

UNIVERSIDAD POLITÉCNICA DE MADRID

Escuela Técnica Superior de Ingeniería de Sistemas Informáticos



**Digitalización del Trail Making Test
con una aplicación multiplataforma
para personas con Esclerosis
Múltiple**

PROYECTO FIN DE GRADO

Xiaoyang Peng

Grado en Ingeniería de Software

Madrid, 2025



UNIVERSIDAD POLITÉCNICA DE MADRID
Escuela Técnica Superior de Ingeniería de
Sistemas Informáticos

Grado en Ingeniería de Software

**Digitalización del Trail Making Test
con una aplicación multiplataforma
para personas con Esclerosis
Múltiple**

PROYECTO FIN DE GRADO

Autor:

Xiaoyang Peng

Grado en Ingeniería de Software

Directores:

Bernardo Tabuenca Archilla

Joaquín Bienvenido Ordieres Mere

Madrid, 2025

Agradecimientos

En primer lugar, me gustaría agradecer a mi familia, y muy especialmente a mi madre. Gracias a vuestro apoyo he podido terminar estos cuatro años de carrera.

También quiero expresar mi agradecimiento a mis tutores de proyecto, Bernardo Tabuena Archilla y Joaquín Bienvenido Ordieres Mere, por su ayuda y orientación durante el desarrollo de este trabajo.

Resumen

El objetivo de este proyecto es desarrollar una aplicación basada en Flutter para digitalizar el Trail Making Test (TMT), una prueba médica que utilizarán neurólogos y psicólogos para evaluar las capacidades cognitivas de los pacientes.

La aplicación funcionará en tabletas y móviles con sistemas operativos Android e iOS. A diferencia del test TMT tradicional, esta aplicación podrá generar las pruebas de forma aleatoria y registrará automáticamente el tiempo y otras métricas como el número de errores, la duración media de las pausas, la duración media de las elevaciones del dedo, la velocidad media entre círculos y dentro de ellos, el tiempo medio entre círculos y dentro de ellos, la presión media aplicada y el tamaño medio del área de contacto. Posteriormente, será capaz de enviar todas estas métricas a un servidor.

El proyecto se desarrollará en colaboración con el Hospital Universitario de Getafe para facilitar la evaluación neurológica de los pacientes, especialmente aquellos con enfermedades que afectan las capacidades cognitivas, como la esclerosis múltiple.

Palabras Clave

Flutter, Trail Making Test (TMT), Esclerosis Múltiple, Android, iOS, Evaluación cognitiva.

Abstract

The objective of this project is to develop a Flutter-based application to digitize the Trail Making Test (TMT), a medical test that neurologists and psychologists will use to assess patients' cognitive abilities.

The application will run on tablets and mobile devices with Android and iOS operating systems. Unlike the traditional TMT, this application can generate tests randomly and will automatically record the time and other metrics such as the number of errors, average pause duration, average finger lift duration, average speed between and within circles, average time between and within circles, average applied pressure, and average contact area size. Subsequently, it will be able to send all these metrics to a server.

The project will be developed in collaboration with the Hospital Universitario de Getafe to facilitate the neurological assessment of patients, especially those with diseases affecting cognitive abilities, such as multiple sclerosis.

Key words

Flutter, Trail Making Test (TMT), Multiple Sclerosis, Android, iOS, Cognitive Assessment.

Contenido

1	Introducción	11
1.1	Objetivos	12
1.2	La esclerosis múltiple	13
1.3	El Trail Making Test.....	14
1.4	Método de Evaluación Test TMT	16
1.4.1	Tiempo promedio.....	17
1.4.2	Normativas para calcular el percentil del resultado del test TMT	17
2	Estado del arte.....	21
2.1.1	En Google Play Store	21
2.1.2	En App Store	23
2.1.3	Aplicación Web.....	25
2.1.4	Resultados de la revisión.....	26
3	Desarrollo del Proyecto	27
3.1	Análisis de requisitos.....	27
3.1.1	Requisitos funcionales.....	27
3.1.2	Requisitos no funcionales	30
3.2	Planificación	31
3.2.1	Product Backlog	31
3.2.2	Cronograma del Proyecto	35
3.3	Diseño	35
3.3.1	Pantalla Principal de la aplicación.....	35
3.3.2	Pantalla de Alta de Usuario	37
3.3.3	Diálogo Seleccionar perfil	38
3.3.4	Pantalla de Mi Perfil y Eliminación	39
3.3.5	Pantalla de Historial de Pruebas (Mis Tests)	40
3.3.6	Pantalla de Instrucciones del Test Tmt(Ayuda).....	41
3.3.7	Pantalla del Test TMT.....	43
3.3.8	Pantalla de Resultados del Test TMT	44
3.4	Implementación de la solución	46
3.4.1	Arquitectura	46
3.4.1.1	Arquitectura general de la aplicación	46
3.4.1.2	Estructura del proyecto	48
3.4.1.3	Justificación de tecnologías	50
3.4.2	Base de Datos	54

3.4.2.1	Modelo Entidad Relación	55
3.4.2.2	Modelo Relacional	56
3.4.2.3	Implementación de la Base de Datos.....	57
3.4.3	Implementación de la API	57
3.4.3.1	Endpoint: Validación de Código de Referencia	58
3.4.3.2	Endpoint: Envío de Resultados del Test TMT	59
3.4.4	Cálculos y algoritmos para generar test TMT.....	60
3.4.4.1	Calcular el radio del círculo.....	60
3.4.4.2	Algoritmo de generación aleatoria de Círculos.....	61
3.4.4.3	Algoritmo para reordenar los círculos.....	64
3.4.5	Implementación de Funcionalidades Clave.....	67
3.4.5.1	Flujo General de Home.....	67
3.4.5.2	Gestión de Perfiles de Usuario	69
3.4.5.3	Flujo de Navegación del Test TMT	73
3.4.5.4	Cálculo de Métricas del Test TMT.....	76
3.4.5.5	Mecanismo de Almacenamiento de Resultados cuando no hay Internet	81
3.4.6	Herramientas y entorno de desarrollo	83
3.4.6.1	Lenguaje de Programación y Framework.....	83
3.4.6.2	Entorno de Desarrollo Integrado (IDE).....	84
3.4.6.3	Diseño y Prototipado.....	85
3.4.6.4	Pruebas de API.....	86
4	Resultado.....	87
4.1	Pruebas.....	87
4.1.1	Prueba de Gestión de Usuario.....	87
4.1.2	Prueba de Validación de Código de Referencia.....	95
4.1.3	Prueba de Flujo de Envío de Resultados.....	101
4.1.4	Prueba de Flujo Navegación de Test Tmt.....	104
4.2	Prueba Piloto.....	109
4.2.1	Contexto del Estudio.....	109
4.2.2	Análisis de Rendimiento en el Test TMT.....	110
4.2.3	Evaluación de Usabilidad y Experiencia de Usuario	114
4.3	Presupuesto	117
4.3.1	Costes personales de las actividades.....	117
4.3.2	Costes de los recursos físicos	119

4.3.3	Presupuesto del proyecto	119
4.4	Distribución y Publicación del Proyecto	120
5	Conclusiones y trabajos futuros	121
5.1	Impacto social y medioambiental.....	121
5.1.1	Impacto social.....	121
5.1.2	Impacto medioambiental.....	122
5.2	Conclusiones.....	122
5.3	Líneas futuras.....	123
	Acrónimos y abreviaturas	124
6	Bibliografía	126
7	Anexo.....	130

Índice de figuras

Ilustración 1.2-1 Variantes de evolución de la Esclerosis Múltiple según diagramas de Lublin.....	14
Ilustración 1.4-1 Tabla: Regresión Lineal Múltiple por Pasos de Sexo, Edad y Educación sobre las Puntuaciones TMT Transformadas Logarítmicamente (extraída del artículo).....	20
Ilustración 1.4-2 Tabla Desviaciones estándar de los residuos para las puntuaciones Log TMT predichas (extraída del artículo).....	20
Ilustración 1.4-1: Captura de pantalla de la aplicación TMT APP	21
Ilustración 1.4-2 Captura de pantalla de la aplicación Trail Making Test	22
Ilustración 1.4-3 Captura de pantalla de la aplicación Trail Making Test	23
Ilustración 1.4-4 Captura de pantalla de la aplicación Trail Making Test J Lite.....	24
Ilustración 1.4-5 Captura Neura Traid Making test extraída del App Store	25
Ilustración 1.4-6 Capturas de CogniFit Trail Making Test.....	26
Ilustración 3.3-1 Pantalla Principal de la aplicación.....	36
Ilustración 3.3-2 Pantalla de Alta de Usuario	37
Ilustración 3.3-3 Diálogo Seleccionar perfil	38
Ilustración 3.3-4 Pantalla de Mi Perfil y Eliminación	39
Ilustración 3.3-5 Pantalla de Historial de Pruebas (Mis Tests)	40
Ilustración 3.3-6 Pantalla de Instrucciones del Test Tmt con mano izquierda	42
Ilustración 3.3-7 Pantalla de Instrucciones del Test Tmt con mano derecha	42
Ilustración 3.3-8 Pantalla del Test TMT.....	44
Ilustración 3.3-9 Pantalla de Resultados del Test TMT	45
Ilustración 3.4-1 Capas de Clean Architecture ^[23]	48
Ilustración 3.4-2 Estructura del proyecto	50
Ilustración 3.4-3 Diagrama Modelo Entidad Relación.....	55
Ilustración 3.4-4 Modelo Relacional	57
Ilustración 3.4-5 Ejemplo muestra la generación aleatoria de Círculos	63
Ilustración 3.4-6 Diagrama de clase de CircleGenerator	64
Ilustración 3.4-7 Flujo de Usuario en la Pantalla Principal.....	69
Ilustración 3.4-8 Diagrama de Estados del Flujo de Navegación del TMT.....	74

Ilustración 3.4-9 Diagrama de clase TmtMetricController	81
Ilustración 4.2-1 Boxplot de Distribución de Duración del TMT Parte A en la prueba piloto	111
Ilustración 4.2-2 Tabla 4.2 2 Estadísticas Errores de la Parte A del Test TMT en la prueba piloto	112
Ilustración 4.2-3 Boxplot de Distribución de Errores del TMT Parte A en la prueba piloto	112
Ilustración 4.2-4 Boxplot de Distribución de Duración del TMT Parte B en la prueba piloto	112
Ilustración 4.2-5 Tabla 4.2 3 Estadísticas errores de la Parte B del Test TMT en la prueba piloto	113
Ilustración 4.2-6 Boxplot de Distribución de Errores del TMT Parte B en la prueba piloto	113
Ilustración 4.2-7 Boxplot Comparación de las duraciones Test TMT en la prueba piloto	114

Índice de tablas

Tabla 1.4-1 Tiempo medio general de test TMT entre los 55 y los 98 años	17
Tabla 1.4-2 Tiempo medio de test TMT en la población ecuatoriana	17
Tabla 3.2-1 Product Backlog	34
Tabla 3.2-2 Cronograma del Proyecto	35
Tabla 3.4-1 Explicación de los atributos de la entidad Paciente	56
Tabla 3.4-2 Explicación de los atributos de la entidad Test TMT	56
Tabla 3.4-3 Diferentes respuestas del endpoint de Envío de Resultados del Test TMT	60
Tabla 4.1-1 Caso de Prueba 1: Crear Usuario	88
Tabla 4.1-2 Caso de Prueba 2: Crear Usuario con Nickname Duplicado	89
Tabla 4.1-3 Caso de Prueba 3: Eliminar Usuario (Último Perfil)	90
Tabla 4.1-4 Caso de Prueba 4: Eliminar Usuario (Múltiples Perfiles)	91
Tabla 4.1-5 Caso de Prueba 5: Cancelar Eliminación de Usuario	92
Tabla 4.1-6 Caso de Prueba 6: Visualizar Lista de Usuarios	93
Tabla 4.1-7 Caso de Prueba 7: Visualizar Usuario Vacío	94
Tabla 4.1-8 Caso de Prueba 1: Validar Código de Referencia Válido (Sin Usos Previos)	96
Tabla 4.1-9 Caso de Prueba 2: Validar Código de Referencia Inválido (Incorrecto)	97
Tabla 4.1-10 Caso de Prueba 3: Validar Código de Referencia Inválido (Ya Utilizado Dos Veces)	98
Tabla 4.1-11 Caso de Prueba 4: Validar Código de Referencia Válido (Solo Mano Izquierda Utilizada - Obligar Mano Derecha)	99
Tabla 4.1-12 Caso de Prueba 6: Validar Código de Referencia Válido (Solo Mano Derecha Utilizada - Obligar Mano Izquierda)	101
Tabla 4.1-13 Caso de Prueba 1: Envío Exitoso de Resultados de Métricas	103
Tabla 4.1-14 Caso de Prueba 2: Envío con Fallo de Conexión y Reintento Exitoso ..	104
Tabla 4.1-15 Caso de Prueba 1: Flujo Completo de Navegación del Test TMT	107
Tabla 4.1-16 Caso de Prueba 13: Funcionalidad de Ayuda Durante Test	109
Tabla 4.2-1 Muestras de resultados del Test TMT en la prueba piloto	111
Tabla 4.2-2 Estadísticas duración de la Parte A del Test TMT en la prueba piloto	111

Tabla 4.2-3 Estadísticas duración de la Parte B del Test TMT en la prueba piloto....	112
Tabla 4.2-4 Muestras de cuestionarios después del Test TMT en la prueba piloto ...	116
Tabla 4.3-1 Coste de cada actividad.....	118
Tabla 4.3-2 Referencia del puesto para cada actividad.....	119
Tabla 4.3-3 Coste de los recursos físicos	119
Tabla 4.3-4 Presupuesto del proyecto	119
Tabla 5.3-1 Casos de Respuesta del Endpoint de Validación de Código de Referencia	131
Tabla 5.3-2 Parámetros a enviar al endpoint de Envío de Resultados del Test TMT	136

1 Introducción

El Trail Making Test (TMT) es una prueba neuropsicológica ampliamente utilizada para evaluar la función cognitiva, específicamente para analizar la atención visual, la velocidad de procesamiento, la flexibilidad mental y la función ejecutiva.

La forma tradicional de realizar esta prueba es utilizar lápiz y papel bajo la supervisión de un neurólogo.

Sin embargo, el método tradicional presenta diversas limitaciones significativas:

1. **Supervisión continua:** Requiere la vigilancia directa de neurólogos durante la realización de la prueba para dar instrucciones e indicar si el proceso o el orden que está realizando el paciente es correcto o no. Esto causa una pérdida de tiempo, ya que al mismo tiempo solo se puede atender a un único paciente.
2. **Limitaciones métricas y errores de registros^[1]:** Cuando se realiza en papel, las métricas que se pueden obtener del test TMT son muy limitadas, restringiéndose únicamente al número de errores y al tiempo de la parte A y la parte B. Además, dependiendo del estado de los neurólogos, estos datos pueden ser malinterpretados; por ejemplo, pueden contar erróneamente el número de errores o no pausar el cronómetro en el momento adecuado, lo que podría afectar la precisión de los tiempos.
3. **Efecto de aprendizaje^[2]:** Como el test TMT en papel siempre tiene el mismo patrón y distribución de círculos esta situación hace que sea difícil investigar experimentalmente los procesos cognitivos que subyacen a su ejecución. Además, si un paciente lo realiza varias veces, al final puede memorizar el recorrido y el orden de los círculos.

Las investigaciones han demostrado que existen efectos significativos en la práctica con la administración repetida de las formas existentes, que son evidentes incluso tras dos exposiciones en un mismo día y detectables hasta un año después de la administración inicial.

Después de la primera administración, el sujeto aprende los trucos y entresijos de la prueba, la novedad desaparece, y es capaz de refinar sus estrategias y mejorar así los resultados en la prueba. De esta manera, se afecta la validez de la prueba.

4. **Desperdicio de recursos:** La realización de la prueba en formato físico implica el consumo de papel y tinta por cada aplicación, lo que genera un coste y un impacto ambiental.

1.1 Objetivos

El objetivo principal de este proyecto es, basándose en todos los inconvenientes del Trail Making Test de manera tradicional observados anteriormente, y con la colaboración del Instituto de Investigación Sanitaria y la Fundación de Investigación Biomédica del Hospital Universitario de Getafe, transformar el Trail Making Test de formato papel a formato digital.

Esto se realizará mediante una aplicación que se desarrollará en Flutter, dado que es una tecnología avanzada y multiplataforma, compatible tanto con Android como con iOS. De esta manera, se permitirá la realización del test TMT en dispositivos móviles o tabletas, independientemente del sistema operativo.

Una vez creada la aplicación, los neurólogos recolectarán las métricas durante la realización de los tests TMT. Al principio, mantendrán la prueba en papel para poder comparar los resultados con las pruebas realizadas de manera digital.

Posteriormente, estos datos serán calibrados. A partir de estos datos, se analizarán y evaluarán los factores relacionados con la cognición.

Objetivos específicos:

1. Generación test TMT aleatoria:

Un objetivo fundamental de este proyecto es crear un algoritmo que genere disposiciones aleatorias de círculos para cada prueba del test TMT. De esta manera, se supera el efecto de aprendizaje que presenta la versión en papel. También se implementará un sistema de feedback visual en tiempo real que indique al paciente si el orden seguido es correcto o incorrecto, proporcionando una guía instantánea durante la prueba.

2. Diseño adaptativo para múltiples dispositivos:

Para garantizar una amplia accesibilidad y una experiencia de usuario consistente, la aplicación se desarrollará con un diseño adaptativo. Se crearán pantallas e interfaces responsivas que funcionen y se visualicen adecuadamente tanto en móviles como en tabletas. Además, se optimizará la funcionalidad para distintas orientaciones de pantalla (vertical y horizontal) y se asegurará la correcta visualización en diferentes tamaños y resoluciones.

3. Captura y análisis de métricas avanzadas:

La aplicación irá más allá de la evaluación tradicional al recolectar un conjunto de métricas avanzadas durante la ejecución del test. Estas métricas incluyen el número de errores, la duración media de las pausas y de las elevaciones del dedo, la velocidad y el tiempo medio del trazo tanto entre los círculos como dentro de ellos, la presión media aplicada sobre la pantalla y el tamaño medio del área de contacto. Para gestionar estos datos, se desarrollará un sistema de transmisión seguro para enviar toda esta información a un servidor donde podrá ser analizada por los especialistas.

4. Soporte múltiples perfiles:

Finalmente, con el fin de que la aplicación pueda ser utilizada por varias personas en un único dispositivo, se implementará un sistema de soporte para múltiples perfiles. Esto permitirá el registro de distintos usuarios, facilitando la administración independiente de las pruebas para cada uno de ellos. Adicionalmente, se incluirá un sistema de historial individualizado que permitirá a cada usuario consultar sus resultados previos y hacer un seguimiento de su progreso a lo largo del tiempo.

1.2 La esclerosis múltiple

La esclerosis múltiple (EM) es la enfermedad neurológica crónica más común entre los adultos jóvenes, aunque también puede diagnosticarse en menores de 18 años. Afecta a más mujeres que a hombres y, en la mayoría de los casos, se diagnostica entre los 20 y los 40 años, un periodo crucial de desarrollo personal y laboral. Actualmente, la EM impacta a más de 55.000 personas en España y a 2,9 millones en todo el mundo, con un nuevo caso diagnosticado cada cinco minutos.^[3]

Aunque la causa exacta de la EM es desconocida, la hipótesis más aceptada es que se trata de una enfermedad autoinmune. Se caracteriza por episodios de disfunción neurológica que están "diseminados en el espacio y el tiempo", lo que significa que afectan diferentes áreas del sistema nervioso central (SNC) en distintos momentos. Esto da lugar a una amplia variedad de síntomas, dependiendo de la zona del SNC que esté comprometida.^[4]

Cursos Clínicos de la EM

La EM se presenta de diversas formas:

- **Forma en brotes o remitente-recurrente (EMRR):** Aproximadamente el 90% de los pacientes experimentan episodios o "brotes" de disfunción neurológica que son más o menos reversibles. Estos brotes se repiten con el tiempo y, a medida que lo hacen, pueden dejar secuelas funcionales neurológicas.
- **Forma secundariamente progresiva (EMSP):** Tras unos 10 años, cerca del 50% de los pacientes con EMRR evolucionan a un curso progresivo de la enfermedad.
- **Forma progresiva primaria (EMPP):** Un 10% de los pacientes muestran un curso progresivo de la enfermedad desde el inicio.
- **Forma Progresiva Recurrente (EMPR):** Es una forma atípica, en la que hay progresión desde el comienzo, pero a diferencia de los pacientes con EMPP, estos muestran brotes agudos claros, con o sin recuperación completa. Los periodos entre brotes se caracterizan por una progresión continua.

En el siguiente gráfico se muestran las variantes de evolución de la Esclerosis Múltiple^[5]:

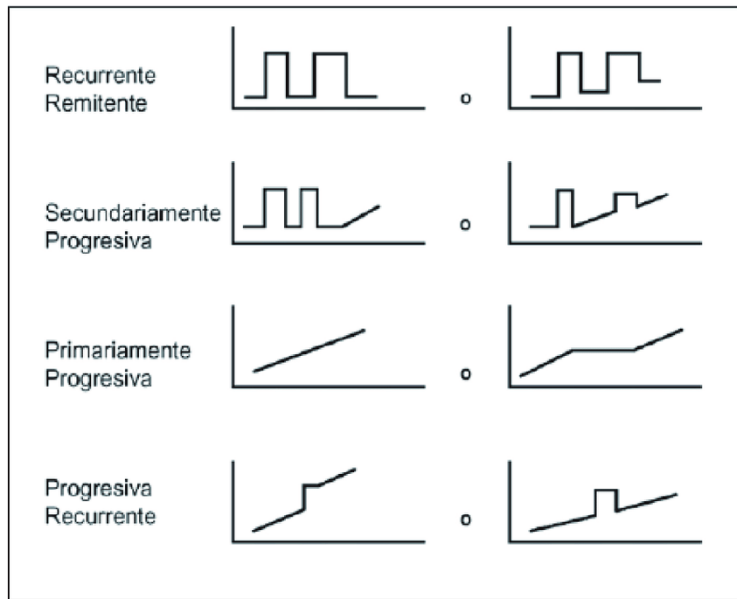


Ilustración 1.2-1 Variantes de evolución de la Esclerosis Múltiple según diagramas de Lublin

Síntomas y Características de la Esclerosis Múltiple ^[6]

La EM se origina por la degradación progresiva de la mielina, la capa protectora de las fibras nerviosas, lo que interrumpe la comunicación entre el cerebro y el resto del cuerpo. Esta condición conduce a una variedad de síntomas, entre los que destaca la falta de equilibrio.

Además de los problemas de equilibrio, los pacientes con EM pueden experimentar:

- **Espasticidad:** Rigidez muscular y espasmos. El tratamiento para este síntoma a menudo requiere fisioterapia y medicamentos como baclofeno o gabapentina.
- **Incontinencia urinaria:** Se puede tratar con medicamentos anticolinérgicos, antidepressivos tricíclicos o autocateterismo intermitente.
- **Fatiga:** Un síntoma común que afecta significativamente la calidad de vida.
- **Problemas de movilidad:** Dificultad para caminar, inestabilidad postural y riesgo de caídas.
- **Deterioro cognitivo:** Afectación de las funciones cognitivas.
- **Funciones respiratorias disminuidas.**

1.3 El Trail Making Test

El Test de Trazados (TMT, por sus siglas en inglés "Trail Making Test") es una prueba neuropsicológica diseñada para evaluar varias funciones cognitivas importantes.

Utilizado ya desde los años 1940 por la batería individual del ejército (Army Individual Test Battery, 1944), este test fue redescubierto como un indicador del daño cerebral en adultos y en niños por Ralph Reitan (Reitan, 1958; Reitan, 1971), que lo incluyó en la

batería diagnóstica de Halstead-Reitan. Spreen y Strauss (1998), mencionaron que el verdadero autor del test fue en realidad John Partington, que lo diseñó en 1938 con el nombre de "Senderos de Partington" o "Partington's Pathways". El test apareció unos años más tarde como un trabajo en común de Partington y Leiter (1949) Recientemente, Albert et al. (2001) han mostrado que el TMT predice la evolución de una demencia hacia formas de mayor severidad, ya que refleja el déficit del control y la planificación ejecutiva que subyace a la pérdida general de la memoria.^[7]

El TMT evalúa la atención sostenida (Parte A) y la función ejecutiva (Parte B), y se presenta en dos versiones adaptadas para diferentes rangos de edad.

Para adultos y adolescentes mayores de 15 años:^[8]

- **Parte A:** Se le pide al participante que conecte 25 números en orden secuencial (1, 2, 3, ..., 25). Esta tarea debe completarse en menos de cinco minutos y evalúa principalmente la velocidad de procesamiento visual y motor.
- **Parte B:** Esta sección incrementa la complejidad al requerir que el participante conecte números (1 hasta 13) y letras (A hasta L) de manera alterna (1-A, 2-B, 3-C, ..., 13-L), también en un tiempo máximo de cinco minutos. Evalúa la flexibilidad cognitiva y la capacidad para alternar entre diferentes conjuntos de estímulos.

Para niños de 4 a 14 años:^[9]

- **Parte A:** Al participante solo se le pide que conecte 15 números, adaptando así la prueba a las capacidades de los niños más pequeños.
- **Parte B:** Se le solicita que conecte 8 números (1 hasta 8) y 8 letras (A hasta H) de manera alterna, diseñado para evaluar la capacidad de los niños para cambiar entre tareas simples de numeración y alfabetización.

Importancia Clínica y Aplicaciones del TMT:^[10]

El TMT es utilizado en diversos contextos clínicos para identificar disfunciones cerebrales y evaluar el impacto de diversas condiciones médicas sobre las capacidades cognitivas. Es capaz de medir dominios cognitivos como la velocidad de procesamiento, la secuenciación, la flexibilidad mental y las habilidades visomotoras lo hace relevante en múltiples escenarios.

Por ejemplo, es eficaz para detectar problemas en pacientes con lesiones cerebrales, enfermedades neurodegenerativas como la enfermedad de Alzheimer y otros trastornos cognitivos:

- **Lesión Cerebral Traumática (LCT):** Se utiliza como parte de una batería de pruebas uno de los indicadores pronósticos válidos de la recuperación funcional después de una LCT. Es muy sensible al grado de lesión.
- **Enfermedad de Alzheimer:** Mediante el análisis de los errores se pueden detectar problemas de desinhibición. Además, la parte A del TMT puede revelar

los efectos farmacológicos sobre la corteza temporal y parietal en esta enfermedad.

- **Esquizofrenia:** Con la parte A del TMT, se pueden detectar deficiencias en la velocidad de procesamiento y, con la parte B, se detectan estrategias ineficientes de procesamiento secuencial. Analizando estos datos, se puede reportar un deterioro severo en (el contexto de) la esquizofrenia.
- **Trastorno Bipolar:** Según el estudio, en pacientes con trastorno bipolar I y II se han encontrado deficiencias moderadas en ambas partes del test TMT. Con un déficit de referencia en la parte B, los pacientes con depresión bipolar obtienen peores resultados que aquellos considerados normales. Por lo tanto, test TMT puede ayudar a detectar si existen síntomas de trastorno bipolar.
- **Evaluación de Habilidades Funcionales:** La combinación de las medidas del test TMT puede ser una herramienta útil. La puntuación B-A puede ser significativa para detectar la enfermedad de Parkinson.
- **Predicción del Potencial Vocacional:** La velocidad de procesamiento, medida de forma robusta por el TMT (especialmente la Parte A), ha demostrado ser un potente predictor del potencial vocacional.

1.4 Método de Evaluación Test TMT

Como el criterio del TMT puede variar mucho según el país o los idiomas que se hablan, en este caso solo nos vamos a centrar en los criterios de España.

En el Test de Trazados (TMT), los resultados se miden en segundos que el individuo tarda en completar la Parte A o la Parte B. Basándose en estos tiempos, se utilizan valores percentiles para evaluar el desempeño del individuo en comparación con una población de referencia. Los valores percentiles reflejan la posición relativa del resultado de un individuo dentro de una distribución normal.

Por ejemplo, un percentil de 50 significa que el rendimiento del individuo está en el promedio de la población de referencia. Si un paciente obtiene un percentil de 75, esto indica que su rendimiento es superior al del 75% de la población normativa.

Parte B-A:

la diferencia entre el tiempo necesario para completar la Parte B y el tiempo necesario para completar la Parte A.

Al restar la influencia de la velocidad psicomotora básica y la velocidad de búsqueda visual de la Parte A, se puede medir de manera más precisa las demandas cognitivas adicionales en la Parte B

Parte B/A

Al calcular la proporción de las puntuaciones de las dos partes, se estandariza el rendimiento de la Parte B en relación con la Parte A, proporcionando un indicador

relativo que refleja el aumento relativo de la dificultad de la Parte B en comparación con la Parte A.

1.4.1 Tiempo promedio

En general el tiempo promedio para TMT-A es de 29 segundos y una puntuación deficiente es mayor de 78 segundos. Para TMT-B, una puntuación promedio es de 75 segundos y una puntuación deficiente es mayor de 273 segundos.^[11]

El tiempo promedio para completar el test TMT puede variar según la edad y género.

La siguiente tabla muestra el tiempo medio general para edades comprendidas entre los 55 y los 98 años:^[12]

Prueba y Grupo de Edad	Promedio	Deficiente
TMT Parte A (edades 55-75)	Hasta 42 segundos	Más de 70 segundos
TMT Parte A (edades 75-98)	Hasta 51 segundos	Más de 79 segundos
TMT Parte B (edades 55-75)	Hasta 101 segundos	Más de 273 segundos
TMT Parte B (edades 75-98)	Hasta 128 segundos	Más de 273 segundos

Tabla 1.4-1 Tiempo medio general de test TMT entre los 55 y los 98 años

La siguiente tabla muestra el promedio del test TMT en la población ecuatoriana, según su edad y género: ^[13]

Rangos de Edad	TMT Parte A	TMT Parte B
Por Género		
Masculino	33,7	76,3
Femenino	34,4	79,1
Por Edad		
17 a 30 años	33,8	78,2
31 a 50 años	34,2	76,5
51 a 64 años	35,5	80,8
65 años o más	32,7	87,8

Tabla 1.4-2 Tiempo medio de test TMT en la población ecuatoriana

1.4.2 Normativas para calcular el percentil del resultado del test TMT

Dependiendo de la edad y el género, se puede calibrar el tiempo de la prueba TMT (Trail Making Test) para luego obtener su percentil. El siguiente ejemplo demuestra cómo sacar el percentil para una persona mayor de 55 años, ya que las personas mayores suelen tener problemas con enfermedades cognitivas.

Para personas de 55 años y más, se aplican los criterios y valores normativos establecidos en el estudio titulado "The Trail Making Test: Association With Other Neuropsychological Measures and Normative Values for Adults Aged 55 Years and Older From a Spanish-Speaking Population-Based Sample". [14]

Este estudio ha proporcionado una base de datos normativos específica para la población española mayor de 55 años.

Para calcular el percentil de sus resultados, siga estos pasos:

Paso 1: Cálculo del valor predicho del log(TMT-A o TMT-B)

$$\begin{aligned} \log(\text{TMT-A o TMT-B})_{\text{predicho}} &= \text{Constant} + (\text{Sex} \times \text{Género}) + (\text{Age} \times \text{Edad}) \\ &+ (\text{LE high} \times \text{Educación Alta}) + (\text{LE low} \times \text{Educación Baja}) \end{aligned}$$

Constant: Según la siguiente tabla (Regresión Lineal Múltiple por Pasos de Sexo, Edad y Educación sobre Puntuaciones TMT Transformadas Logarítmicamente), el valor constante del coeficiente B para TMT-A es 3.758, mientras que para TMT-B es 4.536.

Sex: Según la siguiente tabla (Regresión Lineal Múltiple por Pasos de Sexo, Edad y Educación sobre Puntuaciones TMT Transformadas Logarítmicamente), el valor Sex para TMT-A es 0,051, mientras que para TMT-B es 0,060.

Género: Codificado como binario: masculino = 0, femenino = 1.

Age: Según la siguiente tabla (Regresión Lineal Múltiple por Pasos de Sexo, Edad y Educación sobre Puntuaciones TMT Transformadas Logarítmicamente), el valor Age para TMT-A es 0,017, mientras que para TMT-B es 0,018

Edad: Edad real en años menos la edad media de la muestra (67 años).

LE high: Según la siguiente tabla (Regresión Lineal Múltiple por Pasos de Sexo, Edad y Educación sobre Puntuaciones TMT Transformadas Logarítmicamente), el valor Age para TMT-A es -0,044, mientras que para TMT-B es -0,118.

Educación Alta: Indicador binario: nivel de educación alta = 1, de lo contrario = 0. (Se refiere a personas que tienen más de 12 años de educación formal)

LE low: Según la siguiente tabla, el valor Age para TMT-A es 0,226, mientras que para TMT-B es 0.3.

Educación Baja: Indicador binario: nivel de educación baja = 1, de lo contrario = 0. (se refiere a personas que tienen hasta 8 años de educación forma)

Educación Media, corresponde a personas que tienen entre 9 y 12 años de educación (como haber completado la educación secundaria). Para este grupo, ambos indicadores "Educación Alta" y "Educación Baja" se establecen en 0.

Paso 2: Cálculo del logaritmo del tiempo real

$$\log(\text{TMT-A ó TMT-B})_{\text{real}} = \log(\text{tiempo real})$$

Tiempo real: tiempo real observado para realizar el TMT-A o TMT-B en segundos.

Paso 3: Calcular el residuo:

$$e_i = \log(\text{TMT-A ó TMT-B})_{\text{real}} - \log(\text{TMT-A o TMT-B})_{\text{predicho}}$$

e_i : Valor residual

Paso 4: Obtener puntaje z estandarizado

$$z_i = \frac{e_i}{SD_e}$$

z_i : puntaje z estandarizado

e_i : Valor residual

SD_e : De acuerdo con el valor del $\log(\text{TMT-A o TMT-B})_{\text{predicho}}$ consulta la tabla (Desviaciones estándar de los residuos para las puntuaciones Log TMT predichas) para obtener las desviaciones estándar de los residuos de las puntuaciones Log TMT predichas.

Paso 5: Convertir a percentiles

Convierte el puntaje z en percentiles.

Para ello hay que utilizar Tabla de distribución normal estándar (tabla Z)

Ó utilizar esta fórmula

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$$

Ejemplo en el artículo:

- **Género:** Masculino (0)
- **Edad:** 75
- **Nivel de Educación:** Educación alta (Alta = 1, Baja = 0)
- **Tiempo real hacer test TMT-A :** 61 segundos
-

$$\begin{aligned} \log(\text{TMT-A})_{\text{predicho}} &= 3.758 + (0.051 \times 0) + (0.017 \times 8) + (-0.044 \times 1) + (0.226 \times 0) \\ &= 3.85 \end{aligned}$$

$$\log(\text{TMT-A})_{\text{real}} = \log(61) \approx 4.110$$

$$e_i = 4.110 - 3.850 = 0.261$$

$$Z_i = \frac{0.261}{0.206} \approx -1.265$$

En este ejemplo, la puntuación z (probabilidad) de -1.265 corresponde al percentil 10%

Measure	Variable	β	SE(β)	t	Sd β	ΔR^2
Log TMT-A	Constant	3.758	0.016	—	—	0.284 ^a
	Age	0.017	0.001	18.482*	0.364	0.150
	Sex	0.051	0.015	3.484*	0.068	0.005
	LE low	0.226	0.017	13.151*	0.302	0.118
	LE high	-0.044	0.021	-2.087*	-0.048	0.011
Log TMT-B	Constant	4.536	0.017	—	—	0.369 [†]
	Age	0.018	0.001	19.131*	0.354	0.148
	Sex	0.060	0.015	3.909*	0.072	0.008
	LE low	0.300	0.018	16.767*	0.362	0.175
	LE high	-0.118	0.022	-5.345*	-0.115	0.038
Log TMT-d	Constant	3.830	0.029	—	—	0.226 ^a
	Age	0.020	0.002	11.940*	0.245	0.073
	Sex	0.079	0.026	2.996*	0.061	0.006
	LE low	0.368	0.031	11.860*	0.284	0.111
	LE high	-0.200	0.038	-5.207*	-0.124	0.036
Log TMT-r	Constant	0.779	0.013	—	—	0.039 ^a
	LE low	0.078	0.016	4.434*	0.127	0.016
	LE high	-0.075	0.020	-3.747*	-0.099	0.023

Note. TMT-A = Trail Making Test-A; TMT-B = Trail Making Test-B; TMT-d = Trail Making Test difference; TMT-r = Trail Making Test ratio; LE = level of education; β = regression coefficient; SE(β) = standard error of β ; Sd β = standardized regression coefficient; ΔR^2 = variance explained by variable (Pearson's correlation * Sd β).

[†]R² = coefficient of determination.

*p < .05.

Ilustración 1.4-1 Tabla: Regresión Lineal Múltiple por Pasos de Sexo, Edad y Educación sobre las Puntuaciones TMT Transformadas Logarítmicamente (extraída del artículo)

	Predicted score	SD (Residual)
Log TMT-A	≥4.128	0.422
	Between 4.127 and 3.871	0.206
	Between 3.870 and 3.637	0.201
Log TMT-B	≤3.636	0.379
	≥4.872	0.327
	Between 4.871 and 4.707	0.342
Log TMT-d	Between 4.706 and 4.462	0.327
	≤4.461	0.318
	≥4.237	0.542
Log TMT-r	Between 4.236 and 4.047	0.518
	Between 4.046 and 3.748	0.571
	≤3.747	0.588
Log TMT-r	≥1.005	0.415
	Between 1.004 and 0.796	0.123
	Between 0.795 and 0.593	0.136
	≤0.592	0.388

Note. TMT-A = Trail Making Test-A; TMT-B = Trail Making Test-B; TMT-d = Trail Making Test difference; TMT-r = Trail Making Test ratio.

Ilustración 1.4-2 Tabla Desviaciones estándar de los residuos para las puntuaciones Log TMT predichas (extraída del artículo)

2 Estado del arte

En este apartado se ha realizado un estudio de todas las aplicaciones digitales de test TMT que existen en el mercado, con el fin de tomarlas como referencia y mejorar al respecto.

2.1.1 En Google Play Store

TMT App^[15]

La aplicación está bastante desactualizada; la última actualización fue en febrero de 2020. Es gratuita, lo cual es un punto a favor para accesibilidad general. Sin embargo, la interfaz no es amigable y es muy simple, lo que podría dificultar su uso. Además, la prueba no se genera de forma aleatoria, sino que sigue siempre el mismo formato. El texto siempre está en inglés.

En cuanto a los resultados, la aplicación solo informa los segundos que el usuario ha tardado en completar el TMT-A o TMT-B. No solicita información sobre el nivel educativo del usuario, lo cual podría ser relevante para interpretar los resultados.

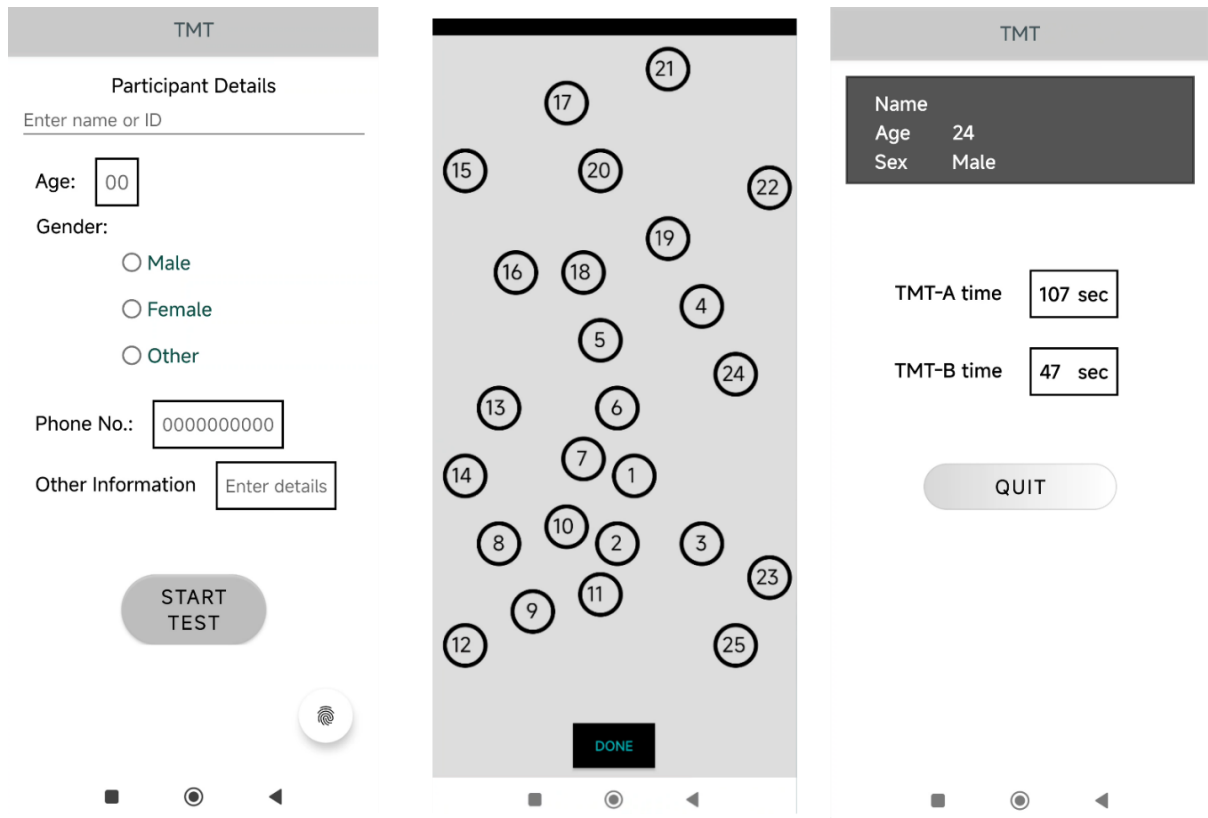


Ilustración 1.4-1: Captura de pantalla de la aplicación TMT APP

Trail Making Test^[16]

La aplicación es gratuita, pero no es compatible con las últimas versiones de Android, lo que ha requerido que se pruebe en un emulador de Android 8. Esto significa que actualmente no se puede instalar en dispositivos con sistemas operativos más recientes.

Además, la aplicación no proporciona mensajes de error claros ni instrucciones sobre cómo resolverlos.

No solicita información sobre la edad ni el género del usuario; el resultado solo se muestra los segundos que el usuario ha tardado en completar las partes TMT-A o TMT-B.

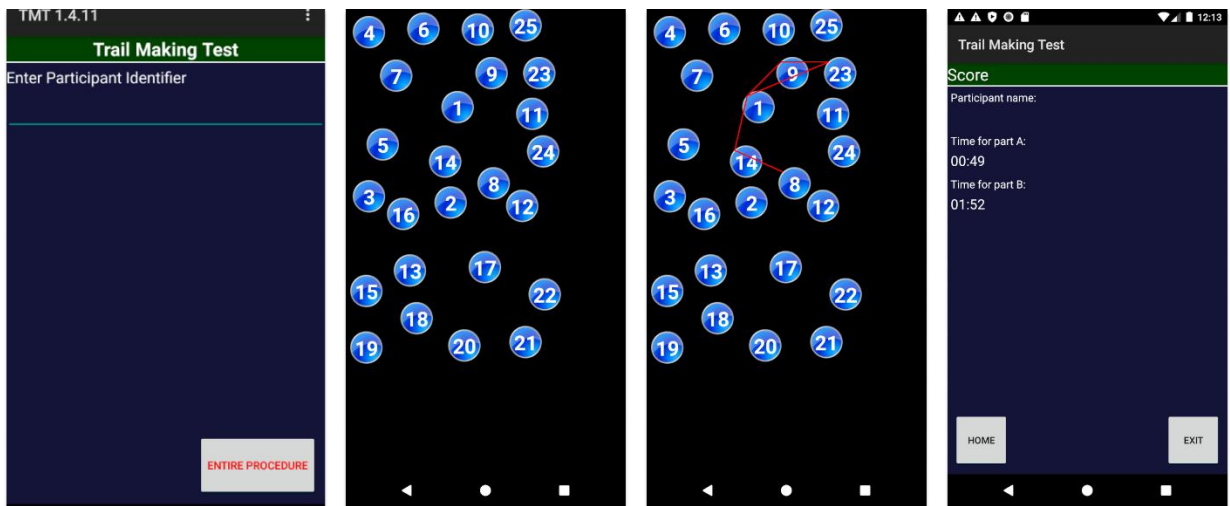


Ilustración 1.4-2 Captura de pantalla de la aplicación Trail Making Test

2.1.2 En App Store

Trail Making Test ^[17]

La aplicación está disponible únicamente para iPad. Es gratuito.

Aunque genera las secuencias de manera aleatoria, en el test TMT-A solo incluye números hasta el 8, y en el TMT-B se limita a 4 números y 4 letras. Además, solo permite realizar el TMT-A o el TMT-B por separado; no es posible hacerlos sucesivamente uno tras otro.

En caso de error al conectar al siguiente número o letra, la aplicación no proporciona ningún tipo de mensaje de error.

Los resultados se presentan solo en segundos.

Además, la aplicación no solicita información sobre la edad, el sexo, ni los años de educación.

Aunque la aplicación ofrece la funcionalidad de guardar y exportar los resultados a lo largo del tiempo, parece que no está funcionando correctamente.

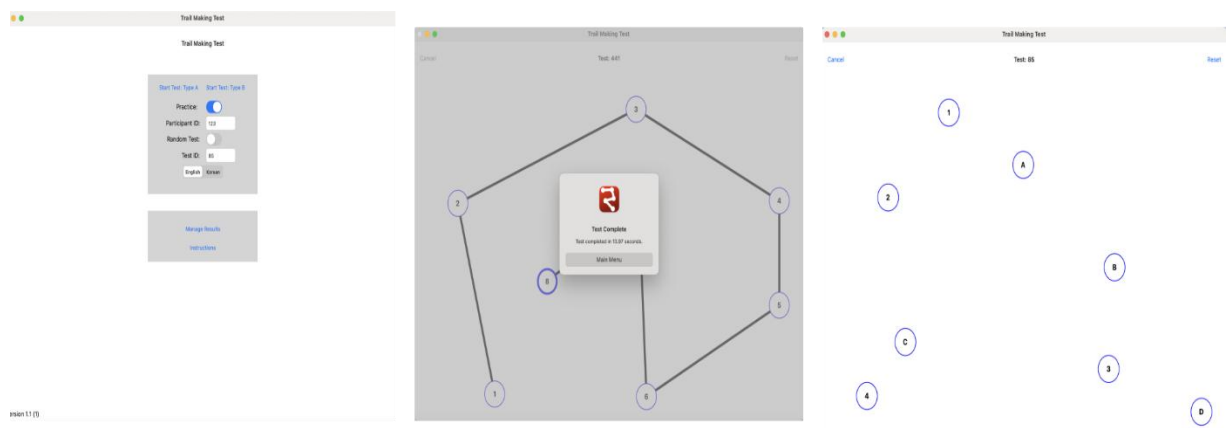


Ilustración 1.4-3 Captura de pantalla de la aplicación Trail Making Test

Trail Making Test J Lite^[18]

Aunque la aplicación es gratuita, pero contiene publicidad. Para eliminar la publicidad, es necesario realizar un upgrade que cuesta 1,99€.

Los números y letras no están dentro de círculos, lo cual se desvía del formato tradicional del Trail Making Test. Cuando la respuesta es correcta, la aplicación reproduce un sonido, y también avisa con un sonido cuando hay un error.

Los resultados muestran solo el tiempo en segundos que se necesita para completar la prueba, también indica el número de errores cometidos.

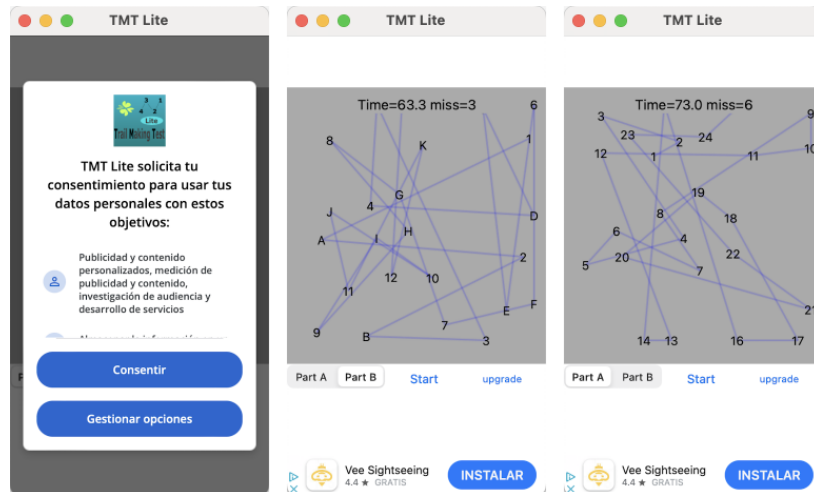


Ilustración 1.4-4 Captura de pantalla de la aplicación Trail Making Test J Lite

NeuRA Trail making test^[19]

No se ha podido realizar la prueba ya que es de pago; el costo es de 59,99€ Según las capturas de pantalla disponibles, la aplicación sí solicita la edad del usuario. Además, proporciona información sobre el tiempo medio y los tiempos deficientes para las pruebas TMT-A y TMT-B.

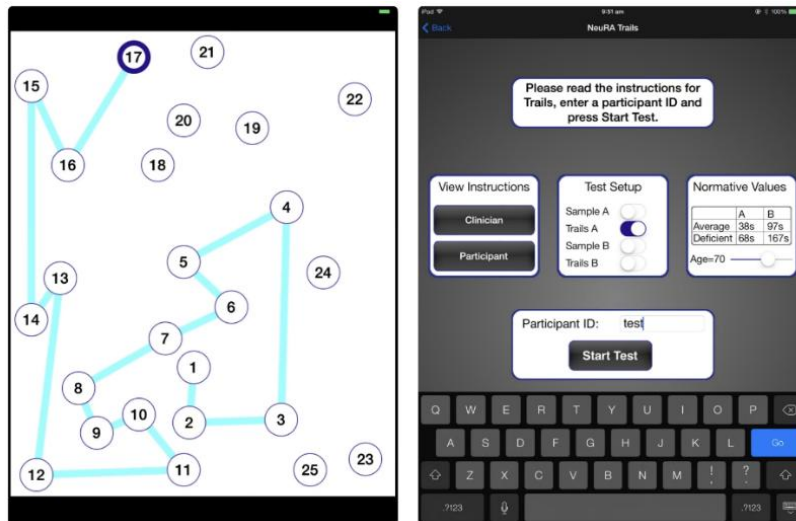


Ilustración 1.4-5 Captura NeuRA Traid Making test extraída del App Store

2.1.3 Aplicación Web

CogniFit Trail Making Test^[20]

Es gratuita, no requiere registro para realizar las pruebas, lo que facilita su acceso y uso inmediato.

Incluye un tutorial de YouTube que enseña cómo realizar los tests, lo que puede ser muy útil para los usuarios nuevos.

Sin embargo, una vez que se sale de la página, es necesario empezar la prueba desde cero, lo cual puede ser inconveniente para los usuarios.

Aunque la aplicación toma en cuenta el género y la edad del usuario, solo es adecuada para personas de 16 a 110 años, por lo que no está diseñada para niños.

Proporciona un percentil, pero este es un resultado conjunto de ambas pruebas (TMT-A y TMT-B) y no ofrece resultados separados para cada una; lo mismo ocurre con los segundos que se reportan como un total combinado.

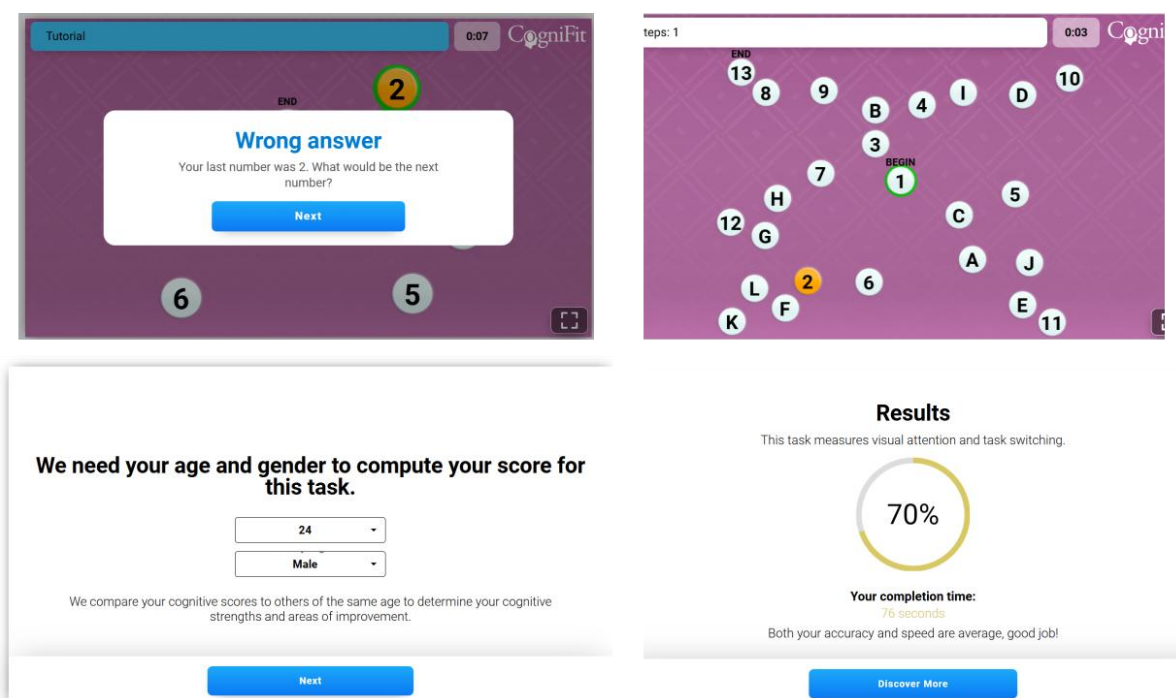


Ilustración 1.4-6 Capturas de CogniFit Trail Making Test

2.1.4 Resultados de la revisión

Basado en las ideas propuestas y el análisis de las aplicaciones existentes, se puede concluir que hay una clara oportunidad de mercado para una aplicación de Trail Making Test que supere las deficiencias de las soluciones actuales. Las aplicaciones existentes presentan limitaciones significativas como:

1. **Falta de personalización:** La mayoría no adapta las pruebas según la edad del usuario o su nivel de experiencia con este tipo de test.
2. **Interpretación limitada de resultados:** Se limitan a mostrar el tiempo en segundos sin contextualizarlo mediante más datos métricas.
3. **Experiencia de usuario deficiente:** Carecen de tutoriales adecuados o modos de práctica para usuarios nuevos.
4. **Generación estática de pruebas:** Utilizan siempre las mismas rutas de números y letras, lo que limita su utilidad para evaluaciones repetidas.
5. **Ausencia de análisis histórico:** No ofrecen herramientas para visualizar el progreso a través del tiempo mediante gráficos.

3 Desarrollo del Proyecto

Para desarrollar este proyecto se ha aplicado una metodología híbrida, con la combinación de elementos de Agile y del Modelo en Cascada.

En el proyecto general se ha seguido el modelo en cascada de la siguiente manera: primero, el análisis de requisitos; luego, el diseño, la implementación, las pruebas, el despliegue y los trabajos futuros (posible mantenimiento)

Adicionalmente, se incorporó elementos de Agile, como el Product Backlog estructurado en épicas e historias de usuario para enumerar y definir las tareas de desarrollo, implementación iterativa por funcionalidades independientes usando Clean Architecture.

3.1 Análisis de requisitos

Antes de comenzar, es imprescindible extraer todos los requisitos de usuario para tener claridad sobre el proyecto.

Se utilizaron técnicas de extracción/obtención de requisitos mediante análisis documental y entrevistas estructuradas.

Tras realizar varias reuniones y entrevistas con Joaquín Bienvenido Ordieres Mere, quien se encarga de comunicarse con los neurólogos del Hospital La Paz, se han extraído los siguientes requisitos:

3.1.1 Requisitos funcionales

RU-1. Como usuario quiero poder validar un código de referencia para acceder a la prueba neurológico

- RF-1.1 El sistema deberá permitir al usuario ingresar un código de referencia proporcionado por su neurólogo.
- RF-1.2 El sistema deberá validar que el código de referencia ingresado sea válido antes de permitir el acceso.
- RF-1.3 El sistema deberá comprobar si el código de referencia ya ha sido utilizado previamente.
- RF-1.4 El sistema deberá informar al usuario si el código de referencia es inválido o ya ha sido utilizado.
- RF-1.5 El sistema deberá permitir modificar el código de referencia en caso de error de escritura.

RU-2. Como usuario quiero poder crear y gestionar mi perfil de usuario para acceder a mis datos e historial.

- RF-2.1 El sistema deberá permitir al usuario crear un nuevo perfil con datos básicos (sexo, fecha de nacimiento, nivel de estudios).
- RF-2.2 El sistema deberá permitir al usuario eliminar su perfil cuando lo desee.

RF-2.3 El sistema deberá garantizar que los datos del perfil se guarden localmente en el dispositivo.

RF-2.4 El sistema deberá asociar los resultados de los tests realizados con el perfil correspondiente.

RU-3. Como usuario quiero gestionar múltiples perfiles en un mismo dispositivo para que varias personas puedan usar la aplicación.

RF-3.1 El sistema deberá permitir crear múltiples perfiles en un mismo dispositivo.

RF-3.2 El sistema deberá mostrar una lista de perfiles disponibles para seleccionar.

RF-3.3 El sistema deberá permitir eliminar perfiles desde la lista de usuarios.

RF-3.4 El sistema deberá cambiar automáticamente entre perfiles cuando el usuario los seleccione.

RU-4. Como usuario quiero poder seleccionar el idioma de la aplicación (inglés o español) para usarla en mi idioma preferido.

RF-4.1 El sistema deberá permitir al usuario cambiar entre inglés y español.

RF-4.2 El sistema deberá guardar la preferencia de idioma seleccionada.

RF-4.3 El sistema deberá aplicar el cambio de idioma inmediatamente sin necesidad de reiniciar la aplicación.

RF-4.4 El sistema deberá presentar todos los textos, mensajes y elementos de la interfaz en el idioma seleccionado.

RU-5. Como usuario quiero poder seleccionar entre 15 o 25 círculos para el test para adaptarlo a mis necesidades.

RF-5.1 El sistema deberá permitir al usuario seleccionar entre 15 y 25 círculos para realizar el test.

RF-5.2 El sistema deberá guardar esta preferencia para futuras sesiones.

RF-5.3 El sistema deberá adaptar el test de acuerdo con la cantidad de círculos seleccionada.

RU-6. Como usuario quiero poder realizar los Tests TMT Parte A y Parte B para evaluar mi capacidad cognitiva.

RF-6.1 El sistema deberá ser capaz de generar círculos solo con números (Parte A) o con números y letras (Parte B) en posiciones aleatorias en la pantalla de cada test TMT.

RF-6.2 El sistema deberá validar en tiempo real, durante la realización del test TMT, que el usuario conecte el orden correcto de números en la Parte A y, en la Parte B, cuando alterne números y letras.

- RF-6.3 El sistema deberá detectar y contabilizar el número de errores exclusivos durante la realización de la Parte A y la Parte B.
- RF-6.4 El sistema deberá mostrar una señal visual (X roja) sobre el círculo cuando el orden en que está conectado por el usuario sea incorrecto.
- RF-6.5 El sistema deberá indicar visualmente al usuario que debe regresar al último círculo correcto cuando, al intentar el siguiente círculo, su orden sea incorrecto.
- RF-6.6 El sistema deberá registrar el tiempo total empleado en completar cada parte del test(Parte A y ParteB).
- RF-6.7 El sistema deberá registrar la duración de las pausas realizadas durante la ejecución del test.
- RF-6.8 El sistema deberá detectar y registrar los levantamientos del dedo durante la realización del test.
- RF-6.9 El sistema deberá medir y registrar la velocidad de trazo cuando el usuario se desplaza entre círculos.
- RF-6.10 El sistema deberá medir y registrar la velocidad de trazo cuando la línea esté dentro de los círculos.
- RF-6.11 El sistema deberá registrar la presión aplicada sobre las pantallas de los dispositivos compatibles con esta funcionalidad durante la realización del test TMT.
- RF-6.12 El sistema deberá registrar los tiempos medios antes de que el usuario conecte al siguiente número para la Parte A del test TMT.
- RF-6.13 El sistema deberá registrar los tiempos medios antes de que el usuario conecte a la siguiente letra para la Parte B del test TMT.
- RF-6.14 El sistema deberá incluir una tolerancia de 5 píxeles alrededor de cada círculo. Cuando el trazo entre en dicho rango de tolerancia, se considerará que ya está conectado al círculo. De esta manera, se compensa la comparación con un bolígrafo real de forma tradicional.
- RF-6.15 El sistema deberá detectar pausas (cuando el usuario mantiene el dedo en la misma posición por más de 0.1 segundos) y levantamientos del dedo durante la realización de cualquiera de las partes del test.

RU-7. Como usuario quiero un modo de práctica para familiarizarme con el test antes de realizarlo formalmente.

- RF-7.1 El sistema deberá generar 9 círculos con solo números en el modo práctica de la parte TMT A y con números y letras en el modo práctica de la parte TMT B en posiciones aleatorias en la pantalla.
- RF-7.2 El sistema deberá validar que el usuario conecte los círculos en orden numérico correcto (Parte A) o alternando números y letras en orden correcto (Parte B).

RF-7.3 El sistema deberá mostrar una pantalla de ayuda antes de iniciar el modo práctica, donde se explique claramente cómo funcionar el test TMT, sus objetivos y la forma correcta de realizarlo.

RF-7.4 El sistema deberá permitir al usuario repetir el modo práctico tantas veces como desee antes de realizar el test formal.

RU-8. Como usuario quiero ver los resultados de mi test para conocer mi desempeño.

RF-8.1 El sistema deberá mostrar el tiempo empleado en cada parte del test (A y B) al finalizar.

RF-8.2 El sistema deberá mostrar el número de errores cometidos en cada parte del test.

RF-8.3 El sistema deberá mostrar un mensaje de agradecimiento después de presentar los resultados.

RU-9. Como usuario quiero que los resultados de mi test se envíen automáticamente para que mi médico pueda revisarlos.

RF-9.1 El sistema deberá enviar los resultados completos del test al servidor automáticamente.

RF-9.2 El sistema deberá almacenar temporalmente los resultados en caso de no tener conexión a internet.

RF-9.3 El sistema deberá reenviar los resultados almacenados cuando se restablezca la conexión.

RF-9.4 El sistema deberá registrar y enviar todas las métricas importantes (duración de pausas, levantamientos del dedo, velocidad de trazo entre círculos y dentro de círculos, presión aplicada, y tiempos específicos antes de números y letras).

RU-10. Como usuario quiero ver mi historial de tests realizados para hacer seguimiento de mi progreso.

RF-10.1 El sistema deberá almacenar localmente el historial de tests realizados por cada usuario.

RF-10.2 El sistema deberá mostrar el historial de tests en una pantalla específica.

RF-10.3 El sistema deberá mostrar la fecha, la referencia del test y los tiempos obtenidos en cada test histórico.

3.1.2 Requisitos no funcionales

Compatibilidad:

COM-1. La aplicación debe ser compatible con sistema operativo Android e iOS.

COM-2. La aplicación debe ser compatible con diferentes tamaños y resoluciones de pantalla para teléfonos móviles y tablets.

Extensibilidad:

EXT-1. La aplicación debe permitir la actualización de las respuestas de las APIs sin afectar al funcionamiento general.

Recuperabilidad:

REC-1. La aplicación debe almacenar los datos de test no enviados en caso de pérdida de conexión.

Seguridad:

SEG-1. El sistema no debe almacenar ningún dato sensible del usuario, como el nombre. Solo debe almacenar los datos de manera anónima, de modo que no se pueda identificar al usuario.

Integridad:

INT-1. El sistema debe validar todos los datos ingresados por el usuario para prevenir errores o manipulaciones.

Usabilidad:

USA-1. El diseño de la interfaz debe ser intuitivo y seguir patrones de diseño similares a la aplicación HealthyWear para mantener consistencia.

Accesibilidad:

ACC-1. La aplicación debe ofrecer suficiente contraste entre el texto y el fondo en ambos modos (claro y oscuro).

ACC-2. Los círculos en el test deben tener un tamaño mínimo para garantizar su visibilidad y facilidad de uso.

3.2 Planificación

Una vez definidos los requisitos y las tareas pendientes de realizar, se procedió a una planificación adecuada para determinar la cantidad de trabajo, su duración estimada y la secuencia de desarrollo.

Antes de planificar fue necesario identificar qué tareas se debían desarrollar y cómo organizarlas.

Para esto se utilizó una planificación basada en la priorización de tareas del Product Backlog, organizando el trabajo en épicas e historias de usuario con diferentes niveles de prioridad (Alta, Media, Baja).

3.2.1 Product Backlog

A partir de los requisitos anteriores, se ha creado la siguiente tabla de Product Backlog con el identificador de cada tarea que debería realizarse. De esta manera, en el futuro, al momento de crear una rama en Git, se podrá identificar fácilmente.

Funcionalidad	Épica (EP)	Historia de Usuario (HU)	Descripción	Prioridad
	EP-1.1 Base de proyecto	HU-1.1.1 Base de proyecto	Crear base inicial del proyecto	Alta
		HU-1.1.2 Multi lenguaje	Implementar soporte multilinguaje	Alta

Estructura Base		HU-1.1.3 Tema dinámica	Implementar tema dinámico (claro/oscur)	Alta
	EP-1.2 API de proyecto	HU-1.2.1 Crear base de gestionar API	Desarrollar estructura base para API	Alta
		HU-1.2.2 Crear API validar referencia	Añadir endpoint validación de códigos de referencia	Alta
		HU-1.2.3 Crear API de lista de formulario	Añadir endpoint para listar formularios	Alta
		HU-1.2.4 Crear API de enviar formulario	Añadir endpoint para envío de datos de formulario	Alta
Pantalla de TMT	EP-2.1 Patalla de TEST TMT	HU-2.1.1 Puede generar círculos aleatorios y conectar entre sí	Implementar algoritmo para colocar círculos aleatoriamente	Alta
		HU-2.1.2 Capaz de validar según orden	Implementar validación del orden correcto al conectar	Alta
		HU-2.1.3 Adaptar según tamaño de pantalla	Hacer que la interfaz sea responsive	Alta
		HU-2.1.4 Capaz de contar variables (errores, tiempo, etc.)	Implementar conteo de métricas del test	Alta
		HU-2.1.5 Adaptar pantalla de TEST TMT	Optimizar la visualización del test	Alta
	EP-2.2 Pantalla de Práctica y su Flujo	HU-2.2.1 Adaptar pantalla de TEST TMT	Adaptar la pantalla de test TMT para el modo práctica	Media
		HU-2.2.2 Diálogo de terminar (gestionar diferentes navegaciones, volver hacer test, volver pantalla anterior)	Gestionar diferentes opciones de navegación (volver a hacer test, volver a pantalla anterior) tras finalizar el test	Media

	EP-2.3 Pantalla de Ayuda	HU-2.3.1 Mostrar diferentes textos para TMTA/TMT-B	Mostrar instrucciones específicas según tipo de test	Media
		HU-2.3.2 Gestionar diferentes navegaciones	Adaptar navegación según contexto	Media
	EP-2.4 Pantalla de resultado	HU-2.4.1 Crear pantalla de resultado	Diseñar interfaz para mostrar resultados	Media
		HU-2.4.2 Vincular con API de enviar formulario	Integrar resultados con API de envío	Media
CRUD usuario	EP-3.1 Crear base de datos	HU-3.1.1 Crear Estructura de Base de datos	Implementar y crear tabla según esquema de base de datos para usuario	Media
		HU-3.1.2 Gestionar métodos de crear y eliminar	Implementar operaciones necesarias para crear y eliminar usuario	Media
	EP-3.2 Crear Usuario	HU-3.2.1 Pantalla de formulario de dar alta	Diseñar interfaz para registro de usuarios	Media
		HU-3.2.2 Vincular pantalla de dar alta con BBDD	Conectar interfaz con base de datos	Media
	EP-3.2 Borrar Usuario	HU-3.2.1 Adaptar Pantalla de formulario de dar alta para demostrar información del usuario	Mostrar datos del usuario seleccionado en la pantalla	Media
		HU-3.2.2 Adaptar Pantalla de formulario de dar alta cambiar botón para eliminar usuario	Modificar la interfaz para permitir eliminación de usuarios	Media
	EP-3.3 Mirar Historia	HU-3.3.1 BBDD demostrar	Implementar consulta de historial	Baja

		usuario con su Historia		
		HU-3.3.2 Pantalla de demostrar historia	Diseñar interfaz para visualizar historial	Baja
Home	EP-4.1 Pantalla de Home	EP-4.1.1 Botones para vincular funcionalidades	Implementar navegación principal (dar alta, visualizar datos, empezar test, mirar historias)	Media
		EP-4.1.2 Diseño responsive	Implementar diseño del home y sea adaptable a tablet y móvil	Media
	EP-4.2 Validar referencia	HU-4.2.1 Validar referencia	Implementar validación de código de referencia	Media
		HU-4.2.2 Modificar referencia	Permitir edición de código de referencia	Baja
	EP-4.3 Elegir usuario	HU-4.3.1 Listar lista de usuario	Mostrar listado de usuarios registrados	Baja
		HU-4.3.2 Seleccionar usuario	Implementar selección y carga de perfil de usuario	Baja
		HU-4.3.3 Eliminar usuario desde lista	Implementar eliminación desde listado	Baja
	Splash	EP-5.1 Pantalla splash	HU-5.1 Pantalla splash	Crear pantalla de inicio/carga
EP-5.2 Vincular con FireBase		HU-5.2 Vincular con FireBase	Implementar carga de configuraciones	Baja

Tabla 3.2-1 Product Backlog

3.2.2 Cronograma del Proyecto

Una vez definidas todas las tareas pendientes de realizar y considerados los tiempos dedicados al preanálisis (es decir, investigar en qué consiste el test TMT, las reuniones y consultas con Joaquín Bienvenido Ordieres Mere vía Teams para obtener los requisitos), se ha creado la siguiente tabla para estimar el cronograma del proyecto:

Tarea	Fecha inicio	Fecha Fin	Días	Horas
Preanálisis	30/10/2024	02/12/2024	33	30
Análisis de Requisitos	03/12/2024	31/12/2024	28	35
Mock up	01/01/2025	09/01/2025	8	20
Diseño	10/01/2025	13/02/2025	34	40
Formación	14/02/2025	28/02/2025	14	40
Desarrollo	01/03/2025	01/05/2025	61	250
Pruebas e Integración	02/05/2025	13/05/2025	11	30
Documentación	14/05/2025	20/05/2025	6	20
Total			195	465

Tabla 3.2-2 Cronograma del Proyecto

Dentro del preanálisis, incluso se debe investigar en qué consiste el test TMT, cómo evaluarlo y los estudios de aplicación existentes en el mercado.

En la formación, principalmente se estudia programación en Flutter y el lenguaje Dart.

La documentación se refiere a la documentación técnica del proyecto, donde se explican la estructura y los requisitos, y cómo debería arrancar el proyecto.

3.3 Diseño

En este apartado se explican los diseños de todas las pantallas que se han decidido crear para la aplicación msDtmt, con el fin de cumplir los requisitos.

3.3.1 Pantalla Principal de la aplicación

La pantalla principal de la aplicación, denominada "Home", está diseñada para ofrecer al usuario acceso a las funcionalidades clave del sistema.

- **Icono de Cambio de Tema:** Un icono permite al usuario cambiar la apariencia visual de la aplicación. Al pulsarlo, se ofrece la posibilidad de seleccionar entre un tema claro, un tema oscuro o que la aplicación siga la configuración de tema del sistema operativo del dispositivo.
- **Icono de Cambio de Idioma:** Un icono circular permite al usuario seleccionar el idioma de la interfaz. La aplicación soporta múltiples idiomas como español, inglés y chino. Al pulsar este icono, se despliega un menú donde el usuario

puede elegir su idioma preferido, y la aplicación actualiza todos los textos de la interfaz inmediatamente sin necesidad de reiniciarla.

- **Selección de Usuario:** Permite al usuario seleccionar un perfil existente o crear uno nuevo. Si no hay perfiles, incita a la creación de uno. Si un perfil está seleccionado, muestra un saludo personalizado con el nombre del usuario y permite cambiar de perfil a través de un diálogo.
- **Entrada de Código de Referencia:** Un campo para introducir un código de referencia. Este campo se activa únicamente si hay un perfil de usuario seleccionado. Permite validar dicho código.
- **Tarjeta de Inicio de Test TMT:** Un botón destacado para "Empezar Test TMT". Este botón se activa y es funcional solo si el código de referencia introducido previamente ha sido validado. Junto a este botón, se encuentra una opción para seleccionar el número de círculos para el test (15 o 25) y un botón de ayuda que explica esta configuración.
- **Botones de Acciones del Usuario:**
 - **Visualizar Mis Datos:** Un botón que dirige al usuario a una sección para ver sus datos. Se activa si hay un perfil de usuario seleccionado.
 - **Mirar Mis Historias:** Un botón que permite al usuario acceder a su historial de resultados de tests TMT. Se activa si hay un perfil de usuario seleccionado.
- **Botón Crear Nuevo Perfil:** Un botón que permite al usuario registrar un nuevo perfil en la aplicación.

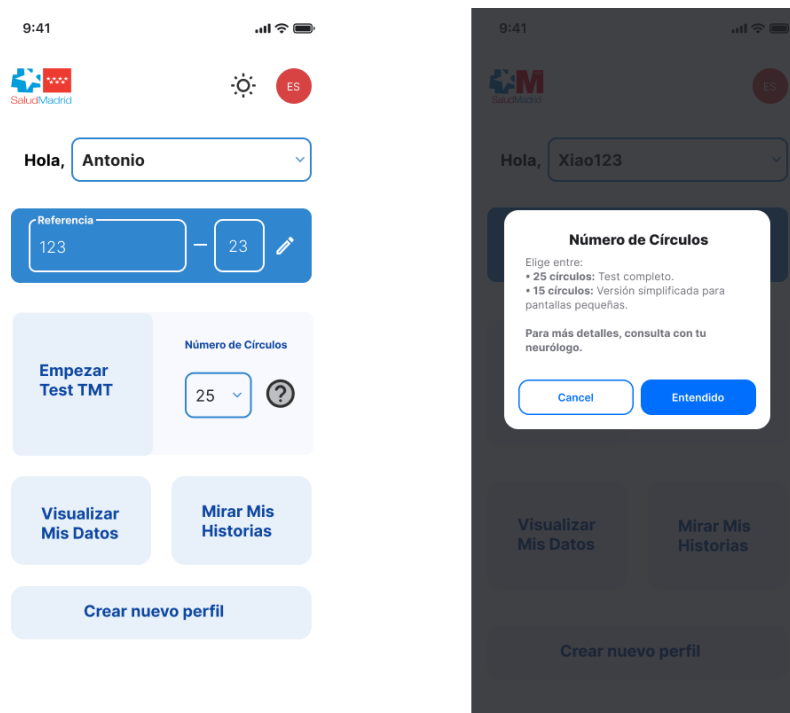


Ilustración 3.3-1 Pantalla Principal de la aplicación

3.3.2 Pantalla de Alta de Usuario

Esta pantalla, titulada "**Crear nuevo perfil**", permite a los usuarios registrar un nuevo perfil dentro de la aplicación.

- **Formulario de Datos:** La pantalla presenta un formulario claro para recoger la información básica necesaria del usuario:
 - **Apodo:** Un campo de texto donde el usuario introduce un nombre único o apodo. Se incluye una sugerencia ("Introduce un apodo") para guiar al usuario.
 - **Sexo:** Selección simple mediante botones de radio entre "Masculino" y "Femenino".
 - **Fecha de nacimiento:** Un campo de texto que, al ser pulsado, despliega un selector de fecha estándar para que el usuario elija su fecha de nacimiento. El campo muestra la fecha seleccionada o una sugerencia ("Seleccionar fecha"). El selector de fecha lleva el título "Seleccionar fecha de nacimiento".
 - **Nivel de estudios:** Un menú desplegable que permite al usuario seleccionar el nivel educativo más alto que ha alcanzado (Primarios, Secundarios, Grado, Máster, Doctorado). Se muestra una sugerencia ("Seleccionar nivel de estudios") hasta que se realiza una selección.

9:41 📶 🔋

< Mis Datos

Apodo

Antonio

Sexo

Masculino Femenino

Fecha de nacimiento

08/17/2025 📅

Nivel de estudios

Grado Universitario ▾

Guardar

Cancelar

Ilustración 3.3-2 Pantalla de Alta de Usuario

3.3.3 Diálogo Seleccionar perfil

Este componente se presenta como una ventana emergente (diálogo) que se superpone a la pantalla principal cuando el usuario interactúa con el control de selección de usuario. La finalidad de este diálogo es permitir la selección del perfil de usuario con el que se desea operar, así como facilitar la eliminación de perfiles existentes.

Funcionalidad y Componentes:

- **Barra de búsqueda:** Esta funcionalidad permite al usuario introducir texto para filtrar la lista de perfiles basándose en el apodo. La barra muestra la indicación "Buscar perfiles" antes de que el usuario introduzca texto.
- **Lista desplazable:** Contiene todos los perfiles de usuario registrados en el dispositivo, cada uno identificado por su apodo. Al pulsar sobre el apodo de un perfil en la lista (sea esta la lista completa o la filtrada por la búsqueda), dicho perfil se establece como el activo para la sesión actual de la aplicación, y el diálogo se cierra automáticamente.
- **Eliminación de Perfil:** Junto a cada apodo listado se visualiza un icono representativo de eliminación (una papelera). Al pulsar este icono, se inicia un proceso de confirmación mediante un sub-diálogo para preguntar al usuario si está seguro de eliminarlo.

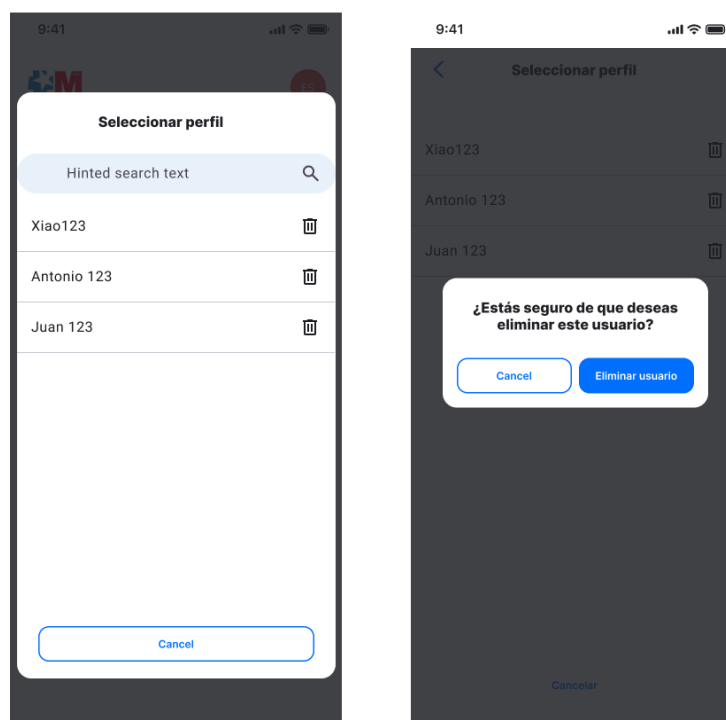


Ilustración 3.3-3 Diálogo Seleccionar perfil

3.3.4 Pantalla de Mi Perfil y Eliminación

En esta pantalla, el usuario puede visualizar los datos de su perfil actualmente activo y, si lo desea, iniciar el proceso de eliminación de este.

Esta pantalla proporciona al usuario un resumen de la información asociada a su perfil seleccionado. La disposición de los elementos prioriza la legibilidad y la facilidad de identificación de los datos clave. Incluye:

- **Identificación del Usuario:** Un campo visible que muestra el nombre del perfil actualmente activo.
- **Detalles del Perfil:** Presenta la información relevante del usuario.
- **Acciones de Perfil:** Un botón claramente visible, como "Eliminar perfil", que permite al usuario iniciar la baja de su cuenta.
- **Diálogo de Confirmación de Eliminación de Perfil:**

Para evitar que el usuario pulse el botón de eliminar de manera accidental, al pulsar el botón de "Eliminar perfil" aparecerá un cuadro de diálogo de confirmación. Este mostrará el mensaje: "¿Estás seguro de que quieres eliminar este perfil?". Habrá dos opciones: "Cancelar" para cancelar la operación de eliminación, y "Eliminar" para confirmar la operación. Si se confirma, la aplicación eliminará dicho perfil y sus datos asociados de la base de datos.

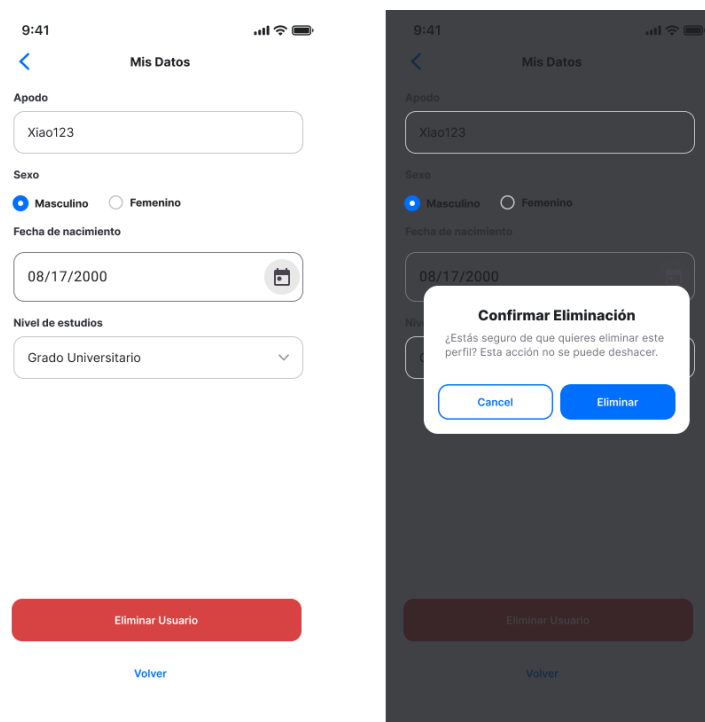


Ilustración 3.3-4 Pantalla de Mi Perfil y Eliminación

3.3.5 Pantalla de Historial de Pruebas (Mis Tests)

Desde la pantalla de Home, puede navegar hacia esta pantalla de “Historia”. En esta pantalla, el usuario puede realizar seguimiento y consultar su historial de tests TMT.

En esta pantalla, la información de los tests TMT se presenta en una tabla. Cada fila corresponde a una partida de test TMT realizada por dicho usuario, y en cada columna se detallan las siguientes informaciones específicas:

- **Fecha:** Indica la fecha en la que el usuario completó la prueba TMT correspondiente.
- **Referencia:** Muestra el código de referencia único que se asoció a esa sesión de prueba en particular.
- **TMT-A:** Presenta el tiempo total, medido en segundos (s), que el usuario tardó en completar la Parte A de la prueba.
- **TMT-B:** Igual que la columna anterior, pero en este caso presenta el tiempo total de la parte B del test TMT.
- **Mano Usado:** Especifica qué mano (Izquierda o Derecha) utilizó el usuario para realizar la prueba en esa ocasión.

Este es un diseño muy simple que muestra los datos más importantes de cada prueba. Si se presentara demasiada información sobre el test TMT, podría causar confusión o ansiedad debido a la gran cantidad de datos. Con esta información simplificada, el usuario puede observar su evolución en el test TMT.



Fecha	Refer	TMT-A	TMT-B	Mano Usado
11/28/2024	123-23	50s	80s	Izq
12/28/2024	123-23	50s	80s	Der

Ilustración 3.3-5 Pantalla de Historial de Pruebas (Mis Tests)

3.3.6 Pantalla de Instrucciones del Test Tmt(Ayuda)

Para el usuario que realiza su test TMT por primera vez o lo hace de forma digital por primera vez, se han diseñado pantallas que indican las instrucciones necesarias para realizar el test TMT.

Esta pantalla aparece antes de comenzar el test de la Parte A o la Parte B

Los componentes clave de esta pantalla son:

- **Título Informativo:** Un encabezado claro que indica la naturaleza de la pantalla y la parte del test a la que se refieren las instrucciones.
- **Descripción de la Tarea:** Un bloque de texto que explica de manera precisa y sencilla las reglas específicas para la parte correspondiente del test:
 - **Para la Parte A:** Se mostrarán números contenidos en círculos. Por favor, conecta con tu dedo los círculos con los números en orden ascendente, empezando desde el número más bajo y continuando con el siguiente en orden. Empieza en 1, luego 2, luego 3, y continúa. (1 → 2 → 3 → ...)
 - **Para la Parte B:** Se mostrarán números y letras contenidos en círculos. Por favor, conecta con tu dedo los círculos alternando entre números y letras en orden ascendente / alfabético. Empieza en el número 1, luego la letra A. Luego el número 2, y luego la letra B. (1 → A → 2 → B → ...)
- **Animación Ilustrativa (GIF):** Para una comprensión aún más clara, debajo del texto descriptivo se presenta una breve animación en bucle (en formato GIF) por cada parte de la prueba. Esta animación muestra cómo se debe desplazar el dedo de la mano sobre el círculo para llegar al siguiente círculo.

Un pequeño detalle aquí es que, dependiendo de la mano que haya elegido el usuario (izquierda o derecha), la mano en la animación de cada parte también cambiará. De esta manera, el usuario tendrá claro qué mano debe utilizar actualmente.

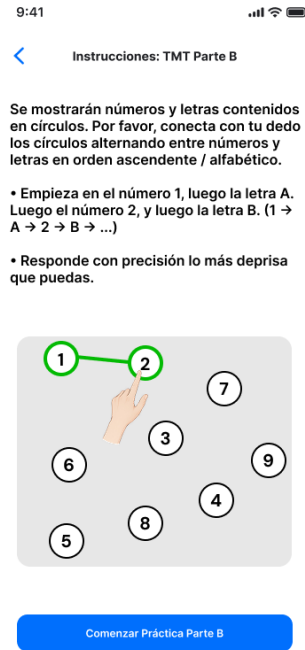
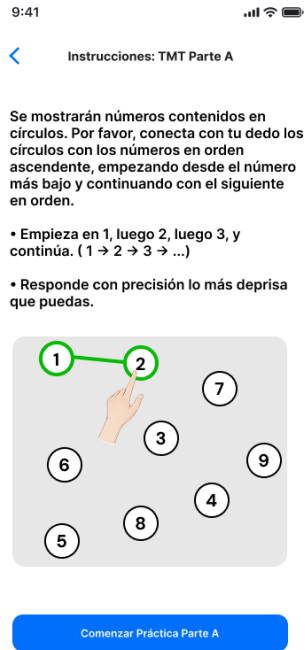


Ilustración 3.3-6 Pantalla de Instrucciones del Test Tmt con mano izquierda

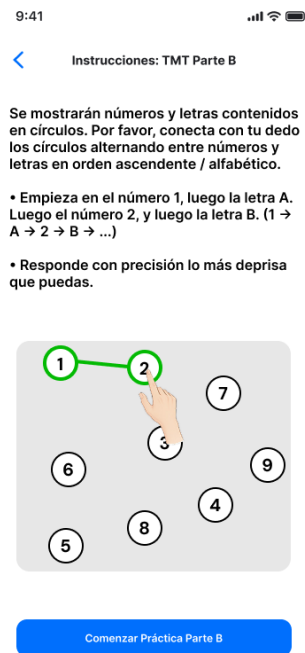
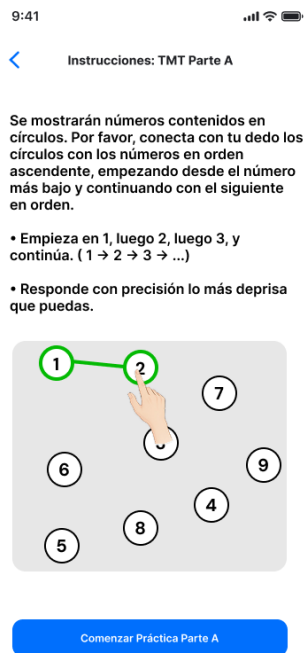


Ilustración 3.3-7 Pantalla de Instrucciones del Test Tmt con mano derecha

3.3.7 Pantalla del Test TMT

Esta pantalla es el núcleo de la aplicación, ya que es donde se realiza la partida del test TMT. Se utiliza tanto para la parte TMT-A como para la TMT-B, y también para hacer la práctica antes de comenzar la prueba

Elementos Clave del Diseño y Funcionalidad:

1. Área de Prueba (Tablero):

- **Generación Aleatoria de Círculos:** Para cada prueba generará una disposición aleatoria de círculos.
- **Contenido de los Círculos:**
 - **Parte A:** Los círculos solo muestran números (por ejemplo, del 1 al 25 o del 1 al 15, según la configuración seleccionada).
 - **Parte B:** Los círculos contienen una secuencia alterna de números y letras (hasta 13-L o una secuencia más corta para la versión de 15 círculos).
- **Diseño Adaptativo:** El tamaño del círculo, es decir, su radio y la distancia entre ellos, así como los márgenes con el borde de la pantalla, se calculan de manera dinámica según el tamaño de la pantalla del dispositivo. De esta manera, se asegura que se pueda visualizar e interactuar bien en diferentes tamaños de pantalla.

2. Barra de Navegación Superior (App Bar):

- **Título de la Prueba:** Muestra qué parte del test se está realizando (ej. "TMT A" o "TMT B").
- **Temporizador:** Un cronómetro visible muestra el tiempo transcurrido desde el inicio de la parte actual del test.
- **Botón de Ayuda (?):** Un icono de interrogación permite al usuario acceder a una pantalla de ayuda si tiene dudas sobre cómo proceder con la prueba.

3. Interacción del Usuario y Feedback Visual:

- **Conexión de Círculos:** Cuando el usuario desliza su dedo entre los círculos, aparecerá una línea negra o blanca, dependiendo del modo de tema. Dicha línea coincidirá con la misma ruta que el usuario haya trazado con su dedo.
- **Feedback de Acierto:** Cuando el usuario conecte los círculos en el orden correcto, la línea anterior cambiará a color verde.

A su vez, los círculos que estén en el orden correcto también cambiarán a color verde, lo cual es una señal que indica al usuario que la secuencia de conexión es correcta.

- **Feedback de Error:**
 - Si el usuario comete un error al seleccionar el siguiente círculo, el sistema proporciona una señal visual. Se resalta en rojo y se muestra una "X" roja.
 - La línea trazada hacia el círculo incorrecto también mostrarse en color rojo.
 - El usuario debe entonces regresar al último círculo correctamente conectado para continuar.

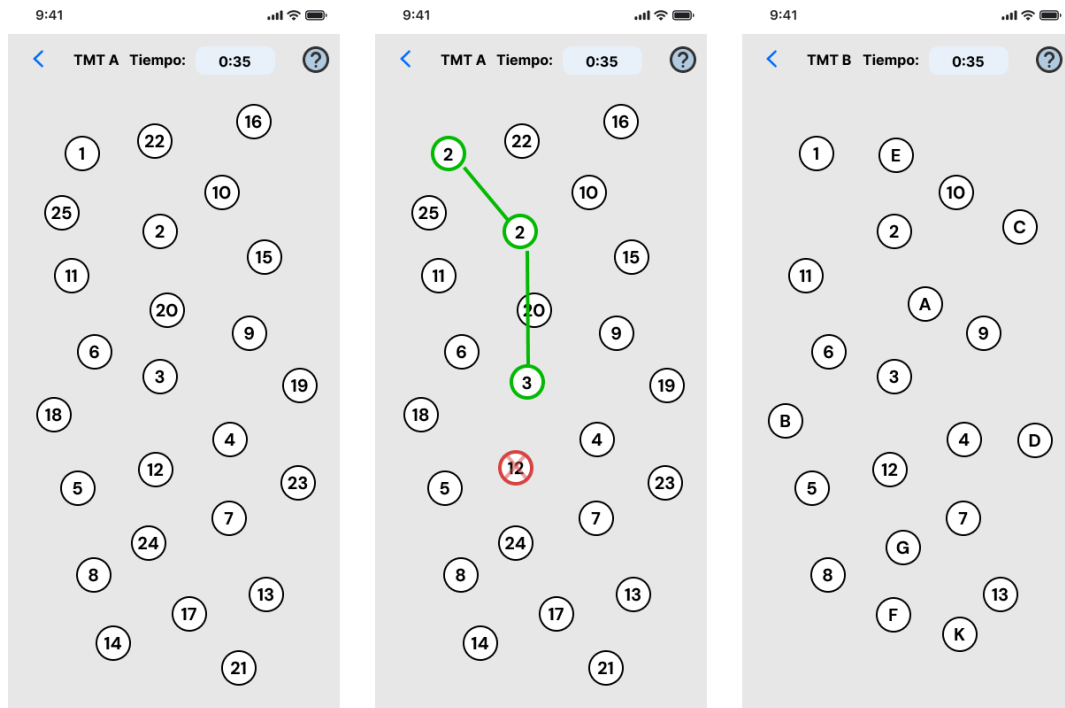


Ilustración 3.3-8 Pantalla del Test TMT

3.3.8 Pantalla de Resultados del Test TMT

Una vez que el usuario completa ambas partes del Trail Making Test (TMT-A y TMT-B), la aplicación presenta la pantalla de "Resultado". Esta pantalla está diseñada para ofrecer un resumen claro y conciso del desempeño del usuario en la prueba recién finalizada.

Elementos Clave del Diseño y Funcionalidad Visualizados:

- **Número de Sesiones:** Justo debajo del título, se muestra el "Número de Sesiones", seguido de un valor numérico. Este dato indica cuántas veces el usuario ha completado la prueba TMT utilizando la aplicación.
- **Tarjetas de Resultados por Parte (TMT-A y TMT-B):** La información principal se organiza en dos tarjetas separadas, una para "TMT A" y otra para "TMT B" .

Este diseño modular facilita la lectura y comparación de los resultados de cada parte del test.

Cada tarjeta contiene:

- **Título de la Parte:** "TMT A" o "TMT B".
 - **Duración:** Muestra el tiempo total que el usuario tardó en completar esa parte específica de la prueba. Se presenta con la etiqueta "Duración" seguida del valor en segundos.
 - **Errores:** Indica el número de errores cometidos por el usuario durante esa parte del test. Se presenta con la etiqueta "Errores" seguida de un valor numérico.
- **Mensaje de Agradecimiento:** Debajo de las tarjetas de resultados, se incluye un mensaje de agradecimiento al usuario por completar la prueba. El texto es: "Le agradecemos la confianza, el tiempo y el esfuerzo en realizar este test dTMT".
 - **Botón de Finalizar:** En la parte inferior de la pantalla, se encuentra un botón prominente con el texto "Terminar". Al pulsar este botón, el usuario normalmente regresaría a la pantalla principal (Home) de la aplicación.

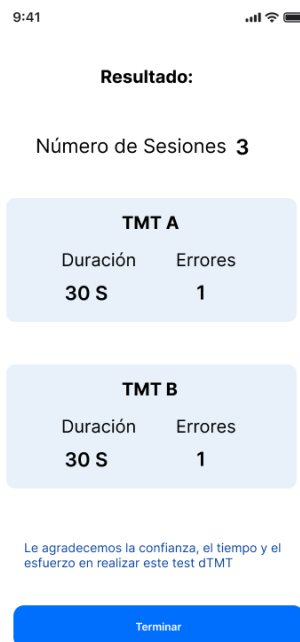


Ilustración 3.3-9 Pantalla de Resultados del Test TMT

3.4 Implementación de la solución

3.4.1 Arquitectura

Para el desarrollo de la aplicación msDTmt, se ha optado por una filosofía arquitectónica que combina la organización por funcionalidades (Feature-First) con los principios de la Arquitectura Limpia (Clean Architecture).

3.4.1.1 Arquitectura general de la aplicación

La estructura principal del proyecto se basa en el enfoque **Feature-First**^[21]. En este paradigma, el código se organiza por funcionalidades, es decir, cada unidad de funcionalidad es distinta y está bien definida dentro de una aplicación, y satisface una necesidad específica del usuario o un objetivo de negocio.

En este caso, la aplicación está separada en funcionalidades como la gestión de perfiles de usuario, la ejecución del Trail Making Test (TMT) y las lógicas de home.

Cada una de estas funcionalidades feature se estructura internamente siguiendo una arquitectura de tres capas bien definidas: **Presentación, Dominio y Datos**. Esta subdivisión interna se inspira directamente en los principios de Clean Architecture, asegurando una separación clara de responsabilidades dentro de cada módulo funcional.^[22]

- **Capa de Presentación (Presentation Layer):**

- **Contenido:** Esta capa es la más externa y es responsable de la interacción directa con el usuario y de la visualización de la información. Contiene los componentes de interfaz de usuario (UI), desde pantallas o vistas (Widgets en Flutter) hasta widgets propios para funciones.

También alberga los controladores de estado (por ejemplo, GetxController si se utiliza GetX), que gestionan la lógica de presentación, responden a las entradas del usuario y preparan los datos para su visualización. Finalmente, los bindings o enlaces de dependencias configuran y proveen las instancias necesarias (controladores, casos de uso) para cada pantalla.

- **Propósito:** Su objetivo principal es mostrar datos al usuario y capturar sus acciones. Se comunica con la capa de Dominio a través de los casos de uso para ejecutar lógica de negocio o solicitar datos, y actualiza la UI en función de los resultados obtenidos. Esta capa depende del framework de UI elegido (Flutter en este caso).

- **Capa de Dominio (Domain Layer):**

- **Contenido:** Es la capa núcleo de la funcionalidad, La Capa de Dominio representa la lógica de negocio de la funcionalidad (o 'característica'). Incluye Casos de Uso, Entidades y Repositorios. La capa de dominio debe ser agnóstica a cualquier framework o tecnología específica.

Sus componentes principales son:

- **Entidades (Entities):** Son los objetos de negocio que representan los conceptos clave y sus atributos (por ejemplo, un Usuario, un ResultadoDeTest). Son clases simples que no dependen de ningún framework.
 - **Casos de Uso (Use Cases):** Orquestan el flujo de datos y encapsulan la lógica para operaciones específicas de la aplicación (por ejemplo, ValidarCodigoReferencia, GuardarResultadoTMT). Utilizan las interfaces de los repositorios para interactuar con la capa de Datos sin conocer sus detalles de implementación.
 - **Interfaces de Repositorio (Repository Interfaces):** Aquí se definen los interfaces o clases abstractas con los métodos que deben utilizarse. Así, luego, desde la capa de Datos se pueden implementar estas interfaces. A través de estos interfaces se realizan las comunicaciones con la capa de Datos. Sin embargo, en esta capa no debería implementarse ninguna interfaz.
- **Propósito:** Centralizar y proteger la lógica de negocios de la aplicación, de forma que esta sea reutilizable y no se encuentre bajo el embate de cambios de la capa externa. El Dominio dicta qué operaciones puede realizar la aplicación y define con qué modelos de datos va a trabajar (Entidades).
- **Capa de Datos (Data Layer):**
 - **Contenido:** La Capa de Datos es la parte encargada de la recuperación y el almacenamiento de datos. Se compone de tres elementos: Modelos, Repositorios y Fuentes de Datos. Las fuentes de datos pueden ser APIs, bases de datos locales u otros proveedores de datos externos.

Sus componentes típicos son:

- **Implementaciones de Repositorio (Repository Implementations):** Aquí son las clases concreta que deberían implementar las interfaces de la capa de Dominio con sus métodos. Aquí, decidiendo si los datos se obtienen de una fuente remota (API), una base de datos local, una caché u otra fuente.
- **Fuentes de Datos (Data Sources):** Clases que interactúan directamente con bibliotecas o marcos de trabajo para la persistencia o la comunicación. Se pueden distinguir entre fuentes de datos remotas (por ejemplo, servicios que hacen llamadas REST API) y fuentes de datos locales (por ejemplo, clases que gestionan el acceso a una base de datos SQLite o a SharedPreferences).
- **Modelos de Datos (Models):** Representaciones de los datos específicos de cada fuente (por ejemplo, modelos para deserializar respuestas JSON de una API o modelos que reflejan

la estructura de las tablas de una base de datos). Estos modelos a menudo necesitan ser mapeados hacia y desde las Entidades de la capa de Dominio para mantener la independencia de esta última.

- **Propósito:** Abstracter el origen y el método de acceso a los datos del resto de la aplicación. Proporciona los datos que el Dominio necesita, gestiona la persistencia y maneja la comunicación con sistemas externos.

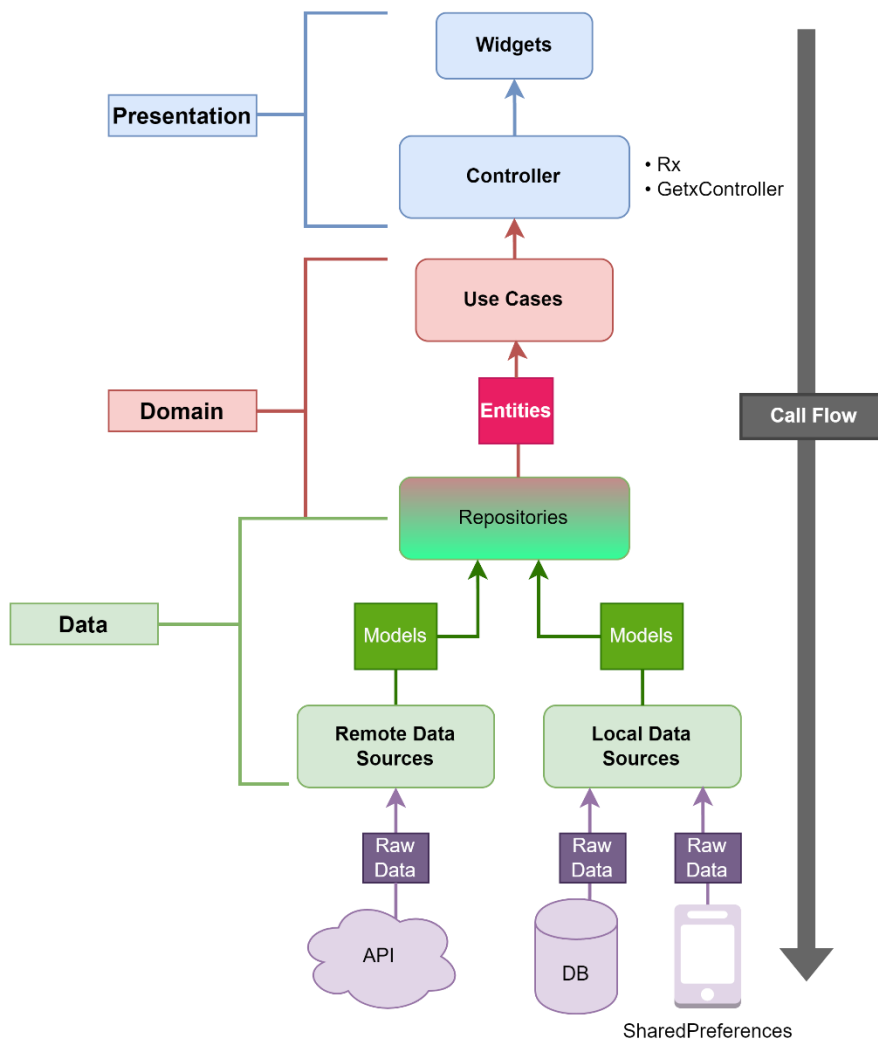


Ilustración 3.4-1 Capas de Clean Architecture^[23]

3.4.1.2 Estructura del proyecto

El código fuente de la aplicación, ubicado dentro del directorio lib/, se organiza jerárquicamente como se describe a continuación, reflejando una clara separación de responsabilidades y una agrupación lógica de los componentes: ^[24]

- **main.dart:** Este archivo representa el punto de entrada principal de la aplicación Flutter. Es responsable de inicializar la aplicación, configurar los servicios globales esenciales y lanzar la primera pantalla o módulo.

- **app/**: Este directorio constituye el núcleo de la aplicación y alberga la mayor parte del código fuente. Su estructura interna se desglosa de la siguiente manera:
 - **config/**: Dedicado a la configuración global de la aplicación. Contiene subdirectorios para aspectos transversales:
 - **routes/**: Define y gestiona la navegación dentro de la aplicación. Aquí se establecen las rutas nombradas, las transiciones entre pantallas y los observadores de navegación, si los hubiera.
 - **themes/**: Contiene la configuración relacionada con la apariencia visual de la aplicación. Esto incluye la definición de los temas principales (claro/oscuro), la paleta de colores, los estilos de texto y tipografías, y cualquier otro aspecto estilístico global.
 - **translation/**: Alberga los recursos necesarios para la internacionalización (i18n) de la aplicación, permitiendo que la interfaz de usuario se presente en múltiples idiomas.
 - **constans/**: Este directorio almacena valores constantes utilizados a lo largo de toda la aplicación. Pueden ser cadenas de texto recurrentes, claves de API (aunque se recomienda gestionarlas de forma segura), rutas de assets, o cualquier otro valor que deba permanecer inmutable y ser accesible globalmente.
 - **features/**: Es el directorio central que materializa el enfoque Feature-First de la arquitectura. Cada subdirectorio dentro de features/ representa un módulo de funcionalidad principal y autocontenido de la aplicación. Para el proyecto msDTmt, se identifican los siguientes módulos de funcionalidad principales:
 - **home/**: Engloba la lógica y la interfaz de usuario de la pantalla principal o de inicio de la aplicación, desde donde el usuario suele iniciar las interacciones clave.
 - **splash/**: Contiene el código de la pantalla de bienvenida o de carga inicial (splash screen) que se muestra al arrancar la aplicación.
 - **tm_tst/**: Dedicado a toda la funcionalidad relacionada con el Trail Making Test (TMT), incluyendo cómo generar el test de manera aleatoria, cómo interactuar durante la prueba, la recopilación de datos de métricas durante la prueba y la presentación de resultados.
 - **user/**: Responsable de la gestión de perfiles de usuario, como el registro, la selección de perfiles existentes y la visualización del historial de tests asociados a un usuario.
 - **shared_components/**: Contiene widgets de Flutter y otros componentes de interfaz de usuario que son reutilizables de la aplicación. El propósito

de este directorio es promover la consistencia visual, evitar la duplicación de código de UI y facilitar el mantenimiento de elementos comunes.

- **utils/**: Agrupa diversas utilidades y servicios genéricos que pueden ser reutilizados por diferentes partes de la aplicación.

Su estructura interna incluye:

- **helpers/**: Clases o funciones auxiliares que proporcionan funcionalidades genéricas o simplifican tareas comunes (ej., formateo de fechas, validaciones simples, acceso a información del dispositivo).
- **mixins/**: Contiene mixins de Dart, que permiten añadir funcionalidades comunes a varias clases de forma concisa sin recurrir a la herencia múltiple.
- **services/**: Módulos que encapsulan la interacción con servicios externos o funcionalidades de bajo nivel, como la comunicación con APIs de red, el acceso a bases de datos locales, la gestión del almacenamiento local (SharedPreferences), o la inicialización y manejo de tareas en segundo plano.

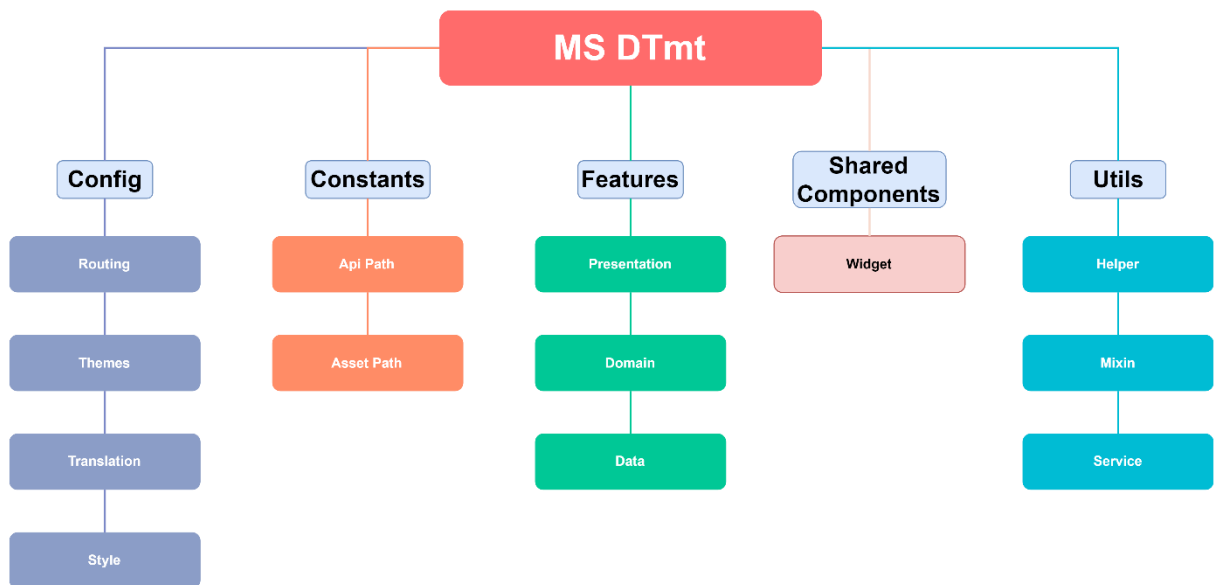


Ilustración 3.4-2 Estructura del proyecto

3.4.1.3 Justificación de tecnologías

Para la aplicación MS DTmt, se ha seleccionado **Flutter** como el framework principal. La razón primordial de esta elección radica en el requisito de que el proyecto funcione de manera nativa y eficiente en una diversidad de dispositivos, incluyendo tanto teléfonos móviles como tablets, y en los dos principales sistemas operativos móviles: Android y iOS.

- **GetX como Framework de Gestión Integral**^[25]

Dentro del ecosistema de Flutter, se ha adoptado GetX como un micro-framework central para la gestión de múltiples aspectos de la aplicación. GetX es una solución extra ligera y potente para Flutter. Combina gestión de estado de alto rendimiento, inyección de dependencia inteligente y gestión de rutas de forma rápida y práctica.

GetX ofrece un conjunto de herramientas para:

- **Gestión de Estado Reactiva:** Mediante GetxController y variables observables (.obs), facilita la creación de interfaces que reaccionan a los cambios de datos con un código conciso y eficiente.
- **Inyección de Dependencias (DI):** Ofrece un sistema de DI integrado y fácil de usar (mediante Get.put(), Get.lazyPut(), Get.find()), para gestionar el ciclo de vida desacoplado y el aprovisionamiento de instancias de clases (como controladores, servicios, casos de uso o repositorios).
- **Navegación intuitiva:** La navegación de pantalla a pantalla se presenta de manera concisa mediante métodos como **Get.to()**, **Get.back()**, y rutas nombradas, lo que facilita considerablemente las transiciones entre las distintas secciones de la aplicación.
- **Internacionalización integrada:** GetX ofrece la posibilidad de soportar varios idiomas, lo que nos permitió implementar el requisito de cambio de idioma entre español e inglés utilizando .tr para fines de traducción.
- **Utilidades Diversas:** Incluye herramientas para la internacionalización, la visualización de diálogos, snackbars, bottom sheets, y otras funcionalidades que agilizan el desarrollo.

En el proyecto se ha utilizado GetX para facilitar mantener la estructura de Clean Code. A su vez, facilita las navegaciones entre diferentes pantallas (pantalla de inicio, test TMT, resultados, etc.). También, gracias a GetX, se facilita el soporte de tres idiomas (español, inglés y chino) diferentes en la aplicación y permite cambiar el idioma de manera directa sin necesidad de reiniciar la aplicación

- **SQLite para Persistencia de Datos Local** ^[26]

SQLite es una biblioteca de software que proporciona un sistema de gestión de bases de datos relacionales. El "lite" en SQLite hace referencia a su naturaleza ligera en cuanto a configuración, administración de la base de datos y recursos necesarios.

SQLite cuenta con las siguientes características destacadas:

- Autocontenida
- Sin servidor (Serverless)
- Configuración cero

- Transaccional

Esta naturaleza auto-contenida, junto con su ligereza, fiabilidad, y el no requerir configuración, lo convierte en una opción muy popular y común para el almacenamiento de datos en aplicaciones móviles y otros dispositivos embebidos.

La aplicación MSDTmt, aprovechando estas ventajas de SQLite, ha creado una base de datos local en los dispositivos para guardar datos de usuario y resultados del test TMT.

- **Dio para Comunicación con el Servidor^[27]**

Dio es un potente paquete de red HTTP para Dart/Flutter, compatible con configuración global, interceptores, FormData, cancelación de solicitudes, carga/descarga de archivos, tiempo de espera (timeout), adaptadores personalizados, transformadores, etc. Además, tiene funciones avanzadas como reintentos automáticos y cancelación de solicitudes. Y es compatible con múltiples plataformas (iOS, Android, web).

Para no reinventar la rueda de realizar las peticiones en cada plataforma, la aplicación MSDTmt ha utilizado esta librería para facilitar la comunicación con el servidor. Todas las lógicas de Dio se encuentran en la clase RestApiServices. Esta clase, a su vez, expone una interfaz para comunicarse con otras clases, lo que permite que la funcionalidad de comunicación HTTP quede más limpia y aislada.

- **WorkManager para la gestión tareas en segundo plano^[28]**

Worker Manager es una librería potente y fácil de usar que te ayuda a gestionar eficientemente las tareas que consumen mucha CPU en tus aplicaciones Flutter. Ofrece varias ventajas sobre la programación asíncrona tradicional o el método compute integrado. Ten en cuenta que las funciones que pases deben ser métodos estáticos, funciones definidas globalmente o lambdas para asegurar que sean accesibles desde otros isolates.

Al usar Worker Manager, se pueden disfrutar de los beneficios de una programación eficiente de tareas, isolates reutilizables, tareas cancelables y comunicación entre isolates. Ofrece una clara ventaja sobre la programación asíncrona tradicional y el método compute integrado, asegurando que las aplicaciones Flutter sigan siendo performantes y responsivas incluso al manejar tareas que consumen mucha CPU. La función adicional executeGentleWithPort proporciona cancelación "gentil" y comunicación entre isolates para las tareas, ofreciendo un mayor control y eficiencia.

Dentro de la aplicación MSDTmt, un requisito muy importante es guardar los resultados y métricas del test TMT en caso de que no haya internet al intentar enviarlos al servidor. Cuando se recupera la conexión a internet, se vuelve a intentar enviar estos datos en primer o segundo plano. Se ha utilizado esta librería para facilitar la programación de tareas en segundo plano con la condición de que se recupere la conexión a internet.

Cuando un usuario completa un test TMT y la aplicación no puede enviar inmediatamente los resultados al servidor debido a problemas de conectividad, la aplicación primero almacena estos datos en SharedPreference. Luego, programa una tarea en segundo plano mediante WorkManagerHandler.scheduleResultsSending(), que monitorizará las condiciones de red y reintentará automáticamente el envío cuando se restablezca la conexión.

Consideraciones sobre WorkManager y Compatibilidad Multiplataforma^[29]:

Aunque WorkManager es más compatible con Android, existen algunas restricciones en iOS, dado que Apple prohíbe estrictamente la ejecución de tareas en segundo plano. Sin embargo, en iOS, el sistema operativo decide cuándo debe ejecutarse la tarea, dependiendo de su propio algoritmo y las condiciones actuales del dispositivo (por ejemplo, estado de carga, estado inactivo, conexión Wi-Fi, etc.). Por lo tanto, nunca se tiene la seguridad de que la tarea se ejecutará inmediatamente después de que su retraso inicial entre en juego. Podría ocurrir unos minutos u incluso horas después. Aun así, esta limitación no es un gran problema para este caso de uso específico porque:

- La funcionalidad solo requiere enviar datos cuando se recupere la conexión.
- Las llamadas a la API son rápidas (completándose dentro de 30 segundos).
- No es crítico el momento exacto de ejecución, sino que eventualmente se ejecute.
- Los datos permanecen seguros en SharedPreferences hasta ser enviados exitosamente.

Por lo tanto, aunque con algunas restricciones en iOS, la utilización de esta librería no afectará su uso en ambas plataformas.

• **Sphinx y ReadTheDocs para Documentación Técnica**

Para escribir el documento técnico de calidad, se pueden usar los programas de línea y otros programas gratuitos:

- **Sphinx:**^[30] Sphinx es un generador de documentación, o una herramienta que transforma un conjunto de archivos fuente de texto plano a varios formatos de salida, produciendo automáticamente referencias cruzadas, índices, etc. Es decir, si tienes un directorio con varios documentos reStructuredText o Markdown, Sphinx puede generar una serie de archivos HTML, un archivo PDF (vía LaTeX), páginas man y mucho más.

Sphinx se enfoca en la documentación, en particular la documentación escrita a mano. Sin embargo, también se puede usar para generar blogs, páginas de inicio e incluso libros. Gran parte del poder de Sphinx proviene de la riqueza de su formato de marcado de texto plano predeterminado, reStructuredText, junto con sus capacidades de extensibilidad significativas.

- **Dart Doc**^[31]: Es una herramienta oficial de Flutter que permite usar el comando `dart doc` para generar documentación de referencia HTML para el código fuente de Dart.

Para este proyecto se ha utilizado Sphinx como el motor principal para la generación de la documentación, donde se explica la descripción general, la arquitectura detallada y cómo está estructurado el proyecto. Luego, dentro de Sphinx, se hace referencia a la documentación detallada de cada clase que ha sido generada con la herramienta Dart Doc. De esta manera, la documentación queda estructurada de forma limpia.

Luego, el proyecto está conectado con la plataforma ReadTheDocs, que es una plataforma en línea donde permite acceder de manera pública y gratuita a la documentación en HTML. Cada vez que se hace un *push* al repositorio donde está alojado el proyecto (en este caso, en GitHub), se utilizarán los comandos necesarios de Sphinx para generar la documentación requerida y transformarla a HTML. Luego, se publicará automáticamente en ReadTheDocs.

Está disponible para consulta en la siguiente dirección URL:

<https://2025-peng-xiaoyang-cogtests.readthedocs.io/en/latest/>

Adicionalmente, se ha creado un archivo README con código Markdown —que es un lenguaje de marcado ligero utilizado para dar formato a texto plano—. Este archivo contiene un resumen del proyecto, incluyendo una descripción general, un resumen de la arquitectura, instrucciones sobre cómo generar documentación con Sphinx y los prerrequisitos para ejecutar el proyecto.

El código está totalmente público y alojado en un repositorio en la plataforma GitHub. Esto se hace con el fin de fomentar la transparencia y para que otras personas que deseen realizar proyectos similares no tengan que empezar desde cero, sino que puedan tener una referencia.

Es accesible a través del siguiente enlace:

https://github.com/jordieres/2025_PENG_XiaoYang_CogTests

3.4.2 Base de Datos

La implementación de un sistema de almacenamiento local en la aplicación responde a varios requisitos funcionales y de rendimiento clave.

En primer lugar, es esencial para garantizar la funcionalidad multiusuario, permitiendo que diferentes perfiles utilicen la aplicación en un mismo dispositivo y mantengan sus datos de forma separada y segura.

En segundo lugar, una base de datos local es fundamental para ofrecer a los usuarios la capacidad de consultar su historial de tests TMT previos, facilitando el seguimiento de su progreso y rendimiento a lo largo del tiempo.

Finalmente, contribuye a optimizar la comunicación con el servidor y aligerar su carga, ya que permite realizar validaciones previas en el dispositivo, como la comprobación de la unicidad de un código de referencia para un test TMT antes de realizar una consulta remota. Esta capacidad de operar con datos locales también asegura la persistencia de la información y una experiencia de usuario fluida, incluso en escenarios sin conexión a internet.

Para la implementación de esta base de datos local en la aplicación móvil, se ha utilizado SQLite, tecnología cuya selección y uso se detallaron previamente en el apartado de arquitectura.

3.4.2.1 Modelo Entidad Relación

Antes de proceder con la implementación, se llevó a cabo una fase de diseño de la estructura de la base de datos. El objetivo era representar de forma clara y organizada las entidades principales del dominio de la aplicación (perfiles de usuario y resultados de los tests TMT) y las relaciones existentes entre ellas.

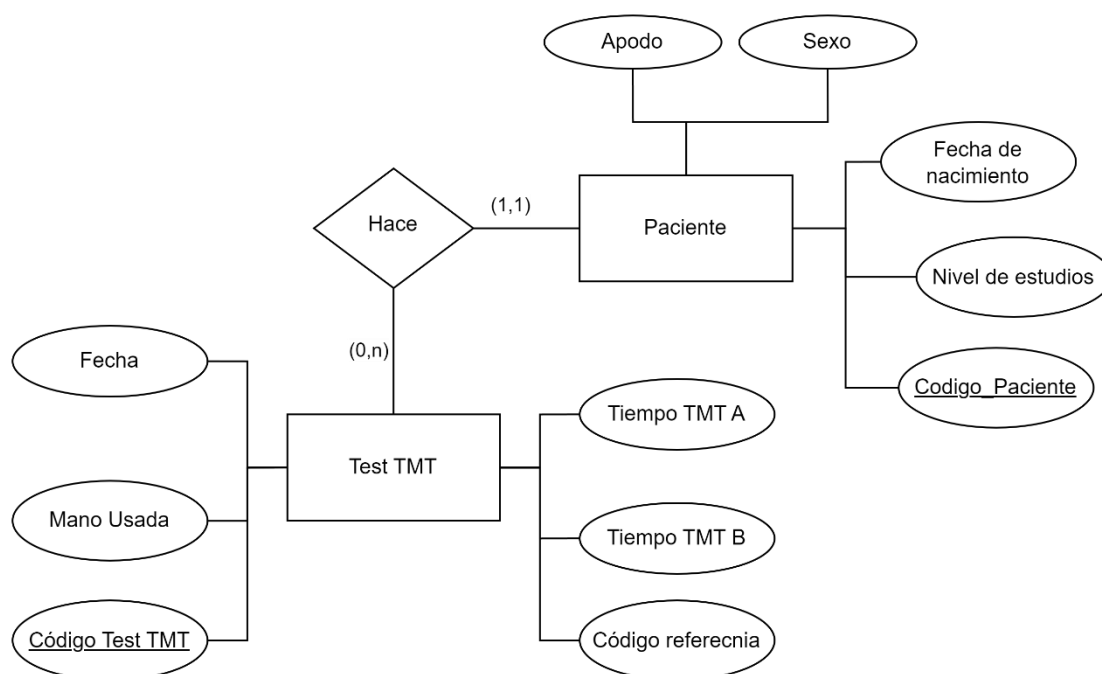


Ilustración 3.4-3 Diagrama Modelo Entidad Relación

En la tabla a continuación se listan todas las entidades presentes en el diagrama, junto a sus respectivos atributos y una breve explicación de qué representan:

Entidad	Atributo	Dominio
Paciente	Codigo_Paciente	Identificador único del paciente generado mediante UUID.v4, garantizando unicidad global incluso entre diferentes dispositivos (clave primaria).

	Apellido	Nombre o alias del paciente para su identificación por parte del usuario.
	Sexo	Género del paciente (Masculino/Femenino)
	Fecha_nacimiento	Fecha de nacimiento del paciente, necesaria para determinar su edad.
	Nivel_estudios	Nivel educativo alcanzado por el paciente (Primarios, Secundarios, Grado, Máster, Doctorado).

Tabla 3.4-1 Explicación de los atributos de la entidad Paciente

Entidad	Atributo	Dominio
Test TMT	Codigo_Test_TMT	Identificador único de la prueba generado mediante UUID.v4, garantizando unicidad global incluso entre diferentes dispositivos (clave primaria).
	Codigo_referencia	Código proporcionado por el neurólogo que autoriza la realización del test.
	Fecha	Fecha y hora en que se realizó el test.
	Mano_Usada	Mano con la que el paciente realizó la prueba (Izquierda/Derecha).
	Tiempo_TMT_A	Tiempo en segundos que tardó el paciente en completar la parte A del test.
	Tiempo_TMT_B	Tiempo en segundos que tardó el paciente en completar la parte B del test.

Tabla 3.4-2 Explicación de los atributos de la entidad Test TMT

3.4.2.2 Modelo Relacional

A partir del modelo Entidad/Relación, se ha derivado el siguiente esquema relacional que representa la estructura final de la base de datos, con sus claves primarias y claves foráneas:

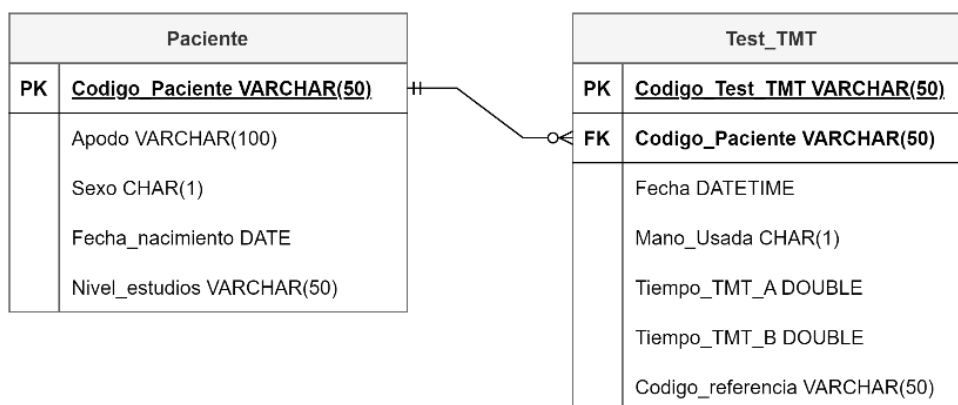


Ilustración 3.4-4 Modelo Relacional

3.4.2.3 Implementación de la Base de Datos

La creación, gestión y manipulación de la base de datos SQLite dentro de la aplicación Flutter se centraliza fundamentalmente en la siguiente clase Dart clave, ubicada en el directorio de utilidades (`lib/app/utills/services/`) del proyecto:

UserDatabaseHelper.dart

Esta clase es la principal encargada de la gestión de todas las operaciones relacionadas con la base de datos:

- Creación y actualización del esquema de la base de datos.
- Métodos para insertar, actualizar, eliminar y consultar perfiles de pacientes.
- Métodos para registrar nuevos resultados de tests TMT.
- Consultas para obtener el historial de tests de un paciente específico.

La clase implementa el patrón **Singleton** para garantizar que solo exista una instancia de conexión a la base de datos en toda la aplicación, optimizando así el rendimiento y evitando problemas de concurrencia.

3.4.3 Implementación de la API

La aplicación msDtmt requiere comunicación HTTP con el servidor backend para dos funcionalidades principales: la validación de códigos de referencia y el envío de los resultados de las métricas de los tests TMT. Esto permite su almacenamiento en la base de datos del servidor, garantizando el acceso y análisis por parte de los neurólogos.

En este proyecto se ha utilizado una API (Interfaz de Programación de Aplicaciones) de tipo RESTful, implementada por el cotutor Joaquín Bienvenido Ordieres Mere. Esta API

utiliza el formato JSON (JavaScript Object Notation) para el intercambio de datos entre la aplicación y el servidor.

3.4.3.1 *Endpoint: Validación de Código de Referencia*

- **Propósito:** El objetivo de este endpoint es validar el código de referencia introducido por el usuario. Una sesión de test TMT se divide en dos partes: una realizada con la mano izquierda y otra con la mano derecha. Ambos partidos de la misma sesión están vinculados a un único código de referencia.

Por lo tanto, este endpoint, además de comprobar la validez del código de referencia, también proporciona información como el número de partidos utilizados con dicho código y con qué mano o manos han sido utilizados. De esta manera, la aplicación puede guiar al usuario sobre qué mano debe usar en cada parte y si este código ya ha sido utilizado en otra sesión.

- **URL Relativa:** /procesar
- **Método HTTP:** POST
- **Solicitud (Request Body):**

```
{  
  "codeid": "string"  
}
```

Donde "codeid" es el código de referencia introducido por el usuario.

- **Respuesta (Response Body):**

La respuesta de API siempre devuelve un código de estado HTTP 200 si no existe ningún error en el servidor, tanto si código es válido o inválido.

Para interpretar correctamente la respuesta, la aplicación cliente debe analizar varios campos clave:

- **status:** Indica si la petición fue procesada correctamente por el servidor.
- **message:** Proporciona información sobre el resultado de la validación.
- **exists:** Valor numérico que indica el grado de utilización previa del código (0, 1 o 2).
- **hands:** Array que enumera las manos con las que ya se ha realizado el test usando este código.

Estas respuestas se clasifican en cuatro casos típicos que determinan el flujo de la aplicación y las opciones disponibles para el usuario, como se detalla en el **Anexo**, específicamente en la [Tabla 5.3-1 **Casos de Respuesta del Endpoint de Validación de Código de Referencia**].

3.4.3.2 Endpoint: Envío de Resultados del Test TMT

- **Propósito:** Este endpoint permite a la aplicación enviar al servidor los datos y métricas recopiladas durante la realización de un test TMT por parte de un usuario. Estos datos se almacenan en el backend para su consulta y análisis.
- **URL Relativa:** /reportar
- **Método HTTP:** POST
- **Solicitud (Request Body):**

Formato: JSON

Estructura: El cuerpo de la solicitud es un objeto JSON complejo que contiene la información del usuario que realizó el test y las métricas detalladas del mismo.

El cuerpo de la solicitud es un objeto JSON complejo que contiene la información del usuario que realizó el test y las métricas detalladas del mismo. La descripción de todos los parámetros que se deben incluir en la solicitud se encuentra detallada en el **Anexo**, específicamente en la **[Tabla 5.3-2 Parámetros a enviar al endpoint de Envío de Resultados del Test TMT]**

- **Respuesta (Response Body):**

El endpoint utiliza códigos de estado HTTP estándar y respuestas en formato JSON con dos campos principales:

- **status:** Indica el resultado de la operación ("ok" para éxito, "error" para fallos)
- **message:** Proporciona información adicional, como el número de registro asignado o detalles del error

La siguiente tabla describe los dos tipos de respuesta posibles:

Tipo de Respuesta	Descripción	Estructura JSON de Respuesta	Código HTTP
Éxito	La solicitud fue procesada correctamente y los resultados del test TMT fueron almacenados exitosamente en el sistema. La respuesta incluye un número de registro único que identifica el test almacenado.	<pre>{ "message": "Rec#:314", "status": "ok"}</pre> <p>Nota: El número después de "Rec#:" corresponde al número del test tmt almacenado en el base de dato.</p>	200 OK

Tipo de Respuesta	Descripción	Estructura JSON de Respuesta	Código HTTP
Error	<p>El error se produce cuando falta algún campo obligatorio en la solicitud o si sucede un error inesperado en el servidor durante el procesamiento de los datos.</p> <p>El mensaje proporciona información específica sobre la causa del error.</p>	<pre>{ "status": "error", "message": "ERROR: llave duplicada viola restricción de unicidad unique_codeid_mano"} </pre> <p>Nota: el contenido del mensaje de error dependerá de la causa del problema, y proporcionará información detallada al respecto.</p>	500 Internal Server

Tabla 3.4-3 Diferentes respuestas del endpoint de Envío de Resultados del Test TMT

3.4.4 Cálculos y algoritmos para generar test TMT

Para cada partido del test TMT, se debe generar un diseño de manera aleatoria y que sea válido.

Para ello, se necesita un algoritmo que distribuya los círculos de forma equilibrada en la pantalla y que se adapte al tamaño de la pantalla del dispositivo. Es fundamental garantizar que el cálculo del radio de los círculos y la separación entre ellos sea adecuado. Adicionalmente, hay que buscar una ruta o trayectoria entre círculos para que no haya obstáculos entre los dos círculos que se van a conectar. De esta manera, se facilita la interacción del usuario al conectar dos círculos. También es importante asegurar la secuencia lógica para cada parte del test: en la Parte A es un orden numérico y en la Parte B es un orden alternando entre números y letras.

En este apartado, se explicarán detalladamente los procesos e implementaciones en el proyecto para lograr estas características.

3.4.4.1 Calcular el radio del círculo

Para garantizar que los círculos del test TMT sean totalmente adaptables a diferentes resoluciones y tamaños de pantalla, se ha implementado un sistema para calcular dinámicamente las dimensiones de los círculos:

1. **Cálculo del área disponible:** Se obtienen el ancho y alto máximos de la pantalla del dispositivo y se resta un margen de borde (18 píxeles en la parte superior, inferior, izquierda y derecha), obteniéndose así el área segura y útil disponible.
2. **Determinación del tipo de dispositivo:** Detecta si está utilizando una Tablet o un teléfono móvil.
3. **Asignación del porcentaje de área:** Dependiendo del tipo de dispositivo, se asigna un porcentaje del área total de pantalla que ocuparán los círculos:
 - Para teléfonos móviles: 14% del área total disponible
 - Para tablets: 12% del área total disponible (proporcionalmente menor debido al mayor tamaño de pantalla)
4. **Distribución del área entre círculos:** El área total designada para círculos se divide entre el número total de círculos.
5. **Conversión de área a radio:** Utilizando la fórmula matemática del área de un círculo $A = \pi \times r^2$ se despeja el radio: $r = \sqrt{\frac{A}{\pi}}$ donde A es el área por círculo.

3.4.4.2 Algoritmo de generación aleatoria de Círculos

El algoritmo para generar círculos aleatorios en el tablero del Test TMT sigue varios pasos para garantizar una distribución adecuada mientras mantiene distancias mínimas entre círculos y evita la concentración de círculos en una misma zona:

1. **División del área en una cuadrícula mediante raíz cuadrada:**
 - El algoritmo determina cuántas filas y columnas se necesitan utilizando la raíz cuadrada del número total de círculos. (Por ejemplo, para 25 círculos, la raíz cuadrada es 5, por lo que se crea una cuadrícula de aproximadamente 5x5.)
 - **El número exacto de filas** se calcula redondeando hacia arriba la raíz cuadrada del número de círculos.
 - **El número de columnas** se calcula dividiendo el número total de círculos entre el número de filas y redondeando hacia arriba.
 - Esta aproximación mediante raíz cuadrada crea una distribución más balanceada que usar solo filas o columnas.
2. **Identificación del punto de origen de cada celda:**
 - Para cada celda en la cuadrícula, se calcula su punto de origen (esquina superior izquierda).
 - Este punto de origen sirve como referencia para generar puntos aleatorios dentro de esa celda específica.
 - Se almacenan estas coordenadas para cada círculo generado, lo que permite regenerar puntos dentro de la misma celda si es necesario.

- Toda esta información se va a guardar en la clase **CircleGenerator** [Ilustración 3.4-6 Diagrama de clase de CircleGenerator] que facilita regenerar coordenadas en caso de que no cumplan los requisitos.
 - Offset: coordenado (x, y) de punto centro de círculo.
 - CellOriginalPoint: punto de origen de cada celda.
 - rowIndex: posición de índice de fila según tabla donde está la celda.
 - columnIndex: posición de índice de columna según tabla donde está la celda.
 - Order: Orden del círculo.

3. Generación de puntos aleatorios confinados a cada celda:

- Para cada celda, se generan coordenadas aleatorias relativas a su punto de origen.
- Las coordenadas finales se calculan sumando el punto de origen con la posición aleatoria dentro de la celda.
- Por cada celda, el algoritmo realiza solo 10 intentos para garantizar la eficiencia y evitar entrar en un bucle, con el objetivo de encontrar un punto que mantenga la distancia mínima requerida.

4. Verificación de distancia mínima entre círculos:

- La distancia mínima es el diámetro del círculo más una distancia adicional de seguridad alrededor de cada círculo, que en este caso es de 5 píxeles, además del margen para el borde de la pantalla, que es de 36 píxele
- Para cada punto candidato, se verifica que esté a una distancia mínima de todos los puntos ya colocados.
- Si el punto no cumple con este requisito, se genera un nuevo punto aleatorio dentro de la misma celda.
- Esta verificación se realiza calculando entre el punto candidato y todos los puntos existentes. Para calcular la distancia de dos puntos, puede utilizar:
 - Fórmula euclidiana

$$d = \sqrt{(x_p - x_c)^2 + (y_p - y_c)^2}$$

- Alternativamente, utiliza el método **distance** de la clase **Offset** de Flutter

5. Método de respaldo cuando faltan círculos:

Si después de intentar colocar círculos en todas las celdas aún no se ha alcanzado el número requerido, se utiliza un método de respaldo.

- En este caso, se generan puntos aleatorios en todo el espacio disponible, sin restricción de celdas.
- Se mantiene la verificación de la distancia mínima, realizando solo 100 intentos para garantizar la eficiencia y evitar entrar en un bucle al completar los círculos faltantes.

6. Ajuste adaptativo de la distancia mínima:

- Si todavía no hay suficientes círculos, el algoritmo reduce la distancia mínima requerida multiplicándola por un factor de 0.9.
- Esta reducción se puede repetir hasta 10 veces, lo que permite que eventualmente la distancia sea lo suficientemente pequeña para colocar todos los círculos necesarios.
- Después de 10 reducciones, la distancia se habrá reducido aproximadamente al 35 % de su valor original (0.9^{10}). Con este último método, se garantiza al 100 % la generación de la cantidad necesaria de círculos.

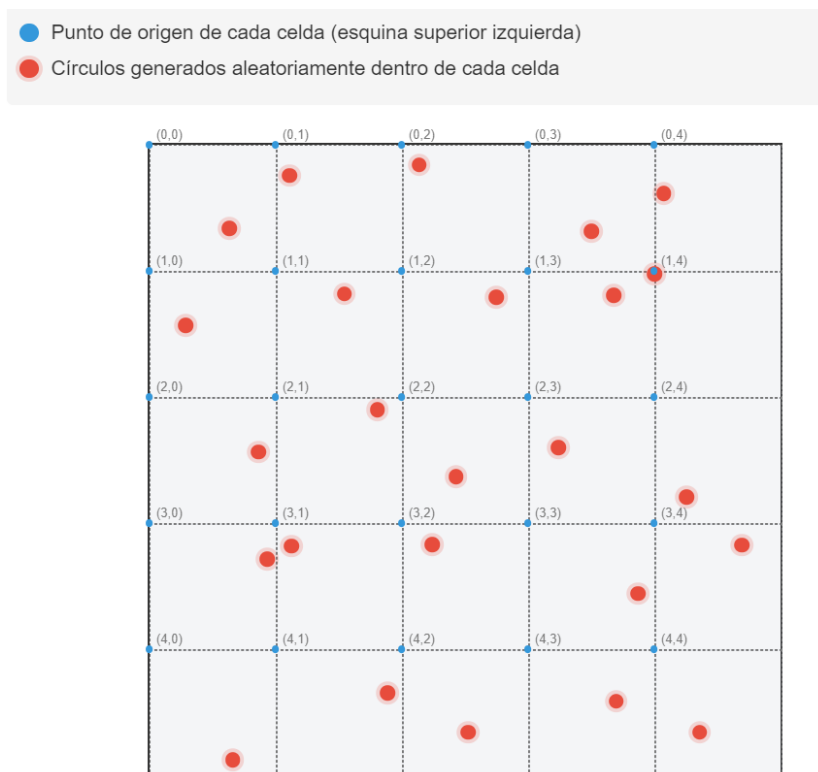


Ilustración 3.4-5 Ejemplo muestra la generación aleatoria de Círculos

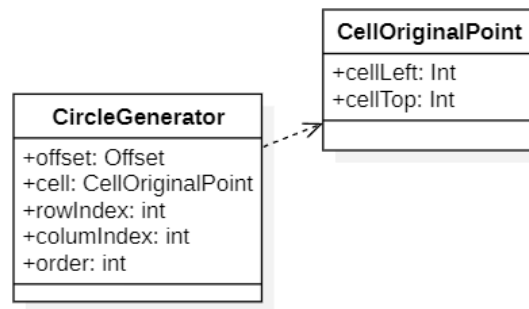


Ilustración 3.4-6 Diagrama de clase de CircleGenerator

3.4.4.3 Algoritmo para reordenar los círculos.

El objetivo de este algoritmo es reordenar los círculos generados aleatoriamente para crear un recorrido lógico y viable para el test TMT, asegurando que, cuando los usuarios conecten los círculos en el orden establecido, no haya otros círculos que actúen como obstáculos entre ellos. Además, garantiza que el orden de los círculos no sea siempre el de círculos cercanos, lo que haría el test demasiado fácil.

1. Análisis de Conexiones Directas

- Para cada círculo, se identifican todos los demás círculos que pueden ser conectados directamente sin que ningún otro círculo bloquee la línea de conexión.

1. Se crea una lista de "círculos conectables" para cada círculo del tablero.

Para saber si una línea recta entre dos círculos intersecta o pasa demasiado cerca de cualquier otro círculo:

- 1) Cálculo de la Distancia entre Círculos:

Se calcula la distancia D entre los centros de los dos círculos a conectar usando la fórmula:

$$D = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Donde (x_1, y_1) y (x_2, y_2) son las coordenadas de los centros de los círculos.

- 2) Verificación de Superposición

Si $D < 2r$ (donde r es el radio de los círculos), los círculos están demasiado cercanos o superpuestos. En este caso, no se considera obstrucción visual y se permite la conexión directa.

3) Cálculo del Vector de Dirección Normalizado

El vector de dirección normalizado \hat{d} se calcula como: $\hat{d} = \left(\frac{x_2 - x_1}{D}, \frac{y_2 - y_1}{D} \right)$

Este vector tiene magnitud 1 y apunta desde el primer círculo hacia el segundo.

4) Cálculo del Vector Perpendicular

El vector perpendicular \hat{p} se calcula como: $\hat{p} = (-\hat{d}_y, \hat{d}_x)$. Este vector es perpendicular al vector de dirección.

5) Análisis de Círculos Potencialmente Obstructores:

Para cada círculo adicional (C) en el tablero, se determina si obstruye la conexión evaluando:

Proyección en la línea: Se calcula la proyección del vector desde el primer círculo al círculo C en la dirección del vector normalizado \hat{d}

Si esta proyección $d_{||}$ está dentro del rango (0, D) entonces el círculo C está entre los dos círculos principales.

Distancia perpendicular: Se calcula la distancia perpendicular d_{\perp} del círculo C a la línea.

Si $d_{\perp} < 2r$ (o más específicamente $d_{\perp} < \frac{w}{2} + r$, donde w es el ancho del corredor), entonces el círculo C está lo suficientemente cerca para obstruir la conexión.

6) Determinación Final

Si se cumplen ambas condiciones (el círculo está entre los puntos de conexión y está lo suficientemente cerca de la línea), entonces se considera que el círculo C obstruye la conexión directa entre los círculos originales.

2. Ordenamiento Inicial por Restricciones

Los círculos se ordenan según la cantidad de conexiones posibles, de menor a mayor número de conexiones.

Este enfoque prioriza los círculos más "restrictivos" (aquellos con menos opciones de conexión).

En la distribución inicial, todos los círculos tienen al menos algunas conexiones disponibles gracias a la generación en cuadrícula con distancias de seguridad.

3. Selección del Primer Círculo

Se selecciona aleatoriamente un círculo para ser el primero en el orden (orden 1).

Si ya existía un círculo designado como orden 1, se intercambia su valor de orden con el del círculo seleccionado.

4. Criterio de Distancia Óptima

Los círculos no deben estar ni demasiado cerca ni demasiado lejos entre sí. Por eso, al seleccionar los siguientes círculos, se considera tanto el número de conexiones como la distancia entre círculos.

Se establece una distancia óptima (aproximadamente 3 veces la distancia mínima de seguridad para móvil, 3,2 para tablet pequeña, 3,5 para tablet mediana y 3,8 para tablet grande).

5. Selección de Círculos Subsiguientes

Para cada posición siguiente en el orden, se selecciona el círculo que:

2. Puede conectarse directamente con el círculo anterior
3. Tiene el menor número de conexiones disponibles con otros círculos.
4. Está a una distancia razonablemente óptima.

6. Mantenimiento del Nuevo Orden

Cada círculo seleccionado se añade a una lista auxiliar que mantiene el nuevo orden.

El valor de orden del círculo se actualiza para reflejar su nueva posición en la secuencia.

7. Actualización Dinámica de Conexiones

- Después de seleccionar cada círculo, se actualiza la lista de conexiones disponibles de todos los círculos restantes.
- Se elimina el círculo recién seleccionado de las listas de conexiones de los demás círculos.

8. Progresión Basada en el Último Círculo

- Para seleccionar cada círculo siguiente, se parte del último círculo colocado.
- Se examinan sus conexiones disponibles y se selecciona aquella con menos opciones de conexión.

9. Continuación del Proceso

Este proceso se repite secuencialmente hasta asignar un orden a todos los círculos o hasta encontrar un punto sin salida.

10. Regeneración en Caso de Bloqueo

- Si un círculo no tiene conexiones disponibles, se intenta regenerar su posición dentro de su celda original.
- Se realizan hasta 5 intentos de regeneración para encontrar una posición que permita las conexiones necesarias.
- Después de cada regeneración, se recalculan todas las conexiones entre círculos.

11. Mecanismo de Seguridad y Finalización

- Si después de 10 intentos completos no se puede encontrar un recorrido viable para todos los círculos, el algoritmo termina.
- Se utiliza el mejor recorrido parcial encontrado hasta ese momento.
- Esto es aceptable, porque una vez que el usuario ha conectado varios círculos, las líneas que cruzan círculos ya conectados no se consideran errores.

3.4.5 Implementación de Funcionalidades Clave

En este apartado se explican algunas implementaciones importantes de la aplicación msDtmt.

3.4.5.1 Flujo General de Home

La pantalla de inicio de la aplicación ha sido diseñada con un flujo de interacción intuitivo que se adapta dinámicamente a la presencia o ausencia de perfiles de usuario preexistentes en la base de datos local. Este enfoque garantiza una experiencia de usuario fluida desde el primer acceso.

El flujo se detalla a continuación:

- **Primer Acceso a la Aplicación (Sin Perfiles Existentes):**

Al iniciar la aplicación, el `SelectUserController` ejecuta una verificación automática en la base de datos a través de `UserProfileDataSource.getAllProfiles()` para determinar si existen perfiles de usuario guardados en la tabla `user_profiles`.

Si el sistema determina que la lista de perfiles está vacía (`allProfiles.isEmpty()`), el componente de selección de perfil (que habitualmente es un desplegable o selector) se transforma en un campo de texto interactivo. Este campo muestra una etiqueta indicativa como "Crear Perfil", guiando al usuario para que registre el primer perfil.

Simultáneamente, la funcionalidad destinada a la introducción de un código de referencia queda bloqueada. Esta restricción se mantiene hasta que, como mínimo, un perfil de usuario haya sido creado, asegurando que las operaciones avanzadas dependan de la existencia de un contexto de usuario válido.

- **Accesos Posteriores: Carga y Gestión del Perfil Existente:**

Cuando un nuevo usuario se registra exitosamente a través de `RegisterUserController.saveProfile()`, el sistema activa automáticamente este perfil como el perfil activo. Esto se logra actualizando `currentProfile.value` e inmediatamente habilitando todas las funcionalidades dependientes. El ID de este perfil activo se almacena de forma persistente utilizando `LocalStorageServices.saveCurrentProfileId()` con la clave `current_profile_id`. Esta acción asegura que el perfil seleccionado permanezca activo incluso después de cerrar y reabrir la aplicación.

En subsiguientes aperturas de la aplicación, el `SelectUserController` ejecuta `loadCurrentProfile()`. Este método recupera el ID de perfil almacenado mediante `LocalStorageServices.getCurrentProfileId()` y valida que corresponda a un perfil existente en la base de datos a través de `UserProfileDataSource.getProfileById()`.

Si tanto los perfiles existen en la base de datos como el `current_profile_id` en `shared_preferences` están presentes, la aplicación carga automáticamente el perfil asociado a dicho ID como el perfil activo.

Una vez que el primer perfil ha sido creado y seleccionado automáticamente, la funcionalidad de introducción de código de referencia se desbloquea, permitiendo su uso.

El siguiente diagrama de flujo ofrece una representación visual del flujo general de la pantalla de inicio (Home), detallando la lógica inicial de la aplicación:

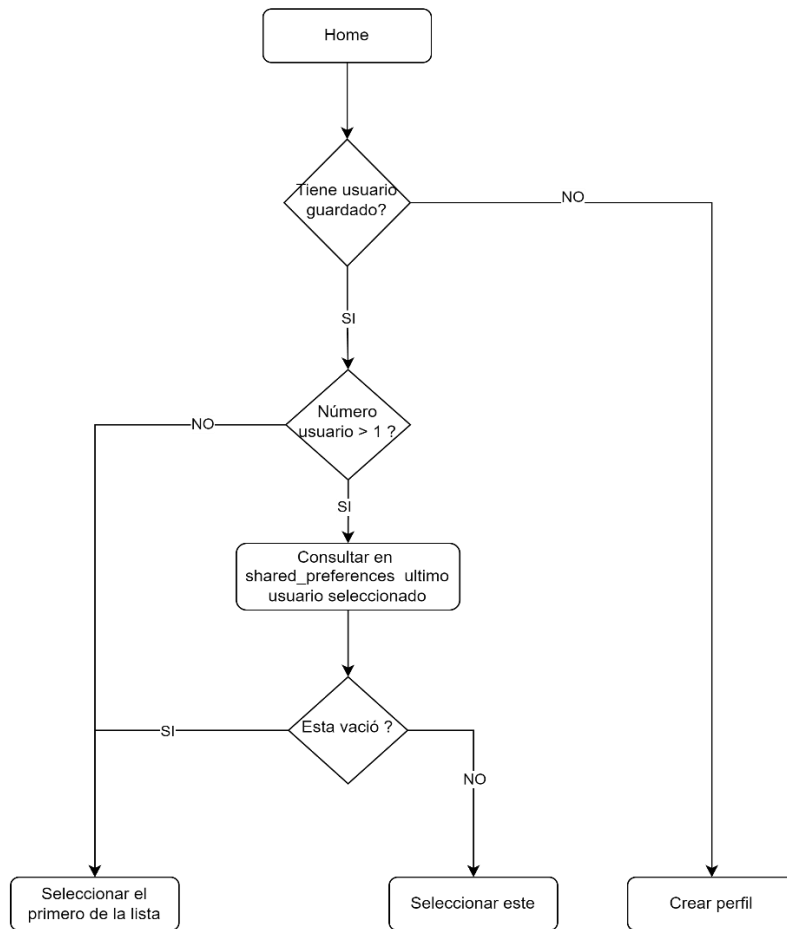


Ilustración 3.4-7 Flujo de Usuario en la Pantalla Principal

3.4.5.2 Gestión de Perfiles de Usuario

La gestión de perfiles de usuario se implementa siguiendo la arquitectura Clean Architecture establecida en el proyecto. La funcionalidad se encuentra organizada dentro del módulo user con la siguiente estructura:

- **Capa de Presentación:** Contiene las pantallas (RegisterUserScreen), controladores (RegisterUserController, SelectUserController) y componentes UI (SelectUserDropdown).
- **Capa de Dominio:** Define las entidades (UserProfile), casos de uso (UserProfileUseCase) y las interfaces de repositorio.
- **Capa de Datos:** Implementa el acceso a datos mediante UserProfileDataSource y los modelos de datos (UserProfileModel).

Proceso de Registro de Usuario

Interfaz de Usuario

La pantalla de registro (RegisterUserScreen) presenta un formulario estructurado que solicita la siguiente información:

- Apodo: Campo de texto obligatorio con validación en tiempo real

- Sexo: Selector de opciones entre "Masculino" y "Femenino"
- Fecha de Nacimiento: Selector de fecha con calendario desplegable
- Nivel de Estudios: Menú desplegable con opciones predefinidas (Primarios, Secundarios, Grado, Máster, Doctorado)

Cuando el usuario completa el formulario y presiona el botón "Guardar", se ejecuta el siguiente proceso:

1. **Validación de Campos:** El RegisterUserController verifica que todos los campos obligatorios estén completos mediante el método `_validateFields()`. Esta validación incluye:
 - Verificación de que el apodo no esté vacío.
 - Confirmación de selección de sexo.
 - Validación de fecha de nacimiento.
 - Verificación de nivel de estudios seleccionado.
2. **Verificación de Unicidad:** El sistema consulta la base de datos local SQLite para comprobar que el apodo introducido no exista previamente. Esta verificación se realiza a través del UserProfileDataSource que ejecuta una consulta SQL específica.
3. **Creación del Perfil:** Si todas las validaciones son exitosas:
 - Se genera un UUID único como identificador del perfil.
 - Se crea una instancia de UserProfile con todos los datos.
 - El perfil se persiste en la base de datos local mediante el método `insertUserProfile()`.
 - Se actualiza el estado global de perfiles disponibles.
4. **Confirmación Visual:**
 - La pantalla se cierra automáticamente al completarse el registro.
 - Se muestra un Snackbar con el mensaje "Perfil registrado correctamente".
 - El usuario es redirigido a la pantalla anterior.

Selección de Perfiles

La selección de perfiles se implementa mediante el componente `SelectUserDropdown` en la pantalla principal:

1. **Visualización:** Un dropdown muestra el perfil actualmente seleccionado o el texto "Crear perfil" si no hay ninguno activo.

2. **Diálogo de Selección:** Al pulsar el dropdown, se abre un diálogo modal que presenta:

- Una lista completa de todos los perfiles registrados.
- Cada perfil muestra su apodo como identificador principal.

3. **Cambio de Perfil:**

- Al seleccionar un perfil diferente, se actualiza inmediatamente el perfil activo.
- La pantalla principal se actualiza para mostrar un saludo personalizado con el apodo del usuario.
- **Persistencia de la Selección:** Guarda automáticamente el ID del perfil seleccionado en `SharedPreferences` mediante `LocalStorageServices.saveCurrentProfileId()`, garantizando que la selección del usuario persista entre sesiones de la aplicación.

4. **Recuperación de la Selección:**

- Al iniciar la aplicación, el `SelectUserController` recupera automáticamente el ID del último perfil seleccionado desde `SharedPreferences` usando `LocalStorageServices.getCurrentProfileId()`.
- Si existe un perfil guardado, se carga automáticamente y se establece como el perfil activo, proporcionando continuidad en la experiencia del usuario sin necesidad de reelegir el perfil en cada sesión.

Búsqueda de Perfiles

Dentro del diálogo de selección se incluye funcionalidad de búsqueda:

1. **Barra de Búsqueda:** Un campo de texto en la parte superior permite filtrar perfiles, controlado mediante un `TextEditingController` que escucha los cambios de texto.
2. **Filtrado en Tiempo Real:**
 - La lista se actualiza dinámicamente mientras el usuario escribe mediante el listener del controlador de texto.
 - El filtro busca coincidencias parciales en los apodos utilizando el método `contains()` combinado con `toLowerCase()` para hacer la búsqueda insensible a mayúsculas.
 - La aplicación mantiene una lista observable (`filteredProfiles.obs`) que se actualiza automáticamente cuando cambia el texto de búsqueda, activando la reactividad de `GetX` para actualizar la UI.
 - Lógica de Filtrado: El método de filtrado itera sobre `allProfiles` y aplica `profile.nickname.toLowerCase().contains(searchQuery.toLowerCase())` para encontrar coincidencias parciales.

- Si no hay resultados después del filtrado, se muestra el mensaje "Ningún perfil coincide con tu búsqueda" mediante una validación que verifica si `filteredProfiles.isEmpty`.

3. Gestión del Estado:

- El `SelectUserController` mantiene tanto la lista completa de perfiles como la lista filtrada, actualizando esta última cada vez que se modifica el término de búsqueda.
- Al cerrar el diálogo, se limpia automáticamente el campo de búsqueda mediante `searchController.clear()` para resetear el estado para futuras búsquedas.

Eliminación de Perfiles

La funcionalidad de eliminación de perfiles de usuario se integra directamente en el diálogo de selección de perfiles, garantizando una gestión centralizada y coherente. El proceso se articula en los siguientes pasos:

1. Identificación del Perfil a Eliminar:

Cada perfil en la lista del diálogo `SelectUserDropdown` muestra un icono de papelera posicionado al lado derecho del apodo del usuario.

2. Confirmación de la Eliminación:

- Al activar el icono de eliminación, el sistema despliega un diálogo de confirmación para prevenir eliminaciones accidentales.
- Este diálogo presenta el mensaje: "*¿Estás seguro de que quieres eliminar este perfil?*", solicitando la validación del usuario.
- Se ofrecen dos opciones explícitas: "Cancelar", para abortar la operación, y "Eliminar", para proceder con la supresión del perfil.

3. Lógica de Eliminación y Gestión del Perfil Activo

El proceso de eliminación implementa la siguiente secuencia lógica a través del `SelectUserController`:

- Si el usuario confirma la eliminación, el sistema procede a suprimir el perfil seleccionado de la base de datos subyacente.
- Se ejecuta la eliminación del perfil mediante `UserProfileDataSource.deleteUserProfile()`, que incluye:
 - Eliminación del registro del perfil de la tabla `user_profiles`.
 - Eliminación en cascada de todos los resultados de tests asociados en la tabla `tmt_results` (mantenimiento de integridad referencial).

4. Manejo del Perfil Actualmente Seleccionado:

Para garantizar la continuidad de la experiencia del usuario, la aplicación implementa una selección automática cuando se elimina el perfil activo:

- Si existen otros perfiles registrados, selecciona automáticamente el primer perfil disponible en orden de creación mediante el método `selectProfile()` del `SelectUserController`.
- Actualización de Estado: La selección automática actualiza inmediatamente:
 - El estado reactivo `currentProfile.value` en el controlador.
 - La persistencia en `SharedPreferences` mediante `LocalStorageServices.saveCurrentProfileId()`.
- Caso sin Perfiles Restantes: Si el perfil eliminado era el último disponible:
 - Se establece `currentProfile.value = null`.
 - Se limpia la selección almacenada con `LocalStorageServices.clearCurrentProfileId()`.
 - La interfaz principal se actualiza para mostrar el estado "Crear perfil".

5. Actualización de la Interfaz y Notificación al Usuario:

- Tras la eliminación y la posible reasignación del perfil activo, la lista de perfiles en el diálogo se actualiza automáticamente para reflejar los cambios.
- Finalmente, se muestra una notificación efímera (`SnackBar`) con el mensaje: "Perfil eliminado correctamente", confirmando al usuario la finalización exitosa de la operación.

3.4.5.3 Flujo de Navegación del Test TMT

Arquitectura de Flujo

El flujo está controlado por la clase `TmtTestNavigationFlowController` que define los siguientes estados mediante el enum `TmtTestNavigationFlow`:

```
enum TmtTestNavigationFlow {  
    START,  
    TMT_A_PRACTICE_IN_PROCESS,  
    TMT_A_IN_PROGRESS,  
    TMT_A_COMPLETED,  
    TMT_B_PRACTICE_IN_PROCESS,  
    TMT_B_IN_PROGRESS,  
    TEST_COMPLETED  
}
```

La siguiente figura Ilustración 3.4-8 Diagrama de Estados del Flujo de Navegación del TMT) ofrece una representación visual de esta máquina de estados. En ella se ilustran los diferentes estados definidos en TmtTestNavigationFlow y las transiciones que pueden ocurrir entre ellos:

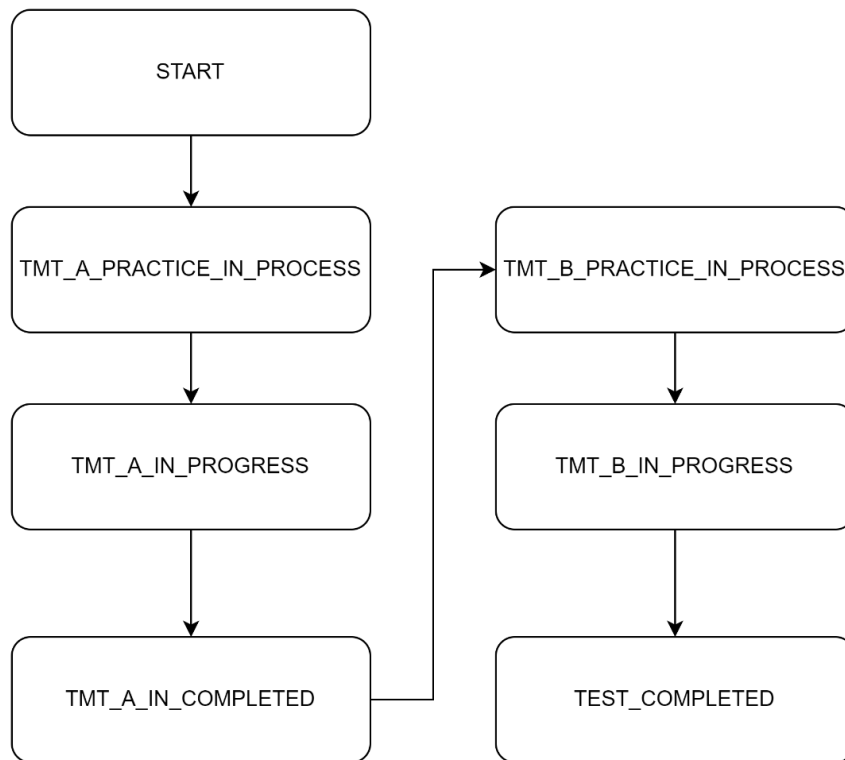


Ilustración 3.4-8 Diagrama de Estados del Flujo de Navegación del TMT

Descripción Detallada del Flujo

El flujo comienza cuando el usuario decide iniciar el TMT test desde la pantalla principal o la sección correspondiente. A partir de ese momento, la secuencia es la siguiente:

1. Inicio del Test (START)

- El usuario inicia el proceso desde la pantalla principal al presionar "Empezar Test TMT"
- Se inicializa el TmtTestNavigationFlowController con estado START
- Se llama al método begin() que establece los datos iniciales del test

2. Instrucción General:

- Al iniciar el test, se presenta al usuario una pantalla de introducción general (gestionada probablemente por tmt_test_help.dart).
- Esta pantalla explica el propósito del TMT, su estructura general (Parte A y Parte B).

- Al presionar "Comenzar", se muestra un diálogo `showTmtSelectHandDialogGetX()` para seleccionar la mano que se debe utilizar para realizar el test. Una vez seleccionada, se avanza al siguiente estado.
3. **Preparación para la Parte A:**
- **Instrucción Específica (Parte A):** Se muestra una pantalla con las instrucciones concretas para la Parte A del test. Se actualiza `TmtTestHelpPage` con modo `TMT_TEST_A`.
 - **Inicio de Práctica (Parte A):** El usuario pulsa un botón para "Comenzar Práctica".
 - **Ejecución de Práctica (Parte A):** Se accede a la página de práctica (`TmtTestPracticePage`) en modo `TMT_A_ONLY`, cargándose la pantalla `tmt_test_practice_screen.dart` configurada para la Parte A. En esta sección, el usuario realiza un ensayo no cronometrado para familiarizarse con la tarea, interactuando con 9 círculos numerados generados específicamente para esta práctica.
4. **Transición al Test Formal (Parte A):**
- **Diálogo de Aviso (Inicio Test A):** Una vez completada la práctica, se muestra un diálogo de aviso. Este diálogo informa al usuario que la práctica ha finalizado y que el test formal de la Parte A está a punto de comenzar, indicándole que se prepare.
 - **Contador Regresivo:** Tras confirmar el diálogo, aparece un contador de 3 segundos (`tmt_count_down_component.dart`) en pantalla. Esto permite al usuario concentrarse antes de que el cronómetro se active.
 - **Ejecución del Test (Parte A):** Una vez finalizada la cuenta atrás, comienza el test formal parte A, presentando en la pantalla `tmt_test_screen.dart` (configurada para la Parte A) 15 o 25 círculos, según corresponda. Se activa el cronómetro en la barra superior y se registran todas las métricas del usuario.
5. **Transición a la Parte B:**
- **Diálogo de Aviso (Fin Test A / Inicio Test B):** Al finalizar la Parte A, aparece otro **diálogo de aviso**. Este informa que la Parte A ha concluido y da paso a las instrucciones y práctica de la Parte B.
6. **Preparación para la Parte B:**
- **Instrucción Específica (Parte B):** Se muestran las instrucciones para la Parte B (conectar números y letras alternativamente: 1-A-2-B....).
 - **Inicio de Práctica (Parte B):** El usuario pulsa "Comenzar Práctica".
 - **Ejecución de Práctica (Parte B):** A continuación, se navega a `TmtTestPracticePage` en modo `TMT_B_ONLY`, cargando la pantalla `tmt_test_practice_screen.dart`, esta vez configurada para la Parte B. Aquí se

generan 9 círculos que alternan números y letras, permitiendo al usuario practicar la nueva secuencia requerida.

7. Transición al Test Formal (Parte B):

- **Diálogo de Aviso (Inicio Test B):** Similar al paso 3, un diálogo advierte que la práctica ha terminado y el test formal de la Parte B va a empezar.
- **Contador Regresivo:** De nuevo, se muestra el contador de 3 segundos.
- **Ejecución del Test (Parte B):** Se presenta `tmt_test_screen.dart`, configurada para la Parte B. El usuario realiza la prueba cronometrada.

8. Finalización y Resultados:

- **Navegación a Resultados:** Una vez completada la Parte B, la aplicación navega automáticamente a la pantalla de resultados (`tmt_test_result_screen.dart`).
- **Presentación de Resultados:** En esta pantalla final se presentan los resultados, mostrando los tiempos y errores registrados en ambas partes (A y B). A continuación, estos datos se envían automáticamente al servidor y, simultáneamente, se guardan de forma local.

3.4.5.4 Cálculo de Métricas del Test TMT

La implementación de la recolección de métricas se inspira en la tesis (*An Analysis of a Digital Variant of the Trail Making Test Using Machine Learning Techniques*)^[32], como la presentada por Dahmen. El objetivo es capturar no solo los resultados tradicionales (tiempo y errores), sino también datos más detallados sobre la ejecución, que pueden ofrecer una visión más profunda del rendimiento cognitivo.

El núcleo de la recolección de datos durante el test TMT se centraliza en la clase `TmtMetricsController`. Esta clase actúa como un orquestador que registra todas las interacciones del usuario y los eventos relevantes que ocurren durante la prueba, tanto para la Parte A como para la Parte B del test. Su diseño, como se puede observar en el diagrama de clases (*Ilustración 3.4-9 Diagrama de clase TmtMetricController*), le permite gestionar diversas métricas específicas.

Como se ilustra en el diagrama, `TmtMetricsController` gestiona y agrega información de varias clases especializadas en métricas:

Tiempo (`TmtTestTimeMetric`):

- Se registra el tiempo total del test, así como los tiempos específicos para la Parte A y la Parte B.
- La clase `TmtTestTimeMetric` es responsable de registrar los momentos clave del test. Al inicio, captura el tiempo actual usando `DateTime.now()` y lo almacena en `timeStartTest`. De manera similar, registra `timeStartTmtA` y `timeStartTmtB` al comenzar cada parte, y `timeEndTmtA`, `timeEndTmtB` y `timeEndTest` al finalizarlas.
- Posteriormente, los métodos como `calculateTimeCompleteTest`, `calculateTimeCompleteTmtA` y `calculateTimeCompleteTmtB` calculan la

diferencia entre estos tiempos (la duración) y la convierten de milisegundos a segundos para su envío y análisis.

Número de Errores (TmtMetricsController):

- Se contabiliza cada vez que el usuario conecta un círculo incorrecto.
- La variable numberError en TmtMetricsController lleva la cuenta total, mientras que numberErrorA y numberErrorB lo hacen por cada parte. Se incrementan mediante onConnectNextCircleError.

Métricas de Pausas (TmtTestPauseMetric):

- Se registra el número total de pausas y su duración promedio.
- Siguiendo el criterio de la tesis una pausa se define como mantener el puntero en la misma área (tolerancia de 5 píxeles) por más de 0.1 segundos (100 ms).
- La clase TmtTestPauseMetric.dart implementa esta lógica:
 - Usa `_pauseThresholdDuration` (100 ms) y `MetricStaticValues.PAUSE_POSITION_TOLERANCE`.
 - `checkPauseStatus` verifica si el movimiento supera la tolerancia; si no, `_startTimer` inicia un temporizador.
 - Si el temporizador se cumple (`_onStayDetected`), se incrementa `numberPause` y se registra el tiempo de inicio.
 - `onEndPause` calcula la duración al salir del estado de pausa.
 - `calculateAveragePause` obtiene la duración promedio.

Métricas de Levantamientos (TmtTestLiftMetric):

- Se mide el número de veces que el usuario levanta el lápiz/dedo y la duración promedio de estos levantamientos.
- `TmtTestLiftMetric.dart` gestiona esto: `onStartLift` registra el inicio al levantar (`onPanEnd` en `TmtMetricsController`) y `onEndLift` calcula la duración al volver a tocar (`onPanStart` en `TmtMetricsController`), incrementando `numberLift` y `totalLiftTime`. `calculateAverageLift` calcula el promedio.

Métricas de Trazado entre Círculos (TmtCircleMetrics):

- Se calcula la velocidad y el tiempo medios empleado al dibujar entre círculos. La velocidad se define como la distancia en línea recta (salida a entrada) dividida por el tiempo.
- `TmtCircleMetrics` implementa esto. `onConnectNextCircleCorrect` calcula la duración y la distancia (descontando los radios) entre círculos correctos. `calculateAverageRateBetweenCircles` y `calculateAverageTimeBetweenCircles` realizan los cálculos promedio.

Métricas de Trazado dentro de Círculos (TmtCircleMetrics):

- Se calcula la velocidad y el tiempo medios empleado mientras se dibuja dentro de los círculos.
- TmtCircleMetrics maneja esto con dragOnInsideCircle y dragEndInsideCircle, registrando el tiempo y la distancia del trazado interno. calculateAverageRateInsideCircles y calculateAverageTimeInsideCircles calculan los promedios.

Métricas Específicas de la Parte B (TmtBMetrics):

- Para la Parte B, se miden específicamente los tiempos y velocidades medias antes de tocar un círculo con letra y antes de tocar uno con número.
- TmtBMetrics.dart se encarga de esto. onConnectLetterCircle discrimina si el círculo es letra o número y almacena las métricas (distancia y duración) en _beforeLetterMetricsList o _beforeNumberMetricsList. Los métodos calculateAverage computan los valores finales.

Métricas de Presión y Tamaño (TmtPressureSizeMetric):

- Se intenta registrar la presión promedio y el tamaño promedio del área de contacto durante el test. Es importante destacar que la disponibilidad de estas métricas depende directamente de si el hardware del dispositivo (tablet o móvil) incorpora los sensores necesarios para detectar la presión y el tamaño del contacto.
- La clase TmtPressureSizeMetric recibe eventos de onPointerMove (desde TmtMetricsController) y almacena los valores si son válidos. En caso de que el dispositivo no soporte estas mediciones o no se detecten valores, los métodos calculateAverageTotalPressure y calculateAverageTotalSize están diseñados para devolver un valor de -1, indicando así la ausencia de esta información específica en los resultados enviados.

Descomposición del Test en Secciones

Inspirados por la metodología de descomposición de la tesis (*Decomposition of the Trail Making Test*)^[33] propuesta por Poreh, que sugiere dividir el TMT en secciones para un análisis más granular del rendimiento, se ha implementado una métrica de tiempo por secciones. El TMT (tanto A como B) se divide en 5 secciones, y se calcula el tiempo empleado en completar cada una.

- **TMT A:** 1-5, 6-10, 11-15, 16-20, 21-25.
- **TMT B:** 1A2B3, C4D5E, 6F7G8, H9I10J, 11K12L13.

Esta funcionalidad se implementa en TmtTestTimeMetric. Los métodos recordTmtACircleTime y recordTmtBCircleTime guardan la marca de tiempo al alcanzar cada círculo. El método _calculateAllScores calcula la diferencia de tiempo entre los círculos límite de cada sección (definidos en MetricStaticValues) y almacena los resultados en las variables _scoreA1 a _scoreA5 y _scoreB1 a _scoreB5. Estos scores representan el tiempo (en segundos) para completar cada una de las 5 secciones, permitiendo un análisis de la curva de rendimiento durante el test,

Flujo de Recolección de Métricas:

El proceso de captura y procesamiento de métricas durante la ejecución del test TMT sigue un flujo bien definido, coordinado entre la interfaz de usuario (Widgets de Flutter), el controlador del tablero de juego (TmtGameBoardController) y el controlador de métricas (TmtMetricsController):

1. **Inicio del Test:** Al comenzar el test, se invoca al método `startTest` del `TmtMetricsController`, que a su vez inicia los cronómetros y prepara las estructuras de datos para recolectar las métricas.
2. **Captura de Interacción:**
 - La pantalla del test (`TmtTestScreen`) utiliza Widgets de Flutter para detectar la interacción del usuario:
 - `GestureDetector`: Observa los gestos de arrastre (`onPanStart`, `onPanUpdate`, `onPanEnd`), que son cruciales para detectar el trazado, el inicio y el fin del contacto (y, por tanto, los levantamientos).
 - `Listener`: Observa eventos de puntero más detallados, específicamente `onPointerMove`. Este evento es fundamental para capturar datos de presión y tamaño, ya que se dispara continuamente mientras el puntero se mueve sobre la pantalla y proporciona acceso a `event.pressure` y `event.size` (clase `PointerMoveEvent` de Flutter).
 - Todos estos eventos capturados por los Widgets de Flutter se redirigen inmediatamente a los métodos correspondientes en el `TmtGameBoardController`.
3. **Procesamiento y Validación:**

El `TmtGameBoardController` actúa como el cerebro del juego, ya que recibe los eventos brutos y los gestiona de varias formas. Por un lado, delega a Métricas, llamando a los métodos `onPanStart`, `onPanUpdate`, `onPanEnd` y `onPointerMove` del `TmtMetricsController`, lo que permite que las clases `TmtTestLiftMetric`, `TmtTestPauseMetric` y `TmtPressureSizeMetric` se actualicen en tiempo real con cada interacción del usuario.

Además, controla el trazado, manteniendo un registro del dibujo que el usuario realiza en la pantalla. Finalmente, verifica las conexiones, comprobando si, al entrar en el área de un círculo, este corresponde al siguiente en la secuencia correcta del juego.

4. Notificación de Aciertos/Errores

- Si la conexión es correcta: El `TmtGameBoardController` invoca `onConnectNextCircleCorrect` en el `TmtMetricsController`. Esto actualiza `TmtCircleMetrics` (calculando tiempo y velocidad entre círculos) y notifica a `TmtTestTimeMetric` para que registre el tiempo de alcance del círculo (`recordTmtACircleTime` o `recordTmtBCircleTime`), lo cual es vital para los scores por secciones.

- Si la conexión es incorrecta: El TmtGameBoardController invoca onConnectNextCircleError en el TmtMetricsController, lo que incrementa los contadores de errores (numberError, numberErrorA o numberErrorB).

5. Pausas

La clase TmtTestPauseMetric (actualizada a través de onPanUpdate y onPanStart) detecta automáticamente las pausas basándose en la falta de movimiento durante un umbral de tiempo, como se describió anteriormente.

6. Fin del Test:

- Al completar el último círculo o cuando finaliza el test por otra razón, se llama al método onTestEnd del TmtMetricsController.
- Este método registra los tiempos finales (timeEndTmtA, timeEndTmtB, timeEndTest) y se asegura de que cualquier métrica pendiente se cierre correctamente.

7. Generación de Datos:

Finalmente, una vez que TmtMetricsController contiene todas las métricas consolidadas, se utiliza el TmtResultMetricModel. Este modelo toma la instancia del TmtMetricsController, extrae todos los valores calculados y utiliza su método toJson para serializarlos en un formato estructurado (JSON).

Este formato JSON se adhiere estrictamente a la estructura de datos definida en el apartado 3.4.3.2 (*Endpoint: Envío de Resultados del Test TMT*) de esta memoria, asegurando que los datos enviados sean compatibles con la API y su backend, tal como se detalla en la tabla correspondiente de dicho apartado. De esta manera, los datos quedan listos para ser enviados a través de la API.

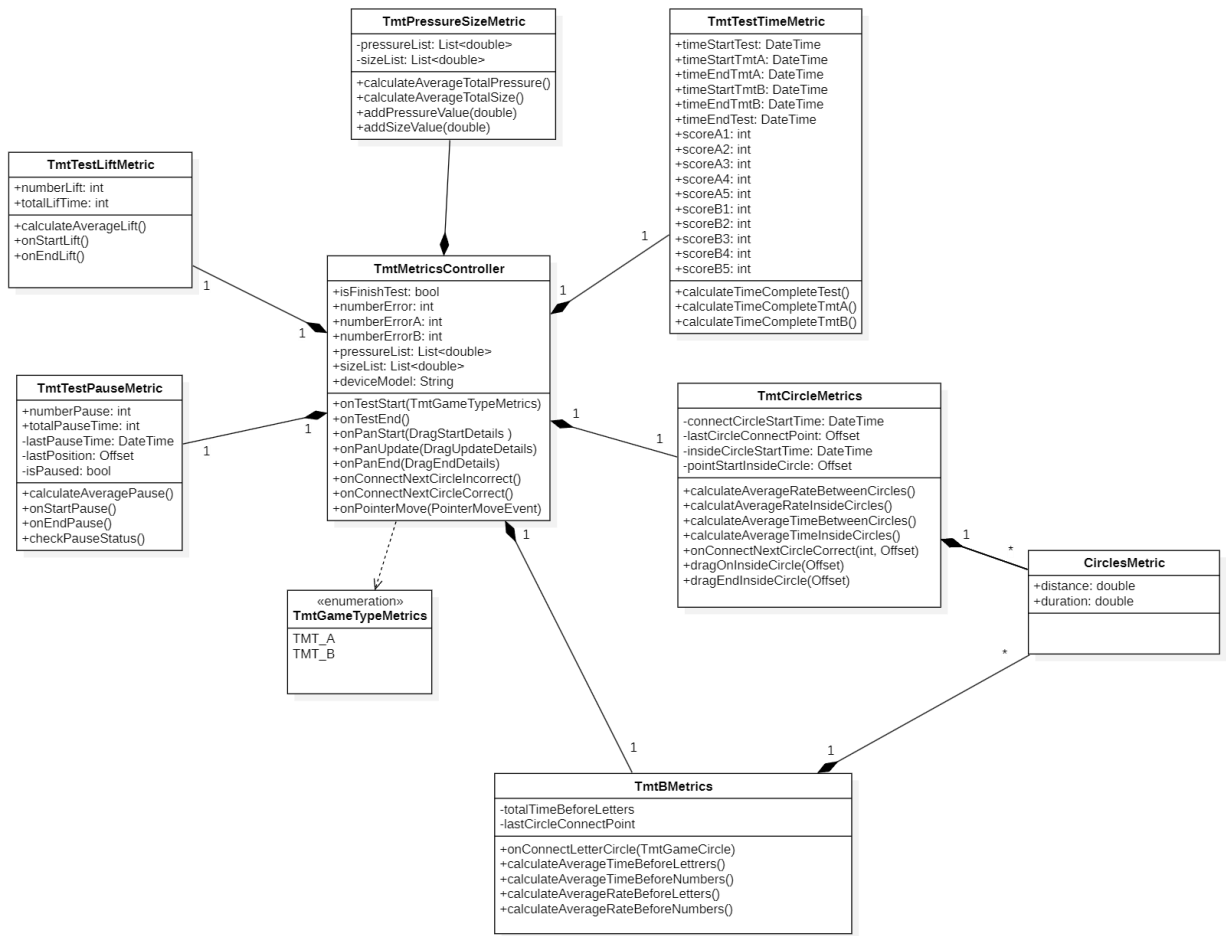


Ilustración 3.4-9 Diagrama de clase TmtMetricController

3.4.5.5 Mecanismo de Almacenamiento de Resultados cuando no hay Internet

Para garantizar que no se pierdan los resultados de los tests TMT realizados en entornos sin conectividad a Internet o cuando el servidor no está disponible, se ha implementado un mecanismo de almacenamiento y envío diferido. Esta funcionalidad asegura la persistencia de los datos y su posterior sincronización con el servidor una vez que se restablece la conexión.

El proceso se activa en el momento de intentar enviar un resultado al servidor:

1. Detección de Falta de Conexión o Fallo del Servidor:

Cuando el usuario completa el test TMT, el sistema intenta enviar automáticamente los resultados al servidor a través del ReportTmtResultsUseCase. Si durante este proceso se detecta un problema de conectividad (clasificado como NetworkError) o si el servidor no responde adecuadamente, se activa el mecanismo de respaldo.

Es importante destacar que la aplicación diferencia entre errores de red y errores del servidor: solo los errores de conectividad activan el almacenamiento temporal, mientras que los errores del servidor (que indican que el resultado no sería válido) no generan respaldo para evitar enviar datos corruptos posteriormente.

2. Almacenamiento Local Temporal:

Cuando se detecta un problema de conectividad, el sistema ejecuta `pendingResultUseCase.savePendingResult()` que convierte los datos del test a formato JSON mediante `TmtResultModel.fromEntity()` y `TmtUserModel.fromUserProfile()`. Estos datos se encapsulan en un objeto `PendingResultData` que incluye tanto la información del resultado como los datos del usuario y un timestamp.

El `PendingResultRepositoryImpl` utiliza `LocalStorageServices` para almacenar estos datos en `SharedPreferences` como una lista de strings JSON, permitiendo acumular múltiples resultados pendientes si es necesario.

3. Creación y Gestión de Tarea en Segundo Plano (WorkManager):

Una vez almacenado exitosamente el resultado pendiente, se ejecuta `WorkManagerHandler.scheduleResultsSending()` que programa una tarea en segundo plano. Esta tarea está diseñada para ejecutarse de forma periódica y verificar tanto el estado de la conectividad como la existencia de resultados pendientes de envío. El `WorkManager` garantiza que la tarea se ejecute incluso si la aplicación está cerrada o el dispositivo se reinicia, proporcionando una solución robusta para la sincronización diferida.

4. Proceso de Envío Diferido:

Una vez que se cumplen todas las condiciones para ejecutar la tarea programada, se intentará llamar al método `sendPendingResults()`. Primero, se utiliza `Connectivity().checkConnectivity()` para verificar el estado de la conexión a internet. Si ya hay conexión a internet, la aplicación recupera la lista de resultados de métricas del test TMT pendientes de enviar que están almacenados en `SharedPreferences`. Para cada resultado pendiente, se intenta el envío utilizando los datos JSON previamente almacenados a través de `reportTmtResults()`.

En caso de que cada registro de métrica de dicha lista sea enviado exitosamente, para garantizar la sincronización, primero se eliminará dicho registro de la lista a través del método `deletePendingResult()`. Posteriormente, convierte los datos del resultado pendiente a `UserTestLocalDataResult` y los almacena en la base de datos local mediante `testResultLocalDataRepositoryImpl.saveTestResult()`, permitiendo que el resultado aparezca en el historial del usuario. Esta separación garantiza que solo los resultados confirmadamente recibidos por el servidor se reflejen en el historial local.

En caso de fallo durante el reintento, si es por problemas de red (`NetworkError`), el resultado permanece pendiente; si es por otra causa, se elimina para evitar acumular datos irrecuperables.

Finalmente, para una gestión eficiente, si no quedan resultados pendientes, `_clearAllPendingResults()` limpia la lista y `WorkManagerHandler.cancelWorkTask()` cancela la tarea programada, evitando verificaciones innecesarias y optimizando el uso de recursos.

3.4.6 Herramientas y entorno de desarrollo

Para la planificación, diseño, desarrollo, pruebas y documentación de este proyecto, se ha hecho uso de un conjunto diverso de herramientas y tecnologías que han facilitado cada una de las fases del ciclo de vida del software. A continuación, se describen las principales herramientas empleadas:

3.4.6.1 Lenguaje de Programación y Framework

Dart

Dart es un lenguaje optimizado para cliente, diseñado para desarrollar aplicaciones rápidas en cualquier plataforma. Su objetivo es ofrecer el lenguaje de programación más productivo para el desarrollo multiplataforma, junto con una plataforma de ejecución flexible para los frameworks de aplicaciones. ^[34]

Los lenguajes se definen por su envolvente técnica (las decisiones tomadas durante su desarrollo que moldean las capacidades y fortalezas del lenguaje). Dart está diseñado para una envolvente técnica particularmente adecuada para el desarrollo de clientes, priorizando tanto la experiencia de desarrollo (recarga en caliente con estado en menos de un segundo) como experiencias de producción de alta calidad en una amplia variedad de destinos de compilación (web, móvil y escritorio).

Dart también es la base de Flutter. Dart proporciona el lenguaje y los entornos de ejecución que impulsan las aplicaciones de Flutter, pero también soporta muchas tareas fundamentales para los desarrolladores, como el formateo, análisis y prueba de código.

Para desarrollar la aplicación msDtmt se ha utilizado este lenguaje de programación.

Flutter ^[35]

Flutter es un kit de herramientas de interfaz de usuario multiplataforma diseñado para permitir la reutilización de código en diferentes sistemas operativos como iOS, Android, web y escritorio, al mismo tiempo que permite que las aplicaciones interactúen directamente con los servicios de la plataforma subyacente. El objetivo es que los desarrolladores puedan entregar aplicaciones de alto rendimiento que se sientan naturales en distintas plataformas, aceptando las diferencias cuando existan, pero compartiendo la mayor cantidad de código posible. ¡Error! No se encuentra el origen de la referencia.

Durante el desarrollo, las aplicaciones de Flutter se ejecutan en una máquina virtual (VM) que ofrece recarga en caliente con estado de los cambios sin necesidad de una recompilación completa. (En la web, Flutter soporta el reinicio en caliente y la recarga en caliente detrás de una flag.) Para la versión de lanzamiento, las aplicaciones de Flutter se compilan directamente a código máquina, ya sean instrucciones Intel x64 o ARM, o a JavaScript si el objetivo es la web.

Como se ha mencionado anteriormente, la aplicación msDtmt de este proyecto se ha desarrollado en Flutter. Esta tecnología es muy avanzada, lo que le permite ser compatible con diferentes resoluciones de pantalla y sistemas operativos, con el fin de cumplir los requisitos del proyecto.

3.4.6.2 Entorno de Desarrollo Integrado (IDE)

Android Studio

Android Studio es el entorno de desarrollo integrado (IDE) oficial que se usa en el desarrollo de apps para Android. Basado en el potente editor de código y las herramientas para desarrolladores de IntelliJ IDEA, Android Studio ofrece aún más funciones que mejoran tu productividad cuando compilas apps para Android, como las siguientes: ^[36]

- Un sistema de compilación flexible basado en Gradle.
- Un emulador rápido y cargado de funciones.
- Un entorno unificado donde puedes desarrollar para todos los dispositivos Android.
- Ediciones en vivo para actualizar elementos componibles en emuladores y dispositivos físicos, en tiempo real.
- Integración con GitHub y plantillas de código para ayudarte a compilar funciones de apps comunes y también importar código de muestra.
- Variedad de marcos de trabajo y herramientas de prueba.
- Herramientas de Lint para identificar problemas de rendimiento, usabilidad y compatibilidad de versiones, entre otros.
- Compatibilidad con C++ y NDK.
- Compatibilidad integrada con Google Cloud Platform, que facilita la integración con Google Cloud Messaging y App Engine.

Aunque Android Studio, como su nombre indica, está diseñado para desarrollar aplicaciones en Android, también ofrece un conjunto de herramientas para Flutter, lo que lo hace compatible y amigable para el desarrollo en esta plataforma. Además, gracias a su emulador de Android, es fácil depurar la aplicación de Flutter directamente en él.

Xcode ^[37]

Xcode es un conjunto de herramientas para crear aplicaciones para las plataformas de Apple. Permite gestionar todo el flujo de trabajo de desarrollo, desde la creación de la aplicación hasta su prueba, optimización y envío a la App Store. Xcode incluye un editor de código de primera clase, herramientas de previsualización de SwiftUI integradas que muestran la interfaz de usuario de la aplicación a medida que se modifica el código, y un potente depurador con puntos de interrupción condicionales.

Xcode también incluye varias herramientas de desarrollo para facilitar el prototipado y la prueba rápida de las aplicaciones. Se utiliza Simulator para el prototipado rápido y para probar la aplicación en un entorno simulado cuando no se dispone de un dispositivo real. Se emplea Instruments para perfilar y analizar la aplicación, mejorar el rendimiento

e investigar el uso de recursos del sistema. Es posible construir contenido 3D con Reality Composer, entrenar modelos de aprendizaje automático personalizados con Create ML, e identificar áreas de la aplicación que no son accesibles con Accessibility Inspector.

En este caso, para garantizar el correcto funcionamiento de la aplicación msDtmt en iOS, se ha utilizado Xcode para probar diferentes simuladores de iOS, tanto de iPhone como de iPad.

3.4.6.3 Diseño y Prototipado

Figma

Figma es la plataforma en línea gratuita en la que las personas diseñan conjuntamente los mejores sitios web del mundo. Desde esquemas de página hasta diseños, pasando por los prototipos y el desarrollo, Figma facilita el diseño web.^[38]

En Figma se dispone de varias herramientas:^[39]

- **Figma Design:** permite a las personas crear, compartir y probar diseños para sitios web, aplicaciones móviles y otros productos y experiencias digitales. Es una herramienta popular para diseñadores, gerentes de producto, redactores y desarrolladores, y ayuda a cualquier persona involucrada en el proceso de diseño a contribuir, dar retroalimentación y tomar mejores decisiones, más rápido.
- **Dev Mode** (Modo de Desarrollo) desbloquea un conjunto de funciones dentro de Figma Design que ayudan a transformar los diseños en código. Con Dev Mode, los diseñadores y desarrolladores pueden mantenerse sincronizados, asegurando que los detalles importantes no se pierdan en el proceso de entrega.
- **Figma Slides** es una herramienta para crear presentaciones que permite a los usuarios y a sus equipos diseñar atractivas diapositivas. Se pueden insertar diseños y prototipos desde Figma Design, cocrear y presentar con confianza usando sus herramientas de colaboración y presentación, y utilizar widgets interactivos para recopilar comentarios de los interesados incluso antes de que finalice la presentación.

En este caso, se ha utilizado Figma para la creación de los diseños de pantalla y prototipos interactivos (mockups) de la aplicación antes de programarla. De esta manera, además de ayudar a entender los requisitos, se pueden editar los diseños sobre la marcha antes de programar.

StartUML

Es un software de modelado UML (Lenguaje Unificado de Modelado) empleado para la creación de diagramas estructurales y de comportamiento del sistema, como :

- Diagramas de casos de uso
- Diagramas de clase
- Diagramas de secuencia
- Diagramas de componentes

- Diagramas de despliegue
- Diagramas de estado
- Diagramas de colaboración

Estos modelos han sido cruciales para definir la arquitectura de la aplicación y las interacciones entre sus componentes.

En este proyecto se ha utilizado StartUML para crear diferentes diagramas de clases. Crear diagramas de clases antes de programar ayuda a clarificar la estructura del proyecto y la interacción entre las clases.

Draw.io (diagrams.net)

Draw.io es un software de diagramación en línea gratuito para crear diagramas de flujo, diagramas de procesos, organigramas, UML, diagramas ER y diagramas de red.

Para este proyecto, se ha utilizado Draw.io para crear diagramas de flujo que representan la lógica de los procesos, diagramas de entidad-relación (ERD) para el diseño de la base de datos y el modelo relacional derivado.

3.4.6.4 Pruebas de API

Postman

Postman es una plataforma API todo en uno para construir y trabajar con APIs. Elimina las dificultades de cada etapa del ciclo de vida de una API, desde el diseño y las pruebas hasta la entrega y la monitorización. Diseñado para equipos, Postman facilita la colaboración, el mantenimiento de la organización y la creación de APIs seguras y fiables de forma más rápida.^[40]

Se ha utilizado Postman para realizar las pruebas de los endpoints de la aplicación msDtmt. De esta manera, se ha podido verificar su correcto funcionamiento, analizar las respuestas y depurar la integración entre el cliente móvil y el servidor.

4 Resultado

4.1 Pruebas

Para garantizar la calidad, fiabilidad y el correcto funcionamiento de la aplicación msDtmt, es fundamental implementar una fase de pruebas rigurosa. Este proceso tiene como objetivo principal verificar que la aplicación cumple con todos los requisitos funcionales y no funcionales definidos previamente en la fase de análisis.

Por lo tanto, antes de su despliegue final, se ha llevado a cabo un plan de pruebas sistemático que abarca diferentes niveles del sistema. En las siguientes secciones se detallan las pruebas realizadas.

4.1.1 Prueba de Gestión de Usuario

En este apartado se detallan las pruebas de extremo a extremo (End-to-End) que se han realizado para validar el ciclo de vida completo de la gestión de perfiles de usuario. El objetivo de estas pruebas es verificar que las funcionalidades de creación, selección, visualización y eliminación de perfiles operan correctamente desde la interfaz de usuario hasta la base de datos local, en cumplimiento con los requisitos funcionales del sistema.

Caso de Prueba 1: Crear Usuario	
Caso de prueba, ID	TC-USER-001
Descripción	Crear un nuevo perfil de usuario con datos válidos.
Pre-condiciones	<ul style="list-style-type: none">▪ Se navega a la pantalla de registro de usuario.▪ No existe un usuario con el mismo nickname.
Secuencia de eventos	
Eventos de entrada	Eventos de salida
El usuario ingresa un nickname válido en el campo "Apodo".	El campo acepta el texto y no muestra errores.

El usuario selecciona el sexo (Masculino/Femenino).	El radio button se marca correctamente.
El usuario toca el campo de fecha de nacimiento.	Se abre el selector de fecha.
El usuario selecciona una fecha válida en el date picker.	La fecha se muestra en el campo con formato localizado.
El usuario selecciona un nivel educativo del dropdown.	El nivel educativo se muestra en el campo.
El usuario presiona el botón "Guardar".	<ul style="list-style-type: none"> ▪ Se muestra un indicador de carga. ▪ El perfil se guarda en la base de datos. ▪ Se muestra un snackbar de éxito. ▪ Se navega de vuelta a la pantalla anterior.
Post-condiciones	<ul style="list-style-type: none"> ▪ El nuevo usuario aparece en la lista de perfiles. ▪ Se puede seleccionar como usuario actual. ▪ Todos los campos del formulario están validados correctamente.

Tabla 4.1-1 Caso de Prueba 1: Crear Usuario

Caso de Prueba 2: Crear Usuario con Nickname Duplicado	
Caso de prueba, ID	TC-USER-002
Descripción	Intentar crear un usuario con un nickname que ya existe.

Pre-condiciones	<ul style="list-style-type: none"> ▪ Existe al menos un usuario registrado. ▪ Se navega a la pantalla de registro de usuario.
Secuencia de eventos	
Eventos de entrada	Eventos de salida
El usuario ingresa un nickname que ya existe.	El campo acepta el texto inicialmente.
El usuario completa todos los demás campos requeridos.	Los campos se completan correctamente.
El usuario presiona el botón "Guardar".	<ul style="list-style-type: none"> ▪ Se muestra un indicador de carga. ▪ Se valida contra la base de datos. ▪ Se muestra error de nickname duplicado. ▪ Se muestra snackbar rojo con mensaje de error. ▪ El formulario se marca como inválido.
Post-condiciones	<ul style="list-style-type: none"> ▪ No se crea ningún usuario nuevo. ▪ El formulario permanece en pantalla con el error mostrado. ▪ El usuario puede corregir el nickname.

Tabla 4.1-2 Caso de Prueba 2: Crear Usuario con Nickname Duplicado

Caso de Prueba 3: Eliminar Usuario (Último Perfil)

Caso de prueba, ID	TC-USER-003
Descripción	Eliminar el último usuario existente en la aplicación.
Pre-condiciones	<ul style="list-style-type: none"> ▪ Existe exactamente un usuario registrado. ▪ Se navega a la pantalla de datos del usuario actual.
Secuencia de eventos	
Eventos de entrada	Eventos de salida
El usuario presiona el botón "Eliminar Perfil".	Se muestra un diálogo de confirmación con título "Confirmar eliminación".
El usuario lee el mensaje de confirmación.	El diálogo muestra el contenido explicativo.
El usuario presiona "Confirmar" en el diálogo.	<ul style="list-style-type: none"> ▪ Se cierra el diálogo. ▪ Se muestra indicador de carga. ▪ Se elimina el perfil de la base de datos. ▪ Se eliminan todos los resultados de pruebas asociados. ▪ Se navega a la pantalla Home. ▪ Se muestra snackbar verde de éxito. ▪ El selector de usuario de la pantalla de Home cambia a un cuadro donde pone 'Crear perfil'.
Post-condiciones	<ul style="list-style-type: none"> ▪ No hay usuarios registrados en la aplicación. ▪ No hay usuario actual seleccionado. ▪ La aplicación está lista para registrar un nuevo usuario

Tabla 4.1-3 Caso de Prueba 3: Eliminar Usuario (Último Perfil)

Caso de Prueba 4: Eliminar Usuario (Múltiples Perfiles)	
Caso de prueba, ID	TC-USER-004
Descripción	Eliminar un usuario cuando existen múltiples perfiles.
Pre-condiciones	<ul style="list-style-type: none"> ▪ Existen al menos 2 usuarios registrados. ▪ Se navega a la pantalla de datos del usuario actual.
Secuencia de eventos	
Eventos de entrada	Eventos de salida
El usuario presiona el botón "Eliminar Perfil"	Se muestra un diálogo de confirmación.
El usuario presiona "Confirmar" en el diálogo	<ul style="list-style-type: none"> ▪ Se elimina el perfil actual. ▪ Se eliminan los resultados de test tmt asociados. ▪ Se navega de vuelta a la pantalla anterior. ▪ Se muestra snackbar de éxito. ▪ Se selecciona automáticamente el último usuario creado de la base de datos.
Post-condiciones	<ul style="list-style-type: none"> ▪ El usuario eliminado ya no aparece en las listas. ▪ Otro usuario queda seleccionado como actual.

Tabla 4.1-4 Caso de Prueba 4: Eliminar Usuario (Múltiples Perfiles)

Caso de Prueba 5: Cancelar Eliminación de Usuario	
Caso de prueba, ID	TC-USER-005
Descripción	Cancelar la eliminación de un usuario.
Pre-condiciones	<ul style="list-style-type: none"> ▪ Existe al menos un usuario registrado. ▪ Se navega a la pantalla de datos del usuario actual.
Secuencia de eventos	
Eventos de entrada	Eventos de salida
El usuario presiona el botón "Eliminar Perfil"	Se muestra el diálogo de confirmación
El usuario presiona "Cancelar" en el diálogo	<ul style="list-style-type: none"> ▪ Se cierra el diálogo. ▪ Se regresa a la pantalla de datos del usuario. ▪ No se realiza ninguna acción de eliminación.
Post-condiciones	<ul style="list-style-type: none"> ▪ El usuario permanece en el sistema sin cambios. ▪ Todos los datos y resultados se mantienen intactos. ▪ El usuario puede continuar usando la aplicación normalmente.

Tabla 4.1-5 Caso de Prueba 5: Cancelar Eliminación de Usuario

Caso de Prueba 6: Visualizar Lista de Usuarios

Caso de prueba, ID	TC-USER-006
Descripción	Visualizar la lista de usuarios disponibles en el dropdown selector.
Pre-condiciones	<ul style="list-style-type: none"> ▪ Existen al menos 3 usuarios registrados con diferentes nombres. ▪ Se está en la pantalla Home.
Secuencia de eventos	
Eventos de entrada	Eventos de salida
El usuario toca el dropdown de selección de la pantalla de Home.	Se despliega la lista completa de usuarios con su nickname , ordenados alfabéticamente.
El usuario selecciona un usuario diferente de la lista.	<ul style="list-style-type: none"> ▪ Se cierra el dropdown. ▪ En el selector de la página de Home aparece el nickname del usuario seleccionado.
Post-condiciones	<ul style="list-style-type: none"> ▪ La lista muestra todos los usuarios registrados correctamente. ▪ El usuario seleccionado queda como usuario actual.

Tabla 4.1-6 Caso de Prueba 6: Visualizar Lista de Usuarios

Caso de Prueba 7: Visualizar Usuario Vacío

Caso de prueba, ID	TC-USER-007
Descripción	Mostrar el estado cuando no existen usuarios registrados.
Pre-condiciones	<ul style="list-style-type: none"> ▪ No existen usuarios registrados en la base de datos. ▪ Se está en la pantalla Home.
Secuencia de eventos	
Eventos de entrada	Eventos de salida
La aplicación se abre y navega a Home.	<ul style="list-style-type: none"> ▪ El selector de usuario de la pantalla de Home cambia a un cuadro donde pone 'Crear perfil'. ▪ El componente (código de referencia ,botón de test , historia tests) se deshabilita. ▪ El botón "Crear Nuevo Perfil" permanece activo.
El usuario intenta interactuar con los componentes deshabilitados.	Los componentes deshabilitados no reaccionan.
El usuario pulsa el cuadro superior donde pone "Crear perfil."	Empieza el flujo para crear un usuario.
Post-condiciones	<ul style="list-style-type: none"> ▪ Solo la opción de crear nuevo perfil está disponible

Tabla 4.1-7 Caso de Prueba 7: Visualizar Usuario Vacío

4.1.2 Prueba de Validación de Código de Referencia

Este apartado detalla el conjunto de pruebas de sistema realizadas para verificar el correcto funcionamiento del módulo de validación de códigos de referencia. El objetivo de estas pruebas es asegurar que cada código introducido sea validado correctamente antes de iniciar una sesión del test TMT, garantizando así que el acceso a la prueba cumple con los requisitos establecidos.

Caso de Prueba 1: Validar Código de Referencia Válido (Sin Usos Previos)	
Caso de prueba, ID	TC-REF-001
Descripción	Validar un código de referencia válido que no ha sido utilizado previamente.
Pre-condiciones	<ul style="list-style-type: none"> ▪ Existe al menos un perfil de usuario seleccionado. ▪ Se está en la pantalla Home. ▪ El código de referencia no ha sido utilizado anteriormente.
Secuencia de eventos	
Eventos de entrada	Eventos de salida
El usuario introduce un código de referencia válido en el campo correspondiente.	El campo acepta el texto sin mostrar errores.
El usuario presiona el botón de validación o presiona Enter.	<ul style="list-style-type: none"> ▪ Se muestra un indicador de carga. ▪ Se realiza la petición al servidor. ▪ El servidor responde: exists: 0, hands: [], message: "OK", status: "ok" ▪ Se muestra mensaje de código válido. ▪ Se habilita el botón "Empezar Test TMT".

El usuario presiona "Empezar Test TMT".	<ul style="list-style-type: none"> ▪ Se muestra un diálogo para seleccionar la mano a utilizar. ▪ El diálogo ofrece opciones: "Mano Izquierda" y "Mano Derecha". ▪ El usuario puede elegir libremente cualquier mano.
Post-condiciones	<ul style="list-style-type: none"> ▪ El código de referencia queda validado. ▪ El botón de iniciar test queda habilitado. ▪ El usuario puede seleccionar cualquier mano para iniciar el test.

Tabla 4.1-8 Caso de Prueba 1: Validar Código de Referencia Válido (Sin Usos Previos)

Caso de Prueba 2: Validar Código de Referencia Inválido (Incorrecto)	
Caso de prueba, ID	TC-REF-002
Descripción	Intentar validar un código de referencia que no existe o es incorrecto.
Pre-condiciones	<ul style="list-style-type: none"> • Existe al menos un perfil de usuario seleccionado. • Se está en la pantalla Home. • El código introducido no existe en el sistema.
Secuencia de eventos	
Eventos de entrada	Eventos de salida
El usuario introduce un código de referencia inválido en el campo correspondiente.	El campo acepta el texto inicialmente.

<p>El usuario presiona el botón de validación.</p>	<ul style="list-style-type: none"> ▪ Se muestra un indicador de carga. ▪ Se realiza la petición al servidor. ▪ El servidor responde message:NoK, status:ok. ▪ Se muestra un mensaje de error debajo del componente de código de referencia : "La referencia no es correcta. Por favor verifique la contacte al neurólogo". ▪ El botón "Empezar Test TMT" permanece deshabilitado. ▪ Se muestra una snackbar roja con el mismo mensaje de error.
<p>Post-condiciones</p>	<ul style="list-style-type: none"> ▪ El código de referencia no queda validado. ▪ El botón "Empezar Test TMT" permanece deshabilitado. ▪ El usuario puede corregir el código e intentar nuevamente.

Tabla 4.1-9 Caso de Prueba 2: Validar Código de Referencia Inválido (Incorrecto)

Caso de Prueba 3: Validar Código de Referencia Inválido (Ya Utilizado Dos Veces)	
<p>Caso de prueba, ID</p>	<p>TC-REF-003</p>
<p>Descripción</p>	<p>Intentar validar un código de referencia que ya ha sido utilizado el máximo número de veces permitidas (2 veces).</p>
<p>Pre-condiciones</p>	<ul style="list-style-type: none"> ▪ Existe al menos un perfil de usuario seleccionado. ▪ Se está en la pantalla Home. • El código de referencia ya ha sido utilizado 2 veces (máximo permitido).
<p style="text-align: center;">Secuencia de eventos</p>	
<p>Eventos de entrada</p>	<p>Eventos de salida</p>

El usuario introduce un código de referencia que ya fue utilizado 2 veces.	El campo acepta el texto inicialmente.
El usuario presiona el botón de validación.	<ul style="list-style-type: none"> ▪ Se realiza la petición al servidor. ▪ El servidor responde: exists: 2, hands: ["I", "D"], message: "OK" ▪ El sistema detecta que numberExists >= 2. ▪ Se muestra mensaje de error "El código de referencia ya ha sido utilizado". ▪ El botón "Empezar Test TMT" permanece deshabilitado. ▪ Se muestra una snackbar roja con el mismo mensaje de error.
Post-condiciones	<ul style="list-style-type: none"> ▪ El código de referencia no queda validado. ▪ El botón "Empezar Test TMT" permanece deshabilitado. ▪ El usuario debe introducir un código diferente.

Tabla 4.1-10 Caso de Prueba 3: Validar Código de Referencia Inválido (Ya Utilizado Dos Veces)

Caso de Prueba 4: Validar Código de Referencia Válido (Solo Mano Izquierda Utilizada - Obligar Mano Derecha)	
Caso de prueba, ID	TC-REF-004
Descripción	Validar un código de referencia que solo ha sido utilizado con la mano izquierda, obligando al usuario a usar la mano derecha.
Pre-condiciones	<ul style="list-style-type: none"> ▪ Existe al menos un perfil de usuario seleccionado. ▪ Se está en la pantalla Home. ▪ El código de referencia ha sido utilizado solo con mano izquierda.
Secuencia de eventos	

Eventos de entrada	Eventos de salida
El usuario introduce un código de referencia utilizado solo con mano izquierda.	El campo acepta el texto sin mostrar errores.
El usuario presiona el botón de validación.	<ul style="list-style-type: none"> ▪ Se realiza la petición al servidor. ▪ El servidor responde: exists: 1, hands: ["I"], message: "OK" ▪ Se muestra mensaje de código válido. ▪ Se habilita el botón "Empezar Test TMT".
El usuario presiona "Empezar Test TMT".	<ul style="list-style-type: none"> ▪ Se muestra diálogo con título " Iniciar test con mano derecha" ▪ Se muestra mensaje explicativo sobre por qué debe usar la mano derecha. ▪ Se muestra solo un botón: "Mano Derecha 🖐️".
El usuario confirma usar mano derecha.	<ul style="list-style-type: none"> ▪ Se cierra el diálogo. ▪ Se establece mano derecha como obligatoria. ▪ Se inicia el flujo del test TMT. ▪ Se navega a la pantalla de ayuda del test.
Post-condiciones	<ul style="list-style-type: none"> • El código queda validado con mano derecha obligatoria. • Se inicia el flujo del test TMT. • Se registra que se utilizará la mano derecha.

Tabla 4.1-11 Caso de Prueba 4: Validar Código de Referencia Válido (Solo Mano Izquierda Utilizada - Obligar Mano Derecha)

Caso de Prueba 6: Validar Código de Referencia Válido (Solo Mano Derecha Utilizada - Obligar Mano Izquierda)

Caso de prueba, ID	TC-REF-005
Descripción	Validar un código de referencia que solo ha sido utilizado con la mano derecha, obligando al usuario a usar la mano izquierda.
Pre-condiciones	<ul style="list-style-type: none"> ▪ Existe al menos un perfil de usuario seleccionado. ▪ Se está en la pantalla Home. ▪ El código de referencia ha sido utilizado solo con mano derecha.
Secuencia de eventos	
Eventos de entrada	Eventos de salida
El usuario introduce un código de referencia utilizado solo con mano derecha.	El campo acepta el texto sin mostrar errores.
El usuario presiona el botón de validación.	<ul style="list-style-type: none"> ▪ Se realiza la petición al servidor. ▪ El servidor responde: exists: 1, hands: ["D"], message: "OK" ▪ Se muestra mensaje de código válido. ▪ Se habilita el botón "Empezar Test TMT".
El usuario presiona "Empezar Test TMT".	<ul style="list-style-type: none"> ▪ Se muestra diálogo con título "Iniciar test con mano izquierda" ▪ Se muestra mensaje explicativo sobre por qué debe usar la mano izquierda. ▪ Se muestra solo un botón: "Mano izquierda".
El usuario confirma usar mano izquierda.	<ul style="list-style-type: none"> ▪ Se cierra el diálogo. ▪ Se establece mano izquierda como obligatoria. ▪ Se inicia el flujo del test TMT.

	<ul style="list-style-type: none"> ▪ Se navega a la pantalla de ayuda del test.
Post-condiciones	<ul style="list-style-type: none"> • El código queda validado con mano izquierda obligatoria. • Se inicia el flujo del test TMT. • Se registra que se utilizará la mano izquierda.

Tabla 4.1-12 Caso de Prueba 6: Validar Código de Referencia Válido (Solo Mano Derecha Utilizada - Obligar Mano Izquierda)

4.1.3 Prueba de Flujo de Envío de Resultados

Este apartado detalla las pruebas de sistema realizadas sobre el flujo de envío de resultados de métricas de test TMT al servidor. Con estas pruebas se garantiza que los resultados se envían como pide el requisito y se asegura la fiabilidad en caso de no haber conexión a internet.

Las pruebas se centran en verificar la robustez del sistema para:

- Enviar los datos de forma exitosa cuando el dispositivo está en línea.
- Almacenar los resultados localmente de forma segura si no hay conexión a internet o si el servidor devuelve un error.
- Reintentar el envío de los resultados pendientes de forma automática y transparente para el usuario una vez que se recupera la conexión.

Caso de Prueba 1: Envío Exitoso de Resultados de Métricas	
Caso de prueba, ID	TC-METRICS-001
Descripción	Enviar exitosamente las métricas del test TMT al servidor y guardar el resultado en la base de datos local.
Pre-condiciones	<ul style="list-style-type: none"> ▪ El usuario ha completado tanto TMT-A como TMT-B. ▪ Existe conexión a internet estable. ▪ El servidor está disponible y funcionando.
Secuencia de eventos	

Eventos de entrada	Eventos de salida
Una vez terminado TMT-A y TMT-B, la aplicación muestra la pantalla de resultados TMT.	Se muestra un indicador de carga.
El sistema recopila automáticamente todas las métricas calculadas (tiempos, errores, métricas de presión, etc.).	Se crea el objeto TmtResultModel con todos los datos.
Se obtiene automáticamente el perfil de usuario actual.	Se crea el objeto TmtUserModel con los datos del usuario.
El sistema envía los datos al servidor vía API REST.	<ul style="list-style-type: none"> ▪ El servidor responde con status 200 OK. ▪ Se confirma que los datos fueron procesados exitosamente.
Se guarda automáticamente el resultado en la base de datos local.	<ul style="list-style-type: none"> ▪ Se crea el registro UserTestLocalDataResult. ▪ Se guarda con: código de referencia, fecha, tiempos TMT-A y TMT-B, mano utilizada.
Se oculta el indicador de carga.	<ul style="list-style-type: none"> ▪ La pantalla muestra los resultados finales. ▪ El botón "Terminar" está disponible.
El usuario pulsa el botón "Terminar".	Cerrar la pantalla de resultado y navegar a la pantalla de Home.
Navegar a "Ver mis Tests".	Aparece el nuevo registro con código de referencia, fecha, tiempos TMT-A y TMT-B y mano utilizado del último test TMT realizado en la tabla.
Post-condiciones	<ul style="list-style-type: none"> ▪ Los resultados están guardados en el servidor.

	<ul style="list-style-type: none"> ▪ Los resultados están guardados en la base de datos local.
--	---

Tabla 4.1-13 Caso de Prueba 1: Envío Exitoso de Resultados de Métricas

Caso de Prueba 2: Envío con Fallo de Conexión y Reintento Exitoso	
Caso de prueba, ID	TC-METRICS-002
Descripción	Manejar el fallo de envío por falta de conexión, guardar como pendiente y reenviar automáticamente cuando se restaure la conexión.
Pre-condiciones	<ul style="list-style-type: none"> ▪ El usuario ha completado tanto TMT-A como TMT-B. ▪ No existe conexión a internet al momento de enviar.
Secuencia de eventos	
Eventos de entrada	Eventos de salida
Una vez terminado TMT-A y TMT-B, Quita el internet del móvil. La aplicación intenta enviar los resultados al servidor.	Se detecta NetworkError por falta de conexión.
El sistema guarda automáticamente los resultados como pendientes.	<ul style="list-style-type: none"> ▪ Se crea PendingResultData en LocalStorage. ▪ Se programa WorkManager para reintentos automáticos.
Se muestra la pantalla de resultados normalmente.	<ul style="list-style-type: none"> ▪ El usuario ve sus resultados. ▪ Aparece una snackbar con el mensaje de error.

El usuario pulsa el botón "Terminar".	Cerrar la pantalla de resultado y navegar a la pantalla de Home.
Se restaura la conexión a internet.	<ul style="list-style-type: none"> • El WorkManager detecta la conectividad. • Se ejecuta automáticamente <code>sendPendingResults()</code>.
El sistema reintentará el envío automáticamente.	<ul style="list-style-type: none"> • El servidor responde exitosamente. • Se elimina el resultado pendiente. • Se guarda en la base de datos local.
Navegar a "Ver mis Tests".	Aparece el nuevo registro con código de referencia, fecha, tiempos TMT-A y TMT-B y mano utilizado del último test TMT realizado en la tabla.
Post-condiciones	<ul style="list-style-type: none"> ▪ Los resultados están guardados en el servidor. ▪ Los resultados están guardados en la base de datos local. ▪ No quedan resultados pendientes en LocalStorage.

Tabla 4.1-14 Caso de Prueba 2: Envío con Fallo de Conexión y Reintento Exitoso

4.1.4 Prueba de Flujo Navegación de Test Tmt

Este apartado detalla las pruebas realizadas para el flujo del Test TMT, desde que se pulsa el botón 'Empezar Test TMT' hasta que aparece la pantalla de resultados.

El objetivo es asegurar que la secuencia de pantallas, los diálogos y las instrucciones aparecen adecuadamente y son entendibles.

Caso de Prueba 1: Flujo Completo de Navegación del Test TMT

Caso de prueba, ID	TC-NAVIGATION-001
---------------------------	-------------------

Descripción	Verificar el flujo completo de navegación desde el inicio del test hasta la pantalla de resultados, incluyendo prácticas, tests y diálogos intermedios.
Pre-condiciones	<ul style="list-style-type: none"> ▪ Existe al menos un usuario registrado y seleccionado. ▪ Se ha validado correctamente un código de referencia. ▪ El usuario está en la pantalla Home. ▪ El botón "Empezar Test TMT" está habilitado.
Secuencia de eventos	
Eventos de entrada	Eventos de salida
El usuario presiona el botón "Empezar Test TMT" en la pantalla Home.	Se navega a la pantalla explicativa general del test TMT con instrucciones generales y botón "Comenzar".
El usuario presiona el botón "Comenzar" en la pantalla explicativa.	Aparece un diálogo para seleccionar la mano a utilizar: "Mano Izquierda" o "Mano Derecha".
El usuario selecciona una mano (izquierda o derecha) en el diálogo.	<ul style="list-style-type: none"> ▪ Se cierra el diálogo. ▪ Se navega a la pantalla de instrucciones TMT parte A.
El usuario presiona "Comenzar práctica parte A" en la pantalla de instrucciones.	Se navega a la pantalla de práctica TMT-A con círculos numerados del 1 al 9.
El usuario completa la práctica TMT-A conectando todos los círculos correctamente.	<ul style="list-style-type: none"> ▪ Aparece un diálogo de confirmación: "Práctica A completada". ▪ El diálogo contiene el botón "Iniciar Test A".

<p>El usuario presiona "Iniciar Test A" en el diálogo.</p>	<ul style="list-style-type: none"> ▪ Se cierra el diálogo. ▪ Aparece la pantalla de cuenta atrás mostrando "TMT A" y contando 3, 2, 1. ▪ El cronómetro se inicia automáticamente.
<p>Termina la cuenta atrás de 3 segundos.</p>	<p>Se muestra la pantalla de test TMT-A con 25 o 15 (Depende de la configuración) círculos numerados y cronómetro activo en la app bar.</p>
<p>El usuario completa el test TMT-A conectando todos los círculos del 1 al 25 o 15.</p>	<ul style="list-style-type: none"> ▪ Se pausa automáticamente el cronómetro. ▪ Aparece un diálogo: "Test A completado". ▪ El diálogo contiene el botón "Sí, comenzar TMT parte B".
<p>El usuario presiona "Sí, comenzar TMT parte B" en el diálogo.</p>	<ul style="list-style-type: none"> • Se cierra el diálogo. • Se navega a la pantalla de instrucciones TMT parte B. • Se muestra la animación de TMT-B con la mano previamente seleccionada.
<p>El usuario presiona "Comenzar práctica parte B" en la pantalla de instrucciones.</p>	<p>Se navega a la pantalla de práctica TMT-B con círculos alternando números y letras (1-A-2-B-3-C-4-D-5).</p>
<p>El usuario completa la práctica TMT-B conectando correctamente la secuencia alternada.</p>	<ul style="list-style-type: none"> ▪ Aparece un diálogo de confirmación: "Práctica B completada". ▪ El diálogo contiene el botón "Iniciar Test B".
<p>El usuario presiona "Iniciar Test B" en el diálogo.</p>	<ul style="list-style-type: none"> ▪ Se cierra el diálogo. ▪ Aparece la pantalla de cuenta atrás mostrando "TMT B" y contando 3, 2, 1. ▪ El cronómetro continúa desde el tiempo de TMT-A.

Termina la cuenta atrás de 3 segundos.	Se muestra la pantalla de test TMT-B con 25 o 15 (Depende de la configuración) círculos alternando números y letras, cronómetro activo continuando desde TMT-A.
El usuario completa el test TMT-B conectando todos los círculos en secuencia alternada.	<ul style="list-style-type: none"> ▪ Se detiene automáticamente el cronómetro. ▪ Se navega directamente a la pantalla de resultados. ▪ Se muestran los tiempos de TMT-A, TMT-B y errores cometidos.
Post-condiciones	<ul style="list-style-type: none"> ▪ El usuario ha completado todo el flujo de test TMT. ▪ Se han registrado todos los tiempos y métricas. ▪ Los resultados se muestran correctamente en la pantalla final. ▪ El usuario puede finalizar y regresar al Home.

Tabla 4.1-15 Caso de Prueba 1: Flujo Completo de Navegación del Test TMT

Caso de Prueba 13: Funcionalidad de Ayuda Durante Test	
Caso de prueba, ID	TC-NAVIGATION-002
Descripción	Verificar que el usuario puede acceder a la ayuda durante TMT-A o TMT-B y regresar correctamente al test con cuenta atrás.
Pre-condiciones	<ul style="list-style-type: none"> ▪ El usuario está ejecutando TMT-A o TMT-B (pantalla de test activa). ▪ El cronómetro está funcionando. ▪ El usuario puede ver el icono de ayuda en la app bar.

Secuencia de eventos	
Eventos de entrada	Eventos de salida
El usuario presiona el icono de ayuda (?) en la app bar durante TMT-A.	<ul style="list-style-type: none"> ▪ Se pausa automáticamente el cronómetro. ▪ Se navega a la pantalla de instrucciones TMT-A. ▪ La pantalla muestra el botón "Volver".
El usuario lee las instrucciones y presiona "Volver".	<ul style="list-style-type: none"> ▪ Se navega a la pantalla de cuenta atrás mostrando "TMT A". ▪ Se cuenta 3, 2, 1 segundos. ▪ El cronómetro permanece pausado durante la cuenta atrás.
Termina la cuenta atrás de 3 segundos.	<ul style="list-style-type: none"> ▪ Se muestra la pantalla de test TMT-A desde un estado nuevo, es decir, empezar desde cero. ▪ El cronómetro se reanuda automáticamente.
El usuario presiona el icono de ayuda (?) en la app bar durante TMT-B.	<ul style="list-style-type: none"> ▪ Se pausa automáticamente el cronómetro. ▪ Se navega a la pantalla de instrucciones TMT-B. ▪ La pantalla muestra el botón "Volver".
El usuario lee las instrucciones y presiona "Volver".	<ul style="list-style-type: none"> ▪ Se navega a la pantalla de cuenta atrás mostrando "TMT B". ▪ Se cuenta 3, 2, 1 segundos. ▪ El cronómetro permanece pausado durante la cuenta atrás.

Termina la cuenta atrás de 3 segundos.	<ul style="list-style-type: none"> • Se muestra la pantalla de test TMT-B desde un estado nuevo, es decir, empezar desde cero. • El cronómetro se reanuda automáticamente desde el tiempo registrado en el TMT Parte A.
El usuario completa el test normalmente después de usar la ayuda.	Los resultados finales incluyen el tiempo total correcto (también incluye el tiempo que ha utilizado para consultar la ayuda).
Postcondiciones	<ul style="list-style-type: none"> ▪ El cronómetro maneja correctamente las pausas y reanudaciones. ▪ El estado del test se reinicia después de la consulta de ayuda. ▪ El usuario puede usar la ayuda múltiples veces sin afectar la funcionalidad.

Tabla 4.1-16 Caso de Prueba 13: Funcionalidad de Ayuda Durante Test

4.2 Prueba Piloto

Para evaluar la aplicación msDtmt en un entorno práctico y obtener una primera impresión sobre su funcionamiento y la experiencia de usuario, se llevó a cabo una prueba piloto. Esta prueba se realizó aprovechando una actividad académica coordinada a nivel universitario.

4.2.1 Contexto del Estudio

Para la evaluación de la aplicación, se contó con la participación de 29 personas voluntarias y sin problemas cognitivos conocidos que accedieron a participar en la prueba piloto. El objetivo de esta prueba era valorar el funcionamiento de la aplicación y obtener una primera retroalimentación sobre su usabilidad y la experiencia de usuario.

Adicionalmente, 26 participantes respondieron la encuesta de evaluación de usabilidad, representando una tasa de respuesta del 89.66%.

Todas las pruebas se realizaron mediante la misma Tablet del modelo Huawei Mate 10, donde tenía instalada la aplicación msDtmt de Android.

La prueba consta de los siguientes pasos:

1. Se ofreció una breve introducción a los participantes sobre el propósito general de la aplicación msDtmt y el Trail Making Test.

2. Se solicitó a cada participante que utilizara la aplicación msDtmT para completar las dos partes del Trail Making Test (Parte A y Parte B).
3. Tras finalizar las pruebas en la aplicación, se les solicitó responder un cuestionario cuyo objetivo era recoger sus impresiones respecto a la usabilidad y satisfacción con la aplicación.

Consideraciones Éticas y de Privacidad:

- La participación fue completamente anónima y voluntaria.
- No se recopiló información personal identificable de los participantes.
- Los datos obtenidos se utilizaron exclusivamente para fines de validación técnica de la aplicación.
- Se informó a los participantes sobre el propósito académico del estudio.

4.2.2 Análisis de Rendimiento en el Test TMT

A continuación, se presentan los resultados obtenidos directamente de la interacción de los participantes con la aplicación msdtmt:

Personas	TMT A Duración	TMT A Errores	TMT B Duración	TMT B Errores
1	13	0	27	1
2	15	0	33	3
3	18	1	22	0
4	16	0	22	0
5	18	1	27	0
6	17	1	34	1
7	18	0	34	0
8	12	0	20	1
9	12	0	24	2
10	16	1	24	1
11	15	0	33	0
12	26	0	32	1
13	22	1	27	0
14	19	0	25	0
15	25	0	43	1
16	17	0	33	1
17	15	0	30	0
18	14	1	20	0
19	34	0	45	1
20	15	0	29	0
21	16	1	24	1
22	18	1	23	0
23	19	0	29	1
24	21	1	29	2
25	25	1	48	2

26	25	0	43	1
27	15	0	26	2
28	24	1	35	3
29	13	0	24	1

Tabla 4.2-1 Muestras de resultados del Test TMT en la prueba piloto

Los datos de rendimiento obtenidos de los 29 participantes muestran los siguientes resultados estadísticos:

TMT Parte A

Estadísticas de Duración del TMT (Parte A)

Métrica	Valor(Segundos)
Media Duración	18,38
Mediana Duración	17
Máximo Duración	34
Mínimo Duración	12
Moda Duración	15
Desviación Estándar	4,99
Rango Duración	12 - 34

Tabla 4.2-2 Estadísticas duración de la Parte A del Test TMT en la prueba piloto

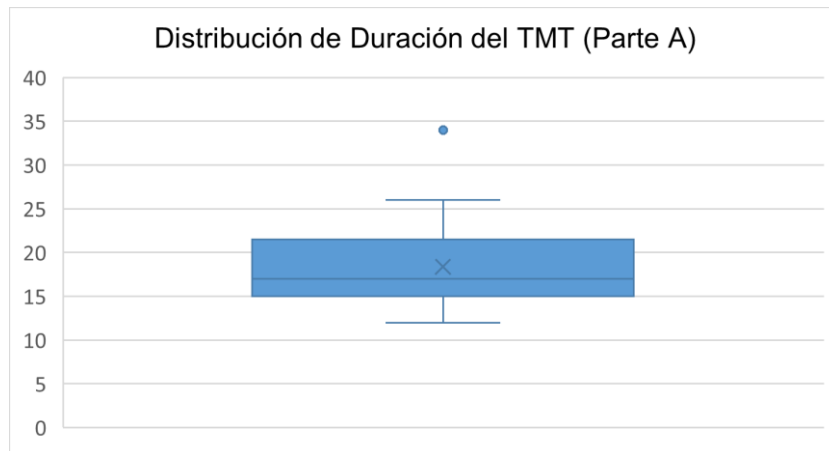


Ilustración 4.2-1 Boxplot de Distribución de Duración del TMT Parte A en la prueba piloto

Estadísticas de Errores del TMT (Parte A)

Métrica	Valor
Media Error	0,38
Mediana Error	0
Máximo Error	1
Mínimo Error	0
Moda Error	0
Desviación Error	0,49
Rango Error	0-1 error

Ilustración 4.2-2 Tabla 4.2 2 Estadísticas Errores de la Parte A del Test TMT en la prueba piloto

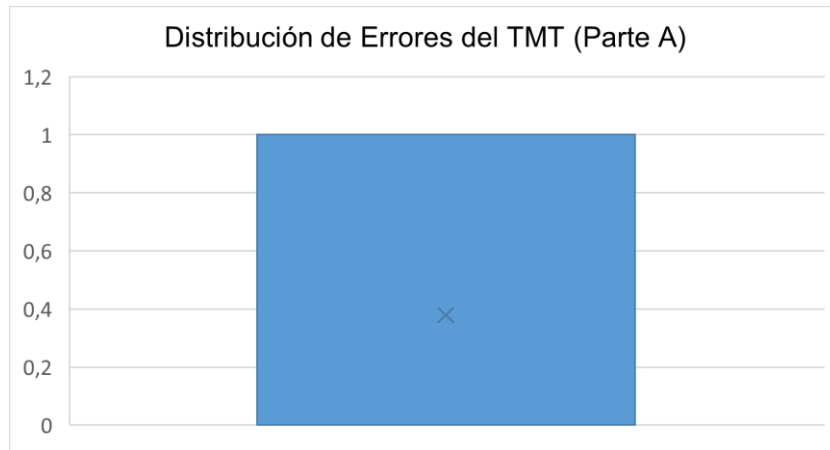


Ilustración 4.2-3 Boxplot de Distribución de Errores del TMT Parte A en la prueba piloto

TMT Parte B

Estadísticas de Duración del TMT (Parte B)

Métrica	Valor(segundos)
Media Duración B	29,83
Mediana Duración B	29
Máximo Duración B	48
Mínimo Duración B	20
Moda Duración B	24
Desviación Estándar	7,35
Rango Duración B	20 - 48

Tabla 4.2-3 Estadísticas duración de la Parte B del Test TMT en la prueba piloto

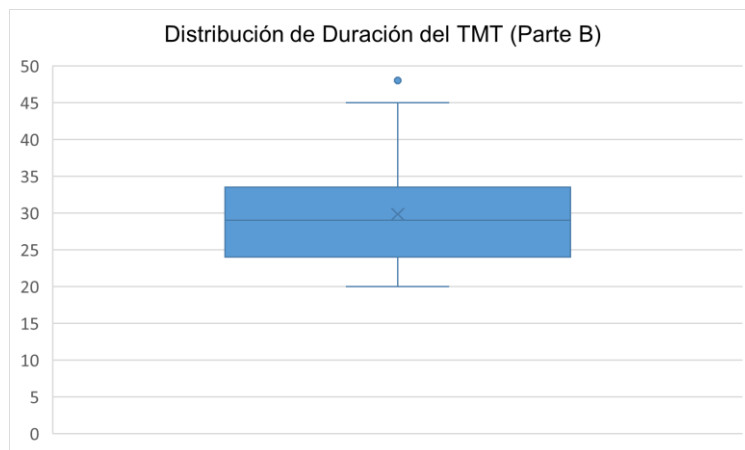


Ilustración 4.2-4 Boxplot de Distribución de Duración del TMT Parte B en la prueba piloto

Estadísticas de Errores del TMT (Parte B)

Métrica	Valor
Media Error	0,9
Mediana Error	1
Máximo Error	3
Mínimo Error	0
Moda Error	1
Desviación Error	0,88
Rango Error	0-3

Ilustración 4.2-5 Tabla 4.2 3 Estadísticas errores de la Parte B del Test TMT en la prueba piloto

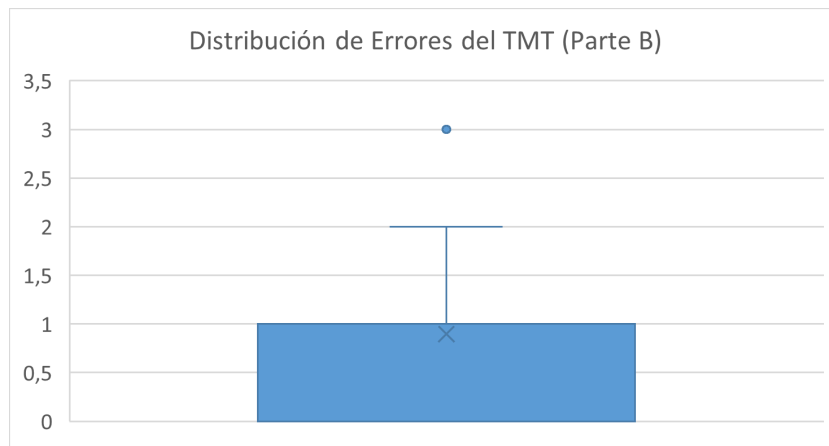


Ilustración 4.2-6 Boxplot de Distribución de Errores del TMT Parte B en la prueba piloto

Los resultados obtenidos son coherentes con los valores esperados para la población joven y demuestran un rendimiento superior al de la población adulta de referencia:

TMT-A (Procesamiento Secuencial): Según la referencia anterior, el tiempo promedio para TMT-A es de 29 segundos para población adulta, mientras que nuestros participantes (15-16 años) obtuvieron una media de 18.38 segundos. Esta diferencia de 10.62 segundos (36.6% más rápido) refleja las mejores capacidades cognitivas y de procesamiento visual-motor características de esta edad.

TMT-B (Alternancia y Flexibilidad Cognitiva): Para TMT-B establece un tiempo promedio de 75 segundos en población adulta, comparado con los 29.83 segundos obtenidos por nuestros participantes jóvenes. Esta notable diferencia de 45.17 segundos (60.2% más rápido) evidencia la superior flexibilidad cognitiva y capacidad de alternancia atencional típica de los adolescentes.

Observaciones Relevantes:

- **TMT-A:** El 95% de los participantes completaron la prueba sin errores, indicando una alta comprensión de las instrucciones.
- **TMT-B:** El incremento en la complejidad se refleja en un aumento del 62% en el tiempo de ejecución (18.38s vs 29.83s).

- **Patrón de Errores:** El aumento de errores en TMT-B (0.38 vs 0.90) es consistente con la mayor demanda cognitiva de alternancia entre números y letras.

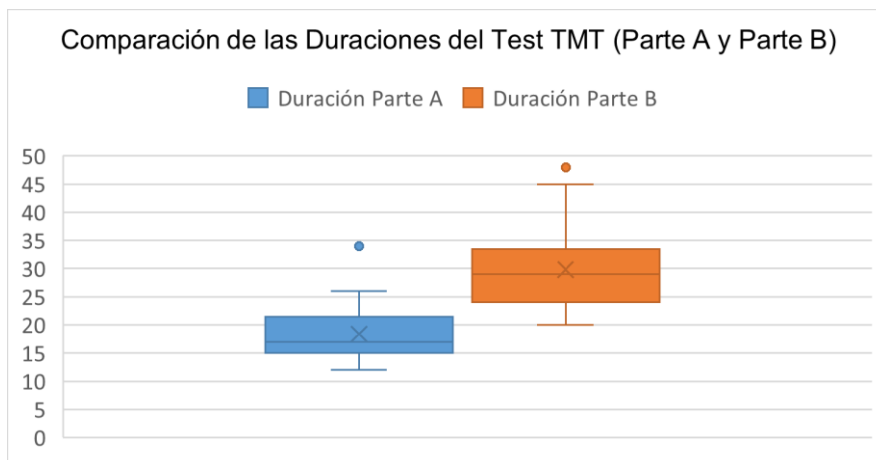


Ilustración 4.2-7 Boxplot Comparación de las duraciones Test TMT en la prueba piloto

4.2.3 Evaluación de Usabilidad y Experiencia de Usuario

Tras completar las pruebas con la aplicación msDmt, los 26 participantes valoraron su experiencia mediante un cuestionario anónimo diseñado para evaluar la usabilidad. Este se componía de cuatro preguntas específicas que combinaban escalas de Likert (1-5) y preguntas abiertas:

Pregunta 1: ¿Son entendibles las señalizaciones del test TMT?

- **Pregunta específica:**
 - Se puede entender, cuando levanta la mano, que la señalización (Es decir, el círculo cambiar a color amarillo) indica dónde está el último círculo.
 - Se puede entender cuando está mal conectado
 - ¿Se puede entender que el orden en el que está conectado es correcto?
- **Escala utilizada:** 1 hasta 5 ; 1 = Muy deficiente, 5 = Excelente

Pregunta 2: ¿El flujo de navegación entre pantallas fue intuitivo?

- **Pregunta específica:** Es entendible e intuitivo navegar entre las diferentes pantallas de la aplicación. Es decir, cuando hay elementos de navegación, como un botón, una barra de para volver o un diálogo, las acciones que se pueden realizar con ellos son comprensibles.
- **Escala utilizada:** 1 hasta 5 ; 1 = Muy deficiente, 5 = Excelente

Pregunta 3: ¿Qué características o cambios añadirías a la aplicación para mejorar tu experiencia al realizar el test TMT?

- **Tipo de pregunta:** respuesta abierta.

Pregunta 4: Describe qué opinas sobre los elementos visuales de la aplicación (colores, tamaño de los círculos, contraste, tamaño del texto, etc.). ¿Hay algún aspecto visual que mejorarías o que te resultó particularmente útil?

- **Tipo de pregunta:** respuesta abierta.

El resultado obtenido puede verse en la siguiente tabla:

Personas	Pregunta 1	Pregunta 2	Pregunta 3	Pregunta 4
1	4	4	Está muy bien	Nada , está muy claro
2	5	5	-	-
3	5	5	Mas dificultad a lo mejor en algunas partes	están bien diseñados
4	5	5	nada	esta todo bien
5	5	5	nada, está muy bien.	todo muy bien
6	5	5	-	Poner el fondo de pantalla en el modo oscuro de gris oscuro y no de negro
7	5	5	Que haya que repetirlo varias veces para que sea más preciso.	No mejoraría nada visual.
8	5	5		
9	5	5	Yo pondría un poco más de decoración a la pantalla inicial	yo lo veo todo bastante bien
10	5	4	está muy bien	no le cambiaría nada
11	4	4	Animación	Todo muy bien
12	5	4	Animaciones, y más botones en el home	Lo de los colores en los números está bien
13	5	5	Animaciones	Los colores de los números
14	5	5	-	Está muy bien, es simple y fácil de entender
15	5	4	-	-
16	5	5	-	-
17	5	5	-	-
18	4	4	-	-
19	4	5	-	es muy visual
20	5	5	creo que falta la H, por lo demás todo perfecto	me parece muy intuitivo y útil, no lo cambiaria
21	5	4	Yo haría que la app fuera un poco más desarrollada en cuanto a características o cosas que se puedan hacer y mejorar el diseño	Tiene un diseño minimalista y agradable a la vista, pero yo mejorar el aspecto visual de la App

22	5	5	Lo que cambiaría es que se puedan ver más datos para tener más información	Están muy bien y no cambiaría nada
23	5	5	Una opción para poder ver la media de resultados adquirida	Los círculos están correctamente, pero mejoraría la visibilidad del menú. Al tener dos opciones, podrían hacerlas de distintos colores o agrandar el tamaño.
24	5	5	-	-
25	5	5	Ninguna, está muy bien	A lo mejor podrían poner los colores del arcoíris en orden
26	5	5	-	son correctos

Tabla 4.2-4 Muestras de cuestionarios después del Test TMT en la prueba piloto

Resumen de Resultados

Los resultados de la evaluación de usabilidad demuestran una **alta satisfacción general** de los usuarios con la aplicación msDtm. Los datos cuantitativos revelan puntuaciones consistentemente elevadas en ambas dimensiones evaluadas:

Entendibilidad de Señalizaciones (Pregunta 1):

- **Media:** 4.85/5 (excelente)
- **Desviación estándar:** 0,36
- **Distribución:** 84.6% de respuestas excelentes (5/5) y 15.4% muy buenas (4/5)
- **Interpretación:** Los usuarios comprenden claramente el sistema de retroalimentación visual, incluyendo las señalizaciones de círculo activo (amarillo), errores y progreso correcto.

Intuitividad del Flujo de Navegación (Pregunta 2):

- **Media:** 4.73/5 (excelente)
- **Desviación estándar:** 0,44
- **Distribución:** 73.1% de respuestas excelentes (5/5) y 26.9% muy buenas (4/5)
- **Interpretación:** La navegación entre pantallas, botones y diálogos resulta comprensible y natural para los usuarios.

Aspectos Más Valorados:

- Claridad del diseño: Múltiples comentarios destacan que "está muy claro", "es simple y fácil de entender".

- Elementos visuales: Los usuarios califican positivamente los colores, tamaño de círculos y contraste general.
- Funcionalidad: El sistema es percibido como "muy visual", "intuitivo y útil"

Sugerencias de Mejora Identificadas:

- Enriquecimiento de la experiencia visual:
 - Incorporación de animaciones para mejorar la interactividad.
 - Mejoras en el diseño de la pantalla principal.
 - Refinamientos en el tema oscuro (fondo gris oscuro vs negro).
- Funcionalidades adicionales:
 - Visualización de estadísticas y promedios de resultados.
 - Más información disponible para el usuario.
- Ajustes menores de interfaz:
 - Mejora en la visibilidad del menú principal.
 - Diferenciación visual entre opciones.
 - Optimización de algunos elementos de navegación.

En conclusión, la prueba piloto realizada con estudiantes de 4º de ESO indica que la aplicación msDtmt posee un alto grado de usabilidad y genera una experiencia de usuario positiva. Los participantes encontraron las señalizaciones del test claras y la navegación intuitiva, y el diseño visual fue bien recibido en general.

Aunque la mayoría se mostró satisfecha con la funcionalidad actual, las sugerencias recogidas ofrecen vías valiosas para futuras mejoras. Por tanto, msDtmt demuestra cumplir efectivamente con los objetivos de diseño establecidos, proporcionando una base sólida para su futura implementación en entornos clínicos reales.

4.3 Presupuesto

A continuación, se va a detallar la estimación de costes asociados al proyecto, teniendo en cuenta las horas dedicadas, la experiencia de los desarrolladores y el coste del material de trabajo.

4.3.1 Costes personales de las actividades

La información salarial de referencia se ha obtenido de Glassdoor^[41] España, según los puestos correspondientes.

Debido a que no se dedicó tiempo completo a cada tarea, el cálculo se basó en el salario promedio anual. Este salario anual se dividió entre 12 para obtener el promedio mensual.

Posteriormente, para determinar el coste por hora, se consideró una jornada de 8 horas diarias y 22 días laborables al mes (excluyendo fines de semana).

Por lo tanto, el coste por hora se calculó mediante la siguiente fórmula:

$$\text{Coste por hora} = \{\text{Salario Anual Promedio}\} \left\{ 12 \text{ meses} \times \frac{176 \text{ horas}}{\text{mes}} \right\}$$

En la siguiente tabla se detalla el coste personal de cada actividad:

Tarea	Horas	Euros / Hora	Coste
Preanálisis	30	15,00 €	450,00 €
Análisis de Requisitos	35	15,00 €	525,00 €
Mock up	20	12,00 €	240,00 €
Diseño	40	12,00 €	480,00 €
Formación	40	14,00 €	560,00 €
Desarrollo backend	100	13,00 €	1.300,00 €
Desarrollo Aplicación	250	14,00 €	3.500,00 €
Pruebas e Integración	30	12,00 €	360,00 €
Documentación Técnica	20	14,00 €	280,00 €
Total			7.695,00 €

Tabla 4.3-1 Coste de cada actividad

La siguiente tabla muestra el puesto de referencia para cada actividad:

Puesto	Actividades	Sueldo Anual
Ingeniero De Requisitos	Preanálisis	31.000,00 €
	Análisis de Requisitos	
Diseñador Ux/Ui	Mock up	25.000,00 €
	Diseño	
Desarrollador backend	Desarrollo backend	27.000,00 €
Desarrollador Flutter	Desarrollo Aplicación	30.000,00 €
	Formación	
	Documentación Técnica	

Ingeniero Qa	Pruebas e Integración	24.000,00 €
--------------	-----------------------	-------------

Tabla 4.3-2 Referencia del puesto para cada actividad

4.3.2 Costes de los recursos físicos

A la hora de calcular el presupuesto, también hay que considerar todos los gastos de hardware a utilizar para desarrollar el proyecto.

En la siguiente tabla se muestran los recursos físicos a utilizar y sus costes:

Recurso físico	Coste
Ordenador	900,00 €
Tablet	300,00 €
Teléfonos Móviles	500,00 €
Máquina Virtual Para Servidor	200,00 €
Total	1.900,00 €

Tabla 4.3-3 Coste de los recursos físicos

4.3.3 Presupuesto del proyecto

Después de obtener todos los costes anteriores, se ha elaborado la tabla de presupuesto, como se demuestra en la siguiente tabla:

Coste total de las actividades del proyecto (€)	9.595,00 €
Reserva para contingencias general (€)	960 €
Línea base de costes final (€)	10.554,50 €
Reserva de gestión (€)	1.055 €
Coste Total	22.164 €
Margen de beneficio (€)	5.541 €
Presupuesto (€)	27.705,56 €

Tabla 4.3-4 Presupuesto del proyecto

- **Coste total de las actividades del proyecto:** Es la suma de todos los costes personales dedicados a las actividades y los costes de recursos físicos.
- **Reservar para contingencias general:** Es una cantidad de dinero incluida en la línea base de costes del proyecto, destinada a cubrir los costes de los riesgos identificados y previsibles que podrían ocurrir. En este caso, se ha tomado un 10% respecto al coste total de las actividades del proyecto.

- **Línea base de costes final:** Es la suma del coste total de las actividades del proyecto y la reserva para contingencias general.
- **Reserva de gestión:** Es una cantidad de dinero adicional al presupuesto base (no forma parte de la línea base de costes) que se reserva para hacer frente a cambios o problemas imprevistos y no identificados inicialmente en el proyecto. Se ha tomado un 10% de la línea base de costes final.
- **Coste Total:** Es la suma del coste total de las actividades del proyecto, la reserva para contingencias general y la reserva de gestión.
- **Margen de beneficio:** Los márgenes de beneficio de un proyecto suelen estar entre el 25% y el 40%^[42]. Siendo este un proyecto muy simple y pequeño, en este caso se ha tomado un 25% de beneficio respecto al coste total.

4.4 Distribución y Publicación del Proyecto

Publicación del Código Fuente como Proyecto Open Source

Como se ha mencionado, y en consonancia con el objetivo de fomentar la transparencia científica y servir de referencia para quienes deseen desarrollar aplicaciones similares, el proyecto de la aplicación msDtmt se ha alojado en un repositorio público en la plataforma GitHub bajo la licencia GPL-3.0.^[43] cuyas libertades fundamentales se detallan a continuación:

- **Libertad de Uso:** El software puede ser utilizado para cualquier propósito.
- **Libertad de Estudio y Modificación:** Los usuarios tienen acceso al código fuente y pueden modificarlo para que haga lo que deseen.
- **Libertad de Distribución:** Se pueden redistribuir copias del software original.
- **Libertad de Distribuir Modificaciones:** Se pueden distribuir las versiones modificadas del software.

Distribución de la Aplicación msDtmt en Plataformas Oficiales

Con el objetivo de asegurar una amplia accesibilidad y facilitar la instalación de la aplicación msDtmt en los dispositivos móviles y tabletas de los usuarios finales, se ha procedido a su publicación en las principales tiendas de aplicaciones. Esta estrategia de distribución es coherente con la elección de Flutter como framework de desarrollo, el cual permite generar aplicaciones nativas y eficientes para los dos principales sistemas operativos móviles a partir de una única base de código.

La aplicación msDtmt se encuentra disponible para su descarga a través de los siguientes enlaces oficiales:

- **Google Play Store (para dispositivos Android):**
<https://play.google.com/store/apps/details?id=com.upm.msdtmt>

- **Apple App Store (para dispositivos iOS):**
<https://apps.apple.com/us/app/msdtmt/id6744974705>

La publicación en estas plataformas oficiales no solo simplifica el proceso de instalación para los usuarios, sino que también proporciona un canal estructurado para la distribución de futuras actualizaciones y el mantenimiento de la aplicación. Este paso es fundamental para la transición del proyecto desde la fase de desarrollo hacia su uso práctico en el contexto para el cual fue diseñado.

5 Conclusiones y trabajos futuros

5.1 Impacto social y medioambiental

Es imprescindible analizar el impacto que cada proyecto genera, tanto en el ámbito social como en el medioambiental. El objetivo de este análisis es ir más allá de un simple requisito académico, permitiendo reflexionar sobre los efectos positivos que este proyecto puede aportar a la sociedad y al medioambiente.

Tras analizar la información detallada en los apartados anteriores de esta memoria, se expone a continuación una explicación del impacto social y medioambiental que supone el desarrollo y la implementación de la aplicación msDtmt, en el marco de los Objetivos de Desarrollo Sostenible (ODS) establecidos por la Organización de las Naciones Unidas en 2015.^[44]

5.1.1 Impacto social

El desarrollo de la aplicación msDtmt para la digitalización del Trail Making Test (TMT) genera un impacto social positivo y significativo, alineándose principalmente con el ODS 3: Salud y Bienestar, que busca "Garantizar una vida sana y promover el bienestar para todos en todas las edades".

- **Detección y Seguimiento de Enfermedades Cognitivas:** La aplicación ofrece una evaluación cognitiva rápida y estandarizada. Al recoger métricas avanzadas (como pausas, velocidad del trazo y presión), proporciona a los neurólogos datos más objetivos para diagnosticar y seguir la evolución de enfermedades como la Esclerosis Múltiple o el Alzheimer.
- **Acceso a Pruebas Neuropsicológicas:** Al digitalizar la prueba TMT, se eliminan las barreras del método tradicional (lápiz y papel). La aplicación permite realizar el test de forma flexible en móviles y tabletas (Android/iOS), lo que reduce costes y libera tiempo del personal sanitario, permitiendo atender a más pacientes.
- **Mayor Fiabilidad y Reducción del Efecto Aprendizaje:** A diferencia de la versión en papel con un patrón fijo, la aplicación genera los círculos de forma aleatoria en cada test. Esto evita que el paciente memorice el recorrido, aumentando la validez de los resultados a lo largo del tiempo, algo crucial para el seguimiento de una enfermedad.

- **Contribución a la Investigación:** El sistema recopila de forma centralizada y anónima una gran cantidad de datos sobre el rendimiento en el TMT. Esta información es muy valiosa para la comunidad científica, ya que facilita estudios a gran escala sobre la cognición y la progresión de diversas patologías.

5.1.2 Impacto medioambiental

El proyecto msDtmt contribuye positivamente a la sostenibilidad medioambiental, principalmente en relación con el ODS 12: Producción y Consumo Responsables y, en menor medida, con el ODS 9: Industria, Innovación e Infraestructura.

- **Reducción del Consumo de Papel(ODS 12):** Al transformar el Test TMT a formato digital, se reduce significativamente la necesidad de utilizar papel y tintas para realizar la prueba en su formato tradicional. Cada prueba que se hace en papel puede consumir muchísimos recursos naturales y generar residuos por el uso de tintas. Con la digitalización del Test TMT se promueve un modelo de consumo más sostenible.
- **Fomento de la Innovación y la Infraestructura Sostenible (ODS 9):** El desarrollo de una solución tecnológica como msDtmt es un ejemplo de cómo la innovación puede transformar y optimizar procesos en el sector de la salud. Al crear una herramienta digital reutilizable y escalable, se promueve una infraestructura de diagnóstico más eficiente y con menor impacto ambiental que las alternativas tradicionales. La aplicación representa una modernización de las herramientas clínicas, alineándose con el objetivo de construir infraestructuras resilientes y promover la industrialización inclusiva y sostenible.

5.2 Conclusiones

A nivel de proyecto, se ha cumplido con éxito el objetivo principal de desarrollar una aplicación funcional para digitalizar el Trail Making Test, satisfaciendo todos los requisitos funcionales y no funcionales establecidos previamente. La aplicación msDtmt no solo automatiza un proceso tradicionalmente manual, sino que lo enriquece mediante la captura de métricas avanzadas que son inviables en el formato de papel y lápiz. Esto representa una herramienta de gran valor que, en el futuro, podrá ayudar a los neurólogos a optimizar su tiempo y a obtener información más objetiva y detallada sobre el estado cognitivo de sus pacientes, especialmente en el contexto de la colaboración con el Hospital Universitario de Getafe.

Desde una perspectiva académica, este proyecto ha permitido consolidar y aplicar de manera integrada las competencias adquiridas a lo largo del Grado en Ingeniería de Software. Se han puesto en práctica disciplinas clave como la Ingeniería de Requisitos para la obtención de las necesidades, la gestión de proyectos mediante la elaboración de un Product Backlog y un cronograma, el diseño de arquitecturas de software robustas como Clean Architecture, la construcción de interfaces gráficas usables y la verificación y validación del sistema a través de un plan de pruebas exhaustivo.

En el plano personal, el proyecto ha supuesto una valiosa oportunidad de crecimiento y aprendizaje autónomo, destacando la adquisición de competencias en tecnologías

modernas de desarrollo multiplataforma como Dart y el framework Flutter. La experiencia de llevar una idea desde su concepción hasta una aplicación completa y funcional ha sido sumamente enriquecedora y fundamental para el desarrollo de mi perfil como futuro ingeniero de software.

5.3 Líneas futuras

El desarrollo de la aplicación msDtmt ha sentado una base sólida y funcional, pero su potencial puede expandirse significativamente a través de varias líneas de trabajo futuras. Estas se centran en la validación clínica, el enriquecimiento funcional y la aplicación de tecnologías avanzadas.

- **Validación Clínica y Recopilación de Feedback:** El paso más inmediato y crucial es desplegar la aplicación en el entorno clínico del Hospital Universitario de Getafe para ser utilizada por pacientes reales. La recopilación sistemática de su feedback será fundamental para realizar ajustes en la usabilidad y la experiencia de usuario, garantizando que la herramienta sea accesible y cómoda para la población objetivo.
- **Calibración de Métricas y Análisis de Relevancia Clínica:** Una vez se disponga de un volumen de datos suficiente, los neurólogos podrán realizar los estudios de calibración necesarios. El objetivo es analizar qué métricas avanzadas (como la velocidad del trazo, la duración de las pausas o la presión ejercida) tienen una mayor correlación con el diagnóstico y la progresión de enfermedades como la esclerosis múltiple. Este análisis permitirá refinar la herramienta para que se enfoque en la información de mayor valor clínico.
- **Desarrollo de un Módulo de Diagnóstico Asistido:** Basándose en los resultados de la calibración, se podría desarrollar una funcionalidad avanzada que ofrezca una interpretación preliminar de los resultados. La aplicación podría calcular e indicar la gravedad o el posible estado cognitivo del paciente (por ejemplo, leve, moderado o severo) a partir de sus métricas, proporcionando un apoyo valioso al especialista.
- **Transición a Herramienta de Acceso Público:** Actualmente, la aplicación está diseñada para un uso clínico controlado mediante un código de referencia. Una vez validada y calibrada, podría evolucionar hacia una versión de acceso público. Esto permitiría a cualquier persona realizar el test TMT, fomentando la autoevaluación y la concienciación sobre la salud cognitiva.
- **Integración de Inteligencia Artificial (IA):** A futuro, se podría explorar la integración de modelos de Machine Learning para analizar las complejas métricas recogidas. Un sistema de IA podría identificar patrones sutiles en los datos del TMT que no son evidentes para el análisis estadístico tradicional, mejorando la precisión del diagnóstico y ofreciendo un pronóstico más personalizado sobre la evolución del paciente.

Acrónimos y abreviaturas

TMT	Trail Making Test (Test de Trazados)
msDTmt	Multiple Sclerosis Digital Trail Making Tes (Test de Trazados Digital para Esclerosis Múltiple)
EM	Esclerosis Múltiple
SNC	Sistema Nervioso central
LCT	Lesión Cerebral Traumática
RU	Requisito de Usuario
RF	Requisito Funcional
COM	Compatibilidad
EXT	Extensibilidad
REC	Recuperabilidad
INT	Integridad
USA	Usabilidad
ACC	Accesibilidad
EP	Épica
HU	Historia de Usuario
UI	Interfaz de Usuario
JSON	Notación de Objeto de JavaScrip

API	Interfaz de Programación de Aplicaciones
REST	Representational State Transfer (Transferencia de Estado Representacional)
i18n	Internacionalización
HTTP	Hypertext Transfer Protocol (Protocolo de Transferencia de Hipertexto)
CPU	Central Processing Unit (Unidad Central de Procesamiento)
HTML	HyperText Markup Language (Lenguaje de Marcado de Hipertexto)
UUID	Universally Unique Identifier (Identificador Único Universal)
Ms	Milisegundo
ESO	Educación Secundaria Obligatoria
GPL	GNU General Public License (Licencia Pública General de GNU)
ODS	Objetivos de Desarrollo Sostenible

6 Bibliografía

- [1] Ignacio Linari, "Unveiling Trail Making Test: visual and manual trajectories indexing multiple executive processes", *nature.com*, [En línea]. Disponible en: <https://www.nature.com/articles/s41598-022-16431-9> [Accedido: en May 2025]
- [2] Alberto García-Molina, "Evaluación de las funciones ejecutivas", *researchgate.net*, [En línea]. Disponible en: https://www.researchgate.net/profile/Alberto-Garcia-Molina/publication/322959778_Evaluacion_de_las_funciones_ejecutivas/links/5a79c6240f7e9b41dbd5f0cc/Evaluacion-de-las-funciones-ejecutivas.pdf [Accedido: en May 2025]
- [3] AEDEM-COCEMFE, "30 de mayo, Día Mundial de la Esclerosis Múltiple", *aedem.org*, [En línea]. Disponible en: <https://aedem.org/30-de-mayo-dia-mundial-de-la-esclerosis-multiple-2/> [Accedido: en May 2025]
- [4] O. Fernández, "Esclerosis múltiple Multiple sclerosis", *sciencedirect.com*, [En línea]. Disponible en: <https://www.sciencedirect.com/science/article/abs/pii/S0304541215000803> [Accedido: en May 2025]
- [5] María Paz Moya Daza, "ESCLEROSIS MÚLTIPLE REMITENTE-RECURRENTE: HALLAZGOS DE LA EVALUACIÓN CLÍNICA DESDE UNA PERSPECTIVA FONOAUDIOLÓGICA Y KINESIOLÓGICA", *researchgate.net*, [En línea]. Disponible en: https://www.researchgate.net/publication/319703834_ESCLEROSIS_MULTIPLE_REMITENTE-RECURRENTE_HALLAZGOS_DE_LA_EVALUACION_CLINICA_DESDE_UNA_PERSPECTIVA_FONOAUDIOLOGICA_Y_KINESIOLOGICA [Accedido: en Jun 2025]
- [6] Jesús Sánchez Lozano, "Eficacia de los ejercicios de pilates para mejorar el equilibrio en pacientes con esclerosis múltiple. Revisión sistemática", *revistacmfr.org*, [En línea]. Disponible en: <https://revistacmfr.org/index.php/rcmfr/article/view/444/511> [Accedido: en May 2025]
- [7] Manuel A. Sedó, "Trazados orales: un test neurológico multicultural con bajos requerimientos académicos", *dialnet.unirioja.es* [En línea]. Disponible en: <https://dialnet.unirioja.es/descarga/articulo/2875672.pdf> [Accedido: en May 2025]
- [8] Katie Marvin, "Trail Making Test (TMT)", *strokengine.ca*, [En línea]. Disponible en: <https://strokengine.ca/en/assessments/trail-making-test-tmt/> [Accedido: en Dic 2024]
- [9] Amy Y. C. Chan, "Assessing children's cognitive flexibility with the Shape Trail Test", *journals.plos.org*, [En línea]. Disponible en: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0198254> [Accedido: en Dic 2024]
- [10] Christopher Bowie, "Administration and interpretation of Trail Making Test", *researchgate.net* [En línea]. Disponible en: https://www.researchgate.net/publication/6416422_Administration_and_interpretation_of_Trail_Making_Test [Accedido: en Dic 2024]

- [11] Glenn T. Stebbins, “Neuropsychological Testing”, *sciencedirect.com* [En línea]. Disponible en: <https://www.sciencedirect.com/topics/medicine-and-dentistry/trail-making-test#:~:text=An%20average%20score%20for%20TMT,based%20on%20age%20and%20education.> [Accedido: en Dic 2024]
- [12] Esther Heerema, “The Trail Making Test for Dementia”, *verywellhealth.com*, [En línea]. Disponible en: <https://www.verywellhealth.com/dementia-screening-tool-the-trail-making-test-98624>. [Accedido: en May 2025]
- [13] Alexandra Yakeline Meneses-Meneses, “Trail Making Test TMT: Primera Aproximación A Las Propiedades Psicométricas En Población Ecuatoriana Adulta”, *scielo.senescyt.gob.ec*, [En línea]. Disponible en: http://scielo.senescyt.gob.ec/scielo.php?script=sci_arttext&pid=S2631-25812023000200049 [Accedido: en May 2025]
- [14] Llinàs-Reglà, “The Trail Making Test: Association With Other Neuropsychological Measures and Normative Values for Adults Aged 55 Years and Older From a Spanish-Speaking Population-Based Sample Jordi”, *journals.sagepub.com*, [En línea]. Disponible en: <https://journals.sagepub.com/doi/abs/10.1177/1073191115602552> [Accedido: en Dic 2024]
- [15] Dr Rohit Verma, Aplicación Android TMT App, *play.google.com*, [En línea]. Disponible en: <https://play.google.com/store/apps/details?id=com.rohitverma.tmt> [Accedido: en Dic 2024]
- [16] Simone Sacchi, Trail Making Test Android, *play.google.com*, [En línea]. Disponible en: <https://play.google.com/store/apps/details?id=com.zsimo.trailmakingtest> [Accedido: en Dic 2024]
- [17] Roland Rabien, Trail Making Test iOS, *apps.apple.com*, [En línea]. Disponible en: <https://apps.apple.com/us/app/trail-making-test/id6470496820> [Accedido: en Dic 2024]
- [18] Takuya Kawakami, Trail Making Test J Lite, *apps.apple.com*, [En línea]. Disponible en: <https://apps.apple.com/es/app/trail-making-test-j-lite/id1078507612> [Accedido: en Dic 2024]
- [19] Neuroscience Research Australia, NeuRA Trail making test, *apps.apple.com*, [En línea]. Disponible en: <https://apps.apple.com/us/app/neura-trail-making-test/id759200886> [Accedido: en Dic 2024]
- [20] CogniFit, CogniFit Trail Making Test, *cognifit.com*, [En línea]. Disponible en: <https://www.cognifit.com/us/es/bateria-de-pruebas-y-tareas/trail-making-test> [Accedido: en Dic 2024]
- [21] Andrea Bizzotto, “Flutter Project Structure: Feature-first or Layer-first?”, *github.com*, [En línea]. Disponible en: <https://github.com/bizz84/flutter-tips-and-tricks/blob/main/tips/0039-flutter-project-structure-feature-first-or-layer-first/index.md>
- [22] Semih Altin, “Flutter-Clean Architecture”, *medium.com*, [En línea]. Disponible en: <https://medium.com/@semihaltin99/flutter-clean-architecture-8759ad0213dd> [Accedido: en Mar 2025]

- [23] Duc Pham, “Flutter getx Clean Architecture”, *github.com*, [En línea]. Disponible en: <https://github.com/phamdinhduc795397/flutter-getx-clean-architecture> [Accedido: en Mar 2025]
- [24] Mochamad Firgia, “Clean Code Architecture Flutter Template”, *github.com*, [En línea]. Disponible en: https://github.com/firgia/Flutter_Template [Accedido: en Mar 2025]
- [25] Jonny Borges, “Readme de GetX en Español”, *github.com*, [En línea]. Disponible en: <https://github.com/jonataslaw/getx/blob/master/README-es.md> [Accedido: en Mar 2025]
- [26] SQLite, “What is SQLite”, *sqlitetutorial.net*, [En línea]. Disponible en: <https://www.sqlitetutorial.net/what-is-sqlite/> [Accedido: en Mar 2025]
- [27] Alex Li, “Readme de Dio” . *github.com*, [En línea]. Disponible en: <https://github.com/cfug/dio/tree/main/dio> [Accedido: en Abr 2025]
- [28] renesanse.net, “Worker Manager”, *pub.dev*, [En línea]. Disponible en: https://pub.dev/packages/worker_manager [Accedido: en Abr 2025]
- [29] Absaar, “Worker Manager iOS Steup”, *github.com*, [En línea]. Disponible en: https://github.com/fluttercommunity/flutter_workmanager/blob/main/IOS_SETUP.md [Accedido: en Abr 2025]
- [30] Sphinx, “Sphinx Getting started”, *sphinx-doc.org*, [En línea]. Disponible en: <https://www.sphinx-doc.org/en/master/usage/quickstart.html> [Accedido: en May 2025]
- [31] Dart Doc, “Dar doc doumentation”, *dart.dev*, [En línea]. Disponible en: <https://dart.dev/tools/dart-doc> [Accedido: en May 2025]
- [32] Jessamyn Dahmen, “An Analysis of a Digital Variant of the Trail Making Test Using Machine Learning Techniques”, *pmc.ncbi.nlm.nih.gov*, [En línea]. Disponible en: <https://pmc.ncbi.nlm.nih.gov/articles/PMC5384876/>[Accedido: en Ene 2025]
- [33] Amir Poreh, “Decomposition of the Trail Making Test - Reliability and Validity of a Computer Assisted Method for Data Collection”, *researchgate.net*, [En línea]. Disponible en: https://www.researchgate.net/publication/235963838_Decomposition_of_the_Trail_Making_Test_-_Reliability_and_Validity_of_a_Computer_Assisted_Method_for_Data_Collection [Accedido: en Ene 2025]
- [34] Dart, “Dart overview”, *dart.dev*, [En línea]. Disponible en: <https://dart.dev/overview> [Accedido: en May 2025]
- [35] Flutter, “Flutter architectural overview”, *docs.flutter.dev*, [En línea]. Disponible en: <https://docs.flutter.dev/resources/architectural-overview> [Accedido: en May 2025]
- [36] Android Developers, “Introducción a Android Studio”, *developer.android.com*, [En línea]. Disponible en: <https://developer.android.com/studio/intro?hl=es-419> [Accedido: en May 2025]
- [37] Developer Apple , “Documentation Xcode”, *developer.apple.com*, [En línea]. Disponible en : <https://developer.apple.com/documentation/xcode/> [Accedido: en May 2025]

- [38] Figma, “Crea diseños web atractivos y flexibles”, *figma.com*, [En línea]. Disponible en : <https://www.figma.com/es-es/web-design/> [Accedido: en May 2025]
- [39] Figma Learn, “What is Figma?”, *help.figma.com*, [En línea]. Disponible en : <https://help.figma.com/hc/en-us/articles/14563969806359-What-is-Figma> [Accedido: en Jun 2025]
- [40] Postman, “What is Postman”, *postman.com*, [En línea]. Disponible en : <https://www.postman.com/product/what-is-postman/> [Accedido: en Jun 2025]
- [41] Glassdoor, “Sueldos para el puesto”, *glassdoor.es*, [En línea]. Disponible en : <https://www.glassdoor.es/Sueldos/index.htm> [Accedido: en Jun 2025]
- [42] Richard Blatcher, “Rentabilidad de un proyecto”, *wrike.com*, [En línea]. Disponible en : <https://www.wrike.com/es/professional-services-guide/rentabilidad-de-un-proyecto/> [Accedido: en Jun 2025]
- [43] GNU GPLv3, “GNU General Public License v3.0” , *choosealicense.com*, [En línea]. Disponible en : <https://choosealicense.com/licenses/gpl-3.0/> [Accedido: en Jun 2025]
- [44] Naciones Unidas, “Objetivos de desarrollo sostenible”, *un.org*, [En línea]. Disponible en: <https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/> [Accedido: en Jun 2025]

7 Anexo

- **Casos de Respuesta del Endpoint de Validación de Código de Referencia**

Caso	Descripción	Estructura JSON de Respuesta	Interpretación en la Aplicación
Caso 1: Código Válido y Sin Usos Previos	Indica que el código es válido y puede ser utilizado para una sesión de test con cualquiera de las dos manos.	{ "exists": 0, "hands": [], "message": "OK", "status": "ok"} Ejemplo: usado con mano derecha "D"	El usuario puede seleccionar libremente la mano para realizar el test (izquierda o derecha).
Caso 2: Código Válido y Usado con Una Mano	Indica que el código es válido, pero ya ha sido utilizado para una sesión con una mano específica. La aplicación debe guiar al usuario para que realice el test con la mano restante.	{ "exists": 1, "hands": ["D"], "message": "OK", "status": "ok"} Ejemplo: usado con mano derecha "D"	Si hands contiene "D", la aplicación debe iniciar el test para la mano izquierda. Si hands contiene "I", se debe iniciar para la mano derecha.
Caso 3: Código Válido y Usado con Ambas Manos	Indica que el código es válido, pero ya se ha completado el ciclo de tests para ambas manos con este código.	{ "exists": 2, "hands": ["D", "I"], "message": "OK", "status": "ok"} Ejemplo: usado con mano derecha "D"	La aplicación debe informar al usuario que este código ya ha sido utilizado completamente y, solicitar un código diferente.
Caso 4: Código Inválido	Indica que el código de referencia introducido no es válido o no existe.	{ "message": "NoK", "status": "ok"} Ejemplo: usado con mano derecha "D"	La aplicación debe informar al usuario que el código es incorrecto e invitarle a

Caso	Descripción	Estructura JSON de Respuesta	Interpretación en la Aplicación
			que intente con otro código válido.

Tabla 5.3-1 Casos de Respuesta del Endpoint de Validación de Código de Referencia

- **Parámetros a enviar al endpoint de Envío de Resultados del Test TMT**

Parámetro	Descripción	Formato	Obligatorio
Identificación y Configuración			
codeid	Código de referencia proporcionado por el neurólogo	Formato específico establecido por el sistema (ej. "123456-78")	Sí
Mano	Mano utilizada para realizar el test	"D" para derecha, "I" para izquierda	Sí
NumCirc	Número de círculos que se presentaron en el test	15 o 25, según la configuración del test	Sí
Date_Data	Fecha en que se realizó el test	Formato ISO: "YYYY-MM-DD"	Sí
Device	Modelo del dispositivo utilizado	Formato "[marca]_[modelo]"	Sí
DiagInch	Tamaño diagonal de la pantalla en pulgadas	Valor decimal (ej. 5.8)	Sí

Parámetro	Descripción	Formato	Obligatorio
Tiempos			
Time_complete	Tiempo total para completar ambas partes del test	Segundos con decimales	Sí
Time_complete_A	Tiempo para completar la parte A del test	Segundos con decimales	Sí
Time_complete_B	Tiempo para completar la parte B del test	Segundos con decimales	Sí
Contadores de Errores			
Number_Errors	Número total de errores cometidos en ambas partes	Valor entero ≥ 0	Sí
Number_Errors_A	Número de errores cometidos en la parte A	Valor entero ≥ 0	Sí
Number_Errors_B	Número de errores cometidos en la parte B	Valor entero ≥ 0	Sí
Métricas de Pausas y Levantamientos			
Number_Pauses	Número total de pausas detectadas durante el test	Valor entero ≥ 0	Sí
Number_Lifts	Número de veces que se levantó el dedo de la pantalla	Valor entero ≥ 0	Sí

Parámetro	Descripción	Formato	Obligatorio
Average_Pause	Duración media de las pausas registradas	Segundos decimales con	Sí
Average_Lift	Duración media de los levantamientos del dedo	Segundos decimales con	Sí
Métricas de Velocidad General			
Average_Rate_Between_Circles	Velocidad media de trazo entre círculos	Píxeles segundo por	Sí
Average_Rate_Inside_Circles	Velocidad media de trazo dentro de círculos	Píxeles segundo por	Sí
Average_Time_Between_Circles	Tiempo medio empleado para trazar entre círculos	Segundos decimales con	Sí
Average_Time_Inside_Circles	Tiempo medio empleado dentro de cada círculo	Segundos decimales con	Sí
Métricas Específicas de la Parte B (Alternancia Números-Letras)			
Average_Rate_Before_Letters	Velocidad media de trazo antes de conectar con una letra	Píxeles segundo por	Sí
Average_Rate_Before_Numbers	Velocidad media de trazo antes de conectar con un número	Píxeles segundo por	Sí

Parámetro	Descripción	Formato	Obligatorio
Average_Time_Before_Letters	Tiempo medio empleado antes de conectar con una letra	Segundos con decimales	Sí
Average_Time_Before_Numbers	Tiempo medio empleado antes de conectar con un número	Segundos con decimales	Sí
Métricas de Presión y Tamaño de Toque			
Average_Total_Pressure	Presión media aplicada durante el trazo	Valor normalizado (0.0 a 1.0) o -1.0 si no es soportado	Sí
Average_Total_Size	Tamaño medio del área de contacto del dedo	Valor normalizado (0.0 a 1.0) o -1.0 si no es soportado	Sí
Puntuaciones por Secciones			
ScoreA	Puntuación total de la parte A (equivalente a Time_complete_A)	Segundos con decimales	Sí
ScoreA1	Tiempo para completar la primera sección (hasta círculo 5) de la parte A	Segundos con decimales	Sí
ScoreA2	Tiempo para completar la segunda sección (círculos 6-10) de la parte A	Segundos con decimales	Sí

Parámetro	Descripción	Formato	Obligatorio
ScoreA3	Tiempo para completar la tercera sección (círculos 11-15) de la parte A	Segundos con decimales	Sí
ScoreA4	Tiempo para completar la cuarta sección (círculos 16-20) de la parte A	Segundos con decimales o -1.0 si NumCirc=15	Solo si NumCirc=25
ScoreA5	Tiempo para completar la quinta sección (círculos 21-25) de la parte A	Segundos con decimales o -1.0 si NumCirc=15	Solo si NumCirc=25
ScoreB	Puntuación total de la parte B (equivalente a Time_complete_B)	Segundos con decimales	Sí
ScoreB1	Tiempo para completar la primera sección (hasta círculo 5) de la parte B	Segundos con decimales	Sí
ScoreB2	Tiempo para completar la segunda sección (círculos 6-10) de la parte B	Segundos con decimales	Sí
ScoreB3	Tiempo para completar la tercera sección (círculos 11-15) de la parte B	Segundos con decimales	Sí

Parámetro	Descripción	Formato	Obligatorio
ScoreB4	Tiempo para completar la cuarta sección (círculos 16-20) de la parte B	Segundos con decimales o -1.0 si NumCirc=15	Solo si NumCirc=25
ScoreB5	Tiempo para completar la quinta sección (círculos 21-25) de la parte B	Segundos con decimales o -1.0 si NumCirc=15	Solo si NumCirc=25
Información del Usuario (combinada con los resultados)			
F_nacimiento	Fecha de nacimiento del paciente	Formato ISO: "YYYY-MM-DD"	Sí
Sexo	Género del paciente	"M" para masculino, "F" para femenino	Sí
Nivel_Educ	Nivel educativo del paciente	"1" para estudios primarios "2" para estudios secundarios "G" para grado universitario "M" para máster "D" para doctorado	Sí

Tabla 5.3-2 Parámetros a enviar al endpoint de Envío de Resultados del Test TMT