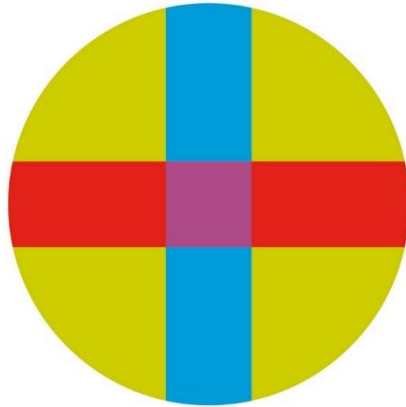


UNIVERSIDAD CEU - SAN PABLO

ESCUELA POLITÉCNICA SUPERIOR

GRADO EN INGENIERÍA DE SISTEMAS DE TELECOMUNICACIÓN



TRABAJO FIN DE GRADO

Diseño de una aplicación educativa para conocimiento biomédico

Autor: María Victoria Ayuso Acedo
Director: Carlos Oscar Sorzano

Junio 2022



Datos del alumno

NOMBRE: MARIA VICTORIA AYUSO ACEDO

Datos del Trabajo

TÍTULO DEL PROYECTO: Diseño de una aplicación educativa para conocimiento biomédico

Tribunal calificador

PRESIDENTE:

FDO.:

SECRETARIO:

FDO.:

VOCAL:

FDO.:

Reunido este tribunal el ____/____/____, acuerda otorgar al Trabajo Fin de Grado presentado por Don _____ la calificación de _____.

RECONOCIMIENTOS

Una vez terminado el trabajo, quería agradecer a todas aquellas personas que me han permitido llegar hasta aquí.

Quería darle las gracias a mi familia y a mis amigos en especial a mis padres, mi pareja y mi mejor amiga.

A mi madre por dedicarme toda su vida en cuerpo y alma. Por ser mi confidente y mi roca, y no dejarme caer nunca. A día de hoy, no estaría aquí si no fuera por ella y jamás podre reflejar en palabras lo que significa para mí, mas que es lo más importante de mi vida.

A mi padre por creer en mi pase lo que pase y no darse por vencido. Por demostrarme más que nadie y quererme hasta el infinito incondicionalmente. Gracias por acompañarme y apoyarme durante toda mi carrera, que es la tuya.

A mi pareja por aguantarme contra viento y marea, seguir apostando por mí y quererme hasta el infinito. Y por último a mi mejor amiga Celia, que no solo es lo mejor que llevo de la universidad, sino que ella se ha convertido en parte esencial de mi vida.

Por último, agradecer a mi tutor Carlos Oscar el esfuerzo y la ayuda que me ha proporcionado en la realización de este trabajo.

1 ÍNDICE

ÍNDICE DE CONTENIDOS

1 ÍNDICE.....	1
2 RESUMEN	5
3 OBJETIVO	6
4 INTRODUCCIÓN	8
4.1 EVOLUCIÓN DE LAS COMUNICACIONES MÓVILES, LOS TERMINALES MÓVILES INTELIGENTES Y SUS SISTEMAS OPERATIVOS ASOCIADOS	9
4.2 ESTADO DEL ARTE EN APLICACIONES COMO LA PROPUESTA EN ESTE TRABAJO	11
5 SOLUCIÓN PROPUESTA.....	19
6 METODOLOGÍA.....	21
6.1 TIPOS DE USUARIOS DE LA APLICACIÓN / ROLES	21
6.2 REQUERIMIENTOS FUNCIONALES	22
6.2.1 Principales funciones integradas en la aplicación	23
6.2.2 Estrategia de generación de preguntas de test de la aplicación.....	29
6.2.3 Flujo lógico para la generación de preguntas “automáticas” a partir de la base de datos Copenmed	31
6.3 REQUERIMIENTOS NO FUNCIONALES	34
6.4 REQUERIMIENTOS LEGALES	36
7 DESARROLLO	38
7.1 ARQUITECTURA HARDWARE.....	38
7.2 ARQUITECTURA SOFTWARE	39
7.2.1 Características principales de la arquitectura software.....	39
7.2.2 Diagrama de navegación a través de la aplicación.....	41
7.2.3 Estructura del software desarrollado para la aplicación	41
7.2.4 Detalle de la generación de preguntas de test	44
7.2.5 Pantallas (mockups) de la aplicación	48
8 CONCLUSIONES	52
9 LÍNEAS FUTURAS DE TRABAJO	55
10 REFERENCIAS.....	57

ÍNDICE DE FIGURAS

FIGURA 1 ESTRUCTURA DE HOJAS DEL FICHERO EXCEL DE DESCARGA DE INFORMACIÓN DE COPENMED. ...	31
FIGURA 2 EJEMPLO DE TIPO DE ENTIDAD EN LA BASE DE DATOS EXCEL (HOJA “TIPOS_ENTIDAD”)	32
FIGURA 3 EJEMPLO DE ENTIDAD EN LA BASE DE DATOS EXCEL (HOJA “ENTIDADES”).....	32
FIGURA 4 EJEMPLO DE ASOCIACIÓN EN LA BASE DE DATOS EXCEL (HOJA “TIPOS_RELACIONES”).....	32
FIGURA 5 EJEMPLO DE POSIBLE COMBINACIÓN DE ENTIDAD2 PARA UNA ENTIDAD1 Y TIPOASOCIACION ESPECÍFICAS EN LA BASE DE DATOS EXCEL (HOJA “RELACIONES”)	34
FIGURA 6 DETALLES DE CONFIGURACIÓN DEL SDK	40
FIGURA 7 DETALLES DE CONFIGURACIÓN DEL ENTORNO DE DESARROLLO FIREBASE	40
FIGURA 8 DETALLES DE CONFIGURACIÓN DEL FRAMEWORK DE NAVEGACIÓN ENTRE ELEMENTOS DE LA APLICACIÓN	40
FIGURA 9 DIAGRAMA DE NAVEGACIÓN ENTRE LOS DIFERENTES “FRAGMENTS” DESARROLLADOS PARA LA APLICACIÓN	41
FIGURA 10 NIVEL 1 / BLOQUES DEL PROGRAMA DESARROLLADO	41
FIGURA 11 NIVEL 2 / BLOQUES DEL PROGRAMA DESARROLLADO / DETALLE DE “APP / MANIFEST”.....	42
FIGURA 12 NIVEL 3 / BLOQUES DEL PROGRAMA DESARROLLADO / CONTENIDO DEL FICHERO “ANDROIDMANIFEST”	42
FIGURA 13 NIVEL 5 / BLOQUES DEL PROGRAMA DESARROLLADO / DETALLE DE LOS “ADAPTERS” DESARROLLADOS.....	42
FIGURA 14 NIVEL 5 / BLOQUES DEL PROGRAMA DESARROLLADO / DETALLE DE LOS “MODELS” DESARROLLADOS.....	42
FIGURA 15 NIVEL 5 / BLOQUES DEL PROGRAMA DESARROLLADO / DETALLE DE LOS “VIEWS” DESARROLLADOS.....	43
FIGURA 16 NIVEL 5 / BLOQUES DEL PROGRAMA DESARROLLADO / DETALLE DEL OBJETO “MAINACTIVITY”	43
FIGURA 17 NIVEL 4 / BLOQUES DEL PROGRAMA DESARROLLADO / DETALLE DE LOS “TEST” DE LA APLICACIÓN	43
FIGURA 18 NIVEL 3 / BLOQUES DEL PROGRAMA DESARROLLADO / DETALLE DE LOS DIRECTORIOS “ASSETS” Y “RES”	44
FIGURA 19 NIVEL 1 / BLOQUES DEL PROGRAMA DESARROLLADO / DETALLE DEL DIRECTORIO “GRADLE SCRIPTS”.....	44
FIGURA 20 NIVEL 5 / BLOQUES DEL PROGRAMA DESARROLLADO / DETALLE DEL DIRECTORIO “QUIZFRAGMENT”.....	44
FIGURA 21 FRAGMENTO “QUIZFRAGMENT” / INICIALIZACIÓN DE “ASSETMANAGER”	45
FIGURA 22 FRAGMENTO “QUIZFRAGMENT” / DETERMINACIÓN DEL NÚMERO DE PREGUNTAS POR TEST.....	45
FIGURA 23 FRAGMENTO “QUIZFRAGMENT” / OBTENCIÓN DE PREGUNTAS DESDE LA BASE DE DATOS – 1/2	45
FIGURA 24 FRAGMENTO “QUIZFRAGMENT” / OBTENCIÓN DE PREGUNTAS DESDE LA BASE DE DATOS – 2/2	46
FIGURA 25 FRAGMENTO “QUIZFRAGMENT” / GENERACIÓN AUTOMÁTICA DE PREGUNTAS / INICIO – 1/2.....	46

FIGURA 26 FRAGMENTO “QUIZFRAGMENT” / GENERACIÓN AUTOMÁTICA DE PREGUNTAS / INICIO – 2/2....	46
FIGURA 27 FRAGMENTO “QUIZFRAGMENT” / GENERACIÓN DE RESPUESTAS VERDADERAS Y ERRÓNEAS....	47
FIGURA 28 FRAGMENTO “QUIZFRAGMENT” / GENERACIÓN DE UN BLOQUE DE 25 PREGUNTAS.....	47
FIGURA 29 PÁGINA SIG IN.....	48
FIGURA 30 PÁGINA SIG UP.....	48
FIGURA 31 PÁGINA DE EXAMEN / USUARIO NO REGISTRADO / “HOMEFRAGMENT”.....	48
FIGURA 32 PÁGINA DE EXAMEN / USUARIO REGISTRADO / “HOMEFRAGMENT”.....	49
FIGURA 33 PÁGINA DE EXAMEN / USUARIO REGISTRADO / SELECCIÓN DE PALABRA CLAVE / “HOMEFRAGMENT”.....	49
FIGURA 34 PÁGINA DE EXAMEN / GESTIÓN DE LAS RESPUESTAS / “QUIZFRAGMENT”.....	49
FIGURA 35 PÁGINA DE RESULTADOS PARA USUARIO REGISTRADO / “RESULTFRAGMENT”.....	50
FIGURA 36 PÁGINA DE RESULTADOS PARA USUARIO NO REGISTRADO / “RESULTFRAGMENT”.....	50
FIGURA 37 PÁGINA DE CONSTRUCCIÓN MANUAL DE PREGUNTAS / “ADDQUESTIONFRAGMENT”.....	50
FIGURA 38 PÁGINA DE EDICIÓN DE PREGUNTAS DE EXAMEN / “ADDQUESTIONFRAGMENT”.....	51

ÍNDICE DE TABLAS

TABLA 1. FUNCIÓN “CREACIÓN DE UNA NUEVA CUENTA”.....	23
TABLA 2. FUNCIÓN “ACCESO PARA USUARIOS REGISTRADOS”.....	23
TABLA 3. FUNCIÓN “ACCESO PARA USUARIOS NO REGISTRADOS”.....	24
TABLA 4. FUNCIÓN “GESTIÓN DE PREGUNTAS REALES DE TEST YA CARGADAS”.....	24
TABLA 5. FUNCIÓN “CARGA MANUAL DE NUEVAS PREGUNTAS”.....	25
TABLA 6. FUNCIÓN “PRUEBA ALEATORIA PARA USUARIO REGISTRADO_1”.....	25
TABLA 7. FUNCIÓN “PRUEBA ALEATORIA PARA USUARIO REGISTRADO_2”.....	26
TABLA 8. FUNCIÓN “PRUEBA POR PALABRA CLAVE PARA USUARIO REGISTRADO”.....	26
TABLA 9. FUNCIÓN “PRUEBA POR ESPECIALIDAD MÉDICA PARA USUARIO REGISTRADO”.....	27
TABLA 10. FUNCIÓN “PRUEBA ALEATORIA PARA USUARIO NO REGISTRADO_1”.....	27
TABLA 11. FUNCIÓN “PRUEBA ALEATORIA PARA USUARIO NO REGISTRADO_2”.....	28
TABLA 12. FUNCIÓN “PRUEBA POR PALABRA CLAVE PARA USUARIO NO REGISTRADO”.....	28
TABLA 13. FUNCIÓN “PRUEBA POR ESPECIALIDAD MÉDICA PARA USUARIO NO REGISTRADO”.....	29
TABLA 14. FUNCIÓN “LOGOUT”.....	29
TABLA 15. CARACTERÍSTICAS PRINCIPALES DE LA ARQUITECTURA SOFTWARE DE LA APLICACIÓN.....	39

2 RESUMEN

El objetivo de este trabajo es la creación de una aplicación para dispositivos móviles (smartphones y tablets) capaz de evaluar el conocimiento de sus usuarios sobre temas biomédicos. Dicha evaluación se apoya en la realización de tests, basados en preguntas con opciones múltiples de respuesta y una sola válida.

Para la construcción de las preguntas se ha optado por dos estrategias posibles. Por una parte, el uso de bases de datos de conocimiento biomédico estructurado, basadas en páginas web de contenido abierto que permiten la generación de preguntas de manera automática. En este caso se ha usado la web Copenmed.

Y, por otra parte, el uso de preguntas reales, habitualmente asociadas a casos clínicos, cuyo origen, en nuestro caso y por simplicidad, es el examen MIR (Médico Interno Residente) que es una oposición de acceso a una plaza de médico especialista en formación en el Sistema Nacional de Salud español.

En conclusión, la plataforma desarrollada gestiona un entorno de gestión de tests, permitiendo administrar la progresión del usuario en su entrenamiento. Y todo esto basado en el uso de preguntas / respuestas correctamente formuladas, con opciones de respuesta realistas y con posibilidad de actualización periódica.

3 OBJETIVO

El objetivo de este trabajo es la realización de una plataforma (en adelante aplicación) capaz de ser ejecutada en dispositivos móviles con sistema operativo Android, potencialmente disponible a través de sistemas de distribución como 'Play Store' y de fácil instalación.

La aplicación debe ser capaz de generar un entorno confiable, simple y fácil de usar para personas vinculadas con temas biomédicos, cuyo objetivo prioritario sea la gestión del conocimiento.

En base a lo anterior, la aplicación debe ser capaz de actualizar su base de datos de preguntas de forma sencilla y, sobre todo, debe asegurarse de que dichas preguntas sean reales o, al menos, realistas y fiables. Nos referimos a que, independientemente de la fuente de información utilizada, los criterios de 'respuesta correcta' y 'respuesta incorrecta' deben ser sólidos, fundamentalmente cuando no se transcriben preguntas “reales” (realmente usadas), sino que se construyen preguntas de “manera automática”.

Pensando en el usuario de la aplicación, ésta debe ofrecer un entorno amigable para su uso y debe permitir que la autoevaluación del usuario sea simple y efectiva. Saber en todo momento si las respuestas son correctas o no, y tras cada prueba, el nivel de éxito sobre cada posible disciplina/especialidad se convierten en criterios de aplicación necesarios.

En base a todo lo anterior, los principales objetivos a cubrir con la aplicación son:

- 1) Desarrollo de una aplicación susceptible de ser utilizada en dispositivos móviles tipo smartphone y tablet, basados en el sistema operativo Android.
- 2) Fácil de usar por todo tipo de personas, independientemente de su nivel de cultura informática.
- 3) Con mecanismos sencillos de generación/actualización de preguntas de examen.
- 4) Con mecanismos de autoevaluación que permitan cuantificar el progreso en la formación.

La información de partida para la aplicación desarrollada es aquella relacionada con las preguntas a utilizar en los tests. Tener la capacidad de generar / utilizar preguntas reales, bien estructuradas y correctamente configuradas es fundamental.

En línea con lo anterior, en este trabajo se ha establecido como objetivo el abordar dos estrategias para la actualización de dicha información:

- La 'manual', mediante el acceso a publicaciones que recogen preguntas reales de tests y gestionando su transcripción. Un derivado de esta opción sería el volcado de preguntas ya digitalizadas en la propia aplicación, lo cual no ha sido abordado en la aplicación desarrollada.
- La 'automática', accediendo a bases de datos relacionales que permitan determinar vinculaciones entre conceptos y que establezcan criterios para determinar asociaciones correctas que deriven en respuestas acertadas a preguntas generadas. En este caso, en el desarrollo de la aplicación se han tenido en cuenta algunas consideraciones tales como:
 - Disponibilidad/accesibilidad de dichas bases de datos.
 - Idioma en el que se recoge la información en dichas bases.
 - Capacidad de estructurar preguntas a partir de la información recogida en ellas.
 - Necesidad de aplicar algún tipo de tratamiento del lenguaje.
 - Fiabilidad de la información.
 - Capacidad para detectar respuestas correctas y respuestas incorrectas pero coherentes.

Por tanto, conseguir para nuestra aplicación un entorno operativo sencillo de utilizar, fácil de adaptar a diferentes fuentes de información, modular y escalable, así como sencilla de instalar y mantener, nos parece esencial.

A lo largo de este trabajo, el autor pretende trasladar los conceptos clave aquí presentados, justificando en la medida de lo posible las estrategias, decisiones de diseño y prioridades aplicadas.

4 INTRODUCCIÓN

En la actualidad, el desarrollo de las tecnologías de la información y las comunicaciones está llegando prácticamente a todos los ámbitos de la vida de las personas y, en particular, a la educación.

La capacidad de computación de los dispositivos (PCs, ordenadores portátiles, tabletas, smartphones, ...), la disponibilidad de medios de comunicación cada vez más eficientes (fijos e inalámbricos) con mayor ancho de banda, y el cada vez mejor manejo de las bases de datos están permitiendo el desarrollo de aplicaciones más eficientes en la gestión de grandes volúmenes de información.

En concreto, el desarrollo de las comunicaciones móviles y el de los equipos que las soportan están permitiendo el desarrollo de multitud de aplicaciones que combinan conceptos como la ubicuidad y la inmediatez.

En base en lo anterior, el desarrollo de aplicaciones móviles de apoyo a la consolidación de conocimiento parece de gran utilidad ya que éstas garantizan prestaciones iguales o superiores a los medios tradicionales fijos y portátiles, permitiendo su uso en prácticamente cualquier lugar y condición y ofreciendo un entorno de información, educación y formación que permite al usuario utilizarlo de manera eficiente.

Adicionalmente, los temas biomédicos y todo lo que se mueve en torno a ellos están de gran actualidad en este momento y son fuente de un volumen ingente de información a gestionar.

Por lo tanto, trabajando en el entorno del conocimiento biomédico, desarrollar herramientas que permitan a sus usuarios, de forma cómoda y con una disponibilidad muy amplia, realizar tests de conocimiento puede constituir un mecanismo poco estresante y de gran utilidad para que consoliden conocimientos y se mantengan actualizados a través de la propia actualización de las preguntas utilizadas.

4.1 Evolución de las comunicaciones móviles, los terminales móviles inteligentes y sus sistemas operativos asociados

Desde un principio, el objetivo de este trabajo ha sido desarrollar la aplicación en un entorno 100% móvil, basado en dispositivos tipo smartphone y tablet, entendiendo esto como un requisito básico para conseguir la máxima usabilidad, sencillez y ubicuidad. En este apartado pretendemos justificar que nuestra decisión es adecuada tanto con respecto a estos dispositivos como en relación con las tecnologías de comunicación que los soportan.

Partiendo del análisis de las comunicaciones que utilizan los smartphones (fundamentalmente inalámbricas), podemos establecer una clara distinción entre los canales basados en tecnología WiFi y los basados en tecnología móvil digital. En ambos casos hablamos de comunicaciones 100% digitales y con velocidades de transmisión crecientes.

Actualmente la tecnología WiFi [1] es el principal modo de acceso a Internet (fuente Cisco) y, especialmente en España, supone gestionar dos tercios del tráfico de datos de Internet y el 50% del tráfico generado a nivel mundial. El último estándar WiFi que se está implementando actualmente es el 802.11ax, denominado “WiFi 6”, que representa un avance significativo en términos de:

- Velocidad: Hasta 4,8 Gbps por punto de acceso.
- Capacidad de transmisión de datos: Hasta un 400% de aumento con respecto a tecnologías anteriores.
- Latencia: Reducción significativa respecto de versiones anteriores.

En relación con las comunicaciones móviles [2], estas se diseñaron por primera vez 100% digitales en la década de los 80 y, desde entonces, no han parado de evolucionar, aumentando sus capacidades en cuanto a velocidad de transmisión, latencia, seguridad, confiabilidad, etc.

Por ejemplo, con respecto a la velocidad de transmisión, en cinco décadas se ha pasado de los 14 Kbps iniciales a los 10 Gbps actuales en las redes 5G. Y por lo que respecta a la latencia (un elemento importantísimo a la hora de caracterizar la experiencia

de usuario), actualmente se trabaja en redes 4G con latencias inferiores a 200 ms, que se reducen a 1 ms en las nuevas redes 5G.

En relación con los dispositivos móviles [3], actualmente casi todos se basan en el concepto de 'smartphone', que significa hablar de un dispositivo que combina las funciones de un teléfono móvil y una computadora. Evidentemente, estos dispositivos operan sobre plataformas informáticas móviles, con alta capacidad de almacenamiento de información y la posibilidad de realizar tareas simultáneamente. El término "inteligente" se usa principalmente con fines comerciales para distinguir estos dispositivos de los teléfonos celulares básicos. Desde principios de este siglo XXI (1ª década), se lanzaron los que podríamos llamar 'smartphones modernos', los cuales adquirieron una gran popularidad en la década siguiente, superando en ventas a los celulares básicos.

Una de las características más significativas de estos teléfonos es la pantalla táctil capacitiva, que básicamente reemplaza los botones y aumenta la capacidad de la pequeña pantalla visual anterior de los teléfonos convencionales. Otras características relevantes de estos dispositivos son:

- Acceso a Internet a través de WiFi o redes móviles (desde 2G hasta 5G).
- Función multitarea.
- Funciones multimedia (cámara y reproductor de audio/video),
- Disponibilidad de instalación de aplicaciones adicionales, generalmente de terceros, que permiten que estos teléfonos soporten muchas funciones en diferentes áreas.

Finalmente, y en relación con los principales sistemas operativos que soportan el funcionamiento de los smartphones, diversas fuentes [4] consultadas establecen hasta en 13 los diferentes sistemas que se han desarrollado históricamente para soportar las funcionalidades de los smartphones. Actualmente, de todos ellos, solo dos se consideran los más utilizados: Android e iOS, apoyados por marcas como: Xiaomi, Samsung, Apple, Oppo, Realme, Vivo, Motorola y Huawei.

Tras este análisis, parece razonable pensar que la selección de un dispositivo smartphone (y, por extensión, aquellos tipo "tablet") para soportar nuestra aplicación

proporciona los recursos suficientes para garantizar una tecnología robusta, una interfaz de usuario lo suficientemente amigable, una experiencia de usuario adecuada y una plataforma de desarrollo alineada con el tipo de servicios que queremos ofrecer.

4.2 Estado del arte en aplicaciones como la propuesta en este trabajo

Como se mencionó anteriormente, el objetivo de este trabajo es el desarrollo de una aplicación capaz de generar un entorno confiable, simple y fácil de usar para personas vinculadas con temas biomédicos, cuyo objetivo prioritario sea la gestión del conocimiento.

Obviamente, dejando de lado aspectos básicos de diseño de la aplicación, en cuanto a su usabilidad, el elemento fundamental a considerar aquí es el contenido a manejar, es decir, las preguntas de test.

Una forma sencilla pero básica y garantizada de asegurar la utilidad de la aplicación es recurrir a fuentes de información que periódicamente recogen preguntas reales sobre conocimientos biomédicos, por ejemplo, de ediciones ya realizadas de la oposición MIR. Esto asegura que disponemos de material real utilizable de manera segura en el proceso de consolidación del conocimiento.

La alternativa a lo anterior es construir nuevas preguntas que respondan a conocimientos reales de temas biomédicos. La complejidad de seguir este camino radica en que debemos contar con:

- Fuentes confiables de información.
- Estas fuentes deben ser capaces de establecer relaciones entre conceptos diversos.
- Asimismo, estas fuentes deben permitir el establecimiento de mecanismos que permitan de alguna manera “crear razonamientos” y, por tanto, establecer ciertos vínculos entre conceptos. Por ejemplo:
 - “Concepto A puede tratarse con Concepto B”.
 - “Concepto C puede causar Concepto D”.

- En general, estas estructuras de bases de datos difícilmente podrán presentar, al menos actualmente, enfoques como “casos clínicos”. Solo parecen poder abordar, por el momento, estructuras lógicas en las que se establecen relaciones de 'causa - efecto' entre entidades (como se mencionó anteriormente).
- Finalmente, estas fuentes deben ser capaces de determinar vínculos erróneos y plausibles entre conceptos. Esto nos llevará a establecer respuestas “erróneas” y “correctas” respectivamente. Recordemos que se necesita desarrollar preguntas de test de respuesta múltiple, con una sola de ellas correcta.

La búsqueda del estado del arte al respecto nos ha llevado a investigar en diferentes fuentes:

- Bases de datos de preguntas reales de tests como el mencionado MIR y los recursos disponibles actualmente para la gestión online de los procesos de formación y entrenamiento a través de tests.
- Estructuras de bases de datos que se pueden utilizar para crear patrones de preguntas a partir de información médica.

Partiendo de la primera línea de investigación y simplemente haciendo una búsqueda en Internet sobre ‘Preguntas del examen MIR’ (dado que actualmente puede que sea una de las mayores fuentes de información sobre preguntas “reales” sobre temas biomédicos), obtenemos como resultado una gran variedad de posibilidades:

- Páginas web tales como:
 - PrepMIR [5]
 - Mirial [6]

que basan su servicio en la creación de simuladores de exámenes MIR, a través de preguntas reales, ofreciendo servicios tales como:

- Actualización frecuente de preguntas.
- Inclusión de comentarios y explicaciones destinadas a ayudar a los usuarios a comprender las respuestas correctas.

- Creación de rankings de participantes.
- Desarrollo de estrategias para la memorización efectiva de conceptos.
- Tutorías en línea.
- Diferentes publicaciones impresas [7] que recogen bloques de preguntas de varias ediciones del examen, incluyendo las respuestas correctas en cada caso.
- Aplicaciones para smartphones [8] con diferentes niveles de servicio:
 - “MirMeApp”:
 - Funciona con preguntas reales de examen.
 - Permite adaptar las respuestas a la estructura vigente en cada momento para la oposición (número de respuestas posibles).
 - Permite al usuario configurar hasta 3 tipos de test diferentes: de 10, 25 y 50 preguntas.
 - Permite guardar preguntas consideradas especialmente complicadas.
 - Es una “aplicación de pago” disponible para sistemas operativos iOS y Android.
 - “MIR” (Akaro):
 - Funciona con preguntas reales de examen.
 - Permite seleccionar preguntas de convocatorias específicas.
 - También permite conocer, tras contestar cada pregunta, si la respuesta es correcta o no.
 - Permite dejar la prueba sin terminar y continuar más tarde.
 - También es posible elegir el modo aleatorio para que las preguntas se realicen sin un orden preestablecido.
 - Si no desea utilizar la aplicación en modo 'quiz', el usuario puede ir a la sección de revisión, de tal forma que la

- aplicación permite avanzar y retroceder sobre las preguntas a su conveniencia.
- También hay una sección de preguntas fallidas que se pueden rescatar para su tratamiento.
 - Es una aplicación gratuita disponible para sistema operativo Android.
- “MIR” (iCarpa):
 - Al igual que los casos anteriores, la aplicación da la opción de contestar preguntas del examen MIR de años anteriores.
 - También ofrece controlar el tiempo de cada prueba y guardar las preguntas favoritas y hasta un examen completo.
 - Permite ver los resultados según se contestan las preguntas y da la opción de responder preguntas aleatorias.
 - Es una aplicación gratuita disponible para sistema operativo Android.
 - “Desgloses MIR”:
 - Esta es la aplicación de la academia Grupo CTO.
 - Reúne preguntas oficiales de exámenes MIR de diversos años y permite repasarlas de una forma más amena.
 - El alumno puede gestionar en la aplicación los siguientes argumentos:
 - Seleccionar el nivel de dificultad.
 - Personalizar las preguntas por temas.
 - Revisar los comentarios una vez finalizada la prueba.
 - Es una aplicación gratuita disponible para sistemas operativos iOS y Android.

- “MIR Quiz”:
 - Es una aplicación desarrollada por la academia MIR Asturias.
 - Contiene más de 4.000 preguntas cortas de respuesta verdadero / falso y así como de opción múltiple.
 - Permite seleccionar el tiempo máximo permitido para responder cada pregunta y elegir el tema.
 - Con el fin de revisar los fallos, al final de la prueba el usuario puede comprobar los resultados y la puntuación obtenida de los aciertos y errores, así como la velocidad con la que se ha respondido el test.
 - Es una aplicación gratuita disponible para sistemas operativos iOS y Android.
- “VideoAmir”:
 - Es una aplicación desarrollada por la Academia Amir.
 - Ofrece una plataforma para seguir las clases impartidas por sus profesores, con explicaciones, esquemas y comentarios añadidos.
 - Las clases están divididas por materias y los profesores que las imparten son los oficiales de la academia.
 - Es una aplicación gratuita disponible para sistema operativo iOS.
- “Mirial”:
 - Es una aplicación desarrollada por la Mutual Médica.
 - Se basa en un juego en línea que permite dos usuarios por juego.
 - Es una especie de Trivial donde se deben responder preguntas que se pueden encontrar en los exámenes MIR.
 - Las preguntas se clasifican en siete categorías correspondientes a distintas especialidades médicas y existen hasta 1.000 preguntas diferentes.

- A medida que se ejecutan los juegos, la aplicación establece una clasificación de los jugadores.
- Es una aplicación gratuita disponible para sistemas operativos iOS y Android.

Siguiendo con la segunda línea de investigación, en primer lugar, es importante aclarar algunos conceptos importantes:

- Información biomédica: [9] Se define como el conocimiento explícito obtenido de investigaciones científicas, clínicas o de evaluación de servicios de salud que se publica formalmente o como artículos científicos en revistas biomédicas, tales como revisiones sistemáticas o narrativas, guías de práctica clínica o como informes de agencias de calificación. Este conocimiento se engloba dentro de las denominadas fuentes de información biomédica (FIB).
- Bases de datos de información biomédica: [9] Son fuentes de información independientes y objetivas, con un contenido de información estructurado según la metodología de la medicina basada en la evidencia y con un costo asequible al profesional o institución de salud.
- Clasificación de las bases de datos de información biomédica: [9] Clásicamente, las fuentes de información se han organizado en:
 - Primarias (revistas).
 - Secundarias (bases de datos).
 - Terciarias (libros).
- DeCS: Es el acrónimo de ‘Descriptores en Ciencias de la Salud’ y es el vocabulario estandarizado que utiliza descriptores para indexar artículos científicos y otros documentos en el campo biomédico.

Actualmente, las fuentes de información biomédica en biomedicina representan un escenario mucho más complejo, debido principalmente a:

- El desarrollo de nuevas tecnologías de la información.
- La medicina basada en la evidencia,

En línea con lo anterior, existen múltiples fuentes de información biomédica organizados según el nuevo modelo de clasificación de Haynes, denominado "5 S" y estructurado piramidalmente. Así, en la base de la pirámide se incluyen los artículos o estudios originales (fuentes primarias antiguas), y en la cúspide se encuentran los llamados sistemas inteligentes o informatizados de toma de decisiones, que responden a las preguntas clínicas con información analizada, sistematizada y sintetizada, según la metodología de la medicina basada en la evidencia.

Llegados hasta aquí, tras haber intentado aclarar algunos conceptos y, una vez más, simplemente haciendo una búsqueda en Internet sobre "bases de datos de información biomédica", concluimos que actualmente hay muchas bases de datos disponibles. Pero entre ellas y basándonos en 'ciencias de la salud' las tres bases de datos más importantes a nivel mundial son: pubmed, cinhal y scopus.

Tras revisar estas bases de datos, se concluye que no son las fuentes más adecuadas para intentar automatizar procedimientos de generación de preguntas de test basadas en conocimientos médicos (no disponen de grafos de conocimiento). Para ello parece más razonable dirigir la búsqueda hacia un concepto diferente. el de la 'web de contenidos abiertos para la estructuración del conocimiento médico'. Haciendo esto, llegamos a la conclusión de que este tipo de base de datos termina siendo un soporte adecuado para ayudar a diagnosticar enfermedades y, de paso, puede ser una buena fuente de información para utilidades como aquella a la que se refiere a este trabajo.

Recordando que la investigación que nos ocupa (segunda línea) se centró en determinar fuentes veraces de información, capaces de soportar procesos automáticos de generación de preguntas relacionadas con el conocimiento biomédico, las principales conclusiones serían:

- Las fuentes de información más adecuadas son las catalogadas como "sistemas de toma de decisiones inteligentes o informatizadas".
- Estas bases de datos deben estar basadas en el vocabulario DeCS.

En conclusión, trabajando sobre los criterios antes señalados, se concluye que la base de datos CopenMed [10] tiene atributos suficientes para ser utilizada como fuente

de información para plantear un generador básico de preguntas relacionadas con el conocimiento biomédico que puede ser utilizado como parte de las funcionalidades de la aplicación descrita en este trabajo.

5 SOLUCIÓN PROPUESTA

Sobre la base de los objetivos establecidos en el capítulo 3 de este documento y a partir de los detalles apuntados en el capítulo 4, la solución propuesta y efectivamente desarrollada ha sido la siguiente:

- Desarrollo de una aplicación para dispositivos móviles (smartphones y tablets) sobre sistema operativo Android.
- Uso del lenguaje de programación Kotlin, 100% compatible con Java.
- Uso de Android Studio como plataforma de desarrollo de la aplicación.
- Simplificación máxima del interfaz de usuario de la aplicación para hacer lo más utilizable posible.
- Criterios muy sencillos de estructuración de los posibles usuarios de la aplicación, como se describe más adelante. Básicamente: Administrador, Probador, Usuario registrado y Usuario no registrado.
- Requisitos funcionales y no funcionales muy claros, descritos en detalle en el siguiente capítulo (6 Metodología).
- Capacidad de uso de preguntas “reales” de tests procedentes de diversas fuentes. En concreto y como ejemplo, uso de preguntas efectivamente usadas en diversas ediciones de la oposición MIR.
- Capacidad de generar preguntas automáticas a partir de la base de datos asociada a la página Web Copenmed.
- Capacidad de generar tests de manera aleatoria (solo preguntas generadas “automáticamente” o mezcla de estas y “reales”), a partir de selección de preguntas de una especialidad médica o de una palabra clave concretas (en ambos casos, solo “reales”).
- Capacidad de proporcionar una información detallada del resultado de cada test realizado, en función del tipo de usuario, del tipo de test y de la información relacionada disponible.
- Idioma inglés para los mensajes de las diferentes páginas de la aplicación y posibilidad de preguntas tanto en inglés (generadas “automáticamente”) como en español (preguntas “reales”).

A la largo de los siguientes capítulos se dará todo el detalle necesario sobre la solución finalmente desarrollada y aquí planteada a alto nivel.

6 METODOLOGÍA

En este capítulo, fundamentalmente, se describen los requerimientos que se han definido y se han tenido en cuenta en el desarrollo de la aplicación.

Conviene recordar que los criterios aquí expuestos se han establecido buscando la máxima usabilidad, sencillez y eficacia de la aplicación para sus potenciales usuarios.

Este capítulo se ha estructurado en los siguientes apartados:

- Tipos de usuarios de la aplicación / Roles.
- Requerimientos funcionales
- Requerimientos no funcionales
- Requisitos legales

6.1 Tipos de usuarios de la aplicación / Roles

La aplicación se ha desarrollado en base a los siguientes tipos de usuarios:

- Administrador.
- Probador.
- Usuario registrado.
- Usuario no registrado.

El **Administrador** es quien tiene acceso a todo el proceso de gestión de preguntas “reales” de la aplicación. Sus permisos incluyen:

- La carga manual de preguntas reales de test. En nuestro caso, usando diversas fuentes.
- La edición de preguntas ya cargadas en la aplicación. Incluye su caracterización por especialidad y por palabras clave.
- La eliminación de preguntas.

El **Probador** es un perfil que se ha creado únicamente para poder realizar pruebas sobre la aplicación, a lo largo de su desarrollo, con el fin de no vincularlas a ningún usuario final en concreto. Sus capacidades para utilizar la aplicación son las

mismas que las de un usuario registrado en la aplicación y con inicio de sesión realizado con éxito.

El **Usuario registrado** es aquel que está interesado en utilizar la aplicación como forma de consolidar su conocimiento en temas biomédicos, disponiendo de mecanismos para su autoevaluación. En este sentido, el usuario dispondrá de datos relativos al resultado general de la prueba realizada (respuestas correctas y respuestas erróneas) así como resultados específicos por especialidad.

El **Usuario no registrado** es el que representa a aquellos usuarios que quieren evaluar la utilidad de la aplicación, pero aún no están convencidos de utilizarla como mecanismo de consolidación del conocimiento. Por tanto, la aplicación les permite un uso restringido de la misma, no gestionando mediciones de ranking.

6.2 Requerimientos funcionales

Esta sección describe las principales funciones integradas en la versión de la aplicación disponible al momento de cerrar este documento. La idea es, para cada una de las funciones, definir sus entradas y salidas asociadas, así como la actividad a realizar prevista en cada caso.

Adicionalmente, se indicará el nivel en el que se encuentra dicha función, dentro de la estructura de la aplicación, entendiendo por “Nivel 0” aquel inicial en el que, entre otras, se encuentra la opción de iniciar sesión para un usuario ya registrado. Los sucesivos niveles se numerarán a partir de éste como resultado de la navegación a través de la aplicación.

En las tablas adjuntas, organizadas por funcionalidad, se recoge toda la información previamente mencionada.

También se describirá en esta sección la forma en la que se plantea la generación “automática” de preguntas.

6.2.1 Principales funciones integradas en la aplicación

A continuación, se describen (en modo tabla) las principales funciones integradas en la aplicación desarrollada como objeto de este trabajo.

Tabla 1. Función “Creación de una nueva cuenta”.

Función:	Creación de una nueva cuenta.
Nivel:	0
Entrada:	<ul style="list-style-type: none"> • Alias • Dirección de correo electrónico. • Clave de acceso.
Salida:	Creación de un nuevo usuario en la aplicación, a almacenar en su base de datos.
Actividad:	<p>Requiere completar un formulario que solicita:</p> <ul style="list-style-type: none"> • Un alias del usuario. • Una dirección de correo electrónico. • Una clave de acceso. En esta versión de la aplicación no se han introducido requisitos específicos de construcción de dicha clave. <p>La ausencia de alguno de estos datos impide la creación del nuevo usuario en la aplicación.</p>

Tabla 2. Función “Acceso para usuarios registrados”.

Función:	Acceso para usuarios registrados.
Nivel:	0
Entrada:	<ul style="list-style-type: none"> • Dirección de correo electrónico. • Clave de acceso, previamente cargadas.
Salida:	Acceso a todas las funcionalidades de usuario, lo que supone acceso a todos los tipos posibles de prueba (aleatorio (mezcla de preguntas o solo las generadas “automáticamente”), por especialidad y por palabra clave).
Actividad:	<p>Requiere completar un formulario que solicita:</p> <ul style="list-style-type: none"> • Una dirección de correo electrónico. • Una clave de acceso previamente validada para dicha dirección de correo electrónico

Tabla 3. Función “Acceso para usuarios no registrados”.

Función:	Acceso para usuarios no registrados.
Nivel:	0
Entrada:	No se requiere información alguna para el acceso.
Salida:	El usuario accede a la aplicación con funcionalidades reducidas.
Actividad:	La aplicación permite al usuario acceder en “modo restringido”, lo que supone: <ul style="list-style-type: none">• Realización de pruebas de solo 10 preguntas (en lugar de 25).• Presentación de resultados en modo reducido (solo números de aciertos y fallos en bruto).

Tabla 4. Función “Gestión de preguntas reales de test ya cargadas”.

Función:	Gestión de preguntas reales de test ya cargadas.
Nivel:	1
Entrada:	Registro mediante coordenadas (correo electrónico y clave de acceso) del usuario Administrador.
Salida:	El Administrador accede a la página que permite la gestión de las preguntas ya cargadas, así como la carga de otras nuevas (manualmente).
Actividad:	Las acciones habilitadas en este nivel de la aplicación son las siguientes: <ul style="list-style-type: none">• El filtrado, para su visualización, de preguntas ya cargadas (botón “Filter”).• La edición de preguntas ya cargadas (botón símbolo “lápiz”).• La eliminación de preguntas ya cargadas (botón “papelera”).• La carga manual de preguntas (botón “+”).• El retorno al nivel anterior de la aplicación (botón “Logout”).

Tabla 5. Función “Carga manual de nuevas preguntas”.

Función:	Carga manual de nuevas preguntas.
Nivel:	2
Entrada:	Acceso, desde el Nivel 1, haciendo clic en el botón “+”.
Salida:	<p>La aplicación presenta un formulario estandarizado que contiene, por este orden, los siguientes campos:</p> <ul style="list-style-type: none"> • Enunciado de la pregunta. • Opciones de respuesta (4 posibles). • Indicador de respuesta correcta (espacio a la izquierda de cada respuesta que, al hacer clic, se sombrea en “verde”). Solo una respuesta puede ser indicada por cada pregunta. • Especialidad médica asociada a la pregunta. • Posibles palabras clave (atributos) de la pregunta.
Actividad:	<p>El administrador procede a la carga manual de una pregunta, indicando su respuesta correcta, su especialidad médica, así como sus posibles palabras clave.</p> <p>Todos los campos de una pregunta son de carga obligada a excepción del de “palabras clave”.</p>

Tabla 6. Función “Prueba aleatoria para Usuario registrado_1”.

Función:	Prueba aleatoria para usuario registrado.
Nivel:	2
Entrada:	Acceso, desde el Nivel 1, haciendo clic en el botón “Random”.
Salida:	Se genera una prueba de 25 preguntas de todo tipo (reales y automáticas) y, al finalizar la misma, presenta una página de resultados.
Actividad:	<p>Tras pulsar el botón “Random” se desencadenan las siguientes acciones:</p> <ul style="list-style-type: none"> • Se pone en marcha la prueba de 25 preguntas. • Tras contestar cada pregunta, la aplicación indica la respuesta correcta (sombreada en verde) y, si fuera el caso, la respuesta errónea (sombreada en rojo). • Tras contestar la última pregunta, se accede a la página de resultados.

Tabla 7. Función “Prueba aleatoria para Usuario registrado_2”.

Función:	Prueba aleatoria para usuario registrado.
Nivel:	2
Entrada:	Acceso, desde el Nivel 1, haciendo clic en el botón “G. Random”.
Salida:	Se genera una prueba de 25 preguntas (solo automáticas) y, al finalizar la misma, presenta una página de resultados.
Actividad:	Tras pulsar el botón “G. Random” se desencadenan las siguientes acciones: <ul style="list-style-type: none">• Se pone en marcha la prueba de 25 preguntas.• Tras contestar cada pregunta, la aplicación indica la respuesta correcta (sombreada en verde) y, si fuera el caso, la respuesta errónea (sombreada en rojo).• Tras contestar la última pregunta, se activa accede a la página de resultados.

Tabla 8. Función “Prueba por palabra clave para Usuario registrado”.

Función:	Prueba por palabra clave para usuario registrado.
Nivel:	2
Entrada:	Acceso, desde el Nivel 1, haciendo clic en el botón “Keyword”.
Salida:	Se presenta una pantalla de selección de “palabra clave”. Posteriormente se genera una prueba de 25 preguntas y, al finalizar la misma, presenta una página de resultados.
Actividad:	Tras pulsar el botón “Keyword” se desencadenan las siguientes acciones: <ul style="list-style-type: none">• Se presenta la página de selección de “palabra clave” para su selección.• Se pone en marcha la prueba de 25 preguntas.• Tras contestar cada pregunta, la aplicación indica la respuesta correcta (sombreada en verde) y, si fuera el caso, la respuesta errónea (sombreada en rojo).• Tras contestar la última pregunta, se accede a la página de resultados.

Tabla 9. Función “Prueba por especialidad médica para Usuario registrado”.

Función:	Prueba por especialidad médica para usuario registrado.
Nivel:	2
Entrada:	Acceso, desde el Nivel 1, haciendo clic en la especialidad médica seleccionada de entre las opciones que aparecen en el espacio “Specific Medical Quiz”.
Salida:	Se genera una prueba de 25 preguntas y, al finalizar la misma, presenta una página de resultados.
Actividad:	Tras pulsar el botón de la especialidad elegida se desencadenan las siguientes acciones: <ul style="list-style-type: none">• Se pone en marcha la prueba de 25 preguntas.• Tras contestar cada pregunta, la aplicación indica la respuesta correcta (sombreada en verde) y, si fuera el caso, la respuesta errónea (sombreada en rojo).• Tras contestar la última pregunta, se accede a la página de resultados, en este caso restringidos a la especialidad elegida.

Tabla 10. Función “Prueba aleatoria para Usuario no registrado_1”.

Función:	Prueba aleatoria para usuario no registrado.
Nivel:	2
Entrada:	Acceso, desde el Nivel 1, haciendo clic en el botón “Random”.
Salida:	Se genera una prueba de 10 preguntas de todo tipo (reales y automáticas) y, al finalizar la misma, presenta una página reducida de resultados.
Actividad:	Tras pulsar el botón “Random Quiz” se desencadenan las siguientes acciones: <ul style="list-style-type: none">• Se pone en marcha la prueba de 10 preguntas.• Tras contestar cada pregunta, la aplicación indica la respuesta correcta (sombreada en verde) y, si fuera el caso, la respuesta errónea (sombreada en rojo).• Tras contestar la última pregunta, se accede a la página reducida de resultados.

Tabla 11. Función “Prueba aleatoria para Usuario no registrado_2”.

Función:	Prueba aleatoria para usuario no registrado.
Nivel:	2
Entrada:	Acceso, desde el Nivel 1, haciendo clic en el botón “G. Random”.
Salida:	Se genera una prueba de 10 preguntas (solo automáticas) y, al finalizar la misma, presenta una página reducida de resultados.
Actividad:	Tras pulsar el botón “Random Quiz” se desencadenan las siguientes acciones: <ul style="list-style-type: none">• Se pone en marcha la prueba de 10 preguntas.• Tras contestar cada pregunta, la aplicación indica la respuesta correcta (sombreada en verde) y, si fuera el caso, la respuesta errónea (sombreada en rojo).• Tras contestar la última pregunta, se accede a la página reducida de resultados.

Tabla 12. Función “Prueba por palabra clave para Usuario no registrado”.

Función:	Prueba por palabra clave para usuario no registrado.
Nivel:	2
Entrada:	Acceso, desde el Nivel 1, haciendo clic en el botón “Keyword Quiz”.
Salida:	Se genera una prueba de 10 preguntas y, al finalizar la misma, presenta una página reducida de resultados.
Actividad:	Tras pulsar el botón “Keyword Quiz” se desencadenan las siguientes acciones: <ul style="list-style-type: none">• Se presenta la página de selección de “palabra clave” para su selección.• Se pone en marcha la prueba de 10 preguntas.• Tras contestar cada pregunta, la aplicación indica la respuesta correcta (sombreada en verde) y, si fuera el caso, la respuesta errónea (sombreada en rojo).• Tras contestar la última pregunta, se accede a la página reducida de resultados.

Tabla 13. Función “Prueba por especialidad médica para Usuario no registrado”.

Función:	Prueba por especialidad médica para usuario no registrado.
Nivel:	2
Entrada:	Acceso, desde el Nivel 1, haciendo clic en la especialidad médica seleccionada de entre las opciones que aparecen en el espacio “Specific Medical Quiz”.
Salida:	Se genera una prueba de 10 preguntas y, al finalizar la misma, presenta una página de resultados.
Actividad:	Tras pulsar el botón de la especialidad elegida se desencadenan las siguientes acciones: <ul style="list-style-type: none">• Se pone en marcha la prueba de 10 preguntas.• Tras contestar cada pregunta, la aplicación indica la respuesta correcta (sombreada en verde) y, si fuera el caso, la respuesta errónea (sombreada en rojo).• Tras contestar la última pregunta, se accede a la página reducida de resultados.

Tabla 14. Función “Logout”.

Función:	Logout.
Nivel:	1
Entrada:	Acceso, desde el Nivel 1, haciendo clic en el botón “Logout”.
Salida:	Acceso al Nivel 0.
Actividad:	Esta función asegura el retorno al Nivel 0 desde diferentes páginas del Nivel 1.

6.2.2 Estrategia de generación de preguntas de test de la aplicación

La aplicación desarrollada se basa en la gestión de preguntas test que:

- Estén correctamente formuladas.
- Se adapten adecuadamente a los conocimientos médicos.
- Sean compatibles con los criterios establecidos para la consolidación del conocimiento.

En línea con lo anterior y como ya se mencionó en su momento en este documento, la aplicación se ha basado en 2 fuentes de información principales:

1. Publicaciones existentes en el mercado, cuyo objeto sea la recopilación de preguntas de examen reales, tales como: “MIR 1.175 preguntas de examen de opción múltiple, Oposiciones 2014 a 2018”. [11]
2. Bases de datos de acceso público existentes como: Copenmed (<https://copenmed.org/#/paginaInicio>). [10]

Para el primer caso, la aplicación permite la transcripción de textos para reproducir las preguntas reales del examen. En el caso de contar con esta información ya digitalizada, una posible evolución de la aplicación sería permitir el volcado de las preguntas en modo “batch”, siguiendo criterios de carga predefinidos y, en su caso, el formato para adaptar la estructura de las preguntas y sus “respuestas verdaderas” a los mecanismos de trabajo de la aplicación.

Para el segundo caso, en el presente trabajo se ha utilizado un volcado, en un fichero Excel, del contenido de la citada base de datos (Copenmed), proponiendo una estrategia para la obtención de posibles preguntas de examen según los siguientes criterios:

- Se ha trabajado con la estructura establecida en la base de datos, que fundamentalmente reconoce los siguientes elementos:
 - Entidades: Por ejemplo, enfermedades, síntomas, etc.
 - Asociaciones: Posibles relaciones entre entidades.
- Para evitar complicaciones derivadas del correcto tratamiento de la lengua española, se ha propuesto utilizar (literalmente) las llamadas “entidades” y “asociaciones” ya preestablecidas en la base de datos. Por ejemplo: ‘Enfermedad1 (Entidad1) can be treated with con Tratamiento (Entidad2)’. No se ha abordado en este caso, una conversión de la lengua de los textos, por ejemplo, al español. De nuevo, esto supondría una evolución futura de la aplicación.
- Se ha trabajado con los “criterios de veracidad” que utiliza la propia base de datos, basados en la asignación de valores entre 0 (veracidad nula) y

1 (veracidad máxima) de las opciones posibles de construcción de preguntas y respuestas.

- Para para una determinada '1ª Entidad' y para una determinada asociación, se ha de generar la búsqueda de posibles y diferentes '2ª Entidad' según los siguientes criterios:
 - Al menos una posibilidad con 'criterio de veracidad' no nulo (posible “Respuesta verdadera”).
 - Al menos 3 posibilidades con “criterios de veracidad” nulo (potenciales “Respuestas falsas”).

A continuación, se reproduce la secuencia lógica en la que se basa el desarrollo software de generación de preguntas de test “automáticas” (incluyendo ejemplos reales del contenido de la base de datos).

6.2.3 Flujo lógico para la generación de preguntas “automáticas” a partir de la base de datos Copenmed

(I) Se actualiza la base de datos de Copenmed.

Significa que una nueva versión de la base de datos (un nuevo archivo de Excel) está disponible para generar un nuevo conjunto de preguntas.

Este archivo de Excel es la entrada para la herramienta (que se definirá en este capítulo) para generar automáticamente preguntas relacionadas con conceptos biomédicos.

Analizando la estructura del archivo Excel, se están tomando en consideración diferentes carpetas para el manejo de la información.



tipos_entidad	tipos_relaciones	idiomas	relaciones	entidades	detalles	recursos	descripciones
---------------	------------------	---------	------------	-----------	----------	----------	---------------

Figura 1 Estructura de hojas del fichero Excel de descarga de información de Copenmed.

(II) Se selecciona un tipo específico de entidad desde la hoja ‘tipos_entidad’.

Por ejemplo, “Disease”; IdTipoEntidad=6.

IdTipoEntidad	TipoEntidad	IdIdioma	Idioma
6	Disease	0	null

Figura 2 Ejemplo de tipo de entidad en la base de datos Excel (hoja “tipos_entidad”)

(III) Se selecciona una entidad específica desde la hoja ‘entidades’.

Por ejemplo, “Tendinitis calcificada”; IdEntidad=6171;

“Disease”, IdTipoEntidad=6.

IdEntidad	Entidad	IdTipoEntidad	TipoEntidad	Idioma	Comentario
6171	Tendinitis calcificada	6	Disease	Español	null

Figura 3 Ejemplo de entidad en la base de datos Excel (hoja “entidades”)

(IV) Se selecciona una asociación específica.

En la hoja “tipos_relaciones” hay diferentes asociaciones predefinidas entre entidades siguiendo la siguiente estructura: Entidad1 + “asociación predefinida” + Entidad2.

Por ejemplo, “Disease1 (Entidad1) is diagnosed with this Test (Entidad2)”;

IdTipoAsociacion = 30

IdTipoEntidad1 = 6; “Disease”

IdTipoEntidad2 = 11; “Test”

IdTipoAsociacion	IdTipoEntidad1	IdTipoEntidad2	IdIdioma	TipoAsociacion	Idioma	TipoEntidad1	TipoEntidad2
30	6	11	0	Disease is diagnosed with this Test	null	Disease	Test

Figura 4 Ejemplo de asociación en la base de datos Excel (hoja “tipos_relaciones”)

(V) Se construye una lista de posibles opciones de “Entidad2” y sus “niveles de veracidad” asociados (de 0,0 a 1,0).

En la hoja de “relaciones” y a partir de una “Entidad1” y un “TipoAsociacion” específicos, se identifican todas las posibles “Entidad2” asociadas con sus “niveles de veracidad” no nulos.

(VI) Se seleccionan candidatos adecuados para la construcción de la pregunta.

Con el fin de definir la pregunta posible, las condiciones a aplicar sobre la lista resultante de trabajar con una “Entidad1”, un “TipoAsociacion” y todas las posibles “Entidad2”, son las siguientes:

1. Al menos debe existir 1 posible combinación que disponga de un “nivel de veracidad” no nulo (valor de la variable “Nivel”). Esto determinará la “respuesta correcta” para la pregunta a construir.
2. Se buscarán alternativas de “Entidad2” que no tengan “nivel de veracidad” no nulo (valor de la variable “Nivel”). Esto determinará las “respuestas erróneas” para la pregunta a construir.
3. Si para una pareja “Entidad1” + “TipoAsociacion” se cumplen las condiciones anteriores, entonces es posible construir una pregunta.

En la figura siguiente se puede ver un ejemplo real de posible pregunta a partir del cumplimiento de las anteriores condiciones:

IdTipoAsociacion = 30; “Disease1 is diagnosed with this Test”

IdTipoEntidad1 = 6; “Disease”

IdTipoEntidad2 = 11; “Test”

Y para el mencionado ejemplo:

- Hay 2 opciones “verdaderas”, con valor de Nivel > 0:
“Ecografía”, IdEntidad2=602;
“Radiología convencional”, IdEntidad2=3295.
La selección de la respuesta correcta es aleatoria.
- Hay, múltiples opciones de Entidad2, con valor de Nivel = 0 con las que se construyen 3 respuestas erróneas. Por ejemplo:
“Test serológico H. Pylori”, IdEntidad2=7106;
“Prueba de precipitinas para aspergilosis”, IdEntidad2=6536;
“Test de la presión intraocular”, IdEntidad2=6305;

IdAsociacion	IdEntidad1	IdEntidad2	IdTipoAsociacion	TipoAsociacion	Descripcion	Nivel
25390	6171	602	30	Disease is diagnosed with this Test	null	0.6
25391	6171	3295	30	Disease is diagnosed with this Test	null	0.6

Figura 5 Ejemplo de posible combinación de Entidad2 para una Entidad1 y TipoAsociacion específicas en la base de datos Excel (hoja “relaciones”)

(VII) Se construye la pregunta para su incorporación a un test.

La herramienta de construcción de preguntas asegura la generación de un nuevo bloque de ellas que habilitan un nuevo test.

6.3 Requerimientos no funcionales

En esta sección se describen los requerimientos que se espera que cumplan la aplicación y sus sistemas de soporte para que se utilicen correctamente.

Facilidad de uso

La aplicación ha de contar con un interfaz de usuario que cumpla con las siguientes condiciones:

- Que sea de uso cómodo.
- Que sea de uso intuitivo.
- Que sea muy accesible, es decir, que permita, en un número limitado de “clics”, acceder a la totalidad de las funcionalidades programadas.

Escalabilidad

El sistema ha de permitir introducir nuevas funcionalidades de manera sencilla y ha de ser capaz de manejar bases de datos adecuadas al crecimiento de la aplicación en cuanto:

- Número de preguntas “reales” a cargar manualmente o, en el futuro, automáticamente.
- Bases de datos de datos, potencialmente cada vez mayores, sobre las que trabajar en la generación de preguntas “automáticas”.

Accesibilidad

Se han de seguir los patrones básicos de accesibilidad de Android. En este sentido, la aplicación se ha de diseñar para ser utilizada directamente desde la pantalla del dispositivo, mediante botones “táctiles”. No se desarrollarán funcionalidades por voz.

Robustez

La aplicación ha de configurarse para gestionar eficazmente los datos de usuario. En este sentido:

- No se permitirá la gestión de usuarios duplicados.
- Solo se trabajará con los siguientes datos personales:
 - Dirección de correo electrónico.
 - Contraseña de usuario
- Se garantizará el nivel de confidencialidad necesario para cumplir con las directivas vigentes de protección de datos.

Conectividad

Básicamente debido al tipo de dispositivos para los que se diseñará la aplicación, ésta dispondrá de acceso a Internet para gestionar comunicaciones de datos. Dicho acceso se garantizará, básicamente, vía WiFi y redes móviles de comunicaciones. No se requerirá conectividad adicional, por ejemplo, de tipo GPS.

Seguridad

La aplicación ha de respetar los criterios básicos de seguridad exigibles a aplicaciones de este tipo en el entorno Android. El manejo confidencial de los datos personales ha de estar garantizado. No se considerarán datos confidenciales aquellos asociados con la construcción y presentación de las preguntas de test.

Persistencia

Se dará por hecho que la persistencia estará garantizada al trabajar sobre el entorno Android y Google a través de sus utilidades. Dada la utilidad de la aplicación

desarrollada no parece necesario dotarla de un elevado nivel de persistencia en aspectos tales como: ausencia de conectividad (no previsible) o caída del servicio.

Integración

El uso de la aplicación ha de estar garantizada para versiones del sistema operativo Android posteriores a la versión 7.

Mantenimiento

La aplicación requerirá de un mínimo mantenimiento que se circunscribirá, fundamentalmente, a la actualización de su banco de preguntas reales y a la disponibilidad de ficheros Excel actualizados con información de la base de datos de conocimientos biomédicos.

6.4 Requerimientos legales

A continuación, se enumeran los principales requerimientos legales exigibles a una aplicación del tipo de la desarrollada para este trabajo y su estado a la hora de concluir el presente trabajo:

- Permisos, licencia y condiciones de uso: Por el momento, nada de esto se ha incluido en el desarrollo de la aplicación al no haberse planteado, como objetivo de diseño, su disponibilidad para ser comercializada.
- Derechos propios y de terceros: En el caso de las preguntas “reales”, supuestamente de dominio público y disponibles en publicaciones específicas y en Internet, sería necesario corroborar la posibilidad de su uso en la plataforma a través de la carga manual de las mismas. Otro caso diferente sería la opción de descarga de bloques concretos de preguntas, en modo batch ya filtradas por alguna “tercera parte” y los derechos de dicha “tercera parte” sobre el bloque generado. En el caso de la generación “automática” de preguntas a partir de la Web Copenmed, la información contenida en dicha Web se cataloga como “abierta” y hace pensar que su uso no estaría sujeto a algún tipo de licencia de explotación por aplicaciones como la desarrollada aquí. En cualquier caso, sería

necesario un estudio legal al respecto si se buscara la explotación comercial de la aplicación.

- Protección a menores: La aplicación desarrollada no está orientada a su uso por menores ni contiene contenidos que puedan ser sensibles para ese tipo de población. Es por esto por lo que no se entiende necesario el cumplimiento de condiciones específicas reguladas en materia de derechos de consumidores y usuarios para el colectivo considerado.
- Funcionalidades ilícitas: La aplicación carece de funcionalidades ilícitas. Otra cosa podría ser considerar como ilícitas aquellas funcionalidades derivadas del uso de información susceptible de estar sujeta a concesión de licencia de explotación y cuyo análisis ya ha sido planteado anteriormente.
- Privacidad y geolocalización: El único dato personal que la aplicación recoge y almacena es el de la dirección de correo electrónico del usuario registrado. En esta misma línea, la aplicación no solicita ni el nombre ni los apellidos del usuario sino, tan solo, un alias de éste. Por otro lado, la aplicación no requiere gestionar la localización del usuario. Si ésta está siendo gestionada por el terminal que el usuario utilice para la implementación de la aplicación, se supone que dicho usuario ya ha dado su consentimiento en un ámbito de uso generalista del dispositivo y al margen del de la aplicación.
- Información y cookies: No se ha incluido este tipo de información en la aplicación, entendiendo que la misma sería objeto de atención en el caso de activar la comercialización de la aplicación.
- Markets: Por el momento, la aplicación cumple los requisitos mínimos que habilitarían la gestión de su disponibilidad a través de los “markets” al uso.
- Publicidad: No ha sido objeto de este trabajo el analizar la posible monetización de la publicidad de esta aplicación.

7 DESARROLLO

A continuación, se describen para la aplicación desarrollada:

- Su arquitectura hardware.
- Su arquitectura software.
- Su “look&feel” a través de sus pantallas (mockups) desarrolladas.

7.1 Arquitectura Hardware

Para el desarrollo y depuración de la aplicación se ha utilizado el siguiente hardware:

- **Ordenador portátil:**
 - Modelo: 14s – dq1021ns
 - Procesador: Intel® Core™ i7 – 1065 G7 CPU 1.30 GHz
 - Memoria RAM: 8 GB
 - Sistema operativo: Windows 11 Home
- **Tablet:**
 - Modelo: Galaxy Tab S6 Lite
 - Procesador: Eynos 9611 2.3 GHz
 - Memoria RAM: 4 GB
 - Sistema operativo: Android 12
- **Smartphone:**
 - Modelo: Xiaomi Redmi 9
 - Procesador: Octa-core Max 2.00 Ghz
 - Memoria RAM: 3 GB
 - Sistema operativo: Android 11

El ordenador portátil ha sido utilizado para el desarrollo del software de la aplicación.

La tablet y el smartphone han sido utilizados como bancos de pruebas para la depuración y testeo de las diferentes versiones desarrolladas para la aplicación.

7.2 Arquitectura Software

A continuación, se describen las principales características de la arquitectura software de la aplicación desarrollada, así como detalles de los principales bloques de programación desarrollados.

7.2.1 Características principales de la arquitectura software

Tabla 15. Características principales de la arquitectura software de la aplicación.

Lenguaje de programación:	Kotlin 1.6.20
Kit de desarrollo software:	Se ha usado un 'software development kit' (SDK) para el desarrollo de la aplicación sobre sistema operativo Android, con las siguientes características: <ul style="list-style-type: none"> • Minsdk: 24 (Android 7) • Targetsdk: 32 (Android 12) • Compilesdk: 32 (Ver Figura 16)
Herramientas de compilación:	Se ha usado el paquete de herramientas de compilación Android Gradle Plugin 7.2. [12]
Interfaz de usuario:	Se ha usado la arquitectura software denominada: Model-View-Presenter (MVP).
Entorno de desarrollo:	Para el desarrollo de la aplicación se ha usado el entorno de desarrollo Firebase, versión 30.0.0. (Ver Figura 17)
Bases de datos:	La gestión de la base de datos se ha implementado con Firebase Firestore. [13] Se trata de una base de datos No SQL. Se han desarrollado dos tipos de tablas: <ul style="list-style-type: none"> • <u>Tabla de Usuarios</u>. Contiene la información de usuarios y sus campos de contenido son los siguientes: <ul style="list-style-type: none"> ○ Identificador de usuario / 'userid' ○ Nombre de usuario / 'username' ○ Dirección de correo electrónico / 'e-mail' ○ Administrador / 'admin' • <u>Tabla de Preguntas</u>. Contiene la información relativa a cada pregunta y sus campos de contenido son los siguientes: <ul style="list-style-type: none"> ○ Texto de la pregunta / 'question text' ○ Opciones de respuesta / 'options' ○ Palabras clave asociadas a la pregunta / 'keywords' ○ Especialidad médica / 'medical speciality'
Autenticación de usuarios:	La autenticación de usuarios se ha implementado con Firebase Authentication. [14]

A continuación se muestran diferentes imágenes representando la configuración del script 'built.gradle' (Figuras 6 y 7).

```
android {  
    compileSdk 32  
  
    defaultConfig {  
        applicationId "com.openmed"  
        minSdk 24  
        targetSdk 32  
        versionCode 5  
        versionName "1.0"  
    }  
}
```

Figura 6 Detalles de configuración del SDK

```
// Import the BoM for the Firebase platform  
// When using the BoM, you don't specify versions in Firebase library dependencies  
implementation platform('com.google.firebase:firebase-bom:30.0.0')  
//firebase authentication (for registering the users using email and password)  
implementation 'com.google.firebase:firebase-auth-ktx'  
//firebase firestore (for storing data)  
implementation 'com.google.firebase:firebase-firestore-ktx'  
// Declare the dependencies for the Crashlytics and Analytics libraries  
implementation 'com.google.firebase:firebase-crashlytics-ktx'  
implementation 'com.google.firebase:firebase-analytics-ktx'
```

Figura 7 Detalles de configuración del entorno de desarrollo Firebase

La aplicación se ha apoyado en el framework 'Navigation' para configurar la navegación interna y externa entre elementos de la aplicación. En la Figura 8 pueden verse sus detalles tal y como se ha configurado en el script 'built.gradle'.

```
//navigation components library  
def nav_version :String = "2.4.2"  
implementation "androidx.navigation:navigation-fragment-ktx:$nav_version"  
implementation "androidx.navigation:navigation-ui-ktx:$nav_version"
```

Figura 8 Detalles de configuración del framework de navegación entre elementos de la aplicación

7.2.2 Diagrama de navegación a través de la aplicación

A continuación, se representa (ver Figura 9) el diagrama de navegación a través de las diferentes “fragments” desarrollados para la aplicación y cullos “mockups” han sido descritos en el apartado 7.2.5 de este documento:

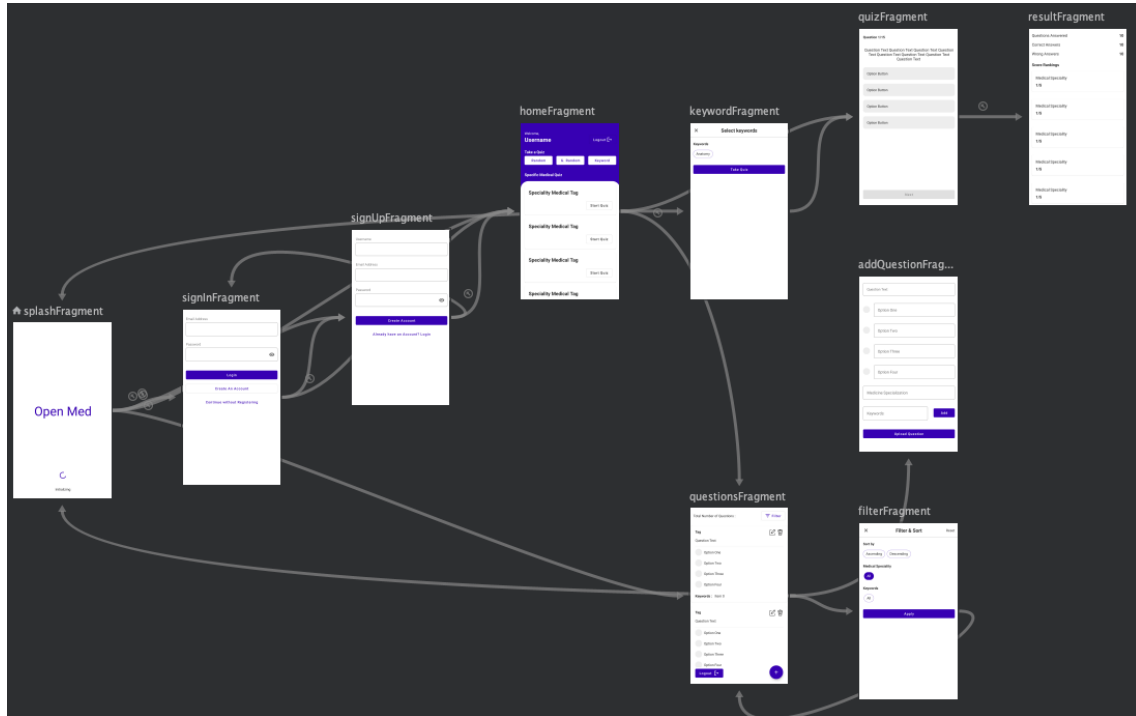
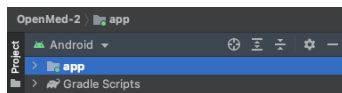


Figura 9 Diagrama de navegación entre los diferentes “fragments” desarrollados para la aplicación

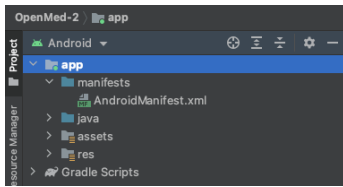
7.2.3 Estructura del software desarrollado para la aplicación

A continuación, se describen los bloques que se han desarrollado para la aplicación objeto de este trabajo, visualizados mediante el entorno de desarrollo ‘Android Studio’:



Aquí se ven los dos bloques básicos del programa: La aplicación en si misma y el bloque de scripts ‘Gradle’.

Figura 10 Nivel 1 / Bloques del programa desarrollado



En esta imagen pueden verse los bloques en los que se ha estructurado la aplicación, haciendo foco especial en el fichero ‘AndroidManifest’.

Figura 11 Nivel 2 / Bloques del programa desarrollado / Detalle de “app / manifest”

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.openmed">

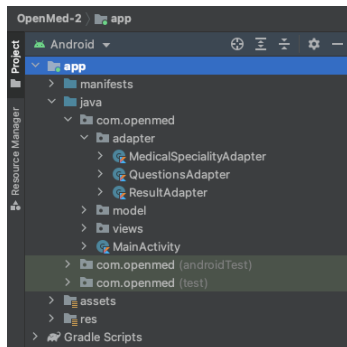
    <uses-permission android:name="android.permission.INTERNET"/>

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/Theme.OpenMed">
        <activity
            android:name=".MainActivity"
            android:exported="true"
            android:configChanges="orientation|screenSize|screenLayout|navigation|smallestScreenSize"
            android:windowSoftInputMode="adjustInspecified|adjustPan">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

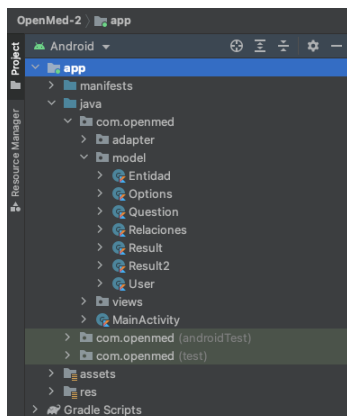
En esta imagen puede verse el contenido del archivo ‘AndroidManifest’. Dicho archivo permite definir los principales componentes de la aplicación.

Figura 12 Nivel 3 / Bloques del programa desarrollado / Contenido del fichero “AndroidManifest”



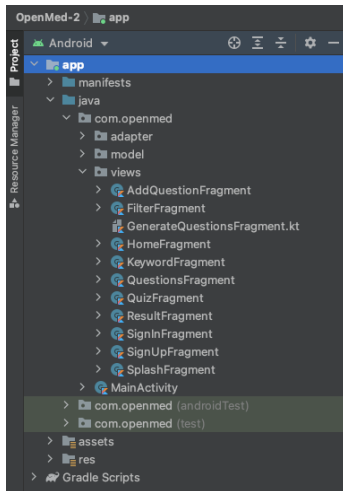
En esta imagen puede verse cómo han sido estructurados los adaptadores para la gestión de las especialidades médicas, las preguntas y los resultados correctos o incorrectos de la aplicación.

Figura 13 Nivel 5 / Bloques del programa desarrollado / Detalle de los “Adapters” desarrollados



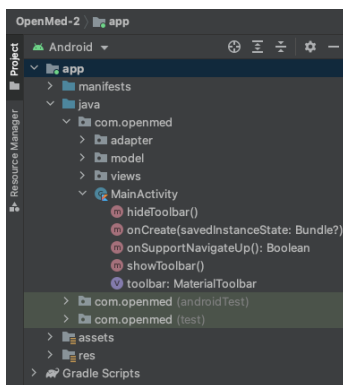
En esta imagen puede verse cómo han sido estructurados los modelos de la base de datos de la aplicación.

Figura 14 Nivel 5 / Bloques del programa desarrollado / Detalle de los “Models” desarrollados



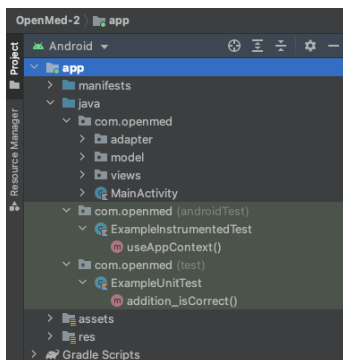
En esta imagen puede verse cómo han sido estructurados los objetos ‘view’ desarrollados para la aplicación.

Figura 15 Nivel 5 / Bloques del programa desarrollado / Detalle de los “Views” desarrollados



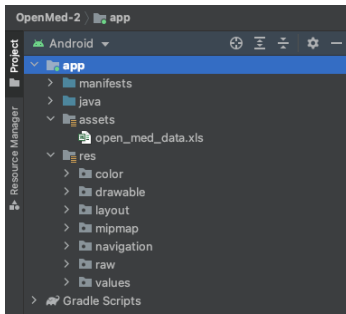
En esta imagen puede verse cómo han sido estructurados los principales componentes de la navegación de la aplicación, a través de “MainActivity”.

Figura 16 Nivel 5 / Bloques del programa desarrollado / Detalle del objeto “MainActivity”



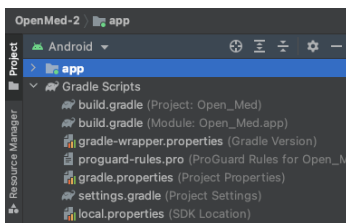
En esta imagen se visualiza la forma en la que se han implementado los tests de la aplicación. [15]

Figura 17 Nivel 4 / Bloques del programa desarrollado / Detalle de los “Test” de la aplicación



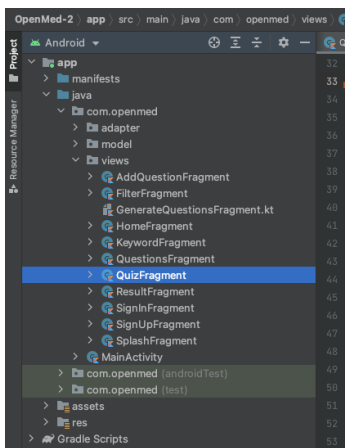
En esta imagen se visualizan los directorios “Assets” y “Res”. El primero contiene el fichero de datos para la generación automática de preguntas. Y el segundo contiene los ficheros y descriptores de imágenes, en formato ‘.xml’.

Figura 18 Nivel 3 / Bloques del programa desarrollado / Detalle de los directorios “Assets” y “Res”



En esta imagen se visualizan los diferentes scripts “gradle” definidos para la aplicación, identificando diferentes detalles de configuración de la aplicación.

Figura 19 Nivel 1 / Bloques del programa desarrollado / Detalle del directorio “Gradle Scripts”



Finalmente, y por su relevancia para la aplicación, en la figura adjunta se ve la ubicación del fragmento “QuizFragment” donde se establece el proceso de gestión de preguntas de test. Se incluye el proceso de generación de preguntas automáticas, a partir del volcado realizado sobre un fichero Excel (en nuestro caso denominado “open_med_data.xls”) a su vez a partir de la información contenida en la web “Copenmed”.

Figura 20 Nivel 5 / Bloques del programa desarrollado / Detalle del directorio “QuizFragment”

7.2.4 Detalle de la generación de preguntas de test

A continuación, y en relación con el fragmento “QuizFragment” antes mencionado, se recorre el código generado para dicho fragmento y se presentan capturas de pantalla con los elementos principales, propios del proceso de generación de preguntas por parte de la aplicación.

En la figura adjunta a continuación, se representa el código que permite inicializar el “assetManager”, habilitar la apertura del fichero Excel con información procedente de la base de datos Copenmed, inicializar el “POIFSFileSystem” y abrir el “Workbook”.

```

65 //for generating random questions from the excel sheet (read this for how to do that)
66 //https://www.blueappsoftware.com/how-to-read-excel-file-in-android-tutorial/
67 //https://www.baeldung.com/java-microsoft-excel
68 private lateinit var myInput: InputStream
69 // initialize asset manager
70 private lateinit var assetManager : AssetManager
71 // Create a POI File System object
72 private lateinit var myFileSystem : POIFSFileSystem
73 // Create a workbook using the File System
74 private lateinit var myWorkBook : HSSFWorkbook
75 private var dataList = mutableListOf<Relaciones>()
76 //private var dataList2 = mutableListOf<Relaciones>()
77 private var entidadList = mutableListOf<Entidad>()
78

```

Figura 21 Fragmento “quizFragment” / Inicialización de “assetManager”

En las siguientes 2 figuras se aprecia la forma en que se activan los tests de 25 o de 10 preguntas, según el tipo de usuario considerado y la forma de obtener las preguntas a presentar en el test (los 3 posibles modelos).

```

106 //if the user is a registered user he would answer 25 questions while an unregistered user would answer 10 questions
107 questionsToAnswer = if(isRegistered){25}else{10}
108
109 //getting the questions from firebase database
110 getQuestionsFromDatabase()

```

Figura 22 Fragmento “quizFragment” / Determinación del número de preguntas por test

```

114 private fun getQuestionsFromDatabase() {
115 //Getting the questions from firebase firestore
116 firebaseFirestore.collection( collectionPath: "Questions").get().addOnSuccessListener { it: QuerySnapshot?
117 //getting the list from firebase and converting it to a mutable list
118 questionsList = it.map { it.toObject(Question::class.java) } as MutableList<Question>
119 if (questionsList.size >= questionsToAnswer) {
120 //this means there are enough questions to display to the user
121 if(quizType != "random" && isSpecial){
122 //give the user quiz based on the selected medical specialty
123 val specificList = mutableListOf<Question>()
124 //this would check for the tag related questions to display to the user
125 questionsList.forEach { question ->
126 if(question.tag.lowercase().trim() == quizType.lowercase().trim()){
127 specificList.add(question)
128 }
129 }
130 questionsList2 = specificList.shuffle() as MutableList<Question> // this would shuffle the questions
131 displayQuestion()

```

Figura 23 Fragmento “quizFragment” / Obtención de preguntas desde la base de datos – 1/2

```

132     }else if(quiztype == "keyword" && !keywords.isNullOrEmpty()){
133         //conditions for a keyword quiz has been satisfied
134         val keywordListQuestion = mutableListOf<Question>()
135         keywords?.forEach { keyword ->
136             keywordListQuestion.addAll(
137                 //this would add all questions that contains the keyword
138                 questionsList.filter { it: Question
139                     it.keywords.contains(keyword)
140                 }
141             )
142         }
143         questionsList2 = keywordListQuestion.shuffled() as MutableList<Question>
144         displayQuestion()
145         Log.i( tag: "QuizFragment", msg: "$questionsList2")
146     }else{
147         //this means the user selected random quiz or random generated quiz button6
148         lifecycleScope.launch(IO) { this: CoroutineScope
149             //because the data from the excel sheet is large we run the task in the background thread
150             //whenStarted {
151                 assetManager = requireContext().assets
152                 myInput = assetManager.open( fileName: "open_med_data.xls")//open_med_data.xls is the name of the ex
153                 myFileSystem = POIFSFileSystem(myInput)
154                 myWorkbook = HSSFWorkbook(myFileSystem)
155
156                 getData()
157
158                 getEntidadData()
159             //}

```

Figura 24 Fragmento “quizFragment” / Obtención de preguntas desde la base de datos – 2/2

En las siguientes 2 figuras, se representa el inicio de la sección de código destinada a la generación automática de preguntas a partir de la información procedente de la base de datos Copenmed.

```

276     }
277 }
278
279 private fun generateRandomQuestions() {
280     val questionText = mutableListOf<String>()
281     val pharseText = mutableListOf<String>()
282     val optionText = mutableListOf<Options>()
283     val questions = mutableListOf<Question>()
284     val tags = mutableListOf<String>()
285
286
287     val entidad_one_id_list = mutableListOf<String>()
288     val entidad_two_id_list = mutableListOf<String>()
289
290     dataList.forEach { it: Relaciones
291         questionText.add(it.tipo_association)

```

Figura 25 Fragmento “quizFragment” / Generación automática de preguntas / Inicio – 1/2

```

287     val entidad_one_id_list = mutableListOf<String>()
288     val entidad_two_id_list = mutableListOf<String>()
289
290     dataList.forEach { it: Relaciones
291         questionText.add(it.tipo_association)
292
293         //adding dummy data into this lists
294         pharseText.add("null")
295         optionText.add(Options( option_one: "null", option_two: "null", option_three: "null", option_four: "null", answer: "0"))
296         tags.add("null")
297         entidad_one_id_list.add("null")
298         entidad_two_id_list.add("null")
299     }

```

Figura 26 Fragmento “quizFragment” / Generación automática de preguntas / Inicio – 2/2

En la figura siguiente se representan detalles de la generación de respuestas correctas y erróneas como parte de la generación automática de preguntas de test.

```

361 //we have about 40000 question so we create a list of the question numbers
362 val list = mutableListOf<Int>()
363 val _list = mutableListOf<Int>()
364 for (i in 1..dataList.size){
365     list.add(i)
366 }
367
368 for (i in 1..25){
369     val random = list.random() // this would generate a random number from the number list
370     val list2 = questionText[random-1].split( ...delimiters: " ") as MutableList<String> // this would split the question
371     list2.removeLast() //this would remove the last word in the question text
372     val entidad_one_id = dataList[random-1].entidad_one_id //the id of the first word to use in the question text
373     val entidad_two_id = dataList[random-1].entidad_two_id //the id of the answer to the question
374     val nivel = dataList[random-1].nivel.toDouble() //the nivel of the question
375     entidadList.forEach { entidad ->
376         //we loop through the entidad list to get our first word and answer
377         if(entidad.entidad_id == entidad_one_id){
378             //if the entidad id is the same with our id then this is the correct word
379             list2[0] = entidad.entidad //this replaces the first dummy word in the question text with the right word
380         }else if(entidad.entidad_id == entidad_two_id){
381             //if the entidad id is the same with our answer id then this is the correct option
382             val falseoptions = getFalseOptions(entidad.entidad_tipo_id) // this would generate random false options
383             randomOptions(random, entidad, optionText,falseoptions) // this would rearrange the options randomly
384             tags[random -1] = entidad.entidad_tipo //adding this to the medical specialty tags
385             entidad_one_id_list[random -1] = entidad_one_id //adding this to the entidad_one_id list
386             entidad_two_id_list[random -1] = entidad_two_id //adding this to the entidad_two_id list
387         }
388     }
389     val newword = list2.joinToString( separator: " ") // this would join the words together
390     phraseText[random -1] = newword //adding the word to our new pharsetext list
391     list.remove(random) //we remove the number from the list to avoid duplicates
392     _list.add(random) // we add the question number to this list to know the location of the questions in the list
393 }

```

Figura 27 Fragmento “quizFragment” / Generación de respuestas verdaderas y erróneas

Finalmente, en la figura adjunta se detalla el proceso de construcción del bloque de 25 preguntas de un test.

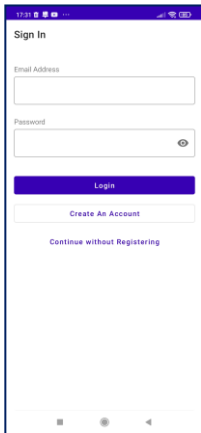
```

395 //we loop through the question 25 times and create random questions with answers
396 for (i in 0..24){
397     val question = Question( id = "$i", phraseText[_list[i]-1], image = "", optionOne = optionText[_list[i]-1].option_one,
398     optionTwo = optionText[_list[i]-1].option_two, optionThree = optionText[_list[i]-1].option_three, optionFour = optionText[_list[i]-1].option_four,
399     correctAnswer = optionText[_list[i]-1].answer.toInt(), timestamp = System.currentTimeMillis(), tag = tagsList[i]-1.)
400     questions.add(question)
401     Log.i( tag = "QuizFragment", msg = "$i Question $question")
402     Log.i( tag = "QuizFragment", msg = "$i Entidad 1 ${entidad_one_id_list[_list[i]-1]}, Entidad 2 ${entidad_two_id_list[_list[i]-1]}")
403 }

```

Figura 28 Fragmento “quizFragment” / Generación de un bloque de 25 preguntas

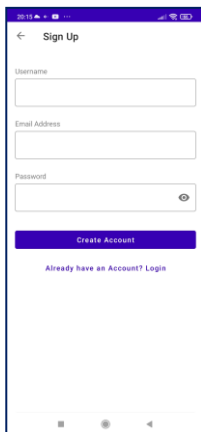
7.2.5 Pantallas (mockups) de la aplicación



Esta imagen corresponde a la página de inicio de la aplicación. En ella, el usuario puede elegir:

- Acceder con un login y password.
- Crear una nueva cuenta.
- Acceder a la aplicación de forma anónima.

Figura 29 Página Sig In

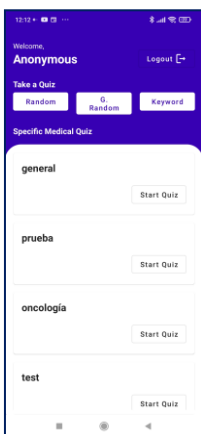


Esta imagen corresponde a la página de la aplicación que permite la creación de una nueva cuenta. En ella, el usuario ha de introducir:

- Un alias.
- Una dirección de correo electrónico.
- Una clave de acceso.

Adicionalmente, el usuario puede salir de la pantalla para registrarse con una cuenta ya existente (“Already have an Account? Login”).

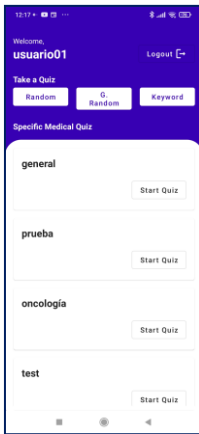
Figura 30 Página Sig Up



Esta imagen corresponde a la página de gestión de pruebas en modo “usuario anónimo”. La aplicación habilita cuatro posibles tipos de prueba:

- Prueba aleatoria mixta (preguntas reales y automáticas) “Random”.
- Prueba aleatoria de preguntas automáticas “G. Random”.
- Prueba basada en palabra clave “Keyword”.
- Prueba basada en especialidad médica “Specific Medical Quiz”.

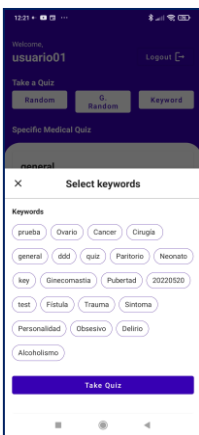
Figura 31 Página de examen / usuario no registrado / “homeFragment”



Esta imagen corresponde a la página de gestión de pruebas en modo “usuario registrado”. La aplicación habilita tres posibles tipos de prueba:

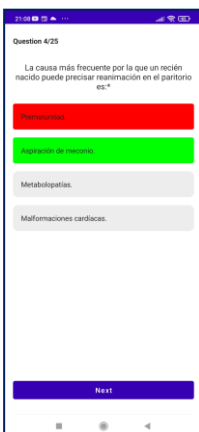
- Prueba aleatoria mixta (preguntas reales y automáticas) “Random”.
- Prueba aleatoria de preguntas automáticas “G. Random”.
- Prueba basada en palabra clave “Keyword”.
- Prueba basada en especialidad médica “Specific Medical Quiz”.

Figura 32 Página de examen / usuario registrado / “homeFragment”



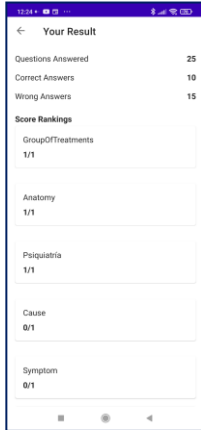
Esta imagen corresponde a la página de selección de palabra clave para el lanzamiento de una prueba.

Figura 33 Página de examen / usuario registrado / Selección de palabra clave / “homeFragment”



Esta imagen corresponde a un ejemplo de gestión de la respuesta a una pregunta. La imagen representa el caso de una respuesta incorrecta (sombreada en rojo) y la indicación de la respuesta correcta (sombreada en verde).

Figura 34 Página de examen / Gestión de las respuestas / “quizFragment”



Esta imagen representa un ejemplo de pantalla de resultados de una prueba para usuario registrado.

Es posible ver:

- El resumen general (número de aciertos y errores).
- Las cifras de aciertos y errores por tipo de especialidad cubierta por las preguntas.

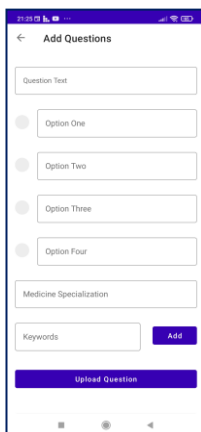
Figura 35 Página de resultados para usuario registrado / “resultFragment”



Esta imagen representa un ejemplo de pantalla de resultados de una prueba para usuario no registrado.

En este caso solo es posible ver el resumen general (número de aciertos y errores).

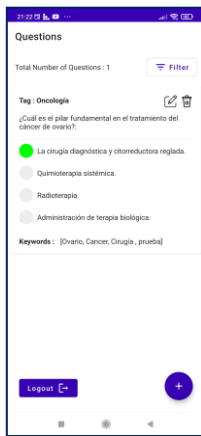
Figura 36 Página de resultados para usuario no registrado / “resultFragment”



Esta imagen representa la forma de cargar en la aplicación, manualmente, preguntas de examen. Además del enunciado de la pregunta y las opciones de respuesta, la aplicación permite:

- Seleccionar la respuesta correcta.
- Indicar la especialidad médica.
- Asociar posibles palabras clave.

Figura 37 Página de construcción manual de preguntas / “addQuestionFragment”



Esta imagen representa la manera en la que se visualiza una pregunta ya cargada. Además, se pueden ver las opciones que la aplicación ofrece para gestionar una pregunta cualquiera:

- Filtrado de preguntas por especialidad y palabra clave.
- El botón de edición.
- El botón de borrado de preguntas.
- El botón de “logout”.
- Y el botón de adición de nueva pregunta.

Figura 38 Página de edición de preguntas de examen / “addQuestionFragment”

8 CONCLUSIONES

En base a todo lo descrito en el presente documento se entiende que el resultado de este trabajo ha sido la generación de una aplicación operativa, capaz de gestionar la asimilación de conocimiento de las personas vinculadas con temas biomédicos, que permite ofrecer diferentes tipos de test, basados en preguntas directamente cargadas en la aplicación (denominadas a lo largo de este documento como “reales”) o bien generadas de manera “automática” a partir de la base de datos Copenmed.

En base a lo anterior y a la vista del funcionamiento de la propia aplicación, se entiende que ésta cumple con los objetivos y requisitos definidos, que podemos resumir así:

- La aplicación permite gestionar a sus usuarios de manera eficiente:
 - Mediante un sistema simple y eficiente de registro de usuarios y uso de la aplicación por parte de ellos.
 - Mediante la opción de uso de la aplicación por parte de usuarios no registrados, los cuales disponen de un número limitado de funcionalidades, entendida esta opción como un primer mecanismo de aproximación a la aplicación.
 - Mediante la opción de usuario administrador, principalmente orientada a la gestión de preguntas “reales” de test.
- La aplicación permite gestionar la incorporación de preguntas:
 - “Reales” procedentes de diversas fuentes, a través de un interfaz de carga manual de las mismas.
 - “Automáticas” a partir de bases de datos biomédicos procedentes de la Web Copenmed.
- Asimismo, permite la ejecución de diferentes tipos de test, a conveniencia del usuario, a partir de:
 - Mezcla de preguntas “reales” y “automáticas” en un mismo test.
 - Gestión de tests basados solo en preguntas “automáticas”.
 - Gestión de tests basados en especialidades concretas.
 - Gestión de tests basados en palabras clave asignadas a preguntas “reales”.

- La aplicación habilita la presentación de resultados de los tests (respuestas correctas totales y por especialidad, así como las incorrectas), con diferentes niveles de detalle según el tipo de usuario (registrado o no).
- Se puede verificar que el uso de la aplicación no exige un esfuerzo excesivo de navegación a través de ella (medible mediante el número de “clics” necesarios para alcanzar todas sus funcionalidades) ya que todas ellas están accesibles con un máximo de 3 clics.
- Adicionalmente, los tiempos de respuesta de la aplicación son inferiores al segundo en todas sus funcionalidades a excepción de la de generación de tests que incorporen la construcción “automática” de preguntas, en cuyo caso la aplicación requiere de menos de 5 segundos para presentar el test solicitado.
- En el caso de la ejecución de los tests, en todo momento el usuario tiene información de cuántas preguntas ya ha realizado y cuántas le quedan por hacer, para gestionar mejor sus expectativas a lo largo de cada prueba.
- En cuanto a experiencia de usuario, no existen diferencias apreciables tanto si se usa la aplicación desde un smatphone o desde una Tablet.
- Finalmente, tras un proceso concienzudo de pruebas a las que se ha sometido a la aplicación, ésta no presenta “zonas muertas” en las que la aplicación se quede congelada o en un estado de no prestación de su servicio.

Si ahora abordamos el tipo de información que la aplicación gestiona, se entiende que actualmente existe una gran oferta de aplicaciones destinadas a la gestión de conocimiento de temas biomédicos, si bien dicha oferta está basada, fundamentalmente, en el uso de preguntas de test “reales”, es decir, ya definidas por diferentes razones y usadas, por ejemplo, en oposiciones como la del MIR. A partir de este punto, el factor diferencial del presente trabajo es el uso de bases de datos de conocimiento biomédico que permiten la generación “automática” de preguntas de test. Haber sido capaces de implementar el mencionado mecanismo de construcción de las preguntas abre la

posibilidad de beneficiar al usuario de la aplicación de un universo de conocimiento en constante actualización y, por lo tanto, amplía de manera significativa su utilidad.

Asimismo, haber trabajado en el desarrollo del código a partir del sistema operativo móvil Android, habilita su uso por un colectivo potencial amplísimo de personas. En este sentido y según diversas fuentes consultadas, se estima que, a nivel mundial, la relación de usuarios de equipos móviles basados en Android versus aquellos basados en iOS es de aproximadamente 3 a 1 actualmente. Y, en particular, en Estados Unidos dicha relación es aproximadamente de 1 a 1. [16]

En base a todo lo aquí planteado se entiende que la aplicación desarrollada cumple con los requisitos mínimos establecidos y constituye un buen punto de partida para sucesivas mejoras, como se indica en el siguiente capítulo de este documento.

9 LÍNEAS FUTURAS DE TRABAJO

A partir del desarrollo realizado y analizando su nivel actual de prestaciones, se han identificado una serie de nuevas funcionalidades que pueden fácilmente ser incorporadas al programa y que constituyen potenciales futuras líneas de trabajo a desarrollar:

- En relación con la generación de preguntas de test:
 - Desarrollar filtros capaces de crear tests, mezclando más de una especialidad.
 - Crear una base de datos de preguntas generadas automáticamente, en vez de crearlas específicamente para cada test solicitado por el usuario (en las opciones “Random” y “G. Random”).
 - A partir de la mejora anterior, introducir la posibilidad de asignar “palabras clave” a las preguntas generadas de manera automática, toda vez que se quedarían almacenadas en la aplicación.
 - A su vez, las mejoras anteriores implicarían la necesidad de manejar actualizaciones de dichas preguntas (como consecuencia de las actualizaciones de la propia base de datos Copenmed) como un delta de “nuevas preguntas” y no como una carga nueva completa (que haría perder las palabras clave, salvo que estas se pudieran también generar de manera automática).
 - Incorporar la funcionalidad de carga de preguntas ya existentes en otras fuentes (por lo tanto, no generadas automáticamente por la aplicación), mediante un proceso batch para simplificar el proceso.
 - Asegurar la unicidad de la lengua de presentación de las preguntas “automáticas”. Por ejemplo, al español.
- En relación con la gestión de resultados de los tests realizados:
 - Desarrollar históricos de estadísticas de resultados de tests por usuario registrado.
 - Desarrollar listas de preguntas erróneamente contestadas por usuario registrado.
- En relación con el ‘look&feel’ general de la aplicación:

- Mejorar su apariencia (por ejemplo, en el tratamiento de los colores) y la gestión general de nuevas funciones.
- En relación con los requisitos legales de la aplicación, realizar un análisis en profundidad de estos y su adecuación de cara a una posible explotación comercial de la aplicación.

10 REFERENCIAS

- [1] <https://www.telefonica.com/es/sala-comunicacion/el-wifi-cumple-22-anos-y-alberga-dos-tercios-del-trafico-de-internet-en-espana/>
- [2] <https://www.universidadviu.com/es/actualidad/nuestros-expertos/evolucion-de-la-red-de-comunicacion-movil-del-1g-al-5g>
- [3] https://es.wikipedia.org/wiki/Tel%C3%A9fono_inteligente
- [4] https://www.elespanol.com/elandroidelibre/20180812/historia-sistemas-moviles-toda-competencia-android/329717692_0.html
- [5] https://www.prepmir.es/?gclid=Cj0KCQjw1N2TBhCOARIsAGVHQc4IQ3netvt16coTUuDILfS7eczqfHFurZwBfdwMm6bPRLysJbYQsaAjbJEALw_wcB
- [6] https://mirial.es/?utm_source=gads&utm_medium=sear&utm_campaign=brand&utm_content=432422936145&gclid=Cj0KCQjw1N2TBhCOARIsAGVHQc6ZP9X18WHCCtk7L98V2WGuE65HjtIHV2vSozrsL8n_a1wqPXj8F_gaAsruEALw_wcB
- [7] https://www.consalud.es/especial-mir/preguntado-examen-mir-2022-consulta-cuadernillos-imagenes_109206_102.html
- [8] <https://www.cmb.eus/actualidad-mir/8-aplicaciones-moviles-para-preparar-examen-mir>
- [9] https://www.cedimcat.info/index.php?option=com_content&view=article&id=183:fuentes-de-informacion-biomedica&catid=46:farmacoterapia-esp&lang=es
- [10] <https://sites.google.com/view/copenmed/inicio>
- [11] MIR Médico Interno Residente 1.175 preguntas de examen tipo test <oposiciones 2014 a 2018. Colección SaniTest (ISBN: 9781987464399)
- [12] <https://developer.android.com/studio/releases/build-tools.html>
- [13] <https://firebase.google.com/docs/firestore?authuser=0>
- [14] <https://firebase.google.com/docs/auth?authuser=0>
- [15] <https://developer.android.com/studio/test>
- [16] <https://clipset.com/ios-contra-android-quien-gana-la-batalla/>