



Universitat
de les Illes Balears

TRABAJO DE FIN DE GRADO

EXERGAMES PARA REHABILITACIÓN CERVICAL

Mario Ventura Burgos

Grado en Ingeniería Informática

Escuela Politécnica Superior

Año Académico 2023-24

EXERGAMES PARA REHABILITACIÓN CERVICAL

Mario Ventura Burgos

Trabajo de Fin de Grado

Escuela Politécnica Superior

Universitat de les Illes Balears

Año Académico 2023-24

Palabras clave del trabajo:

Exergames, rehabilitación cervical, terapia, detección de rostro, aplicación Android, inteligencia artificial, procesamiento de imágenes, hilos de ejecución.

Trabajo tutelado por Maria Francesca Roig Maimó

Autoritz la Universitat a incloure aquest treball en el repositori institucional per consultar-lo en accés obert i difondre'l en línia, amb finalitats exclusivament acadèmiques i d'investigació

Autor/a		Tutor/a	
Sí	No	Sí	No
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Agradecimientos

En primer lugar, quiero agradecer a mi tutora, Xisca, por su apoyo, orientación y colaboración durante el proceso que ha supuesto este proyecto, y por confiar en mí para participar en él.

También quiero agradecer a mi familia, por su apoyo incondicional durante los estudios del grado, y a mis amigos y compañeros de la universidad, por ayudarme y acompañarme durante estos 4 años, haciéndome crecer como persona y estudiante.

Finalmente, quiero expresar mi agradecimiento hacia aquellos que han participado en las pruebas de usuario, haciendo posible la realización del presente proyecto.

Este trabajo de fin de grado es parte del proyecto de I+D+i TED2021-131358A-I00, financiado por MICIU/AEI/10.13039/501100011033 y por la “Unión Europea NextGenerationEU/PRTR”.

Índice de contenidos

Agradecimientos	V
Índice de figuras.....	XI
Índice de tablas.....	XIV
Acrónimos.....	XVI
Resumen.....	XVIII
1. INTRODUCCIÓN	1
1.1 Objetivos	2
1.2 Estructura de la memoria	2
2. CONTEXTO	4
2.1 Definición, evolución y contexto de los exergames.....	4
2.2 Clasificación de los exergames	5
2.3 Beneficios de los exergames en aplicaciones de rehabilitación y medicina	7
2.4 Aplicaciones de exergames en contextos de rehabilitación y medicina.....	8
2.5 Comparación con métodos tradicionales de ejercicio	9
2.6 Conclusiones de cara a la implementación de la aplicación	10
3. ANÁLISIS	11
3.1. Requisitos del sistema.....	11
3.1.1. Requisitos funcionales	12
3.1.2. Requisitos no funcionales	15
3.2. Metodología de desarrollo	17
3.2.1. Descripción de los sprints	18
3.3. Planificación del desarrollo.....	19
3.3.1. Planificación detallada con subtarear.....	20
3.3.2. Diagrama de Gantt	22
4. TECNOLOGÍAS Y DISEÑO	24
4.1 Estado del arte.....	24
4.1.1 Entornos de desarrollo	25
4.1.2 Tecnologías de detección de movimiento	27
4.1.3 Lenguajes de programación	29
4.2 Plataforma de desarrollo	30

4.2.1 Entorno de desarrollo (IDE).....	30
4.2.2 Otras herramientas de desarrollo.....	31
4.3 Diseño	34
5. IMPLEMENTACIÓN.....	35
5.1 Registro de usuarios.....	35
5.1.1 Información necesaria para la aplicación.....	35
5.1.2 Entorno social motivacional.....	36
5.1.3 Creación de una base de datos relacional.....	37
5.1.4 Creación de usuarios desde la aplicación.....	38
5.2 Autenticación de usuarios	39
5.3 Petición y concesión de permisos	40
5.3.1 Obtención de permisos necesarios para la aplicación	40
5.3.2 Gestión de la respuesta del usuario	41
5.4 Seguimiento facial en tiempo real.....	42
5.4.1 Conexión con el framework de Google	42
5.4.2 Primeras pruebas con ML Kit	42
5.4.3 Detección estática de coordenadas de puntos faciales clave.....	44
5.4.4 Detección dinámica de coordenadas de puntos faciales clave	47
5.4.5 Algoritmo para procesamiento de imágenes en tiempo real en búsqueda de coordenadas de puntos faciales.	49
5.4.6 Detección de variaciones significativas en coordenadas faciales	50
5.4.7 Interpretación de variaciones en coordenadas y conversión en controles útiles para exergames	52
5.5 Implementación de exergames o videojuegos serios	53
5.5.1 Lista de exergames.....	53
5.5.2 Implementación de exergames.....	54
5.5.3 Comunicación entre módulos de la aplicación.....	55
5.6 Instrucciones del exergame como ejercicio de rehabilitación.....	56
5.7 Registro del progreso de los usuarios.....	57
5.8 Notificaciones y recordatorios	59
6. PRUEBAS Y RESULTADOS	61
6.1 Pruebas con usuarios reales.....	61
6.2 Pruebas de rendimiento.....	65
6.3 Pruebas de experiencia de usuario	69

6.4 Resultados	70
7. TRABAJO FUTURO	71
8. CONCLUSIONES	72
BIBLIOGRAFÍA	74
ANEXOS	80
ANEXO A.1. Evolución y etapas en la creación de exergames desde sus inicios.....	80
ANEXO A.2. Modelo final de datos implementado en la base de datos de exergames	82
ANEXO A.3. Cálculo de velocidades en movimientos faciales en tiempo real y posibles aplicaciones en exergames	85
ANEXO A.4. Medición de la usabilidad de la aplicación con el Sistema de Escalas de Usabilidad (SUS)	88

Índice de figuras

Figura 1. Actividad física mediante exergames.	4
Figura 2. Visión global sobre la clasificación de exergames bajo diferentes criterios.....	5
Figura 3. Representación gráfica de un desarrollo en 5 sprints.....	17
Figura 4. Diagrama de Gantt que ilustra la cronología de implementación de las subtarefas en base a su prioridad.....	23
Figura 5. Algunas de las tecnologías, herramientas o metodologías consideradas.	25
Figura 6. Ejemplo de reconocimiento facial y de puntos clave del rostro.	29
Figura 7. Cambios en el repositorio ‘exergames_beta’.....	31
Figura 8. Comandos de Git ejecutados en el Powershell de Windows.	32
Figura 9. Interacción de distintas tecnologías y frameworks con la aplicación de exergaming. .	33
Figura 10. Diseño estructural de la aplicación de exergaming.....	34
Figura 11. Modelo conceptual de datos para la base de datos de la aplicación de exergames.....	37
Figura 12. Creación de usuarios en la aplicación y generación de contraseñas seguras.	38
Figura 13. Autenticación de usuarios para iniciar sesión en la aplicación.	39
Figura 14. interfaz de usuario de la aplicación app1.....	43
Figura 15. Pruebas y resultados de la clasificación de imágenes con la aplicación app1.	43
Figura 16 que muestra el diagrama de flujo seguido desde que se captura una imagen con el dispositivo hasta que se obtiene las coordenadas de la nariz en un rostro.	45
Figura 17. Representación de coordenadas de una nariz encontradas en un rostro.	46
Figura 18. Coordenadas de la nariz detectadas estáticamente en un rostro.	46
Figura 19. Dos puntos y su distancia en el eje X e Y sobre un mismo plano.	50
Figura 20. Pantalla principal de la aplicación con un slider de exergames y pantalla de detalle de un exergame.	53
Figura 21. Texturas usadas para el exergame del juego de la serpiente.....	55
Figura 22. Transcurso habitual de una partida al “Snake Game”.	55
Figura 23. Instrucciones de un exergame mostradas en la pantalla previa al juego.....	57
Figura 24. Gráfico que muestra la evolución de un usuario en un exergame a través de las puntuaciones que ha obtenido (caso hipotético)	58
Figura 25. Gráfico que ilustra un ejemplo del análisis de las medias de puntuaciones obtenidas por un usuario para monitorizar su progreso en el tratamiento.	59
Figura 26. Notificaciones y recordatorios enviados por la aplicación.	60

Figura 27. Pruebas con usuarios reales.	61
Figura 28. Puntuaciones obtenidas por los usuarios en las partidas jugadas propuestas para el experimento.....	62
Figura 29. Promedio de puntos de cada usuario en las diez partidas jugadas	63
Figura 30. Promedio de puntuaciones de todos los usuarios en cada partida.....	64
Figura 31. Número de análisis realizados por distintos dispositivos en 60 segundos.	66
Figura 32. Versión de Android de los dispositivos y análisis por segundo realizados.	68
Figura 33. Modelo de datos que ilustra las estructura real implementada en la base de datos tras el paso a relacional y normalización.	83
Figura 34. Distancia absoluta entre dos puntos.....	85
Figura 35. Juegos que podrían implementarse a modo de exergame que incluye la simulación de fuerzas.	87
Figura 36. Puntuación obtenida en el SUS para la aplicación de exergames.....	89

Índice de tablas

Tabla 1. Clasificación de exergames bajo distintos criterios	6
Tabla 2. Matriz de dependencias de los requisitos funcionales del sistema.....	15
Tabla 3. Requisitos funcionales y user stories correspondientes a estos.....	20
Tabla 4. Subtarefas identificadas para la implementación con su identificador y dependencias. .	21
Tabla 5. Ejemplo de evolución de un usuario a través de sus puntuaciones.	58
Tabla 6. Resultados obtenidos por usuarios reales en pruebas de usuario.	62
Tabla 7. Dispositivos usados para el experimento, especificaciones y número de análisis realizados en un lapso de 60 segundos.....	66

Acrónimos

IDE – Entorno de desarrollo (Integrated Development Environment).

IA – Inteligencia artificial (Artificial Intelligence).

ML – Aprendizaje automático (Machine Learning).

RF – Requisito funcional.

RNF – Requisito no funcional.

CC – Comienzo a comienzo.

FF – Fin a Fin.

CF – Comienzo a Fin.

FC – Fin a Comienzo.

UI – Interfaz de usuario (User Interface).

SQL – Structured Query Language. Lenguaje estructurado de consultas.

VR – Virtual Reality. Realidad virtual.

AR – Augmented Reality. Realidad aumentada.

GO – Graphic Overlay.

Resumen

¿Se pueden realizar tratamientos de rehabilitación cervical y mejorar la adherencia a éstos mediante el uso de videojuegos que implementen técnicas de inteligencia artificial?

Los trastornos musculoesqueléticos en la zona cervical tienen un creciente número de casos en la sociedad moderna, con una prevalencia cada vez mayor y un impacto económico significativo. Estos trastornos son cada vez más comunes debido a una mala higiene postural, relacionada comúnmente con exceso de trabajo o incluso con la postura cuando se usan dispositivos como *smartphones*. Pueden provocar dolor y limitación funcional, afectando negativamente la calidad de vida de los pacientes. Si bien el ejercicio terapéutico es una intervención comúnmente prescrita para estos casos, su efectividad a largo plazo puede ser limitada debido a la falta de adherencia por parte de los pacientes.

En respuesta a este desafío, este proyecto se enfoca en el desarrollo de una aplicación de exergaming para rehabilitación cervical para dispositivos Android, bajo la hipótesis de que el uso de esta aplicación (app) aumentará la motivación de los pacientes a realizar sus ejercicios terapéuticos prescritos (con el objetivo final de aumentar la adherencia al tratamiento). La app deberá poder contener una serie de juegos serios (exergames) que induzcan a los pacientes a realizar ejercicios físicos específicos de movimientos de cabeza y cuello, pretendiendo que estos imiten los ejercicios terapéuticos realizados en una consulta tradicional.

La implementación de la aplicación se ha llevado a cabo en Android Studio como IDE, usando Java como lenguaje principal y XML para el diseño de elementos gráficos. Además, se ha usado SQL y el sistema gestor de base de datos PostgreSQL para la creación y gestión de una base de datos alojada en un servidor y que se conecta a la aplicación. La conexión entre la aplicación y la base de datos se hace mediante llamadas explícitas a funciones SQL sin una API RESTful que actúe como intermediario. También se ha usado la librería independiente ML Kit, de Google, para poder implementar funcionalidades relacionadas con el aprendizaje automático orientadas a aplicaciones móviles sin necesidad de crear y entrenar una IA propia. En concreto, la librería ML Kit se ha usado para la detección en tiempo real de puntos faciales clave sobre las imágenes capturadas por la cámara frontal del dispositivo móvil.

Este trabajo no sólo se centra en el desarrollo de un exergame que permita la interacción con el dispositivo móvil mediante el movimiento de la cabeza del usuario, sino que se analiza su usabilidad con el objetivo final de desarrollar una aplicación usable que permita, en un futuro, realizar un ensayo clínico para analizar si se cumple la hipótesis de que el exergame desarrollado aumentará la motivación de los pacientes a realizar los ejercicios terapéuticos prescritos (y, en consecuencia, su adherencia al tratamiento).

1. INTRODUCCIÓN

Este trabajo de fin de grado se engloba dentro del proyecto de I+D+i TED2021-131358A-I00: “Ejercicio terapéutico mediante juego serio en dispositivos móviles para la rehabilitación cervical (RehbeCa)” de la convocatoria de 2021 Proyectos Estratégicos Orientados a la “Transición Ecológica y a la Transición Digital”. Este proyecto cuenta con un equipo multidisciplinar compuesto por personal del área de tecnología y del área de salud.

En la presente era digital, la integración de la tecnología en el ámbito de la salud ha emergido como una solución prometedora ante diversas problemáticas médicas. En particular, los trastornos musculoesqueléticos en la zona cervical representan una creciente preocupación en de la sociedad moderna [1], dada su alta prevalencia y el significativo impacto en la calidad de vida de quienes sufren de estos trastornos [2]. Tradicionalmente, el ejercicio terapéutico ha sido la intervención predilecta para el manejo de estas afecciones; sin embargo, la efectividad de dicha práctica se ve frecuentemente comprometida por la baja adherencia de los pacientes a los programas de rehabilitación [3], a menudo por lo aburridos y poco rutinarios que resultan.

Motivado por la necesidad de encontrar alternativas que fomenten la participación del paciente en su proceso de recuperación, este trabajo se centra en el desarrollo de una aplicación de exergaming para la rehabilitación cervical, diseñada para dispositivos móviles, de modo que pueda instalarse en el dispositivo del propio paciente y éste pueda usarlo en cualquier momento y lugar. Un exergame es aquel videojuego que implica un esfuerzo físico, por lo que el objetivo final es la implementación de un videojuego que induzca al paciente a realizar los ejercicios terapéuticos de rehabilitación cervical intentando, al mismo tiempo, aumentar la motivación del paciente. La propuesta se basa en la hipótesis de que la gamificación de los ejercicios de rehabilitación mediante una aplicación móvil puede incrementar significativamente la motivación del paciente, fomentando así una mayor adherencia al tratamiento. Por lo que se pretende desarrollar una aplicación que incorpore al menos un juego serio (exergame) con el que se interactúe mediante movimiento de la cabeza del usuario, simulando los ejercicios terapéuticos prescritos por los profesionales fisioterapeutas.

La innovación de este proyecto radica en su capacidad para detectar en tiempo real las coordenadas de puntos faciales clave (en este caso, la nariz), utilizando dicha información para interpretar los movimientos realizados por la cabeza del usuario y traducirlos en acciones interactivas sobre el dispositivo móvil. De esta manera, se establecen los controles para interactuar con los exergames propuestos. Además, con el objetivo de poder ser utilizado de forma cómoda por el usuario en cualquier momento y lugar, no debe requerir ningún sensor adicional a los integrados en el dispositivo móvil ni tampoco la colocación de ningún marcador sobre el usuario.

El objetivo principal de este trabajo es desarrollar un exergame **usable** para dispositivos móviles que induzca al usuario a realizar movimientos con la cabeza que simulen los ejercicios terapéuticos de rehabilitación cervical. El cumplimiento de este objetivo permitirá contar con una aplicación

que permita realizar el ensayo clínico planificado dentro del proyecto TED2021-131358A-I00 para analizar si es posible mejorar la adherencia a los programas de rehabilitación cervical mediante la implementación de soluciones tecnológicas innovadoras que combinen el ejercicio físico con elementos lúdicos. Con esta meta final en mente, se han establecido objetivos específicos relacionados con el diseño, desarrollo e implementación de la aplicación móvil, así como la evaluación de su impacto en la motivación de los pacientes.

1.1 Objetivos

Los objetivos de este trabajo de fin de grado, por tanto, son los siguientes:

1. Desarrollar una aplicación móvil que permita jugar, mediante movimientos de la cabeza, a videojuegos básicos en dos dimensiones; bajo la hipótesis de que estos videojuegos podrían usarse para inducir al paciente a realizar los ejercicios terapéuticos tradicionales prescritos para la rehabilitación cervical. La validación de la adecuación de los movimientos cervicales inducidos por la aplicación será responsabilidad del personal del área de salud que forma parte del proyecto TED2021-131358A-I00 y, por lo tanto, se encuentra fuera del ámbito de este trabajo de fin de grado.
2. La aplicación móvil desarrollada debe permitir la detección del movimiento de la cabeza del usuario sin usar ningún sensor adicional a los sensores integrados en el propio dispositivo móvil ni debe requerir la colocación de ningún marcador sobre el usuario.
3. La aplicación móvil usará la biblioteca de ML Kit para la detección de rostros con el objetivo de evaluar su rendimiento en la detección de características faciales.
4. La aplicación móvil desarrollada debe ser usable.

1.2 Estructura de la memoria

La memoria para este proyecto se ha desarrollado siguiendo la estructura descrita a continuación:

- **Capítulo 2: Contexto.** Se proporciona una visión general del contexto y la situación en la que se enmarcan los exergames, abordando las circunstancias, antecedentes y factores más relevantes. También se los compara con métodos tradicionales de ejercicio.
- **Capítulo 3: Análisis.** Se muestra el análisis, donde se recogen todos los requisitos funcionales y no funcionales, se descomponen en tareas, y se ilustra su planificación con un diagrama de Gantt.
- **Capítulo 4: Tecnologías y diseño.** Se presentan las tecnologías, métodos y plataformas apropiadas para la implementación de la aplicación, así como el diseño de esta. Este capítulo se centra en exponer las tecnologías utilizadas, mencionar alternativas, justificar la elección de estas e ilustrar el diseño de la arquitectura del sistema.
- **Capítulo 5: Implementación.** Se expone la implementación de la aplicación, explicando cómo se llevó a cabo su desarrollo.
- **Capítulo 6: Pruebas y resultados.** Se presentan las pruebas de usuario, pruebas de rendimiento, pruebas de usabilidad, y se analizan los resultados obtenidos en éstas.

- **Capítulo 7: Trabajo futuro.** Se presentan brevemente propuestas de valor y se discuten posibles mejoras y proyectos complementarios que podrían enriquecer la aplicación de exergaming.
- **Capítulo 8: Conclusiones.** Se exponen las conclusiones obtenidas tras la realización del proyecto.
- **Anexos:** Se aportan materiales complementarios que apoyan la información presentada en los capítulos anteriores.

2. CONTEXTO

En este capítulo se presenta un análisis del contexto en el que se enmarcan los exergames, y se menciona el estado actual de estos. Además, se explora el impacto de las tecnologías emergentes en el tratamiento de problemas cervicales, especialmente a través del uso de exergames, con el objetivo de comprender la definición y evolución de exergames, los distintos tipos de estos y su posible impacto.

2.1 Definición, evolución y contexto de los exergames.

En este apartado se expone la reciente historia y evolución de los exergames, así como los principales hitos en el desarrollo de estos videojuegos.

Como se ha mencionado anteriormente, se define un exergame como un videojuego activo o de ejercicio, es decir, un **videojuego interactivo que requiere y promueve la actividad física**, intentado convertirla en algo gratificante. Los exergames enfocan de forma tecnológica el ejercicio, centrándose en objetivos concretos como el fitness, la diversión, o en este caso, la rehabilitación cervical [4].

Son muchas las personas que consideran que el ejercicio físico, en especial los ejercicios terapéuticos, son monótonos y aburridos. Esto provoca que una parte significativa de los pacientes que requieren de estos ejercicios para llevar a cabo su rehabilitación acaben abandonándolos. Además de monotonía o aburrimiento, es frecuente la falta de motivación, que provoca una reducción de la adherencia a estos tratamientos físicos. A largo plazo, muchos pacientes acaban abandonando por completo los ejercicios pautados para su rehabilitación.

Tal y como se muestra en la [figura 1](#), los exergames plantean una posible solución, ya que permiten amenizar y hacer más divertida la Actividad Física (AF) gracias a los videojuegos.



Figura 1. Actividad física mediante exergames [5].

En la actualidad, los exergames han avanzado mucho, incluso hasta el punto de permitir a los usuarios sumergirse totalmente en un mundo virtual donde jugar. La evolución de los exergames se remonta desde los años 80 hasta la actualidad [6] [7], y puede dividirse en distintas etapas. Los exergames son, hoy en día, una parte integral de la industria del deporte, con un gran impacto en la salud digital. La creciente innovación en campos como la inteligencia artificial han propiciado el nacimiento de una creciente variedad de juegos y aplicaciones diseñadas para distintos objetivos, entornos, niveles de intensidad, etc.

Para profundizar más y conocer con detalle la evolución de los exergames desde sus inicios en los años 80, se puede consultar el [anexo A.1](#).

2.2 Clasificación de los exergames

Una vez establecida la definición de un exergame, es evidente que existen distintos tipos de estos. La separación de exergames en distintos tipos, depende de bajo que criterio se quiera hacer esta separación: exergames en interiores y exergames en exteriores, fines recreativos y fines médicos o terapéuticos, etc. La [figura 2](#) ilustra un ejemplo básico de cómo se pueden clasificar los exergames bajo distintos criterios.

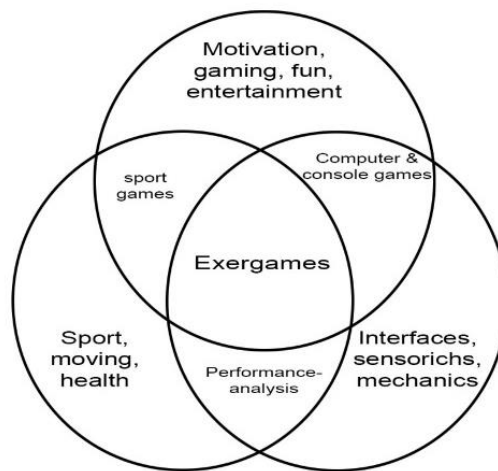


Figura 2. Visión global sobre la clasificación de exergames bajo diferentes criterios.

A continuación, la [tabla 1](#) muestra una clasificación completa a grandes rasgos, que permite comprender el alcance de los exergames y su clasificación en base a diferentes criterios [8].

Tabla 1. Clasificación de exergames bajo distintos criterios

Criterio	Tipo	Ejemplos reales	Descripción
Naturaleza del movimiento	Juegos de movimiento completo	Just Dance, Kinect Sports	Juegos que requieren movimientos corporales completos y que normalmente hacen uso de sensores de movimiento o cámaras.
	Juegos de movimiento parcial	Wii Sports, BoxVR	Juegos que requieren movimientos específicos con partes concretas del cuerpo.
	Juegos basados en gestos	Fruit Ninja Kinect	Juegos que requieren gestos simples y específicos para ser jugados.
Objetivo del juego	Juegos de fitness	Wii Fit	Juegos diseñados para mejorar la condición física de los usuarios.
	Juegos de deportes	Wii Sports, Kinect Sports	Juegos diseñados para simular movimientos de algún deporte como tenis, fútbol, etc.
	Juegos de baile	Dance Revolution, Just Dance	Juegos enfocados en hacer movimientos corporales para seguir coreografías.
Entorno	Juegos de exterior (freegaming)	Pokemon GO, Run!	Juegos que requieren ser jugados en zonas exteriores como calles, parques u otras.
	Juegos de interior	Beat Saber	Juegos diseñados para ser jugados en espacios interiores, en ambientes tranquilos y controlados.
Tecnología utilizada	Uso de controladores de movimiento	PlayStation Move, Wii Sports	Juegos que usan controladores para detectar el movimiento del usuario.
	Uso de cámaras o sensores	Shape Up, Kinect Sports	Juegos que usan cámaras y sensores para detectar el movimiento del usuario.
	Juegos de realidad virtual	BoxVR	Juegos que utilizan cascos de realidad virtual y controladores manuales para una experiencia inmersiva.
Nivel de intensidad	Intensidad baja	Wii Fit Yoga	Juegos que requieren una intensidad baja.
	Intensidad media	Just Dane, Ring Fit Adventure	Juegos que requieren una intensidad media.
	Intensidad Alta	Beat Saber, BoxVR	Juegos que requieren una intensidad alta.

Esta clasificación permite entender más fácilmente que, en función de factores como el objetivo del exergame, la naturaleza del movimiento que requiere, o el nivel de intensidad requerida para los movimientos; los exergames suponen una herramienta útil con la que fomentar la actividad física a través de los videojuegos, con objetivos tan dispares como, en este caso, realizar ejercicios de rehabilitación cervical a la vez que los pacientes se divierten jugando.

2.3 Beneficios de los exergames en aplicaciones de rehabilitación y medicina

Los exergames, combinando el entretenimiento de los videojuegos con el ejercicio físico, han demostrado ser una herramienta efectiva en la promoción de la actividad física, demostrando que pueden ser de gran utilidad especialmente en entornos de rehabilitación y medicina. Los beneficios físicos que estos juegos ofrecen son numerosos. A través de la interacción activa, los exergames pueden mejorar la fuerza muscular, la coordinación, la flexibilidad y la resistencia cardiovascular. Esto es particularmente relevante en pacientes que enfrentan desafíos en su movilidad, donde el ejercicio tradicional podría resultar monótono o desalentador.

Además de los beneficios físicos, los exergames también ofrecen ventajas psicológicas significativas, ya que se sabe que esta es otra de las ventajas de la actividad física. Uno de los principales beneficios es el aumento de la motivación y la adherencia al ejercicio. Al transformar el ejercicio en una actividad lúdica e interactiva, los exergames reducen la percepción de esfuerzo y aumentan el disfrute, lo que lleva a una mayor participación regular. La retroalimentación instantánea y el sentido de logro que proporcionan estos juegos también contribuyen a una mayor motivación, reduciendo el estrés y mejorando el bienestar emocional. En el contexto de la rehabilitación, esto es crucial para mantener a los pacientes comprometidos con su tratamiento a largo plazo.

En el caso de este proyecto, éste nace bajo la idea de esta mejora de adherencia a tratamientos de rehabilitación cervical gracias a los exergames, y su impacto positivo en los pacientes. En este contexto de uso, se pueden deducir, entre otros, los siguientes beneficios específicos:

- **Aumento de la motivación y adherencia durante procesos de rehabilitación:** El carácter interactivo y lúdico de los exergames hace que el proceso de rehabilitación sea más ameno, diferenciándose de los métodos tradicionales.
- **Reducción de dolores o rigidez y mejoras posturales:** Los exergames diseñados específicamente para tratar problemas cervicales pueden corregir malas posturas, lo que ayuda a reducir o eliminar el dolor y la rigidez muscular.
- **Personalización del tratamiento:** Se puede evaluar el desempeño de los usuarios en los exergames para hacer un seguimiento de su proceso de rehabilitación y ajustarlo a sus necesidades específicas.
- **Seguimiento y mejora continua del tratamiento:** El uso de tecnologías avanzadas en exergames permite una retroalimentación inmediata sobre la ejecución de los ejercicios.

2.4 Aplicaciones de exergames en contextos de rehabilitación y medicina

Los exergames, que combinan elementos de juego con actividad física, han ganado popularidad en el ámbito de la rehabilitación y la medicina debido a su capacidad para motivar a los pacientes y mejorar la adherencia a los programas de tratamiento. Estas aplicaciones no solo se enfocan en el entretenimiento, sino que también tienen el potencial de proporcionar beneficios terapéuticos significativos. En este apartado, se explorarán diversas aplicaciones de los exergames en la rehabilitación física y en otros contextos médicos, así como su impacto en la calidad de vida de los pacientes.

- **Rehabilitación física.** Los exergames han sido ampliamente usados en programas de rehabilitación física donde se utilizan para mejorar la movilidad, la fuerza y la coordinación en pacientes que se recuperan de lesiones, cirugías o enfermedades. Un área en la que los exergames han mostrado particular eficacia es la rehabilitación tras un accidente cerebrovascular (ACV) [\[9\]](#). En estos casos, los pacientes que sufren secuelas como la hemiparesia (debilidad de un lado del cuerpo) pueden beneficiarse de ejercicios específicos diseñados en forma de juegos, que los motivan a practicar movimientos repetitivos necesarios para la recuperación. Además, en la rehabilitación de lesiones ortopédicas, como las fracturas de extremidades, los exergames pueden ser utilizados para fomentar la movilidad articular y la fuerza muscular en un entorno controlado y seguro.
- **Rehabilitación neurológica.** En pacientes con enfermedades neurodegenerativas como el Parkinson, los exergames pueden ayudar a mejorar el equilibrio, la postura y la marcha. Por ejemplo, juegos diseñados para simular caminatas en diferentes terrenos pueden ser utilizados para practicar la estabilidad y la coordinación, elementos clave en la rehabilitación de estos pacientes [\[10\]](#).
- **Salud mental.** Como se ha mencionado anteriormente, los beneficios de los exergames no son solamente físicos, sino que también incluyen beneficios psicológicos. Algunos exergames han sido utilizados en el tratamiento de problemas de salud mental [\[11\]](#), tratando de mejorar distintas variables psicosociológicas como la depresión, ansiedad o impactos en la autoestima.
- **Rehabilitación pediátrica.** Los exergames son particularmente efectivos en la rehabilitación pediátrica [\[12\]](#), donde la motivación y el compromiso de los niños pueden ser un desafío. Los juegos que incorporan actividad física permiten a los niños participar en su rehabilitación de una manera divertida y atractiva, lo que ayuda a mantener su interés y adherencia al tratamiento. Por otro lado, esto resulta especialmente útil en niños con problemas como, por ejemplo, parálisis cerebral, ya que participar en juegos tradicionales con otros niños sin estas condiciones pueden resultar complicado. Los exergames suponen una alternativa que proporciona a los niños la posibilidad de disfrutar jugando a la vez que realizan su terapia.

2.5 Comparación con métodos tradicionales de ejercicio

En este apartado se realizará una comparación de la eficacia y el compromiso de los pacientes con su tratamiento cuando este incluye el uso de exergames en contraposición con el uso de métodos tradicionales de ejercicio. Además, se presenta una comparación detallada entre los exergames y los métodos tradicionales de ejercicio, considerando tanto los resultados en términos de salud como el feedback de los usuarios en diversos estudios.

Comenzando por la eficacia de los exergames en comparación con los métodos tradicionales de ejercicio, esta ha sido objeto de numerosos estudios. En algunos de estos estudios [7], se ha demostrado que los exergames pueden ser tan eficaces como los métodos tradicionales para mejorar la condición física general, la coordinación motora y la fuerza muscular, especialmente en contextos de rehabilitación y prevención.

Respecto a la intensidad o calidad de la actividad física en los exergames respecto a los métodos tradicionales, existen estudios [13] que comparan el gasto energético y la exigencia física de los exergames con el ejercicio físico tradicional, destacando que, pese a que no siempre es igual el nivel de exigencia, los exergames demuestran ser una alternativa efectiva en poblaciones especialmente sedentarias. Por ejemplo, se ha observado que los exergames son particularmente efectivos en poblaciones que pueden no sentirse atraídas por los métodos de ejercicio tradicionales, como niños, personas mayores, o pacientes en rehabilitación. En estos casos, la interactividad y el componente lúdico de los exergames ayudan a mantener el interés y la participación a lo largo del tiempo, lo que puede traducirse en mejoras significativas en los resultados de salud.

Respecto a la adherencia y compromiso al tratamiento, uno de los principales desafíos de los métodos tradicionales de ejercicio es mantener la motivación a largo plazo. La monotonía y aburrimiento que causan en los pacientes las rutinas repetitivas pueden llevar al abandono del programa de ejercicio. Sin embargo, existen estudios [14] que muestran que los exergames ofrecen una experiencia diferente, dinámica y variada, que suele ser percibida como más entretenida y menos ardua, lo que puede conducir a una mayor adherencia.

Además, los exergames permiten una personalización y adaptación en tiempo real, ajustando el nivel de dificultad según el rendimiento del usuario. Esto no solo evita el aburrimiento, sino que también asegura que el usuario esté constantemente desafiado a un nivel apropiado para su capacidad actual. En estudios de usuarios, los participantes a menudo reportan una mayor satisfacción y un mayor deseo de continuar con su programa de ejercicios cuando utilizan exergames en lugar de métodos tradicionales. Por otro lado, este fomento de la competitividad en los videojuegos también puede traer como consecuencia un aumento de la motivación y compromiso de los pacientes con su tratamiento.

2.6 Conclusiones de cara a la implementación de la aplicación

A partir del análisis detallado del contexto en el que se enmarcan los exergames en entornos de rehabilitación y medicina, se pueden extraer varias conclusiones clave que guiarán la implementación de la aplicación desarrollada en este proyecto.

En primer lugar, la principal ventaja de los exergames frente a los métodos tradicionales de ejercicio radica en su capacidad para aumentar la motivación y la adherencia al tratamiento. Por lo tanto, la implementación de la aplicación deberá centrarse en maximizar este aspecto, asegurando que los ejercicios terapéuticos sean atractivos e interactivos. Esto se logrará mediante la gamificación de los ejercicios, utilizando sistemas de puntos y niveles, así como a través de la creación de un entorno social que fomente la motivación de los usuarios.

Además de los beneficios físicos, los exergames ofrecen significativas ventajas psicológicas, como la reducción del estrés y la mejora del bienestar emocional. Por ello, la aplicación debe generar un entorno que no solo fomente la actividad física, sino que también promueva un ambiente lúdico y agradable, donde los usuarios se sientan acompañados en su proceso de rehabilitación y conectados con otros en situaciones similares. Los sistemas de puntos y niveles de experiencia se utilizarán para incrementar la motivación de los usuarios, incentivando su participación continua y proporcionando una sensación de logro y progreso en su tratamiento.

Finalmente, la aplicación debe incluir herramientas que permitan el seguimiento y la evaluación continua del progreso de los usuarios, facilitando la recopilación de datos que puedan ser analizados para ajustar y mejorar continuamente la experiencia del usuario y la efectividad del tratamiento. Los puntos obtenidos por los usuarios no solo actuarán como elementos de gamificación, sino que también permitirán un seguimiento efectivo del tratamiento al evaluar el desempeño en los juegos.

3. ANÁLISIS

El capítulo de análisis describe las necesidades o requisitos de la aplicación para que funcione de forma eficaz. Para ello, se clasifican estas necesidades en dos categorías.

1. **Requisitos funcionales (RF):** Aquellos requisitos que describen qué debe hacer el sistema, es decir, sus funcionalidades y servicios que ofrece.
2. **Requisitos no funcionales (RNF):** Aquellos requisitos que describen cómo debe comportarse el sistema.

El objetivo es proporcionar una base sólida para el desarrollo del proyecto, que posteriormente se detallará para proporcionar la metodología de desarrollo del proyecto. Finalmente, se mostrará la planificación completa de la implementación con un diagrama de Gantt [\[15\]](#).

3.1. Requisitos del sistema

Como ya se ha mencionado anteriormente, los RF especifican cuáles son las funcionalidades que debe tener la aplicación. Estos requisitos son de vital importancia para asegurar que la aplicación implementada cumpla con las expectativas de los usuarios finales que la usarán. Permitirán cuantificar el grado de cumplimiento con las necesidades detectadas previas al desarrollo y, por tanto, si la aplicación es funcional o no.

Por otro lado, los RNF son aquellos requisitos que describen cuál debe ser el comportamiento esperado del sistema (en este caso, la aplicación de exergaming). Estos requisitos hacen referencia a aspectos como la usabilidad, rendimiento, seguridad, escalabilidad, etc.

Los RF se centran en responder a “qué debe poder hacer la aplicación”, mientras que los RNF se centran en responder a “cómo debe comportarse la aplicación”. Ambos tipos de requisitos son cruciales para el desarrollo de un sistema completo y funcional.

Los requisitos funcionales y no funcionales de la aplicación se detallarán a modo de tabla para que se entiendan mejor. Cada tabla cuenta con un código/ID para el requisito, un nombre, una descripción, un objetivo, y una lista de dependencias. La lista de dependencias hace referencia a si un requisito depende directa o indirectamente de otro requisito. Esta lista es de una relevancia capital ya que ayuda a establecer el orden de las tareas de desarrollo.

3.1.1. Requisitos funcionales

Los RF identificados para la aplicación son los descritos a continuación.

3.1.1.1. Requisitos

ID	RF1
Nombre	Registro de usuarios
Descripción	La aplicación debe permitir a los usuarios registrarse y crear un perfil con sus datos personales.
Objetivo	Permitir a los usuarios tener un perfil personalizado con información relevante para su rehabilitación.
Dependencias	-

ID	RF2
Nombre	Autenticación de usuarios
Descripción	La aplicación debe autenticar a los usuarios antes de permitirles el acceso a sus perfiles y ejercicios.
Objetivo	Garantizar que solo usuarios registrados puedan acceder a sus datos y ejercicios personalizados.
Dependencias	RF1

ID	RF3
Nombre	Concesión de Permisos
Descripción	La aplicación debe solicitar y gestionar los permisos necesarios del dispositivo (acceso a la cámara, almacenamiento y red) para su correcto funcionamiento.
Objetivo	Garantizar que la aplicación tenga los permisos necesarios para acceder a los recursos del dispositivo y ejecutar sus funcionalidades correctamente.
Dependencias	-

ID	RF4
Nombre	Seguimiento facial en tiempo real
Descripción	La aplicación debe utilizar la cámara frontal del dispositivo para monitorizar y hacer un seguimiento en tiempo real de los movimientos del rostro del usuario.
Objetivo	Evaluar en tiempo real los movimientos de los usuarios.
Dependencias	RF3

ID	RF5
Nombre	Lista de exergames o videojuegos serios
Descripción	La aplicación debe poder contener una lista de exergames específicos para la rehabilitación cervical, de los cuales al menos uno de ellos debe ser jugable.
Objetivo	Ofrecer a los usuarios diferentes juegos terapéuticos para realizar sus ejercicios de rehabilitación de forma interactiva y amena.
Dependencias	RF4

ID	RF6
Nombre	Instrucciones del ejercicio/juego
Descripción	Dado que cada exergame es en realidad un ejercicio, la aplicación debe guiar a los usuarios en la realización de estos ejercicios mediante instrucciones visuales y auditivas.
Objetivo	Asegurar que los usuarios realicen los ejercicios correctamente siguiendo las instrucciones.
Dependencias	RF5

ID	RF7
Nombre	Registro de progreso
Descripción	La aplicación debe registrar y almacenar el progreso del usuario en cada sesión de ejercicio.
Objetivo	Permitir a los usuarios y a los profesionales de la salud monitorear el progreso en la rehabilitación.
Dependencias	RF2, RF4

ID	RF8
Nombre	Recordatorios y notificaciones
Descripción	La aplicación debe enviar recordatorios y notificaciones a los usuarios para realizar sus ejercicios diarios.
Objetivo	Mejorar la adherencia del usuario a la rutina de ejercicios mediante recordatorios oportunos.
Dependencias	RF2

ID	RF9
Nombre	Entorno social motivador
Descripción	La aplicación debe crear un entorno social que incluya la posibilidad de que un usuario busque a otros usuarios, y que estos puedan seguirse (followers).
Objetivo	Fomentar la creación de un entorno social que motive y divierta al usuario para mejorar la adherencia al tratamiento.
Dependencias	RF2, RF7

3.1.1.2. Dependencias de requisitos funcionales

Para expresar de forma gráfica las dependencias entre los requisitos identificados, se ha creado una matriz de dependencias que ilustra las relaciones y dependencias de cada requisito funcional de la aplicación con el resto de los requisitos. Se entiende una dependencia como la relación en la que un requisito depende de la implementación o el funcionamiento de otro requisito. Es importante identificar estas dependencias para planificar adecuadamente el desarrollo y garantizar que los requisitos que dependen unos de otros se implementen en el orden correcto. Por ejemplo, el requisito RF2 (Autenticación de usuarios) depende de RF1 (Registro de usuarios), ya que, si RF1 no se implementa correctamente, los usuarios no pueden crearse una cuenta y no se puede tener un registro de usuarios, por tanto, resulta imposible llevar a cabo un proceso de autenticación de usuarios.

En la matriz, cada requisito aparece en una fila y en una columna, y viene identificado por su ID. Dado que un requisito no puede depender de sí mismo, las posiciones de la diagonal de la matriz no pueden marcar ninguna dependencia. De hecho, la mitad superior e inferior de la matriz representan la misma información, de forma que esta solo se representará en una de las mitades (típicamente, la inferior a la diagonal). Este formato de matriz facilita la visualización de dependencias y permite planificar la implementación de manera efectiva, asegurando que los requisitos críticos se aborden en el orden correcto.

Las dependencias identificadas son las siguientes:

- **RF1 (Registro de usuarios):** No tiene dependencias.
- **RF2 (Autenticación de usuarios):** Depende de RF1.
- **RF3 (Concesión de permisos):** No tiene dependencias.
- **RF4 (Seguimiento facial en tiempo real):** Depende de RF3.
- **RF5 (Lista de exergames o videojuegos serios):** Depende de RF4.
- **RF6 (Instrucciones del ejercicio/juego):** Depende de RF5.
- **RF7 (Registro de progreso):** Depende de RF2 y RF4.
- **RF8 (Recordatorios y notificaciones):** Depende de RF2.
- **RF9 (Entorno social motivador):** Depende de RF7, RF2

Basándonos en esto, la matriz de dependencias de los requisitos funcionales de la aplicación es la que se muestra en la [tabla 2](#).

	RF1	RF2	RF3	RF4	RF5	RF6	RF7	RF8	RF9
RF1									
RF2	X								
RF3									
RF4			X						
RF5				X					
RF6					X				
RF7		X		X					
RF8		X							
RF9		X					X		

Tabla 2. Matriz de dependencias de los requisitos funcionales del sistema.

3.1.2. Requisitos no funcionales

Los RNF identificados para la aplicación son los descritos a continuación.

ID	RNF1
Nombre	Rendimiento
Descripción	La aplicación debe ser capaz de procesar y analizar los datos de seguimiento facial en tiempo real sin retrasos perceptibles.
Objetivo	Garantizar una experiencia de usuario fluida durante la realización de ejercicios.

ID	RNF2
Nombre	Usabilidad
Descripción	La interfaz de usuario de la aplicación debe ser intuitiva y fácil de usar, permitiendo a los usuarios encontrar y usar las funciones principales con facilidad. La aplicación debe obtener una puntuación mínima de 70 en el cuestionario SUS [16] .
Objetivo	Hacer una app universal fácil de usar para personas de diferentes edades y habilidades técnicas.

ID	RNF3
Nombre	Seguridad
Descripción	La aplicación debe garantizar que todos los datos personales de los usuarios estén protegidos contra accesos no autorizados.
Objetivo	Proteger la privacidad y seguridad de los datos del usuario.

ID	RNF4
Nombre	Compatibilidad
Descripción	La aplicación debe ser compatible con una amplia gama de dispositivos Android, pudiendo funcionar en el 95% de los dispositivos (Android 8.0 en adelante) como mínimo. [17]
Objetivo	Asegurar que la aplicación funcione en la mayoría de los dispositivos móviles utilizados por los usuarios.

ID	RNF5
Nombre	Fiabilidad
Descripción	La aplicación debe ser estable y no debe fallar durante el uso normal.
Objetivo	Proveer una experiencia confiable para los usuarios, minimizando interrupciones y fallos.

ID	RNF6
Nombre	Interfaz adaptable
Descripción	La interfaz de usuario de la aplicación debe adaptarse y mantener su estilo en distintos tamaños de pantalla y distintas resoluciones, incluyendo tabletas u otros dispositivos.
Objetivo	Hacer que la experiencia del usuario sea coherente y no cambie dependiendo del dispositivo que use.

ID	RNF7
Nombre	Respuesta en tiempo real
Descripción	Cada uno de los juegos debe ser capaz de detectar en tiempo real con latencias mínimas los movimientos del rostro del usuario, y responder a ellos con los respectivos movimientos de cada juego.
Objetivo	Ofrecer una experiencia fluida al usuario en la que no haya retrasos que afecten a la realización de los ejercicios.

ID	RNF8
Nombre	Entorno social motivador
Descripción	La aplicación debe incluir un entorno social motivador, incentivando a los usuarios a seguir participando en los exergames y, por ende, en sus ejercicios de rehabilitación.
Objetivo	Generar un entorno social que fomente la inclusión y motivación de los usuarios y haga que se sientan acompañados.

3.2. Metodología de desarrollo

Para el desarrollo/implementación de la aplicación se ha optado por seguir la **metodología Scrum**, marco de trabajo para el desarrollo ágil de software. Esta metodología es muy común en empresas que se dedican al desarrollo de software a medida, y que prefieren mantener un contacto estrecho y comunicación frecuente con el cliente o *stakeholders*. Scrum es particularmente adecuado para proyectos que requieren flexibilidad y adaptabilidad, como en este caso, el desarrollo de una aplicación móvil. Usar Scrum presenta ventajas como desarrollo incremental, adaptabilidad, mejora continua comunicación y colaboración frecuente, etc.

En este caso, se ha optado por adoptar Scrum a un contexto en el cual el desarrollo es individual, es decir, que la implementación de la aplicación corre a cargo de una sola persona. Se recomienda que los sprints tengan una duración constante y relativamente corta, generalmente entre 1 a 4 semanas, para permitir al equipo recibir y actuar sobre el feedback de manera rápida y eficiente.

Teniendo en cuenta los RF identificados anteriormente, usar sprints de 3 semanas se ha considerado como la mejor opción, ya que es suficientemente corto para mantener la agilidad, pero suficientemente largo para completar una cantidad significativa de trabajo.

De esta forma, el proyecto consta de 5 Sprints, cada uno de 3 semanas de duración. Esto resulta en un total de 15 semanas para completar la primera versión de la aplicación, suponiendo que no haya retrasos significativos y que cada Sprint se complete a tiempo.



Figura 3. Representación gráfica de un desarrollo en 5 sprints.

Se crea un *Product Backlog*, es decir, una lista completa de todas las tareas necesarias para el proyecto, y se priorizan estas tareas. Al inicio de cada sprint se hace una planificación del trabajo que se realizará en el sprint, creando el *Sprint Backlog*, que es la lista de tareas del *Product Backlog* que se realizarán durante el sprint. Las tareas del *Sprint Backlog* se escogen en base a la prioridad de la tarea y la capacidad de trabajo disponible. Dado que el desarrollo corre a cargo de una sola persona, esta debe adoptar todos los roles típicos de Scrum.

Respecto a las ceremonias de Scrum, estas son prácticas clave que estructuran y guían el proceso de desarrollo ágil, y se deben adaptar a un proyecto de desarrollo individual como el presente. Las ceremonias de Scrum y sus participantes pueden adaptarse a este contexto.

- Se entenderá por clientes o stakeholders al tutor del proyecto.
- La ceremonia de **Daily Scrum** (reuniones diarias de unos 15 minutos donde se revisa el progreso del día anterior, se planifica el trabajo de ese día y se abordan imprevistos) se traducen en las revisiones diarias del trabajo realizado, muchas veces, por correo electrónico.
- La ceremonia de **Sprint Review** (reuniones al final de cada sprint donde se presenta a los stakeholders el trabajo realizado durante el sprint, se discute el trabajo realizado y se recogen comentarios o dudas) se traducen en reuniones presenciales con el tutor en las cuales se muestra el trabajo realizado durante el sprint. Esto también incluye la Sprint Retrospective, donde discute qué ha ido bien, qué no ha ido bien y qué se puede mejorar. El objetivo es presentar el trabajo realizado, validarlo, exponer los desafíos enfrentados y obtener el visto bueno del stakeholder (tutor).

3.2.1. Descripción de los sprints

Las consideraciones anteriores permiten realizar una planificación inicial y una definición de los sprints de forma detallada. A continuación, se sugiere una planificación inicial basada en los requisitos funcionales y consideraciones identificadas anteriormente:

1. **Sprint 1:** Implementación de RF1, RF2 y RF3.
2. **Sprint 2:** Implementación de RF4.
3. **Sprint 3:** Implementación de RF5.
4. **Sprint 4:** Implementación de RF6, RF7, RF8 y RF9.
5. **Sprint 5:** Pruebas de usuario, posibles optimizaciones y redacción del informe.

La cantidad de RF implementados en cada sprint se ha decidido en base a la estimación de horas de trabajo que llevará cada uno. Los sprints 2 y 3 implementan solamente los requisitos funcionales de seguimiento facial en tiempo real y lista de exergames, respectivamente, debido a que se considera que estos requisitos son de mayor complejidad que el resto. Por ejemplo, se considera que implementar el seguimiento facial en tiempo real llevará más horas de programación que implementar y gestionar concesión de permisos de acceso a cámara. Estos dos sprints, por tanto, constituyen el núcleo del proyecto y el eje central en torno al cual gira la aplicación.

La planificación divide las funcionalidades de la aplicación en bloques lógicos que se pueden abordar de manera secuencial. Cada sprint se enfoca en un conjunto específico de funcionalidades que son necesarias para construir las siguientes. Este enfoque escalonado y bien organizado garantiza que cada funcionalidad se construya sobre una base sólida y que el desarrollo progrese de manera lógica y eficiente. La planificación permite ajustes durante el desarrollo si se encuentran desafíos técnicos o si se identifican mejoras durante las pruebas de usuario. El último sprint puede ser utilizado para ajustar y optimizar las funcionalidades. La revisión y retrospectiva del sprint se realizan al final de cada sprint. En estas ceremonias se muestra el estado actual de la aplicación, se proporciona feedback sobre el trabajo realizado y se revisa el progreso comparándolo con los objetivos establecidos en la línea base del proyecto (Sprint Review).

También se identifican áreas de mejora y se analiza el desempeño del trabajo y la gestión del trabajo realizado durante el sprint (Sprint Retrospective).

3.3. Planificación del desarrollo

El proyecto comienza el **19 de febrero del 2024**, coincidiendo con el inicio del segundo cuatrimestre del curso escolar 2023/2024. Antes de comenzar el proyecto, este debe planificarse meticulosamente para asegurar su éxito.

El desarrollo de la aplicación se ha planificado de forma que se asegure el cumplimiento de todos los requisitos funcionales descritos en el apartado anterior. La implementación de cada uno de estos debe coordinarse con la implementación del resto, garantizando de esta manera que el desarrollo se hará de forma estructurada y coordinada. Cada uno de los requisitos funcionales a implementar puede ser entendido como una historia de usuario o *User Story*. En el contexto de metodologías ágiles como Scrum, las *User Stories* son descripciones breves de una funcionalidad desde la perspectiva del usuario final y suelen descomponerse en subtarefas para una implementación más detallada. Este enfoque no solo facilita la gestión del trabajo a realizar, sino que además permite centrar el desarrollo en torno a las necesidades y expectativas del usuario, asegurando que cada funcionalidad tenga un propósito claro.

Las User Stories obtenidas fruto de la reinterpretación de los RF son las siguientes:

	Requisito	User Story
RF1	Registro de usuarios	Como usuario, quiero poder registrarme y crear un perfil con mis datos personales y tener un perfil útil para mi rehabilitación.
RF2	Autenticación de usuarios	Como usuario registrado, quiero poder autenticarme para acceder a mis datos y ejercicios.
RF3	Concesión de permisos	Como usuario, quiero que la aplicación solicite los permisos necesarios para su correcto funcionamiento.
RF4	Seguimiento facial en tiempo real	Como usuario, quiero que la aplicación haga un seguimiento en tiempo real de los movimientos de mi rostro para evaluar mis ejercicios y mi movilidad.
RF5	Lista de exergames	Como usuario, quiero acceder a una lista de juegos específicos para distintos problemas cervicales.
RF6	Instrucciones del exergame	Como usuario, quiero recibir instrucciones claras y detalladas sobre cómo realizar los ejercicios correctamente.
RF7	Registro de progreso	Como usuario, quiero que la aplicación registre y almacene mi progreso en cada sesión de ejercicio.
RF8	Recordatorios y notificaciones	Como usuario, quiero recibir recordatorios y notificaciones para realizar mis ejercicios diario.

RF9	Entorno social motivador	Como usuario, quiero poder interactuar en un entorno social que me motive en mi proceso de rehabilitación.
------------	--------------------------	--

Tabla 3. Requisitos funcionales y user stories correspondientes a estos.

Para poder coordinar las tareas y subtareas, es necesario conocer cuáles son las dependencias entre distintas tareas y de qué tipo son estas dependencias. Existen distintos tipos de dependencias entre tareas y/o subtareas [18]:

- **CC** (comienzo a comienzo): Dos tareas o subtareas deben iniciarse a la vez.
- **FF** (fin a fin): Dos tareas o subtareas vinculadas deben finalizar a la vez.
- **FC** (fin a comienzo): El final de una tarea o subtaska marca el inicio de la siguiente.
- **CF** (comienzo a fin): El comienzo de una tarea o subtaska marca el fin de otra tarea

Para cada subtaska extraída de una tarea se detallará las dependencias con otras subtareas y el tipo de dependencia del que se trata. Es importante definir el tipo de dependencia ya que esto definirá el orden en el que se implementarán las subtareas y, por tanto, marcará la cronología de desarrollo de la aplicación.

3.3.1. Planificación detallada con subtareas

A continuación, [la tabla 4](#) detalla las subtareas identificadas y el tipo de dependencia de cada una de ellas respecto a las demás.

	Subtarea	ID	Dependencias
RF1: Registro de usuarios.	Configuración del IDE de Android Studio y PostgreSQL.	RF1-1	-
	Creación de una base de datos con información sobre los usuarios, juegos, problemas cervicales, etc.	RF1-2	RF1-1 → FC
	Crear pantalla de registro con formularios para datos personales.	RF1-3	RF1-2 → FC
	Implementación de la lógica del registro de usuarios con Java y SQL.	RF1-4	RF1-3 → FC
	Pruebas unitarias para comprobar el correcto funcionamiento de la creación de usuarios.	RF1-5	RF1-4 → FC
RF2: Autenticación de usuarios.	Crear pantalla de inicio de sesión con formularios para introducir usuario y contraseña.	RF2-1	RF1-5 → FC
	Implementación de la lógica de la autenticación de usuarios con Java y SQL.	RF2-2	RF2-1 → FC
	Pruebas unitarias para comprobar el correcto funcionamiento de la autenticación de usuarios.	RF2-3	RF2-2 → FC
RF3: Concesión de permisos.	Identificar todos los permisos necesarios para la app.	RF3-1	RF 2-3 → FC
	Implementar solicitud de permisos al iniciar la app.	RF3-2	RF3-1 → FC
	Manejar la concesión o denegación de permisos.	RF3-3	RF3-2 → FC

	Pruebas unitarias para comprobar la correcta gestión y petición de permisos al usuario.	RF3-4	RF3-3 → FC
RF4: Seguimiento facial en tiempo real.	Implementar la lógica de detección de rostros con las librerías de ML Kit para Java.	RF4-1	RF3-4 → FC
	Procesar y mostrar en tiempo real los resultados del seguimiento facial a través de la interfaz de usuario.	RF4-2	RF4-1 → FC
	Pruebas unitarias para comprobar la correcta implementación del seguimiento facial.	RF4-3	RF4-2 → FC
RF5: Lista de exergames o videojuegos serios.	Diseñar y desarrollar con Java y XML la interfaz de usuario con la pantalla principal de la aplicación y las pantallas previas a los juegos.	RF5-1	RF4-3 → FC
	Implementar al menos un juego en el que el usuario pueda jugar moviendo el rostro con Java y XML.	RF5-2	RF5-1 → FC
	Pruebas de navegación entre las distintas pantallas de la aplicación para ir de un juego a otro.	RF5-3	RF5-2 → FC
	Pruebas unitarias para comprobar el correcto funcionamiento de cada juego con movimientos faciales.	RF5-4	RF5-3 → FC
RF6: Instrucciones del ejercicio/juego	Redactar un apartado previo al juego donde se detallen las instrucciones de este.	RF6-1	RF5-4 → FC
	Implementar la lógica para mostrar las instrucciones antes de iniciar cada ejercicio usando como lenguajes Java y XML.	RF6-2	RF6-1 → FC
	Asegurar que la información proporcionada es suficiente, coherente y correcta.	RF6-3	RF6-1 → FC
RF7: Registro de progreso	Implementar lógica para registrar en la base de datos el progreso del usuario tras cada sesión de juego.	RF7-1	RF6-2 → FC
	Comprobar que la monitorización del desempeño del usuario se está llevando a cabo exitosamente.	RF7-2	RF7-1 → FC
RF8: Recordatorios y notificaciones	Implementar sistema de notificaciones push.	RF8-1	RF7-2 → FC
	Crear lógica para enviar recordatorios diarios.	RF8-2	RF8-1 → FC
	Comprobar que las notificaciones se envían correctamente al usuario tal y como se espera.	RF8-3	RF8-2 → FC
RF9: Entorno social motivador	Implementar un buscador de usuarios que permita encontrar a otros usuarios de la aplicación y ver los datos de sus perfiles.	RF9-1	RF8-3 → FC
	Implementar la posibilidad de seguir y dejar de seguir a otros usuarios.	RF9-2	RF9-1 → FC
	Pruebas unitarias para comprobar el correcto funcionamiento del entorno social.	RF9-3	RF9-2 → FC

Tabla 4. Subtareas identificadas para la implementación con su identificador y dependencias.

3.3.2. Diagrama de Gantt

Para ilustrar la planificación de la implementación y comprender de forma más fácil, rápida y visual el proceso que esta supone, la [figura 4](#) muestra un diagrama de Gantt. Este diagrama muestra las subtarefas identificadas para cada requisito funcional, detalladas cronológicamente teniendo en cuenta sus dependencias y, por tanto, su prioridad.

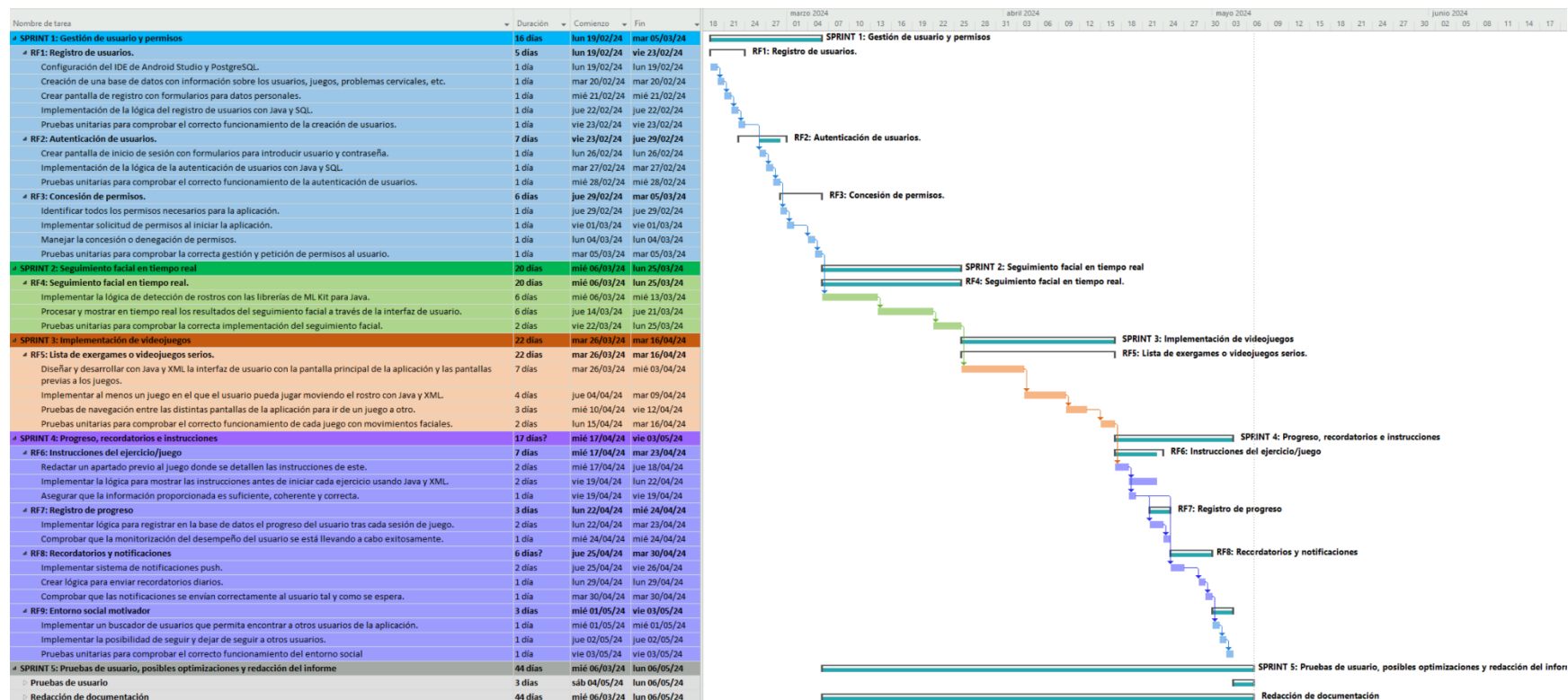


Figura 4. Diagrama de Gantt que ilustra la cronología de implementación de las subtarefas en base a su prioridad.

Dado que el proyecto tiene una duración de 60 días y, teniendo en cuenta que se debe trabajar en él una media de 8 horas diarias, esto supone un total de 480 horas estimadas. Teniendo en cuenta que el trabajo de fin de grado corresponde a 18 ECTS, esto cumple con las expectativas del trabajo estimado, ya que 1 ECTS es equivalente a 25 horas y esto supone que se supera el total de 450 horas de trabajo dedicadas al proyecto.

4. TECNOLOGÍAS Y DISEÑO

En este capítulo se exponen las tecnologías usadas y el diseño del sistema. Se identifican los elementos que forman el sistema, sus propiedades. Para ello, se tendrá en cuenta los requisitos funcionales y no funcionales definidos con anterioridad, poniendo especial foco en los requisitos no funcionales.

La descripción de estos elementos se hará con un lenguaje general y sin entrar en tecnicismos. También se razonarán las decisiones tomadas en referencia a la elección de las tecnologías usadas, mencionando alternativas en algún caso.

En primer lugar, se estudiará el estado del arte, donde se proporciona una visión general de las diversas tecnologías para la detección de movimiento existentes, así como entornos de desarrollo y lenguajes de programación disponibles para la creación de aplicaciones de exergaming y proyectos similares. En este subapartado, se analizarán las opciones existentes, evaluando su aplicabilidad y relevancia en el contexto del presente proyecto.

4.1 Estado del arte

Para cumplir el objetivo de desarrollar un exergame para dispositivos móviles que permita la interacción mediante el movimiento de la cabeza del usuario sin el uso de sensores adicionales ni marcadores, es imprescindible indagar en las distintas tecnologías y métodos existentes, de forma que se pueda hacer una elección óptima de las que se usarán para abordar el problema.

En este capítulo se presentan las tecnologías y métodos apropiadas para la resolución de la problemática que se presenta.

Se realiza un análisis detallado en el que se identifican puntos fuertes y débiles para cada uno, presentando así las ventajas y desventajas que supondría el uso de cada tecnología o método. De esta forma, se crea una base sólida de conocimiento en base a la cual comenzar a desarrollar una solución que permita, mediante una aplicación para dispositivos móviles, realizar el seguimiento de la cabeza del usuario y transformarlo en acciones interactivas sobre el dispositivo.

Ya que se pretende poder llegar al máximo número de usuarios, se ha elegido desarrollar la aplicación para dispositivos Android debido a que su cuota de mercado es de un 70% aproximadamente. Esto implica que Android posee alrededor del 70% del mercado global de sistemas operativos móviles [\[19\]](#).

En la [figura 5](#), pueden verse distintas tecnologías de detección de movimiento, plataformas de desarrollo, lenguajes de programación, etc. típicamente usados para abordar problemas similares.



Figura 5. Algunas de las tecnologías, herramientas o metodologías consideradas.

En el transcurso de este apartado, se detallan las áreas clave relevantes para el desarrollo de la aplicación de exergaming mencionando los siguientes aspectos:

- **Entornos de desarrollo:** Dado que el objetivo de este proyecto es desarrollar una aplicación para usuarios de Android, se mencionan los entornos de desarrollo integrados (IDEs) que suelen usarse con este fin.
- **Tecnologías de detección de movimiento:** Se detallan las tecnologías que se utilizan convencionalmente con el fin de conseguir que el sistema detecte movimientos o gestos del usuario que lo está usando.
- **Lenguajes de programación:** Se exponen los lenguajes de programación con los que se puede trabajar para llegar a una solución, describiendo para cada uno sus ventajas y desventajas.

Por tanto, a continuación, se proporciona un análisis y explicación detallados para cada uno de estos bloques.

4.1.1 Entornos de desarrollo

Como ya se ha mencionado anteriormente, el objetivo de este proyecto es, en pocas palabras, el desarrollo de una aplicación para dispositivos Android. Para ello es evidente la necesidad de usar un entorno de desarrollo integrado donde poder implementar el código que definirá la lógica de la aplicación, así como el diseño de su interfaz de usuario.

Los IDEs que se contemplan para desarrollar la aplicación son Android Studio, React Native e IntelliJ IDEA.

4.1.1.1 Android Studio

Android Studio es el entorno de desarrollo integrado oficial para el desarrollo de aplicaciones Android [20]. Es propiedad de Google y ofrece una amplísima variedad de utilidades, funcionalidades y herramientas integradas, entre las cuales se incluyen un editor de código

avanzado, un emulador de dispositivos Android de distinta gama, generación y versión para probar la aplicación desarrollada, herramientas que facilitan el diseño de interfaces de usuario, herramientas de depuración, consola, compilador basado en Gradle, terminal de comandos y control de versiones integrado, entre otros.

Android Studio permite el uso de lenguajes varios lenguajes, entre los cuales destacan *Java*, *Kotlin*, y en menor medida, *C++*. Además de estos, el entorno permite el uso de otros lenguajes y frameworks gracias al uso de extensiones o plugins. Algunos de estos lenguajes son *Python*, *JavaScript*, *Dart* o *Groovy*. Sin embargo, los lenguajes predominantes son Java y Kotlin, ya que los otros no resultan tan útiles para desarrollar aplicaciones Android.

La versión actual (febrero de 2024) es la 2023.2.1, también conocida como “Iguana” [21]. Esta versión ofrece mejoras significativas de rendimiento y funcionalidad respecto a versiones anteriores, facilitando y haciendo más eficiente y productivo el proceso de desarrollo.

4.1.1.2 React Native

React Native es un framework de código abierto creado por Meta, y permite el desarrollo de aplicaciones Android y iOS usando JavaScript y React [22]. Una de las principales ventajas que presenta respecto a otros IDEs es su capacidad de desarrollo multiplataforma, ya que con React Native se pueden crear aplicaciones para Android y iOS. Permite el uso de lenguajes como JavaScript, Java, Python, C++, Objective-C. Al igual que Android Studio, ofrece funcionalidades como herramientas de diseño de interfaces de usuario (UI), editor de código avanzado u otras tantas.

Sin embargo, esto hace que algunas de sus funcionalidades dependan de la creación de “puentes de código” [23], lo cual puede ser propenso a errores e influye negativamente en el rendimiento de aplicaciones que requieran ser usadas en escenarios de cierta complejidad.

4.1.1.3 IntelliJ IDEA

En este caso, se trata de un entorno de desarrollo integrado desarrollado por JetBrains [24], capacitado para integrarse con diversas tecnologías y herramientas, y para soportar distintos lenguajes como Java, Kotlin y JavaScript. Incluye características avanzadas como ayudas a la concentración o completado automático de código u otras muchas.

Sin embargo, esta gran versatilidad y potencia hacen que sea pesado en términos de uso de recursos del sistema, afectando negativamente a la experiencia del programador por falta de rendimiento en máquinas poco potentes. Por otro lado, la complejidad de configuraciones y características que presenta el entorno provoca que el proceso de aprendizaje sea lento para nuevos usuarios.

Por otro lado, dado que la aplicación se conectará a una base de datos en la que almacenará información sobre exergames, usuarios y otros, se necesita un entorno de desarrollo en el que implementar scripts en SQL. Para ello, por su versatilidad y facilidad de uso, se propone VSCode.

4.1.1.4 VSCode

Visual Studio Code (más comúnmente conocido como VSCode) [\[25\]](#) es un entorno de desarrollo o editor de código abierto creado por Microsoft, que no es específico de ningún lenguaje de programación, sino que permite el uso de distintos lenguajes mediante la descarga de extensiones. Pese a no ser un IDE específico para Android, la gran cantidad de extensiones, su versatilidad y facilidad de uso, lo convierten en una opción de uso plausible. Es de gran utilidad para desarrolladores que trabajan con distintos lenguajes de programación y plataformas a la vez, y cuenta con una consola en la que se puede ejecutar código. En el contexto de este proyecto, es de gran utilidad para implementar y depurar los scripts SQL que interactúan con la base de datos.

4.1.2 Tecnologías de detección de movimiento

Para que el desarrollo de los exergames sea exitoso, es de vital importancia conocer cómo integrar tecnologías de detección de movimiento en tiempo real con videojuegos. La detección de movimientos de los usuarios es crucial para poder asegurar la interacción entre el usuario y la aplicación, permitiéndole jugar a varios exergames que sirven como rehabilitación.

Actualmente, existen tecnologías de detección de movimiento basadas en sensores, controladores u otros dispositivos remotos, y, por otro lado, existen técnicas de detección de movimiento como la visión por computador.

Sin embargo, puesto que se pretende desarrollar una aplicación de exergames para dispositivos Android, el uso de sensores, cámaras o controladores externos que permitan monitorizar y analizar el movimiento de los usuarios, probablemente no sea la mejor opción, por motivos evidentes como el coste, la comodidad o facilidad de uso. Además, uno de los objetivos del proyecto era no utilizar sensores adicionales. La mejor opción es llevar a cabo la detección de movimiento de usuarios aprovechando las propiedades de las tabletas y smartphones que usarán la aplicación, es decir, usando técnicas de visión por computador.

A continuación, se detallan algunas técnicas basadas en visión por computador.

4.1.2.1 Técnicas de detección de movimiento basadas en visión por computador

Las técnicas de detección de movimiento en tiempo real basadas en visión por computador se centran en la ejecución de algoritmos con la capacidad de analizar imágenes en tiempo real, con el objetivo de poder identificar objetos o animales, o rastrear y analizar el movimiento de usuarios. Estas técnicas se usan no solamente en videojuegos, sino en contextos de seguridad, estadística, análisis de imágenes o vídeos, medicina u otros muchos campos. La capacidad de un sistema de analizar y comprender imágenes en tiempo real es la que otorga el nombre a esta técnica, ya que la capacidad de un computador de “ver” su entorno y reconocerlo genera la sensación de que el computador tiene capacidad de visión [\[26\]](#).

Algunas de las técnicas principales de visión por computador con fines de detección de movimiento, son las descritas a continuación.

4.1.2.1.1 Reconocimiento de gestos

El reconocimiento de gestos [\[27\]](#) [\[28\]](#) es una técnica que consiste en que el sistema sea capaz de ver e interpretar gestos que el usuario hace con manos, brazos, piernas o incluso con el cuerpo entero. Estos gestos son interpretados por el sistema y permiten la interacción con el usuario sin necesidad de contacto físico con este con, por ejemplo, clicar en una pantalla u otros. Esta técnica es altamente usada en realidad virtual, en videojuegos o en sistemas que implementan este tipo de interacción para personas con discapacidades.

El reconocimiento de gestos podría ser de gran utilidad en el contexto de este proyecto, ya que permitiría al sistema reconocer el movimiento que el usuario está haciendo y utilizar estos movimientos para diseñar los controles de los exergames.

4.1.2.1.2 Seguimiento de puntos clave

El seguimiento de puntos clave [\[29\]](#) [\[30\]](#), o keypoint tracking, es otra técnica de visión por computador que consiste en el análisis de imágenes o videos con el objetivo de encontrar puntos característicos en ellos y hacer un seguimiento. Esta técnica puede usarse en todo tipo de objetos, pero habitualmente se usa en tareas de seguimiento de puntos clave del cuerpo humano. Es de gran utilidad para comprender movimientos o posturas, y en el contexto de una aplicación de exergaming, algunos de los puntos clave que se pueden identificar son características faciales, manos, articulaciones u otras zonas del cuerpo de los usuarios.

Esta técnica es comúnmente utilizada en análisis de imágenes médicas, para crear animaciones en la industria del cine y los videojuegos, para rehabilitación deportiva o física en general, etc. En el contexto del presente proyecto también puede ser de gran utilidad, para hacer seguimiento de puntos faciales clave y detectar movimientos que permitan al usuario interactuar con el sistema.

4.1.2.1.3 Reconocimiento y seguimiento facial

Las técnicas de reconocimiento y seguimiento facial consisten en la detección de la presencia de rostros en imágenes o videos. Se analizan los entornos capturados en foto o video en búsqueda de rostros y, al detectarse uno, se comienza a hacer un seguimiento.

Para conseguir detectar rostros con un porcentaje de fiabilidad que asegure un seguimiento exitoso, algunas de las técnicas de seguimiento y reconocimiento facial pueden incluir [técnicas de seguimiento de puntos clave](#). En este caso, los puntos clave serían ciertos puntos faciales como ojos, nariz, boca, orejas, etc.

Esto permite, entre otras tantas utilidades, el análisis de expresiones o gestos faciales, haciendo posible que los sistemas que lo implementan sean capaces de reconocer las emociones o estados de ánimo de los usuarios en base a sus expresiones faciales. Un ejemplo de este caso es el ilustrado en la [figura 6](#).

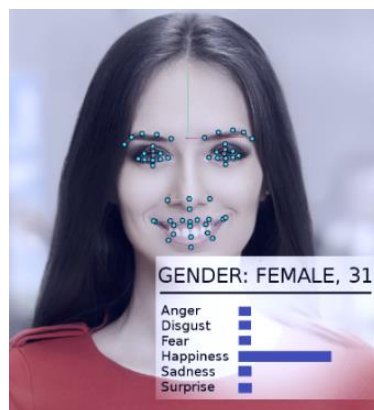


Figura 6. Ejemplo de reconocimiento facial y de puntos clave del rostro [31].

4.1.3 Lenguajes de programación

Dado que el propósito de este proyecto es implementar una aplicación de exergaming, es crucial seleccionar lenguajes de programación que no solo sean completos y eficientes, sino que también faciliten el desarrollo multiplataforma y la integración de características avanzadas como gráficos, interacción en tiempo real y conectividad con frameworks externos. A continuación, se describen los lenguajes de programación considerados para este proyecto:

4.1.3.1 Java

Es uno de los lenguajes más utilizados para el desarrollo de aplicaciones Android. Es un lenguaje complejo pero completo, maduro y con una sintaxis clara. Tiene un ecosistema de bibliotecas muy completo y robusto, que hacen de Java una muy buena opción para la implementación de interfaces de usuario complejas y la gestión de procesos en tiempo real.

4.1.3.2 Kotlin

Como el lenguaje oficial para el desarrollo de Android, Kotlin debe ser considerado como opción ya que ofrece una sintaxis moderna y concisa, mejorando la seguridad y reduciendo la cantidad de código necesario en comparación con Java. Es más simple y fácil de aprender que Java, pero al ser un lenguaje más reciente o moderno, no cuenta con tantas librerías o funcionalidades, es decir, no es un lenguaje tan completo.

Sin embargo, es el lenguaje oficial para aplicaciones Android, y cuenta con una gran interoperabilidad con Java, permitiendo aprovechar bibliotecas y frameworks existentes.

4.1.3.3 C#

Este lenguaje, junto con el motor de desarrollo Unity, es ideal para el desarrollo de juegos que requieren gráficos 3D o interacciones complejas. Unity permite la implementación multiplataforma, haciendo posible desplegar el exergame tanto en dispositivos móviles como en consolas y ordenadores.

Debe ser considerado como posible lenguaje ya que la combinación de C# con Unity es altamente usada en el mundo de los videojuegos complejos, pero dado que se pretende implementar una aplicación de exergaming con videojuegos en 2D, la utilización de C# junto con Unity resultaría, probablemente, contraproducente. Esto se debe a que, pese a ser un lenguaje completo y que ofrece grandes posibilidades, la complejidad de su utilización junto a Unity no resulta útil si no se tiene una necesidad clara de utilización.

4.2 Plataforma de desarrollo

A continuación, se describirán las opciones seleccionadas entre todas las posibilidades evaluadas en el [apartado 4.1](#), incluyendo el entorno de desarrollo, los lenguajes de programación elegidos, y la tecnología de detección de movimiento adoptada. Como se mencionó previamente, la plataforma de desarrollo es Android, dado que la aplicación está destinada a dispositivos móviles con este sistema operativo.

4.2.1 Entorno de desarrollo (IDE)

Para desarrollar la aplicación se han utilizado los lenguajes de programación **Java** y **XML**. El entorno de desarrollo integrado (IDE) empleado ha sido **Android Studio** en su versión “Iguana 2023.2.1” [\[21\]](#) y seleccionando Android 8.0 como versión base de Android [\[32\]](#).

Para el diseño e implementación de una base de datos relacional conectada con la aplicación, se ha usado **PostgreSQL** [\[33\]](#) en su versión 15.1 como sistema gestor de bases de datos, junto con el lenguaje **SQL** para interactuar con dicha base de datos. El código SQL se ha implementado en **Visual Studio Code** (VSCode) [\[34\]](#) como IDE para programar scripts SQL. Además, se han utilizado las herramientas **SQLShell** y **pgAdmin 4** (versión 8.2) para probar y ejecutar esos scripts en la base de datos.

XML se ha escogido como lenguaje para diseñar las UI porque es el estándar para definir interfaces de usuario en Android. Por otro lado se ha optado por usar Java como lenguaje para programar toda la lógica de la aplicación porque ya se tiene experiencia usando este lenguaje, además de que es uno de los más usados para aplicaciones Android, ofreciendo un gran soporte, rendimiento y compatibilidad. Una posible alternativa a Java hubiese sido desarrollar la lógica de la aplicación usando React Native (JavaScript) [\[22\]](#) o Kotlin [\[35\]](#), lenguaje altamente usado para aplicaciones Android. Se ha optado por Java porque Kotlin no cuenta con tanta documentación y recursos, y es un lenguaje con el que no se tiene tanta experiencia y soltura.

Android Studio se ha escogido como IDE debido a que es el IDE oficial para el desarrollo de aplicaciones Android y está optimizado específicamente para este propósito. Otra opción podría haber sido IntelliJ IDEA [\[24\]](#), pero se tiene más experiencia usando Android Studio.

Por último, SQL se ha escogido porque es el lenguaje estándar y más comúnmente usado para bases de datos relacionales. Se ha optado por utilizar VSCode para programar SQL porque es

flexible, ligero y rápido, y se ha usado PostgreSQL por su robustez y facilidad de uso con respecto a otras opciones como SQLite, con los cuales además no se tiene experiencia.

4.2.2 Otras herramientas de desarrollo

Durante el desarrollo de la aplicación, se ha implementado el uso de Git para control de versiones, permitiendo una correcta organización y gestión del código, las ramas, los cambios en cada rama, etc. El uso de Git permitirá hacer un buen seguimiento de los cambios en el código y de las subidas de estos cambios a sus respectivas ramas.

Para poder manejar esto de forma eficaz y sencilla, se ha usado **Fork** [36], que se conecta con Git y permite llevar un control de versiones desde una interfaz gráfica amigable y fácil de usar. Fork cuenta con 4 paneles principales:

1. **Panel superior izquierdo:** Este panel muestra todos los commits y cambios locales que todavía no han sido subidos a la rama de desarrollo en la cual se está trabajando.
2. **Panel inferior izquierdo:** Este panel muestra una lista con todas las ramas locales y las ramas en el repositorio remoto (*origin*). También muestra los *stashes* o los submódulos (packages u otros). Este panel nos permitirá saber en qué rama estamos, y podremos cambiar de rama de desarrollo con un simple doble clic en la rama a la que se quiere ir.
3. **Panel superior derecho:** Muestra una representación gráfica en forma de árbol de todas las ramas de desarrollo, los cambios y commits sobre estas, y las fusiones (*merge*) de una rama a otra.
4. **Panel inferior derecho:** Nos muestra los elementos que se van a subir con un commit o los que se subieron con un commit que hayamos seleccionado.

A continuación, la [figura 7](#) muestra los cambios generados con un commit inicial al crear un nuevo repositorio llamado “exergames_beta” y subir los cambios en él. El código que se ve marcado en verde es código nuevo que no existía previamente en la rama. Se puede apreciar que existen un total de 10 ramas creadas (master, development y ex-dev-1/8), actualmente se trabaja en la rama development y no hay cambios locales

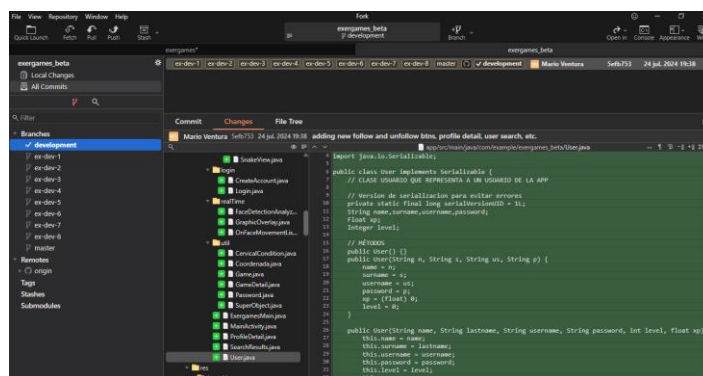


Figura 7. Cambios en el repositorio ‘exergames beta’.

Alternativamente, todos los comandos de Git se pueden lanzar desde la pantalla de comandos que nos ofrece el Powershell de Windows, aunque evidentemente, es más cómodo e intuitivo hacerlo desde Fork. La [figura 8](#) ilustra un ejemplo de uso de comandos de Git.

```
PS D:\tfg\exergames_beta> git branch
* master
PS D:\tfg\exergames_beta> git status
On branch master
nothing to commit, working tree clean
PS D:\tfg\exergames_beta> |
```

Figura 8. Comandos de Git ejecutados en el Powershell de Windows.

Para asegurar un correcto control de versiones, se trabaja bajo el siguiente método:

- **Rama *master* para la versión final:** La rama *master* contiene la aplicación real o “final”. El código en esta rama siempre debe compilar y funcionar, y debe haber sido probado previamente.
- **Rama *development* para la última versión funcional:** En la rama *development* se encuentra el código de la última versión funcional de la aplicación, que, tras ser probado, se mueve a *master*. El código en esta rama presenta una versión diferente con funcionalidades nuevas respecto a la aplicación que hay en *master*, y también debe funcionar correctamente siempre.
- **Ramas de características *ex-dev-X*:** Cada nuevo requisito o tarea se programa en una rama nueva que partirá de *development* y tendrá el nombre de *ex-dev-X* (exergames-development-X), donde *X* representa el número de la tarea o requisito funcional a implementar.
- **Merge de características a *development*:** Cuando se acaba de programar una tarea y se comprueba que funciona bien en la rama, se hace merge de *development* a esa rama y se prueba si sigue funcionando una vez integrado con el resto de la aplicación. Si funciona correctamente, se mueve ese código a *development* y se borra la rama *ex-dev-X* en local y en *origin*, para evitar manejar versiones innecesarias de la aplicación.
- **Movimiento de cambios de *development* a *master*:** Al final de cada sprint se mueve todo el código de *development* a *master*, de forma que *master* siempre contenga una versión funcional de la aplicación.

Esto proporciona un método sólido con el cual trabajar y que sigue buenas prácticas de desarrollo y control de versiones.

La [figura 9](#) muestra la interacción de los distintos elementos de la aplicación, las tecnologías que usa y los frameworks empleados. La parte más importante es la detección de las coordenadas de la cabeza de los usuarios sin el uso de sensores adicionales. Por ello, se usa visión por computador. Dado que no se quieren colocar marcadores sobre el usuario, se ha considerado que detectar las coordenadas de la nariz del usuario en tiempo real sobre las imágenes capturadas por la cámara

frontal del dispositivo es la mejor opción. Esto se debe a que la nariz cuenta con unas características (las fosas nasales son claramente más oscuras que muchas partes del rostro, por ejemplo) que la hacen fácilmente detectable con técnicas de visión por computador. Por tanto, se implementa una técnica de seguimiento de puntos clave.

Para este propósito, se ha optado por el uso del framework ML Kit de Google.

ML Kit es una biblioteca de herramientas de machine learning que facilita la integración de modelos de inteligencia artificial en aplicaciones móviles [37]. Proporciona una variedad de funcionalidades para tareas como el reconocimiento de texto, la detección de rostros y el análisis de imágenes.

Se ha optado por usar ML Kit por su robustez y facilidad de integración en aplicaciones móviles, así como por su capacidad para realizar detección de características faciales de manera eficiente y precisa. En este caso particular, se ha decidido utilizar ML Kit para evaluar su rendimiento en la detección de características faciales, lo cual se ha establecido como uno de los objetivos del Trabajo de Fin de Grado (TFG).

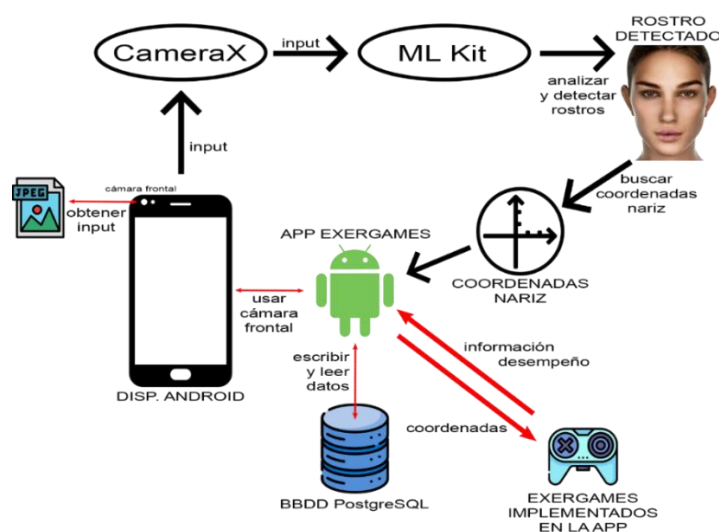


Figura 9. Interacción de distintas tecnologías y frameworks con la aplicación de exergaming.

A modo de observación, cabe mencionar que el exergame implementado forma parte de la propia aplicación de exergaming, es decir, no es un elemento separado, sino que está incluido dentro de la aplicación. En la imagen se muestra por separado ('APP EXERGAMES' y 'EXERGAMES IMPLEMENTADOS EN LA APP') con el objetivo de ilustrar de qué forma se utiliza la información detectada con la cámara frontal para poder jugar. Aunque actualmente la aplicación incluye un único exergame, está preparada para integrar más juegos en el futuro, cada uno de los cuales podría emplear esta información de distintas maneras según sus mecánicas específicas. Las coordenadas detectadas se usan en cada juego para hacer cálculos específicos para este, y en base

a los resultados obtenidos en estos cálculos, se realizan distintas acciones en los juegos (mover personajes, saltar, u otras acciones típicas en videojuegos), permitiendo de esta manera la jugabilidad.

Por otro lado, cada juego tiene un sistema de puntuación en base a sus objetivos, que permite evaluar el desempeño del usuario. Tras cada partida, esta información es enviada a la base de datos, que contiene un registro histórico de todas las puntuaciones de cada paciente en cada juego. Esto permite una monitorización y seguimiento continuo del desempeño y evolución del usuario en su proceso de rehabilitación.

4.3 Diseño

En este apartado se ilustra el diseño y la arquitectura de la aplicación implementada para el proyecto. En la [figura 10](#) se ilustra la aplicación diseñada, mediante un diagrama de flujo de las pantallas (UI) que proporciona una visión global de la arquitectura funcional.



Figura 10. Diseño estructural de la aplicación de exergaming.

5. IMPLEMENTACIÓN

A continuación, se detallan los pasos seguidos en el proceso de implementación de la aplicación de exergaming.

5.1 Registro de usuarios

En primer lugar, para que la aplicación de exergaming pueda ser de utilidad para pacientes reales, estos deben poder registrarse como usuarios de la aplicación. Sin un usuario no se puede usar la aplicación porque toda la información sobre el desempeño y progreso de los pacientes en los juegos va asociada a un usuario en concreto.

5.1.1 Información necesaria para la aplicación

Como primer paso para crear un registro de usuarios, se debe crear una base de datos que permita almacenar la información necesaria para que la aplicación funcione. Esta información incluye, entre otros, la asociada a los usuarios de la aplicación, permitiendo que se den de alta nuevos usuarios y se almacene su información. Los usuarios también tendrán la posibilidad de modificar la información de su perfil una vez lo hayan creado.

La aplicación de exergaming tiene que contener información sobre:

- **Usuarios:** Se debe saber el nombre, apellido, nombre de usuario y contraseña de los usuarios de la aplicación. Además, se debe poder almacenar el progreso de cada usuario en cada juego mediante el almacenamiento de los puntos que obtenga en una partida de un juego en concreto, y la fecha en la que ha obtenido esa puntuación.
- **Exergames:** La aplicación cuenta con una lista de videojuegos, de los cuales se quiere saber el nombre, la descripción, la dificultad, instrucciones y el problema cervical asociado. Esto último se debe a que cada videojuego está diseñado para poder usarse como terapia de algún problema cervical, de forma que cada uno de estos juegos está estrechamente relacionado con un problema cervical concreto.
- **Problemas cervicales:** Como consecuencia del punto anterior, se debe tener constancia de qué problemas cervicales se pueden tratar mediante los videojuegos disponibles en la aplicación. De cada problema cervical se debe saber el nombre.

Teniendo esto en cuenta, ya se puede diseñar un modelo de datos que ilustre cómo se relaciona esta información en la base de datos. Sin embargo, se pretende crear un entorno social que fomente la motivación de los pacientes a realizar sus ejercicios de terapia, y esto requerirá el almacenamiento de información adicional. Por ello, el modelo de datos se diseñará cuando se conozca la totalidad de la información a almacenar.

5.1.2 Entorno social motivacional

Como se ha mencionado anteriormente, la aplicación de exergaming no solamente pretende permitir a los pacientes realizar ejercicios terapéuticos de forma más amena e interactiva, sino que además cuenta con un sistema de puntuaciones que favorece la creación de un entorno social que fomente la motivación de los usuarios a realizar sus ejercicios de rehabilitación. Como consecuencia, se pretende que esto ayude también a mejorar la adherencia al tratamiento.

Este sistema de puntuaciones consiste en el uso de niveles y puntos de experiencia, junto con la posibilidad de seguir y dejar de seguir a usuarios.

Todos los exergames tienen una forma de puntuar el desempeño de los usuarios en el juego y, por tanto, en el ejercicio de rehabilitación. Por ejemplo, en el juego de la serpiente, éste consiste en hacer que una serpiente coma cuanta más fruta mejor. Cada vez que la serpiente come una fruta, el usuario suma 100 puntos y la serpiente crece, de forma que cada vez es más complicado manejar la serpiente dado que el espacio es limitado. El objetivo es sumar todos los puntos posibles y hacer que la serpiente crezca cuanto más, mejor.

Cuando un usuario juegue a un exergame, tras cada partida, las puntuaciones que este usuario obtenga en el juego no servirán solamente para evaluar el desempeño del paciente en el ejercicio, sino que también servirán para crear el entorno social que fomenta la adherencia al tratamiento. Esto se consigue de la siguiente manera:

- **Puntos de experiencia:** Los puntos que un usuario obtenga en un juego se contarán como “Puntos de experiencia”. Estos se van sumando a cada usuario en función de su desempeño. Por ejemplo, si un usuario obtiene 300 puntos en un juego, se le sumarán 300 puntos de experiencia. Un usuario podrá tener como máximo 1000 puntos de experiencia.
- **Niveles:** Cuando un usuario alcance los 1000 puntos de experiencia, subirá 1 nivel y los puntos de experiencia volverán a ser 0. De esta forma, cada 1000 puntos de experiencia los usuarios subirán de nivel y los puntos de experiencia se restablecerán.
- **Seguidores:** La aplicación de exergaming cuenta con un buscador de usuarios que permite encontrar a otros usuarios de la aplicación y comprobar de qué nivel son y cuantos puntos de experiencia tienen. Se podrá seguir y dejar de seguir a usuarios.

Cabe mencionar que un usuario que acaba de crear su perfil y aún no ha jugado a ningún exergame, será un usuario de nivel 0 y con 0 puntos de experiencia.

Esto facilita la creación de un entorno social que fomente la competitividad entre los usuarios. Teniendo en cuenta que para subir de nivel se debe jugar a los exergames y, por tanto, hacer rehabilitación, el sistema de niveles y puntos de experiencia actúa como sistema de gamificación que premia mediante niveles o rangos la constancia de los usuarios con su tratamiento. De forma opcional y a modo de propuesta, se podría implementar un sistema de recompensas que entregue premios a los pacientes al alcanzar ciertos niveles o conseguir algunos hitos.

Además, se puede buscar a otros compañeros o pacientes en la aplicación y comprobar de qué nivel son y cuantos puntos de experiencia deben conseguir para subir de nivel. Esto se hace para unir a todos los usuarios en un mismo espacio, de forma que no solamente se cultiva la competitividad, sino que también se fomenta el compañerismo y la inclusión social entre los pacientes, ya que se sabe que muchos de ellos pueden sentirse solos durante el proceso de rehabilitación.

Todo esto implica que también se debe almacenar en la base de datos la siguiente información:

- Nivel y puntos de experiencia de cada usuario.
- Seguidores de cada usuario.

5.1.3 Creación de una base de datos relacional

Una vez se sabe la información que se debe almacenar, se puede diseñar un modelo conceptual que ilustre cómo se relaciona la información en la base de datos. El modelo de datos planteado es el que se muestra en la [figura 11](#).

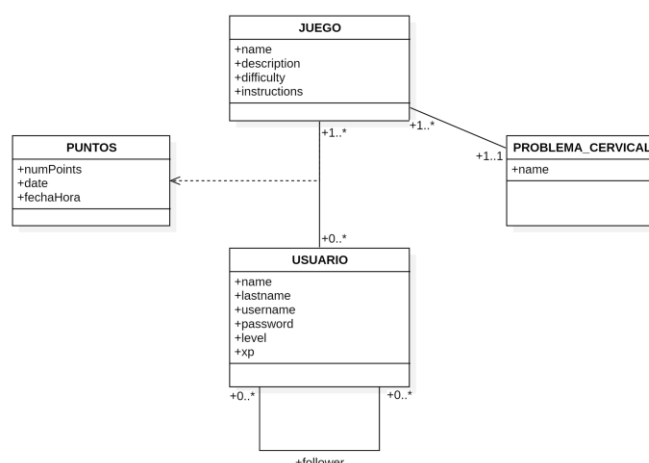


Figura 11. Modelo conceptual de datos para la base de datos de la aplicación de exergames.

Este modelo todavía necesita ser convertido en un modelo relacional y normalizado hasta la mayor forma normal posible. Para consultar el modelo final implementado tras el paso a relacional y la normalización, consultar el [anexo A.2](#).

Una vez pasado a modelo relacional y normalizado, se detalla el código SQL que permite la creación de la base de datos y de las tablas resultantes que esta contendrá. Este código se encuentra en el repositorio del proyecto [\[38\]](#). Para crear la base de datos, se usará el sistema gestor de base de datos de PostgreSQL, junto con las herramientas de pgAdmin 4 y SQL Shell.

Una vez creada la base de datos, se ha generado un conjunto de datos para insertar en ella, de forma que haya usuarios realistas en la aplicación y se puedan hacer pruebas con ella. Este

conjunto de datos incluye una cantidad aproximada de 10.000 usuarios, todos ellos con registros de partidas jugadas en ciertos exergames, con niveles y puntos de experiencia distintos, y con seguidores diferentes. Esto se ha hecho implementando un script usando Python como lenguaje, y permite la creación de un dataset suficientemente relacionado como para que las pruebas que se hagan en un futuro en la aplicación puedan ser consideradas realistas. El código de este script también se encuentra en el repositorio del proyecto [\[39\]](#).

5.1.4 Creación de usuarios desde la aplicación

Dado que la creación de usuarios en la base de datos ya está operativa y funcional desde las herramientas que proporciona el gestor de base de datos, se puede comenzar con la implementación de las pantallas que permitan la creación de usuarios desde la propia aplicación de exergames.

Para ello, se ha diseñado un formulario que permite a los usuarios crear una cuenta o perfil introduciendo sus datos. Cuando los usuarios pulsen el botón de crear cuenta, esta información se validará y, en caso de ser correcta, se insertará en la tabla de usuarios de la base de datos, creando de esta manera un usuario.

Adicionalmente, para aumentar la seguridad relacionada con el uso de contraseñas en la aplicación, se ha habilitado un botón que permite generar una contraseña segura. Al pulsar este botón, se genera una cadena aleatoria de letras mayúsculas, minúsculas, números y otros caracteres como barras o guiones. Esta cadena se establecerá automáticamente en el campo de contraseña para que no lo tenga que hacer el usuario, y dado su carácter aleatorio, muy probablemente será una contraseña más segura que alguna contraseña típica introducida por un usuario.

La [figura 12](#) muestra la pantalla habilitada para la creación de un perfil y la generación de una contraseña segura aleatoria.

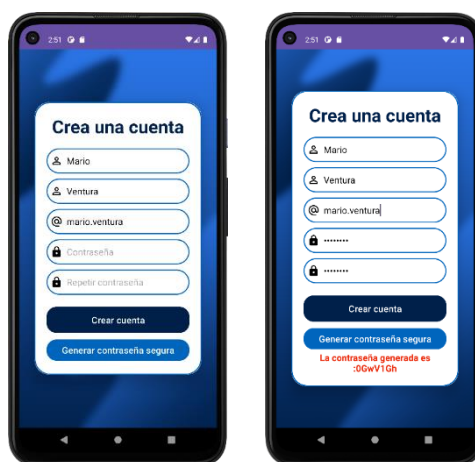


Figura 12. Creación de usuarios en la aplicación y generación de contraseñas seguras.

5.2 Autenticación de usuarios

Una vez que los usuarios se hayan creado una cuenta, deben poder usarla para iniciar sesión en la aplicación y realizar sus ejercicios de rehabilitación mediante juegos. Por tanto, es necesaria la implementación de una pantalla que permita la autenticación de usuarios mediante nombre de usuario y contraseña.

Esta pantalla será la primera que aparezca al abrir la aplicación, y contará con un botón para crear una cuenta en caso de que el usuario no tenga una. Al pulsar este botón, se redirigirá al usuario a la pantalla mostrada en la [figura 12](#).

Para autenticarse, un usuario introducirá su usuario y contraseña, y la lógica de autenticación será la típica de una autenticación con usuario y contraseña. Esta consiste en comprobar que existe un usuario en la base de datos cuyo nombre de usuario es el introducido, y cuya contraseña coincide para ese nombre de usuario.

El diseño planteado para la autenticación de usuarios el mostrado en la [figura 13](#).



Figura 13. Autenticación de usuarios para iniciar sesión en la aplicación.

5.3 Petición y concesión de permisos

Cuando los usuarios inicien sesión podrán acceder a la pantalla principal de la aplicación, donde existe una lista de exergames disponibles para jugar. Sin embargo, es evidente que no se podrá jugar a estos exergames si no se puede acceder a la cámara frontal del dispositivo, ya que el seguimiento facial no estaría disponible. Se debe pedir al usuario permisos de acceso a la cámara.

Por ello, la gestión de los permisos es crucial no solo para obtener los permisos adecuados en tiempo de ejecución, sino también para gestionar las respuestas del usuario a estas solicitudes de permisos de manera que la experiencia del usuario sea fluida.

Se debe poder acceder a la cámara del dispositivo Android del paciente y grabar vídeo, al menos mientras se esté usando la app de exergaming. Esto se debe a que los exergames se juegan mediante movimientos del rostro, que son detectados mediante variaciones significativas en las coordenadas de ciertos puntos faciales del usuario. Es vital, por tanto, que se pueda acceder a la cámara del dispositivo, ya que de lo contrario no se podrá ver la cara del usuario y, como consecuencia, no se podrá jugar a ningún exergame al no poder detectar las coordenadas de los puntos faciales clave.

Para poder solicitar al usuario permisos de acceso a la cámara del dispositivo, será necesario realizar ciertos pasos específicos tanto en los archivos de configuración como en el código de la app.

5.3.1 Obtención de permisos necesarios para la aplicación

La obtención de permisos de acceso a la cámara del dispositivo Android requiere de dos pasos, descritos con detalle a continuación.

1. **Informar al sistema operativo.** En primer lugar, debe informarse al sistema operativo de la necesidad de usar este permiso concreto. Como ya se ha explicado con anterioridad, para la aplicación es imprescindible el uso de la cámara, de forma que, junto con la petición del permiso, se debe especificar que este permiso es de carácter obligatorio y la aplicación no puede funcionar sin él.
2. **Solicitar permiso en tiempo de ejecución.** Una vez se ha informado al sistema operativo de la necesidad del uso de la cámara del dispositivo, se debe solicitar al usuario el acceso a esta.

Una consideración a tener cuenta es que, para solicitar permiso de acceso a la cámara, para versiones de Android estrictamente inferiores a la versión **Android 6.0** (API nivel 23), este permiso ya está garantizado de forma predeterminada. Esta versión es la llamada *Android Marshmallow*, que salió al mercado a partir del 5 de Octubre del 2015 [\[40\]](#). Desde entonces, debido a la creciente preocupación en términos de seguridad y privacidad, este permiso debe ser solicitado explícitamente al usuario y aceptado por él mismo. Además, por razones descritas en el [apartado 5.8](#), también será necesario el permiso para enviar notificaciones.

Teniendo en cuenta que actualmente ya existen dispositivos con versiones Android 13 (API nivel 33), la solicitud de este permiso se llevará a cabo de forma distinta en función del nivel de API o de versión de Android del dispositivo.

El primer paso para la solicitud del permiso en tiempo de ejecución será, por tanto, conocer la versión de Android del dispositivo en cuestión. Si la versión de Android es inferior a Android 6.0, no será necesario solicitar el permiso ya que este ya está otorgado de forma predeterminada. De lo contrario, si el dispositivo usa una versión de Android superior o igual a la 6.0 (opción más probable teniendo en cuenta que esta versión de Android fue lanzada hace 9 años y que los sistemas operativos tienen la capacidad de actualizarse), se tendrá que pedir al usuario que conceda el permiso específicamente para la cámara cuando la aplicación esté en uso, generalmente justo antes de realizar alguna acción que requiera el uso de esta.

Para las comprobaciones sobre la versión de Android del dispositivo, se ha optado por comprobar el nivel de la API ya que se considera una comprobación más precisa. Si la API es nivel M (nivel 23, Android 6.0) o superior, se pide el permiso explícitamente al usuario.

Cabe mencionar que, como buena práctica, podría informarse a usuarios con versiones de Android inferiores a 6.0 de que va a usarse la cámara de su dispositivo. Aunque esta comunicación no sea estrictamente necesaria ya que el permiso está garantizado de forma predeterminada, es importante considerar que la transparencia siempre beneficia la confianza del usuario. Aunque técnicamente los permisos en estas versiones se conceden al instalar la aplicación y no es necesario solicitarlos en tiempo de ejecución, informar a los usuarios sobre cómo se usan sus datos o el hardware de su dispositivo siempre es una buena práctica.

Esta comunicación podría realizarse mediante la interfaz de usuario de la aplicación, como parte del flujo de introducción o, alternativamente, en secciones típicas como “*Configuración*” o “*Ayuda*”. Esto aseguraría que todos los usuarios, independientemente de la versión de Android de la que dispongan en sus dispositivos, entiendan cómo y por qué su aplicación requiere el acceso a ciertos recursos del sistema tales como la cámara.

5.3.2 Gestión de la respuesta del usuario

Tras solicitar al usuario el permiso de acceso a la cámara, es importante gestionar las posibles respuestas que este nos pueda dar. En este caso, se está tratando con una pregunta del tipo “Sí o No”, de forma que solo hay dos posibles respuestas por parte del usuario y, por tanto, dos posibles casos a tratar.

1. **Caso 1: El usuario concede el permiso para acceder a la cámara.** Si el usuario concede los permisos, simplemente se podrá acceder a la cámara del dispositivo y proceder con la acción que requería dicho permiso. En este caso, se podrá acceder a los exergames y jugar a ellos.
2. **Caso 2. El usuario deniega el permiso para acceder a la cámara.** En caso de que el permiso sea denegado se deberá informar al usuario de que sin este permiso no se puede

acceder a los exergames que permiten comenzar con las actividades terapéuticas de rehabilitación. Esto puede hacerse mediante una pequeña ventana emergente con el mensaje de advertencia correspondiente, y con un botón “Aceptar” para cerrar esta ventana. Podría implementarse de forma que, cuando se pulse el botón “Aceptar”, se abra una nueva ventana que pregunte una vez más por los permisos de acceso a la cámara bajo un mensaje del tipo “*¿Estás seguro? Este permiso es necesario para el funcionamiento básico de la aplicación*”. Si el usuario vuelve a denegar el permiso, los exergames no podrán usarse y la aplicación no debe permitir usarlos.

5.4 Seguimiento facial en tiempo real

Para implementar correctamente el seguimiento facial en tiempo real que permite a los usuarios de la aplicación jugar correctamente a los exergames, como primer paso se debe conectar la aplicación con el framework de Google que permite estas funcionalidades.

5.4.1 Conexión con el framework de Google

La conexión de la aplicación al framework de Google es esencial para asegurar el éxito en el desarrollo de esta, ya que, sin las funcionalidades que este framework presenta, no se pueden obtener las coordenadas en tiempo real de ciertos puntos faciales y, por tanto, no se podría jugar a ninguno de los exergames propuestos.

En el proyecto de Android Studio, debe asegurarse la inclusión de la configuración necesaria de la aplicación para gestionar las dependencias de la aplicación Android, de forma que se pueda importar todas las bibliotecas pertinentes del ML Kit ofrecido por Google. Esto se hará en el bloque de dependencias, donde se detallan bibliotecas externas que la aplicación necesita para funcionar correctamente. Se añadirán las dependencias al framework de Google siguiendo esta metodología, consiguiendo que el sistema de automatización de compilación de Android Studio descargue e integre estas dependencias en el proyecto, haciendo posible su uso en el código. Esto permite la conexión de la aplicación y el framework de ML Kit de Google, facilitando la incorporación de funcionalidades de Machine Learning (ML de ahora en adelante) en la aplicación y permitiendo, en este caso, la detección de rostros o el etiquetado de imágenes mediante el reconocimiento de objetos, lugares, personas, etc.

Se seguirá este razonamiento para añadir todas las dependencias que se consideren necesarias.

5.4.2 Primeras pruebas con ML Kit

Como paso previo al desarrollo de la aplicación final de exergaming, y como parte del proceso de aprendizaje realizado en el proceso de implementación de dicha aplicación, se ha implementado una pequeña aplicación Android que hace uso del ML Kit de Google. Esta aplicación recibirá el nombre de **app1** de ahora en adelante, y permite capturar imágenes y detectar objetos en ellas.

Pese a que no es de gran utilidad en el contexto de creación de una aplicación de exergaming, la aplicación **app1** permite hacer pruebas para conectarse a la cámara frontal del dispositivo Android

y probar las funcionalidades propias de ML Kit. El planteamiento es el siguiente: crear una aplicación para dispositivos Android que sea capaz de reconocer y etiquetar objetos, personas, animales, etc. en imágenes mediante la funcionalidad de etiquetado de imágenes que presenta el framework de Google con el que se trabaja. Para llevar a cabo esta tarea, se ha creado un proyecto en Android Studio que ya cuenta con los archivos y paquetes necesarios para trabajar. Dado que el objetivo es familiarizarse con el framework de Google, su funcionamiento y limitaciones, el diseño de la aplicación será simple, permitiendo que los esfuerzos se centren en la comprensión de los retos asociados al uso de ML Kit. El diseño de la aplicación app1 es el mostrado en la [figura 14](#).



Figura 14. interfaz de usuario de la aplicación app1.

Tal y como muestran las figuras anteriores, la aplicación tiene un botón para iniciar la cámara y tomar una foto. Una vez se tome una foto, esta aparecerá en la pantalla junto con los resultados obtenidos tras el proceso de análisis de esta imagen.

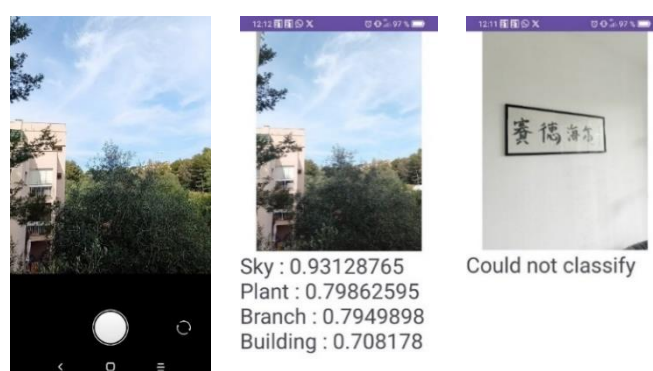


Figura 15. Pruebas y resultados de la clasificación de imágenes con la aplicación app1.

Como puede verse en las imágenes anteriores, la detección y clasificación de objetos, animales, personas y demás, parece funcionar como se pretendía al plantear la aplicación. La [figura 15](#)

muestra lo que el usuario ve cuando pulsa el botón “*Start Camera*”, tras aceptar que la aplicación pueda acceder a la cámara y pudiendo usarla para tomar cualquier fotografía.

Por otro lado, la segunda pantalla de la [figura 15](#) muestra la imagen tomada y todos los objetos identificados en ella por parte de la funcionalidad “*image labelling*” de ML Kit. Estos objetos identificados aparecen junto a la precisión de la clasificación, expresada a modo de probabilidad (por ejemplo, se detecta el cielo con una probabilidad de acierto de 0.9312, de forma que se puede asegurar con un porcentaje de acierto del 93.12% que la detección del cielo es correcta).

Además, puede apreciarse que los objetos etiquetados aparecen ordenados de forma descendente en cuanto a precisión, y que no aparecen objetos con una precisión menor a 0.7, mínimo de precisión establecido anteriormente.

Por último, la captura de la derecha de la [figura 15](#) muestra el escenario en el cual no se consigue identificar o etiquetar ningún objeto con la precisión mínima establecida.

Puede comprobarse, por tanto, que la implementación de la aplicación app1 ha sido exitosa, y se considera que los desafíos enfrentados para implementarla han sido de utilidad para la posterior implementación de la aplicación de exergaming.

5.4.3 Detección estática de coordenadas de puntos faciales clave

Para desarrollar un algoritmo que, mediante la cámara frontal, permita la detección en tiempo real de rostros y puntos faciales en ellos, primeramente, y como parte del proceso de aprendizaje, se ha desarrollado un algoritmo que permite la detección de coordenadas de puntos faciales clave en imágenes tomadas con la cámara. Esto permite probar y familiarizarse con las funcionalidades de detección de rostro de ML Kit sin afrontar de forma prematura los desafíos asociados a las necesidades de detección en tiempo real.

Como consideración previa, se debe tener en cuenta que para detectar rostros las imágenes de entrada deben ser de un tamaño mínimo de 480x360 píxeles [\[42\]](#). De lo contrario, la detección de rostros no funcionará. Además, en una imagen de cualquier tamaño, el rostro debe tener unas dimensiones mínimas de 100x100 píxeles como norma general, aunque en el caso concreto de querer detectar el contorno del rostro, estos rostros deben tener unas dimensiones mínimas de 200x200 píxeles.

Para la detección de coordenadas será necesario usar detector de rostros propio de ML Kit. En primer lugar, se crea una configuración de detección, que puede considerar distintos parámetros para obtener un mayor rendimiento dependiendo del objetivo de su uso. Posteriormente, se creará un detector de rostros usando la configuración creada previamente. Este detector procesará una imagen obtenida con la cámara, buscando rostros y creando una lista con todos los que encuentre, en caso de encontrarlos.

Tras esto, se obtendrá una lista que contiene todos los rostros detectados en la imagen. En caso de que la lista tenga un tamaño o longitud mayor a 0 significa que se ha encontrado al menos un rostro. Se hará un recorrido de los rostros de la lista y, para cada rostro contenido en ella, se tratará de obtener todos los puntos faciales clave, generando una lista de puntos faciales.

El framework de Google permite que, para cada punto facial, se pueda obtener su coordenada X e Y con una llamada a métodos propios.

La [figura 16](#) muestra un diagrama del planteamiento seguido desde que se obtiene una imagen como input hasta que se obtienen las coordenadas de cada punto facial detectado en cada rostro encontrado.

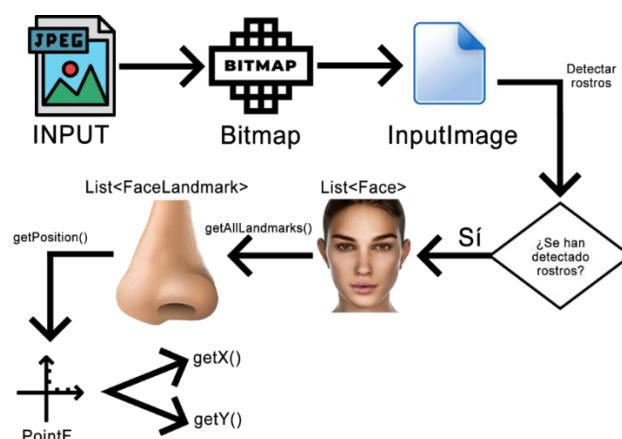


Figura 16 que muestra el diagrama de flujo seguido desde que se captura una imagen con el dispositivo hasta que se obtiene las coordenadas de la nariz en un rostro.

ML Kit presenta un total de diez tipos de puntos faciales clave (ojo izquierdo, ojo derecho, oreja derecha, parte superior de la boca, etc.), de forma que para cada uno de ellos encontrado, solo si es la nariz, se invocará a las funcionalidades que permiten obtener un objeto que represente las coordenadas.

Estas coordenadas se representan en formato (X,Y), permitiendo conocer exactamente la posición de la nariz en la imagen. El valor de estas coordenadas será el que se mostrará en la pantalla para que el usuario compruebe si las coordenadas de la nariz se han detectado correctamente.

La [figura 17](#) ilustra cual es el valor de X e Y en las coordenadas detectadas para la nariz en un rostro de una imagen.



Figura 17. Representación de coordenadas de una nariz encontradas en un rostro.

De esta forma, partiendo de una imagen que contenga un rostro o selfi cualquiera, la aplicación seguirá la lógica descrita, en busca del reconocimiento de la nariz de los rostros detectados (en caso de haber). Posteriormente, mostrará las coordenadas en las que se encuentra en la imagen, dándoles valor a X e Y asumiendo que el punto (0,0) es origen de las coordenadas. Este punto está situado en la esquina superior izquierda de la pantalla, e implica que las coordenadas X e Y tienen un valor de 0.

Para comprobar que el funcionamiento del planteamiento implementado sea correcto, se ha habilitado un apartado provisional en la aplicación de exergaming (no estará disponible en la versión final), donde se puede probar el planteamiento y ver sus resultados.

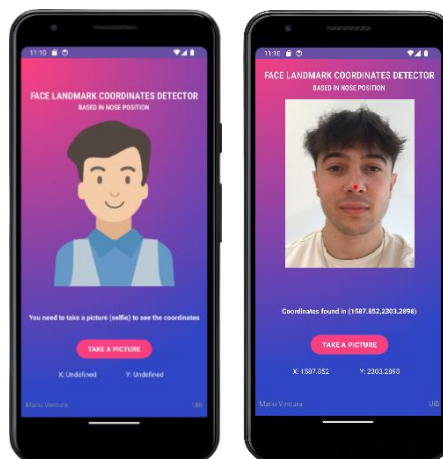


Figura 18. Coordenadas de la nariz detectadas estáticamente en un rostro.

Al iniciar la aplicación, la primera pantalla muestra el título "*Face Landmark Coordinates Detector Based in Nose Position*" en la parte superior, seguido de una ilustración de un hombre sonriente con un mensaje que invita al usuario a tomar una selfie para visualizar las coordenadas.

Un botón central etiquetado como "TAKE A PICTURE" permite capturar la foto necesaria. Las coordenadas, inicialmente indefinidas, se muestran como "X: Undefined" y "Y: Undefined".

La segunda pantalla muestra el resultado tras tomar una foto, incluyendo la imagen del usuario y las coordenadas específicas de la nariz identificadas, como "*Coordinates found in (1587.852,2303.2898)*" y actualiza los campos de coordenadas respectivos a "X: 1587.852" y "Y: 2303.2898".

Además, para demostrar que las coordenadas detectadas son, efectivamente, de la nariz; aparece un punto rojo dibujado justo encima de las coordenadas detectadas. Este punto rojo tiene un diámetro de 10 píxeles y recibe el nombre de "Graphic Overlay" (GO de ahora en adelante). El GO permite obtener una comprobación rápida de si las coordenadas detectadas realmente son las de la nariz, o del nivel de precisión de la detección.

5.4.4 Detección dinámica de coordenadas de puntos faciales clave

Una vez se ha entendido el planteamiento que se debe seguir para detectar coordenadas de puntos faciales con ML Kit sobre un input (hasta ahora una imagen), el siguiente paso es implementar la detección en tiempo real. Esta capacidad es fundamental para que la aplicación responda de manera dinámica a los movimientos del usuario, permitiendo una interacción fluida y continua con los juegos diseñados para la terapia.

Para lograr la detección en tiempo real, se utiliza el mismo ML Kit de Google, pero con una diferencia significativa: el input. En este caso se debe configurar para procesar flujos de video en lugar de imágenes estáticas. Esto es de vital importancia en muchas aplicaciones modernas, especialmente en aquellas que requieren interacción en tiempo real y monitoreo continuo. En este contexto de rehabilitación mediante exergames, donde la dinámica de los movimientos del usuario debe ser capturada y respondida instantáneamente para garantizar la efectividad del tratamiento y la interacción con la aplicación, esto es particularmente crítico.

La razón principal por la cual se hace esto es por el rendimiento de la aplicación. Si se detectaran las coordenadas en imágenes estáticas, deberían tomarse varias fotografías por segundo y detectar las coordenadas en todas ellas, haciendo que la ejecución de la aplicación sea muy costosa. Además, si no se gestionara correctamente el tratamiento de estas imágenes, la galería del dispositivo Android se llenaría de imágenes al usar la app, a no ser que tras detectar las coordenadas se eliminara la imagen automáticamente, cosa que truncaría todavía más el rendimiento de la app.

Una de las principales razones de peso por las que se deben procesar flujos de vídeo en lugar de imágenes estáticas es la interacción en tiempo real. Para la aplicación es crucial capturar y responder a los movimientos del usuario en tiempo real, y para ello, el procesamiento de video permite analizar y responder a los cambios dinámicos y continuos en la posición y orientación del usuario, lo que es fundamental para ajustar los ejercicios y juegos en el momento exacto y de manera precisa.

El primer desafío por afrontar para implementar la interacción en tiempo real es comprender cómo obtener el input necesario para después hacer el procesamiento y obtener rostros. Para obtener un input sin capturar imágenes se ha optado por usar un *PreviewView*. Un *PreviewView* es la vista previa en vivo que muestra lo que la cámara está capturando en tiempo real. Esencialmente, es una interfaz gráfica que muestra el video o las imágenes que están siendo captadas por la cámara del dispositivo antes de tomar una fotografía o grabar un video.

La implementación de este tipo de previsualización implica configurar la cámara del dispositivo, definir el tamaño y la orientación de la previsualización y asegurarse de que los datos de la cámara se procesen correctamente para su visualización. Para ello, tradicionalmente, en Android se hacía a través de un framework llamado *Camera2*. Sin embargo, actualmente existe *CameraX*, otro framework más moderno y completo que simplifica mucho la tarea de codificación necesaria para manejar visualizaciones previas complejas y operaciones de cámara.

En este caso, se ha optado por usar *CameraX* para la obtención de una previsualización que nos permita saber qué está viendo la cámara en todo momento y poder procesarlo.

La aplicación contiene un bloque de código que actúa como núcleo del procesamiento de video para la detección facial en tiempo real. Este bloque es responsable de analizar los fotogramas de video capturados por la cámara del dispositivo, identificando y extrayendo las coordenadas de los puntos faciales clave. La implementación se realiza utilizando ML Kit, que proporciona herramientas robustas para la detección de características faciales en flujos de vídeo.

En primer lugar, antes de aplicar la detección de rostros en imágenes, tal y como se recomienda en la propia página de ML Kit [\[41\]](#), se ha aprovechado las posibles opciones que presenta ML Kit para crear un objeto que modifique la configuración predeterminada del detector de rostros.

Esto permitirá crear una configuración para el detector de rostros. Teniendo en cuenta que se pretende realizar un procesamiento de imágenes en tiempo real, se ha optado por aplicar una configuración que priorice la velocidad a la precisión en la detección de rostros.

Una vez creada la configuración con la que se quiere detectar el rostro del usuario, se crea el propio detector de rostros usando esta configuración. Posteriormente se vincula este detector al ciclo de vida de la cámara y se añade un listener (oyente de eventos) que permita obtener sus previsualizaciones y procesarlas.

Este listener obtiene la previsualización definida en la aplicación, y configura la vista previa de la cámara para mostrarla sobre esta previsualización, indicándole que está preparado para recibir información.

Para asegurar que el análisis de imágenes de la aplicación puede seguir el ritmo de los requisitos de velocidad de fotogramas de *CameraX*, se configura el analizador de imágenes de forma que no cause bloqueos. Esto se hace aplicando una estrategia llamada “*keep only latest*”, que asegura que

una vez se analiza una imagen, esta se borra y no se almacena, de forma que solamente se almacena en caché la última imagen recibida. Esto permite que, si se recibe una imagen nueva antes de que la aplicación termine de procesar la anterior, la nueva imagen reemplaza a la anterior guardándose en el mismo búfer que esta [\[42\]](#).

Cabe mencionar que, para recibir y analizar imágenes, se crea un nuevo hilo simple de ejecución. Esto permite controlar la concurrencia, hacer un seguimiento de tareas asíncronas y gestionar métodos que pueden producir un *Future*, es decir, un resultado producido tras la finalización de una tarea asíncrona.

Por último, debe considerarse la orientación que tendrá el dispositivo cuando los usuarios usen la aplicación, ya que para poder realizar la detección de coordenadas en tiempo real se debe usar la cámara frontal, que es la que apuntará al rostro del usuario. Se seleccionará esta cámara explícitamente para asegurar que sea esta la que se use en todo momento en la aplicación para la detección de rostros.

Ahora ya se sabe qué cámara usar, se ha creado y configurado un analizador de rostros, se ha vinculado al ciclo de vida de la cámara, y se ha definido una forma de tratar la concurrencia y gestionar los posibles bloqueos y la obtención de resultados de tareas asíncronas durante el proceso de análisis de imágenes en tiempo real. En este punto, se puede dar paso al procesamiento de imágenes.

5.4.5 Algoritmo para procesamiento de imágenes en tiempo real en búsqueda de coordenadas de puntos faciales.

Una vez se carga la aplicación de exergaming, se puede acceder a una sección llamada “Probar tracker”, que contiene una vista previa de lo que ve la cámara. Esta muestra lo que ve la cámara en tiempo real, tal y como se vería si se pretendiese sacar una foto con la propia cámara del dispositivo. Bajo esta previsualización, aparece un texto que muestra el valor de las coordenadas faciales detectadas, junto con otra información como los movimientos detectados, entre otros.

En el código destinado al procesamiento de imágenes en tiempo real, se ha añadido un bloque específico para actualizar periódicamente este texto. Sin embargo, esto se hace en un hilo de ejecución por separado, ya que el procesamiento de imágenes y la actualización del texto en la UI son dos tareas distintas y, por tanto, son tratadas en hilos distintos con propósitos diferentes. La actualización de la UI se hará en un hilo de ejecución específico y destinado para este.

El algoritmo implementado para la detección de rostros en tiempo real sigue el planteamiento descrito a continuación.

```

var coordenada;
detectarRostrosEnTiempoReal(prevImagen) {
    imagen = obtenerImagen(prevImagen);
    listaRostros = analizar(imagen);
    IF (listaRostros.longitud > 0)
        FOR (rostro IN listaRostros)
            var coordenadaX = rostro.NarizX;
            var coordenadaY = rostro.NarizY;
            coordenada = (coordenadaX, coordenadaY);
            actualizarUI(coordenada);
            return coordenada;
        ENDFOR
    ENDIF
}

```

5.4.6 Detección de variaciones significativas en coordenadas faciales

Una vez se consigue detectar las coordenadas de la nariz en rostros de usuarios en tiempo real, se debe saber cómo usar estas coordenadas para convertirlas en controles útiles para los exergames. Este paso es esencial para interpretar los movimientos del rostro en tiempo real y tratarlos como controles de juego.

Exergames es una aplicación interactiva destinada a la rehabilitación cervical o el entretenimiento mediante videojuegos (exergames) a los que se juega usando tecnologías de seguimiento facial. Por ello, es necesario saber interpretar el valor de X e Y en las coordenadas detectadas, y entender qué significan las variaciones en sus valores.

La capacidad para detectar y responder a las variaciones significativas en las coordenadas faciales es crucial para garantizar que el proceso de entrenamiento y/o rehabilitación sea óptimo y asegurar una experiencia de usuario calidad. Además, también se medirá la velocidad con la que un usuario realiza un movimiento con el rostro.

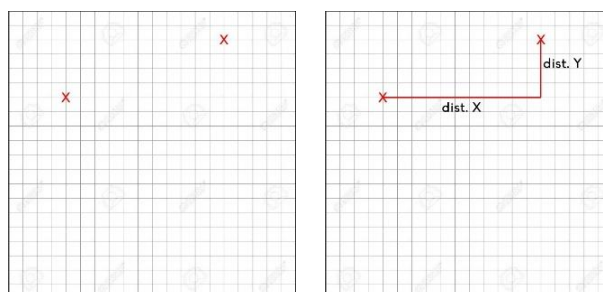


Figura 19. Dos puntos y su distancia en el eje X e Y sobre un mismo plano.

Las variaciones en las distancias entre distintas coordenadas, es decir, la diferencia de sus valores X e Y, se usarán para poder desarrollar controles útiles que permitan jugar a los exergames.

Una vez obtenida una coordenada, se puede saber la distancia de esta respecto a la última coordenada detectada mediante operaciones básicas como obtener las diferencias de sus valores en el eje X e Y. Se evidencia, por tanto, la relevancia de almacenar siempre las coordenadas anteriormente detectadas, puesto que sin ellas resulta imposible detectar los movimientos que hace el usuario con el rostro.

Sin embargo, es importante almacenar solamente el mínimo indispensable para poder llevar a cabo el cálculo, ya que de lo contrario resultaría en un programa lento e ineficiente. Desde un punto de vista de rendimiento y uso de recursos, almacenar únicamente las coordenadas necesarias para el cálculo actual maximiza la eficiencia. Las aplicaciones, especialmente las que operan en tiempo real, necesitan ser lo más ligeras y rápidas posible. Almacenar datos adicionales que no se utilizan para cálculos inmediatos ralentizaría el proceso y aumentaría el consumo de memoria innecesariamente.

Para obtener la distancia entre los valores X e Y de cada coordenada detectada con respecto a la coordenada anterior, se ha usado un planteamiento como el descrito a continuación.

pseudocódigo

```
var primeraVez = 1;
var coordAnt = NULL;
var distancia = 0;
programaEnTiempoReal {
    coordAct = detectarCoordenada
    IF (primeraVez == 1)
        coordAnt = coordAct;
        primeraVez = 0;
        distancia = 0;
    ELSE
        distancia = calcularDistancia(coordAct, coordAnt);
        coordAnt = coordAct;
    ENDIF
    return distancia;
}
```

Donde *coordAnt* y *coordAct* representan un par de valores (X, Y) para las coordenadas anterior y actual respectivamente.

Además, dada la precisión de la detección de coordenadas ([figura 18](#)), puede resultar de gran utilidad establecer un umbral de sensibilidad para diferenciar entre movimientos intencionales y movimientos pequeños o involuntarios. A este umbral se le llamará “*sensitivity threshold*” o

umbral de sensibilidad, y permitirá ignorar movimientos indeseados por parte de los usuarios, de forma que estos no trunquen la experiencia de usuario al usar la aplicación ni dificulten el proceso de rehabilitación.

Por defecto, se establecerá el umbral de sensibilidad en **25 píxeles**, de forma que los movimientos que los usuarios hagan con la cabeza que impliquen que las coordenadas anterior y actual tienen una separación inferior a esta cantidad, serán ignorados al considerarse involuntarios.

5.4.7 Interpretación de variaciones en coordenadas y conversión en controles útiles para exergames

Para interpretar y usar correctamente las variaciones de los valores X e Y de las coordenadas detectadas, se parte del punto de que se sabe que, dado que los exergames son videojuegos en 2 dimensiones, solo existen cuatro tipos de movimientos simples que el usuario puede realizar: mover la cabeza hacia la izquierda, moverla hacia la derecha, hacia arriba o hacia abajo.

Es evidente que también se pueden hacer movimientos más complejos como mover la cabeza hacia arriba y a la derecha simultáneamente, pero estos movimientos surgen de la composición de dos (y solamente dos) movimientos simples. Partiendo de esta idea inicial, se debe saber interpretar a la perfección cuando un usuario está haciendo cada uno de estos movimientos, y esto debe hacerse solamente usando las coordenadas.

Para ello, en primer lugar, se calcula la diferencia de valores en el eje X, y posteriormente en el eje Y. En función de los resultados obtenidos se podrá deducir el movimiento que el usuario ha realizado.

Dadas una coordenada ACT(X_{ACT} , Y_{ACT}) y una coordenada ANT(X_{ANT} , Y_{ANT}), se sabe que:

- El usuario ha movido la cabeza a la derecha si $X_{ACT} - X_{ANT} > 0$.
- El usuario ha movido la cabeza a la izquierda si $X_{ACT} - X_{ANT} < 0$.
- El usuario no ha movido la cabeza en el eje X si $X_{ACT} = X_{ANT}$.

Lo mismo sucede en el eje Y, donde se sabe que:

- El usuario ha movido la cabeza hacia abajo si $Y_{ACT} - Y_{ANT} > 0$.
- El usuario ha movido la cabeza hacia arriba si $Y_{ACT} - Y_{ANT} < 0$.
- El usuario no ha movido la cabeza en el eje Y si $Y_{ACT} = Y_{ANT}$.

Por tanto, cada exergame usará esta lógica de una forma específica dado el propósito del juego, ya sea para mover un personaje a la derecha, izquierda, arriba o abajo, o realizar alguna otra acción dentro del juego.

Respecto al proceso seguido para la interpretación de la velocidad con la que el usuario ha realizado el gesto y las posibles aplicaciones que esto podría tener en un contexto de exergaming, la explicación detallada se encuentra en el [anexo A.3](#).

5.5 Implementación de exergames o videojuegos serios

En cuanto se sabe cómo detectar movimientos con el rostro de los usuarios, se puede comenzar con la implementación de los exergames con los que contará la aplicación. Teniendo en cuenta que estos están diseñados para problemas cervicales específicos, se planteará el diseño de un juego que permita probar de forma completa los cuatro movimientos básicos descritos en el [apartado 5.4.7](#).

5.5.1 Lista de exergames

La aplicación de exergaming cuenta con un información detallada para un total de cuatro juegos en la base de datos. Estos son: Pacman, Flappy Bird, Bricks Breaker y el Snake Game o juego de la serpiente.

La pantalla principal de la aplicación, mostrada en la [figura 20](#), cuenta con un slider con la portada de cada uno de los juegos, donde al clicar en una de las portadas se abrirá una pantalla que mostrará en detalle la información asociada a ese exergame junto con un botón para iniciar el juego. Esta información incluye el nombre, una descripción, la dificultad del ejercicio, los problemas cervicales asociados al ejercicio y las instrucciones sobre cómo debe ser practicado.

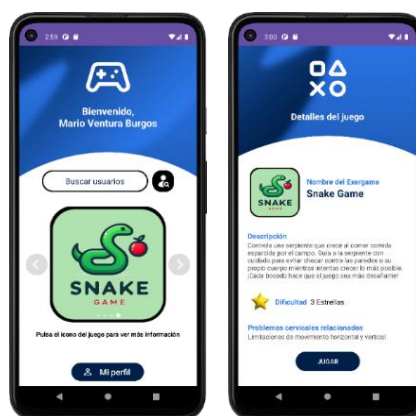


Figura 20. Pantalla principal de la aplicación con un slider de exergames y pantalla de detalle de un exergame.

Sin embargo, dado el alcance del proyecto que supone la implementación de la aplicación, bastará con desarrollar un exergame para poder verificar que realmente se pueden usar exergames con fines de rehabilitación mediante un futuro ensayo clínico y tener la aplicación preparada para contener más de uno.

Con este objetivo, se ha implementado el juego de la serpiente, ya que este permite probar movimientos horizontales y verticales, de forma que genera una experiencia de juego más completa que otros juegos y se puede asociar con problemas cervicales relacionados con limitaciones de movimiento horizontal y/o vertical.

Este juego consiste en alimentar a una serpiente para que crezca y sumar puntos, de forma que cada vez que la serpiente come, crece, y se hace más difícil manejarla puesto que el espacio es limitado. La serpiente morirá si se choca con una de las cuatro paredes (los límites de la pantalla) o contra sí misma.

Para este juego, la detección de las coordenadas de la nariz del usuario en tiempo real se usa de forma que cuando el usuario mueva la cabeza a la derecha, la serpiente girará a la derecha, y lo mismo sucederá cuando el usuario mueva la cabeza hacia la izquierda, hacia arriba o hacia abajo. Esto permite probar el planteamiento descrito para la interpretación de movimientos de la cabeza del usuario.

Como ya se mencionó en su momento, de entre todos los puntos faciales clave en un rostro, se ha optado por usar la nariz como punto de referencia para la detección de coordenadas porque esta se encuentra en la parte central del rostro, además de tener unas características (las fosas nasales) que la hacen fácilmente detectable mediante visión por computador.

El próximo reto por afrontar consiste en hacer que los juegos sean capaces de estar constantemente a la espera de la detección de movimientos de la cabeza del usuario, para usarlos para sus propósitos específicos (en este caso, mover a la serpiente en una dirección concreta). Se debe saber cómo comunicar ambas partes de la aplicación, ya que la detección de movimientos de la cabeza y los exergames son partes distintas e independientes en la aplicación.

5.5.2 Implementación de exergames

Para implementar el juego de la serpiente como exergame para dispositivos Android, se ha implementado una pantalla en la que se establece un tablero donde se moverá la serpiente. Este tablero, como muchos otros juegos en dos dimensiones, sigue una estructura matricial mediante el uso de listas de dos dimensiones, y supone la totalidad de la superficie en la que la serpiente se puede mover.

El tablero, por tanto, será una matriz a la cual se le dará forma y color mediante imágenes usadas para establecer texturas de césped o hierba sobre el terreno, y también para establecer la textura de la serpiente o de las frutas que esta se come. Las texturas usadas son las mostradas en la [figura 21](#).

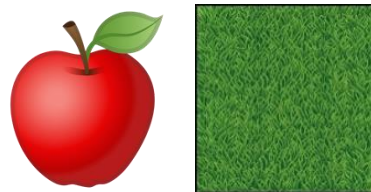


Figura 21. Texturas usadas para el exergame del juego de la serpiente.

En el caso del juego de la serpiente el tablero siempre ocupa 20 casillas de ancho, pero la altura depende del tamaño de la pantalla del dispositivo. Esto permite que el tablero tenga un tamaño considerable a la vez que la experiencia de juego se adapte y sea satisfactoria para todos los usuarios independientemente del tipo de dispositivo que usen o el tamaño de este.

La [figura 22](#) muestra el transcurso de una partida al juego de la serpiente, donde se puede apreciar (de izquierda a derecha) que la serpiente crece por cada fruta que come. Cuando la serpiente se come una manzana, aparece otra en un lugar aleatorio del mapa y se sigue la partida.



Figura 22. Transcurso habitual de una partida al “Snake Game”.

5.5.3 Comunicación entre módulos de la aplicación

Para que la serpiente pueda realizar sus movimientos (o en su defecto, cualquier otro personaje en otro exergame pueda realizar alguna acción) es necesario que los distintos módulos o partes de la aplicación se comuniquen. Esto se debe a que la detección de movimientos de la cabeza del usuario se realiza desde una parte distinta de la aplicación que la ejecución de los exergames. Ambas tareas se llevan a cabo en hilos de ejecución separados.

La implementación de un sistema de comunicación entre ambas partes es crucial para que la experiencia de juego sea fluida y eficaz, y que no haya retrasos por problemas de concurrencia o de comunicación entre distintas clases.

Realmente, los juegos no son capaces de detectar el movimiento de la cabeza de un usuario, sino que siguen las órdenes del módulo de seguimiento facial, que sí es capaz de detectarlo. Por otro lado, el detector de rostros en tiempo real no incluye juegos, sino que simplemente detecta los movimientos del usuario e informa de ellos al resto de módulos de la aplicación.

La comunicación entre ambas partes tiene lugar implementando un sistema de *listeners* u oyentes. Esto puede entenderse de la siguiente manera: El seguimiento facial en tiempo real es un narrador que tiene capacidad de ver lo que sucede y narrarlo, pero no tiene capacidad de actuar. Los exergames no tienen capacidad de ver y, por tanto, se basan en acciones que se llevan a cabo cuando son informadas por parte de un narrador. Una parte puede ver, pero no actuar, y la otra puede actuar, pero no ver.

Esto implica que el seguimiento facial en tiempo real comenzará a ejecutarse en el momento en que se inicie un juego, e irá “narrando” todo lo que “ve” al exergame que esté siendo jugado en ese momento. Mediante las operaciones descritas en el [apartado 5.4.7](#) el seguimiento facial será capaz de detectar lo que pretende hacer el usuario e informará de ello al exergame, que estará a la escucha de órdenes y las llevará a cabo en cuanto se ordenen.

Esto permite que ambas instancias se ejecuten de forma separada pero simultánea, comunicándose de forma efectiva y consiguiendo una experiencia de juego óptima para el usuario.

5.6 Instrucciones del exergame como ejercicio de rehabilitación

Cada exergame está específicamente diseñado para poder tratar algún problema cervical específico mediante la simulación del ejercicio físico tradicional que se realizaría para tratarlo. Por ello, es importante tener en cuenta que los usuarios realmente son pacientes que necesitan un mentor en la realización del ejercicio. En este caso, la aplicación debe actuar como mentor y dar al usuario las instrucciones y pautas necesarias para que pueda realizar los ejercicios correctamente, asegurando, por tanto, que pueda jugar exitosamente a los exergames.

Tal y como se menciona en el [apartado 5.1.1](#), en la base de datos de la aplicación, cada exergame cuenta con información sobre las instrucciones para realizarlo correctamente. Estas instrucciones están redactadas de forma clara y concisa, asegurando que se puedan entender fácilmente, y facilitando el aprendizaje de los movimientos para realizar el ejercicio por parte de los usuarios.

Al seleccionar un exergame en la aplicación, se hace una llamada a base de datos que devuelve la información detallada sobre este juego, entre la que se encuentran las instrucciones que debe seguir el usuario en la realización del ejercicio. Estas pautas se muestran siempre en la pantalla de detalle de un exergame, justo encima del botón de jugar, asegurando que no aparezcan de forma invasiva y molesta, pero permitiendo que el usuario siempre pueda leerlas antes de iniciar una partida.



Figura 23. Instrucciones de un exergame mostradas en la pantalla previa al juego.

5.7 Registro del progreso de los usuarios

Para que la aplicación pueda registrar el progreso de los usuarios en su proceso de rehabilitación mediante el uso de exergames, se almacenan las puntuaciones obtenidas por cada usuario en cada exergame disponible en la aplicación.

La aplicación está diseñada para que, después de cada partida, se almacene en la base de datos la siguiente información: nombre del usuario que ha jugado, exergame al que ha jugado, puntos obtenidos en la partida y fecha en la que se han obtenido los puntos. De esta forma, si se consulta la información del usuario en la base de datos, se puede ver su evolución en el tratamiento a través de las puntuaciones obtenidas a lo largo del tiempo.

Dado que cada exergame está diseñado para un problema cervical concreto, si se analiza la evolución de un usuario mediante su desempeño en el exergame a través del tiempo, se pueden extraer conclusiones sobre si este usuario está respondiendo favorablemente a su tratamiento o no.

Por ejemplo, en el hipotético caso de que un usuario jugara diez partidas a un juego, se registrarían las puntuaciones obtenidas en cada una de esas diez partidas, dejando como resultado una información en la base de datos fácilmente interpretable, como la que se muestra como ejemplo en la tabla y gráfico de la [figura 24](#).

	numpoints integer	idgame [PK] integer	username [PK] character varying (50)	fechahora [PK] timestamp without time zone
1	100	2	mario.ventura	2024-01-05 17:34:48
2	200	2	mario.ventura	2024-01-05 17:35:48
3	400	2	mario.ventura	2024-01-05 17:37:48
4	300	2	mario.ventura	2024-01-05 17:38:21
5	500	2	mario.ventura	2024-01-05 17:39:06
6	500	2	mario.ventura	2024-01-05 17:41:49
7	600	2	mario.ventura	2024-01-05 17:42:56
8	400	2	mario.ventura	2024-01-05 17:45:03
9	600	2	mario.ventura	2024-01-05 17:47:38
10	600	2	mario.ventura	2024-01-05 17:52:18



Figura 24. Gráfico que muestra la evolución de un usuario en un exergame a través de las puntuaciones que ha obtenido (caso hipotético)

En el caso de la figura anterior, se ha planteado un ejemplo en el que un usuario hipotéticamente juega diez partidas al exergame con identificador número 2. Mediante las puntuaciones que obtiene en cada partida, puede apreciarse como el usuario evoluciona favorablemente a medida que va jugando, demostrando que cada vez se desenvuelve mejor en el juego y/o que le resulta más fácil realizar los movimientos necesarios para jugar. Esto puede verse en el hecho de que cada vez obtiene más puntos. La media de puntuaciones que este usuario ha obtenido en las primeras cinco partidas es de 300 puntos, mientras que la media de puntuaciones obtenidas en las últimas cinco partidas es de 540, es decir, un 80% más que en las últimas cinco partidas.

Mediante medias aritméticas, la [tabla 5](#) muestra junto con la [figura 25](#) la evolución del usuario en este ejemplo.

Partidas	1-2	3-4	5-6	7-8	9-10
Media de puntos	150	350	500	500	600

Tabla 5. Ejemplo de evolución de un usuario a través de sus puntuaciones.

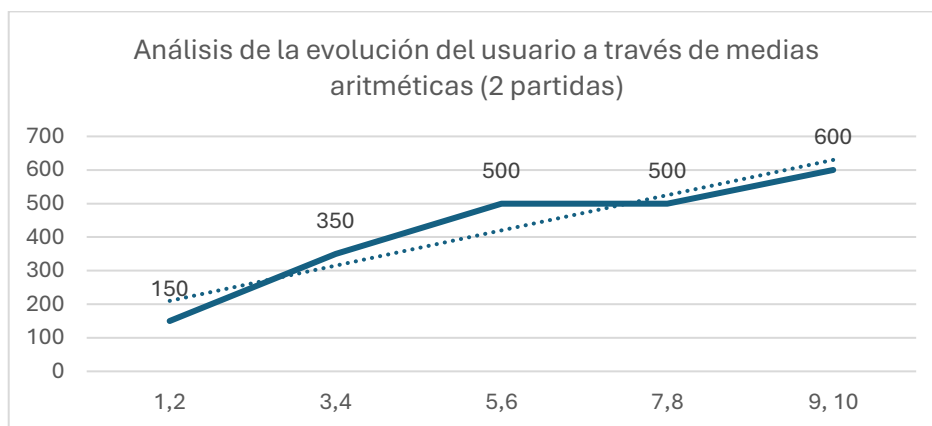


Figura 25. Gráfico que ilustra un ejemplo del análisis de las medias de puntuaciones obtenidas por un usuario para monitorizar su progreso en el tratamiento.

Cabe destacar que, como ya se ha mencionado, este no es un caso real, sino que se trata de un ejemplo con datos arbitrarios que sirven para ilustrar de forma sencilla el hipotético progreso de un usuario. Las diferencias entre las puntuaciones y la evidente evolución favorable del usuario en tan poco tiempo probablemente no sean tan comunes en un caso real de rehabilitación cervical, pero permiten entender de qué forma se puede usar la aplicación para analizar la evolución de los pacientes en su tratamiento mediante las puntuaciones que estos obtienen en los exergames.

Probablemente, en un caso real, la evolución sería más lenta y no necesariamente favorable. Los gráficos no mostrarían tantos picos y no se verían desviaciones tan drásticas y repentinas.

5.8 Notificaciones y recordatorios

Por último, con el objetivo de aumentar la adherencia de los usuarios al tratamiento, se ha implementado un sistema de notificaciones y recordatorios que avisa al usuario sobre la necesidad de realizar sus ejercicios de rehabilitación. Este sistema es de vital importancia para asegurar que el usuario sigue el tratamiento de manera consistente y no lo abandona, lo que es crucial para progresar satisfactoriamente en el tratamiento.

Mediante mensajes amigables y motivacionales, la aplicación informa al usuario de que debe realizar su tratamiento, recordándole la importancia de mantener la constancia en su programa de ejercicios. Las notificaciones incluyen frases motivadoras y recordatorios de realizar una sesión de juego. Esto ayuda a que los usuarios no olviden jugar diariamente y, por tanto, que no abandonen las actividades terapéuticas que la aplicación de exergaming pretende que hagan para ayudarles a completar su tratamiento.

Este enfoque proactivo no solo promueve la adherencia al tratamiento, sino que también contribuye a que los usuarios se sientan apoyados y acompañados en su proceso de rehabilitación, mejorando así su experiencia general con la aplicación.

Estas notificaciones se configuran a nivel interno al iniciar la aplicación por primera vez, tras crear una cuenta e iniciar sesión con ella. Al acceder a la pantalla principal de la aplicación por primera vez, se configuran una serie de recordatorios que se mandarán diariamente a las 08:00pm de la zona horaria en la que se encuentre el dispositivo Android. Mediante mensajes personalizados y motivacionales, se pretende que estos recordatorios ayuden a los usuarios a recordar que deben realizar su tratamiento, aumentando la adherencia a este como consecuencia.

La [figura 26](#) muestra algunos ejemplos de notificaciones que la aplicación manda a los usuarios para recordarles que dediquen un rato a jugar a los exergames para realizar sus ejercicios de rehabilitación.



Figura 26. Notificaciones y recordatorios enviados por la aplicación.

Cabe mencionar que, tal y como se mencionó en el [apartado 5.3.1](#), para que la aplicación pueda mandar notificaciones a los usuarios, será necesario pedir permiso explícitamente al usuario en dispositivos con una versión de Android igual o superior a la 13 (API nivel 33) [\[43\]](#).

6. PRUEBAS Y RESULTADOS

En este apartado se detallan las pruebas realizadas para evaluar el impacto y efectividad del uso de exergames para rehabilitación en usuarios reales, así como el rendimiento de la aplicación en distintos dispositivos y pruebas de usabilidad para evaluar la experiencia del usuario.

6.1 Pruebas con usuarios reales

En este subapartado, se describen las pruebas llevadas a cabo con usuarios reales, explicando el planteamiento de estas pruebas, la metodología empleada, el perfil de los participantes, y los resultados obtenidos en términos de desempeño en los exergames y satisfacción del usuario.

La prueba realizada consiste en hacer que un conjunto de 10 usuarios juegue un total de 10 partidas cada uno al exergame del juego de la serpiente, evaluando las puntuaciones obtenidas por cada usuario en cada partida para descubrir si puede apreciarse una evolución. La edad media de los participantes es de 23,8 años, con una desviación estándar de 1,72 años. La muestra está compuesta por un 60% de hombres y un 40% de mujeres. Todos los participantes cuentan con experiencia previa con dispositivos móviles.

Se anotarán las puntuaciones de cada usuario en sus partidas y posteriormente se analizarán los resultados para descubrir si ha habido algún progreso o adaptación al juego o tratamiento.

Cabe mencionar que los usuarios que participan en el experimento son usuarios reales pero que no padecen de ningún trastorno musculoesquelético o problema cervical. Sin embargo, su participación permitirá evaluar la usabilidad y eficacia del exergame en un entorno controlado antes de aplicarlo a pacientes con necesidades específicas de rehabilitación. Aunque estos usuarios no tienen problemas cervicales, su interacción con el juego proporciona valiosa información sobre el aprendizaje, la adaptabilidad, y el nivel de dificultad del juego, lo que podría permitir realizar ajustes necesarios para que el exergame sea más efectivo cuando se aplique a usuarios con condiciones médicas. Además, su participación ayudará a identificar posibles mejoras en la interfaz y en la experiencia del usuario, garantizando que el exergame sea accesible y motivador para todos los usuarios, independientemente de su estado de salud.

La [figura 27](#) muestra imágenes sobre los usuarios durante la realización del experimento.

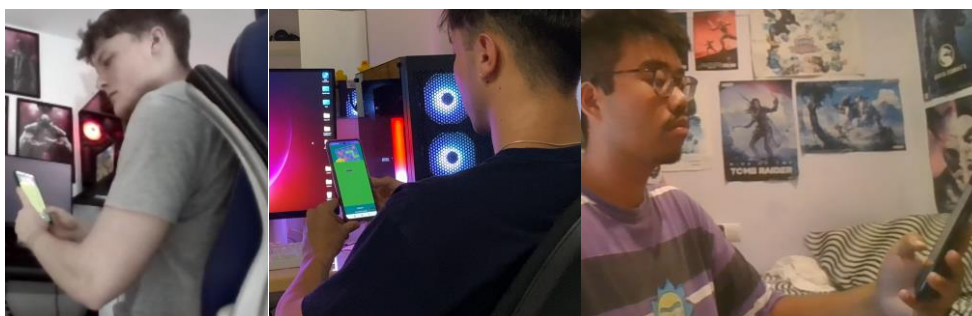


Figura 27. Pruebas con usuarios reales.

Siendo la prueba 1 (P1) la primera, y la prueba 10 (P10) la última, las puntuaciones obtenidas por los usuarios en sus respectivas partidas son las siguientes:

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Promedio
Usuario 1	0	0	100	200	200	300	200	300	100	300	170
Usuario 2	0	0	100	200	200	300	300	300	400	300	210
Usuario 3	100	100	100	200	200	200	200	300	400	400	220
Usuario 4	0	100	100	100	200	300	300	300	400	400	220
Usuario 5	0	100	100	200	200	200	300	400	200	600	230
Usuario 6	0	0	200	200	200	300	200	300	300	300	200
Usuario 7	0	100	100	200	200	200	300	300	400	500	230
Usuario 8	100	100	100	200	200	300	300	400	300	400	240
Usuario 9	0	100	100	100	200	200	300	300	400	500	220
Usuario 10	100	100	100	200	200	200	300	300	400	400	230
Promedio	30	70	110	180	200	250	320	320	330	410	

Tabla 6. Resultados obtenidos por usuarios reales en pruebas de usuario.

La [figura 28](#) muestra las puntuaciones obtenidas por los usuarios en las diez partidas.

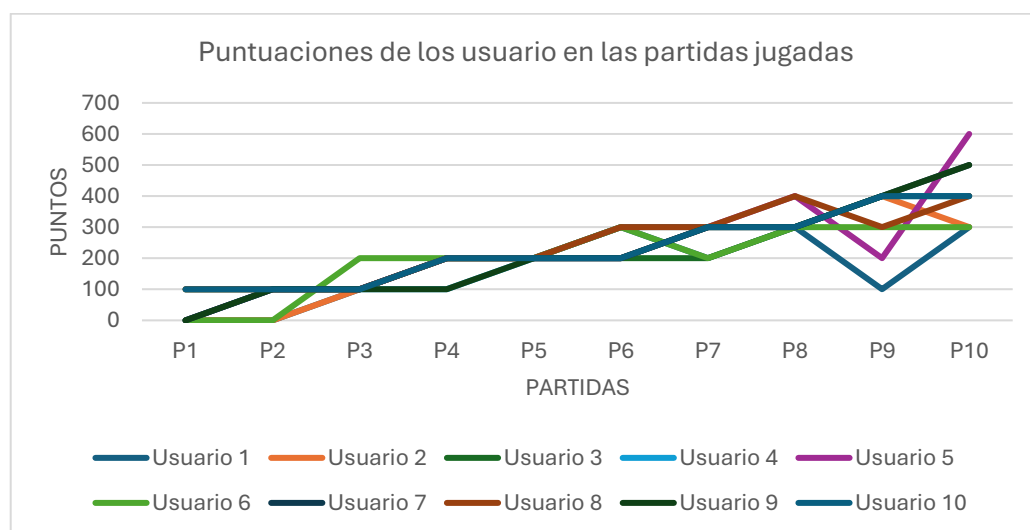


Figura 28. Puntuaciones obtenidas por los usuarios en las partidas jugadas propuestas para el experimento.

Puede observarse un evidente patrón general de mejora en las puntuaciones de la mayoría de los usuarios a medida que avanzan en las partidas. Este incremento sugiere un proceso de adaptación y aprendizaje, donde los usuarios se familiarizan con la aplicación y las mecánicas del juego, mejorando su desempeño como consecuencia.

Pese a que los usuarios empiezan con puntuaciones bajas de entre 0 y 100 puntos, la mayoría muestra un evidente crecimiento consistente en el tiempo, alcanzando puntuaciones con valores más altos como 300 o 400 puntos en las últimas partidas. De 100 partidas jugadas en total, solamente en 3 se logró superar la barrera de los 400 puntos, alcanzando 500 puntos o más. Estas partidas fueron jugadas por tres usuarios distintos, suponiendo que un 30% de los usuarios logró superar los 400 puntos en alguna ocasión, pero ninguno lo consiguió más de una vez. Solamente un usuario alcanzó los 600 puntos en una ocasión, suponiendo un 1% de las partidas jugadas.

Esto indica que el exergame permite a los usuarios adaptarse progresivamente, lo cual es una señal positiva en términos de adaptación a la aplicación, aunque también indica que debería revisarse la dificultad del juego, ya que cada movimiento correcto supone un incremento de 100 puntos, por lo que el máximo global de movimientos consecutivos realizados correctamente es de 6 (gesta sólo conseguida por uno de los 10 usuarios).

La [figura 29](#) muestra que el desempeño de los usuarios que han participado en el experimento es similar. Se observa que la mayoría de los usuarios alcanzaron promedios de puntuación parecidos, en torno a 220-240 puntos. Esto sugiere que, en general, los usuarios fueron capaces de adaptarse al juego a un ritmo comparable.

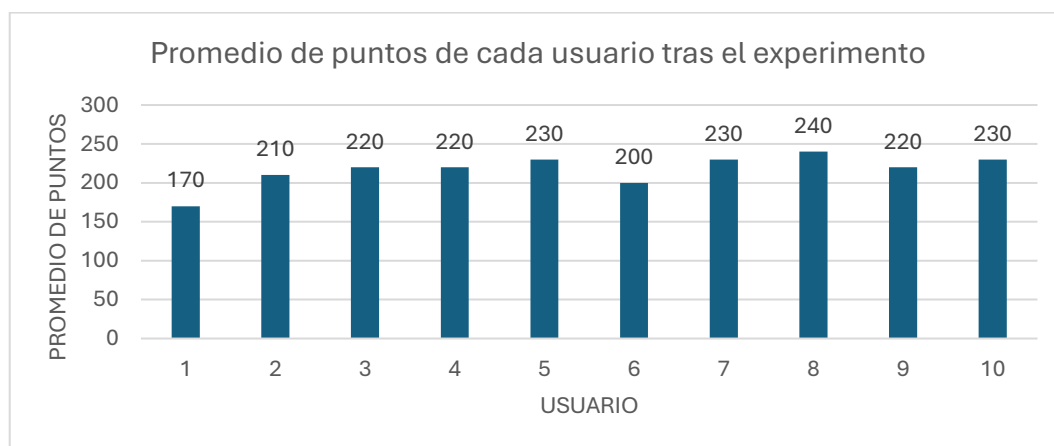


Figura 29. Promedio de puntos de cada usuario en las diez partidas jugadas

Los promedios de puntos más bajos (usuario 1) sugieren un desempeño menor, y pueden advertir de una mayor dificultad de adaptación al juego o a la aplicación por parte del usuario. Pese a que puede ser un caso aislado y deberse a causas individuales, podrían explorarse a fondo las razones

de este bajo desempeño para conocer si existe alguna forma de mejorar la aplicación y hacer que los usuarios se adapten a ella y a los juegos más fácilmente.

La [figura 30](#) muestra el promedio de puntuaciones obtenidas por los diez usuarios en cada partida, permitiendo analizar la tendencia general de los usuarios a lo largo de las partidas propuestas en el experimento.

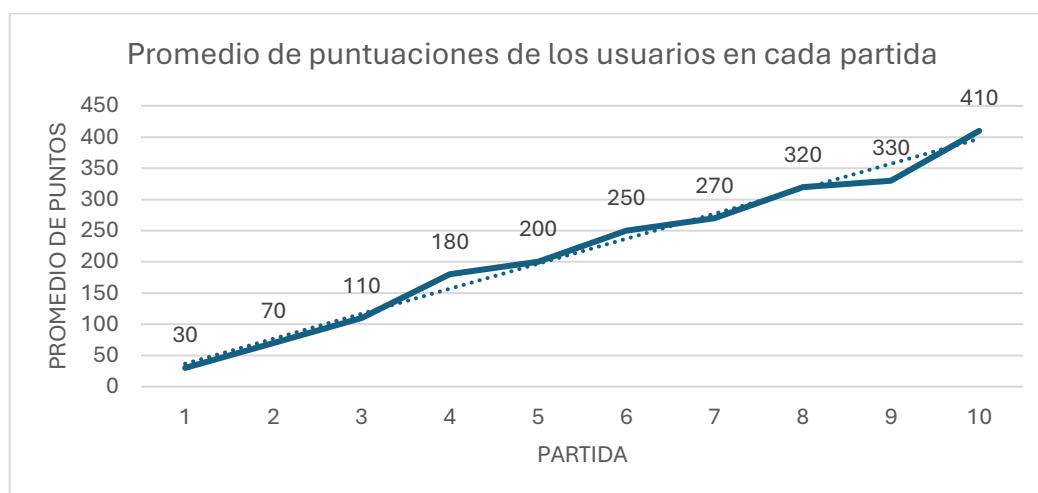


Figura 30. Promedio de puntuaciones de todos los usuarios en cada partida.

Se observa un incremento constante en el promedio de puntos a lo largo de las partidas, comenzando con menos de 50 puntos en la primera y superando los 400 puntos en la última. Esto indica que los usuarios mejoraron progresivamente con la práctica, lo que sugiere un proceso de aprendizaje y adaptación al juego. Esta progresión continua y ascendente refleja una clara curva de aprendizaje, en la que los usuarios, a medida que adquieren más experiencia con el exergame, logran obtener resultados significativamente mejores. Este patrón es un indicador positivo de la efectividad del juego para captar la atención de los usuarios y mejorar su rendimiento con el tiempo, lo cual también sugiere que se podrían obtener resultados igualmente favorables en un experimento con usuarios que presenten necesidades reales de rehabilitación.

Además, la línea de tendencia permite identificar variaciones en el rendimiento a lo largo de las partidas. Por ejemplo, se observan picos de puntuación superiores a lo esperado en partidas como la 4, así como una ligera caída en la partida 9. No obstante, la tendencia general se mantiene estable y, al finalizar el experimento en la partida 10, el promedio de puntuaciones alcanzado por los usuarios se alinea con las expectativas marcadas por la tendencia anterior. Este comportamiento evidencia un progreso creciente que, en su mayor parte, sigue un patrón lineal desde el inicio hasta el final del experimento.

Cabe destacar que, muy probablemente, si el experimento se llevara a cabo con usuarios reales que padeciesen algún trastorno musculoesquelético o problema cervical, la adaptación y progreso

de los usuarios sería más lento. Esto se debe a que estos usuarios sí tendrían dificultades reales para realizar los movimientos necesarios para jugar a los exergames, y probablemente se apreciarían más picos y desviaciones, y una curva de aprendizaje menos pronunciada.

6.2 Pruebas de rendimiento

Esta sección se centra en evaluar el rendimiento de la aplicación en dispositivos de distinta gama o categoría (baja, media, alta) para analizar las diferencias en cuanto a consumo de recursos y tiempos de respuesta en función del dispositivo que usen los usuarios. Esto es de vital importancia ya que para que la aplicación funcione debe realizar una serie de acciones de cierta complejidad, y la capacidad del dispositivo para abordar estas tareas de forma efectiva puede traducirse en una mejor (o peor) jugabilidad en los exergames y experiencia de usuario.

La aplicación debe ser capaz de acceder a la cámara del dispositivo, obtener un input, usar el framework de ML Kit para analizar el input, obtener un resultado y usarlo para realizar movimientos en exergames. Se considera que esta prueba puede ser de gran utilidad para evaluar las diferencias entre distintos dispositivos, ya que diferentes dispositivos cuentan con distintas cámaras de diferente calidad y tamaño de imagen, y distinto procesador.

Además, también podrán evaluarse aspectos como tamaños de pantalla diferentes, lo cual resulta relevante porque el tamaño del tablero o mapa donde se juega a los exergames se adapta al tamaño de la pantalla del dispositivo, de forma que no es lo mismo jugar en un teléfono que en una Tablet.

Para realizar el experimento, se aprovechará la funcionalidad de “Probar tracker” de la aplicación. Esta muestra una pantalla en la que se ve en tiempo real el seguimiento facial que se está haciendo, y que permite conocer las coordenadas detectadas, coordenadas anteriores, desplazamiento, movimientos detectados y número de análisis realizados sobre la cámara.

Para la prueba se usarán **6 dispositivos de distinta categoría y características**, y el experimento consistirá en utilizar la información mostrada en esta pantalla para cuantificar el número de análisis que la aplicación es capaz de realizar en un lapso de 1 segundo. Este dato es crucial, ya que cada análisis corresponde a una posible actualización del movimiento en el exergame. Si se realiza un análisis, se puede reflejar un nuevo movimiento (o mantener el estado actual si no se detecta ningún cambio respecto al análisis anterior) dentro del juego.

Una menor frecuencia de análisis se traduce en una menor actualización de movimientos, lo que puede provocar retrasos y afectar negativamente la fluidez del juego y, por consiguiente, en la efectividad del uso de la aplicación con fines de rehabilitación. Esto, a su vez, impacta directamente en la experiencia del usuario, ya que una baja tasa de análisis podría dar lugar a una latencia perceptible entre la acción del jugador y la respuesta del exergame, lo que podría resultar en una experiencia frustrante y poco inmersiva. Por tanto, optimizar el número de análisis por segundo es fundamental para garantizar un rendimiento fluido y una interacción en tiempo real dentro del exergame.

La [tabla 7](#) muestra los detalles sobre los dispositivos usados para el experimento, sus características principales, y el número de análisis realizados.

Dispositivo	Tipo	Gama	Pantalla	Cámara frontal	Versión Android	Núm. análisis / s
Samsung Galaxy Tab A	Tableta	Baja	10.1"	2 MP	7.1.1	2.73
Realme C55	Teléfono	Media	6,72"	8MP	13	23.1
Xiaomi Redmi Note 9 Pro	Teléfono	Media	6,67"	8MP	10	19.7
Xiaomi Redmi Note 10 Pro	Teléfono	Media	6,67"	16 MP	11	25.93
OPPO Poco F4	Teléfono	Media/Alta	6,67"	20MP	13	23.5
Samsung Galaxy S22	Teléfono	Alta	6,1"	10MP	12	33.4

Tabla 7. Dispositivos usados para el experimento, especificaciones y número de análisis realizados en un lapso de 60 segundos.

Tras las pruebas realizadas sobre distintos dispositivos, la [figura 31](#) muestra un gráfico de barras donde se aprecia la diferencia de análisis realizados en distintos dispositivos.

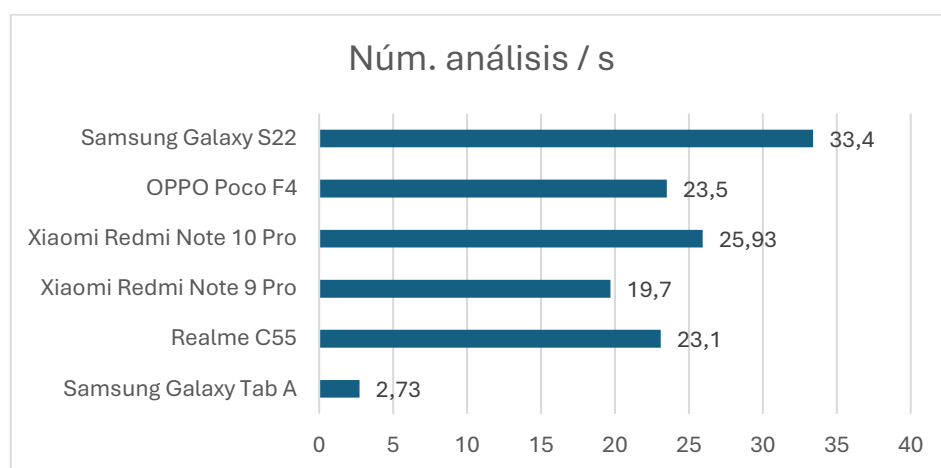


Figura 31. Número de análisis realizados por distintos dispositivos en 60 segundos.

A partir de los datos recopilados tras el experimento sobre el número de análisis de rostro que cada dispositivo puede realizar en 60 segundos, pueden extraerse diversas conclusiones.

En primer lugar, se observa una clara diferencia de rendimiento según la gama del dispositivo. El Samsung Galaxy S22, un dispositivo de gama alta, muestra un rendimiento superior en el análisis de rostros, con 33.4 análisis por segundo. Esto refleja el mayor poder de procesamiento de los dispositivos de gama alta, lo que les permite realizar más análisis en menos tiempo, garantizando así una mejor experiencia de usuario.

Debe tenerse en cuenta que se considera que un análisis se realiza “en tiempo real” cuando alcanza unas cifras superiores a 30 fotogramas por segundo. Sin embargo, dependiendo del contexto, puede considerarse “tiempo real” si se puede procesar imágenes a una velocidad que es suficiente para que el usuario no perciba ningún retraso o interrupción en la interacción. Puede asegurarse, por tanto, que el dispositivo de gama alta cumple con las necesidades de análisis en tiempo real.

Los dispositivos de gama media o media/alta, en cambio, aunque no alcanzan el mismo nivel de análisis por segundo que el Galaxy S22, ofrecen un rendimiento más que suficiente. Con cifras que oscilan entre 19 y 26 movimientos por segundo durante un exergame. Esto no alcanza la cota de 30 fotogramas por segundo descritos previamente, pero aun así permite una jugabilidad fluida y sin interrupciones al exergame propuesto para el experimento.

Dado que los exergames suelen ser juegos de ritmo lento o moderado, donde los movimientos se controlan mediante la cabeza, este nivel de análisis es más que adecuado. Aunque la diferencia en el número de análisis entre los dispositivos de gama media y el de gama alta es significativa, ambos garantizan una experiencia de juego fluida, superando con creces las exigencias mínimas recomendadas.

Por otro lado, el Samsung Galaxy Tab A, un dispositivo de gama baja, muestra el peor rendimiento en este experimento, con solo 2.73 análisis por segundo. Este resultado sugiere que los dispositivos de gama baja podrían no ser adecuados para una experiencia de exergaming fluida, ya que el bajo número de análisis podría provocar retrasos en la detección de movimientos, afectando negativamente la experiencia del usuario. Cabe destacar que este dispositivo tiene más de 8 años y no era de gama alta en el momento de su lanzamiento. Por lo tanto, se puede concluir que prácticamente cualquier dispositivo de gama media actual, en condiciones normales, debería ser capaz de ejecutar exergames sin ningún problema.

Además, cabe recalcar que el experimento se hizo con los dispositivos en “condiciones óptimas”, es decir, ninguno de ellos realizó los análisis de rostros mientras realizaba otras tareas en segundo plano como descargar archivos, reproducir música, etc. Es probable que, en caso de ser así, el número de análisis por segundo se viese reducido por disponer de menos recursos de procesador. Esto agrava más el caso observado con el dispositivo de gama baja y resalta la importancia de los buenos resultados obtenidos con los dispositivos de gama media, media/alta y alta.

Respecto a la versión de Android, esta también parece influir en el rendimiento, aunque no de manera determinante. Por ejemplo, el Realme C55 con Android 13 realiza 23.1 análisis, mientras que el Xiaomi Redmi Note 9 Pro con Android 10 realiza 19.7 análisis. Esto sugiere que las

optimizaciones en las versiones más recientes del sistema operativo pueden contribuir a un mejor rendimiento, aunque no es el único factor decisivo.

En la [figura 32](#) puede verse como el número de análisis del dispositivo con Android 10 es mayor al del dispositivo con Android 7.1.1. A su vez, el dispositivo con Android 11 ha realizado más análisis en el mismo lapso que el dispositivo con Android 10. Sin embargo, llegados a versiones de Android como la 11 o superiores, puede verse que las diferencias en el número de análisis entre los dispositivos son mínimas.

De hecho, el dispositivo que tiene Android en su versión 12 ha realizado una cantidad notablemente mayor de análisis por segundo que los dos dispositivos con Android 13. Esto puede deberse a que el dispositivo con Android 12 es de gama alta y tiene más capacidad de procesamiento. Esto permite concluir que no existe ninguna relación directa entre las versiones de Android de los dispositivos y la cantidad de análisis por segundo que pueden hacer, sino que esta depende de características como la capacidad de procesamiento del dispositivo en concreto.

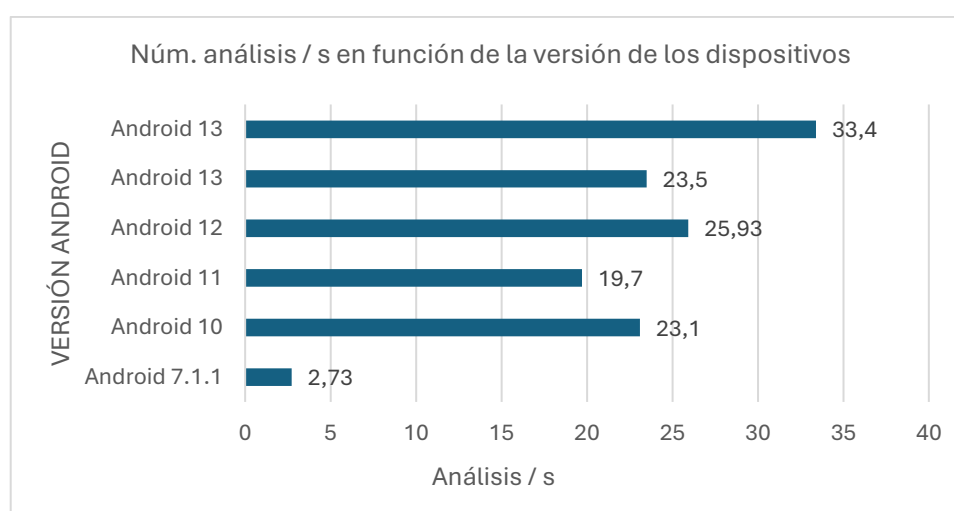


Figura 32. Versión de Android de los dispositivos y número de análisis por segundo realizados.

Por otro lado, aunque no está tan directamente relacionado, el tamaño de la pantalla puede influir en el rendimiento, especialmente en dispositivos menos potentes como el Samsung Galaxy Tab A. Este dispositivo no solo tiene la peor calidad de cámara, sino que también es el más lento en términos de procesamiento de imágenes. Dado que el tablero de los exergames se adapta al tamaño de la pantalla, la tableta debe mostrar un mapa relativamente grande y actualizar sus gráficos constantemente. Estas demandas adicionales, combinadas con su limitado poder de procesamiento, pueden haber contribuido a su bajo rendimiento en el experimento.

Por último, los dispositivos con pantallas más pequeñas, pero de gama alta, como el Samsung Galaxy S22, demuestran que un dispositivo puede ofrecer un rendimiento excepcional en el análisis de rostros independientemente del tamaño de la pantalla.

Sin embargo, no se puede concluir que exista una correlación directa entre el tamaño de la pantalla y el rendimiento en el análisis de imágenes, ya que otros factores como la capacidad de procesamiento, parecen ser más importantes.

6.3 Pruebas de experiencia de usuario

En este subapartado, se realizará una evaluación sobre la usabilidad y satisfacción del usuario. También permitirá evaluar la interfaz para saber cómo de intuitiva resulta para los usuarios, permitiendo cuantificar la facilidad de navegación.

Con este objetivo, se ha optado por medir la usabilidad usando el cuestionario SUS (*System Usability Scale*, o SUS). Este es un método rápido y validado que lleva usándose durante más de 30 años para evaluar la usabilidad de cualquier sistema informático, permitiendo obtener un valor global de la usabilidad [44]. El concepto de usabilidad de un sistema considera tres dimensiones principales que pueden medirse:

- **Eficacia:** Se refiere a si los usuarios del sistema pueden alcanzar sus objetivos.
- **Eficiencia:** La cantidad de esfuerzo necesario para alcanzar los objetivos.
- **Satisfacción:** Nivel de satisfacción con el uso del sistema.

La prueba consiste en encuestar a los usuarios del sistema con 10 preguntas o afirmaciones (todos los SUS responden a las mismas preguntas, ya vienen predefinidas y no cambian dependiendo del sistema informático que se evalúe) que deben ser respondidas siguiendo una **Escala de Likert** de 5 puntos [45]. Para la prueba, se ha encuestado a los mismo 10 usuarios que participaron en el experimento del [apartado 6.1](#), ya que estos han tenido que usar la aplicación y desenvolverse con ella para su propósito específico: jugar a exergames para rehabilitación cervical.

Una vez que los usuarios han respondido, se realizan una serie de cálculos en base a las respuestas, que permiten obtener una cantidad numérica del 1 al 100 que suele interpretarse de la siguiente manera:

- **Por debajo de 50:** Se considera una usabilidad pobre o inaceptable.
- **Entre 51 y 68:** Se considera aceptable, pero con áreas de mejora significativas.
- **Entre 69 y 80:** Se considera una usabilidad buena o aceptable.
- **Por encima de 80:** Se considera una usabilidad muy buena o excelente, indicando que el sistema es muy bien recibido por los usuarios.

En el caso de la aplicación de exergaming, se ha obtenido una puntuación total de **72'5 puntos**, lo cual supone una usabilidad buena o aceptable, aunque con ciertas áreas de mejora. El cálculo detallado sobre cómo se ha obtenido esta puntuación puede verse en el [anexo A.4](#).

6.4 Resultados

En este apartado se proporcionan las conclusiones extraídas en base a la información obtenida tras los experimentos y pruebas realizados en los apartados anteriores.

Los usuarios que han usado la aplicación muestran un rápido proceso de adaptación al entorno que presenta y a los exergames de los que dispone. Cuanto más juegan, más fácil les resulta entender el juego y los movimientos que deben realizar. Los resultados de los usuarios tras las 10 partidas propuestas para el experimento del [apartado 6.1](#) muestran una clara tendencia de mejora en el desempeño a lo largo del tiempo, lo que sugiere que la aplicación es efectiva en el proceso de aprendizaje y motivación de los usuarios, y además hace que se adaptan bien a los exergames y a los movimientos que deben realizar para su tratamiento.

Por otro lado, las pruebas de rendimiento realizadas en el [apartado 6.2](#) revelan que los dispositivos de gama media o alta ofrecen una mejor experiencia de usuario debido a sus mejores prestaciones con respecto a otros dispositivos. Una gran frecuencia de análisis por segundo permite una experiencia de juego fluida, permitiendo jugar sin retrasos ni interrupciones. Los dispositivos de gama más baja han mostrado más dificultades para cumplir con las necesidades de análisis en tiempo real y, pese a que se puede jugar a los exergames en estos dispositivos, es evidente que la experiencia de usuario y la fluidez de los juegos se ven afectadas por esta baja frecuencia de análisis.

Además, no se ha podido demostrar ninguna relación directa entre la frecuencia de análisis realizados y la versión de Android, tamaño de la pantalla o calidad de la cámara del dispositivo. Pese a que pudiera parecer que la versión de Android sí que influye, debe tenerse en cuenta que a medida que pasa el tiempo, cuanto más modernos son los dispositivos, estos cuentan con mejores prestaciones y versiones de Android más avanzadas. Dispositivos de alta gama que cuenten con características como una gran capacidad de procesamiento mostrarán mejores resultados, pero no por la nueva versión de Android de la que disponen, sino por la capacidad de procesamiento u otras características.

Por último, en el [apartado 6.3](#), la aplicación obtuvo una puntuación de 72'5 en el cuestionario de *System Usability Scale* (SUS), lo que sugiere que la aplicación es amigable, entendible y fácil de usar. Este resultado revela una aceptación general por parte de los usuarios que han usado la aplicación, indicando que la usabilidad de la aplicación es aceptable, aunque existe margen de mejora hasta alcanzar la excelencia.

7. TRABAJO FUTURO

El prototipo funcional desarrollado es operativo y usable, por lo que puede usarse en un futuro ensayo clínico para realizar ejercicios terapéuticos de rehabilitación cervical mientras se juega a videojuegos. Sin embargo, esto no impide que se puedan plantear proyectos o modificaciones complementarias en un futuro.

Un posible trabajo futuro en el ámbito de la rehabilitación cervical mediante el uso de exergames sería la implementación de exergames que impliquen la simulación de algún tipo de fuerzas, siguiendo las técnica descrita en el [anexo A.3](#) o usando otras. Hasta ahora, la aplicación permite evaluar si los usuarios han realizado algún movimiento, y se juega a los exergames en base a ello; pero gracias a la implementación de esta técnica u otras similares, también se podría evaluar la soltura o facilidad con la que los usuarios hacen los movimientos para los cuales en un principio mostraban dificultades, ayudando a realizar un seguimiento efectivo del progreso de los usuarios en su tratamiento.

Otra solución futura sería la posibilidad de que la aplicación genere datos clínicos valiosos. Se puede implementar algún tipo de sistema que permita a la aplicación realizar de forma automática un análisis de la información recopilada sobre el desempeño de los usuarios en los exergames. De esta manera, se podría obtener de forma automatizada ciertas conclusiones sobre el avance de los usuarios en su tratamiento, convirtiendo la aplicación en una herramienta útil para la obtención de datos clínicos de valor.

Esto podría permitir a los profesionales de la salud ajustar los planes de rehabilitación de manera más precisa y en tiempo real, basándose en el rendimiento individual de cada usuario y personalizando la terapia al ajustarla a las necesidades concretas del usuario detectadas por la aplicación. Además, este sistema automatizado podría identificar patrones en el progreso de los usuarios, ayudando a detectar posibles dificultades o estancamientos en el proceso de rehabilitación.

Otra posibilidad viable que podría enriquecer la experiencia del usuario en la aplicación sería la introducción de doctores o profesionales de la salud en la aplicación, haciendo que cada doctor tenga una serie de pacientes asociados y cada paciente tenga su doctor. Esto podría permitir que la aplicación envíe directamente al doctor la información asociada a cada sesión de terapia, facilitando el acceso a esta información y simplificando el proceso de seguimiento.

Para mejorar el entorno social, también podría implementarse funcionalidades como algún sistema de chat o llamada (o ambas) entre distintos usuarios ya sean pacientes o doctores, la creación de un sistema de recompensas reales para los usuarios al alcanzar ciertos hitos en su rehabilitación, o la implementación de exergames con función multijugador, que podría ayudar a mejorar todavía más la motivación al tratamiento gracias al fomento de la competitividad entre dos pacientes o más que realizan sus ejercicios terapéuticos simultáneamente mientras juegan.

8. CONCLUSIONES

Este proyecto ha logrado cumplir los objetivos propuestos al diseñar, implementar, probar y analizar una aplicación que permite jugar a exergames específicamente diseñados para tratar problemas cervicales, con el fin de hacer el proceso de rehabilitación más ameno y entretenido. La aplicación no solo ofrece a los usuarios la posibilidad de disfrutar mientras tratan sus problemas cervicales, sino que también incorpora un entorno social motivador que les permite sentirse acompañados durante el proceso. Esto puede fomentar la integración social de ciertos pacientes y mejora la adherencia al tratamiento.

En cuanto al cronograma del proyecto, se identificó una desviación significativa en el requisito no funcional 4 (RNF4), que abarcó todo el sprint 2. La implementación de esta funcionalidad llevó más tiempo del esperado debido a dificultades técnicas, y en lugar de completarse el 25 de marzo, como estaba planeado, finalizó el 10 de abril, resultando en una desviación de 16 días. Este retraso afectó el inicio de tareas en sprints posteriores que dependían de la finalización del seguimiento facial en tiempo real. A pesar de este contratiempo, el proyecto se completó sin otros problemas significativos.

Para el desarrollo de la aplicación, se utilizó el IDE Android Studio con Java y XML como lenguajes de programación, y Visual Studio Code junto con PostgreSQL y sus herramientas asociadas (SQL Shell y pgAdmin 4) para el diseño y creación de una base de datos relacional que se conecta a la aplicación y permite el intercambio de información. Se realizaron numerosas pruebas y optimizaciones, especialmente en los algoritmos de seguimiento facial en tiempo real, que son fundamentales para el funcionamiento de los exergames, el eje central del proyecto.

Este proyecto subraya la significativa utilidad de diversas tecnologías y técnicas al integrarlas con videojuegos para lograr objetivos que combinan elementos lúdicos con fines terapéuticos. Destaca la relevancia de técnicas como el reconocimiento facial mediante visión por computador, que permite adaptar la experiencia de juego a las necesidades del usuario en tiempo real, así como el uso de una base de datos relacional para la creación de un entorno motivador, o la relevancia de operaciones matemáticas que, pese a ser simples, cumplen un papel para lograr grandes propósitos en el contexto de la rehabilitación.

Cabe destacar que este proyecto ha implementado satisfactoriamente un exergame usable para rehabilitación cervical que podrá ser usado en el ensayo clínico planificado dentro del proyecto TED2021-131358A-I00 para analizar si realmente es posible mejorar la adherencia a los programas de rehabilitación cervical mediante la implementación de soluciones tecnológicas que combinen el ejercicio físico con elementos lúdicos.

BIBLIOGRAFÍA

- [1] G. Hernández, ““El dolor cervical es una de las principales causas de discapacidad en el mundo””, UDEP Hoy, 28-ago-2023. [En línea]. Disponible en: <https://www.udep.edu.pe/hoy/2023/08/el-dolor-cervical-es-una-de-las-principales-causas-de-discapacidad-en-el-mundo/>. [Consultado: 02-mar-2024].
- [2] Á. Parra, “Dolor Cervical O Cervicalgia. Consecuencias Y Tratamiento De Fisioterapia”, Clinica Dolor Orofacial, 03-sep-2018. [En línea]. Disponible en: <https://www.clinicadolororofacial.es/dolor-cervical-consecuencias-y-tratamiento-de-fisioterapia/>. [Consultado: 02-mar-2024].
- [3] A. al T. y. S. de L. C. de A. G. en M. del D. S. I. de la pro Formosa. Periodo, “UNIVERSIDAD NACIONAL DE ROSARIO CENTRO DE ESTUDIOS INTERDISCIPLINARIOS MAESTRÍA EN GESTIÓN DE SISTEMAS Y SERVICIOS DE SALUD”, Edu.ar. [En línea]. Disponible en: <https://rephip.unr.edu.ar/server/api/core/bitstreams/702e3e45-8c39-41d7-aa8b-d18dc9da11e7/content>. [Consultado: 02-mar-2024].
- [4] J. E. Muñoz, J. F. Villada, y J. C. Giraldo Trujillo, “Exergames: una herramienta tecnológica para la actividad física”, Rev. médica Risaralda, vol. 19, núm. 2, pp. 126–130, 2013.
- [5] SNSF Scientific Image Competition y By: SNSF Scientific Image Competition, ““nice!’ by Sofia anzeneder, university of Bern”, Flickr. [En línea]. Disponible en: https://www.flickr.com/photos/snsf_image_competition/52793680400/in/album-72177720307264813/. [Consultado: 02-mar-2024].
- [6] Researchgate.net. [En línea]. Disponible en: https://www.researchgate.net/publication/230794344_Defining_exergames_exergaming. [Consultado: 02-mar-2024].
- [7] A. E. Staiano y S. L. Calvert, “Exergames for physical education courses: Physical, social, and cognitive benefits: Exergames for physical education courses”, Child Dev. Perspect., vol. 5, núm. 2, pp. 93–98, 2011.
- [8] Uva.es. [En línea]. Disponible en: <https://uvadoc.uva.es/bitstream/handle/10324/36567/TFG-B.1264.pdf;jsessionid=B2D91C347A589D5E3EBF36728B0DBB1C?sequence=1>. [Consultado: 08-mar-2024].
- [9] H. Holguín, “Este videojuego peruano tiene el poder de la rehabilitación médica”, CNN en Español, 16-ago-2018. [En línea]. Disponible en: <https://cnnespanol.cnn.com/2018/08/16/videojuego-peruano-ayudar-personas-derrames-cerebrales-acv/>. [Consultado: 14-abr-2024].

- [10] “Dispositivos de bajo coste en rehabilitación neurológica”, Neuraxis, 04-may-2022.
- [11] “Vista de Videojuegos activos y salud mental: una revisión sistemática con metaanálisis”, Uma.es. [En línea]. Disponible en: <https://revistas.uma.es/index.php/riccafd/article/view/15805/16713>. [Consultado: 14-abr-2024].
- [12] Ugr.es. [En línea]. Disponible en: <https://produccioncientifica.ugr.es/documentos/618f57789ff8c939aacc1588>. [Consultado: 02-mar-2024].
- [13] L. E. F. Graves, N. D. Ridgers, y G. Stratton, “The contribution of upper limb and total body movement to adolescents’ energy expenditure whilst playing Nintendo Wii”, *Eur. J. Appl. Physiol.*, vol. 104, núm. 4, pp. 617–623, 2008.
- [14] W. Peng, J.-H. Lin, y J. Crouse, “Is playing exergames really exercising? A meta-analysis of energy expenditure in active video games”, *Cyberpsychol. Behav. Soc. Netw.*, vol. 14, núm. 11, pp. 681–688, 2011.
- [15] D. E. Meardon, “Diagramas de Gantt”, Atlassian. [En línea]. Disponible en: <https://www.atlassian.com/es/agile/project-management/gantt-chart>. [Consultado: 14-abr-2024].
- [16] J. A. Del Prado, “Medir con el sistema de escala de usabilidad (SUS)”, UXABLES | Blog, 17-abr-2024. [En línea]. Disponible en: <https://www.uxables.com/investigacion-ux/medir-con-el-sistema-de-escala-de-usabilidad-sus/>. [Consultado: 05-may-2024].
- [17] E. Belinski, “Android API Levels”, Apilevels.com. [En línea]. Disponible en: <https://apilevels.com/>. [Consultado: 05-may-2024].
- [18] A. D. R. Formación, “Vinculación de tareas y sus dependencias en Project 2016 - ADR Formación”, Adrformacion.com, 17-mar-2017. [En línea]. Disponible en: https://www.adrformacion.com/knowledge/ingenieria-y-proyectos/vinculacion_de_tareas_y_sus_dependencias_en_project_2016.html. [Consultado: 05-may-2024].
- [19] I. Ramírez, “El mapa que muestra si se usa más Android o iOS: así queda el reparto de sistemas operativos móviles a nivel mundial”, Xatakamovil.com, 04-abr-2023. [En línea]. Disponible en: <https://www.xatakamovil.com/mercado/mapa-que-muestra-se-usa-android-ios-asi-queda-reparto-sistemas-operativos-moviles-a-nivel-mundial>. [Consultado: 05-may-2024].
- [20] “Cómo descargar Android Studio y App Tools”, Android Developers. [En línea]. Disponible en: <https://developer.android.com/studio?hl=es-419>. [Consultado: 05-may-2024].

- [21] “Android Studio Iguana”, Android Developers. [En línea]. Disponible en: <https://developer.android.com/studio/releases/past-releases/as-iguana-release-notes?hl=es-419>. [Consultado: 05-may-2024].
- [22] “React Native · Learn once, write anywhere”, Reactnative.dev. [En línea]. Disponible en: <https://reactnative.dev/>. [Consultado: 05-may-2024].
- [23] M. Frachet, “Comprender el concepto de puente de React Native”, Hackernoon.com, 10-nov-2017. [En línea]. Disponible en: <https://hackernoon.com/lang/es/compreesion-reaccionar-nativo-puente-concepto-e9526066ddb8>. [Consultado: 05-may-2024].
- [24] “IntelliJ IDEA – the leading Java and Kotlin IDE”, JetBrains. [En línea]. Disponible en: <https://www.jetbrains.com/idea/>. [Consultado: 05-may-2024].
- [25] “¿Qué es Visual Studio Code y cuáles son sus ventajas?”, Arsys. [En línea]. Disponible en: <https://www.arsys.es/blog/que-es-visual-studio-code-y-cuales-son-sus-ventajas>. [Consultado: 05-may-2024].
- [26] Marketing, “Visión por Computador. Qué es, Aplicaciones y Objetivos”, EDS Robotics, 31-ene-2022. [En línea]. Disponible en: <https://www.edsrobotics.com/blog/vision-computador-que-es/>. [Consultado: 12-ago-2024].
- [27] S. Mitra y T. Acharya, “Gesture Recognition: A Survey”, IEEE Trans. Syst. Man Cybern. C Appl. Rev., vol. 37, núm. 3, pp. 311–324, 2007.
- [28] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, y X. Twombly, “Vision-based hand pose estimation: A review”, Comput. Vis. Image Underst., vol. 108, núm. 1–2, pp. 52–73, 2007.
- [29] R. Redlich, “4 tipos de visión por computadora que deberías conocer (códigos incluidos)”, Bitstream, 14-mar-2024. [En línea]. Disponible en: <https://blog.zambia.cl/4-tipos-de-vision-artificial-que-deberias-conocer-codigos-incluidos/>. [Consultado: 12-ago-2024].
- [30] P. Potrimba, “What is keypoint detection?”, Roboflow Blog, 31-oct-2023. [En línea]. Disponible en: <https://blog.roboflow.com/what-is-keypoint-detection/>. [Consultado: 12-ago-2024].
- [31] “File:Visage technologies face tracking and analysis.Png”, Wikimedia.org. [En línea]. Disponible en: https://commons.wikimedia.org/wiki/File:Visage_Technologies_Face_Tracking_and_Analysis.png. [Consultado: 12-ago-2024].
- [32] “Android – 8.0 Oreo”, Android. [En línea]. Disponible en: <https://www.android.com/versions/oreo-8-0/>. [Consultado: 16-ago-2024].

- [33] “PostgreSQL”, PostgreSQL, 04-sep-2024. [En línea]. Disponible en: <https://www.postgresql.org/>. [Consultado: 16-ago-2024].
- [34] L. M. A. Extensions, “Visual Studio Code - code editing. Redefined”, Visualstudio.com. [En línea]. Disponible en: <https://code.visualstudio.com/>. [Consultado: 16-ago-2024].
- [35] P. Buitrago, “Kotlin vs Java: ¿Cuál usar para desarrollar tu nueva app?”, Wortise, 01-mar-2024. [En línea]. Disponible en: <https://wortise.com/blog/kotlin-vs-java-cual-usar-para-desarrollar-tu-nueva-app/>. [Consultado: 16-ago-2024].
- [36] “Fork - a fast and friendly git client for Mac and Windows”, Fork - a fast and friendly git client for Mac and Windows. [En línea]. Disponible en: <https://git-fork.com/>. [Consultado: 16-ago-2024].
- [37] “ML Kit”, Google for Developers. [En línea]. Disponible en: <https://developers.google.com/ml-kit?hl=es-419>. [Consultado: 01-jul-2024].
- [38] Mario, SQL Changes.Sql at development · venturamario/Exergames. .
- [39] Mario, INSERTS_Exergames.py at development · venturamario/Exergames. .
- [40] Wikipedia contributors, “Android Marshmallow”, Wikipedia, The Free Encyclopedia. [En línea]. Disponible en: https://es.wikipedia.org/w/index.php?title=Android_Marshmallow&oldid=161932890.
- [41] “Detecta rostros con ML Kit en Android”, Google for Developers. [En línea]. Disponible en: <https://developers.google.com/ml-kit/vision/face-detection/android?hl=es-419>. [Consultado: 06-jul-2024].
- [42] “ImageAnalysis”, Android Developers. [En línea]. Disponible en: <https://developer.android.com/reference/androidx/camera/core/ImageAnalysis>. [Consultado: 18-jul-2024].
- [43] “Permiso de tiempo de ejecución: notificaciones”, Android Developers. [En línea]. Disponible en: <https://developer.android.com/develop/ui/views/notifications/notification-permission?hl=es-419>. [Consultado: 18-jul-2024].
- [44] “System Usability Scale (SUS)”, Formaciongrafica.net. [En línea]. Disponible en: <https://www.formaciongrafica.net/blog/tutoriales-online/uxd/system-usability-scale-sus/>. [Consultado: 04-ago-2024].
- [45] A. Zunzunegui, “Escala Likert: qué es y cómo usarla en tus encuestas”, Blog de Acumbamail, 18-sep-2023. [En línea]. Disponible en: <https://acumbamail.com/blog/escala-likert/>. [Consultado: 04-ago-2024].

[46] Washingtonpost.com. [En línea]. Disponible en: <https://www.washingtonpost.com/news/to-your-health/wp/2014/12/22/puffer-the-great-atari-exercise-bike-that-never-was/>. [Consultado: 04-ago-2024].

[47] E. Albisúa, “La evolución de los videojuegos para ejercitarte: de los años 80 a nuestros días”, El Sol de Puebla | Noticias Locales, Policiacas, sobre México, Puebla y el Mundo, 23-nov-2021. [En línea]. Disponible en: <https://www.elsoldepuebla.com.mx/circulos/la-evolucion-de-los-videojuegos-para-ejercitarte-de-los-anos-80-a-nuestros-dias-7510207.html>. [Consultado: 23-ago-2024].

[48] Wikipedia contributors, “Power Pad”, Wikipedia, The Free Encyclopedia. [En línea]. Disponible en: https://es.wikipedia.org/w/index.php?title=Power_Pad&oldid=149200961.

[49] Wikipedia contributors, “Wii Balance Board”, Wikipedia, The Free Encyclopedia. [En línea]. Disponible en: https://es.wikipedia.org/w/index.php?title=Wii_Balance_Board&oldid=153684804.

[50] A. González, Wii Sports: el juego de deportes que cambió la forma de jugar. Nintenderos, 2022.

[51] M. B. Frankie, “EyeToy, la cámara mágica de PlayStation 2”, Vidaextra.com, 15-ago-2021. [En línea]. Disponible en: <https://www.vidaextra.com/ps2/eyetoy-camara-magica-playstation-2>. [Consultado: 23-ago-2024].

[52] G. R. Rocío, “Cómo funciona y en qué se usa la realidad virtual”, ADSLZone, 11-jun-2021. [En línea]. Disponible en: <https://www.adslzone.net/reportajes/tecnologia/realidad-virtual-rv/>. [Consultado: 23-ago-2024].

[53] A. S. Gillis, “augmented reality (AR)”, WhatIs, 21-mar-2024. [En línea]. Disponible en: <https://www.techtarget.com/whatis/definition/augmented-reality-AR>. [Consultado: 23-ago-2024].

[54] Wikipedia contributors, “Teorema de Pitágoras”, Wikipedia, The Free Encyclopedia. [En línea]. Disponible en: https://es.wikipedia.org/w/index.php?title=Teorema_de_Pit%C3%A1goras&oldid=162226410.

ANEXOS

ANEXO A.1. Evolución y etapas en la creación de exergames desde sus inicios

Como se ha mencionado en el [apartado 2.1](#), la creación de exergames se remonta a principios de los años 80. Han existido numerosos intentos por crear y popularizar exergames por parte de distintas compañías, haciendo que estos juegos hayan ido evolucionando progresivamente a lo largo de los años.

La evolución de los exergames puede dividirse en distintas etapas, descritas brevemente a continuación.

1. **Primeros intentos:** Los primeros intentos de crear exergames surgen en los años 80 y 90. Concretamente, en 1980 la empresa Atari, trató de combinar la actividad física con los videojuegos, creando como resultado a “HighCycle”, un videojuego que conectaba una bicicleta estática a unos sensores que permitían al usuario conocer la velocidad de la bicicleta, las calorías quemadas, etc. [\[46\]](#) [\[47\]](#). Seis años más tarde, Nintendo crearía el PowerPad, dispositivo para la Nintendo Entertainment System (NES) y el Sega Activator. Este era un dispositivo rudimentario que se usaba con los pies y permitía monitorizar la actividad del usuario para poder jugar a un total de 6 Exergames de carácter deportivo [\[48\]](#). Otros intentos destacables son Computainer, de Nintendo (1986) o Exertainment System, también de Nintendo (1993).
2. **Popularización:** La popularización de los exergames llegó en los años 2000, cuando Nintendo creó la “Wii”, marcando un punto de inflexión en la historia de los exergames. Gracias al controlador remoto “Wii Remote” y accesorios como “Wii Balance Board” [\[49\]](#), una plataforma que detectaba el equilibrio y peso del jugador se consiguió que los juegos y la actividad física se unieran y popularizaran. Juegos como “Wii Sports” [\[50\]](#) o “Wii Fit” marcaron un antes y un después en la industria e introdujeron de forma sólida el concepto de ejercicio mediante videojuegos.
3. **Innovación:** Tras la base sólida establecida por Nintendo, otras compañías comenzaron a innovar e investigar este campo. En 2005, Play Station en colaboración con Nike lanzó EyeToy [\[51\]](#), un videojuego para PlayStation 2 que introducía prácticas como combate, ejercicios de resistencia u otros. Microsoft lanzó Microsoft Kinect en 2010 para la Xbox 360, que usaba cámaras y sensores para detectar la actividad del jugador. Gracias a estos avances, empezaron a surgir videojuegos como “Just Dance” o “Kinect Sports”.

Con el auge de los smartphones, la innovación fue todavía mayor, propiciando la aparición de tecnologías de realidad virtual (VR) [\[52\]](#) y realidad aumentada (AR) [\[53\]](#).

- 4. Actualidad:** En la actualidad, una parte significativa de los exergames se juega en dispositivos móviles. Juegos como “Pokemon GO” utilizan Augmented Reality (AR) para hacer que sus jugadores tengan que salir al mundo real y moverse o realizar actividad física para cumplir con los objetivos del juego.

ANEXO A.2. Modelo final de datos implementado en la base de datos de exergames

Para la creación de una base de datos relacional funcional, el modelo de datos presentado en el [apartado 5.1.3](#) debe ser pasado a relacional y normalizado. Se comenzará pasando a relacional el modelo de datos siguiendo los cinco pasos típicos para ello. En estos cinco pasos para hacer el modelo relacional de la base de datos se usará el carácter # para indicar las claves primarias (Primary Key o PK) y se usarán subrayados para indicar las claves foráneas (Foreign Key o FK). A continuación, los 5 pasos del paso de modelo conceptual a relacional.

Pasos 1 y 2: Escribir las tablas y escoger/crear una clave para cada una.

USUARIO(**#username**, name, lastname, password, level, xp)
JUEGO(**#idGame**, name, description, difficulty, instructions)
PROBLEMA_CERVICAL(**#idCerv**, name)
PUNTOS(**#numPoints**, **#fechaHora**)

Pasos 3 y 4: Cada relación del modelo conceptual es una tabla del modelo relacional. Para cada una de estas tablas se escoge una clave primaria (PK).

R_USUARIO_USUARIO(**#username1**, **#username2**)
R_USUARIO_JUEGO(**#username**, **#numPoints**, **#fechaHora**)
R_PUNTOS_JUEGO(**#idPoints**, idGame)
R_PUNTOS_USUARIO(**#idPoints**, username)
R_JUEGO_PROBLEMA(**#idGame**, idCerv)

Paso 5: Se estudia la posibilidad de realizar una fusión de tablas y que aparezcan tablas nuevas. El modelo final de datos que se usará para desarrollar el código de creación de la base de datos es el siguiente:

USUARIO(**#username**, name, lastname, password, level, xp)
JUEGO(**#idGame**, name, description, instructions, difficulty, idCerv)
PROBLEMA_CERVICAL(**#idCerv**, name)
PUNTOS(numPoints, **#fechaHora**, **#idGame**, **#username**)
R_USUARIO_USUARIO(**#username1**, **#username2**)

Para la tabla PUNTOS se ha definido como clave primaria las combinaciones únicas de nombres de usuario (username), identificadores de exergames (idGame) y fechas (fechaHora). Esto se debe a la necesidad de almacenar las fechas y horas exactas en las que un usuario determinado ha obtenido una puntuación concreta en un exergame en específico. El nombre de usuario y el identificador del juego permiten conocer quién ha obtenido la puntuación y en qué exergame, pero usar solamente estos valores como clave primaria provocaría que solo se pudiera guardar una puntuación por cada usuario y cada juego, negando la posibilidad de monitorizar la evolución de

un usuario en un juego. Teniendo en cuenta que se quiere poder almacenar el progreso de cada usuario en cada juego, tiene que existir la posibilidad de que se almacenen varios registros para un mismo usuario y juego. El elemento que diferencia cada uno de estos registros es la fecha y la hora en la que se obtiene esta puntuación en concreto, y por esta razón se incluye la columna fechaHora en la clave primaria.

La [figura 33](#) muestra el modelo de datos final implementado para la aplicación de exergames. Para cada tabla se puede apreciar la visibilidad (pública o privada), el nombre de la tabla y los atributos. Además, para cada atributo, puede verse si es una clave primaria, foránea o ambas.

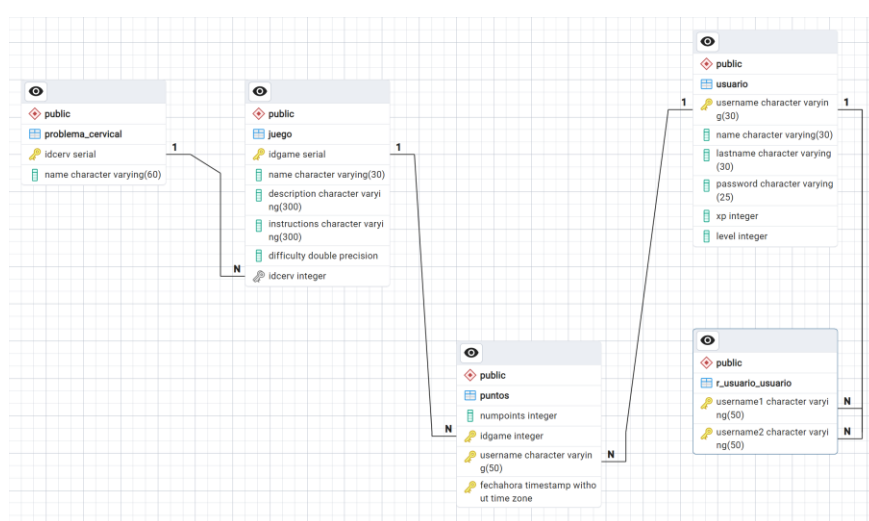


Figura 33. Modelo de datos que ilustra la estructura real implementada en la base de datos tras el paso a relacional y normalización.

Respecto a la normalización, se analiza la forma normal en la que se encuentra la base de datos tras el paso a relacional y se observa que:

- **La base de datos está en la primera forma normal (1FN)** porque ningún atributo de la relación es, en sí mismo, una relación. Es decir, todo atributo de la relación es atómico, no descomponible ni repetitivo.
- **La base de datos está en la segunda forma normal (2FN)** porque está en la primera forma normal y todos los atributos que no son parte de la clave principal (PK) dependen funcionalmente en forma completa de ella. Dependen de toda la clave y no solamente de parte de ella.
- **La base de datos está en la tercera forma normal (3FN)** porque está en la segunda forma normal y ningún atributo no clave depende funcionalmente de ningún atributo o subconjunto de ellos no clave. Es decir, los atributos que no forman parte de la clave son independientes entre ellos.

- **La base de datos está en la forma normal de Boyce-Codd (FNBC)** porque está en 1FN y todos los determinantes son clave candidata de la relación, es decir, no hay una clave candidata que apunte a una clave primaria de la relación. Dicho de otra manera, para cada dependencia funcional del tipo $X \rightarrow Y$, X es una superclave.

Dado el estado actual, ya se considera que el estado actual de la base de datos es suficiente para ser implementada sin buscar grados de normalización más elevados. Además, dada la naturaleza de la cuarta y quinta forma normal, muy probablemente se generarían más tablas, complicando innecesariamente la implementación de la base de datos y provocando futuros problemas de rendimiento al necesitar numerosas operaciones de **JOIN** para realizar consultas simples de información.

ANEXO A.3. Cálculo de velocidades en movimientos faciales en tiempo real y posibles aplicaciones en exergames

El cálculo de la velocidad con la que un usuario es capaz de hacer un movimiento con la cabeza se obtendrá en base al desplazamiento de las coordenadas detectadas en un intervalo de tiempo definido, de forma que cuanto mayor sea el desplazamiento en ese intervalo de tiempo, mayor será la velocidad con la que el usuario ha realizado el movimiento. Esto podría ser de utilidad en juegos en los que se plantee acciones basadas en la “fuerza” de movimientos. En un contexto de rehabilitación, podría ayudar a medir la mejora en la capacidad de los pacientes para controlar movimientos específicos.

Suponiendo que se está consiguiendo detectar rostros en el proceso de detección en tiempo real, y por tanto, se está pudiendo detectar las coordenadas de la nariz, puede obtenerse la distancia entre ambas coordenadas mediante el cálculo de la distancia entre dos puntos, que no es lo mismo que obtener la distancia entre dos coordenadas en X y en Y, ya que ahora se pretende conocer la distancia absoluta entre ambas coordenadas, teniendo en cuenta X e Y a la vez.

Esto se entiende mejor si se plantea como puntos sobre un mismo plano, tal y como muestra la [figura 34](#), donde se puede ver, para los puntos A y B, la distancia en X, la distancia en Y, y la distancia absoluta.

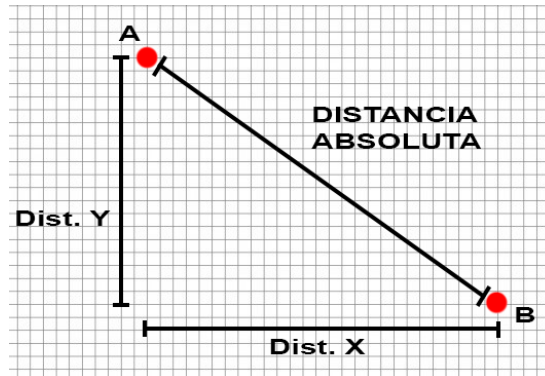


Figura 34. Distancia absoluta entre dos puntos.

Dados estos dos puntos sobre un mismo espacio, para conocer la distancia absoluta entre ambos, se puede aplicar la fórmula de la distancia euclidiana, donde debemos obtener el valor del módulo $|\overrightarrow{AB}|$.

$$d(A, B) = |\overrightarrow{AB}| = \left| \sqrt{(X_B - X_A)^2 + (Y_B - Y_A)^2} \right|$$

Alternativamente, podremos llegar a la misma conclusión si hacemos el cálculo aplicando el conocido Teorema de Pitágoras [54], que se usa para calcular la longitud de la hipotenusa o de los catetos en triángulos rectángulos. Si nos fijamos en la [figura 34](#), nos daremos cuenta de que casi siempre se puede dibujar un triángulo rectángulo entre dos puntos, donde dos de los vértices de la figura (los que forman la hipotenusa) son las coordenadas detectadas de la nariz. El valor de los catetos será el valor obtenido de las distancias en X e Y, y la hipotenusa, a la que llamaremos h será exactamente la distancia $d(A, B)$ calculada anteriormente. De esta forma el cálculo de la distancia puede hacerse aplicando este teorema y se llegará a la misma conclusión.

Con el Teorema de Pitágoras se obtendría lo siguiente:

$$h^2 = c_1^2 + c_2^2 \quad \rightarrow \quad h = \sqrt{c_1^2 + c_2^2}$$

Y como los catetos c_1 y c_2 son, respectivamente, la distancia en X e Y, entonces:

$$distancia\ absoluta = \sqrt{dist(X)^2 + dist(Y)^2}$$

Se puede apreciar una única diferencia entre ambas fórmulas: el uso del valor absoluto. Si se calcula la distancia entre ambos puntos obteniendo el valor del módulo, la fórmula incluye siempre el uso del valor absoluto para evitar problemas asociados a valores negativos inesperados. Sin embargo, el Teorema de Pitágoras no introduce el uso del valor absoluto ya que el teorema se usa para calcular la longitud de los costados de un triángulo, que nunca pueden tener valor negativo. Por simplicidad, se recomienda usar la fórmula de la distancia euclidiana.

De esta forma, si se establece un intervalo de tiempo de, por ejemplo, 5 segundos, puede calcularse la distancia absoluta del movimiento que ha realizado un usuario con la cabeza mediante la aplicación de las operaciones descritas, siendo A el punto en el que se encuentra la nariz al inicio de los 5 segundos, y B el punto en el que se encuentra la nariz tras 5 segundos. Cuanto mayor sea la distancia absoluta entre coordenadas, mayor será el movimiento que ha realizado el usuario.

Esto podría permitir implementar exergames que no solamente usen la detección de coordenadas para realizar movimientos hacia la izquierda, derecha, arriba o abajo; sino que también se podría usar la distancia absoluta entre dos puntos en un intervalo definido de tiempo para simular movimientos que requieran algún tipo de fuerza como golpear, lanzar algo, etc.

La [figura 35](#) muestra algunas ideas de exergames que podrían implementarse y ser jugados siguiendo esta lógica.

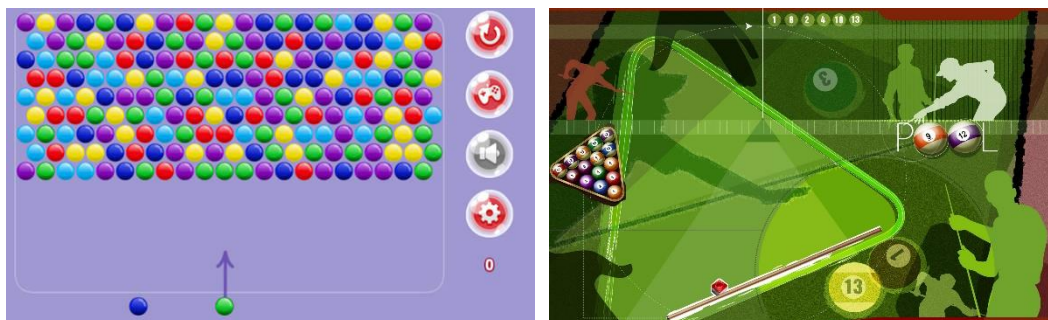


Figura 35. Juegos que podrían implementarse a modo de exergame que incluye la simulación de fuerzas.

A nivel de exergaming, muchos juegos o deportes reales que impliquen el uso de algún tipo de fuerza pueden implementarse siguiendo la lógica descrita a continuación. Puede usarse la distancia absoluta en intervalos cortos de tiempo para deducir la velocidad o agresividad con la que los usuarios realizan algunos movimientos, y usarse para simular la fuerza de lanzamientos en juegos clásicos como el billar, los bolos, o juegos de combate o boxeo donde se simula fuerza del golpe, entre otros.

A nivel de rehabilitación, la monitorización de cómo los usuarios hacen estos movimientos también podría ayudar a llevar un registro de cómo están evolucionando. Por ejemplo, suponiendo la existencia de un usuario con dificultades para realizar movimientos de cabeza con cierta velocidad, inicialmente podría observarse que la distancia absoluta promedio de sus movimientos es de cierta cantidad en concreto. Si tras un periodo de tiempo jugando a los exergames, la aplicación muestra que esta cantidad ha subido, es decir, la distancia absoluta promedio de sus movimientos ha incrementado, podría deducirse que el usuario está evolucionando favorablemente ya que no presenta tantas dificultades para mover la cabeza con velocidad y soltura.

Esta mejora se podría producir gracias a la motivación de los usuarios para jugar, haciendo que la aplicación de exergaming se convierta en la herramienta con la que hacer terapia y, a la vez, con la que evaluar cómo está evolucionando un paciente en ella.

ANEXO A.4. Medición de la usabilidad de la aplicación con el Sistema de Escalas de Usabilidad (SUS)

Tal y como se ha mencionado en el [apartado 6.3](#), el cálculo detallado sobre la puntuación de la aplicación “Exergames” usando el sistema SUS requiere encuestar a una serie de usuarios del sistema sobre 10 enunciados usando la escala de Likert, es decir, con 5 posibles respuestas cada uno.

Los enunciados predefinidos son los siguientes:

1. Creo que me gustaría utilizar este sistema con frecuencia.
2. Encontré el sistema innecesariamente complejo.
3. Pensé que el sistema era fácil de usar.
4. Creo que necesitaría el apoyo de un técnico para poder utilizar este sistema.
5. Encontré que las diversas funciones de este sistema estaban bien integradas.
6. Pensé que había demasiada inconsistencia en este sistema.
7. Me imagino que la mayoría de la gente aprendería a utilizar este sistema muy rápidamente.
8. Encontré el sistema muy complicado de usar.
9. Me sentí muy seguro usando el sistema.
10. Necesitaba aprender muchas cosas antes de empezar con este sistema.

Las posibles respuestas que usan la escala de Likert son:

1. Totalmente en desacuerdo
2. En desacuerdo
3. Neutro
4. De acuerdo
5. Totalmente de acuerdo

Puede verse que los enunciados pares afirman cosas negativas sobre la usabilidad del sistema y los enunciados impares afirman cosas positivas, y esto se usará para obtener la puntuación del sistema.

Para los enunciados impares, se sumará las respuestas (totalmente en desacuerdo equivale a 1, en desacuerdo equivale a 2, etc.) y se restará el valor 5:

$$(4 + 4 + 4 + 4 + 3) - 5 = 19 - 5 = \mathbf{14}$$

Para los enunciados pares, se restará a 25 el valor de la suma de las respuestas:

$$25 - (2 + 2 + 1 + 2 + 3) = 25 - 10 = \mathbf{15}$$

Por último, se suman los dos valores obtenidos y se multiplica el resultado por 2'5:

$$(14 + 15) \times 2'5 = 29 \times 2'5 = \mathbf{72'5}$$

De esta forma puede deducirse que, debido a que se ha obtenido una puntuación entre 68 y 80, el sistema presenta una usabilidad buena o aceptable, pero mejorable en algunos aspectos.

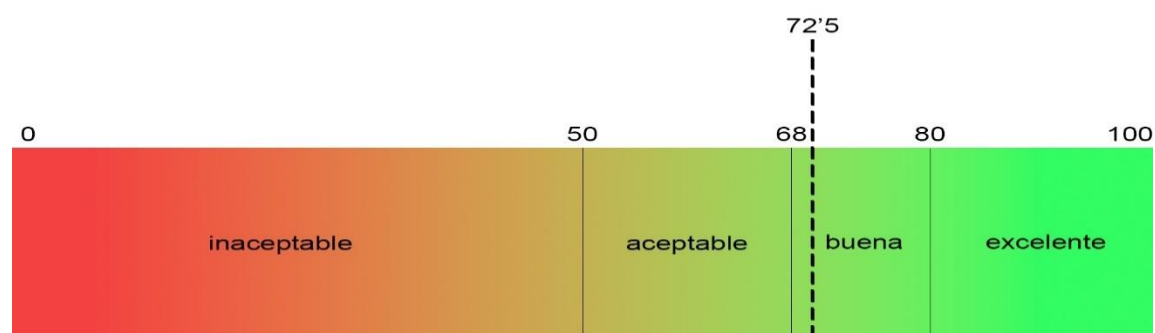


Figura 36. Puntuación obtenida en el SUS para la aplicación de exergames.