



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior de Jaén

Trabajo Fin de Grado

**PROTOTIPO DE APLICACIÓN
MÓVIL ASISTENTE PERSONAL
PARA PERSONAS CON TEA**

Alumno: Manuel Jesús Romero Mateos

Tutor: Prof. D. Francisco de Asís Conde
Rodríguez.

Dpto: Departamento de Informática.

Octubre, 2023

(Página intencionalmente en blanco)



Universidad de Jaén

Departamento de Informática

Don Francisco de Asís Conde Rodríguez, tutor del Trabajo Fin de Grado titulado: **'Prototipo de aplicación móvil asistente personal para personas con TEA'**, que presenta Don Manuel Jesús Romero Mateos, otorga el visto bueno para su entrega y defensa en la Escuela Politécnica Superior de Jaén.

Jaén, septiembre de 2023

El alumno:

El tutor:

Manuel Jesús Romero Mateos

Francisco de Asís Conde Rodríguez

(Página intencionalmente en blanco)

Agradecimientos

A mi tutor, Francisco de Asís, por brindarme la oportunidad de acompañarme en la realización de este Trabajo,

a la Asociación HorizonTEAsperger de Jaén, y en concreto a Aurora, por su inestimable ayuda a la hora de definir los requisitos del proyecto, y por ayudarme a salir adelante en lo personal,

a las personas que, durante las fases de desarrollo, han probado el prototipo para verificar su calidad, por ayudarme y además estar ahí en los malos momentos,

y a demás gente (familia, pareja, actuales y antiguos amigos, y demás compañeros y profesores) por estar ahí de alguna forma a pesar de los complicadísimos momentos que hemos pasado juntos durante mi paso por la Universidad.

FICHA DEL TRABAJO FIN DE TÍTULO	
Titulación	Grado en Ingeniería Informática
Modalidad	Proyecto de Ingeniería
Especialidad (solo TFG)	Tecnologías de la Información
Mención (solo TFG)	Sin mención
Idioma	Español
Tipo	Específico
TFT en equipo	No
Autor/a	Manuel Jesús Romero Mateos
Fecha de asignación	04/11/2022
Descripción corta	El trabajo consiste en la elaboración del análisis, diseño e implementación de un prototipo de una aplicación móvil que sirva de asistente personal para personas con TEA (Trastorno del Espectro Autista). La aplicación contará con diversas funciones que ayudarán a estas personas en su desarrollo y autonomía personal, como gestor de tareas, gestor de rutinas cotidianas, registro de emociones y estado de ánimo cotidiano, visualizador de tablas de planes basadas en diagramas de Gantt, y sistema de notas de texto y voz. Se ha enfocado el análisis de requisitos de la aplicación contando con profesionales de la psicología que han proporcionado información orientativa relativa al trabajo en terapia con una persona con TEA. La aplicación se ha construido con el framework multiplataforma Flutter, favoreciendo la agilidad en el desarrollo y la expansión de características en un futuro en el que se plantea su publicación.

NORMAS APLICADAS EN ESTE DOCUMENTO	
LOCALES	
TFT-UJA:2017	Normativa de Trabajos Fin de Grado, Fin de Máster y otros Trabajos Fin de Título de la Universidad de Jaén (Normativa marco UJA aprobada en Consejo de Gobierno)
TFT-EPSJ:2017	Normativa sobre Trabajos Fin de Grado y Fin de Máster en la Escuela Politécnica Superior de Jaén (Normativa EPSJ aprobada en Junta de Escuela)
TFT-EPSJ	Criterios de evaluación y normas de estilo para TFG y TFM de la Escuela Politécnica Superior de Jaén
NACIONALES E INTERNACIONALES	
ISO 2145:1978	Documentación - Numeración de divisiones y subdivisiones en documentos escritos
UNE 50132:1994	Traducción de la ISO 2145
APA 6ª edición	Estilo de referencias y citas de APA (American Psychological Association)

NORMAS UTILIZADAS COMO BASE O REFERENCIA	
NACIONALES	
UNE 157001:2014	Criterios generales para la elaboración formal de los documentos que constituyen un proyecto técnico
UNE 157801:2007	Criterios generales para la elaboración de proyectos de sistemas de información
<i>Estas normas se han utilizado como base o referencia para la inclusión de algunos contenidos y definiciones sobre elaboración de proyectos, entendiendo como proyecto la documentación consensuada entre una empresa y un cliente, que da lugar al perfeccionamiento de un contrato para la elaboración de una obra o la prestación de un servicio. Por consiguiente, no debe esperarse la aplicación de estas normas en cuanto a la completitud de los contenidos ni a la organización de los mismos.</i>	

Contenido

1	Especificación del trabajo	13
1.1	Introducción.....	13
1.2	Objetivos del trabajo	14
1.3	Antecedentes y estado del arte	14
1.3.1	El Trastorno del Espectro Autista	14
1.3.2	La terapia psicológica en una persona con TEA	16
1.3.3	Facilidades informáticas para personas con TEA	17
1.3.4	Desarrollo de aplicaciones multiplataforma	18
1.4	Descripción de la situación de partida.....	19
1.4.1	Descripción del entorno actual.....	19
1.4.2	Resumen de las deficiencias y carencias identificadas.....	21
1.5	Requisitos iniciales	21
1.5.1	Propuestas iniciales – propias	22
1.5.2	Sondeo a profesionales	23
1.5.3	Clasificación de requisitos	24
1.6	Alcance.....	26
1.7	Hipótesis y restricciones	27
1.8	Estudio de alternativas y viabilidad	29
1.9	Descripción de la solución propuesta	30
1.10	Tecnologías utilizadas.....	31
1.11	Metodología de desarrollo de software.....	32
1.12	Estimación del tamaño y esfuerzo	34
1.13	Planificación temporal.....	35
1.14	Presupuesto.....	36
2	Diseño inicial	38
2.1	Especificaciones del sistema	38
2.1.1	Requisitos funcionales	38
2.1.2	Requisitos no funcionales	40
2.1.3	Casos de uso	41
2.2	Análisis y diseño del sistema	45
2.2.1	Almacenamiento de archivos.....	45
2.2.2	Interacción con notas de texto y voz	46
2.2.3	Notas de voz	47
2.2.4	Tablas de tareas	48
2.2.5	Vista de calendario y anotaciones diarias	49
2.2.6	Diseño de la interfaz de inicio	50
3	Desarrollo	52

3.1	Primera Iteración	52
3.1.1	Diagramas de clases	52
3.1.2	Diagrama de secuencia	53
3.2	Segunda iteración.....	55
3.2.1	Diagramas de clases	56
3.2.2	Diseño de la interfaz	57
3.3	Tercera iteración.....	58
3.3.1	Diseño de la interfaz	59
3.4	Cuarta iteración	60
3.4.1	Diagramas de secuencia	61
3.4.2	Diagramas de clases	62
3.4.3	Diseño de la interfaz	62
3.5	Quinta iteración	64
3.5.1	Diseño de la interfaz	64
3.6	Sexta iteración	64
3.7	Séptima iteración.....	65
3.7.1	Funciones nuevas y descartadas	67
3.7.2	Diagramas de clases	69
3.7.3	Diagramas de casos de uso	70
3.7.4	Diseño de la interfaz	71
3.8	Pruebas finales	71
3.8.1	Pruebas de verificación y validación del sistema	72
3.8.2	Validación de la usabilidad del sistema	72
4	Conclusiones y trabajos futuros	75
5	Apéndices	76
5.1	Instalación y configuración del sistema	76
5.2	Manuales de usuario	77
5.2.1	Puesta en marcha de la aplicación	77
5.2.2	Trabajo con tareas	77
5.2.3	Trabajo con notas de texto y voz	79
5.2.4	Resumen del día	80
5.2.5	Estado de ánimo diario	81
5.2.6	Tablas de planes.....	82
5.2.7	Ruleta de decisiones.....	83
5.2.8	Pantalla de ajustes.....	84
6	Definiciones y abreviaturas.....	85
7	Bibliografía	87

Índice de ilustraciones

Ilustración 1.....	17
Ilustración 2.....	20
Ilustración 3.....	28
Ilustración 4.....	33
Ilustración 5:.....	34
Ilustración 6.....	42
Ilustración 7.....	43
Ilustración 8.....	44
Ilustración 9.....	46
Ilustración 10.....	47
Ilustración 11.....	48
Ilustración 12.....	49
Ilustración 13.....	51
Ilustración 14.....	51
Ilustración 15.....	53
Ilustración 16.....	54
Ilustración 17-.....	56
Ilustración 18.....	57
Ilustración 19.....	58
Ilustración 20	59
Ilustración 21.....	61
Ilustración 22.....	62
Ilustración 23.....	63
Ilustración 24.....	63
Ilustración 25.....	64
Ilustración 26.....	69
Ilustración 27.....	70
Ilustración 28.....	71
Ilustración 29.....	77
Ilustración 30.....	78
Ilustración 31.....	79
Ilustración 32.....	80
Ilustración 33.....	81
Ilustración 34.....	82
Ilustración 35.....	83
Ilustración 36.....	84

Índice de tablas

Tabla 1.1 – Desglose de requisitos (Must y Should)	25
Tabla 1.2 – Desglose de requisitos (Could y Won't)	26
Tabla 2 – Desglose de costes de personal humano	36
Tabla 3 – Presupuesto total.....	37

1 ESPECIFICACIÓN DEL TRABAJO

En este capítulo se presenta la especificación del trabajo, con una estructura y contenidos **inspirados** en los criterios y recomendaciones que establece la norma UNE 157801:2007 - “*Criterios Generales para la elaboración de proyectos de Sistemas de Información*”.

A lo largo del documento se utilizarán términos y acrónimos cuya descripción aparecen en el apartado 6 (Definiciones y abreviaturas).

1.1 Introducción

El presente Trabajo de Final de Grado consiste en la elaboración de una aplicación para dispositivos móviles, consistente en un asistente personal para personas con TEA (Trastorno del Espectro Autista). Este trabajo abarca el análisis y el diseño de la solución, así como la implementación de un prototipo.

Esta aplicación incorporará utilidades para ayudar a las personas con TEA (que serán el usuario objetivo de la presente solución), principalmente aquellas que están entrando en la edad adulta, con su **organización personal y toma de decisiones**, mediante herramientas como un gestor básico de tareas y rutinas y un planificador de días. Además, se incorporarán herramientas de toma de apuntes textuales o por voz, que serán cruciales para el manejo de ciertas situaciones personales y su posterior análisis en una sesión de terapia psicológica. Por último, la aplicación incorporará instrucciones para el manejo de ciertas situaciones que resultan complejas en el día a día del potencial usuario.

A la hora de la elaboración de la aplicación, se ha estado en un permanente contacto y comunicación con personas profesionales de la psicología, a fin de perfilar la mejor solución posible para concebir una aplicación completa.

1.2 Objetivos del trabajo

El objetivo principal es el análisis, diseño e implementación de un prototipo de aplicación Asistente Virtual para Personas con TEA. El desarrollo estará fraccionado en función de la implementación de cada función específica.

La aplicación debe ser capaz de aunar todas las funciones básicas de la aplicación en una interfaz sencilla y accesible, de tal forma que no sea compleja su utilización en caso de que su usuario la necesite con urgencia.

Las funciones básicas determinarán los subobjetivos que deberemos alcanzar:

1. Sistema básico de gestión de tareas.
2. Sistema de toma de notas:
 - a. Notas de voz.
 - b. Notas de texto.
3. Pantalla de inicio que recoja las tareas a hacer y un sistema de rutinas, así como un pequeño diario personal para expresar de forma breve el estado de ánimo del usuario cada día.
4. Vista de calendario para consultar lo que cada día se ha ido anotando.
5. Sistema de rutinas básicas para la autonomía personal del usuario.
6. Herramienta de planificación horaria de los días, que permitirá agregar tablas de horarios de forma ilimitada para poder contemplar varias posibilidades (por ejemplo, un plan para un día de viaje, para el primer día que el usuario pase en determinado lugar, etc.).
7. Menú de configuración para controlar algunos aspectos de la aplicación, con alta flexibilidad para configurar los colores de la interfaz de usuario y la extensión de las tablas de planificación horaria.

1.3 Antecedentes y estado del arte

1.3.1 El Trastorno del Espectro Autista

De acuerdo con la Guía de Consulta de los Criterios Diagnósticos del DSM-5, publicada por la APA (Asociación Americana de Psicología (American Psychological

Association, 2014)), el **Trastorno del Espectro del Autismo** se caracteriza por las siguientes posibles condiciones presentadas en el desarrollo neurológico del ser humano:

- Déficits en la comunicación social en diversos contextos, así como falta de reciprocidad comunicativa y dificultad para la adaptación a determinados contextos sociales.
- Patrones restrictivos y repetitivos de comportamiento, que se caracterizan por:
 - Movimientos, utilización de objetos o habla estereotipados o repetitivos (por ejemplo, estereotipias motoras simples).
 - Insistencia en la monotonía, inflexibilidad de rutinas, y dificultades ante las adversidades que puedan romper la rutina (por ejemplo, angustia ante cambios repentinos de planes, rigidez a la hora de realizar ciertas acciones, etc.).
 - Intereses especiales muy marcados y enfocados.
 - Hiper- o hiporeactividad a los estímulos sensoriales, como los sonidos a muy alto volumen o frecuencia, los leves roces con otros objetos o personas, o estímulos visuales como pantallas.
- La alterada reacción a los estímulos sensoriales provoca divergencias cognitivas que pueden traducirse en dificultades en la planificación personal y organización, así como en la toma de decisiones. Es un detalle estrechamente relacionado con la monotonía y la rigidez cognitiva, y que puede agudizarse en la edad adulta de no tratarse adecuadamente.
- Estos síntomas suelen detectarse con precocidad, si bien se pueden manifestar a lo largo de la vida de la persona. Se hace más complicado desarrollar aptitudes sociales, laborales, etc. a medida que la persona crece, por lo que una intervención psicológica es crucial.
- No se advierte una relación entre el Trastorno del Espectro Autista y un posible cuadro clínico de discapacidad intelectual o retraso del desarrollo, aunque con frecuencia coinciden.

En antiguos manuales de psicología y psiquiatría de la APA y la CIE (Clasificación Internacional de Enfermedades de la Organización Mundial de la Salud) se hacía distinción entre el autismo y la condición anteriormente conocida como síndrome de Asperger. Éste último compartía varios de los aspectos diagnósticos citados con anterioridad.

El TEA se suele clasificar en distintos grados, en función de la cantidad y tipo de ayuda que la persona requiera, si bien la clasificación por grados resulta, en determinados casos, imprecisa. En grado 1 se suelen diagnosticar las personas que normalmente necesitarán una ayuda más enfocada en su autonomía personal, mientras que las personas diagnosticadas con grados superiores necesitarán más ayuda a priori, incluso necesitando una dependencia más intensa de terapeutas, familiares, etc.

1.3.2 La terapia psicológica en una persona con TEA

Al no ser una enfermedad, no existe como tal un tratamiento farmacológico para el TEA, ni tampoco es coherente hablar de una “cura” para el mismo. Sin embargo, algunos de los síntomas se mitigan a medida que la persona diagnosticada desarrolla su crecimiento, sobre todo si existe una adecuada ayuda por parte de profesionales.

Las terapias más comunes para el TEA son la terapia cognitivo-conductual, terapia en habilidades sociales, comunicación y lenguaje, y terapia ocupacional, así como otras modalidades con determinada generalización o especialización en ciertos aspectos del trastorno. Debido a que el trastorno se puede manifestar de forma distinta entre cada individuo, esto también involucrará la elección de determinadas terapias en función de su encaje y resultado en el paciente (What is Autism Spectrum Disorder?, 2022).

En la terapia cognitivo-conductual (MayoClinic, 2021) se incita al paciente a expresar los pensamientos y sentimientos acerca de las situaciones y circunstancias preocupantes para el mismo. Se identifican los aspectos mentales con más taras (principalmente pensamientos negativos) y progresivamente se tratan de reformar para, en última instancia, resolver cada situación planteada por el paciente.

En el TEA la terapia psicológica suele necesitar de un amplio tiempo para conseguir resultados tangibles, y ésta depende de las circunstancias del entorno en el que la persona vive.

1.3.3 Facilidades informáticas para personas con TEA

Al no tener una alta prevalencia por sobre el conjunto de la población, con un ratio de alrededor de 1 sobre 36 en niños (Data & Statistics on Autism Spectrum Disorder, 2023), la cantidad de soluciones es reducida.

Ejemplos como *Proloquo2Go* o *i Create* (Davis, 2019) tienen el enfoque puesto en niños con TEA que carecen de capacidad de comunicación, o presentan déficits severos en ésta. Sin embargo, presentan limitaciones importantes al estar solamente enfocadas a los niños con TEA y ser complicado su aprovechamiento por parte de un adulto, además de estar disponibles solamente para dispositivos móviles de Apple, que a fecha de agosto de 2023 sólo suponen un 28,52 % de la cuota de mercado total de dispositivos móviles (Mobile Operating System Market Share Worldwide, 2023).

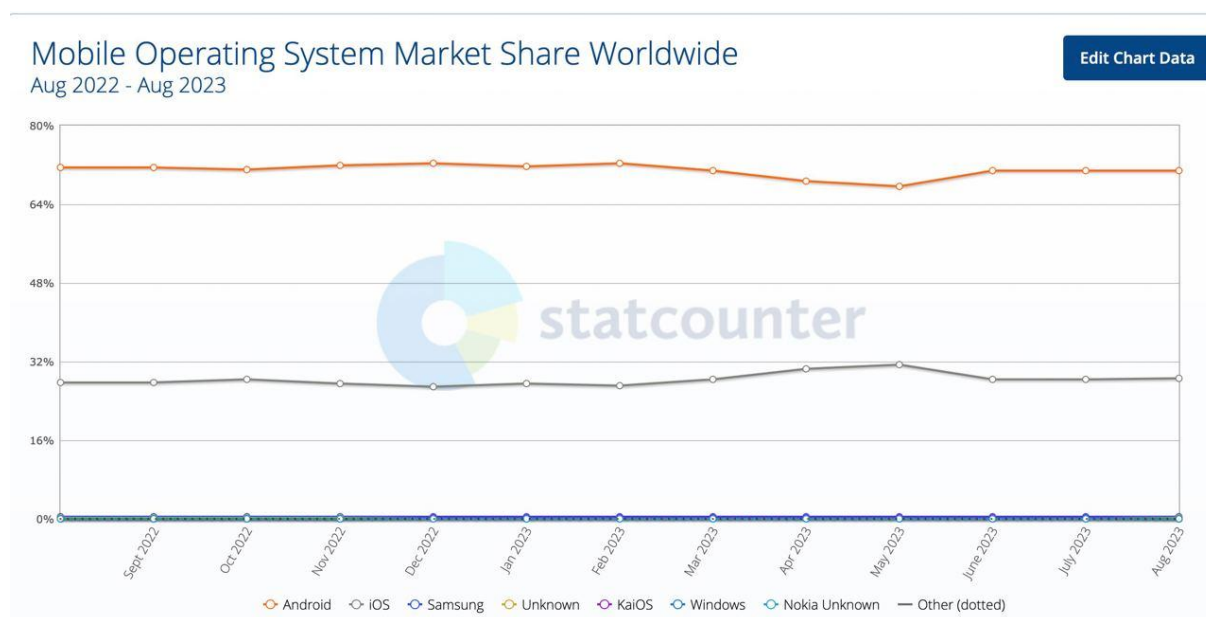


Ilustración 1 – Detalle de la cuota de mercado de dispositivos móviles a fecha de agosto de 2023 (fuente: statcounter.com)

Otras soluciones existentes en el mercado, como *Language Therapy for Kids – MITA*, u *Otsimo* (Davis, 2019), son también compatibles con Android, pero su enfoque también está centrado en niños, o adultos con altos déficits en el habla y la comunicación.

En cuanto a la gestión de tareas y planificación diaria, hay aplicaciones respaldadas por grandes empresas, como Microsoft To-Do, Evernote o Google Keep. También es digna de destacar la utilidad de otras soluciones como Obsidian.md, Notion o Logseq para este menester, pero su uso dista de ser intuitivo para las personas que no están instruidas de forma intensa en el uso de herramientas informáticas de este calibre.

Para un correcto aprovechamiento de la terapia cognitivo-conductual en las personas con TEA que la reciben, es interesante inducirles a anotar cómo se sienten ante determinadas situaciones que experimentan en su vida, así como saber identificar las emociones básicas que los acompañan en dichas situaciones. Sin embargo, a la fecha de elaboración del presente trabajo no existe ninguna solución enfocada específicamente en esto.

1.3.4 Desarrollo de aplicaciones multiplataforma

Para una mayor accesibilidad a la aplicación de cara a un posible lanzamiento al público, se ha tomado la consideración de utilizar una plataforma de desarrollo (framework) multiplataforma. A continuación se detallan algunos de los framework de mayor renombre en la actualidad (Srivastava, 2023):

- **ionic:** se basa en AngularJS y proporciona acceso a lenguajes de marcado y estilo como HTML5 y CSS3, así como lenguajes de programación como JavaScript, usando Apache Cordova para dar acceso a funciones específicas de cada plataforma y dispositivo.
- **React Native:** está basado en JavaScript y permite crear aplicaciones multiplataforma con apariencia nativa para cada sistema. Integra ReactJS y permite escribir módulos en Objective-C, Swift o Java.
- **Xamarin:** en lugar de utilizar tecnologías web, se usan lenguajes como C# o .Net. Con la ayuda de APIs (interfaces de programación de aplicaciones) y plugins se pueden obtener multitud de funcionalidades en el desarrollo de una aplicación determinada. Además, se permite el uso de bibliotecas de otros lenguajes como C++ o Java, haciéndolo una opción interesante.

- **Node.js:** Node.js construye aplicaciones basadas en tecnologías web utilizando el motor V8 de Google Chrome. Permite el desarrollo de aplicaciones de red escalables, así como desarrollos a nivel de servidor.
- **Flutter:** Su primera versión estable fue introducida por Google en 2018 (Ramírez, 2018), y usando el lenguaje de programación Dart como base (también potenciado por Google), se trata de un SDK (kit de desarrollo de software) que tiene un enfoque especial en el desarrollo de aplicaciones para Android e iOS, así como el sistema operativo Google Fuchsia. Permite uniformidad entre distintas plataformas sin descuidar la sencillez en el desarrollo. Su base conceptual es el árbol de widgets, y esto permite trabajar de forma eficiente replicando árboles de widgets cuando resulte necesario. Flutter permite desarrollar aplicaciones de calidad de una forma sumamente acelerada, sin tener que diseñar interfaces distintas para cada plataforma de las que soporta. Actualmente, la cátedra de la asignatura Desarrollo de Software para Dispositivos Móviles (código 13312008 del plan de estudios del Grado en Ingeniería Informática de la Escuela Politécnica Superior de Jaén) se basa en Flutter y Dart.

A partir de este análisis, y sabiendo el enfoque de la solución que se quiere implementar, se ha optado por **Flutter** como plataforma de desarrollo.

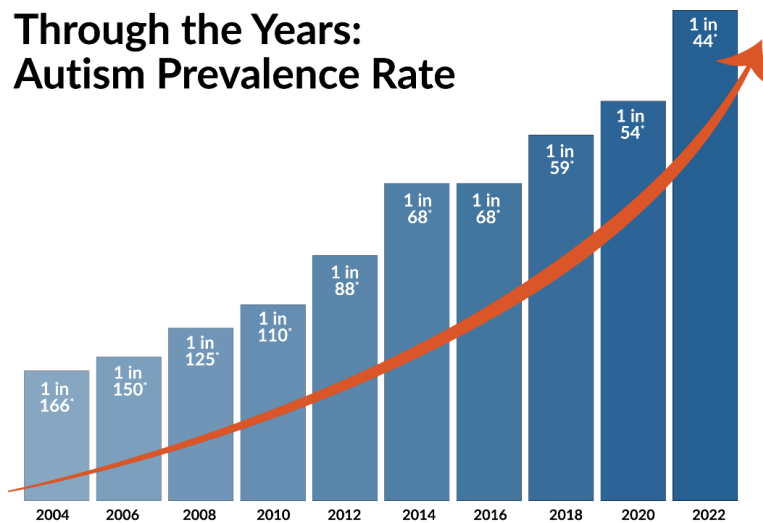
1.4 Descripción de la situación de partida

En los siguientes puntos, se detallará la situación de partida en la que se pone en contexto el Trabajo, para poder desarrollar los siguientes aspectos con la debida corrección.

1.4.1 Descripción del entorno actual

Con la presente solución, se pretende crear una aplicación que acompañe a una persona con TEA hacia su plena autonomía personal. Si bien no será posible lograr este objetivo con personas con TEA que requieran de una gran cantidad de ayuda externa, es factible alcanzarlo para aquellas personas cuyas necesidades y dificultades no interfieren de una forma intensa en los aspectos básicos de la vida de la persona.

Al no existir soluciones enfocadas en una persona adulta, esta aplicación vendrá a intentar redimir esta carencia.



*Centers for Disease Control and Prevention prevalence estimates are for four years prior to the report data.

Ilustración 2 – Incremento de la prevalencia de TEA en Estados Unidos entre 2004 y 2022, datos recogidos por los Centros para el Control y la Prevención de Enfermedades de EE.UU.

En esta gráfica podemos comprobar que la prevalencia del TEA ha incrementado en los últimos 19 años en Estados Unidos de América. De estos datos se puede interpretar el hecho de que 1 de cada 150 personas en 2006 tenían diagnóstico. Si el diagnóstico se suele realizar en niños, la mayoría de los niños que fueran diagnosticados ese año estarán llegando o habrán llegado a la edad adulta. Extrapolando estos resultados al conjunto mundial, no es descabellado plantear que las cifras sean similares, y que, por tanto, hay una considerable cantidad de personas con TEA, cuyas necesidades se concentren en el plano de las relaciones y habilidades sociales, y la proyección académica y laboral, que están por entrar en la edad adulta y podrían recibir de buen grado un asistente personal del que puedan disponer de forma ubicua a través de su dispositivo móvil, en todo momento.

Asimismo, dado el hecho de que la solución planteada en este Trabajo tiene margen de escalabilidad, existe la posibilidad de que en el futuro sea más refinada y pueda, por tanto, acoger a más usuarios con TEA que dentro de unos años serán adultos y probablemente requieran de las herramientas que ofrecerá.

1.4.2 Resumen de las deficiencias y carencias identificadas

Las soluciones exploradas en el punto 1.3.3 no se amoldan a las necesidades que pueda tener un adolescente o adulto con necesidades poco disruptivas en los aspectos más básicos de su vida, como hemos analizado. Por tanto, la aplicación que acompaña al presente Trabajo pretende cubrir ese vacío, enfocándose en los problemas más comunes de las personas diagnosticadas que tienen a partir de 16 años. Como se ha comentado con anterioridad, estos problemas se suelen circunscribir en la organización personal y la toma de ciertas decisiones, así como el exceso de información manejada que podría dificultar el paso por terapias y, por tanto, dificultar la ayuda que se necesita.

1.5 Requisitos iniciales

Para el estudio de los requisitos, se ha consultado con el tutor del Trabajo, y se ha concluido que se usaría la metodología **MoSCoW**.

MoSCoW es una técnica de priorización bastante popular en la gestión de requisitos. Este nombre se debe a la propia metodología en sí, que divide los requisitos según su importancia, así:

- **Must-have:** características que el proyecto **debe** tener.
- **Should-have:** características que el proyecto **debería** tener. Un peldaño por debajo de las características 100 % obligatorias.
- **Could-have:** características que el proyecto **podría** tener.
- **Won't-have:** características que el proyecto **no va a** tener.

Para el estudio de los requisitos se han realizado exhaustivos análisis, que han contado con la colaboración de profesionales del ámbito de la psicología a los que un servidor ha podido acceder.

En el apartado “Alcance” se detallará más información sobre los roles contemplados en el proyecto. El presente Trabajo contemplará la existencia de un único rol: el “usuario”. El usuario será el que controle todos los aspectos de la aplicación y su utilización. Al no contar con una estructura de cliente-servidor, no es

necesaria la existencia de un “administrador”, si bien se contempla la posibilidad de que un “tutor” gestione el contenido que el usuario maneja en la aplicación.

En cuanto a los requisitos más secundarios (“could have” y “won’t have”) se detallarán más adelante.

1.5.1 Propuestas iniciales – propias

El alumno creador de este trabajo tiene diagnóstico de TEA, y sería un potencial usuario objetivo de la aplicación, por lo que a fecha de la proposición formal del Trabajo se propusieron varios requisitos:

1. La aplicación debería integrarse con los servicios de accesibilidad del sistema operativo.
2. El usuario debería poder personalizar su experiencia de usuario mediante el cambio de colores de la interfaz de la aplicación.
3. El usuario debe poder acceder a una completa gestión de tareas diarias.
4. El usuario debería poder almacenar estas tareas de forma externa a la aplicación.
5. El usuario debe poder acceder a un sistema de gestión de tiempo para planes y tareas que impliquen determinado grado de concentración y productividad.
6. El usuario debería poder gestionar recordatorios para determinados aspectos, como tareas o rutinas esenciales.
7. El usuario debería tener acceso a información que le ayude en tiempo real con ciertos aspectos del día a día que les resulten difíciles.
8. El usuario debería disponer en la aplicación de su información burocrática (informes de diagnóstico en PDF, tarjeta de discapacidad) para su posible uso en circunstancias académicas, laborales..., así como actividades de ocio que cuentan con personas con discapacidad reconocida.
9. El usuario deberá disponer de un pequeño “diario personal” en el que se pueda escribir o narrar por audio una situación que le ha ocurrido, y los pensamientos laterales sobre tal situación (cómo debería haber ocurrido, cómo actuó, etc.).

10. El usuario debería tener un control sobre las notificaciones que recibe sobre las tareas y rutinas.
11. La aplicación debería integrarse con el control parental en caso de que sea utilizada por un usuario en edad infantil.
12. La aplicación debería implementar un sistema de sincronización basado en inicio de sesión, que permita el acceso a los datos desde cualquier dispositivo y plataforma.
13. Un tercero autorizado (un tutor legal, familiar, terapeuta...) debería poder acceder a la información que el usuario maneja en la aplicación, para así controlar mejor sus avances y cuestiones.
14. La aplicación deberá tener un modo 100 % offline, para aquellas personas con mayores dificultades para conectarse a Internet. La mayoría de funcionalidades deberían estar disponibles de forma offline.

1.5.2 Sondeo a profesionales

El 2 de mayo, se tuvo una reunión con la directora de la Asociación HorizonTEAsperger, una asociación formada en Jaén (España) a principios de 2023 por varios psicólogos especialistas en autismo, que trata con personas con TEA en edad adulta y pre-adulta que potencialmente podrían ser usuarios de la aplicación que saldrá de este proyecto.

En total hubo dos entrevistas con esta Asociación, siendo la segunda el 21 de junio de 2023, en las que se perfilaron más aspectos en relación a requisitos:

1. El usuario debería poder introducir determinados datos por voz.
2. El usuario debe planificarse tareas y planes para sí mismo, en una base diaria. Esto además debería implicar un cuidado especial de la interfaz gráfica, a fin de evitar problemas de legibilidad. Asimismo, se debería incluir un contador de porcentaje de tareas activas que el usuario ha acabado.
3. El usuario debe poder disgregar las tareas que debe hacer según un código de colores, para así tomar en mejor consideración la prioridad de las mismas.

4. El usuario debería poder disponer de herramientas que le faciliten la anticipación de situaciones complicadas, sobre todo situaciones imprevistas. Sería interesante que estas herramientas se delimitaran por pasos a seguir, así como guías de estrategias ante situaciones.
5. El usuario deberá poder registrar situaciones que le ocurran usando la voz.
6. El usuario debería tener a disposición un sistema de autoinstrucciones que le permitan afrontar con mejor ánimo sus situaciones y problemas.
7. El usuario debería tener acceso a alternativas de ocio compartidas por otros usuarios de la aplicación.
8. El usuario debería poder visualizar y contactar con otros usuarios.
9. La aplicación debería poder organizar según área o ámbito vital las tareas y apuntes que el usuario haga, y organizarlas temporalmente en un calendario.
10. El usuario deberá poder organizar de forma gráfica sus planes diarios y anticipar situaciones en base a su extensión horaria e importancia.
11. El usuario debería recibir recordatorios de vida saludable, así como rutinas de mantenimiento doméstico y personal.

1.5.3 Clasificación de requisitos

Los requisitos sondeados en los apartados anteriores se verán clasificados según la metodología MoSCoW. En las siguientes tablas se dilucida este aspecto:

Must have	Should have
El usuario debe poder acceder a una completa gestión de tareas diarias.	El usuario debería poder personalizar su experiencia de usuario mediante el cambio de colores de la interfaz de la aplicación.
El usuario debe poder acceder a un sistema de gestión de tiempo para planes y tareas que impliquen determinado grado de concentración y productividad.	El usuario debería poder gestionar recordatorios para determinados aspectos, como tareas o rutinas esenciales.
El usuario deberá poder organizar de forma gráfica sus planes diarios y anticipar situaciones en base a su extensión horaria e importancia.	La aplicación debería poder organizar según área o ámbito vital las tareas y apuntes que el usuario haga, y organizarlas temporalmente en un calendario.

El usuario deberá disponer de un pequeño “diario personal” en el que se pueda escribir o narrar por audio una situación que le ha ocurrido, y los pensamientos laterales sobre tal situación (cómo debería haber ocurrido, cómo actuó, etc.).	El usuario debería tener un control sobre las notificaciones que recibe sobre las tareas y rutinas.
La aplicación deberá tener un modo 100 % offline, para aquellas personas con mayores dificultades para conectarse a Internet. La mayoría de funcionalidades deberían estar disponibles de forma offline.	El usuario debería poder introducir determinados datos por voz
El usuario debe poder disgregar las tareas que debe hacer según un código de colores, para así tomar en mejor consideración la prioridad de las mismas.	El usuario debería recibir recordatorios de vida saludable, así como rutinas de mantenimiento doméstico y personal.
El usuario debe planificarse tareas y planes para sí mismo, en una base diaria. Esto además debería implicar un cuidado especial de la interfaz gráfica, a fin de evitar problemas de legibilidad. Asimismo, se debería incluir un contador de porcentaje de tareas activas que el usuario ha acabado	El usuario debería poder disponer de herramientas que le faciliten la anticipación de situaciones complicadas, sobre todo situaciones imprevistas. Sería interesante que estas herramientas se delimitaran por pasos a seguir, así como guías de estrategias ante situaciones.
	El usuario debería tener a disposición un sistema de autoinstrucciones que le permitan afrontar con mejor ánimo sus situaciones y problemas.

Tabla 1.1 – Desglose de requisitos (Must y Should)

Could have	Won't have
La aplicación debería integrarse con los servicios de accesibilidad del sistema operativo.	La aplicación debería integrarse con el control parental en caso de que sea utilizada por un usuario en edad infantil.
El usuario debería poder almacenar estas tareas de forma externa a la aplicación	El usuario debería tener acceso a alternativas de ocio compartidas por otros usuarios de la aplicación.
El usuario debería tener acceso a información que le ayude en tiempo real con ciertos aspectos del día a día que les resulten difíciles.	El usuario debería poder visualizar y contactar con otros usuarios
El usuario debería disponer en la aplicación de su información burocrática (informes de diagnóstico en PDF, tarjeta de discapacidad) para su posible uso en circunstancias académicas, laborales..., así como actividades de ocio que cuentan con personas con discapacidad reconocida.	
La aplicación debería implementar un sistema de sincronización basado en inicio	

de sesión, que permita el acceso a los datos desde cualquier dispositivo y plataforma	
Un tercero autorizado (un tutor legal, familiar, terapeuta...) debería poder acceder a la información que el usuario maneja en la aplicación, para así controlar mejor sus avances y cuestiones	

Tabla 1.2 – Desglose de requisitos (Could y Won't)

1.6 Alcance

Con el alcance del proyecto nos referimos al trabajo necesario para realizar el mismo de una forma satisfactoria, definiendo límites sobre qué se incluirá o no. Una correcta gestión del alcance del proyecto es crucial para el cumplimiento de las expectativas y su éxito (Riveros, 2020).

En el caso que compete con el presente Trabajo, se deben establecer límites nítidos, ya que el proyecto viene a intentar mitigar circunstancias de una disciplina, la psicología, que no funciona en base a patrones y modelos perfectamente definidos y delimitados.

Ciñéndonos a la temática del Trabajo, el TEA se puede manifestar por una serie de comportamientos y actitudes que no serán exactamente iguales en todas las personas que posean la condición. Por ejemplo, una persona con TEA puede manifestar problemas en la comunicación verbal, pero a la vez tener una gestión de estímulos normalizada, otra persona puede tener graves problemas con la hipersensibilidad ante estímulos y en consecuencia con la gestión de su vida cotidiana, pero tener unas habilidades sociales equiparables a la mayoría de sus personas coetáneas.

Así, en este proyecto se quiere hacer un enfoque en los problemas de gestión cotidiana: gestión de tiempos, tareas y rutinas, principalmente, sin descuidar a aquellas personas que, en su búsqueda de autonomía personal, encuentran dificultades en forma de situaciones difíciles de manejar. Por ello, se quiere destinar esta aplicación a las personas con TEA que presenten de una forma más aguda este tipo de necesidades. No obstante, en el hipotético caso de que la aplicación sea publicada, estará disponible para cualquier persona, con TEA o no. Algunas de las funciones incluso podrían ser útiles para otras personas con otro tipo de neurodivergencias contempladas en el manual DSM-5 de la APA.

1.7 Hipótesis y restricciones

El TFG se define como una asignatura de 12 créditos, lo que supone que la duración total del proyecto será de 300 horas, incluyendo todas las etapas del ciclo de vida, con la excepción del mantenimiento. Por consiguiente, la principal restricción aplicable es la limitación de la duración del trabajo.

El mayor condicionante en la realización de este Trabajo ha sido el tiempo dedicado, debido a las responsabilidades académicas que el autor debía asumir. En base a dicho factor restrictivo, se elabora una planificación temporal suficientemente amplia para contrarrestar este problema, la cual puede consultarse en el apartado 1.13 “Planificación temporal”.

Las restricciones también abarcan el desarrollo en sí. A pesar de contemplar un desarrollo para Android e iOS (factor determinante en la elección de Flutter como plataforma de desarrollo), el autor del Trabajo carece de dispositivos físicos con iOS. Por tanto, para las pruebas de la aplicación en iOS se ha trabajado íntegramente con el Simulador de iOS disponible en Xcode, entorno de desarrollo para dispositivos Apple en macOS (sistema al que sí se tiene acceso). En cuanto a Android sí se han podido hacer exhaustivas pruebas en dispositivos reales, algo que ha resultado de amplia utilidad para verificar el funcionamiento de la solución en un entorno real. A pesar de ello, en Android han existido dificultades relativas a la amplia fragmentación de versiones derivadas del sistema para cada marca de dispositivos (Global Smartphone Market Share: By Quarter, 2023).

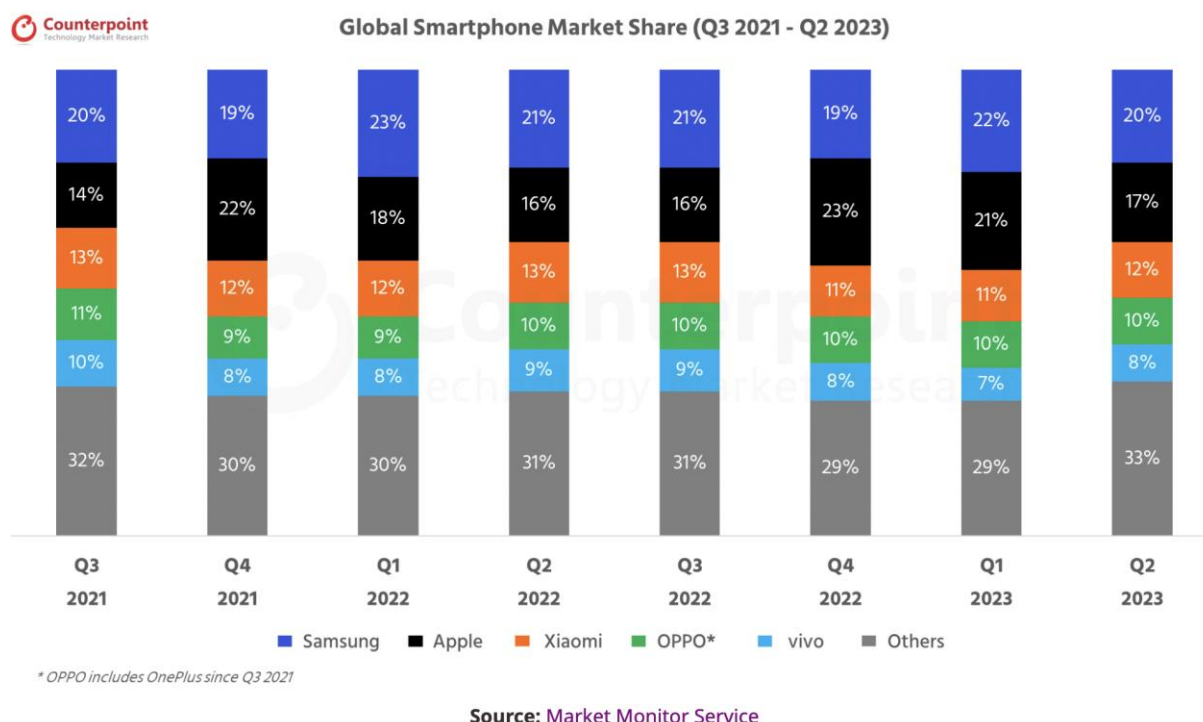


Ilustración 3 – Cuota de mercado de las principales marcas de dispositivos móviles entre 2021 (tercer trimestre) y 2023 (segundo trimestre).

Por ejemplo, el derivado de Android utilizado por los dispositivos Xiaomi es MIUI, y el utilizado por Oppo se denomina ColorOS. Son destacables las diferencias entre cada derivado con respecto a los demás y en relación a la versión original de Android mantenida por Google; en lo que nos compete ha resultado problemática esta fragmentación, sobre todo en cómo maneja cada derivado la gestión de energía en los dispositivos móviles, algo que ha dificultado la implementación de las notificaciones (véase apartado 3.7).

Otra cuestión a tener en cuenta es la necesidad, o no, de una constante conexión a Internet para intercambiar datos con servicios en la nube relacionados con la aplicación. Tras análisis informales realizados por el autor del Trabajo, se determina que en algunos lugares cercanos a su domicilio aún hay serios problemas de cobertura de las redes de telefonía móvil 5G, 4G e incluso 3G, factor condicionado por la geografía y la morfología de ciertos terrenos no cubiertos adecuadamente por antenas de comunicaciones, además de tratarse de áreas rurales notablemente alejadas de núcleos de población mayores. Esto lastra sobremanera la posibilidad de implementar una solución para las notificaciones (y en general, para cualquier otra funcionalidad

de la aplicación) basada en la nube, por lo que, al menos en lo que respecta al presente prototipo, resultará ser una mejor solución la implementación de notificaciones locales.

1.8 Estudio de alternativas y viabilidad

A lo largo del proceso que ha llevado a la gestión de este proyecto, se han tomado en cuenta varias alternativas que han condicionado de diversas maneras el desarrollo del prototipo que se presenta.

Ya se ha expuesto en los apartados 1.3.4 y 1.10 la elección de Flutter como plataforma de desarrollo. El tutor del Trabajo propuso al autor el uso de este framework sobre los demás descritos en el apartado 1.3.4, debido a la agilidad en el desarrollo, a la flexibilidad del lenguaje Dart, y al contundente respaldo por parte de comunidades de desarrolladores y empresas que diseñan aplicaciones para todo tipo de dispositivos.

Con el entorno de desarrollo decidido, se han sopesado más alternativas que han estado enfocadas en la dirección a tomar para implementar una solución. Así, en el estudio de requisitos (apartado 1.5) se abarcaron numerosas propuestas que sobredimensionarían el alcance de la aplicación, y que dentro del marco de este Trabajo se hallan descartadas. No obstante, ante la posible expansión de la solución más allá del ámbito que compete a este Trabajo, y con vistas a una futura publicación (véase apartado 4, “Conclusiones y trabajos futuros”), se toman en cuenta en caso de que se considere oportuna su implementación.

La idea anterior también se ha reflejado en la calidad del prototipo. El diseño de varias de las funciones ha estado limitado por el factor temporal y el alcance del prototipo, y ello ha sido factor condicionante la implementación de dichas funciones. A pesar de esto, se contempla que en el futuro estas funciones tengan una implementación de mejor calidad.

Un ejemplo de lo comentado previamente está en el ámbito de las notificaciones que la aplicación envía. Durante el desarrollo se contempló el posible

uso de Firebase, aunque esto supondría un incremento del tiempo de desarrollo (que incluiría, además, un estudio y diseño de lo que corresponde a las notificaciones push que implementa esta plataforma de desarrollo en la nube). Además, en consonancia con lo comentado en el apartado 1.7, y dado que se pretendía que la aplicación funcione ubicuamente sin importar la presencia o ausencia de conexión a Internet, se descartó el uso de Firebase en el proyecto y se optó por la implementación de notificaciones locales, notablemente más rudimentarias en su puesta en funcionamiento, pero más efectivas de cara a este prototipo.

1.9 Descripción de la solución propuesta

Tenidos en cuenta los factores analizados en los apartados anteriores, se propone una solución basada en los siguientes fundamentos:

- Se desarrollará tomando como referencia la posible futura disponibilidad para Android e iOS. A pesar de usar macOS como sistema donde se lleva a cabo el desarrollo, los respectivos SDKs de cada sistema operativo móvil cuentan con simuladores y emuladores de sus entornos en los que probar las aplicaciones de forma paralela a su desarrollo antes de desplegarlas a otro tipo de público.
- Contará con un sistema de gestión de tareas personales que vendrá acompañado de notificaciones básicas.
- Se usarán notificaciones periódicas para recordar al usuario que acceda a la aplicación para controlar rutinas cotidianas.
- Proveerá al usuario de pequeñas utilidades de toma de apuntes y notas, gestión personal de tiempos, y ayuda en la toma de decisiones.
- Servirá al usuario como diario personal en el que podrá expresar aspectos relacionados con su día a día y las dificultades a las que se enfrenta, algo crucial para aquellas personas con TEA que se hallen en terapia continua.

1.10 Tecnologías utilizadas

Al ser éste un proyecto centrado en el desarrollo de una solución potencialmente destinada al público, se ha considerado la utilización de tecnologías que favorezcan una mayor difusión de la aplicación.

En ese sentido, se optó por la utilización del framework Flutter, para desarrollo de aplicaciones multiplataforma.

Flutter está potenciado por el lenguaje de programación Dart, el cual ha sido empleado en la construcción de los modelos de datos requeridos por los distintos componentes de la aplicación en su desarrollo.

Considerando la fecha en la que se ha llevado a cabo la implementación, se ha utilizado la versión 3.13.1 del framework Flutter, y la versión 3.1.0 del lenguaje Dart.

Durante el desarrollo se ha trabajado con la referencia de las dos plataformas móviles más populares:

- **Android:** se ha utilizado la versión 2022.1.1 del IDE (*Integrated Development Environment*, entorno integrado para desarrollo) Android Studio, el cual da acceso a varias versiones del Android SDK (*Software Development Kit*, kit para el desarrollo de software). A la hora de la compilación de aplicaciones ejecutables es necesario establecer, en la configuración del proyecto de Flutter para Android, la versión mínima y la versión destino del SDK, ambas correspondientes a determinadas versiones de Android, y por tanto a determinados dispositivos en variable grado de antigüedad. Así, se ha escogido como la versión mínima del SDK la 19, la cual corresponde a Android 4.4, una versión del sistema lanzada a finales de 2013 (Kelion, 2013), con la idea de que la aplicación funcione en cualquier dispositivo de bajo rendimiento lanzado durante los últimos 10 años y que cuente con una versión de Android igual o posterior a la citada.
- **iOS:** se ha utilizado la versión 14.2 del IDE Xcode, destinado a desarrollo para todas las plataformas de Apple. En la configuración del proyecto de Flutter para iOS, se ha establecido iOS 11 como la versión más antigua donde se ejecutará el proyecto, decisión equiparable con la tomada para los dispositivos Android, en materia de soporte de dispositivos con hasta 10 años de antigüedad – el dispositivo más antiguo compatible con iOS 11 es

el iPhone 5s, lanzado el 20 de septiembre de 2013 (Dillet, 2017) (Patel, 2013).

Los IDE y el framework de Flutter se han puesto en marcha en un entorno basado en macOS 12.6.8 arm64. Aunque Flutter puede ser utilizado desde cualquier sistema operativo de escritorio, la utilización de software de simulación de dispositivos móviles de Apple requiere el uso del sistema de escritorio ofrecido por dicha empresa. En búsqueda de una mayor agilidad en la puesta en funcionamiento de la aplicación para ambas plataformas durante la implementación, se determinó que macOS sería la opción más indicada para ejercer el desarrollo.

Para la escritura de código se ha utilizado Visual Studio Code, que a fecha de 12 de septiembre de 2023 está en su versión 1.82.0. Es una de las opciones recomendadas en la web oficial del framework Flutter, y se ha escogido por la flexibilidad que ofrece con su sistema de plug-ins, así como una experiencia previa en su uso.

1.11 Metodología de desarrollo de software

Se define como **metodología** al conjunto de métodos que se siguen en una investigación científica o en una exposición doctrinal.

Partiendo de la mencionada definición, la metodología en el desarrollo de software puede entenderse como la guía a seguir a lo largo de las distintas etapas del proyecto: análisis, diseño, desarrollo, pruebas y mantenimiento). Este procedimiento es crucial de cara al cumplimiento de los requisitos pedidos por un potencial cliente, en tiempo y forma, y nos arriesgamos a un posible fracaso del proyecto de no cumplir con la organización necesaria.

Visto esto, y sabiendo las dimensiones del proyecto, se ha optado por la utilización de Scrum. Scrum es un proceso de desarrollo ágil incremental basado en el trabajo conjunto entre los clientes y desarrolladores, en el que se divide el trabajo a hacer en sprints o iteraciones. Cada uno de estos sprints debe dar un resultado completo que ha de ser entregado al cliente (Huambachano, 2017).

SCRUM FRAMEWORK

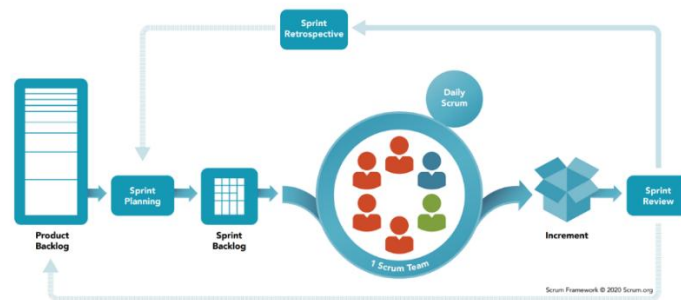


Ilustración 4 – Esquema que explica la Metodología Scrum

Es habitual asignar roles en la metodología Scrum:

- Propietario del producto (Product Owner): gestiona todo el proyecto desde un plano general, velando por el cumplimiento correcto de los objetivos.
- Scrum master: coordina la cohesión entre los distintos sprints. Es una figura esencial si el proyecto está siendo realizado por varias personas.
- Equipo de desarrollo: lo componen todos los miembros que colaboran en el proyecto, y se deben adherir a los límites determinados por el Product Owner.
- Cliente: aporta ideas y comentarios para asegurar un buen rumbo del proyecto.

Al ser un proyecto llevado a cabo por una sola persona, se deben hacer varias adaptaciones a la metodología. El autor de este Trabajo ha desempeñado los roles de Product Owner, Scrum Master y equipo de desarrollo. Las tareas a realizar a lo largo del proyecto se han fraccionado en la medida de lo posible para dividirlo en varios sprints que han permitido enfocar esfuerzos en una sola parte del análisis, diseño y desarrollo en cada iteración, agilizando drásticamente el proceso de creación de la solución y, por tanto, contrarrestando las limitaciones en materia de tiempo que se han comentado con anterioridad. Por otro lado, se ha contado con la ayuda externa y puntual de personas que han examinado y comprobado la aplicación en dispositivos reales, en sus distintas fases de desarrollo, a fin de encontrar bugs y problemas de funcionalidad. De esto se hará un detalle más exhaustivo en el apartado 3.8.

A continuación, se expone el tablero de tareas seguido para aplicar Scrum a este proyecto:

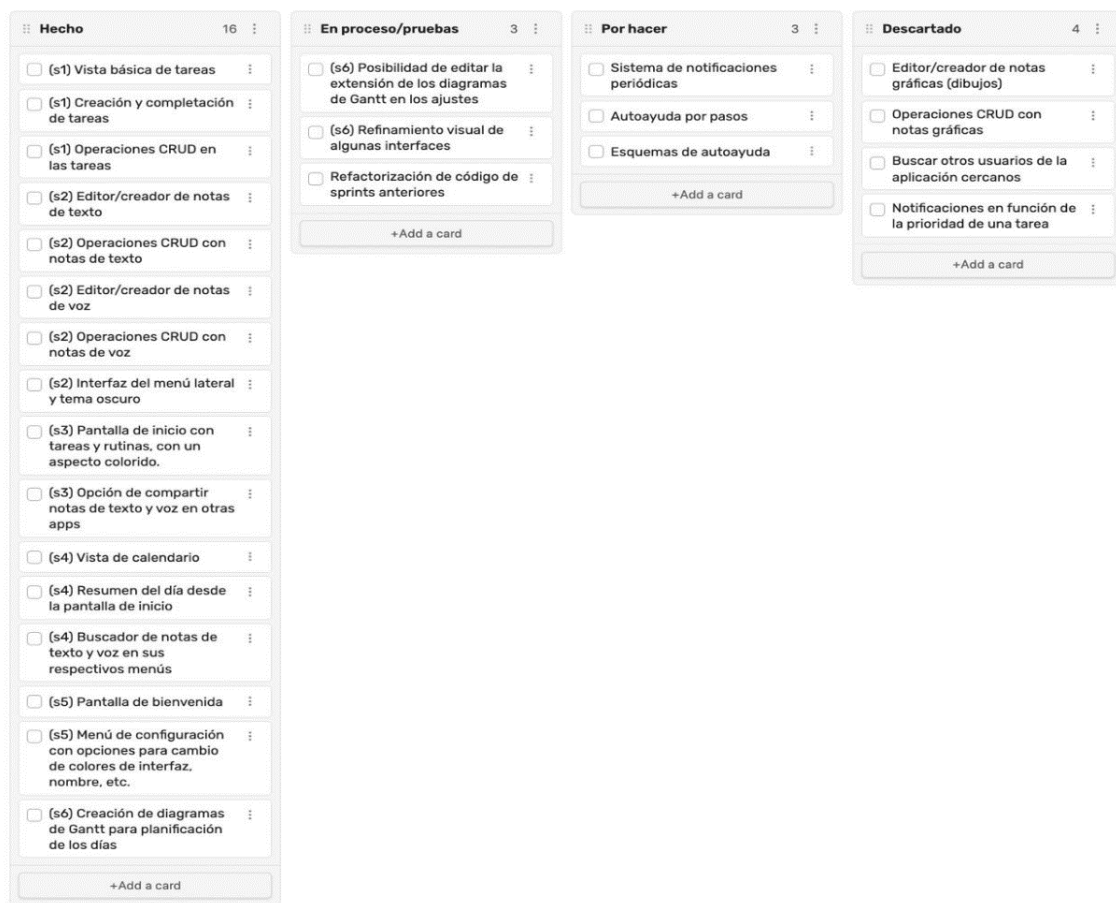


Ilustración 5: Ejemplo de tabla de tareas de Scrum en el sprint 6

1.12 Estimación del tamaño y esfuerzo

Ya que el presente proyecto es un TFT, no existen restricciones de tipo económico, sino de tipo temporal (un número aproximado de horas). Por consiguiente, los cálculos de tamaño del proyecto están supeditados el tiempo disponible. En cuanto al esfuerzo, se dispone de tan un solo efectivo (la persona autora del trabajo).

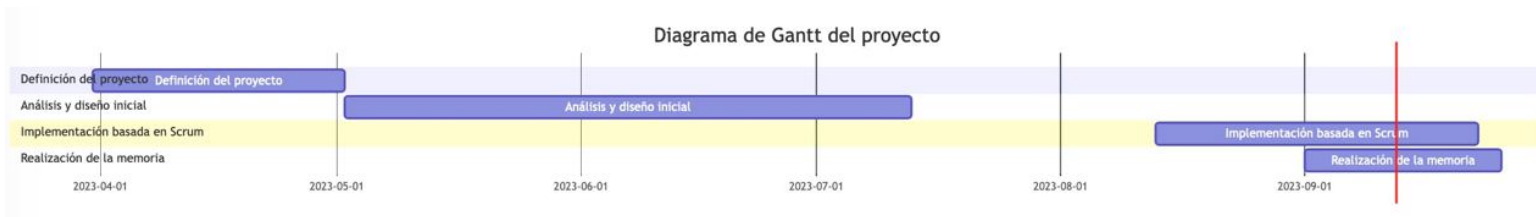
Las 300 horas se repartirán a lo largo de las siguientes fases:

- **Definición del proyecto:** 15 horas. Se dedican varias horas de forma espaciada debido a restricciones temporales tenidas por el autor del Trabajo.

- **Análisis y diseño iniciales:** 85 horas. Se dedican varias horas de forma espaciada debido a restricciones temporales tenidas por el autor del Trabajo y por las personas que han colaborado con la elaboración de requisitos y propuestas de diseño.
- **Implementación:** 180 horas. Estimación considerada en base a una dedicación de entre 6 y 7 horas al día en el período dedicado a la implementación (ver sección 1.13).
- **Elaboración de la memoria:** 20 horas. Se compagina con las fases finales de la implementación.

1.13 Planificación temporal

El presente es un diagrama de Gantt que estructura las fases de elaboración y desarrollo de este proyecto. Las limitaciones horarias a las que está sometido el Trabajo están supeditadas a la información detallada en el apartado 1.12.



Para cada parte primordial del desarrollo se han estipulado estas fechas:

- Del 31 de marzo al 2 de mayo de 2023 – Definición del proyecto.
- Del 2 de mayo al 13 de junio de 2023 – Análisis y diseño inicial, en conjunto con profesionales.
- Del 13 de julio al 13 de agosto de 2023 – Pausa por período vacacional.
- Del 13 de agosto al 23 de septiembre – Fase Scrum
- Del 1 de septiembre al 25 de septiembre de 2023 – Redacción de la memoria del Trabajo.

En la sección 3, Desarrollo, se hará un desglose temporal de la fase de implementación, acometiendo en cada sprint las pertinentes alteraciones al análisis y diseño inicial.

1.14 Presupuesto

Según datos contrastados en fuentes de la Universidad Alfonso X El Sabio (¿Cuánto gana un ingeniero informático? Descubre su sueldo, 2023), el sueldo medio de un ingeniero informático en España oscila entre los 20.450 € y los 76.800 €, horquilla variable según la experiencia del trabajador (junior o senior). Estimaremos los costes humanos considerando una experiencia equiparable a la de un desarrollador junior, así que partiremos de la cifra inicial de 20.450 €.

Al haber habido solamente una persona a cargo del proyecto, y considerando que el plan de estudios establece 300 horas para la realización del trabajo, podemos hacer una primera estimación de los costes humanos:

Coste por año	Coste por mes (12 meses)	Coste por día (suponiendo 22 días de trabajo en un mes)	Coste por hora (suponiendo 8 horas por jornada laboral)
20.450	1.704,17	77,46	9,68

Tabla 2 – Desglose de costes de personal humano

$$9,68 \text{ €} \times 300 \text{ horas} = 2904 \text{ €}$$

Para el desarrollo de la aplicación se han usado los siguientes elementos de hardware:

- Análisis, diseño e Implementación: Apple MacBook Pro 13inch (2020, M1), adquirido el 22 de febrero de 2021 (precio: 1.333 €). Se estima una vida útil de 5 años.
- Pruebas de desarrollo y calidad: Xiaomi Mi 10T Lite 5G, adquirido el 3 de febrero de 2021 (precio: 298 €). Se estima una vida útil de 3 años.

Suponiendo 8 horas de desarrollo diario a lo largo de 25 días para la parte de análisis y diseño (100 horas), y 30 días para la implementación (180 horas) y pruebas (20 horas), podemos estipular los siguientes costos:

$$\begin{aligned}
 &\text{Amortización/hora (hardware desarrollo)} = \\
 &= \frac{1333 \text{ € hardware desarrollo}}{5 \text{ años amortización} \times 365 \text{ meses/año}} = 0,73 \text{ €} \\
 &\text{Coste desarrollo} = 0,73 \text{ €} \times 25 \text{ días} = 18,25 \text{ €} \\
 &\text{Amortización/hora (hardware pruebas)} = \\
 &= \frac{298 \text{ € hardware pruebas}}{3 \text{ años amortización} \times 365 \text{ días/año}} = 0,27 \text{ €} \\
 &\text{Coste pruebas} = 0,27 \text{ €} \times 30 \text{ días} = 8,10 \text{ €} \\
 &\text{Coste análisis y diseño} = 0,73 \text{ €} \times 30 \text{ días} = 21,90 \text{ €}
 \end{aligned}$$

** Estimación temporal aproximada, basada en las ventanas temporales asignadas en el apartado 1.13. Los datos pueden diferir del esfuerzo horario real.*

Así, los costes totales del proyecto se definen de la siguiente forma:

Conceptos	Coste
Sueldo ingeniero	2904 €
Análisis y diseño	21,90 €
Desarrollo/redacción	18,25 €
Pruebas	8,10 €
Coste total	2952,10 €
Plusvalía	3542,52 €
Beneficio neto	590,42 €

Tabla 3 – Presupuesto total

2 DISEÑO INICIAL

En los siguientes apartados se hará una exhaustiva exposición del análisis y diseño previos a la primera iteración Scrum.

2.1 Especificaciones del sistema

2.1.1 Requisitos funcionales

Visto el análisis preliminar de requisitos realizado en el apartado 1.5, se parte de los requisitos iniciales allí expuestos para elaborar una relación de requisitos funcionales definitivos:

2.1.1.1 Gestión de tareas.

1. Un usuario podrá crear tareas nuevas.
2. Un usuario podrá establecer parámetros en dichas tareas, como prioridad y ámbito vital.
3. Un usuario podrá establecer si una tarea está completada.
4. Un usuario podrá determinar la eliminación de una tarea.

2.1.1.2 Toma de apuntes

5. Un usuario podrá disponer de un menú en el que acceder a sus apuntes de texto, y otro menú para las notas de voz.
6. Un usuario podrá crear una nota de texto con una extensión amplia para facilitar una expresión libre de ideas o pensamientos, acompañada de etiquetas en forma de “ámbito personal” y “estado emocional”.
7. Un usuario podrá crear una nota de voz con un formato de audio compatible con los dispositivos, y una extensión del audio ilimitada.
8. Un usuario podrá editar las notas de texto cuando éste lo requiera, de forma total.
9. Un usuario podrá editar los parámetros que acompañan a las notas de voz (nombre, ámbito personal y estado de ánimo) cuando se requiera

- 10.** Un usuario podrá reproducir las notas de audio, pausarlas y acelerar su reproducción.
- 11.** Un usuario podrá compartir notas en otra aplicación que admita compartición de archivos y texto. El formato de compartición de las notas de texto buscará la mayor legibilidad posible.
- 12.** Un usuario podrá eliminar notas cuando éste lo considere oportuno. Las notas serán eliminadas permanentemente en este caso.

2.1.1.3 Gestión de rutinas y pantalla de inicio

- 13.** Un usuario podrá gestionar de un vistazo sus rutinas diarias.
- 14.** Un usuario podrá disgregar de un vistazo la importancia de las tareas y rutinas que deberá realizar, mediante la utilización de recursos gráficos como colores e iconos.
- 15.** Un usuario podrá visualizar las tareas y rutinas realizadas en el transcurso de un día, así como un porcentaje orientativo de tareas resueltas.
- 16.** Un usuario podrá determinar cuál es su estado de ánimo en el presente día, e informar con un pequeño resumen sobre la trayectoria personal llevada durante el transcurso del día.
- 17.** Un usuario podrá acceder desde la pantalla de inicio a todas las funciones de la aplicación, a menos de 3 toques cada una, mediante el uso de menús y botones contextuales.

2.1.1.4 Gestión de planes por días

- 18.** Un usuario podrá crear tablas de planes para un día concreto. Estas tablas consistirán en diagramas de Gantt en cascada, que además permitirán visualizar colisiones en los planes.
- 19.** Un usuario podrá dilucidar, con un código de colores basado en el ámbito vital de cada plan, la prioridad o importancia de cada plan, para poder resolver posibles colisiones visibles en el diagrama de Gantt.
- 20.** Un usuario podrá editar de forma totalmente flexible la extensión horaria de cada plan, así como su nombre y el ámbito vital al que pertenece.

- 21. Un usuario podrá eliminar y agregar tareas a una tabla con total libertad.
- 22. Un usuario podrá crear tantas tablas de planes como considere conveniente, para planificar varias posibilidades de días, y favorecer un entorno y situación predecibles tan necesitados a veces por una persona con TEA.

2.1.1.5 Calendario con historial de días

- 23. Un usuario podrá consultar en una vista de calendario los registros que ha venido haciendo cada día desde la pantalla de inicio
- 24. Un usuario podrá consultar las notas de audio y texto que realizó en un día concreto, así como acceder a su edición, compartición o eliminación.

2.1.1.6 Experiencia de usuario general y configuración

- 25. Un usuario será recibido por vez primera por una pantalla de bienvenida en la que podrá configurar aspectos básicos de su experiencia en la aplicación.
- 26. Un usuario podrá acceder a un menú de configuración en el que podrá cambiar su nombre, colores de la aplicación, así como la extensión de las tablas de planes (diagramas de Gantt) en una dimensión horaria.

2.1.1.7 Adenda a los requisitos funcionales realizada en el sprint 7 del desarrollo: sistema de toma de decisiones mediante ruleta

- 27. Un usuario podrá verse ayudado en la toma de decisiones personales mediante un aleatorizador de opciones dispuesto en forma de ruleta.
- 28. Un usuario podrá disponer de varios aleatorizadores y personalizarlos con los elementos que considere oportunos.

2.1.2 Requisitos no funcionales

- 1. La aplicación tendrá un despliegue visual llamativo. Esto se traducirá en un uso destacado de colores, iconos, etc.
- 2. La aplicación deberá ser ligera en cuanto a su funcionamiento, pero a la vez contar con numerosas características ágiles y flexibles para el usuario.

3. La aplicación será flexible en su configuración, ofreciendo numerosas posibilidades al usuario.
4. El centro neurálgico de la aplicación será su pantalla de inicio. Desde ahí se podrá acceder a la configuración de la misma, creación de nuevas tareas o notas de texto/voz, vista de todas las tareas y rutinas, resumen del día presente, y acceso a un menú contextual para acceder a otras funciones como la vista de calendario, el menú de las tablas de planes, entre otras funciones.

2.1.3 Casos de uso

En los siguientes subapartados se plasmarán diversos diagramas de flujo que representarán los casos de uso más relevantes que se han contemplado en el proyecto. Al haberse considerado solamente un rol en la definición del proyecto y requisitos, así será en el diseño de los casos de uso, con la salvedad de que se considerarán las posibles necesidades diversas que el usuario pueda tener (Definición de casos de uso, 2021).

2.1.3.1 Caso de uso – Gestión de tareas diarias

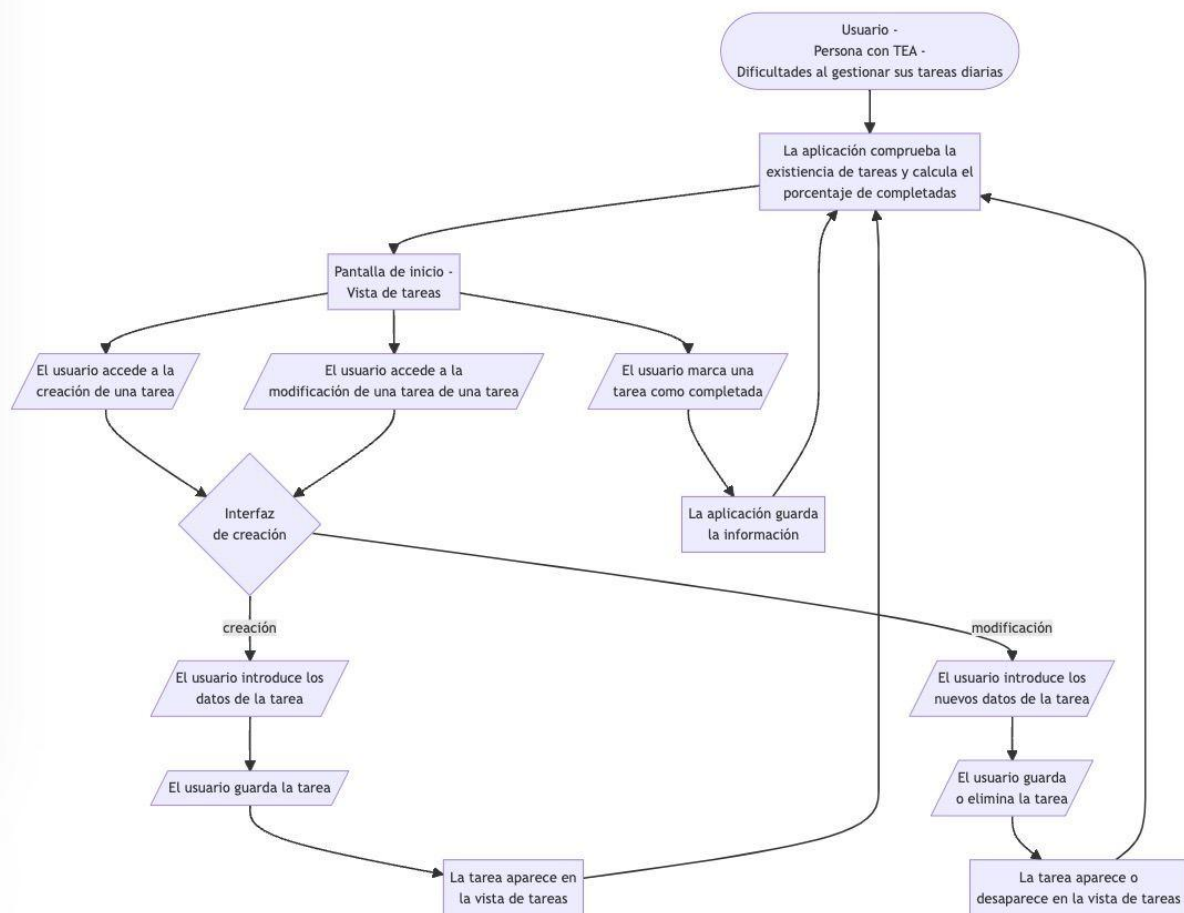


Ilustración 6 – Diagrama de casos de uso en el que se expone el funcionamiento teórico del sistema de tareas en la aplicación

2.1.3.2 Caso de uso – Gestión de notas de texto y voz

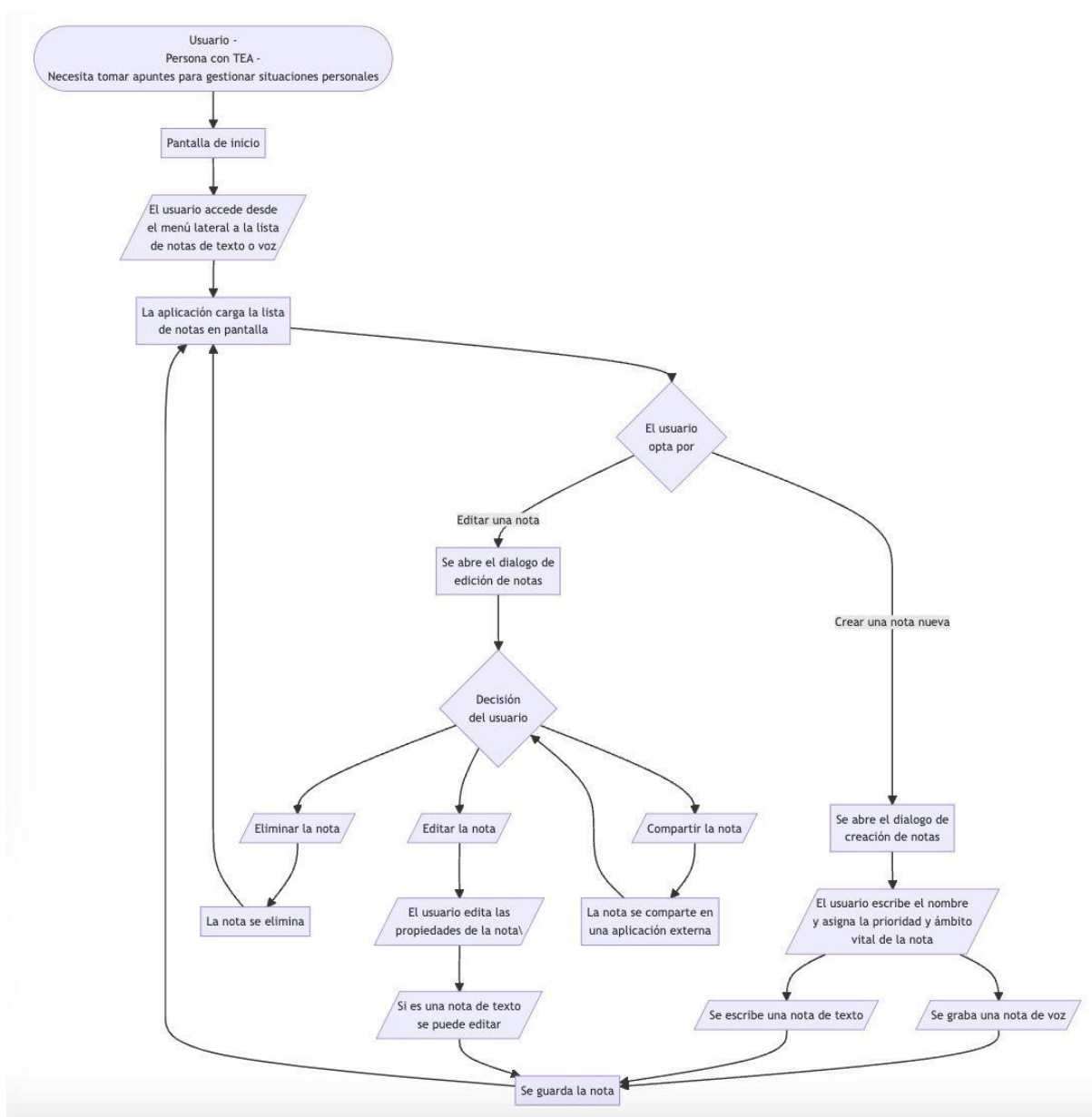


Ilustración 7 - Diagrama de casos de uso en el que se expone el funcionamiento teórico del sistema de notas de texto y voz en la aplicación

2.1.3.3 Caso de uso – Gestión de tablas de planes (diagramas de Gantt)

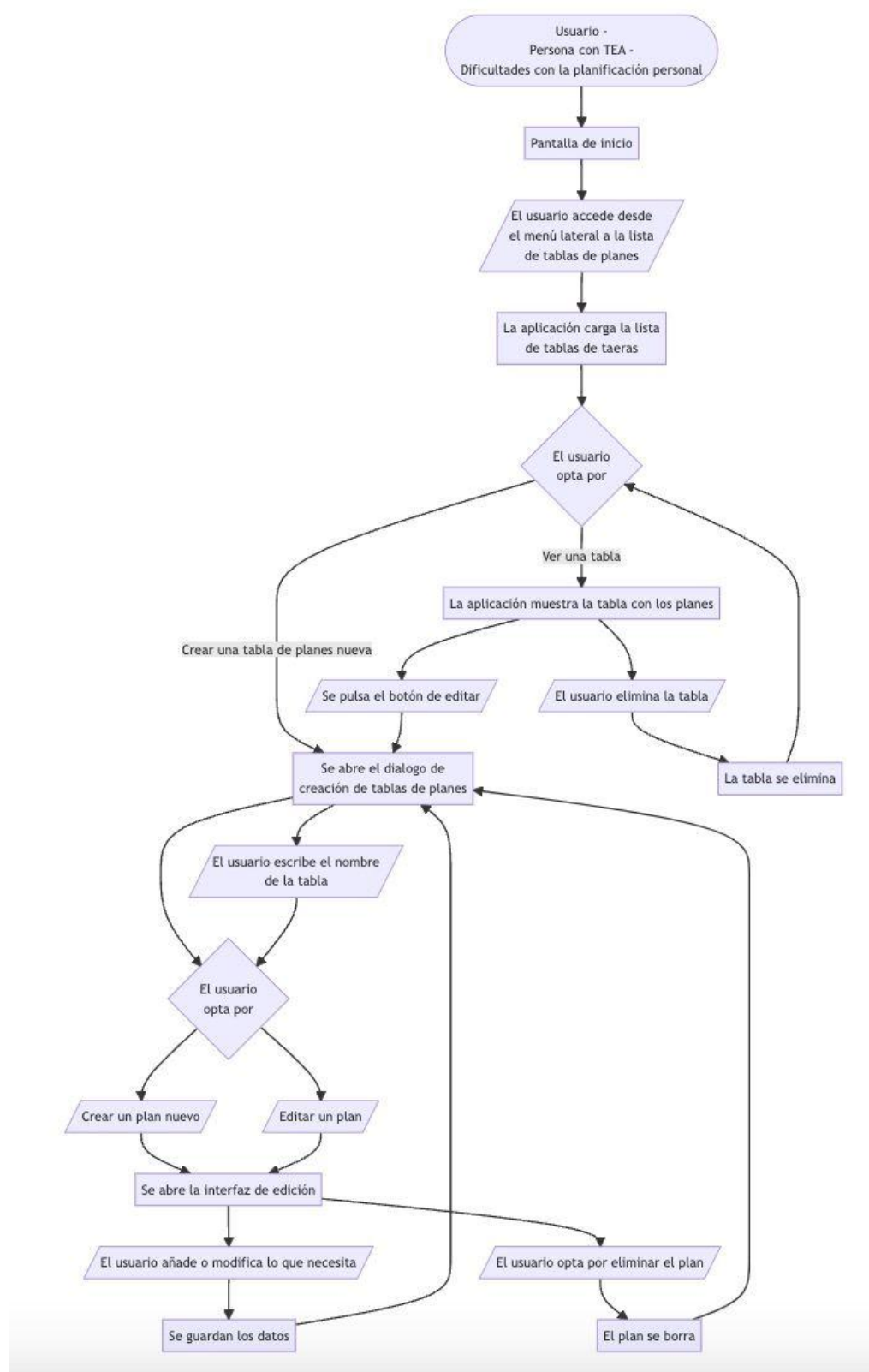


Ilustración 8 – Diagrama de casos de uso en el que se expone el funcionamiento teórico del sistema de tablas de planificaciones en la aplicación

2.2 Análisis y diseño del sistema

En el apartado anterior se han expuesto los requisitos y los casos de uso que gozan de mayor relevancia en el proyecto. A continuación, se darán detalles relativos al diseño de la interfaz y las características que se implementarán en la aplicación.

Para varias de las características planteadas, como las notas de voz, el almacenamiento de datos en archivos JSON, o la implementación de botones flotantes con menús contextuales, es necesario recurrir a plugins externos de Flutter.

Una de las ventajas más significativas de Flutter es su extensa colección de plugins de terceros que permiten implementar un número bastante nutrido de características y funcionalidades.

Por consiguiente, y visto todo lo anterior, se comenzará el diseño de la aplicación sobre el framework Flutter (Flutter - Get started, 2023).

2.2.1 Almacenamiento de archivos

En función de los requisitos, la aplicación manejará una serie de datos (notas de texto, notas de voz, tareas, rutinas, tablas de planes...) que deben ser almacenados internamente en la propia aplicación.

Para el prototipo implementado, se ha considerado el uso de archivos JSON. Los archivos JSON poseen la ventaja de tener un formato cómodo para ser leídos por máquina, y, si cabe, favorecer una expansión futura de características, como almacenamiento y manejo en la nube de estos archivos.

En búsqueda de un debido método para evitar colisiones por nombres en el archivo JSON, se ha optado por el plugin de Flutter `uuid`, que genera códigos UUID (*universally unique Identifier*: números de 128 bits para disminuir drásticamente la probabilidad de colisión) (Rombiola, 2014) asignables a cada objeto dentro del archivo JSON (sea una rutina, tarea, tabla de planes, nota de texto o referencia a nota de voz).

Para evitar pérdidas de datos de otras partes de la aplicación en el guardado en los archivos, se ha determinado el uso de varios archivos JSON que controlen las diversas funcionalidades. El gasto extra de almacenamiento es mínimo, pues hablamos solamente de pocos kilobytes.

2.2.2 Interacción con notas de texto y voz

Para crear una nota de texto o voz, se plantea la modelación de una interfaz que permita establecerles un nombre, ámbito vital y un estado de ánimo que determine el contexto emocional del usuario en el momento en el que toma la nota.

Hablar o escribir sobre situaciones ocurridas es muy importante para toda persona con TEA que esté pasando por una terapia cognitivo-conductual, por tanto se hace mucho énfasis en una toma de apuntes con unos límites difíciles de alcanzar por procedimientos comunes. Así, se plantea que las notas escritas tengan un mínimo de 4000 caracteres de límite.

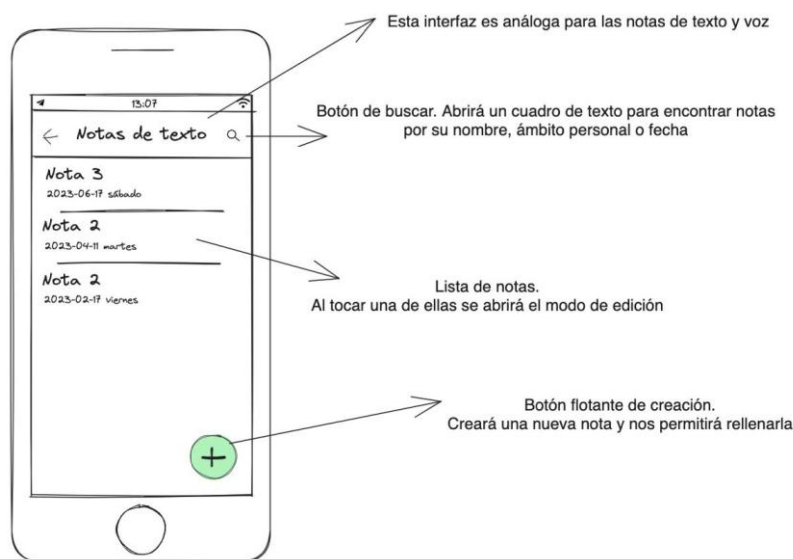


Ilustración 9 – Sketch diseñado para modelar las vistas de las listas de notas.

Para la efectividad de la búsqueda, se ha requerido del uso del plugin **diacritic**, para que el buscador implementado ignore acentos y diéresis existentes en los títulos de las notas.

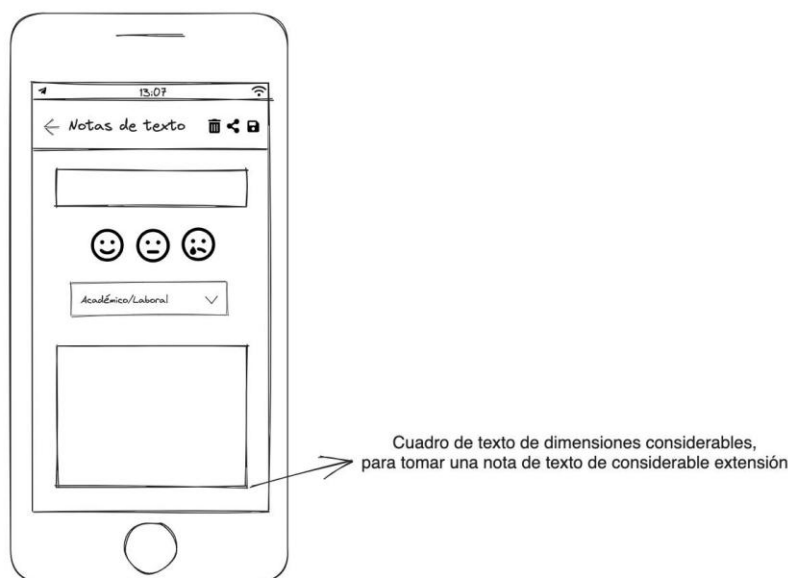


Ilustración 10 - Sketch diseñado para la interfaz de redacción de nuevas notas de texto

2.2.3 Notas de voz

Las notas de voz deberán almacenarse en un formato eficiente que encuentre un buen compromiso entre almacenamiento ocupado y buena calidad de los clips de audio.

Para este menester, se ha tomado a consideración el plugin `record`. `record` es un plugin de Flutter que permite implementar la grabación de audio proveniente del micrófono del dispositivo. Incorpora varias funciones para cambiar aspectos como la tasa de bits, formato de codificación, o dispositivo de entrada predeterminado.

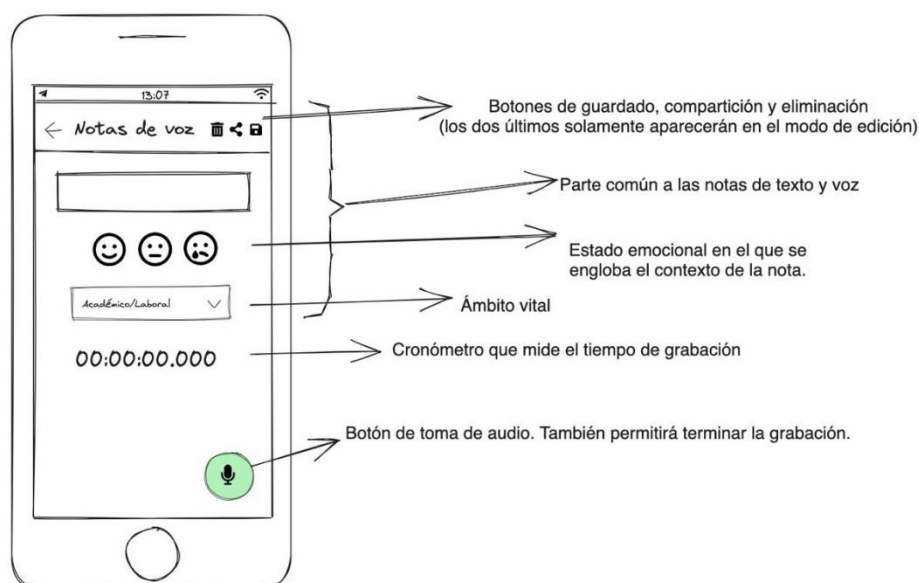


Ilustración 11 - Sketch diseñado para la interfaz de grabación de nuevas notas de voz

Por defecto, en las plataformas móviles principales se usa el codificador de audio AAC LC (el cual se usa por defecto en el plugin, y así se especifica en el código del mismo (lifbandit, 2023)), que equilibra una calidad correcta de grabación y una gestión comedida del almacenamiento del audio.

Para la reproducción del audio, se contempla el uso del plugin `flutter_sound`, el cual permite crear botones que permitan la reproducción del medio, así como alterar aspectos como su velocidad de reproducción.

2.2.4 Tablas de tareas

La modelación de tablas de tareas quiere responder a la necesidad del potencial usuario de organizar de forma precisa sus días, favoreciendo circunstancias predecibles que le permitan desarrollar sus quehaceres con normalidad.

Las tablas de tareas se basarán en **diagramas de Gantt**. Los diagramas de Gantt permiten estructurar espacios horarios de forma visual, así como facilitar la toma de decisiones en caso de colisión horaria entre dos planes o eventos que se tuvieran en consideración al crear una tabla.

Así, por cada tarea establecida se ocupará una fila en el diagrama de Gantt, permitiendo una vista en cascada horizontal. La prioridad de cada plan se establecerá según el ámbito vital al que pertenece, siendo así más sencillo de solucionar colisiones, que vendrán representadas por el hecho de que en un determinado espacio horario habrá dos barras correspondientes a dos planes diferentes que coinciden en el tiempo.

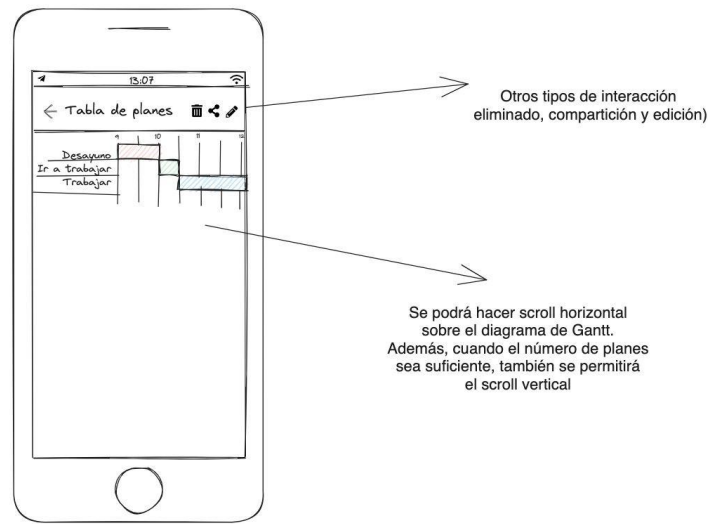


Ilustración 12 - Sketch diseñado para las tablas de tareas, con su estructura de diagramas de Gantt.

2.2.5 Vista de calendario y anotaciones diarias

A modo de diario, se tiene planteada la posibilidad de que el usuario pueda anotar cada día aspectos sobre su experiencia en el mismo. Para esto hay varias opciones, bien desde la pantalla de inicio o bien desde el menú lateral.

La idea concebida reside en incitar al usuario a que resuma brevemente cómo le ha ido, y estos datos se almacenarán en una vista de calendario por meses. Para el calendario, se ha utilizado el plugin `table_calendar` de Flutter (Wozniak, 2023), que dispone todos los días de un mes en una vista de tabla para poder interactuar con cada uno.

Al acceder a un día concreto, se podrá comprobar lo que se apuntó dicho día, es decir, un resumen de cómo fue. Esto facilitaría mucho la puesta en contexto de muchas situaciones en el dado caso que el usuario acuda a terapia, alcanzando mejores soluciones a las mismas. Además, se plantea también que se puedan acceder a las notas de voz y texto llevadas a cabo ese día, así como las rutinas realizadas (estas representarán una mejor radiografía de lo efectiva que ha sido la organización personal en el día seleccionado).

2.2.6 Diseño de la interfaz de inicio

La interfaz deberá ser lo más sencilla posible, y contener en la primera pantalla todos los aspectos esenciales para el usuario: el potencial usuario de la aplicación podría no tener experiencia ante el uso de tecnologías informáticas de cierta complejidad, y por ello se primará el carácter intuitivo de la interfaz. Además, se ha planteado que el color juegue un papel protagonista en la experiencia de usuario, y esto pasa por los siguientes aspectos:

- Colores de interfaz personalizables.
- Pantalla de inicio con un llamativo despliegue cromático.
- Posibilidad de alternar el tema claro y oscuro.

Así, los diseños preliminares de la interfaz van en este sentido:

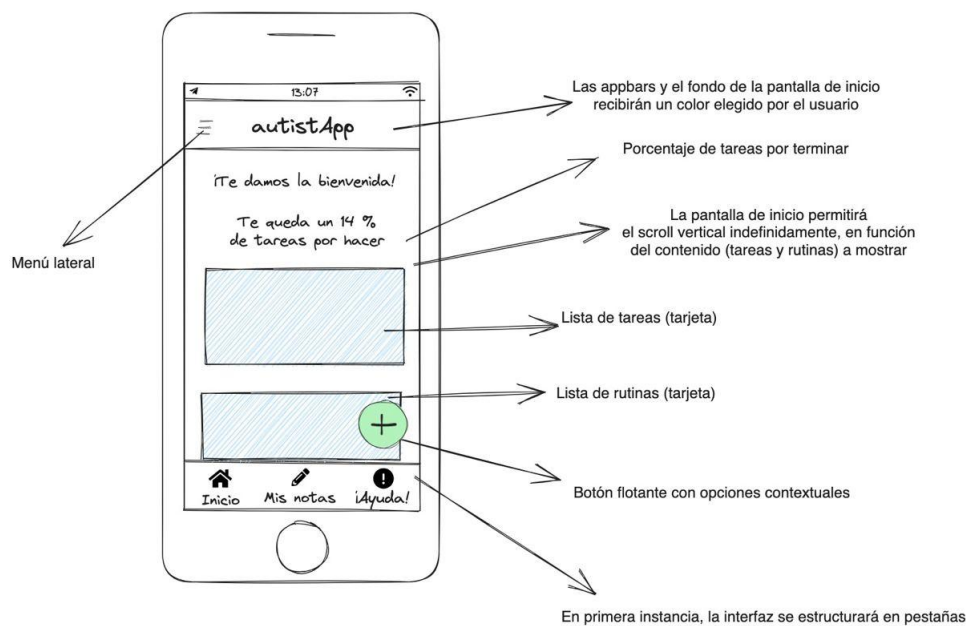


Ilustración 13 – Sketch diseñado para modelar la pantalla de inicio de la aplicación.

Flutter estructura las interfaces en base a los widgets. Cada widget, además, puede contener otros widgets (como botones, tarjetas, menús contextuales o cuadros de texto), así como elementos de control y posición (`Padding`, `Center`, `SingleChildScrollView`...). Lo concebido en este sketch será altamente similar a lo que se puede modelar en el prototipo final.

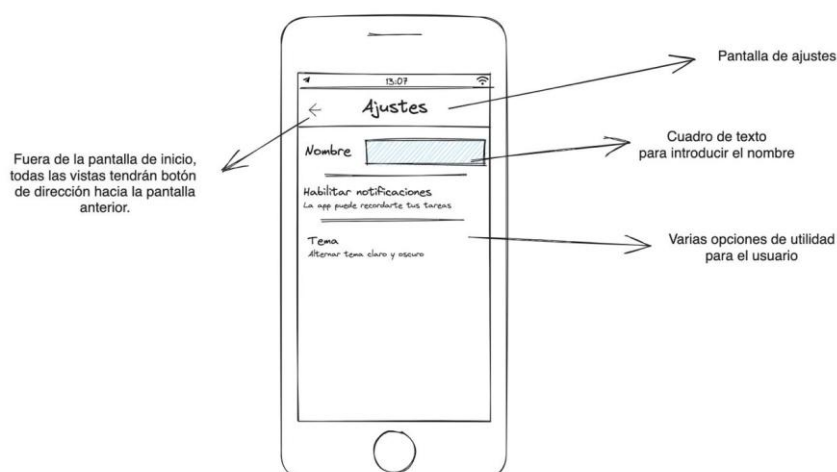


Ilustración 14 – Sketch diseñado para modelar la futura vista de ajustes de la aplicación.

3 DESARROLLO

Dado que se ha optado por una metodología de desarrollo ágil como es Scrum, se va a detallar el desarrollo en base a iteraciones (sprints). En cada sprint se van a tomar consideraciones en la alteración del diseño original, que a continuación se detallarán debidamente, junto al desarrollo y pruebas de cada parte.

3.1 Primera Iteración

La prioridad a seguir en esta primera iteración será la construcción de un sistema de gestión de tareas, que a la postre constituirá parte de la pantalla de inicio.

En la pantalla de inicio (aún poco desarrollada en este sprint) se ha dispuesto un cuadro con las tareas pendientes de terminar. Al mantener una tarea pulsada por 1 segundo, se abrirá un diálogo de edición de la misma. Si se desliza a la izquierda o derecha, la tarea se eliminará irreversiblemente de la lista. Para crear tareas se ha dispuesto un botón que da la opción para la creación de las mismas.

En la pantalla de creación/edición hay opciones para establecer un nombre para la tarea, un ámbito en el que la tarea se engloba dentro del paradigma vital del usuario, y una prioridad, estos dos últimos siendo elementos decisivos para que el usuario pueda dilucidar por sí mismo cuán importante es una tarea sobre las demás; además, se dispone de un diálogo para establecer recordatorios. Esto se desarrollará con más detalle en futuras iteraciones.

3.1.1 Diagramas de clases

Este diagrama de clases representa el modelo de datos que sustenta el concepto de “tareas” dentro de la aplicación:

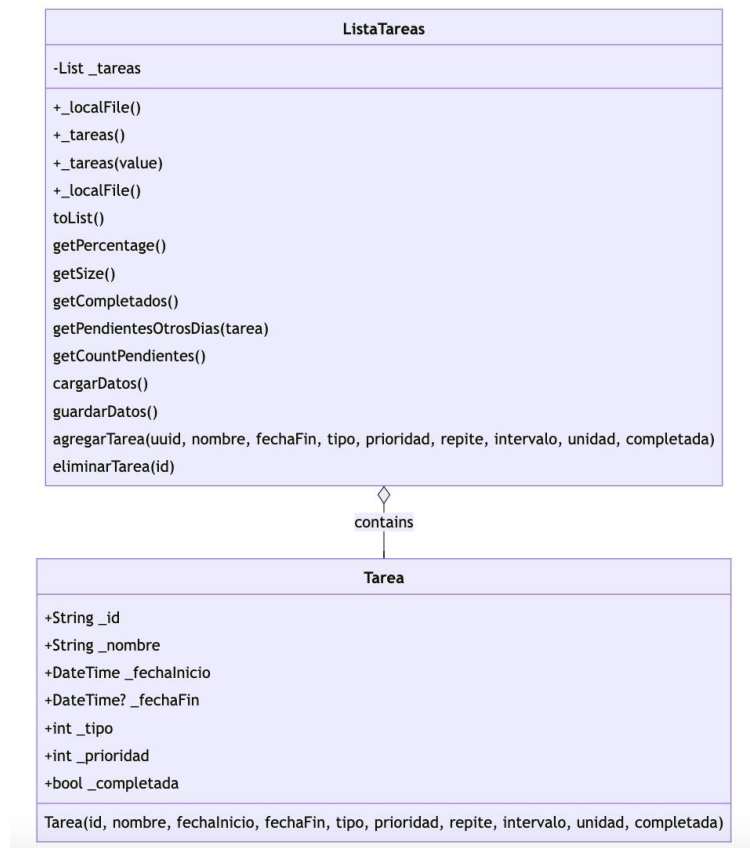


Ilustración 15 – Diagrama de clases del modelo de datos para las tareas

En la clase **ListaTareas**, los métodos **cargarDatos()** y **guardarDatos()** se encargan de guardar y cargar la información sobre las tareas a un archivo JSON. Los métodos **agregarTarea()** y **eliminarTarea()** también modifican los datos de dicho JSON. La clase **Tarea** no se ha modelado para usarse independientemente, sino para ser contenida dentro de una **ListaTareas**.

3.1.2 Diagrama de secuencia

Este es el flujo que el usuario sigue para gestionar una tarea:

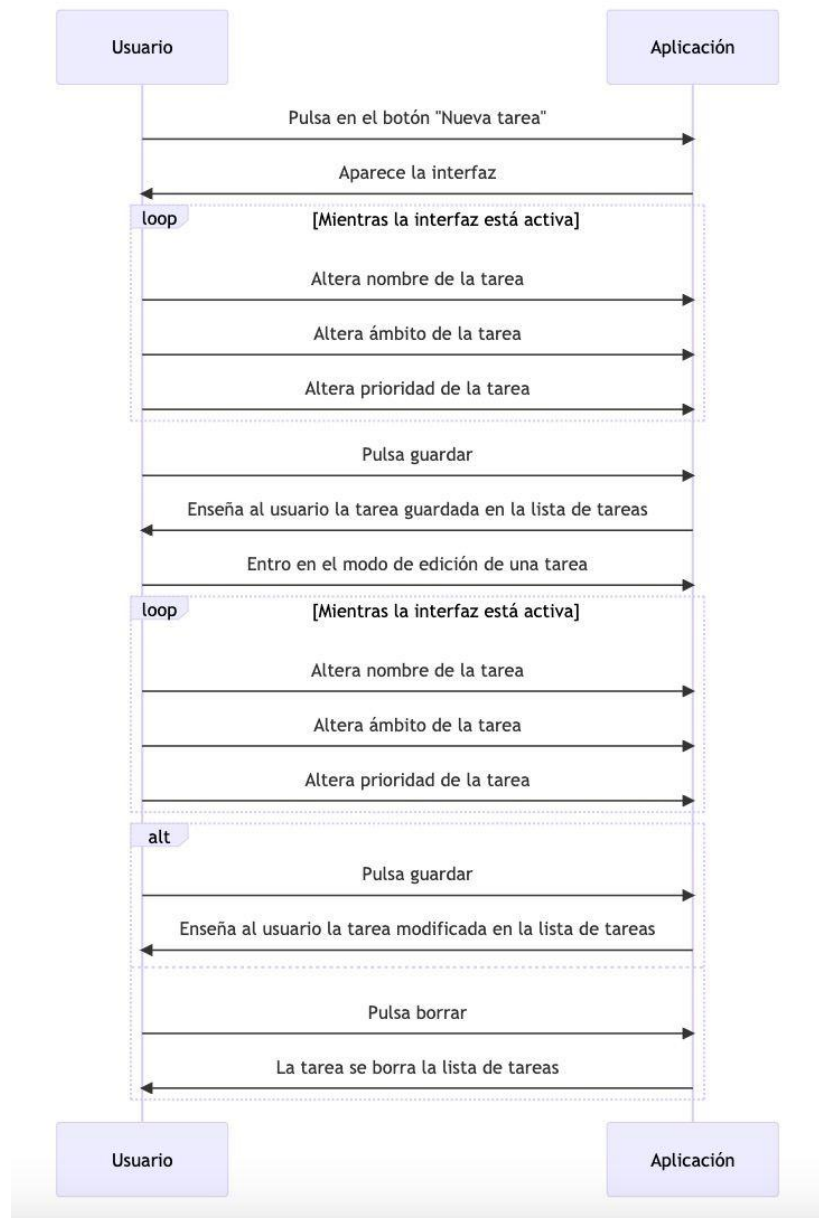


Ilustración 16 - Diagrama de secuencia en el que se expone la interacción entre el usuario y la aplicación al manipular tareas.

En los primeros sprint nos hemos valido de una clase `VistaTareas` que ha heredado del `StatefulWidget` de Flutter. Este widget consistía solamente en una vista `Scaffold` con una lista de tareas y un botón flotante que facultaba la adición de nuevas tareas.

3.2 Segunda iteración

En esta segunda iteración se construirá el sistema de toma de notas, tanto de voz como de texto. Se implementarán de forma básica las interfaces y menús que apoyan estas características de la aplicación.

Asimismo, se implementarán de forma mínima características de apariencia personalizada, como el cambio entre el tema claro y oscuro.

Por otro lado, se ha considerado la eliminación de la interfaz por pestañas de la pantalla de inicio. La interfaz es más limpia sin la implementación de pestañas realizada en el sprint 1, y además las listas de notas de voz y texto pueden ser accesibles con facilidad desde el menú lateral, que también será introducido en este sprint. En el futuro se contempla la adición de un menú contextual sobre un botón flotante, con el plugin de Flutter `speed_dial`.

Dos opciones en un menú lateral en la pantalla de inicio conducen a sendas listas de notas, de voz y texto respectivamente, con un botón para crear nuevas notas. Al acceder a la creación (o edición, al pulsar una nota existente) se dispondrá un cuadro de texto con el nombre, y dos filas de botones para determinar el ánimo emocional con el que se toma la nota y el ámbito vital al que pertenece. En el caso de las notas de texto, existirá un segundo cuadro de texto para escribir ahí la nota, mientras que para las notas de voz habrá controles multimedia de reproducción, pausa, grabación para notas nuevas, etc.

Es posible compartir las notas fuera de la aplicación, para su difusión a contactos a través de un mensaje, la descarga en la memoria principal del dispositivo, etc. Asimismo, la eliminación de notas es factible desde su pantalla de edición, una vez ésta ha sido creada.

3.2.1 Diagramas de clases

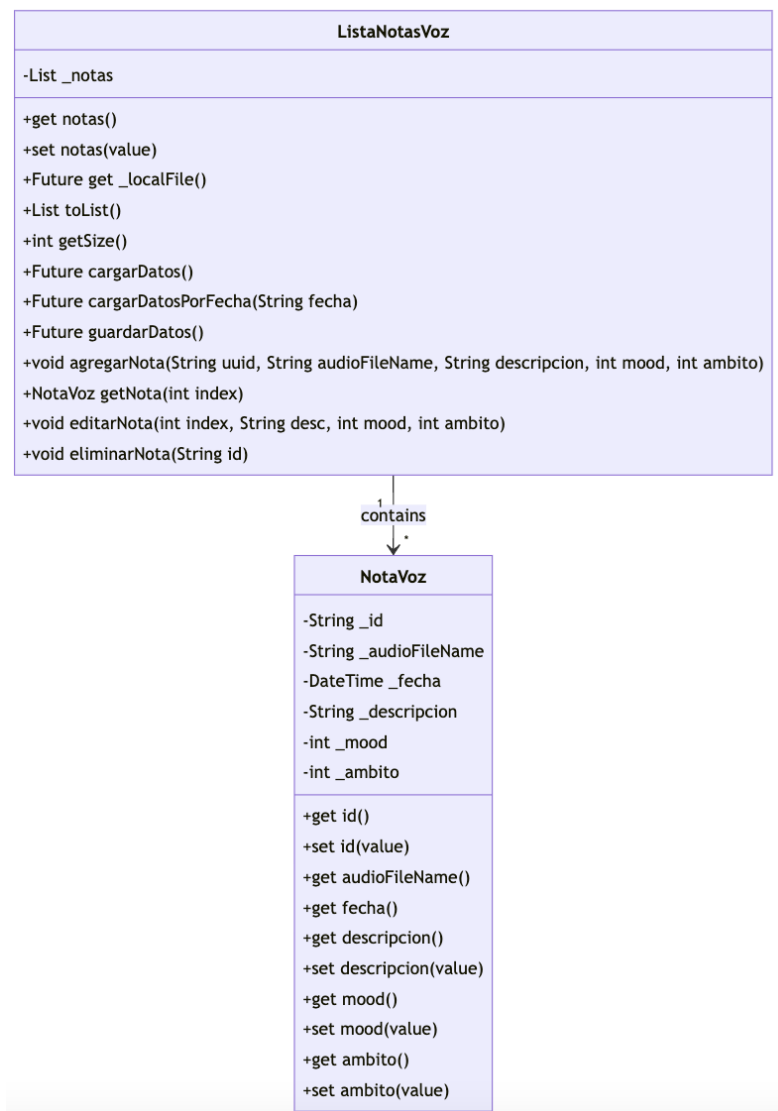


Ilustración 17- Diagrama de clases que exhibe la relación existente entre las clases *NotaVoz* y *ListaNotasVoz*, modelo de datos utilizado para las notas de voz

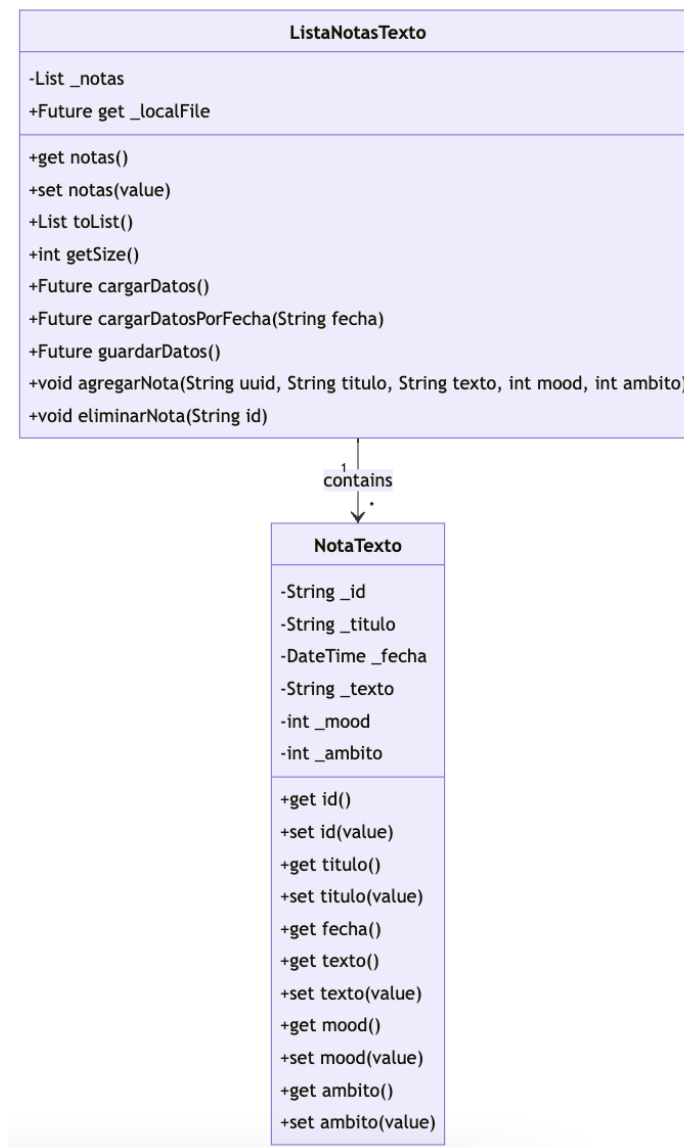


Ilustración 18 - Diagrama de clases que exhibe la relación existente entre las clases *NotaTexto* y *ListaNotasTexto*, modelo de datos utilizado para las notas de texto.

Los modelos de datos para ambos tipos de notas son altamente similares. Debido a que Dart permite declarar nulidad condicional con el parámetro `?` junto al tipo de dato (dart.dev, n.d.), se podría considerar una única clase *Nota* en futuras iteraciones del desarrollo, que pueda contener condicionalmente una `String texto` o una `String audioFileName`, para fomentar un código de mejor calidad y legibilidad.

3.2.2 Diseño de la interfaz

El diseño de la interfaz de lo que se va a implementar en esta iteración se ha cubierto mayoritariamente en el apartado 2.2.2 y 2.2.3.

El siguiente sketch define el acceso a la creación de nuevas notas de texto y voz, desde la futura pantalla de inicio:



Ilustración 19 – Botón flotante que invoca un menú con varias opciones, como la creación de nuevas tareas.

En el anterior sketch, se comprueba que en la interfaz ya no se cuenta con la interfaz con pestañas. Se ha tomado la decisión en búsqueda de una experiencia de usuario menos recargada, pero igualmente funcional. El plugin `speed_dial` que proporciona el menú del botón flotante ha sido clave en la toma de dicha decisión.

El diagrama de casos de uso definido en el punto 2.1.3.2 define de forma precisa las interacciones que el usuario hará con la función implementada en este sprint; no obstante, cabe reseñar que en dicho diagrama no se contempló la utilización del botón flotante previamente citado, dado que este ha sido un componente agregado durante la fase de desarrollo incremental.

3.3 Tercera iteración

En esta iteración el objetivo está en la construcción de una interfaz para la pantalla de inicio. Al haber eliminado las pestañas en el sprint 2, existe más espacio vertical (*screen real estate*) y por tanto la visibilidad de las tareas será mayor.

La lista de tareas, por su parte, se encerrará en un contenedor Card. La idea es crear una sensación de profundidad entre la mencionada lista y el fondo de la pantalla de inicio, algo que busca un mayor atractivo visual. Sobre la lista de tareas, se introducirá una sección de texto en la que se indicará el porcentaje de tareas completadas de entre las que hay en pantalla.

En la pantalla de inicio, además, se implementará en un futuro sprint un sistema para registrar el estado emocional del usuario en base a pictogramas (emojis) en forma de cara, representando las emociones. La interfaz que sustentará esta función, basada en botones, se implementará junto a la pantalla de inicio como tal.

Bajo la lista de tareas se encontrará una lista de rutinas, de apariencia similar pero distinto enfoque, destinada a ayudar al usuario con cuestiones vertebrales de la vida cotidiana, como consejos de salud, higiene u organización doméstica.

3.3.1 Diseño de la interfaz

En el apartado 2.2.6 se definió una pantalla de inicio sobre un sistema de pestañas, que carecía de varias de las opciones que se han planteado en la presente iteración.

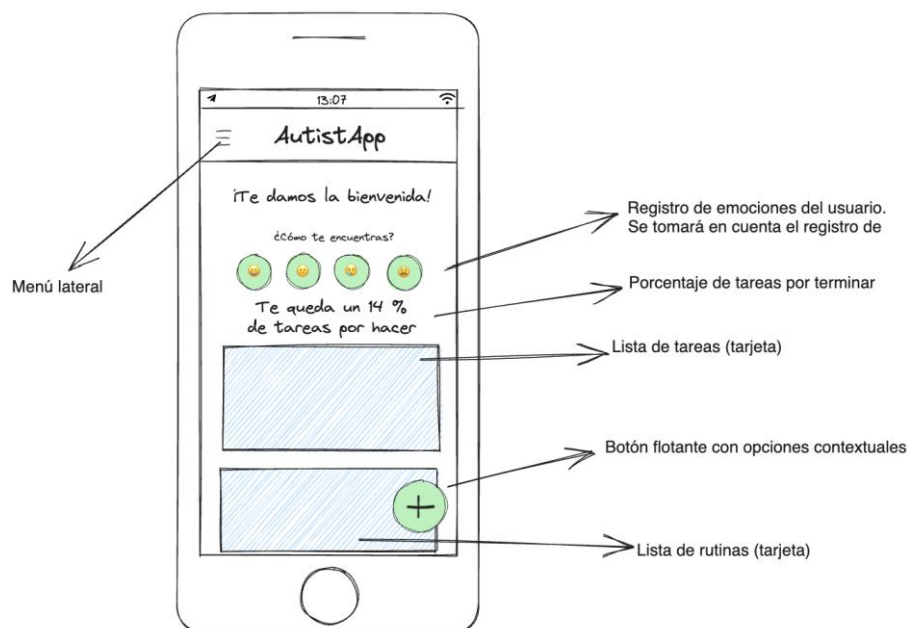


Ilustración 20 – Sketch donde se define la pantalla de inicio según lo estipulado en la presente iteración del desarrollo.

La revisión del diseño para la pantalla de inicio se define según el anterior sketch. Esta interfaz presenta todas las funciones básicas de la aplicación accesibles desde un único punto, en base a los nuevos iconos de emojis (que representarán el registro del estado emocional en el futuro), las listas de tareas y rutinas, el botón flotante implementado en la iteración anterior, y el menú lateral.

3.4 Cuarta iteración

Como se mencionó en la iteración anterior, se planteaba para el futuro la finalización de una característica cuya interfaz ya fue iniciada: el registro del estado emocional del usuario. Finalmente, esta característica será implementada junto a una de las más importantes: la vista de calendario. En la vista de calendario, se podrá acceder a los registros emocionales creados cada día, así como las notas de texto y voz generadas.

Además, en las listas de notas de texto y voz, se implementará un buscador que ejerza un sondeo en función al nombre de las notas, a su ámbito vital o a la fecha en que se tomaron.

3.4.1 Diagramas de secuencia

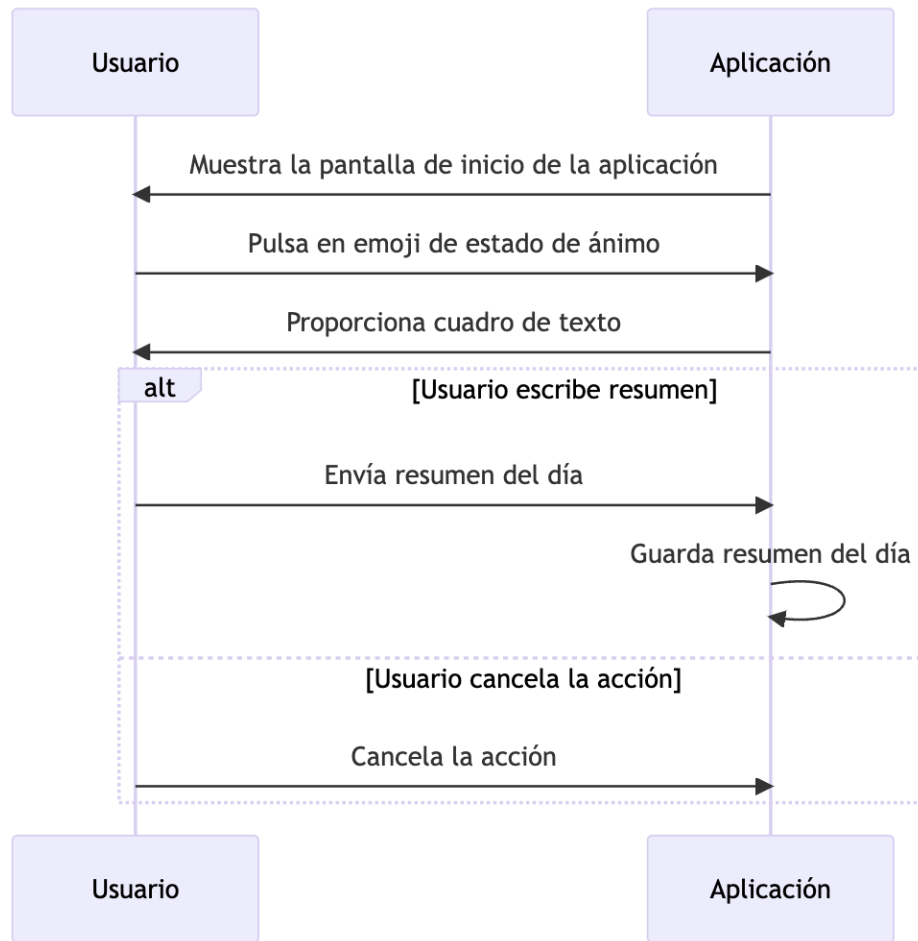


Ilustración 21 – Diagrama de secuencia que representa el funcionamiento básico de la función de registro del estado

3.4.2 Diagramas de clases

El modelo de datos básico que maneja esta función se sustenta en la clase Dia, una clase que se define así:

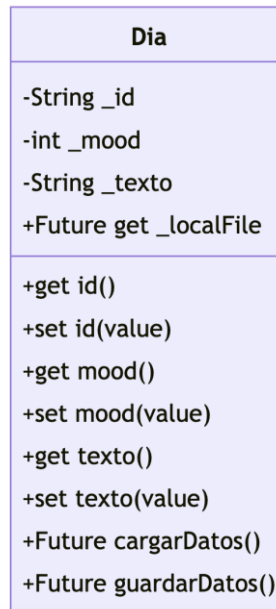


Ilustración 22 – Diagrama de clases de Dia.

Esta clase incorpora atributos como el día, el estado de ánimo elegido en la pantalla de inicio y el texto escrito tras dicha acción, así como atributos y métodos auxiliares para cargar y guardar estos datos en un archivo JSON que internamente tendrá un nombre único cada día, en base al estándar ISO 8601 para la estandarización de formatos de fecha (Dubost, 2003).

3.4.3 Diseño de la interfaz

Las nuevas interfaces diseñadas en esta iteración son la vista de calendario y la vista de resumen del día. Su diseño estará en consonancia con las interfaces ya implementadas.

En el caso de la vista de resumen del día, ésta contará con un scroll vertical continuo en caso de que el número de elementos exceda los que la pantalla del dispositivo pueda mostrar.

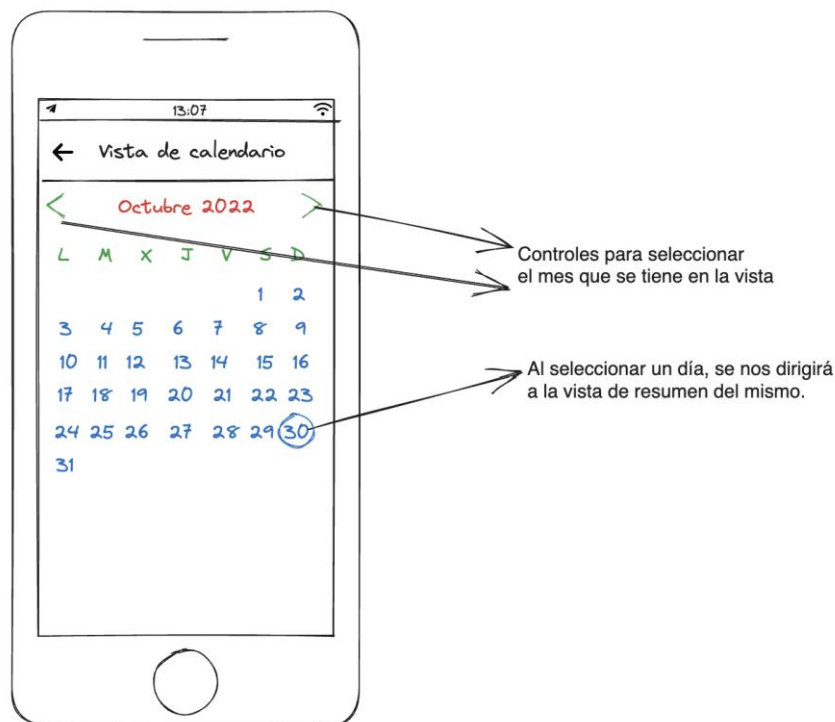


Ilustración 23 – Sketch que representa la vista de calendario.

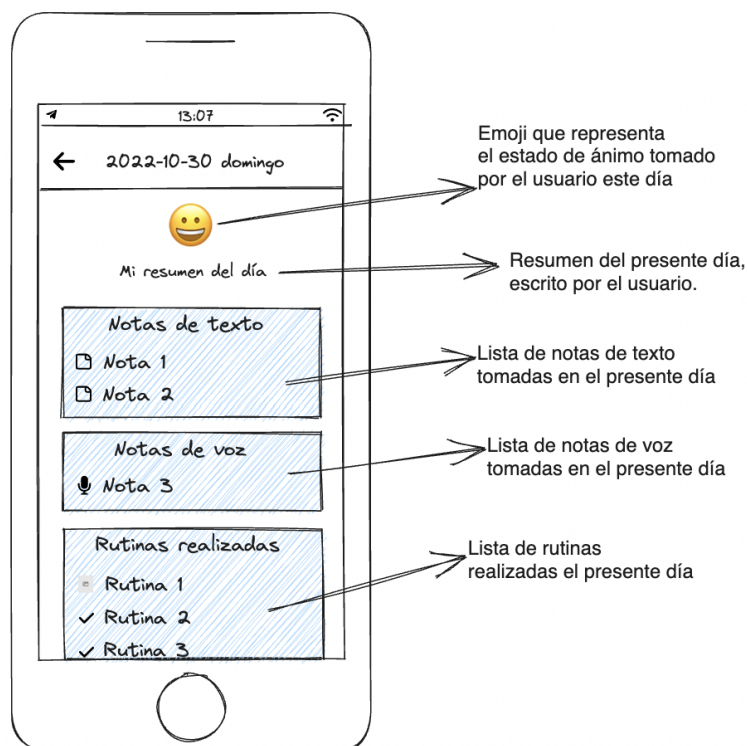


Ilustración 24 – Sketch que representa la vista de resumen de un día seleccionado en la vista de calendario.

3.5 Quinta iteración

Habiendo implementado buena parte de las características principales, se ha decidido crear una pantalla de bienvenida que reciba al usuario en su primer uso de la aplicación, y que permita al mismo establecer un nombre y un color para la interfaz.

Por su parte, la interfaz de la aplicación completa se adaptará al uso de dicho color. Para ello, cada clase de interfaz necesitará ahora cargar un objeto Ajustes, que contendrá la información necesaria en términos de colores de la interfaz. Este objeto puede ser alterado en una nueva pantalla, pantalla de Ajustes, en la que también se podrá cambiar el nombre con el que queremos que la aplicación nos reciba en la pantalla de inicio.

3.5.1 Diseño de la interfaz

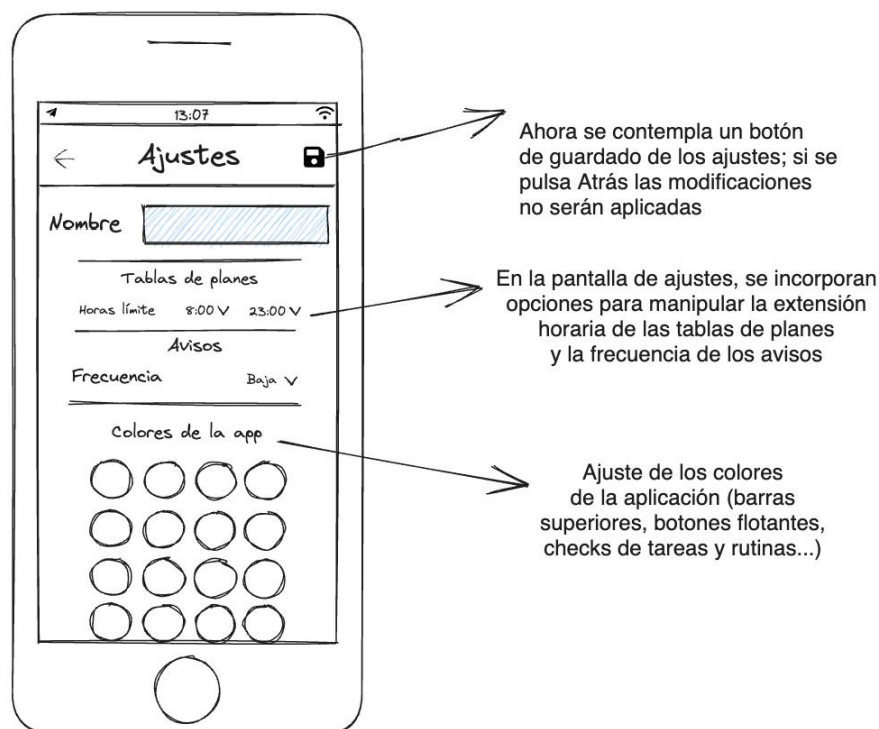


Ilustración 25 – Nuevo diseño contemplado para la pantalla de ajustes, que difiere del generado en la fase de análisis y diseño general (apartado 2.2).

3.6 Sexta iteración

En esta iteración se realizarán numerosas revisiones y refactorizaciones para incrementar la calidad y legibilidad del código y la interfaz de usuario. Además, se

implementará la característica de las tablas de tareas, pudiendo configurar en la vista de Ajustes la extensión que estas tendrán.

En el apartado 2.2.4 se ha detallado cómo se diseñarán las tablas de tareas, en base a diagramas de Gantt que definirán la dimensión temporal de cada tarea, ayudando al usuario a predecir las situaciones nuevas, y favoreciendo su autonomía.

Si bien existen plugins para Flutter que permiten la implementación sencilla de diagramas de Gantt, se va a optar por utilizar vistas de tablas (GridView) que simularán la estructura del diagrama.

Por otro lado, para la pantalla de edición de planes se prevé un diseño parecido al editor de notas y al de tareas, consistente en un editor de nombre, de ámbito vital (factor que determinará el color de un plan en el diagrama) y las horas de inicio y fin.

3.7 Séptima iteración

En esta última iteración se implementará una de las características que más importancia y utilidad toman de cara al usuario final: las notificaciones.

La utilidad de las notificaciones viene a solventar uno de los desafíos a los que se puede enfrentar el usuario: no recordar que debe abrir la aplicación para gestionar su día a día.

Como se ha analizado en el punto 1.3, “Antecedentes y estado del arte”, el hecho de que el usuario objetivo sea una persona con TEA implica que presenta una serie de dificultades en la organización y planificación personal. La sobrecarga sensorial a la que potencialmente se someten puede provocar una mayor agudeza de dichas dificultades. Por tanto, resulta de amplia relevancia que un sistema de notificaciones automático recuerde al usuario que debe planificar y organizar ciertos aspectos (algo que se traduciría en la acción de “abrir la aplicación”).

Sin embargo, se han encontrado numerosas dificultades técnicas a la hora de la implementación de las mismas:

- En los inicios de la fase de implementación, se estuvo contemplando la posibilidad de implementar notificaciones push utilizando servicios de mensajería de Firebase. Firebase es una plataforma de despliegue de aplicaciones para la nube respaldada por Google, y que aparte de una multitud de extensiones para implementar numerosas características, cuenta con integración para Flutter con el fin de crear aplicaciones móviles que dependan de conexión a internet para funcionar. Se tuvo que descartar esta posibilidad debido a que se buscaba que la aplicación de este Proyecto funcionara sin conexión, además de ciertas limitaciones técnicas en la implementación de notificaciones periódicas que no se consiguieron solventar.
- Por tanto, se tuvo que tomar la vía de las notificaciones locales. Mediante el plugin `flutter_local_notifications` (dexterv.dev, 2023) se consigue una implementación básica de notificaciones que puede ser fácilmente adaptable a las necesidades que el desarrollo exige. Esta extensión faculta la creación de notificaciones de una sola tirada mediante sus métodos `send()` y `zonedSchedule()`, para lanzar, respectivamente, avisos en el mismo instante o a una hora establecida con un objeto `TZDateTime`. El problema que ocasiona esto reside en que las notificaciones así lanzadas solamente se mostrarán una única vez. Para establecer recordatorios para tareas es, sin embargo, una solución ciertamente interesante porque permitirá recordarlas al usuario con posterioridad. Se han implementado mediante una interfaz dentro del editor de tareas, que permite establecer cuándo la aplicación nos recordará dicha tarea en el futuro, en un rango temporal de 1 hora, 3 horas, 6 horas, 1 día, 3 días o 1 semana. Estos plazos temporales se han escogido de forma arbitraria, pero con vocación de buscar un equilibrio para que el usuario tenga flexibilidad a la hora de gestionar sus tareas a corto plazo de cara al futuro
- Para el establecimiento de notificaciones periódicas que insten al usuario a abrir la aplicación, se ha optado por el plugin `workmanager` para implementar una tarea cada cierto tiempo, determinado al inicializar el plugin dentro del código de la aplicación. Se lanza una notificación con `send()` cada vez que el método `workManagerCallbackDispatcher()` se ejecuta; este último está

definido en la inicialización de `workmanager` y contiene la llamada al servicio de notificaciones.

Cabe resaltar que los métodos utilizados para las notificaciones no pueden lanzarlas con total exactitud, debido a limitaciones para las que no se ha tenido la capacidad de combatir, relativas al funcionamiento de los sistemas iOS y Android en materia de ahorro de energía. Por tanto, la frecuencia de las notificaciones periódicas es aproximada.

3.7.1 Funciones nuevas y descartadas

Respecto a otros aspectos introducidos en previas iteraciones, como el sistema de autoayuda frente a situaciones de emergencia, o el manual de uso que integrará la aplicación, se ha decidido descartarlos. El alcance del proyecto y su carácter de prototipo ha determinado que, habiendo sido definidos como aspectos importantes en la fase de análisis y diseño, y teniendo el proyecto aspiraciones de ser continuado más allá de la redacción del presente Trabajo, dichas características pueden ser implementadas con posterioridad, y sería más idóneo centrar esfuerzos en esta última iteración en la puesta en marcha de las notificaciones y alguna otra característica útil.

En consonancia con lo anterior, se ha ideado una nueva característica: ruletas de decisiones. Remitiéndonos a la sección 1.3 del Trabajo, se menciona que las personas con TEA suelen presentar dificultades en la toma de decisiones, que se pueden agudizar a medida que la persona crece y desarrolla su vida.

Un sistema sencillo basado en una ruleta con varias opciones podría representar una buena solución de cara al mencionado inconveniente: ante una situación con varias alternativas difíciles de clasificar por importancia (por ejemplo, planes para el tiempo libre personal), el usuario puede representar dichas alternativas en la ruleta, y ésta aleatoriamente escogerá una opción.

Así, se ha implementado una función en torno a este concepto. Para ello se ha optado por el plugin `flutter_fortune_wheel` (kevlatus.de, 2023), que ofrece amplia flexibilidad para la puesta en funcionamiento de widgets que simulan el comportamiento de una ruleta en la vida real.

De la misma forma que con las notas y las tablas de planes, se ha planteado la creación de un modelo de datos construido sobre archivos JSON que serán cargados y escritos según el contexto (botones pulsados, cambios de pantallas, etc.). El fundamento de este modelo está en la clase `Ruleta`, con campos que almacenan nombre, ID y fecha, así como arrays con los elementos importantes de cada ruleta (nombres de los elementos, veces que éstos han sido elegidos en la ruleta, etc.).

3.7.2 Diagramas de clases

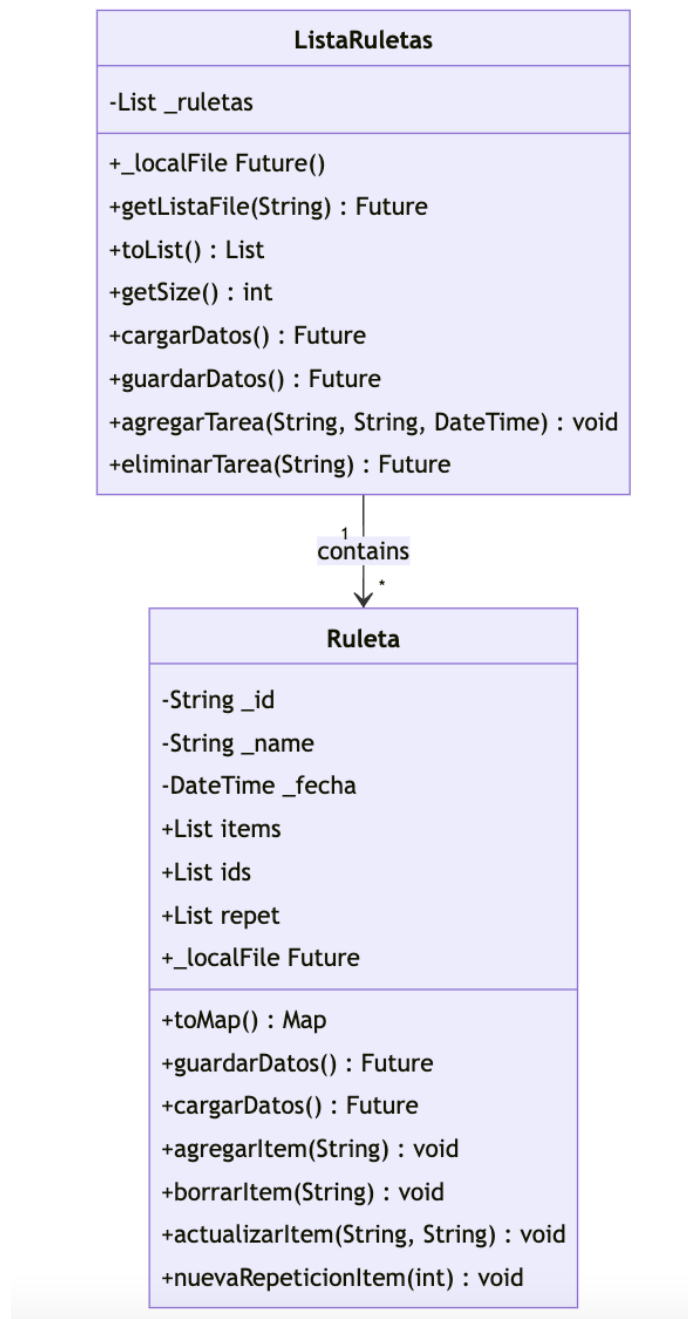


Ilustración 26 – Diagrama de clases para Ruleta y ListaRuletas. Se puede apreciar que, al igual que otros componentes de la aplicación, ésta se vale de archivos JSON para almacenar información persistente sobre cada ruleta y la referencia a la lista de ruletas.

Asimismo, también se ha implementado una clase **ListaRuletas**, que carga las referencias a los archivos de cada ruleta y los convierte a objetos **Ruleta**. Además, guarda los cambios en la cantidad y nombre de las ruletas, valiéndose de identificadores UUID para garantizar unicidad entre cada ruleta de forma interna.

3.7.3 Diagramas de casos de uso

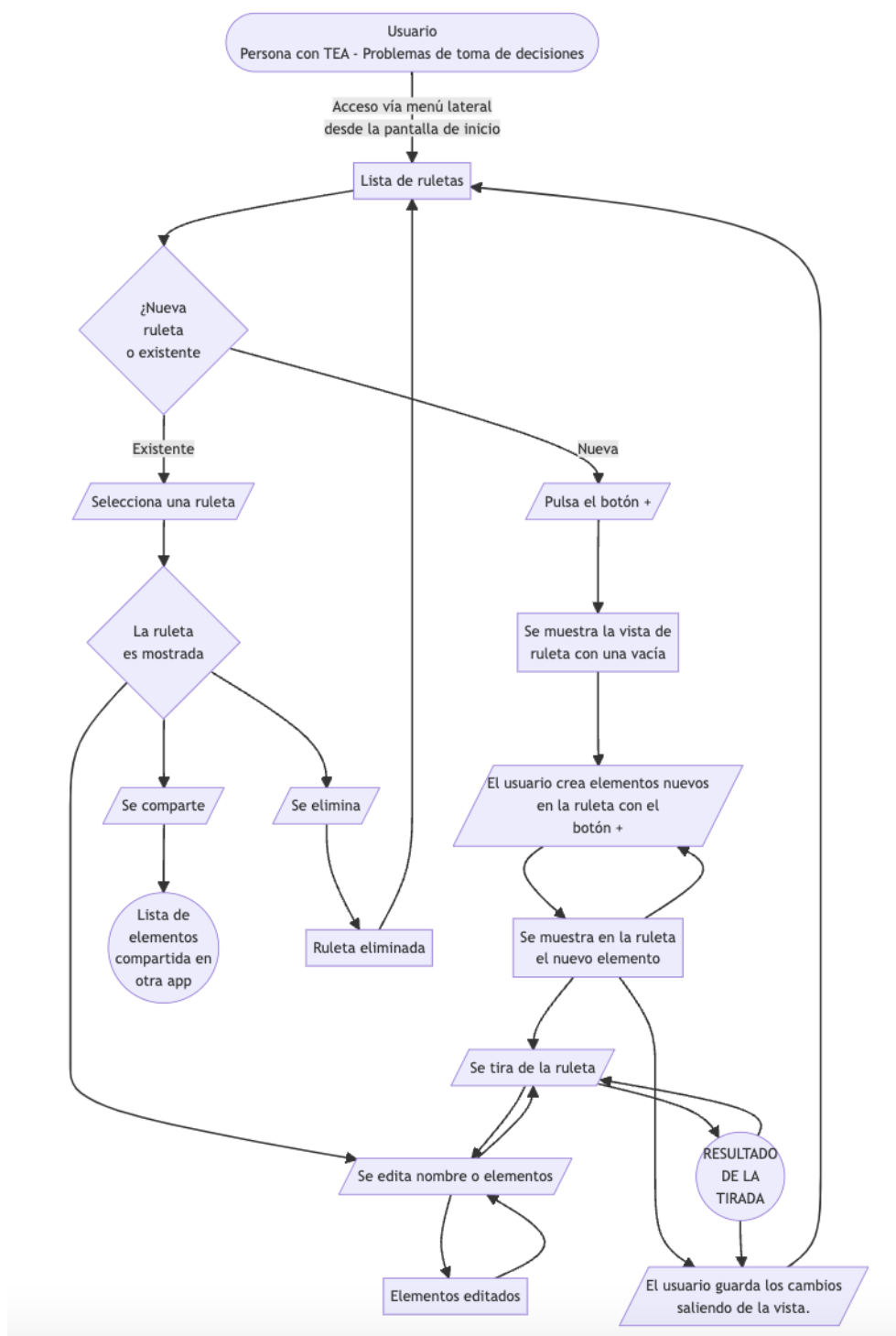


Ilustración 27 – Diagrama de casos de usos correspondientes a la nueva función de la ruleta de decisiones.

3.7.4 Diseño de la interfaz

La nueva característica de ruleta de decisiones prevé el diseño de la siguiente interfaz:

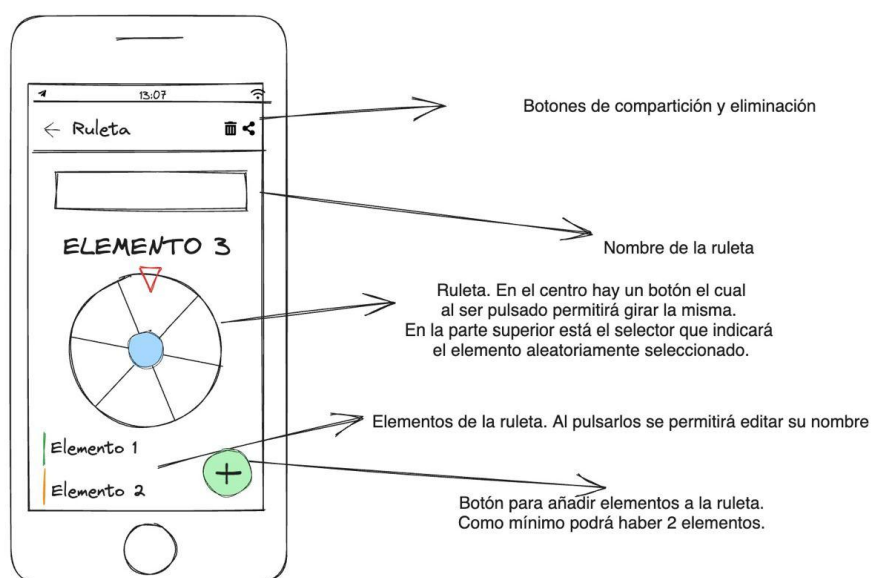


Ilustración 28 – Sketch que representa la función de la ruleta de decisiones, con sus elementos más destacables.

Para el acceso a la interacción con las ruletas que el usuario genere, se toma en consideración un menú similar al de las otras características anteriormente implementadas: notas y tablas de planes.

3.8 Pruebas finales

Después de la finalización del desarrollo incremental, se toma en consideración la ejecución de pruebas finales que constaten el alcance de la calidad del prototipo.

Al ser una aplicación planificada en torno al ámbito de la psicología, se ha visto de buen grado volver a tener tomas de contacto con profesionales de la materia para la realización de pruebas. Además, se ha tenido la oportunidad de que potenciales usuarios objetivo de la aplicación hayan podido realizar pruebas y comprobaciones de la funcionalidad de la misma, con el fin de obtener conclusiones de cara a aspectos mejorables en la interfaz o funcionalidades.

3.8.1 Pruebas de verificación y validación del sistema

Las pruebas de verificación y validación del sistema se centran en dar una garantía de cumplimiento de los requisitos estipulados. Dado que es un proyecto alrededor de una aplicación sencilla para dispositivos móviles, y el modelo de desarrollo incremental seguido, en cada iteración se detalla qué respuesta se ha dado a las necesidades planteadas inicialmente en el apartado 1.5, y posteriormente en el apartado 2.1.1 y 2.1.2, por lo que a priori no es necesaria una amplia exposición en el presente apartado.

3.8.2 Validación de la usabilidad del sistema

Para la validación de la usabilidad se ha seguido un exhaustivo proceso que ha prolongado levemente el tiempo de redacción de la presente Memoria de Trabajo. El tutor del Trabajo expuso, a fecha de 22 de septiembre de 2023, una sugerencia consistente en proporcionar el prototipo a determinados usuarios experimentales para que realicen pruebas de usabilidad.

Así, en los días posteriores, se dispuso de un archivo instalable de la versión de Android, que fue proporcionado a varios potenciales usuarios que poseen el diagnóstico de TEA. Por otro lado, el autor del trabajo ha realizado varias demostraciones de usabilidad a personas profesionales que trabajan o han trabajado ayudando a personas con TEA.

3.8.2.1 Demostraciones con profesionales

- El 23 de septiembre, se volvió a tener una tercera reunión con responsables de la Asociación HorizonTEAsperger de Jaén. En dicha reunión se concluyó un veredicto mayoritariamente positivo de la usabilidad de la aplicación, en consonancia con los requisitos estipulados y el desarrollo que sobre éstos se elaboró en la fase de análisis y diseño, sin advertirse detalles de gran índole sujetos a problemas.
- El 27 de septiembre, se tuvo una reunión con la Subdirectora de Estudiantes de la Escuela Politécnica Superior de Jaén, reunión con la se buscaron

objetivos análogos con el encuentro citado en el anterior punto, ya que la mencionada Subdirectora es responsable de alumnos con Necesidades Educativas Especiales que acceden a los estudios de Grado universitario y Master ofertados en la institución. En esta reunión se dieron algunas pinceladas acerca del desarrollo futuro de la aplicación:

- En la interfaz de la pantalla de inicio, sería interesante dotar de mayor flexibilidad al usuario a la hora de establecer los estados de ánimo (emojis) a definir cada día: más estados y mejor expresividad son claves para lograr este objetivo.
- Para introducir al usuario en las funcionalidades de la aplicación cuando éstas se usan por primera vez, es recomendable que existan ejemplos de las mismas. Por ejemplo: una nota de texto, ruleta o conjunto de tareas ya definidas; cada elemento en su respectiva sección.
- Dado el carácter de prototipo que condiciona el desarrollo actual de la aplicación, durante las iteraciones de Scrum se decidió establecer rutinas predefinidas, que sirven a modo de ejemplo y están consensuadas por la asociación con la que se definió la fase de análisis y diseño. No obstante, en la reunión se ha analizado la posibilidad de que en el futuro se puedan definir rutinas de forma libre y adaptada a las necesidades del propio usuario.
- El enfoque dado a las notas de texto y voz en la propia aplicación debería mejorar, para que el usuario pueda identificar fácilmente la función con sus objetivos. Al tener amplias posibilidades de uso, puede resultar problemático que el usuario encuentre una amplia utilidad en estas herramientas. Para presentar las funciones de notas de texto y voz se sugiere el uso de una palabra clave, fácil de identificar por la mayoría de usuarios potenciales: por ejemplo “autoayuda”.
- Se ha mencionado la posibilidad de que una persona con TEA puede ser propensa a asignar mentalmente un color dado a una situación positiva o negativa, algo que puede condicionar en la efectividad de la función de ruleta de decisiones. Es por esto que se sugiere que la ruleta

no sea multicolor, o que esta característica no sea obligatoria y pueda determinarse en la pantalla de Ajustes de la aplicación.

3.8.2.2 Pruebas con usuarios reales

En cuanto a las pruebas realizadas por potenciales usuarios, realizadas entre los días 22 y 27 de septiembre de 2023, se ha estado experimentando con dos usuarios, del entorno personal del autor del Trabajo. De ellos se esperaba un análisis en entorno real, explotando las características implementadas y el amoldamiento de las mismas en su vida cotidiana.

- La primera persona ha dado un veredicto positivo de la aplicación, si bien ha destacado cuestiones a refinar, relativas a la eliminación de elementos superfluos de la vista de calendario que podrían alterarla. Además, se ha tomado en consideración la lesa utilidad del cambio automático de colores en la pantalla de bienvenida.
- La segunda persona ha establecido posibles características a considerar:
 - La posibilidad de establecer tareas cuya ejecución deba comenzar en el futuro, y no a un plazo tan corto como el definido por el actual prototipo.
 - La capacidad de establecer una completa flexibilidad en la programación de notificaciones para las tareas.
 - La posibilidad de vincular tablas de planes con la vista de día y de calendario, para que se puedan establecer planificaciones para un día en concreto
 - Asimismo, ha realizado pruebas con las notificaciones, que al estar sujetas a los ajustes de ahorro de energía de los sistemas operativos móviles, han presentado dificultades de funcionamiento.

A las demandas de los mencionados usuarios de prueba, se dará una respuesta en el desarrollo posterior de la aplicación. Debido al carácter de prototipo que ésta posee, hay en general numerosas cuestiones de usabilidad que deben tomarse en consideración.

4 CONCLUSIONES Y TRABAJOS FUTUROS

El presente ha sido un Trabajo de Final de Grado consistente en el análisis, diseño y desarrollo de un prototipo de aplicación asistente personal para personas con TEA. La principal motivación residía en la falta de herramientas enfocadas al fomento de la autonomía personal de estas personas, así como la vocación de intentar facilitar la labor a los profesionales de la psicología que trabajan a diario con personas con TEA en edades en las que las exigencias académicas, laborales o sociales son más intensas.

A partir de lo desarrollado en este proyecto, se abre un abanico de opciones, que pasan por la expansión de prestaciones de la aplicación, ya sea con las ideas descartadas al aplicar la metodología MoSCoW, o ideas nuevas. La aplicación se ha diseñado con la siempre presente posibilidad de expandirla y publicarla, bien como aplicación de pago o como solución gratuita, o aplicando un modelo mixto en el que el código fuente esté alojado en un repositorio de código online y la aplicación final se consiga, previo pago, en las tiendas de aplicaciones. Esta cuestión merecerá una mayor instrucción en materia de licencias de software por parte del autor del presente Trabajo.

En cuanto a la disponibilidad para la mayoría de públicos, Flutter ha sido una opción acertada para el desarrollo de la aplicación, ya que en cuestión de poco tiempo se pueden generar binarios ejecutables e instaladores para las tres principales plataformas de escritorio, las dos principales plataformas móviles, e incluso como aplicación web. Así, en función de la decisión que se tome alrededor de la continuación del desarrollo de la misma, se valorarán las posibilidades de publicar la solución en múltiples plataformas.

5 APÉNDICES

5.1 Instalación y configuración del sistema

El código fuente de la aplicación se encuentra en el siguiente repositorio remoto de Git:

<https://github.com/manu-romero-411/autistapp.git>

En un sistema con Flutter instalado y configurado (en [este enlace](#) se pueden consultar más detalles sobre la puesta en marcha del framework) se clona el repositorio de Git previamente citado. Para compilarlo, se necesitará ejecutar un emulador de terminal, dirigir el Shell al directorio raíz del proyecto, y ejecutar:

```
flutter build apk --release
```

El anterior ejemplo se empleará para generar instaladores para Android, en formato APK (*Android Package*, paquete de Android). Para generar instaladores para otras plataformas, será suficiente con cambiar `apk` por las siguientes opciones:

- `appbundle`: construye una aplicación de Android en formato AppBundle, una alternativa al APK más reciente.
- `ios`: construye los archivos necesarios para ejecutar la versión de iOS (solo disponible en Flutter para macOS).
- `ipa`: construye la versión de iOS preparada para su distribución.
- `macos`: construye la versión de macOS.
- `web`: genera los archivos necesarios para la versión web (que será ejecutada por defecto en Google Chrome).
- `windows`: construye archivos ejecutables para Windows.
- `linux`: construye binarios para sistemas GNU/Linux.

El output de consola nos indicará la ruta del sistema donde se han generado los archivos. Una vez obtenidos, se trasladan a los dispositivos que vayan a ejecutar la aplicación, para acto seguido proceder a la instalación y ejecución.

5.2 Manuales de usuario

5.2.1 Puesta en marcha de la aplicación

En la primera apertura de la aplicación, tras su instalación, se recibirá al usuario con una pantalla de bienvenida. El usuario podrá escoger un color para la interfaz, así como escribir un nombre, aunque no son acciones obligatorias.

Al terminar la puesta en marcha de la aplicación, aparecerá la pantalla de usuario. A partir de aquí, se deben seguir los pasos expuestos en los diagramas de secuencia, para acceder a las determinadas funciones de la aplicación.

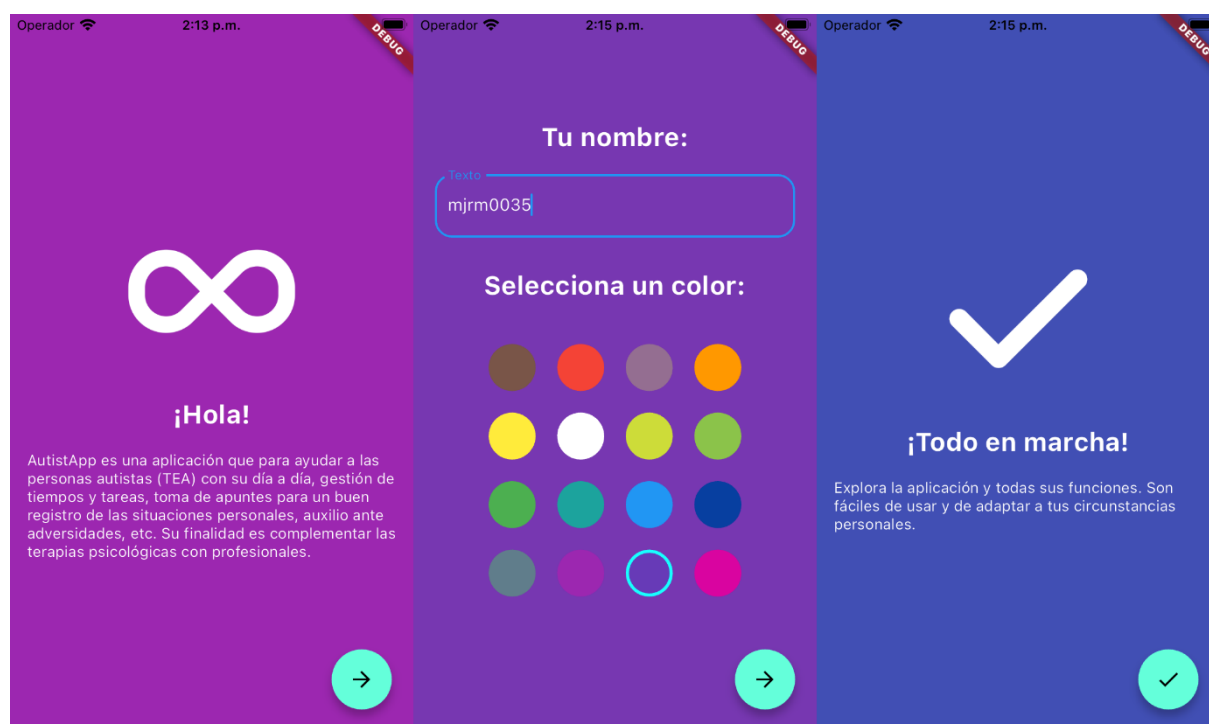


Ilustración 29 – Secuencia de la pantalla de bienvenida en una versión compilada de la aplicación para iOS.

5.2.2 Trabajo con tareas

Para eliminar tareas cuando éstas están establecidas, se pueden deslizar a la izquierda o a la derecha en la lista. Esta acción puede también realizarse desde la pantalla de edición, accesible al mantener pulsada una tarea.

En la mencionada pantalla hay controles para borrar y compartir la información de la tarea con otras aplicaciones. Asimismo, se puede manipular cualquier aspecto relativo a la propia tarea.

Para crear tareas nuevas, desde el botón flotante “∞” se accede al menú de creación de recursos, donde aparecerá la opción “Nueva tarea”.

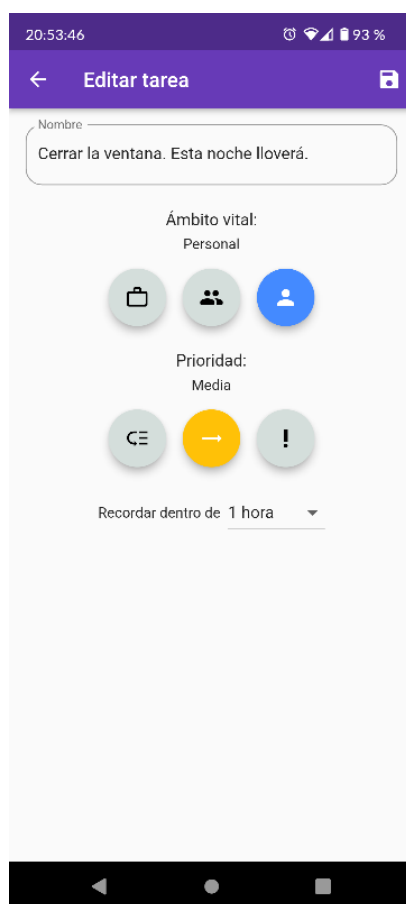


Ilustración 30 – Pantalla de creación de nuevas tareas. Nótese la ausencia de botones de borrado y compartición, no disponibles para una tarea que se está creando.

5.2.3 Trabajo con notas de texto y voz

En la pantalla de inicio, desde el botón flotante “∞” podemos acceder a acciones de creación de nuevas notas de texto y voz.

Común a ambos tipos es la introducción de nombre y el establecimiento de un estado de ánimo y ámbito vital para contextualizar la nota en cuestión.

Para las notas de texto contaremos con un cuadro de texto en el que se puede escribir cualquier cuestión que se requiera mantener por escrito. Para las notas de voz se provee de un botón de “Grabar”, que iniciará una grabación de voz.

Para editar una nota hay dos vías:

- Desde el menú lateral, acceder respectivamente a “Notas de voz” o “Notas de texto”. Al pulsar sobre una nota podremos editarla, borrarla o compartirla.
- Desde la vista de calendario (apartado 5.2.4), en la vista de un día completo podremos ver todas las notas de texto y voz tomadas este día. Al pulsar sobre una de ellas podremos acceder a su edición, borrado o compartición.

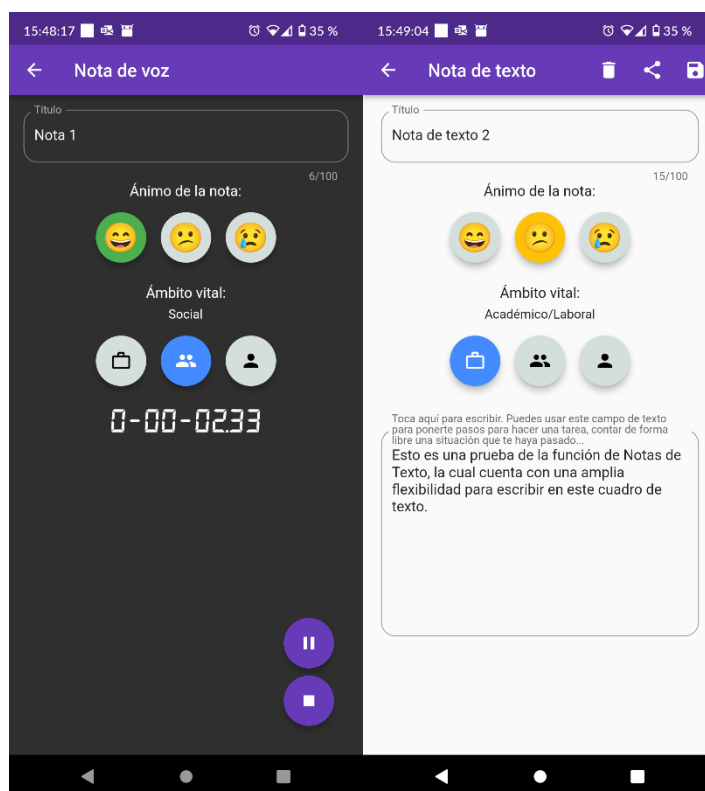


Ilustración 31 – Ejemplos de toma y edición de notas de texto y voz en la versión de Android.

5.2.4 Resumen del día

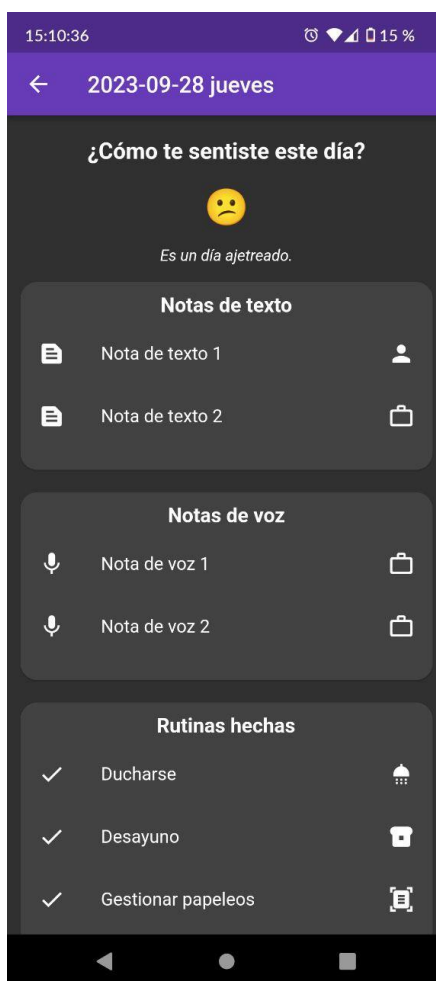


Ilustración 32 – Pantalla de resumen del día en la versión de Android.

Para acceder a la vista de calendario y al resumen de los días, se realizará desde el menú lateral existente en la pantalla de inicio.

En la vista de calendario se podrá navegar mes a mes para buscar días. Al acceder a un día en concreto podremos ver los siguientes elementos:

- Resumen del día y estado de ánimo (elementos definidos en la pantalla de inicio).
- Lista de notas de texto y voz tomadas en el día seleccionado.
- Lista de rutinas que el usuario ha logrado realizar.

5.2.5 Estado de ánimo diario



Ilustración 33 – Pantalla de inicio de la versión de Android; en la que se incluyen los botones que representan el estado de ánimo del usuario, el cual puede interactuar con ellos.

En la pantalla de inicio, en la parte superior aparece una fila de botones con iconos de caras que representan emociones.

Al pulsar en una cara, se abrirá un cuadro de texto en el que podremos escribir el resumen del día, para guardarlo una vez que finalicemos.

Este texto es editable a lo largo de todo el día, y a priori no podrá ser editado a la finalización de éste.

El texto se guardará en el resumen del día, accesible desde la vista de calendario (apartado 5.2.4).

5.2.6 Tablas de planes

Para acceder a la función de tablas de planes, hay que abrir el menú lateral en la pantalla de inicio, estando allí la opción.

Para crear tablas de planes nuevas, se pulsará sobre el botón +. A partir de ahí, se podrán añadir con el botón + sucesivos planes, que tendrán definido nombre, ámbito vital (parámetro que definirá el color), horas de inicio y fin. Asimismo, podrá editarse el nombre de la tabla.

Para ver la tabla creada tras haberla definido, hay que volver a la lista de tablas de planes y pulsar sobre la que hemos realizado ahora mismo. Desde la vista de diagrama de Gantt podremos acceder a opciones para editar, borrar o compartir la información de la tabla.

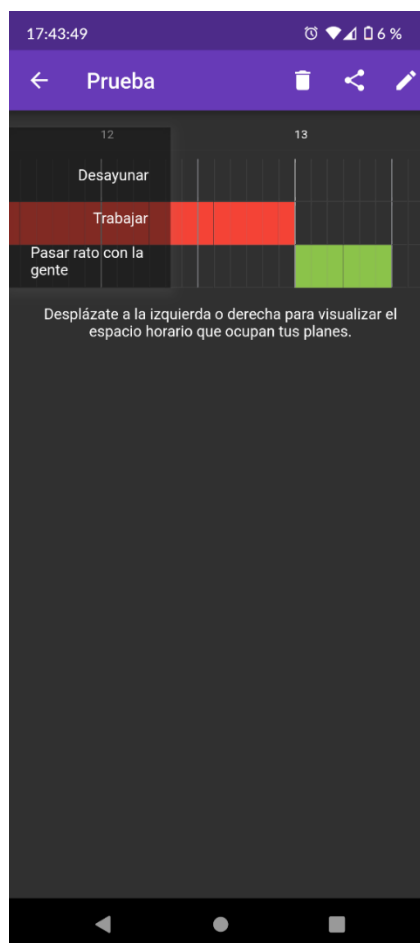


Ilustración 34 – Detalle de la vista de tablas de planes con una planificación de ejemplo, en la versión de Android.

5.2.7 Ruleta de decisiones

Para acceder a la función de tablas de planes, hay que abrir el menú lateral en la pantalla de inicio, estando allí la opción.

Para crear tablas de planes nuevas, se pulsará sobre el botón +. A partir de ahí, se podrán añadir con el botón + sucesivos planes, que tendrán definido nombre, ámbito vital (parámetro que definirá el color), horas de inicio y fin. Asimismo, podrá editarse el nombre de la tabla.

Para ver la tabla creada tras haberla definido, hay que volver a la lista de tablas de planes y pulsar sobre la que hemos realizado ahora mismo. Desde la vista de diagrama de Gantt podremos acceder a opciones para editar, borrar o compartir la información de la tabla.



Ilustración 35 – Ejemplo de una ruleta de decisiones en la versión de Android, tanto en tema claro como en tema oscuro.

5.2.8 Pantalla de ajustes

Desde la pantalla de inicio es accesible la vista de ajustes. Para ello, pulsaremos en el icono en la parte superior derecha de la aplicación.

En esta vista tendremos la facultad de cambiar el nombre con el que la aplicación nos recibe en la pantalla de inicio, ajustar los límites de la tabla de planificaciones, la frecuencia de las notificaciones y el color de la interfaz.

Si pulsamos Atrás no se guardarán los cambios. Esto solamente es factible al pulsar el botón Guardar.

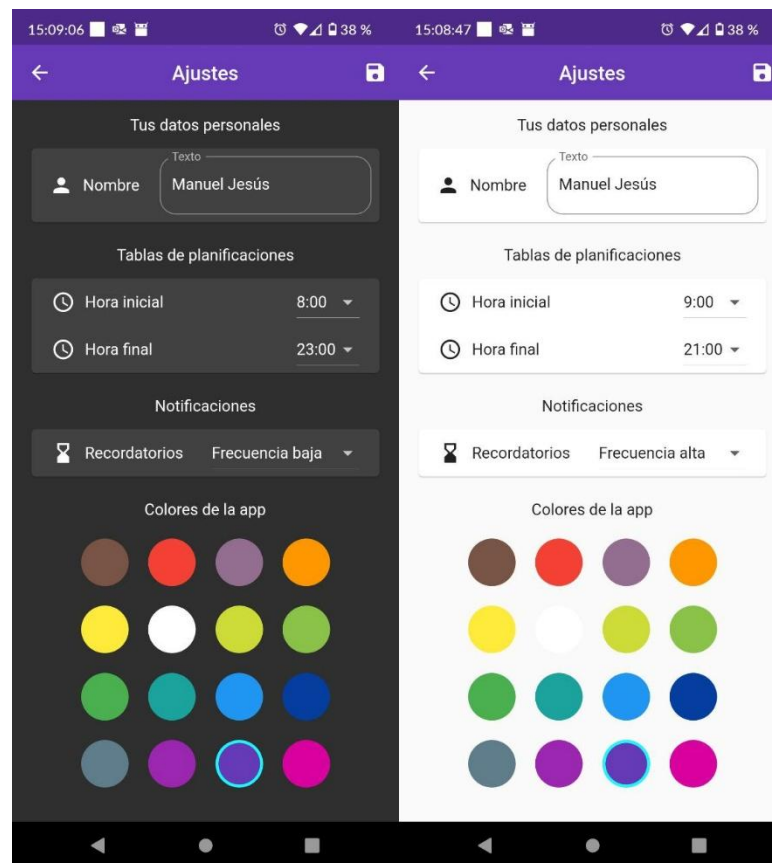


Ilustración 36 – Detalle de la pantalla de Ajustes en la versión de Android, tanto en modo claro como en modo oscuro. La versión de iOS no cuenta con un ajuste de frecuencia de notificaciones debido a las limitaciones con las mismas (véase apartado 3.7).

6 DEFINICIONES Y ABREVIATURAS

- APA: *American Psychological Association* (Asociación Americana de Psicología).
- API: *Application Programming Interface* (Interfaz de programación de aplicaciones).
- APK: *Android PackAge* (paquete de Android).
- CIE: Clasificación internacional de Enfermedades de la OMS (en inglés, *International Statistical Classification of Diseases and Related Health Problems* (ICD)).
- DSM-5: Manual de psicología publicado por la APA entre 2013 y 2014, que incorpora como tal el diagnóstico de TEA.
- Estereotipia: comportamiento motor repetitivo fruto de una alteración neurológica, observado en diversas condiciones neurológicas como el TEA (J, 2001).
- RF: Requisito funcional.
- RNF: Requisito no funcional.
- SDK: *Software Development Kit* (Kit de desarrollo de software).
- TEA: Trastorno del Espectro Autista.
- UUID: *universally unique Identifier* (Identificador único universal). (Davis, 2019)

7 BIBLIOGRAFÍA

- ¿Cuánto gana un ingeniero informático? Descubre su sueldo.* (Abril de 2023). Obtenido de Universidad Alfonso X El Sabio: <https://www.uax.com/blog/ingenieria/cuanto-cobra-un-ingeniero-informatico>
- American Psychological Association. (2014). *Guía de consulta de los criterios diagnósticos del DSM-V.*
- dart.dev. (s.f.). *Sound null safety.* Obtenido de Sound null safety: <https://dart.dev/null-safety>
- Data & Statistics on Autism Spectrum Disorder.* (04 de 04 de 2023). Obtenido de Centers for Disease Control and Prevention: <https://www.cdc.gov/ncbddd/autism/data.html>
- Davis, H. (19 de 12 de 2019). *6 Best Autism Apps For Skill Development and Confidence.* Obtenido de Carmen B. Pringlee: <https://carmenbpringlee.com/blog/9-best-autism-apps-for-skill-development-and-confidence/>
- Definición de casos de uso.* (09 de 03 de 2021). Obtenido de IBM: <https://www.ibm.com/docs/es/engineering-lifecycle-management-suite/lifecycle-management/6.0.3?topic=requirements-defining-use-cases>
- dexterv.dev. (08 de 2023). *flutter_local_notifications.* Obtenido de pub.dev: https://pub.dev/packages/flutter_local_notifications
- Dillet, R. (12 de 09 de 2017). *iOS 11 is coming out on September 19.* Obtenido de TechCrunch: <https://techcrunch.com/2017/09/12/ios-11-release-date/?guccounter=2>
- Dubost, K. (16 de 08 de 2003). *Use el formato de fecha internacional.* Obtenido de Use el formato de fecha internacional: <https://www.w3.org/QA/Tips/iso-date.html.es>
- Flutter - Get started.* (2023). Obtenido de flutter.dev: <https://docs.flutter.dev/get-started>
- Global Smartphone Market Share: By Quarter.* (17 de 08 de 2023). Obtenido de Counterpoint: <https://www.counterpointresearch.com/insights/global-smartphone-share/>
- Huambachano, J. F. (25 de 09 de 2017). *¿Qué es Scrum?* Obtenido de Scrum, the home of Scrum: <https://www.scrum.org/resources/blog/que-es-scrum>
- J, J. (2001). *Differential diagnosis and etiology of tics.* Obtenido de Differential diagnosis and etiology of tics.
- Kelion, L. (03 de 09 de 2013). *Android KitKat unveiled in Google surprise move.* Obtenido de BBC News - Technology: <https://www.bbc.co.uk/news/technology-23926938>
- kevlatus.de. (07 de 2023). *flutter_fortune_wheel.* Obtenido de pub.dev: https://pub.dev/packages/flutter_fortune_wheel

- lifbandit. (2023). *record*. Obtenido de Flutter packages | pub.dev: <https://pub.dev/packages/record>
- MayoClinic. (29 de 07 de 2021). *Terapia Cognitivo Conductual*. Obtenido de MayoClinic: <https://www.mayoclinic.org/es/tests-procedures/cognitive-behavioral-therapy/about/pac-20384610>
- Mobile Operating System Market Share Worldwide*. (31 de 08 de 2023). Obtenido de StatCounter: <https://gs.statcounter.com/os-market-share/mobile/worldwide>
- Patel, N. (13 de 10 de 2013). *The iPhone 5s: fingerprint sensor and improved camera, starts at \$199 and coming September 20th*. Obtenido de The Verge: <https://www.theverge.com/2013/9/10/4713720/apple-iphone-5s-release-date-price-cost>
- Pedro BM, Pilowsky LS, Costa DC, Hemsley DR, Ell PJ, Verhoeff NP, Kerwin RW, Gray NS. Stereotypy, schizophrenia and dopamine D2 receptor binding in the basal ganglia. *Psychol Med*. 1994 May;24(2):423-9. doi: 10.1017/s0033291700027392. PMID: 8084937.
- Pressman, R. (2010). *Ingeniería del Software*. McGraw-Hill.
- Ramírez, I. (5 de 12 de 2018). *Flutter, el framework de Google para crear apps multiplataforma, llega a su versión 1.0*. Obtenido de Xataka Android: <https://www.xatakandroid.com/aplicaciones-android/flutter-framework-google-para-crear-apps-multiplataforma-llega-a-version-1-0>
- Riveros, A. (17 de 11 de 2020). *Qué es el alcance de un proyecto y cómo gestionarlo*. Obtenido de EALDE Business School: <https://www.ealde.es/alcance-proyecto-gestion/>
- Rombiola, N. (14 de 09 de 2014). *¿Qué es el UUID?* Obtenido de Los impuestos: <https://web.archive.org/web/20200522141455/https://losimpuestos.com.mx/que-es-el-uuid/>
- Srivastava, S. (15 de 09 de 2023). *Top 10 Best Cross-Platform App Development Frameworks*. Obtenido de AppInventiv: <https://appinventiv.com/blog/cross-platform-app-frameworks/>
- What is Autism Spectrum Disorder?* (2022). Obtenido de What is Autism Spectrum Disorder?: <https://www.cdc.gov/ncbddd/autism/facts.html>
- Wozniak, A. (2023). *table_calendar*. Obtenido de pub.dev: https://pub.dev/packages/table_calendar