



Universidad Politécnica
de Madrid



**Escuela Técnica Superior de
Ingenieros Informáticos**

Grado en Ingeniería Informática

Trabajo Fin de Grado

**Desarrollo de Funcionalidades de
Recogida de Información en una App para
la Ayuda al Diagnóstico Temprano de la
Enfermedad de Parkinson**

Autor: Martín Bellido Virtudes

Tutores: Alfonso Mateos Caballero y Carlos Javier Pérez
Sánchez

Madrid, Junio 2024

Este Trabajo Fin de Grado se ha depositado en la ETSI Informáticos de la Universidad Politécnica de Madrid para su defensa.

Trabajo Fin de Grado

Grado en Ingeniería Informática

*Título: DESARROLLO de FUNCIONALIDADES de RECOGIDA de
INFORMACIÓN en una APP para la AYUDA al DIAGNÓSTICO TEMPRANO de la
ENFERMEDAD de PARKINSON*

Junio 2024

Autor: Martín Bellido Virtudes

Tutores: Alfonso Mateos Caballero

Departamento de Inteligencia Artificial

ETSI Informáticos

Universidad Politécnica de Madrid

Carlos Javier Pérez Sánchez

Departamento de Matemáticas

Facultad de Veterinaria

Universidad de Extremadura

Resumen

Este Trabajo de Fin de Grado se centra en el avance de funcionalidades de una aplicación para dispositivos Android, diseñada para detectar de manera temprana la enfermedad de Parkinson, un trastorno neurodegenerativo que impacta a una considerable parte de la población de edad avanzada. A menudo, esta enfermedad se manifiesta a través de síntomas motores como la rigidez, los temblores y la lentitud de movimientos, así como síntomas no motores que pueden afectar profundamente el bienestar de los afectados.

Las tecnologías emergentes, como los sistemas de ayuda al diagnóstico (CAD), han empezado a jugar un papel fundamental en la identificación y el seguimiento de esta enfermedad. Estos sistemas facilitan la evaluación temprana mediante la extracción y análisis de datos obtenidos a través de tareas sencillas efectuadas por los pacientes. La aplicación desarrollada en este trabajo extiende su funcionalidad más allá de los registros básicos y análisis vocales, incorporando la técnica de *tapping*. Esta técnica permite medir de forma no invasiva la bradicinesia y la coordinación motora a través de simples interacciones con la pantalla del dispositivo.

El valor de este trabajo reside en su capacidad para integrar y mejorar las herramientas de diagnóstico de la enfermedad, proporcionando así una base sólida para futuras investigaciones y el desarrollo de nuevas soluciones tecnológicas. Al optimizar estas herramientas, el proyecto no solo busca beneficiar a los pacientes mediante una detección más temprana y un seguimiento continuo de su condición, sino que también aspira a ofrecer a los profesionales médicos un recurso valioso para la evaluación de la enfermedad de Parkinson, marcando un avance hacia la integración de tecnologías digitales en el tratamiento de trastornos neurodegenerativos.

Este enfoque sistematizado y la implementación de nuevas funcionalidades refuerzan el potencial de las aplicaciones móviles para revolucionar el diagnóstico y tratamiento de enfermedades como el Parkinson, subrayando la importancia de la innovación tecnológica en el campo de la salud neurológica.

Palabras clave: Enfermedad de Parkinson; tecnología móvil; diagnóstico precoz; tapping test; análisis de movilidad; aplicaciones de salud; innovación médica, recogida automática de datos, aplicaciones móviles.

Abstract

This project focuses on advancing an Android application designed for the early detection of Parkinson's disease, a neurodegenerative disorder that impacts a relevant portion of the global population. The disease often manifests through motor symptoms such as stiffness, tremors, and slowness of movement, as well as non-motor symptoms that can profoundly affect the well-being of those affected.

Emerging technologies, such as Computer-Aided Diagnosis (CAD) systems, have begun to play a crucial role in identifying and monitoring this disease. These systems facilitate early evaluation by extracting and analyzing data obtained through simple tasks performed by patients. The application developed in this work extends its functionality beyond basic records and vocal analysis, incorporating an innovative *tapping* technique. This technique allows for the non-invasive measurement of bradykinesia and motor coordination through simple interactions with the device's screen.

The value of this work lies in its ability to integrate and enhance diagnostic tools for the disease, thus providing a solid foundation for future research and the development of new technological solutions. By optimizing these tools, the project not only seeks to benefit patients through earlier detection and continuous monitoring of their condition but also aims to provide medical professionals with a valuable resource for evaluating and managing Parkinson's, marking a significant advancement toward the integration of digital technologies in the treatment of neurodegenerative disorders.

This systematic approach and the implementation of new functionalities reinforce the potential of mobile applications to revolutionize the diagnosis and treatment of diseases like Parkinson's, underscoring the importance of technological innovation in the field of neurological health.

Keywords: Parkinson's Disease; mobile technology; early diagnosis; tapping test; mobility analysis; health applications; medical innovation; automatic data collection; smartphone applications.

Agradecimientos

En primer lugar, deseo expresar mi más sincero agradecimiento a mis tutores, Alfonso Mateos Caballero y Carlos Javier Pérez Sánchez, cuya guía experta, paciencia y apoyo fueron fundamentales para la realización de este trabajo. Su constante disposición y preocupación por el avance de este proyecto fueron cruciales para mi desarrollo académico y personal.

A mi familia y mis amigos más cercanos, quiero agradecerles por estar siempre ahí, ofreciendo su apoyo incondicional y acompañándome en cada paso de este desafiante pero gratificante viaje que ha sido completar esta ingeniería. Su amor, motivación y comprensión han sido pilares en mi vida y en este proceso.

También agradezco a Marcos Cebreiro, por su ayuda en la búsqueda de información, debido a la similitud entre su propuesta de TFG y la mía. Su colaboración no solo enriqueció mi trabajo sino que también fortaleció nuestro compañerismo.

No puedo dejar de mencionar a todos los profesores que han dejado una huella imborrable en mi trayectoria educativa. Gracias por transmitirme no solo conocimientos, sino lecciones de vida que llevaré conmigo siempre. Cada uno de ustedes ha contribuido a moldear la persona y el profesional que soy hoy.

Finalmente, mis compañeros de clase merecen un agradecimiento especial. Juntos hemos compartido tanto los buenos momentos como los desafíos, y siempre hemos encontrado la manera de superar los obstáculos y continuar adelante. Esta experiencia compartida ha sido fundamental para mi crecimiento y les estoy profundamente agradecido por ser parte de ella.

Gracias a todos por su inestimable aporte a mi viaje académico y personal.

Este trabajo forma parte de los proyectos de I+D+i PID2021-122209OB-C31 y PID2021-122209OB-C32 financiado por MCIN/AEI/10.13039/501100011033 y los fondos FEDER "Una forma de hacer Europa"



Tabla de contenidos

| | | |
|----------|--|-----------|
| 1 | Introducción | 1 |
| 2 | Estado del arte | 3 |
| 3 | Desarrollo | 7 |
| 3.1 | Configuración de la aplicación en el equipo local..... | 7 |
| 3.2 | Estado inicial de la app | 9 |
| 3.3 | Desarrollo de <i>tapping</i> | 11 |
| 3.3.1 | Tapping Estático | 14 |
| 3.3.2 | Tapping Dinámico..... | 20 |
| 4 | Resultados | 27 |
| 4.1 | Síntesis de Resultados | 27 |
| 4.2 | Comparación con Estado del Arte | 27 |
| 4.3 | Discusión de Métricas para Diagnóstico..... | 28 |
| 4.4 | Impacto Clínico y Tecnológico | 31 |
| 5 | Conclusión | 32 |
| 6 | Análisis de Impacto | 33 |
| | Bibliografía | 35 |
| | ANEXO I | 37 |
| | ANEXO II | 39 |

1 Introducción

La enfermedad de Parkinson (EP) es un trastorno neurodegenerativo progresivo que afecta principalmente a la población de edad avanzada, aunque no exclusivamente. Se caracteriza por la pérdida de neuronas dopaminérgicas en la *substantia nigra* del cerebro, lo que lleva a síntomas motores distintivos como temblores en reposo, bradicinesia (lentitud de movimiento), rigidez muscular y, en etapas avanzadas, inestabilidad postural. Además, la EP puede manifestar síntomas no motores, incluidos trastornos del sueño, depresión, ansiedad y problemas cognitivos, que pueden preceder a los trastornos motores y afectar a la calidad de vida [1].

La detección temprana de la EP es esencial, ya que un diagnóstico precoz puede permitir el inicio oportuno de tratamientos que alivien los síntomas, mejoren la calidad de vida y, en algunos casos, ralenticen la progresión de la enfermedad. Sin embargo, el diagnóstico de la EP sigue siendo un desafío, principalmente debido a la variabilidad de sus manifestaciones y la superposición con otros trastornos neurodegenerativos. Tradicionalmente, el diagnóstico se basa en la evaluación clínica de los síntomas motores.

En este contexto, las tecnologías digitales han supuesto un avance prometedor para facilitar la detección y el seguimiento de la EP. Las aplicaciones móviles, en particular, ofrecen un medio accesible y no invasivo para recopilar datos sobre los síntomas motores y no motores de los usuarios. Entre las diversas funcionalidades que estas aplicaciones pueden incorporar, el método de "tapping" o pruebas de pulsación representa una técnica valiosa para evaluar la bradicinesia y la coordinación motora. Al analizar la velocidad, ritmo y regularidad con la que un individuo realiza secuencias de toques en la pantalla del dispositivo, es posible obtener indicadores objetivos de alteraciones motoras típicas de la EP.

El presente Trabajo de Fin de Grado (TFG) se centra en el desarrollo de funcionalidades de recogida de información en una aplicación para dispositivos móviles, destinada a asistir en el diagnóstico temprano de la enfermedad de Parkinson. Este proyecto aborda específicamente la implementación del método de *tapping* como herramienta de apoyo al diagnóstico, seleccionado tras una revisión exhaustiva de las aplicaciones existentes y los síntomas característicos de la EP. El objetivo principal de este proyecto es incorporar en la aplicación una técnica sencilla pero efectiva que permita la recogida automática de datos relevantes, facilitando así el diagnóstico temprano de la enfermedad. En la aplicación que se proporcionó para continuar el desarrollo ya estaba implementada la función de grabación de voz como método de recogida de datos. La integración del análisis de voz y análisis de *tapping* conduce a la utilización de una herramienta que puede mejorar los resultados predictivos de cada una de las técnicas por separado.

Además de contribuir al ámbito clínico mediante la mejora de las herramientas de diagnóstico, este TFG busca explorar el potencial de las tecnologías móviles en el manejo de trastornos neurodegenerativos. Al hacerlo, este trabajo no solo tiene la potencialidad de beneficiar a los pacientes mediante la detección temprana y el seguimiento de la enfermedad, sino también de ofrecer a los profesionales de la salud una herramienta adicional para la evaluación y manejo de la EP, marcando un paso significativo hacia la integración de la tecnología digital en el cuidado de la salud neurológica.

Los objetivos específicos de este Trabajo de Fin de Grado se dividen en tres puntos principales, diseñados para garantizar una aproximación sistemática al desarrollo y evaluación de una aplicación orientada a facilitar el diagnóstico temprano de la enfermedad de Parkinson. Estos objetivos se distribuyen temporalmente para optimizar los recursos y maximizar la eficiencia del proyecto:

- 1. Investigación del Estado del Arte en Aplicaciones de Diagnóstico para la Enfermedad de Parkinson:** Se ha desarrollado un estudio detallado sobre las aplicaciones disponibles que contribuyen al diagnóstico de la enfermedad de Parkinson, examinando las tareas específicas que abordan y la eficacia de estas herramientas. Este análisis permite identificar las mejores prácticas y determinar cuál de las tareas diagnosticadas es más relevante y prometedora para su implementación.
- 2. Implementación y Demostración de Funcionalidades Seleccionadas:** Tras identificar la tarea más relevante a través de la investigación, se procederá a su implementación en la aplicación. Este objetivo central buscará no solo incorporar la funcionalidad elegida sino también estudiar su viabilidad en casos simulados, ya que no se dispone de acceso a enfermos con Parkinson.
- 3. Elaboración de la Memoria del TFG:** Este documento final recogerá de manera exhaustiva todo el proceso de desarrollo del proyecto, incluyendo la metodología empleada, los resultados obtenidos, y las conclusiones derivadas. La memoria servirá como registro detallado del trabajo realizado y como reflexión sobre el impacto y las implicaciones del proyecto.

Este Trabajo de Fin de Grado consta de seis capítulos más la bibliografía y los anexos:

- **Introducción:** Se presenta la enfermedad de Parkinson, destacando la importancia de su diagnóstico temprano y cómo la aplicación móvil desarrollada contribuye a este propósito.
- **Estado del arte:** Se revisan las tecnologías y aplicaciones existentes relacionadas con el diagnóstico y seguimiento de la enfermedad de Parkinson, evaluando sus funcionalidades y limitaciones.
- **Desarrollo:** Describe detalladamente el proceso de implementación de la funcionalidad de tapping en la aplicación, incluyendo aspectos técnicos y desafíos encontrados.
- **Resultados:** Se analizan los resultados obtenidos de las pruebas de tapping implementadas, comparando con el estado del arte y discutiendo las métricas para el diagnóstico.
- **Conclusión:** Se recapitulan los logros del proyecto, su impacto en el diagnóstico de la enfermedad de Parkinson y se sugieren direcciones para futuras investigaciones.
- **Análisis de Impacto:** Evalúa el impacto potencial del proyecto en términos de beneficios para pacientes con Parkinson y contribuciones a objetivos de desarrollo sostenible.

2 Estado del arte

Para analizar el estado del arte de este proyecto buscamos las aplicaciones más relevantes del diagnóstico del Parkinson en los últimos 10 años. Se hizo una búsqueda que abarcó varios pasos.

Definición de palabras clave: Se seleccionaron términos como “*Parkinson's disease*”, “*mobile application*”, “*diagnostic tools*”, “*machine learning*”, “*neurological disorders*”, “*Smartphone*”, y “*digital health*” para capturar un amplio espectro de desarrollos relevantes.

Se utilizaron PubMed, y Google Scholar para acceder a estudios recientes que evalúan aplicaciones móviles en el contexto del Parkinson.

Repositorios de aplicaciones y reseñas: Plataformas como Google Play Store y Apple App Store, junto con sitios especializados como MDS (Movement Disorders Society) y Digital Health, proporcionaron datos sobre aplicaciones comerciales.

También los tutores nos proporcionaron un artículo en el que se hacía un estudio de algunas aplicaciones móviles para trastornos del movimiento.

Tabla 1. Aplicaciones para trastornos del movimiento.

| Nombre de la Aplicación | Trastorno(s) de Movimiento | Propósito | Dominios de Uso | Sistema Operativo |
|------------------------------------|-----------------------------------|---|--|--------------------------|
| iPrognosis | Parkinson | Recopilación de datos de comportamiento | Voz, Estabilidad al sostener, Análisis de movimiento | Android, iOS |
| Parkinson mPower 2 | Parkinson | Rastreo de síntomas mediante sensores y cuestionarios | Actividad de los síntomas | Android, iOS |
| TREMOR12 | General (temblores) | Cuantificación de temblores | Monitoreo de temblores | iOS |
| APKINSON | Parkinson | Evaluación diaria de síntomas | Habla, Congelación de la marcha | Android, iOS |
| Mon4t | Alzheimer, Huntington, Parkinson | Monitoreo neurológico | Cognitividad, Temblor, Emociones | Android, iOS |
| HopkinsPD | Parkinson | Detección de respuesta a medicamentos | Salud general | Android, iOS |
| Roche PD Mobile Application | Parkinson | Monitoreo remoto de síntomas | Biomarcadores digitales | Android, iOS |
| iSeismometer | General (temblores) | Monitoreo de temblores | Temblores finos | iOS |

| | | | | |
|----------------------|--------------------------|-----------------------|---|--------------|
| StudyMyTremor | Trastornos con temblores | Análisis de temblores | Temblores | Android, iOS |
| Lift Pulse | General (temblores) | Medición de temblores | Intensidad y características de temblores | Android, iOS |

iPrognosis: Esta aplicación, lanzada en 2017 como parte de un proyecto de investigación de la UE, recopila datos de comportamiento de manera discreta a través de la interacción natural de los usuarios con sus dispositivos inteligentes. Utiliza la interacción del usuario con dispositivos inteligentes para recoger datos sobre características de la voz, estabilidad al sostener el teléfono y análisis de movimiento. Esta innovadora aproximación permite un seguimiento continuo y discreto que podría revelar signos tempranos de la enfermedad de Parkinson [2].

Parkinson mPower 2: Ofrecida por Sage Bionetworks, esta aplicación recopila datos mediante cuestionarios y sensores del teléfono para rastrear los síntomas del Parkinson y ha sido utilizada por más de 10,000 personas a nivel mundial. Se destacan sus innovadoras mediciones basadas en la actividad de los síntomas, incluyendo el toque de dedos y la medición de temblores [3].

TREMOR12: Lanzada en 2016, esta aplicación utiliza los sensores incorporados en dispositivos iOS para cuantificar los temblores en pacientes, ofreciendo una herramienta de bajo costo para la evaluación del tratamiento [4].

APKINSON: Basada en métodos ya establecidos, esta aplicación evalúa aspectos como la articulación y pronunciación en el habla y la congelación de la marcha, entre otros, a través de ejercicios diarios [5].

Mon4t: Ofrece monitorización neurológica mediante pruebas de cognitivdad, evaluación del temblor y cuestionarios para evaluar emociones, proporcionando una perspectiva profunda para detectar precozmente enfermedades como el Alzheimer, la enfermedad de Huntington o el Parkinson [6].

HopkinsPD: Diseñada para detectar la respuesta a medicamentos, la aplicación HopkinsPD podría asistir en la toma de decisiones terapéuticas. Ha sido diseñada para recoger datos sensibles sobre la salud de manera segura. [7].

Roche PD Mobile Application: Esta aplicación ha sido evaluada en un ensayo clínico y ha recopilado biomarcadores digitales clínicamente relevantes para la enfermedad de Parkinson. Se usa para la monitorización remota de síntomas y ha demostrado ser confiable y válida para el seguimiento temprano de la enfermedad [8].

iSeismometer: Es una aplicación que convierte un dispositivo iOS en un sismógrafo en tiempo real, permitiendo medir y registrar movimientos. Aunque originalmente diseñada para detectar vibraciones del suelo, su sensibilidad la hace útil para registrar temblores humanos, ofreciendo una manera innovadora de monitorear los temblores en pacientes con trastornos del movimiento [9].

Aunque su aplicación primaria no es médica, la capacidad de iSeismometer para capturar movimientos finos puede ser aprovechada para evaluar la

presencia y severidad de temblores en pacientes, proporcionando datos útiles para la evaluación clínica.

StudyMyTremor: aprovecha los sensores de los dispositivos móviles para analizar y cuantificar los temblores en los usuarios. Esta aplicación está específicamente diseñada para personas con trastornos que incluyen temblores como síntoma, facilitando la autoevaluación y el seguimiento de la efectividad de los tratamientos [10].

Lift Pulse: es otra aplicación que utiliza la tecnología de los smartphones para medir y analizar los temblores. Se centra en la facilidad de uso, permitiendo a los usuarios realizar mediciones con solo sostener el dispositivo en la mano, ofreciendo resultados inmediatos sobre la intensidad y características de sus temblores [11].

La aplicación más interesante fue Parkinson mPower 2. Esta herramienta no solo permite a los usuarios rastrear sus síntomas y compartir esta información con investigadores, sino que también introduce mediciones innovadoras basadas en actividades diarias para una comprensión más profunda de la enfermedad. La aplicación utiliza cuestionarios y aprovecha los datos de sensores del móvil para recopilar información de forma continua. Los participantes pueden elegir cómo desean compartir sus datos, ya sea solo con el equipo de estudio de mPower y sus socios o de manera más amplia con investigadores cualificados en todo el mundo. Durante los primeros seis meses del estudio (960 participantes), más del 75% de los participantes de mPower optaron por compartir sus datos de forma amplia con investigadores cualificados [12]. Estudios más recientes indican que la aplicación se ha usado previamente en un estudio observacional de más de 15000 participantes [21].

Entre las características descritas en el estudio, se incluyen:

- Medición de la marcha y el equilibrio, usando sensores de giroscopio y acelerómetro.
- Juegos de memoria visoespacial que evalúan la capacidad de memoria a corto plazo.
- Pruebas de destreza que implican tocar la pantalla del teléfono con dedos alternos para evaluar la precisión, velocidad y ritmo de los toques.
- Grabaciones de voz de los participantes diciendo "Aaah" durante 10 segundos para evaluar características vocales, incluido el temblor vocal.

mPower destaca no solo por su amplio alcance sino también por su enfoque en permitir una comprensión más profunda de la variabilidad diaria en los síntomas de la enfermedad de Parkinson y los efectos de la medicación, con el objetivo final de mejorar la calidad de vida general de los pacientes. La aplicación está disponible de forma gratuita y es accesible para cualquier persona mayor de 18 años que tenga un iPhone, sin importar si han sido diagnosticados con Parkinson o no, lo que también permite recolectar datos de control de participantes saludables. Se descargó esta aplicación para realizar pruebas y monitorear mis propios síntomas, destacando su utilidad y accesibilidad para los pacientes.

El desarrollo de aplicaciones móviles para el diagnóstico y seguimiento del Parkinson es un área prometedora que continúa evolucionando, con el potencial de mejorar significativamente la calidad de vida de los pacientes. El uso de tecnologías móviles permite una detección más temprana de los síntomas, un seguimiento más detallado de la progresión de la enfermedad y una gestión más efectiva del tratamiento.

3 Desarrollo

En este capítulo, se explica detalladamente el proceso de desarrollo de la función de “tapping” en la aplicación móvil original. Este desarrollo incluye la configuración inicial del entorno, el estado de la aplicación al momento de recibirla, y los pasos específicos tomados para implementar las pruebas de “tapping” estático y dinámico.

3.1 Configuración de la aplicación en el equipo local

Para garantizar el correcto funcionamiento de la aplicación en el entorno de desarrollo, fue esencial establecer una configuración local que pudiera simular el entorno de producción. XAMPP [13], un paquete de software que incluye Apache, MySQL, y PHP, se utilizó como la base para esta configuración debido a su amplia compatibilidad y facilidad de uso.

Instalación de XAMPP:

El primer paso fue la instalación de XAMPP, descargando la versión más reciente del software desde su página oficial. Tras la instalación, se realizó la configuración inicial para asegurar que los módulos de Apache y MySQL [14] estuvieran activos y funcionando correctamente, como se muestra en la Figura 1, que corresponde al Panel de Control de XAMPP.

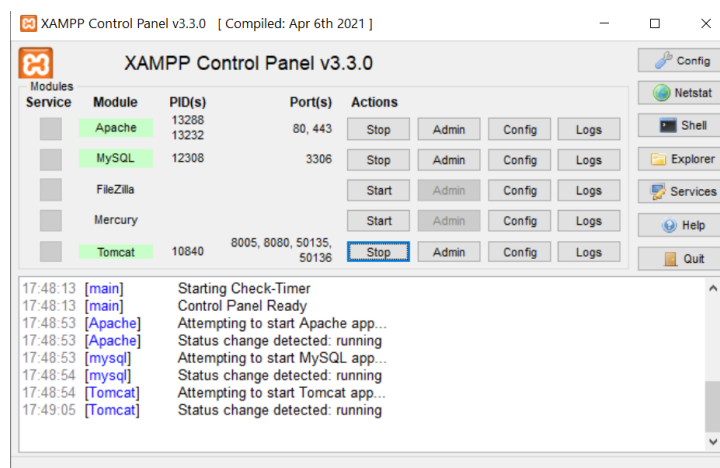


Figura 1. Panel de Control de XAMPP.

Creación y Configuración de la Base de Datos:

Con los servicios esenciales en ejecución, procedí a la creación de la base de datos a través de la interfaz de phpMyAdmin [15], un administrador de bases de datos incluido en XAMPP, accesible a través del navegador. En la Figura 2, se ilustra la estructura de la base de datos denominada “test”, que contiene las tablas necesarias para la aplicación, como grabaciones, médicos, y pacientes.

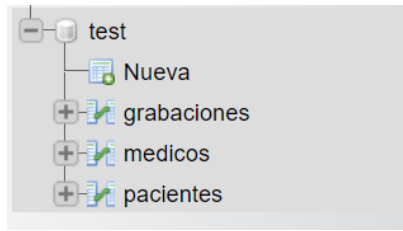


Figura 2. Estructura de la base de datos.

Conexión con la Aplicación:

La siguiente fase implicó ajustar la configuración de la aplicación para conectar con la base de datos recién creada. Se modificaron los archivos de configuración relevantes para establecer los parámetros de conexión correctos, garantizando así una comunicación fluida entre la aplicación y la base de datos.

Verificación de Datos y Funcionalidad:

Para validar la configuración realizada, se efectuaron varias pruebas de conexión y manipulación de datos dentro de la aplicación. Como se observa en la Figura 3, se realizaron consultas en *phpMyAdmin* para verificar que los datos estuvieran siendo gestionados adecuadamente y que la interacción con la base de datos fuera consistente con las expectativas del funcionamiento de la aplicación.

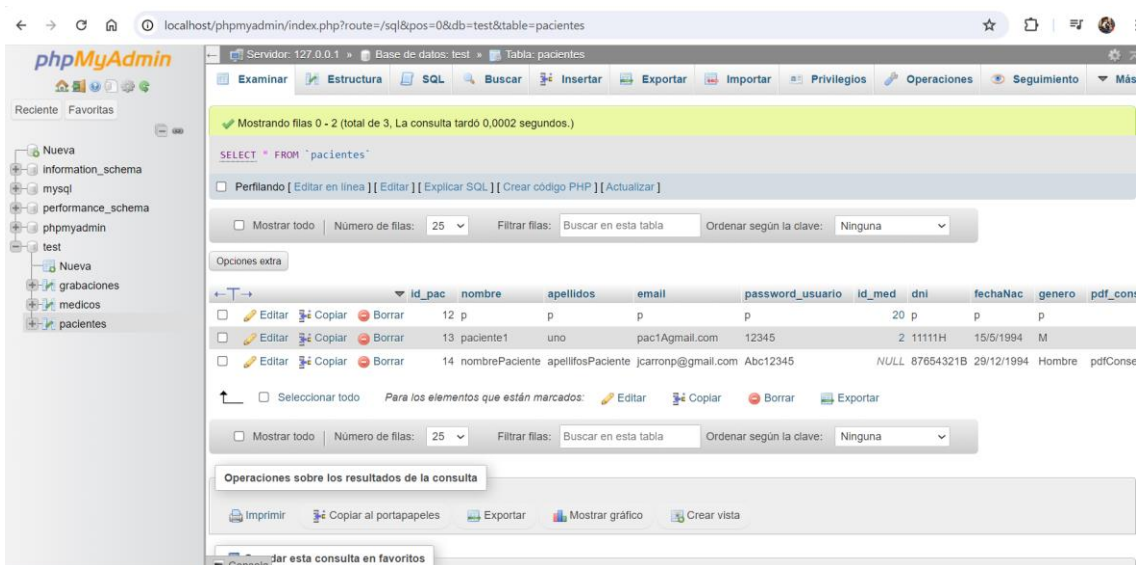


Figura 3. Base de datos de *phpMyAdmin* en la que se muestran los datos de prueba de los pacientes.

3.2 Estado inicial de la app

Al inicio del Trabajo de Fin de Grado (TFG), la aplicación se encontraba operativa con una interfaz de usuario intuitiva centrada en la facilidad de acceso y manejo para los usuarios. Como se muestra en la Figura 4, la pantalla de bienvenida permitía a los usuarios iniciar sesión o registrarse como pacientes o médicos, ofreciendo una entrada directa y sin complicaciones a la aplicación.



Figura 4. Pantalla de bienvenida.

La función principal disponible para los usuarios era la grabación de voz. En las figuras 5 y 6 se muestra como los pacientes podían grabar su voz siguiendo instrucciones claras y concisas. Estas grabaciones, una vez realizadas, presentaban opciones de escucha, envío o descarte, lo que indica un proceso de revisión y manejo de las grabaciones antes de su finalización.

Todo esto estaba disponible en el caso de que el tipo de cuenta elegida fuera la de paciente, que para este proyecto es en la que nos hemos centrado para poder implementar la funcionalidad del *tapping*.

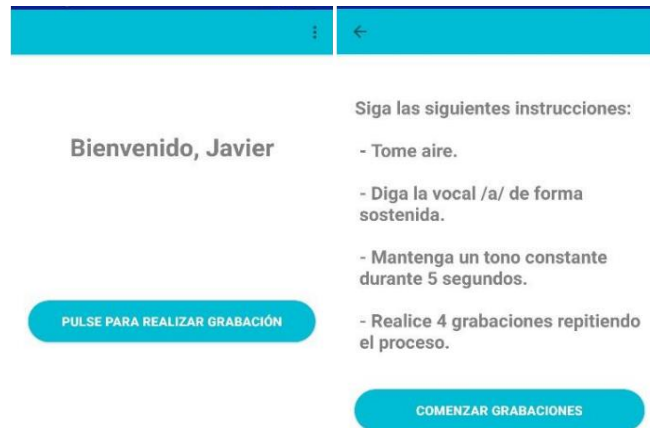


Figura 5. Pantallas para el paso de realizar las grabaciones de voz.



Figura 6. Función de realizar la grabación

Adicionalmente, la aplicación proporcionaba funcionalidades administrativas para los profesionales de la salud, como el registro de nuevos pacientes, destacando la estructura de la aplicación para atender las necesidades de ambos grupos de usuarios, representado en la Figura 7.

Bienvenido, Javier

Introduzca los datos del paciente:

Nombre _____

Apellidos _____

DNI/Pasaporte _____

DAR DE ALTA UN PACIENTE

DAR DE ALTA

Figura 7. Pantalla de dar de alta a un paciente con sus respectivos datos a introducir.

La aplicación, en su estado original, estaba por lo tanto enfocada en proporcionar un medio para la recolección de datos acústicos y la administración de perfiles de usuario, estando pendiente de ampliación en cuanto a la implementación de nuevas funcionalidades que podrían enriquecer las utilidades para diagnosticar y mejorar la experiencia del usuario.

Con este fundamento establecido, mi aportación al proyecto fue la introducción del método de *tapping*, tanto estático como dinámico, lo cual representó una expansión significativa de las capacidades de la aplicación. Estos nuevos módulos permitirían una evaluación más compleja de los síntomas motores, complementando así la información recopilada a través de las grabaciones de voz.

3.3 Desarrollo de *tapping*

Lo primero que se hizo fue añadir el botón para ir a la prueba de *tapping* en el *main* del paciente, justo abajo del botón de realizar la prueba de grabación como se puede ver en la Figura 8.

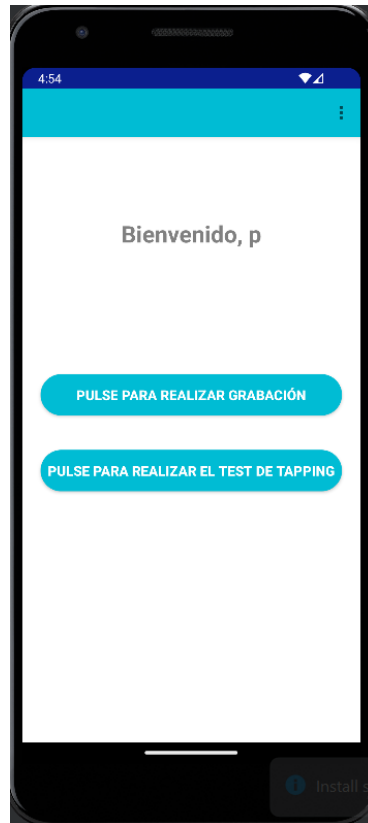


Figura 8. Pantalla de inicio para elegir la prueba a realizar

El botón de “PULSE PARA REALIZAR EL TEST DE TAPPING” lleva al usuario a la pantalla mostrada en la Figura 9, donde el usuario podrá elegir qué tipo de prueba de *tapping* realizar, el *tapping* estático o *tapping* dinámico. Estas pruebas ofrecen enfoques complementarios para evaluar capacidades motoras en el contexto de trastornos del movimiento como el Parkinson.

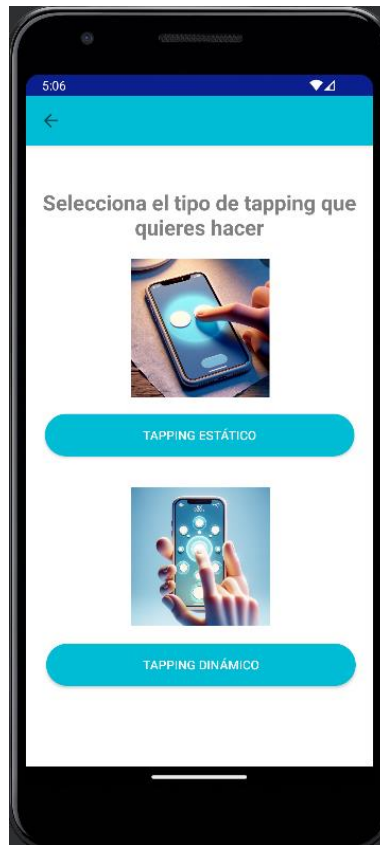


Figura 9. Pantalla para seleccionar entre *tapping* estático y dinámico. Fuente de gráficos: generados con inteligencia artificial con la herramienta DALL-E [16].

En la prueba de *tapping* estático, los participantes interactúan con botones fijos en la pantalla, apuntando a evaluar la velocidad y consistencia con la que pueden realizar *taps*. Esta prueba se centra en la capacidad de ejecutar una tarea repetitiva y constante, donde se registran métricas como la velocidad de *tapping* (toques por segundo), la desviación estándar de la velocidad y la cantidad de errores (toques fuera de los botones designados). La visualización de la velocidad media de *tapping* a lo largo del tiempo puede indicar variaciones en el desempeño de la tarea por parte del participante, como aquellos producidos por la fatiga.

Por otro lado, la prueba de *tapping* dinámico desafía a los participantes a tocar objetivos móviles que aparecen en diferentes lugares de la pantalla. Esta prueba evalúa la reactividad y la capacidad de adaptación a cambios en el entorno, midiendo no solo el número de toques exitosos y los errores, sino también el tiempo de inicio de movimiento (TMI), que es el tiempo que tarda el participante en reaccionar y tocar un objetivo desde su aparición. A diferencia de la prueba estática, donde la interacción es predecible, la prueba dinámica introduce un elemento de incertidumbre y requiere seguimiento visual y coordinación motora ajustada.

Ambas pruebas tienen su lugar en la evaluación de trastornos del movimiento. Mientras que la prueba estática ofrece una medida de la habilidad para mantener una tasa constante de interacción con objetivos conocidos y fijos, la prueba dinámica proporciona información sobre cómo los individuos se ajustan a estímulos cambiantes y su habilidad para gestionar tareas que requieren una adaptación rápida. En conjunto, ofrecen una visión de las

capacidades motoras de un individuo, crucial para entender y tratar efectivamente trastornos como el Parkinson.

3.3.1 Tapping Estático

Para abordar el desarrollo del módulo de *tapping* estático en la aplicación, primero se identificó la necesidad de crear una interfaz intuitiva que pudiera guiar a los usuarios a través del proceso de la prueba. El diseño de la interfaz debía ser sencillo pero funcional, con botones claramente marcados para iniciar y terminar la prueba, así como para indicar errores y registrar la velocidad de tapping.

Lo primero que se hizo fue crear la pantalla en la que se le muestra al usuario las instrucciones de la prueba que va a realizar como se muestra en la Figura 10.

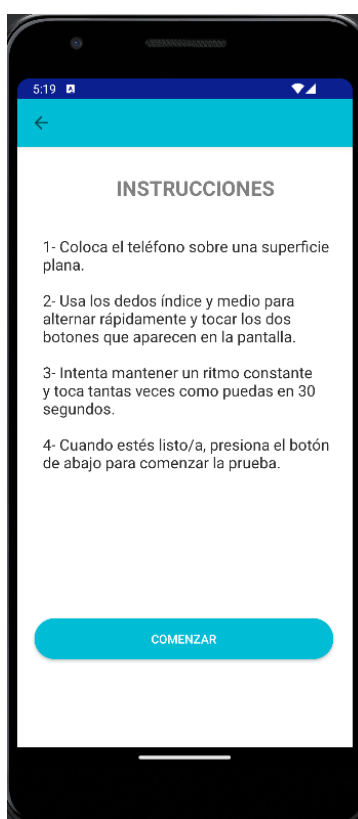


Figura 10. Instrucciones de la prueba de tapping estático.

El núcleo del desarrollo se centró en la programación de la lógica que detecta y responde a los toques en la pantalla. Para esto, se implementó un sistema que registra cada toque, calcula la velocidad de los toques y monitoriza la consistencia del usuario a lo largo del tiempo. Se tuvieron en cuenta las posibles variaciones en la velocidad de *tapping* para poder detectar patrones que podrían indicar fatiga o irregularidad motora, comunes en pacientes con enfermedad de Parkinson.

Otro aspecto importante fue la gestión de errores, identificando y contabilizando los toques que se hacían fuera de los botones designados. Esto implicaba no solo detectar la ubicación de los toques en la pantalla, sino

también ofrecer una retroalimentación visual para informar al usuario de los errores cometidos, de forma que la prueba fuera también una herramienta educativa y de autoevaluación.

Para los análisis de resultados, se desarrollaron funciones que generaban representaciones gráficas de la fatiga del usuario a lo largo del tiempo, midiendo la velocidad en cada segundo. Esto permite a los usuarios y médicos obtener una comprensión visual inmediata del rendimiento motriz durante la prueba.

La implementación del módulo de *tapping* estático también incluyó una rigurosa fase de prueba y depuración para garantizar que la funcionalidad fuera confiable y precisa. Se realizaron pruebas simulando el comportamiento de un paciente con Parkinson y un paciente sano que se comentará más adelante.

La culminación de este proceso de desarrollo fue la integración exitosa del módulo de *tapping* estático en la aplicación existente, lo que amplía la capacidad de recogida de datos y mejora la utilidad de la aplicación para el diagnóstico de la enfermedad de Parkinson.

Una vez iniciada la prueba, llevaba a la pantalla de la Figura 11, en la cual se mostraba el contador de toques, el tiempo restante, y los dos botones que tenía que pulsar el usuario.



Figura 11. Pantalla de la actividad del *tapping* estático

Para llevarlo a cabo hizo falta crear dos clases java, *StaticTappingActivity* y *StaticTappingTestActivity*.

StaticTappingActivity

El *StaticTappingActivity.java* es una actividad de lanzamiento que proporciona la interfaz para iniciar la prueba de tapping estático. Contiene un botón que, cuando se toca, inicia *StaticTappingTestActivity*.

Las razones por las cuales se han decidido recoger y calcular estos datos son las siguientes:

1. Evaluación de la *bradicinesia*: La *bradicinesia*, o la lentitud de movimiento, es uno de los síntomas del Parkinson. Al medir la velocidad de *tapping* (número de toques por segundo), se puede evaluar cuán rápido puede moverse el paciente. Una disminución en la velocidad de *tapping* puede indicar la presencia de *bradicinesia*.
2. Monitoreo de la fatiga muscular: La fatiga puede afectar significativamente a los pacientes con Parkinson, y puede ser evaluada observando cómo cambia la velocidad de *tapping* a lo largo del tiempo. Una disminución progresiva en la velocidad podría sugerir que el paciente se fatiga rápidamente, un síntoma común en el Parkinson.
3. Detección de irregularidades motoras: La consistencia en la velocidad de *tapping* (evaluada a través de la desviación estándar de la velocidad) puede revelar variabilidad en la capacidad motora. En los pacientes con Parkinson, puede haber una mayor variabilidad debido a la alteración del control motor.

La decisión de calcular la velocidad máxima, la velocidad media y la desviación estándar en una prueba de *tapping* se basa en que estos datos son fáciles de obtener y analizar, ofrecen resultados reproducibles y pueden ser comparados con estándares de referencia o con mediciones previas del mismo paciente para evaluar el progreso o deterioro de la condición.

StaticTappingTestActivity

Esta clase gestiona la prueba de *tapping* estático y contiene elementos de interfaz de usuario como botones, texto y gráficos, además de lógica para manejar el toque en la pantalla y calcular métricas de rendimiento.

Componentes clave

Contadores y cronómetros: Se utiliza una variable para manejar la cuenta regresiva antes de que comience la prueba y durante la prueba misma.

Detectores de toque: se usó una variable para detectar toques, calcular la velocidad de toques por segundo y registrar errores si los toques ocurren fuera de los botones.

Visualización de datos: La librería *MPAndroidChart* [17] se usa para mostrar un gráfico de líneas que representa la velocidad de toque a lo largo del tiempo, esto es útil para identificar patrones como la fatiga.

Manejo de errores y toques fuera de los botones: Se añade un detector de toques en el área general de la pantalla para contar los errores, es decir, toques fuera de los botones designados. Esto es crucial para evaluar la precisión del usuario durante la prueba.

Funciones matemáticas y lógica detallada

Registro y cálculo de velocidad de toques: En el método `setupTouchListener`, se establece un escuchador para los botones que representan áreas de toque. Cada vez que un botón es presionado, se registra el tiempo del evento y se actualiza un contador de toques por segundo (`tapsPerSecond`). Como se puede observar en la Figura 12, este código permite calcular la velocidad en tiempo real, que es el número de toques en un segundo específico.

```
153         long currentSecond = (currentTime - startTime) / 1000;
154
155         if (!tapsPerSecond.containsKey(currentSecond)) {
156             tapsPerSecond.put(currentSecond, 1);
157         } else {
158             tapsPerSecond.put(currentSecond, tapsPerSecond.get(currentSecond) + 1);
```

Figura 12. Código para el registro de tiempo en cada toque

Durante el inicio de la prueba, se inicializa el `HashMap` `tapsPerSecond` para mantener un registro del número de toques por cada segundo transcurrido, como se observa en la Figura 13.

```
133         long testDurationSeconds = 30;
134         for(long i = 0; i < testDurationSeconds; i++) {
135             tapsPerSecond.put(i, 0); // Inicializar cada segundo con 0 toques
136         }
```

Figura 13. Código para la inicialización del registro de toques

Cada vez que el usuario toca uno de los botones designados para la prueba (`circleLeft` o `circleRight`), el manejador de eventos de click incrementa el conteo de toques para el segundo actual. Esto se calcula tomando la diferencia de tiempo desde el inicio de la prueba hasta el momento del toque, dividido por 1000 (para convertir milisegundos a segundos).

Si se intenta tocar el mismo botón varias veces seguidas entonces el contador de errores va aumentando, ya que si el usuario no va alternando entre el botón derecho e izquierdo no se cuentan los toques. La lógica de esta parte del código se muestra en la Figura 14.

```
// Variable para almacenar el ID del último botón pulsado
final int[] lastButtonId = {-1};
View.OnClickListener tapListener = new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        if (lastButtonId[0] == v.getId()) {
            // El mismo botón fue pulsado consecutivamente
            errors++;
            return;
        }
        lastButtonId[0] = v.getId(); // Actualiza el último botón pulsado
```

Figura 14. Código que comprueba si el botón ha sido pulsado consecutivamente

Como muestra la Figura 15, al final de la prueba, la lista *speeds* se llena con los valores de *tapsPerSecond* para cada segundo registrado. Esto ocurre justo antes de calcular la velocidad media, máxima y la desviación estándar.

```
275 ArrayList<Integer> speeds = new ArrayList<>(tapsPerSecond.values());
```

Figura 15. Código para inicializar la variable donde se guardan las velocidades

Cálculo de velocidad media: Para calcular la velocidad media, se suman todas las velocidades registradas y luego se divide esta suma por el número de entradas (número de segundos en que se registraron toques). El código mostrado en la Figura 16 proporciona un promedio de la velocidad de toques por segundo durante la duración total de la prueba.

```
276 double sum = 0;
277 for (int speed : speeds) {
278     sum += speed;
279 }
```

Figura 16. Código para el cálculo de la velocidad media.

Cálculo de desviación estándar: Para calcular la desviación estándar de las velocidades, se utiliza el método `calculateStdDev`, mostrado en la Figura 17. Este método toma la lista de velocidades y el promedio, calcula la suma de las diferencias al cuadrado del promedio y luego toma la raíz cuadrada de la suma dividida por el número de velocidades. La desviación típica mide la variabilidad en la velocidad de toques, en concreto se utiliza la fórmula siguiente:

$$s = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2}$$

```
230 @private double calculateStdDev(ArrayList<Integer> speeds, double mean) {
231     double sum = 0.0;
232     for (int speed : speeds) {
233         sum += Math.pow(speed - mean, 2);
234     }
235     return Math.sqrt(sum / speeds.size());
236 }
```

Figura 17. Función en el programa para calcular la desviación estándar

Después de la prueba, como se puede ver en la Figura 18, la aplicación muestra estadísticas como la velocidad máxima, la velocidad media y la desviación estándar, junto con el número total de toques y errores, y la gráfica de la evolución en toques/segundos a lo largo de los 30 segundos.



Figura 18. Resultados de la prueba del tapping estático

Una vez terminada la prueba aparecen los botones de *Elegir prueba de tapping*, el cual lleva al usuario a la pantalla de la Figura 9 comentada anteriormente, el botón de *Repetir Test*, el cual permite al usuario volver a realizar la prueba, y un último botón llamado *Descargar Fichero* el cual permite descargar en el smartphone del usuario un fichero Excel con los resultados de la prueba realizada, dicho fichero se guarda con el formato de nombre *StaticNombreApellido-AñoMesDiaHoraMinutosSegundos.csv*. Al pulsar este botón de descarga de fichero salta una notificación al smartphone, la cual indica que el fichero ha sido guardado, como se puede ver en la Figura 19. El usuario será capaz de acceder a este archivo pulsando la notificación o accediendo manualmente a la carpeta de descargas.



Figura 19. Notificación de fichero descargado.

3.3.2 Tapping Dinámico

El módulo de *tapping* dinámico fue diseñado para crear un entorno interactivo donde se generan círculos que el usuario debe tocar lo más rápidamente posible.

Al igual que en el tapping estático, lo primero que se hizo fue crear la pantalla para iniciar la prueba en la que se muestran las instrucciones al usuario (véase la Figura 20).

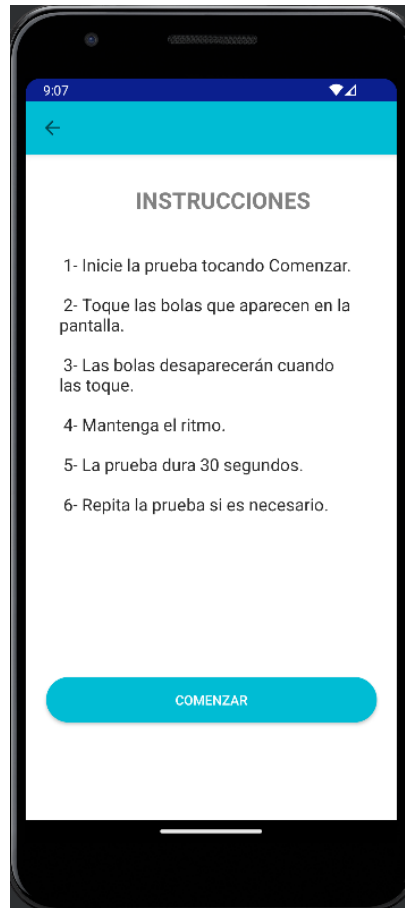


Figura 20. Instrucciones de la prueba del tapping dinámico.

Se usaron dos clases para conseguir este objetivo: *DinamycTappingActivity.java* y *DynamicTappingTestActivity.java*

DynamicTappingActivity

Su diseño y funcionalidad están orientados a facilitar el acceso y la gestión de la prueba de *tapping* dinámico dentro de la aplicación. Se implementa un botón *startTestButton*, que al ser pulsado inicia la actividad *DynamicTappingTestActivity*. Esta acción está diseñada para llevar al usuario directamente a la pantalla donde se realizará la prueba de *tapping*.

DynamicTappingTestActivity

Estos son los pasos que se utilizaron para desarrollar esta funcionalidad:

Área de Juego: Se utilizó un *ViewGroup* como área principal donde los círculos aparecen y se desplazan. Se realizaron ajustes específicos en el área de juego y en el proceso de generación de círculos para estandarizar la prueba y garantizar que todos los usuarios estén sujetos a las mismas condiciones durante su evaluación.

El área de juego se ajustó a un formato cuadrado, independientemente de las dimensiones de la pantalla del dispositivo, como se puede ver en la Figura 21. Este diseño asegura que todas las bolas tengan el mismo recorrido potencial desde el punto de generación hasta el punto de final, facilitando una evaluación uniforme de la rapidez y precisión de respuesta de los usuarios.

Esta configuración es crucial para minimizar las variaciones debidas a la diferencia de las distancias de recorrido que puede haber, por ejemplo, si una bola saliese de la parte superior céntrica de la pantalla hasta la parte inferior céntrica haría mucho más recorrido que una bola que saliese de la parte derecha céntrica de la pantalla hasta la parte izquierda céntrica de la pantalla.

```
13 <FrameLayout
14     android:id="@+id/gameArea"
15     android:layout_width="410dp"
16     android:layout_height="410dp"
17     app:layout_constraintBottom_toBottomOf="parent"
18     app:layout_constraintEnd_toEndOf="parent"
19     app:layout_constraintStart_toStartOf="parent"
20     app:layout_constraintTop_toTopOf="@+id/toolbarDynamic" >
21
```

Figura 21. Layout del área de juego donde se observan las medidas

Generación de Bolas: Las bolas se crean dinámicamente con propiedades específicas de tamaño y color, utilizando el método `getDrawable()` para definir su apariencia. Las bolas aparecen en posiciones aleatorias alrededor de los bordes del área de juego, determinadas por una lista de bordes (`edgeList`) que se baraja para asegurar la aleatoriedad.

Como se puede ver en las Figuras 22 y 23, se implementó un sistema de generación de bolas donde siempre se crean exactamente 32 bolas en total, distribuidas equitativamente entre las esquinas y los centros de cada lado del cuadrado del área de juego. Cada uno de estos ocho puntos de generación (cuatro esquinas y cuatro centros de los lados del cuadrado) produce cuatro bolas durante la prueba. Esta distribución garantiza que cada sesión de prueba sea idéntica en términos de número y posición inicial de las bolas, proporcionando una base consistente para comparar el desempeño de diferentes usuarios bajo las mismas condiciones estandarizadas.

```
151 private void initializeEdgeList() {
152     for (int i = 0; i < 8; i++) { // 8 ejes
153         for (int j = 0; j < 4; j++) { // 4 bolas por eje
154             edgeList.add(i);
155         }
156     }
157     Collections.shuffle(edgeList);
158 }
```

Figura 22. Código de la función donde se inicializan las bolas que saldrán.

```

181 public void run() {
182     // Genera una posición aleatoria en el borde
183     int edge = edgeList.get((ballsGenerated - 1) % edgeList.size()); // 0: arriba, 1: arriba derecha
184     int startPositionX = 0;
185     int startPositionY = 0;
186     switch (edge) {
187         case 0: // arriba
188             startPositionX = (gameArea.getWidth() - size) / 2;
189             startPositionY = 0;
190             break;
191         case 1: // arriba derecha
192             startPositionX = gameArea.getWidth() - size;
193             startPositionY = 0;
194             break;
195         case 2: // arriba izquierda
196             startPositionX = 0;
197             startPositionY = 0;
198             break;
199         case 3: // izquierda
200             startPositionX = 0;
201             startPositionY = (gameArea.getHeight() - size) / 2;
202             break;
203         case 4: // derecha
204             startPositionX = gameArea.getWidth() - size;
205             startPositionY = (gameArea.getHeight() - size) / 2;
206             break;
207         case 5: // abajo
208             startPositionX = (gameArea.getWidth() - size) / 2;
209             startPositionY = gameArea.getHeight() - size;
210             break;
211         case 6: // abajo derecha
212             startPositionX = gameArea.getWidth() - size;
213             startPositionY = gameArea.getHeight() - size;
214             break;
215         case 7: // abajo izquierda
216             startPositionX = 0;
217             startPositionY = gameArea.getHeight() - size;
218             break;
219     }
220
221     ball.setX(startPositionX);
222     ball.setY(startPositionY + verticalOffset);

```

Figura 23. Código de la función de generación de círculos en los ejes

Animación de Círculos: Cada círculo se anima para moverse a lo largo de trayectorias predefinidas usando *ObjectAnimator*, ajustando su velocidad y dirección para aumentar la dificultad de la tarea. La interacción se maneja mediante eventos táctiles, donde cada toque en un círculo es registrado y evaluado.

Registro de Interacciones: Se registra el momento en que el usuario toca cada bola (*circleAppearanceTime*), y se calcula el Tiempo de Inicio de Movimiento (TMI) desde que la bola aparece hasta que es tocada. Esto se complementa con el registro de la velocidad de los toques y la cuenta de errores, que ocurren cuando un toque es registrado fuera de las bolas, como se muestra en la Figura 24.

```

160     private void addNewBall() {
161         final View ball = new View( context: this);
162         int size = calculateBallSize();
163         int screenWidth = gameArea.getWidth();
164         int screenHeight = screenWidth;
165         int verticalOffset = (gameArea.getHeight() - screenHeight) / 2;
166
167         isBallActive = true;
168
169         ball.setLayoutParams(new FrameLayout.LayoutParams(size, size));
170         ball.setBackground(getDrawable(R.drawable.circled));
171         // Registrar el momento de aparición del círculo
172         circleAppearanceTime = System.currentTimeMillis();
173         isFirstTap = true; // Restablecer para el nuevo círculo
174         gameArea.addView(ball);
175         balls.add(ball);
176         ballsGenerated++;

```

Figura 24. Código del método para añadir una nueva bola y registrar el TMI

Conteo Regresivo y Temporización de la Prueba: Antes de comenzar, se muestra un conteo regresivo de tres segundos para preparar al usuario. La prueba tiene una duración fija, y se utiliza un `CountDownTimer` para gestionar este tiempo, finalizando la prueba cuando todas las bolas han sido generadas y el tiempo ha expirado.

La elección de implementar un módulo de *tapping* dinámico se basa en la necesidad de proporcionar una herramienta que evalúe múltiples aspectos del desempeño neuromotor que son afectados por la enfermedad de Parkinson. Las características evaluadas incluyen:

Coordinación y agilidad: Fundamental para determinar cómo los pacientes manejan tareas que requieren precisión bajo presión.

Velocidad de reacción: Crucial para evaluar cómo los pacientes responden a estímulos cambiantes, un indicador común de deterioro en condiciones neurológicas.

Precisión: Importante para entender la capacidad del paciente para interactuar con objetivos específicos, reflejando la funcionalidad motora fina.

Esta metodología proporciona datos valiosos que pueden ayudar en el diagnóstico temprano y seguimiento de la enfermedad de Parkinson [18]. Como se puede ver en la Figura 25, en la pantalla del *tapping* dinámico se muestra el tiempo restante mientras los círculos van apareciendo por la pantalla en constante movimiento.

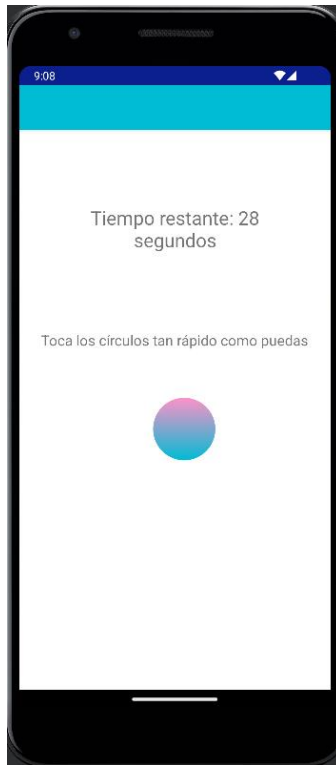


Figura 25. Pantalla de actividad del tapping dinámico

Los resultados obtenidos de la prueba de *tapping* dinámico son presentados en una interfaz clara e intuitiva para facilitar su interpretación tanto por parte de los pacientes como de los profesionales médicos. La pantalla de resultados, que se puede observar en la Figura 26, muestra las métricas clave de rendimiento que se han calculado durante la prueba.



Figura 26. Pantalla de los resultados de la prueba del tapping dinámico

Velocidad Media:

1. Cada vez que el usuario toca una bola, se registra el tiempo actual (`currentTimeMillis`).
2. Se calcula el intervalo de tiempo entre el toque actual y el último toque registrado (`lastTapTimeMillis`).
3. La velocidad de *tapping* individual se calcula como el inverso del intervalo de tiempo (es decir, 1 dividido por el intervalo de tiempo en segundos). Esto da una medida en "toques por segundo".
4. Cada velocidad individual se suma a un total acumulativo (`totalTapSpeed`), y se incrementa un contador de la cantidad de velocidades medidas (`tapSpeedCount`).
5. Al finalizar la prueba, se calcula la velocidad media de *tapping* dividiendo el total acumulativo de las velocidades de *tapping* (`totalTapSpeed`) por la cantidad de velocidades medidas (`tapSpeedCount`).

Tiempo medio de inicio (TMI):

1. Cuando aparece una bola, se registra el momento de su aparición (`circleAppearanceTime`).
2. Al detectar el primer toque en una bola, se registra ese momento (`tapTime`).
3. El TMI para ese toque se calcula restando el `circleAppearanceTime` del `tapTime`, resultando en el tiempo que tardó el usuario en reaccionar y tocar la bola desde que apareció.
4. Este TMI se agrega a una lista de todos los TMIs registrados durante la prueba.
5. Al finalizar la prueba, se calcula el tiempo medio de inicio sumando todos los TMIs de la lista y dividiendo por el número total de TMIs registrados. Esto da el tiempo medio de inicio en milisegundos, que luego se convierte a segundos para su presentación.

Ambos cálculos tienen como objetivo proporcionar una medida objetiva de la capacidad del usuario para responder a estímulos visuales en un tiempo limitado, que es especialmente relevante para la evaluación de trastornos motores como la enfermedad de Parkinson. Estas métricas permiten a los médicos evaluar el rendimiento motor y la velocidad de procesamiento, que pueden verse afectados en esta y otras condiciones neurológicas.

Al igual que en el *tapping* estático tenemos los mismos tres botones al finalizar la prueba, la única diferencia es que esta vez al pulsar el botón de *Descargar Fichero*, el fichero csv que se descargará tendrá el nombre en el formato *DynamicNombreApellidoAñoMesDiaHoraMinutosSegundos.csv*

4 Resultados

4.1 Síntesis de Resultados

Implementación del *Tapping*

La implementación de las pruebas de *tapping* en la aplicación diseñada se ha llevado a cabo con éxito. La prueba de *tapping* estático y dinámico se ha integrado de manera eficiente, permitiendo la recopilación de datos que reflejan la coordinación motora y la bradicinesia, síntomas comunes del Parkinson. La naturaleza interactiva de la aplicación asegura la participación del usuario y proporciona una forma no invasiva de monitorear los síntomas a lo largo del tiempo.

Métricas desarrolladas

Se han calculado varias métricas clave a través de la aplicación:

- Velocidad de *tapping*: La rapidez con la que los usuarios pueden tocar los círculos en la pantalla, que tiende a disminuir en los pacientes con Parkinson debido a la bradicinesia.
- Tiempo de Inicio de Movimiento (TMI): El tiempo que tarda un usuario en responder a un estímulo visual, lo cual es valioso para evaluar los reflejos y la agilidad motora.
- Número de Errores: Los toques realizados fuera de los círculos objetivo, que pueden indicar una disminución en la precisión motora.

4.2 Comparación con Estado del Arte

Innovaciones de la Aplicación

En comparación con aplicaciones existentes como Parkinson mPower 2 y iPrognosis, esta aplicación presenta innovaciones significativas, como algoritmos mejorados para la detección de errores y una interfaz más accesible y amigable para los usuarios. Además, la aplicación se ha optimizado para ofrecer una experiencia de usuario más personalizada y resultados más precisos.

Necesidades Atendidas y Mejoras Implementadas

Este trabajo ha abordado necesidades específicas previamente no resueltas en el estado del arte, incluyendo la necesidad de una mayor precisión en la medición de la bradicinesia y la falta de pruebas de *tapping* dinámico. Hemos incorporado estos elementos dentro de la aplicación para ofrecer una herramienta más completa y robusta para el diagnóstico y seguimiento del Parkinson.

4.3 Discusión de Métricas para Diagnóstico

Validez de las Métricas

La validez de las métricas utilizadas en la aplicación se basa en su capacidad para reflejar los síntomas motores característicos del Parkinson y su correlación con estudios clínicos existentes. La velocidad de *tapping* y el TMI se han demostrado como indicadores confiables de disfunción motora [19].

Diferenciación entre Paciente Sano y con Parkinson

Los umbrales y rangos típicos podrían establecerse de la siguiente manera:

Velocidad de *Tapping* Estático y Dinámico: La velocidad de *tapping*, que mide la capacidad de un individuo para realizar toques rápidos y precisos, se ha observado de forma diferenciada entre pacientes simulados sanos y aquellos con Parkinson. En pruebas con sujetos sanos, la velocidad media de la prueba del *tapping* estático fue de aproximadamente entre 6 y 8 toques por segundo como se puede observar en la Figura 27, con un rendimiento bastante estable y una menor fatiga con el tiempo. Por otro lado, en la simulación de pacientes con Parkinson, la velocidad media disminuyó notablemente a valores de entre 1 y 4 toques por segundo como se puede observar en la Figura 28, exhibiendo una mayor variabilidad y un incremento significativo en la fatiga durante la misma duración de prueba.



Figura 27. Prueba de *tapping* estático en paciente sano

Respecto a la velocidad en la prueba del *tapping* dinámico, los pacientes sanos presentaban una velocidad media mayor de 1 toque por segundo, como se puede observar en la Figura 29, al contrario que los pacientes simulando la enfermedad del parkinson que obtenían valores inferiores a 1 toque por segundo.



Figura 28. Prueba de tapping estático en paciente simulando el Parkinson.



Figura 29. Prueba tapping dinámico en paciente sano y en paciente simulando el Parkinson

Tiempo de Inicio de Movimiento (TMI): Para un sujeto sano, el TMI promedio podría estar en el rango de 400-700 milisegundos, mientras que, para un paciente con Parkinson, este tiempo podría incrementarse significativamente a 700-1000 milisegundos o más, lo cual fue observado en las pruebas donde los sujetos simulando Parkinson mostraron un TMI promedio de alrededor de 0.80 segundos.

Número de Errores: Los errores durante las pruebas de *tapping* pueden indicar una disminución en la precisión motora. Los sujetos sanos cometieron muy pocos errores, lo cual contrasta con los resultados de los pacientes simulados con Parkinson, quienes cometieron un número mayor de errores, reflejando una disminución en la coordinación motora. En la prueba estática, se podría considerar que un número de errores mayor de 20 con un número de aciertos poco elevado junto con las demás medidas mencionadas podría considerarse como un posible caso de Parkinson.

Desviación Estándar Media de la Velocidad de *Tapping*: La variabilidad en la velocidad de *tapping* revela diferencias entre pacientes sanos y aquellos simulados con Parkinson. Mientras que los individuos sanos exhiben una mayor consistencia en su velocidad de *tapping*, evidenciada por una desviación estándar más baja (con media más alta), los pacientes simulados con Parkinson muestran una desviación típica más alta (con una media más baja). Para comparar la dispersión relativa se utiliza el coeficiente de variación de Pearson. Este coeficiente, al ser una medida adimensional, permite comparar la variabilidad entre diferentes grupos. En el caso de los pacientes con Parkinson, el coeficiente de variación de Pearson es mayor debido a la combinación de una menor media y una mayor desviación estándar, en contraste con los individuos sanos que muestran una mayor media y una desviación estándar menor. Esta diferencia de variabilidad puede atribuirse a la bradicinesia y otras fluctuaciones motoras asociadas con la enfermedad.

Hay que considerar que la media de la velocidad de *tapping* y otros factores como el número de aciertos y errores puede proporcionar una visión más completa de la situación.

Fatiga a lo largo del tiempo: La prueba de *tapping* evaluó la fatiga mediante la observación de la disminución en la velocidad de *tapping* a lo largo de los 30 segundos de duración de la prueba. Los pacientes con Parkinson, simulados para este estudio, mostraron una disminución más pronunciada en la velocidad y un aumento en la desviación estándar con el tiempo en comparación con los individuos sanos, lo que refleja el incremento de la fatiga motora característica de la enfermedad.

Esta métrica se presenta visualmente a través de gráficos de líneas en la aplicación, donde la pendiente negativa de la línea es más acentuada en pacientes con Parkinson. Estos resultados son consistentes con la literatura que documenta la fatiga como un síntoma común y debilitante del Parkinson.

Estos resultados se basan en simulaciones y deberían ser validados con datos experimentales recolectados de sujetos reales. Además, estos rangos pueden variar significativamente entre individuos, por lo que es importante tener en cuenta la variabilidad individual al interpretar los resultados. También es crucial que estos umbrales se validen en un entorno clínico para asegurar su precisión y fiabilidad.

4.4 Impacto Clínico y Tecnológico

Mejora en el Diagnóstico

La app puede usarse para mejorar el proceso de diagnóstico y seguimiento de los pacientes con Parkinson, proporcionando una herramienta accesible para la evaluación continua de los síntomas y la respuesta al tratamiento.

Futuras Direcciones de Investigación

Sería importante contrastar los resultados con pacientes de Parkinson reales ya que actualmente para este TFG las pruebas realizadas han sido simulando los comportamientos de esta enfermedad. Además, es crucial explorar la integración de inteligencia artificial para mejorar la precisión del diagnóstico y proporcionar predicciones proactivas.

5 Conclusión

Este Trabajo de Fin de Grado ha desarrollado y evaluado dos funcionalidades de *tapping* en una aplicación para dispositivos móviles, específicamente diseñada para ayudar en el diagnóstico y seguimiento de la enfermedad de Parkinson.

A través de este proyecto, se ha demostrado que la tecnología móvil puede ser una plataforma eficaz para la implementación de herramientas de ayuda al diagnóstico que antes estaban confinadas a entornos clínicos. Sin embargo, es fundamental reconocer ciertas limitaciones del estudio, como la variabilidad en la interacción de los usuarios con la aplicación, que puede influir en la consistencia de los datos recogidos. Además, la validación rigurosa de las funcionalidades implementadas es crucial para garantizar su fiabilidad y eficacia en un contexto real.

Para futuras investigaciones y desarrollos, se sugiere explorar una personalización más profunda de las pruebas de *tapping*, adaptándolas a las necesidades específicas de los usuarios con diferentes grados y tipos de limitaciones motoras. También es esencial ampliar el alcance de la accesibilidad de la aplicación para asegurar que todos los pacientes, independientemente de sus capacidades motoras, puedan beneficiarse de ella.

En resumen, este trabajo subraya la potencialidad de las aplicaciones móviles como herramientas revolucionarias en el campo de la neurología, con la capacidad de ayudar en el diagnóstico y el seguimiento de enfermedades como la enfermedad de Parkinson. Así, se abre el camino para futuras innovaciones que puedan integrar aún más la tecnología digital en la salud neurológica y mejorar la calidad de vida de los pacientes.

6 Análisis de Impacto

En este capítulo, se evalúa el impacto potencial de los resultados obtenidos durante la realización del Trabajo de Fin de Grado (TFG) en diversos contextos, considerando tanto beneficios esperados como posibles efectos adversos. Se hace hincapié en la contribución de este proyecto a los Objetivos de Desarrollo Sostenible (ODS) de la Agenda 2030.

Contribución a los Objetivos de Desarrollo Sostenible

Este proyecto contribuye especialmente a varios objetivos de desarrollo sostenible:

ODS 3: Salud y Bienestar. Promueve el bienestar para todos al proporcionar herramientas que mejoran la calidad de vida de los pacientes con Parkinson [20].

ODS 9: Industria, Innovación e Infraestructura. Fomenta la innovación en el sector tecnológico médico [20].

ODS 11: Ciudades y Comunidades Sostenibles. Reduciendo la necesidad de desplazamientos físicos, contribuye a ciudades más sostenibles [20].

Decisiones Basadas en la Consideración de Impacto

A lo largo del desarrollo de este Trabajo de Fin de Grado, se tomaron varias decisiones claves que reflejan una consideración profunda del impacto potencial del proyecto en diferentes ámbitos:

- Estandarización de las Pruebas de Tapping: Se ajustaron cuidadosamente las pruebas de *tapping* para garantizar que fueran igualmente desafiantes y accesibles para todos los usuarios, independientemente de sus habilidades motoras iniciales. Esto no solo promueve la igualdad en la evaluación de los síntomas del Parkinson, sino que también asegura la consistencia en la recopilación de datos para una comparación precisa entre individuos, lo cual es crucial para el análisis clínico y la investigación.
- Diseño Inclusivo y Accesible: La interfaz de la aplicación se diseñó teniendo en cuenta la accesibilidad, con controles grandes y respuestas visuales claras para acomodar a usuarios con diversas capacidades motoras y visuales. Esto refleja un compromiso con la inclusión social y el respeto por la diversidad de los usuarios, alineándose con los principios de diseño universal.
- Reducción del Impacto Ambiental: Al optar por una solución digital que minimiza la necesidad de recursos físicos y visitas presenciales a centros médicos, se tomó una decisión consciente para reducir el impacto medioambiental del proyecto. Esta decisión también contribuye a la sostenibilidad de las ciudades, al disminuir los desplazamientos y, por ende, las emisiones de carbono asociadas.
- Consideración de Impacto Económico y Empresarial: Al desarrollar una herramienta que puede ser utilizada tanto en investigaciones clínicas

como en seguimientos individuales, se abrió la posibilidad de colaboraciones con entidades médicas y empresariales. Esto no solo amplía el impacto del proyecto en la comunidad médica, sino que también crea oportunidades de negocio y desarrollo profesional en el sector de tecnología de la salud.

- Mejora en la Calidad de la Asistencia Sanitaria: Una consideración fundamental durante el desarrollo de este Trabajo de Fin de Grado fue la mejora en la calidad de la asistencia sanitaria para los pacientes con Parkinson. Al proporcionar una herramienta digital para el seguimiento continuado de los síntomas motores, se abre la posibilidad de un monitoreo más frecuente y personalizado en comparación con las visitas anuales tradicionales al neurólogo. Esta evolución hacia un seguimiento más regular y detallado permite una detección temprana de cambios en los síntomas, lo que potencialmente mejora la calidad de vida de los pacientes al ofrecer una atención más proactiva y adaptada a sus necesidades individuales.

Estas decisiones no solo muestran un compromiso con la creación de una herramienta eficaz y eficiente, sino que también reflejan un enfoque ético y responsable hacia el impacto que el proyecto puede tener en la sociedad, la economía y el medio ambiente. Además, al alinear estas decisiones con los Objetivos de Desarrollo Sostenible, se subraya la relevancia global del proyecto y su contribución a metas internacionales de salud y bienestar.

Bibliografía

- [1] Parkinson's Foundation. Enfermedad de Parkinson. Parkinson's Foundation. Acceso el 13 de Marzo de 2024 <https://www.parkinson.org/library/fact-sheets/enfermedad-parkinson>
- [2] Tena, J. N. (2019, 4 abril). Una aplicación móvil para detectar señales tempranas de Parkinson. Euronews. Acceso el 15 de marzo de 2024. <https://es.euronews.com/2019/04/04/una-aplicacion-movil-para-detectar-senales-tempranas-de-parkinson>
- [3] Bot, B. M., Suver, C., Neto, E. C., Kellen, M., Klein, A., Bare, C., Doerr, M., Pratap, A., Wilbanks, J., Dorsey, E. R., Friend, S. H., & Trister, A. D. (2016). The mPower study Parkinson disease mobile data collected using ResearchKit. *Scientific Data*, 3, 160011. Acceso el 17 de marzo de 2024. <https://www.nature.com/articles/sdata201611>
- [4] García Dimas, F. J., Olivares Zepahua, B. A., Alor Hernández, G., Reyes Hernández, L. Á., & Cruz Ramos, N. A. (2022). Una revisión exploratoria de aplicaciones para dispositivos móviles para el tratamiento de la enfermedad de Parkinson. En Congreso Estudiantil de Inteligencia Artificial Aplicada a la Ingeniería y Tecnología (pp. 82-89). Universidad Nacional Autónoma de México, Facultad de Estudios Superiores Cuautitlán. Acceso el 17 de marzo de 2024. <https://virtual.cuautitlan.unam.mx/intar/ceiaait/wp-content/uploads/sites/14/2023/02/Int-Art-82-89.pdf>
- [5] Apkinson. (s.f.). Acceso el 24 de marzo de 2024. <https://www.movementdisorders.org/MDS/Application-Repository/Apkinson>
- [6] Tchelet, K., Stark-Inbar, A., & Yekutieli, Z. (2019). Pilot study of the encephalog smartphone application for gait analysis. *Sensors*, 19(23), 5179.
- [7] Zhan, A., Little, M. A., Harris, D. A., Abiola, S. O., Dorsey, E., Saria, S., & Terzis, A. (2016). High frequency remote monitoring of Parkinson's disease via smartphone: Platform overview and medication response detection. arXiv preprint arXiv:1601.00960. Acceso el 24 de marzo de 2024
- [8] Lipsmeier, F., Taylor, K. I., Postuma, R. B., Volkova-Volkmar, E., Kilchenmann, T., Mollenhauer, B., ... & Lindemann, M. (2022). Reliability and validity of the Roche PD Mobile Application for remote monitoring of early Parkinson's disease. *Scientific reports*, 12(1), 12081.
- [9] App Store. (2009, 22 febrero). ISeismometer. Acceso el 6 de abril de 2024, de <https://apps.apple.com/us/app/iseismometer/id304190739>
- [10] StudyMyHealth. User manual - StudyMyTremor. Acceso el 6 de abril de 2024, de <http://studymyhealth.com/user-manuals/user-manual-studymytremor/>
- [11] Lynx Design. (s. f.). Acceso el 07 de Abril de 2024 de <https://lift-pulse.soft112.com/>
- [12] Omberg, L., Neto, E. C., Perumal, T. M., Pratap, A., Tediario, A., Adams, J., Bloem, B. R., Bot, B. M., Elson, M., Goldman, S. M., Kellen, M. R., Kiebertz, K., Klein, A., Little, M. A., Schneider, R., Suver, C., Tarolli, C., Tanner, C. M., Trister, A. D., . . . Mangravite, L. M. (2021). Remote smartphone monitoring of Parkinson's disease and individual response to therapy. *Nature Biotechnology*, 40(4), 480-487. <https://doi.org/10.1038/s41587-021-00974-9>

- [13] XAMPP Installers and Downloads for Apache Friends. (s. f.). <https://www.apachefriends.org/index.html>
- [14] Oracle. MySQL. (s.f.). Acceso el 16 de abril de 2024 de <https://www.mysql.com/>
- [15] Contributors, P. (s. f.). PhpMyAdmin. phpMyAdmin. Acceso el 16 de abril de 2024 de <https://www.phpmyadmin.net/>
- [16] OpenAI. (2015-2024). DALL-E. Acceso el 20 de abril de 2024 de <https://openai.com/dall-e>
- [17] Jahoda, P. (2023). MPAndroidChart. GitHub. Acceso el 30 de abril de 2024 de <https://github.com/PhilJay/MPAndroidChart>
- [18] Lee, C. Y., Kang, S. J., Hong, S. K., Ma, H. I., Lee, U., & Kim, Y. J. (2016). A validation study of a smartphone-based finger tapping application for quantitative assessment of bradykinesia in Parkinson's disease. *PloS one*, 11(7), e0158852.
- [19] Goodman, A. O., Silva, A. H., Lazic, S. E., Morton, A. J., & Barker, R. A. (2008). Hand tapping: a simple, reproducible, objective marker of motor dysfunction in Huntington's disease. *Journal of Neurology*. SpringerLink. <https://link.springer.com/article/10.1007/s00415-008-0958-0>
- [20] Naciones Unidas. Objetivos de Desarrollo Sostenible. Acceso el 17 de mayo de 2024 de <https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/>
- [21] Page, A., Yung, N., Auinger, P., Venuto, C., Glidden, A., Macklin, E., ... & Dorsey, E. R. (2022). A smartphone application as an exploratory endpoint in a phase 3 Parkinson's disease clinical trial: A pilot study. *Digital Biomarkers*, 6(1), 1-8.

ANEXO I

MANUAL DE USUARIO

1. Introducción

Este manual acompaña a la aplicación diseñada para asistir en el diagnóstico temprano de la enfermedad de Parkinson (EP) a través de pruebas de *tapping*. Estas pruebas evalúan la bradicinesia y la coordinación motora, síntomas clave de la EP, mediante interacciones con la pantalla táctil simples pero efectivas.

2. Requisitos del Sistema

Para utilizar la aplicación de manera eficiente, los usuarios deben tener un dispositivo con las siguientes características:

- Sistema Operativo Android versión 10.0 o superior.
- Espacio mínimo de almacenamiento de 100 MB.

3. Pantalla de Inicio

Al iniciar la aplicación, los usuarios se encuentran con una interfaz clara que les permite acceder rápidamente al registro o al inicio de sesión. Los usuarios nuevos pueden elegir entre crear una cuenta de paciente o médico, proporcionando los detalles necesarios para cada tipo de registro.

4. Registro de Usuarios

4.1 Registro de Pacientes

Los pacientes necesitan proporcionar información personal básica como nombre, edad y detalles de contacto. Es necesario firmar el PDF de consentimiento informado.

4.2 Registro de Médicos

Los médicos deben ingresar su información profesional incluyendo el centro médico. Este registro permite el acceso a funciones adicionales como la monitorización de pacientes y el acceso a datos agregados para fines de investigación o seguimiento.

4. Funcionalidades de la Aplicación

La aplicación permite a los usuarios realizar pruebas de *tapping*, tanto estático como dinámico, que son evaluaciones interactivas donde los usuarios deben tocar círculos que aparecen en la pantalla en un intervalo y orden específicos. Los datos recogidos se analizan automáticamente para proporcionar métricas sobre la velocidad de *tapping* y la regularidad de los toques, ofreciendo una evaluación cuantitativa de la capacidad motora del usuario. Los resultados

ayudan a identificar patrones que pueden indicar la presencia de síntomas de la enfermedad de Parkinson.

5 Realización de Pruebas de *Tapping*

5.1 Prueba de *Tapping* Estático

La prueba de *tapping* estático evalúa la velocidad y precisión con la que un usuario puede tocar círculos que aparecen en posiciones fijas en la pantalla del dispositivo. Para iniciar esta prueba, el usuario debe seleccionar la opción "*Tapping* Estático" desde el menú principal. Una vez iniciada, la pantalla mostrará dos círculos en posiciones fijas que el usuario debe tocar lo más rápidamente posible durante 30 segundos. Cada toque es registrado y analizado para calcular la velocidad media de *tapping* y la precisión, ayudando a identificar posibles síntomas de bradicinesia asociados con la enfermedad de Parkinson.

5.2 Prueba de *Tapping* Dinámico

En la prueba de *tapping* dinámico, los círculos aparecen en diferentes ubicaciones de la pantalla en intervalos aleatorios. Esta prueba está diseñada para evaluar no solo la velocidad sino también la coordinación y la capacidad de reacción del usuario. Para comenzar, seleccione "*Tapping* Dinámico" en el menú. Durante la prueba, los círculos se generarán en posiciones aleatorias y el usuario debe tocar cada uno tan pronto como aparezca. Al igual que en la prueba estática, los datos recogidos se utilizan para analizar la velocidad de *tapping* y la regularidad de los toques.

Ambas pruebas proporcionan información valiosa sobre la funcionalidad motora del usuario y son relevantes para el diagnóstico y seguimiento de la enfermedad de Parkinson. Los resultados se muestran al final de cada prueba, ofreciendo un resumen detallado de la actuación del usuario, incluyendo la cantidad de toques realizados, la velocidad promedio de *tapping* y la cantidad de errores cometidos.

6 Opciones de la Aplicación

La aplicación también incluye opciones adicionales para personalizar la experiencia del usuario y mejorar la utilidad de las pruebas:

- Visualización de Resultados: Después de cada sesión de prueba, los resultados se visualizan en forma de gráficos y tablas, facilitando la comprensión y seguimiento de la evolución del usuario.
- Exportación de Datos: Para facilitar el seguimiento médico, los usuarios pueden exportar los datos de sus pruebas en csv para ser revisados por profesionales de la salud.

ANEXO II

MANUAL DEL PROGRAMADOR

1. Introducción

Este manual del programador se ha creado para facilitar la comprensión y el mantenimiento de la aplicación Android desarrollada para el diagnóstico temprano de la enfermedad de Parkinson. Este documento detalla la estructura y lógica detrás de las pruebas de *tapping*, tanto estáticas como dinámicas.

2. Descripción del Proyecto

El proyecto está estructurado en el entorno de Android Studio, utilizando Java como lenguaje principal de programación. La aplicación se compone de varias actividades que gestionan desde el registro de usuarios hasta la ejecución y análisis de pruebas de *tapping*. Todos los componentes y sus funciones están detalladamente descritos para facilitar las actualizaciones y la depuración del código.

3. Configuración del Entorno de Desarrollo

SDK Utilizado: Android 10.0 (Q).

IDE Recomendado: Android Studio.

Gradle: Utilizar la versión compatible con Android 10.0, típicamente gradle-6.1.1-all.zip.

4. Estructura del Proyecto

Paquetes Principales:

com.example.Parkinson_voice: Contiene todas las actividades y fragmentos utilizados.

Actividades Relevantes:

DynamicTappingTestActivity: Gestiona la prueba de *tapping* dinámico.

StaticTappingTestActivity: Gestiona la prueba de *tapping* estático.

Recursos:

Drawables: Incluye imágenes y xmls para los botones y fondos utilizados en las pruebas de *tapping*.

Layouts: Definiciones de la interfaz de usuario para cada actividad y fragmento.

Values: Contiene strings, colores y dimensiones utilizadas en la aplicación

5. Clases de *Tapping*

Descripción de la Clase *DynamicTappingTestActivity*

Librerías y Versiones Utilizadas:

- Librerías de Animación de Android: Se utiliza Animator, AnimatorSet, y ObjectAnimator para crear animaciones fluidas en los objetos dentro de la interfaz.
- Manejo de Permisos: Uso de ActivityCompat para solicitar permisos en tiempo de ejecución, esencial para la descarga de resultados y acceso a almacenamiento externo.

Componentes y Funciones Principales:

- 1) Inicialización y Configuración de UI:
 - Inicializa componentes de la interfaz de usuario como botones, TextViews, y ProgressBar.
 - Prepara el área de juego donde los objetos dinámicos (círculos) serán mostrados y animados.
- 2) Manejo de Eventos de Tocar en la Pantalla:
 - Detecta toques sobre los objetos dinámicos y maneja eventos cuando los toques ocurren fuera de estos objetos para contabilizar errores.
 - Registra el tiempo de respuesta del usuario desde la aparición de cada objeto hasta el toque.
- 3) Cronometraje y Control del Flujo de la prueba:
 - Utiliza CountdownTimer para gestionar el cronometraje de la actividad, desde la cuenta regresiva inicial hasta el tiempo total asignado para el test.
 - Controla la visibilidad y la interacción con elementos de UI basados en el estado del test.
- 4) Generación y Animación de Objetos:
 - Crea visualmente animaciones para los objetos (círculos) que el usuario debe tocar, utilizando una lista predefinida de posiciones de inicio basadas en el diseño del área de juego.
 - Implementa lógica para determinar la secuencia de aparición y movimiento de estos objetos en la pantalla.
- 5) Cálculo de Resultados y Estadísticas:
 - Calcula y muestra resultados como la velocidad media de *tapping*, el número de errores y el tiempo promedio de inicio de movimiento (TIM).
 - Proporciona funcionalidades para descargar estos resultados en formato CSV, permitiendo un análisis posterior fuera de la aplicación.

6) Notificaciones y Permisos:

- Gestiona las notificaciones para informar al usuario cuando los resultados están disponibles para visualización o han sido guardados exitosamente.
- Solicita y verifica permisos necesarios para el almacenamiento de archivos y otros recursos.

Descripción de la Clase StaticTappingTestActivity

Librerías y Versiones Utilizadas:

- MPAndroidChart: Para la visualización de datos, se usa la biblioteca MPAndroidChart, que facilita la creación de gráficos dinámicos y atractivos para representar la fatiga a lo largo del tiempo.
- Gestión de Permisos Android: Emplea clases de ActivityCompat y FileProvider para gestionar permisos y acceso a archivos, crucial para la funcionalidad de descarga de resultados.

Componentes y Funciones Principales:

1) Inicialización y Configuración de UI:

- Configura elementos de UI como Button, TextView, y LineChart, preparándolos para la interacción del usuario y la presentación de resultados.
- Establece visibilidad inicial y prepara el entorno para la prueba.

2) Gestión de Eventos de Tocar:

- Implementa listeners para botones que el usuario debe tocar alternadamente, midiendo la velocidad y precisión de respuesta.
- Captura errores cuando un botón es tocado consecutivamente o si se toca fuera de los botones objetivo.

3) Control de Tiempo y Eventos:

- Utiliza CountdownTimer para manejar el tiempo de inicio de la prueba y el tiempo total de duración, proporcionando feedback visual y temporal al usuario.

4) Registro y Análisis de Datos:

- Monitorea y registra cada toque, calculando estadísticas en tiempo real como velocidad máxima, velocidad media y desviación estándar.
- Utiliza un HashMap para acumular toques por segundo, facilitando el cálculo de la aceleración máxima y otros parámetros dinámicos.

5) Visualización de Resultados con Gráficos:

- Configura y muestra un gráfico de línea en tiempo real para visualizar la fatiga, utilizando la biblioteca MPAndroidChart para una representación detallada y comprensible.

6) Manejo de Resultados y Exportación de Datos:

- Proporciona opciones para descargar los resultados de la prueba en formato CSV, permitiendo un análisis más detallado fuera de la aplicación.

- Implementa notificaciones para informar al usuario sobre el estado de la descarga y el acceso al archivo.
- 7) Gestión de Permisos y Seguridad:
- Maneja permisos necesarios para escribir en el almacenamiento externo, asegurando que la aplicación cumpla con las normativas de seguridad y privacidad de Android.

Descripción de la Clase LoginActivity

Librerías y Versiones Utilizadas:

- Volley: Para las operaciones de red, se usa Volley para simplificar la comunicación con el servidor y el manejo de solicitudes HTTP.

Componentes y Funciones Principales:

- 1) Inicialización y Configuración de UI:
 - Inicializa y configura elementos de la interfaz de usuario como EditText para usuario y contraseña, Button para acciones de inicio de sesión y registro, y TextView para recuperación de contraseña.
 - Define RadioGroup y RadioButton para seleccionar el tipo de cuenta durante el proceso de login o recuperación de contraseña.
- 2) Gestión de Sesión:
 - Maneja el inicio de sesión verificando las credenciales con el servidor y direccionando al usuario a la pantalla correspondiente según su tipo de cuenta, ya sea médico o paciente.
 - Ofrece funcionalidad para recuperar contraseña mediante un diálogo y la interacción con un servicio web para el envío del correo electrónico de recuperación.
- 3) Interacción con el Servidor:
 - Implementa solicitudes POST a través de Volley para enviar y recibir datos del servidor, manejando respuestas en formato JSON para decidir el flujo de la aplicación.
Utiliza StringRequest para gestionar las solicitudes de inicio de sesión y recuperación de contraseña, manejando las respuestas del servidor de forma eficiente.
- 4) Navegación y Transiciones:
 - Controla la navegación entre diferentes actividades dependiendo del resultado del inicio de sesión o del proceso de registro, usando Intent para cambiar de pantalla.
- 5) Validación de Usuario:
 - Valida la entrada del usuario en los campos de texto y asegura que la selección del tipo de cuenta es correcta antes de enviar la solicitud al servidor.
 - Muestra mensajes de error relevantes si las entradas son incorrectas o si hay problemas con la comunicación del servidor.

6) Manejo de Errores y Notificaciones:

- Maneja errores de red y responde a fallos en la autenticación o en la recuperación de contraseña mediante mensajes Toast.
- Utiliza NotificationManager para generar notificaciones cuando los resultados de las acciones, como la recuperación de contraseña, están listos.

Análisis del Archivo build.gradle de la Aplicación

El archivo build.gradle es esencial para la configuración del proyecto Android, definiendo las versiones del SDK, las dependencias y otras configuraciones importantes para el entorno de desarrollo.

compileSdkVersion: Define la versión del API de Android contra la cual se compilará la aplicación. La actual, es la versión 29, lo que corresponde a Android 10.

defaultConfig:

- applicationId: Identificador único de la aplicación en el ecosistema Android, utilizado para identificar recursos de la aplicación.
- minSdkVersion y targetSdkVersion: Especifican la versión mínima de Android necesaria para ejecutar la app y la versión objetivo de Android para la cual la app está diseñada, respectivamente. Ambas están configuradas a la versión 29, asegurando compatibilidad con dispositivos que ejecutan Android 10 y superior.
- versionCode y versionName: Utilizados para gestionar las diferentes versiones de la app, donde versionCode es un número entero que debería incrementarse con cada versión de lanzamiento, y versionName es el nombre de la versión.

Tipos de Build

- buildTypes: Define cómo se construirán las versiones de lanzamiento de la aplicación.
 - minifyEnabled: Indica si el código será minimizado.
 - proguardFiles: Referencia a los archivos ProGuard que controlan la minimización del código. Esencial para proteger el código en la versión de lanzamiento.


Configuración de dependencias

Las dependencias son bibliotecas externas que la aplicación necesita para funcionar correctamente:

- implementation fileTree(dir: 'libs', include: ['*.jar']): Incluye automáticamente cualquier archivo .jar en el directorio libs como una dependencia.
- Varias implementaciones de androidx: Aseguran la compatibilidad y facilitan el diseño de la interfaz con componentes como appcompat y constraintlayout.
- com.google.android.material:material:1.0.0: Proporciona componentes de diseño Material Design para la interfaz de usuario.
- com.android.volley:volley:1.1.1: Utilizada para facilitar la comunicación de la red, como las solicitudes HTTP que se hacen desde la app.

- `com.tom_roush:pdfbox-android` y `com.github.barteksc:android-pdf-viewer`: Permiten la manipulación y visualización de archivos PDF dentro de la aplicación.
- `com.github.PhilJay:MPAndroidChart:v3.1.0`: Biblioteca para crear gráficos avanzados y visualizaciones de datos dentro de la app.

Este documento esta firmado por



| | |
|-------------------------------|---|
| Firmante | CN=tfgm.fi.upm.es, OU=CCFI, O=ETS Ingenieros Informaticos - UPM, C=ES |
| Fecha/Hora | Mon Jun 03 00:50:57 CEST 2024 |
| Emisor del Certificado | EMAILADDRESS=camanager@etsiinf.upm.es, CN=CA ETS Ingenieros Informaticos, O=ETS Ingenieros Informaticos - UPM, C=ES |
| Numero de Serie | 561 |
| Metodo | urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.sha1 (Adobe Signature) |