

UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR



TRABAJO FIN DE MÁSTER

Tic-Tac-TEA: Herramienta de autor para autorregulación emocional en situaciones de crisis

**Doble máster en Ingeniería informática y en
Investigación e innovación en TICs**

**AUTOR: Chinchay Manco, Yussy Miriam
TUTOR: Montoro Manrique, Germán**

Septiembre 2016

Resumen

Dentro del espectro de dificultades a las que se enfrentan las personas con autismo, se encuentra la disregulación emocional, es decir, la incapacidad de tener una adecuada respuesta emocional. Esto influye en el comportamiento de las personas, provocando sentimientos negativos y obstaculizando el proceso de comunicación. Si un individuo no es capaz de percibir o transmitir adecuadamente sus emociones, verá aumentada sus problemas conductuales y de adaptación social.

Si bien es cierto que existen terapias para prevenir situaciones de crisis, éstas suelen requerir la asistencia de terapeutas profesionales, y la familia adquiere un papel secundario en la enseñanza de esta regulación.

En la actualidad, gracias a las aplicaciones móviles, podemos cambiar la forma de aplicar las terapias. Ahora, los familiares pueden encontrar actividades educativas y pueden usarlas en cualquier lugar. Sin embargo, existen pocas aplicaciones destinadas a la regulación emocional y, la gran mayoría de ellas, no detectan situaciones de crisis, sino que deben ser ejecutadas manualmente cuando se necesiten.

Así pues, por todas las razones anteriormente mencionadas, se propone desarrollar una herramienta de autor para dispositivos móviles dirigida a la autorregulación emocional de personas con TEA. La aplicación, denominada *Tic-Tac-TEA*, tendrá una parte wear (smartwatch) que detectará situaciones de crisis, y una parte móvil (smartphone) que permitirá la personalización de los ejercicios de regulación emocional mostrados en el reloj inteligente. Esta personalización de estrategias es fundamental ya que las personas con autismo suelen tener intereses excesivamente enfocados.

Este proyecto ha sido diseñado en colaboración con el Instituto de Psico-Pediatría Doctor Quintero Lumbreras, y ha sido seleccionado por la Fundación Orange en su convocatoria Soluciones tecnológicas para autismo 2015, de entre más de 150 propuestas.

Palabras clave

TEA, autismo, autorregulación emocional, control emocional, terapia conductual, aplicación móvil, aplicación educativa, disregulación emocional, competencia emocional, intervención conductual, herramienta de autor, tecnología asistida

Abstract

Among the range of difficulties that people with autism face, we can find the emotional dysregulation, which means the inability to have a proper emotional response in some given situations. This influences on the people's behavior, causing negative feelings and hindering the communication process. If people is unable to perceive or properly convey their emotions, they will see increased their behavioral and social adjustment problems.

While it is true that there are therapies to prevent crisis situations, they often require the assistance of specialized therapists. This renders the patients' family unable to assist with the treatment.

Currently, thanks to mobile applications, we can change the way in which therapies are applied. Families can now easily find educational material and engage in educational activities by themselves in a time and place that fits them. However, there are few applications for emotional regulation and most of them do not detect situations of crisis, they must be performed manually when needed.

For all the above reasons, we proposed to develop an authoring tool for mobile devices aimed at emotional self-regulation of people with ASD. The application, called Tic-Tac-TEA, have a wear module (smartwatch) to detect situations of crisis, and a mobile module (smartphone) that allow customization of emotional regulation exercises, that will be shown on the smartwatch. This customization of strategies is fundamental since people with autism tend to have overly focused interests.

This project has been designed in collaboration with the Institute of Psycho-Pediatrics Doctor Quintero Lumbreras, and it has been selected by the Orange Foundation in its Autism Tech Solutions call (2015), out of more than 150 proposals.

Keywords

ASD, autism, emotional self-regulation, emotional control, behavioral therapy, mobile application, educational application, emotional dysregulation, emotional competence, behavioral intervention, authoring tool, assistive technology

Índice de Contenidos

RESUMEN.....	I
PALABRAS CLAVE	I
ABSTRACT	II
KEYWORDS.....	II
ÍNDICE DE CONTENIDOS	III
ÍNDICE DE TABLAS.....	VII
ÍNDICE DE FIGURAS	VIII
GLOSARIO	X
AGRADECIMIENTOS.....	XI
1. INTRODUCCIÓN	1
1.1. TRASTORNO DEL ESPECTRO AUTISTA	1
1.1.1. <i>Problemas de conducta y disregulación emocional</i>	1
1.2. MOTIVACIÓN	2
1.3. OBJETIVOS.....	3
1.4. ESTRUCTURA DE LA MEMORIA	4
2. TECNOLOGÍAS A UTILIZAR Y ESTADO DEL ARTE	5
2.1. DISPOSITIVOS MÓVILES	5
2.1.1. <i>Mobile Learning</i>	5
2.1.2. <i>Wearables y Android Wear</i>	6
2.2. ESTADO DEL ARTE.....	8
2.2.1. <i>Autorregulación emocional en el TEA</i>	8
2.2.1.1. Estudio de mapeo sistemático.....	8
2.2.1.1.1. Preguntas de investigación	9
2.2.1.1.2. Cadenas de búsqueda.....	9
2.2.1.1.3. Criterios de inclusión y exclusión.....	10
2.2.1.1.4. Resultados.....	11
2.2.2. <i>Aplicaciones móviles relacionadas</i>	12
2.3. CONCLUSIONES DEL ESTADO DEL ARTE	14
3. ANÁLISIS.....	16
3.1. RESUMEN DE LOS REQUISITOS DEL SISTEMA	17
3.1.1. <i>Requisitos funcionales</i>	18
3.1.2. <i>Requisitos no funcionales</i>	21
3.1.2.1. De recursos	21
3.1.2.2. De usabilidad	21
3.1.2.3. De operatividad	22
3.1.2.4. De escalabilidad	22
3.1.2.5. De robustez.....	22
4. DISEÑO	23

4.1.	DIAGRAMA DE CLASES	23
4.2.	ANÁLISIS DEL DISEÑO DE LA APLICACIÓN	28
4.3.	MODELO DE DATOS	32
4.4.	DISEÑO DE ESTRATEGIAS.....	34
5.	DESARROLLO.....	36
5.1.	PROTOCOLO DE COMUNICACIÓN	36
5.2.	ENTORNO DE TRABAJO	38
5.3.	APARTADOS DE LA APLICACIÓN	39
5.3.1.	<i>Pantalla inicial Splash</i>	40
5.3.1.1.	Descripción	40
5.3.1.2.	Interfaz.....	40
5.3.1.3.	Lógica	41
5.3.2.	<i>Pantalla PIN de seguridad</i>	41
5.3.2.1.	Descripción	41
5.3.2.2.	Interfaz.....	41
5.3.2.3.	Lógica	42
5.3.3.	<i>Pantalla Home y menú lateral</i>	42
5.3.3.1.	Descripción	42
5.3.3.2.	Interfaz.....	42
5.3.3.3.	Lógica	44
5.3.4.	<i>Pantalla de Perfil de usuario</i>	45
5.3.4.1.	Descripción	45
5.3.4.2.	Interfaz.....	45
5.3.4.3.	Lógica	46
5.3.5.	<i>Pantalla de Editar perfil</i>	46
5.3.5.1.	Descripción	46
5.3.5.2.	Interfaz.....	46
5.3.5.3.	Lógica	47
5.3.6.	<i>Pantalla de Configuración</i>	49
5.3.6.1.	Descripción	49
5.3.6.2.	Interfaz.....	49
5.3.6.3.	Lógica	51
5.3.7.	<i>Pantalla de Estrategias</i>	51
5.3.7.1.	Descripción	51
5.3.7.2.	Interfaz.....	51
5.3.7.3.	Lógica	53
5.3.8.	<i>Pantalla de Nueva estrategia</i>	54
5.3.8.1.	Descripción	54
5.3.8.2.	Interfaz.....	54
5.3.8.3.	Lógica	55
5.3.9.	<i>Pantalla de Nueva estrategia. Tarjeta de texto</i>	56
5.3.9.1.	Descripción	56
5.3.9.2.	Interfaz.....	56
5.3.9.3.	Lógica	57
5.3.10.	<i>Pantalla de Nueva estrategia. Tarjeta de pictogramas</i>	59
5.3.10.1.	Descripción	59

5.3.10.2. Interfaz.....	59
5.3.10.3. Lógica	60
5.3.11. <i>Pantalla de Nueva estrategia. Tarjeta de recursos propios</i>	60
5.3.11.1. Descripción	60
5.3.11.2. Interfaz.....	60
5.3.11.3. Lógica	61
5.3.12. <i>Pantalla de Nueva Estrategia. Configuración</i>	62
5.3.12.1. Descripción	62
5.3.12.2. Interfaz.....	62
5.3.12.3. Lógica	63
5.3.13. <i>Pantalla de Regulaciones</i>	63
5.3.13.1. Descripción	63
5.3.13.2. Interfaz.....	63
5.3.13.3. Lógica	65
5.3.14. <i>Pantalla de Nueva Regulación</i>	66
5.3.14.1. Descripción	66
5.3.14.2. Interfaz.....	66
5.3.14.3. Lógica	67
5.3.15. <i>Pantalla de Tutorial</i>	68
5.3.15.1. Descripción	68
5.3.15.2. Interfaz.....	68
5.3.15.3. Lógica	69
6. PRUEBAS	70
6.1. VALIDACIÓN DE REQUISITOS	70
6.1.1. <i>Nueva estrategia con tarjeta de texto</i>	70
6.1.2. <i>Nueva estrategia con tarjeta con recurso propio</i>	71
6.1.3. <i>Nueva regulación con edición de estrategias</i>	71
6.1.4. <i>Cambio de idioma de aplicación</i>	71
6.2. PRUEBAS DE INTEGRACIÓN	71
6.3. EVALUACIÓN HEURÍSTICA	72
6.4. ENCUESTA SUS	72
7. RESULTADOS	73
7.1. VALIDACIÓN DE REQUISITOS	73
7.1.1. <i>Nueva estrategia con tarjeta de texto</i>	73
7.1.2. <i>Nueva estrategia con tarjeta con recurso propio</i>	76
7.1.3. <i>Nueva regulación con edición de estrategias</i>	79
7.1.4. <i>Cambio de idioma de aplicación</i>	82
7.2. PRUEBAS DE INTEGRACIÓN	82
7.3. EVALUACIÓN HEURÍSTICA	83
7.3.1. <i>Visibilidad del estado del sistema</i>	83
7.3.2. <i>Lenguaje de los usuarios</i>	84
7.3.3. <i>Control y libertad para el usuario</i>	84
7.3.4. <i>Consistencia y estándares</i>	84
7.3.5. <i>Ayuda a los usuarios a reconocer, diagnosticar y recuperarse de los errores</i> 85	
7.3.6. <i>Prevención de errores</i>	85

7.3.7. Reconocimiento antes que cancelación	86
7.3.8. Flexibilidad y eficiencia de uso.....	86
7.3.9. Estética de diálogos y diseño minimalista	87
7.3.10. Ayuda general y documentación	87
7.4. ENCUESTA SUS	88
8. CONCLUSIONES Y TRABAJO FUTURO	89
9. REFERENCIAS	90
10. ANEXOS	92
ANEXO A. EVOLUCIÓN DEL DISEÑO DE TIC-TAC-TEA.....	92
ANEXO B. PROTOCOLO DE COMUNICACIÓN ENTRE SMARTPHONE Y SMARTWATCH	97
ANEXO C. ENCUESTA SUS	99

Índice de Tablas

Tabla 1 - Comparativa de tecnologías de regulación emocional	15
Tabla 2. Valoración de Visibilidad del estado del sistema	83
Tabla 3. Valoración de Lenguaje de los usuarios	84
Tabla 4. Valoración de Control y libertad para el usuario.....	84
Tabla 5. Valoración de Consistencia y estándares	85
Tabla 6. Valoración de Ayuda a los usuarios reconocimiento, diagnóstico y recuperación de errores	85
Tabla 7. Valoración de Prevención de errores	86
Tabla 8. Valoración de Reconocimiento antes que cancelación	86
Tabla 9. Valoración de Flexibilidad y eficiencia de uso	87
Tabla 10. Valoración de Estética de diálogos y diseño minimalista.....	87
Tabla 11. Valoración de Ayuda general y documentación.....	87
Tabla 12. Resultado de encuesta SUS	88

Índice de Figuras

Figura 1. Comparativa del número de usuarios de dispositivos móviles y de ordenador	5
Figura 2. Comparativa de aceptación del uso de diferentes wearables.....	6
Figura 3. Scopus. Análisis de resultados obtenidos.....	11
Figura 4. Manos quietas. Video en reproducción.....	12
Figura 5. Calm counter. Evolución del enfado con pictogramas.....	13
Figura 6. Adiós enfados. Consejos del hada	13
Figura 7. Respira, piensa, actúa. Pantalla de interacción	14
Figura 8. Diagrama de clases Tic-Tac-TEA. Parte I.....	23
Figura 9. Diagrama de clases Tic-Tac-TEA. Parte II.....	24
Figura 10. Diagrama de clases Tic-Tac-TEA. Parte III	25
Figura 11. Diagrama de clases Tic-Tac-TEA. Parte IV	26
Figura 12. Diagrama de clases Tic-Tac-TEA. Parte V - Modelo.....	27
Figura 13. Diagrama de E-R de Tic-Tac-TEA	32
Figura 14. Pictogramas de ARASAAC. Pintar y besar	34
Figura 15. Ejemplo de estrategias con burbujas.....	35
Figura 16. Estructura de las carpetas y ficheros del proyecto	39
Figura 17. Tic-Tac-TEA. Pantalla Splash.....	40
Figura 18. Tic-Tac-TEA. Pantalla PIN de seguridad.....	41
Figura 19. Tic-Tac-TEA. Pantalla Home y menú lateral	44
Figura 20. Tic-Tac-TEA. Pantalla de Perfil.....	45
Figura 21. Tic-Tac-TEA. Pantalla de Editar perfil	47
Figura 22. Tic-Tac-TEA. Diálogo con opciones de edición de foto.....	48
Figura 23. Tic-Tac-TEA. Pantalla de Configuración.....	50
Figura 24. Tic-Tac-TEA. Pantalla de Estrategias.....	53
Figura 25. Tic-Tac-TEA. Pantalla de Nueva estrategia	55
Figura 26. Tic-Tac-TEA. Pantalla de Nueva estrategia. Tarjeta de texto.....	57
Figura 27. Tic-Tac-TEA. Pantalla de Nueva estrategia. Tarjeta con pictograma.....	59
Figura 28. Tic-Tac-TEA. Pantalla de Nueva estrategia. Tarjeta con recurso propio	61
Figura 29. Tic-Tac-TEA. Pantalla de Nueva estrategia. Configuración	62
Figura 30. Tic-Tac-TEA. Pantalla de Regulaciones.....	65
Figura 31. Tic-Tac-TEA. Pantalla de Nueva regulación	67
Figura 32. Tic-Tac-TEA. Pantalla de Tutorial.....	68
Figura 33. Prueba. Nueva estrategia con tarjeta de texto. Parte I	73
Figura 34. Prueba. Nueva estrategia con tarjeta de texto. Parte II	74
Figura 35. Prueba. Nueva estrategia con tarjeta de texto. Parte III	75
Figura 36. Prueba. Nueva estrategia con tarjeta de texto. Parte IV	76
Figura 37. Prueba. Nueva estrategia con tarjeta de recurso. Parte I.....	76
Figura 38. Prueba. Nueva estrategia con tarjeta de recurso. Parte II.....	77
Figura 39. Prueba. Nueva estrategia con tarjeta de recurso. Parte III.....	78
Figura 40. Prueba. Nueva estrategia con tarjeta de recurso. Parte IV	79
Figura 41. Prueba. Nueva regulación con edición de estrategias. Parte I	79
Figura 42. Prueba. Nueva regulación con edición de estrategias. Parte II	80
Figura 43. Prueba. Nueva regulación con edición de estrategias. Parte III	81
Figura 44. Prueba. Cambio de idioma de aplicación	82

Figura 45. Prueba. Adaptación de pictograma a pantalla del smartwatch	82
Figura 46. Prueba. Adaptación de regulación con selector a pantalla del smartwatch .	83
Figura 47. Evolución del diseño de Tic-Tac-TEA. Versión 1.0. Parte I	92
Figura 48. Evolución del diseño de Tic-Tac-TEA. Versión 1.0. Parte II	93
Figura 49. Evolución del diseño de Tic-Tac-TEA. Versión 1.0. Parte III	94
Figura 50. Evolución del diseño de Tic-Tac-TEA. Versión 2.0. Parte I	94
Figura 51. Evolución del diseño de Tic-Tac-TEA. Versión 2.0. Parte II	95
Figura 52. Evolución del diseño de Tic-Tac-TEA. Versión 2.0. Parte III	96
Figura 53. Encuesta SUS.....	99

Glosario

Android wear: Nuevo sistema operativo centrado en los dispositivos wearables actuales y por venir.

Disregulación emocional: Término profesional para referirse a la falta de gestión adecuada de respuestas emocionales, que se muestran inflexibles y restringidas.

DSM-5: Quinta edición del Libro de Asociación Estadounidense de Manual Diagnóstico y Estadístico de Trastornos Mentales (*Diagnostic and Statistic Manual of Mental Disorders*). Se trata de una clasificación de trastornos mentales que sirve de referencia a los profesionales de la salud.

Estrategias: Conjunto de tarjetas o pasos cuyo objetivo es conseguir la autorregulación emocional del usuario, se trata de estrategias de relajación.

Herramienta de autor: Aplicación informática que permite crear, publicar y gestionar otras herramientas con contenido digital y soporte TIC.

m-learning: Sistema de aprendizaje cuya metodología se basa en el uso de tecnologías móviles.

Regulaciones: Las regulaciones o regulaciones emocionales son un conjunto de estrategias que pueden ser sincronizadas con el smartwatch para su ejecución en el momento en que se detecte una situación de crisis.

Tarjetas o pasos: Unidad en la que se dividen las estrategias. Contienen un elemento gráfico que puede ser una imagen estática, un gif o un texto. Son completamente personalizables por los usuarios.

TEA: Trastorno del Espectro Autista.

Tic – Tac – TEA: Herramienta de autor que integra estrategias y regulaciones emocionales para la autorregulación emocional de personas con TEA, que pueden ser sincronizadas con un smartwatch.

Wearable: Dispositivo tecnológico que puede ser llevado puesto como si de un complemento se tratara por los usuarios.

Agradecimientos

En primer lugar, debo agradecer a la Universidad Autónoma de Madrid y, en particular, a la Escuela Politécnica Superior, por contar conmigo para este tipo de proyectos innovadores. Además de utilizar las últimas tecnologías, puedo aplicarlas en un campo que era al que quería aportar algo cuando empecé la carrera. Gracias por permitirme colaborar en áreas centradas en personas con necesidades especiales y gracias a mi Escuela por haberme dado los conocimientos para ello.

Gracias al Dr. Germán Montoro por supervisar mi proyecto y por haber pensado en mí para desarrollarlo. Igualmente, gracias a todo el equipo de personas que han formado parte de este proyecto y se han volcado en él. Especial agradecimiento a las profesionales del I.P.P. Quintero Lumbreras, que tanto me han asesorado y que han aportado grandes ideas. En general, ha sido increíble ver la gran cantidad de personas que me han apoyado desinteresadamente, incluyendo varios compañeros del sector. Sin ellos esto tampoco hubiera sido igual.

Finalmente, gracias a mi familia, a mi entorno más cercano que se vuelve tan distante cuando me entrego a este tipo de proyectos. Agradezco toda la comprensión y cada una de las tazas de café de mi madre. Con este proyecto cierro una gran etapa de mi vida, una vida complicada, de mucho trabajo, esfuerzo, y no puedo más que agradecer a todos los que aún siguen formando parte de ella.

1. Introducción

1.1. Trastorno del Espectro Autista

El Trastorno del Espectro Autista (TEA) es el término incluido en el Manual Diagnóstico y Estadístico de Trastornos Mentales (DSM-5) [1], que engloba diversas alteraciones cognitivas, psicológicas y conductuales.

Dentro de este amplio espectro, las características clínicas pueden variar enormemente de una persona a otra, así como sus grados de afectación. El DSM-5 clasifica el autismo según tres grados de gravedad de los síntomas y según el nivel de apoyo necesario que requiere. Además, establece dos grandes categorías de las manifestaciones clínicas: Deficiencias en la comunicación social y comportamientos restringidos que incluyen autoagresión, ecolalia y apego estricto a rutinas [2 – 3].

En la afectación social destaca la falta de habilidades de interacción, derivadas de los problemas para compartir intereses, atender a las expresiones faciales o a la falta de contacto visual. Esto genera la imposibilidad de inferir segundas intenciones, comprender miradas, deseos y conductas de otros, lo que afecta gravemente a la conducta social y a la empatía [4].

El campo de la comunicación más afectado es el pragmático, que es el que se refiere a cómo atender al uso del lenguaje para comunicar. Las habilidades de comunicación verbal y no verbal pueden variar desde personas con retraso o ausencia del lenguaje hasta personas con habla adecuada, pero con incapacidad para iniciar o mantener conversaciones al hacer un uso incoherente del lenguaje o al mostrar dificultades en la toma de turnos [5].

Además de las alteraciones mencionadas, un 75% de las personas con autismo tienen una discapacidad intelectual asociada, que también puede variar entre distintos rangos de gravedad.

Respecto a la prevalencia del TEA, la última publicación del Centro de Control de Enfermedades de EEUU (CDC), indica una prevalencia en 1 de cada 68 niños, lo que supone un incremento del 30% de casos respecto a hace un par de años. A nivel de Europa, este trastorno afecta a 1 de cada 100 nacimientos, según revela el último estudio realizado por la asociación internacional *Autism – Europe aisbl* en 2015.

1.1.1. Problemas de conducta y disregulación emocional

La incapacidad para encontrar una vía adecuada de comunicación puede generar situaciones de frustración que derivan en problemas de conducta. Asimismo, la limitación en el lenguaje no verbal dificulta la adaptación a nuevos y complejos contextos sociales.

Estos problemas de adaptación a entornos nuevos o estimulantes se ven magnificados debido a que las personas con TEA necesitan que su entorno sea lo más previsible posible. Se tiene una inflexibilidad que puede llegar a ser extrema en sus rutinas, con lo que cualquier leve cambio resulta excesivamente perturbador y puede provocar crisis emocionales. Por ello, las personas con autismo muestran conductas repetitivas, que van relacionadas con un repertorio limitado de sus intereses.

Otro déficit presente en el TEA es la disregulación emocional, es decir, la dificultad para entender y regular emociones de manera apropiada y afectiva. Como respuesta a esta limitación, se producen casos que van desde la depresión y el aislamiento, hasta la falta del control de la ira, irritabilidad y conductas autolesivas [5].

1.2. Motivación

La motivación de este TFM responde al crecimiento de la tecnología innovadora aplicada a la asistencia de personas con necesidades especiales, en este caso, de personas con espectro autista.

Con los programas y atención adecuados, las personas con TEA pueden desarrollar sus habilidades, logrando conseguir una aceptable autonomía para desenvolverse en la vida diaria. Por este motivo, el principal objetivo de los desarrollos tecnológicos realizados ha sido proporcionar independencia a los usuarios, es decir, proporcionar autodeterminación. Este comportamiento autodeterminado se compone de cuatro características necesarias: autonomía, autorregulación, fortalecimiento y realización [6].

Se propone desarrollar un proyecto que involucra varios comportamientos autodeterminados como la elección y toma de decisiones, resolución de problemas, fijación de objetivos, adquisición de habilidades y control interno.

Para llevar a la práctica este enfoque, se plantea emplear tecnologías emergentes [7] para la asistencia a los usuarios. El proyecto *Tic – Tac – TEA* empleará smartwatches junto a smartphones para la asistencia en la autorregulación emocional y contará con la colaboración del Instituto de Psico – Pediatría “Dr. Quintero Lumbreras” [8].

Además, este proyecto ha sido seleccionado por la Fundación Orange, entre más de 150 propuestas recibidas, en su convocatoria “Soluciones tecnológicas aplicadas al autismo 2015”. El fallo del jurado se basó en criterios como la necesidad de la tecnología propuesta para las personas con autismo, la explotación gratuita de los resultados finales, el carácter novedoso y la voluntad de llegar al máximo número de usuarios en toda España.

1.3. Objetivos

Las emociones son consideradas un aspecto fundamental en la toma de decisiones acertadas en un determinado contexto, que permiten la interacción social. El control de éstas, junto a la autodeterminación, son los objetivos clave que se pretenden conseguir con la herramienta a desarrollar en este TFM.

En relación a la autorregulación emocional, el diseño del proyecto debe ayudar en algunos de los siguientes objetivos [10]:

- Ajustar la reacción emocional a la propia intensidad.
- Identificar las situaciones que provoquen variaciones en la intensidad emocional y reaccionar ante ellas.
- Desarrollar estrategias de control emocional.
- Manejar el estrés en situaciones no deseadas.
- Gestionar episodios de ira.

Los fines últimos que se procuran conseguir con el desarrollo de este TFM son:

- Ofrecer una aplicación pragmática que responda a las necesidades de las personas con TEA y de su entorno.
- Brindar opciones de creación y edición de estrategias de autorregulación emocional, recomendadas por expertos y personalizables para cada individuo.
- Además de ofrecer opciones de gestión emocional, dar información y seguimiento de los resultados a los familiares o cuidadores.
- Lograr una mayor integración social derivada del crecimiento de la expresividad emocional.
- Procurar el avance individual en la calidad de vida e independencia de los usuarios.
- Ayudar a las personas con TEA a integrar las emociones en el desarrollo de sus habilidades sociales.

1.4. Estructura de la memoria

Este documento se divide en los bloques relativos a la Gestión de un Proyecto de Desarrollo Software. En particular, se estructura en los siguientes diez bloques:

1. Introducción: En este apartado trataremos los motivos que inspiraron la realización de este trabajo, indicando los problemas particulares de las personas con TEA y comentando la solución propuesta para tratar de afrontarlos. Además, se listará los objetivos buscados.
2. Tecnología a utilizar y estado del arte: Este bloque contendrá información relativa a por qué se decidió utilizar tecnologías móviles para el desarrollo de la herramienta e información del paradigma Android Wear. También, se comentarán las asistencias tecnológicas actuales utilizadas para la autorregulación emocional en el TEA.
3. Análisis: Este capítulo tratará sobre los temas discutidos en las reuniones con el I.P.P. Quintero Lumbreras Quinteras y sobre el análisis de requisitos realizado en ellas.
4. Diseño: Se centrará en la arquitectura de la aplicación. Se mostrará el diagrama de clases y el modelado de base de datos a emplear, explicando sus relaciones, funciones, etc. Asimismo, en este bloque se detallará el diseño de las estrategias de autorregulación emocional.
5. Desarrollo: En esta sección se darán detalles de la implementación del software, explicando la lógica de las soluciones desarrolladas. Se hará referencia a los archivos principales que componen la aplicación, así como a las herramientas de diseño y desarrollo empleadas.
6. Pruebas: Este capítulo incluirá los datos de las pruebas realizadas, tanto de validación de requisitos, evaluación heurística, de integración y de usabilidad.
7. Resultados: Mostrará los resultados de las pruebas realizadas. Además, se evaluará el alcance de los objetivos buscados.
8. Conclusiones y trabajo futuro: Tratará de la evaluación global de este TFM, incluyendo ideas evolutivas a futuro.
9. Referencias
10. Anexos

2. Tecnologías a utilizar y estado del arte

2.1. Dispositivos móviles

2.1.1. Mobile Learning

El Mobile Learning (aprendizaje móvil o *m-learning*) es la evolución del *e-learning* adaptada a dispositivos móviles, cuyo fin sigue siendo aprovechar al máximo la tecnología a servicio de los procesos didácticos [11].

Esta nueva concepción metodológica ha tenido una gran aceptación debido al gran crecimiento de usuarios de dispositivos móviles (ver Figura 1), que ya supera al de usuarios de ordenador, y debido a las ventajas pedagógicas que estos dispositivos ofrecen, tales como:

- Capacidad de elección del entorno y momento de aprendizaje.
- Posibilidad de personalización de los métodos educativos.
- Control y supervisión de las actividades realizadas por parte de los padres o tutores.

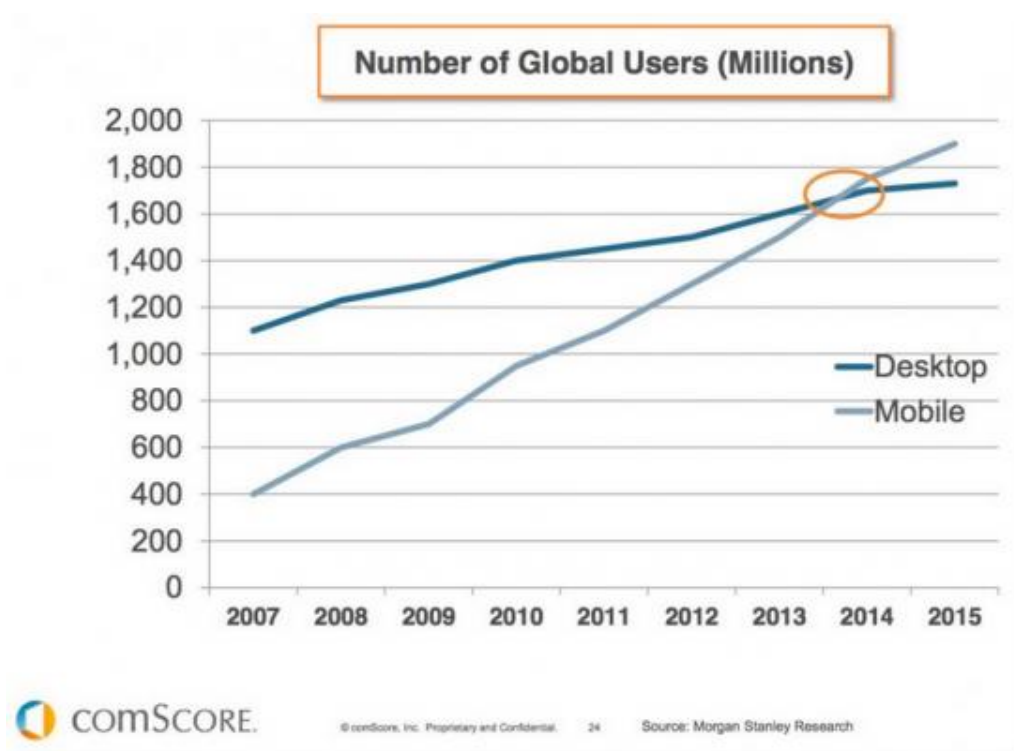


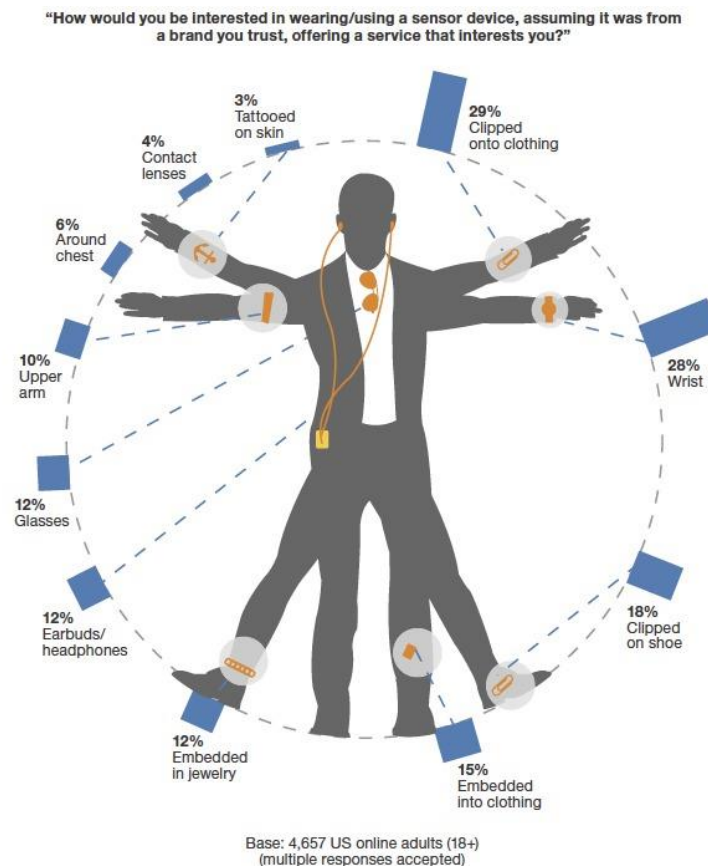
Figura 1. Comparativa del número de usuarios de dispositivos móviles y de ordenador

A estas ventajas hay que sumarle la motivación del aprendizaje derivada del modo del pensamiento digital: es sistemático, visual, de respuesta inmediata, con mínima planificación motora, presenta réplicas y ensayos cognitivos de situaciones reales, etc. Según el doctor Miguel García Coto, médico psiquiatra infantil, magister en Psicología Cognitiva y profesor de Psicopatología Infantil, todas estas características permiten integrar los recursos de las tecnologías de la información a las intervenciones realizadas a personas con TEA, y sirven como complemento a las realizadas en la vida diaria.

Cabe destacar que esta revolución tecnológica está en constante evolución. Las tecnologías disruptivas han ido desde la aparición de smartphones, tablets, phablets hasta la llegada de los actuales wearables.

2.1.2. Wearables y Android Wear

Se define como *wearable* a aquel dispositivo inteligente que se puede llevar puesto, como si de un complemento se tratase. Como vemos en la Figura 2, existe una gran variedad de estos dispositivos como son los relojes inteligentes o smartwatches, las pulseras de actividad, las smartglasses, los dispositivos embebidos en la ropa o los tatuados en la piel, entre otros.



Source: North American Technographics® Consumer Technology Survey, 2013

Figura 2. Comparativa de aceptación del uso de diferentes wearables

La portabilidad de esta tecnología junto a su comodidad de uso crea el ecosistema idóneo para el Internet de las cosas (IoT): Nos mantenemos conectados a internet, prácticamente 24/7, mientras creamos datos, generamos información de valor y ofrecemos la capacidad de tomar acciones respecto a esta información [12]. Según el objetivo final del uso que hacen los *wearables* de los datos obtenidos, estos se pueden dividir en cuatro categorías:

- Deporte y fitness
- Salud y bienestar
- Entretenimiento
- Industrial y militar

Dentro de estas categorías destaca el uso de los smartwatches, que es uno de los dispositivos que mayor aceptación tiene por parte de los usuarios. Los smartwatches disponen de sensores tales como giroscopio, acelerómetro, pulsómetro, geolocalizador, altímetro, etc., y de otras características como micrófono y opciones de conexión a internet.

Estos wearables pueden ser usados de forma independiente, con aplicaciones diseñadas específicamente para ellos. Sin embargo, es frecuente hacer un uso de estos dispositivos como una extensión del smartphone, permitiendo enviar y recibir datos entre aplicaciones e interactuar con ellas de una manera menos intrusiva que como se haría con un smartphone.

En este proyecto vamos a aprovechar el potencial de esta combinación de smartphone con smartwatch para crear una herramienta de asistencia para la autorregulación emocional de personas con TEA en situaciones de crisis.

La aplicación, denominada *Tic – Tac – TEA*, contará, por tanto, con una parte relativa al wearable y con otra relativa al smartphone. Ambas partes se desarrollarán de forma paralela e independiente, como dos proyectos diferentes que tendrán que integrarse.

La parte del smartwatch se encargará de detectar potenciales momentos de crisis a partir de las señales fisiológicas del usuario, capturadas por los sensores. En dichos casos, desplegará una ayuda en la pantalla para asistir al usuario y recuperar el estado de calma. Esta ayuda consistirá en la adaptación de ejercicios y estrategias de intervención a la pantalla del smartwatch, y dependerán del usuario y de la intensidad de la crisis.

Como complemento a la parte del wearable, será necesaria una aplicación móvil que permita la creación de estos ejercicios personalizados y que sirva como herramienta de control y seguimiento por parte de la persona responsable del usuario. Esta herramienta de autor, en la que se centra este proyecto, permitirá, por un lado, configurar de forma sencilla los parámetros específicos que se han de determinar para la medición y comportamiento de la aplicación. Esto permitirá personalizar las estrategias de intervención para su adaptación a los diferentes usuarios a los que vayan dirigidas.

Por otro lado, su función será la de seguimiento de los datos de las actividades realizadas. De este modo, la aplicación no sólo sirve como un recurso de gestión emocional para personas con TEA, sino también como una herramienta de información y seguimiento para sus familiares o cuidadores.

Una vez decidida la tecnología a utilizar, queda por decidir la elección del sistema operativo. Analizando el mercado actual, se ha optado por el uso de Android Wear. Android Wear es el sistema operativo de Google, basado en Android, dedicado exclusivamente a los wearables existentes y pensado para el soporte de los nuevos que puedan aparecer.

Su elección se basó en la gran cuota de mercado de los usuarios de Android (más del 70%) y en su modelo de desarrollo de código abierto. Su Software Development Kit (SDK) proporciona las herramientas necesarias para desarrollar aplicaciones en cualquier dispositivo que soporte esta tecnología. También, se valoró su constante desarrollo y evolución desde su presentación oficial en 2014, con mejoras tales como su compatibilidad con el sistema operativo de iOS.

2.2. Estado del arte

2.2.1. Autorregulación emocional en el TEA

Antes de empezar este proyecto, se ha realizado una investigación de las publicaciones de asistencia tecnológica existentes dedicadas a las personas con TEA y relacionadas con la regulación de emociones, con el fin de conocer sus bases científicas, metodologías, objetivos, etc.

Para realizar la selección de la bibliografía relacionada con el campo de investigación de este trabajo, se ha seleccionado el método de mapeo sistemático o *Systematic Mapping Study* (SMS) [13].

2.2.1.1. Estudio de mapeo sistemático

El estudio de mapeo sistemático es una forma de obtener información sobre el estado del arte, de tal manera que la información sea reproducible y estándar. De esta forma, se consigue abarcar la gran mayoría (o los más destacados) de los documentos presentes en la literatura, sin dejar fuera ninguna publicación importante.

El protocolo a seguir para desarrollar el método de SMS está formado por varias fases: Definición de las preguntas de investigación, realización de las cadenas de búsqueda y elección de los criterios de inclusión y exclusión.

2.2.1.1.1. Preguntas de investigación

El primer paso a realizar consiste en definir los objetivos de la investigación para extraer una serie de preguntas que determinan qué información se desea responder.

En este caso, vamos a centrar la investigación en la asistencia tecnológica aplicada en el TEA. Se buscarán tecnologías no sólo relacionadas con dispositivos wearables sino también con otros dispositivos en general, con el objetivo de tener una información más de ellos para posteriormente estudiarlos y seleccionar aquellos que se puedan adaptar.

Partiendo de esta meta, y considerando el objetivo de la regulación emocional, la pregunta principal de investigación planteada es la siguiente:

RQ1: ¿Cuáles son las asistencias tecnológicas para personas con trastorno de espectro autista para la gestión emocional?

2.2.1.1.2. Cadenas de búsqueda

En esta segunda fase del SMS se extraen las palabras clave o *keywords* a partir de la bibliografía de partida del proyecto y de la pregunta de investigación, que son los conceptos principales a tratar en el proyecto. Una vez seleccionadas, se forman las cadenas de búsqueda, que serán introducidas en los motores de búsqueda de publicaciones para obtener la bibliografía con la que se responderá a la pregunta planteada anteriormente.

Para este trabajo, las palabras clave que se obtuvieron se agruparon en tres grupos:

- Usuarios objetivo: Personas con trastorno del espectro autista.
- Intervención: Autorregulación emocional y, en general, otros tipos de gestión de las emociones.
- Tecnología: Aplicaciones móviles, wearables o cualquier otro tipo de herramienta tecnológica.

A partir de estas palabras clave se ha extraído la consulta de búsqueda, generada tras un proceso iterativo donde se han ido añadiendo términos para refinar los resultados. La cadena de búsqueda final obtenida ha sido:

```
(TITLE-ABS-KEY ("autism" OR "autism spectrum disorder" OR "ASD")  
AND ("assistive technology" OR "application" OR "authoring tool" OR "smartwatch" OR  
"wearable" OR "authoring system")  
AND ("emotion" OR "emotional self-regulation" OR "emotional regulation" OR "social deficits"  
OR " behavioral deficits " OR "emotional dysregulation" OR "emotional learning")) AND  
PUBYEAR > 2006  
AND (LIMIT-TO (LANGUAGE,"English" )) AND (LIMIT-TO (SUBJAREA,"COMP" ) )
```


Para cada uno de los grupos de palabras clave se han agregado numerosos sinónimos que se han considerado necesarios para la obtención de un conjunto de resultados de mayor calidad. Asimismo, se han limitado los resultados al inglés y al área de ciencias de la computación. Otro filtro importante es el de sólo obtener resultados posteriores al año 2006, ya que así tratamos tecnologías más recientes y dado al bajo número de publicaciones en fechas anteriores.

La plataforma de búsqueda de bibliografía que se ha utilizado ha sido *Scopus*. Este buscador pertenece a la editorial Elsevier e incluye en su índice artículos de otras revistas como IEEE, Springer o ACM. Además, la plataforma permite analizar los resultados mostrando datos respecto a un campo concreto, tales como los autores representativos, las universidades destacadas o cuántas publicaciones por año se hacen respecto a dicho campo.

2.2.1.1.3. Criterios de inclusión y exclusión

Tras la realización de las búsquedas, se debe realizar una limpieza de las publicaciones obtenidas como resultado aplicando sobre éstas criterios de inclusión y exclusión, de forma que se obtiene una primera selección de artículos.

Los criterios de inclusión son los siguientes:

- El título o *abstract* habla de los problemas emocionales o conductuales de las personas con autismo.
- El título o *abstract* trata de tecnología dirigida a interceder en la regulación emocional, conductual o alguna otra fuertemente asociada.
- El título o *abstract* habla de terapias en dispositivos móviles o adaptable a ellos.

Los criterios de exclusión son los siguientes:

- El artículo trata los temas anteriores (problemas emocionales, TEA, asistencia tecnológica) de forma secundaria.
- En el artículo no se mencionan tecnologías comparables con dispositivos móviles actuales.
- En el artículo se tratan los objetivos buscados sólo de forma teórica.
- El artículo se centra en la detección de emociones de otras personas y no en la gestión de las propias.
- El artículo habla de la detección de emociones por parte de un sistema y no por parte de un usuario real.

Como puede observarse, todos los criterios de inclusión sólo analizan el contenido del título y abstract (nivel superior), mientras que los de exclusión sí realizan un análisis más profundo del contenido del artículo (nivel inferior).

2.2.1.1.4. Resultados

Tras la búsqueda realizada en Scopus se ha encontrado un total de 41 documentos, cuya distribución por año puede apreciarse en la Figura 3.

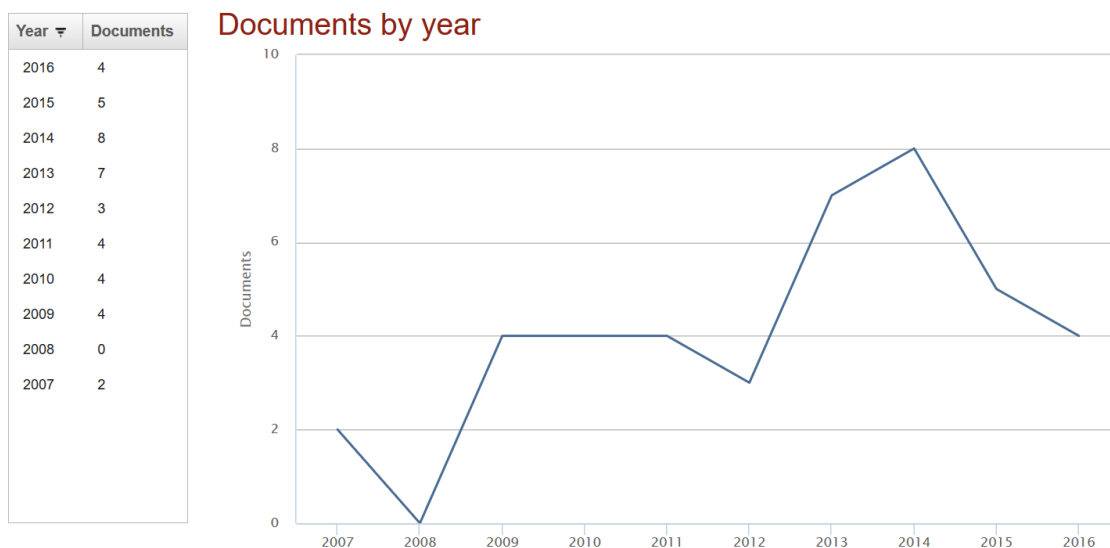


Figura 3. Scopus. Análisis de resultados obtenidos

La selección final tras aplicar los criterios de inclusión y exclusión está formada por un conjunto de dos artículos, que son los que más se aproximan a la pregunta de investigación. La definición y características principales de las tecnologías de asistencia de gestión emocional para personas con TEA (RQ1) se detallan en los siguientes puntos:

- Asistente emocional robot: En la publicación de la Universidad de Ysukuba [14], Japón, se propone una actividad afectiva asistida por un robot (ARAA) dedicada a reforzar las respuestas emocionales positivas de las personas con TEA. Esto se hace con el fin de corregir y desarrollar futuros comportamientos sociales espontáneos que también sean positivos. Para las opciones de personalización del robot por parte de los terapeutas, ofrecen un framework para controlar sus acciones en tiempo real.
- Jessie, agente de tutoría afectiva: Este estudio hace uso de una aplicación integrada de aprendizaje especializado (ISLA) que ayuda a las personas con autismo a gestionar sus emocionales, analizándolas durante sesiones de aprendizaje matemático [15]. El agente pedagógico, llamado Jessie, detecta el estado afectivo de cada niño durante el aprendizaje y le da refuerzos positivos cuando la respuesta es correcta o mensajes motivadores en caso contrario. Además, si el usuario necesita ayuda, el agente es capaz de dar consejos personalizados dependiendo del perfil cognitivo y el perfil emocional del niño.

2.2.2. Aplicaciones móviles relacionadas

En el contexto de la edición 2016 de la Jornada Internacional sobre nuevas tecnologías aplicadas al trastorno del espectro autista (TECTEA), podemos apreciar el gran número de aplicaciones dedicadas a asistir a las personas con TEA.

Dentro del grupo de aplicaciones educativas que podemos encontrar, las que más destacan relacionadas con la gestión emocional son:

- **Manos quietas:** Aplicación para mejorar el autocontrol de una persona con TEA, la cual que debe mantener un número determinado de dedos sobre la pantalla del iPad con el fin de poder ver la reproducción de un video. Permite la personalización del video, de los dedos a utilizar, de la duración del telón que se presenta antes del video, etc. Además, hace uso de refuerzos y premios, estrategia habitual en la intervención en autismo.

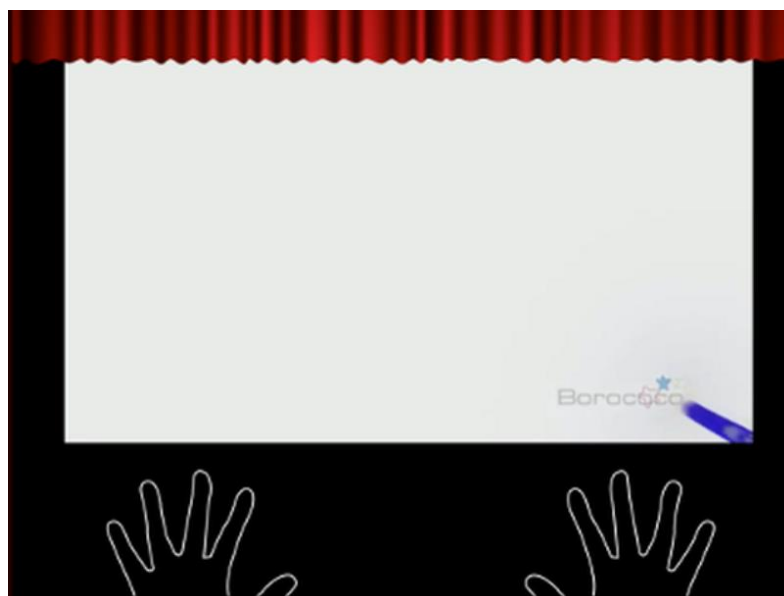


Figura 4. Manos quietas. Video en reproducción

- **Calm counter:** Esta aplicación para tablets muestra una cuenta atrás que va desde un pictograma representando un estado de enfado, que va variando hasta llegar a un pictograma de tranquilidad. El cambio de un pictograma a otro se hace mediante el tacto. Se trata de una herramienta que requiere ser trabajada para llegar a ser eficaz. Dentro de las opciones de configuración son sólo de audio (en español, inglés o sin él).



Figura 5. Calm counter. Evolución del enfado con pictogramas

- **Adiós enfados:** Aplicación para iOS que muestra la historia de un niño que siempre está enfadado y cuyos padres no saben cómo tranquilizarle. La historia cambia cuando un hada le enseña al niño trucos para no enfadarse más. Incluye páginas con contenido interactivo, opciones de personalización con la foto del usuario, selección de idiomas (español e inglés) y opciones de compartir páginas de la historia en Facebook.

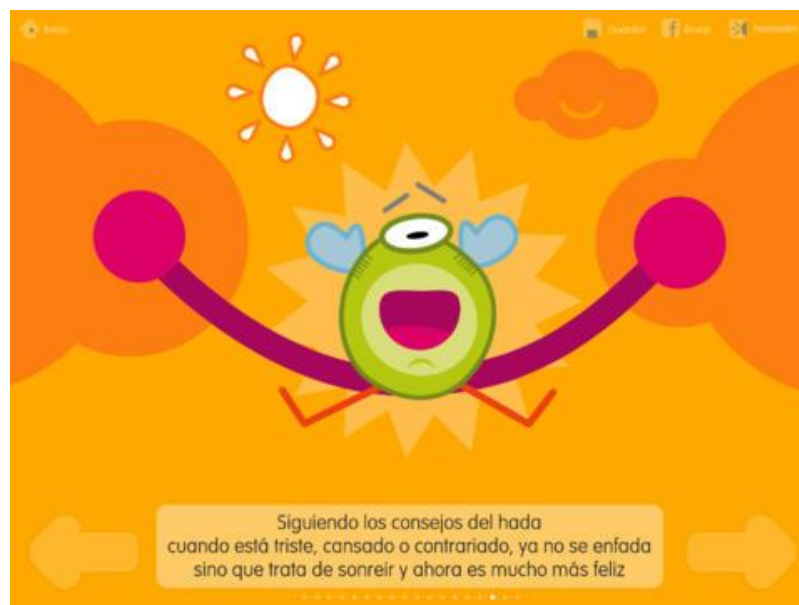


Figura 6. Adiós enfados. Consejos del hada

- **Respira, piensa, actúa:** Sesame Street ofrece una aplicación de regulación emocional cuyo objetivo es ayudar a un monstruo a calmarse. Ofrece cinco escenarios interactivos, actividad para practicar respiraciones con el monstruo, es bilingüe y contiene audio motivador que puede ser personalizado para cada niño.



Figura 7. Respira, piensa, actúa. Pantalla de interacción

2.3. Conclusiones del estado del arte

Tras analizar las distintas tecnologías y publicaciones relacionadas con la regulación emocional, hemos realizado una comparativa de las mismas, que queda reflejada en la siguiente tabla:

	Positivo	Negativo
Asistente emocional robot	Potencia respuestas emocionales positivas mediante refuerzos. Su interfaz es llamativa.	Es preventivo, no actúa en el momento de crisis. Es poco portable. El framework de control puede resultar complejo de utilizar.
Agente de tutoría afectiva	Adapta sus acciones al estado emocional actual de cada usuario. Mantiene un seguimiento de los niños.	Está limitado a un contexto específico. No es portátil.
Manos quietas	Ayuda al autocontrol de las emociones. Es motivador y muy personalizable.	Es de pago. Calma la crisis, pero no la detecta automáticamente, es necesario que otra persona la abra.
Calm counter	El uso de pictogramas ayuda a su comprensión por usuarios con TEA.	Es de pago y poco personalizable. Dirigida a momentos de crisis leve, que no detecta automáticamente. Necesita ayuda externa para entenderla inicialmente.

Adiós enfados	Enseña formas de autocontrol de las emociones. Permite personalizar la historia con la foto del usuario. Tiene contenido interactivo.	Algunas opciones de personalización son de pago. Es preventiva, no actúa en el momento de crisis ni la detecta. Compatible sólo con iOS.
Respira, piensa, actúa	Ofrece un asistente con el que se pueden aprender y ensayar técnicas para la autorregulación emocional.	También es preventiva y no detecta estados actuales de crisis. Dirigida a niños de hasta 5 años y sólo compatible con iOS.

Tabla 1 - Comparativa de tecnologías de regulación emocional

Podemos concluir que las intervenciones de regulación emocional usan, generalmente, refuerzos positivos como estímulo para el control adecuado de las emociones. Muchas de estas tecnologías requieren la asistencia de otra persona, ya sea para controlar las respuestas en tiempo real, como en el caso del robot, o para iniciarlas cuando el usuario se encuentre en una situación de crisis. El que no puedan detectar crisis e iniciarse automáticamente es una de las principales desventajas que presentan.

Otra limitación importante es que, en los dos primeros casos, no se trata de dispositivos fácilmente portables. En el resto de aplicaciones se usan tablets como dispositivo móvil, pero pueden no estar siempre accesibles al usuario final, como sí sería con el uso de wearables. Esta portabilidad ha limitado el número de artículos estudiados ya que varios trataban de sistemas complejos y el más portable que encontramos era uno que hacía uso de FPGAs, por lo que fueron descartados.

Otro motivo que redujo enormemente el número de publicaciones obtenidas en el SMS fue que una gran mayoría se centraba en el reconocimiento de emociones por parte de los sistemas y no en su control por parte de los usuarios. Trataban de computación afectiva capaz de aprender emociones, detectarlas, adaptarse y responder a ellas mostrando determinados contenidos, imitar expresiones faciales, etc. Si bien el objetivo final era su extrapolación para ayudar a las personas con TEA con la gestión emocional, este tema era tratado de forma secundaria como una aplicación a futuro.

La aplicación global Tic-Tac-TEA, considerando su parte *wear* y su parte *mobile*, supone una herramienta completa capaz de detectar situaciones de crisis y de responder ante ellas de forma automática gracias a sus sensores. Además, permite una completa personalización de la intervención a mostrar en el reloj y es totalmente portable. Responde a una necesidad cuya solución no existe en el mercado actual.

3. Análisis

Este proyecto cuenta con el asesoramiento y aprobación del Instituto de Psico – Pediatría “Dr. Quintero Lumbreras”. En particular, se han mantenido una serie de reuniones con su directora, Guadalupe Montero de Espinosa Espino, y con otras terapeutas de apoyo.

En estas reuniones se ha tratado las necesidades de las personas con TEA y se ha hecho referencia a los recursos educativos actuales existentes. Las profesionales indicaron que eran pocas las aplicaciones que ofrecían opciones de autorregulación emocional y que implementar una herramienta como la propuesta podría suponer un factor diferencial.

A partir de estas reuniones, y de forma iterativa, se fueron refinando los requisitos necesarios para la aplicación. Este proyecto, por tanto, sigue una metodología iterativa incremental con un diseño de software centrado en el usuario.

Las características obtenidas para la herramienta de autor debían ser:

- Ser una aplicación para la creación y edición de regulaciones emocionales, que podrán ser enviadas al smartwatch para mostrarse cuando se detecte una situación de crisis.
- Las regulaciones deben estar formadas por un conjunto de estrategias cuyo objetivo sea recuperar la calma del usuario. El número de estrategias será personalizable.
- Ser una aplicación de creación y edición de estrategias, que representan distintas actividades de relajación.
- Las estrategias deben estar formadas por tarjetas o pasos. El número y el contenido de pasos será personalizable.
- Los pasos o tarjetas pueden contener textos o imágenes.
- La aplicación debe ofrecer pictogramas que, cuando sean mostrados en el reloj, faciliten la comprensión por parte de los usuarios con TEA.
- Se debe dar la opción de ofrecer un refuerzo positivo tras completar una regulación o tras recuperar la calma (detectada por los sensores).

- Es necesario que las estrategias sean muy personalizables tanto en contenido como en configuración (duración, forma de transición de una a otra, entre otras opciones).
- Permitir el seguimiento del progreso de los usuarios.
- Permitir exportar e importar el backup de la herramienta.

Una vez definido el concepto general de la aplicación, queda por especificar un ejemplo de lo que sería crear una regulación emocional personalizada. Por ejemplo, si queremos crear una regulación con dos estrategias, en primer lugar, tendríamos que crear esas estrategias con sus opciones de configuración. Vemos:

- **Contar hasta 10:** Estrategia que contiene 10 tarjetas o pasos. Cada tarjeta contiene una imagen con los números que van del 1 al 10. La transición entre tarjeta y tarjeta se define que sea mediante tacto en la pantalla del reloj. Se activa la opción de ofrecer refuerzo positivo y se guarda la estrategia.
- **Sentarse y descansar:** Estrategia compuesta por un único paso en el que se muestra el pictograma de descanso. En lugar de tacto, se define la transición (en este caso el fin) del paso por tiempo. Se selecciona una duración de 60 segundos y se activa la opción de mostrar un temporizador en la pantalla. Se guarda la estrategia.

Una vez creadas las estrategias, ya podemos crear la regulación. Empezamos indicando si queremos que el reloj vibre o no al iniciar la regulación. Luego definimos si queremos que las estrategias se muestren una tras otra o si mostraremos un selector para elegir sólo una. Después de esta configuración inicial, añadimos las dos estrategias a la regulación y guardamos.

Por último, una vez creada la regulación, podemos sincronizarla con el smartwatch para mandar la información. De este modo, esta regulación personalizada se mostrará al usuario, de la forma que se ha especificado, cuando se detecte un estado de crisis emocional.

3.1. Resumen de los Requisitos del Sistema

Después de identificar las características del sistema a implementar, procedimos a realizar el análisis de los requisitos. A continuación, se listan los requisitos básicos.

3.1.1. Requisitos funcionales

- **Req. 1. SO y API:** El sistema debe ser implementado para smartphones con sistema operativo Android. Debe funcionar correctamente con la versión 5.0 (*Lollipop*) como mínimo.
- **Req. 2. Perfil:** La aplicación debe mostrar un perfil de usuario con la información de su nombre, sexo, foto (opcional) y tipo de reloj que vaya a utilizar (redondo o cuadrado).
- **Req. 3. Edición de perfil:** Se debe permitir cambiar la información del perfil en cualquier momento que el usuario quiera.
- **Req 4. Gestión de foto de perfil:** El usuario podrá agregar, modificar o eliminar su foto, tomada desde la cámara o la galería del dispositivo móvil. En caso de no agregar ninguna foto, se debe dar la opción de usar una por defecto.
- **Req 5. Crear copia de seguridad:** La herramienta debe proporcionar la opción de crear copia de seguridad del backup de forma manual y, de ser posible, también automática.
- **Req 6. Restaurar copia de seguridad:** Siempre que exista una copia de seguridad del backup creada, el sistema debe ser capaz de restaurarla por petición manual.
- **Req 7. Exportar resultados:** La aplicación debe permitir obtener los registros de resultados guardados en el smartwatch y exportarlos a la ubicación en el smartphone que el usuario elija.
- **Req 8. Protección por PIN:** El sistema debe tener un sistema de protección por PIN para poder acceder a la información. Esta protección es opcional y puede ser activada o desactivada por el usuario.
- **Req 9. Actividad offline:** Se debe poder usar todas las opciones de la aplicación sin necesidad de conexión a internet.
- **Req 10. Lista de estrategias:** La herramienta ofrecerá la opción de visualización de todas las estrategias existentes o, en su defecto, mostrará un mensaje indicando que no existe ninguna.

- **Req. 11. Crear estrategia:** La aplicación debe permitir crear estrategias y asignarles un nombre e icono personalizado. También debe ofrecer opciones por defecto si no se quiere personalizar.
- **Req. 12. Personalización de tarjetas:** Deberá ofrecerse la opción de insertar texto en la tarjeta o insertar una imagen.
- **Req 13. Opciones de texto:** Se podrá introducir texto en una tarjeta y personalizar su tamaño, color y modo (normal, minúsculas y mayúsculas).
- **Req 14. Opciones de imagen por defecto:** La herramienta debe proporcionar una serie de pictogramas por defectos para utilizar en los pasos.
- **Req 15. Opciones de imagen como recurso propio:** Un usuario puede añadir un recurso propio desde la cámara o la galería de imágenes del dispositivo móvil.
- **Req 16. Contenido de tarjetas adaptado:** La aplicación debe permitir ver el contenido de las tarjetas adaptado a la pantalla del tipo del reloj seleccionado en el perfil del usuario.
- **Req 17. Número de tarjetas:** El sistema debe tener un botón para añadir nuevas tarjetas, con un máximo de 15 tarjetas por estrategia.
- **Req 18. Visualización de tarjetas:** Se debe poder hacer una navegación sencilla entre todas las tarjetas que componen una estrategia.
- **Req 19. Mostrar refuerzo:** La configuración de la estrategia debe mostrar la opción de activar el uso de refuerzos. De este modo, si el usuario del smartwatch recupera la calma podrá ver una imagen motivadora.
- **Req 20. Comprobación emocional:** La configuración de la estrategia debe permitir elegir si se quiere comprobación emocional o no. Esto se refiere a si el reloj debe comprobar mediante sus sensores el estado emocional del usuario para detectar si se ha calmado o, por el contrario, se debe hacer toda la regulación hasta el final.
- **Req 21. Opciones de transición:** La configuración de la estrategia permite elegir entre pasar de una tarjeta a otra mediante el toque en la pantalla del smartwatch o si cada tarjeta tiene una duración en segundos.

- **Req 22. Transición por tiempo:** Al activar esta opción también se deben activar otras relacionadas como el uso del temporizador, la definición del tiempo y la opción para saber si el tiempo es global. El tiempo, por defecto, será de 20 segundos por tarjeta (no global).
- **Req 23. Tiempo global:** Opción que indica si el tiempo es por cada tarjeta o si es el tiempo total que debe durar la estrategia.
- **Req 24. Temporizador:** La herramienta ofrecerá cuatro opciones de temporizador: Sin temporizador, temporizador tipo donut, tipo llenado de pantalla y tipo vaciado de pantalla.
- **Req 25. Guardar estrategia:** La aplicación debe permitir guardar todas las estrategias del usuario.
- **Req 26. Lista de regulaciones:** La herramienta ofrecerá la opción de visualización de todas las regulaciones existentes o, en su defecto, mostrará un mensaje indicando que no existe ninguna.
- **Req 27. Crear regulación:** La aplicación debe permitir crear regulaciones y asignarles un nombre e icono personalizado. También debe ofrecer opciones por defecto si no se quiere personalizar.
- **Req 28. Vibración inicial:** La herramienta ofrecerá la opción de si se quiere que el smartwatch vibre antes de lanzar la ejecución de la regulación
- **Req. 29. Tipo selector:** Deberá ofrecerse la opción de elegir si las estrategias de la regulación han de mostrarse de forma secuencial o si se eligen a partir de un selector en el smartwatch.
- **Req 30. Añadir estrategias a la regulación:** El sistema debe permitir agregar tantas estrategias como el usuario desee, incluyendo repeticiones.
- **Req 31. Orden de las estrategias:** Se debe permitir cambiar el orden de ejecución que tendrán las estrategias en la regulación.
- **Req 32. Eliminar estrategia:** La herramienta ofrecerá la opción de eliminar las estrategias no deseadas por el usuario.
- **Req 33. Guardar regulación:** La aplicación debe permitir guardar todas las regulaciones del usuario.

- **Req 34. Sincronizar regulación:** El sistema debe ofrecer al usuario la opción de sincronizar la regulación con el smartwatch para enviar los datos.
- **Req 35. Editar estrategia:** Se debe permitir editar las estrategias creadas.
- **Req 36. Editar regulación:** Se debe permitir editar las regulaciones creadas.
- **Req 37. Tutorial:** El sistema debe ofrecer una sección de tutorial con las explicaciones de las acciones principales de la aplicación.
- **Req 38. Popups informativos:** Siempre que se trate de una opción de configuración compleja, ya sea en estrategias o regulaciones, la herramienta debe mostrar un popup con una breve descripción del funcionamiento de la opción.
- **Req 39. Recursos GIF:** La aplicación debería permitir el uso de recursos de tipo gif en la creación de las tarjetas de las estrategias.

3.1.2. Requisitos no funcionales

3.1.2.1. *De recursos*

- **Req 1.** Aunque sólo se tiene una sincronización de usuario por smartwatch y smartphone, la base de datos debe considerar el registro de más usuarios, por si esta configuración cambia a futuro.
- **Req 2.** Las fotos y recursos propios de los usuarios deben ser almacenados en una carpeta propia y privada de la aplicación.

3.1.2.2. *De usabilidad*

- **Req 3.** La interfaz deberá estar disponible en castellano y en catalán.
- **Req 4.** El sistema deberá contar con un manual de usuario y acciones de ayuda para asistir las dudas que le puedan surgir al usuario.
- **Req 5.** La aplicación proporcionará mensajes de error informativos y orientativos.

- **Req 6.** La herramienta contará con interfaces gráficas bien formadas.

3.1.2.3. De operatividad

- **Req 7.** Los usuarios no familiarizados con la aplicación no deberían tardar más de quince minutos en manejar y comprender la dinámica de las estrategias y las regulaciones.

3.1.2.4. De escalabilidad

- **Req 8.** El sistema debe ser construido sobre una base que posibilite un desarrollo evolutivo e incremental. Nuevos requerimientos y definiciones de estrategias deben ser fácilmente incorporados.

3.1.2.5. De robustez

- **Req 9.** En caso de errores excepcionales, la herramienta debe ser capaz de recuperarse, mostrando al usuario la información de los últimos datos guardados.

4. Diseño

4.1. Diagrama de clases

En esta sección veremos el diagrama de clases simplificado de la aplicación. Lo hemos dividido en cinco partes: cuatro que representan las actividades y fragmentos de lógica, y otra representando el diagrama de clases de los modelos de datos diseñados.



Figura 8. Diagrama de clases Tic-Tac-TEA. Parte I



Figura 9. Diagrama de clases Tic-Tac-TEA. Parte II

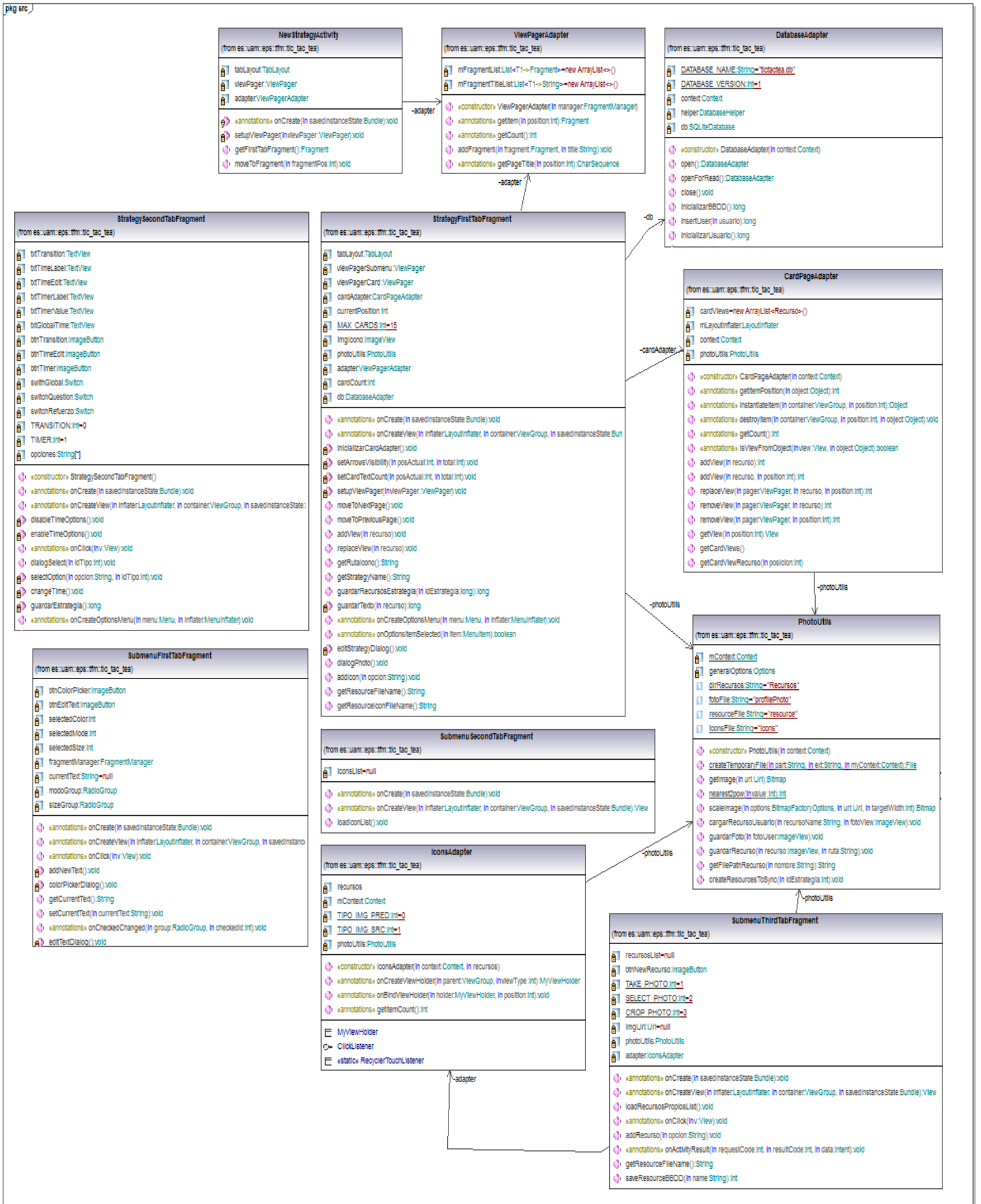


Figura 10. Diagrama de clases Tic-Tac-TEA. Parte III



Figura 11. Diagrama de clases Tic-Tac-TEA. Parte IV

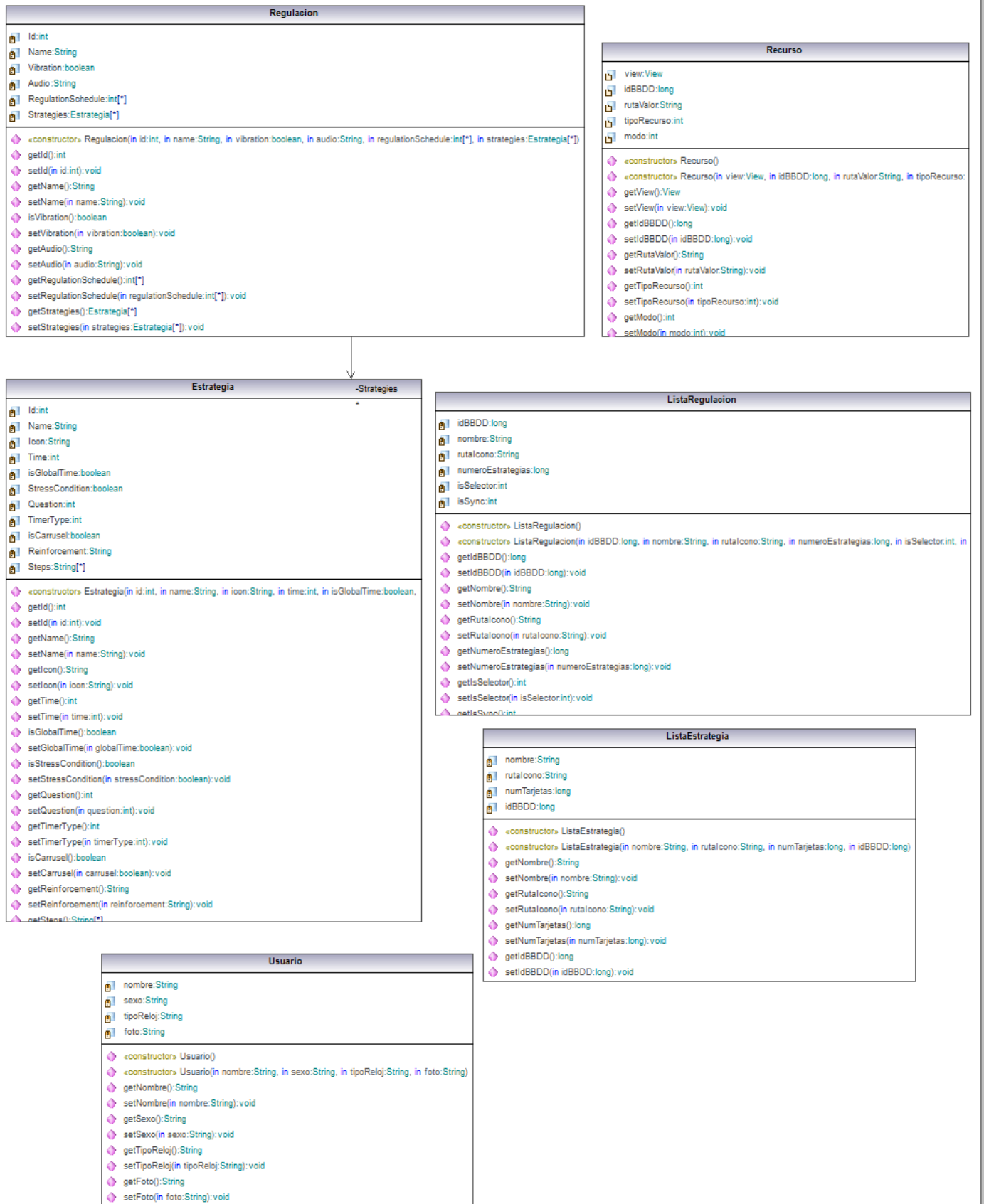


Figura 12. Diagrama de clases Tic-Tac-TEA. Parte V - Modelo

4.2. Análisis del diseño de la aplicación

Para el diseño de la aplicación se tuvo en cuenta el Material Design de Google, introducido a partir de Android 5.0 (nivel de API 21). Se trata de una guía completa para el diseño visual, de movimientos y de interacción en las aplicaciones. Es multiplataforma y se adapta a distintos dispositivos. Además de las pautas de diseño que ofrece, se incluyen nuevas componentes y funcionalidades como widgets para vistas complejas, nuevas APIs y un nuevo tema.

Dado que se siguió un diseño iterativo de la aplicación, centrado en el usuario, en cada reunión con el I.P.P. “Quintero Lumbreras” se fue refinando la maqueta de la interfaz. Esta evolución (ver Anexo A) refleja tanto la evolución de los requisitos de la herramienta como la aplicación de las pautas del Material Design.

El resultado final del diseño queda reflejado en el diagrama de clases. Como se trata de una aplicación Android, a parte de las clases estándar de Java tendremos unas clases especiales que serán las *Activities*. Cada clase de tipo *Activity* estará relacionada con una pantalla principal de la herramienta móvil y tendrá su correspondiente fichero *XML* en el que se definirá la interfaz de usuario. Además, haremos usos de *Fragmentos*, que son porciones de interfaz embebidas en actividades anfitrionas. Estos serán útiles para los casos en que necesitemos navegación entre pestañas o en la pantalla principal, en la que varias vistas comparten un mismo menú lateral. En estos casos, cada pestaña sería un fragmento y cada vista dentro de la pantalla principal, también.

Empezaremos realizando el análisis de las clases de la Figura 8, que son las primeras que la aplicación ejecutaría (sin contar el Main) y las que irían en el menú lateral. Véanse:

- **SplashActivity:** Esta clase contendrá el logo de la Fundación Orange y del I.P.P. “Quintero Lumbreras” con una animación de tipo *alpha*. Además, en esta clase se hará uso de las Preferencias de la aplicación para saber si se tiene la BBDD inicializada o no. En caso negativo, se inicializarán los valores de los recursos predeterminados.
- **PasswordActivity:** Si el usuario ha establecido un PIN de seguridad, tras el SplashActivity se iniciará esta actividad. Aquí se le solicitará al usuario el PIN correspondiente y no se podrá acceder al contenido de la aplicación a menos que se efectúe una entrada correcta.
- **ProfileActivity:** Si en el menú lateral se selecciona la cabecera de la información del usuario, se crea esta actividad. En esta clase, el usuario podrá ver su nombre, género, tipo de reloj, foto de perfil y número de estrategias y de regulaciones que tenga. Además, habrá un botón para editar estos datos.

- **EditProfileActivity:** Si en la pantalla anterior se selecciona el botón de editar, se creará esta actividad. Esta clase contendrá los campos de datos del usuario que podrán ser editados. Habrá un botón para guardar los cambios y otro para cancelar y volver a la pantalla anterior.
- **SettingsActivity:** Esta actividad corresponde a la opción de ‘Configuración’ que estará disponible en el menú lateral de la aplicación. Contendrá las opciones de backup de la herramienta, de exportación de logs y de PIN de seguridad.
- **TutorialActivity:** Ésta será otra opción disponible en el menú lateral. Se tratará de una serie de imágenes con información explicativa de los conceptos y acciones principales de la aplicación como el uso de estrategias y regulaciones.
- **Preferencias:** Si bien no hemos relacionado esta clase con el resto en el diagrama (por cuestiones de simplicidad), la clase de preferencias será utilizada por varias clases de la aplicación. En las preferencias tendremos información del usuario, sus opciones de configuración y la variable booleana para saber si la BBDD está inicializado o no.

Continuamos analizando la Figura 9, en la que podemos ver la actividad principal de la aplicación. Tenemos:

- **MainActivity:** Es la actividad inmediatamente siguiente a *Splash* o a *PasswordActivity*, dependiendo de si el usuario tiene puesto un PIN de seguridad o no. Esta actividad principal contendrá tres fragmentos correspondientes a las pantallas de Home, Estrategias y Regulaciones, y un menú lateral desplegable que será común a los fragmentos.
- **HomeFragment:** Este fragmento será el principal y el que primero se mostrará en el MainActivity. Mostrará un fondo de smartwatch y dos botones para navegar a la pantalla de regulaciones o a la pantalla de estrategias.
- **StrategyFragment:** Este fragmento muestra la lista de estrategias que el usuario haya creado o un mensaje informativo en caso de que no haya ninguna. Además del nombre de la estrategia, se mostrará su icono y el número de tarjetas que contiene. Será posible editar las estrategias seleccionando alguna. Por último, habrá un botón para crear una nueva estrategia.
- **RegulationFragment:** Este tercer fragmento contendrá la lista de regulaciones del usuario o un mensaje informativo en caso de no haber ninguna, como en el caso del fragmento anterior. Se mostrará el nombre de la regulación, su icono y

el número de estrategias que contiene. Se podrá acceder a la edición de la regulación al seleccionar una de la lista. Habrá un botón para sincronizar la regulación con el smartwatch y otro para crear una regulación nueva.

En la Figura 10 vemos las clases relativas a la creación de una nueva estrategia. Dado que los requisitos que se solicitaron para esta funcionalidad eran muchos y muy complejos, se tuvo especial cuidado y refinamiento en su diseño. Se propuso el uso de pestañas para dividir la información, y de un submenú de opciones para cada tipo de contenido de tarjeta de la estrategia (texto, imagen predeterminada y recurso propio). El resultado queda definido en las siguientes clases:

- **NewStrategyActivity:** Actividad principal para la creación de nuevas estrategias. Contendrá dos fragmentos que se corresponderán con dos pestañas de división de información: En la primera se realizará la edición de tarjetas y en la segunda se configurará las opciones globales de la estrategia.
- **StrategyFirstTabFragment:** El primer fragmento de la actividad anterior. Se encargará de la creación y edición de las tarjetas. Tendrá un submenú inferior que permitirá elegir si agregar un recurso de tipo texto, imagen predeterminada o imágenes agregadas por el usuario. Este submenú estará compuesto por tres sub-fragmentos.
 - **SubmenuFirstTabFragment:** Aquí podremos configurar los recursos de tipo texto a incluir en las tarjetas. Se podrá editar el texto, cambiar el color, el tamaño y el modo (normal, mayúsculas y minúsculas).
 - **SubmenuSecondTabFragment:** Esta opción del submenú nos mostrará una lista en grid de todos los pictogramas predeterminados de la aplicación, incluidos los de tipo gif.
 - **SubmenuThirdTabFragment:** En la tercera opción se mostrarán las imágenes que haya subido el usuario, si existen. Además, habrá un botón para incluir nuevos recursos desde la cámara o la galería de imágenes.
- **StrategySecondTabFragment:** En esta segunda pantalla de la creación de una estrategia, podremos configurar opciones tales como la forma de transición de las tarjetas (tacto o tiempo), si mostrar o no un temporizador, si se requiere el uso de refuerzos, etc.

La última actividad por revisar es la encargada de crear una nueva regulación. Vemos el análisis de su clase, correspondiente a la Figura 11:

- **NewRegulationActivity:** Clase encargada de la creación de nuevas regulaciones. Hace uso de clases auxiliares para la carga de estrategias y su manipulación. Permite añadir estrategias, repetirlas, eliminarlas y cambiar el orden de las mismas.

Tras analizar las *Activities* de la aplicación, detallaremos las clases restantes que se muestran en la Figura 12, que representan modelos creados para la manipulación de datos y clases auxiliares. Éstas son:

- **DatabaseAdapter:** Clase encargada de la gestión completa de la base de datos de la aplicación (creación, actualización, lectura, etc.). Contendrá las queries para la creación de todas las tablas, además de funciones de acceso a las mismas, tanto de lectura como escritura.
- **Recurso:** Esta clase representará un objeto caracterizado por tener un tipo (imagen o texto), una ruta y un identificador de base de datos relacionado, principalmente. Con ella podremos gestionar fácilmente y de forma genérica los recursos utilizados por las tarjetas de las estrategias. Se utilizará específicamente para la gestión de las mismas al crear o editar estrategias.
- **ListaEstrategia:** Es un modelo de objeto necesario para mostrar la lista de estrategias. Aquí cargamos los atributos principales que definen a una estrategia, es decir, su id, nombre, icono y número de tarjetas que contiene.
- **ListaRegulacion:** Aquí almacenaremos los atributos que definen al resumen de una regulación, es decir, su id, nombre, icono, numero de estrategias que contiene, si es de tipo selector y si está sincronizada con el reloj inteligente o no. Esta clase nos facilitará la gestión para mostrar la lista de regulaciones.
- **Estrategia:** Esta es la representación del modelo común de estrategia que usará la aplicación móvil y el smartwatch. Aquí almacenaremos todos los datos que definen a una estrategia completa.
- **Regulación:** Esta es la representación del modelo común de regulación que usará la aplicación móvil y el smartwatch. Aquí almacenaremos todos los datos que definen a una estrategia completa, incluyendo la lista de modelo de estrategias anterior.

El resto de clases mostradas en los diagramas son clases auxiliares tales como adaptadores, necesarios para el soporte de las nuevas componentes del Material design. Por último, la clase auxiliar PhotoUtils, de código abierto (crédito a la “Comunidad de Desarrolladores”) se utilizará para la manipulación de gráficos.

4.3. Modelo de Datos

Para entender el modelo de datos utilizado en la herramienta estudiaremos primero el diagrama E-R de la base de datos y luego, detallaremos las tablas empleadas junto con sus relaciones. Veamos el diagrama:

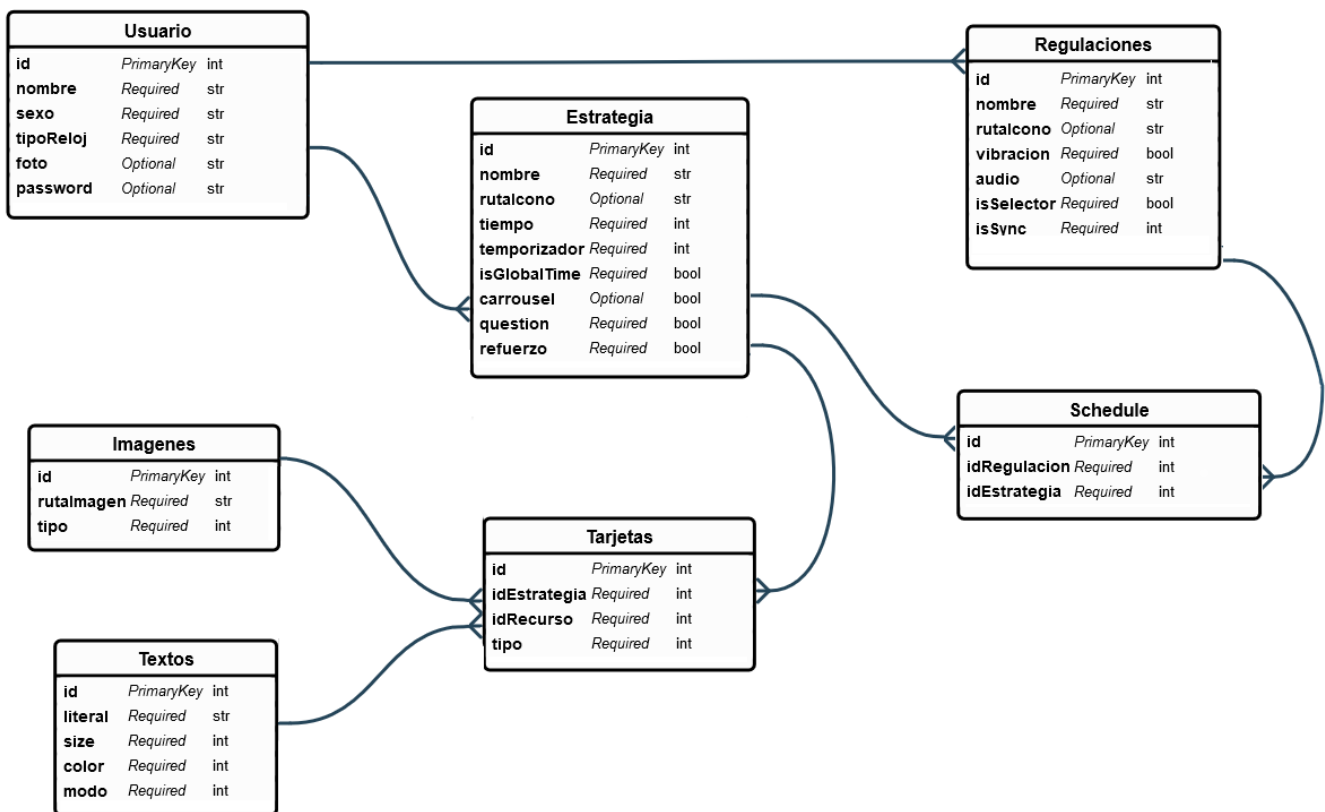


Figura 13. Diagrama de E-R de Tic-Tac-TEA

Como podemos observar, la aplicación constará de siete tablas de base de datos. Sus utilidades y relaciones serán las siguientes:

- **Usuario:** En esta tabla guardaremos la información de los usuarios de la aplicación. Aunque, de momento, sólo se tenga uno por aplicación, la tabla está pensada para ser escalable por si en un futuro se permite un número mayor de usuarios.

- **Imágenes:** Esta tabla representa dos de los tres posibles tipos de tarjetas a tener en una estrategia. La tabla contendrá el identificador de la imagen, su ruta y su tipo (predeterminado o recurso propio).
- **Textos:** Esta tabla representa el tercer tipo de tarjeta posible en una estrategia. En esta tabla se almacenarán el literal del texto y sus opciones de configuración (tamaño, color y modo).
- **Tarjetas:** Esta tabla actuará como medio de relación entre las tablas de Imágenes, Textos y Estrategia. Dado que una tarjeta de estrategia puede tener opciones tanto gráficas como textuales, esta tabla almacenará el tipo de recurso (Imagen, Texto) y el identificador del recurso para saber cuál está utilizando una estrategia en particular. Esta tabla también contendrá, lógicamente, el identificador de la estrategia a la que pertenece la tarjeta. Al diseñar las tarjetas en una tabla aparte (no incluida en la propia tabla de Estrategias), también permitiremos tener un número variable de tarjetas.
- **Estrategia:** Esta tabla queda definida por su identificador de estrategia, su nombre, la ruta de su icono (opcional), su opción de tiempo (0 si transición por tacto, mayor a 0 indica los segundos de duración), tipo de temporizador (0 sin temporizador y mayor que 0 indica el tipo de temporizador), tiempo global (si es verdadero el tiempo definido es para toda la estrategia, si no, el tiempo es por cada tarjeta), si es de tipo carrusel (opcional), si tiene una pregunta de comprobación emocional y si debe utilizar refuerzos una vez que el usuario del reloj recupere la calma.
- **Regulaciones:** En esta tabla guardaremos la información referente a las regulaciones emocionales de la herramienta de autor. Éstas estarán definidas por su id, su nombre, la ruta de su icono (opcional), si tienen vibración inicial, su audio (opcional), si son de tipo selector y si están sincronizadas o no con el smartwatch.
- **Schedule:** Esta tabla actuará como medio de relación entre las tablas de Regulaciones y Estrategia, por lo que se tendrá una referencia del identificador de cada entrada de ellas. Al diseñar las *schedule* en una tabla aparte (no incluida en la propia tabla de Regulaciones), permitimos tener un número variable de estrategias, con lo que facilitamos la gestión de las mismas.

En la arquitectura de la base de datos podemos observar que tanto la tabla de Estrategias como la de Regulaciones poseen un atributo de ruta de icono. En este atributo opcional guardaremos la ruta del recurso de imagen subido por el usuario, que

estará almacenado en una carpeta privada de la aplicación. En caso de no tener valor en este campo, se entiende que el usuario hará uso del icono por defecto brindado por la aplicación.

También podemos observar que la tabla de Imágenes y la tabla de Tarjetas poseen un atributo de “tipo”. Este campo será un número entero que resultará verdaderamente útil para la gestión de las diferentes clases de imágenes y tarjetas que podamos tener. En el caso de las imágenes, el tipo representará si se trata de un recurso predeterminado o si es un recurso creado por el usuario. En el caso de las tarjetas, diferenciará entre el uso de recursos de imagen o de texto.

Respecto a la tabla de Tarjetas y Schedule, hemos indicado que servirán de conexión entre las tablas de Imágenes, Textos y Estrategia, y entre las tablas de Estrategia y Regulaciones. Esta versatilidad en el número de tarjetas y schedule no sería posible si almacenáramos toda la información sólo en la propia tabla de Estrategias y Regulaciones. Así pues, con el diseño de estas tablas evitamos la redundancia de datos y la mezcla de información. Conseguimos, de este modo, proteger la integridad de los datos y facilitar su actualización.

En conclusión, el diseño de la base de datos propuesto pretende permitir la máxima escalabilidad posible. Con este diseño agregar un nuevo tipo de tarjeta (un nuevo tipo de recurso) sería una tarea sencilla.

4.4. Diseño de Estrategias

Para el diseño pedagógico de las estrategias hemos tenido en cuenta las recomendaciones realizadas por las expertas del I.P.P. “Quintero Lumbreras”, y las características de las personas con autismo.

Debido a las dificultades para la comunicación no verbal de ideas y sentimientos de las personas con TEA, se ha decidido hacer uso de pictogramas como material alternativo de comunicación. En particular, se utilizarán los pictogramas del Portal Aragonés de la Comunicación Aumentativa y Alternativa (ARASAAC) [16] (ver Figura 14), debido a que están especialmente diseñados para facilitar la comunicación y resultan accesibles a las personas con TEA.

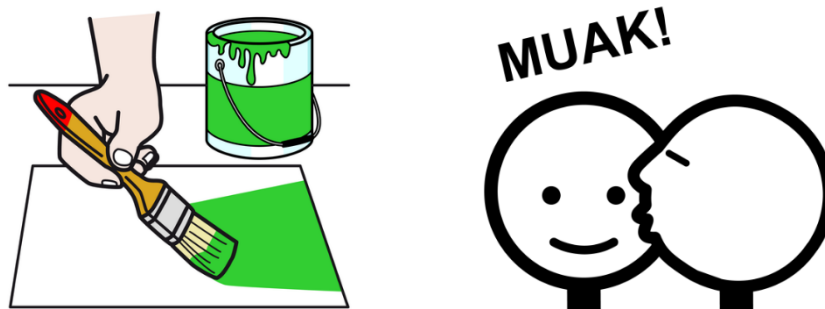


Figura 14. Pictogramas de ARASAAC. Pintar y besar

También se definió utilizar sólo un recurso de tipo imagen o de tipo texto por tarjeta. En un principio se pensó en combinar ambos pero, finalmente, esa idea se descartó porque hubiera sido mucho contenido adaptado a la pantalla de un smartwatch y debido a que no es completamente necesario acompañar los pictogramas por texto, estos ya se explican por sí solos. De este modo, la información de las tarjetas de las estrategias se presenta de forma simple y comprensible.

Esta estructura de estrategias, aunque pueda parecer rígida, permite una gran combinación de posibilidades. Además, el hecho de poder incluir recursos propios hace que se pueda personalizar en gran medida cada estrategia a las preferencias del usuario. En la Figura 15 vemos un ejemplo que utiliza tres tarjetas con imágenes de burbujas. Si en cada imagen aparece una burbuja menos que en la anterior y se define la transición de tarjetas mediante tacto, se da la impresión de estar reventando burbujas, como si de un mini juego de relajación se tratara.



Figura 15. Ejemplo de estrategias con burbujas

Otra consideración de diseño es la relacionada con el uso de temporizadores. Las expertas comentaron la idea abstracta del tiempo que tienen las personas con TEA, con lo que mostrar una cuenta atrás de minutos o segundos no resultaría útil para ellos. En su lugar, como representación temporal, se decidió hacer uso de tres elementos visuales: donut, llenado de pantalla y vaciado de pantalla. Estos elementos se mostrarán sobre las tarjetas con una determinada transparencia e irán respondiendo al transcurso del tiempo.

5. Desarrollo

En este apartado hablaremos de los pasos seguidos en la implementación del proyecto y comentaremos el porqué de las decisiones que hemos tomado.

Empezaremos indicando el protocolo de comunicación entre el smartphone y el smartwatch, el entorno de trabajo utilizado, el lenguaje de desarrollo y la organización de los archivos del proyecto.

Continuaremos detallando en profundidad los distintos apartados que componen la aplicación. Además, comentaremos el trabajo adicional que se ha realizado en alguno de ellos, más allá de sólo la programación.

5.1. Protocolo de comunicación

Antes de empezar con el desarrollo del proyecto, se mantuvieron una serie de reuniones con el desarrollador de la parte wear de la aplicación. El objetivo de estas reuniones era definir el modelo de datos para la sincronización entre el proyecto móvil y el proyecto smartwatch.

Se optó por usar un protocolo JSON de comunicación ya que facilita la creación de estructuras y su uso es sencillo en Android. La estructura principal que tendremos será la de regulación. Vemos:

```
{
  "Regulation": {
    "Id": 15,
    "Name": "Regulacion 20",
    "Vibration": true,
    "Audio": null,
    "RegulationSchedule": [
      [
        3,
        7,
        19,
      ],
      [
        5,
        8
      ],
    ],
    "Strategies": [ ]
  }
}
```

Podemos ver que estamos incluyendo las características principales de una regulación tales como su id, su nombre, si tiene vibración inicial, su audio, un apartado especial de *RegulationSchedule*, y las estrategias que la conforman.

El campo *RegulationSchedule* define el orden en que se mostrarán las estrategias, las

repeticiones que pueda haber y si se está haciendo uso de selectores. Para ello, lo que se tiene en este campo son los identificadores de las estrategias. Si los identificadores están sencillamente separados por comas, se tratarán de estrategias secuenciales. Si por el contrario los identificadores están separados por comas, pero englobados dentro de corchetes, se tratará de un selector de estrategias. Es decir, se mostrará una lista con esas estrategias y el usuario del reloj deberá elegir una.

En el ejemplo, se tiene una regulación que mostrará primero un selector de tres estrategias y después ejecutará la estrategia 5 y 8 de forma secuencial.

El campo de *Strategies* define todas las estrategias de las que puede hacer uso la regulación (las secuenciales y las posibles a elegir en un selector). Su estructura será:

```
"Strategies": [
  {
    "Id": 3,
    "Name": "9 y 8",
    "Icon": "one.png",
    "Time": 0,
    "isGlobalTime": false,
    "StressCondition": false,
    "Question": 1,
    "TimerType": 0,
    "isCarrusel": false,
    "Reinforcement": true,
    "Steps": [
      "nine.png",
      "eight.png"
    ]
  }
]
```

Al igual que en el caso de las regulaciones, estamos definiendo todos los campos de configuración necesarios para las estrategias. Tenemos su id, su nombre, la ruta de su icono, el tiempo, etc. De esta estructura podemos destacar los siguientes dos puntos:

- El campo *isCarrusel* se definió en las primeras reuniones con las expertas, pero finalmente se descartó. Se trataba de un campo que indicaba que todas las tarjetas de la estrategia se mostraban como un carrusel cíclico. Dado que se pretende retomar el uso de este campo en el futuro se ha mantenido, pero en su uso actual siempre tendrá el valor *false*.
- En el array de *Steps* (pasos o tarjetas) siempre se pasarán el nombre de archivos de imagen. Esto se debe a que, incluso cuando tengamos tarjetas de tipo texto, enviaremos al reloj la imagen de la vista de la tarjeta. De este modo el smartwatch sólo tiene que mostrar las tarjetas y no ocuparse de su configuración.

Un ejemplo completo de este protocolo JSON puede verse en el Anexo B.

5.2. Entorno de trabajo

Para este proyecto hemos utilizado Android Studio 2.1.2. junto al SDK 23.0.0. de Android, lo que nos ha permitido tener un entorno robusto y con todas las librerías y funcionalidades necesarias para la programación en el sistema de Google.

La estructura final de las carpetas del proyecto y sus archivos contenidos para la parte “mobile” ha sido la siguiente:

- **AndroidManifest.xml:** Éste es el fichero de configuración de la aplicación móvil. En él hemos indicado que nuestro sistema usa permisos de lectura y escritura en memoria externa. Aquí también señalamos la versión mínima de la herramienta (*Req. 1 Funcional*) y todas las actividades de las que se compone. Cabe mencionar que todas las actividades se mostrarán en portrait.
- **Paquete es.uam.eps.tfm.tic_tac_tea:** Éste es el paquete principal de la aplicación, donde tendremos las clases Java mencionadas en el diseño, en particular, 34 clases.
- **Paquete es.uam.eps.tfm.tic_tac_tea.model:** Éste paquete, contenido en el anterior, tendrá las clases Java relacionadas con los modelos de datos necesarios.
- **Carpeta res/anim:** En la carpeta de recursos de la aplicación, tendremos otra destinada a las animaciones utilizadas en el proyecto. En este caso, esta carpeta contiene sólo el fichero de animaciones alpha.xml.
- **Carpeta res/drawable:** En esta carpeta almacenaremos todas las imágenes que vayamos a utilizar. Además, guardaremos ficheros XML que tendrán configuraciones personalizadas de elementos gráficos.
- **Carpeta res/layout:** Todos los ficheros de interfaz de las pantallas estarán en esta carpeta. Tendremos un fichero XML por cada pantalla o componente personalizada del sistema. En particular, el proyecto cuenta con 24 ficheros XML.
- **Carpeta res/menu:** En este lugar estarán contenidos todos los ficheros de configuración del menú lateral y del menú de opciones que irán en la toolbar.
- **Carpeta res/values:** Esta carpeta contendrá los archivos strings.xml, styles.xml y dims.xml. En ellos especificaremos constantes de texto, estilos propios definidos para determinadas vistas y tamaños de elementos varios.

- **Carpeta res/xml:** Aquí tendremos el fichero con la definición de las preferencias de la aplicación a mostrar en la pantalla de Configuración.

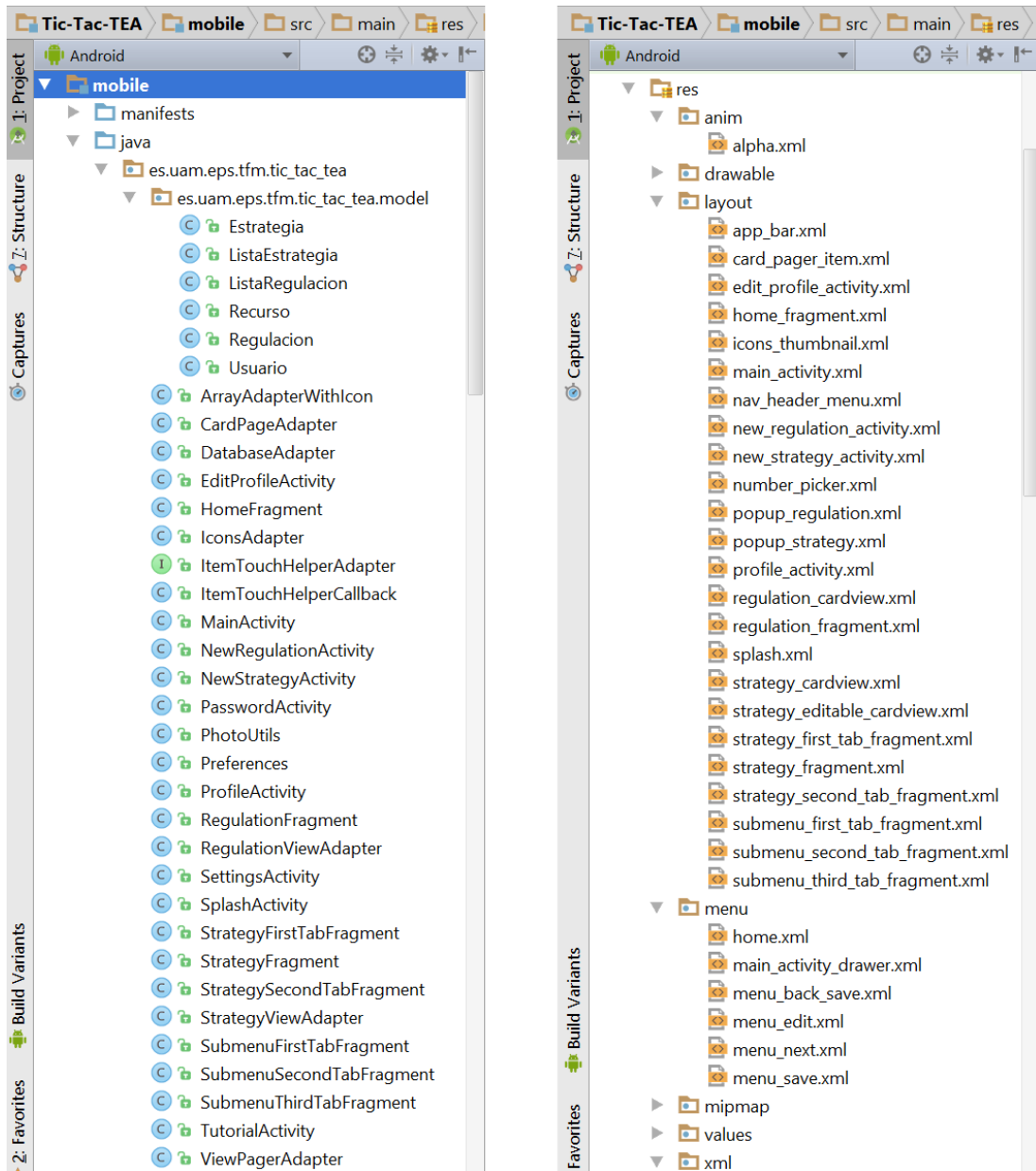


Figura 16. Estructura de las carpetas y ficheros del proyecto

5.3. Apartados de la aplicación

En esta sección hablaremos de la implementación de las distintas pantallas de la aplicación. Daremos una breve descripción de la funcionalidad de cada una de ellas, del diseño de su interfaz y de la lógica utilizada.

Sobre la interfaz de las pantallas, hemos decidido usar ficheros XML que contengan el diseño visual de cada una. De este modo, separamos la lógica de la vista de la aplicación, independizando la una de la otra.

A pesar de que existe una paleta de componentes gráficas para la creación de interfaces, hemos decidido no utilizarla y programar nuestras propias componentes en XML. Así conseguimos ajustar el diseño a nuestras preferencias y obtenemos código más limpio.

5.3.1. Pantalla inicial Splash

5.3.1.1. Descripción

Esta es la primera pantalla de la aplicación. En ella mostramos el logo de la Fundación Orange seguido del logo del I.P.P. “Quintero Lumbreras”, como referencia a la colaboración realizada con ellos y como sello de calidad del trabajo.

5.3.1.2. Interfaz

El fichero relativo a la interfaz es el de *splash.xml*. En él simplemente tenemos un *LinearLayout* que contiene un *ImageView*.

Con el fin de utilizar una animación en los logos, hemos creado en la carpeta */res/anim* el fichero *alpha.xml*. En este fichero hemos definido una animación de desvanecimiento alfa.



Figura 17. Tic-Tac-TEA. Pantalla Splash

5.3.1.3. Lógica

La lógica de esta pantalla se encuentra en la clase *SplashActivity.java*. Esta clase se encarga de obtener el *ImageView* y asignarle la animación creada. El listener del objeto *Animation* nos permitirá gestionar las acciones relacionadas con la animación. Gracias a él, conseguimos el efecto de tener dos pantallas distintas cuando sólo usamos una. Para lograr esto, asignamos un listener a la primera animación y en su método *onAnimationEnd* cambiamos el contenido del *ImageView* y lanzamos la segunda animación. Cuando la segunda termina, se lanza la siguiente actividad.

Esta clase también se ocupa de la inicialización de la BBDD, que se dará siempre que la variable de *BBDD_INI_KEY* de las Preferencias tenga su valor por defecto.

5.3.2. Pantalla PIN de seguridad

5.3.2.1. Descripción

Esta es la pantalla de seguridad de la aplicación, requerida por el usuario final. Siempre que se defina un PIN de seguridad, esta pantalla saldrá al iniciarse la aplicación y solicitará un PIN para poder acceder.

5.3.2.2. Interfaz

El fichero relativo a la interfaz es el de *password.xml*. En él se muestra una cabecera con un icono de candado, un *EditTextView* y un botón de aceptar en la toolbar.

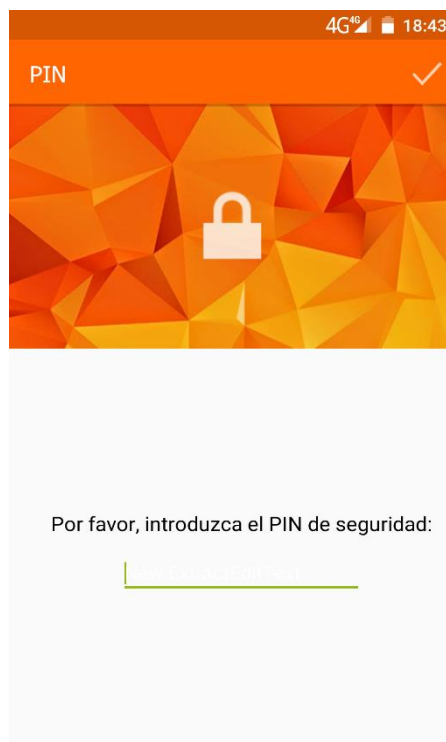


Figura 18. Tic-Tac-TEA. Pantalla PIN de seguridad

5.3.2.3. Lógica

La lógica de esta pantalla se encuentra en la clase *PasswordActivity.java*. Esta actividad se ejecuta tras la pantalla de Splash si el usuario tiene activado un PIN de seguridad. Por ello, en la lógica de la actividad simplemente se recupera el PIN guardado en las preferencias y se compara con la entrada que haga el usuario. En caso de error se mostrará un mensaje informativo y se permanecerá en la misma pantalla.

5.3.3. Pantalla Home y menú lateral

5.3.3.1. Descripción

Ésta es la pantalla principal de la aplicación. En ella tenemos la navegación a las dos opciones más destacadas del usuario: Estrategias y Regulaciones. También tendremos un menú lateral con esas opciones de navegación y con el acceso al perfil del usuario, a la pantalla de configuración y al tutorial.

5.3.3.2. Interfaz

La interfaz de esta pantalla está definida en *main_activity.xml*. En este fichero usamos varias componentes nuevas del Material design. El contenido del fichero es:

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.v4.widget.DrawerLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    tools:openDrawer="start">

    <!-- Contenido principal -->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <!-- Incluimos toolbar -->
        <include
            layout="@layout/app_bar"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" />

        <!-- Layout de fragmentos -->
        <FrameLayout
            android:id="@+id/fragment_menu"
            android:layout_width="match_parent"
            android:layout_height="match_parent">
            </FrameLayout>
    </LinearLayout>
```

```

<!-- Menu drawer -->
<android.support.design.widget.NavigationView
    android:id="@+id/nav_view"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_gravity="start"
    android:fitsSystemWindows="true"
    app:headerLayout="@layout/nav_header_menu"
    app:menu="@menu/main_activity_drawer" />

</android.support.v4.widget.DrawerLayout>

```

A continuación, detallamos los componentes utilizados y sus características:

- Usamos un `DrawerLayout`, necesario para poder tener el elemento del menú lateral. Dividimos el contenido entre un layout principal y el `NavigationView`, donde se carga el menú.
- En el contenido principal estamos incluyendo el layout `AppBarLayout`. En este fichero tenemos un `AppBarLayout` en el que hemos definido una `Toolbar`. Ésta es la barra superior que saldrá en casi todas las pantallas de la aplicación y en la que podremos poner títulos y botones.
- Bajo la toolbar tenemos un `FrameLayout`. Este layout nos permite trabajar con distintos fragmentos y lo usaremos para cargar los correspondientes a Home, Estrategias y Regulaciones. De este modo, estos tres fragmentos tendrán acceso al menú lateral que hemos definido.
- El menú lateral utiliza otros dos ficheros de interfaz: el layout `nav_header_menu.xml` en el que se definen los elementos de la cabecera del menú, y `main_activity_drawer.xml` que contiene la lista de elementos del menú, agrupados por categoría.
- La cabecera del menú lateral mostrará la foto del usuario, su nombre y el nombre de la aplicación. En caso de no tener información del usuario, se mostrarán valores por defecto.

Como hemos comentado, esta actividad principal hará uso de tres fragmentos. El que primero se mostrará será el de Home, cuya interfaz está definida en el fichero `home_fragment.xml`. En este fichero tendremos un fondo de smartwatch y dos botones para el acceso a Estrategias y Regulaciones. En la toolbar de este fragmento se mostrará el nombre de la aplicación.

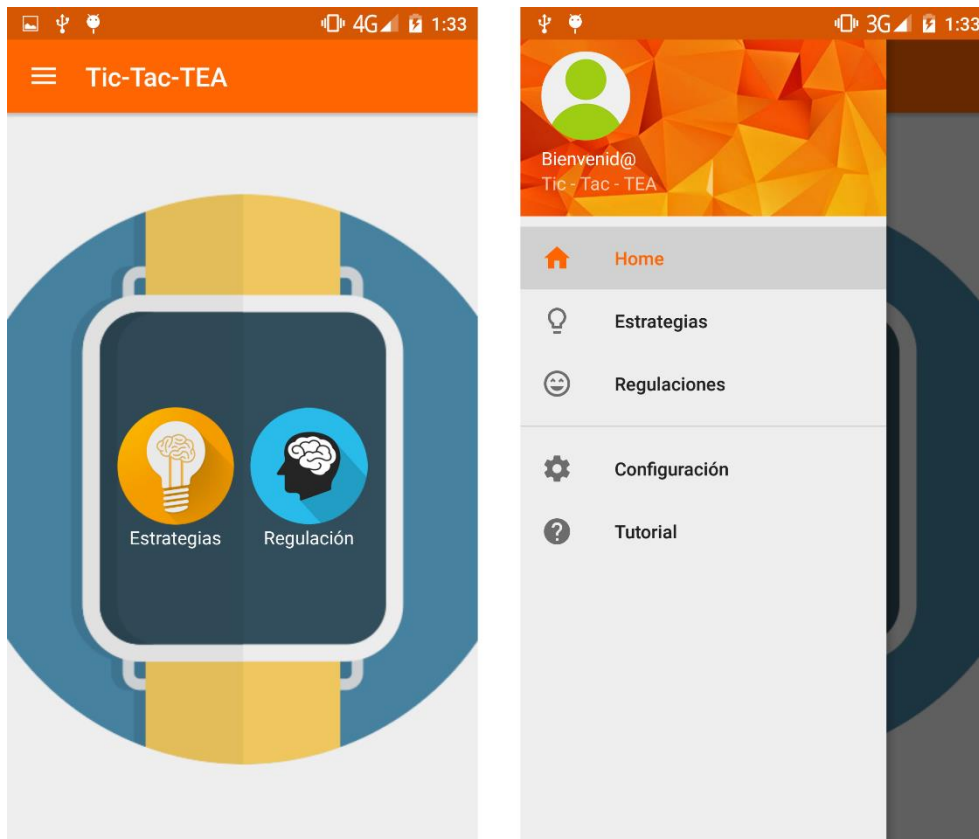


Figura 19. Tic-Tac-TEA. Pantalla Home y menú lateral

5.3.3.3. Lógica

Las distintas funcionalidades que ofrece esta actividad se encuentran definidas en la clase *MainActivity.java* y son las siguientes:

- Inicialización del menú lateral. Se carga en la cabecera del menú la información del nombre y foto del usuario en caso de tenerla registrada. En caso contrario, se muestra una imagen por defecto y un mensaje de bienvenida donde iría el nombre.
- Carga de las opciones del menú y definición de sus métodos *onClick* para la navegación a otros fragmentos o a otras actividades.
- Gestión de los fragmentos a mostrar en la actividad. Se tiene un *FragmentManager* para esta gestión según la navegación a los fragmentos a través de los botones o usando el menú lateral.

Las funcionalidades relativas al fragmento Home se encuentran definidas en la clase *HomeFragment.java*. Esta clase únicamente se encarga de la gestión de los dos botones de navegación, avisando a la clase Main sobre a qué fragmento navegar.

5.3.4. Pantalla de Perfil de usuario

5.3.4.1. Descripción

El sistema consta de una pantalla de visualización de los datos del usuario, a la que se accede mediante la cabecera del menú lateral. En esta pantalla, el usuario ve sus datos personales registrados o los valores por defecto si aún no ha editado los campos.

5.3.4.2. Interfaz

El fichero de interfaz de esta sección es el de *profile_activity.xml*. En este fichero usamos varios Linearlayouts, a los que les hemos asignado pesos para mantener las proporciones de la posición de los elementos y no tener que depender, de este modo, de los distintos tamaños de las pantallas existentes. Esta técnica la aplicaremos al resto de interfaces de la herramienta. Los componentes a destacar de la interfaz son:

- Botones de back y editar. Debajo, una cabecera con un CircleImageView para mostrar la foto del usuario en un layout redondo, y el nombre del mismo.
- Un separador con la información del número de estrategias y de regulaciones creadas.
- TextViews para el género y el tipo del reloj del usuario. Además, estos textos irán acompañados de un ImageView representativo.

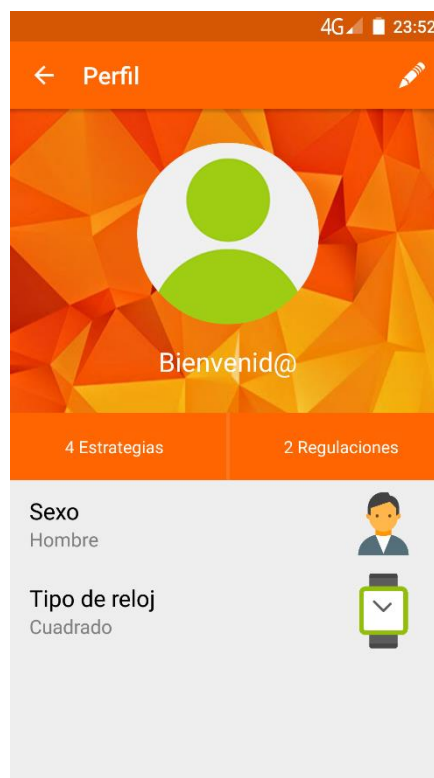


Figura 20. Tic-Tac-TEA. Pantalla de Perfil

5.3.4.3. Lógica

La clase *ProfileActivity.java* contiene la lógica de esta parte del sistema. Los métodos y funcionalidades que tiene son:

- **onCreate:** En esta función, propia de la Activity de Android, recuperamos las vistas a las que necesitamos acceder. Luego, llamamos a los métodos inicializadores de los datos.
- **Carga de datos de usuario:** Con el método *loadUserData* se recuperarán los datos del usuario con una llamada a la BBDD, en caso de tener inicializado el id del usuario en las preferencias de la aplicación. En caso contrario se cargará una imagen por defecto y se mostrará un mensaje de bienvenida.
- **Número de estrategias y regulaciones:** *setStrategiesAndRegulationNumber* será el método que, en caso de tener registrado un usuario, recupere el número de estrategias y regulaciones de la BBDD.
- **Volver:** Si se quiere volver a la pantalla en que se estuviera, sólo habrá que tocar el botón de back. La actividad Profile se ha definido como hija de la Main, por lo que la navegación hacia atrás es automática. Lo que no es automático es el control sobre a qué fragmento volver, pero esa funcionalidad de control ha sido definida en el Main.
- **Menú de opciones de la toolbar:** Se define un botón para editar en la toolbar, que nos llevará a otra actividad.

5.3.5. Pantalla de Editar perfil

5.3.5.1. Descripción

En esta pantalla mostraremos las opciones de edición de datos del usuario, incluyendo la modificación de su foto de perfil.

5.3.5.2. Interfaz

El estilo de la interfaz se ha conseguido utilizando la técnica de asignación de pesos comentada anteriormente. El fichero de interfaz *edit_profile_activity.xml* contiene las siguiente cuatro partes diferenciadas:

- La toolbar con la opción de volver, el título de la pantalla y un botón de guardar.

- Una cabecera con un CircleImageView mostrando la foto del usuario y debajo una opción para editar esa foto.
- Un separador con la información estática del número de estrategias y regulaciones.
- Un EditText para cambiar el nombre de usuario y checkboxes para seleccionar el género y el tipo de reloj.

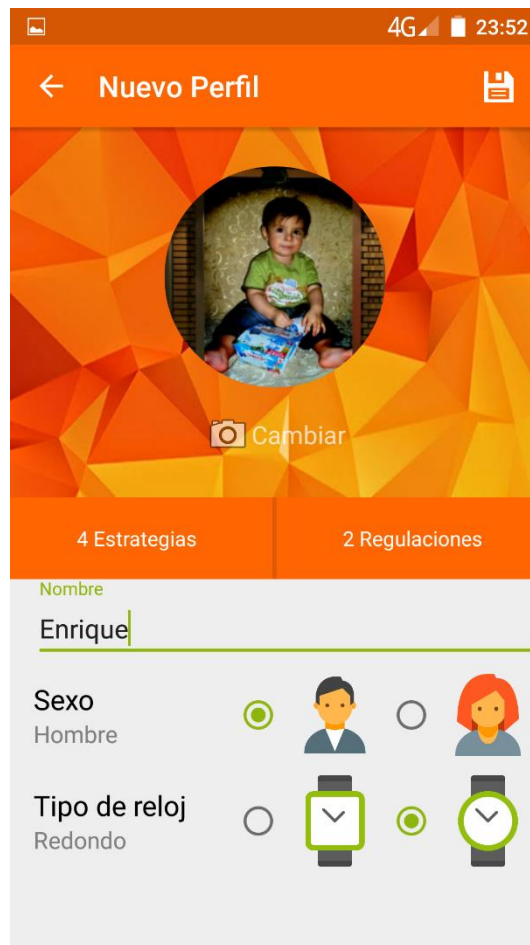


Figura 21. Tic-Tac-TEA. Pantalla de Editar perfil

5.3.5.3. Lógica

En la clase NewAccount.java se tiene la lógica de esta parte del sistema. Vemos:

- **onCreate:** En este método recuperamos las vistas a las que necesitamos acceder. Además, determinamos los *listeners* necesarios para los botones y llamamos a otros métodos de inicialización de datos.

- **Inicialización de datos:** Al igual que en la pantalla anterior, en caso de existir un usuario, se cargan sus datos tanto personales como los relativos al número de estrategias y regulaciones que tiene.
- **Selección de género:** Mediante RadioButtons el usuario podrá elegir su sexo. Al seleccionar alguna opción se llamará al método *onCheckedChanged* que encargará de realizar el cambio.
- **Selección de tipo de reloj:** Mediante RadioButtons se podrá elegir entre un smartwatch de pantalla cuadrada o redonda. Esta información es útil para el momento en que se vayan a crear tarjetas de estrategias, ya que la vista de ellas se adaptará al tipo de reloj seleccionado.
- **Añadir foto de perfil:** Si presionamos el botón con forma de cámara o el texto de “Cambiar” se mostrará un diálogo con tres opciones: Cámara, Galería y Borrar imagen. Al escoger alguna de las dos primeras, se llamará al método *addPhoto* que recibirá el tipo de opción seleccionada. Según la opción solicitada creará un Intent con una acción o bien de tipo *MediaStore.ACTION_IMAGE_CAPTURE* o bien de tipo *Intent.ACTION_PICK*. En ambos casos se iniciará la actividad requerida con el método *startActivityForResult*. Por último, en *onActivityResult* se recuperará la imagen cargada, se llamará al método de crop y después se asignará el resultado al *ImageView* de nuestra vista.

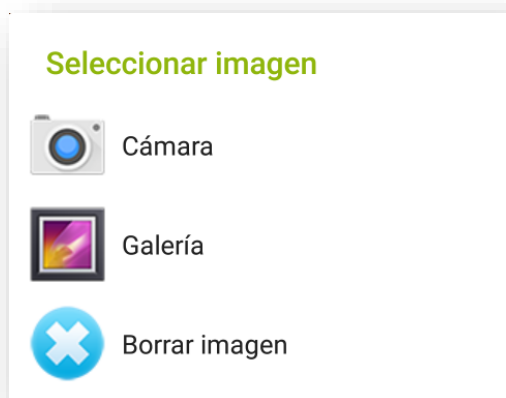


Figura 22. Tic-Tac-Tea. Diálogo con opciones de edición de foto

- **Eliminar foto:** En el caso anterior, si usuario selecciona la tercera opción de Borrar imagen, ésta desaparecerá del *ImageView*. En su lugar se colocará la imagen por defecto.

- **Crop de foto:** Se ha añadido una opción extra a la edición de la foto del usuario. Hemos hecho uso de la librería *android-cropimage* para permitir seleccionar una parte de la imagen a cargar y escalarla. De este modo, siempre se cargarán fotos cuadradas de una proporción fija.
- **Guardar usuario:** Al pulsar el botón de guardar de la toolbar, la aplicación realizará una serie de acciones. Primero comprobará que todos los campos están rellenos correctamente y, de ser así, empezará el registro de datos. Antes de guardar al usuario, guardará primero su foto (si hay alguna cargada) utilizando el método *guardarFoto*. Esta función se encarga de guardar la imagen en una carpeta privada de la aplicación que hemos creado. Una vez guardada la foto del usuario, se procede al registro del mismo en la BBDD.
- **Actualizar preferencias:** Tras guardar la información del usuario en BBDD, se registra su información en las preferencias. De este modo, se tendrá el nombre, la foto y el tipo de reloj para ser usados en otras pantallas sin necesidad de hacer nuevas queries a BBDD. El proceso finaliza volviendo a la pantalla anterior de Perfil.
- **Cancelar:** Si se cancela la operación se finaliza la actividad y se vuelve a la pantalla del Perfil, que está definida como padre de Editar perfil.

5.3.6. Pantalla de Configuración

5.3.6.1. Descripción

Esta pantalla ofrece las opciones especiales de configuración que se le da al usuario. Tiene dos bloques diferenciados: uno para el backup y otro para la seguridad. Las opciones que se ofrecen son: Crear copia de seguridad, restaurar copia de seguridad, exportar resultados y protección por PIN.

5.3.6.2. Interfaz

En este caso, el fichero de interfaz no se encuentra en la carpeta de *layout* sino en *xml*, con nombre *settings.xml*. Esto se debe a que está definido como un *PreferenceScreen*. En su interior tendremos elementos de tipo *PreferenceCategory* (para crear los grupos) y *Preference* (para cada elemento). De este modo, el primer grupo de configuración queda definido de la siguiente forma:

```
<PreferenceCategory
    android:title="@string/backup"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
```



```

<Preference
    android:layout_height="wrap_content"
    android:layout_width="wrap_content"
    android:key="@string/backup"
    android:title="@string/create_backup"
    android:defaultValue="" />

<Preference
    android:layout_height="wrap_content"
    android:layout_width="wrap_content"
    android:key="@string/restore"
    android:title="@string/restore_backup"
    android:defaultValue="" />

<Preference
    android:layout_height="wrap_content"
    android:layout_width="wrap_content"
    android:key="@string/export"
    android:title="@string/export_logs"
    android:defaultValue="" />
</PreferenceCategory>

```

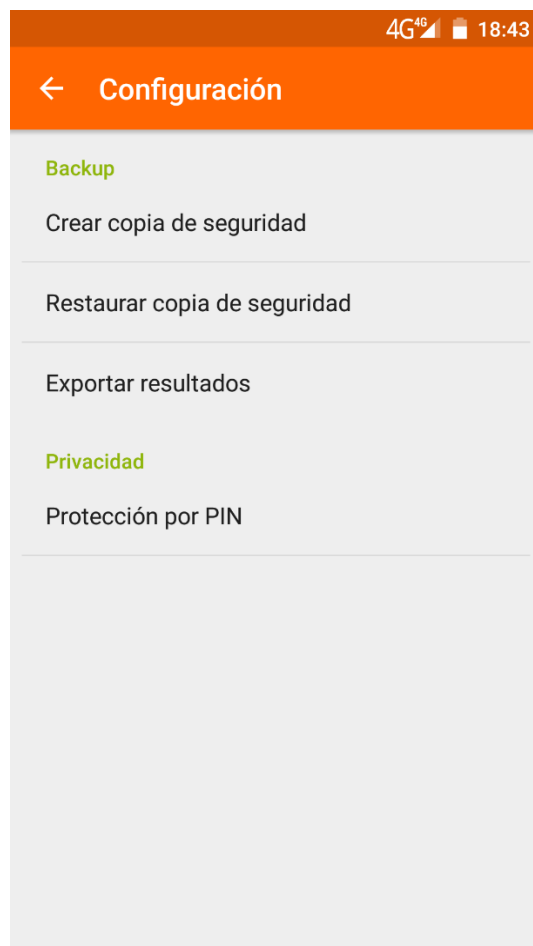


Figura 23. Tic-Tac-TEA. Pantalla de Configuración

5.3.6.3. Lógica

En *SettingsActivity.java* encontramos la lógica de este apartado. Vemos que esta clase extiende de otra especial que es *PreferenceActivity* y que define los siguientes métodos:

- **onCreate:** En esta función inicializamos las preferencias y les asignamos el listener de *setOnPreferenceClickListener*.
- **Backup de aplicación:** Tendremos los métodos *createBackup* y *restoreBackup*, que serán los encargados de crear la copia de seguridad de la aplicación y de poder restaurarla posteriormente. Se guarda tanto el estado de la BBDD como el contenido de la carpeta privada en la que se almacenan las imágenes subidas por el usuario. Además, se mantienen los valores de las preferencias.
- **Exportar logs:** El método *exportUserLogs* se encarga de hacer una petición al smartwatch para el envío de los logs que el reloj tenga registrado.
- **PIN de seguridad:** Mostraremos un dialog con un EditText para que el usuario pueda escribir su PIN de seguridad. Si ya hay alguno establecido, éste se mostrará en el componente editable. Si se elimina su valor, se estará desactivando esta opción de seguridad. Las preferencias de la aplicación tendrán un valor para saber cuándo está activada esta seguridad o no.

5.3.7. Pantalla de Estrategias

5.3.7.1. Descripción

Esta pantalla muestra la lista de estrategias creadas por el usuario. Sirve como un resumen de ellas ya que muestra la información más importante como el nombre, el icono y el número de tarjetas que contiene cada una. Es un fragmento del *MainActivity*.

5.3.7.2. Interfaz

Al tratarse de un fragmento, su interfaz se encuentra en el fichero *strategy_fragment.xml* y su contenido será:

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
```

```

        android:textAlignment="center">

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <!-- Listas con cardviews de las estrategias -->
    <android.support.v7.widget.RecyclerView
        android:id="@+id/strategies_recycler"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>

    <!-- Texto a mostrar cuando no hay estrategias -->
    <TextView
        android:id="@+id/no_strategies"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:text="@string/none_strategies"
        android:visibility="visible"
        android:gravity="center"/>
</LinearLayout>

<android.support.design.widget.FloatingActionButton
    android:id="@+id/btn_add_strategy"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="bottom|end"
    android:layout_margin="@dimen/fab_margin"
    android:src="@drawable/ic_action_add" />

</android.support.design.widget.CoordinatorLayout>

```

Los detalles a destacar de la interfaz son:

- Usamos un `CoordinatorLayout`, que nos permite establecer un contenido principal y hacer uso de un botón flotante en la parte inferior derecha para crear una nueva estrategia.
- El contenido principal está formado por un `LinearLayout` que contiene un texto a mostrar para cuando no hay estrategias creadas, y un `RecyclerView` que mostrará una lista de *cards* con la información resumida de cada estrategia.
- El contenido del `RecyclerView` está definido en otra clase, en `StrategyViewAdapter`. El layout de esta clase se encuentra en `strategy_cardview.xml` y contiene los elementos a mostrar de la estrategia como el `CircleImageView` para su icono, dos `TextViews` para el nombre y número de tarjetas, y un botón para acceder a la edición de esa estrategia.

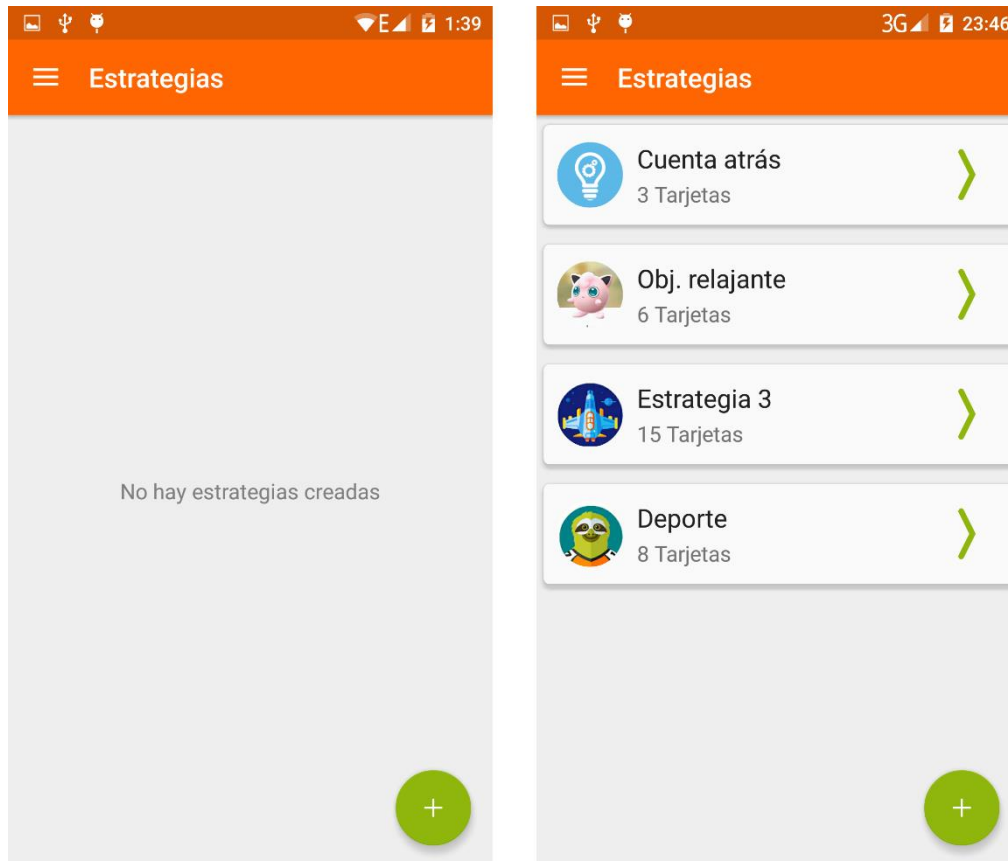


Figura 24. Tic-Tac-TEA. Pantalla de Estrategias

5.3.7.3. Lógica

La clase *StrategyFragment.java* se encarga de recuperar los elementos de la interfaz y de establecer el *listener* para la acción del botón flotante. Además, llama al método para inicializar la lista de estrategias.

En caso de que el usuario no tenga estrategias registradas, se inicializará el valor del *TextView* y se activará su visibilidad. Por el contrario, si existen estrategias se inicializará el adaptador del *RecyclerView* con la lista de estrategias (el resumen) obtenido de BBDD y se asignará este adaptador a la vista.

La lógica del adaptador está definida en *StrategyViewAdapter.java*. Esta clase crea un elemento *card* por cada entrada de la lista de estrategias que reciba. Cada entrada de la lista es de tipo *ListaEstrategia.java*, por lo que contiene el identificador de BBDD, el nombre, el icono y el número de tarjetas de la estrategia.

Además, el adaptador se encarga de la gestión de los iconos de la estrategia, es decir, dependiendo del valor llamará a *PhotoUtils* para obtener el icono personalizado guardado en la carpeta privada, o mostrará el icono por defecto de la aplicación.

5.3.8. Pantalla de Nueva estrategia

5.3.8.1. Descripción

Esta pantalla es una de las más importantes de la aplicación y contiene la lógica principal. Dado que usa mucha información y un número elevado de componentes y fragmentos, vamos a dividir su explicación por bloques. En este primer bloque explicaremos la primera vista que obtendremos nada más acceder a la actividad.

5.3.8.2. Interfaz

El fichero relativo a la interfaz es el de *new_strategy_activity.xml*. En él podremos ver el siguiente contenido:

```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
android:fitsSystemWindows="true">

<!-- Incluimos toolbar -->
<include
    layout="@layout/app_bar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />

<!-- Pestañas de configuración -->
<android.support.design.widget.TabLayout
    android:id="@+id/strategy_tabs"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="?attr/colorPrimary"
    app:tabIndicatorColor="@color/tabIndicatorColor"
    android:minHeight="?attr/actionBarSize"
    android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar"
    app:tabMode="fixed"
    app:tabGravity="fill"/>

<!-- Contenido de fragmentos -->
<android.support.v4.view.ViewPager
    android:id="@+id/new_strategy_viewpager"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"/>

</LinearLayout>
```

Podemos ver que los principales elementos que componen la interfaz son: La toolbar, un TabLayout en el que se definirán las 2 pestañas de la configuración de estrategias (Tarjetas y Configuración) y, por último, un ViewPager con el contenido de cada fragmento correspondiente a las pestañas. El primer fragmento que se cargará será el de "Tarjetas" y mostrará, como primer paso, un popup para registrar el nombre de la

estrategia. Este popup será personalizado y su fichero de interfaz vendrá definido en *popup_strategy.xml*. Si se cancela este popup, se mostrará un nombre de estrategia por defecto. El nombre de la estrategia será como máximo de 16 caracteres, adaptados a dos líneas de 8 caracteres en la pantalla del smartwatch.

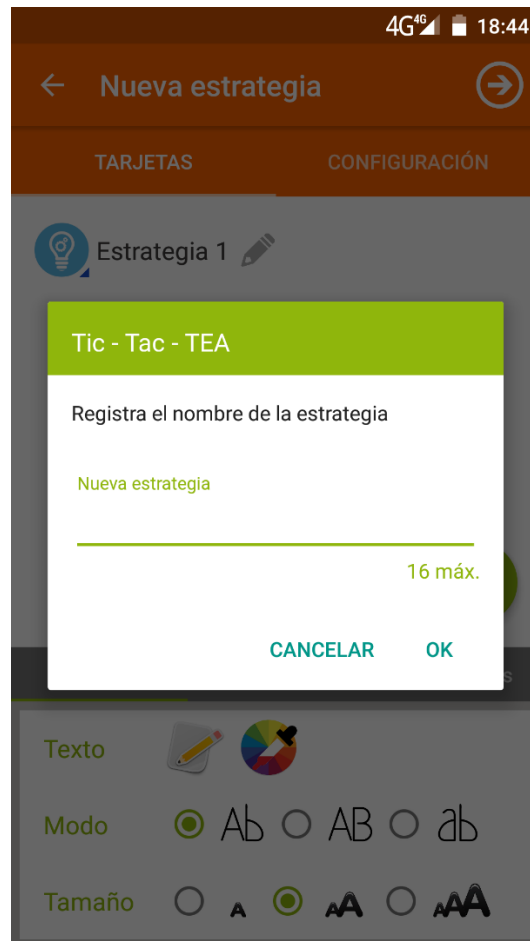


Figura 25. Tic-Tac-TEA. Pantalla de Nueva estrategia

5.3.8.3. Lógica

La lógica de esta pantalla se encuentra en la clase *NewStrategyActivity.java*. Esta clase se encarga, principalmente de la carga y gestión de los fragmentos definidos en cada pestaña. El fragmento que se mostrará por defecto será el primero y su lógica se encuentra en la clase *StrategyFirstTabFragment.java*.

En el primer fragmento tendremos la información del nombre e icono de la estrategia, la edición de las tarjetas y un submenú con las opciones del contenido de las tarjetas. Para cambiar la información de la estrategia contamos con dos métodos:

- **editStrategyDialog**: Este método crea un popup que se mostrará al crearse la actividad y cuyo objetivo es editar el nombre de la estrategia. También se

mostrará si se toca el icono de 'lápiz' visible al lado derecho del nombre de la estrategia.

- **dialogPhoto:** Se trata de otro popup, en este caso para la edición del icono de la estrategia. Se podrá elegir un recurso desde la cámara, galería o uno predeterminado.

5.3.9. Pantalla de Nueva estrategia. Tarjeta de texto

5.3.9.1. Descripción

Continuamos la explicación de la pantalla anterior y de la primera pestaña de Tarjetas. En esta pestaña la primera opción del submenú que veremos será la de Texto, que nos permitirá incluir tarjetas con textos personalizados. Encima de este submenú, veremos la vista de cómo quedaría la tarjeta, englobada en un marco cuadrado o redondo, según el tipo de reloj que haya seleccionado el usuario.

5.3.9.2. Interfaz

El fichero relativo a la interfaz general es el de *strategy_first_tab.xml* y el relativo a la parte de edición de texto es el de *submenú_first_tab_fragment.xml*. De estas vistas destacamos:

- En la vista general de la primera pestaña, usaremos un elemento ViewPager que será el que nos permitirá hacer la gestión de las tarjetas, ya sea mostrando el contenido, navegando entre las ya creadas o añadiendo nuevas.
- El ViewPager tendrá superpuesto un ImageView que actuará como marco de la pantalla del smartwatch, redondo o cuadrado.
- Para facilitar la gestión, por parte del usuario, del ViewPager se mostrará un rótulo con la posición actual de la tarjeta que se esté editando y el número total de tarjetas creadas. Además, habrá flechas de navegación para moverse entre ellas, aunque también servirá hacer el gesto swipe en la pantalla.
- En el submenú de texto, las opciones estarán divididas en tres partes: El primer grupo para modificar el texto y su color, el segundo para modificar el modo de escritura (normal, mayúsculas y minúsculas) y el tercero para modificar el tamaño (normal, mediano, grande).



Figura 26. Tic-Tac-TEA. Pantalla de Nueva estrategia. Tarjeta de texto

5.3.9.3. Lógica

La lógica de la primera pestaña se encuentra en la clase *StrategyFirstTabFragment.java*, mientras que en *SubmenuFirstTabFragment.java* tenemos la del submenú de texto.

Empezaremos explicando la parte de gestión de tarjetas usando el ViewPager. Vemos:

- En primer lugar, debemos crear y asignar un adaptador al ViewPager. Para ello hemos creado la clase *CardPageAdapter.java*. Este adaptador facilitará la gestión de las tarjetas, ocupándose de cargar la información a mostrar, añadiendo contenido, eliminándolo y reemplazándolo.
- El adaptador *CardPageAdapter* mantendrá una lista de elementos de tipo *Recurso.java*. La clase de tipo recurso contiene información necesaria para manejar tanto elementos de texto como imágenes.

- El adaptador utilizará el fichero de interfaz *card_pager_item.xml* para mostrar el contenido de las tarjetas. Esta interfaz contiene tanto un *ImageView* como un *TextView*, para usar según el tipo de recurso requerido.
- Para la carga de imágenes se utilizará la librería *Glide*. Esta librería será la que nos permita mostrar imágenes de tipo gif.
- Se inicializará el *ViewPager* a través de su adaptador, añadiendo una vista vacía.
- Se definirán como métodos de comunicación entre el componente *ViewPager* y su adaptador los siguientes: *addView*, *replaceView*, *moveToNextPage*, *moveToPreviousPage*. Con estos métodos también se controlará la posición actual de una tarjeta respecto al total.
- La clase *StrategyFirstTabFragment* gestionará la visibilidad de las flechas de navegación entre tarjetas con el método *setArrowsVisibility*. Cuando sólo se tenga una tarjeta no se mostrarán las flechas, cuando se tenga más de una y se esté en la primera no se mostrará la flecha de la izquierda y cuando se esté en la última tarjeta no se mostrará la flecha de la derecha.

Tras ver los principales métodos de gestión del *ViewPager*, continuamos explicando la segunda componente principal de esta primera pestaña. Se trata de un *TabLayout* junto a otro *ViewPager*. Estos componentes conforman el submenú de opciones de contenido de la estrategia. El primero de ellos, el submenú de texto, tiene en su lógica los siguientes métodos:

- **onCreateView:** Obtiene los elementos de la interfaz e inicializa los valores por defecto.
- **editTextDialog:** Muestra un popup en el que poder añadir o modificar el texto a mostrar en la tarjeta.
- **colorPickerDialog:** Abre un diálogo en el que poder seleccionar el color del texto a utilizar. Usa la librería *color-picker*.
- **onCheckedChanged:** Cambia la configuración del texto según la opción elegida en los radiobuttons.
- **addNewText:** Crea un objeto *Recurso* con el texto personalizado y llamado al método *addView* del fragmento padre para añadir la tarjeta al *ViewPager*.

5.3.10. Pantalla de Nueva estrategia. Tarjeta de pictogramas

5.3.10.1. Descripción

Continuamos en la primera pestaña de edición de tarjetas y vemos la segunda opción del submenú de opciones. Esta segunda pestaña del submenú mostrará una matriz de imágenes predeterminadas, que serán los pictogramas de ARASAAC.

5.3.10.2. Interfaz

Dado que ya hemos visto el fichero de interfaz del primer fragmento de estrategia, nos vamos a centrar sólo en las interfaces del submenú. El fichero relativo a la interfaz de la segunda pestaña del submenú es el de *submenú_second_tab_fragment.xml*. En él tendremos el siguiente contenido:

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.v7.widget.RecyclerView
xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/lst_icon_thumbnails"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/submenuBackground"
    android:scrollbars="vertical" />
```

En este caso sólo tenemos un elemento de tipo RecyclerView en el que cargaremos la listas de los thumbnails de los pictogramas a utilizar en las tarjetas.



Figura 27. Tic-Tac-TEA. Pantalla de Nueva estrategia. Tarjeta con pictograma

5.3.10.3. Lógica

La lógica de esta pantalla se encuentra en la clase *SubmenuSecondTabFragment.java*. La funcionalidad básica de esta clase consiste en:

- Recuperar el elemento RecyclerView y asignarle un layout de cuadrícula con cinco elementos por línea.
- Inicializar la lista de pictogramas a mostrar.
- Creamos una clase adaptadora para el RecyclerView y le pasamos la lista de pictogramas a mostrar. Esta clase será la de *IconsAdapter.java*.
- La clase adaptadora usará de layout *icons_thumbnail.xml*, que será el ImageView con características específicas donde mostrar los iconos de cada pictograma.
- IconsAdapter utilizará la librería Glide para cargar la miniatura de los pictogramas en la rejilla del RecyclerView.
- Se asigna el listener *addOnItemTouchListener* al RecyclerView para que cada vez que se seleccione un pictograma se cargue en la tarjeta mediante el método *replaceView* del fragmento padre.

5.3.11. Pantalla de Nueva estrategia. Tarjeta de recursos propios

5.3.11.1. Descripción

Ésta es la última opción del submenú de contenido y servirá para que el usuario pueda incluir en las tarjetas recursos propios que ya haya agregado o que puede agregar en el mismo momento. Se mostrará las miniaturas de estos recursos en una cuadrícula al igual que en el contenido del submenú anterior.

5.3.11.2. Interfaz

El fichero relativo a la interfaz de esta tercera pestaña del submenú es el de *submenú_third_tab_fragment.xml*. Su contenido es muy parecido al del submenú anterior, con la diferencia de que se agrega un botón que permitirá añadir un recurso nuevo.



Figura 28. Tic-Tac-TEA. Pantalla de Nueva estrategia. Tarjeta con recurso propio

5.3.11.3. Lógica

La lógica de esta parte de la aplicación es muy parecida a la anterior y se encuentra en el fichero *SubmenuThirdTabFragment.java*. Al igual que en el caso anterior, usaremos un *RecyclerView* al que le asignaremos un *IconsAdapter*. A esta clase adaptadora se le pasará una lista con las imágenes propias del usuario. Los elementos de esta lista son de tipo *Recurso*, por lo que la clase adaptadora puede diferenciar cuándo está trabajando con imágenes predeterminadas y cuándo son imágenes propias.

Como principal diferenciación a la sub-pestaña anterior, se tienen métodos relativos a la carga de imágenes. Como hemos visto en otras pantallas, mostraremos un popup para la carga desde la cámara o la galería de imágenes. Igualmente, se utilizará la librería de *crop-image* para editar la imagen antes de agregarla al *RecyclerView*. Por último, se almacenará el recurso nuevo en la carpeta privada de la aplicación y en la BBDD, indicando su ruta.

5.3.12. Pantalla de Nueva Estrategia. Configuración

5.3.12.1. Descripción

Esta pantalla corresponde a la segunda pestaña de la creación de una nueva estrategia. En ella podremos elegir las opciones de configuración de la estrategia tales como el tipo de transición de las tarjetas, si mostrar temporizador, refuerzo, etc.

5.3.12.2. Interfaz

La interfaz se encuentra en *strategy_second_tab.xml*. Aquí se muestran todas las opciones de configuración. Por defecto, se activa la transición de tarjetas por tacto y se muestran las opciones relativas al tiempo desactivadas. En caso de cambiar la transición a tiempo, las opciones relativas se activarían.

Para las opciones booleanas se ha utilizado el elemento switch y para las múltiples se hará uso de un popup en el que el usuario pueda elegir la opción que desee.

En las opciones de Comprobación emocional y Refuerzo habrá un icono de información que mostrará en un popup el uso de esas opciones, ya que pueden resultar poco intuitivas para usuarios que no sean profesionales en TEA.

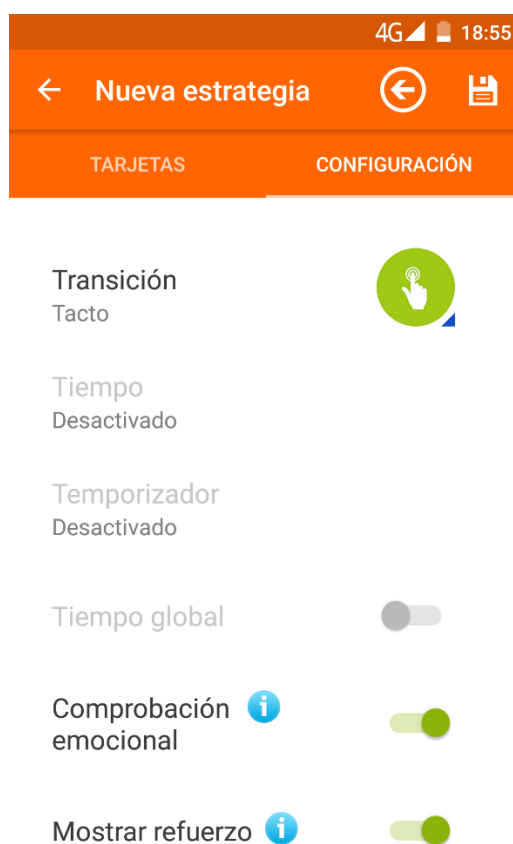


Figura 29. Tic-Tac-TEA. Pantalla de Nueva estrategia. Configuración

5.3.12.3. Lógica

La lógica de esta pantalla se encuentra en la clase *StrategySecondTabFragment.java*. La funcionalidad clave que desarrolla es:

- Recuperar los elementos de interfaz, asignarle listeners e inicializar los valores por defecto.
- Desarrollar el método *disableTimeOptions* para deshabilitar las opciones de tiempo cuando la transición sea por tacto, y *enableTimeOptions* para activarlas.
- Crear un popup de selección de opciones que se mostrará cuando el usuario decida cambiar la forma de transición o cuando quiera cambiar el tipo de temporizador a mostrar.
- Para la opción de tiempo de estrategia se ha desarrollado un *numberPicker*, que se mostrará en un popup personalizado. Esta opción permitirá incluir valores numéricos de segundos mayores a 1.
- Por último, en la toolbar tenemos el botón de guardar que, una vez que haya comprobado que todos los datos de la estrategia son correctos (de ambas pestañas) procederá a guardarla en BBDD. Se guardan tanto las tarjetas como todas las opciones de configuración en las tablas correspondientes. Tras este proceso se volverá a la pantalla de lista de estrategias y se incluirá la nueva creada.

5.3.13. Pantalla de Regulaciones

5.3.13.1. Descripción

Esta pantalla muestra la lista de regulaciones creadas por el usuario. Sirve como un resumen de ellas ya que muestra la información más importante como el nombre, el icono, el número de estrategias que contiene cada una y si es de tipo selector o no. Además, tiene un botón para sincronizar las regulaciones y según su estado se podrá saber si una regulación ya está sincronizada o no. Es un fragmento del *MainActivity*.

5.3.13.2. Interfaz

Al tratarse de un fragmento, su interfaz se encuentra en el fichero *strategy_fragment.xml* y su contenido será:

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout
```

```

xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity"
android:textAlignment="center">

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <!-- Listas con cardviews de las regulaciones -->
    <android.support.v7.widget.RecyclerView
        android:id="@+id/regulations_recycler"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>

    <!-- Texto a mostrar cuando no hay regulaciones -->
    <TextView
        android:id="@+id/no_regulations"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:text="@string/none_regulations"
        android:visibility="visible"
        android:gravity="center"/>

</LinearLayout>

<android.support.design.widget.FloatingActionButton
    android:id="@+id/btn_add_regulation"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="bottom|end"
    android:layout_margin="@dimen/fab_margin"
    android:src="@drawable/ic_action_add" />

</android.support.design.widget.CoordinatorLayout>

```

Esta interfaz sigue la misma base que la pantalla que lista las estrategias. Los detalles a destacar de la interfaz son:

- Volvemos a usar un `CoordinatorLayout` en el que tenemos un contenido principal y un botón flotante para crear una nueva regulación.
- El contenido principal está formado por un `LinearLayout` que contiene un texto a mostrar para cuando no hay regulaciones creadas, y un `RecyclerView` que mostrará una lista de *cards* con la información resumida de cada regulación.
- El contenido del `RecyclerView` está definido en otra clase, en `RegulationViewAdapter`. El layout de esta clase se encuentra en `regulation_cardview.xml` y contiene los elementos a mostrar de la regulación como el `CircleImageView` para su icono, dos `TextViews` para el nombre y

número de estrategias junto a la información del tipo (si es selector), y un botón para sincronizar la regulación.

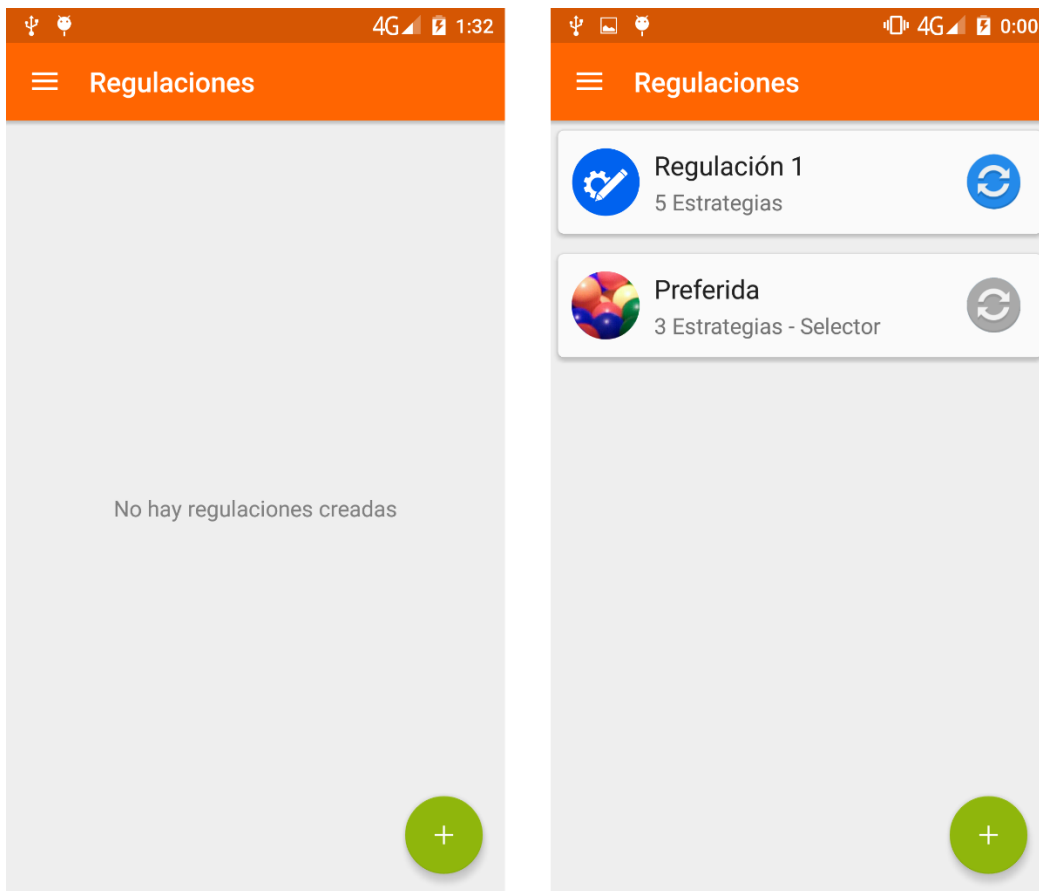


Figura 30. Tic-Tac-TEA. Pantalla de Regulaciones

5.3.13.3. Lógica

La clase *RegulationFragment.java* se encarga de recuperar los elementos de la interfaz y de establecer el *listener* para la acción del botón flotante. Además, llama al método para inicializar la lista de regulaciones.

En caso de que el usuario no tenga regulaciones registradas, se inicializará el valor del *TextView* y se activará su visibilidad. Por el contrario, si existen regulaciones se inicializará el adaptador del *RecyclerView* con la lista de regulaciones (el resumen) obtenido de BBDD y se asignará este adaptador a la vista.

La lógica del adaptador está definida en *RegulationViewAdapter.java*. Esta clase crea un elemento *card* por cada entrada de la lista de regulaciones que reciba. Cada entrada de la lista es de tipo *ListaRegulacion.java*, por lo que contiene el identificador de BBDD, el nombre, el icono, el número estrategias y los flags de si es de tipo selector y si está sincronizado.

Además, el adaptador se encarga de la gestión de los iconos de la estrategia, es decir, dependiendo del valor llamará a *PhotoUtils* para obtener el icono personalizado guardado en la carpeta privada, o mostrará el icono por defecto de la aplicación.

Respecto a las opciones de sincronización, esta clase tiene dos métodos principales: *regulationToJson* y *synchronizeRegulation*. El primer método se encarga de recuperar toda la información de la regulación, junto a sus estrategias, y de convertir esos datos a la estructura JSON esperada por el protocolo de comunicación con el smartwatch. El segundo método prepara la comunicación y realiza la sincronización de la regulación al reloj.

5.3.14. Pantalla de Nueva Regulación

5.3.14.1. Descripción

Esta pantalla se muestra tras tocar el botón de añadir de la pantalla que muestra el listado de regulaciones. Aquí podremos agregar un conjunto de estrategias que formarán una regulación nueva que podremos sincronizar con el smartwatch.

5.3.14.2. Interfaz

El fichero relativo a la interfaz es el de *new_regulation_activity.xml*. En él podemos ver la interfaz estructurada en 3 partes. Primero, se muestra el nombre de la regulación junto a su icono y los botones para editarlos, luego tenemos un RecyclerView para mostrar la lista de estrategias y, por último, tenemos un botón flotante para añadir estrategias a la lista.

Al igual que como pasaba en la pantalla de Nueva estrategia, lo primero que se mostrará al iniciarse esta actividad será un popup para registrar el nombre de la regulación y seleccionar determinadas opciones de configuración. Este popup será personalizado y su fichero de interfaz vendrá definido en *popup_regulation.xml*. Si se cancela este popup, se mostrará un nombre de regulación por defecto. El nombre de la regulación será como máximo de 16 caracteres, adaptados a dos líneas de 8 caracteres en la pantalla del smartwatch.

También se tendrá otro popup con la lista de estrategias, que se mostrará cuando el usuario quiera añadir una nueva. Al agregar una estrategia, ésta será mostrada en una *card*, como las del listado de estrategias, es decir, podremos ver su icono, nombre y número de tarjeta. La principal diferencia es que en este caso cada *card* será editable, podremos cambiar la posición de ellas e incluso eliminarla.

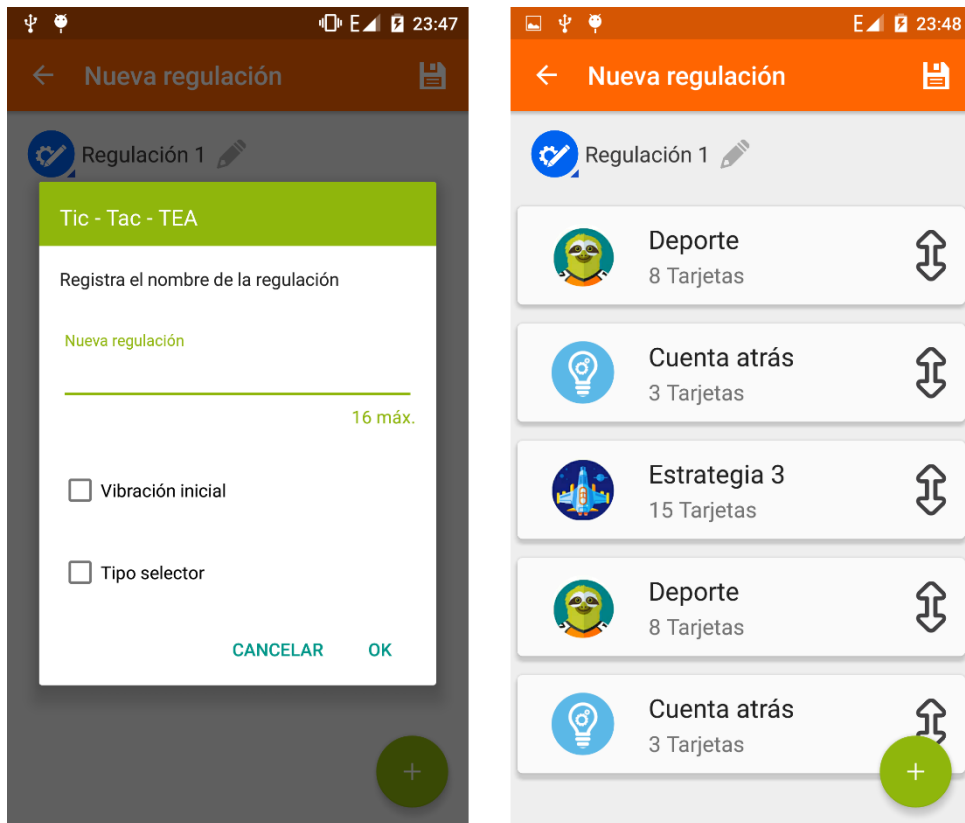


Figura 31. Tic-Tac-TEA. Pantalla de Nueva regulación

5.3.14.3. Lógica

La lógica de esta pantalla se encuentra en la clase *NewRegulationActivity.java*. Esta clase se encarga de recuperar los elementos de la interfaz y de establecer el *listener* para la acción del botón flotante. Además, llama al método para inicializar la lista de estrategias (vacía en caso de regulación nueva o con contenido si se está editando una regulación).

Dado que usaremos un *RecyclerView* para la lista de estrategias, necesitaremos asignarle un adaptador a la vista. El adaptador necesario estará definido en *StrategyViewAdapter.java*. Esta clase crea un elemento *card* por cada entrada de la lista de estrategias que reciba. Cada entrada de la lista es de tipo *ListaEstrategia.java*, por lo que contiene el identificador de BBDD, el nombre, el icono y el número de tarjetas de la estrategia.

Para permitir que el adaptador nos deje manejar la lista de cards con manipulación de gestos, vamos a hacer que su clase implemente la interfaz *ItemTouchHelperAdapter.java*. En esta interfaz hemos declarado los métodos necesarios para mover un elemento mediante *drag* y eliminarlo mediante *swipe*.

Para editar la configuración de la regulación y para añadir estrategias vamos a usar dos métodos que mostrarán popups, de la misma forma que hemos hecho en pantallas

anteriores. El popup de estrategias siempre mostrará un listado completo de todas ellas, con lo que será posible añadir a la regulación la misma estrategia más de una vez.

Una vez que se hayan añadido y editado las estrategias deseadas, el botón de guardar de la toolbar nos permitirá crear la nueva regulación. Si todos los datos de la regulación son correctos (si no está vacía), se procederá a guardarla en BBDD. Se guardan tanto la configuración global de la regulación como todas las estrategias que la componen en las tablas correspondientes (Regulaciones y Schedule). Tras este proceso se volverá a la pantalla de lista de regulaciones y se incluirá la nueva creada.

5.3.15. Pantalla de Tutorial

5.3.15.1. Descripción

Esta pantalla mostrará una serie de imágenes con información de ayuda relativa a la herramienta de autor. Podremos encontrar desde aclaraciones de concepto hasta indicaciones de cómo realizar ciertas actividades.

5.3.15.2. Interfaz

En el fichero *tutorial.xml* tendremos un ViewPager que se encargará de gestionar las distintas imágenes del tutorial. Bajo él, tendremos un indicador de navegación para informar en qué pantalla de ayuda nos encontramos respecto al total.



Figura 32. Tic-Tac-TEA. Pantalla de Tutorial

5.3.15.3. Lógica

La lógica de esta pantalla se encuentra en TutorialActivity.java y se centra en los siguientes métodos:

- **onCreate:** Como en apartados anteriores, aquí inicializamos valores y vistas de la clase. En este caso recuperamos el ViewPager y le asignamos su adaptador con la lista de imágenes a mostrar.
- **moveToNextPage:** Una vez creado el ViewPager con las imágenes del tutorial, usaremos este método para comunicar al adaptador que queremos ver la siguiente imagen, siempre que no estemos ya en la última.
- **moveToPreviousPage:** Equivalente al método anterior, pero para mostrar una imagen previa.

6. Pruebas

Para realizar una prueba completa de la aplicación nos hemos propuesto centrarnos en cuatro aspectos principales a considerar: Validación de requisitos, pruebas de integración, evaluación heurística y encuesta SUS.

Para la validación de requisitos vamos a definir una serie de actividades a realizar por un usuario y mostraremos el resultado que ha tenido la aplicación para ellos. De este modo, podremos comprobar qué requisitos se están cumpliendo y en qué medida.

En las pruebas de integraciones veremos los resultados obtenidos en el smartwatch según las opciones de configuración creadas en la herramienta de autor.

En la evaluación heurística, como expertos en usabilidad, examinaremos y evaluaremos la interfaz de la herramienta. Entre los beneficios que proporciona este método de evaluación se encuentran que da información para realizar mejoras de forma rápida y de poco coste, y que los problemas de usabilidad encontrados son fácilmente demostrables.

Por último, con la encuesta SUS obtendremos una valoración subjetiva de la usabilidad de la aplicación y obtendremos un feedback directo de la satisfacción de los usuarios.

6.1. Validación de requisitos

El objetivo de esta prueba es validar el mayor número de requisitos de la herramienta, que se sumarán a los ya vistos en las pantallas de la sección de Desarrollo y que estarán complementados con los resultados de las pruebas de integración. Se proponen cuatro actividades de validación.

6.1.1. Nueva estrategia con tarjeta de texto

Esta actividad estará formada por los siguientes pasos obligatorios:

- Crear una nueva estrategia.
- Añadir texto personalizado.
- Cambiar el color del texto.
- Probar los tres modos de texto (normal, mayúsculas, minúsculas).
- Probar los tres tipos de tamaño de texto.
- Cambiar el nombre de la estrategia a “Tarjeta Texto”.
- Cambiar el icono de la estrategia.
- Asignar transición de tacto entre tarjetas.
- Guardar la estrategia.

6.1.2. Nueva estrategia con tarjeta con recurso propio

Esta actividad estará formada por los siguientes pasos obligatorios:

- Crear una nueva estrategia llamada “Tarjeta Recurso”.
- Añadir un nuevo recurso propio desde la cámara de fotos.
- Usar ese recurso en una tarjeta.
- Asignar la opción de transición por tiempo (17 segundos) entre tarjetas.
- Asignar un temporizador en la configuración de la estrategia.
- Guardar la estrategia.

6.1.3. Nueva regulación con edición de estrategias

Esta actividad estará formada por los siguientes pasos obligatorios:

- Crear una nueva regulación y mantener el nombre por defecto.
- Añadir las estrategias “Deporte”, “Cuenta atrás” y “Estrategia 3”, en ese orden.
- Repetir las estrategias “Deporte” y “Cuenta atrás”.
- Cambiar el orden de la estrategia “Cuenta atrás” a la tercera posición.
- Eliminar la estrategia “Estrategia 3”.
- Añadir otra estrategia.
- Guardar la regulación.

6.1.4. Cambio de idioma de aplicación

Esta actividad estará formada por los siguientes pasos obligatorios:

- Cambiar el idioma de la aplicación a catalán.
- Navegar al Perfil del usuario.

6.2. Pruebas de integración

Las pruebas de integración con el smartwatch están en su fase inicial. Sin embargo, queremos mostrar el resultado de dos contenidos, de forma que pueda verse cómo se muestra su integración final. Probaremos:

- Adaptación de una tarjeta a la pantalla del reloj.
- Regulación de tipo selector.

6.3. Evaluación heurística

En este apartado comentaremos la prueba de usabilidad realizada, que mide cómo de bien puede un usuario utilizar la aplicación. Para ello nos basaremos en las 10 heurísticas de Nielsen [17]. Estas heurísticas nos proporcionan unas pautas para evaluar el diseño de la interfaz de usuario.

6.4. Encuesta SUS

Nuestra última prueba consiste en una encuesta SUS (*System Usability Scale*), que es una encuesta de 10 preguntas para obtener una métrica subjetiva de la usabilidad del sistema según la hayan percibido los usuarios.

La mayor parte de los usuarios que realizarán esta encuesta serán los profesionales del centro “Quintero Lumbreras”, por lo que los resultados serán especialmente relevantes. Sin embargo, para la realización de la evaluación no se ha podido hacer una demostración final de la interacción con la aplicación y ésta se ha basado en una documentación de cada pantalla de la herramienta, que simulaba todos los recorridos posibles que puede hacer un usuario.

7. Resultados

7.1. Validación de requisitos

7.1.1. Nueva estrategia con tarjeta de texto

Esta actividad ha sido superada y los resultados obtenidos han sido los siguientes:

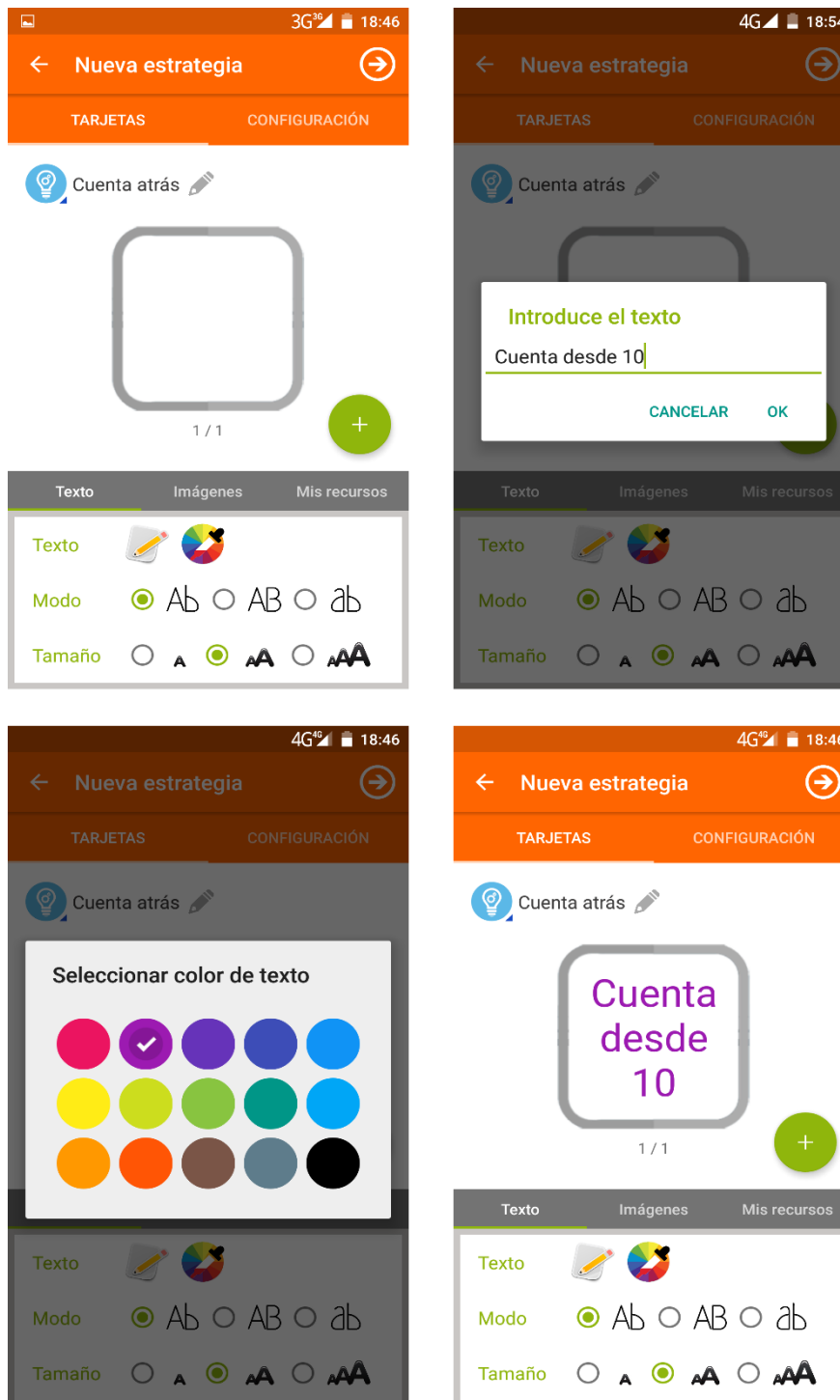


Figura 33. Prueba. Nueva estrategia con tarjeta de texto. Parte I



Figura 34. Prueba. Nueva estrategia con tarjeta de texto. Parte II

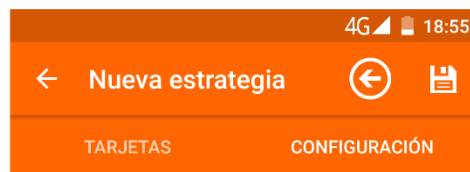
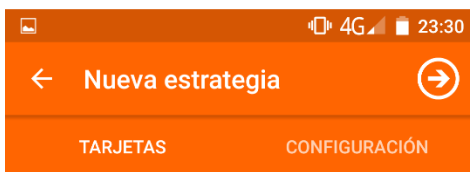
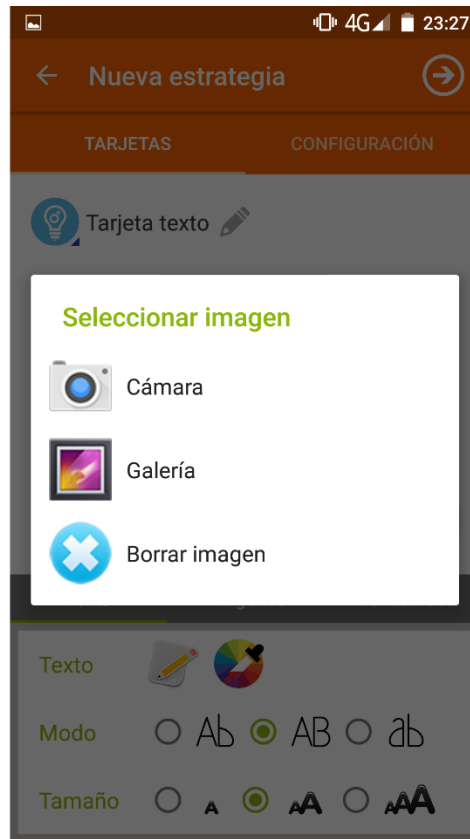
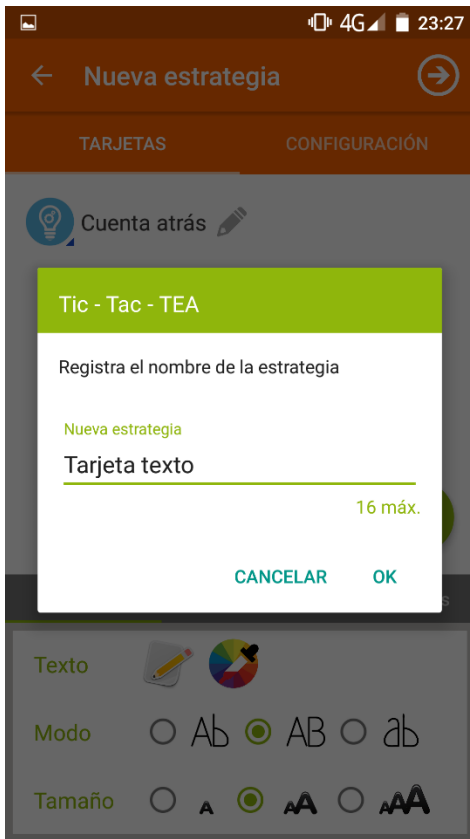


Figura 35. Prueba. Nueva estrategia con tarjeta de texto. Parte III

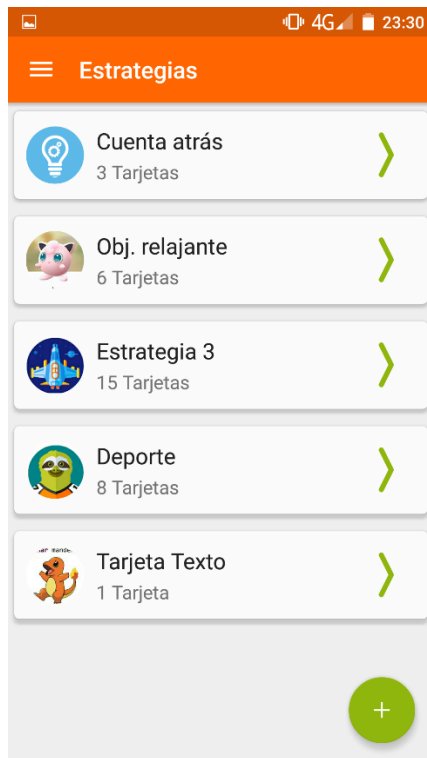


Figura 36. Prueba. Nueva estrategia con tarjeta de texto. Parte IV

7.1.2. Nueva estrategia con tarjeta con recurso propio

Esta actividad ha sido superada y los resultados obtenidos han sido los siguientes:

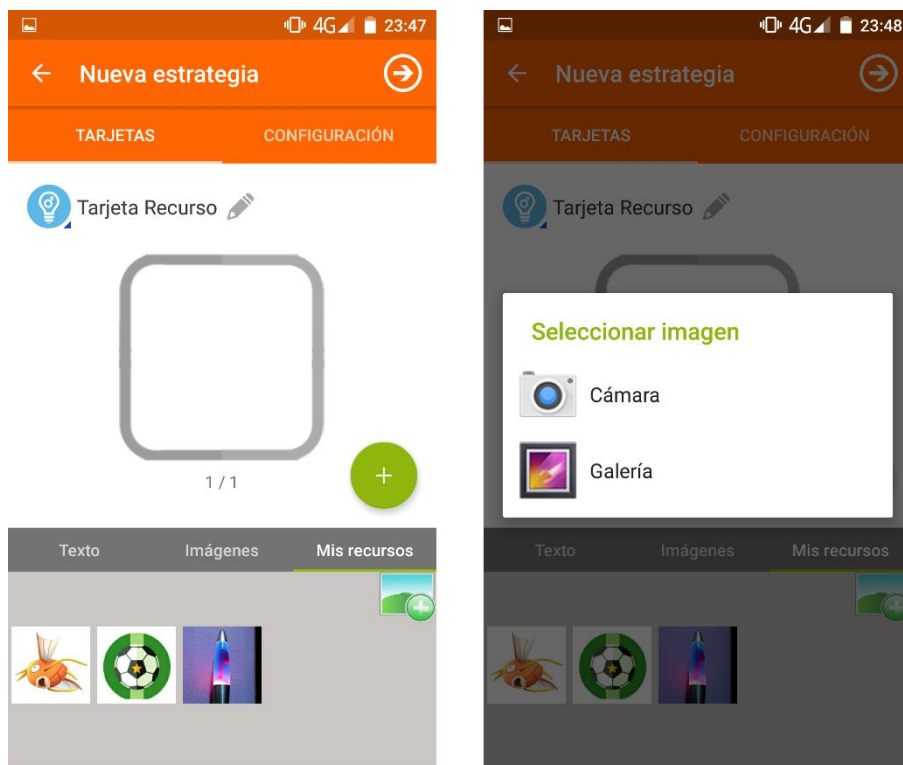
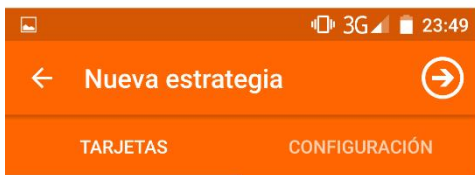
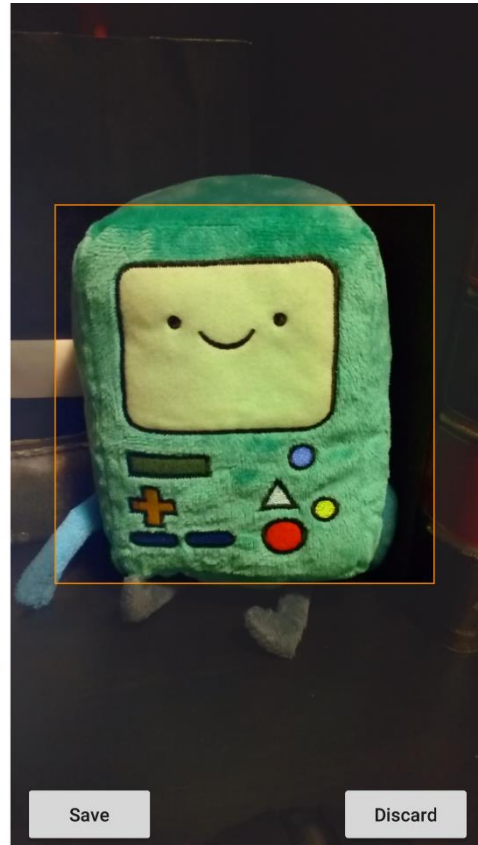


Figura 37. Prueba. Nueva estrategia con tarjeta de recurso. Parte I



 Tarjeta Recurso 

 Tarjeta Recurso 



1/1



1/1

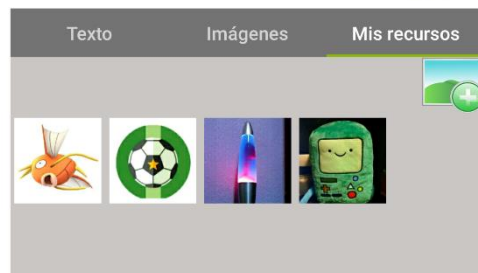
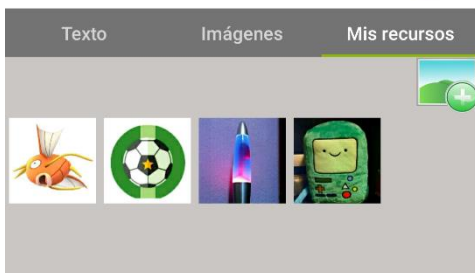


Figura 38. Prueba. Nueva estrategia con tarjeta de recurso. Parte II

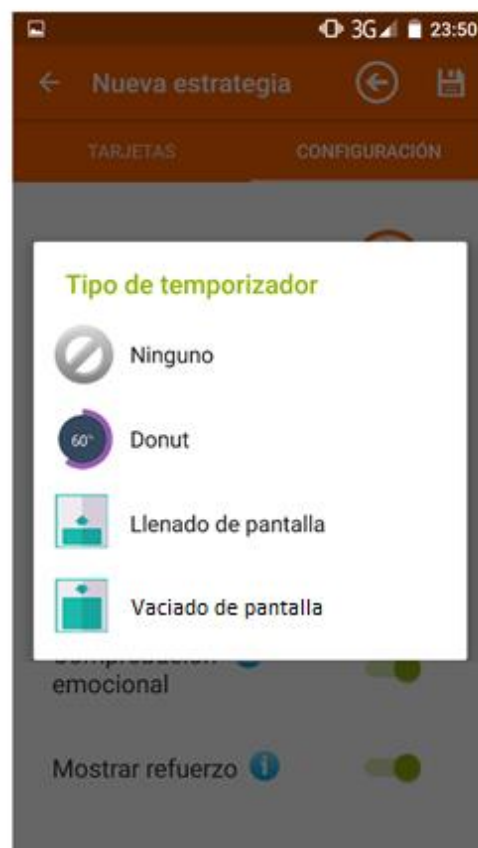
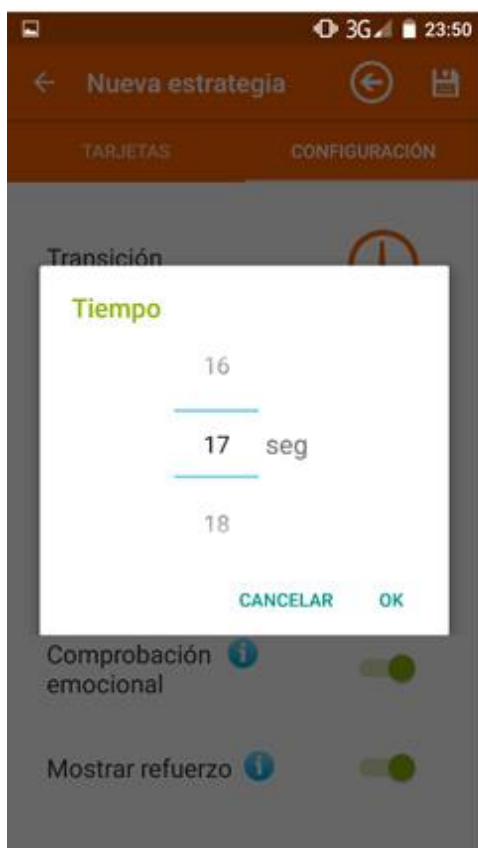
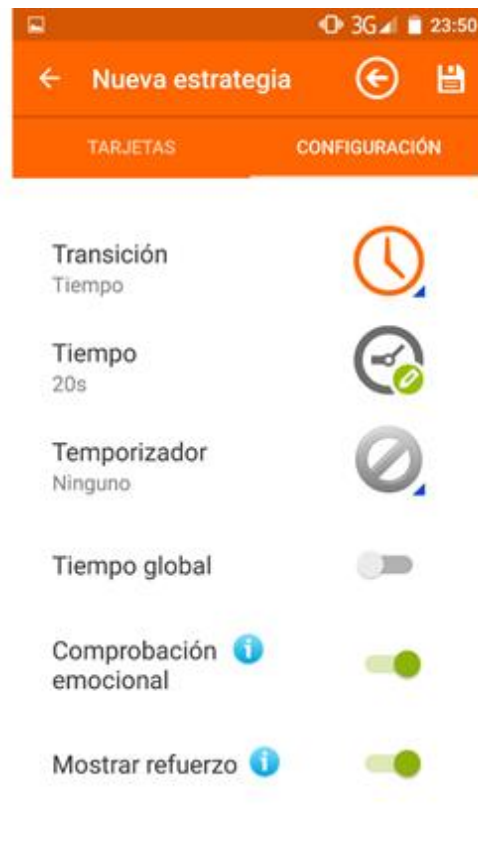


Figura 39. Prueba. Nueva estrategia con tarjeta de recurso. Parte III

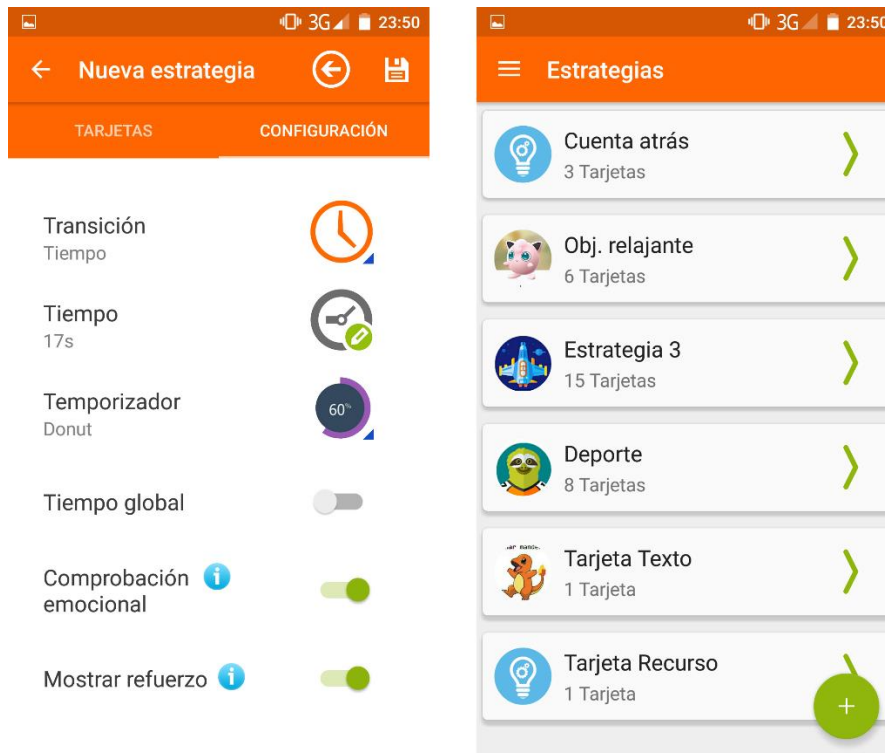


Figura 40. Prueba. Nueva estrategia con tarjeta de recurso. Parte IV

7.1.3. Nueva regulación con edición de estrategias

Esta actividad ha sido superada y los resultados obtenidos han sido los siguientes:

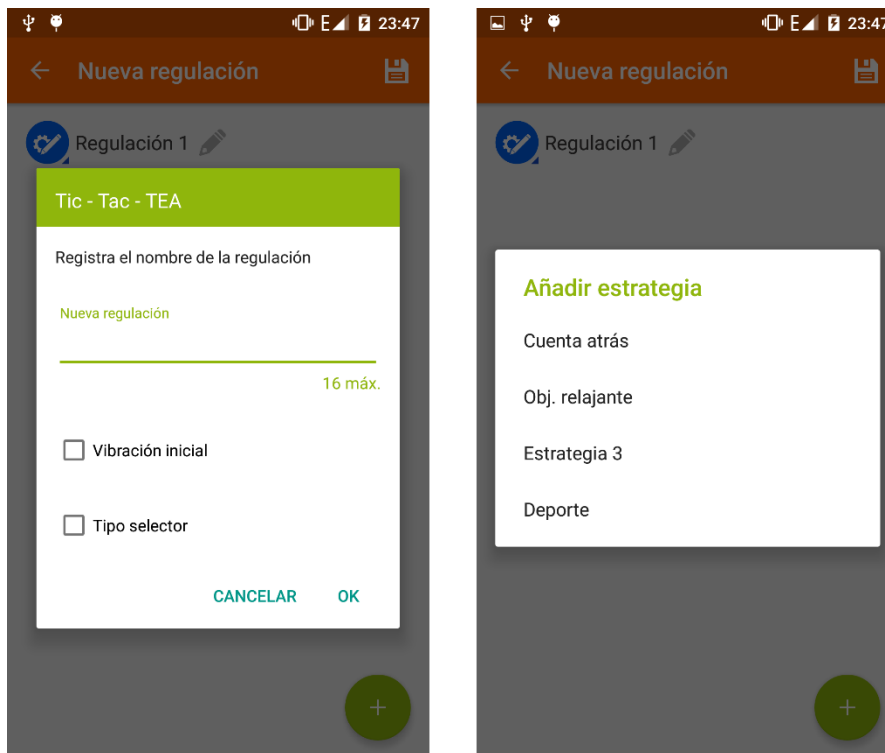


Figura 41. Prueba. Nueva regulación con edición de estrategias. Parte I

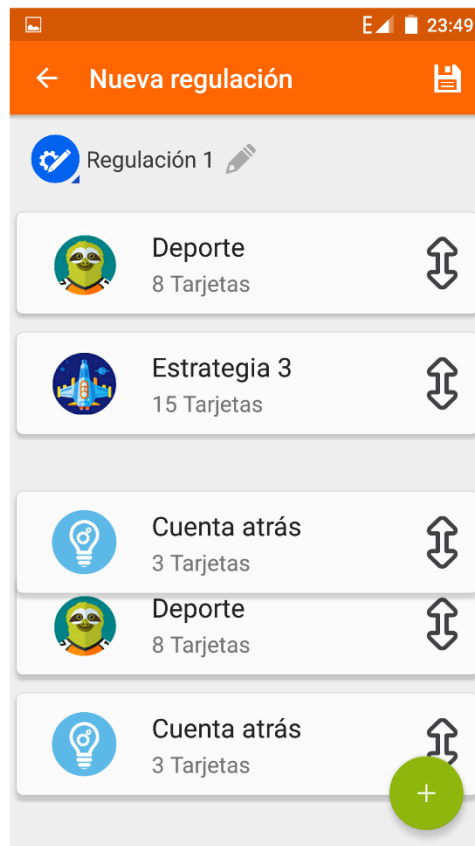
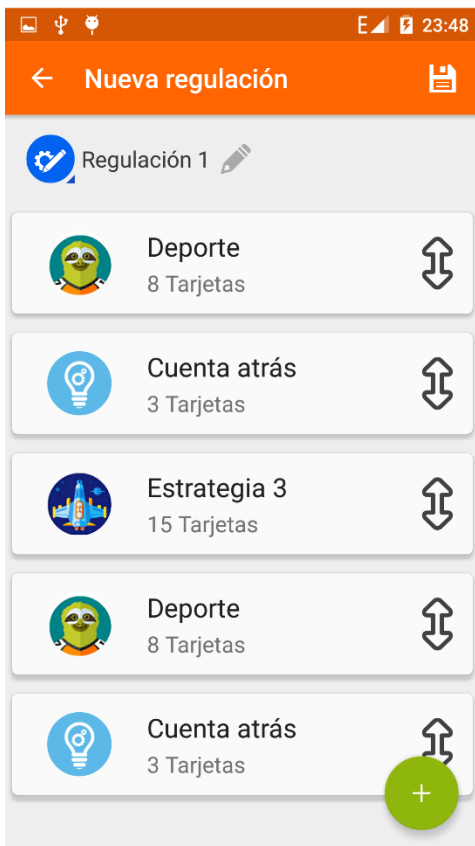
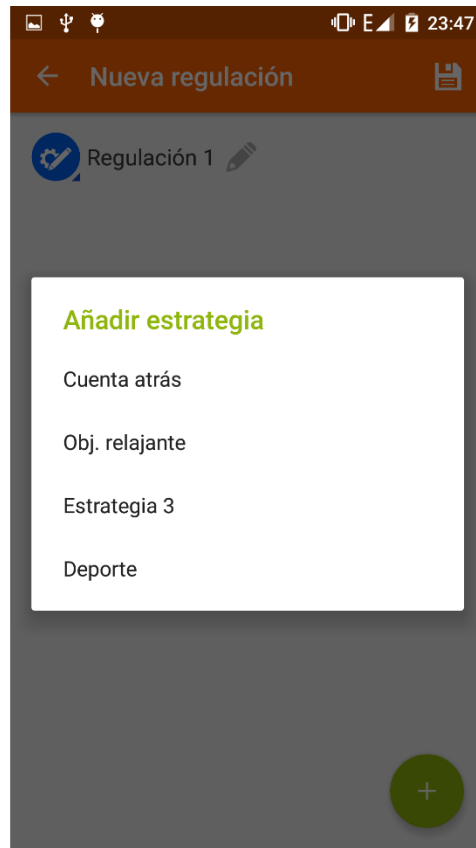
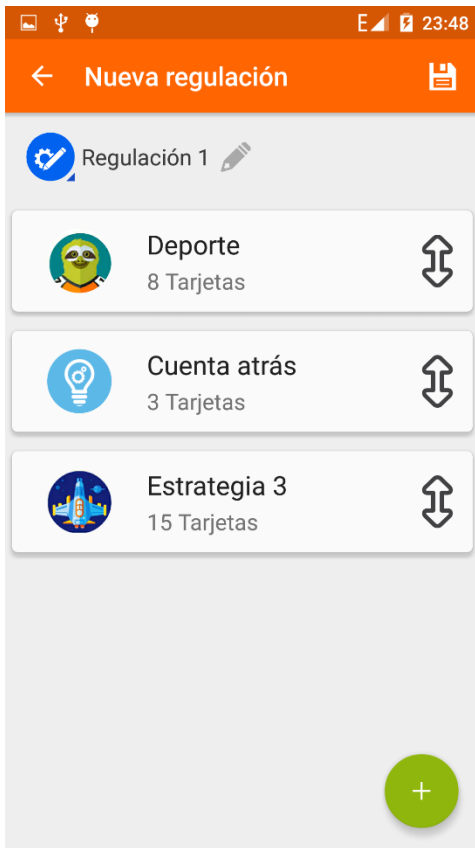


Figura 42. Prueba. Nueva regulación con edición de estrategias. Parte II

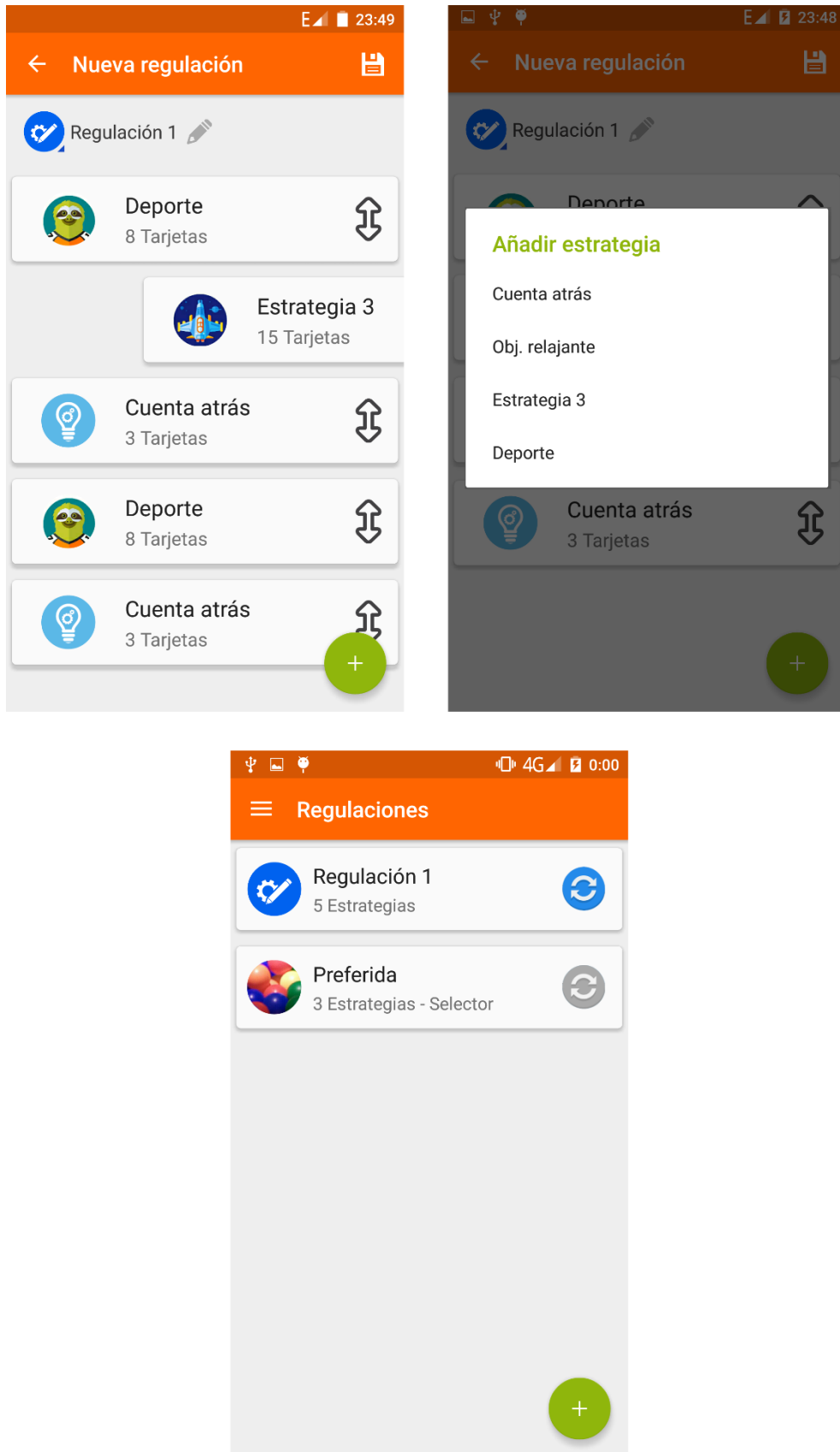


Figura 43. Prueba. Nueva regulación con edición de estrategias. Parte III

7.1.4. Cambio de idioma de aplicación

Esta actividad ha sido superada y el resultado obtenidos ha sido el siguiente:

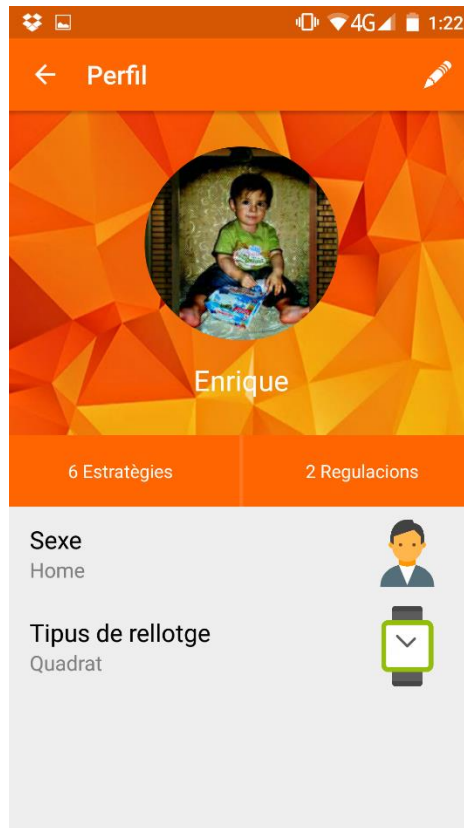


Figura 44. Prueba. Cambio de idioma de aplicación

7.2. Pruebas de integración

El resultado de la prueba de adaptación de un pictograma a la pantalla del reloj es:



Figura 45. Prueba. Adaptación de pictograma a pantalla del smartwatch

El resultado de la integración de una regulación de tipo selector, da como resultado en el reloj:

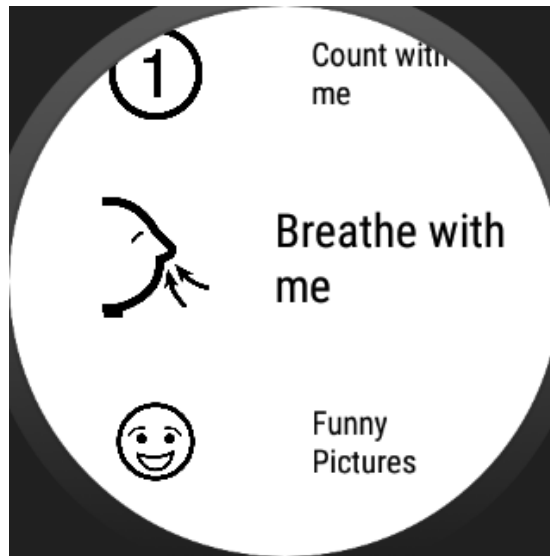


Figura 46. Prueba. Adaptación de regulación con selector a pantalla del smartwatch

7.3. Evaluación heurística

A continuación, se definen cada una de las heurísticas de Nielsen y su valoración obtenida.

7.3.1. Visibilidad del estado del sistema

El usuario siempre debe saber exactamente qué es lo que el sistema está haciendo. Además, se debe ofrecer al usuario una permanente retroalimentación, en un tiempo razonable, de las acciones que esté realizando el sistema.

Acción	Interpretación	Valoración
El sistema informa al usuario de donde se encuentra siempre.	El nombre de la pantalla actual se muestra siempre en la toolbar.	10
Ofrece retroalimentación en un tiempo razonable.	Ofrece feedback de cada acción, ya sea visual o con un mensaje.	9
Informa de las acciones realizadas por el usuario.	Cuando se realiza una acción se informa al usuario.	9

Tabla 2. Valoración de Visibilidad del estado del sistema

7.3.2. Lenguaje de los usuarios

El sistema debe hablar el lenguaje del usuario, las palabras, frases y conceptos deben de ser familiares para el usuario, en lugar de usar los términos orientados al sistema. Además, se deberá seguir las convenciones usadas en el mundo real, haciendo que la información aparezca en un orden lógico y natural.

Acción	Interpretación	Valoración
El sistema debe hablar el mismo lenguaje que el usuario.	Se considera el lenguaje utilizado por los usuarios (pasos, estrategias, regulaciones).	10
Las palabras, frases y conceptos son familiares para el usuario.	Los términos utilizados en la página son conocidos y de uso cotidiano para el usuario.	8
Muestra la información en un orden lógico y natural.	En las distintas páginas la información se muestra con un orden natural y lógico.	9

Tabla 3. Valoración de Lenguaje de los usuarios

7.3.3. Control y libertad para el usuario

En ocasiones, los usuarios pueden elegir funciones por error y necesitarán “salidas de emergencia” claramente marcadas. Se debe de proveer al usuario con funciones para deshacer y rehacer las acciones que haya realizado.

Acción	Interpretación	Valoración
El sistema debe proporcionar “salidas de emergencia” claramente marcadas.	Se puede volver a páginas anteriores mediante botones de retroceso.	10
Debe proveer funciones de deshacer y rehacer las acciones realizadas por el usuario.	Cuando se esté realizando un proceso como el crear una nueva estrategia, se permite retroceder a la pestaña anterior y realizar modificaciones.	9
Proporciona control y libertad al usuario.	En cualquier momento se puede acceder a página disponible.	9

Tabla 4. Valoración de Control y libertad para el usuario

7.3.4. Consistencia y estándares

Los usuarios no tienen por qué preguntarse si palabras, situaciones o acciones distintas significan lo mismo. Se deben seguir las convenciones de las plataformas a las que pertenece la aplicación. Es decir, escoger los estilos por default de botones, popups,

etc. provistas por la plataforma.

Acción	Interpretación	Valoración
Sigue las convenciones de las plataformas a las que pertenece la aplicación.	Sigue las convenciones del Material Design de Google (Android).	10
Las palabras, situaciones o acciones significan lo mismo.	En general, las acciones o metáforas de ellas significan lo mismo, sin llevar a la confusión de los usuarios.	9
Utiliza botones, popup, switches, etc. por defecto.	Mantiene el estilo por defecto de botones y otros elementos, según las recomendaciones del Material design.	10

Tabla 5. Valoración de Consistencia y estándares

7.3.5. Ayuda a los usuarios a reconocer, diagnosticar y recuperarse de los errores

Los mensajes de error deben de ser expresados en un lenguaje claro, indicando exactamente el problema, y proveyendo constructivamente una solución.

Acción	Interpretación	Valoración
Los mensajes de error tienen que ser claros.	Los errores se muestran de forma clara y en la zona respectiva para que se visualicen antes.	9
El sistema debe sugerir una solución.	La aplicación se mantiene en la página donde se ha producido el error, pero no sugiere soluciones.	5
El sistema identifica el problema.	Se indica específicamente el error detectado.	8

Tabla 6. Valoración de Ayuda a los usuarios reconocimiento, diagnóstico y recuperación de errores

7.3.6. Prevención de errores

Aún mejor que desarrollar buenos mensajes de error es tener un diseño cuidadoso que evite la ocurrencia de ellos. Por ejemplo, si se requiere completar una pantalla con campos obligatorios, se debería destacarlos sobre el resto de las entradas. También, es muy conveniente hacer una validación de los datos antes de guardarlos.

Acción	Interpretación	Valoración
Destacar los campos obligatorios de una pantalla.	No se destaca que un dato sea obligatorio, pero sí se controla su valor antes de guardar.	8
Valida los datos antes de guardarlos.	No se permite guardar estructuras si algún dato obligatorio no ha sido introducido.	8
El sistema informa y ejemplifica las acciones que puede realizar el usuario.	Se informa con detalle las acciones que se pueden realizar y se ejemplifican claramente.	8

Tabla 7. Valoración de Prevención de errores

7.3.7. Reconocimiento antes que cancelación

Minimizar la carga de la memoria del usuario haciendo que los objetos, acciones y opciones sean visibles. Las instrucciones de uso del sistema deben ser visibles y accesibles cuando se considere necesario.

Acción	Interpretación	Valoración
Hace visibles y accesibles las instrucciones cuando es necesario.	Hay un menú lateral que va mostrando jerárquicamente opciones.	7
Utiliza mecanismos de búsqueda y accesos directos.	En la pantalla principal se tiene acceso directo a la pantalla de estrategias y regulaciones, pero no hay opciones de búsqueda.	6
El usuario no tiene que memorizar los datos que ya ha introducido.	Mientras el usuario navega entre pestañas al crear una regulación, estos datos se mantienen en memoria.	8

Tabla 8. Valoración de Reconocimiento antes que cancelación

7.3.8. Flexibilidad y eficiencia de uso

Los aceleradores (invisibles para el usuario novato) pueden hacer más rápida la interacción para el usuario experto. El sistema debe ser eficiente tanto para los usuarios expertos como inexpertos. Para lograr esto, es conveniente permitir a los usuarios que personalicen ciertas acciones frecuentes.

Acción	Interpretación	Valoración
El sistema ofrece aceleradores para acciones frecuentes.	No se ofrecen aceleradores.	1

Permite a los usuarios personalizar acciones.	Permite a los usuarios personalizar sus recursos propios.	5
Proporciona opciones para que la interacción sea más rápida.	Se permite la interacción con la aplicación mediante gestos.	8

Tabla 9. Valoración de Flexibilidad y eficiencia de uso

7.3.9. Estética de diálogos y diseño minimalista

Los diálogos no deben contener información que sea irrelevante o que rara vez sea de utilidad. La presentación visual ha de respetar las guías de diseño.

Acción	Interpretación	Valoración
La información más importante se coloca en la parte superior de la página.	La información más importante para el usuario se destaca claramente.	7
No debe contener información irrelevante.	Se muestra la información justa y necesaria en todos los diálogos utilizados.	10
La presentación visual respeta los principios de contraste, repetición, alineamiento y proximidad.	La distribución de los elementos y del menú se presentan de forma clara y ordenada, con el alineamiento y el contraste adecuados. Se sigue la guía del Material design.	9

Tabla 10. Valoración de Estética de diálogos y diseño minimalista

7.3.10. Ayuda general y documentación

Es recomendable tener documentación y opciones de ayuda en la herramienta. La información de soporte debe estar orientada a las acciones del usuario y no debe ser muy extensa.

Acción	Interpretación	Valoración
La información debe de ser fácil de buscar.	Se proporciona una pantalla de tutorial accesible desde el menú lateral.	10
La documentación debe estar orientada a las acciones del usuario y no debe ser muy extensa.	Las explicaciones del tutorial son cortas, claras y utilizan contenido visual centrado en las acciones del usuario.	10
Además de documentación debe ofrecer otras opciones de ayuda.	Además del tutorial, se usan íconos de ayuda para aclarar información en un popup sobre determinados elementos.	10

Tabla 11. Valoración de Ayuda general y documentación

7.4. Encuesta SUS

Una vez realizadas las encuestas SUS (ver Anexo C), se han transcrito las respuestas de cada participante para obtener el promedio total de la valoración (ver Tabla 12).

Usuario	q1	q2	q3	q4	q5	q6	q7	q8	q9	q10	SUS Score
Lupe	5	2	4	1	5	1	3	2	5	1	87,5
Laura	4	2	4	1	4	1	4	1	3	1	82,5
Juan Carlos	5	2	5	1	5	1	3	2	4	3	82,5
Andrés	4	2	4	1	5	1	5	1	4	3	85,0
										PROMEDIO	84,4

Tabla 12. Resultado de encuesta SUS

Como podemos observar, el promedio obtenido a partir de la encuesta SUS, es de 84,4. Esto supera nuestro mínimo esperado que era del 75,0. Las dos primeras usuarias son expertas del centro I.P.P. “Quintero Lumbreras”, mientras que los dos restantes son investigadores y profesionales de la Escuela Politécnica Superior de la UAM. A pesar de tratarse de usuarios con perfiles diferentes, la usabilidad ha sido altamente satisfactoria para ambos y se han obtenido promedios cercanos.

Cabe destacar que la máxima puntuación obtenida ha venido por parte de la directora del centro I.P.P. “Quintero Lumbreras”. Además, el feedback en forma de comentarios recibidos han mostrado el alto grado de apoyo y conformidad con el resultado de este proyecto.

8. Conclusiones y trabajo futuro

Tras las pruebas y resultados obtenidos de la aplicación, podemos concluir que la herramienta cumple perfectamente los objetivos por los cuales fue desarrollada.

Hemos creado una herramienta de autor que permite a usuarios no expertos crear interfaces adaptadas para mostrarse en un smartwatch. Estas interfaces son, además, completamente personalizables para adecuarse a cada usuario final y tienen un objetivo que responde a una necesidad no resuelta actualmente: La autorregulación emocional en el momento en que se detecte una crisis.

La valoración positiva y el reconocimiento recibido a este proyecto por parte de diversas entidades, certifican la calidad obtenida.

Como trabajo futuro, necesitamos refinar aún más la aplicación teniendo nuevas reuniones con las expertas en TEA. A partir de esta versión final 1.0, esperamos incluir nuevos requisitos de personalización de estrategias y regulaciones. Además, esperamos mejorar la usabilidad de forma que familiares o tutores de personas con autismo puedan usarla sin dificultades, sin necesidad de ser terapeutas profesionales.

Otro tema pendiente son las pruebas de integración con el smartwatch. Es necesario hacer una batería completa de pruebas para comprobar que todos los contenidos se ajustan al smartwatch y que todas las funcionalidades responden como se espera.

Respecto al diseño y usabilidad, se tiene pensado trabajar con una diseñadora para cambiar algunas imágenes, íconos y mejorar las pantallas del tutorial. Tras estos cambios, la aplicación pasará a estar disponible en Google Play.

Respecto a funcionalidades que nos hubiera gustado incluir, queda pendiente internacionalizar más la herramienta, añadir más opciones de personalización de texto y mostrar una vista previa o simulador de la estrategia completa antes de enviarla al reloj.

En conclusión, tras la realización de este proyecto tenemos el deseo de seguir trabajando en esta línea, colaborando con organizaciones centradas en personas con necesidades especiales. Vemos que, aunque la aplicación desarrollada va dirigida a personas con TEA, dadas sus características, también podríamos ampliarlas a otros usuarios. Además, la tecnología wearable y las aplicaciones de smartphone junto a smartwatch abren un gran abanico de posibilidades para crear herramientas innovadoras e interesantes.

9. Referencias

- [1] American Psychiatric Association. (2013). *Diagnostic and statistical manual of mental disorders (DSM-5®)*. American Psychiatric Pub.
- [2] García-Maldonado G. *Trastorno autista. Una revisión*. Revista Mexicana de Pediatría 2004; 71: 33-7.
- [3] AACAP Official action. *Practice parameters for the assessment and treatment of children, adolescents and adults with autism and other pervasive developmental disorders*. J Am Acad Child Adolescents Psychiatry 1999; 38 (Supl. 12): 32-54s.
- [4] Ruggieri, V. L. (2013). *Empatía, cognición social y trastornos del espectro autista*. Rev Neurol, 56(Supl 1), S13-21.
- [5] Samson A.C., Phillips J.M., Parker, K.J., Shah S, Gross J, Hardan A. 2014. *Emotion Dysregulation and the Core Features of Autism Spectrum Disorder*.
- [6] Nota, L., Ferrari, L., Soresi, S., & Wehmeyer, M. (2007). *Self-determination, social abilities and the quality of life of people with intellectual disability*. Journal of Intellectual Disability Research, 51(11), 850-865.
- [7] Bijker, W. E., Hughes, T. P., Pinch, T., & Douglas, D. G. (2012). *The social construction of technological systems: New directions in the sociology and history of technology*. MIT press.
- [8] Instituto de Psico – Pediatría “Dr. Quintero Lumbreras”. <http://www.somosipp.com/>
- [9] Soluciones tecnológicas para autismo - Fundación Orange
<http://www.proyectosfundacionorange.es/convocatoria2015/seleccionados.php>
- [10] Pottie, C. G., & Ingram, K. M. (2008). *Daily stress, coping, and well-being in parents of children with autism: a multilevel modeling approach*. Journal of Family Psychology, 22(6), 855.
- [11] Informe Horizon (2012). Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado. Ministerio de Educación, cultura y deporte.
- [12] Swan, M. (2012). *Sensor mania! the internet of things, wearable computing, objective metrics, and the quantified self 2.0*. Journal of Sensor and Actuator Networks, 1(3), 217-253.
- [13] Petersen, K., Feldt, R., Mujtaba, S., & Mattsson, M. (2008, June). *Systematic mapping studies in software engineering*. In 12th International Conference on Evaluation and Assessment in Software Engineering (Vol. 17, p. 1).
- [14] Hirokawa, M., Funahashi, A., Itoh, Y., & Suzuki, K. (2014). *Design of affective robot-assisted activity for children with autism spectrum disorders*. Paper presented at the IEEE RO-MAN 2014 - 23rd IEEE International Symposium on Robot and Human

Interactive Communication: Human-Robot Co-Existence: Adaptive Interfaces and Systems for Daily Life, Therapy, Assistance and Socially Engaging Interactions, 365-370. doi:10.1109/ROMAN.2014.6926280

[15] Mondragon, A. L., Nkambou, R., & Poirier, P. (2016, June). *Towards an Effective Affective Tutoring Agent in Specialized Education*. In International Conference on Intelligent Tutoring Systems (pp. 402-408). Springer International Publishing.

[16] ARASAAC. Portal Aragonés de la Comunicación Aumentativa y Alternativa. <http://arasaac.org/>

[17] Nielsen, J., (2015) Nielsen Norman Group. *NN/g, 10 Usability Heuristics for User Interface Design*: <http://www.nngroup.com/articles/ten-usability-heuristics/>

10. Anexos

Anexo A. Evolución del diseño de Tic-Tac-TEA.

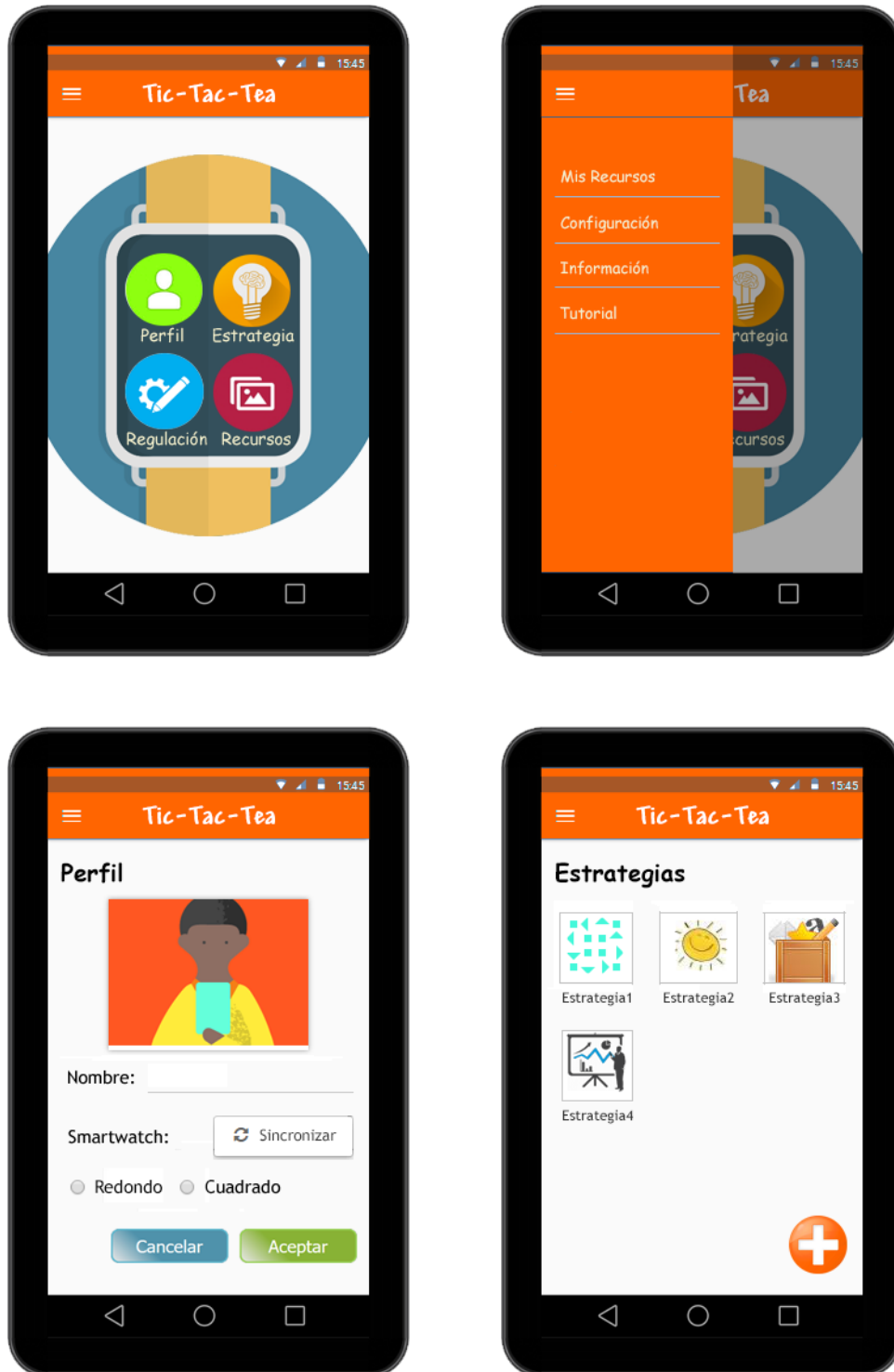


Figura 47. Evolución del diseño de Tic-Tac-TEA. Versión 1.0. Parte I

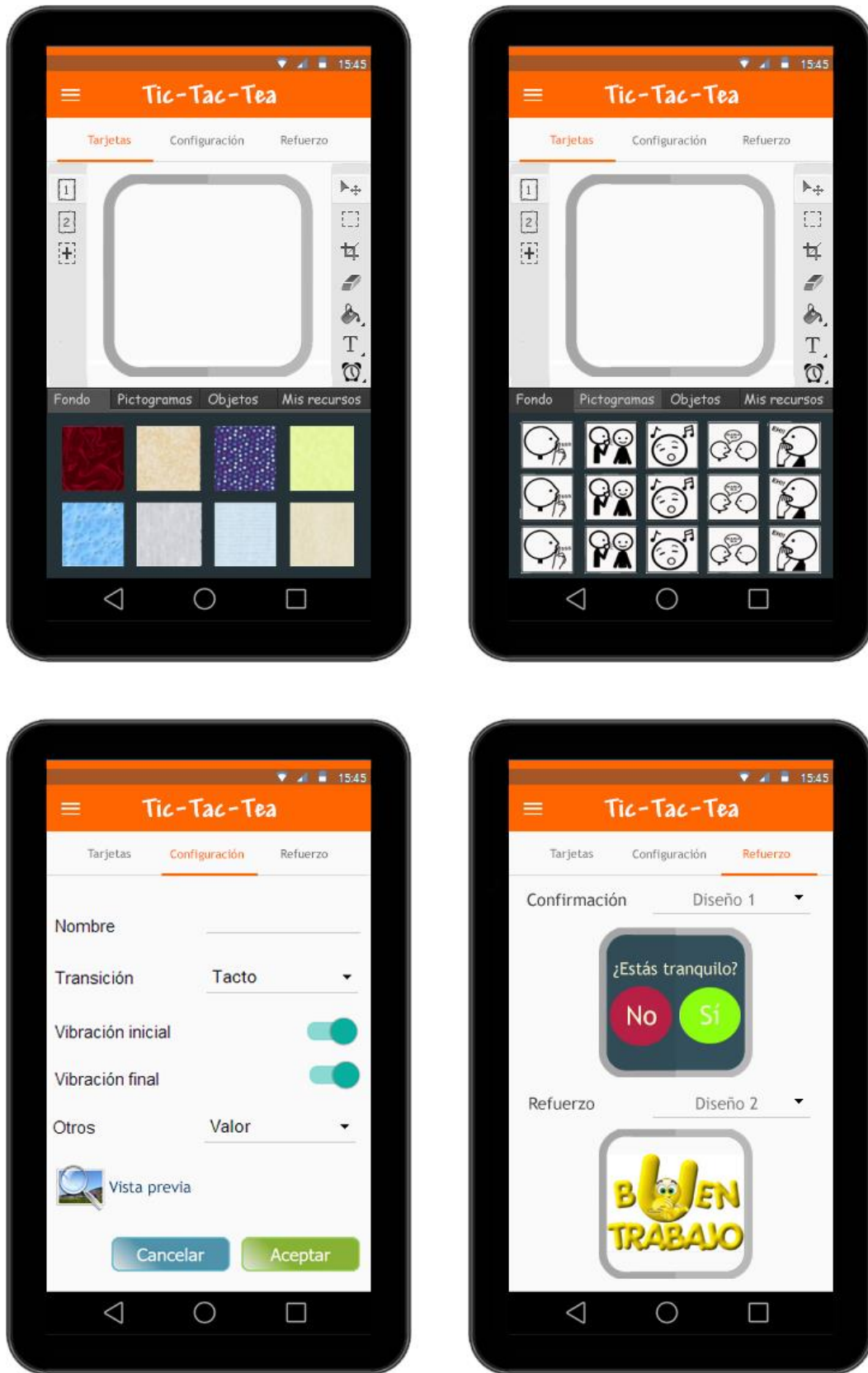


Figura 48. Evolución del diseño de Tic-Tac-TEA. Versión 1.0. Parte II



Figura 49. Evolución del diseño de Tic-Tac-TEA. Versión 1.0. Parte III

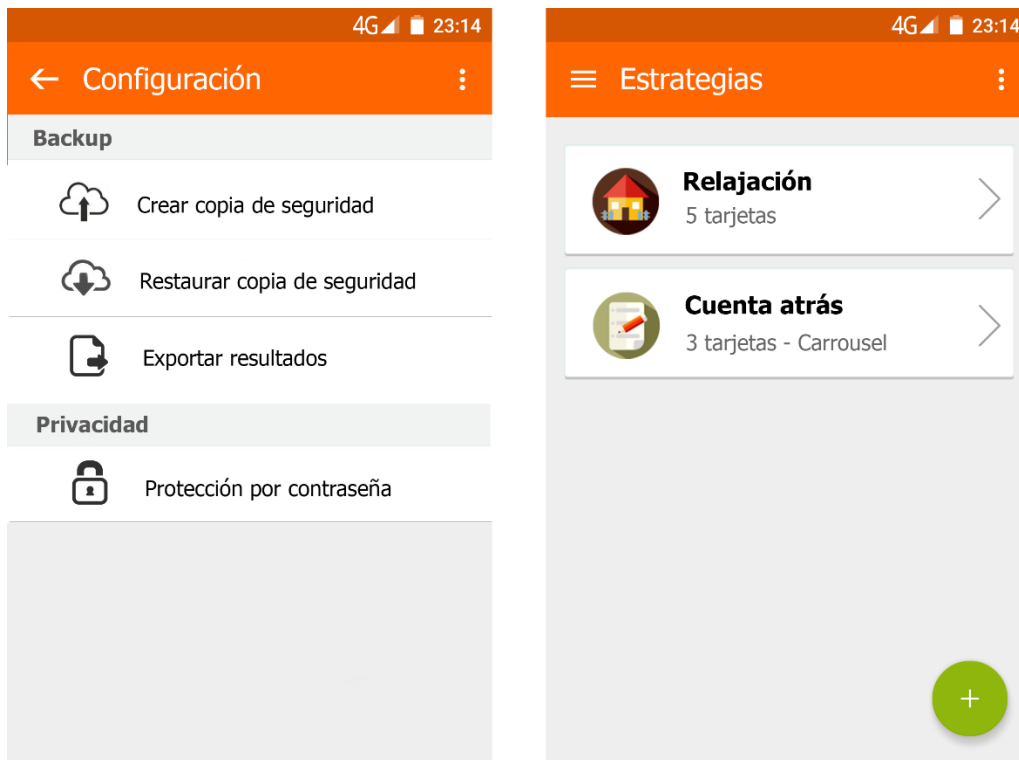


Figura 50. Evolución del diseño de Tic-Tac-TEA. Versión 2.0. Parte I

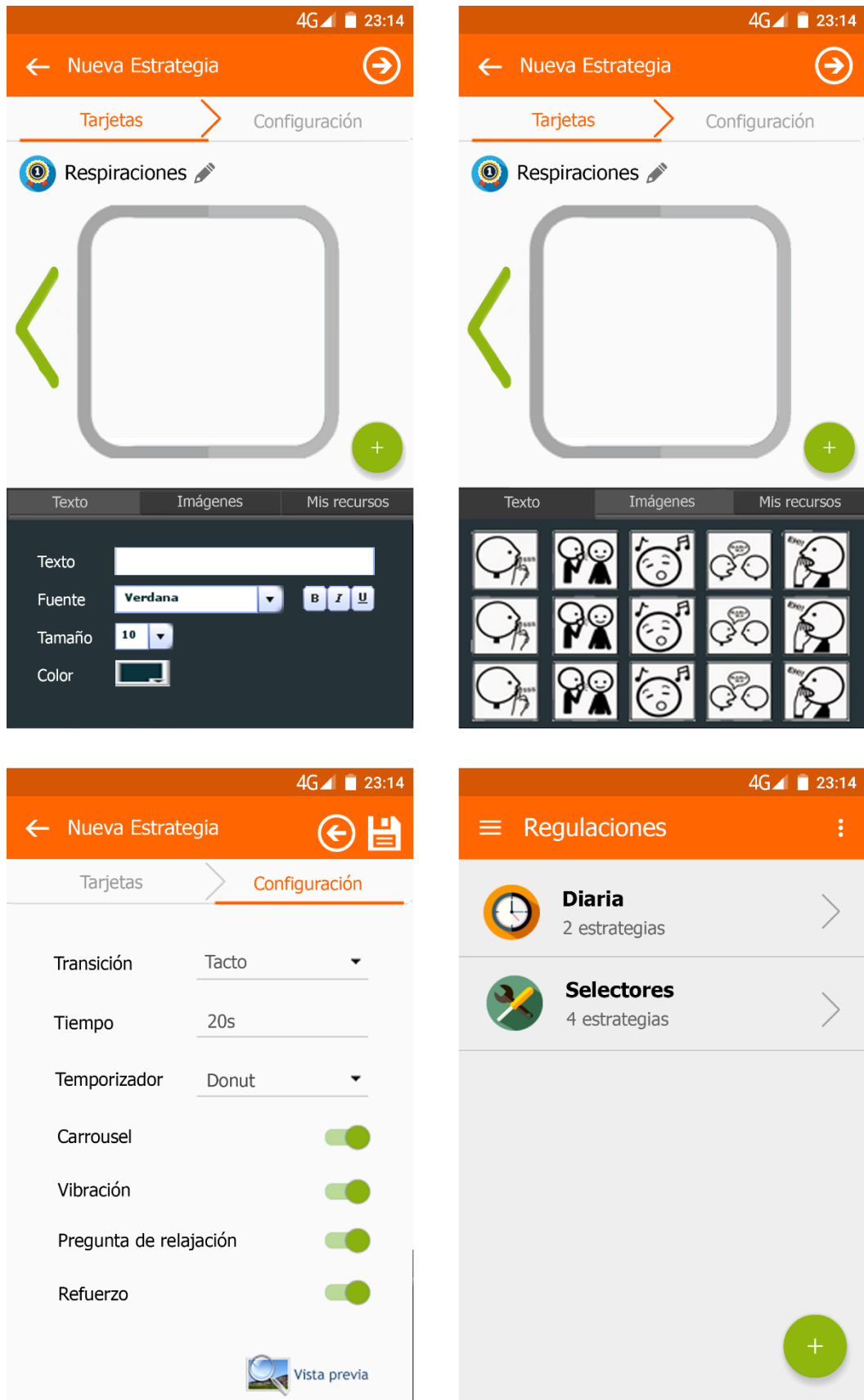


Figura 51. Evolución del diseño de Tic-Tac-TEA. Versión 2.0. Parte II

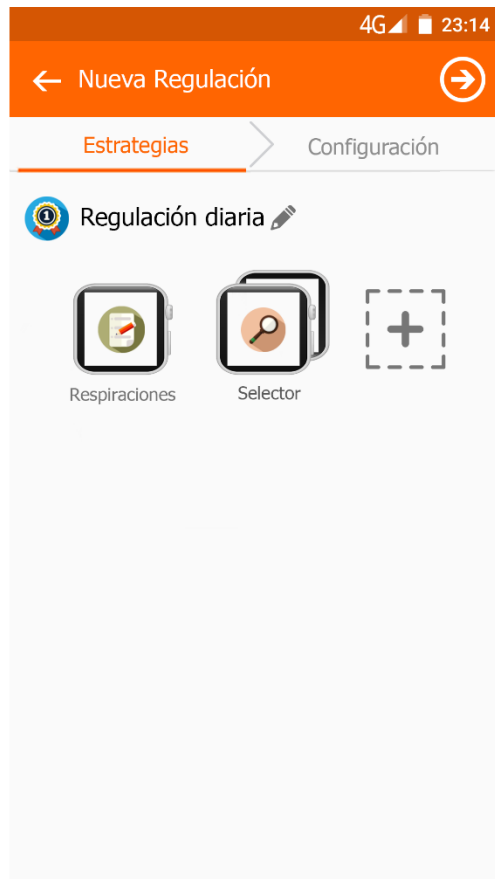
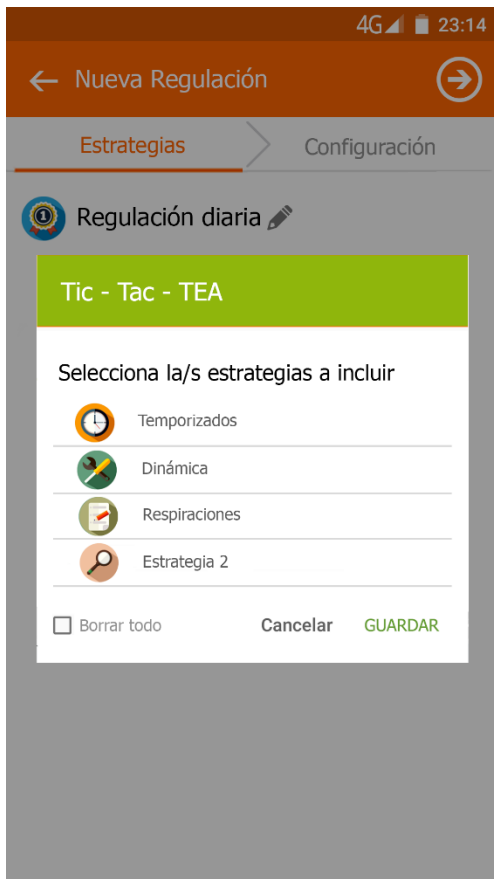


Figura 52. Evolución del diseño de Tic-Tac-TEA. Versión 2.0. Parte III

Anexo B. Protocolo de comunicación entre smartphone y smartwatch

```
{
  "Regulation": {
    "Id": 15,
    "Name": "Regulacion 20",
    "Vibration": true,
    "Audio": null,
    "RegulationSchedule": [
      [
        12,
        21
      ],
      5,
      6,
      2
    ],
    "Strategies": [
      {
        "Id": 5,
        "Name": "9 y 8",
        "Icon": "one.png",
        "Time": 0,
        "isGlobalTime": false,
        "StressCondition": false,
        "Question": 0,
        "TimerType": 0,
        "isCarrusel": false,
        "Reinforcement": true,
        "Steps": [
          "nine.png",
          "eight.png"
        ]
      },
      {
        "Id": 6,
        "Name": "Carrusel fill",
        "Icon": "two.png",
        "Time": 30,
        "isGlobalTime": true,
        "StressCondition": false,
        "Question": 0,
        "TimerType": 2,
        "isCarrusel": true,
        "Reinforcement": true,
        "Steps": [
          "one.png",
          "two.png",
          "reir.png"
        ]
      },
      {
        "Id": 2,
        "Name": "Mira imágenes",
        "Icon": "two.png",
        "Time": 0,
        "isGlobalTime": false,
        "StressCondition": false,

```



```

    "Question": 0,
    "TimerType": 0,
    "isCarrusel": false,
    "Reinforcement": false,
    "Steps": [
      "seven.png",
      "reir.png"
    ]
  },
  {
    "Id": 12,
    "Name": "donut global 20s",
    "Icon": "two.png",
    "Time": 20,
    "isGlobalTime": true,
    "StressCondition": false,
    "Question": 0,
    "TimerType": 1,
    "isCarrusel": false,
    "Reinforcement": false,
    "Steps": [
      "seven.png",
      "reir.png"
    ]
  },
  {
    "Id": 21,
    "Name": "tocar",
    "Icon": "one.png",
    "Time": 0,
    "isGlobalTime": false,
    "StressCondition": false,
    "Question": 0,
    "TimerType": 0,
    "isCarrusel": false,
    "Reinforcement": false,
    "Steps": [
      "TOCA.png"
    ]
  }
]
}
}
}

```

Anexo C. Encuesta SUS

Encuesta SUS: Tic - Tac -TEA					
Por favor, marque la casilla que refleje su respuesta inmediata a cada afirmación. No piense demasiado sobre cada afirmación. Asegúrese de que responde todas las afirmaciones. Si no sabe que responder, simplemente marque la casilla "3".					
	Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo
1. Pienso que me gustaría utilizar este producto con frecuencia					
	1	2	3	4	5
2. Encontré el producto innecesariamente complejo					
	1	2	3	4	5
3. Me pareció que el producto era fácil de usar					
	1	2	3	4	5
4. Creo que voy a necesitar la ayuda de una persona técnica para poder utilizar este producto					
	1	2	3	4	5
5. Me pareció que las diversas funciones de este producto están bien integradas					
	1	2	3	4	5
6. Me pareció que había demasiada inconsistencia en este producto					
	1	2	3	4	5
7. Me imagino que la mayoría de personas aprenderían a usar este producto muy rápidamente					
	1	2	3	4	5
8. Me pareció que el producto es muy complicado de usar					
	1	2	3	4	5
9. Me sentí con mucha confianza al usar el producto					
	1	2	3	4	5
10. Tenía que aprender muchas cosas antes de que pudiera comenzar a utilizar este producto					
	1	2	3	4	5

Figura 53. Encuesta SUS