

Escuela Politécnica Superior

19
20

Trabajo fin de grado

Uso de tecnologías wearable para la asistencia de personas con diversidad funcional intelectual en sus actividades de la vida diaria



Héctor Cuervo Carrizo

Escuela Politécnica Superior
Universidad Autónoma de Madrid
C/ Francisco Tomás y Valiente nº 11

**UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR**



Grado en Ingeniería de Textos

TRABAJO FIN DE GRADO

**Uso de tecnologías wearable para la asistencia de
personas con diversidad funcional intelectual en
sus actividades de la vida diaria**

Autor: Héctor Cuervo Carrizo

Tutor: Javier Gómez Escribano

Ponente: Germán Montoro Manrique

junio 2020

Todos los derechos reservados.

Queda prohibida, salvo excepción prevista en la Ley, cualquier forma de reproducción, distribución comunicación pública y transformación de esta obra sin contar con la autorización de los titulares de la propiedad intelectual.

La infracción de los derechos mencionados puede ser constitutiva de delito contra la propiedad intelectual (*arts. 270 y sgts. del Código Penal*).

DERECHOS RESERVADOS

© 3 de Noviembre de 2017 por UNIVERSIDAD AUTÓNOMA DE MADRID
Francisco Tomás y Valiente, nº 1
Madrid, 28049
Spain

Héctor Cuervo Carrizo

Uso de tecnologías wearable para la asistencia de personas con diversidad funcional intelectual en sus actividades de la vida diaria

Héctor Cuervo Carrizo

C\ Francisco Tomás y Valiente Nº 11

IMPRESO EN ESPAÑA – PRINTED IN SPAIN

A todos los que han estado ahí!

*Soy pregonero del negro
y tengo en cama la opinión,
Sé que no existe el infierno
¡Qué desilusión!*

Rosendo Mercado

PREFACIO

El presente Trabajo de fin de grado, ha sido elegido y desarrollado por dos motivos. El primero de ellos es porque se trata de desarrollar un programa centrado en ayudar a que personas con diversidad intelectual, puedan desarrollar tareas importantes en su día a día, con más seguridad en sí mismos, y puedan sentirse más autónomos y funcionales, dentro de una sociedad que no está diseñada para que personas que difieren un poco de la norma o que necesitan de ciertas ayudas o apoyos extra, puedan llevar a cabo una vida con plenitud.

El segundo motivo es que se desarrolla para un dispositivo innovador, con relativamente poca información en las redes, y con apenas documentación escrita sobre la que apoyarse a la hora de programar o diseñar, sobre todo si lo comparamos con lo que podemos encontrar a la hora de desarrollar para servidores, páginas web, ordenadores o smartphones o tabletas. Esto se convierte en un reto para cualquier aficionado a la informática, pero dificulta y ralentiza enormemente el trabajo, a la vez que aumenta la satisfacción cuando se consigue el resultado esperado.

Como persona con diversidad funcional, en mi caso física, y estudiante de informática, el poder unir los dos ámbitos que rodean mi vida, incrementan todavía más la intención de que esta iniciativa se lleve a cabo y pueda ser de utilidad para aquellas personas que lo necesiten y puedan servirse de este trabajo.

Héctor Cuervo Carrizo

AGRADECIMIENTOS

El primer lugar, como no, agradecer a mi familia directa, la cual ha hecho lo posible y lo imposible para que yo haya podido escalar hasta la universidad, y me han apoyado y ayudado en todo lo necesario, que ha sido mucho.

En segundo lugar, a los amigos con los que me he ido topando a lo largo de mi vida, tanto en las escuelas o institutos de mi León natal, como los que han estado a mi lado en esta nueva andadura universitaria en Madrid, que me han ayudado en multitud de ocasiones y sin ellos no habría sido posible llegar hasta aquí.

Y por último a todos los profesionales, tanto profesores como demás personal, que han sido cruciales para mí y han creído en que mi proyecto de vida podía llevarse adelante.

RESUMEN

El presente trabajo de fin de Grado trata de diseñar y desarrollar una aplicación para dispositivos SmartWatch, que sirva como apoyo o ayuda, para que aquellas personas con diversidad intelectual que se adapten a este tipo de dispositivos, en la realización de tareas diarias de la vida cotidiana.

El usuario final, encontrará en nuestro programa, y en este tipo de dispositivos, el apoyo que necesita, siempre en el lugar donde lo necesita, puesto que este tipo de dispositivos wearable acompañan al usuario allá donde este pueda necesitarlos. En este caso, por tratarse de un reloj, siempre estará en su muñeca.

Para conseguir tal fin, se diseñará un programa que consta de una lista de diferentes tareas que el usuario realiza a lo largo de su día. Al seleccionar una de estas tareas, el dispositivo comenzará a enumerar los pasos que se han de hacer para realizarla correctamente. De este modo se le muestran al usuario todos los pasos que ha de seguir y de este modo no se le olvidará ninguno, o en caso de bloqueo, sabrá cómo seguir adelante.

Siendo los SmartWatch una tecnología sin duda innovadora, el usuario ha de poder adaptarse a ella, por ello para hacer que la curva de aprendizaje sea lo más suave posible, se usarán todos los medios disponibles como pueden ser fotografías o pictogramas, un texto explicativo y si el dispositivo lo permite, un audio que exprese en viva voz la tarea o paso a realizar. De este modo conseguimos que sea el software el que se adapte lo máximo posible al usuario, ofreciendo así una experiencia lo más simple y efectiva posible.

PALABRAS CLAVE

Diversidad funcional, Diversidad cognitiva, discapacidad, sistema de guiado, tecnologías de asistencia, tareas, vida cotidiana, SmartWatch, wearable, wear os, android

ABSTRACT

This dissertation focuses on the design and development of an application for SmartWatch devices that serves as support or aid to those people with intellectual diversity that can adapt to this type of devices when carrying out tasks from daily life.

The final users will find in our program, and in this type of devices, the support they need and always when they need it, as these wearable devices are with the users wherever they may need it. In this case, as it is a watch, it will always be in their wrists.

In order to achieve this, we will develop a program consisting on a list of different tasks that the user carry out during their day. When selecting one of these tasks, the device will detail the steps that must be done in order to properly perform it. In this way, the user is shown every step to follow and thereby they will not forget any of them, or in case of blockade, they will know how to continue.

The SmartWatch is without a doubt an innovative technology, and this is why the users have to be able to adapt to it. Therefore, and in order to make the learning curve as smooth as possible, every mean on hand will be used, such as pictures or pictograms, an explanatory text and, if the device allows it, an audio that puts in words the task or step to perform. In this way, we secure that the software is the one adapting to its fullest to the user, and that it can offer the most straightforward and effective experience possible.

KEYWORDS

Functional diversity, Cognitive diversity, disability, guide system, assistive technology, tasks, daily life, SmartWatch, wearable, wear os, android

ÍNDICE

1	Introducción	1
1.1	Motivación	1
1.2	Objetivos	2
1.3	Organización	3
2	Estado Del Arte	5
2.1	Estado Del Arte	5
2.2	Conclusiones	9
3	Diseño	11
3.1	Patrón de diseño	11
3.2	Modelo de proceso	11
3.3	Requisitos	14
3.4	Maquetas	14
4	Desarrollo	19
4.1	Base de datos	20
4.2	Permisos	20
4.3	Codificación	23
4.4	Pruebas y resultados	28
4.5	Interfaz de usuario	29
5	Conclusiones y trabajo futuro	37
5.1	Conclusiones	37
5.2	Trabajo Futuro	38
	Bibliografía	39

LISTAS

Lista de figuras

2.1	Artículos publicados por año	6
3.1	Modelo vista controlador	12
3.2	Maqueta 1	15
3.3	Maqueta 2	16
3.4	Maqueta 3	16
3.5	Maqueta 4	17
4.1	Base de datos	21
4.2	Permisos	21
4.3	Conexiones	24
4.4	Descargar base de datos	24
4.5	RecyclerView	25
4.6	WearableRecyclerView	26
4.7	Avanzar y retroceder	27
4.8	Estyless Swipe To Dismiss	27
4.9	Leer Tarea	27
4.10	Menú de aplicaciones	30
4.11	Complicaciones de la esfera	31
4.12	Descarga de la base de datos	31
4.13	Lista de actividades	32
4.14	Pasos de una actividad	33
4.15	Tarea terminada	34
4.16	Permiso de almacenamiento	35

INTRODUCCIÓN

1.1. Motivación

La inclusión de las personas con diversidad funcional es sin duda una de las grandes metas de esta sociedad en todos sus ámbitos. Desde la educación inclusiva, a la inclusión en los puestos de trabajo, pasando por la vida personal y familiar. Para conseguirlo es muy importante, que se pongan todos los medios actuales a disposición de tal fin.

Dentro del mundo de las nuevas tecnologías, nos vamos encontrando cada vez con nuevas alternativas tanto de software como de hardware que de una u otra manera pueden ayudar a que todas las personas nos igualemos en oportunidades y estemos más integrados y por lo tanto menos discriminados dentro de la sociedad, independientemente de las necesidades individuales de cada uno.

Más en concreto, este TFG, tiene el fin de adaptar los relojes inteligentes o SmartWatch a tal fin, dándole al usuario que lo necesite, la posibilidad de utilizar una herramienta normalizadora, usable y autónoma, con la que puedan llevar a cabo sus tareas diarias con más independencia.

Por otra parte es necesario destacar, que para el desarrollo de esta aplicación, se tendrán en cuenta tanto experiencias personales como estudios anteriores y trabajos y aplicaciones realizados por el laboratorio y el equipo de trabajo del Ambient Intelligence Lab (AmILab) de la Escuela Politécnica Superior de la Universidad Autónoma de Madrid.

En concreto se quiere diseñar una herramienta para SmartWatch que ofrezca un guiado en la realización de tareas de uso diario para personas con diversidad funcional intelectual. Por ende utilizaremos un SmartWatch como plataforma hardware, con conexión a internet, el cuál deberá disponer de un sistema operativo lo suficientemente abierto para que el usuario pueda elegir el dispositivo que más se acerque a sus requerimientos.

Debido a las características especiales de los usuarios de nuestra aplicación, el programa resultante deberá estar diseñado de tal forma que ofrezca una fácil comprensión por parte de los usuarios finales. Además, tanto su funcionamiento como la información que preste, a de ser sumamente sencillo para brindar el menor tiempo de adaptación posible, ofreciendo una mayor autonomía en el día a día a

dichas personas de una forma rápida.

Con el fin de minimizar el número de aparatos que se han de aprender a manejar y reducir de este modo la curva de aprendizaje, el programa ha de ser autónomo, o sea, no debe depender de ningún otro elemento cómo pueden ser teléfonos móviles o tabletas para su uso por parte del usuario final. Únicamente el SmartWatch se conectará a Internet y descargará las actividades que necesite realizar el usuario.

1.2. Objetivos

La creación de Assist-Task Wear tiene como objetivo ayudar a personas con diversidad funcional intelectual, mostrándole en un SmartWatch los pasos que ha de seguir para realizar diferentes tipos de tareas que el usuario realice en su vida diaria.

Para conseguir un programa funcional, no solamente ha de realizar las tareas para las que ha sido pensado, si no que las ha de realizar cumpliendo una serie de características que hagan que sea plenamente usable y accesible para las personas que van a ser sus usuarios finales.

Debido a la pequeña pantalla del reloj se deberá de mostrar la información de todas las maneras que sea posible. Tanto en imágenes o pictogramas, dependiendo de lo que más acostumbrado esté el usuario, como una pequeña frase descriptiva o un mensaje de viva voz expresado mediante el altavoz interno del smartwatch. De este modo podemos abarcar un mayor rango de usuarios, ya que cada persona puede necesitar o preferir una u otra manera de acceder a la información.

También, debido al tamaño de la pantalla, el uso de botones en esta resulta inadecuado para el desplazamiento por los diferentes elementos de la tarea que el usuario quiera realizar. Es preferible dejar la pantalla completa para mostrar la información, por lo que la navegación dentro de la aplicación se deberá hacer mediante la utilización de gestos en la propia pantalla. También, el uso de gestos resulta más intuitivo lo cual ayudará a que la curva de aprendizaje se reduzca al máximo.

Hacer que el usuario se sienta seguro utilizando la aplicación para SmartWatch, es algo fundamental para conseguir que nuestro trabajo resulte útil y se convierta en una herramienta beneficiosa y conveniente para el día a día de las personas que lo utilicen. La mayoría de estas personas están acostumbradas a tener a alguien que lo supervise y les guíe en sus quehaceres. Por ello debemos de tener en cuenta la novedad que supone el que sea un reloj el encargado de esta tarea y por ello hay que brindar al usuario las posibilidades que el SmartWatch nos ofrezca para conseguir dicha fiabilidad.

Puede suceder que por motivos dispares un usuario realice el gesto de desplazar el dedo por la pantalla de manera impulsiva, sobre todo si el reloj no ha reconocido el gesto correctamente la primera vez que lo realiza. Debemos tener en cuenta estos comportamientos y evitar que pueda cometer errores, o que no vea correctamente algún paso de la actividad que está realizando. De este modo

ayudaremos a que la persona use el SmartWatch con más tranquilidad y no se ponga nervioso si algo no le sale correctamente a la primera.

Por otro lado, si el usuario no está seguro de haber realizado todos los pasos anteriores en un momento dado, siempre podrá volver atrás, como se ha dicho, de forma intuitiva mediante gestos, al paso anterior, ayudando una vez más, a que el usuario se sienta seguro con la utilización de la aplicación, aumentando su seguridad en la tarea que quiere realizar

El hecho de desplazarnos mediante gestos puede chocar con el propio sistema operativo que elijamos, ya que este puede tener gestos predefinidos. Como ejemplo podemos citar sistema operativo Wear Os, en el cual el hecho de desplazar el dedo de izquierda a derecha por la pantalla significa volver a la aplicación anterior. Por ello debemos tener la opción de bloquear dichos gestos del sistema operativo dentro de nuestra aplicación, para que realicen la opción que se desea obtener.

Es conveniente igualmente, que SmartWatch sea el único dispositivo necesario para que los usuarios finales puedan conseguir sus objetivos. Evitando la utilización de otros aparatos como por ejemplo teléfonos inteligentes o tabletas. De este modo la persona que utilice nuestro programa no tendrá que aprender a utilizar nada más. Para ello el SmartWatch tendrá que ser completamente autónomo, conectándose mediante Wi-Fi o mediante 3G para descargar las tareas o actualizaciones de nuestro programa e incorporarlas a una base de datos, para que aunque el SmartWatch esté separado del teléfono o de cualquier fuente de acceso a Internet, funcione correctamente y muestre en todo momento la información necesaria por la pantalla.

La fácil y correcta comprensión de la información mostrada y la fiabilidad y tranquilidad para el usuario final, de que el reloj le brindará todo el apoyo necesario que pueda necesitar, es algo fundamental para la incorporación de esta tecnología en el día a día las personas que lo necesiten.

1.3. Organización

Esta memoria consta de 6 capítulos y la bibliografía.

- **Capítulo 1 Introducción:** en este apartado se explica la motivación y los objetivos del proyecto, así como algún ligero esbozo de las soluciones que planteamos para su realización. Además se incluye la explicación de la estructuración que va a seguir la memoria.
- **Capítulo 2 Estado del arte:** en este apartado se realiza un estudio del estado actual de la tecnología utilizada y de las necesidades de las personas que van a usar nuestro producto, para de este modo, hacer entender la necesidad de la creación de esta aplicación.
- **Capítulo 3 Diseño:** en este apartado explicamos el modelo utilizado para el diseño de nuestro programa, los requisitos tanto funcionales como no funcionales y explicamos como ha de ser el funcionamiento paso a paso del mismo.
- **Capítulo 4 Desarrollo:** en este apartado indicamos cuál ha sido el proceso de codificación de la aplicación, mostrando algunos métodos que han sido importantes a la hora de poner en marcha los requisitos de nuestro

proyecto.

- **Capítulo 5 Pruebas y resultados:** en este apartado se explicarán las pruebas que se han hecho para el correcto funcionamiento de nuestro proyecto.
- **Capítulo 6 Conclusiones y trabajo futuro:** en este apartado daremos las conclusiones sobre el trabajo realizado e indicaremos distintas ideas o proyectos que se pueden realizar en un futuro sobre nuestra aplicación.
- **Bibliografía.**

ESTADO DEL ARTE

2.1. Estado Del Arte

Al investigar sobre el estado del arte en los sectores que pueden influir en este proyecto, se comienza a buscar en la página de Recursos científicos dependiente de la Fundación Española para la Ciencia y la Tecnología (FECYT) y del Ministerio de Ciencia innovación y universidades. En dicha página, encontramos el acceso a la base de datos de la Web of Science la cual cuenta con una amplia colección de bases de datos en las que se encuentran revistas académicas, libros, actas, información bibliográfica, patentes y demás recursos académicos y de investigación.

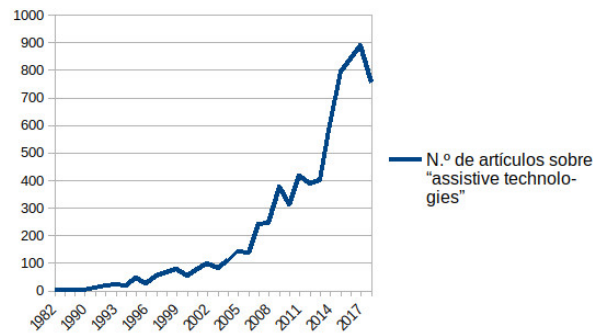
Haciendo búsquedas sobre disabilities o assistive technologies y más concretamente, búsquedas sobre wereables o SmartWatch, podemos observar como cada vez con más popularidad se está investigando y realizando trabajos sobre estos temas concretos.

Cómo se puede observar a continuación el incremento es sustancial en los últimos años. Por ejemplo el número de artículos sobre disabilities comienza a principios de 1900 con un poco más de 10 y, pasando a 100 o 200 en los años 60-70, triplicándose en los 80-90, llegando a 1500 en los 2000 y a 3000 en 2010 y a más de 6000 en los años anteriores a la publicación de este escrito.

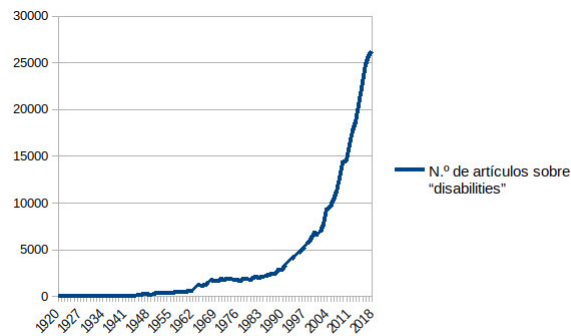
A continuación podemos ver alguno de estos ejemplos en la figura 2.1 de las búsquedas realizadas:

Metidos más de lleno en el mundo de los SmartWatch, la popularidad comienza con la llegada al mercado del sistema operativo diseñado para este tipo de dispositivos llamado Android wear en 2014, saliendo a la venta con relojes tan populares como fueron el LG G Watch, el Samsung Galaxy Gear Live o el Motorola Moto 360. Este sistema operativo de Android fue evolucionando en diferentes actualizaciones hasta llegar a la más importante que fue la Android wear 2.0 que fue presentada en 2016, y que tuvo su mayor cambio en 2018 cuando el propio sistema operativo se volvió independiente de los teléfonos y cambió de nombre a Wear Os.

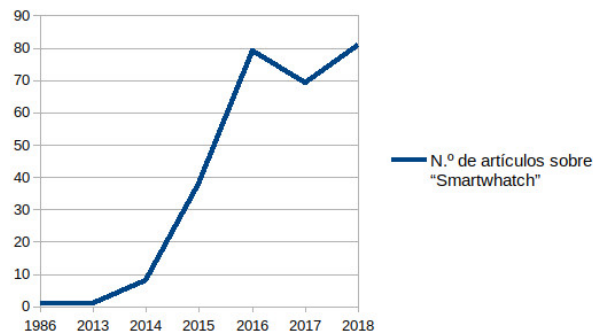
Cabe destacar, que fue también en 2014 cuando Apple anuncio la primera versión de su reloj inteligente llamado Apple Watch el cual, como es seña de identidad de la marca, disponía de su propio sistema operativo y lenguaje de programación, y nada tiene que ver con el sistema operativo de Google.



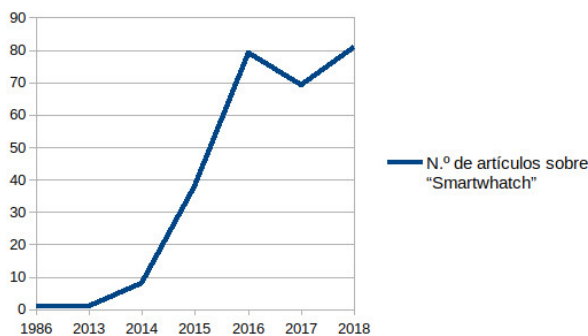
(a) Número de artículos sobre discapacidades



(b) assistive technologies



(c) Número de artículos sobre wereables



(d) Número de artículos sobre smartWatch

Figura 2.1: En este conjunto de gráficas se muestra como la evolución de las publicaciones científicas, sobre las diferentes temáticas en las que podemos encuadrar nuestro trabajo de fin de grado, cómo pueden ser "disabilities" 2.1(a), "assistive technologies" 2.1(b), "wearable" 2.1(c) o "Smart-Watch" 2.1(d), crece de forma exponencial. De este modo podemos ver como nuestro trabajo está encuadrado dentro lo que pueden ser necesidades o tendencias actuales.

El siguiente paso de la investigación nos lleva a buscar más concienzudamente en las publicaciones que podemos encontrar al respecto de los temas que nos influyen, encontrando de este modo matices o aspectos que ya se han tenido en cuenta en anteriores trabajos y buscando en qué podemos avanzar dentro de este sector para crear algo funcional y útil para las personas que lo necesiten.

Según varias fuentes consultadas, la mejor y más completa definición de discapacidad intelectual (según argumentan autores como Schalock, Robert L. [1]) o diversidad funcional intelectual (que sería el término elegido por otros autores tales como Javier Romañach, Manuel Lobato [2]) sería la ofrecida por la Asociación Americana de Discapacidades Intelectuales y del Desarrollo (AAIDD) [3] la cual la define como: “La discapacidad intelectual es un estado individual que se caracteriza por presentar limitaciones significativas tanto en el funcionamiento intelectual como en la conducta adaptativa, tal y como se manifiesta en las habilidades adaptativas conceptuales, sociales y prácticas, y por ser originada antes de los 18 años”.

Además de la definición, es importante subrayar que según la misma asociación, “la diversidad funcional intelectual afecta a diferentes dimensiones de la persona, no solamente a las capacidades estrictamente intelectuales, sino que además afecta también a la conducta adaptativa, tanto en el campo intelectual como en el ámbito social, o en las habilidades de la vida diaria; a la salud, tanto física como mental; a la participación y las interacciones con los demás o el contexto ambiental y cultural en el que la persona se encuentra incluida”.

Por otra parte, también se clasifica la diversidad funcional intelectual dependiendo del cociente intelectual y de los niveles de apoyo que la persona necesita en su día a día, como leve, moderada, grave o profunda.

Uniendo en la búsqueda, la diversidad funcional intelectual junto con otras palabras como tecnologías de asistencia y cómo no, relacionándolas con los SmartWatch, encontramos diversas publicaciones científicas que abordan estos temas. Todos estos estudios, y por supuesto, las conclusiones a las que han llegado, son de gran ayuda a la hora de buscar requisitos para la elaboración de nuestro trabajo.

Un ejemplo de ello, se puede ver en la publicación **A Wearable Face Recognition System Built into a Smartwatch and the Blind and Low Vision Users** [4] en la cual se habla de la utilización de un SmartWatch para reconocimiento facial de personas, el cual puede ser usado por personas con problemas en la visión. Aunque no es nuestro campo de trabajo, ya que nosotros nos centraremos en diversidad psíquica, tratan un tema que se debe tener muy en cuenta. Los autores le llaman ergonomía, y se basa en qué el SmartWatch es un dispositivo fácilmente usable, pues está con la persona en todo momento, y siendo un dispositivo usado en el día a día cómo es un reloj no llama la atención del resto de personas y sin embargo nos puede ayudar en las tareas que necesitemos.

Otro tema a tener en cuenta, es que cada persona es diferente, y requiere que la información se muestre de una determinada manera, acorde a sus necesidades. En **WearSense: Detecting Autism**

Stereotypic Behaviors through Smartwatches [5] los autores encuentran una serie de patrones que pueden seguir las personas con algún tipo de autismo, y se ayudan de un SmartWatch para determinarlas y de este modo poder ayudar a los cuidadores, médicos, o familiares a resolver o ayudar en las diferentes situaciones en las que se puedan ver afectados.

Este estudio es importante, pues tiene en cuenta que cada persona es diferente y necesita que el SmartWatch reaccione a las características específicas del usuario.

Vivir de un modo más independiente y tener más seguridad en las actividades que realiza la persona con diversidad funcional es uno de los objetivos fundamentales de este proyecto. La publicación **Exploring the Potential of Wearables to Support Employment for People with Mild Cognitive Impairment** [6] se acerca mucho a lo que nosotros queremos desarrollar puesto que trabaja con personas con diversidad intelectual que trabajan o viven en centros especiales y que gracias al uso del SmartWatch pueden recibir información, tanto los cuidadores o familiares, como las personas con diversidad para estar más seguros en sus actividades. Por un lado envía información los cuidadores o familiares ofreciendo la seguridad necesaria, y haciendo así, mejorar la productividad de la persona con diversidad. Y por otro lado, enviando recordatorios las personas usuarias que viven independientemente en un hogar inteligente.

Aunque en este caso el SmartWatch está siendo utilizado para informar a cuidadores y familiares también se observa como es de vital importancia mejorar la productividad de la persona con diversidad funcional y ayudar mediante el uso de recordatorios, a dichas personas que quieren vivir una vida de un modo más independiente.

La autonomía de la persona con diversidad funcional es un tema recurrente en los diferentes artículos que podemos encontrar, y se le dota de vital importancia en las diferentes publicaciones al respecto. Un ejemplo de ello es el trabajo llamado **Sistema de guiado en exteriores para personas con necesidades especiales mediante Smartwatches** [7]. En este artículo se desarrolla una aplicación para el guiado en exteriores a personas con diversidad intelectual mediante el uso de SmartWatch, llamada Assist-Out-Android-Wear.

Para nosotros es sin duda una aplicación muy importante ya que nos habla de la importancia de dotar de autonomía a la propia persona con discapacidad. Assist-Out-Android-Wear no está pensada para qué terceras personas sepan dónde se encuentra en todo momento el usuario con diversidad, ni para que otra persona le envíe mensajes diciéndole lo que tiene que hacer. La aplicación, desde un principio, está pensada para que el propio usuario con diversidad intelectual, consiga por sí mismo realizar un trayecto definido con anterioridad.

En **Emotional Self-Regulation of Individuals with Autism Spectrum Disorders: Smartwatches for Monitoring and Interaction** [8] se estudia como las personas con trastorno del espectro autista (TEA), cuando entran en ciertos estados de shock o estrés, tienen comportamientos o situaciones que son de alguna manera reconocibles mediante el uso de un SmartWatch, el cual además de reconocer

el estado en el que se encuentra el usuario, puede mostrar la información que le ayude a recuperarse de dicho estado y continuar con la actividad que estaba realizando.

Este estudio para nosotros es muy importante ya que una vez más, se basa en ayudar al propio usuario con diversidad funcional a autorregularse emocionalmente sin la ayuda de terceros, ofreciéndole al usuario una mayor autonomía y seguridad en sí mismo.

Para nuestro trabajo, resulta de vital importancia encontrar estudios o aplicaciones, que estén actualmente en funcionamiento, que demuestran que nuestros objetivos pueden ser necesarios para determinadas personas. Un ejemplo claro es el artículo **Using Smartphones to Assist People with Down Syndrome in Their Labour Training and Integration: A Case Study** [9], el cual describe el uso smartphones para ayudar a personas con síndrome de Down en el desempeño de su capacitación e integración laboral. Para ello hace uso de una aplicación llamada AssisT-Task, que mediante el uso de códigos QR y pictogramas ayuda al usuario a desempeñar una tarea de forma autónoma dotándola así de una mayor autonomía personal.

Este artículo, junto con esta aplicación es muy importante para nuestro proyecto ya que reúne todas cualidades que nosotros queremos conseguir, pero en esta ocasión vinculadas a un smartphone. Realizar este trabajo con un SmartWatch facilitará la tarea integradora y aumentará las posibilidades para que las personas con diversidad intelectual pueden realizar sus actividades con mayor autonomía y seguridad en sí mismos.

2.2. Conclusiones

Sin duda alguna, se puede observar como los ámbitos que abarca nuestro trabajo de fin de grado, son campos que actualmente están siendo muy importantes en el ámbito de la investigación y el desarrollo.

Puede verse que en los últimos años, temas como los wereables y los SmartWatch han incrementado sustancialmente el número de artículos que los toman como base para su investigación. Esto puede ser debido al auge de las nuevas tecnologías, sin embargo vemos que otros campos que han estado con nosotros desde siempre cómo pueden ser la discapacidad o las tecnologías de asistencia, se suman en la investigación y el desarrollo provocando todas ellas subidas exponenciales en los últimos años en el número de artículos publicados que abordan estos temas.

Si se profundiza un poco más en las aplicaciones y artículos publicados en la comunidad científica, vemos como por un lado la mayoría de artículos se basan en alertar o informar a terceros de dónde se encuentra o qué está haciendo en todo momento la persona con diversidad, mientras que unos pocos se basan en ofrecer ayudas al propio usuario para que éste pueda ser más autónomo en su día a día.

También podemos ver la importancia del uso de SmartWatch para garantizar la ergonomía del

hardware utilizado. Esto es de vital importancia ya que “normaliza” y minimiza el impacto de utilizar una herramienta para realizar una determinada tarea.

Por todo esto, la idea de realizar una aplicación que permita a las personas con diversidad funcional intelectual llevar a cabo sus tareas de manera más independiente mediante el uso de dispositivos SmartWatch, es una idea novedosa con la que podremos aportar a las personas que lo necesiten una mayor independencia y fiabilidad a la hora de realizar las tareas para las que requieran este tipo de ayuda.

Al ser un dispositivo "vestible" que siempre se lleva encima, y que siempre se tiene a mano, esta ayuda extra que necesitan en su vida diaria, incluirá a las personas con diversidad funcional intelectual dentro del grupo de usuarios del ecosistema de aplicaciones para wearables, lo cual aporta un toque de normalización y de modernización a los usuarios finales.

Debido a todos estos motivos, las personas que utilicen nuestra aplicación podrán sentirse más incluidas en la sociedad, ya que el producto de apoyo que utilizan será un dispositivo novedoso, que apenas tiene impacto visual y que además de conseguir guiarles en sus tareas diarias, les hará sentirse a la última en tecnología pues como hemos visto, los SmartWatches son un producto que lleva relativamente poco tiempo en el mercado, pero que unidos a las necesidades reales que observamos en el campo de la diversidad intelectual conformarán un producto completo y necesario.

DISEÑO

3.1. Patrón de diseño

Modelo-Vista-Controlador

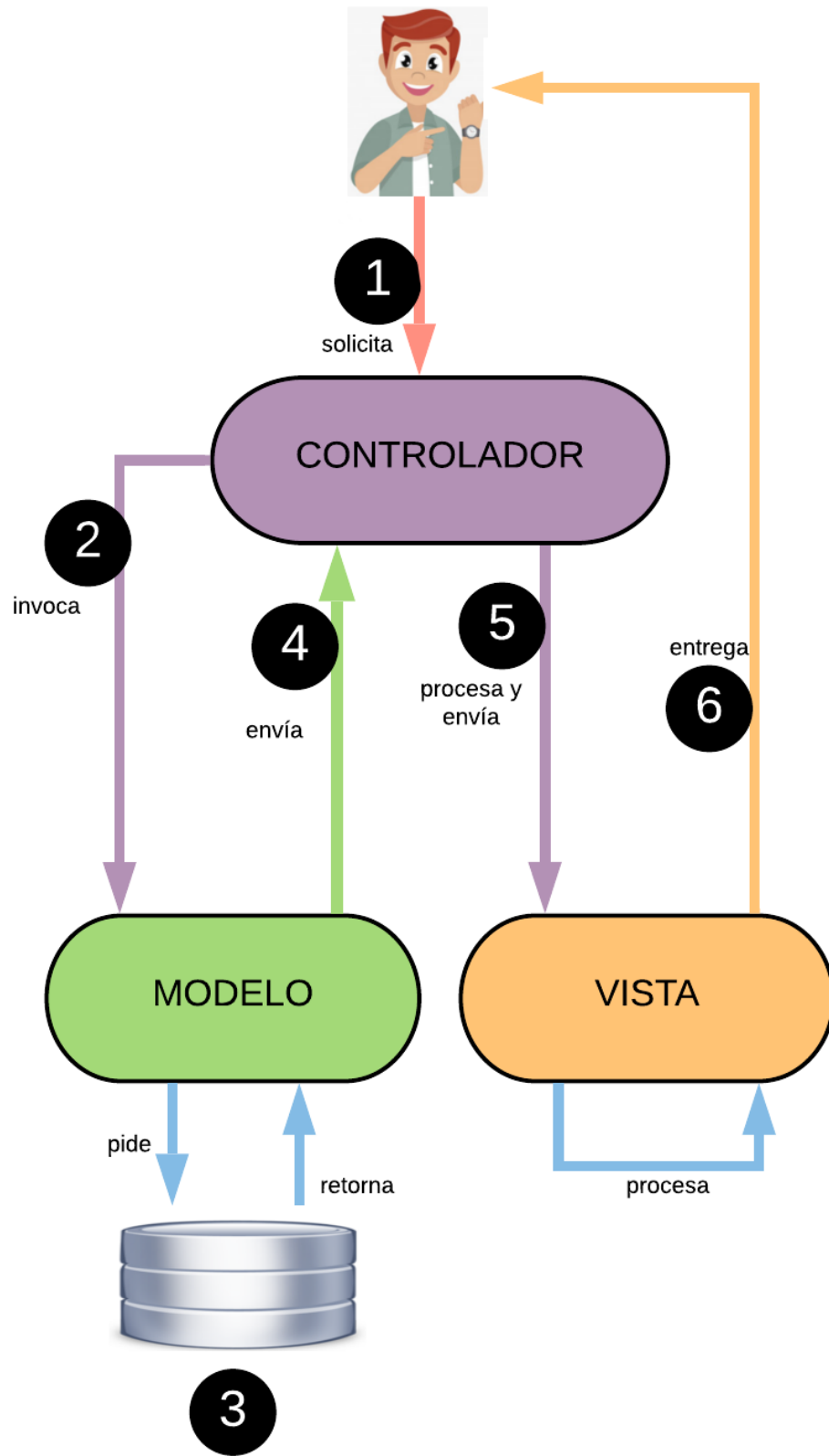
Para diseñar la arquitectura interna de nuestro programa se ha tenido en cuenta el patrón de diseño "modelo vista controlador"(MVC) el cual nos permite estructurar sistemas robustos de forma clara y eficiente.

En la siguiente figura 3.1 se puede observar el funcionamiento de este patrón de diseño, el cual, como su propio nombre indica, consta de tres partes: el modelo, la vista y el controlador. Cada una de ellas se encargan de diferentes aspectos dentro de la arquitectura de nuestro programa. Gracias a ello en el futuro se podrá modificar cada uno de sus componentes sin que se tenga la necesidad de tener que retocar los demás.

- **Modelo:** Este componente es el encargado de conseguir los datos que se necesitan para la ejecución del programa o los que el usuario pide mediante alguna interacción con el mismo. En nuestro caso por ejemplo, al tener una base de datos, es este el componente que se encarga de tenerla actualizada, así como de recoger y gestionar los datos que ella devuelve.
- **Vista:** Este componente es el encargado de mostrarle al usuario los datos que quiere o necesita. Éstos serán colocados en el lugar y formato preciso para su correcta visualización e interacción con ellos. También es conocida esta parte dentro del ámbito informático como frontend.
- **Controlador:** Este componente es el encargado de recibir las instrucciones, ya sean del usuario o de otro tipo de estructura, y procesarlas por medio de la información que recibe del componente "modelo" que hemos estudiado con anterioridad. En nuestro caso, por ejemplo, es el encargado de procesar la información que llega de la vista (tarea seleccionada por el usuario), pedir al modelo la información necesaria y una vez estructurada volver a pasársela a la vista para que la pueda ver y escuchar el usuario.

3.2. Modelo de proceso

En cuanto a la ingeniería de software se refiere, este proyecto será desarrollado utilizando un modelo incremental e iterativo ya que por las peculiaridades del proyecto y lo novedoso del campo en



(a)

Figura 3.1: La imagen nos explica cómo es el funcionamiento y la integración de las diferentes partes que componen el patrón de diseño "modelo vista controlador".

el que se ha de trabajar, será lo más eficiente.

La principal diferencia con otros modelos de desarrollo de software, sin duda alguna, es la parte iterativa de este modelo, es decir, cada vez que se incrementa el diseño y se avanza en la creación del software se podrá, en caso de necesidad, volver a estructurar o modificar pasos o partes realizadas con anterioridad.

Otra de las características de este modelo que tuvimos muy en cuenta, es que con cada incremento que realizábamos al software, siempre se buscará la entrega de un producto funcional, de tal modo que al final se obtenga el proyecto completo.

Las diferentes fases del modelo serán:

- **Análisis de requisitos:** debido al tipo de proyecto del que se trata, el cual es una ampliación/adaptación para un tipo concreto de producto (en este caso un SmartWatch) de una aplicación ya existente (AssisT-Task), los requisitos han sido propuestos por el equipo de trabajo de laboratorio, y se han ido modificando y adaptando, gracias al modelo incremental iterativo que hemos utilizado, a medida que veíamos las carencias y posibilidades que nos brindaba la plataforma para la cual se programa, en este caso Wear Os.
- **Diseño:** La fase de diseño de esta aplicación es relativamente sencilla, puesto que únicamente consta de tres pantallas visibles. La lista de aplicaciones, cada paso de la tarea seleccionada y una pantalla final que nos indica que hemos terminado la tarea con éxito. Lo que es más complicado será discutir qué elementos se mostrarían o no se mostrarían en cada una de estas pantallas así como el tiempo de bloqueo o la forma de interactuar o pasar de unas pantallas a otras. En definitiva, cómo aplicar la lista de requisitos en una pantalla tan pequeña y con unas funciones tan limitadas. Otro punto a tener en cuenta del diseño es el lenguaje que íbamos a utilizar, ya que cabía la opción de utilizar el nuevo lenguaje de desarrollo Kotlin o utilizar Java para Android.
- **Codificación y desarrollo:** Se ha de partir de la base de que la codificación y el desarrollo para un dispositivo con las características de un SmartWatch es totalmente distinto a programar para otros tipos de dispositivos como pueden ser un teléfono móvil inteligente o una tableta. Se necesitará cambiar u omitir funcionalidades, y así mismo crear otras nuevas que se necesitarán para que todo funcione correctamente en un dispositivo de tan reducidas características, tanto de tamaño como de potencia interna. Será determinante el correcto tratamiento de las imágenes para que funcionen correctamente y no sobrecarguen la memoria y bloqueen el dispositivo. También habrá que tener en cuenta las nuevas formas de interactuar con la pantalla del SmartWatch, que son propias de este tipo de dispositivos, por ejemplo para realizar el desplazamiento de un paso al siguiente dentro de una actividad, se deberá evitar en todo momento el uso de botones, utilizando gestos (desplazamientos con el dedo por la pantalla) para tal fin.
- **Pruebas e integración:** debido a la utilización de un sistema incremental e iterativo, y la necesidad de que en cada incremento tuviésemos al finalizar un producto funcional, las pruebas se realizarán unitariamente y paso a paso con cada nueva actualización que se vaya creando. Esto hace que nuestro programa resultante siempre sea útil y completo respecto a los requisitos que vamos completando.
- **Mantenimiento:** debido a las características de este proyecto, y ya que no es un producto final, si no que ha de ser un producto para probar con usuarios y posiblemente cambiar, mejorar o incluir nuevas funcionalidades, la fase de mantenimiento no es motivo de este estudio.

3.3. Requisitos

Requisitos funcionales

- Mostrar lista de tareas. En la pantalla principal se muestra una lista con todas las tareas disponibles en la base de datos.
- Mostrar los pasos de la tarea seleccionada mediante una imagen o pictograma, con un texto y si el SmartWatch lo permite mediante audio.
- Mostrar una pantalla de finalización de una tarea. Al completar todos los pasos de una tarea se le mostrará al usuario una pantalla que indique que ha terminado y posteriormente se regresará a la lista de tareas.

Requisitos no funcionales

- El dispositivo debe ser autónomo. no debe necesitar la presencia del móvil para funcionar con total normalidad. Se podría conectar a redes WiFi o 3G dependiendo del dispositivo y de la presencia o no de dichas redes, el dispositivo podrá actualizar su base de datos a través de alguno de estos métodos, sin requerir estar conectado al teléfono.
- La información descargada de Internet se guardará en una base de datos dentro del propio dispositivo Smart-Watch para que esté disponible en caso de necesitar usarla y no funcionar correctamente y la conexión a Internet
- El programa ha de ser completamente accesible y adaptado a los usuarios finales. La información mostrada de cada tarea ha de poder estar adaptada al usuario final mostrando la información mediante audio, texto y fotografía o pictograma según sean los requerimientos del usuario.
- La navegación para el usuario ha de ser sencilla. Se utilizarán únicamente gestos en la pantalla, deslizamientos del dedo por ésta, hacia delante o hacia atrás para avanzar o retroceder en la tarea.
- Durante la ejecución de una tarea se ha de eliminar el gesto nativo de WearOs de deslizar el dedo de izquierda a derecha para salir a la pantalla anterior. Una vez dentro de la tarea, este gesto, únicamente mostrará la anterior fase de la tarea del mismo modo que el gesto contrario (deslizar el dedo de derecha a izquierda) mostrará la siguiente fase de la tarea a realizar.
- Toda tarea seleccionada puede tener un número alto de pasos para completar dicha tarea. Cada paso llevará una foto o pictograma asociada, esto no debe ser problema y nuestro dispositivo no debe de llenar su memoria y colapsar por muchas fotos que tengamos que mostrar por pantalla.
- Se ha de esperar un tiempo prudencial para poder pasar hacia delante o hacia atrás en las tareas de cada actividad. De este modo evitamos que el usuario, ya sea por error o por demasiada efusividad a la hora de tocar la pantalla, se pase por alto algún paso o tarea de la actividad que desea realizar.

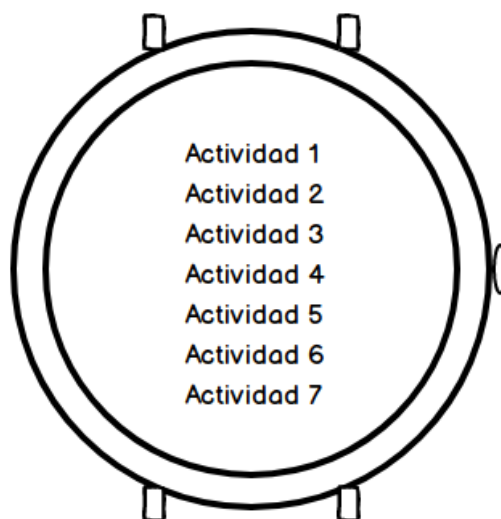
3.4. Maquetas

La aplicación de Assist-Task Wear para SmartWatch que se implementará en este estudio, estará instalada junto con el resto de aplicaciones que tengan reloj dentro de la lista de aplicaciones, y constará de su logo característico, para que sea más fácilmente detectable. Además, contaremos con la implementación propia del sistema operativo Wear os, de las complicaciones. Las complicaciones son elementos de los que consta la esfera o pantalla principal de nuestro dispositivo, los cuales son accesos directos que muestran datos relevantes, o directamente, lanzan aplicaciones, para que estén más

accesibles para ser consultadas o acceder a ellas. Por lo tanto lo ideal es tener nuestra aplicación en una de estas complicaciones, para que con un simple toque en la pantalla principal, se nos muestre por completo nuestro programa, siendo este, un modo sumamente sencillo, visual y fácilmente recordable, para poner en marcha la aplicación.

Una vez dentro, se mostrará una lista con todas las tareas disponibles en la base de datos de forma aproximada como se muestra en la figura 3.2. Como el reloj tiene una pantalla relativamente pequeña, será necesario en la mayoría de los casos, hacer scroll para ver la lista completa y buscar la tarea concreta que se necesite realizar.

El scroll se hace de manera intuitiva apoyando el dedo en la pantalla y deslizando hacia arriba o hacia abajo, pudiendo así buscar las veces que haga falta la tarea que estamos buscando.

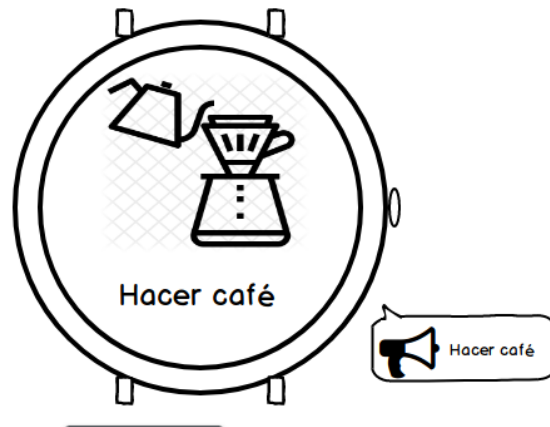


(a)

Figura 3.2: La figura nos muestra como nuestro SmartWatch nos mostraría una lista con las actividades a las que nos puede ayudar.

Cuando se encuentre la tarea deseada, se seleccionará pulsándola con el dedo. De este modo se accede a la primera parte de la tarea requerida, mostrándose así el primer paso. Se verá en la pantalla un dibujo que nuestro usuario reconocerá. Los dibujos pueden ser pictogramas o fotografías reales, ya que la base de datos contendrá estas imágenes dependiendo de las necesidades que exija el cliente, adaptándose así a cada tipo de usuario y a cada necesidad específica.

Nuestra aplicación, mostrará por pantalla, la imagen o el pictograma, que será el primer paso de la tarea que se desea realizar de forma similar a lo que se puede ver en la maqueta 3.3. A su vez, se muestra debajo de ésta un texto que dirá lo que significa dicha imagen. En el caso de que el reloj lo permitiese (actualmente lo son la mayoría), el altavoz de nuestro reloj, reproducirá de viva voz el mismo texto que describe la imagen.

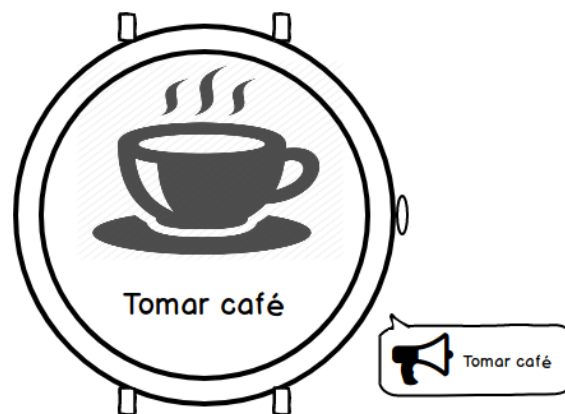


(a)

Figura 3.3: La figura nos muestra como nuestro SmartWatch nos presenta el primer paso de la tarea que queremos realizar. Además, si es posible nos lo indicaría mediante un mensaje de voz.

A continuación de esta primera pantalla viene el resto de pantallas o de pasos de cada tarea, el cual es un número variable, ya que depende de la tarea en sí.

Para pasar al siguiente paso, el usuario, de un modo natural, debe deslizar la tarea hacia atrás, esto es apoyar el dedo en la pantalla y deslizarlo de derecha a izquierda. Este gesto, ha sido elegido, pues aparenta dejar atrás la acción ya realizada, y traer a la pantalla la nueva acción que se ha de realizar. Este paso se intenta reflejar en la figura 3.4



(a)

Figura 3.4: La figura nos muestra como nuestro SmartWatch nos mostraría el siguiente paso que debemos realizar para llevar a cabo nuestra tarea.

Para evitar posibles confusiones, y que no se pasen varios pasos a la vez sin querer, hay un tiempo prudencial que esperar para poder pasar de una acción a otra.

Una vez que estemos en cualquiera de los pasos de nuestra actividad, se podrá navegar por estos

hacia delante o hacia atrás. El gesto de navegar hacia delante, hacia la siguiente diapositiva, ya lo hemos explicado con anterioridad, es deslizar el dedo de derecha a izquierda. Por consiguiente si queremos ir hacia atrás, el gesto que debemos hacer es deslizar el dedo de izquierda a derecha, excepto en la primera pantalla que no sucedería nada, ya que es el primer paso. Esto lo hemos implementado por si el usuario necesita volver a un paso anterior por cualquier causa, por ejemplo si no se acuerda si lo ha realizado o no, de este modo podrá asegurarse.

En Wear Os el gesto de deslizar el dedo de izquierda a derecha, por norma general nos saca del programa que estemos ejecutando y nos muestra la pantalla principal. Para evitar esto, y que el usuario pueda navegar con tranquilidad y no se sienta inseguro por haber perdido el programa, mientras se esté inmerso en alguna actividad, este gesto nativo de Wear Os lo tendremos desactivado, y como hemos explicado servirá para navegar por los pasos de la aplicación. En la única pantalla que este gesto estará activo y funcionará para salir de la aplicación, es en la pantalla principal, o sea, en la lista de tareas. De este modo, deslizando el dedo en la lista de tareas de izquierda a derecha, saldremos de la aplicación y se mostrará la esfera del reloj.

Una vez finalizados todos los pasos de la actividad seleccionada, se mostrará una pantalla final que nos indicará, que la tarea que queríamos realizar no tiene más pasos, y por lo tanto la damos por finalizada, figura 3.5. En la pantalla se mostrará una imagen y un texto que simbolizan ese “fin de la actividad” y pulsando en ella con el dedo, el programa volverá a su pantalla principal con la lista de todas las actividades, por si queremos realizar ésta o cualquier otra, una vez más.



(a)

Figura 3.5: La figura muestra como nuestro SmartWatch nos mostraría una pantalla que indicaría que hemos terminado con éxito nuestra tarea.

DESARROLLO

La fase de codificación comienza con la elección del sistema de la arquitectura para la que se ha de programar, así como el lenguaje que utilizaremos para la realización en este programa.

Como ya se ha dejado claro en la fase de diseño utilizaremos la plataforma para dispositivos wearable de Android, cuya base es el sistema operativo Wear Os. Esta elección ha sido tomada por dos motivos fundamentales, la primera es la expansión del sistema operativo Android, la cual es lo suficientemente amplia como para llegar a un amplio abanico de usuarios. En este aspecto también se ha tenido en cuenta el amplio abanico dispositivos SmartWatch compatibles con esta plataforma, los cuales a su vez, garantizan también que el usuario pueda optar por el modelo de dispositivo que más se ajuste a sus necesidades tanto de diseño (por ejemplo una pantalla más o menos grande), como económicas.

Por otro lado nos encontramos con la elección del lenguaje de programación que utilizaremos. Para Wear Os podemos programar en el lenguaje Java, o en Kotlin, que pese a que corre sobre la máquina virtual de Java tiene la ventaja de que también puede ser compilado a código fuente de JavaScript.

En nuestro caso partimos con que el laboratorio AMILab de la Escuela Politécnica Superior de la Universidad Autónoma de Madrid ya tiene en funcionamiento un programa parecido al que vamos a desarrollar, pero esta vez orientado al uso en teléfonos móviles o tabletas.

Debido a que ambos productos puedan requerir actualizaciones en un futuro, se ha decidido la utilización de Java como lenguaje de programación, para así unificar ambos desarrollos y facilitar las tareas de mejoras que se puedan realizar más adelante.

Por todos estos motivos, se ha utilizado Android Studio como entorno de desarrollo para la realización del proyecto. Por supuesto teniendo en cuenta que se va a desarrollar para un dispositivo wearable, con las características propias de Wear Os.

Para las pruebas y los debugs de cada una de las fases se ha utilizado un SmartWatch Huawei Watch 2, el cuál es plenamente compatible con nuestros requerimientos.

4.1. Base de datos

AssisT-Task Wear no es un programa autónomo que se instala en el SmartWatch y funciona siempre igual, sino que es un programa conectado. Requiere de una conexión a Internet para descargarse una base de datos que ha de estar siempre actualizada y en la cual encuentra todas las actividades y toda la información que el usuario pueda requerir.

En este apartado, centrándonos en la comunicación entre el dispositivo wearable y el servidor que contiene la base de datos, podemos decir que ésta está basada en un sistema REST (Representational State Transfer) el cual utiliza directamente el protocolo HTTP para obtener los datos del servidor con el requerimiento de identificación de usuario que ha de estar debidamente registrado en la base de datos.

A través del protocolo HTTP podemos obtener la información mediante diversos tipos de objetos, los más conocidos son XML y JSON. Debido a las características del cliente, el cual es un dispositivo con bajos recursos hardware, se ha utilizado JSON, pues este ofrece una mayor ligereza a la hora de la codificación y la descodificación de los objetos recibidos, que en nuestro caso sería la base de datos principalmente.

El funcionamiento interno de nuestro programa, se basa en descargar la base de datos del servidor y posteriormente utilizarla para recuperar de ahí toda la información que se requiera mostrar por pantalla.

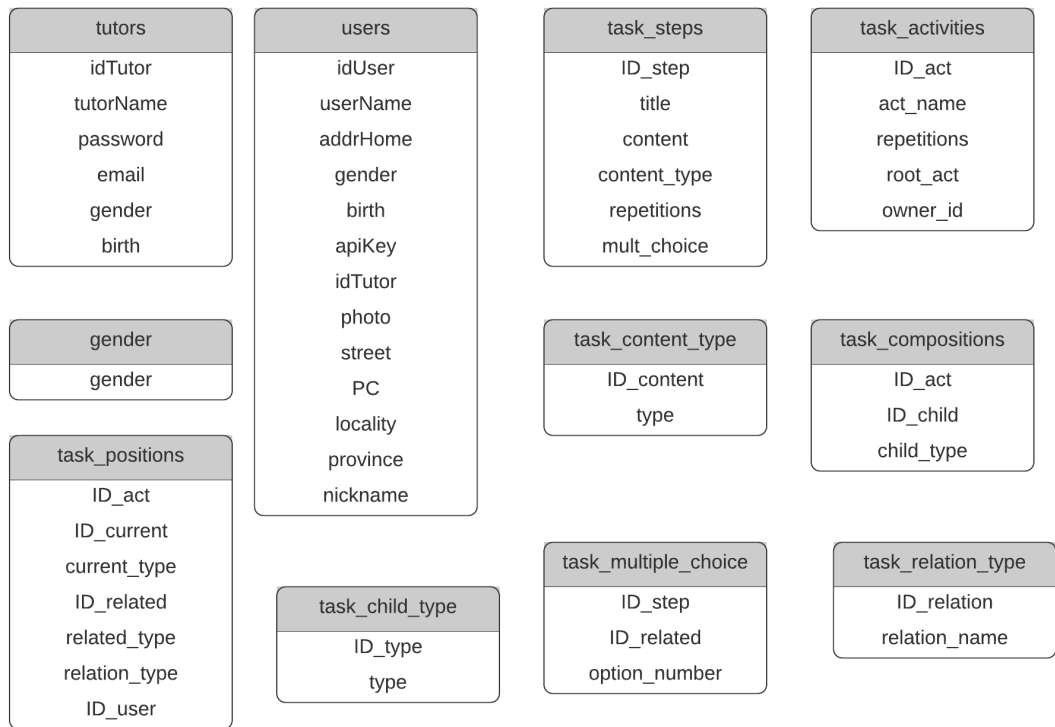
Las tablas y los campos de la base de datos son los que se representan en la figura 4.1.

Teniendo esta copia de la base de datos en memoria nuestro programa es mucho más independiente, pues no requiere continuamente de una conexión a Internet, únicamente sería necesario que estuviese conectado cuando haya algún cambio en la base de datos para actualizarla.

4.2. Permisos

Uno de los archivos más importantes de cualquier aplicación Android, sea del tipo que sea, es el AndroidManifest.xml. En el AndroidManifest se incluye información de vital importancia para el correcto funcionamiento de nuestra aplicación, como son el nombre (en nuestro caso AssisT-Task Wear), el icono que la representa, un listado de todas las actividades de las que consta así como sus propiedades, estilos, características o preferencias y algo muy importante cómo son los permisos que el sistema operativo y/o el usuario deben de concederle a nuestra aplicación para que funcionen correctamente todas sus actividades programadas.

En nuestra aplicación, como se muestra en la figura 4.2, lo primero que se muestra es el tipo de hardware para el que está diseñada, en este caso para un reloj. Este dato es importante ya que se



(a)

Figura 4.1: La figura nos muestra las tablas y los diferentes campos que necesita la base de datos de AssisT-Task Wear para funcionar correctamente.

```

<uses-feature android:name="android.hardware.type.watch" />

<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.VIBRATE" />
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_LOGS" />

```

(a)

Figura 4.2: La figura nos muestra una captura de todos los permisos que necesita AssisT-Task Wear para funcionar correctamente.

tiene en cuenta no solamente para la compilación, sino que también se tiene muy en cuenta en la Play Store de Google, la cual podrá poner nuestra aplicación disponible para que se pueda descargar desde cualquier dispositivo wearable que acceda a ella.

Posteriormente se muestran todos los permisos que se necesitan. Los cuatro primeros, en nuestro caso, están relacionados con las redes de nuestro dispositivo.

INTERNET

ACCESS NETWORK STATE

ACCESS WIFI STATE

READ PHONE STATE

Tanto el permiso para acceder a Internet, como los permisos para conocer el estado de las redes Wi-Fi o el estado de la red del teléfono (en nuestro caso para dispositivos que tengan acceso a la red telefónica GPRS), son utilizados posteriormente en las diferentes actividades para acceder Internet y poder conocer con antelación el estado de conectado o no, de nuestro SmartWatch.

VIBRATE

El poder tener acceso al vibrador de nuestro smartwatch es necesario, puesto que en caso de que nuestro dispositivo no tenga altavoz, al entrar en una actividad o cambiar a otra, el reloj admitirá una respuesta háptica mediante vibración para que el usuario sea consciente de que se habrá realizado algún tipo de movimiento o cambio en la información de la pantalla.

WAKE LOCK

Este permiso es bastante importante, ya que permite que nuestra aplicación no se cierre y permanezca activa en primer plano de la pantalla, aunque el dispositivo SmartWatch apague ésta por ahorro de batería, en el caso de que el usuario esté un tiempo prolongado sin tocar o interactuar con la pantalla.

READ EXTERNAL STORAGE

WRITE EXTERNAL STORAGE

Una de las características de nuestro programa es que si tiene conexión a Internet, se conectará para descargar la base de datos y actualizarla, pero si en ese momento no existe conexión utilizará la base de datos guardada como copia que tiene en memoria. Además una vez que descarga la base de datos de Internet la guarda en el dispositivo para poder acceder a todos los datos ya que el sistema operativo Wear Os es compatible con una versión de SQL, llamada SQL Lite, y nuestro programa ha de poder acceder a ella, mediante sentencias SQL para recuperar, y posteriormente mostrar la información requerida. Para ello necesita estas dos sentencias, para tener permisos para poder acceder a la memoria el dispositivo, y de este modo poder tanto leer cómo escribir en dicha base

de datos.

READ LOGS

Este último permiso se usa para poder acceder a los logs del sistema, en los que nuestro programa va guardando determinadas partes que el usuario va dando a lo largo del uso de la aplicación.

4.3. Codificación

La programación en este proyecto, ha estado basada principalmente en crear el programa haciendo cumplir los requisitos, tanto funcionales, como no funcionales, sin perder la funcionalidad básica para la que se ha desarrollado.

Como se puede observar cuando se analizan los requisitos, estos están basados, en su gran mayoría, en el dispositivo para el que se está programando, esto es, para un SmartWatch.

El resto de la programación, está orientada al funcionamiento interno del programa, ya sea la gestión de la base de datos, la colocación de los distintos elementos para que el usuario pueda visualizar los datos de la mejor manera posible, etcétera.

Por lo tanto vamos a empezar a desglosar y a explicar algunas partes del código para que podamos entender mejor el funcionamiento interno de Assist-Task Wear.

Un punto importante a tener en cuenta de la aplicación, es que ha de ser una aplicación independiente, esto es, que no dependa de ninguna otra que tenga que estar instalada en algún otro dispositivo, véase tableta o teléfono móvil. Una de las grandes ventajas de un SmartWatch, es que una vez que te lo colocas en la muñeca, te sigue a todas partes, y cuando se necesita, ha de estar disponible para darte la información requerida.

Para conseguir este objetivo, nuestro programa AssistTask Wear comprueba si de alguna manera está conectado a la red, y todo depende del número de conexiones que tiene nuestro dispositivo. En la actualidad la gran mayoría de dispositivos cuenta con Bluetooth y con Wi-Fi pero también hay ciertos modelos que cuentan con conexión GPRS, esto es conexión telefónica 3G o 4G.

Como podemos ver en la figura 4.3, si el programa detecta que tiene algún tipo de conexión activa, se conectará a la base de datos del servidor para descargar una copia actualizada de ésta. En caso contrario utilizará la base de datos que tiene ya guardada en la memoria del dispositivo.

Este procedimiento, podemos observarlo en la figura siguiente 4.4 en la cual vemos que tienen cuenta tanto que esté conectado el SmartWatch a Internet, cómo que la variable guardada en preferencias del sistema, llamada "isUpdateableOnStart". En el caso de que estas dos variables sean ciertas, cierra la base de datos anterior lanza una nueva tarea de descarga, mientras que cualquiera de estas dos variables es negativa, lógicamente, mantiene la base de datos anterior guardada en la

```

public static boolean checkConex(Context ctx){
    ConnectivityManager connec = (ConnectivityManager)ctx.getSystemService(Context.CONNECTIVITY_SERVICE);

    /* Con esto recogemos todas las redes que tiene el móvil (wifi, gprs...) */
    NetworkInfo[] redes = connec.getAllNetworkInfo();

    for(int i=0; i<redes.length; i++){
        if (redes[i].getState() == NetworkInfo.State.CONNECTED){
            return true;
        }
    }

    return false;
}

```

(a)

Figura 4.3: La imagen nos muestra un fragmento de código en el que puede observarse como nuestro programa comprueba si está o no conectado a algún tipo de red.

memoria y comienza a trabajar con ésta.

```

public StepInfoProvider(Context context, boolean online){
    this.context = context;

    if(online && Preferences.isUpdateableOnStart(context)){
        clear();
        Preferences.setisUpdateableOnStart( autoupdate: false, context);
        new RetrievalTask(context).execute();
    } else {
        /* Modo offline: abrir base de datos precargada */
        init();
        ((BaseActivity) context).providerCallback();
    }
}

```

(a)

Figura 4.4: La imagen nos muestra un fragmento de código en el que puede observarse como el programa intenta descargar una base de datos actualizada, y si no es así, abrir la copia de la base de datos precargada que tiene memoria.

Otro punto importante que se ha tenido en cuenta a la hora de programar nuestra aplicación, es que los SmartWatch no tienen un hardware demasiado potente, con lo que corremos el riesgo de bloquear nuestro dispositivo si hacemos que cargue demasiada información en memoria principal. Además este proyecto se ha diseñado para Android porque es la plataforma, en la actualidad, más flexible para que cada usuario pueda adquirir el reloj que más se adapte a sus necesidades. Esto es, habrá usuarios que puedan adquirir un SmartWatch último modelo, lo suficientemente potente como para mover cualquier tipo de aplicación, mientras que otros puede que tengan que utilizar un modelo un poco más modesto. Nuestro software ha de funcionar correctamente en todos los relojes.

Por todo esto se han utilizado estructuras de tipo RecyclerView. Estas estructuras han sido

diseñadas para que se pueda mostrar un número indeterminado de elementos, y únicamente cargarán en la memoria del terminal los datos que se muestran actualmente en la pantalla. Cuando el usuario quiera ver más elementos de la lista, haciendo scroll con el dedo en la pantalla, se eliminarán los objetos que ya no sean visibles para dar cabida a los nuevos elementos que se quieren mostrar, de este modo, aunque la lista tenga cien elementos, únicamente se utilizará memoria cinco elementos que se ven en la pantalla.

```
public class RecyclerViewAdapter extends RecyclerView.Adapter<RecyclerViewAdapter.MyViewHolder> {
    private ArrayList<String> items;
    private Context context;

    public RecyclerViewAdapter(Context context, ArrayList<String> items) {
        this.context = context;
        this.items = items;
    }

    @Override
    public MyViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        View itemView = LayoutInflater.
            from(parent.getContext()).
            inflate(R.layout.custom_hcc, parent, attachToRoot: false);

        MyViewHolder holder = new MyViewHolder(itemView);

        return holder;
    }
}
```

(a)

Figura 4.5: La imagen nos muestra un fragmento de código que mediante el uso de RecyclerViewAdapter ayudamos a que toda la información que cargamos en memoria para poder visualizarla e interactuar con ella, no saturé nuestro SmartWatch.

Esto, se realiza, como se ha dicho anteriormente mediante el uso de RecyclerViewAdapter que se implementa como se muestra en la figura 4.5. El cual mediante la sentencia “inflate” muestra por pantalla un layout del tipo WearableRecyclerView (figura 4.6), que adapta la lista a la pantalla de nuestro wearable. Además, esta estructura, cómo curiosidad, si nuestro reloj es redondo, muestra la lista curvada adaptándose a la pantalla y haciendo la más cómoda de ver y manejar.

Una vez seleccionada la actividad que se desea realizar, se mostrará por pantalla el primer paso. A partir de aquí, el usuario ha de interactuar con el programa para poder avanzar o retroceder por las distintas partes o tareas de cada actividad. Debido principalmente, a la pequeña pantalla de los SmartWatch, los botones quedan prácticamente desechados para tal fin. Por ello, el sistema operativo Wear Os pone a nuestra disposición los gestos en pantalla, mediante los cuales, deslizando con el dedo por la pantalla, hacia un lado o hacia otro de ésta, podemos elegir que se realice la acción correspondiente.

Se ha llegado a la conclusión de que la forma más intuitiva para avanzar hacia delante, a la siguiente tarea, es desplazar el dedo horizontalmente, de derecha a izquierda de la pantalla. Del mismo modo si

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <android.support.wear.widget.WearableRecyclerView
        xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:tools="http://schemas.android.com/tools"
        android:id="@+id/wrv"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:scrollbars="vertical" />

</RelativeLayout>

```

(a)

Figura 4.6: La imagen nos muestra un fragmento de código con el que conseguimos utilizar una lista para ser mostrada en nuestro dispositivos wearable.

lo que se quiere es volver a una actividad anterior desplazáramos de izquierda a derecha el dedo por la pantalla de nuestro reloj.

Cómo se muestra en la figura 4.7, los gestos que realizamos en la pantalla se detectan mediante la implementación de la tarea “GestureDetector”, la cual además de detectar un toque, un toque prolongado, etcétera; detecta también si tocamos y arrastramos con el dedo, y en qué eje de coordenadas lo estamos haciendo, esto es, en el eje vertical o en el eje horizontal de la pantalla. De este modo hacemos que si detecta un movimiento en horizontal, de izquierda a derecha, retroceda a una actividad anterior, mientras que si el movimiento es de derecha a izquierda, se muestre una actividad posterior, incluso podemos decirle que si ya no hay más actividades, dado que la actual es la última, nos muestre en pantalla que hemos acabado de realizar tarea correctamente.

La utilización del gesto desplazar el dedo por la pantalla de izquierda a derecha, choca directamente con el sistema operativo, el cual tiene este gesto de uso propio, y lo utiliza para cerrar la aplicación que estamos utilizando, o el layout o fragmento que estemos visualizando en cada momento. Para evitar que el sistema operativo realice la operación que tiene predeterminada, se ha creado un estilo nuevo, con el atributo “android:windowSwipeToDismiss” con valor false, tal y como lo muestra la figura 4.8. Posteriormente se debe indicar, utilizando este nuevo estilo, en cada layout que queramos que no se cierre con el gesto de deslizar el dedo izquierda derecha. De este modo el gesto queda libre para poder utilizarlo en nuestro programa como nosotros nos convenga, en nuestro caso, volviendo a mostrar la tarea anterior para que el usuario pueda volver a visualizarla en la pantalla.

Respecto a la accesibilidad de nuestro programa, siempre se ha buscado que sea la máxima posible. En la actualidad, la mayoría de SmartWatch con sistema operativo Wear Os disponen de altavoz, hemos querido hacer uso este elemento hardware para que a la vez que se muestra una imagen por

```

mDetector = new GestureDetector( context: this, new GestureDetector.SimpleOnGestureListener() {

    @Override
    public boolean onFling(MotionEvent e1, MotionEvent e2,
                          float velocityX, float velocityY) {
        float sensitivity = 10;

        if ((e1.getX() - e2.getX()) > sensitivity) {
            if (_next.isClickable()){
                if(!setFinishButton){
                    new GoNextTask(context).execute();
                }
                /* Si es el último paso */
            }else{
                startActivity(new Intent(context, FinishActivity.class));
                finish();
            }
        }
        else if ((e2.getX() - e1.getX()) > sensitivity) {
            if (_prev.isClickable()&&!( _currentStepIndex == 0 || ( _currentStepIndex == -1 && _record.size() == 1))){
                setFinishButton = false;
                new GoPreviousTask(context).execute();
            }
        }
    }

    return true;
}
}

```

(a)

Figura 4.7: La imagen no se muestra un fragmento de qué es el responsable de capturar los movimientos que realizamos con nuestro dedo en la pantalla del dispositivo para avanzar o retroceder de un paso a otro en las distintas actividades.

```

<?xml version="1.0" encoding="utf-8" ?>
<resources>

    <style name="AppThemeFullscreen" parent="android:Theme.Holo.Light.NoActionBar.Fullscreen" />
    <style name="AppTheme" parent="@android:style/Theme.DeviceDefault">
        <item name="android:windowSwipeToDismiss">false</item>
    </style>
</resources>

```

(a)

Figura 4.8: La imagen nos muestra un fragmento de código gracias al cual podemos evitar que al deslizar el dedo de izquierda a derecha por la pantalla se cierre la aplicación que tenemos abierta en este momento.

```

if(mTts!=null && Preferences.enableVoice(context)){

    String text = Html.fromHtml(b.getString( key: "Title")).toString();
    mTts.speak(text, TextToSpeech.QUEUE_ADD, params: null);

}

```

(a)

Figura 4.9: La imagen nos muestra un fragmento del código gracias al cual podemos escuchar por el altavoz de nuestro dispositivo SmartWatch una locución de lo que se ve en pantalla.

pantalla, y un texto descriptivo debajo de ésta, también se pueda locutar ese texto por a través del altavoz del propio dispositivo, para que el usuario pueda hacer uso del elemento que mejor se adapte a sus necesidades en cada momento, para llevar a cabo con éxito la tarea. El Código que incluimos en nuestra activity es el que se muestra en la figura 4.9 y el fragmento de audio que se escucha es una transcripción del texto que aparece en pantalla y se reproduce cuando el usuario da paso a una nueva tarea, así como cuando se da un toque con el dedo sobre la imagen de la tarea a desarrollar, para poder escuchar la locución las veces que el usuario estime necesarias.

4.4. Pruebas y resultados

Durante el desarrollo de toda la aplicación se han realizando pruebas individuales para cada módulo que se iba creando, y únicamente, cuando todas estas pruebas ofrecían el resultado correcto se daba por cerrada esta fase de desarrollo y se podía avanzar a la siguiente fase. Esta es una de las premisas de la forma de trabajar que se explicó con anterioridad en el modelo de proceso. El modelo utilizado ha sido el Modelo Incremental Iterativo, el cual no solo permite hacer pruebas individuales dentro de la fase de desarrollo, sino que además, si una prueba da un resultado que no es coherente con lo que se espera, se debe volver a fases anteriormente diseñadas para que el resultado sea el correcto, sin esperar a tener terminado todo el conjunto de tareas a desarrollar.

Por esta popularidad del modelo de proceso las pruebas han sido puntuales y constantes a lo largo de todo el desarrollo de la aplicación, pero también se han hecho pruebas al conjunto de la aplicación una vez terminada.

Una de las pruebas realizadas en medio del desarrollo de una fase, y que es importante y un buen ejemplo como para remarcarlo en este apartado, ha sido la utilización de diferentes tipos de imágenes para mostrar en cada uno de los pasos de una actividad. Se probaron tanto imágenes pequeñas como grandes, tanto en tamaño como en densidad, con diferentes tipos de formato (.jpg, .bmp, .png) y por supuesto se probaron imágenes y dibujos o pictogramas.

El comportamiento del programa es el esperado con cualquiera de las opciones. Independientemente del formato o el tamaño de la imagen, ésta se muestra redimensionada al tamaño que se le ha asignado. La fluidez en el cambio a otra imagen siempre ha sido correcta. Pero debido al pequeño tamaño de la pantalla, no podemos olvidarnos que utilizamos un dispositivo SmartWatch, se ven más claramente los dibujos o pictogramas. Será el usuario final el que decida qué tipo de imágenes se adaptan mejor a sus necesidades pero lo importante es que el programa funcione correctamente con cualquiera.

Otra prueba importante que se ha realizado, esta vez al finalizar el desarrollo completo del programa ha sido la conexión a Internet para la descarga de la base de datos.

La primera vez que se instala AssisT-Task Wear, necesita estar conectado a Internet para descargar por primera vez la base de datos desde el servidor. Posteriormente ya no hace falta tener conexión para poder utilizar la aplicación, puesto que puede utilizar la copia de la base de datos que guarda en memoria.

Es recomendable, que para descargar la base de datos, el reloj esté conectado una fuente de datos rápido. No se han encontrado problemas tanto con el wi-fi como con la conexión telefónica GPRS (se recomienda usar Wi-Fi puesto que al tener que descargar imágenes la conexión telefónica puede verse interrumpida al exceder el límite de datos contratado). Por otro lado si lo que disponemos es conexión Bluetooth, utilizando la conexión de datos a través del teléfono móvil, la descarga de la información es lenta y puede verse cortada en diversas ocasiones sin haber terminado de descargar todos los datos correctamente.

También se ha comprobado que en el caso de que el usuario intente pasar rápidamente de un paso de una actividad a otro, el reloj no responde, ya que toma un tiempo para poder realizar dicho cambio. Esto está hecho así evitando que el usuario pase varias veces seguidas la actividad sin llegar a ver la información necesaria y permitiendo así que realice todos los pasos sin que ninguno de ellos pase advertido.

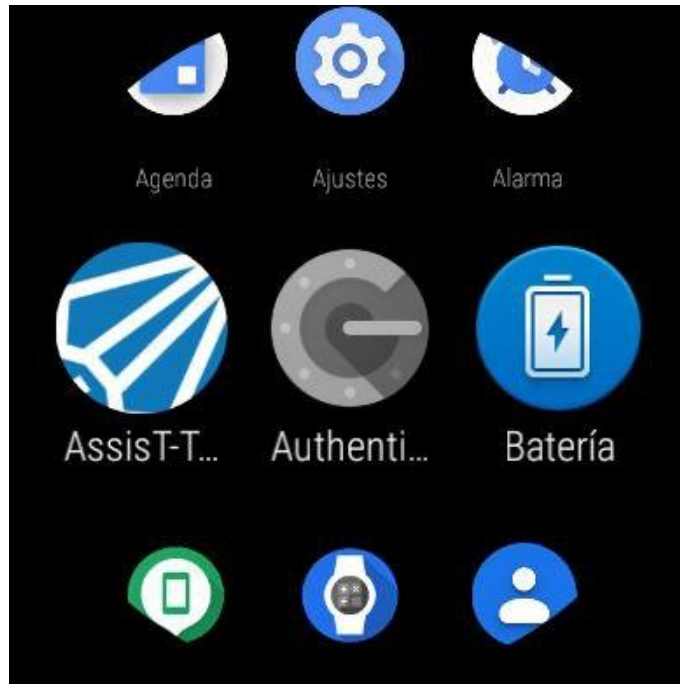
En alguna ocasión, cuando se desliza el dedo por la pantalla para pasar a la siguiente actividad no se reproduce el audio con el contenido o explicación de la actividad. Para ello, se programó que también se emita el sonido al tocar con el dedo en la imagen de la pantalla. Esto también nos sirven para poder escuchar el contenido en cualquier momento, por si el usuario no lo ha podido entender.

4.5. Interfaz de usuario

En este apartado se muestra como es el funcionamiento en nuestro programa AssisT-Task Wear, mostrando capturas de pantalla y explicando paso a paso lo que el usuario puede observar en ella, y como ha de interactuar con el SmartWatch en cada ocasión.

Partiremos de la base de que el usuario ya tiene instalada la aplicación en su reloj inteligente. Partiendo de esta premisa, hay dos formas de acceder a ella. La primera es entrando en el menú de aplicaciones del dispositivo, la manera común de hacerlo es pulsando alguno de los botones físicos que tiene nuestro reloj inteligente, lo que dependerá normalmente del tipo de dispositivo que usemos. Se nos mostrará una lista con todas las aplicaciones instaladas de las cuales puede hacer uso el usuario, se buscará AssisT-Task Wear y se pulsará sobre ella con el dedo. 4.10.

La otra forma de acceder a nuestra aplicación sería utilizando una de las características del sistema Wear Os que son las llamadas complicaciones 4.11. Las complicaciones no son más que accesos directos en la esfera del reloj, en los que entre otras cosas, podemos colocarle un acceso a nuestra



(a)

Figura 4.10: En la imagen podemos observar el menú de aplicaciones de nuestro dispositivo Smart-Watch en el que podemos observar que está instalada nuestra aplicación Assist-Task Wear. Se debe pulsar sobre ella para abrirla y poder usarla.

aplicación, de tal modo que desde la pantalla principal del reloj, y simplemente dando un toque en la pantalla se abriría AssisTask-Wear y el usuario podría utilizarlo de la manera más sencilla posible.

Es necesario, al menos al abrir el programa por primera vez, que el SmartWatch esté conectado a internet, para que este pueda descargar la base de datos 4.12 con la información y las imágenes que posteriormente se mostrarán por la pantalla del dispositivo. Este proceso conlleva un cierto tiempo, el cual dependerá de la cantidad sobre todo de imágenes que se descarguen a la memoria interna del dispositivo.

Una vez abierto nuestro programa, se muestra una lista con las actividades disponibles para ser consultadas por el usuario 4.13. Deslizando el dedo hacia arriba o hacia abajo se pueden ver todas las actividades disponibles para las que nuestro programa puede sernos de ayuda. Cabe mencionar que esta es una versión de prueba de la base de datos. En una versión más avanzada se podrían mostrar actividades diferentes orientadas a cada usuario, por ello, aunque aquí puedan parecer que son muchas o pocas, en una versión final podrían verse únicamente dos o tres actividades, o muchas más. Dependería de las necesidades de cada usuario.

Sobre esta lista el usuario puede realizar dos tipos de acciones. La primera es deslizar el dedo hacia arriba o hacia abajo para ver el nombre de todas y cada una de las actividades disponibles. La segunda opción que se puede realizar es un pulsar con el dedo sobre una de ellas. Una vez que el



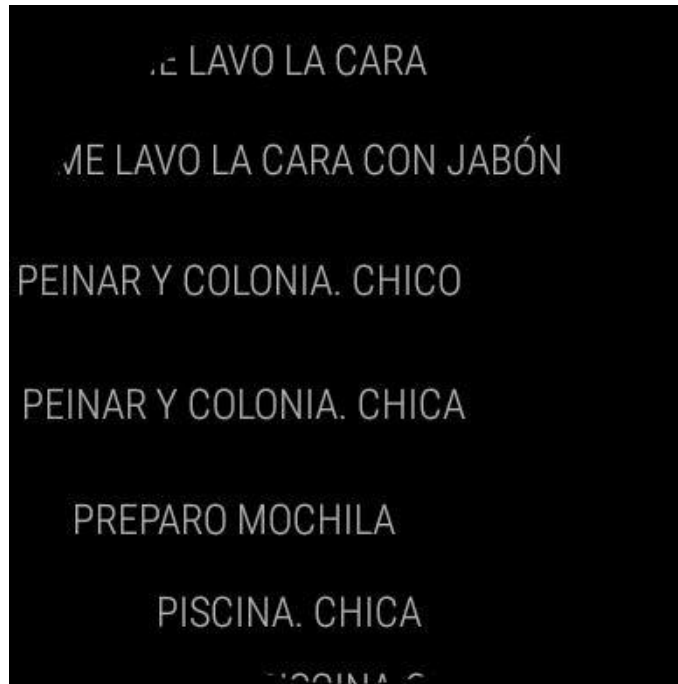
(a)

Figura 4.11: La imagen se puede observar como en la propia esfera del reloj se puede colocar un acceso directo a nuestra aplicación Assist-Task Wear para poder acceder a ella con un solo toque con el dedo.



(a)

Figura 4.12: Imagen que muestra el proceso de descarga de la base de datos desde el servidor a nuestro dispositivo.



(a)

Figura 4.13: En esta imagen podemos observar una lista de las actividades que contiene la base de datos.

usuario encuentre la actividad que desea realizar, y para la que el programa puede serle útil, debe pulsar sobre ella con el dedo y es entonces cuando la actividad comenzará a mostrar los pasos que el usuario debe seguir para realizarla completamente.

A continuación 4.14 se va a mostrar una lista de imágenes, que son capturas de pantalla de una actividad en concreto.

Para pasar de un paso al siguiente, el usuario debe desplazar el dedo de derecha a izquierda por la pantalla, haciendo un gesto parecido al de pasar página de un libro. Del mismo modo, si necesita volver a ver algún paso anterior, se haría deslizando el dedo en sentido contrario, esto es, de izquierda a derecha. Cabe destacar que si se intenta pasar rápido de un paso a otro el sistema no nos deja, con esto evitamos que no se vea algún paso por error, o por cualquier manipulación indebida por parte del usuario.

Por otro lado también debe de quedar constancia de que en todo momento, si el dispositivo lo permite, se escuchará a través del altavoz del SmartWatch la locución del texto que aparece en pantalla junto a la imagen, de tal modo que el usuario no tiene ni siquiera que ver o leer la pantalla directamente para poder seguir los pasos.

En el caso de que hayamos realizado todos los pasos que nos ha ido enumerando el programa, éste nos indicará que hemos terminado correctamente la tarea, y se mostrará la siguiente pantalla 4.15.



Figura 4.14: imágenes de los pasos de la actividad seleccionada.

Tendremos en este caso que pulsar el botón verde que indica “Salir”, el cual es un botón amplio, grande y visible. Se debe de pulsar un botón en lugar de seguir con desplazamiento con el dedo, ya que de este modo sería muy fácil salir de esta pantalla, tocar sin querer otra actividad y que Assist-Task Wear nos muestre erróneamente pasos que nada tengan que ver con lo que el usuario necesita realizar. Al obligar al usuario a pulsar botón para seguir adelante, se tiene un grado de seguridad añadido para que éste sea completamente consciente de que ha terminado la tarea.



(a)

Figura 4.15: La imagen nos muestra un mensaje que nos indica que hemos terminado de realizar correctamente la tarea.

Al pulsar el botón de salir, volvemos a la pantalla inicial con la lista de tareas, esperando a que el usuario pueda elegir otra para volver a ayudarlo a que llegue a buen fin.

Por último, vamos a terminar con uno de los problemas más típicos con los que nos hemos encontrado en las sucesivas pruebas que le hemos hecho al producto. Pese a que en el archivo Android manifest, hemos señalado como uno de los permisos que el sistema debe otorgarle a nuestro programa sea el uso tanto para lectura como para escritura de la memoria interna de nuestro dispositivo, en ocasiones no lo reconoce, y es aquí cuando debemos de hacerlo manualmente entrando en el menú de ajustes del dispositivo, aplicaciones y notificaciones, permisos de aplicaciones, buscar nuestra aplicación Assist-Task Wear y activar el permiso de almacenamiento.

A continuación le mostramos una serie de capturas de pantalla 4.16.

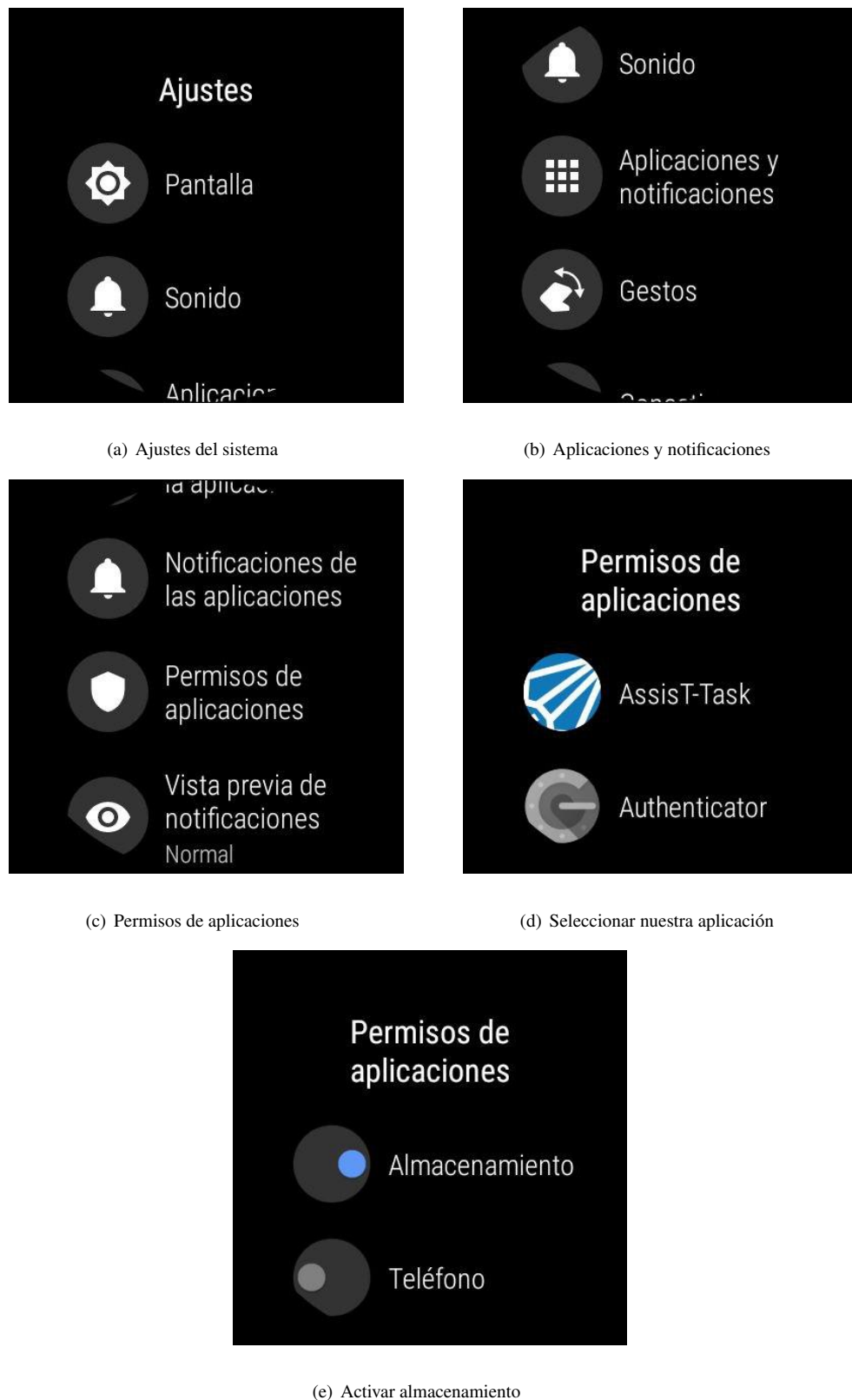


Figura 4.16: En esta imagen se muestran los pasos a seguir para activar el permiso de almacenamiento si nuestro dispositivo no lo hace automáticamente.

CONCLUSIONES Y TRABAJO FUTURO

5.1. Conclusiones

AssisT-Task Wear, es una aplicación diseñada para la realización de un cometido concreto, el cual no es otro más que ayudar a las personas con diversidad intelectual a realizar tareas de forma autónoma, que ellos solos no podrían realizarlas.

Este programa no es fiable para todo tipo de personas con diversidad intelectual, está enfocado aquellas personas que pueden realizar una actividad, pero necesitan ayuda externa que compruebe que realizan todos los pasos pertinentes y les ayuden a terminar correctamente su cometido.

Es aquí donde interviene AssisT-Task Wear, el cual se convierte en esa ayuda imprescindible, que los usuarios necesitan, tanto para realizar todos los pasos de la actividad, como para brindarle la seguridad necesaria de que están realizando la tarea correctamente, puesto que ellos mismos van a ir revisando todos los pasos a seguir.

El haber podido implementar este programa en un SmartWatch pone al servicio de las personas con diversidad intelectual una tecnología innovadora cómo son los wearables, los cuales, además de ser un dispositivo de ayuda, son dispositivos innovadores y tecnológicos de última generación, que ayudan a que el usuario se sienta más integrado en una sociedad que cada vez necesita más de la tecnología, para realizar la mayoría de las tareas diarias.

Hay que ser conscientes de que este programa no requiere de la realización de una actividad para poder pasar a otra, sino que es el propio usuario, el que debe indicar si la tarea o el paso han sido realizadas o no. Por lo que no debe ser utilizado para tareas de vital importancia o peligrosas, pero sin embargo se presumen plenamente válido para la ayuda a la realización de tareas que normalmente el usuario realiza en su día a día pero que necesite de esa ayuda extra que AssisT-Task Wear puede ofrecerle.

Serán los usuarios finales los que nos ofrezcan la garantía de que el programa es útil y aporte una mejora en la calidad de vida tanto a las personas diversidad intelectual, como a sus cuidadores o familiares mediante el uso en el día a día.

5.2. Trabajo Futuro

El programa que hemos creado, sin duda alguna, necesita un trabajo futuro para poder asegurar su finalidad y comprobar su eficacia, así como avanzar en funcionalidades.

Algo imprescindible en todo programa de desarrollo, es probarlo con usuarios finales. De este modo podremos observar cuáles son sus carencias y virtudes, para poder seguir avanzando en el desarrollo y mejorar la experiencia del usuario, que sin duda, es el principal objetivo de un proyecto de este tipo.

Otra de las mejoras que se pueden realizar es la posibilidad de tener una tabla de actividades para cada usuario. En la actualidad Assist-Task Wear se conecta al servidor y nos descarga la misma base de datos para cualquier persona. Lo ideal será cada usuario únicamente puede ver las actividades que se consideren oportunas, y a su vez, éstas muestren las imágenes o los textos que más fácil pueda llegar a identificar y con las que más cómodo se sienta.

Una vez conseguido que cada usuario disponga de su propia tabla actividades, es simple escalar el programa otro tipo de entornos. Actualmente Assist-Task Wear ha sido diseñado para ayudar a las personas con diversidad intelectual a desarrollar labores de su vida cotidiana, pero si en lugar de este tipo de acciones, introducimos en la base de datos actividades que el usuario realiza en su centro de estudios o en su lugar de trabajo, nuestro programa se ve mucho más completo para todo tipo de ámbitos en los que se pueda necesitar.

BIBLIOGRAFÍA

- [1] R. L. Schalock *et al.*, “El nuevo concepto de retraso mental: comprendiendo el cambio al término discapacidad intelectual,” 2012.
- [2] J. Romañach and M. Lobato, “Diversidad funcional, nuevo término para la lucha por la dignidad en la diversidad del ser humano,” *Foro de vida independiente*, vol. 5, pp. 1–8, 2005.
- [3] A. A. on Intellectual, D. Disabilities, and M. Alonso, *Discapacidad intelectual: Definición, clasificación y sistemas de apoyo - 11 Edición*. Alianza Ensayo, Alianza Editorial, 2011.
- [4] L. de Sousa Britto Neto, V. R. M. L. Maíke, F. L. Koch, M. C. C. Baranauskas, A. de Rezende Rocha, and S. K. Goldenstein, “A wearable face recognition system built into a smartwatch and the blind and low vision users,” in *Enterprise Information Systems* (S. Hammoudi, L. Maciaszek, E. Teniente, O. Camp, and J. Cordeiro, eds.), (Cham), pp. 515–528, Springer International Publishing, 2015.
- [5] A. Amiri, N. Peltier, C. Goldberg, Y. Sun, A. Nathan, S. Hiremath, and K. Mankodiya, “Wearsense: Detecting autism stereotypic behaviors through smartwatches,” *Healthcare*, vol. 5, p. 11, Feb 2017.
- [6] V. Dibia, S. Trewin, M. Ashoori, and T. Erickson, “Exploring the potential of wearables to support employment for people with mild cognitive impairment,” in *Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility*, ASSETS '15, (New York, NY, USA), p. 401–402, Association for Computing Machinery, 2015.
- [7] R. d. Castro Muñoz *et al.*, “Sistema de guiado en exteriores para personas con necesidades especiales mediante smartwatches,” B.S. thesis, 2016.
- [8] J. C. Torrado, J. Gomez, and G. Montoro, “Emotional self-regulation of individuals with autism spectrum disorders: smartwatches for monitoring and interaction,” *Sensors*, vol. 17, no. 6, p. 1359, 2017.
- [9] J. Gomez, J. C. Torrado, and G. Montoro, “Using smartphones to assist people with down syndrome in their labour training and integration: a case study,” *Wireless Communications and Mobile Computing*, vol. 2017, 2017.

UAM

UNIVERSIDAD AUTONOMA

DE MADRID