



Universidad de

Valladolid

E.T.S.I. Telecomunicación
Trabajo de Fin de Grado

Grado en Ingeniería de Tecnologías de Telecomunicación

App móvil para ayuda en el estudio de la asignatura
“Simulación Clínica Avanzada” en el grado de Medicina

Autor:

D. Miguel Cantón González

Tutora:

Dra. Isabel de la Torre Díez

TÍTULO: **App móvil para ayuda en el estudio de la asignatura “Simulación Clínica Avanzada” en el grado de Medicina**

AUTOR: **Miguel Cantón González**

TUTOR: **Isabel de la Torre Díez**

DEPARTAMENTO: **Teoría de la Señal y Comunicaciones e Ingeniería Telemática**

TRIBUNAL

PRESIDENTE: **Beatriz Sainz de Abajo**

VOCAL: **David González Ortega**

SECRETARIO: **Isabel de la Torre Díez**

SUPLENTE: **Carlos Gómez Peña**

SUPLENTE: **Salvador Dueñas Carazo**

FECHA: **14 de junio de 2022**

CALIFICACIÓN:

Resumen

Hoy en día la tecnología, a través de los *smartphones* y las *tablets*, se está adentrando en la vida cotidiana de las personas, tanto en el hogar como en el trabajo. El incremento de uso de este tipo de dispositivos está directamente relacionado con el crecimiento de las aplicaciones móviles. Estas herramientas destacan por su facilidad de manejo y la gran variedad de utilidades que ofrecen, puesto que encontramos aplicaciones casi para cualquier ámbito.

Por estos motivos no es de extrañar que estén cobrando importancia aplicaciones desarrolladas para dos campos muy relevantes de nuestra sociedad, la educación y la salud, recogiendo estos trabajos bajo los términos *mLearning* y *mHealth* respectivamente. A nivel educativo, se están empezando a observar los beneficios de introducir dispositivos móviles en el aprendizaje, sobre todo en universidades. En el área de la salud, la mayoría de las aplicaciones en la actualidad están destinadas a mejorar la vida del paciente. Sin embargo, cada vez se están desarrollando más herramientas enfocadas en facilitar el trabajo de los profesionales sanitarios, sobre todo en la toma de decisiones, un aspecto clave de su labor.

Dentro de este contexto surge la idea del desarrollo de la aplicación e-poliTrauma. Esta app, disponible tanto para Android como para iOS, se centra en un área muy importante de la medicina prehospitalaria, la evaluación inicial de pacientes politraumatizados. Su objetivo principal es dotar de apoyo a los estudiantes de la asignatura Simulación Clínica Avanzada del grado de Medicina en la realización de sus prácticas. A pesar de esto, al haber empleado las últimas recomendaciones médicas, será también una herramienta útil para los profesionales de la salud. En concreto se ha empleado la metodología XABCDE, que evalúa de forma secuencial y sistemática cada uno de los sistemas del paciente, priorizando la búsqueda y el tratamiento de lesiones con compromiso vital.

Palabras clave

Aplicación móvil, *mLearning*, *mHealth*, Android, iOS, evaluación inicial, politrauma, XABCDE.

Abstract

Nowadays the technology, through smartphones and tablets, is entering people's daily lives, both at home and at work. The increased use of this type of devices is directly related to the growth of mobile applications. These tools stand out for their ease of use and the wide variety of utilities they offer, since we find applications for almost any field.

For these reasons, it is not surprising that applications developed for two very relevant fields of our society, education and health, are gaining in importance, collecting these works under the terms mLearning and mHealth respectively. At an educational level, the benefits of introducing mobile devices in learning are beginning to be observed, especially in universities. In health area, most applications today are aimed at improving patient's life. However, more and more tools are being developed focused on facilitating health professionals work, especially in decision-making, a key aspect of their job.

Within this context, the idea of developing the e-poliTrauma application arises. This app, available for both Android and iOS, focuses on a very important area of prehospital medicine, the initial assessment of polytraumatized patients. Its main objective is to provide support to the students of the Advanced Clinical Simulation subject of the Medicine degree in carrying out their practical classes. Despite this, having used the latest medical recommendations, it will also be a useful tool for health professionals. Specifically, the XABCDE methodology has been used, which sequentially and systematically evaluates each of the patient's systems, prioritizing the search for life-threatening injuries and their treatment.

Keywords

Mobile application, mLearning, mHealth, Android, iOS, initial assessment, polytrauma, XABCDE.

Agradecimientos

Para empezar, agradecer a mis padres, Luis y Leonor, y a mi hermano, Víctor, el apoyo incondicional que me han dado desde el comienzo de la carrera. Aunque hemos pasado por momentos difíciles durante estos años, juntos los hemos superado. Para mí son un ejemplo de lucha y superación, de caer y levantarse, y espero llegar a ser una parte de lo que son.

También dar las gracias a las personas que han hecho posible la realización de este Trabajo de Fin de Grado. A mi tutora, Isabel de la Torre, por siempre estar pendiente y guiarme en este proceso. Así como a Francisco Martín y Miguel Ángel Castro, doctores y profesores del grado de Medicina, por tener la idea de este proyecto y acompañarme mano a mano en su desarrollo.

Por último, agradecer a mis amigos, muchos de ellos compañeros de carrera, y al resto de familiares, por estar siempre a mi lado, levantándome en los malos momentos y disfrutando conmigo de los buenos, y siempre tener confianza plena en mí.

A todos, GRACIAS.

Tabla de contenido

CAPÍTULO 1. INTRODUCCIÓN.....	17
1.1 Motivación	17
1.2 mHealth y mLearning	18
CAPÍTULO 2. DESCRIPCIÓN DEL PROBLEMA	22
2.1 Contexto de la asignatura	22
2.2 Requisitos iniciales	23
2.3 Metodología de trabajo	24
CAPÍTULO 3. TECNOLOGÍAS Y HERRAMIENTAS EMPLEADAS.....	26
3.1 Tipos de aplicaciones	26
3.1.1 Aplicaciones web.....	26
3.1.2 Aplicaciones nativas	27
3.1.3 Aplicaciones híbridas.....	28
3.1.4 Solución planteada.....	29
3.2 Android.....	29
3.3 iOS	33
3.4 Herramientas utilizadas	34
3.4.1 Android Studio	34
3.4.2 Xcode.....	35
3.4.3 Sourcetree	36
3.4.4 ClickUp.....	36
3.4.5 VMware Workstation Player	37
CAPÍTULO 4. CONTENIDO MÉDICO	38
4.1 Motivación	38
4.2 Metodología XABCDE	39
4.2.1 Seguridad en la escena.....	40
4.2.2 Impresión de la gravedad.....	41
4.2.3 X: Exanguinación	41
4.2.4 A: Vía aérea	42
4.2.5 B: Respiración.....	43
4.2.6 C: Circulación.....	45
4.2.7 D: Deterioro.....	47
4.2.8 E: Exposición.....	48
4.3 Aplicaciones comerciales	49
CAPÍTULO 5. MANUAL DE USUARIO	55
5.1 Pantalla principal.....	55

5.2 Tutorial	56
5.3 Información	57
5.4 Contacto	57
5.5 Pantalla Covid-19.....	58
5.6 Elementos comunes	59
5.7 Seguridad en la escena.....	61
5.8 Impresión de la gravedad.....	62
5.9 Pantalla X: Exanguinación	64
5.10 Pantalla A: Vía aérea	67
5.11 Pantalla B: Respiración.....	71
5.12 Pantalla C: Circulación.....	76
5.13 Pantalla D: Deterioro.....	82
5.14 Pantalla E: Exposición.....	87
5.15 Pantalla tiempo finalizado.....	89
5.16 Pantalla final.....	90
5.17 Datos del paciente.....	91
5.18 Pantalla resumen	93
5.19 Listado simulaciones	94
5.20 Fichero Excel.....	96
5.20.1 Android.....	96
5.20.2 iOS	97
CAPÍTULO 6. DESARROLLO TÉCNICO	98
6.1 Librerías externas	98
6.2 Gifs animados.....	99
6.3 Listado de simulaciones	99
6.4 Base de datos	100
6.5 Fichero Excel.....	101
6.6 Preferencias de usuario.....	102
6.7 Pantallas emergentes.....	103
6.8 Localización del foco de sangrado.....	104
6.9 Diseño adaptativo	105
6.10 Aplicación multilingüe	110
CAPÍTULO 7. CONCLUSIONES Y LÍNEAS FUTURAS.....	111
7.1 Conclusiones.....	111
7.2 Líneas futuras	112
Bibliografía	114

Índice de figuras

Figura 1. Cuota de mercado mundial de sistemas operativos móviles. Enero 2022. Fuente: [22].	30
Figura 2. Cuota de mercado de sistemas operativos en todo el mundo. Enero 2022. Fuente: [23].	30
Figura 3. Distribución de versiones de Android. Noviembre 2021. Fuente: [25].	32
Figura 4. Distribución acumulativa de versiones de Android. Fuente: Android Studio.	32
Figura 5. Cuota de mercado de proveedores de móviles en todo el mundo. Enero 2022. Fuente: [26].	33
Figura 6. Mecanismo lesional y biomecánica en la colisión de un automóvil. Fuente: propia.	40
Figura 7. Colocación adecuada de la cánula de Guedel. Fuente: [35].	43
Figura 8. Drenaje de neumotórax a tensión. Fuente: [36].	44
Figura 9. Concentración y flujo de O ₂ para dispositivos de oxigenoterapia. Fuente: propia.	45
Figura 10. Regla de los pulsos. Fuente: propia.	46
Figura 11. Prueba de relleno capilar. Fuente: propia.	47
Figura 12. Interfaz de la app "Valoración primaria ABCDE". Fuente: Play Store.	49
Figura 13. Interfaz de la app "ABCDE". Fuente: Google Play.	50
Figura 14. Interfaz de la app "Urgencias extrahospitalarias". Fuente: Google Play.	51
Figura 15. Interfaz de la app "Polytrauma". Fuente: propia.	52
Figura 16. Interfaz de la app "MyATLS". Fuente: Google Play.	53
Figura 17. Pantalla principal. Fuente: propia.	56
Figura 18. Opciones del menú. Fuente: propia.	56
Figura 19. Pantalla del tutorial. Fuente: propia.	57
Figura 20. Pantalla de información. Fuente: propia.	58
Figura 21. Pantalla de contacto. Fuente: propia.	58
Figura 22. Pantalla dedicada a las medidas por la Covid-19. Fuente: propia.	59
Figura 23. Elementos comunes. Fuente: propia.	60
Figura 24. Pop-up de salida. Fuente: propia.	60
Figura 25. Pantalla dedicada a la seguridad en la escena. Fuente: propia.	61
Figura 26. Diagrama de flujo de la pantalla seguridad en la escena. Fuente: propia.	62
Figura 27. Pantalla dedicada a la impresión de la gravedad inicial. Fuente: propia.	63
Figura 28. Pantalla dedicada a la impresión de la gravedad completada. Fuente: propia.	63
Figura 29. Diagrama de flujo de la pantalla impresión de la gravedad. Fuente: propia.	63
Figura 30. Pantalla X inicial. Fuente: propia.	64
Figura 31. Pantalla X completada. Fuente: propia.	64
Figura 32. Pop-up localización de hemorragia. Fuente: propia.	65
Figura 33. Pop-up tipo de hemorragia. Fuente: propia.	65
Figura 34. Pop-up torniquete. Fuente: propia.	65
Figura 35. Diagrama de flujo pantalla X. Fuente: propia.	66
Figura 36. Pantalla A1 inicial. Fuente: propia.	67
Figura 37. Pantalla A1 completada. Fuente: propia.	67
Figura 38. Pantalla A2 inicial. Fuente: propia.	68
Figura 39. Pantalla A2 completada. Fuente: propia.	68
Figura 40. Pop-up control cervical. Fuente: propia.	69
Figura 41. Pop-up cánula de Guedel. Fuente: propia.	69
Figura 42. Pop-up maniobra de tracción mandibular. Fuente: propia.	69
Figura 43. Pop-up maniobra frente-mentón. Fuente: propia.	69
Figura 44. Diagrama de flujo pantalla A. Fuente: propia.	70
Figura 45. Pantalla B inicial. Fuente: propia.	71
Figura 46. Pantalla B completada. Fuente: propia.	71
Figura 47. Pop-up neumotórax a tensión. Fuente: propia.	72
Figura 48. Pantalla oxígeno inicial. Fuente: propia.	73
Figura 49. Pantalla oxígeno completada. Fuente: propia.	73

Figura 50. Pop-up frecuencia respiratoria. Fuente: propia.....	73
Figura 51. Pop-up saturación de oxígeno. Fuente: propia.....	73
Figura 52. Pop-up tabla de concentración de O2 para dispositivos de oxigenoterapia. Fuente: propia. ...	74
Figura 53. Diagrama de flujo pantalla B. Fuente: propia.	75
Figura 54. Pantalla C inicial. Fuente: propia.....	76
Figura 55. Pantalla C completada. Fuente: propia.....	76
Figura 56. Pop-up reponer volumen sanguíneo. Fuente: propia.	76
Figura 57. Pop-up estado de la piel. Fuente: propia.....	77
Figura 58. Pop-up tensión arterial sistólica. Fuente: propia.....	78
Figura 59. Pop-up frecuencia cardiaca. Fuente: propia.....	78
Figura 60. Pop-up relleno capilar. Fuente: propia.	79
Figura 61. Pop-up shock hipovolémico. Fuente: propia.....	80
Figura 62. Diagrama de flujo pantalla C. Fuente: propia.	81
Figura 63. Pantalla D inicial. Fuente: propia.	82
Figura 64. Pantalla D completada. Fuente: propia.....	82
Figura 65. Pantalla Escala de Coma de Glasgow inicial. Fuente: propia.	83
Figura 66. Pantalla Escala de Coma de Glasgow completada. Fuente: propia.	83
Figura 67. Pop-up apertura ocular. Fuente: propia.....	83
Figura 68. Pop-up respuesta verbal. Fuente: propia.	83
Figura 69. Pop-up respuesta motora. Fuente: propia.....	84
Figura 70. Pop-up pupila izquierda. Fuente: propia.	85
Figura 71. Pop-up pupila derecha. Fuente: propia.	85
Figura 72. Diagrama de flujo pantalla D. Fuente: propia.	86
Figura 73. Pantalla E inicial. Fuente: propia.	87
Figura 74. Pantalla E completada. Fuente: propia.	87
Figura 75. Pop-up para finalizar la simulación. Fuente: propia.....	88
Figura 76. Diagrama de flujo pantalla E. Fuente: propia.	89
Figura 77. Pantalla tiempo finalizado. Fuente: propia.	90
Figura 78. Pantalla final. Fuente: propia.	91
Figura 79. Pantalla datos del paciente inicial. Fuente: propia.	92
Figura 80. Error en la pantalla datos del paciente. Fuente: propia.	92
Figura 81. Pantalla datos del paciente completada. Fuente: propia.....	92
Figura 82. Pantalla resumen (1/3). Fuente: propia.	93
Figura 83. Pantalla resumen (2/3). Fuente: propia.	93
Figura 84. Pantalla resumen (3/3). Fuente: propia.	94
Figura 85. Pantalla listado de simulaciones. Fuente: propia.	95
Figura 86. Captura fichero Excel en Android (1/2). Fuente: propia.	96
Figura 87. Captura fichero Excel en Android (2/2). Fuente: propia.	96
Figura 88. Captura fichero Excel en iOS (1/2). Fuente: propia.....	97
Figura 89. Captura fichero Excel en iOS (2/2). Fuente: propia.....	97
Figura 90. Ejemplo pop-up Android. Fuente: propia.....	104
Figura 91. Ejemplo pop-up iOS. Fuente: propia.....	104
Figura 92. Pop-up localización de hemorragia en Android inicial. Fuente: propia.	105
Figura 93. Pop-up localización de hemorragia en Android completada. Fuente: propia.	105
Figura 94. Portada en Galaxy Nexus (Android). Fuente: propia.	107
Figura 95. Portada en Pixel C (Android). Fuente: propia.	107
Figura 96. Seguridad en la escena en Galaxy Nexus (Android). Fuente: propia.	107
Figura 97. Seguridad en la escena en Pixel C (Android). Fuente: propia.	107
Figura 98. Resumen en Galaxy Nexus (Android). Fuente: propia.	108
Figura 99. Resumen en Pixel C (Android). Fuente: propia.	108
Figura 100. Portada iPhone SE (iOS). Fuente: propia.	109
Figura 101. Portada iPad (iOS). Fuente: propia.	109
Figura 102. Suministrar oxígeno iPhone SE (iOS). Fuente: propia.	109

<i>Figura 103. Suministrar oxígeno iPad (iOS). Fuente: propia.....</i>	<i>109</i>
<i>Figura 104. Resumen iPhone SE (iOS). Fuente: propia.....</i>	<i>110</i>
<i>Figura 105. Resumen iPad (iOS). Fuente: propia.</i>	<i>110</i>

CAPÍTULO 1. INTRODUCCIÓN

En los últimos años estamos presenciando un gran desarrollo tecnológico. Cada vez se producen más avances y en periodos de tiempo más reducidos, consiguiendo un ritmo de permanente evolución que no parece que vaya a frenar en el futuro. Además, muchos de estos avances afectan directamente a la sociedad, que está integrando soluciones tecnológicas tanto en el hogar como en el trabajo.

Uno de los principales motivos de esta evolución tecnológica es facilitar la vida de las personas. Un ejemplo claro lo podemos ver en la actualidad, dentro del marco de la pandemia producida por la Covid-19. En una época en la que hemos tenido que hacer gran parte de nuestra vida desde casa, los avances en el ámbito de las telecomunicaciones nos han permitido estar conectados con el resto de las personas y, en muchos casos, seguir con nuestra labor de forma telemática. Por ello han cobrado importancia programas y aplicaciones de comunicaciones sobre Internet, como los destinados a la realización de videollamadas.

Hoy en día, la tecnología se ha adentrado en la mayoría de los ámbitos de la vida. Por tanto, no es de extrañar que haya ganado especial relevancia en dos campos vitales para la sociedad, como son la educación y la salud. En el primero de ellos, facilita tanto el trabajo de profesores como el aprendizaje de los alumnos, dotándoles de herramientas que hace no mucho tiempo eran impensables. En cuanto a la salud, la tecnología ayuda a los profesionales sanitarios, aportando herramientas para diagnóstico, análisis y cura de patologías e incluso para la toma de decisiones. También permite llevar un control de los pacientes y facilita la comunicación con los mismos.

En concordancia con estos avances, la tecnología se está afianzando cada vez más en la vida cotidiana de las personas. Buena parte de este hecho reside en el desarrollo de los *smartphones*. En la actualidad, la mayoría de las personas cuentan con uno de estos dispositivos y lo usan diariamente [1]. Este tipo de tecnología presenta varios beneficios, entre los que destacan la movilidad que ofrecen y que nos permiten estar permanentemente conectados, tanto a Internet como a otras personas. Además, las aplicaciones instalables en estos dispositivos aportan funcionalidades para casi cualquier aspecto de la vida, ofreciendo soluciones sencillas y accesibles de una forma muy rápida. Entre sus aportaciones tienen mucha relevancia las destinadas a la salud, aumentando en gran medida durante la pandemia.

1.1 Motivación

Dentro de este contexto en el que se está comprobando la eficacia de aplicaciones relacionadas con la salud, surge la idea de crear una aplicación móvil para dar apoyo a una asignatura del grado de Medicina de la Universidad de Valladolid, denominada “Simulación Clínica Avanzada”. En concreto, esta aplicación está vinculada a los ámbitos de la salud y de la educación, puesto que va a servir de ayuda a los estudiantes de la asignatura para completar conocimientos teóricos y, además, servirá de apoyo en el correcto desarrollo de las prácticas médicas.

La aplicación, que se llama e-poliTrauma, se centra en la evaluación inicial del paciente politraumatizado en el marco de una emergencia médica prehospitalaria para encontrar un diagnóstico adecuado. Se busca identificar, en primer lugar, lesiones de compromiso vital y determinar las actuaciones pertinentes de forma rápida. Aunque la aplicación está enfocada en tratar con pacientes cuyas lesiones son de origen traumático, de ahí su nombre, también sirve para abordar otros tipos de afecciones en pacientes críticos, puesto que se efectúa una primera

evaluación completa. Hay que tener en cuenta que la rapidez es una pauta fundamental en la realización de este proceso, ya que se trabaja para detectar lesiones que podrían causar el fallecimiento del paciente en caso de no ser atendidos correctamente en un corto periodo de tiempo. Por este motivo, la rapidez de uso de la aplicación ha sido un aspecto muy importante en su desarrollo.

El objetivo principal de la aplicación es dotar a los estudiantes de una herramienta interactiva que permita complementar las simulaciones, siguiendo secuencialmente los pasos que deben realizar para encontrar un correcto diagnóstico del problema que se presente, así como indicaciones y guías de las posibles actuaciones que haya que aplicar para encontrar la solución. De esta forma se busca que los alumnos afiancen los conocimientos de la asignatura, y que ganen en seguridad y rapidez en la realización de sus prácticas al tener a su alcance una guía con las recomendaciones de actuación.

Además, la aplicación presenta también funcionalidades para los profesores de la asignatura, en este caso los doctores Francisco Martín Rodríguez y Miguel Ángel Castro Villamor, puesto que los datos recogidos del desarrollo de la simulación por parte del alumno serán almacenados en una base de datos local dentro del dispositivo en que se realice, y podrán ser exportados a un fichero Excel. Por ello, el correcto empleo de la aplicación en busca de un diagnóstico acertado podrá ser tenido en cuenta por parte de los profesores en la nota de la práctica.

Fuera del ámbito de la asignatura, y aunque la aplicación está desarrollada para fines educativos, no deja de ser una herramienta médica que podrá ser usada por profesionales de la salud. Por esta razón ha sido muy importante seguir de forma estricta las últimas recomendaciones en cuanto a evaluación inicial sistematizada, empleando la metodología conocida como "XABCDE". En ella se recoge un análisis de todos los sistemas del cuerpo humano agrupado en secciones (cada una de ellas designada por una letra que viene de su nomenclatura en inglés) siendo muy importante el orden en que se abordan, ya que se centran primero en los problemas clínicos más peligrosos para la vida del paciente.

Cabe destacar que, en la actualidad, aunque el seguimiento de estas recomendaciones para la realización de la evaluación inicial del paciente está ampliamente extendido entre los expertos en emergencias médicas y profesionales de la salud en general, apenas existen aplicaciones móviles que recojan estas pautas, por lo que ha sido una buena oportunidad de crear una herramienta completa que realmente sirva de ayuda tanto a estudiantes de medicina como a la comunidad médica en general.

1.2 mHealth y mLearning

Un sector que avanza de la mano de la tecnología es la medicina. Este campo siempre se ha apoyado en los últimos avances tecnológicos para su desarrollo y lo ha hecho de diferentes formas, como puede ser el uso de las Tecnologías de la Información y la Telecomunicación (TIC), que han permitido mejorar procesos tanto a nivel puramente médico como en lo relativo a gestión de consultas. Dentro de este contexto surge un término denominado eSalud, *eHealth* en inglés. Aunque no se ha encontrado una definición ampliamente consensuada en la comunidad médica, la mayoría lo entienden, en términos generales, como el uso de tecnologías digitales aplicadas a la salud, resaltando sus funciones comunicativas, para diferenciarlo del campo de la informática médica, tal y como se recoge en [2]. Además, se añade en este artículo que, a pesar de que algunas definiciones se centran en los profesionales de la salud o los pacientes, normalmente se abarcan bajo este término aplicaciones para todas las partes implicadas.

La tecnología ayuda en casi todos los procesos llevados a cabo en los centros sanitarios. Entre ellos destacan los puramente médicos, donde entra la creación de equipamiento que permita la identificación y el tratamiento de enfermedades, o herramientas de análisis de datos, más conocidas en la actualidad como *BigData*, que permiten extraer información de utilidad a partir de datos recogidos de muchos pacientes. Además, también es útil en ámbitos hospitalarios de una forma más transversal, en lo relativo a almacenamiento de información, planificación y gestión de consultas, comunicación con los pacientes, etc.

Uno de los procesos en los que la tecnología ha adquirido más relevancia es en el diagnóstico clínico del paciente, por ser una de las tareas más relevantes de los médicos. Este es un procedimiento con un alto grado de incertidumbre, en el cual los profesionales de la salud combinan su conocimiento y experiencia en la búsqueda de una posible enfermedad, cuya detección temprana y acertada puede ser clave en muchas ocasiones. Por ello, gran parte de los actuales avances van destinados a la ayuda en este proceso.

Las tecnologías móviles también se han introducido en los entornos sanitarios, y más recientemente con el gran avance de los *smartphones*, permitiendo dotar a la medicina de otra herramienta más que puede ser muy beneficiosa. El uso de este tipo de dispositivos está ampliamente extendido en nuestra vida diaria, tanto a nivel global como nacional en España. Ya en 2017, un 94% de los encuestados en nuestro país en un estudio de *Global Mobile Consumer Survey*, que recoge la consultora Deloitte en [3], afirmaba haber utilizado un *smartphone* en las últimas 24 horas. Y el uso de esta tecnología no ha hecho más que crecer, en detrimento de otras como los ordenadores. Otro dato relevante recogido en la Encuesta sobre Equipamiento y Uso de Tecnologías de Información y Comunicación en los Hogares realizada por el Instituto Nacional de Estadística (INE) en 2021 [4] es que el porcentaje de hogares (con algún miembro entre 16 y 74 años) que cuentan con al menos un teléfono móvil es del 99,5%. Con estos datos no es de extrañar que se trate de implementar esta tecnología en uno de los ámbitos más relevantes a nivel social como es el de la salud.

En consonancia con este crecimiento se encuentra el desarrollo de aplicaciones móviles, un sector que sigue creciendo año tras año. Según recoge *Data AI* (antes llamado *App Annie*) en su informe anual, *State of Mobile 2022* [5], dos millones de aplicaciones móviles fueron lanzadas solo en 2021. Esto es debido a los grandes beneficios que aportan. En primer lugar, hay que destacar la rapidez de acceso a las mismas desde cualquier dispositivo y su facilidad de manejo. Además, cuestiones como la interactividad de forma visual por parte del usuario es un aspecto muy relevante. Y, por último, un gran beneficio que nos ofrecen es la movilidad. Hoy en día la mayor parte de la población no sale a la calle sin su teléfono móvil, y poder consultar una aplicación en cualquier momento y lugar es uno de sus principales atractivos. Por estas razones se busca desarrollar aplicaciones con un buen diseño de la interfaz de usuario, priorizando en muchas ocasiones su facilidad de uso.

El desarrollo de aplicaciones relacionadas con la salud es útil tanto para profesionales sanitarios, principalmente en cuestiones de diagnóstico, como para pacientes y población en general. También están apareciendo apps de gestión de citas médicas de administraciones públicas o privadas. Y es que en los últimos años se ha producido un amplio incremento de aplicaciones con fines sanitarios. Esto se constata de nuevo en el informe del INE [4] en el que se recoge que, de la población situada entre 16 y 74 años, un 48,7% en hombres y 54,0% en mujeres utilizan alguna página web o app móvil para concertar una cita médica. Y se espera que estos datos sigan creciendo en los años próximos, pues el empleo de estas herramientas está más instaurado entre los jóvenes.

El uso de las tecnologías móviles en el ámbito de la salud se conoce como *mHealth*, proveniente de *mobile health*, y es una parte de la *eHealth*. Una de las posibles definiciones de este término viene recogida en un informe de la Organización Mundial de la Salud realizado por el Global Observatory for eHealth (GOe) [6]. En él se relaciona la *mHealth* con las prácticas médicas y de salud pública soportadas por dispositivos móviles, como teléfonos móviles, dispositivos de monitorización, asistentes digitales personales (PDAs) y otros dispositivos inalámbricos.

Las implicaciones de la *mHealth* van desde las destinadas a profesionales sanitarios como a la población en general, pues son muy comunes, y cada vez más, las aplicaciones dedicadas a monitorizar el ejercicio físico y el bienestar. Entre las ventajas que ofrecen podemos encontrar: beneficios en regiones rurales donde el acceso a asistencia médica presencial no es posible de forma inmediata, ayuda en el diagnóstico de enfermedades, grandes funcionalidades a bajo coste en relación con otros equipos médicos especializados que en general son muy costosos y suelen necesitar de personal cualificado para su uso, capacidad para ceder la participación activa al paciente liberando de esta manera entornos hospitalarios en la medida de lo posible, gestión de información del paciente, reserva de consultas médicas y proporcionar información de forma rápida e interactiva.

En los tiempos actuales, con la pandemia producida por la Covid-19, se ha constatado y ha aumentado la importancia de este tipo de aplicaciones. Se han argumentado propuestas sobre el uso de la *mHealth* para el seguimiento de la enfermedad entre la población, con el fin de conseguir una detección temprana de los infectados, además de realizar análisis de los síntomas para una rápida actuación en caso de que los pacientes empeoren, como se recoge en [7]. Una de las posibles metodologías para aplicar estas funcionalidades es su implementación en aplicaciones móviles. Este tipo de herramientas han sido analizadas en distintos artículos como [8] y [9], donde se constata que la mayoría se han desarrollado con el objetivo de monitorizar síntomas y rastrear contactos. En relación con las apps de rastreo se concluye en [9] que solo son efectivas si son empleadas por toda la comunidad. Por estas razones va a ser interesante en los años venideros realizar estudios sobre sus implicaciones y beneficios en la pandemia vivida, con el fin de extraer información relevante que se pueda aplicar en futuros problemas sociales.

Hay que tener en cuenta que tanto Internet como las aplicaciones móviles son de tipo abierto y libre, por lo que cualquiera puede subir información. Y en un campo tan delicado como es el de la salud, hay que verificar que provenga de una fuente fiable, como pueden ser profesionales sanitarios u organismos oficiales.

Otro ámbito en el que el uso de aplicaciones móviles está ganando relevancia es en la educación. A raíz de esto, ha surgido el término *mLearning*, que forma parte del *eLearning*, y se puede definir como el uso de dispositivos móviles y tabletas como herramienta de apoyo en el proceso de aprendizaje [10]. Este es un campo todavía en desarrollo que se está empezando a utilizar sobre todo en universidades, donde la madurez de los estudiantes permite emplear estos métodos de forma adecuada. Sin embargo, ya hace años se apreciaban sus ventajas como se constata en un estudio realizado en la Universidad de Oviedo en el año 2013 [11]. Los autores concluyeron que los dispositivos móviles tenían un componente motivacional para los estudiantes aportando varias ventajas: popularidad, accesibilidad, inmediatez y motivación. En este estudio también se destacaba que los estudiantes universitarios de ciencias de la salud percibían estos recursos como beneficiosos, sobre todo en cuanto a realización de actividades prácticas.

Este TFG va a formar parte de estos dos aspectos tratados, mHealth y mLearning, ya que consiste en el desarrollo de una aplicación móvil médica para fines educativos en un entorno universitario. Por ello se ha buscado realizar un trabajo de calidad desde un punto de vista médico, empleando las últimas recomendaciones oficiales, sin dejar de lado que formará parte del desarrollo de una asignatura universitaria, por lo que hacer una aplicación interactiva, visual y fácil de emplear son cuestiones muy relevantes.

CAPÍTULO 2. DESCRIPCIÓN DEL PROBLEMA

Para continuar con la memoria conviene explicar la problemática de la que ha surgido la creación de la aplicación e-poliTrauma. Primero se presenta el contexto de la asignatura a la que va a dar soporte, para después analizar los requisitos iniciales solicitados y la metodología de trabajo seguida en el desarrollo del proyecto.

2.1 Contexto de la asignatura

La asignatura a la que la aplicación servirá de apoyo se denomina Simulación Clínica Avanzada (SCA). Forma parte del sexto curso del grado de Medicina de la Universidad de Valladolid y es impartida actualmente por los doctores Francisco Martín Rodríguez y Miguel Ángel Castro Villamor.

Se basa en sesiones prácticas desarrolladas en el Centro de Simulación Clínica Avanzada de la Facultad de Medicina, que fue inaugurado de forma oficial el 5 de mayo de 2021. La SCA se define como “una herramienta de aprendizaje que mediante el uso de simuladores y modelos anatómicos permite la recreación de escenarios clínicos con una gran exactitud, desde situaciones críticas difícilmente reproducibles hasta escenarios más sencillos donde la evaluación de la situación es la competencia fundamental por desarrollar”, como se informa por parte del Gabinete de Comunicación de la Universidad de Valladolid en [12].

Las prácticas educativas en SCA son muy novedosas, ya que se desarrollan en un entorno tecnológico que era inviable no hace mucho tiempo. La ventaja que presenta este tipo de simulaciones es que se trata de una experiencia inmersiva muy cercana a la realidad, por lo que los estudiantes pueden practicar lo que posteriormente se encontrarán en sus prácticas reales en un hospital, en condiciones de seguridad tanto para el paciente como para ellos. Con estos nuevos procedimientos educativos se busca conseguir la excelencia de los estudiantes de Medicina, adquiriendo competencias clínicas y comunicativas, de manera que cuenten con experiencia previa cuando se enfrenten a pacientes reales, disminuyendo así el margen de error humano.

En estas simulaciones, los alumnos se dividen en grupos de dos (uno es líder y otro actúa de apoyo) y se les presenta el caso de un paciente basado en situaciones clínicas reales experimentadas por los profesores en su trabajo. Con esto se busca potenciar aspectos relevantes del desarrollo profesional de los estudiantes como la toma de decisiones, la rapidez, el trabajo en equipo y la autocrítica.

La simulación se desarrolla en un periodo de diez minutos en el que se debe encontrar un diagnóstico del problema que presenta el paciente. Para ello pueden hablar con el supuesto paciente, un modelo anatómico equipado con una gran cantidad de sensores e interpretado (mediante voz) por el profesor Francisco Martín Rodríguez desde una zona colindante en la que monitoriza su estado. La sala cuenta con equipamiento médico real, como dispositivos de monitorización y oxigenoterapia, igual que el que se encontraría en un box de urgencias de un hospital. De esta forma los estudiantes, a partir de la información previa sobre el caso clínico (como pueden ser síntomas del paciente, antecedentes médicos, etc.) y los procedimientos médicos que consideren oportunos buscan determinar un diagnóstico adecuado. En el desarrollo de esta práctica están acompañados por una enfermera de la facultad de Enfermería que sirve de apoyo y hace todavía más real la experiencia. Además, si se precisa alguna prueba médica complementaria, es simulada por ordenador y presentada a los alumnos en pantalla

para su análisis. También tienen la posibilidad de emplear fármacos, e incluso si necesitan hablar con un especialista, lo realizan de manera real, siendo este interpretado por el profesor.

Además, hay que destacar que mientras dos alumnos están realizando la práctica, el resto de los estudiantes de la asignatura se encuentra en otra sala junto al profesor Miguel Ángel Castro Villamor donde pueden seguir el proceso de la simulación. Al finalizarla se debate si los pasos seguidos han sido correctos y si el diagnóstico final es acertado.

En el desarrollo de la práctica los alumnos se ven sometidos a la presión de determinar un diagnóstico adecuado en un tiempo de diez minutos, ya que la rapidez en la evaluación inicial del paciente es muy importante, pues se trata de determinar en muchas ocasiones problemas de compromiso vital, siendo cruciales los tiempos de detección y actuación. Otro de sus problemas es que se encuentran en un marco de incertidumbre en el que tienen que emplear los conocimientos que han ido adquiriendo a lo largo de la carrera, sin poder consultar ningún tipo de documentación médica en su toma de decisiones. Además, tienen nervios añadidos por estar siendo examinados, aunque los profesores se centran más en evaluar si los procedimientos y pasos seguidos son adecuados y bien argumentados, que en haber llegado a un diagnóstico correcto.

Dentro de este contexto surge la idea de la aplicación móvil en la que se basa este TFG. Su finalidad principal es presentar los pasos que los estudiantes deben seguir en el desarrollo de la práctica de una forma rápida, concisa e interactiva. Se busca mejorar uno de los problemas que se presentan en la realización de las simulaciones. Los alumnos, al estar bajo presión, en bastantes ocasiones se olvidan de ciertas actuaciones que deben realizar y su orden. Hay que destacar nuevamente que el orden en que se evalúan los distintos sistemas humanos es fundamental, pues se busca identificar en primer lugar las lesiones de compromiso vital que puedan afectar a la vida del paciente. Por ello, la aplicación viene presentada con un conjunto de pasos desarrollados de manera secuencial siguiendo las recomendaciones médicas actuales. Además, cuenta con información relevante y guías de realización de distintos procedimientos que los alumnos pueden consultar en el desarrollo de la práctica con la finalidad de que ganen en seguridad y rapidez, pudiéndose centrar en la búsqueda del diagnóstico sin la necesidad de ir recordando en cada momento el siguiente paso que deben seguir.

Para el correcto uso de la aplicación, el alumno de apoyo de los dos que están siendo evaluados contará con la aplicación móvil en un iPad proporcionado por los profesores. Además, hay que destacar que la pantalla del dispositivo será compartida en tiempo real al resto de alumnos de la clase y a los profesores de la asignatura. También podrá ser considerado su adecuado uso como una parte de la nota, por lo que será grabada, al igual que el resto de los procedimientos seguidos en estas sesiones.

Por último, cabe destacar que la aplicación también servirá de utilidad para los profesores de la asignatura, ya que los datos introducidos por los alumnos serán recogidos en una base de datos local en el dispositivo en que se realice la simulación y podrán ser exportados a un fichero Excel para su posterior análisis y posible calificación.

2.2 Requisitos iniciales

En una primera reunión con los profesores de la asignatura SCA se establecieron las principales características y funcionalidades que debía tener la aplicación. Aunque estas se fueron modificando y ampliando con el tiempo, tanto en lo relativo a interfaz gráfica como en

contenidos, el esquema principal, el cual presento a continuación, se mantuvo como fue diseñado inicialmente.

Para comenzar, se acordó emplear el método X-A-B-C-D-E como documentación médica para la aplicación. Este es un procedimiento ampliamente aceptado entre los profesionales sanitarios para la evaluación inicial del paciente. Para su correcta implementación, se planteó el uso de al menos una pantalla para cada campo que aborda.

Entre las características de la aplicación, se propuso que esta fuera muy interactiva y visual, incluyendo imágenes descriptivas de los distintos procedimientos cuando fuese necesario. Hay que tener en cuenta que la aplicación está diseñada principalmente para fines educativos, por lo que presentar información de calidad de forma concisa, interactiva y visual ha sido una constante en su desarrollo.

Desde el punto de vista de la interfaz gráfica se planteó el uso de un color de fondo de pantalla distinto para cada campo de la metodología en cuestión. En la medida de lo posible se intentó que fuese un color representativo del sistema humano a tratar, como el rojo para la circulación sanguínea y el azul para el sistema respiratorio. Esto es útil para identificar el campo que estamos abordando de forma rápida.

Entre las funcionalidades de la aplicación se estableció que contase con un temporizador de diez minutos y que pasado ese tiempo la simulación finalizase. Este tiempo es el mismo que tienen los alumnos de la asignatura para realizar la práctica y dar un diagnóstico.

Un aspecto muy importante es que la aplicación está diseñada para ser empleada de forma secuencial, de igual manera que la metodología en la que se apoya. Por esta razón se ha buscado que no se pueda avanzar de pantalla sin haber completado todos sus pasos. Se estableció que cuando se intentase avanzar sin haber finalizado todos los campos anteriores se informase por pantalla esta situación.

Otras de las funcionalidades que se solicitaron fueron el almacenamiento de los resultados de simulaciones anteriores dentro de la propia aplicación, lo cual ha sido implementado en una base de datos local, y, por último, la capacidad de exportar esta información a ficheros Excel para su posterior análisis.

Como cuestión final, en un principio la idea era desarrollar una aplicación nativa en el sistema operativo Android. Posteriormente, se planteó el poder hacer uso de esta aplicación en iPads para el desarrollo de las prácticas de la asignatura. Por esta razón se ha terminado realizando una aplicación nativa para el sistema operativo iOS idéntica a la primera en la medida de lo posible. Esto también es beneficioso para que todos los alumnos puedan descargarse la aplicación en sus dispositivos personales y así poder practicar con ella de forma individual, independientemente del sistema operativo que utilicen.

2.3 Metodología de trabajo

Respecto al proceso de trabajo, en el desarrollo de la aplicación he seguido varios pasos. En primer lugar, me he centrado en aprender los fundamentos del desarrollo de aplicaciones en Android. Antes de enfrentarme con este TFG tenía ciertos conocimientos previos, pero a un nivel muy básico. Por esta razón fue necesario seguir guías y cursos sobre este tema.

Tras contar con los conocimientos suficientes, tuve una reunión con los profesores de la asignatura SCA, Francisco Martín Rodríguez y Miguel Ángel Castro Villamor, en la que se

establecieron las funcionalidades principales que iba a tener la aplicación, desarrolladas en el apartado sobre requisitos iniciales de este documento (2.2). Además, asistí a una de las sesiones prácticas de la propia asignatura para entender el marco de trabajo en profundidad y los beneficios que iba a aportar la aplicación.

A partir de este momento comencé a desarrollar la aplicación en Android, manteniendo el contacto tanto con los profesores de la asignatura SCA como con mi tutora, la doctora Isabel de la Torre Díez. Tras varias reuniones en las que se iban añadiendo y modificando funcionalidades y contenido, se fue puliendo la aplicación hasta llegar a su estado final.

Cuando terminé el desarrollo de la aplicación en Android casi en su totalidad, comencé con el aprendizaje de la realización de aplicaciones en iOS, lo que supone el estudio de un nuevo lenguaje de programación y un entorno de desarrollo distinto. Además, para poder realizar aplicaciones para iOS, se necesita un dispositivo con este sistema operativo (uno de los *Mac* de *Apple* con *macOS*). Al no contar con uno, fue necesario realizar una virtualización sobre mi ordenador en Windows. Esto me ha permitido realizar la aplicación sin la necesidad de adquirir un *Mac*, pero tiene desventajas, ya que al realizar una virtualización el rendimiento de la máquina se ve mermada y el sistema en ocasiones se ralentiza.

En lo relativo al propio desarrollo de la aplicación en iOS, he intentado realizar una aplicación idéntica, en la medida de lo posible, a la app en Android. Aunque hay ligeras diferencias por las particularidades que presentan ambos entornos de desarrollo, sobre todo en cuestiones de interfaz, ambas cuentan con las mismas funcionalidades y se pueden usar de forma análoga. Esto es importante para que todos los estudiantes, tanto con dispositivos Android como con dispositivos iOS, cuenten con la misma información y que nadie se vea perjudicado.

CAPÍTULO 3. TECNOLOGÍAS Y HERRAMIENTAS EMPLEADAS

En el capítulo 3 se realiza un análisis de los tipos de aplicaciones existentes, tratándolos de forma individual. Se explican las ventajas y desventajas que presenta cada uno de ellos, y se concluye presentando la opción escogida y los motivos de esta decisión. Además, se recogen las principales características de los dos sistemas operativos a los que va a dar soporte la aplicación desarrollada, Android y iOS, destacando el lenguaje de programación utilizado en cada caso. Por último, se abordan las herramientas empleadas en la realización de este proyecto, entre las que se encuentran los entornos de desarrollo integrado (IDE) de cada sistema operativo, así como otros programas que han servido de ayuda en este trabajo.

3.1 Tipos de aplicaciones

Dentro de los tipos de aplicaciones que se pueden encontrar en los dispositivos móviles, tanto *smartphones* como *tablets*, se pueden diferenciar tres: aplicaciones web, aplicaciones nativas y aplicaciones híbridas. Cada uno de ellos presenta ciertos beneficios y desventajas respecto al resto, por lo que no hay una solución única. A la hora de desarrollar una aplicación hay que tener en cuenta qué opción se adapta mejor en cada caso, en función de los requerimientos que tengamos.

3.1.1 Aplicaciones web

Las aplicaciones web, más conocidas por su nombre en inglés *web apps*, utilizan un navegador para acceder al contenido web y mostrarlo, como viene definido en [13]. Son desarrolladas usando tecnologías habituales en el desarrollo web, siendo las principales: *HTML5*, *Javascript* y *CSS*. Además, se necesita programar la parte del *backend* en el lado del servidor, para lo cual se utilizan lenguajes como pueden ser: *PHP*, *ASP.NET*, *Python*, etc.

La principal ventaja de este tipo de apps es que son multidispositivo y multiplataforma. Esto quiere decir que se pueden utilizar en cualquier dispositivo móvil, mediante la implementación de un diseño adaptativo para distintos tamaños de pantalla, y son independientes del sistema operativo sobre el que se ejecutan (y de su versión). Con esto conseguimos una completa abstracción de la aplicación, necesitando únicamente de un navegador para su uso.

Estas características están directamente relacionadas con los costes y la dificultad que conlleva su desarrollo. Al necesitar un único código para abarcar todos los dispositivos móviles del mercado, esta solución es la más económica, sencilla y rápida de implementar respecto a los rivales que veremos a continuación.

Otra de sus ventajas es que no precisa de instalación en el dispositivo en el cual se vaya a ejecutar, sino que son accesibles directamente a través de cualquier navegador web del que dispongamos. Esto permite el ahorro en recursos *hardware*, principalmente en cuanto a procesador y memoria. Aunque el no necesitar instalación es mayormente un aspecto positivo, también tiene una parte negativa, ya que al no poder realizar su distribución en tiendas de apps oficiales (como son *Play Store* en Android y *App Store* en iOS) hay que hacer un mayor esfuerzo en cuestiones de marketing para poder llegar a los usuarios.

Al estar el código unificado, van a resultar aplicaciones con mayor facilidad de mantenimiento y actualización, ya que cualquier tipo de cambio que queramos realizar solo tendremos que implementarlo una única vez. Además, esta modificación se presentará de forma automática, sin necesidad de esperar su validación por parte de las tiendas oficiales.

Por último, hay que destacar que se gana en homogeneidad de la interfaz gráfica, ya que se va a presentar la información de igual manera en cualquier dispositivo, pero, en consecuencia, se pierde la riqueza de personalización que ofrecen las aplicaciones nativas.

En cuanto a las desventajas, la más relevante es que, al estar basadas en recursos web, necesitan conexión a Internet para funcionar, normalmente mediante un enlace de tipo cliente-servidor. Esta característica puede provocar ralentizaciones si se requiere de un gran volumen de datos y/o no se cuenta con buena conectividad, por lo que su uso no será óptimo. Por lo general, este tipo de aplicaciones es el más lento de todos, aunque dependerá de diversos factores.

Otro de los puntos negativos es que no exprimen la potencia del dispositivo físico sobre el que se presentan. A la hora de acceder a funcionalidades propias de sus módulos nativos (como pueden ser el GPS, el acelerómetro, la cámara, los altavoces, etc.), esto va a ser muy difícil de implementar, necesitando en muchos casos la creación de complejas *APIs* para su uso.

Por todas estas razones, las *web apps* van a ser útiles en casos concretos, como pueden ser aplicaciones basadas en el consumo de recursos por parte de un servidor. En contraposición, van a ser peor opción que las aplicaciones nativas cuando se necesite hacer un amplio uso del hardware del dispositivo, tal y como se discute en [14].

3.1.2 Aplicaciones nativas

Las aplicaciones móviles nativas se pueden definir como aquellas desarrolladas específicamente para una plataforma móvil (como puede ser Android o iOS), tal y como refleja *Bambú Mobile*, empresa especializada en desarrollo de software y creación de apps, en [15].

Los beneficios que presenta este tipo de aplicaciones son claros. En primer lugar, no se requiere el uso constante de conexión a Internet. Muchas de sus funcionalidades se podrán ejecutar sin contar con ella, aunque siempre será necesaria para comunicarse con servidores en caso de tener que acceder a recursos externos.

Estas apps hacen uso de toda la potencia del dispositivo, pudiendo emplear sus módulos físicos de una forma más o menos sencilla, así como la mayoría de sus funcionalidades. Por estos motivos, en general se va a obtener mejor rendimiento que con el resto de las opciones disponibles.

Una de las ventajas que siempre se ha tenido en cuenta al realizar este tipo de análisis es la rapidez de la propia aplicación. Al precisar de conexión, las apps nativas suelen ser más rápidas que las opciones basadas en tecnologías web. Además, para su desarrollo se emplea un lenguaje de programación admitido por el sistema operativo del dispositivo, por lo que se recortan tiempos al no necesitar realizar conversiones de código. Pero como se desarrolla en [16], la penalización en rendimiento, que es determinante en aplicaciones que requieran de la potencia del dispositivo (como puede ocurrir en juegos 3D o en procesamiento de imagen), resulta insignificante e incluso inapreciable en ciertas aplicaciones comerciales bien desarrolladas que usan tecnología web.

Respecto a la interfaz gráfica, este tipo de aplicaciones suele contar con mayor aprobación por parte de los usuarios. Al estar desarrolladas para un sistema operativo específico, la interfaz cuenta con detalles propios del mismo. Se consiguen crear apps con características comunes al propio entorno gráfico que presenta el dispositivo en cuestión, asemejándose al resto de aplicaciones del sistema. Con esto se consigue, por lo general, mejor experiencia de usuario.

El último beneficio por destacar es que se pueden distribuir mediante las tiendas oficiales. Con ello se consigue una mayor exposición al público y confianza de que el producto es seguro. Pero esto también conlleva aspectos negativos. Para su publicación hace falta adquirir una licencia por cada tienda. El coste para poder subir aplicaciones a *Google Play* de *Android* es de 25 dólares [17] siendo este un pago único. Por otro lado, formar parte del programa de desarrolladores de *Apple* [18] tiene un precio asociado de 99 dólares anuales. Con estas dos licencias se permite subir a las respectivas tiendas un número de aplicaciones ilimitado. Además de necesitar estos pagos, otra desventaja que presenta la publicación oficial de las apps es que conlleva revisiones, por lo que cualquier cambio que hagamos en la aplicación tendrá que ser aprobado por el organismo correspondiente y la actualización no será inmediata.

Pasando a los aspectos puramente negativos, hay que destacar principalmente la complejidad. Para poder hacer que una aplicación sea accesible para todos los dispositivos del mercado, se necesita crearla de forma independiente para los distintos sistemas operativos. Esto conlleva el uso de varios lenguajes de programación y diferentes entornos de desarrollo. Por lo general serán necesarios profesionales especializados para cada app, lo que implica un aumento considerable del coste de desarrollo.

Además, para realizar cualquier actualización de la aplicación se necesita implementar el cambio varias veces, una por cada sistema operativo sobre el que estemos desarrollando. Por lo tanto, aumentan también el coste y la complejidad de su mantenimiento a lo largo del tiempo.

3.1.3 Aplicaciones híbridas

El tercer tipo de aplicaciones se conoce como híbridas. Se trata de una unión de los dos anteriores, buscando tomar los aspectos positivos de ambos, aunque esto conlleva que presenten a su vez algunas de sus desventajas.

Las apps híbridas se desarrollan usando tecnología web, al igual que las *web apps*, por lo que un único código sirve para todos los tipos de dispositivos. A su vez, este código se asienta sobre un *framework* específico nativo del sistema operativo para el que se quiera desarrollar.

Entre las ventajas que presenta esta tercera clase podemos destacar varias. En primer lugar, al igual que las nativas no necesita de un navegador para funcionar, las apps son instaladas directamente en el dispositivo. Como hemos comentado, el grueso del código es único, lo que conlleva beneficios en sencillez, coste y tiempo de desarrollo, al igual que las *web apps*. Esto implica al mismo tiempo facilidades en cuestiones de mantenimiento y escalabilidad.

Estas aplicaciones se pueden distribuir en las tiendas oficiales, aunque en ocasiones se muestran reticentes si gran parte del código está implementado en lenguajes web, ya que los organismos correspondientes prefieren que se trabaje de forma nativa con sus entornos de desarrollo y lenguajes oficiales.

Entre las partes negativas que presenta destacamos que son más difíciles de implementar que las *web apps*, ya que se necesita una capa de conversión que conecte el código web con los diferentes sistemas operativos y plataformas. Por lo tanto, aunque la mayor parte del desarrollo sea común, tenemos varias aplicaciones independientes. Esta adaptación a cada plataforma se puede resolver con herramientas como *Apache Cordova*, *Ionic* o *Flutter* [19].

Además, aumenta la complejidad en el uso de los componentes hardware del dispositivo frente a las aplicaciones nativas. Y su implementación no está optimizada para cada sistema operativo, lo que puede conllevar peor rendimiento y producir ralentizaciones.

A nivel de usuario, se suele preferir el uso de aplicaciones puramente nativas por dos motivos: en general son más rápidas, y la interfaz está específicamente desarrollada para un sistema operativo concreto asimilándose a la del resto de componentes del sistema. Aunque es cierto que en determinadas aplicaciones que no hagan uso excesivo de la potencia del dispositivo, estas desventajas son casi imperceptibles para el público [19].

3.1.4 Solución planteada

Por lo general, el desarrollo de aplicaciones nativas va a obtener iguales o mejores resultados que sus rivales en cuanto a optimización, tanto en lo relativo al rendimiento como a la interfaz gráfica. Sin embargo, los otros tipos de aplicaciones, web e híbridas, ofrecen soluciones competentes reduciendo en gran medida costes en producción y mantenimiento, y recortando los tiempos en desarrollo. Esto conlleva un *trade-off* entre la optimización de la app y su coste asociado.

Por lo tanto, cada aplicación habrá que evaluarla de forma individual teniendo en cuenta el factor que sea más relevante. Al no haber una solución perfecta, la que se debe aplicar va a ser siempre la que mejor se adapte en cada caso, valorando los beneficios y desventajas que presente cada alternativa.

En nuestro caso de estudio, en un principio solo se iba a desarrollar una aplicación para dispositivos Android, por lo que se comenzó con el desarrollo de una aplicación nativa para este sistema operativo. Tras determinar la necesidad de implementarla para dispositivos iOS, se realizó una app nativa análoga a la primera para este nuevo sistema.

Aunque la solución fue en parte impuesta por los tiempos, es bueno hacer un análisis de cuál hubiese sido la mejor opción. La idea de crear una *web app* queda rápidamente descartada por la necesidad de conexión a Internet y su menor interactividad por parte del usuario. Al ser una aplicación que no hace uso excesivo de los sistemas físicos del dispositivo, salvo el acceso a memoria, se podría haber implementado con una solución híbrida empleando herramientas multiplataforma y así precisando un único código. Con esto hubiésemos conseguido recortar los tiempos de desarrollo logrando un resultado bastante similar al obtenido.

Sin embargo, desde un punto de vista académico y a nivel personal, la mejor idea era implementar la aplicación de forma nativa en los dos sistemas operativos en los que ha sido desarrollada. Aunque es una solución más compleja por tener que utilizar dos lenguajes de programación y dos entornos de desarrollo distintos, también es la más enriquecedora al adquirir conocimientos y experiencia en cada sistema operativo de forma individualizada.

3.2 Android

Android es un sistema operativo y una plataforma software basados en Linux. Fue creado inicialmente para teléfonos móviles, más tarde su uso se extendió a tabletas y, actualmente, podemos encontrarlo en todo tipo de dispositivos como relojes inteligentes, televisores, automóviles, etc.

Fue desarrollado por *Android Inc.*, empresa comprada por *Google* en 2005. Gran parte de su código fuente, inicialmente escrito en Java junto con C y C++, fue liberado al público a través del software Apache [20].

En la actualidad Android es usado por 2500 millones de dispositivos activos [21], y es el sistema operativo mayoritario. En la Figura 1, obtenida de la herramienta en línea que proporciona *Statcounter* en [22], en enero de 2022 el 69,74% de dispositivos móviles usan Android como

sistema operativo. El porcentaje de iOS es del 28,49%. Entre los dos obtienen un 99,23%, siendo el resto de los sistemas operativos residuales en comparación. Con estos datos podemos afirmar que Android es ampliamente el SO más usado del mercado en cuanto a dispositivos móviles.

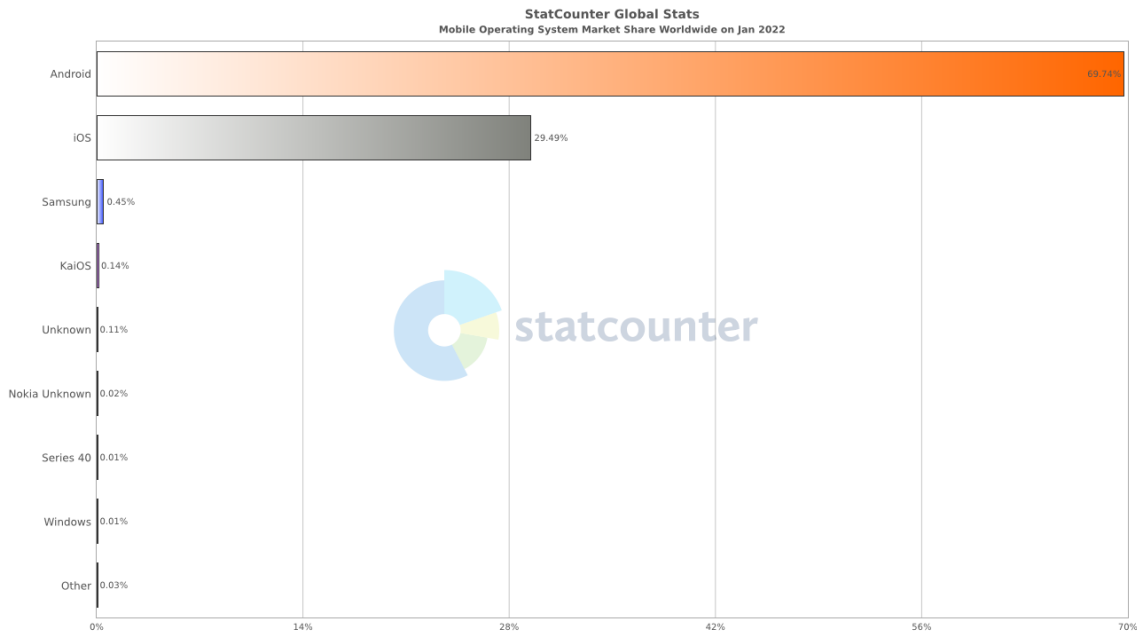


Figura 1. Cuota de mercado mundial de sistemas operativos móviles. Enero 2022. Fuente: [22].

Pero, además, Android es actualmente el sistema operativo más usado de todos los existentes en cualquier tipo de dispositivo, estando en un 39,45%, superando a Windows que se encuentra en el segundo puesto con un 32,11% y a iOS con un 17,56%, tal y como se puede ver en la Figura 2.

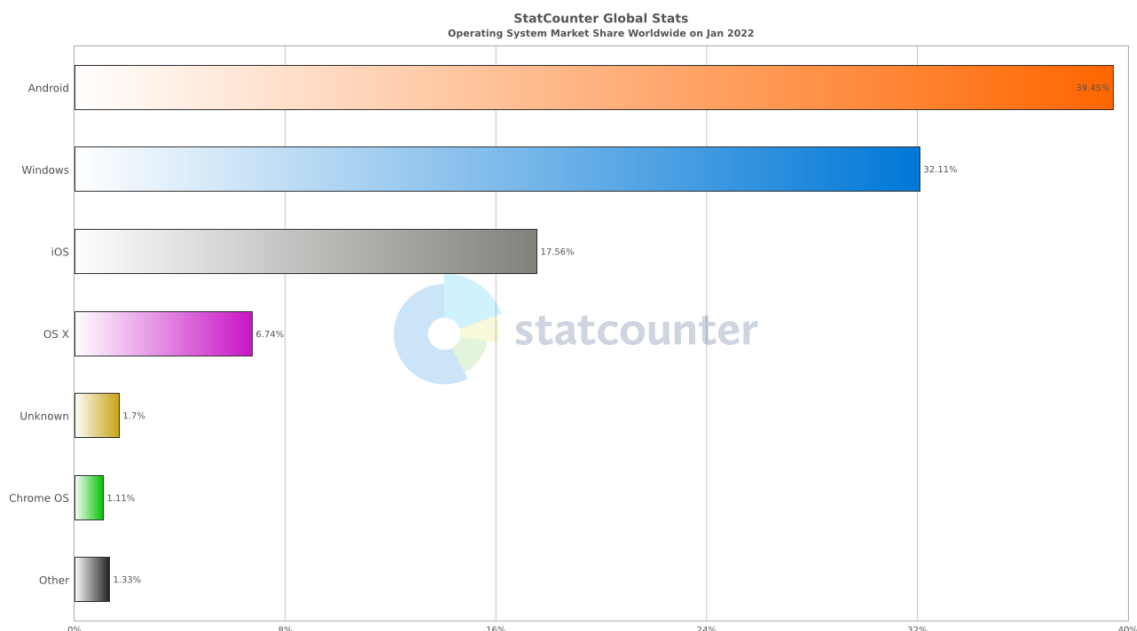


Figura 2. Cuota de mercado de sistemas operativos en todo el mundo. Enero 2022. Fuente: [23].

Para el desarrollo de aplicaciones, necesitamos descargar el *Android Software Development Kit* (Android SDK), que es un conjunto de herramientas y bibliotecas requeridas. Para trabajar con él, Google proporciona un entorno de desarrollo integrado (IDE) llamado *Android Studio*,

presentado de forma oficial en 2014 y basado en *IntelliJ IDEA* de *JetBrains*. Antes de esta fecha las aplicaciones se creaban en *Eclipse*.

En cuanto al lenguaje que se emplea, tradicionalmente se ha usado Java, un lenguaje orientado a objetos ampliamente extendido en el mundo de la programación. Sin embargo, en 2010 surge *Kotlin*, un lenguaje desarrollado por *JetBrains* que combina la programación orientada a objetos y la funcional, que se ejecuta en la *Java Virtual Machine* (JVM) y es interoperable con Java. Este nuevo lenguaje se caracteriza por ser conciso, eficaz y seguro, realizando lo mismo que antes con menos código, disminuyendo así el riesgo de presentar errores. Entre sus funcionalidades podemos destacar el uso de expresiones lambdas, funciones de extensión, control de nulidad en variables y corrutinas.

En la actualidad, el uso de Java está mucho más extendido que Kotlin, por lo que gana en cuestiones de experiencia de implementación (Kotlin no fue utilizado de forma oficial para desarrollo de aplicaciones móviles hasta 2017) y comunidad. Sin embargo, las características que ofrece Kotlin hace que sea mejor lenguaje de programación para el desarrollo de aplicaciones móviles. Y tal es así que, desde 2019, Google recomienda el uso de este lenguaje para el desarrollo de aplicaciones por delante de Java, determinando Android como “*Kotlin-first*”. Actualmente, más de 60 de las aplicaciones de Google (como *Google Maps*, *Google Play*, *Google Messages*, etc.) están desarrolladas en este nuevo lenguaje [24].

En el desarrollo de la aplicación que nos atañe he decidido usar Java por contar con conocimientos previos de este lenguaje, y tener a disposición mayor documentación sobre su uso. Sin embargo, en el futuro será una buena idea migrar la app a Kotlin.

La plataforma oficial de distribución de aplicaciones móviles para el sistema operativo Android se denomina Google Play. Para publicar aplicaciones propias, como ya hemos comentado, es necesario adquirir una licencia con un pago único asociado de 25 dólares. Con ella podemos subir un número de aplicaciones indeterminado. Además de publicar las apps de forma oficial, también es posible su distribución a través de un fichero llamado APK (*Android Application Package*), aunque siempre va a ser más seguro descargarlas de la tienda oficial ya que han pasado por una revisión para verificar que no contienen ningún tipo de *malware*.

Un problema que presenta el sistema operativo Android es la fragmentación. Al estar extendido entre la mayoría de los dispositivos del mercado, hay un gran número de tamaños de pantalla y resoluciones distintas a las que dar soporte con nuestra aplicación. Además, cada uno de ellos puede tener una versión de Android diferente, ya que muchos fabricantes dejan de dar soporte mediante actualizaciones a modelos antiguos.

Actualmente hay doce versiones del sistema operativo Android, tradicionalmente conocidos por el nombre de un dulce. La versión de Android 12 fue lanzada el 19 de octubre de 2021 pero todavía no ha sido implementada salvo en algunos dispositivos propios de Google. En la Figura 3 disponible en [25] podemos ver claramente este problema. En ella se ven los porcentajes de la distribución de cada versión de Android entre los dispositivos que usan este sistema operativo en noviembre de 2021.

Nov. 2021

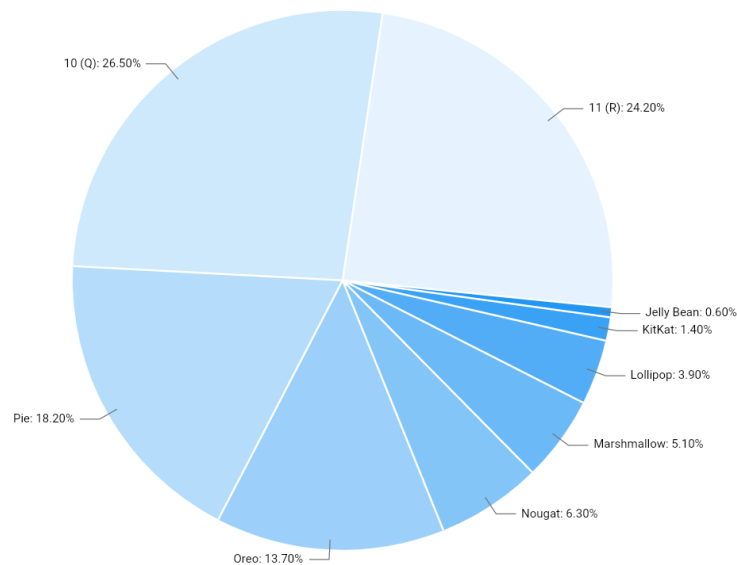


Figura 3. Distribución de versiones de Android. Noviembre 2021. Fuente: [25].

En el desarrollo de aplicaciones conviene dar soporte al mayor número de versiones de Android posible, para así poder dar cobertura a un mayor número de dispositivos. Sin embargo, hay ciertas funcionalidades y librerías que necesitan una cierta versión mínima para su implementación. En concreto, para mi aplicación he necesitado determinar la versión mínima del SDK en 25. Esto se corresponde con Android 7.1 denominado *Nougat*. Con esto conseguimos dar acceso a un 85,6% de los dispositivos Android en la actualidad, tal y como viene reflejado en la Figura 4 obtenida en *Android Studio*.

ANDROID PLATFORM VERSION	API LEVEL	CUMULATIVE DISTRIBUTION
4.1 Jelly Bean	16	
4.2 Jelly Bean	17	99,8%
4.3 Jelly Bean	18	99,5%
4.4 KitKat	19	99,4%
5.0 Lollipop	21	98,0%
5.1 Lollipop	22	97,3%
6.0 Marshmallow	23	94,1%
7.0 Nougat	24	89,0%
7.1 Nougat	25	85,6%
8.0 Oreo	26	82,7%
8.1 Oreo	27	78,7%
9.0 Pie	28	69,0%
10. Q	29	50,8%
11. R	30	24,3%

Figura 4. Distribución acumulativa de versiones de Android. Fuente: Android Studio.

3.3 iOS

iOS es el sistema operativo propio de *Apple* para dispositivos móviles que únicamente se puede emplear en los teléfonos de dicha marca. Fue presentado bajo el nombre de *iPhone OS* en enero de 2007, pero no fue hasta 2009 cuando se lanzó de forma estable junto con la presentación del *iPhone 4*.

Originariamente fue creado para smartphones, pero al igual que ha ocurrido en Android, se ha ido adaptando para ser utilizado en otro tipo de dispositivos como los *iPads* (cuya versión de sistema operativo se conoce como *iPadOS*) y los *Apple Watch* (relojes inteligentes de la propia compañía).

Como ya hemos comentado, iOS es el segundo sistema operativo más usado del mercado para dispositivos móviles. Aunque está claramente por debajo de Android, abarca casi un 30% de los dispositivos en datos de enero de 2022, tal y como se presentaba en la Figura 1. Para analizar estos resultados hay que tener en cuenta que Android está extendido entre la mayoría de los fabricantes de este tipo de dispositivos, mientras que iOS es un sistema operativo de *Apple*, implementado únicamente en sus propios smartphones. Esto lo vemos claramente reflejado en la Figura 5. En ella se muestran los porcentajes de cuota de mercado que presentan los principales proveedores de teléfonos móviles a nivel global. Como vemos, el porcentaje de *Apple* (29,49%) coincide con el que tenía iOS en la Figura 1, siendo la principal marca de dispositivos a nivel global, con *Samsung* como principal competidor (27.18%). A parte de *Apple*, el resto de los fabricantes emplean Android como sistema operativo, o una versión adaptada del mismo, por lo que se aprecia de forma clara la fragmentación que explicábamos en el apartado anterior.

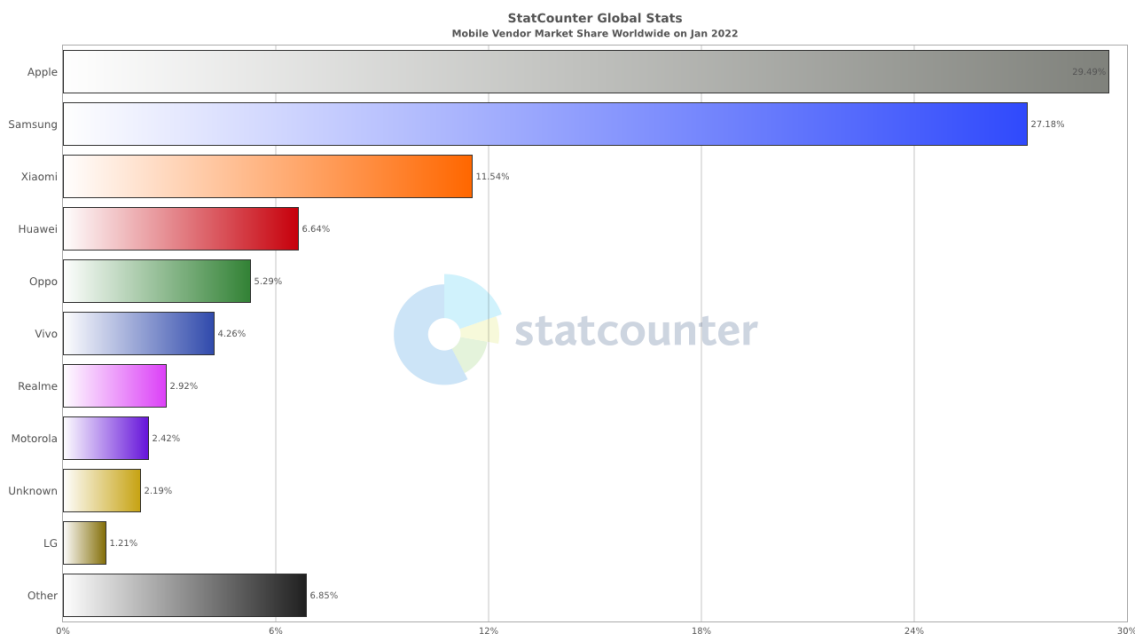


Figura 5. Cuota de mercado de proveedores de móviles en todo el mundo. Enero 2022. Fuente: [26].

Para el desarrollo de aplicaciones en iOS es necesario disponer de un dispositivo de *Apple*, normalmente uno de sus ordenadores conocidos como *Mac* (que cuentan con el sistema operativo *macOS*). Sin embargo, hay otra opción para programar apps sin contar con un *Mac* y es la virtualización. Esto consiste en ejecutar una máquina virtual sobre una máquina física que puede tener otro sistema operativo distinto. En este proyecto se ha realizado una virtualización de una máquina virtual con *macOS Monterey* sobre una máquina física con *Windows 10*. Para

realizar esta tarea, existen varias herramientas disponibles como pueden ser *VirtualBox* o *VMware Workstation Player*, habiendo elegido para este trabajo la segunda opción.

Además, para la creación de apps en iOS hay que hacer uso del entorno de desarrollo integrado que proporciona *Apple* de forma gratuita, llamado *Xcode*, que cuenta con todas las herramientas necesarias para esta tarea. En cuanto a lenguajes de programación, tradicionalmente se ha usado *Objective-C*, una extensión de C para adaptarlo a programación orientada a objetos haciéndolo más flexible. Sin embargo, en 2014 se presenta *Swift*, un lenguaje de programación de código abierto que busca ser seguro, rápido y conciso. Es compatible con *Objective-C*, y es más sencillo de utilizar que el anterior. Actualmente es el lenguaje que recomienda *Apple* para el desarrollo de aplicaciones, y al no conocer previamente a este trabajo ninguno de los dos lenguajes, he decidido usar *Swift*.

En cuanto al desarrollo de interfaces gráficas existen dos opciones que podemos usar dentro de *Xcode*, *UIKit* y *SwiftUI*. *UIKit* es un *framework* que permite crear interfaces utilizando los llamados *storyboards* (mediante un editor visual), o directamente de manera programática. Esta herramienta es la que más se ha usado porque era la única opción existente. Sin embargo, en 2019 fue presentado *SwiftUI*, un nuevo *framework* que permite diseñar y desarrollar interfaces declarativas, empleando menos código y más sencillo de entender. Esta es la apuesta que ha hecho *Apple* con vistas a futuro, y es la que recomienda utilizar actualmente si se empieza a desarrollar una app desde cero. Por esta razón, para la aplicación e-poliTrauma en su versión de iOS he empleado *SwiftUI*.

Al igual que en Android, *Apple* cuenta con una plataforma oficial de distribución de aplicaciones, denominada *App Store*. Para poder publicar en esta tienda es necesario formar parte del programa de desarrolladores de *Apple*, lo que implica un pago anual de 99 dólares. Con esta licencia se permite subir un número ilimitado de aplicaciones. Cuando queramos realizar alguna publicación, nuestro trabajo tiene que pasar por un proceso de revisión bastante estricto con la finalidad de asegurar su seguridad y su calidad.

Por último, conviene destacar que en iOS no es tan relevante el problema de fragmentación que sí importaba en Android. Esto es debido a que las aplicaciones para iOS van a ser distribuidas entre dispositivos de un único proveedor (*Apple*), por lo que el número de modelos distintos a los que adaptar la app es muy reducido. Además, las actualizaciones del sistema operativo se suelen implementar de una forma más o menos homogénea en todos los dispositivos de la marca (por los menos los más recientes). Con ello evitamos tener que hacer código soportable para muchas versiones del sistema operativo, como ocurría en Android.

3.4 Herramientas utilizadas

En este subapartado se realiza un pequeño análisis de todas las herramientas empleadas en el desarrollo del proyecto, indicando la función que ha tenido cada una de ellas.

3.4.1 Android Studio

Para el desarrollo de la aplicación en su versión de Android se ha usado el entorno de desarrollo integrado (IDE) oficial, proporcionado por *Google* y presentado oficialmente en 2014. Está basado en *IntelliJ IDEA*, un IDE desarrollado por *JetBrains* para el desarrollo de software principalmente en lenguaje Java, y más recientemente en Kotlin.

Este IDE busca unificar todas las herramientas necesarias para el desarrollo de aplicaciones Android de forma completa, incluyendo la creación del proyecto, escritura de código,

comprobación de errores, diseño de la interfaz y simulación en dispositivos virtuales. De esta forma, y al ser el programa oficial y recomendado por Google para el desarrollo de apps en su sistema operativo, ha sido la opción escogida. Entre las características que presenta esta herramienta se pueden destacar las siguientes [27]:

- Emplea un sistema de compilación flexible basado en *Gradle*, que es un paquete de herramientas para automatizar y administrar el proceso de compilación.
- Cuenta con un emulador rápido con múltiples funciones para visualizar la aplicación en dispositivos virtuales, pudiendo utilizar distintos modelos con diferentes tamaños de pantalla y versiones del sistema operativo Android. Además, permite realizar cambios en la aplicación mediante código en periodo de ejecución sin necesidad de reiniciarla.
- Tiene una herramienta bastante completa para el desarrollo de interfaces, pudiendo realizar este proceso mediante código (en archivos *XML*), de forma visual con un editor gráfico o empleando ambas opciones en paralelo.
- Es un entorno unificado para cualquier tipo de dispositivo Android, entre los que destacan principalmente los *smartphones* y las tabletas, pero están empezando a ganar importancia otros como los relojes inteligentes.
- Integra de herramientas *Git* de control de versionado como *GitHub*.
- Aporta herramientas de testeo y múltiples *frameworks*.
- Usa herramientas *Lint* para la comprobación de problemas de rendimiento y errores, como pueden ser los derivados de la compatibilidad de versionado.
- Compatibilidad con otros lenguajes de programación (además de Java y Kotlin) como C y C++.
- Da soporte para *Google Cloud Platform*, pudiendo realizar integraciones de *Google Cloud Messaging* y *App Engine* de forma sencilla.

Es conveniente mencionar que *Google* proporciona una gran cantidad de documentación en su web de *Android Developers*, incluyendo tutoriales, ejemplos y ejercicios. Esto permite adquirir conocimientos sobre el desarrollo de aplicaciones Android en *Android Studio* de una forma sencilla, gratuita y al alcance de todo el mundo. Además, se establecen las recomendaciones oficiales para unificar cuestiones programación dentro de cada equipo de trabajo.

3.4.2 Xcode

De forma análoga a *Android Studio* para Android, *Apple* ha proporcionado un IDE oficial para el desarrollo de aplicaciones en su sistema operativo. Este IDE, denominado *Xcode*, fue presentado en 2003 y actualmente va por su versión número 13.

Un aspecto a tener en cuenta es que *Xcode* se puede descargar de forma gratuita a través de la *App Store*, pero solo está disponible para dispositivos de *Apple*, por lo que es necesario tener uno de ellos de forma física o virtualizada.

Al igual que *Android Studio*, *Xcode* busca reunir todas las herramientas necesarias para el desarrollo de aplicaciones, en este caso en iOS y los sistemas operativos que derivan de este. Entre sus características destacan las siguientes [28]:

- Tiene herramientas de compilación propias para realizar este proceso de forma sencilla.
- Cuenta con un editor de interfaces muy completo, permitiendo el desarrollo de estas sin necesidad de escribir código.
- Tiene un simulador que, junto con el iOS SDK, permite instalar, ejecutar y depurar las aplicaciones que hemos creado en dispositivos de *Apple* virtualizados.

- Recientemente se ha añadido la herramienta *Canvas*, que permite visualizar la parte gráfica en tiempo real, aplicándose los cambios que hagamos en el código de forma inmediata.
- Este IDE unifica el desarrollo de aplicaciones para cualquier dispositivo de *Apple* y su sistema operativo asociado: *smartphones* (iOS), tabletas (iPadOS), ordenadores (macOS), relojes inteligentes (watchOS) e incluso televisores (tvOS).
- Cuenta con sistemas de construcción de las aplicaciones, lo que incluye su firma y su instalación.
- También tiene disponibles herramientas dedicadas al testeo de la aplicación, incluyendo su parte gráfica.
- Cuenta con un editor de versiones integrado, que permite el uso de sistemas del control *Git*.
- Es compatible con diversos lenguajes de programación, entre los que destacan C, C++, Objective-C, Objective-C++, Java, AppleScript, y el recomendado actualmente por *Apple*, Swift.

Al igual que en el caso de *Google* con Android, *Apple* también proporciona gran cantidad de información gratuita y de utilidad para hacer uso de sus herramientas oficiales, lo que permite adquirir conocimientos sobre el desarrollo de aplicaciones es su sistema operativo.

3.4.3 Sourcetree

Para el control de versionado y cambios en el desarrollo de la aplicación, la herramienta que he escogido ha sido *Sourcetree*. Este programa es uno de los mejores clientes GUI (*Graphical User Interface*) para la gestión de repositorios *Git* y *Mercurial*. Además, es perfectamente integrable con un repositorio en línea a través de *GitHub* o *Bitbucket* (el que he utilizado).

Esta herramienta es muy potente y nos facilita el trabajo con repositorios *Git* de forma sencilla a través de la interfaz gráfica que ofrece. Está disponible tanto para *Windows* como para *MacOS*, por lo que he podido utilizarlo de forma análoga en el desarrollo de la aplicación en ambos sistemas operativos, cuyos IDE son perfectamente integrables con repositorios *Git* a través de *Sourcetree*.

El trabajo con repositorios *Git* es necesario cuando se trabaja en equipo para llevar un control de los avances y unificar el código. Sin embargo, también es realmente útil en proyectos personales, pudiendo crear varias ramas de trabajo, gestionar los cambios que se realicen sobre cada fichero, tener grabaciones de respaldo de cada paso, y volver a alguna de ellas en caso de que sea necesario.

3.4.4 ClickUp

ClickUp es una herramienta de gestión de proyectos y tiempos, bastante extendida en el sector de desarrollo de software por las grandes funcionalidades que ofrece de forma visual y rápida. Tal y como describe la propia compañía, es una aplicación para “reemplazarlas a todas” queriendo unificar en una misma plataforma todas las herramientas de gestión necesarias. Además, añaden que su objetivo es “hacer el mundo más productivo” [29]. Entre las funcionalidades que ofrece se pueden destacar las siguientes:

- Listas *To-Do*.
- Personalización de las fases de cada tarea.
- Documentos y *Wikis*.
- Hojas de cálculo.

- Correo y chat entre miembros del equipo de trabajo.
- Eventos.
- Recordatorios.
- Seguimiento de objetivos (*goal tracking*).
- Registro de tiempos (*time tracking*).
- Administración de recursos.
- Diagramas de Gantt.

Aunque es una plataforma principalmente diseñada para el desarrollo de proyectos de software en equipos de varias personas, considero que también es bastante útil en la realización de trabajos personales. Además, cuenta con una versión gratuita con sus principales utilidades, lo que lo hace bastante atractivo frente a su competencia, que suele tener productos únicamente de pago.

Para el desarrollo de este TFG he decidido usar esta herramienta para hacer una planificación del desarrollo de la app, tener un lugar donde especificar los cambios a realizar tras cada reunión con los profesores de la asignatura y llevar un control de los tiempos para ser consciente del trabajo dedicado en cada fase.

3.4.5 VMware Workstation Player

Como ya hemos comentado previamente, para el desarrollo de aplicaciones en el sistema operativo de *Apple* es necesario contar con algún dispositivo de la propia marca, generalmente uno de sus ordenadores llamados *Mac*.

Sin embargo, existe otra opción para poder realizar esta tarea sin necesidad de contar con un *Mac* físico, y esto es mediante una virtualización. Este proceso consiste en recrear un entorno informático simulado sobre un dispositivo físico. En el caso de este proyecto, lo que se ha necesitado es virtualizar un *Mac* con *macOS* sobre un ordenador equipado con sistema operativo *Windows*.

Para este proceso existen herramientas que nos facilitan la creación de máquinas virtuales, siendo las más comunes *Virtual Box*, y la que he usado en este proyecto, *VMware Workstation Player*.

VMware Workstation Player es la versión gratuita del programa de virtualización que ofrece *VMware*. Esta herramienta aporta la capacidad de ejecutar una sola máquina virtual en equipos *Windows* o *Linux*, lo cual es suficiente para este proyecto y no se ha precisado de la opción de pago (*VMware Workstation Pro*).

Esta plataforma ofrece un producto bastante completo y sencillo de utilizar, permitiendo la elección del *hardware* del dispositivo físico que queremos reservar para la máquina virtual, como puede ser la memoria RAM, el disco duro o los núcleos. En nuestro caso, he realizado una virtualización de una máquina con *macOS 12 Monterey*, reservando 8GB de memoria RAM y 120GB para el almacenamiento.

CAPÍTULO 4. CONTENIDO MÉDICO

En este capítulo se aborda el contenido médico recogido en la aplicación desarrollada. Primero se presenta la motivación de la creación de este proyecto, resaltando la importancia de evaluar de forma rápida y sistemática a los pacientes politraumatizados dentro del marco de una emergencia prehospitalaria. Después se explica la metodología recomendada en la valoración inicial de este tipo de pacientes, denominada XABCDE, profundizando en cada una de sus fases. Por último, se analizan una serie de aplicaciones comerciales similares y se comparan con la app e-poliTrauma.

4.1 Motivación

Una de las principales causas de mortalidad a nivel mundial son los traumatismos. Según datos recogidos por la Organización Mundial de la Salud en [30], las lesiones, tanto las accidentales como las relacionadas con la violencia, suponen el fallecimiento de 4.4 millones de personas por todo el mundo cada año, lo que constituye casi el 8% de todas las muertes.

Entre las causas de fallecimiento de estos 4.4 millones anuales destacan los traumatismos causados por el tránsito (1.3 millones), las lesiones relacionadas con la violencia (1.25 millones), suicidios (700.000), caídas (674.000), ahogamientos, quemaduras e intoxicaciones.

Estos datos sobre mortalidad son especialmente relevantes en personas con edades comprendidas entre 5 y 29 años, ya que tres de sus cinco primeras causas de muerte están relacionadas con los traumatismos. Estas tres causas son: accidentes de tráfico, homicidios y suicidios.

Con estas estadísticas no es de extrañar que un campo con gran importancia dentro de la asistencia sanitaria sea el tratamiento de pacientes traumáticos. Se puede definir el trauma como “una lesión severa a nivel orgánico, resultante de la exposición aguda a un tipo de energía (mecánica, térmica, eléctrica, química o radiante), en cantidades que exceden el umbral de la tolerancia fisiológica” [31]. Dentro de las lesiones traumáticas, la mayoría de las que se producen son de carácter leve. Sin embargo, hay un cierto tipo de lesiones que pueden afectar a la vida del paciente o a su estado funcional, conocido de forma genérica como trauma grave. Otro término muy relevante es el de paciente politraumatizado, definido en [32] como “aquel que presenta lesiones a consecuencia de un traumatismo que afectan a dos o más órganos, o bien aquel que presenta al menos una lesión que pone en peligro su vida”.

Dentro de este contexto surge la idea de la aplicación en la que se basa este Trabajo de Fin de Grado. Su objetivo principal es aportar ayuda en la evaluación inicial de pacientes politraumatizados (de ahí su nombre, “e-poliTrauma”), priorizando la identificación y tratamiento de lesiones de compromiso vital. Sin embargo, también será útil para la evaluación de cualquier tipo de paciente crítico.

Esta primera valoración del paciente debe ser lo más completa posible y desarrollarse de manera sistemática, de forma que se comprueben todos los sistemas secuencialmente. Hasta que no se haya concluido la exploración de un determinado campo, se deben ignorar el resto de las lesiones, aunque estas sean obvias y llamativas. Esto se debe a que el orden seguido aborda en primer lugar las lesiones con mayor riesgo para la vida del paciente. El trabajo se puede realizar en equipo, comprobando de forma simultánea diferentes sistemas para iniciar su correspondiente tratamiento lo antes posible.

Un aspecto muy importante es la rapidez con la que se realiza la valoración. En la aplicación, el tiempo determinado para completar la evaluación es de diez minutos, ya que se encuentra destinado a estudiantes de Medicina. Sin embargo, el tiempo para un correcto desarrollo de este procedimiento está entre dos y cinco minutos.

Otros factores importantes a la hora de realizar la valoración primaria son el de no hacer más daño al paciente del que ya presenta, priorizar en todo momento la vida sobre la función, y esta sobre la estética. Y, por último, hay que tener en cuenta que el trauma es un proceso dinámico, por lo que continuamente se deberá realizar una reevaluación, haciendo especialmente hincapié en lesiones de compromiso vital como pueden ser las hemorragias exanguinantes.

Este procedimiento está especialmente diseñado para entornos prehospitalarios, dentro del contexto de una emergencia sanitaria. Sin embargo, el desarrollo de esta evaluación va a ser la misma en centros hospitalarios, en los que se complementarán las actuaciones de esta metodología con las herramientas disponibles en el box. En los primeros casos, antes de comenzar la evaluación del paciente hay que evaluar la seguridad de la escena, confirmando que el entorno sea seguro. Si esto no se consigue, se intentará movilizar a la víctima a una zona sin peligros potenciales o a un centro hospitalario.

Entre los beneficios que va a aportar la aplicación desde un punto de vista médico se pueden destacar varios. En primer lugar, se va a ganar en rapidez. Con el uso de la app, el personal sanitario ve en todo momento los pasos que se deben seguir, por lo que no se pierde tiempo en recordar cada uno de ellos. Por otro lado, como ya se ha comentado, la evaluación del paciente politraumatizado debe ser secuencial, un aspecto que se recoge de forma clara en la aplicación. De esta forma se van a identificar en primer lugar las lesiones de compromiso vital. Además, la aplicación cuenta con explicaciones sobre la manera de realizar ciertas maniobras sobre el paciente de forma visual, por lo que se podrá consultar esta información rápidamente para que los procedimientos seguidos sean más precisos. Por último, se van a ir recogiendo datos del paciente en el desarrollo de la valoración, que pueden ser útiles de consultar con posterioridad.

4.2 Metodología XABCDE

Tradicionalmente, para la evaluación inicial de pacientes politraumatizados se ha seguido la metodología ABCDE, denominada así por las siglas en inglés de los campos que trata: *Airway*, *Breathing*, *Circulation*, *Disability* y *Exposure* (en español: Vía Aérea, Respiración, Circulación, Deterioro, y Exposición). Este procedimiento aborda dichos sistemas humanos de manera secuencial, de forma que se priorizan, como ya se ha comentado, las lesiones que puedan afectar a la vida del paciente. Con esta regla se comenzaba con la valoración de la vía aérea del paciente, ya que esta era una de las principales causas de mortalidad en estas situaciones.

Sin embargo, en 2019 PHTLS (*PreHospital Trauma Life Support*) [33], el programa de educación prehospitalaria más reconocido a nivel mundial, cambió el procedimiento a XABCDE, en el que la X se corresponde con el término en inglés *Exsanguination* (Exanguinación). Por tanto, en este nuevo orden de la metodología, se tratan en primer lugar las hemorragias exanguinantes, ya que se vio que era la principal causa de muerte evitable entre pacientes politraumatizados. Las hemorragias antes eran tratadas en el campo C dedicado a la circulación. Con este cambio pasan a tratarse en primer lugar para realizar una inmediata actuación sobre las mismas, pero se vuelven a evaluar en el apartado C, junto con otros aspectos propios de la circulación sanguínea.

En el desarrollo de la aplicación se han seguido las recomendaciones expuestas en el manual PHTLS, puesto que son ampliamente reconocidas a nivel global y hay suficientes evidencias

científicas sobre los efectos beneficiosos del uso de la metodología XABCDE en la evaluación primaria de pacientes politraumatizados.

En los próximos subapartados se van a ir exponiendo los distintos sistemas sobre los que se apoya esta metodología, desarrollando la finalidad de cada paso seguido y los tratamientos recomendados. Los dos primeros no se corresponden a ninguna letra de las que dan nombre a este procedimiento, pero son cuestiones fundamentales que hay que tratar en las evaluaciones prehospitalarias.

4.2.1 Seguridad en la escena

Como primer paso de cualquier asistencia médica en un entorno prehospitalario, se debe realizar una valoración de la seguridad de la escena. Esto no forma parte de la regla de actuación XABCDE, pero es de vital importancia antes de iniciar su ejecución. Para ello se comienza con la utilización de equipo de protección individual, siendo lo más recomendable el uso de ropa de alta visibilidad, casco, gafas y guantes. Además, en el contexto de la pandemia derivada de la Covid-19 que estamos viviendo, es conveniente el uso de mascarillas, tanto por parte de los profesionales sanitarios como de los pacientes, y tomar la temperatura en caso de que la gravedad de la situación lo permita.

El siguiente paso consiste en la propia valoración de la seguridad en el entorno. En este momento se evaluarán los riesgos presentes o potenciales de la escena, siempre priorizando la autoseguridad sobre el resto de las acciones. En caso de que haya policía presente será obligatorio seguir sus indicaciones, y si la situación no permite realizar la asistencia en ese lugar o trasladar al paciente, se deberá abandonar la escena hasta que esta sea segura para todas las personas implicadas.

Si se tiene la seguridad de que no hay peligros en la escena, se procederá con la evaluación del paciente, realizando un análisis del mecanismo lesional de todas las secciones dañadas y la biomecánica implicada ante un posible impacto de carácter mecánico. Un ejemplo se presenta en la Figura 6 en la que se observan los mecanismos lesivos de un conductor de automóvil en caso de colisión de su vehículo.

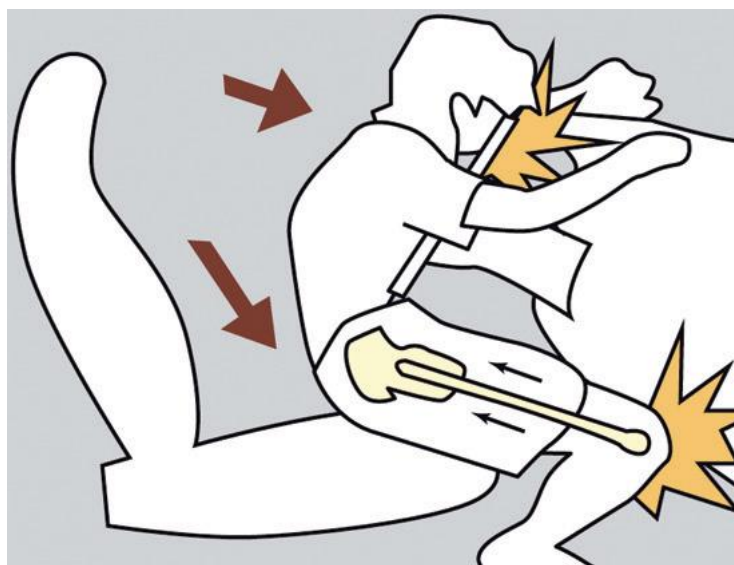


Figura 6. Mecanismo lesional y biomecánica en la colisión de un automóvil. Fuente: propia.

4.2.2 Impresión de la gravedad

Antes de realizar la evaluación sistematizada del paciente siguiendo la metodología XABCDE, se analiza la impresión de gravedad de la situación, pero siempre en caso de que esté garantizada la seguridad en la escena. En este paso se busca detectar en un corto periodo de tiempo el número de heridos para tener una visión global de la situación.

Ya enfocándonos en cada uno de los pacientes, se trata de determinar información relevante sobre los heridos y su situación. Entre estos aspectos se deberá determinar la edad, el sexo, el peso, el estado general, la posición y las lesiones evidentes. En este paso se pretende recoger información aproximada para tener un análisis previo antes de comenzar la propia evaluación, por lo que el tiempo dedicado debe ser muy reducido. También se recomienda detectar si el o los pacientes son capaces de responder a órdenes sencillas, ya que con ello se pueden descartar posibles lesiones que afecten directamente a su vida (aunque se deberá seguir realizando el análisis en los pasos posteriores).

Por último, es conveniente determinar en este momento la necesidad de recursos adicionales para realizar la evaluación, como puede ser el apoyo de más personal sanitario en caso de que haya múltiples pacientes.

4.2.3 X: Exanguinación

El primer campo a tratar en la evaluación del paciente politraumatizado, siguiendo la metodología XABCDE, es la exanguinación. En este paso se busca la identificación precoz de focos activos de hemorragia masiva que puedan afectar a la vida del paciente. Como ya se ha comentado, los focos exanguinantes son la principal causa de muerte evitable entre los pacientes politraumatizados, por lo que va a ser el primer paso que valorar.

Ante una hemorragia externa exanguinante hay que actuar de forma inmediata con la finalidad de controlarla y detenerla en la medida de lo posible. Es necesario diferenciar dos casos, si el foco de sangrado se encuentra en las extremidades, las recomendaciones de actuación para el control de la hemorragia consisten primero en una compresión directa manual o con vendaje compresivo (de ser posible transparente para poder observar la evolución). En caso de no conseguir detenerla se procederá con la aplicación de un torniquete. Esta es una solución efectiva en muchas ocasiones, pero debe ser empleada de manera correcta, puesto que puede provocar otro tipo de complicaciones locales y sistémicas [34]. Sus recomendaciones de uso son:

- Emplearlo solo si es estrictamente necesario.
- Aplicarlo tan cerca de la lesión como sea posible.
- Apretar justo hasta que la hemorragia se haya reducido lo suficiente como para poder controlarla mediante la aplicación de vendaje compresivo.
- Anotar la hora en la que se ha puesto, ya que una aplicación excesivamente prolongada (más de 1.5 – 2 horas) puede provocar lesiones musculares, nerviosas y vasculares, con un síndrome denominado parálisis del torniquete [34].

En caso de que el uso del torniquete tampoco sea efectivo, se emplearán agentes hemostáticos. Esta solución tiene por objetivo promover una rápida coagulación sanguínea, sobre todo en heridas sangrantes con hemorragia arterial, con el fin de favorecer la hemostasia (detección de la hemorragia).

Si nos encontramos ante una hemorragia exanguinante en una localización anatómica compleja (como puede ser en la cabeza, el cuello, el pecho, el abdomen, etc.) el primer paso es la presión

externa manual, ya que los vendajes compresivos no van a ser efectivos en estas zonas. En caso de no conseguir controlar hemorragia se pasa directamente al uso de agentes hemostáticos.

Para la valoración del foco de sangrado nos apoyaremos en la aplicación, primero indicando su localización en el cuerpo del paciente, y después identificando el tipo de hemorragia, diferenciando tres:

- Capilar: caracterizada por la salida de sangre en poca cantidad.
- Venosa: salida de sangre continua de color rojo oscuro.
- Arterial: salida intermitente con sangre de color rojo brillante.

En caso de conseguir controlar la hemorragia con alguna de las técnicas indicadas, obteniendo una hemostasia aceptable, se continúa con el resto de los sistemas de la evaluación inicial del paciente, pasando al análisis de la vía aérea. Hay que tener presente que los focos de sangrado pueden cambiar rápidamente, por lo que hay que realizar una valoración regular de los mismos, y en caso de empeoramiento regresar a este paso de la evaluación. Si el foco de sangrado no se consigue detener con los procedimientos citados, se buscará la evacuación rápida del paciente a un centro hospitalario para que pueda ser tratado con cirugía. Además, en los casos en los que haya algún cuerpo extraño enclavado en el paciente, también será necesario su traslado, puesto que retirarlo es muy peligroso en entornos prehospitalarios.

4.2.4 A: Vía aérea

El segundo campo que tratar en la metodología XABCDE se corresponde con la evaluación de la vía aérea (en inglés *airway*). El manejo inadecuado de la vía aérea en pacientes politraumatizados es la segunda causa de mortalidad evitable, justo detrás de las hemorragias exanguinantes, por lo que es importante realizar este paso cuanto antes.

El objetivo de este apartado es conseguir y mantener una vía aérea abierta y permeable, teniendo presente una posible lesión medular. Para comenzar, en caso de tener un paciente que presente algún tipo de traumatismo, se procederá con el control cervical siguiendo los siguientes pasos:

1. Tracción longitudinal de la cabeza.
2. Alineación cabeza-cuello-tronco.
3. Colocación de collarín cervical.

Tras esto comenzamos con el control de la vía aérea. Si el paciente habla o llora su vía se encuentra abierta y permeable, por lo que se podrá continuar con el siguiente campo de la evaluación inicial. En caso de que el paciente se encuentre inconsciente o con un bajo nivel de consciencia es necesario proceder con la apertura de la vía aérea, ya que en estas situaciones la lengua se desplaza hacia la parte posterior de la hipofaringe, pudiendo obstruir el paso del aire y derivar en una asfixia que podría causar la muerte del paciente.

Para realizar la apertura se pueden emplear dos maniobras distintas. Si hay sospechas de lesión en la columna cervical se aplica la maniobra de elevación o tracción mandibular. En caso contrario, se recomienda aplicar la maniobra frente-mentón. Tras realizar uno de estos procedimientos se continúa con la limpieza de posibles cuerpos extraños o secreciones (sangre, vómitos, piezas dentales, etc.) para permitir el paso de aire hasta los pulmones, estando siempre alerta ante signos de obstrucción real o potencial de la vía aérea. Otro aspecto a tener en cuenta es vigilar y valorar posibles quemaduras por inhalación que puedan afectar en la respiración del paciente.

Por último, si se ha conseguido tener una vía aérea abierta se busca mantener su permeabilidad. En caso de que sea necesario (pacientes inconscientes o con bajo nivel de consciencia), se emplea la cánula de Guedel. Para su correcta colocación, la cual se presenta en la Figura 7, se deben seguir los siguientes pasos:

1. Elegir el tamaño de la cánula, midiendo desde la comisura labial hasta el trago de la misma oreja o desde el ángulo mandibular hasta los incisivos.
2. Insertar cánula con la concavidad hacia el paladar duro.
3. Girar 180° al llegar al paladar blando.



Figura 7. Colocación adecuada de la cánula de Guedel. Fuente: [35].

4.2.5 B: Respiración

En este apartado se trata la respiración del paciente. Su letra asociada en la evaluación XABCDE es la B por el término en inglés *breathing*. El objetivo es comprobar que la función ventilatoria sea adecuada, ya que tener una vía aérea abierta y permeable no implica que el paciente esté respirando correctamente.

En primer lugar, se descubre el tórax y se realizan los siguientes pasos:

1. Auscultación: buscando deformidades, heridas, asimetría de ambos hemitórax, desplazamientos traqueales, signos de dificultad respiratoria, etc.
2. Auscultación: determinando si existe asimetría de ruidos respiratorios, abolición del murmullo vesicular, presencia de roncus y sibilantes, ruidos cardiacos apagados, etc.
3. Palpación: buscando presencia de enfisema subcutáneo, deformidades no visibles, fracturas, puntos dolorosos, etc.
4. Percusión: para encontrar zonas timpánicas (presencia de aire) o zonas mate (presencia de sangre).

En este punto es recomendable cerciorarse de que no haya neumotórax a tensión, que consiste en la acumulación de aire entre el pulmón y la pared torácica, aumentando la presión en el tórax y reduciendo la cantidad de sangre que llega al corazón. En caso de presentar signos claros del mismo, junto con extrema disnea, hay que abordarlo de forma inmediata, ya que puede afectar directamente a la vida del paciente. El tratamiento consiste en el drenaje mediante un catéter

de calibre 14 o 16 en el segundo espacio intercostal a la altura de la línea clavicular media, tal y como se puede ver en la Figura 8.

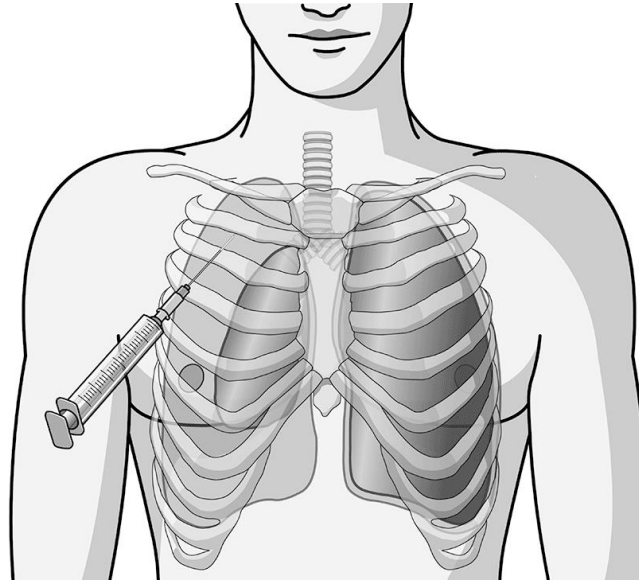


Figura 8. Drenaje de neumotórax a tensión. Fuente: [36].

El siguiente paso es conseguir una buena ventilación, con el fin de aportar un ambiente rico en oxígeno a las células. Será necesario administrar oxígeno al paciente, cuyo flujo y método de aplicación dependerá de la frecuencia respiratoria y de la saturación de oxígeno que presente.

Primero nos debemos cerciorar de que el paciente esté consciente y respirando, ya que de lo contrario habrá que realizar una intubación orotraqueal con control de la columna cervical. Si respira, se evalúa su ventilación determinando valores aproximados de la frecuencia respiratoria y la saturación de oxígeno. Los valores seleccionables en la aplicación para la frecuencia son:

- < 12 respiraciones por minuto: bradipnea.
- 12 – 20 respiraciones por minuto: ritmo normal.
- 20 – 30 respiraciones por minuto: taquipnea.
- > 30 respiraciones por minuto: taquipnea grave.

Mientras que para la saturación de oxígeno distinguimos tres estados:

- > 95%.
- 90% – 95%.
- < 95%.

En función de estos datos se elige el tipo de ventilación que aplicar al paciente con la finalidad de mantener una saturación por encima del 95% y un ritmo adecuado. Para ello es realmente útil la aplicación, ya que simplemente seleccionando los valores de saturación y frecuencia nos dice el tipo de ventilación que se recomienda aplicar, entre los que se encuentran:

- Gafas nasales: en caso de que el paciente esté respirando a un ritmo adecuado y con un buen valor de saturación (por encima del 95%), se considera la aplicación de oxígeno suplementario mediante este tipo de ventilación de forma preventiva para mantener su estado.
- Mascarilla tipo Venturi: para las situaciones en las que se presente una frecuencia respiratoria adecuada, pero con un valor de saturación de oxígeno algo bajo (90% - 95%).

- Mascarilla con reservorio: empleada si el paciente respira a un ritmo correcto, pero con un valor de saturación por debajo del 95%, o en situaciones de taquipnea (20 – 30 respiraciones por minuto) independientemente de la saturación.
- Ventilación asistida o total con balón reservorio, cánula de Guedel y oxígeno en concentración superior al 85%: este tipo de oxigenoterapia se aplica en los casos más graves, en situaciones donde la frecuencia respiratoria está por debajo de las 12 respiraciones por minuto (bradipnea) o por encima de las 30 (taquipnea grave).

La aplicación, además de recomendarnos el tipo de ventilación que aplicar, presenta una tabla de consulta sobre el flujo y la concentración de oxígeno correcto para los distintos dispositivos de oxigenoterapia, la cual puede verse en la Figura 9.

	FiO2 (%)	Flujo O2 (l/min)
Sistemas de alto flujo		
Máscara Venturi	25	3
	26	4
	28	5
	30	7
	35	10
	40	12
	50	15
Sistemas de bajo flujo		
Cánula nasal	24	1
	28	2
	32	3
	36	4
	40	5
	44	6
Tienda facial	40	5-6
Mascara de O2	40	5-6
	50	6-7
	60	7-8
Máscara reservorio	100	10-15

Figura 9. Concentración y flujo de O2 para dispositivos de oxigenoterapia. Fuente: propia.

4.2.6 C: Circulación

Si conseguimos mantener una buena ventilación en el paciente, se comienza con la evaluación de la circulación sanguínea. El objetivo de esta sección es obtener lo antes posible una adecuada perfusión tisular y evitar complicaciones potenciales asociadas a la pérdida de volumen sanguíneo.

En primer lugar, si la víctima presentaba hemorragias exanguinantes conviene revalorar su estado. Además, se determinará una pérdida estimada de volumen sanguíneo, sobre todo si presenta hipotensión después de un traumatismo, ya que será producida por hipovolemia mientras no se demuestre lo contrario. En estos casos se procederá a reponer el volumen, comenzando con un bolo de fluidos cristaloides de 500 mililitros y valorando el estado cada poco tiempo.

Tras este paso, es necesario evaluar una serie de datos clínicos que nos sirven para valorar la situación del paciente. El primero de ellos es determinar el estado de su piel, verificando su color y temperatura. Como norma general si un paciente politraumatizado presenta la piel rosada y caliente se puede descartar un shock hemorrágico grave. Por el contrario, si se detecta piel pálida y fría puede ser indicativo de pérdidas importantes de volumen sanguíneo, estimando al menos un 30% de la volemia.

El siguiente aspecto a tratar es el pulso del paciente. Por regla general, si el paciente politraumatizado tiene pulso rápido y filiforme (apenas perceptible), es un signo precoz de shock hemorrágico. Un estado correcto del pulso sería regular y no muy rápido. También es necesario evaluar la tensión arterial sistólica (TAS), empleando para ello la llamada “regla de los pulsos”. Esta es una técnica para encontrar una estimación rápida del TAS. Consiste en detectar la presencia de tres pulsos sobre el cuerpo del paciente: radial, femoral y carotídeo. Con la regla de los pulsos, cuyos puntos de palpación pueden verse en la Figura 10, se distinguen tres estados:

- TAS > 80 mmHg: si se palpa pulso radial.
- TAS entre 70 y 80 mmHg: si se detecta pulso femoral pero no radial.
- TAS entre 60 y 70 mmHg: si solo presenta pulso carotídeo.

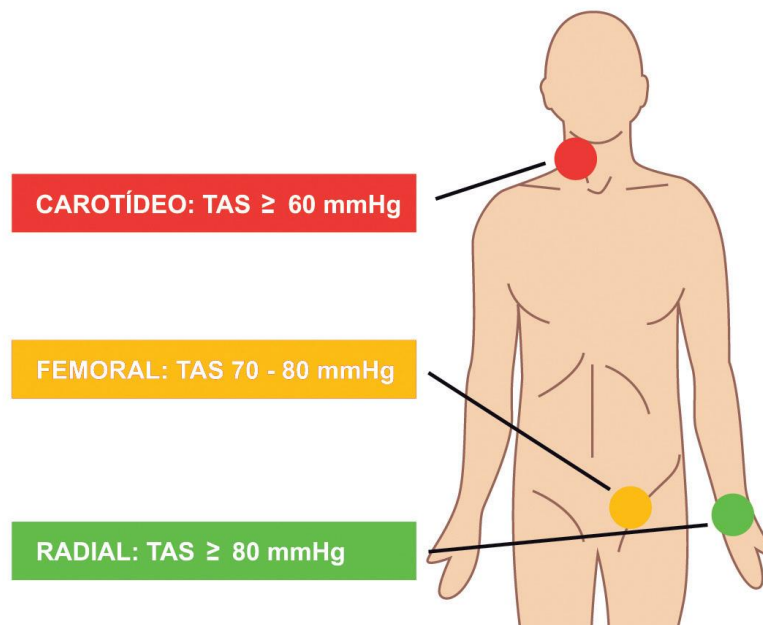


Figura 10. Regla de los pulsos. Fuente: propia.

Tras esto es necesario comprobar otros cuatro indicativos claros de shock hipovolémico. El primero de ellos es el relleno capilar. Para su evaluación se usa la técnica de apretar el lecho ungueal que puede verse en la Figura 11. Si la perfusión periférica del paciente es normal, el lecho recupera su color original en menos de dos segundos. Por el contrario, si se retrasa el llenado más de dos segundos, la perfusión periférica está disminuida.



Figura 11. Prueba de relleno capilar. Fuente: propia.

Los siguientes posibles indicativos de shock hipovolémico son la presencia de fracturas en huesos largos (como el fémur), la inestabilidad o fractura de pelvis y la presencia de defensa abdominal. Si se detecta al menos uno de estos cuatro indicativos deberemos considerar la presencia de shock hipovolémico.

4.2.7 D: Deterioro

La cuarta fase de la metodología de evaluación inicial que estamos utilizando tiene asociada la letra D, correspondiente al deterioro o déficit neurológico. El objetivo de este campo es realizar una valoración neurológica rápida, teniendo en cuenta en este momento solo el nivel de consciencia y la exploración pupilar. Con esto se busca evitar una posible lesión cerebral derivada de hipoxia o hipovolemia, así como descartar la presencia de lesiones ocupantes de espacio que requieran intervención quirúrgica inmediata.

Para evaluar el nivel de consciencia se recomienda el uso de la escala de coma de Glasgow (aunque si no se recuerda se puede emplear la regla AVDN). Esta técnica consiste en estimar la disminución del nivel de consciencia observando tres indicativos. A cada uno de ellos se le asigna el estado que presente el paciente, con un número asociado al mismo.

El primero de los indicativos es la apertura ocular, que cuenta con cuatro opciones dependiendo de cuándo el paciente abra los ojos:

- 4: Espontánea.
- 3: Al hablarle.
- 2: Al dolor.
- 1: Ausente.

El segundo es la respuesta motora, presentando seis estados:

- 6: Obedece órdenes.
- 5: Localiza el dolor.
- 4: Retira ante el dolor.
- 3: Flexión anormal.
- 2: Extensión.
- 1: No reacciona.

El tercero y último de los aspectos tratados es la respuesta verbal, diferenciando cinco opciones según la facilidad de palabra del paciente:

- 5: Orientado.
- 4: Conversación confusa.
- 3: Palabras inapropiadas.
- 2: Sonidos incomprensibles.
- 1: Ninguna.

El resultado de la escala de coma de Glasgow se obtiene sumando los números asociados a la apertura ocular, la respuesta motora y la respuesta verbal, quedando el valor comprendido entre 3 y 15. El número 15 se corresponde con la respuesta normal del paciente. Todo lo que sea inferior a ese valor significará que está alterado de alguna manera. Además, si el resultado es inferior a 9 y la respuesta motora está afectada (inferior a 6) se corresponde con un nivel de consciencia alterado, teniendo que considerar la intubación orotraqueal si no la hemos empleado en la fase de respiración.

El siguiente paso que realizar es la exploración pupilar, observando cada pupila por separado. Primero se evalúa la simetría, si ambas tienen el mismo tamaño se denomina isocoria y en caso contrario, anisocoria (con diferencia mayor a 1 milímetro entre ambas pupilas). Después el propio tamaño, detectando si el paciente tiene las pupilas puntiformes (miosis) o muy grandes (midriasis). También se analiza si presentan reactividad a la luz contrayéndose ante una iluminación directa con la linterna.

Por último, en esta fase conviene cerciorarse de que el paciente movilice las cuatro extremidades y que no presente hormigueos o falta de sensibilidad en estos miembros. En caso de tener falta de movilidad hay que poner especial atención en la focalidad neurológica, ya que podría estar afectada.

4.2.8 E: Exposición

Este apartado se corresponde con la última fase de la evaluación inicial. Su primer paso es el examen del paciente, que comienza exponiendo por completo el cuerpo en búsqueda de heridas, deformidades o cuerpos extraños no observados con anterioridad. Tras esto se procede con la estabilización del cuerpo.

El siguiente paso consiste en el control de la temperatura, teniendo especialmente cuidado con las pérdidas de calor, ya que un paciente hipotérmico presenta un estado inestable y empeorará su pronóstico.

Después se comienza con la monitorización disponible si no se ha realizado con anterioridad. Con ello conseguimos un mayor control sobre las constantes y el estado general del paciente, pudiendo detectar de forma más rápida alteraciones del mismo.

Por último, es conveniente valorar si se precisa de analgesia. El paciente por regla general no debe tener dolor, no solo por los componentes psicológicos que produce, sino que reduciendo el dolor se mejora de forma general su situación hemodinámica, por lo que es una medida necesaria y útil en la estabilización de pacientes críticos.

En este momento concluye la última fase de la evaluación inicial, pero no finaliza por completo esta valoración, puesto que se debe realizar una reevaluación permanente de su estado para anticiparnos y atajar problemas de forma rápida y efectiva. Además, será conveniente realizar una exhaustiva evaluación secundaria cuando sea posible.

4.3 Aplicaciones comerciales

En este subapartado se presentan varias aplicaciones móviles que guardan relación con la desarrollada para este TFG. Para ello, se ha realizado una búsqueda en las tiendas oficiales de apps tanto de Android como de iOS, *Google Play* y *App Store* respectivamente.

“Valoración primaria ABCDE”

Aplicación disponible tanto en Android como en iOS a un precio de 0,99 euros en castellano. Recoge la valoración primaria del paciente politraumatizado en el ámbito extrahospitalario siguiendo el protocolo ABCDE. Presenta el contenido casi en su totalidad de manera informativa en formato textual, por lo que es una herramienta más útil a nivel de estudio ya que no se pueden introducir resultados ni se ofrecen recomendaciones de actuación adaptadas a cada paciente. Su última actualización se realizó el 30 de diciembre de 2017, por lo que no cuenta con las últimas recomendaciones del protocolo incluyendo la X de exanguinación como primera fase de la metodología. En la Figura 12 se muestran capturas de su interfaz.

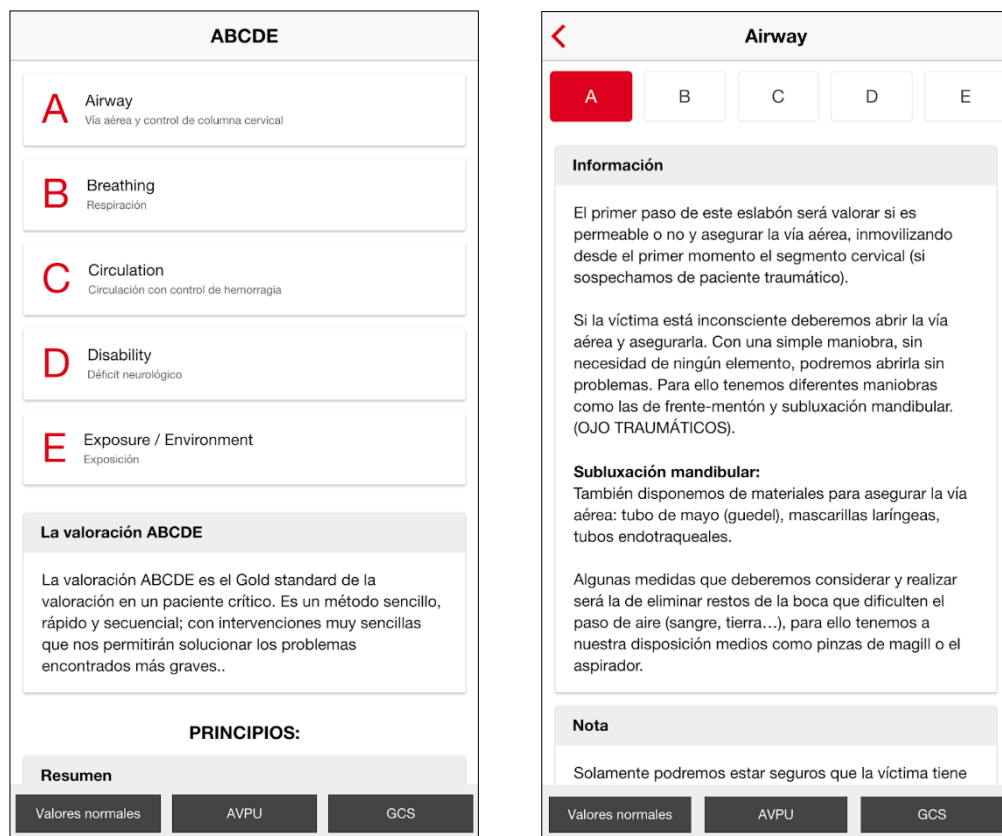


Figura 12. Interfaz de la app "Valoración primaria ABCDE". Fuente: Play Store.

“ABCDE”

Aplicación disponible en Android y iOS, nuevamente de pago con un precio de 0,99 euros. Desarrollada por la Fundación de Medicina Aguda del Norte de los Países Bajos (SAGNN). Está disponible tanto en idioma neerlandés como en inglés. Tal y como viene indicado en la información que proporciona en *Google Play*, está destinada al tratamiento de pacientes sin traumatismos en emergencias sanitarias por parte de profesionales de la salud. Recoge el

método ABCDE para el diagnóstico y tratamiento de enfermedades de compromiso vital, ofreciendo información básica práctica y concisa sobre los parámetros más relevantes. Su última actualización data del 7 de marzo de 2016. Se presenta la interfaz de la versión en neerlandés en la Figura 13.

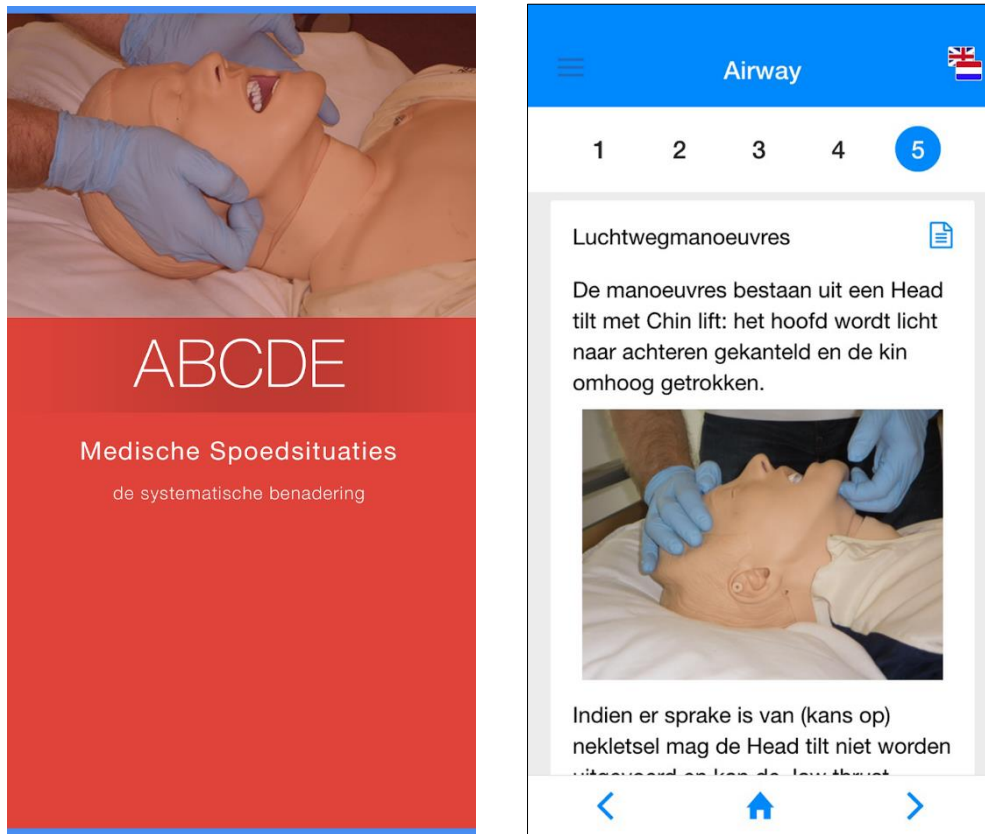


Figura 13. Interfaz de la app "ABCDE". Fuente: Google Play.

"Urgencias Extrahospitalarias"

Aplicación disponible en Google Play en castellano. Es gratuita, pero cuenta con anuncios. Está destinada exclusivamente a profesionales sanitarios, y viene definida como un libro de bolsillo sobre emergencias y urgencias extrahospitalarias. Contiene diagramas de flujo, imágenes, enlaces a vídeos de YouTube y buscadores para diversos tipos de contenido. Aborda cualquier tipo de paciente en emergencias médicas, pero cuenta con un apartado dedicado a la atención inicial del paciente politraumatizado en el que recoge la metodología ABCDE en formato texto como se muestra en la Figura 14. Aunque su última versión se implementó el 11 de febrero de 2020, no cuenta con la última actualización del protocolo XABCDE incluyendo la exanguinación como primer aspecto a tratar.

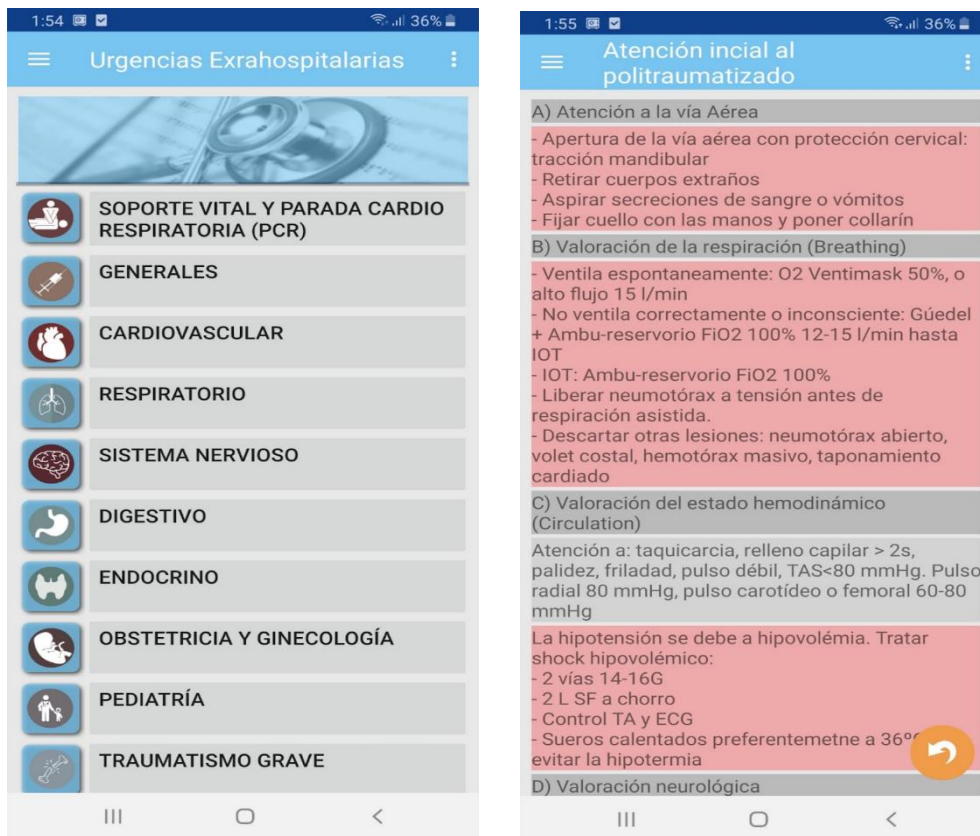


Figura 14. Interfaz de la app "Urgencias extrahospitalarias". Fuente: Google Play.

"Polytrauma"

Disponible para Android de forma gratuita en inglés. Aborda el tratamiento de pacientes gravemente lesionados. Cuenta con definiciones, diagramas de flujo, tablas y enlaces a información adicional. Tiene de nuevo un apartado destinado a la valoración inicial del paciente politraumatizado presentado en formato de texto que sigue el protocolo ABCDE según las recomendaciones de ATLS (*Advanced Trauma Life Support*), mostrado en la Figura 15. Esta aplicación ha sido actualizada por última vez el 26 de abril de 2017.

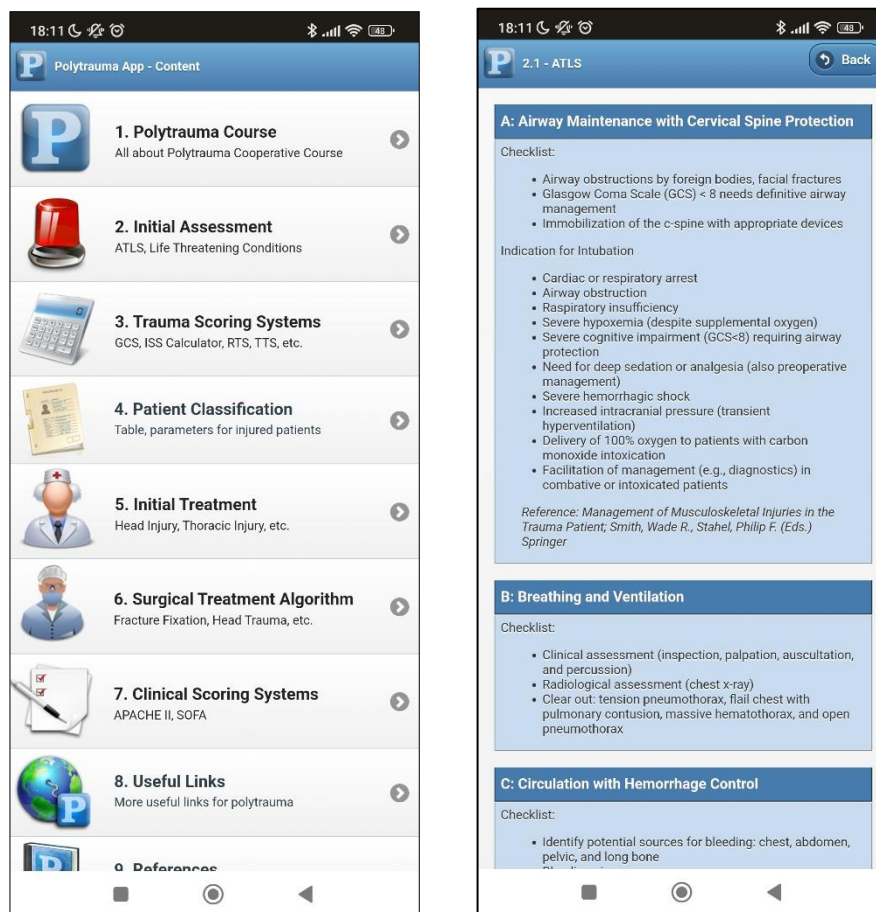


Figura 15. Interfaz de la app "Polytrauma". Fuente: propia.

"MyATLS"

Aplicación disponible en inglés tanto en Android como en iOS, creada por el ACS (Colegio Americano de Cirujanos). Muestra las recomendaciones de ATLS (*Advanced Trauma Life Support*) para la atención inicial del paciente politraumatizado. Cuenta con información en formato texto, además de imágenes interactivas, calculadoras y vídeos en los que se explica la correcta aplicación de los principales tratamientos. En uno de estos vídeos se presenta una evaluación inicial completa. Aunque su instalación es gratuita, su contenido solo está disponible si se realiza un pago de 10 dólares o si se introduce un código del libro ATLS. Su última actualización, del 19 de octubre de 2017, no cuenta con la información más actualizada del manual. Se muestran capturas de su interfaz en la Figura 16.

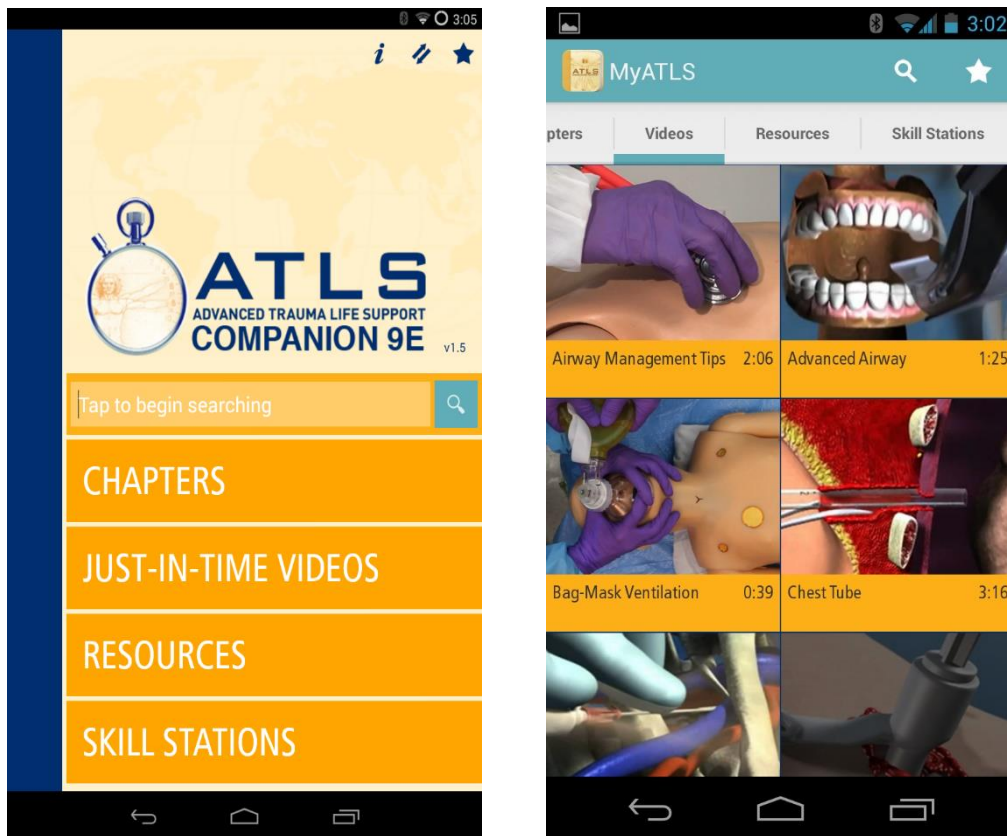


Figura 16. Interfaz de la app "MyATLS". Fuente: Google Play.

Conclusiones

En este apartado se han presentado varias aplicaciones disponibles en las tiendas oficiales que guardan relación con la app e-poliTrauma desarrollada en este TFG. Tres de las cinco aplicaciones evaluadas son de pago, otra es gratuita pero cuenta con anuncios ("Emergencias prehospitalarias"), y solo una de ellas ("Polytrauma") es gratis en su totalidad. Respecto al idioma en el que presentan la información, tres de ellas están disponibles en inglés y dos en castellano.

Las aplicaciones abordadas están destinadas al tratamiento de pacientes en emergencias sanitarias, la mayoría de ellas centrándose en el manejo del trauma. Para ello, emplean la metodología ABCDE en la evaluación inicial de este tipo de pacientes, por lo que no están actualizadas siguiendo las últimas recomendaciones. No se ha encontrado en el mercado ninguna app que presente la nueva versión de este protocolo incluyendo la X de exanguinación como primer aspecto a tratar (XABCDE).

Además, estas aplicaciones se basan casi en su totalidad en mostrar información mediante guías médicas, por lo que serán útiles en cuestiones de estudio. Sin embargo, no servirán realmente en la realización de la propia evaluación inicial del paciente salvo alguna consulta puntual, ya que al estar la mayor parte de la información en formato textual se perdería mucho tiempo. Presentan el contenido de una manera similar a lo que encontraríamos en una página web, sin hacer excesivo uso de las funcionalidades que ofrecen las aplicaciones móviles, por lo que la interacción resultante es pobre.

Por estos motivos la aplicación e-poliTrauma desarrollada se diferencia de las anteriores, presentando las recomendaciones más actualizadas con la metodología XABCDE, y ofreciendo una herramienta que presenta al usuario casi la misma información, pero de forma concisa, visual e interactiva.

Por último, cabe destacar que ninguna de las aplicaciones analizadas almacena valores y resultados de las evaluaciones. Por ello resaltan las funcionalidades que ofrece e-poliTrauma almacenando la información de cada simulación en la propia app, con la posibilidad de descargar los datos en un fichero Excel.

CAPÍTULO 5. MANUAL DE USUARIO

En este capítulo se presenta el contenido de la aplicación e-poliTrauma. Se realiza una explicación exhaustiva de la interfaz gráfica de las distintas pantallas, los elementos que la conforman, y las interacciones que podrán realizar los usuarios con cada uno de ellos. Para esta tarea se va a hacer uso de capturas de la aplicación en su versión de iOS, realizadas en el propio simulador de Xcode con el dispositivo iPhone SE. La app desarrollada para Android tiene, casi en su totalidad, la misma interfaz gráfica que para iOS. Ambas cuentan con los mismos componentes, por lo que el usuario final puede interactuar de forma similar en terminales de los dos sistemas operativos y, por tanto, este manual de usuario es perfectamente válido para ambas versiones de la aplicación.

5.1 Pantalla principal

Al abrir la aplicación aparece la pantalla inicial correspondiente a la portada, mostrada en la Figura 17, en la que vemos el color de fondo azul oscuro que presentan las pantallas genéricas de la app. En ella aparece el nombre de la aplicación (e-poliTrauma), la imagen de un paciente con traumatismos en una camilla, y varios componentes con los que el usuario puede interactuar:

- Botón “NUEVA SIMULACIÓN”: inicia una nueva simulación llevándonos a la pantalla sobre la Covid-19 si se encuentra esta opción activada, o directamente a la destinada a la seguridad en la escena en caso contrario.
- Botón “SIMULACIONES GUARDADAS”: nos redirige a la pantalla donde se presenta la lista de simulaciones guardadas en el dispositivo, lo que se puede corresponder con el historial de la aplicación.
- Botón menú: se encuentra en la esquina superior derecha. Al pulsarlo se despliega el menú mostrado en la Figura 18 que presenta cuatro opciones:
 - Activar/Desactivar pantalla Covid-19: con esta opción podemos determinar si queremos que aparezca la pantalla dedicada a las medidas en tiempos de Covid-19 al iniciar cada simulación. Al tratarse de medidas concretas para la pandemia en la que nos encontramos, se permite al usuario elegir si se muestran o no. Tras pulsar esta opción se presenta por pantalla un mensaje temporal indicando la elección tomada.
 - Tutorial: opción que redirige a la pantalla homónima en la que se explica el funcionamiento de la aplicación. También se especifica el lugar del dispositivo donde se encuentran los ficheros descargados.
 - Información: opción que nos lleva a la pantalla homónima en la que se presenta el contexto de la aplicación desarrollada, el título del TFG y el nombre de las personas involucradas en el proyecto.
 - Contacto: opción que nos lleva a otra pantalla en la que aparece un correo electrónico de contacto para temas relacionados con la aplicación.



Figura 17. Pantalla principal. Fuente: propia.

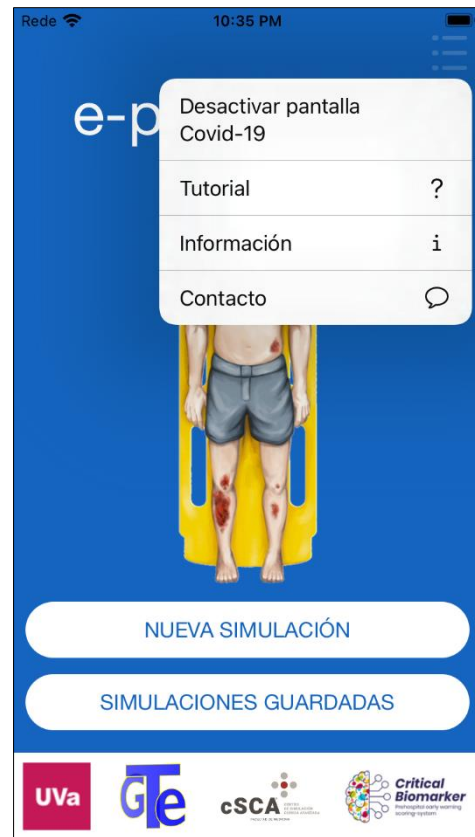


Figura 18. Opciones del menú. Fuente: propia.

En la parte inferior de la pantalla inicial, Figura 17 y Figura 18, aparecen los logos de las organizaciones involucradas en el proyecto: la Universidad de Valladolid, GTe (Grupo de Telemedicina y eSalud de la Escuela Técnica Superior de Ingenieros de Telecomunicación), cSCA (Centro de Simulación Clínica Avanzada de la Facultad de Medicina) y *Critical Biomarker*.

5.2 Tutorial

Pantalla a la que se accede desde el menú de la Figura 18. Presenta una breve explicación del funcionamiento de la aplicación. En ella se expone que está diseñada para completar cada una de las pantallas de forma secuencial, al ser el orden en que están dispuestas importante, ya que se tratan en primer lugar los sistemas que puedan provocar lesiones de compromiso vital. Por esta razón se recomienda completar todos los pasos de cada sección (momento en que el botón se ilumina) antes de pasar a la siguiente. Pese a esto puede haber ocasiones en las que necesitemos ir a una pantalla concreta rápidamente o retroceder a una previa en caso de cambios en el estado del paciente. Por ello hay dos flechas en la parte inferior de cada pantalla para ir al nivel anterior o posterior de forma inmediata.

También se presenta la ruta de la carpeta en la que se encuentran los ficheros que descarguemos con la aplicación. Por lo tanto el texto mostrado será distinto en Android (Archivos – Android – data – com.ePoliTrauma) y en iOS (Archivos – En mi iPhone/iPad – e-poliTrauma).

En la parte inferior de la interfaz, que se puede ver en la Figura 19, hay un botón con el texto “VOLVER” que nos lleva de vuelta a la pantalla principal de la app.



Figura 19. Pantalla del tutorial. Fuente: propia.

5.3 Información

A la pantalla de información se accede pulsando el botón correspondiente dentro de las opciones del menú de la pantalla principal. En ella se explica el contexto de la aplicación, exponiendo que se ha desarrollado como parte de este Trabajo de Fin de Grado, añadiendo el título del mismo. También se presentan las personas implicadas en el proyecto, la doctora Isabel de la Torre Díez ejerciendo de tutora y los doctores y profesores de la Facultad de Medicina Francisco Martín Rodríguez y Miguel Ángel Castro Villamor, que han colaborado en su desarrollo. De nuevo en la parte inferior de la interfaz, mostrada en la Figura 20, tenemos un botón de volver para regresar a la pantalla principal.

5.4 Contacto

En la pantalla contacto, mostrada en la Figura 21, a la que también se accede desde las opciones del menú en el inicio, se presenta una dirección de correo electrónico en la que se puede contactar en caso de que algún usuario de la app tenga alguna duda, comentario o sugerencia. El texto está acompañado de un gif animado de un sobre para hacer más visual el mensaje. Nuevamente cuenta con el botón de volver para regresar a la pantalla principal.



Figura 20. Pantalla de información. Fuente: propia.



Figura 21. Pantalla de contacto. Fuente: propia.

5.5 Pantalla Covid-19

Al comenzar una nueva simulación, aparece en primer lugar la pantalla relativa a las medidas por la Covid-19, la cual se ve en la Figura 22. En ella se recogen los procedimientos esenciales que hay que adoptar en cualquier contacto sanitario. Estos son el uso de mascarilla, el uso de guantes, y tomar la temperatura del paciente. Cada una de estas medidas está acompañada de un gif animado que sirve para representarla de una forma más visual y hacer la aplicación más atractiva a nivel de interfaz.

Esta pantalla se muestra por defecto cuando comenzamos una simulación (al pulsar el botón “nueva simulación” de la pantalla principal). Como son medidas básicas y comunes en cada simulación, y puede que en un tiempo sean innecesarias, he dispuesto una casilla (*check box*) en la parte inferior acompañada del texto “No volver a mostrar esta pantalla”. Al pulsar esta casilla o el texto correspondiente se guarda la elección. Aunque se cierre la aplicación no se volverán a mostrar estas medidas al comenzar una nueva simulación, ya que se almacena en la memoria del propio dispositivo mediante las preferencias de usuario. Si se cambia de idea y queremos que aparezca de nuevo, la app cuenta con la opción de activarla o desactivarla desde el menú de la pantalla principal, indicando la opción escogida con un mensaje temporal.

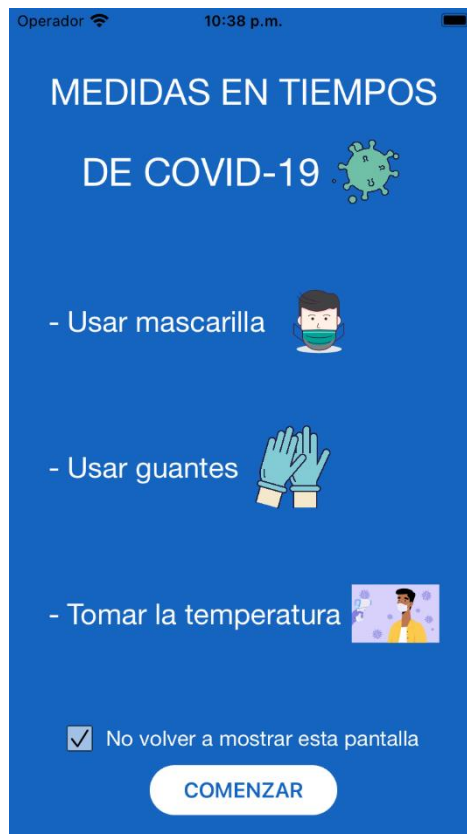


Figura 22. Pantalla dedicada a las medidas por la Covid-19. Fuente: propia.

5.6 Elementos comunes

En los próximos subapartados se tratan las pantallas dedicadas a la propia simulación. Aunque cada una de ellas va a tener un color característico correspondiente a la cuestión médica que se esté tratando (las propias de la metodología XABCDE), todas ellas cuentan con una serie de elementos comunes en la interfaz, los cuales presento en la Figura 23. Entre estos componentes, que vemos señalados en la imagen, encontramos:

1. Barra de progreso: esta *progress bar* nos permite ver el avance en la simulación hasta completar todos los pasos. Se va actualizando de dos formas: visualmente con la barra que se va rellenando al avanzar en la aplicación, y con un porcentaje, justo encima de esta, que nos traduce el progreso a un valor numérico.
2. Temporizador: dado que el objetivo final de la aplicación es ayudar a los alumnos de la asignatura “Simulación Clínica Avanzada” en la realización de sus simulaciones, he añadido un temporizador de diez minutos, que es el tiempo que tienen para determinar un diagnóstico al caso clínico propuesto y realizar la primera evaluación completa del paciente. Cuando este temporizador llega a cero se termina la simulación.
3. Letra X rodeada por un círculo: este símbolo es un botón que nos redirige a la pantalla X sobre la exanguinación. Dado que el estado de las hemorragias puede variar a lo largo de la atención médica y es necesario atajar el problema en cuanto se produzca, podemos tener la necesidad de retroceder de forma inmediata a esa pantalla en el transcurso de la simulación, y ahí reside la utilidad de este botón. Va a ser común a todas las pantallas posteriores a la X, por lo que lo encontraremos en los niveles A, B, C, D y E.
4. Puerta: este símbolo es un botón que nos permite finalizar la simulación en cualquier momento y regresar a la pantalla principal de la aplicación. Al pulsarla se abre un diálogo

en una pantalla emergente donde se nos pregunta si realmente queremos finalizar la simulación, ya que al hacerlo se perderán todas las acciones realizadas hasta el momento. Por lo tanto, aunque se pulse la puerta tenemos la opción de continuar la evaluación. Este *pop-up* se muestra en la Figura 24.

5. Letra del nivel: cada una de las cuestiones médicas que trata la aplicación lleva asociada una letra en la metodología seguida, por lo general asociada a su nombre en inglés. Estas van a ser: X, A, B, C, D y E. Con ellas, junto con un color característico asociado, conseguimos identificar de una forma rápida el contenido del que trata la pantalla.
6. Nombre del nivel: justo debajo de la letra indicativa del nivel aparece la cuestión médica que se aborda.
7. Flecha atrás: este símbolo es un botón que nos lleva inmediatamente a la pantalla anterior, para el caso de que se necesite repasar o cambiar valores en pasos anteriores, ya que el trauma y la evaluación primaria son procesos dinámicos.
8. Botón "SIGUIENTE": este botón es el que deberíamos usar con un funcionamiento habitual de la aplicación. La utilidad que tiene es que solo nos permite avanzar si se han completado todos los campos de esa pantalla. En caso de que falte algún paso por realizar aparece al pulsarle un mensaje temporal informando de esta situación. Por el contrario, cuando se completan todos los pasos de la sección, el botón se ilumina (su fondo pasa a ser blanco con las letras de "SIGUIENTE" del color de la pantalla) indicando que ya se puede avanzar en la simulación.
9. Flecha adelante: este símbolo es un botón que nos lleva inmediatamente a la pantalla siguiente sin necesidad de completar todos los campos previos, para los casos en los que se necesite abordar un sistema concreto de forma rápida.

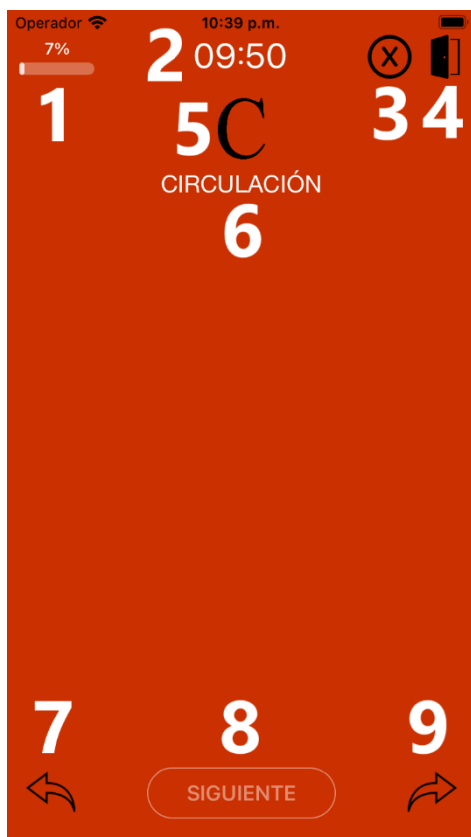


Figura 23. Elementos comunes. Fuente: propia.

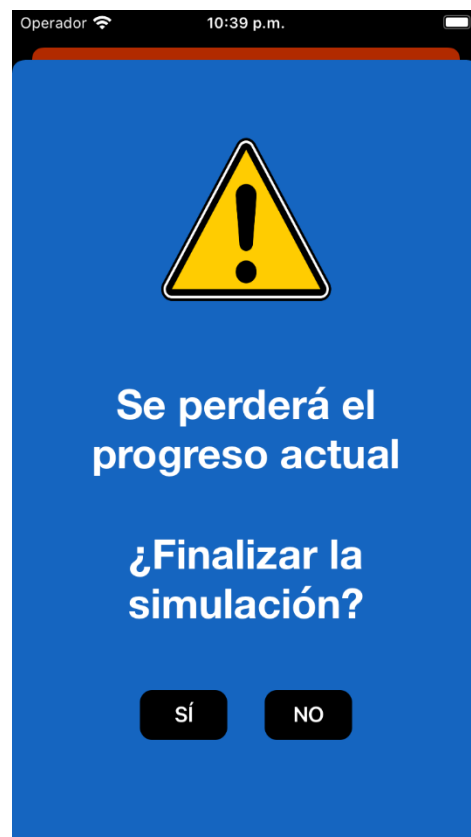


Figura 24. Pop-up de salida. Fuente: propia.

5.7 Seguridad en la escena

El primer apartado de la evaluación se corresponde con la comprobación de la seguridad de la escena. No pertenece a la metodología XABCDE, pero es un aspecto clave en cualquier emergencia prehospitalaria.

Esta pantalla es puramente informativa, recordando al personal sanitario las recomendaciones con las que atajar la asistencia médica. Las medidas, mostradas en la Figura 25, son el uso de equipo de protección individual, la valoración de los riesgos que pueda presentar la escena, priorizando en todo momento la autoseguridad, y, por último, el análisis del mecanismo lesional y la biomecánica que presente el paciente. La interfaz está acompañada de una imagen representativa con el ejemplo de un conductor de vehículo en caso de accidente.

El botón siguiente propio de esta pantalla aparece desde el primer momento iluminado, permitiendo el paso a la siguiente sección, al no haber ningún campo que deba ser completado por el usuario de la aplicación. Al accionarlo se pasa a tratar la impresión de la gravedad de la escena.



Figura 25. Pantalla dedicada a la seguridad en la escena. Fuente: propia.

A continuación, en la Figura 26, se presenta el diagrama de flujo de la pantalla. Aunque en esta sección es sencilla al tratarse de un apartado informativo, se pueden apreciar de un vistazo los pasos a seguir por el usuario, además de la siguiente sección. Por esta razón ha sido incluido en esta memoria, siguiendo el mismo esquema de contenido en cada apartado.

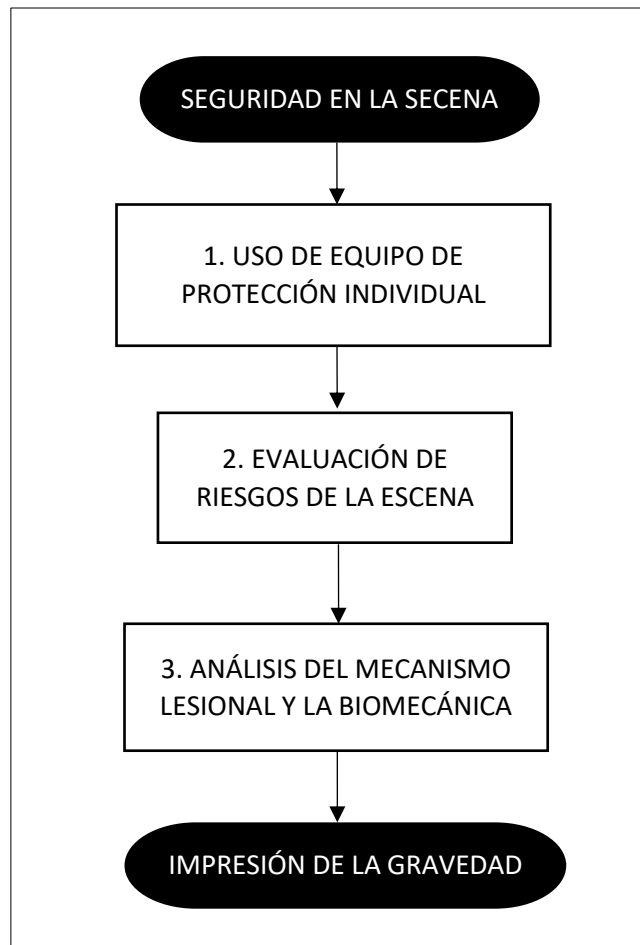


Figura 26. Diagrama de flujo de la pantalla seguridad en la escena. Fuente: propia.

5.8 Impresión de la gravedad

En la segunda sección de la evaluación se comprueba la impresión de la gravedad, realizando una valoración de la escena en su totalidad. Tampoco pertenece a la metodología XABCDE, como la anterior, pero es realmente útil antes de comenzar a tratar al paciente para tener una visión general de la situación.

De nuevo se trata de una pantalla bastante informativa, en la que se recomienda tener en cuenta una serie de datos antes de iniciar con la evaluación sistematizada. Primero se comprueba el número de heridos y se valora la necesidad de recursos adicionales. Tras esto, es recomendable obtener una información aproximada del paciente y su estado, como su edad, sexo, peso, posición y lesiones evidentes. Por último, conviene comprobar que el paciente sea capaz de responder ante órdenes sencillas, ya que con ello se descartan muchas lesiones de compromiso vital.

En este caso, el botón siguiente aparece primero sin iluminar, como se ve en la Figura 27, y no permite continuar hasta que el usuario de la aplicación responda a la pregunta de si el paciente es capaz de comunicarse. Cuando se introduzca un valor, el botón se ilumina (con el fondo negro y letras blancas, Figura 28), y se puede comenzar con la evaluación sistemática del paciente mediante la metodología XABCDE, comenzando por la exanguinación.

De nuevo, presento el diagrama de flujo seguido en esta pantalla en la Figura 29.

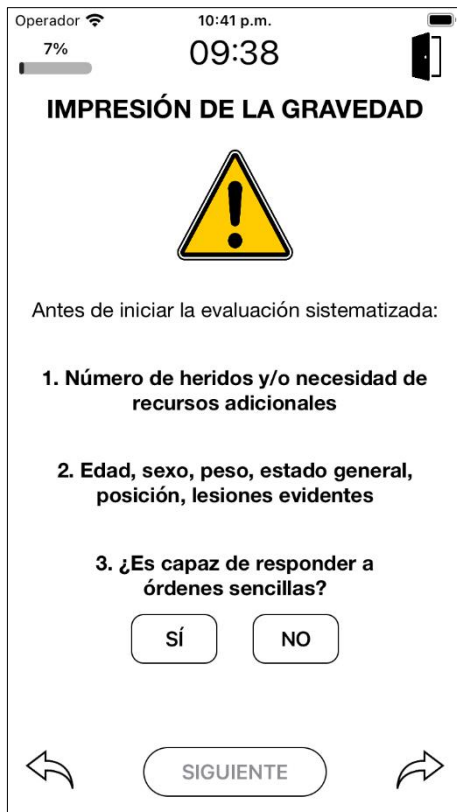


Figura 27. Pantalla dedicada a la impresión de la gravedad inicial. Fuente: propia.

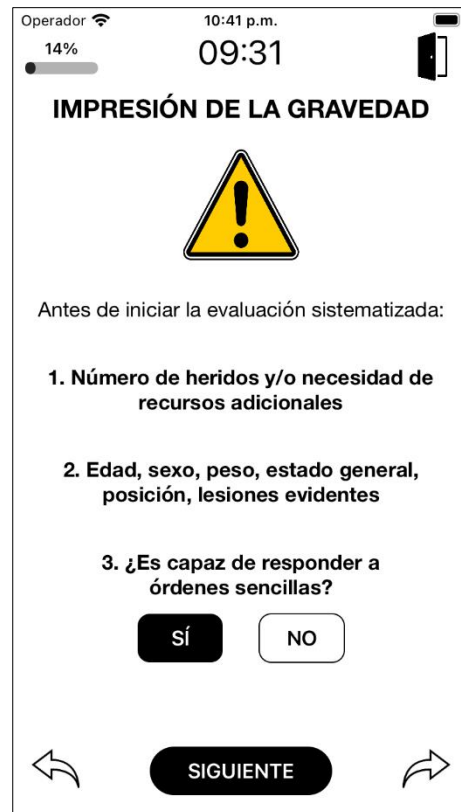


Figura 28. Pantalla dedicada a la impresión de la gravedad completada. Fuente: propia.

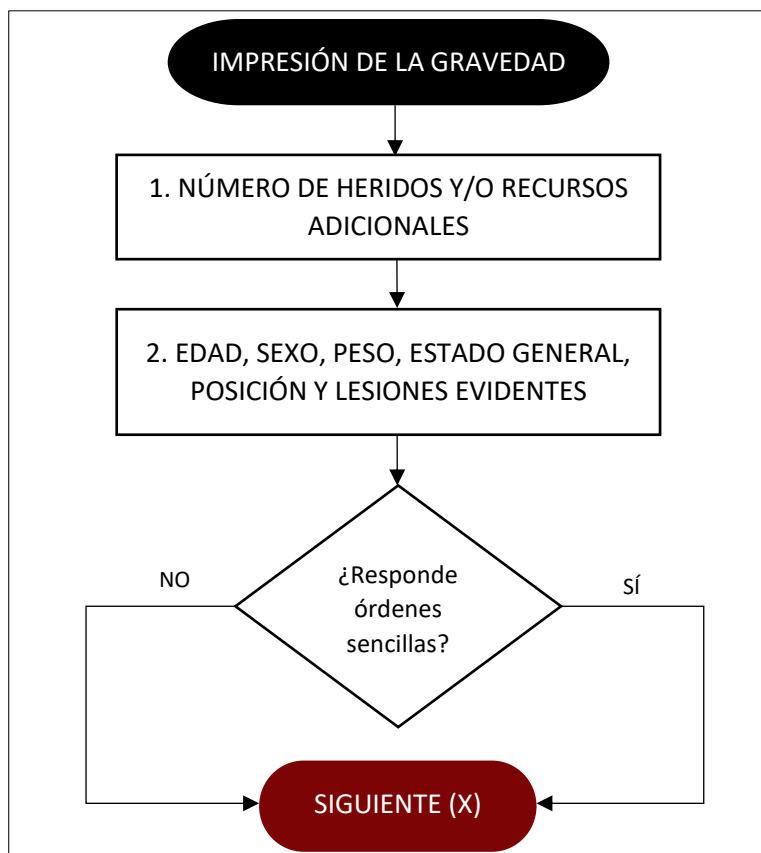


Figura 29. Diagrama de flujo de la pantalla impresión de la gravedad. Fuente: propia.

5.9 Pantalla X: Exanguinación

En esta pantalla, llamada exanguinación (o *exsanguination* en inglés), se tratan los focos activos de sangrado. Esta es una de las cuestiones más importantes que tratar de forma rápida porque una hemorragia grave es la principal causa de muerte evitable entre pacientes politraumatizados. Como color se ha escogido un rojo oscuro característico que representa bastante bien el aspecto a tratar.

Al comenzar esta sección tenemos una pregunta sobre la presencia de focos de sangrado activos, como podemos ver en la Figura 30. Si no presenta ninguno, se ilumina el botón siguiente porque ya no hay ningún paso más por hacer, y se puede avanzar en la simulación. En caso de que sí tengamos alguna hemorragia, aparecen dos *pop-ups* para identificarla. La primera sirve para localizar la zona del cuerpo en la que se encuentra pulsando sobre la imagen del paciente. Al hacerlo, la parte del cuerpo se iluminará, como se puede ver en la Figura 32 y se puede pasar al siguiente *pop-up*, el cual se muestra en la Figura 33. Este segundo consiste en determinar el tipo de hemorragia que presenta, escogiendo entre capilar, venosa o arterial, mostrando por pantalla las características más identificativas de la opción escogida.

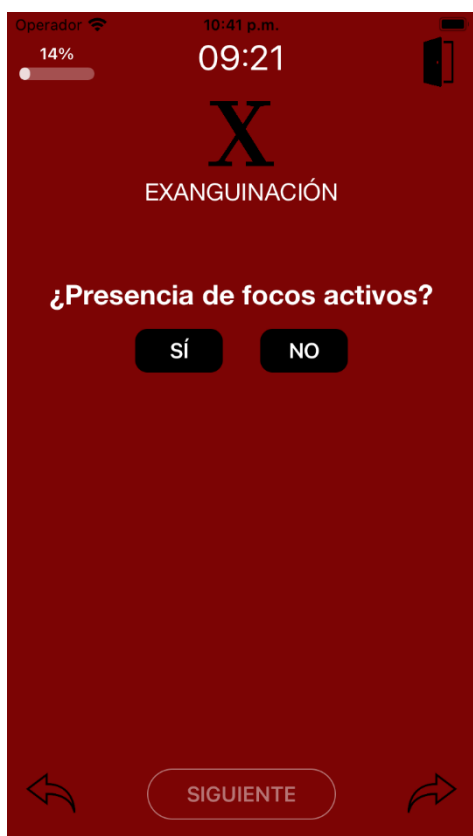


Figura 30. Pantalla X inicial. Fuente: propia.



Figura 31. Pantalla X completada. Fuente: propia.

Tras estos dos pasos, se muestra por pantalla lo que el usuario de la aplicación debería hacer, que es presión directa manual (o con vendaje compresivo en caso de las extremidades), como se ve en la Figura 31. En la parte final de la pantalla, tenemos la pregunta de si se ha conseguido controlar la hemorragia. En caso negativo, si la hemorragia se encuentra en las extremidades y solo cuando sea estrictamente necesario, se hará uso del torniquete. Esto lo vemos en el *pop-up* de la Figura 34, en el que se muestran las indicaciones para aplicar el torniquete de forma

correcta. Por último, se expone el uso de agentes hemostáticos, que se pueden utilizar para hemorragias en cualquier localización.

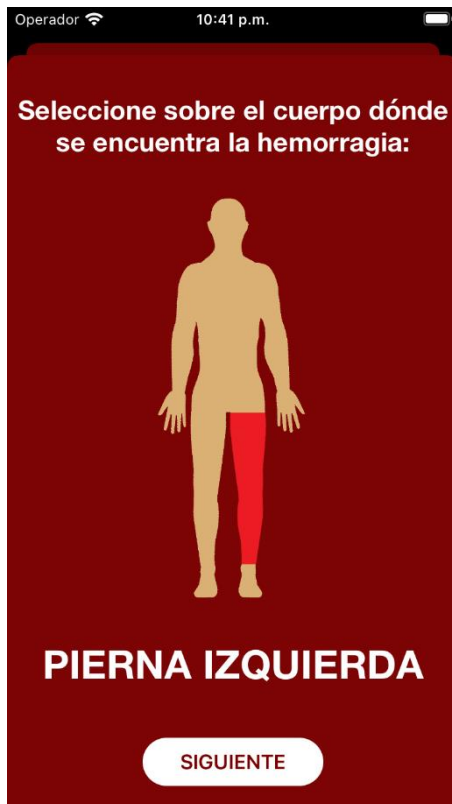


Figura 32. Pop-up localización de hemorragia. Fuente: propia.



Figura 33. Pop-up tipo de hemorragia. Fuente: propia.



Figura 34. Pop-up torniquete. Fuente: propia.

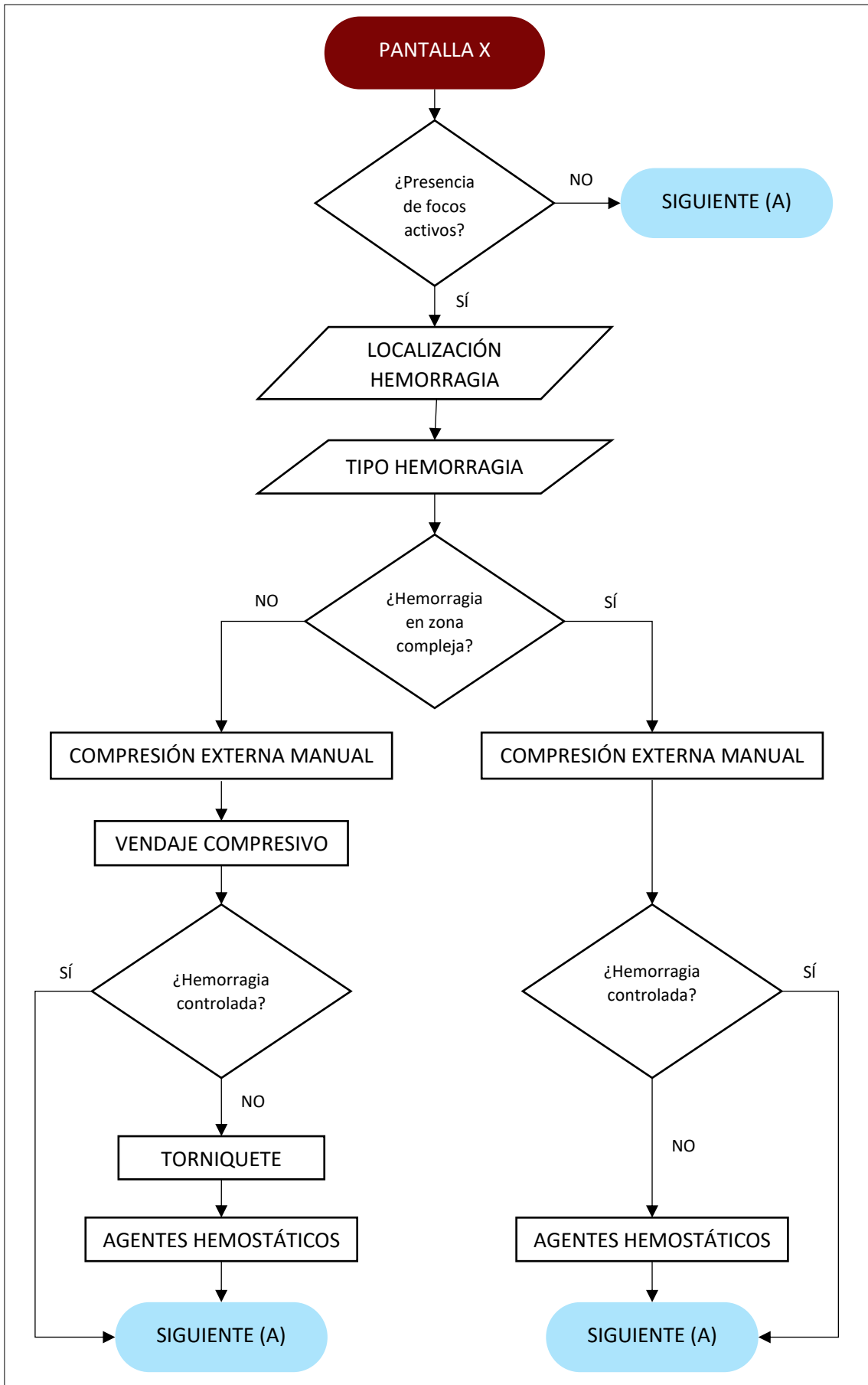


Figura 35. Diagrama de flujo pantalla X. Fuente: propia.

5.10 Pantalla A: Vía aérea

Pasamos a la pantalla A donde se trata todo lo relacionado con la vía aérea, buscando como objetivo final que permanezca abierta y permeable. La letra asignada A viene determinada por el término en inglés *airway*. Como se ve en la Figura 36, el color elegido es un tono azul claro bastante característico. Esta sección está dividida en dos pantallas para que pueda entrar toda la información en dispositivos de pequeño tamaño, como es el caso del que se han realizado las capturas.

Para comenzar, se busca saber si el paciente ha tenido algún problema de tipo traumático. En caso afirmativo se abre un *pop-up* con las indicaciones a tener en cuenta para un correcto control cervical y así evitar posibles riesgos en la columna. Esto lo vemos en la Figura 40.

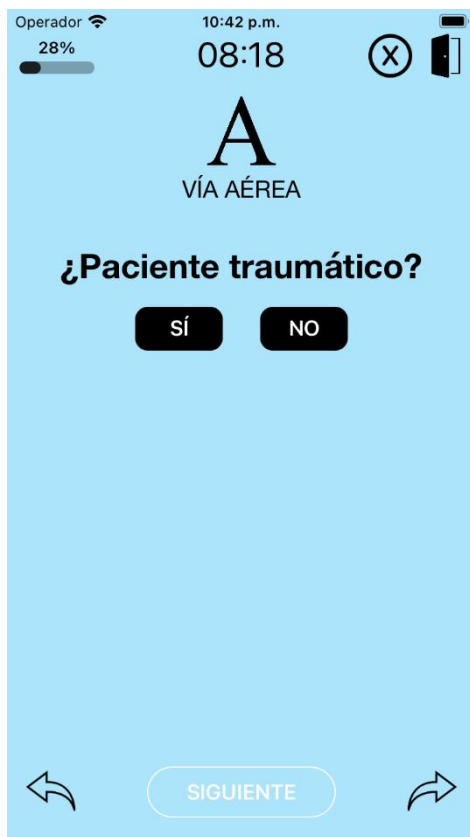


Figura 36. Pantalla A1 inicial. Fuente: propia.

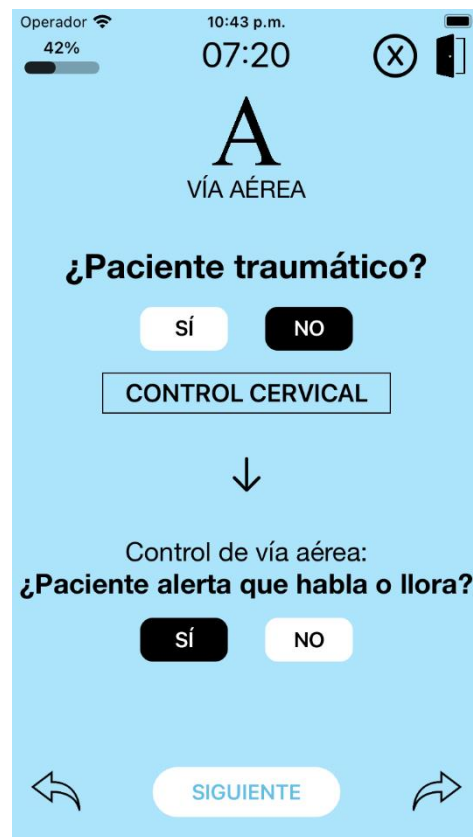


Figura 37. Pantalla A1 completada. Fuente: propia.

Tras este primer paso se pregunta si el paciente está alerta hablando o llorando, ya que es un buen indicativo de que la vía está abierta y permeable. Por lo tanto, si así se encuentra, no hay más riesgos en cuestiones referidas a la vía aérea y se puede pasar a la siguiente sección de la simulación. Por otro lado, si la respuesta es negativa la aplicación nos redirige a la segunda pantalla dedicada a este nivel, que se puede ver en la Figura 38.

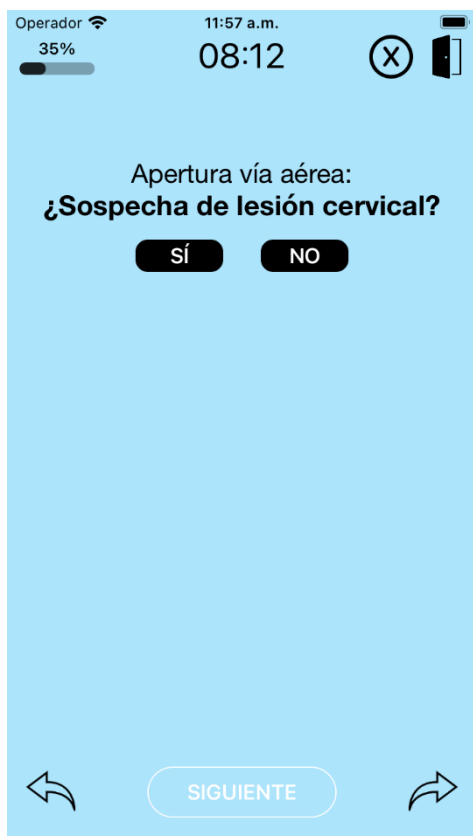


Figura 38. Pantalla A2 inicial. Fuente: propia.

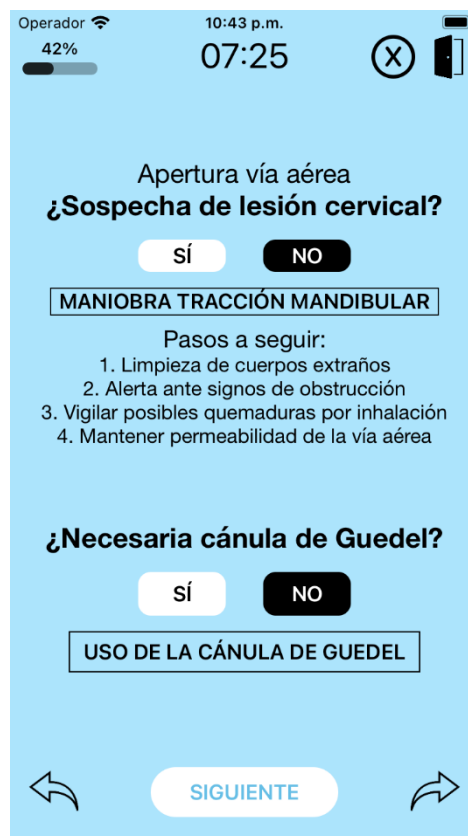


Figura 39. Pantalla A2 completada. Fuente: propia.


En esta segunda pantalla identificamos en primer lugar si hay sospecha de lesión cervical. Con ello se determina el tipo de maniobra que hay que llevar a cabo para abrir la vía aérea. Si el paciente presenta lesión, se procederá a realizar la maniobra de tracción mandibular que se puede ver en el *pop-up* de la Figura 42. En caso contrario se realiza la maniobra frente-mentón de la Figura 43.

Tras aplicar la maniobra correspondiente se exponen una serie de pasos, comunes en ambas situaciones, para controlar otros posibles problemas de la vía aérea. Al final se consulta si es necesaria la cánula de Guedel. Esta será la última opción para obtener una vía aérea abierta y permeable, para los casos en que no se haya conseguido previamente. Si se necesita su uso aparece el *pop-up* de la Figura 41 en el que se explica su correcta colocación.

Para finalizar con lo relativo a la vía aérea, se muestra el diagrama de flujo seguido en este apartado en la Figura 44.

Operador 11:56 a.m.

CONTROL CERVICAL



1. Tracción longitudinal de la cabeza.
2. Alineación cuello-cabeza-tronco.
3. Colocación de collarín cervical.


LISTO

Figura 40. Pop-up control cervical. Fuente: propia.


Operador 10:43 p.m.

CÁNULA DE GUEDEL

1. Elegir tamaño de la cánula midiendo desde la comisura labial hasta el trago de la misma oreja.
2. Insertar cánula con la concavidad hacia el paladar duro.



3. Girar 180° al llegar al paladar blando.
4. Ver colocación adecuada.



LISTO

Figura 41. Pop-up cánula de Guedel. Fuente: propia.

Operador 10:42 p.m.

MANIOBRA TRACCIÓN MANDIBULAR



LISTO

Figura 42. Pop-up maniobra de tracción mandibular. Fuente: propia.

Operador 10:42 p.m.

MANIOBRA FRENTE-MENTÓN



LISTO

Figura 43. Pop-up maniobra frente-mentón. Fuente: propia.

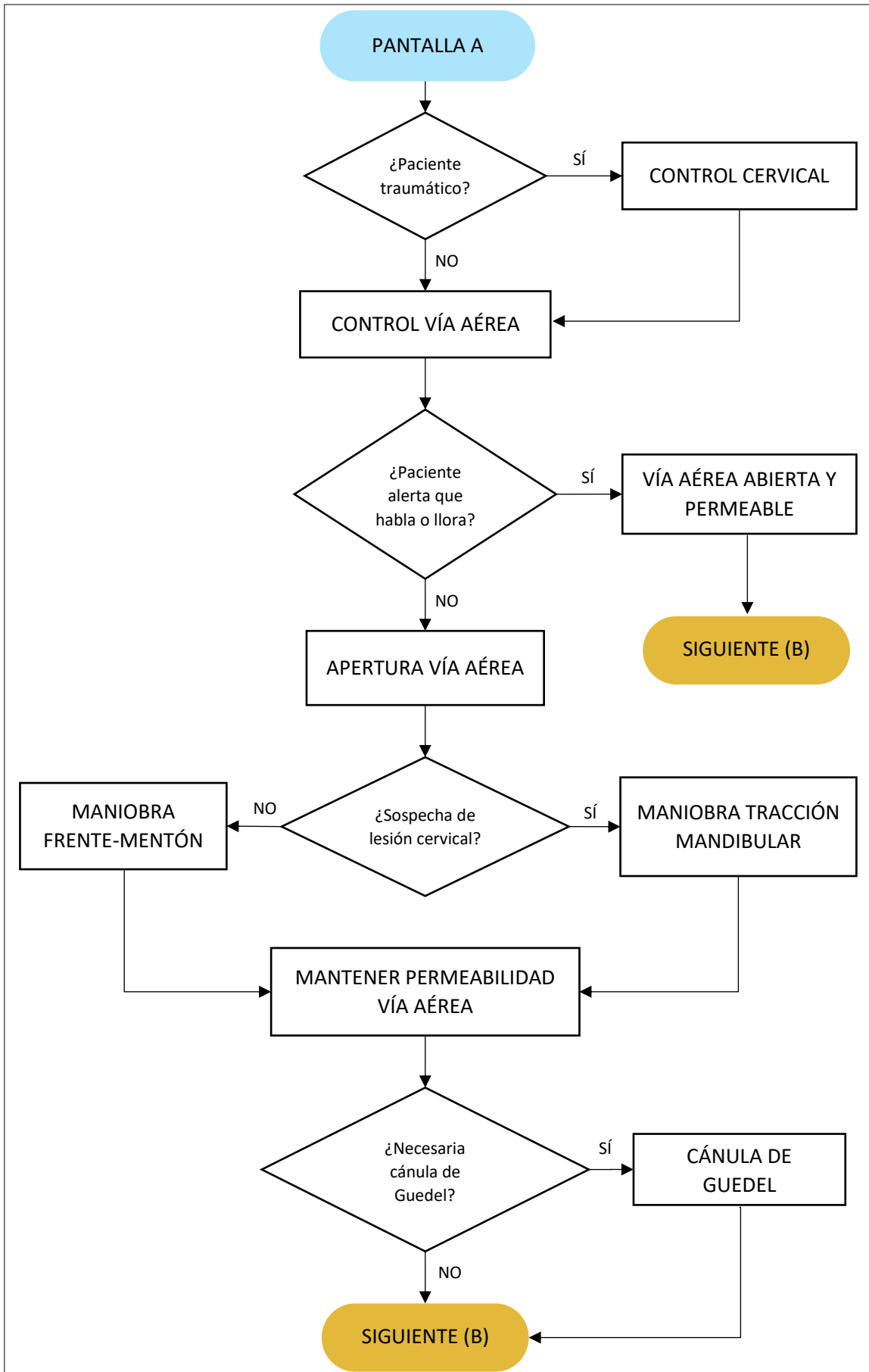


Figura 44. Diagrama de flujo pantalla A. Fuente: propia.

5.11 Pantalla B: Respiración

En este apartado se valora la calidad de la respiración del paciente y se establece el tipo de mascarilla o ventilación que haya que aplicar en función de los datos introducidos por el usuario de la aplicación. La letra asignada a esta pantalla es la B, ya que viene del término en inglés *breathing*. De nuevo, esta sección está compuesta de dos pantallas para poder mostrar toda la información necesaria en dispositivos de pequeño tamaño. A diferencia del apartado anterior en que la aparición de la segunda pantalla era opcional (solo se abría si la vía aérea no estaba abierta y permeable), en este caso siempre la usaremos ya que contiene un procedimiento indispensable que es el de suministrar oxígeno.

Para comenzar con la valoración de la función respiratoria se solicita realizar cuatro pasos, que son: inspección, auscultación, palpación y percusión. Según el usuario de la aplicación realice cada una de estas acciones marcará la casilla correspondiente y aparecerá en pantalla la siguiente. Esto se puede ver en la Figura 45 en el que solo está visible el primer paso (inspección), frente a la Figura 46 en la que se muestran todos los pasos.

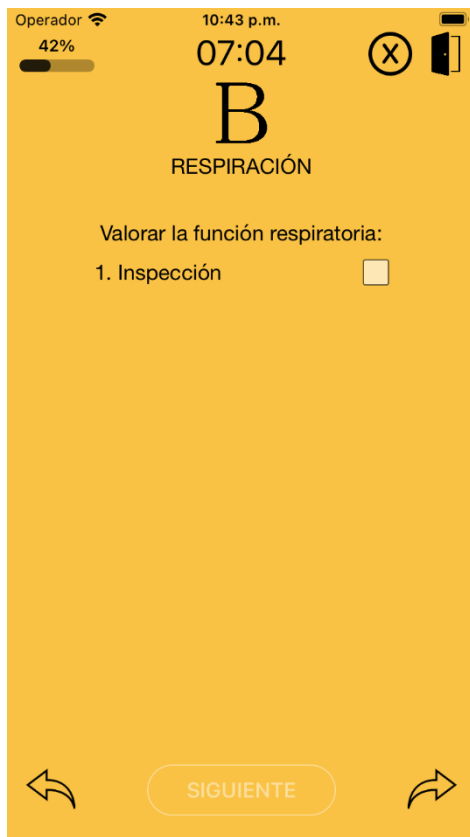


Figura 45. Pantalla B inicial. Fuente: propia.

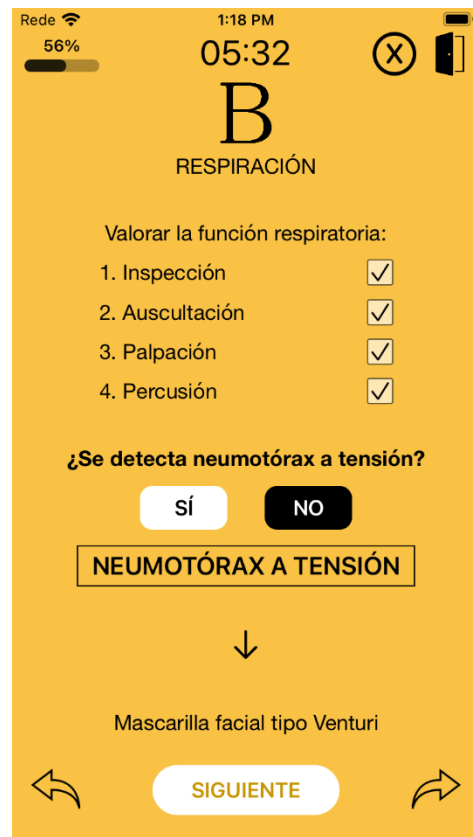


Figura 46. Pantalla B completada. Fuente: propia.

Tras esto se consulta si se detecta neumotórax a tensión. Este es un problema bastante grave que hay que solucionar de inmediato en caso de encontrarse. Sin embargo, hay que asegurarse de que se produce, y necesita ser tratado de manera correcta, por eso hay un pop-up dedicado, el cual vemos en la Figura 47. En ella se mencionan los síntomas para identificar el neumotórax y el proceso a seguir para solventarlo, acompañado de una imagen ilustrativa.

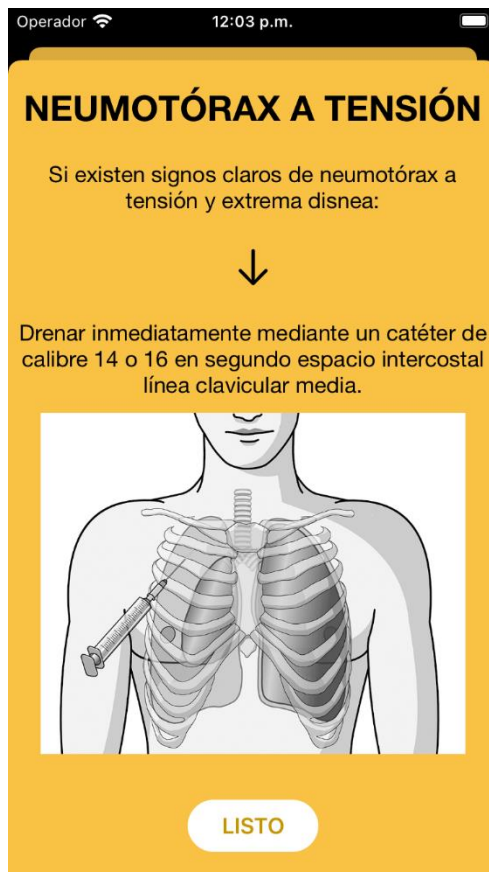


Figura 47. Pop-up neumotórax a tensión. Fuente: propia.

Después pasamos a la pantalla para suministrar oxígeno. En ella se comienza preguntando si el paciente está consciente y respirando, ya que de lo contrario habría que practicarle una intubación orotraqueal con control de la columna cervical. Esto se indicaría en pantalla acompañado de la imagen de un laringoscopio (instrumento médico necesario para realizar la intubación).

Por el contrario, si el paciente sí que está consciente y respirando se pasa a medir la calidad de la respiración. Esto se lleva a cabo con dos preguntas, cada una implementada en un *pop-up*. La primera dedicada a la frecuencia respiratoria como vemos en la Figura 50, en la que tenemos cuatro opciones en función del número de respiraciones por minuto, y la segunda dedicada a la saturación de oxígeno, mostrada en la Figura 51, con tres opciones dependiendo del porcentaje que presenta el paciente. Al introducir estos dos campos se muestra por pantalla el tipo de ventilación que debemos aplicar con su correspondiente imagen. Con esto ya habremos completado el suministro de oxígeno, retrocedemos a la pantalla B donde se ve de nuevo el tipo de ventilación empleado (en el ejemplo de la Figura 46 mascarilla facial tipo Venturi) y se puede avanzar a la siguiente sección.

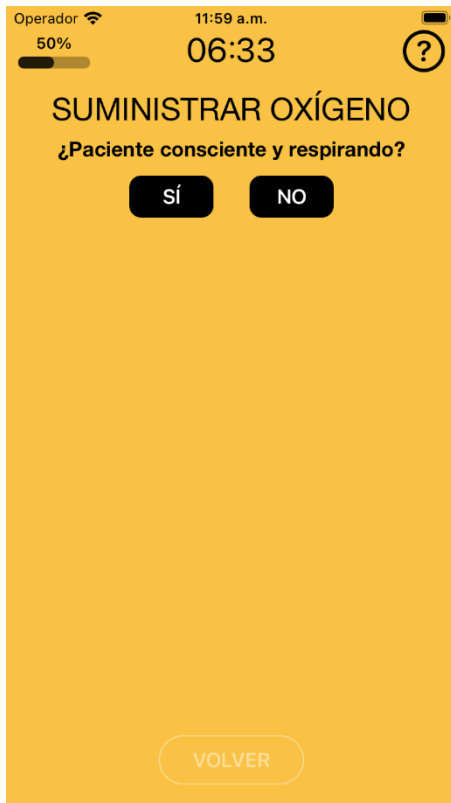


Figura 48. Pantalla oxígeno inicial. Fuente: propia.



Figura 49. Pantalla oxígeno completada. Fuente: propia.

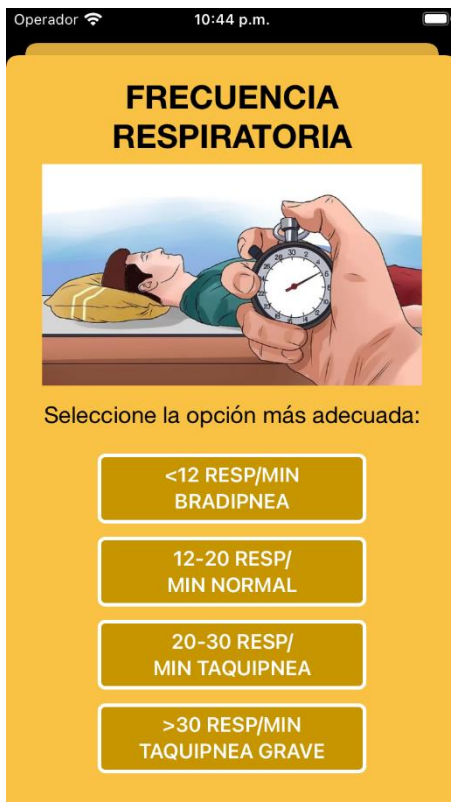


Figura 50. Pop-up frecuencia respiratoria. Fuente: propia.



Figura 51. Pop-up saturación de oxígeno. Fuente: propia.

Cabe destacar el símbolo de interrogación presente en la esquina superior derecha de la pantalla “Suministrar oxígeno”, el cual puede verse en la captura de la Figura 48. Este botón abre otro *pop-up* con una tabla en la que se informa de los valores de flujo y concentración de oxígeno adecuados para los distintos dispositivos de oxigenoterapia, presentado en la Figura 52.

	FiO2 (%)	Flujo O2 (l/min)
Sistemas de alto flujo		
Máscara Venturi	25	3
	26	4
	28	5
	30	7
	35	10
	40	12
	50	15
Sistemas de bajo flujo		
Cánula nasal	24	1
	28	2
	32	3
	36	4
	40	5
	44	6
Tienda facial	40	5-6
Mascara de O2	40	5-6
	50	6-7
	60	7-8
Máscara reservorio	100	10-15

Figura 52. Pop-up tabla de concentración de O2 para dispositivos de oxigenoterapia. Fuente: propia.

Para acabar, al igual que en el resto de los subapartados, muestro el diagrama de flujo dedicado a la pantalla B en la Figura 53.

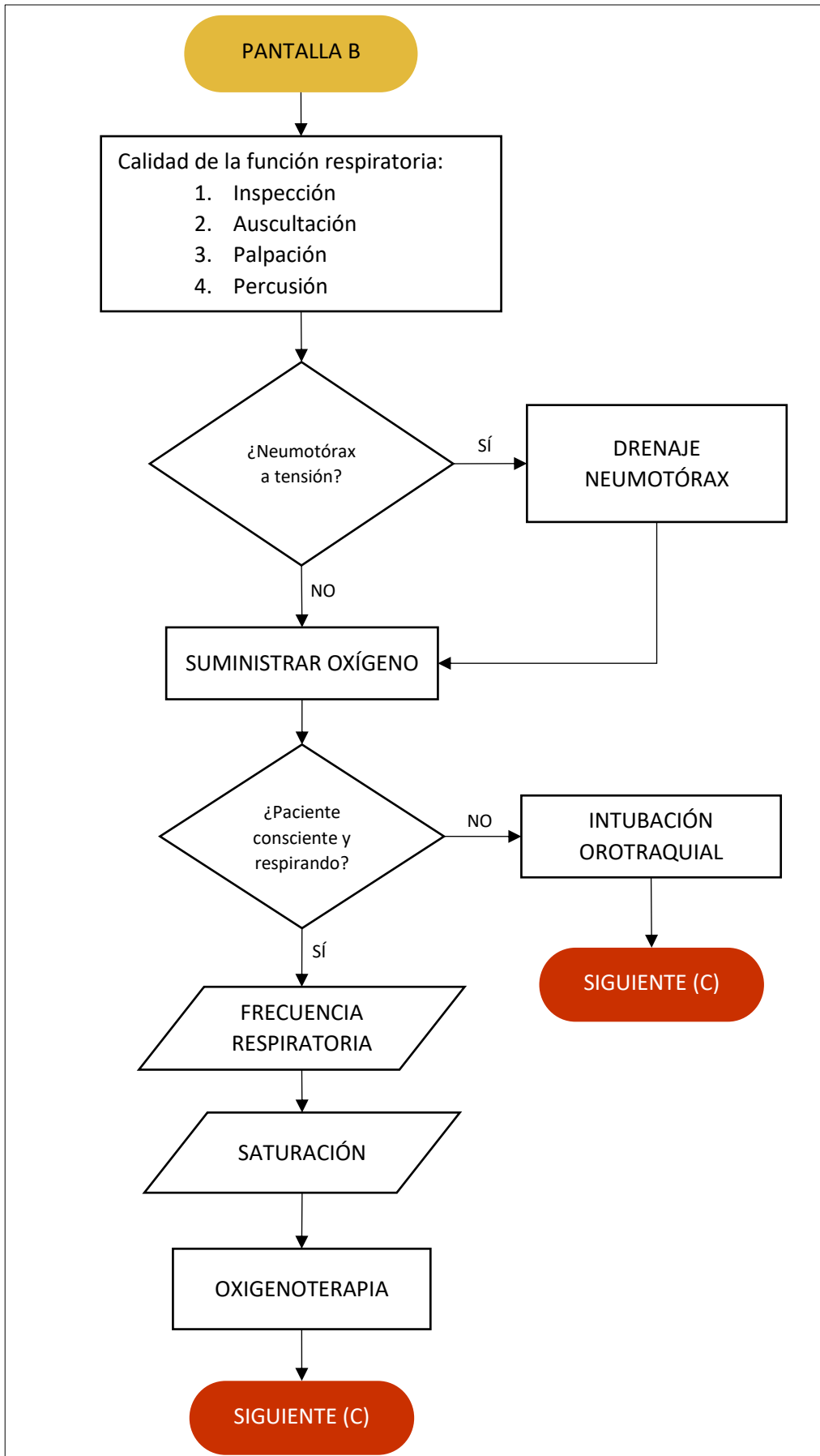


Figura 53. Diagrama de flujo pantalla B. Fuente: propia.

5.12 Pantalla C: Circulación

La pantalla C está dedicada a comprobar la calidad de la circulación sanguínea que presenta el paciente. Para realizar esta tarea se van a ir desarrollando un conjunto de pasos de manera secuencial para evaluar cada uno de los indicadores de la función sanguínea, así como la posible presencia de shock hemorrágico. El color escogido en este caso es un rojo vivo característico. Para comenzar, se pregunta si es necesario reponer el volumen sanguíneo. En caso de que así sea, se abre el *pop-up* de la Figura 56 en el que se indican los pasos a seguir.

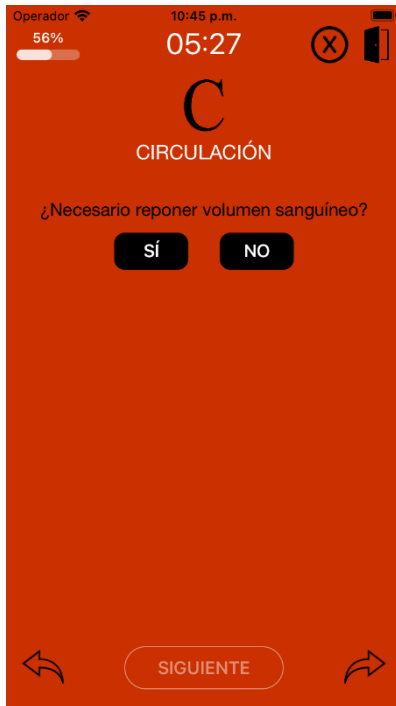


Figura 54. Pantalla C inicial. Fuente: propia.

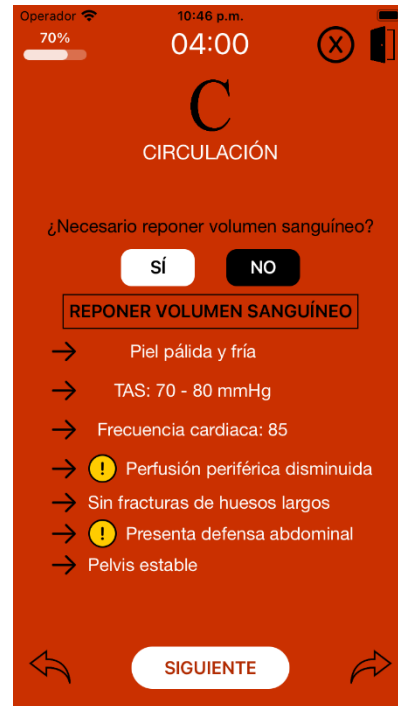


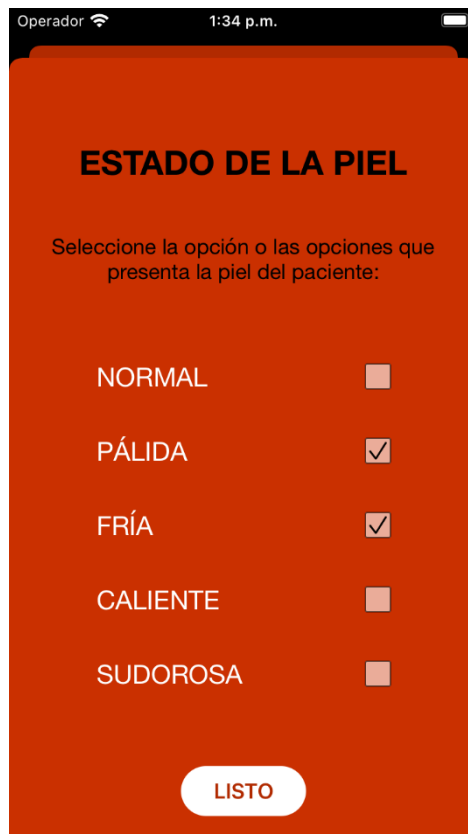
Figura 55. Pantalla C completada. Fuente: propia.



Figura 56. Pop-up reponer volumen sanguíneo. Fuente: propia.

Tras esto tratamos un conjunto de indicativos de la calidad de la circulación sanguínea que aparecen por pantalla según se introduce el valor anterior. Todos ellos se muestran en la interfaz con una flecha para destacar cada cuestión, como podemos ver en la Figura 54. Estos pasos son:

- Estado de la piel: se evalúa mediante un *pop-up* (Figura 57) en el que tenemos una serie de opciones: normal, pálida, fría, caliente y sudorosa. La selección se realiza mediante casillas o *check boxes* que se marcan al pulsarlas. El máximo de selecciones permitidas es de tres, ya que con las opciones dadas no tendría sentido marcar más al haber términos contrapuestos.



Operador 1:34 p.m.

ESTADO DE LA PIEL

Seleccione la opción o las opciones que presenta la piel del paciente:

NORMAL	<input type="checkbox"/>
PÁLIDA	<input checked="" type="checkbox"/>
FRÍA	<input checked="" type="checkbox"/>
CALIENTE	<input type="checkbox"/>
SUDOROSA	<input type="checkbox"/>

LISTO

Figura 57. Pop-up estado de la piel. Fuente: propia.

- Tensión arterial sistólica (TAS): se comprueba en un *pop-up* dedicado realizando la llamada “regla de los pulsos”. Esta se emplea para obtener una aproximación bastante acorde a la realidad comprobando los tipos de pulsos que se detectan: carotídeo, femoral y/o radial. Al seleccionar las opciones adecuadas la aplicación da un valor del TAS en unidades de milímetros de mercurio (mmHg). Se puede ver esta implementación en la captura de la Figura 58.

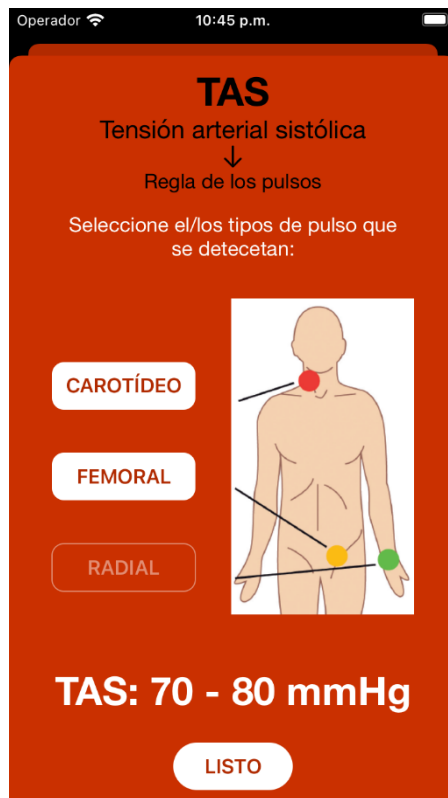


Figura 58. Pop-up tensión arterial sistólica. Fuente: propia.

- Frecuencia cardiaca: en este campo se introduce el valor de la frecuencia cardiaca del paciente con el fin de tener un control de la misma y guardar el valor en la aplicación. La forma en la que se ha implementado se ve en la Figura 59.



Figura 59. Pop-up frecuencia cardiaca. Fuente: propia.

- **Perfusión periférica:** para comprobar su correcto estado se utiliza una técnica de evaluación del relleno capilar. Para ello se aprieta el lecho ungueal y se determina el tiempo en que este recupera su color original. Si se consigue en menos de dos segundos tenemos perfusión periférica normal. Por el contrario, si tarda más de dos segundos en recuperar el color, estaremos ante un estado de perfusión periférica disminuida, como es el caso del ejemplo mostrado en la Figura 60. En ella se presenta el *pop-up* en el que se dan las dos opciones comentadas, mayor o menor de dos segundos.



Figura 60. Pop-up relleno capilar. Fuente: propia.

- **Fracturas de huesos largos:** se realiza una pregunta de si hay fracturas en algún hueso de los llamados “largos”. Su presencia puede provocar otros problemas médicos por lo que conviene tenerlo presente en esta sección.
- **Defensa abdominal:** este es un síntoma de algún problema mayor por lo que también hay que tenerlo en cuenta. Se comprobará manualmente mediante una técnica de palpación.
- **Pelvis inestable:** al igual que la defensa abdominal, la presencia de pelvis inestable puede ser un síntoma grave por lo que hay que estar pendientes en caso de que se detecte.

El último aspecto a tener en cuenta es la presencia de shock hipovolémico. Para detectarlo, hacemos uso de los cuatro últimos indicativos recogidos en la aplicación: perfusión periférica disminuida, fracturas de huesos largos, presencia de defensa abdominal e inestabilidad de pelvis. Con que se detecte solo uno de estos cuatro factores se abrirá un pop-up resaltando el campo o los campos que lo han activado (acompañados de una señal de alerta), y un mensaje indicando que se considere el shock hipovolémico. Un ejemplo de esta situación en la que se presenta perfusión periférica disminuida y defensa abdominal se presenta en la Figura 61.



Figura 61. Pop-up shock hipovolémico. Fuente: propia.

Tras completar todos estos campos, y cerrar si ha sido necesario el pop-up dedicado al shock hipovolémico, se ilumina el botón “Siguiente”, indicativo de que hemos terminado todos los aspectos relacionados con la circulación sanguínea del paciente y podemos avanzar a la siguiente sección.

El diagrama de flujo de este apartado se muestra en la siguiente página en la Figura 62.

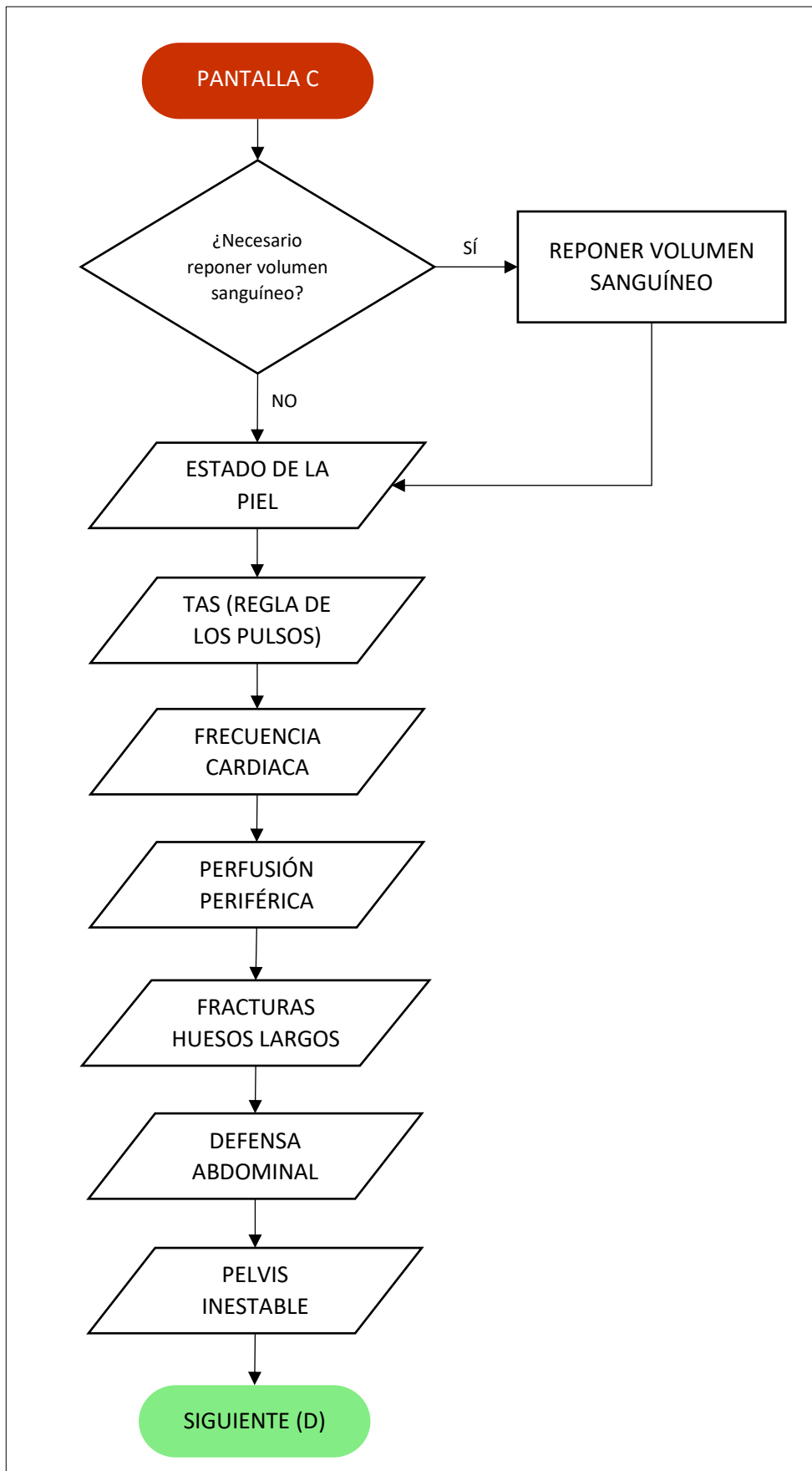


Figura 62. Diagrama de flujo pantalla C. Fuente: propia.

5.13 Pantalla D: Deterioro

La pantalla D está destinada al deterioro (en inglés disability) que pueda presentar el paciente, sobre todo al control de un posible déficit neurológico.

El primer paso de este apartado es realizar la llamada “Escala de Coma de Glasgow”. Con este sistema se mide el nivel de conciencia que presenta el paciente. Está implementada en otra pantalla dedicada solo a este procedimiento, la cual vemos en la Figura 65. Se compone de tres aspectos que se deben tener en cuenta: apertura ocular, respuesta verbal y respuesta motora. Cada uno de ellos tiene asociado un *pop-up*, los cuales vemos en la Figura 67, Figura 68 y Figura 69 respectivamente. En cada uno tenemos una imagen representativa del parámetro que estamos comprobando y todas las opciones disponibles. Tras pulsar el usuario de la aplicación uno de los estados, se actualiza el valor individual y total del procedimiento. Cuando se selecciona una opción para cada aspecto, se aporta información del estado del paciente en función del valor numérico obtenido (en una escala que va de 3 a 15). Un ejemplo de esta implementación se puede ver en la Figura 66.

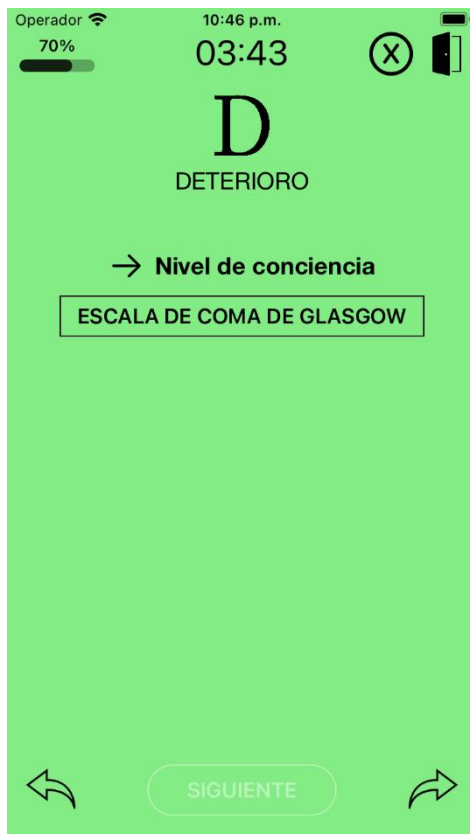


Figura 63. Pantalla D inicial. Fuente: propia.

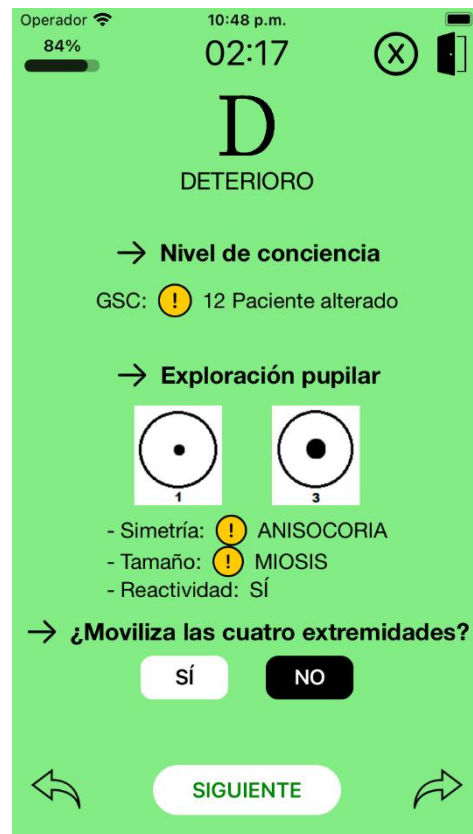


Figura 64. Pantalla D completada. Fuente: propia.

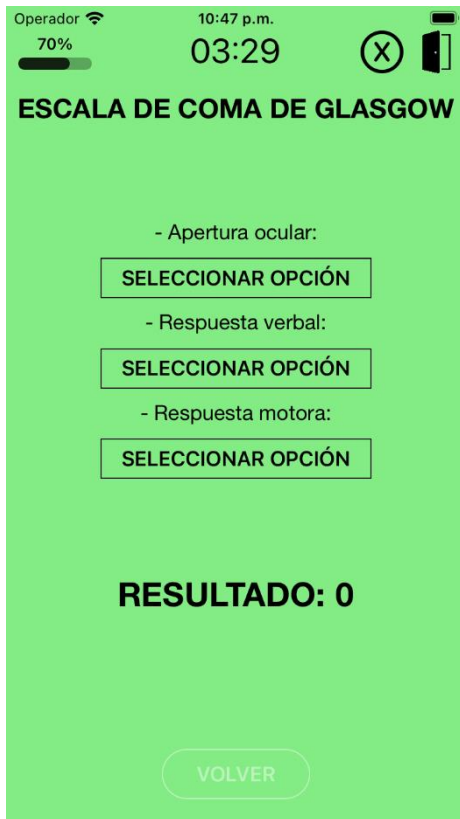


Figura 65. Pantalla Escala de Coma de Glasgow inicial. Fuente: propia.

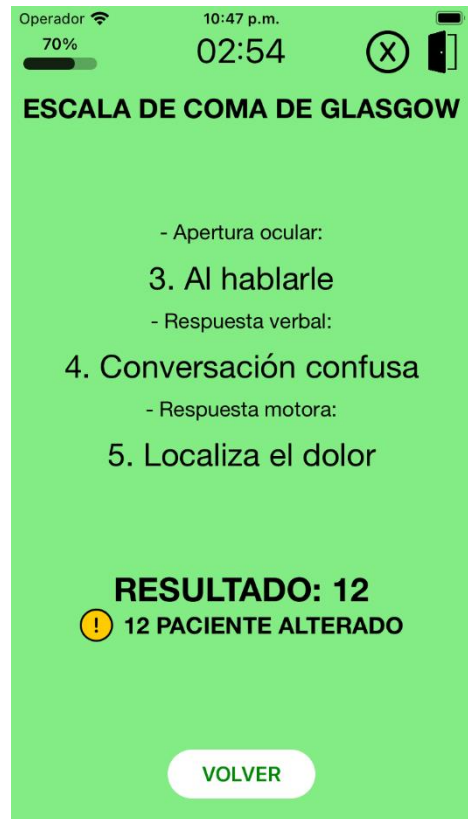


Figura 66. Pantalla Escala de Coma de Glasgow completada. Fuente: propia.



Figura 67. Pop-up apertura ocular. Fuente: propia.

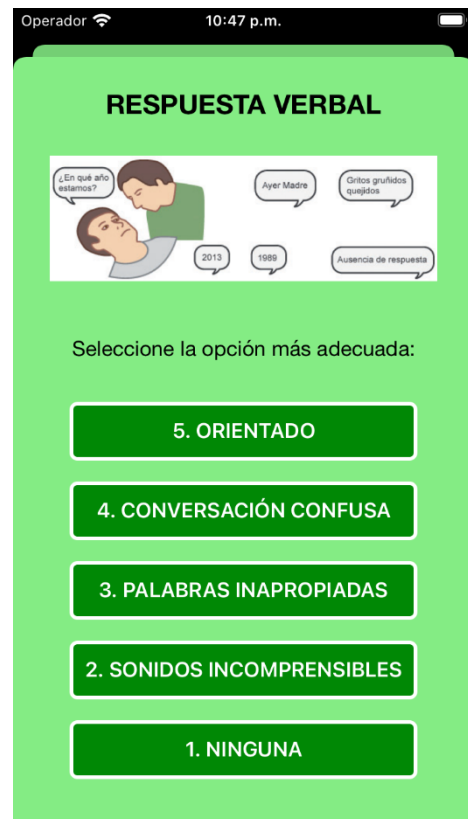


Figura 68. Pop-up respuesta verbal. Fuente: propia.



Figura 69. Pop-up respuesta motora. Fuente: propia.

Tras la Escala de Coma de Glasgow se realiza una exploración pupilar, implementada mediante dos *pop-ups* (Figura 70 y Figura 71), encadenadas una detrás de otra, en las que se pide al usuario de la aplicación que introduzca la medida del diámetro de cada pupila en milímetros. Para ello se ofrecen unas imágenes con tamaños de pupila de 1 a 8 milímetros, sobre las que el usuario tendrá que pulsar en la opción adecuada. Con estos dos valores, se aporta información sobre la simetría que presentan las pupilas (isocoria o anisocoria), su tamaño (miosis o midriasis), y se pregunta si las pupilas reaccionan a la luz. Se ve un ejemplo de su implementación en la Figura 64.

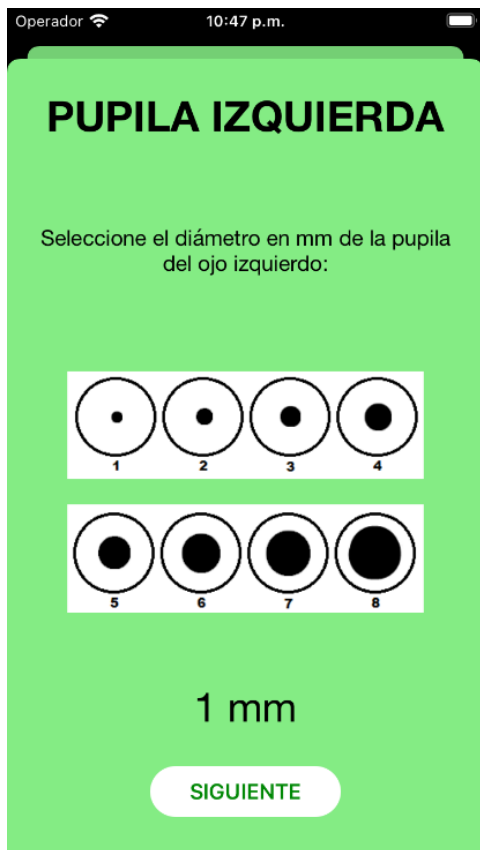


Figura 70. Pop-up pupila izquierda. Fuente: propia.

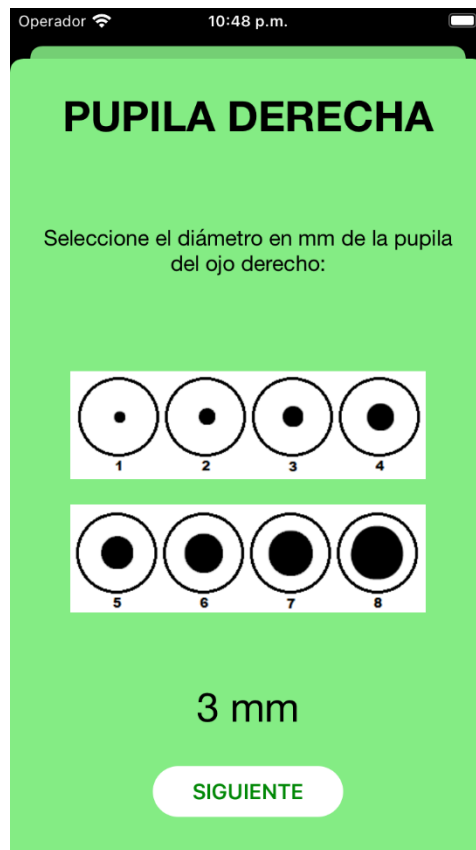


Figura 71. Pop-up pupila derecha. Fuente: propia.

Para acabar con este apartado hay una cuestión para que el usuario de la aplicación se cerciore de que el paciente es capaz de movilizar las cuatro extremidades, ya que en caso contrario podríamos estar ante algún tipo de daño neurológico. Esta pregunta se puede ver en la Figura 64.

Y por último, al igual que en el resto de las secciones muestro el diagrama de flujo, Figura 72, en este caso basado en la introducción de valores por parte del usuario de la aplicación.

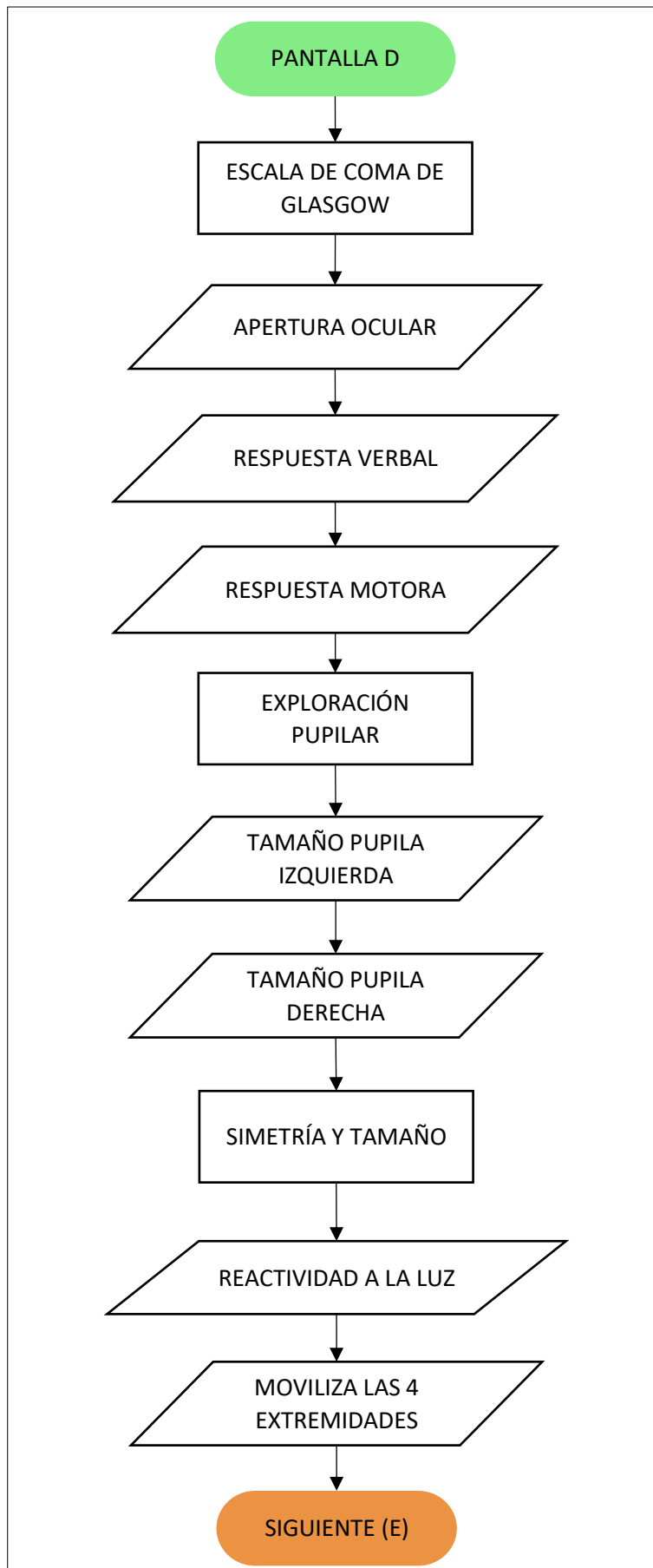


Figura 72. Diagrama de flujo pantalla D. Fuente: propia.

5.14 Pantalla E: Exposición

La última pantalla dedicada a la evaluación primaria tiene la letra E que se corresponde con la exposición (del inglés *exposure*). En concreto se realizan una serie de actuaciones sobre el paciente con el objetivo de detectar, prevenir y evitar posibles problemas que no se hayan abordado en las secciones anteriores. Hay que ir completando cada campo para que salga el siguiente. Esto se realiza mediante unas casillas que acompañan a cada procedimiento que tengamos que realizar, como se aprecia en la Figura 74.

Para empezar, se establece un examen del paciente, exponiendo por completo su cuerpo para detectar posibles problemas que estén ocultos a la vista. Después tenemos un campo para realizar una correcta estabilización.

Otro aspecto importante a tener en cuenta es el control de la temperatura del paciente, ya que es un síntoma de diversos problemas médicos, y así evitar la hipotermia. Tras esto se realiza una monitorización para tener la mayor información posible sobre su estado.

Para acabar con este apartado, hay que valorar si se precisa de analgesia, lo cual es importante ya que reduciendo el dolor se mejora la situación general del paciente.

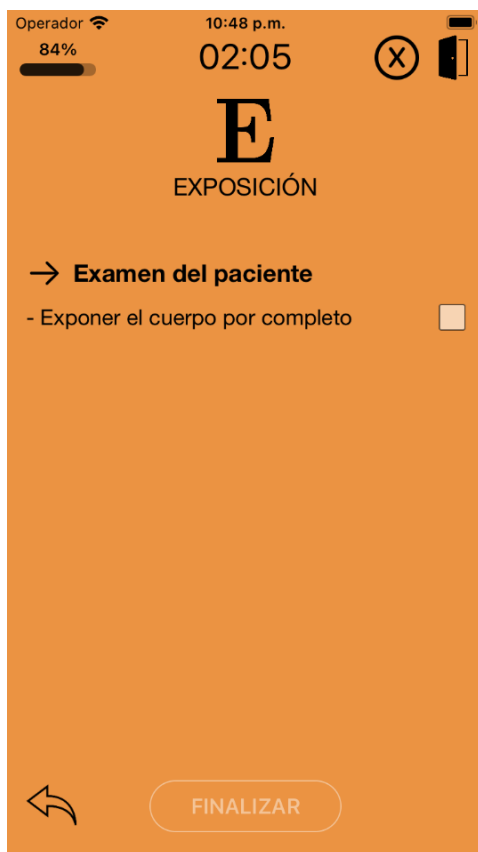


Figura 73. Pantalla E inicial. Fuente: propia.

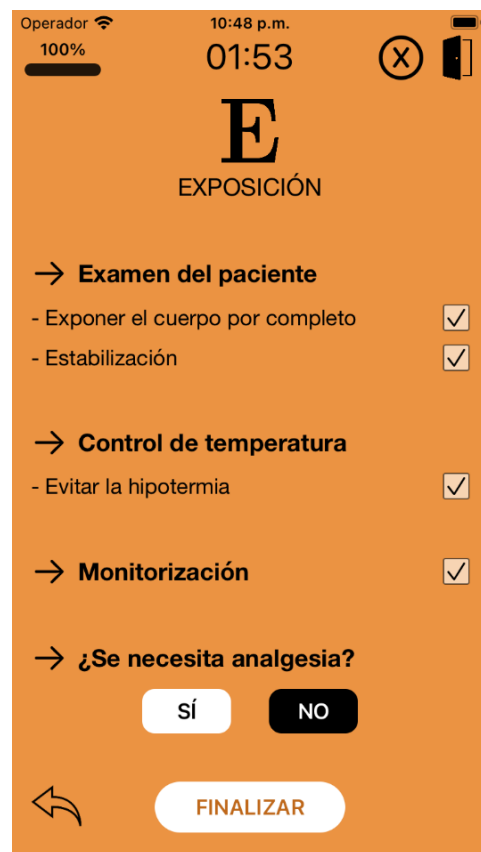


Figura 74. Pantalla E completada. Fuente: propia.

Por último, cabe destacar que el botón finalizar solo se iluminará en caso de que se haya completado el 100% de la simulación, es decir, que se hayan realizado todos los pasos tanto de la pantalla E como de las anteriores. A pesar de esto, siempre vamos a poder finalizar la simulación. Para ello al pulsar el botón finalizar se abre un diálogo en un *pop-up* informando que el usuario quiere terminar la simulación sin haber completado todos los pasos. Esto se puede ver en la Figura 75.



Figura 75. Pop-up para finalizar la simulación. Fuente: propia.

El diagrama de flujo correspondiente a esta sección se presenta en la Figura 76.

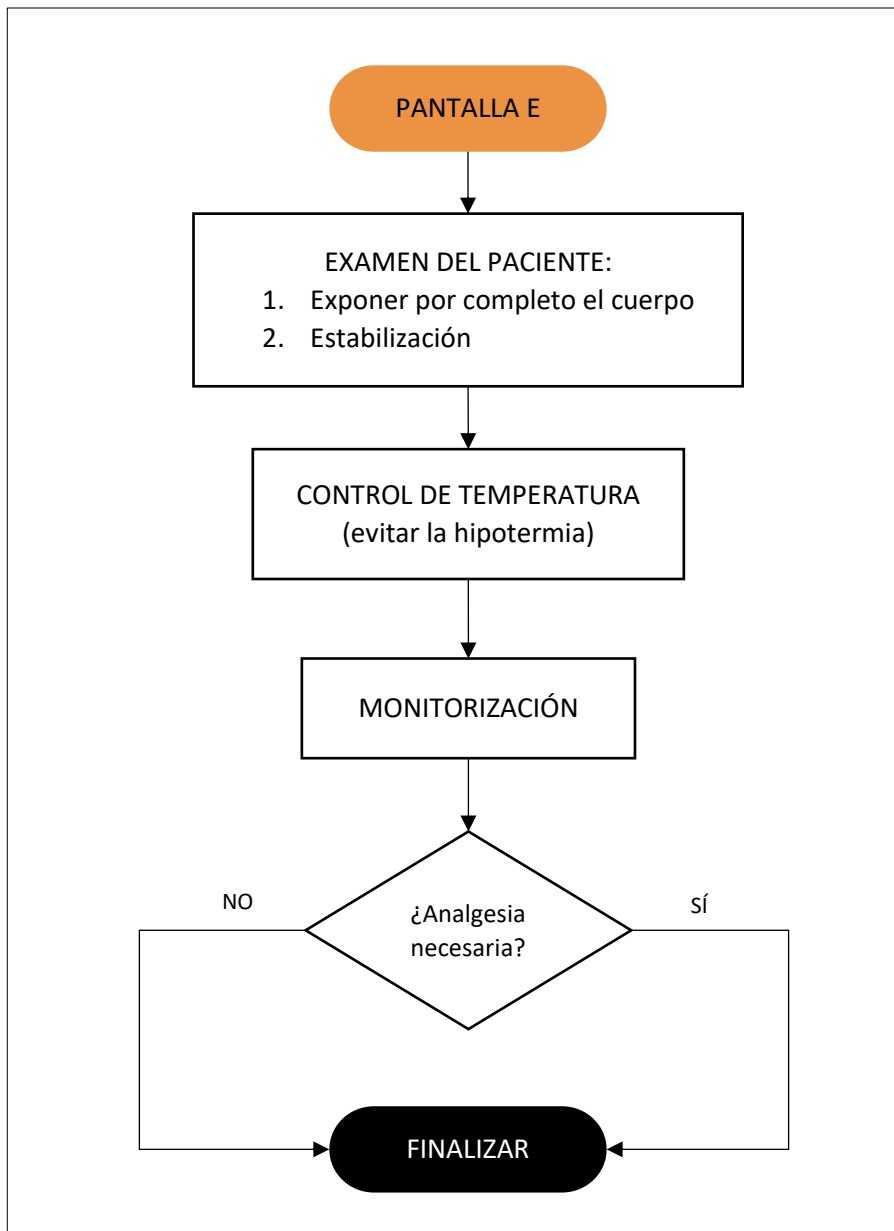


Figura 76. Diagrama de flujo pantalla E. Fuente: propia.

5.15 Pantalla tiempo finalizado

Si en el transcurso de la simulación se termina el tiempo dado, que es de diez minutos, esta finaliza y la aplicación nos redirige a una pantalla en la que se informa de esta situación. Tal y como se ve en la Figura 77, esta viene acompañada de un *gif* animado de un reloj con el mensaje “Time’s up”. Se ha dispuesto con la finalidad hacer más visual el hecho de que se ha acabado el tiempo y aportar dinamismo a la aplicación. Al pulsar el botón “Siguiente” se pasa a la pantalla final que se presenta en el siguiente subapartado.

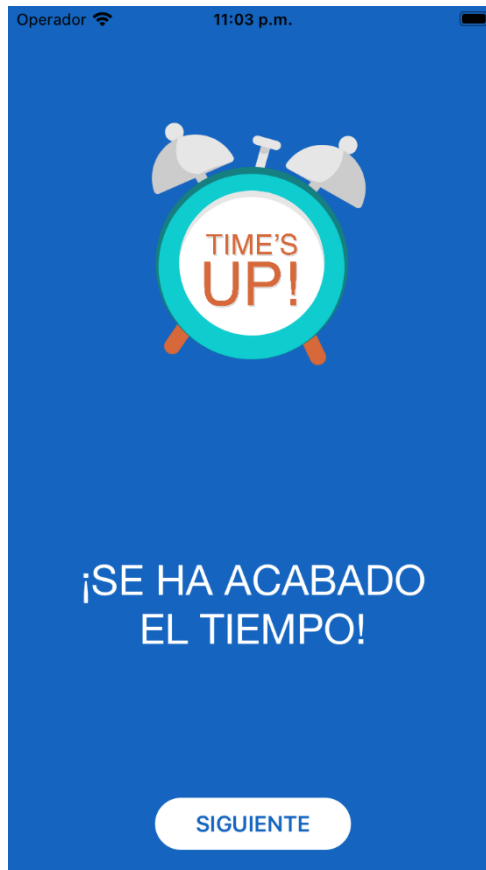


Figura 77. Pantalla tiempo finalizado. Fuente: propia.

5.16 Pantalla final

Esta pantalla aparece cuando se finaliza una simulación, hecho que se puede dar de dos formas:

- Si se acaba el temporizador de diez minutos: en ese caso se llegaría desde la pantalla de tiempo finalizado visto en el subapartado anterior.
- Si se termina voluntariamente la simulación: mediante el botón finalizar de la pantalla E, tanto si se ha completado el 100% de la evaluación del paciente como si no (confirmando en ese caso la elección en el *pop-up* correspondiente, Figura 75).

La interfaz está compuesta de un mensaje indicando que se ha terminado la simulación y un campo con el tiempo empleado. Por último, cuenta con un botón “siguiente” que nos redirige a la introducción de datos del paciente. Estos componentes se pueden ver en la captura de la aplicación de la Figura 78.

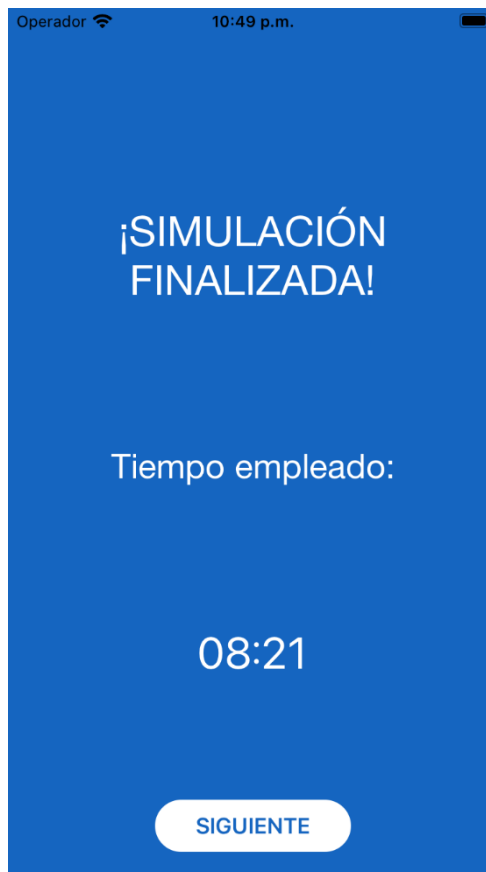


Figura 78. Pantalla final. Fuente: propia.

5.17 Datos del paciente

Tras finalizar la evaluación primaria, se solicitan una serie de datos relevantes del paciente en cuestión, como son el nombre y la edad. Esta información nos servirá para identificar cada simulación dentro de la lista que trataremos en el apartado 5.19. Además, he añadido un campo de observaciones, con un máximo de 200 caracteres, que es opcional y se pueden añadir en él los comentarios que el usuario de la aplicación considere oportunos (en el ejemplo de la Figura 81 se ha incluido que el paciente no presenta alergias).

Los campos nombre (que cuenta con un máximo de 25 caracteres) y edad son obligatorios y no se permite pasar de pantalla sin rellenarlos. En caso de que el usuario pulse el botón “Siguiente” sin haber introducido estos valores aparece por pantalla un mensaje temporal indicándolo, como se muestra en la Figura 80. Cuando se complete la información requerida y se vuelva a pulsar el botón “Siguiente” se pasará a la sección con el resumen de la valoración del paciente.



Figura 79. Pantalla datos del paciente inicial. Fuente: propia.



Figura 80. Error en la pantalla datos del paciente. Fuente: propia.

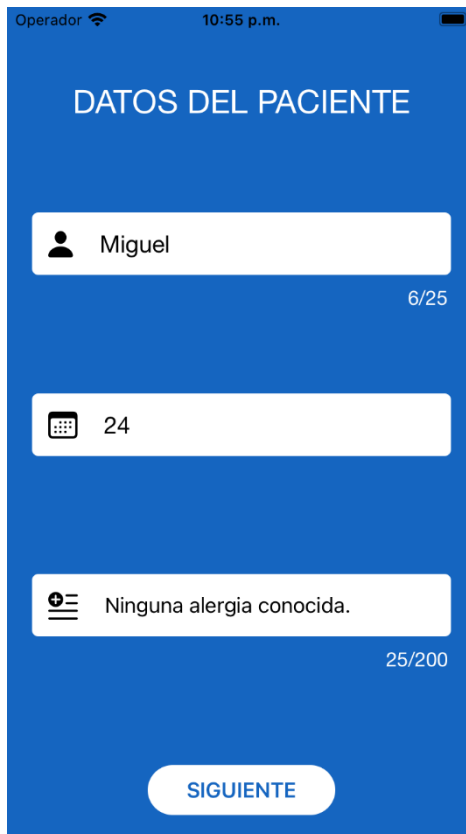


Figura 81. Pantalla datos del paciente completada. Fuente: propia.

5.18 Pantalla resumen

En esta parte de la aplicación se muestra la información relevante de cada uno de los campos tratados en la evaluación inicial. En primer lugar, se presentan los datos de la simulación (tiempo empleado, hora y fecha en que se ha realizado, progreso alcanzado y si se ha finalizado por completo o no), así como los datos del paciente introducidos con anterioridad. A continuación, se muestran los detalles marcados en cada sección de la simulación, separando cada apartado (Seguridad en la escena, Impresión de la gravedad, X, A, B, C, D y E). Cada uno de ellos viene en un recuadro con el mismo color que tenía su correspondiente nivel. Como no cabe toda la información en una sola pantalla, se ha dispuesto en forma de *scroll* en vertical para que se puedan ver todos los campos desplazando hacia arriba con el dedo. Las siguientes tres imágenes, Figura 82, Figura 83 y Figura 84, presentan capturas habiendo realizado este desplazamiento para ver toda la información que contiene.

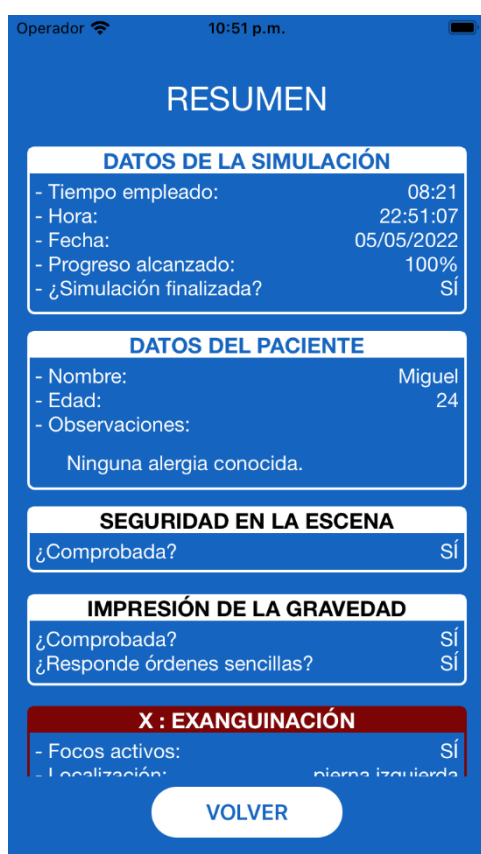


Figura 82. Pantalla resumen (1/3). Fuente: propia.

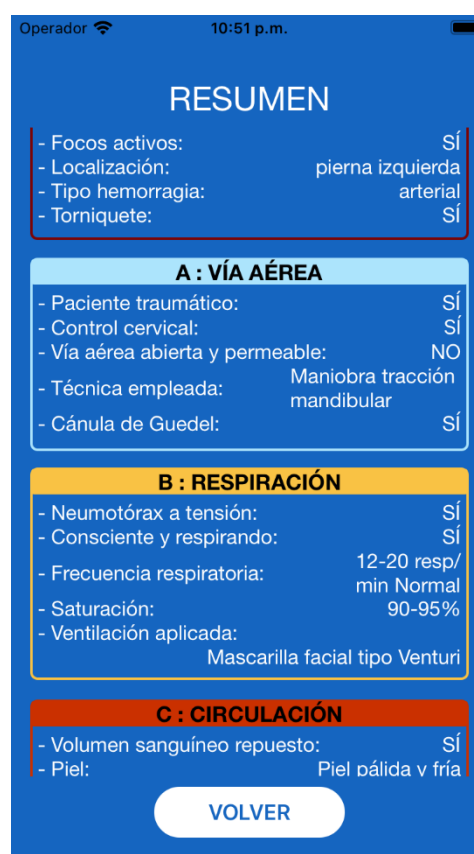


Figura 83. Pantalla resumen (2/3). Fuente: propia.

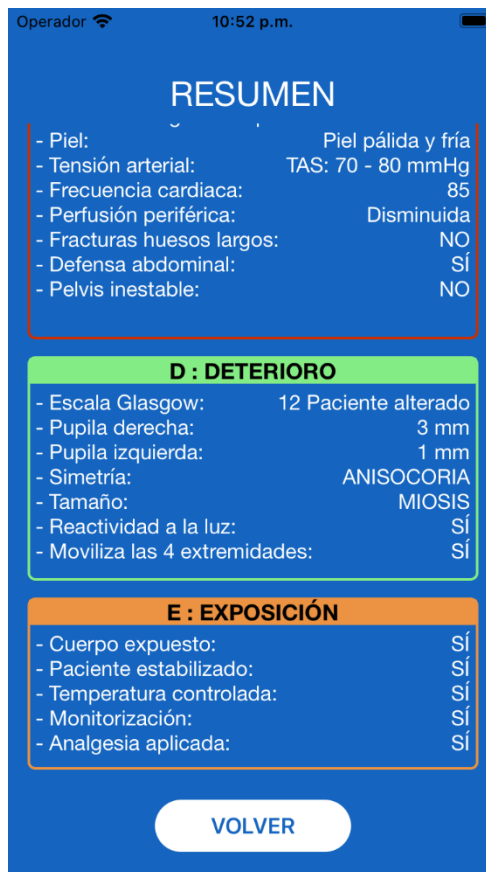


Figura 84. Pantalla resumen (3/3). Fuente: propia.

Por último, hay que destacar que al pulsar el botón “volver” se regresa a la pantalla principal de la aplicación y es en este momento en el que la simulación se guarda en la base de datos. Si este paso se realiza correctamente aparece por pantalla un mensaje indicando que se ha almacenado de forma satisfactoria.

5.19 Listado simulaciones

Al pulsar el botón “simulaciones guardadas” de la pantalla principal, que se ha presentado en la Figura 17, accedemos a una pantalla en la que se muestra un listado de simulaciones guardadas en una base de datos local almacenada en el propio dispositivo. Cada una de ellas se muestra individualmente en formato tarjeta y contiene los datos de la simulación necesarios para identificarla. Estos campos son:

- Número ordinal: asignado a cada simulación según se van almacenando en la lista. En caso de que alguna de ellas se elimine, este número se actualiza de forma que no haya saltos en la numeración.
- Nombre: campo introducido en la pantalla correspondiente a los datos del paciente.
- Edad: campo también introducido en la pantalla sobre los datos del paciente.
- Hora: en formato de horas, minutos y segundos. Hora en que se finalizó la simulación.
- Fecha: en formato día, mes, año. Fecha en que se realizó la simulación.
- Progreso: valor numérico indicando el porcentaje de la aplicación que se llegó a completar antes de finalizarla.

En la parte inferior de la tarjeta correspondiente a cada simulación, además de la información ya mencionada, tenemos tres acciones que podemos realizar (cada una de ellas acompañada de su correspondiente símbolo identificativo):

- **Resumen:** botón que nos permite ver todos los detalles de la simulación dentro de la propia aplicación. Al pulsarlo nos redirige a la pantalla del resumen y se muestra la información de la misma forma que en el apartado 5.18.
- **Descargar:** botón que genera un fichero Excel con la información de la simulación correspondiente. Hay que tener en cuenta que, si ya teníamos un fichero con un cierto número de simulación, volver a dar al botón descargar asociado a ese mismo número sobrescribirá la información anterior.
- **Eliminar:** botón que borra de la base de datos la simulación asociada. Al hacerlo la pantalla se actualiza, reordenando las simulaciones para que no haya saltos en la numeración.

Además de todos los componentes presentados, esta pantalla cuenta con otro símbolo de descarga en la esquina superior derecha, como se puede ver en la Figura 85. Pulsando este botón se genera un fichero Excel con información de la lista completa, es decir, de los detalles de todas las simulaciones que contiene. Cuando se acciona se informa por pantalla mediante un mensaje temporal si se ha generado correctamente el fichero.

Por último, la pantalla cuenta con un botón de “volver” para regresar a la portada de la aplicación.



Figura 85. Pantalla listado de simulaciones. Fuente: propia.

5.20 Fichero Excel

Como hemos comentado en el subapartado anterior, hay dos tipos de fichero Excel que se pueden generar con la aplicación: individual de cada simulación o de la lista completa. El nombre asignado a estos ficheros será “Registro_simulacion_1” donde el 1 se sustituirá por el número asociado en cada caso, y en la lista completa será “Listado_simulaciones”. El formato del fichero Excel con el cual se muestran los datos va a ser diferente por la implementación en Android y en iOS, así como la ruta en la que se almacenará dentro del dispositivo. A continuación, se presentan estos dos aspectos en cada sistema operativo.

5.20.1 Android

En el sistema operativo Android se generan ficheros Excel con formato “Hoja de cálculo XLS”. Esta es una buena elección para mostrar el contenido en cualquier dispositivo. Además, su implementación nos permite añadir colores, lo que va a ser útil para separar visualmente cada sección que se trata en la simulación.

En cuanto a la situación de los ficheros, se podrán encontrar en la carpeta dedicada a la aplicación en el propio dispositivo. Esta es accesible siguiendo la ruta: Archivos – Android – Data – com.ePoliTrauma – Files.

Por último, se muestra en la Figura 86 y en la Figura 87 un ejemplo de fichero generado a partir de una única simulación, en la que se puede observar la forma en la que se presenta la información. Las capturas (que son dos por no caber todo el contenido en una imagen) están realizadas en un dispositivo Android físico.

	A	B	C	D	E	F	G	H
1	DATOS DE LA SIMULACIÓN							
2	- Tiempo empleado:			04:14				
3	- Hora:			18:54:15				
4	- Fecha:			15/05/2022				
5	- Progreso alcanzado:			100%				
6	- ¿Simulación finalizada?			SÍ				
7								
8	DATOS DEL PACIENTE							
9	- Nombre:			Miguel				
10	- Edad:			24				
11	- Observaciones:			Ninguna alergia conocida.				
12								
13	SEGURIDAD EN LA ESCENA							
14	¿Comprobada?			SÍ				
15								
16	IMPRESIÓN DE LA GRAVEDAD							
17	¿Comprobada?			SÍ				
18	¿Responde órdenes sencillas?			SÍ				
19								
20	X : EXANGUINACIÓN							
21	- Focos activos:			SÍ				
22	- Localización:			pierna izquierda				
23	- Tipo hemorragia:			arterial				
24	- Torniquete:			SÍ				
25								
26	A : VIA AÉREA							
27	- Paciente traumático:			SÍ				
28	- Control cervical:			SÍ				
29	- Vía aérea abierta y permeable:			NO				
30	- Técnica empleada:			Maniobra tracción mandibular				
31	- Cánula de Guedel:			SÍ				
32								
33	B : RESPIRACIÓN							
34	- Neumotórax a tensión:			SÍ				
35	- Consciente y respirando:			SÍ				
36	- Frecuencia respiratoria:			12-20 resp/min Normal				
37	- Saturación:			90-95%				
38	- Ventilación aplicada:			Mascarilla facial tipo Venturi				
39								

Figura 86. Captura fichero Excel en Android (1/2).
Fuente: propia.

	A	B	C	D	E	F	G	H
32								
33	B : RESPIRACIÓN							
34	- Neumotórax a tensión:			SÍ				
35	- Consciente y respirando:			SÍ				
36	- Frecuencia respiratoria:			12-20 resp/min Normal				
37	- Saturación:			90-95%				
38	- Ventilación aplicada:			Mascarilla facial tipo Venturi				
39								
40	C : CIRCULACIÓN							
41	- Volumen sanguíneo repuesto:			SÍ				
42	- Piel:			Piel pálida y fría				
43	- Tensión arterial:			TAS: 70 - 80 mmHg				
44	- Frecuencia cardíaca:			85				
45	- Perfusión periférica:			Disminuida				
46	- Fracturas huesos largos:			NO				
47	- Defensa abdominal:			SÍ				
48	- Pelvis inestable:			NO				
49								
50	D : DETERIORO							
51	- Escala Glasgow:			12	Paciente alterado			
52	- Pupila derecha:			3 mm				
53	- Pupila izquierda:			1 mm				
54	- Simetría:			ANISOCORIA				
55	- Tamaño:			MIOSIS				
56	- Reactividad a la luz:			SÍ				
57	- Moviliza las 4 extremidades:			SÍ				
58								
59	E : EXPOSICIÓN							
60	- Cuerpo expuesto:			SÍ				
61	- Paciente estabilizado:			SÍ				
62	- Temperatura controlada:			SÍ				
63	- Monitorización:			SÍ				
64	- Analgesia aplicada:			SÍ				
65								
66								
67								
68								
69								
70								
71								

Figura 87. Captura fichero Excel en Android (2/2).
Fuente: propia.

5.20.2 iOS

En cuanto al sistema operativo iOS, la implementación del fichero Excel se ha realizado mediante un formato "CSV". En concreto se trata de un fichero de valores separados por comas que se puede traducir fácilmente a cualquier formato de archivo Excel deseado.

Los ficheros creados con la aplicación se encuentran en el propio dispositivo en el que se han generado siguiendo la ruta: Archivos – En mi iPhone/iPad – e-poliTrauma. Como se aprecia, es ligeramente distinta si estamos usando un *smartphone* o una tableta.

A continuación, se presentan imágenes de un fichero Excel del listado completo de simulaciones (con dos en este caso) que genera la aplicación, habiendo realizado estas capturas, Figura 88 y Figura 89, en un iPad físico.

NÚMERO SIMULACIÓN	1	2
DATOS DE LA SIMULACIÓN		
- Tiempo empleado:	05:09	02:14
- Hora:	19:55:13	20:00:37
- Fecha:	15/05/2022	15/05/2022
- Progreso alcanzado:	100%	100%
- ¿Simulación finalizada?	SÍ	SÍ
DATOS DEL PACIENTE		
- Nombre:	Miguel	Victor
- Edad:	24	33
- Observaciones:	Ninguna alergia conocida.	
SEGURIDAD EN LA ESCENA		
¿Comprobada?	SÍ	SÍ
IMPRESIÓN DE LA GRAVEDAD		
¿Comprobada?	SÍ	SÍ
¿Responde órdenes sencillas?	SÍ	SÍ
X : EXANGUINACIÓN		
- Focos activos:	SÍ	NO
- Localización:	pierna izquierda	
- Tipo hemorragia:	arterial	
- Torniquete:	SÍ	
A : VÍA AÉREA		
- Paciente traumático:	SÍ	NO
- Control cervical:	SÍ	NO
- Vía aérea abierta y permeable:	NO	NO
- Técnica empleada:	Maniobra tracción mandibular	Maniobra frente-mentón
- Cánula de Guedel:	SÍ	NO
B : RESPIRACIÓN		
- Neumotórax a tensión:	SÍ	NO
- Consciente y respirando:	SÍ	SÍ
- Frecuencia respiratoria:	12-20 resp/min Normal	12-20 resp/min Normal
- Saturación:	90-95%	>95%
- Ventilación aplicada:	Mascarilla facial tipo Venturi	Considerar oxígeno suplementario

Figura 88. Captura fichero Excel en iOS (1/2). Fuente: propia.

B : RESPIRACIÓN		
- Neumotórax a tensión:	SÍ	NO
- Consciente y respirando:	SÍ	SÍ
- Frecuencia respiratoria:	12-20 resp/min Normal	12-20 resp/min Normal
- Saturación:	90-95%	>95%
- Ventilación aplicada:	Mascarilla facial tipo Venturi	Considerar oxígeno suplementario
C : CIRCULACIÓN		
- Volumen sanguíneo repuesto:	SÍ	NO
- Piel:	Piel pálida y fría	Piel normal
- Tensión arterial:	TAS: 70 - 80 mmHg	TAS > 80 mmHg
- Frecuencia cardiaca:	85	72
- Perfusión periférica:	Disminuida	Normal
- Fracturas huesos largos:	NO	NO
- Defensa abdominal:	SÍ	NO
- Pelvis inestable:	NO	NO
D : DETERIORO		
- Escala Glasgow:	12 Paciente alterado	15 Respuesta normal
- Pupila derecha:	3mm	5mm
- Pupila izquierda:	1mm	5mm
- Simetría:	ANISOCORIA	ISOCORIA
- Tamaño:	MIOSIS	NORMAL
- Reactividad a la luz:	SÍ	SÍ
- Moviliza las 4 extremidades:	SÍ	SÍ
E : EXPOSICIÓN		
- Cuerpo expuesto:	SÍ	SÍ
- Paciente estabilizado:	SÍ	SÍ
- Temperatura controlada:	SÍ	SÍ
- Monitorización:	SÍ	SÍ
- Analgesia aplicada:	SÍ	NO

Figura 89. Captura fichero Excel en iOS (2/2). Fuente: propia.

CAPÍTULO 6. DESARROLLO TÉCNICO

En este apartado se van a tratar los principales aspectos técnicos de la aplicación. Esto abarca la implementación y uso de librerías externas para resolver algún requerimiento, cuestiones relacionadas con la interfaz de usuario, y alguna de las funcionalidades de la app con cierto interés.

La aplicación se ha desarrollado de manera análoga en su versión de Android y de iOS, buscando ser lo más idéntica posible tanto en funcionalidades como en interfaz. El objetivo de esto es que los estudiantes de la asignatura a la que sirve de apoyo tengan la misma herramienta de aprendizaje, independientemente del dispositivo del que dispongan. Sin embargo, se van a encontrar ciertas diferencias en la aplicación impuestas por el sistema operativo que estemos usando y su entorno de desarrollo integrado correspondiente. Además, la implementación de la mayoría de las funcionalidades va a ser distinta, por lo que en cada uno de los subapartados se explicarán, cuando sea necesario, las diferencias entre la versión de la aplicación para Android y para iOS.

6.1 Librerías externas

A lo largo del desarrollo de la aplicación, tanto en la versión de Android como de iOS, han surgido funcionalidades que necesitan realizar su implementación mediante librerías externas. Con ellas conseguimos resolver los requerimientos solicitados de forma sencilla en la mayoría de las ocasiones. Aunque estas librerías van a ser distintas en Android y iOS, en general se necesitan las mismas funcionalidades en ambas aplicaciones, por lo que he buscado librerías con implementaciones similares en las dos versiones para facilitar el desarrollo de la app.

En primer lugar, se ha empleado la librería externa Lottie para implementar imágenes con movimiento, disponible en [37]. Este es un aspecto relevante desde un punto de vista visual de la aplicación. Con ella la adición de gifs animados es relativamente sencilla. Además, cuentan con versiones de implementación, tanto para Android como para iOS, ofrecidas por la propia compañía. Este elemento está más explicado en el siguiente apartado del documento, el 6.2.

Para la base de datos se ha buscado una implementación que fuese similar en ambos sistemas operativos. La opción escogida, tal y como se explica en el apartado 6.4, es el uso de la base de datos SQLite. Para realizar esta tarea, se ha precisado añadir una librería externa en la versión de iOS, disponible en [38], la cual aporta una capa de adaptación para emplear SQLite3 sobre el lenguaje de programación Swift. En Android, sin embargo, no se ha necesitado ninguna librería externa, ya que las librerías que aporta de base Android Studio cuentan con una implementación de SQLite.

El siguiente requerimiento de la aplicación que se ha solventado con una librería externa es la creación de ficheros Excel desde la propia app para exportar los resultados de las simulaciones. En la versión de Android se ha usado una API de Java para la creación y lectura de documentos Microsoft denominada Apache POI, cuyo contenido e implementación se puede ver en [39]. Por otro lado, para la aplicación desarrollada en iOS se ha hecho uso de una librería disponible en [40], destinada al trabajo con ficheros Excel en formato CSV para el lenguaje Objective-C, pero que también es compatible con Swift. Estas dos implementaciones se explican de forma detallada en el apartado 6.5 del documento.

Por último, para el uso de los mensajes temporales que aparecen por pantalla en la aplicación, se ha usado una librería externa en la versión de iOS. Para Android su desarrollo es sencillo, ya

que se pueden implementar directamente a través del método *makeText* de la clase *Toast*, con el que se puede determinar el texto que se aparece por pantalla y el tiempo que se muestra. Sin embargo, en iOS (empleando SwiftUI) no existe una clase análoga que facilite esta tarea. Para solventarlo se ha empleado la librería publicada en GitHub en [41]. Esta librería aporta distintos diseños para cada tipo de mensaje, como alertas o errores, en la que se muestra el texto deseado acompañado de un icono representativo, lo que la convierte en una herramienta atractiva en cuestiones de diseño y realmente útil.

6.2 Gifs animados

En la aplicación se han usado una serie de imágenes con movimiento (también llamados *gifs* animados). Para la implementación de dicho tipo de recurso he usado una librería denominada Lottie, creada por la empresa Airbnb. Tal y como presenta la propia compañía en [37], Lottie es un formato de archivos de animación basado en JSON. Estas animaciones se pueden usar en casi cualquier plataforma, y son fácilmente escalables sin pixelarse. Lottie cuenta con una biblioteca de animaciones creadas por la comunidad, siendo muchas de ellas gratuitas.

Además, su implementación tanto para Android como para iOS es sencilla. En la propia página web de Lottie tienen enlaces a repositorios de GitHub donde se encuentra la dependencia necesaria para integrar las animaciones en el proyecto de cada sistema operativo. Tras añadir la librería mencionada, basta con descargar los ficheros Lottie que queramos e incluirlos en la carpeta correspondiente dentro de cada IDE (*raw* en Android). En la implementación gráfica en Android se usa un *LottieAnimationView* con similares características a un *ImageView*, mientras que para iOS se usa un *animationView*. En ambos casos se pueden ajustar parámetros de la animación como pueden ser su velocidad, si se quiere repetir una vez termine, el número de veces que queramos reproducirlo, etc.

La finalidad del uso de los *gifs* animados es hacer más visual la aplicación. Con el uso del movimiento se logra una mejor experiencia de usuario, siendo elementos diseñados para atraer su atención. En concreto, en esta aplicación se han usado un total de cinco animaciones, todas ellas implementadas a través de Lottie. La primera de ellas es un sobre cerrándose, visible en el apartado de contacto presentado en la Figura 21. Las otras cuatro se encuentran en la pantalla dedicada a las medidas en tiempos de Covid-19. Entre ellas tenemos los *gifs* de un virus, el uso de mascarilla, el uso de guantes y la toma de temperatura mediante un termómetro infrarrojo. Estas últimas animaciones pueden verse en la Figura 22.

6.3 Listado de simulaciones

Una de las principales funcionalidades de la aplicación es la lista de simulaciones. Este listado es un historial de evaluaciones pasadas, pudiendo identificar fácilmente la que estemos buscando mediante el nombre del paciente, la hora y la fecha en que se realizó. Además, cada una de estas simulaciones aparece en la lista acompañada de un número ordinal según se van almacenando en la base de datos. Cada una de ellas cuenta con tres acciones que podemos utilizar, ver resumen, eliminar y descargar, las tres acompañadas de un símbolo representativo.

El objetivo de este listado es dotar a la aplicación de un lugar donde consultar simulaciones pasadas. Además, tenemos las funcionalidades de ver sus principales características en su pantalla resumen, y la más importante, descargarla en un fichero Excel. Este era un requerimiento por parte de los profesores de la asignatura Simulación Clínica Avanzada para poder almacenar los resultados de las prácticas y así poder usar la aplicación en la propia

evaluación de la asignatura. Para ello, además de poder descargar un fichero Excel con cada simulación de forma individual, la pantalla cuenta con un símbolo en la esquina superior derecha que permite descargar un listado completo de todas las simulaciones, para así poder compararlas y tenerlas almacenadas en un único archivo.

La implementación de este listado es bastante relevante en la versión de Android. Se ha realizado a través del llamado *RecyclerView*, empleado dentro de un *ScrollView* vertical para poder tener un número ilimitado de simulaciones y permitir visualizarlas desplazando la pantalla del dispositivo. Para su creación se han necesitado tres elementos, el primero de ellos es el *RecyclerView* mencionado, cuya funcionalidad es crear listas infinitas sin que esto suponga una penalización en el rendimiento. Para ello recicla las vistas que emplea en la presentación de cada elemento. Solo se necesita acceder a un cierto número de elementos de la lista (determinado por la cantidad de tarjetas que quepan en la pantalla), reduciendo así el procesamiento de las vistas en caso de que tengamos una lista extensa. El segundo elemento necesario es el *LayoutManager* que define cómo se presentan las vistas hijas dentro del *RecyclerView*. Y el último componente necesario es el llamado *Adapter*. Este es usado por el *RecyclerView* para crear vistas nuevas cuando sea necesario y para asignar los datos que correspondan a cada una de ellas, tanto en su creación como en su actualización. Además, esto nos va a permitir identificar la simulación de la lista sobre las que se quiera realizar alguna acción pulsando sobre alguno de los botones con los que cuenta la vista.

6.4 Base de datos

Para la implementación del listado de simulaciones se necesita hacer uso de una base de datos en la que almacenar información relevante de cada una de ellas. Para esta función existen diversas opciones disponibles, tanto en la versión de Android como de iOS, por lo que se ha buscado la que más se adecúa a nuestro proyecto.

En primer lugar, hay que elegir entre usar una base de datos local o remota. Entre estas dos opciones me he decantado por la base de datos local, utilizando la memoria del propio dispositivo en que se ejecuta la aplicación. Esto aporta varias ventajas, entre las que destacan que no es necesario disponer de conexión a Internet para hacer uso de la app en su totalidad, además de la rapidez al tener la información almacenada de forma local, pudiendo acceder a la misma de forma inmediata.

La base de datos remota ofrece otro tipo de funcionalidades, como puede ser el control de las simulaciones realizadas en diversos dispositivos desde un único punto, o almacenar la información de forma más fiable usando servidores externos en los que los datos, por lo general, estarán más seguros que en el dispositivo físico del usuario. Sin embargo, para la utilidad que se le ha otorgado a la base de datos de la aplicación, que se centra en la revisión de simulaciones antiguas por parte de los alumnos y del profesor de la asignatura en cuestión, no es necesario disponer de una base de datos remota, por lo que la mejor opción y la escogida ha sido trabajar en local.

En este punto se ha elegido también el tipo de base de datos local que utilizar. Existen diversas librerías en la actualidad para implementar esta funcionalidad, como puede destacar *Room* en Android. Para la aplicación de análisis en este proyecto, al precisar de una base de datos tanto en la versión de Android como de iOS, se ha buscado una solución común (o similar) para ambas plataformas que pudiese facilitar su implementación.

La opción elegida finalmente ha sido SQLite. SQLite es una librería de gestión de bases de datos, escrita en C y de dominio público, que permite almacenar información de forma rápida y sencilla, destacando sobre todo su uso en dispositivos móviles por su pequeño tamaño. Tal y como se expone en [42], SQLite es el motor de bases de datos más utilizado en el mundo. Por estos motivos, el uso de SQLite ha sido la opción escogida, pudiendo encontrar bastante información y tutoriales sobre su implementación tanto en Android como en iOS. Para la versión de Android se emplean librerías propias de Android Studio, *SQLiteDatabase* y *SQLiteOpenHelper*, mientras que para la versión en iOS he utilizado una librería externa. Esta última, disponible en [38], aporta una capa de adaptación para hacer uso del lenguaje de programación Swift sobre SQLite3.

Respecto al uso de esta base de datos, las simulaciones se guardan tras pulsar el botón “volver” de la pantalla de resumen (a la que se accede directamente tras finalizar cada simulación). En este momento se le asigna un número ordinal y se almacena toda la información relevante en una tabla. El contenido de esta base de datos se muestra en la pantalla “Listado de simulaciones”. En un primer momento se presentan los datos más característicos para su identificación en la propia lista, pudiendo acceder a la información completa pulsando el botón de resumen asociado a una simulación concreta, o descargando el fichero Excel, acción que también hace uso de la base de datos. El último caso para modificar la *database* es el botón de eliminar una cierta simulación. Tras pulsarlo se borra dicha entrada en la tabla de simulaciones, reordenando el resto para presentar los números ordinales sin saltos. Por último, se vuelve a mostrar el listado con la información actualizada.

El contenido de esta base de datos son un conjunto de valores relevantes, guardados en una clase con variables globales, que se determinan en función de las acciones del usuario de la aplicación. Muchos de ellos son enteros tratados en binario, con valor de 0 si no se ha pulsado un determinado botón o 1 si se ha accionado. Esto sirve para mostrar el contenido de una pantalla anterior con la información aportada por el usuario (sin que esta se reinicie al pasar entre pantallas), permitiendo además el uso de propiedades como la opacidad de las vistas, vinculada a alguna de estas variables globales, que muestran la vista si está en 1 o la ocultan si está en 0. Estas variables también se utilizan para actualizar la barra de progreso de la simulación. También hay otro tipo de valores almacenados como *Strings*, que permiten guardar cadenas de texto por el usuario como pueden ser el nombre del paciente o las observaciones.

6.5 Fichero Excel

Uno de los primeros requerimientos impuestos en la realización de la aplicación fue la creación de un fichero Excel con los resultados de la simulación. Por ello se ha implementado un botón en la pantalla del listado de simulaciones para tal efecto. En cada una de las tarjetas vinculadas a una simulación concreta aparece el botón “descargar”. Tras pulsarle se muestra por pantalla un mensaje informando si este proceso se ha realizado de forma satisfactoria o si se ha producido algún error. Además, se ha dispuesto de otro botón en la esquina superior derecha de la misma pantalla mediante el cual se genera un fichero Excel con todas las simulaciones que tengamos en el listado. De esta forma podemos verlas todas juntas en un único archivo para poder compararlas.

Uno de los objetivos de esta funcionalidad es puramente académico. La aplicación va a ser empleada por los alumnos de la asignatura Simulación Clínica Avanzada en la realización de sus prácticas. Por tal motivo, el correcto uso de la app y los resultados que se obtengan pueden ser objetos de calificación por parte de los profesores. Así pues, conviene que los datos que quedan

recogidos en el dispositivo en que se realizan las simulaciones de forma local se puedan exportar para su posterior análisis en cualquier dispositivo, tanto por parte de los estudiantes como de los profesores.

El contenido de estos ficheros es el mismo que recoge el apartado resumen. En concreto, presentan información sobre la realización de la simulación (tiempo empleado, hora y fecha en que se ha realizado, progreso alcanzado y si se ha finalizado por completo o no) y los datos del paciente introducidos por el usuario. Tras esto, se muestran los detalles más relevantes de cada sección de la simulación separándolos en apartados (Seguridad en la escena, Impresión de la gravedad, X, A, B, C, D y E).

En Android se ha hecho uso de una API de Java para la creación y lectura de documentos Microsoft denominada Apache POI, cuyo contenido incluyendo guías de uso está disponible en [39]. En nuestro caso se ha necesitado la librería para ficheros de tipo HSSF, una implementación de Java para el trabajo con archivos Excel de formato XLS. Este es un tipo de archivo adecuado para su uso en cualquier dispositivo. Además, esta implementación permite añadir colores, lo que ha resultado útil para separar visualmente cada apartado tratado en la simulación, tal y como se puede ver en la Figura 86 y en la Figura 87. Al descargar los ficheros en la aplicación Android, quedarán almacenados en la carpeta destinada a la app en el propio dispositivo. Esta es accesible siguiendo la ruta: Archivos – Archivos del dispositivo – Android – Data – com.e-poliTrauma – Files.

Para la aplicación destinada al sistema operativo iOS, la implementación del fichero Excel se ha realizado mediante el formato CSV, usando una librería disponible en [40] para el lenguaje de programación Objective-C pero compatible con Swift. Este formato almacena valores separados por comas, pero se puede traducir fácilmente a cualquier tipo de archivo Excel. Los ficheros generados con la aplicación se encuentran en el propio dispositivo donde se han generado siguiendo la ruta: Archivos – En mi iPhone/iPad – e-poliTrauma. Un ejemplo del mismo se ha presentado en la Figura 88 y en la Figura 89.

6.6 Preferencias de usuario

En la mayoría de las aplicaciones es necesario el almacenamiento de ciertas preferencias del usuario, valores que este decide y que se deben guardar aunque se cierre la aplicación. Para ello se utiliza cierta parte de memoria local que el dispositivo en el que se descarga la app destina a la misma.

En la aplicación de análisis en este trabajo se ha empleado para que los usuarios puedan decidir si quieren que la pantalla destinada a las medidas por la Covid-19 aparezca al comenzar una simulación. En ella se recogen procedimientos fundamentales que tomar en la actualidad en cualquier emergencia sanitaria, pero que pueden ser repetitivos y una vez tenidos en cuenta no es necesario que aparezcan en cada evaluación. Por este motivo se ha incluido en la parte inferior de dicha pantalla una casilla de selección acompañada del mensaje “No volver a mostrar esta pantalla”. Si se marca la casilla, pulsando sobre ella o sobre el texto correspondiente, se guarda esta selección para futuras simulaciones (mediante una variable booleana) aunque se cierre la aplicación. Si en algún momento se quiere volver a activar, tenemos en el menú de la pantalla principal una opción para hacerlo, cambiando el valor de la variable mencionada y mostrando en un mensaje emergente la opción escogida.

Para la implementación de esta funcionalidad, Android ofrece una librería llamada *SharedPreferences*. Esta consiste en una colección de pequeño tamaño con pares clave-valor,

por lo que está especialmente dedicada a almacenar las preferencias del usuario sobre diversos valores de la aplicación. Por la parte de iOS, existe una clase que realiza una función similar, denominada *NSUserDefaults*, y es la que he empleado en el proyecto.

6.7 Pantallas emergentes

A lo largo de la realización de la evaluación existen circunstancias en las que se necesita del uso de pantallas emergentes, también conocidas como *pop-ups*. Consisten en una pantalla que aparece tras realizar algún paso de la aplicación, mostrándose por encima de la que la invoca. Los beneficios que este tipo de funcionalidad ofrece son las de llamar la atención del usuario de la aplicación cuando sea preciso, para que introduzca algún tipo de valor solicitado o para mostrar información relevante en ese momento.

Existen numerosos usos de *pop-ups* en la aplicación que estamos tratando, los cuales pueden verse en diversas capturas en el apartado 5 de esta memoria. Su implementación tiene alguna diferencia entre la versión de Android y de iOS, aunque la funcionalidad es completamente idéntica. En general se han usado los mismos colores en el *pop-up* que en la pantalla que lo invoca para verificar de una forma visual que nos seguimos encontrando en el mismo nivel de la aplicación.

En Android se ha hecho uso de una clase propia de Android Studio, *AlertDialog*. Con esto generamos un cuadro de diálogo cuyo tamaño se adapta al contenido que tengamos en su correspondiente vista, y que aparece en el centro. Para cerrar esta pantalla emergente se puede utilizar el botón que se ha implementado para tal uso, o bien pulsar en la pantalla del dispositivo fuera del *pop-up*, regresando al punto anterior.

En iOS, he usado el método *sheet*, que abre pantallas emergentes apareciendo desde la parte inferior del dispositivo cuando un determinado valor booleano se hace *true*. Para desactivarla nuevamente se puede emplear el botón correspondiente, o bien se puede desplazar el *pop-up* hacia abajo, volviendo a la pantalla original.

En las siguientes dos capturas se pueden apreciar estas diferencias comentadas en cada sistema operativo. En la Figura 90 se muestra una captura del *pop-up* de neumotórax a tensión en Android (en este caso en idioma inglés), en la que se aprecia que la ventana emergente empleada en este sistema operativo no ocupa todo el espacio. En este caso la ventana de la que emerge aparece visible pero más oscura en un segundo plano. En la Figura 91 se ha dispuesto una captura del mismo *pop-up* en iOS. En ella se aprecia que la ventana emergente ocupa casi todo el espacio, salvo una parte en la zona superior en la que se queda almacenada la pantalla desde donde ha sido invocada. Como vemos, aunque hay ligeras diferencias de interfaz, la utilidad en ambos casos es exactamente la misma.

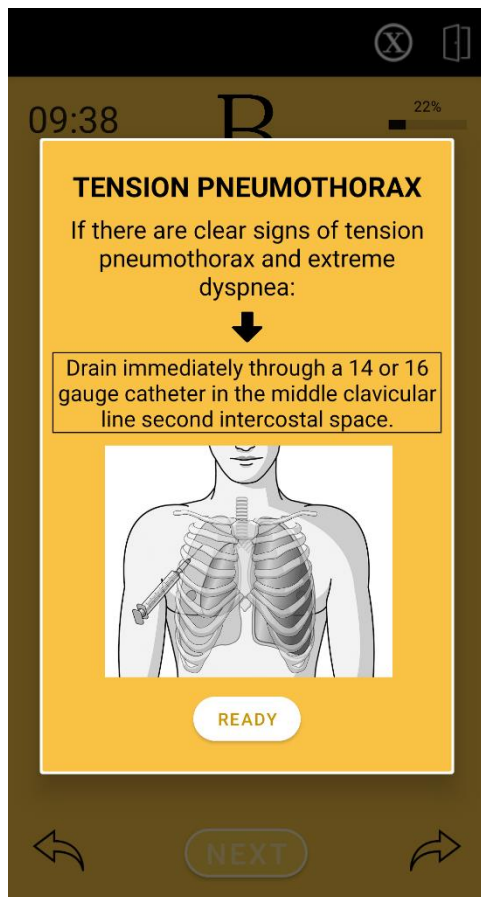


Figura 90. Ejemplo pop-up Android. Fuente: propia.

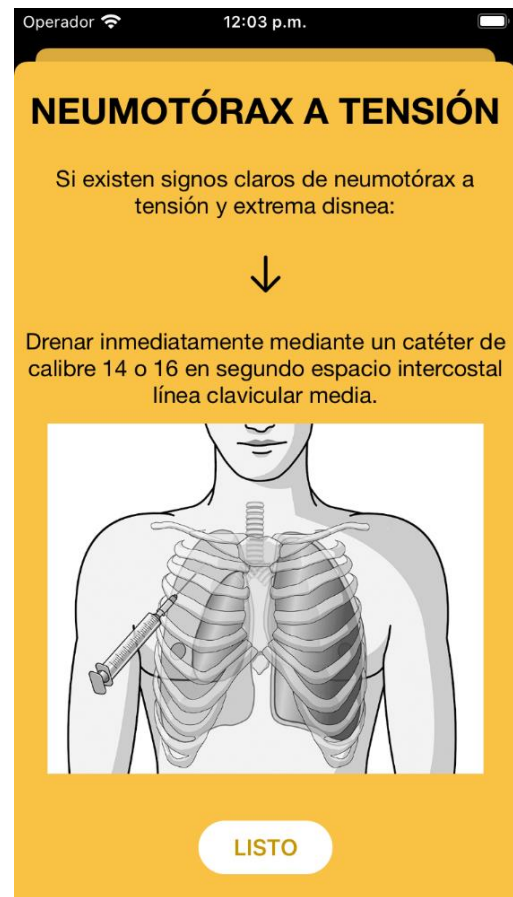


Figura 91. Ejemplo pop-up iOS. Fuente: propia.

6.8 Localización del foco de sangrado

Una de las funcionalidades que ofrece la aplicación es la de seleccionar sobre la imagen del cuerpo del paciente el punto en el que se encuentra el foco activo de sangrado, en caso de que presente alguno. Esto aparece en un *pop-up* dentro de la pantalla dedicada a la exanguinación. Su implementación no es sencilla, ya que la imagen usada para el cuerpo humano se debe redimensionar adaptándose al tamaño de pantalla del dispositivo. Esto se consigue utilizando una altura de `Odp` en Android, o con los parámetros *resizable* y *scaledToFit* propios de SwiftUI en iOS.

Para poder seleccionar cada parte de la figura por separado he empleado un conjunto de imágenes transparentes superpuestas sobre el cuerpo completo del paciente. Cada una de ellas se adapta al tamaño de la parte del cuerpo a la que corresponden, habiendo tenido en cuenta cabeza, cuello, pecho, abdomen, brazo derecho e izquierdo, mano derecha e izquierda, pierna derecha e izquierda y pie derecho e izquierdo. Todas estas imágenes transparentes, que funcionan como botones, se redimensionan de la misma forma que la imagen original del paciente.

Al mismo tiempo, cuando se selecciona la parte del cuerpo en la que se encuentra la hemorragia, esta se pinta de rojo sobre la imagen original. Para ello se selecciona la figura correspondiente con la parte del cuerpo iluminada de los recursos de la aplicación, teniendo una imagen diferente por cada una de las secciones diferenciadas. En las dos siguientes capturas se muestra

un ejemplo en Android, en la versión en inglés, primero con la pantalla en su estado inicial, Figura 92, y después habiendo pulsado sobre el abdomen del paciente, Figura 93. Otro ejemplo en iOS seleccionando la pierna izquierda fue presentado en la Figura 32.

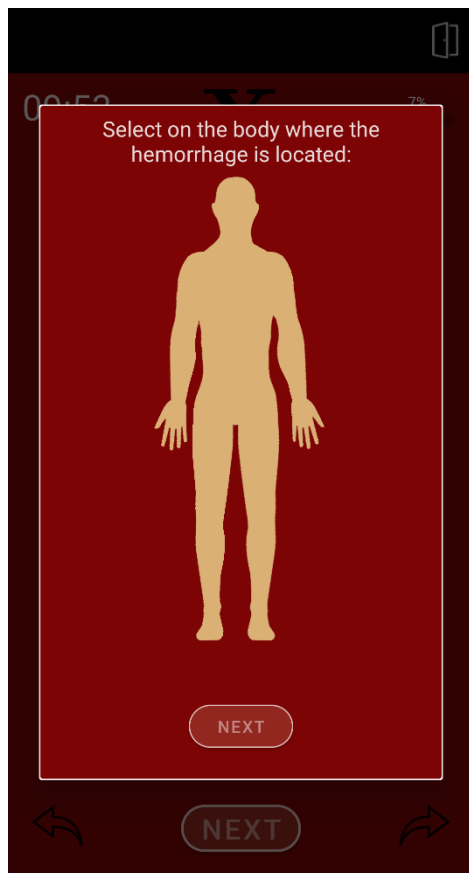


Figura 92. Pop-up localización de hemorragia en Android inicial. Fuente: propia.



Figura 93. Pop-up localización de hemorragia en Android completada. Fuente: propia.

6.9 Diseño adaptativo

Uno de los aspectos más importantes a tener en cuenta en la creación de la aplicación es la adaptación de la misma a distintos tamaños de pantalla. Su uso principal a nivel educativo en la asignatura Simulación Clínica Avanzada se va a llevar a cabo en las tabletas propias de *Apple*, los iPads, por lo que debe tener una buena presencia en tamaños de pantalla grandes. Pero al mismo tiempo, tanto los estudiantes de la asignatura, como otros posibles usuarios de la aplicación, deben poder descargarla en sus dispositivos personales, entre los que puede haber tamaños de pantalla muy dispares.

Este es una característica muy relevante en la aplicación diseñada para Android. En este sistema operativo, como ya se ha comentado, existe mucha fragmentación en el mercado con gran cantidad de distribuidores de dispositivos móviles implicados, muchos modelos distintos y, por lo tanto, existen bastantes tamaños de pantalla diferentes e incluso varias versiones del sistema operativo a los que hay que dar cobertura. Para poder adaptar la aplicación se han usado diversas funcionalidades que ofrece Android Studio.

En primer lugar, el uso de diferentes tipos de *layouts* según sean los requerimientos de la pantalla. Entre estos destacan el *ConstraintLayout*, empleado para anclar determinados elementos de la pantalla a un lateral de la misma de forma fija, el *FrameLayout*, principalmente aplicado a situaciones en que necesitemos superponer varios elementos unos encima de otros, y el *LinearLayout* (tanto vertical como horizontal), el cual se usa para concatenar elementos en la pantalla, pudiendo determinar características del mismo como la distancia entre los elementos. Además, en Android Studio también existe la posibilidad de usar un componente denominado *Space* que permite separar otros elementos modificando el tamaño del mismo, que se adapta a distintos tamaños de pantalla.

Otro de los aspectos relevantes es el uso de la asignación de pesos (*weight*), especialmente útil para los *LinearLayout*. Podemos determinar la sección de pantalla que se le otorga al *layout*, y con la propiedad *weight* asignamos a cada elemento la parte que le corresponde de forma relativa en función del total. De esta manera podemos tener tres elementos en una pantalla, y que, por ejemplo, ocupen cada uno de ellos un tercio de la misma. Esta relación en el diseño se va a mantener independientemente del tamaño de pantalla del dispositivo, y por eso es realmente útil.

Por último, en lo relativo a Android, es adecuado hacer uso de dos tipos de unidades que usa Android Studio, el *dp* y el *sp*. El *dp*, *density-independent pixel* o píxel independiente de la densidad, es una unidad abstracta basada en la densidad física de píxeles de la pantalla. Es una buena solución para mantener la relación entre el tamaño de cada elemento y el total, independientemente del tamaño del dispositivo. Se utiliza para la gran mayoría de componentes o vistas que añadamos a una pantalla, destacando también su uso en los tamaños de las imágenes que empleemos. Un aspecto importante asociado al *dp* es el uso como medida de *Odp*, indicando que se utilizará toda la parte de la pantalla disponible, independientemente de cuál sea el valor específico. El *sp*, *scale-independent pixel* o píxel, es similar al *dp* pero empleado para el tamaño de fuente de los textos de la aplicación. Además, el *sp* se calcula según la preferencia de fuente que el usuario tenga asignado en su dispositivo. Por lo tanto, el tamaño de los textos se ajusta tanto a la densidad de pantalla como a las preferencias del usuario.

A continuación, se muestran capturas de pantalla de la aplicación en dos dispositivos de Android distintos realizadas directamente sobre el simulador de *Android Studio* con el idioma inglés. El primero de ellos es un *smartphone* con unas dimensiones bastante reducidas, el Galaxy Nexus, con un tamaño de pantalla de 4.65 pulgadas. El siguiente dispositivo se trata del modelo de tableta Google Píxel C con 10.2 pulgadas de pantalla. Aunque en las siguientes imágenes estos dos dispositivos no presentan mucha diferencia de tamaño para poder mostrarlos en paralelo, en realidad el Galaxy Nexus tiene unas dimensiones de 13,55 cm de alto por 6,79 cm de ancho, mientras que el Píxel C mide 17,9 cm por 24,2 cm, por lo que sí que presentan mucha diferencia en la realidad.

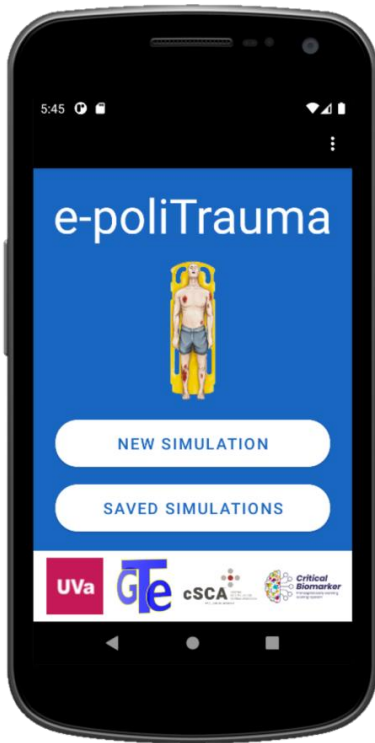


Figura 94. Portada en Galaxy Nexus (Android).
Fuente: propia.

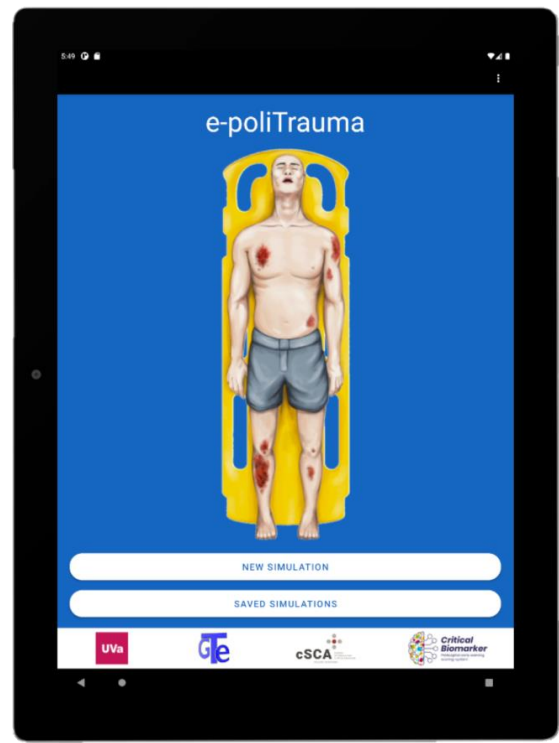


Figura 95. Portada en Pixel C (Android). Fuente:
propia.

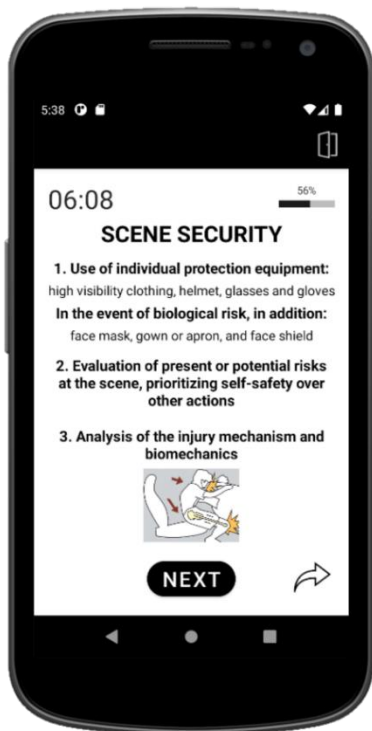


Figura 96. Seguridad en la escena en Galaxy Nexus
(Android). Fuente: propia.

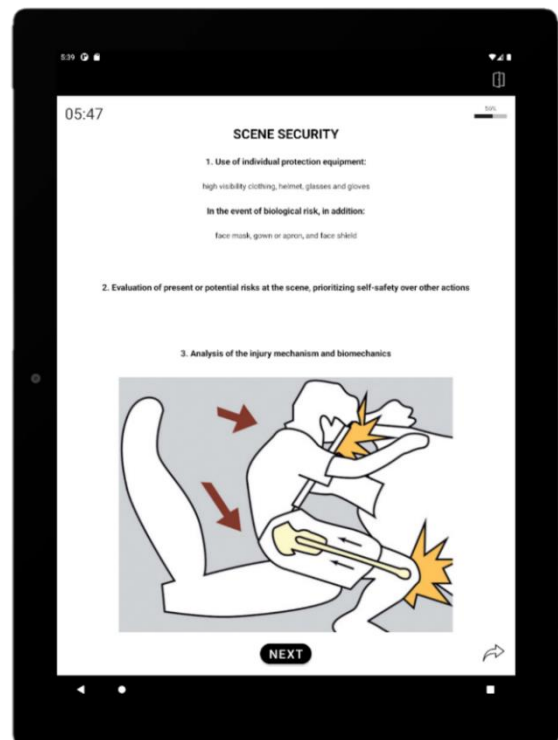


Figura 97. Seguridad en la escena en Pixel C
(Android). Fuente: propia.



Figura 98. Resumen en Galaxy Nexus (Android).
Fuente: propia.



Figura 99. Resumen en Pixel C (Android). Fuente: propia.

Por la parte de la aplicación destinada al sistema operativo iOS, el problema de fragmentación es mucho menor por tener un solo fabricante, *Apple*, y contar con pocos modelos de dispositivos. Sin embargo, sigue siendo importante la adaptación de la app a distintos tamaños de pantalla, para que se pueda usar tanto en *iPhones* como en *iPads*.

Para la realización de este requerimiento de la aplicación usando SwiftUI existen elementos similares a los encontrados en Android, *VStack* y *HStack* (parecidos a los *LinearLayout*) y *ZStack* (similar al *FrameLayout*). El que más se ha empleado ha sido el *VStack*, utilizando con él un elemento muy útil denominado *Spacer*. Este crea una vista adaptativa sin contenido que se expande tanto como puede dentro del *stack* en el que se encuentra, quedando el resto de los elementos empujados él, de forma que se aprovecha todo el tamaño de pantalla independientemente de cuál sea. En cuanto la adaptación de las imágenes incluidas en la aplicación, existen dos métodos en SwiftUI que nos permiten realizar esta tarea, los cuales son *resizable*, que permite modificar el tamaño de la imagen, y *scaledToFit*, que adapta su tamaño hasta alcanzar todo el espacio disponible para la misma, pero manteniendo la relación de aspecto.

Esta adaptación se puede ver en las siguientes figuras, de la 100 a la 105, donde se muestra una comparativa entre la aplicación presentada en un iPhone SE y en un iPad, con capturas realizadas en el simulador de Xcode. Aunque se presentan en tamaño de manera similar para poder visualizarlas mejor en paralelo, hay que destacar que las medidas del iPhone SE son de 13,84 cm de alto y 6,73 cm de ancho, mientras que las del iPad de novena generación usado en las capturas son de 25,06 cm de alto y 17,41 cm de ancho.



Figura 100. Portada iPhone SE (iOS). Fuente: propia.



Figura 101. Portada iPad (iOS). Fuente: propia.



Figura 102. Suministrar oxígeno iPhone SE (iOS). Fuente: propia.



Figura 103. Suministrar oxígeno iPad (iOS). Fuente: propia.



Figura 104. Resumen iPhone SE (iOS). Fuente: propia.

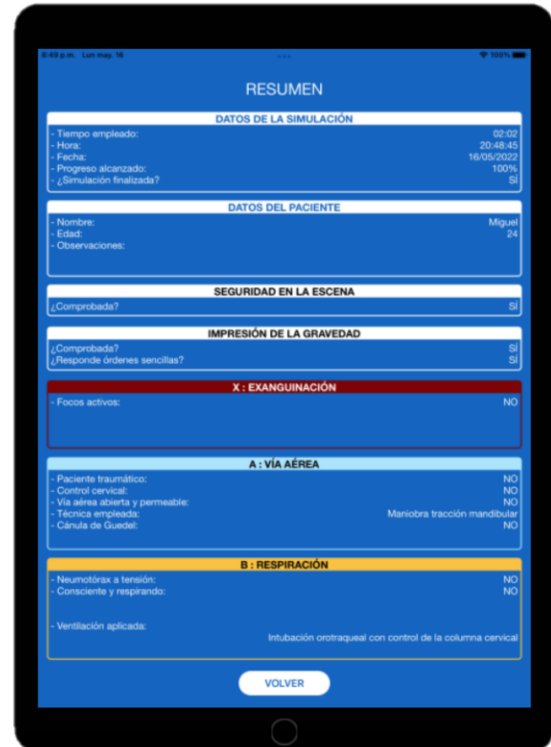


Figura 105. Resumen iPad (iOS). Fuente: propia.

6.10 Aplicación multilingüe

Por último, para hacer la aplicación más completa, se ha desarrollado tanto en español como en inglés, lo que la convierte en aplicación multilingüe. El idioma que presenta la app viene dispuesto por el que se tenga como preferencia en el dispositivo si se está usando español o inglés. En caso de que se esté empleando cualquier otro idioma se toma por defecto el inglés.

Para su implementación se han empleado las herramientas que ofrecen los IDEs tanto en Android como en iOS para esta tarea. En Xcode se ha hecho uso de un fichero de tipo *localizable* para cada idioma. Además, se ha creado una de las *extensions* propias del lenguaje Swift, añadiendo una nueva funcionalidad a la clase existente *String* para hacer más accesible la traducción de los textos añadidos. De forma análoga se ha implementado la aplicación multilingüe en Android. Para ello se han creado dos ficheros "*strings*" de tipo *xml* distintos, uno por cada idioma (español e inglés).

CAPÍTULO 7. CONCLUSIONES Y LÍNEAS FUTURAS

Para finalizar con la memoria de este TFG conviene exponer unas breves conclusiones sobre los distintos aspectos de su desarrollo. Para ello voy a explicar los motivos de la creación de este proyecto, así como las ventajas que ofrece la aplicación e-poliTrauma tanto a nivel educativo como desde un punto de vista puramente médico. Tras esto se establecen las líneas a seguir en futuras versiones de la aplicación, recogiendo el procedimiento que se empleará para su análisis.

7.1 Conclusiones

En primer lugar, cabe destacar el elevado uso de la tecnología móvil en la actualidad. Este tipo de dispositivos, destacando los *smartphones* y las *tablets*, ya forman parte de la vida cotidiana de las personas. Además, su crecimiento está directamente relacionado con el incremento de uso de las aplicaciones móviles, las cuales son empleadas tanto a nivel de ocio como a nivel profesional. Y esto no parece que vaya a frenar en los años venideros.

Dos campos en los que el uso de dispositivos y aplicaciones móviles se están adentrando cada vez más son la educación y la salud. Por ello se han destacado a lo largo de la memoria dos términos con especial relevancia que forman parte del contexto de la aplicación desarrollada, *mHealth* y *mLearning*. Cada vez el uso de apps está más integrado en el campo de la educación, sobre todo en universidades, pero está mucho más extendida en el ámbito del bienestar y la salud, encontrando a día de hoy infinidad de aplicaciones en el mercado.

Muchas de las herramientas dedicadas a la salud se centran en el paciente, pero están ganando importancia las que ayudan directamente a los profesionales sanitarios, sobre todo en la toma de decisiones. Dentro de este contexto se ha desarrollado la aplicación e-poliTrauma, de forma nativa tanto en Android como en iOS. Esta app se ha centrado en un aspecto clave de la medicina prehospitalaria, la evaluación inicial de pacientes politraumatizados, aunque va a ser útil para cualquier tipo de paciente crítico. Como ya se ha expuesto, una intervención temprana, rápida y sistematizada puede salvar vidas, al estar tratando en muchas ocasiones con lesiones de compromiso vital.

La aplicación e-poliTrauma se ha creado con la finalidad de dar apoyo a los estudiantes de la asignatura Simulación Clínica Avanzada en el ejercicio de sus prácticas. Sin embargo, al haber seguido las últimas recomendaciones de actuación en este tipo de emergencias prehospitalarias, la metodología XABCDE, va a ser útil para cualquier profesional sanitario. Las normas de actuación seguidas son bastante novedosas al incorporar la exanguinación como primera sección a tratar en la evaluación del paciente. Aunque ya están ampliamente extendidas y validadas por la comunidad médica, apenas existen herramientas de apoyo en el mercado que faciliten este trabajo, por lo que el proyecto desarrollado va a ser realmente útil.

Centrándonos en las virtudes y los beneficios que aporta la aplicación se pueden destacar varios puntos:

- Facilidad de uso: se ha conseguido crear una aplicación sencilla de manejar, ya que va a ser empleada en situaciones de mucha presión, acompañándola de un diseño atractivo para obtener una interfaz amigable para el usuario.
- Concisa y rápida: puesto que la evaluación inicial de pacientes politraumatizados se debe efectuar en el menor tiempo posible, uno de los mejores aspectos que tiene la app es la velocidad de uso. Además, es muy concisa, presentando solo información relevante

y útil para el personal sanitario, pero a la vez completa, tratando todos los aspectos necesarios.

- Rigurosidad de información: al ser una aplicación destinada a profesionales sanitarios con el objetivo de salvar vidas, es de crucial importancia disponer de contenido de calidad. Para ello se han seguido las últimas recomendaciones médicas en emergencias prehospitalarias para pacientes politraumatizados que expone PHTLS en [33], resumidas y supervisadas por el doctor y profesor de la facultad de Medicina Francisco Martín Rodríguez. Con esto se va a conseguir que sea útil, tanto para estudiantes en sus prácticas, como para los profesionales de la salud.
- Herramienta educativa: al haber dispuesto de una base de datos donde guardar las simulaciones, así como de la posibilidad de exportar los resultados a ficheros Excel, el correcto uso de la aplicación podrá formar parte de la calificación de las prácticas de la asignatura Simulación Clínica Avanzada. De esta forma va a servir de utilidad tanto a los estudiantes como a los profesores.
- Diseño adaptativo: la aplicación, que se ha desarrollado tanto para Android como para iOS, cuenta con un diseño que se adapta a distintos tamaños de pantalla y resoluciones. Se podrá usar en prácticamente la totalidad de dispositivos del mercado, abarcando casi cualquier *smartphone* y *tablet*.
- Aplicación multilingüe: para hacer más completo el trabajo desarrollado, y llegar a un mayor número de potenciales usuarios, se ha traducido toda la aplicación al inglés.

Por último, quiero destacar lo gratificante que ha sido el desarrollo de este TFG. Primero desde un punto de vista educativo, completando mi formación y habiendo permitido adentrarme en el desarrollo de aplicaciones móviles. Pero también a nivel personal, habiendo realizado un trabajo muy enriquecedor basado en el autoaprendizaje y enfocado en un área con gran interés como es el de la salud.

7.2 Líneas futuras

Por último, tras haber finalizado el desarrollo de la aplicación y habiendo destacado los beneficios que ofrece en su estado actual, conviene exponer una serie de actuaciones a seguir para mejorar el proyecto en futuras versiones.

El próximo paso que se va a tomar en este trabajo es la valoración de la aplicación en el contexto de la asignatura Simulación Clínica Avanzada. Los alumnos, al estar bajo presión en esos momentos, suelen tener fallos en la evaluación sistematizada del paciente, ya que en bastantes ocasiones olvidan los pasos a seguir y su orden. Con la aplicación se espera que se solventen gran parte de estos problemas, al poder consultar información y obtener recomendaciones de actuación en todo momento. De esta forma se busca mejorar los resultados obtenidos, pero no será hasta que los estudiantes prueben a realizar sus prácticas apoyándose en esta herramienta cuando se puedan valorar realmente los beneficios que ofrece. El modo de análisis consistirá en prácticas evaluando un caso clínico que ya se haya empleado en años anteriores, para así poder realizar una comparativa con los nuevos resultados empleando la aplicación.

Además, será conveniente obtener *feedback* por parte de los estudiantes para poder mejorar o añadir los aspectos que consideren oportunos. Para ello se dispondrá de un cuestionario tras el empleo de la aplicación en el que podrán valorar diferentes cuestiones como su diseño, la facilidad de uso, las funcionalidades que ofrece, la efectividad e incluso sugerir nuevas ideas con las que hacer la herramienta más completa.

Aunque todavía no se ha puesto en marcha el uso de la aplicación, ya existen algunos aspectos que podrían mejorar el producto de cara a futuras versiones:

- Traducir la aplicación a más idiomas. Pese a estar disponible tanto en español como en inglés, se podrían seguir añadiendo traducciones a otros lenguajes con los que llegar a más público.
- Continuar con las simulaciones guardadas para los casos en los que no se haya finalizado por completo. Tras terminar las evaluaciones y almacenar sus resultados en la base de datos, podría ser útil permitir al usuario seguir con la valoración más adelante desde el punto en que la dejó.
- Añadir casos clínicos dentro de la aplicación, con la finalidad de ofrecer ejemplos de evaluaciones en pacientes politraumatizados con las que poder practicar antes de enfrentarse a casos reales.
- Mejorar el código. Al ser mi primer trabajo en desarrollo de aplicaciones, gran parte del código de las funcionalidades está dispuesto en los mismos ficheros que las vistas. Sería buena idea mejorar su disposición añadiendo algún tipo de arquitectura como MVVM (Model-View-ViewModel).
- Desarrollar la misma aplicación en formato multiplataforma y realizar una comparativa en cuestiones de diseño y rendimiento con las aplicaciones nativas creadas en este proyecto.
- Presentar indicadores de los sensores en el desarrollo de las prácticas en la propia aplicación. En el Centro de Simulación Clínica Avanzada se trabaja con modelos anatómicos que acercan la experiencia a la realidad. Podría ser conveniente presentar los resultados de estos sensores y de la maquinaria disponible desde la propia aplicación, para así condensar toda la simulación en un mismo dispositivo.

Bibliografía

- [1] A. Phongtraychack y D. Dolgaya, «Evolution of Mobile Applications,» *MATEC Web of Conferences*, vol. 155, ene. 2018.
- [2] C. Pagliari, D. Sloan, P. Gregor, F. Sullivan, D. Detmer, J. P. Kahan, W. Oortwijn y S. MacGillivray, «What Is eHealth (4): A Scoping Exercise to Map the Field,» *J Med Internet Res*, vol. 7, nº 1, mar. 2005.
- [3] «Estudio de Consumo Móvil en España,» Deloitte, 2017. [En línea]. Disponible en: <https://www2.deloitte.com/es/es/pages/technology-media-and-telecommunications/articles/consumo-movil-espana.html>. [Último acceso: 19 abril 2022].
- [4] «Encuesta sobre Equipamiento y Uso de Tecnologías de Información y Comunicación en los Hogares,» Instituto Nacional de Estadística, 2021. [En línea]. Disponible en: https://www.ine.es/prensa/tich_2021.pdf. [Último acceso: 19 abril 2022].
- [5] «State of Mobile 2022,» Data AI, 2022. [En línea]. Disponible en: <https://www.data.ai/en/go/state-of-mobile-2022/>. [Último acceso: 19 abril 2022].
- [6] Global Observatory for eHealth, «mHealth: new horizons for health through mobile technologies,» Organización Mundial de la Salud, 2011. [En línea]. Disponible en: https://apps.who.int/iris/bitstream/handle/10665/44607/9789241564250_eng.pdf?. [Último acceso: 19 abril 2022].
- [7] C. P. Adans-Dester et al., «Can mHealth Technology Help Mitigate the Effects of the COVID-19 Pandemic?,» *IEEE Open Journal of Engineering in Medicine and Biology*, vol. 1, pp. 243-248, 2020.
- [8] H. Kondylakis, D. G. Katehakis, A. Kouroubali, F. Logothetidis, A. Triantafyllidis, I. Kalamaras, K. Votis y D. Tzovaras, «COVID-19 Mobile Apps: A Systematic Review of the Literature,» *J Med Internet Res*, vol. 22, nº 12, dic. 2020.
- [9] H. John Leon Singh, D. Couch y K. Yap, «Mobile Health Apps That Help With COVID-19 Management: Scoping Review,» *JMIR Nursing*, vol. 3, nº 1, 2020.
- [10] J. Rodríguez Arce y J. P. C. Juárez Pegueros, «Impacto del m-learning en el proceso de aprendizaje: habilidades y conocimiento,» *RIDE. Revista Iberoamericana para la Investigación y el Desarrollo Educativo*, vol. 8, nº 15, pp. 363-386, 2017.
- [11] Fombona Cadavienco, Javier; Pascual Sevillano, M.ª Ángeles; Universidad de Oviedo, «Beneficios del m-learning en la Educación Superior,» *Educatio Siglo XXI*, vol. 31, nº 2, pp. 211-234, 2013.
- [12] «Inauguración del Centro de Simulación Clínica Avanzada de la Facultad de Medicina,» Gabinete de Comunicación. Universidad de Valladolid, 5 jul. 2021. [En línea]. Disponible en: https://comunicacion.uva.es/es_ES/detalle/Inauguracion-del-Centro-de-Simulacion-Clinica-Avanzada-de-la-Facultad-de-Medicina/. [Último acceso: 20 abril 2022].

- [13] P. Mutchler, A. Doupé, J. Mitchell, C. Kruegel y G. Vigna, «A Large-Scale Study of Mobile Web App Security,» *Conference: Mobile Security Technologies Workshop (MoST)*, may. 2015.
- [14] W. Jobe, «Native Apps Vs. Mobile Web Apps,» *International Journal of Interactive Mobile Technologies (IJIM)*, vol. 7, pp. 27-32, oct. 2013.
- [15] «App nativa vs app híbrida: cuál elegir para el desarrollo de aplicaciones,» Bambú Mobile, dic. 2020. [En línea]. Disponible en: <https://www.bambu-mobile.com/desarrollo-de-aplicaciones-app-nativa-vs-app-hibrida-cuando-usar-cada-una/>. [Último acceso: 20 abril 2022].
- [16] A. Charland y B. Leroux, «Mobile application development: web vs. native,» *Communications of the ACM (Association for Computing Machinery)*, vol. 54, nº 5, pp. 49-53, 2011.
- [17] «Cómo utilizar Play Console,» Ayuda de Play Console. Soporte de Google, 2022. [En línea]. Disponible en: <https://support.google.com/googleplay/android-developer/answer/6112435?hl=es#zippy=%2Cpaso-paga-la-cuota-de-registro>. [Último acceso: 20 abril 2022].
- [18] «Apple Developer Program,» Apple Developer, 2022. [En línea]. Disponible en: <https://developer.apple.com/programs/>. [Último acceso: 20 abril 2022].
- [19] A. Rahman, «A comparative study of hybrid mobile application development,» FOCI - Foundations of Computation and Intelligence, 2020. [En línea]. Disponible en: https://easychair.org/publications/preprint_open/KLPH. [Último acceso: 20 abril 2022].
- [20] M. Báez et al., «Introducción a Android,» V. López y Grupo Tecnología UCM, Edits., E.M.E. Editorial, 2019, p. 3.
- [21] «Qué es Android,» Android, 2022. [En línea]. Disponible en: https://www.android.com/intl/es_es/what-is-android/. [Último acceso: 20 abril 2022].
- [22] «Mobile Operating System Market Share Worldwide. Jan 2022,» Statcounter. GlobalStats, 2022. [En línea]. Disponible en: <https://gs.statcounter.com/os-market-share/mobile/worldwide/#monthly-202201-202201-bar>. [Último acceso: 20 abril 2022].
- [23] «Operating System Market Share Worldwide. Jan 2022,» StatCounter. GlobalStats, 2022. [En línea]. Disponible en: <https://gs.statcounter.com/os-market-share#monthly-202201-202201-bar>. [Último acceso: 20 abril 2022].
- [24] «Android's Kotlin-first approach,» Kotlin. Android Developers, 2021. [En línea]. Disponible en: <https://developer.android.com/kotlin/first>. [Último acceso: 20 abril 2022].
- [25] K. Bradshaw y D. Rousell, «Distribution data for Android,» Android Distribution, nov. 2021. [En línea]. Disponible en: <https://androiddistribution.io/#/>. [Último acceso: 20 abril 2022].
- [26] «Mobile Vendor Market Share Worldwide. Jan 2022,» Statcounter. GlobalStats, 2022. [En línea]. Disponible en: <https://gs.statcounter.com/vendor-market-share/mobile/worldwide/#monthly-202201-202201-bar>. [Último acceso: 20 abril 2022].

- [27] «Meet Android Studio,» Android Studio. Android Developers, 2021. [En línea]. Disponible en: <https://developer.android.com/studio/intro>. [Último acceso: 20 abril 2022].
- [28] «Xcode IDE,» Xcode. Apple Developer, 2022. [En línea]. Disponible en: <https://developer.apple.com/xcode/features/>. [Último acceso: 20 abril 2022].
- [29] «We are ClickUp,» ClickUp, 2022. [En línea]. Disponible en: <https://clickup.com/about>. [Último acceso: 20 abril 2022].
- [30] «Injuries and violence,» World Health Organization, 2021. [En línea]. Disponible en: <https://www.who.int/news-room/fact-sheets/detail/injuries-and-violence>. [Último acceso: 3 mayo 2022].
- [31] S. P. Baker et al., «The injury fact book,» Oxford University Press, 1992.
- [32] Y. Ballesterero Díez, «Manejo del paciente politraumatizado,» *Protocolos y diagnósticos terapéuticos en pediatría. SEUP*, pp. 247-262, 2020.
- [33] National Association of Emergency Medical Technicians (NAEMT), PHTLS. Soporte Vital de Trauma Prehospitalario, 9 ed., Jones and Bartlett Publishers, Inc, 2019.
- [34] G. S. Doyle y P. P. Taillac, «Tourniquets: a review of current use with proposals for expanded prehospital use,» *Prehospital emergency care*, vol. 12, nº 2, pp. 241-256, 2008.
- [35] «Cánula de Guedel,» Enfermanía, [En línea]. Disponible en: <https://www.enfermania.com/canula-de-guedel.html>. [Último acceso: 6 mayo 2022].
- [36] S. Schachter, «Neumotórax y toracocentesis,» *Destrezas médicas para salvar vidas*, [En línea]. Disponible en: http://destrezasmedicasparasalvarvidas.com/capitulo9_celular.html#neumo. [Último acceso: 6 mayo 2022].
- [37] «What is a Lottie?,» LottieFiles, [En línea]. Disponible en: <https://lottiefiles.com/what-is-lottie>. [Último acceso: 15 mayo 2022].
- [38] S. Celis, «SQLite.swift,» GitHub, [En línea]. Disponible en: <https://github.com/stephencelis/SQLite.swift>. [Último acceso: 15 mayo 2022].
- [39] «POI-HSSF and POI-XSSF/SXSSF - Java API To Access Microsoft Excel Format Files,» Apache POI, 2022. [En línea]. Disponible en: <https://poi.apache.org/components/spreadsheet/>. [Último acceso: 15 mayo 2022].
- [40] D. DeLong, «CHCSVParser,» GitHub, 2020. [En línea]. Disponible en: <https://github.com/davedelong/CHCSVParser>. [Último acceso: 15 mayo 2022].
- [41] E. Zuberger, «AlertToast,» GitHub, [En línea]. Disponible en: <https://github.com/elai950/AlertToast>. [Último acceso: 15 mayo 2022].
- [42] «What Is SQLite?,» SQLite, 2022. [En línea]. Disponible en: <https://www.sqlite.org/index.html>. [Último acceso: 15 mayo 2022].