase el *Smartwatch* para poder usar la aplicación, pero debido a la necesidad obligada de uso de datos y GPS, se ha optado por hacer la aplicación usando el apoyo de un *Smartphone*. A pesar de ello, la aplicación ha sido implementada desligando todo lo posible la funcionalidad del *Smartwatch* de la funcionalidad del *Smartphone*, de manera que el trabajo del último es el mínimo indispensable para el correcto funcionamiento de la aplicación.

5.2. Módulo del Smartphone

En el módulo del *Smartphone* únicamente se ejecutan las siguientes funciones:

1. Recibir la petición de ruta del *Smartwatch*
2. Obtener la ruta desde el punto actual y el destino elegido por el usuario
3. Enviar los puntos obtenidos al *Smartwatch*

Para realizar esto, se ha implementado un *listener* en el *Smartphone* que permanece a la espera de mensajes enviados desde el *Smartwatch*. Cuando un usuario elige su destino, se envían las coordenadas del destino al *Smartphone*. Una vez recibidas, éste se

conecta a un servidor el cual ejecuta una petición a *Google Maps* con el origen y destino mandados por el *Smartphone*. Posteriormente el servidor recoge la ruta calculada y devuelve el conjunto de pasos a realizar al *Smartphone*. Por último, el dispositivo móvil envía los puntos calculados al reloj para comenzar el guiado.

5.3. Módulo del Smartwatch

Este módulo se compone de las siguientes funciones:

1. Selección de destino
2. Sistema de guiado

5.3.1. Selección de destino

Al iniciar la aplicación al usuario se le muestra una ruleta de posibles destinos entre los que elegir. Al elegir uno, se realiza el flujo definido en el apartado anterior. Estos destinos en esta primera versión son fijos porque se van a hacer pruebas más adelante con esos mismos destinos, pero en un futuro los destinos serán programables mediante una solicitud a un servidor.



Figura 5.1. Ejemplo de destino de domicilio



Figura 5.2. Ejemplo de destino de escuela



Figura 5.3. Ejemplo de destino de transporte público

5.3.2. Sistema de guiado

Una vez se han recogido los puntos de la ruta y se han enviado al *Smartwatch*, el guiado da comienzo. Lo primero que se realiza es el cálculo del ángulo entre los puntos siguientes: el punto actual en el que se encuentra el usuario y el primer punto de la ruta.

5.3.2.1. Obtención de la orientación actual

Para obtener la orientación se debe implementar la interfaz *SensorEventListener*. Se deben declarar dos tipos de sensores: el acelerómetro y el magnetómetro, los cuales irán registrando los cambios de posición y la brújula. A su vez se declaran dos matrices que contendrán los datos de la orientación.

```

private Sensor accelerometer;
private Sensor magnetometer;
private float [] lastAccelerometer = new float [3];
private float [] lastMagnetometer = new float [3];
float [] r = new float [9];
float [] orientation = new float [3];

```

Para poder rellenar estas variables hay que sobrescribir la implementación del método *onSensorChanged* e insertar el código siguiente:

```

@Override
public void onSensorChanged(SensorEvent event) {
    if (event.sensor.getType() \== Sensor.TYPE_ACCELEROMETER) {
        System.arraycopy(event.values, 0, lastAccelerometer,
            0, event.values.length);
        mLastAccelerometerSet = true;
    } else if (event.sensor \== magnetometer) {
        System.arraycopy(event.values, 0, lastMagnetometer,
            0, event.values.length);
        mLastMagnetometerSet = true;
    }
    if (mLastAccelerometerSet \&\& mLastMagnetometerSet) {
        SensorManager.getRotationMatrix(r, null,
            lastAccelerometer, lastMagnetometer);
        SensorManager.getOrientation(r, orientation);
        float azimuthInRadians = orientation[0];
        float azimuthInDegrees = (float) (Math.toDegrees(
            azimuthInRadians) + 360) \% 360;
    }
}

```

De esta manera obtendremos en la variable *azimuthInDegrees* el valor de los grados acimut de la posición del usuario. Posteriormente se realiza la conversión de grados acimut a cartesiano:

```

if(acimutAngle < 90){
    return 90 - acimutAngle;
} else {
    if(450 - acimutAngle == 360){
        return 0.0;
    }
    else {
        return 450 - acimutAngle;
    }
}

```

5.3. MÓDULO DEL SMARTWATCH

Android dispone de un sistema de brújula que permite al *Smartphone* obtener los grados acimut en la orientación actual. Los grados acimut se definen como los grados en sentido horario respecto al polo norte magnético.

El sistema implementado sigue la representación de ángulos cartesianos, por lo tanto, debido al sentido inverso de los mismos, se debe realizar una conversión de ángulos entre acimut y cartesiano.

5.3.2.2. Cálculo del ángulo de dirección

Aparte de calcular el ángulo hacia el cual mira el usuario, hay que calcular el ángulo que forman el punto actual del usuario, y el punto destino.

A continuación se expone el código acorde al diseño explicado en 4.5.1.

```
public static double calcularAnguloDireccion(Location actual,
    Location destiny) {

    //distancia entre los puntos
    double distance = Math.hypot((destiny.getLongitude() -
        actual.getLongitude()), (destiny.getLatitude() -
        actual.getLatitude()));

    //diferencia de latitud (diferencia en eje y)
    double latdiff = Math.hypot(destiny.getLatitude() -
        actual.getLatitude(), 0);

    //cálculo del ángulo
    Double angle = (Math.toDegrees(Math.asin(latdiff /
        distance)));

    //cálculo de variación en eje x e y
    double longitudediff = destiny.getLongitude() - actual.
        getLongitude();
    double latitudediff = destiny.getLatitude() - actual.
        getLatitude();

    //caso de movimiento al 2 cuadrante
    if(longitudediff < 0 && latitudediff > 0 ){
        angle = 90 + (90 - angle);
    }
    //caso de movimiento al 3 cuadrante
    else if(longitudediff < 0 && latitudediff < 0){
        angle = 270 - (90 - angle);
    }
    //caso de movimiento al 4 cuadrante
    else if(longitudediff > 0 && latitudediff < 0){
        angle = 270 + (90 - angle) ;
    }
}
```

```
    }else{  
        // caso primer cuadrante  
    }  
    return angle;  
}
```

Después a ese resultado se le aplica los grados acimut de la orientación del usuario, de manera que cuando el usuario está mirando en dirección hacia el punto, la flecha se sitúa en 90° , y si giramos sobre nuestros pies, la flecha varía su indicación para seguir apuntando en dirección al destino.

Mientras se va avanzando hacia el destino, la barra interna de la flecha se va rellenando indicando el progreso, de manera que cuanto más cerca se encuentre el usuario de su destino, más se rellenará la flecha.

Una vez el usuario alcanza su destino, sin que éste sea el destino final, (con una aproximación de 10 metros) el reloj vibra y muestra un mensaje con un botón. Esto obliga al usuario a mirar al reloj y a interactuar con él y tiene el objetivo adicional de intentar dejar claro al usuario de que va a cambiar de punto de destino. Hasta que el usuario no pulse el botón, no se calculará la dirección al siguiente punto.

Una vez el destino es alcanzado, se muestra un mensaje parecido al anterior pero en este caso se informa al usuario de que ya ha alcanzado su destino y se le muestra un icono que represente la idea.



Figura 5.4. Imagen de próximo punto de la ruta



Figura 5.5. Imagen de final de ruta

5.3.2.3. Cálculo del ángulo final

```
public void dibujaFlecha() {  
    //cálculo del angulo vista  
    Double anguloVista = Utils.acimutToCartesiano(  
        currentAzimutDegree);  
  
    //cálculo del angulo dirección  
    Double anguloDireccion = calcularAnguloDireccion(  
        currentLocation, destinationLocation);  
  
    nuevoAngulo = anguloVista - anguloDireccion;  
  
    circleProgress.setAngulo(newAngle + 180);  
  
    circleProgress.setProgress( utils.calculaProgreso(  
        totalDistance, partialDistance));  
  
    //normalización del angulo  
    nuevoAngulo %=360;  
  
    //rotación de la imagen de la flecha  
    RotateAnimation ra = new RotateAnimation(  
        (float) lastNewAngle, //punto origen del  
        movimiento  
        (float) newAngle, //punto final al que rota  
        Animation.RELATIVE_TO_SELF, 0.5f,  
        Animation.RELATIVE_TO_SELF,  
        0.5f);  
  
    ra.setDuration(250);  
  
    ra.setFillAfter(true);  
}
```

```

        mPointer.startAnimation(ra);

        lastNewAngle = newAngle;
    }

```

5.3.2.4. Cálculo y representación del progreso

Para el relleno de la flecha se ha usado una modificación de la clase *View* proporcionada por *lzyzsd* [15] la cual dibuja un círculo de unas medidas preestablecidas desde un punto de la pantalla.

El cálculo de la distancia se realiza midiendo la distancia a la que se encuentra *distanciaTotal* y la distancia a la que se encuentra en ese momento o *distanciaParcial*. Ambas distancias son calculadas mediante la fórmula de la distancia entre dos puntos geográficos y el cálculo del progreso del interior de la flecha se calcula de la siguiente manera:

```

public int calculaProgreso(double distanciaTotal , double
    distanciaParcial){

    double res = 100 - (distanciaParcial*100)/
        distanciaTotal;

    return (int) res;
}

```

Esta función lo que hace es que en un inicio la *distanciaParcial* y la *distanciaTotal* son iguales, por lo que *res* vale 0 o un valor muy cercano. Según el usuario se va acercando a su destino, la *distanciaParcial* disminuye, haciendo la fracción

$$\frac{distanciaParcial * 100}{distanciaTotal} \quad (5.1)$$

cada vez más pequeña, de manera que *res* aumenta hasta llegar al 100 %.

Para establecer el progreso simplemente se utiliza la función *setProgress*:

```

circleProgress.setProgress( utils.calculaProgreso( totalDistance ,
    partialDistance ));

```

Y se le establece el ángulo del cual empieza a rellenar:

5.3. MÓDULO DEL SMARTWATCH

```
circleProgress.setAngulo(nuevoAngulo + 180);
```

En este caso hay que aplicarle un desfase de 180° puesto que en condiciones normales, sin aplicar ningún ángulo en concreto, el círculo se empieza a rellenar desde la parte superior de la pantalla.

5.3.2.5. Diagrama de clases

Por último se muestra un diagrama general de la aplicación. El diagrama muestra las clases más significativas puesto que un diagrama con todas las clases del proyecto sería excesivamente grande para mostrar aquí.

El diagrama viene dividido en dos paquetes, correspondientes a cada uno de los dispositivos. Las líneas de asociación representan la comunicación y flujo de la aplicación. En cuanto a las composiciones, todas son unitarias, es decir, una clase contenedor no utiliza más de una instancia de una clase agregada.

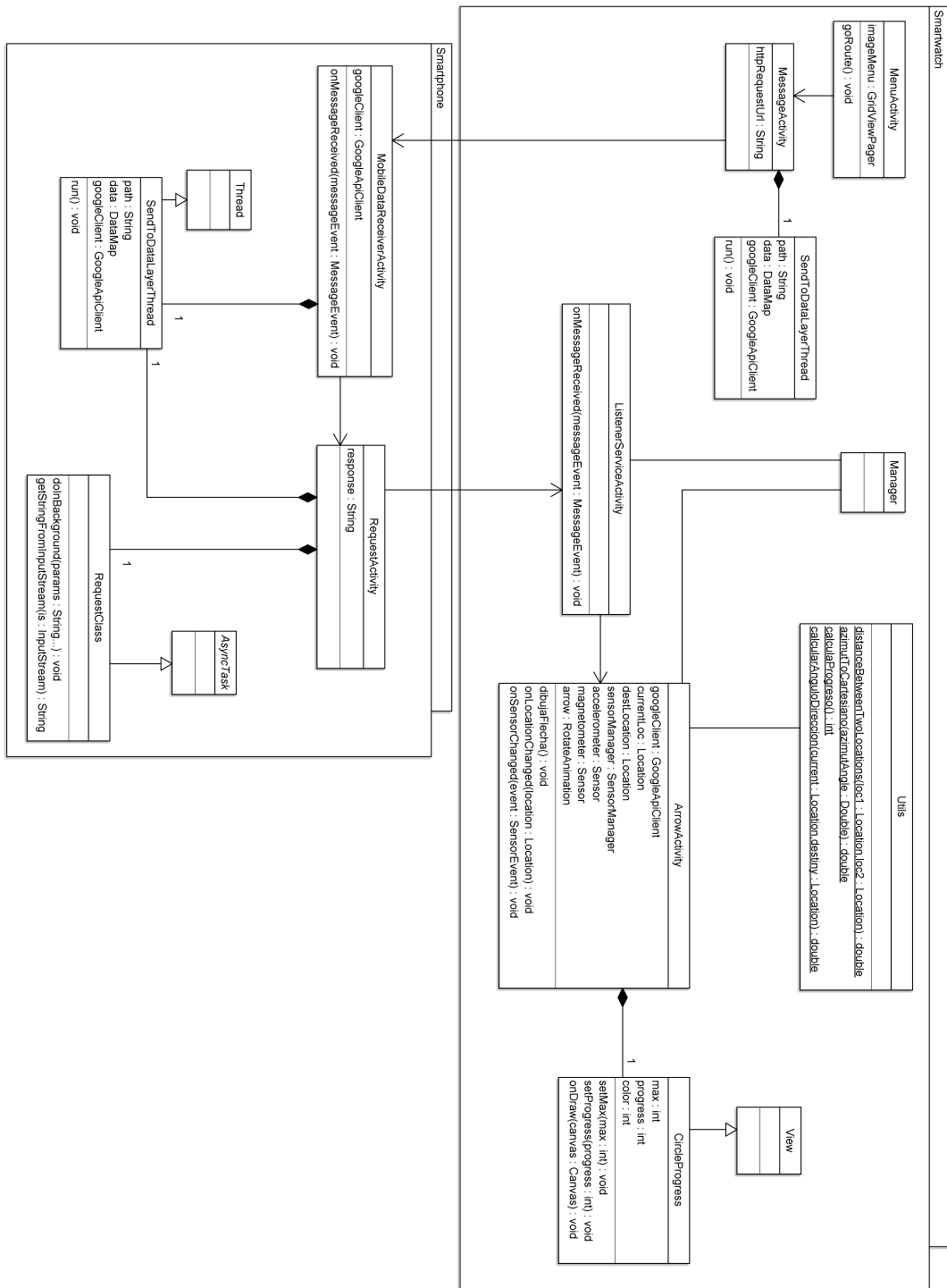


Figura 5.6. Imagen del diagrama de clases



6 Pruebas

Es este apartado se detallan las pruebas realizadas con la aplicación. En primer lugar se han ido realizando pruebas unitarias de cada módulo de la aplicación. Estas pruebas se realizaban mediante emulación de coordenadas GPS con una aplicación externa [16] así como en un entorno real realizando físicamente la ruta.

Posteriormente, una vez los distintos módulos fueron implementados y conectados, se probaron con personas reales con discapacidad. Debido a las dificultades para concertar citas con los centros y el limitado tiempo de desarrollo del *TFG*, sólo se ha conseguido realizar una prueba con un niño de 12 años. Aun así, han quedado programadas para Septiembre del 2016 unas pruebas con 10 usuarios con el centro **Prodis** [17].

6.1. Pruebas de componentes

6.1.1. GPS

Dado que uno de los ejes fundamentales de la aplicación es el sistema de *GPS*, dicho sistema fue el primero en diseñarse y probarse. Para ello se usó de apoyo la aplicación de *Google Maps*. Mediante la comparación entre las coordenadas obtenidas a través del *GPS* del *Smartphone* y las mismas coordenadas mostradas en *Google Maps* se pudo determinar que las coordenadas del *Smartphone* eran correctas.

6.1.2. Brújula

Para probar el sistema de la brújula se recurrió a la comparación de los grados de acimut obtenidos en el *Smartwatch* con otras aplicaciones de brújula en el mismo dispositivo y en otros del laboratorio.

6.1.3. Obtención de coordenadas

Se comprobó usando *Google Maps* que el *Smartwatch* mandaba la petición de ruta al *Smartphone*, éste a su vez conectaba con el servidor, obtenía los puntos de la ruta y éstos eran devueltos al *Smartwatch*.

6.1.4. Enrutado

Se comprobó mediante aplicaciones de emulación *Fake GPS* que el direccionamiento de la flecha era el correcto y que se indicaba el progreso de manera adecuada.

6.2. Pruebas con usuarios

La aplicación fue probada con un total de 10 usuarios distintos en 3 entornos diferentes.

6.2.1. Usuarios sin discapacidad

9 de los usuarios no presentaban ninguna discapacidad en especial y sirvieron como primera toma de contacto de la aplicación con un caso real además de poder pulir fallos y realizar mejoras de cara a un caso con un usuario final. En esas pruebas se vieron factores como:

1. Capacidad de un usuario de seguir el direccionamiento.
2. Interacción del usuario con los cambios de puntos.
3. Distancia a la que debe estar el *Smartphone* del *Smartwatch*.
4. Sensibilidad del sistema *GPS* para detectar que se ha alcanzado un punto.

6.2.2. Usuario con discapacidad

Uno de los usuarios fue un niño de 12 años con discapacidad cognitiva. El niño tiene conocimientos previos con dispositivos electrónicos como *Smartphone* o *Tablets* y tiene independencia a la hora de moverse por unos pocos lugares que conoce. Es capaz de leer y de copiar un texto de forma adecuada pero sin embargo debe ir supervisado por sus padres cuando se mueven por otras zonas y posee alguna dificultad al hablar. Este usuario realizó dos rutas preestablecidas: una pequeña de entrenamiento y toma de contacto con el sistema y otra ruta para tomar resultados.

La primera ruta fue sencilla para que al usuario no le costase comprender los mecanismos de la aplicación y corta para no cansarlo. La segunda ruta se enfocó como un juego en el cual el usuario debía recorrer el trayecto para poder obtener un premio al final. Durante la prueba el usuario llevaba el *Smartwatch* y yo le acompañaba de cerca haciéndole preguntas para comprobar cómo se comportaba con la aplicación.

Los resultados fueron estos:

1. No tuvo problemas a la hora de entender el sistema de direccionamiento. El usuario supo en todo momento adónde tenía que ir.
2. Sabía en todo momento qué dirección estaba siguiendo y si iba o no en la dirección correcta. A la pregunta "*¿Vamos bien?*" el usuario era capaz de contestar si estaba realizando el trayecto correctamente.
3. Entendió el sistema de rellenado de la flecha. El usuario fue capaz de decir si nos encontrábamos cerca o lejos del próximo destino.
4. Reaccionaba bastante rápido a la hora de tocar la pantalla para cambiar de punto. El sistema de vibración funcionó perfectamente alertando al usuario del siguiente punto y éste no presentó ningún problema pulsando la pantalla para pasar al siguiente.

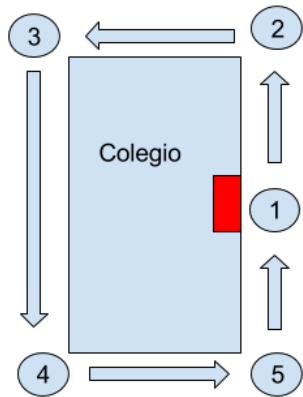


Figura 6.1. Ruta de calentamiento

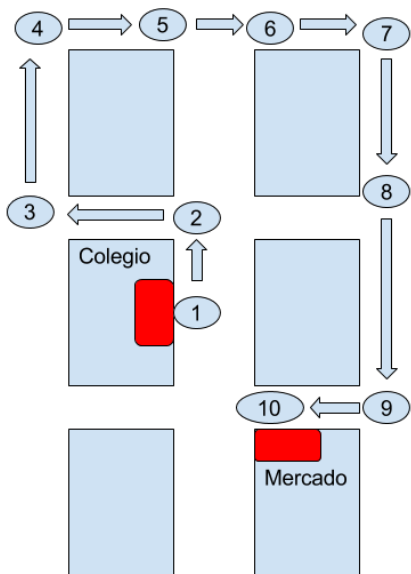


Figura 6.2. Ruta de prueba del sistema



7 Conclusiones

7.1. Conclusiones

El desarrollo de aplicaciones en tecnologías emergentes tiene dos características fundamentales: innovación y escaso soporte/documentación. La plataforma de los *Smartwatches* es muy joven aún (nació en 2012) y a pesar de estar apadrinada por *Android* o *Apple* aún tiene mucho camino que recorrer. En la actualidad la funcionalidad de un *Smartwatch* es bastante reducida, básicamente realizan tareas de notificación o medición de datos, pero a día de hoy no existen *Smartwatches* que posean una SIM propia, con lo cual no se dispone de acceso a datos ni a GPS. Las API de desarrollo aún están bastante limitadas y en muchas ocasiones es necesario consultar foros para obtener información acerca de alguna funcionalidad o solucionar algún error. Otro hándicap muy importante son los tiempos de despliegue. En mi caso trabajé con un LG G Watch R el cual no disponía de cable USB que lo conectase al pc, por lo que los despliegues se realizaban mediante Bluetooth, tardando una media de 4 minutos de reloj en ejecutar cada prueba. Sin embargo esto ya está cambiando. Los entornos de desarrollo están incorporando nuevas tecnologías que aceleran muy notablemente el tiempo de despliegue y nuevas APIs están apareciendo con nuevas y mejoradas funcionalidades y una mayor documentación. Los proyectos que se han realizado y se están realizando son la base para el desarrollo de proyectos mayores en el futuro, y todos los problemas e ideas que se sufren ahora servirán para que la plataforma mejore. Los *Smartwatches* siempre tendrán que rendir con el problema de su limitada pantalla y su limitado espacio para hardware, pero yo creo que en unos años, con avances como reproducción de sonido o capacidad para una SIM propia pueden convertirse en un dispositivo muy utilizado. Todas estas mejoras a su vez, ayudarán a su empleo en tareas de asistencia para personas con necesidades especiales como por ejemplo, se podría proporcionar ayuda por voz a personas con ansiedad o un sistema de recordatorio por voz con sincronización

en la nube. El futuro de los *SmartWatches* pinta esperanzador pero será el tiempo y en su medida los usuarios, los que decidan su destino.

7.2. Trabajo Futuro

Como ideas de trabajo futuro se presentan las siguientes opciones:

1. **Stand-alone:** Convertir la aplicación a stand-alone de manera que sólo sea necesario el uso de un *Smartwatch*.
2. **Selección de destinos dinámica:** Diseño de un sistema para la configuración de destinos.
3. **Ayuda por voz:** Diseño de un sistema de voz que vaya informando al usuario de las acciones a realizar o del progreso del trayecto.
4. **Colores daltónicos:** Uso de una paleta de colores apta para personas daltónicas.



Bibliografía

- [1] R. Lardoeyt Ferrer, H. R. Guas, O. P. Estévez, M. J. Arguello, R. R. Pujal, H. H. Madera, S. B. Constanten, and M. G. Fonseca, “Etiología genética en el origen de la discapacidad intelectual en la república del ecuador,” *Revista Cubana de Genética Comunitaria*, vol. 5, no. 2, pp. 44–49, 2011.
- [2] S. Garcia and E. ESPECIAL, “Deficiência mental,” *Aspetos psicoevolutivos y educativos*, 1994.
- [3] H. S. Basile, “Retraso mental y genética síndrome de down,” *Revista Argentina de Clínica Neuropsiquiátrica*, vol. 15, no. 1, pp. 9–23, 2008.
- [4] R. Novell Alsina, P. Rueda Quitllet, L. Salvador Carulla, and E. Forgas Farre, “Salud mental y alteraciones de la conducta en las personas con discapacidad intelectual: Guía práctica para técnicos y cuidadores,” 2015.
- [5] DISCAPACIDAD_INTELECTUAL2009, “<http://discapacidadintelectual09.blogspot.com.es/2009/06/grados-de-afectacion-de-la-discapacidad.html>,”
- [6] V. Tomberg, T. Schulz, and S. Kelle, “Applying universal design principles to themes for wearables,” in *International Conference on Universal Access in Human-Computer Interaction*, pp. 550–560, Springer, 2015.
- [7] A. Gillespie, C. Best, and B. O’Neill, “Cognitive function and assistive technology for cognition: A systematic review,” *Journal of the International Neuropsychological Society*, vol. 18, no. 01, pp. 1–19, 2012.
- [8] W. H. Organization, *International Classification of Functioning, Disability and Health: ICF*. World Health Organization, 2001.

- [9] G. Lancioni, J. Sigafos, M. F. O'Reilly, and N. N. Singh, *Assistive technology: Interventions for individuals with severe/profound and multiple disabilities*. Springer Science & Business Media, 2012.
- [10] J. C. Torrado, G. Montoro, and J. Gomez, "The potential of smartwatches for emotional self-regulation of people with autism spectrum disorder," 2016.
- [11] P. Chandrasekara, T. Mahaulpatha, D. Thathsara, I. Koswatta, and N. Fernando, "Landmarks based route planning and linear path generation for mobile navigation applications," *Spatial Information Research*, pp. 1–11, 2016.
- [12] D. Dobbstein, P. Henzler, and E. Rukzio, "Unconstrained pedestrian navigation based on vibro-tactile feedback around the wristband of a smartwatch," in *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, pp. 2439–2445, ACM, 2016.
- [13] AXSMAP, "www.axsmap.com,"
- [14] handimap, "www.handimap.org,"
- [15] lzyzsd, "<https://github.com/lzyzsd/CircleProgress/>,"
- [16] Fake GPS, "<https://play.google.com/store/apps/details?id=com.blogspot.newapphorizons.fakegps&hl=es/>,"
- [17] Prodis, "<http://www.fundacionprodis.org/>,"