

UNIVERSIDAD DE VALLADOLID

**ESCUELA TÉCNICA SUPERIOR DE
INGENIEROS DE TELECOMUNICACIÓN**

TRABAJO DE FIN DE GRADO

**GRADO EN INGENIERÍA DE TECNOLOGÍAS ESPECÍFICAS
DE TELECOMUNICACIÓN
MENCIÓN EN INGENIERÍA TELEMÁTICA**

**APLICACIÓN iOS DE APOYO A PACIENTES Y
PROFESIONALES DE FISIOTERAPIA**

**AUTOR: ALBERTO MARTÍN MATEOS
TUTOR: MÍRIAM ANTÓN RODRÍGUEZ**



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

APLICACIÓN iOS DE APOYO A PACIENTES Y
PROFESIONALES DE FISIOTERAPIA



TÍTULO: “APLICACIÓN iOS DE APOYO A
PACIENTES Y PROFESIONALES DE
FISIOTERAPIA”

AUTOR: ALBERTO MARTÍN MATEOS

TUTORA: MÍRIAM ANTÓN RODRÍGUEZ

DEPARTAMENTO: TEORÍA DE LA SEÑAL Y
COMUNICACIONES E INGENIERÍA
TELEMÁTICA

Miembros del Tribunal

PRESIDENTE: M^a ÁNGELES PÉREZ JUÁREZ

SECRETARIO: DAVID GONZÁLEZ ORTEGA

VOCAL: MÍRIAM ANTÓN RODRÍGUEZ

SUPLENTE: MARIO MARTÍN ZARZUELA

SUPLENTE: FRANCISCO JAVIER DÍAZ PERNAS

FECHA DE LECTURA:

CALIFICACIÓN:



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA



Resumen del proyecto

Hoy en día, el mundo de la tecnología está creciendo a un ritmo vertiginoso, una situación que provoca que la demanda del uso de las aplicaciones móviles, entre otros sectores, sea uno de los entornos en los que más se está trabajando en estos últimos años.

Este Trabajo Fin de Grado surge de la necesidad de una herramienta que sirva de apoyo a la intercomunicación entre profesionales y pacientes en un área de la salud que necesita un empujón, la fisioterapia. Además, a nivel personal, nace con el reto principal del aprendizaje de uno de los lenguajes de programación más utilizados en la actualidad para el desarrollo de aplicaciones móviles nativas.

La herramienta que se propone para que sirva de soporte para los diferentes centros de fisioterapia es una aplicación móvil iOS, desarrollada en el lenguaje Swift 3. Esta aplicación es una adaptación de la aplicación Android y de la página Web ya implementadas. Se desarrolla con la intención de proporcionar funcionalidades al usuario, similares a las de la aplicación Android, de una manera más dinámica y rápida que si se tuviera que usar navegación Web, destinándola para labores administrativas y de puesta a punto si se produjesen cambios relevantes como, por ejemplo, contratación de nuevos trabajadores, incorporación de nuevos tratamientos, nuevas clínicas o nuevos ejercicios.

La aplicación está ideada para el uso en dispositivo iPhone, a partir de la versión de 4,7 pulgadas (iPhone 5) en adelante, intentando garantizar una interfaz intuitiva y sencilla de manejar, para poder conseguir acercarnos a todos los grupos de la sociedad, sin la necesidad de tener unas nociones avanzadas en el uso de la tecnología móvil.

Palabras claves

Fisioterapia, Xcode, Swift, JSON, PHP, Bases de datos, iPhone, Apple.



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

Abstract

Nowadays, the technological world is growing at a dizzying pace, a situation that causes the demand, among some other sectors, of the mobile applications, together with its growth in recent years as a consequence.

This project emerges from the need of a tool that allows the intercommunication between professionals and patients in one of the health areas that needs a boost in this respect, physiotherapy. In addition, this project is born with the main goal to learn one of the programming languages more used to develop mobile applications nowadays.

The tool proposed to support the physiotherapy centers is an iOS mobile application, developed in Swift 3 programming language. This application is an adaptation of the Android one and the webpage already in place. It is developed with the intention of providing functionalities to the user, similar to those of the Android application, in a more dynamic and fast way. It is as well meant to help the webpage for administrative and set-up tasks, in the event of some relevant changes such as new workers' recruitment, or the incorporation of new treatments, exercises, or clinics.

The application is designed to be used in an iPhone device, from the 4.7 inches' version (iPhone 5) onwards, trying to ensure an intuitive and easy to handle interface, in order to approach more easily to all kind of society groups, being able to overcome the need of advanced notions in the use of mobile technology.

Keywords

Physiotherapy, Xcode, Swift, JSON, PHP, Databases, iPhone, Apple.

APLICACIÓN iOS DE APOYO A PACIENTES Y
PROFESIONALES DE FISIOTERAPIA





APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA



AGRADECIMIENTOS

Agradecer enormemente el apoyo de toda mi familia desde el inicio de la carrera, sin él tengo el convencimiento que no podría haber llegado hasta aquí. Gracias por los valores inculcados durante toda mi vida.

Agradecer a mis amigos su preocupación por mí a lo largo de todos estos años, son una parte importante de lo que soy yo ahora mismo.

Agradecer también el apoyo de todos los profesores, compañeros y amigos de carrera, por estrecharme la mano en cualquier instante.

Por último, dar las gracias a mi tutora Míriam por su ánimo en todos los momentos del proyecto.

A todos, gracias por el sacrificio de vuestro tiempo en mí.

Dedicado a mi abuelo Gerardo.



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

APLICACIONES DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA



“El destino no es cuestión de casualidad, sino cuestión de elección. No es algo a lo que se deba esperar, sino algo a conseguir”.

Jeremy Kiston



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA



Índice de contenidos

Resumen del proyecto	5
Abstract.....	6
Índice de contenidos	13
Capítulo 1: Introducción general	23
1.1 Objetivos	23
1.2 Fases y métodos	24
1.3 Medios	24
1.4 Estructura del documento	25
Capítulo 2: Estudio de mercado	26
2.1 Mercado de la telefonía móvil	26
2.2 Mercado de aplicaciones para la salud	28
Capítulo 3: Estudio de las diferentes tecnologías	34
3.1 Tipo de aplicaciones	35
3.1.1 Aplicaciones nativas.....	35
3.1.1.1 Android.....	35
3.1.1.2 Windows Phone.....	41
3.1.1.3 iOS.....	42
3.1.1.4 Comparativa entre sistemas	46
3.1.2 Aplicaciones web	47
3.1.3 Aplicaciones híbridas.....	48
3.2 Tecnologías del lado del servidor	49
3.2.1 PHP.....	49
3.2.2 Bases de datos	52
3.2.2.1 Bases de datos relacionales.....	52
3.2.2.2.1 MySQL	53
3.2.2.2 Bases de datos orientadas a objetos.....	54
3.2.3 Servicios Web	55
3.2.3.1 SOAP.....	56



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

3.2.3.2 REST	57
3.2.3.3 Comparativa de servicios Web	59
3.3 Elección de la tecnología utilizada	63
Capítulo 4: Descripción técnica de la aplicación	65
4.1 Estructura de la base de datos	65
4.1.1 Tabla clinics	66
4.1.2 Tabla treatments	68
4.1.3 Tabla clinics_treatments.....	69
4.1.4 Tabla users.....	69
4.1.5 Tabla exercises	70
4.1.6 Tabla users_exercises	71
4.1.7 Tabla professionals.....	72
4.1.8 Tabla reservations	73
4.1.9 Tabla news.....	74
4.2 Relación entre las tablas de la base de datos	74
4.2.1 Clinics y treatments	75
4.2.2 Professionals, exercises y users.....	75
4.2.3 Clinics, treatments, reservations y professionals	76
4.2.4 Professionals, reservations y users_Exercise	78
4.2.5 Professionals, reservations, clinics y users.....	79
4.3 Descripción de las funcionalidades y diagramas de flujo.....	80
4.3.1 Clase LoginVC y clase ViewController.....	81
4.3.2 Clase RegisterPageVC	83
4.3.3 Clase UsuarioVC.....	84
4.3.3.1 Clases ClinicasVC, EspecificacionesClinicasVC, ComoLlegarVC y TratamientosVC	85
4.3.3.2 Clases ClinicasVC, EspecificacionesClinicaReservaVC, DatosParaReservarViewController, CalendarioVC y ConsultarCitaVC.....	86
4.3.3.3 Clases ListaEjerUsuarioVC y EspecificacionesEjercicioUsuarioVC	89
4.3.3.4 Clases ListaMisReservasVC y AnularReservaVC	90
4.3.4 Clase ProfesionalVC	91

APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA



4.3.4.1 Clases TratamientosVC y EspecificacionesTratamientoVC	92
4.3.4.2 Clases EjerciciosVC, EspecificacionesEjercicioVC, ListaPacProVC y AsignarEjercicioVC.....	93
4.3.4.3 Clase ListaMisReservasVC y EspecificacionesCitaProfesionalVC.....	96
4.3.5 Clase EncargadoViewController.....	96
4.3.5.1 Clases ListaReservasVC, EspecificacionesReservasVC y ListaProfesionalesVC	97
4.3.5.2 Clases TratamientosVC y EspecificacionesTratamientoVC	99
4.3.5.3 Clases EjerciciosVC, EspecificacionesEjercicioVC, ListaPacProVC y AsignarEjercicioVC.....	100
4.3.5.4 Clases VerReservasVC y AnularCitaVC	101
Capítulo 5: Manual del usuario.....	103
5.1 Login.....	103
5.2 Registro de pacientes	105
5.3 Menú principal del encargado.....	108
5.3.1 Adjudicar paciente.....	109
5.3.2 Asignar ejercicio al paciente	113
5.3.3 Ver tratamientos	118
5.3.4 Ver reservas de las clínicas	119
5.4 Menú principal del profesional	122
5.4.1 Ver tratamientos	123
5.4.2 Asignar ejercicio	123
5.4.3 Contactar con la clínica.	124
5.4.4 Lista de citas.....	125
5.5 Menú principal del usuario/paciente.....	127
5.5.1 Clínicas.....	128
5.5.2 Reservar cita.....	130
5.5.3 Lista de ejercicios del paciente.....	135
5.5.4 Mis reservas.....	137
5.5 Desconexión.....	137
5.6 Vista global de la aplicación	139
Capítulo 6: Presupuesto económico.	141



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

Capítulo 7: Conclusiones, líneas futuras y experiencia personal	142
7.1 Conclusiones	142
7.2 Líneas futuras.....	142
7.3 Experiencia personal.....	144
Capítulo 8: Bibliografía	145
ANEXO: Solución a los problemas con acentos en PHP, MySQL y SWIFT	147



Índice de figuras

Figura 1.	Mercado de la distribución de los sistemas operativos móviles. (Statista, 2017).....	26
Figura 2.	Número de aplicaciones móviles disponibles en cada una de las tiendas móviles 2016 (Statista, 2016).	27
Figura 3.	Mercado de la distribución de los sistemas operativos móviles por continentes 2016 (Expansión, 2016).	28
Figura 4.	Crecimiento de las diferentes categorías de aplicaciones móviles (Statista, 2015).....	29
Figura 5.	Crecimiento del número de app de la salud (Pwc, 2016).....	30
Figura 6.	Crecimiento de ingresos en app de la salud. (TAD, 2016).....	30
Figura 7.	Mercado de las aplicaciones móviles de la salud. (TAD, 2016).	31
Figura 8.	Mejores aplicaciones móviles de la salud 2016 (TAD, 2016).....	33
Figura 9.	Plataformas y lenguajes usados por los desarrolladores (Inmobi, 2016). 34	
Figura 10.	Pantalla principal sistema operativo Android.....	36
Figura 11.	Reparto de las diferentes versiones de los dispositivos Android. (Google, 2016).....	37
Figura 12.	Arquitectura de Android. (Android Developers, 2017).....	38
Figura 13.	Pantalla principal del sistema operativo Windows Phone.....	41
Figura 14.	Pantalla principal sistema operativo iOS.....	42
Figura 15.	Reparto de las diferentes versiones de los dispositivos iOS (Mixpanel, 2016).....	44
Figura 16.	Arquitectura iOS.....	45
Figura 17.	Aplicaciones Web.....	48
Figura 18.	Aplicaciones híbridas.	49
Figura 19.	Logo PHP	50
Figura 20.	Uso de las tecnologías del lado del servidor (W3C, 2017).	51
Figura 21.	Logo MySQL.....	54
Figura 22.	Servicio REST, nivel 3.	58
Figura 23.	Logo Swift	62
Figura 24.	Situación de Swift con respecto al resto de lenguajes (Fernández Muñoz, J.C, 2015). 63	



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

Figura 25.	Relación entre tablas clinics y treatments.	75
Figura 26.	Relación entre tablas users, professionals y exercises	76
Figura 29.	Relación entre las tablas reservations, clinics, professionals y users.	80
Figura 30.	Diagrama de flujo de “Login”.	83
Figura 31.	Diagrama de flujo de “Registro”.	84
Figura 32.	Diagrama de flujo menú principal del usuario/paciente.	85
Figura 33.	Diagrama de flujo “Clínicas”.	86
Figura 34.	Diagrama de flujo “Reservar cita”	89
Figura 35.	Diagrama de flujo “Lista de ejercicios del paciente”.	90
Figura 36.	Diagrama de flujo “Ver reservas del paciente”.	91
Figura 37.	Diagrama de flujo del menú principal del profesional.	92
Figura 38.	Diagrama de flujo “Ver tratamientos”.	93
Figura 39.	Diagrama de flujo “Asignar ejercicio al paciente”.	95
Figura 40.	Diagrama de flujo “Lista de citas”.	96
Figura 41.	Diagrama de flujo del menú principal del encargado.	97
Figura 42.	Diagrama de flujo “Adjudicar paciente”.	99
Figura 43.	Diagrama de flujo “Ver tratamientos (encargado)”	99
Figura 44.	Diagrama de flujo “Asignar ejercicio (encargado)”	100
Figura 45.	Diagrama de flujo “Ver reservas clínicas”.	102
Figura 46.	Pantalla inicial de la aplicación.	103
Figura 47.	Errores a la hora de producirse el logueo en la aplicación (móvil).	104
Figura 48.	Errores a la hora de producirse el logueo en la aplicación.	104
Figura 49.	Pantalla de registro de la aplicación.	105
Figura 50.	Errores en el proceso de registro del paciente.	107
Figura 51.	Errores producidos al introducir datos correctos.	107
Figura 52.	Registro del paciente.	108
Figura 53.	Pantalla de menú principal del encargado.	109
Figura 54.	No hay reservas pendientes de adjudicar.	110
Figura 55.	Lista de reservas sin adjudicar profesional.	110
Figura 56.	Detalles de la reserva.	111
Figura 57.	Lista de profesionales disponibles para adjudicar.	112

APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA



Figura 58.	No hay profesionales disponibles para adjudicar.	112
Figura 59.	Paciente adjudicado al profesional.	113
Figura 60.	Lista de ejercicios disponibles.	114
Figura 61.	Detalles del ejercicio	115
Figura 62.	Compartir ejercicio	115
Figura 64.	Profesional sin pacientes.	116
Figura 65.	Comentario del profesional sobre el ejercicio.	117
Figura 66.	Ejercicio asignado correctamente.	118
Figura 67.	Lista de tratamientos.	118
Figura 68.	Detalles del tratamiento.	119
Figura 69.	No hay reservas en ninguna clínica.	120
Figura 70.	Lista de reservas.	120
Figura 71.	Detalles de la reserva	121
Figura 72.	Anular cita	122
Figura 73.	Cita anulada	122
Figura 74.	Pantalla del menú principal del profesional.	123
Figura 75.	Menú de correo del teléfono.	124
Figura 76.	Mensaje de e-mail enviado.	125
Figura 77.	No hay citas para el profesional.	126
Figura 78.	Lista de citas concertadas.	126
Figura 79.	Detalles de la cita.	127
Figura 80.	Pantalla de menú principal del paciente.	128
Figura 81.	Lista de clínicas	128
Figura 82.	Detalles de la clínica seleccionada.	129
Figura 83.	Cómo llegar a la clínica.	130
Figura 84.	Detallas de la clínica para reservar cita.	131
Figura 85.	Datos para reservar.	132
Figura 87.	Detalles de la reserva.	133
Figura 88.	Fecha y hora de la posible reserva.	134
Figura 89.	Cita reservada.	135
Figura 90.	No hay ejercicios para el paciente.	136



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

Figura 91.	Listado de ejercicios del paciente.....	137
Figura 92.	Detalles del ejercicio.	137
Figura 93.	Encargado desconectado.....	138
Figura 94.	Profesional desconectado.	138
Figura 95.	Paciente desconectado.	139
Figura 96.	“Main StoryBoard” de la aplicación.....	140
Figura 97.	Configuración lenguaje para la conexión al servidor.	148
Figura 98.	Configuración lenguaje para la base de datos.	148
Figura 99.	Configuración lenguaje para las tablas (1).	149
Figura 100.	Configuración lenguaje para las tablas (2).	149
Figura 101.	Configuración lenguaje para los campos.....	150



Índice de tablas

Tabla 1.	Ventajas y desventajas de las aplicaciones nativas.	35
Tabla 2.	Ventajas y desventajas de las aplicaciones web.	48
Tabla 3.	Ventajas y desventajas de las aplicaciones híbridas.	49
Tabla 4.	Información sobre el servicio Web SOAP.	57
Tabla 5.	Información sobre el servicio Web REST.	59
Tabla 6.	Comparativa de servicios Web.	61
Tabla 8.	Tabla clinics.	68
Tabla 9.	Tabla treatments.	69
Tabla 10.	Tabla clinics_treatments.	69
Tabla 11.	Tabla users.	70
Tabla 12.	Tabla exercises.	71
Tabla 13.	Tabla users_exercises.	72
Tabla 14.	Tabla professionals.	73
Tabla 15.	Tabla reservations.	74
Tabla 16.	Tabla news.	74



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA



Capítulo 1: Introducción general

La industria, la economía, la sanidad, las telecomunicaciones, las empresas, el mundo tecnológico, en definitiva, el mundo, está atravesando por la etapa más importante de su historia, una etapa que no se entendería sin la tecnología.

En los últimos años, la introducción de los teléfonos inteligentes o Smartphones en el mercado ha supuesto una verdadera revolución. Actualmente no se concibe un mundo sin el uso de estos dispositivos, desde el momento en el que una persona se levanta hasta el momento en el que se acuesta. Además, si nos centramos en el uso de este tipo de teléfonos en la vida laboral, es aún más grande su repercusión.

La principal ventaja del uso de esta tecnología es la accesibilidad a la gran mayoría de los servicios que se utilizaban en Web. Antes, cuando se quería consultar cualquier tipo de información a través de internet, era necesario estar enfrente de un ordenador con una conexión a la red. En cambio, ahora, gracias a los Smartphones, esta dificultad desaparece y, la mayor parte de la información es alcanzable, bien a través de la navegación Web o a través de aplicaciones móviles.

Analizando los diferentes campos de aplicación de la tecnología, y más en concreto, el de las aplicaciones móviles, se observa que es uno de los que más ha crecido en la última década. Si uno se pone a buscar el motivo, se encontrará con que la principal razón es la de permitir el acceso a los principales recursos de manera más rápida en cualquier lugar y en cualquier momento. No obstante, la principal intención de las aplicaciones móviles es la de servir de complemento a la navegación Web.

1.1 Objetivos

Este Trabajo Fin de Grado se realiza con el objetivo adquirir la mayoría de las habilidades necesarias para el desarrollo y puesta en funcionamiento de aplicaciones en un lenguaje de programación totalmente desconocido al iniciar este proyecto. Por otra parte, se trata de complementar a la plataforma Web y a la aplicación Android, herramientas de apoyo a centros de fisioterapia, que han sido previamente desarrolladas.

El proyecto consiste en el desarrollo de una aplicación iOS que permita la intercomunicación entre pacientes y profesionales y que sirva de complemento a la ya mencionada plataforma Web. Se trata de que permita realizar ciertas funcionalidades como, por ejemplo, reservar una cita o anularla, sin la necesidad de tener que acudir al centro, únicamente usando un dispositivo móvil. Además, se pretende que reúna todas las funcionalidades de la aplicación Android para que, si surgieran líneas futuras, fuera más sencillo añadirlas en ambas aplicaciones.



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

1.2 Fases y métodos

Para lograr los objetivos presentados previamente, en el proyecto hemos seguido las siguientes fases:

- Fase de documentación, en la que se ha recopilado información sobre los objetivos que tenía que reunir la aplicación, sobre el lenguaje de programación en iOS, manipulación y puesta a punto de las bases de datos y sobre la interacción entre los lenguajes JSON y PHP.
- Fase de iniciación y familiarización con el lenguaje de programación Swift 2 y su entorno de desarrollo “*Xcode*” (funcionamiento de la interfaz de vistas) y prueba del entorno “*PlayGround*”, donde se han hecho pruebas de la documentación recopilada previamente.
- Desarrollo de la aplicación en Swift 2 utilizando la herramienta “*Xcode*” junto con los servicios ofrecidos por “*MAMP*”. Posteriormente, se adapta la aplicación a la última versión de Swift, Swift 3.
- Realización de pruebas de la aplicación completa en terminal real y depuración de fallos.
- Extracción de conclusiones y líneas futuras.

1.3 Medios

Para la realización de este proyecto se ha utilizado como referencia algunos Trabajos Fin de Grado relacionados con el desarrollo de aplicaciones móviles.

Se han visualizado videos y leído libros y artículos que estudian el desarrollo de aplicaciones móviles.

Para su implementación se ha dispuesto del siguiente equipo, en el que se incluyen tanto programas como dispositivos físicos:

- Ordenador Mac mini Intel Core i5 @2.6GHz, 8GB de RAM y un 1TB. Sistema operativo Mac OS X, versión 10.11.6, “*El Capitan*”.
 - Empleado para el desarrollo y diseño de la aplicación.
- Teléfono móvil iPhone 5.
 - Utilizado para realizar pruebas de ejecución.
- Xcode 8.1: Entorno de Desarrollo creado por Apple Inc. para el desarrollo de aplicaciones para sistemas operativos Mac OS X, iOS y tvOS.
- Coda 2.5.16: Editor de texto utilizado para la edición de ficheros PHP.
- MAMP 4.0.6: Usa una recopilación de servicios “*MAMP*”, formada por “*Apache*”, servidor Web, “*MySQL*”, servidor de bases de datos y,



“*PHP*”, lenguaje de script de programación que se usa como complemento para acceder a la base de datos.

1.4 Estructura del documento

El documento se ha estructurado de la siguiente forma:

- En el primer capítulo, se realiza una introducción sobre el contenido del trabajo y los objetivos, fases, metodologías y medios utilizados para la realización del proyecto.
- En el segundo capítulo, se hace un estudio del mercado, a nivel de la telefonía móvil y a nivel del sector de la salud.
- En el tercer capítulo, se hace una introducción sobre las tecnologías móviles y los sistemas operativos más destacados. Asimismo, se estudiarán las tecnologías por parte del servidor, bases de datos y servicios Web, finalizando con una comparativa entre los lenguajes más utilizados y una breve explicación de la elección del sistema operativo elegido.
- En el cuarto capítulo, se realiza una descripción técnica de la aplicación, donde se muestra la estructura de la base de datos, las relaciones entre tablas y la descripción de las funcionalidades de la aplicación.
- En el quinto capítulo, se ofrece un manual de usuario de la aplicación, explicando paso a paso todas las pantallas que atraviesa.
- En el sexto capítulo, se muestra un presupuesto económico de la aplicación.
- En el séptimo capítulo, se exponen las conclusiones y las líneas futuras de la aplicación.
- En el octavo capítulo, se recoge la bibliografía empleada al realizar este documento.
- Finalmente, se describen los anexos para la mejor comprensión de la aplicación en algunas partes.

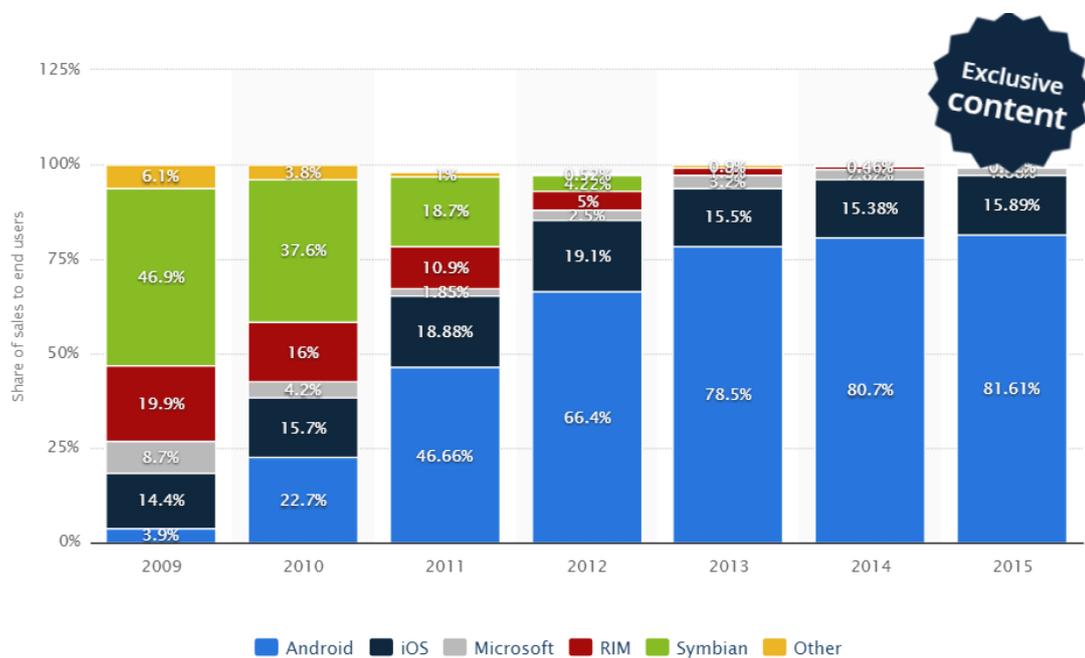


Capítulo 2: Estudio de mercado

2.1 Mercado de la telefonía móvil.

En los últimos años, el mercado de la telefonía móvil ha sido uno de los que más ha aumentado, año tras año se han ido desarrollando nuevas funcionalidades en cada uno de los sistemas operativos que ha hecho que la competencia fuese máxima y la competición por crear el mejor dispositivo fuera constante, día a día.

Hace unos años, en 2009, cuando todavía no había comenzado Android a emerger en el mercado, Symbian, era el sistema operativo dominante, era un sistema operativo propiedad de Nokia y que en el pasado fue propiedad entre otras empresas, de Samsung. Estas dos marcas dominaban el mercado en esos años, por lo que también lo hacía Symbian. Blackberry también tuvo su clientela a partir de ese año con su famoso terminal con teclado incorporado.



© Statista 2017

Figura 1. Mercado de la distribución de los sistemas operativos móviles. (Statista, 2017).



CÁPITULO 2: ESTUDIO DE MERCADO

A partir de 2011, Symbian y Blackberry empezaron a ser atropellado por los dos grandes gigantes del momento. Android iba creciendo exponencialmente año tras año gracias a la incorporación del sistema operativo en los terminales Samsung mientras que iOS iba manteniendo su hueco en el mercado.

A partir de 2013, Android se posiciona como el sistema operativo predominante siguiendo con su crecimiento exponencial mientras iOS se tiene que conformar con ser la alternativa

De manera similar, se puede observar que el predominio de Android frente a iOS se ve reflejado en el índice de descargas, pero la diferencia no es tan dispar como a la hora de ver el sistema operativo utilizado, lo cual indica que el uso de aplicaciones en personas que utilizan iOS es mucho mayor que el uso de aplicaciones por personas que usan Android.

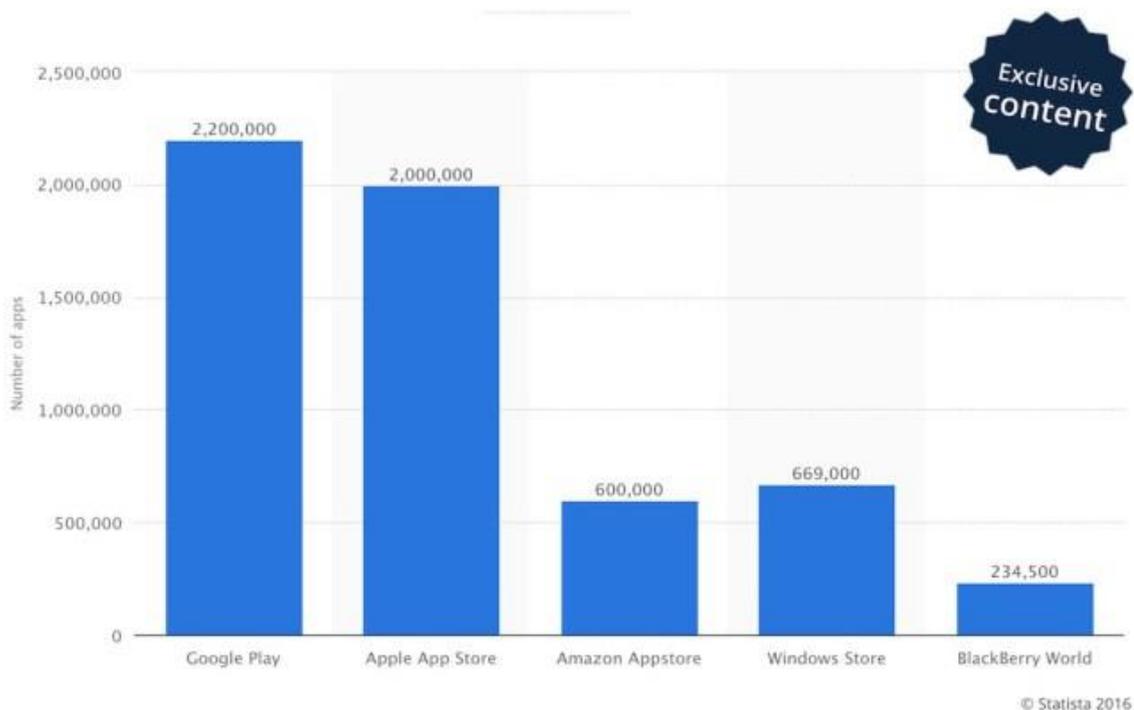


Figura 2. Número de aplicaciones móviles disponibles en cada una de las tiendas móviles 2016 (Statista, 2016).

En cambio, si se analiza desde un punto de vista diferente, se puede observar que la perspectiva indica que, en países como China o Japón, países emergentes en la última década, iOS se puede posicionar como un sistema operativo claramente competitivo en un futuro, al menos, en estos países.



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

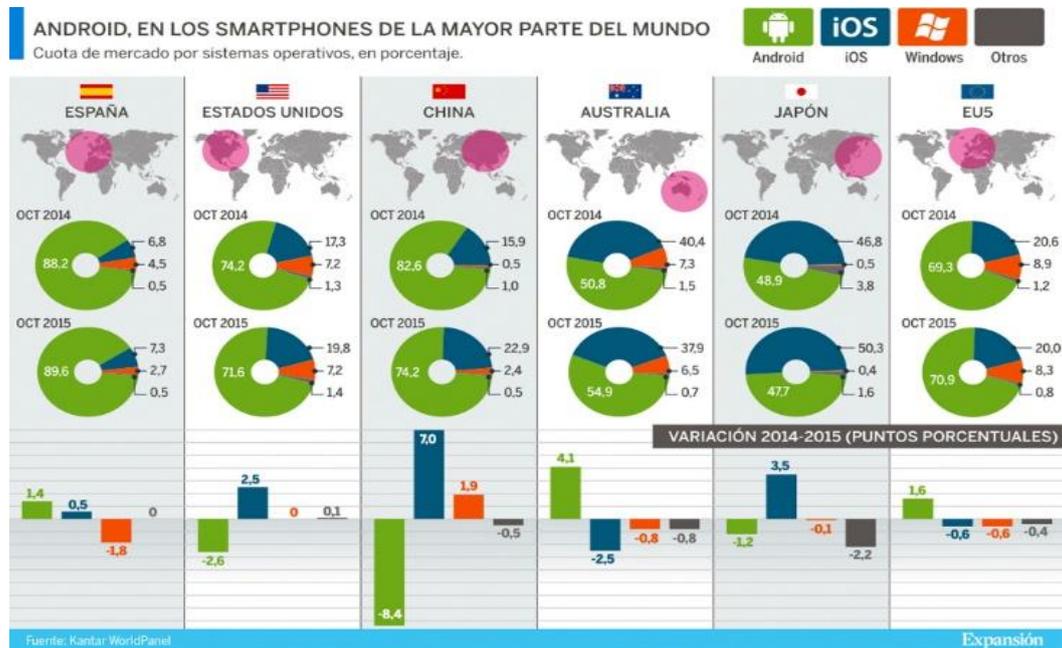


Figura 3. Mercado de la distribución de los sistemas operativos móviles por continentes 2016 (Expansión, 2016).

2.2 Mercado de aplicaciones para la salud

El mercado de las aplicaciones para la salud, las llamadas *mHealth*, está en constante crecimiento en estos últimos tres años, llegando a posicionarse como la tercera categoría más importante, después de los juegos y las utilidades. Así lo avanzaba a principios de 2014 un artículo del diario “El Mundo” (El Mundo, 2015). Además, adelantaba que su crecimiento anual sería de un 23% durante los 5 próximos años, pero las previsiones se cumplieron con creces en el primer año, creciendo más de un 50% respecto al año anterior.



CÁPITULO 2: ESTUDIO DE MERCADO

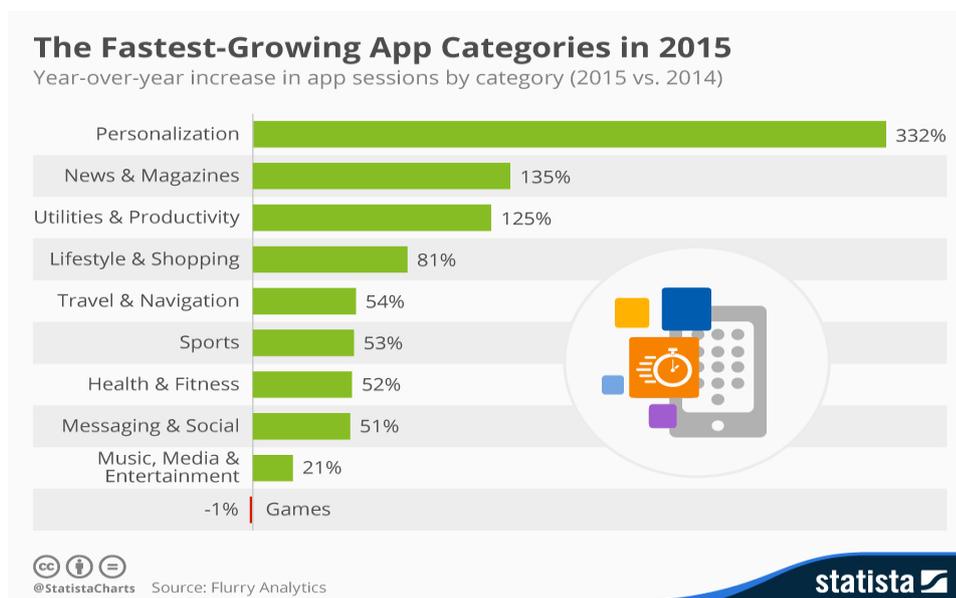


Figura 4. Crecimiento de las diferentes categorías de aplicaciones móviles (Statista, 2015).

El 70% de estas aplicaciones va dirigida al público en general, a través de ejercicios físicos, consejos de bienestar y, el 30% están ideadas para los profesionales sanitarios y los pacientes. La funcionalidad más común a nivel general es la de aportar información (39,8%), seguida de la proveer de instrucciones de uso (21,4%) y registrar o capturar datos del usuario (18,7%).

En el caso concreto de las aplicaciones disponibles en español analizadas en el nuevo estudio, el 24% son informativas, el 22% están dedicadas a la monitorización de parámetros físicos y el 18% las que facilitan el seguimiento del tratamiento.

La educación (16%), la gestión (16%) y la ayuda al diagnóstico (6%) son los ámbitos que cierran las categorías en las que el informe de “The App Date” ha organizado las aplicaciones seleccionadas. Asimismo, el estudio ha comprobado que el 66% de las aplicaciones en español son gratuitas y sólo el 34% de pago (El Mundo, 2015).

En febrero de 2016, se publicó un artículo que corroboraba en cierta manera las previsiones hechas, que el uso de las aplicaciones móviles de salud iba a seguir creciendo a ritmo vertiginoso (Pwc, 2016).

La consultora PwC analizaba en su informe anual “*Top health industry issues of 2016*” las principales tendencias en el sector de la salud para el año 2016 (Pwc, 2016).

Una de las principales conclusiones que destaca es la utilización de apps sanitarias se duplicará en un plazo de dos años. Así, el porcentaje de consumidores con, al menos, una app médica o de fitness en sus móviles pasará del 16% de 2015 a un 32% en 2016. La adopción de este tipo de dispositivos para controlar la tensión arterial, reservar horarios de consulta, ayudar en el cumplimiento de la medicación, realizar



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

seguimiento de enfermedades crónicas o hacer ejercicio, por citar algunas de sus aplicaciones más numerosas, es cada vez mayor y seguirá creciendo en los próximos años.

Mobile health app adoption doubles in two years

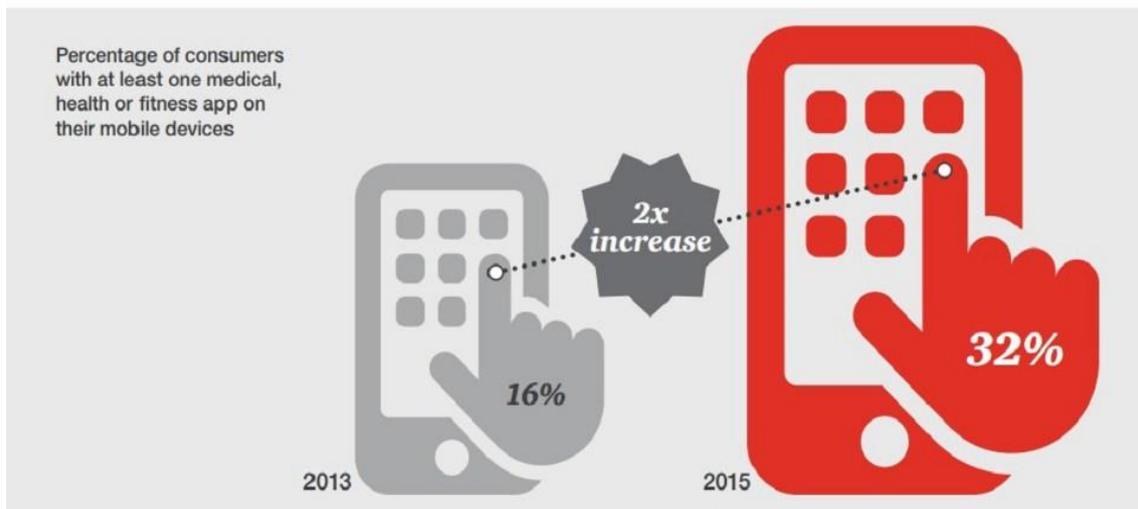


Figura 5. Crecimiento del número de app de la salud (Pwc, 2016).

Sin embargo, de forma paralela al auge de las apps sanitarias, el informe de PWC advierte del riesgo de seguridad que puede entrañar su uso. Este riesgo puede suponer que, en los próximos años, los consumidores puedan tener cierto temor a estas aplicaciones sanitarias por el miedo a ser hackeados. Por este motivo, la industria sanitaria debería invertir en ciberseguridad.



Figura 6. Crecimiento de ingresos en app de la salud. (TAD, 2016).

Según este informe, en apenas 4 años el nivel de ingresos se multiplicará por cinco, alcanzando los 23 millones de dólares en 2017. Las herramientas para



CÁPITULO 2: ESTUDIO DE MERCADO

monitorizar las enfermedades crónicas supondrán el 65% del mercado global de la mHealth, con unos ingresos de 15.000 millones de dólares. La diabetes será el área terapéutica con el mayor potencial de negocio, seguida por las enfermedades cardiovasculares. Por su parte, los servicios de diagnóstico ocuparán el 15% y generarán 3.400 millones de dólares. En tercer lugar, las apps relacionadas con los tratamientos médicos supondrán un 10% del mercado y unos ingresos de 2.300 millones de dólares. Europa y Asia son los continentes que tendrán mayores cuotas de mercado en el futuro próximo, ambas con un 30%, seguidas de EE.UU. y Canadá, con un 28%.

El sistema operativo preferido para estas aplicaciones es iOS con casi la mitad de las aplicaciones desarrolladas llegando a la cifra de 40.000 las apps de la salud y medicina, alcanzando la increíble cifra de 97.000 en este sector juntando las demás tiendas de descarga.

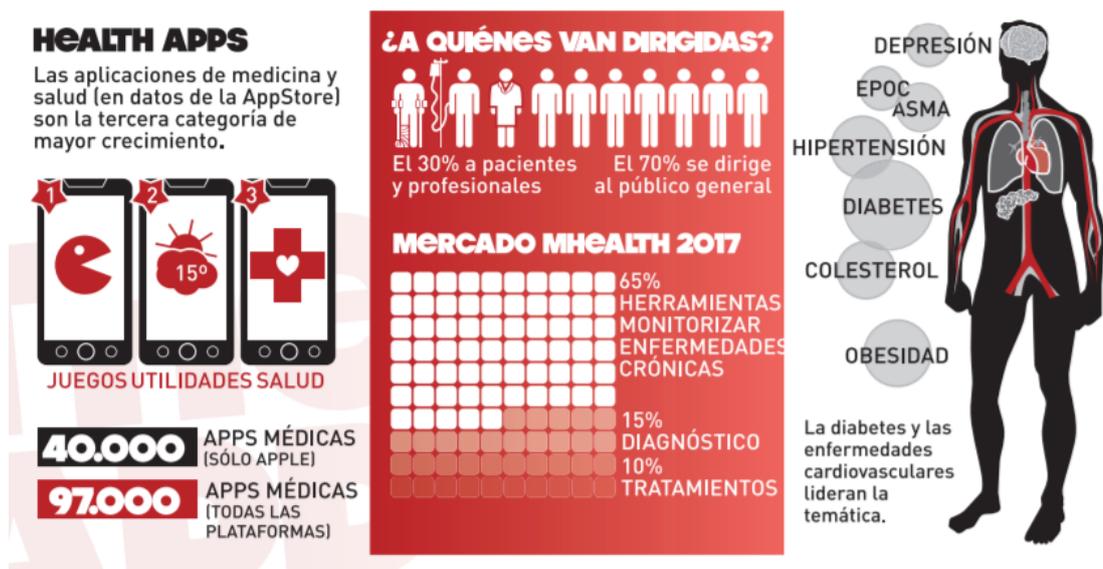


Figura 7. Mercado de las aplicaciones móviles de la salud. (TAD, 2016).

En Europa, según PwC, las aplicaciones de la salud representaran en 2017 una oportunidad que puede rondar los 6.900 millones de dólares y, lo que es más asombroso, es que, según este informe, si la UE se sumergiera de lleno en la salud móvil se podría ahorrar en costes sanitarios nada menos que 99.000 millones de euros. Y se añadirían, además, 93.000 millones al producto interior bruto de la Unión Europea. Concretamente, la utilización de tecnologías móviles podría reducir el coste sanitario per cápita en Europa un 18%, y hasta un 35% en el caso del tratamiento de pacientes crónicos en el año 2017, como señala el último estudio sobre la Sociedad de la Información en España de la Fundación Telefónica (Fundación Telefónica, 2016). Los



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

datos indican, así mismo, que 9,4 millones de europeos en riesgo de padecer enfermedades crónicas podrían acceder a diagnósticos precoces. Lo que queda claro con este tipo de cifras es que las mHealth tienen que tenerse muy en cuenta en estos tiempos.

Por último, si se analiza la situación actual de España, se puede observar que cuenta con una gran ventaja sobre una gran cantidad de países. Según la OMS, España cuenta con uno de los mejores sistemas públicos de salud a nivel mundial, lo que lo sitúa entre uno de los diez mejores en el ranking global (OMS, 2000). Además, para su mejora se ha enfrentado a grandes retos como la sostenibilidad y la incorporación de nuevas tecnologías, como las apps, para mejorar el coste-eficiencia.

Ante la gran cantidad de dispositivos móviles en España, el siguiente paso se concentra en dotar de contenido a estos dispositivos y facilitar aplicaciones que apoyen las necesidades de los usuarios. En lo que se refiere a la salud digital y con los datos que maneja el Ministerio de Sanidad, la eHealth podrían generar en España proyectos con un volumen de negocio de 4.000 millones de euros. En España este sector cuenta, según recoge la GSMA, con cinco principales agentes: pacientes, sistemas médicos, aseguradoras, industria farmacéutica y proveedores de tecnología. A pesar de que España cuenta con una estructura muy positiva para la popularización de las aplicaciones de salud, es un mercado que ha comenzado a arrancar en un corto periodo de tiempo. Nuestro país parte con la ventaja de que el 50% de su población cuenta con la tecnología adecuada y que el grupo de personas situadas en el periodo de edad que va de los 60 a los 80 años ya usa en gran parte smartphones y tabletas y lo hará aún más. Así que, explican desde PwC, serán los propios pacientes los motores de este cambio.

Finalmente, se muestra un resumen de las aplicaciones mejor valoradas en el sector de la salud en España y, como se relacionan ocupan diversos campos de la salud y están orientadas con diversos guiones, pero todos en una misma dirección, complementar la asistencia sanitaria tradicional.



CÁPITULO 2: ESTUDIO DE MERCADO



Nombre: 30 años de VIH
Plataforma: iOS
Categoría: Información
Nº descargas: 10.000
Precio: Gratis
Idioma: ES, EN
Desarrollo: Wake App Health
Publicación: julio 2012
Actualizaciones: 8

DESCRIPCIÓN: Es una guía que ofrece información científica sobre la enfermedad y los avances médicos logrados en este ámbito durante los últimos treinta años. Ofrece una gran experiencia interactiva y está avalada científicamente por GeSIDA (Grupo de Estudio del Sida) y por los doctores Alcamí y Gatell. Ha recibido el Premio de Redacción Médica 2012 a la mejor aplicación sanitaria y el Premio Fundamed-El Global 2012 a la mejor iniciativa 2012.



Nombre: Abla
Plataforma: iOS y Android
Categoría: Seguimiento de tratamiento
Nº descargas: 7.500*
Precio: 14,99€ en iOS y 0,79€ en Android
Idioma: ES, EN, IT, GR, CAT, GAL, FR, PT
Desarrollo: Juan Carlos Gonzalez Montesino
Publicación: 2010
Actualizaciones: en desarrollo

DESCRIPCIÓN: Tiene como principal objetivo mejorar la comunicación en personas con trastornos del lenguaje como autismo, síndrome de down o adultos con ictus o derrames, entre otros. Facilita la interacción de estas personas con sus familiares, terapeutas y entorno social. Ha sido premiada como la mejor aplicación en la categoría de Bienestar en los Smart Accessibility Awards 2012 de la Fundación Vodafone y como la mejor App española 2010 por The App Date.



Nombre: Alerhta
Plataforma: iOS y Android
Categoría: Seguimiento de tratamiento
Nº descargas: 4.000*
Precio: Gratis
Idioma: ES, EN, DE
Desarrollo: Grupo 2- Comunicación Médica
Publicación: 2013
Actualizaciones: 1

DESCRIPCIÓN: Ayuda a los pacientes con hipertensión arterial (HTA) a llevar un control de su enfermedad, a través del registro de sus datos personales, consejos de su médico y alarmas para la toma de medicamentos. Cuenta con el aval de la Sociedad Española de Hipertensión-Liga Española para la Lucha contra la Hipertensión Arterial (SEH-LELHA), el Club del Hipertenso y Conespacar.



Nombre: Babymecum
Plataforma: iOS
Categoría: Registro y monitorización
Nº descargas: 110.000*
Precio: Gratis
Idioma: ES
Desarrollo: Joaquín Luis García Moreno
Publicación: mayo 2012
Actualizaciones: 1

DESCRIPCIÓN: Aporta información sobre la salud del bebé y permite consultar rápidamente la dosis de la mayoría de la medicación pediátrica de uso común, en función de las medidas y el peso del niño. Puede resultar muy útil tanto a padres como a cuidadores. Su contenido ha sido supervisado por pediatras con amplia experiencia en pediatría extrahospitalaria.

Figura 8. Mejores aplicaciones móviles de la salud 2016 (TAD, 2016).



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

Capítulo 3: Estudio de las diferentes tecnologías

La forma de programar aplicaciones es diferente dependiendo del sistema operativo utilizado, aunque también se pueden programar aplicaciones de bajo nivel. Programar aplicaciones de bajo nivel significa que están programadas en un lenguaje no nativo y luego son adaptadas al lenguaje nativo mediante programas auxiliares. Lógicamente, el rendimiento no será el mismo que si esa aplicación ha sido desarrollada en nativo. Este tipo de aplicaciones se les conoce como aplicaciones híbridas.

En los últimos años se habla mucho sobre el oficio de los programadores de aplicaciones móviles. Es un oficio joven en continuo crecimiento, tal y como se ha podido comprobar en el estudio “State of Mobile App Developers 2016” Imobi (Imobi, 2016).

De este estudio se pueden sacar conclusiones muy interesantes a la hora de analizar los diferentes tipos de aplicaciones móviles. Si se atiende a la plataforma de desarrollo, se observa que Android es la preferida para los programadores con un 83%, seguida de iOS con un 57% y de Windows Phone con un 21%. Plataformas como RIM o Symbian han quedado totalmente en el olvido.

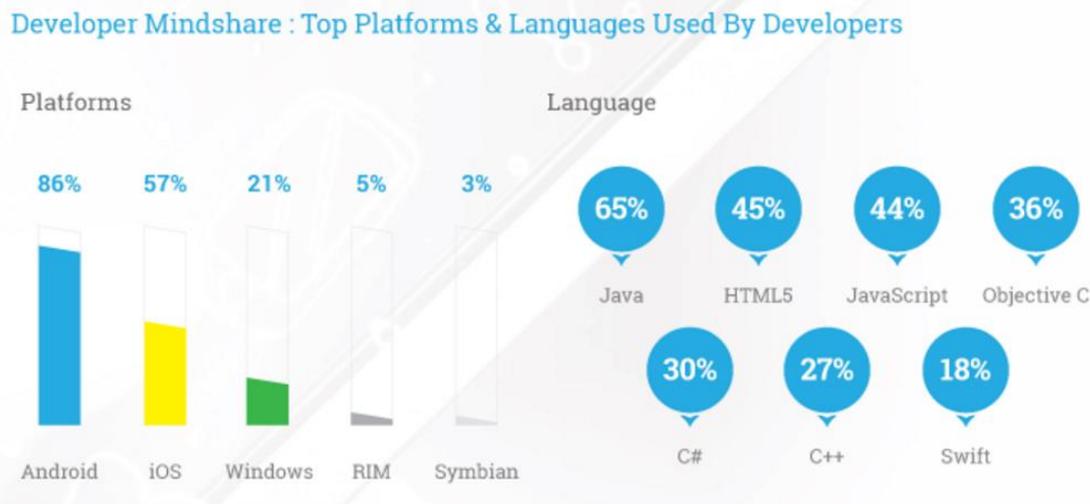


Figura 9. Plataformas y lenguajes usados por los desarrolladores (Inmobi, 2016).

Si ahora se hace hincapié en el lenguaje escogido por los desarrolladores a la hora de programar las aplicaciones, el ganador es Java, el lenguaje de programación para Android. Después, se sitúa HTML5 y Javascript, puesto que son lenguajes utilizados para programar tanto las aplicaciones híbridas anteriormente mencionadas, como las aplicaciones Web. Siguen a estos lenguajes, Objective-C y C#, utilizados para programar aplicaciones nativas para iOS y Windows Phone respectivamente. Por

CÁPITULO 3: ESTUDIO DE LAS DIFERENTES TECNOLOGÍAS



último, se encuentra con menor uso el nuevo lenguaje implementado para iOS, Swift, que se está abriendo un hueco en el mercado desde su creación, y con la idea de ser el lenguaje de desarrollo para iOS sustituyendo a Objective-C.

3.1 Tipo de aplicaciones

Como se ha comentado, las aplicaciones pueden estar implementadas en diferentes lenguajes dependiendo de las necesidades del usuario final. Se estudiarán los distintos tipos a continuación.

3.1.1 Aplicaciones nativas.

Las aplicaciones nativas son un tipo de aplicación que se caracterizan por estar desarrolladas de forma específica para un determinado sistema operativo, son mucho más robustas, estables y veloces que las aplicaciones híbridas, pero tienen el inconveniente de que son más complicadas de desarrollar y su coste es mayor ya que el código no puede ser reutilizado para desarrollar la aplicación en otro lenguaje

Otras ventajas con las que cuenta este tipo de aplicación son la mejor interactividad con el usuario permitiendo la posibilidad de acceder a todas las características del móvil como, por ejemplo, la cámara, GPS, agenda, el uso de notificaciones o la constante actualización.

Ventajas	Desventajas
<ul style="list-style-type: none">• Rendimiento superior.	<ul style="list-style-type: none">• Coste elevado.
<ul style="list-style-type: none">• Buena experiencia del usuario.	<ul style="list-style-type: none">• No reutilización del código.
<ul style="list-style-type: none">• Visibilidad en las apps store.	<ul style="list-style-type: none">• Aprendizaje para un solo sistema operativo.
<ul style="list-style-type: none">• Acceso completo al dispositivo.	
<ul style="list-style-type: none">• Notificaciones.	

Tabla 1. Ventajas y desventajas de las aplicaciones nativas (Lancetalent, 2014).

3.1.1.1 Android



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA



Figura 10. Pantalla principal sistema operativo Android.

Android es un sistema operativo desarrollado por la Open Handset Alliance (OHP), de la que forma parte Google, inicialmente pensado para teléfono móviles, igual que iOS y Windows Phone, además de otros como Symbian y Blackberry OS. Su principal diferencia es que está basado en Linux, un núcleo de sistema operativo libre, gratuito y multiplataforma, algo que le hace diferenciarse totalmente del resto.

En el año 2005, Google adquiere Android Inc, una pequeña compañía que acababa de ser creada, orientada a la producción de aplicaciones para terminales móviles. Ese mismo año empiezan a trabajar en la creación de una máquina virtual Java optimizada para móviles (Dalvik VM). En el año 2007 se crea el consorcio Handset AllianceV con el objetivo de desarrollar estándares abiertos para móviles. Está formado por Google, Intel, Texas Instrumente, Motorola, T-Mobile, Samsung, Ericsson, Toshiba, Vodafone, NTT DoCoMo, Sprint Nextel y otros. Un punto clave de los objetivos de esta

alianza es promover el diseño y difusión de la plataforma Android. Sus miembros se comprometieron a publicar una parte importante de su propiedad intelectual como código abierto bajo licencia Apache v2.0.

En noviembre del 2007 se lanza una primera versión del Android SDK. Al año siguiente aparece el primer móvil con Android (T-Mobile G1). En octubre, Google libera el código fuente de Android, principalmente bajo licencia de código abierto Apache (licencia GPL v2 para el núcleo). Ese mismo mes se abre Android Market para la descarga de aplicaciones.



CÁPITULO 3: ESTUDIO DE LAS DIFERENTES TECNOLOGÍAS

Actualmente, la última distribución introducida por Android es Android Nougat 7.0. Sin embargo, por el momento solo se puede probar en los últimos Nexus, Google Pixel y LG V20. El mercado sigue liderado por la versión KitKat, aunque ya ha sido superada por la suma de las versiones de la distribución Lollipop. Nougat, como todas las versiones, está adaptándose al mercado al ser una versión recién salida y tardará unos meses en llegar a la mayor parte de dispositivos y estabilizarse. Mientras, la última versión estable, Marshmallow, lanzada a principios del año 2016 sigue creciendo mes tras mes.

Version	Codename	API	Distribution
2.2	Froyo	8	0.1%
2.3.3 - 2.3.7	Gingerbread	10	1.3%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	1.3%
4.1.x	Jelly Bean	16	4.9%
4.2.x		17	6.8%
4.3		18	2.0%
4.4	KitKat	19	25.2%
5.0	Lollipop	21	11.3%
5.1		22	22.8%
6.0	Marshmallow	23	24.0%
7.0	Nougat	24	0.3%

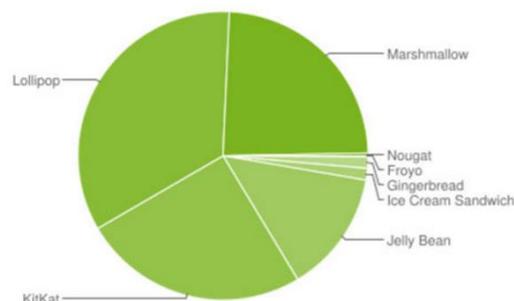


Figura 11. Reparto de las diferentes versiones de los dispositivos Android. (Google, 2016).

El sistema Android en su totalidad consiste en una pila de software que incluye el mencionado sistema operativo basado en Linux, un entorno de ejecución de aplicaciones basado en java, un conjunto de bibliotecas de bajo y medio nivel accesibles por las aplicaciones y un conjunto inicial de aplicaciones para el usuario final. Para desarrollar aplicaciones con este sistema operativo es necesario el Android SDK, Software Development Kit, que aporta las herramientas y APIs necesarias usando como principal lenguaje de programación Java (Rodríguez Cayetano, 2016), (Android Developers, 2017).



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

La arquitectura de Android está formada por cuatro capas, pero cuentan con la gran ventaja de que están basadas en software libre. A continuación, se explican brevemente cada una de ellas:



Figura 12. Arquitectura de Android. (Android Developers, 2017).

- **Applications:** Android incorpora un conjunto de aplicaciones pre-instaladas (teléfono, calendario, mapas, navegador, cliente de correo, contactos, etc..) que se entregan junto con el sistema operativo al cliente. Todas estas aplicaciones han de ejecutarse en la máquina virtual para garantizar la seguridad del sistema. Normalmente, estas aplicaciones están escritas en Java usando Android SDK, aunque es posible su desarrollo en C/C++ usando Android NDK, Native Development Kit.
- **Application framework:** proporciona una plataforma de desarrollo libre para aplicaciones innovadoras y con grandes recursos como pueden ser, sensores, localización, notificaciones, etc...



CÁPITULO 3: ESTUDIO DE LAS DIFERENTES TECNOLOGÍAS

Esta capa ha sido diseñada para simplificar la reutilización de componentes.

Cada aplicación puede publicar sus capacidades para que otras aplicaciones puedan hacer uso de ellas, dado que ofrece soporte para el desarrollo de aplicaciones basadas en Java (compatibilidad con la mayoría de Java JRE).

Los servicios más importantes que incluye son:

- Activity Manager: gestiona el ciclo de vida de aplicaciones.
 - Window Manager: gestiona ventanas de las aplicaciones.
 - Content Providers: permite a las aplicaciones compartir datos con otras (contactos, agenda, mensajes...).
 - View System: elementos para construir GUI (listas, mosaicos, botones, check-boxes...).
 - Package Manager: permite obtener información sobre aplicaciones instaladas en el dispositivo.
 - Telephony Manager: gestiona las funcionalidades del teléfono (llamadas, mensajes...).
 - Resource Manager: gestiona el acceso a recursos.
 - Location Manager: gestión de información de localización.
 - Notification Manager: comunicación de eventos al usuario.
- **Libraries**: bibliotecas escritas en C/C++, y compiladas en código nativo del procesador.
 - **Android runtime**: está basado en el concepto de máquina virtual utilizado en Java. Dadas las limitaciones de los dispositivos en los que ha de ejecutarse Android (poca memoria y un procesador limitado) no es posible utilizar una máquina virtual Java estándar. Google tomó la decisión de crear una nueva, la máquina virtual Dalvik, que respondiera mejor a estas limitaciones.
 - Una característica de la máquina virtual Dalvik que facilita esta optimización de recursos son la ejecución de ficheros Dalvik ejecutables (dex), formato optimizado para ahorrar memoria. Además, está basada en registros. Cada máquina virtual Dalvik delega al kernel de Linux algunas funciones como threading y el manejo de la memoria a bajo nivel. Cada aplicación es un proceso diferenciado que tiene su propia instancia en la máquina Dalvik.
 - **Capa HAL**: La capa de abstracción de Hardware (HAL) brinda interfaces estándares que exponen las capacidades de hardware del dispositivo al framework de la API Java de nivel más alto. Esta capa consiste en varios módulos de biblioteca y cada uno de estos implementa una interfaz para un tipo específico de componente de hardware, como el módulo de la cámara o de bluetooth.
 - **Linux Kernel**: Android se basa en el sistema operativo Linux versión 3.4 (Android 4.4.2) que proporciona servicios básicos como seguridad, gestión de



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

memoria, multiproceso, drivers del dispositivo y pila de protocolos en red. Actúa como capa de abstracción entre el hardware y el resto del software de Android (es la única dependiente del hardware) (Android Developers, 2017).

Android proporciona una gran variedad de ventajas entre las cuales se resaltan las siguientes:

- Software libre y multiplataforma. Cualquier persona que desee programar en este lenguaje lo puede hacer.
- El sistema Android es capaz de hacer funcionar a la vez varias aplicaciones y además se encarga de gestionarlas, dejarlas en modo suspensión si no se utilizan e incluso cerrarlas si llevan un periodo determinado de inactividad.
- Amplia variedad de versiones que permite al usuario tener constantemente actualizado su sistema operativo, siempre y cuando el dispositivo lo soporte.
- Al pertenecer a Google, la interactividad con servicios que ofrece como Hangouts, Gmail es muy beneficiosa
- Gran cantidad de desarrolladores que facilita la retroalimentación de información a programadores noveles.
- Venta en una amplia gama de teléfonos debido a que muchas compañías pertenecen al grupo Open Handset Alliance.
- Amplia variedad de aplicaciones, en su mayor parte de uso gratuito.

Sin embargo, Android también cuenta con una serie de desventajas que son las siguientes:

- Vulnerabilidad. Al ser un software libre, es utilizado por muchos usuarios aprovechando sus fallos de sistema.
- El control de aplicaciones subidas a la tienda de Google hace que haya una gran cantidad de apps sin demasiada utilidad.
- La gestión multitarea también tiene su lado negativo. Al tener varias aplicaciones en segundo plano, el dispositivo se ralentiza y el consumo de batería es superior.
- El problema de tener aplicaciones en segundo plano en ocasiones hace necesario el uso de aplicaciones adicionales que mejoren el rendimiento puesto que el sistema a veces no cierra las aplicaciones correctamente.
- La gran cantidad de actualizaciones de las versiones hace que no todas lleguen a todos los dispositivos al mismo tiempo, de manera que es una desventaja para los desarrolladores que tengan que revisar la misma aplicación para diferentes APIs.

CÁPITULO 3: ESTUDIO DE LAS DIFERENTES TECNOLOGÍAS



3.1.1.2 Windows Phone



Figura 13. Pantalla principal del sistema operativo Windows Phone.

Windows Phone es un sistema operativo para teléfonos móviles desarrollado por la empresa Microsoft. Lanzado al mercado en los últimos meses de 2010 con la finalidad de suplantar al conocido Windows Mobile (Windows Phone, 2016).

Microsoft decidió realizar un cambio completo en su nuevo sistema operativo, presentando una interfaz completamente nueva, mejor comportamiento y un mayor control sobre las plataformas hardware que lo ejecutan, con el principal objetivo de volver a ser competitivo en su sector. Actualmente, todavía no ha alcanzado las dimensiones de los grandes competidores, Android e iOS.

Con esta plataforma, Microsoft llega como una nueva propuesta de movilidad diseñada para integrar y maximizar experiencias a través de la web, la PC y el teléfono,



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

mediante aplicaciones y servicios. Se ha establecido una serie de requisitos mínimos de hardware para que las compañías que manufacturan los teléfonos puedan utilizar Windows Phone 7 en sus equipos.

Todo esto brinda ciertas ventajas a los desarrolladores de aplicaciones pues reduce la fragmentación de la plataforma al no tener una variedad infinita de tamaños, formas y recursos de hardware. Windows Phone se encuentra disponible en los equipos de sus principales socios comerciales, tales como HTC, HP, LG, Toshiba, Sony Ericsson, Samsung, entre otros.

Finalmente, cabe señalar que el Windows Phone presenta incompatibilidad con los Windows Mobile anteriores, los usuarios no serán capaces de actualizar el Windows en su teléfono y por ende deberán comprar uno nuevo con el reciente sistema operativo, una característica que resaltamos como principal desventaja de este nuevo sistema operativo.

3.1.1.3 iOS



Figura 14. Pantalla principal sistema operativo iOS.

iOS (antes iPhone OS) es un sistema operativo para dispositivos móviles desarrollado por Apple Inc. y distribuido exclusivamente para hardware fabricado por

CÁPITULO 3: ESTUDIO DE LAS DIFERENTES TECNOLOGÍAS



Apple. Es el sistema operativo sobre el que funcionan muchos de los productos móviles de la compañía.

Al igual que Android, 2005 fue el año en el que comenzó la historia de Apple en el mundo de la telefonía móvil. Steve Jobs, cofundador y presidente ejecutivo de Apple Inc, comenzó a planificar el iPhone, aunque su decisión final fue cuanto menos llamativa. Tuvo que elegir entre “encoger el Mac, lo que sería un prodigio de la ingeniería, o aumentar el tamaño del iPod”. Jobs prefería la primera opción, pero enfrentó a los equipos de Macintosh (liderado por Scott Forstall) e iPod (dirigido por Tony Fadell) en una competición interna. Finalmente, Forstall fue el ganador de la misma, creando el iPhone OS.

Esta decisión permitió establecer el éxito del iPhone como una plataforma para desarrolladores externos: usando un sistema operativo bien conocido como base y permitiendo que muchos de los desarrolladores externos de Mac pudiesen escribir software para el iPhone con un mínimo re-entrenamiento. Forstall también fue el responsable de crear los primeros SDK (Software Development Kits) para que los desarrolladores pudiesen crear apps para el nuevo teléfono, así como la App Store dentro de iTunes para permitir su distribución.

Inicialmente, no se soportaban las aplicaciones creadas por terceros, según el razonamiento de Jobs, los desarrolladores podrían construir aplicaciones Web que “se comportasen como aplicaciones nativas en iPhone”. Sin embargo, el 17 de octubre de 2007 Apple anunció que un SDK nativo estaba en fase desarrollo y que habían planeado ponerlo en las manos de los desarrolladores en febrero. En marzo de 2008, Apple lanzó la primera beta del SDK, así como un nuevo nombre para el sistema operativo: “iPhone OS”.

El increíble éxito de este trinomio ha permitido a Apple llegar a ser la mayor empresa en capitalización bursátil del mundo, y su modelo de negocio ha impuesto la norma en el mundo del software actual. En 2010, Apple renombra a iPhone OS como iOS. (Apple, 2016)

Hasta el momento, la última versión lanzada por iOS es iOS 10, presentada al mercado a principios del mes de septiembre del año 2016. Ha sido la versión con adopción más rápida por los usuarios en la historia de iOS. Escasamente un mes fue lo que necesitó, como se muestra en la siguiente figura, para situarse como la versión más utilizada en los teléfonos iPhone, coincidiendo también con la presentación del último iPhone lanzado al mercado por Apple, iPhone 7, que ya venía con la versión instalada. Si es verdad que las primeras betas de esta nueva versión contaron con algunos problemas, pero actualmente la versión ya ha sido acogida por el 81.02% de los usuarios. (Apple5x1, 2016).



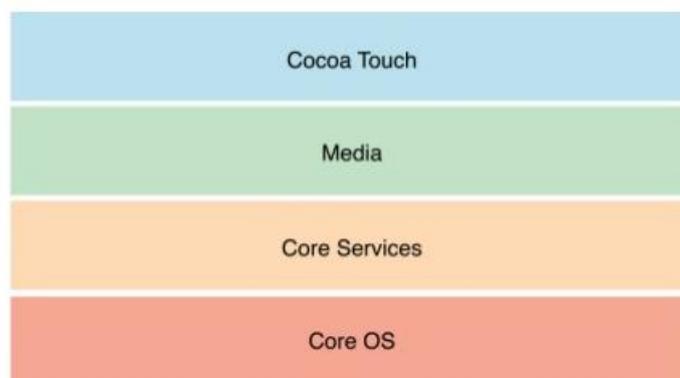
APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA



Figura 15. Reparto de las diferentes versiones de los dispositivos iOS (Mixpanel, 2016).

Los lenguajes utilizados para desarrollar aplicaciones nativas para este sistema operativo son Objective-C, cuyo predecesor es C, un lenguaje estructurado muy utilizado a lo largo de la historia, y Swift, su sucesor. Éste primero ya está siendo absorbido por su sucesor ya que ofrece un entorno mucho más comprensible para un programador novel y su adaptación resulta más sencilla por su similitud en varios puntos con otros lenguajes. Actualmente sigue siendo mejorado por Apple, ofreciendo su tercera versión, Swift 3.

iOS es un derivado de Mac OS X, que a su vez está basado en Darwin BSD, derivado de Unix. La arquitectura de iOS se puede dividir en cuatro capas tal y como se puede apreciar en la siguiente figura. Las capas inferiores ofrecen servicios de bajo nivel, mientras que las superiores contienen servicios y tecnologías mucho más sofisticados. (Antón Rodríguez, 2016), (Apple, 2016).



Capas de iOS

CÁPITULO 3: ESTUDIO DE LAS DIFERENTES TECNOLOGÍAS



Figura 16. Arquitectura iOS.

- Cocoa Touch: capa de más alto nivel de iOS que contiene un conjunto de frameworks orientados a objetos que permiten el desarrollo de aplicaciones nativas. Dentro de esta capa se destacan los siguientes servicios:
 - Multi-tasking.
 - Jerarquía de vistas.
 - Soporte de cámara.
 - Widgets.
 - MapKit framework.
 - Eventos multi-touch.
 - Localización.
 - Internalización.
- Media: capa que tiene las tecnologías de video, audio e imágenes para implementar los servicios orientados a multimedia. Se destacan:
 - Reproducción de video.
 - Formatos de archivos de imágenes.
 - Core Image: manipulación de video e imágenes.
 - OpenGL: framework para manipular gráficos 3D.
 - Quartz: framework para manipular gráficos 2D.
- Core services: capa que recopila los servicios fundamentales para las aplicaciones. Se destacan:
 - Base de datos SQLite.
 - Networking.
 - Soporte XML.
 - Core Location: framework que permite la localización del dispositivo utilizando el hardware disponible.
 - iCloud.
 - Core Foundation: Interfaces escritas en C para la gestión de datos.
- Core OS: actúa como núcleo del sistema operativo. Controla el sistema de memoria virtual, los ficheros, la red e interprocesa con los marcos de la capa superior. Implementa:
 - Gestión de batería.
 - Sockets.
 - TCP/IP.
 - Seguridad.
 - Autenticación.

iOS cuenta con una serie de ventajas en las que se destacan las siguientes:

- Sincronización de todos los dispositivos de Apple, una de las características bandera de este sistema operativo.



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

- Gran calidad en la oferta de sus aplicaciones. Éstas tienen que pasar por numerosos filtros manuales antes de estar disponible en su App Store.
- Existe una gran variedad de aplicaciones.
- Elevada duración de la batería gracias a la buena optimización de sus terminales al contar con una arquitectura particular exclusiva para los dispositivos Apple.
- Alto número de desarrolladores que favorece la retroalimentación para los programadores noveles.
- Elaboración con los mejores materiales del mercado.
- Interfaz intuitiva, iOS focaliza sus elementos en mejorar la experiencia del usuario.

Por el contrario, iOS también cuenta con una serie de desventajas entre las que destacan las siguientes:

- Precio elevado. Al contar con los mejores materiales del mercado y focalizar todos sus esfuerzos en terminales exclusivos, hace que contar con uno de estos dispositivos no esté al alcance de cualquier usuario.
- Objective-C. Cuenta con un lenguaje de programación totalmente diferente al resto, poco intuitivo y difícil de usar. Cabe destacar que se ha mejorado esta experiencia con Swift, su sucesor.
- Para poder desarrollar aplicaciones en este sistema operativo, el único entorno de desarrollo disponible es Xcode, el cual es propiedad de Apple. Para poder instalarlo es necesario tener un ordenador con el sistema operativo Mac OS X y una cuenta de Apple para acceder a la App Store, método necesario para descargarlo. Esto hace que para una persona sea mucho más complicado aprender este lenguaje de programación.
- Los desarrolladores necesitan abonar actualmente una cuota de 99\$ anuales para contar con las herramientas de desarrollo y poder subir aplicaciones a la tienda. Los desarrolladores que abonen esta cuenta entrarán a formar parte del iPhone Developer Program.
- Apple no permite modificar la API de cualquier componente de su framework, lo que resta libertad y capacidad de innovación al desarrollador.

3.1.1.4 Comparativa entre sistemas



CÁPITULO 3: ESTUDIO DE LAS DIFERENTES TECNOLOGÍAS

Tras el análisis de cada uno de los sistemas operativos líderes en el desarrollo de aplicaciones móviles nativas se pueden sacar ciertas conclusiones que hagan que desde diferentes puntos de vista decantarse por uno o por otro sea la mejor opción.

Android cuenta con la gran ventaja de contar con un software abierto, ofreciendo una gran variedad de alternativas tanto a usuarios, al tener una amplia oferta de dispositivos disponibles, como a desarrolladores, puesto que al ser un software multiplataforma es posible programarlo en cualquier sistema operativo y de forma gratuita, al contrario que en Windows Phone y iOS, donde es necesario una cuenta de desarrollador para el lanzamiento en sus tiendas.

Por otra parte, iOS al tener una oferta de dispositivos mínima permite tener con sus usuarios un gran acercamiento y mejorar siempre su experiencia, facilitando cada una de sus versiones de manera casi inmediata a todos sus clientes. Esto no ocurre en el caso de Android, la gran oferta de dispositivos hace que las actualizaciones no lleguen al mismo tiempo a todos los dispositivos ocasionando una gran fragmentación entre las últimas versiones ofrecidas. Ésta situación actúa de manera negativa en el ámbito del programador, que se ve casi obligado a la comprobación de la aplicación en la que trabaja para la mayor parte de sus APIs. Además, iOS ofrece una mejor calidad en su mercado de aplicaciones, punto muy a favor de este sistema operativo que puede prevenir grandes errores en aspectos de seguridad. Android no ofrece este punto en sus aplicaciones y apenas tiene controles.

Windows Phone es un sistema operativo que ha intentado renovarse desde cero para entrar en el mercado con fuerza, pero no ha alcanzado los niveles de Android e iOS, aunque presenta unas grandes innovaciones en sus actualizaciones como su integración total con Windows.

Por el momento, la oferta que presenta Windows Phone es muy pequeña respecto a los grandes competidores, por lo que se puede concluir diciendo que dependiendo de las pretensiones de cada cliente iOS o Android puede ser una buena opción.

3.1.2 Aplicaciones web

Una aplicación web es la desarrollada con lenguajes muy conocidos por los programadores, como es el HTML, CSS y JavaScript. La principal ventaja con respecto a la nativa es la posibilidad de programar independiente del sistema operativo en el que se usará la aplicación. De esta forma se pueden ejecutar en diferentes dispositivos sin tener que crear varias aplicaciones.



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA



Figura 17. Aplicaciones Web

Las aplicaciones web se ejecutan dentro del propio navegador web del dispositivo a través de una URL. El contenido se adapta a la pantalla adquiriendo un aspecto de navegación APP.

La pregunta es si se pueden considerar aplicaciones, ya que no necesita instalación porque se ejecutan en el mismo navegador y no pueden ser descargadas desde ninguna app store. De todas formas, se puede crear un acceso directo que sería como “instalar” la aplicación en el dispositivo.

Las apps web móviles son siempre una buena opción si nuestro objetivo es adaptar la web a formato móvil.

Ventajas	Desventajas
<ul style="list-style-type: none"> • Código reutilizable. 	<ul style="list-style-type: none"> • Conexión a internet.
<ul style="list-style-type: none"> • Desarrollo más sencillo. 	<ul style="list-style-type: none"> • Acceso limitado al hardware.
<ul style="list-style-type: none"> • Sin aprobación externa para su publicación. 	<ul style="list-style-type: none"> • Peor experiencia de usuario.
<ul style="list-style-type: none"> • Pueden utilizarse sitios diseñados. 	<ul style="list-style-type: none"> • Mayor esfuerzo publicitario.
<ul style="list-style-type: none"> • Siempre disponible la última versión. 	

Tabla 2. Ventajas y desventajas de las aplicaciones web (Lancetalent, 2014).

3.1.3 Aplicaciones híbridas.

Una aplicación híbrida es una combinación de las dos anteriores, se podría decir que recoge lo mejor de cada una de ellas. Las apps híbridas se desarrollan con lenguajes propios de las “WebApp”, es decir, HTML, Javascript y CSS, por lo que permite su uso en diferentes plataformas, pero también dan la posibilidad de acceder a gran parte de las características del hardware del dispositivo. La principal ventaja es que, a pesar de estar desarrollada con HTML, Java o CSS, es posible agrupar los códigos y distribuirla en

CÁPITULO 3: ESTUDIO DE LAS DIFERENTES TECNOLOGÍAS



app store. Por el contrario, este tipo de aplicaciones suele ser menos robusta y la experiencia del usuario suele ser peor que en una aplicación nativa.



Figura 18. Aplicaciones híbridas.

PhoneGap es uno de los frameworks más utilizados por los programadores para el desarrollo multiplataforma de aplicaciones híbridas.

Ventajas	Desventajas
<ul style="list-style-type: none">• Posible distribución en todas las tiendas.	<ul style="list-style-type: none">• Peor experiencia de usuario
<ul style="list-style-type: none">• Mismo código para varias aplicaciones.	<ul style="list-style-type: none">• Peor rendimiento que en nativo
<ul style="list-style-type: none">• Acceso a varias características del hardware	

Tabla 3. Ventajas y desventajas de las aplicaciones híbridas (Lancetalent, 2014).

3.2 Tecnologías del lado del servidor

Se llaman tecnologías del lado del servidor a aquellas que consisten en el procesamiento de una petición de un usuario mediante la interpretación de un script en un servidor y producen una respuesta que le llega al cliente. El principal lenguaje utilizado del lado del servidor es PHP, si bien hay otros que pueden usarse como ASP (Microsoft Active Server Pages) o JSP (Java Server Pages).

3.2.1 PHP



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA



Figura 19. Logo PHP

PHP (PHP Hypertext Preprocesor) es un lenguaje de programación interpretado, cuyos comandos se ejecutan en el servidor y permiten la creación de documentos HTML dinámicos. Su sintaxis es similar a la de otros lenguajes como C, Perl, Java o JavaScript (Antón Rodríguez & Pérez Juárez, 2015), (PHP, 2016).

PHP es un producto de código abierto, lo que quiere decir que se puede acceder a su código, usarlo, modificarlo y distribuirlo de forma gratuita sin que suponga coste alguno, al contrario de lo que ocurre con los productos comerciales. Esta es, sin duda, una de las principales ventajas de PHP, ya que la comunidad de usuarios que desarrollan mejoras para el sistema es enorme, lo que proporciona a PHP una vitalidad y capacidad de adaptación que no poseen otros lenguajes de programación.

PHP se encuentra disponible para muchas plataformas incluyendo Windows, Unix o Linux y con la ventaja de que las aplicaciones desarrolladas en PHP se pueden transportar de una plataforma a otra sin necesidad de modificaciones, es decir que PHP presenta una portabilidad elevada. A pesar de ser en la actualidad un lenguaje multiplataforma, debe comentarse que su entorno nativo es Unix/Linux. Con respecto al servidor web también existirían diferentes opciones, ya que podría usarse por ejemplo IIS (Internet Information Server) de Microsoft o Apache, este último con la ventaja de que, al igual que PHP, es también un producto de libre distribución.

Entre las ventajas de PHP se encuentra también la de permitir la conexión y el acceso a multitud de sistemas gestores de bases de datos como MySQL, PostgreSQL u Oracle.

PHP fue concebido en 1994 por Rasmus Lerdorf con la intención de crear un contador para averiguar el número de visitas que recibía su CV virtual. Sin embargo, con el tiempo ha sido adoptado por otros desarrolladores que lo han transformado y convertido en la herramienta que es hoy en la actualidad.

PHP es el lenguaje de programación del lado del servidor más utilizado con un 82.3% (diciembre 2016) de los sitios web donde se conoce la tecnología utilizada, como adelantamos antes, según la web especialista en encuestas <https://w3techs.com/> (W3Techs - World Wide Web Technology Surveys), tal y como puede observarse en la siguiente figura.

CÁPITULO 3: ESTUDIO DE LAS DIFERENTES TECNOLOGÍAS



Most popular server-side programming languages

© W3Techs.com	usage	change since 1 November 2016
1. PHP	82.3%	+0.1%
2. ASP.NET	15.4%	-0.2%
3. Java	2.7%	
4. static files	1.5%	
5. ColdFusion	0.7%	

percentages of sites

Figura 20. Uso de las tecnologías del lado del servidor (W3C, 2017).

Algunos de los sitios web donde se usan este lenguaje son:

- Facebook
- Tuenti
- OScommerce
- WordPress
- MediaWiki
- phpMyAdmin
- Moodle

Este lenguaje ha sido el utilizado para la aplicación desarrollada en este Trabajo Fin de Grado. El funcionamiento es claro, la aplicación solicita información sobre alguna de las funcionalidades. Acto seguido, se consulta el script PHP correspondiente al que se le pasan los parámetros necesarios y una vez obtenidos estos, se consulta la base de datos oportuna y se ofrece al cliente la información solicitada en formato JSON, para su interpretación en el lenguaje del cliente.

Las enormes ventajas que ofrece este lenguaje son:

- Software libre.
- Código abierto.
- Multiplataforma.
- Fácil comprensión y nivel de aprendizaje.
- Gratuito.
- Alto rendimiento, capaz de soportar millones de accesos diarios.
- Comunidad muy grande.
- Interacción con la gran totalidad de las bases de datos.
- No requiere ningún tipo de licencia.



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

3.2.2 Bases de datos

Una base de datos es una entidad en la cual se pueden almacenar datos interrelacionados, con la menor redundancia posible y de manera que se pueda acceder a esos datos de manera eficiente por varias aplicaciones y usuarios.

Para acceder a estos datos se utilizan Sistemas Gestores de Bases de Datos (SGBD) o Data Base Management System (DBMS), en textos ingleses. Son programas o conjuntos de programas cuya función es gestionar a través de su acceso a la base de datos la información en la misma, para realizar operaciones sobre la información contenida en la misma relacionadas con su recuperación, modificación, almacenamiento o eliminación.

Por lo tanto, no se puede confundir los dos conceptos. La base de datos es como un archivador de información y el SGBD es el programa que gestiona ese archivador de datos.

Existen varios modelos de bases de datos según la organización lógica de sus datos:

- Bases de datos jerárquicas.
- Bases de datos en red.
- Bases de datos relacionales.
- Bases de datos orientadas a objetos.
- Bases de datos multidimensionales.

3.2.2.1 Bases de datos relacionales

Una base de datos relacional es una colección de elementos de datos organizados en un conjunto de tablas formalmente descritas desde la que se puede acceder a los datos o volver a montarlos de muchas maneras posibles sin la necesidad de reorganizar las tablas de la base. Fue inventada por E.F.Cood en IBM en 1970 y sigue el modelo relacional, el más utilizado actualmente.

Más específicamente, podemos describir a una base de datos relacional como un conjunto de tablas bidimensionales interrelacionadas que contienen datos provistos en categorías predefinidas. Cada tabla contiene una o más categorías de datos en columnas (o campos). Cada fila (o registro) contiene una instancia única de datos para las categorías definidas en columnas. Para la identificación de las filas se requiere del uso de claves. Una clave es el campo o combinación de campos usados para identificar un registro. Hay dos tipos de claves: principales o primarias y, externas, secundarias, ajenas o foráneas.



CÁPITULO 3: ESTUDIO DE LAS DIFERENTES TECNOLOGÍAS

La interfaz de programa de usuario y de aplicación a una base de datos relacional es el lenguaje de consultas estructuradas (SQL). Los comandos SQL se utilizan tanto para consultas interactivas para obtener información de una base de datos como para la recopilación de datos. Además, son relativamente fáciles de utilizar. Después de su creación original, la posibilidad de añadir nuevas categorías es muy sencilla, sin necesidad de que todas las aplicaciones tengan que ser modificadas.

La base de datos relacional presenta grandes ventajas que hacen que sean las más utilizadas en la actualidad. Se destacan las siguientes:

- Ofrecen sistemas simples y eficaces para la manipulación de datos.
- Provee herramientas que garantizan evitar el duplicado de registros.
- Garantiza la integridad referencial.
- Al eliminar un registro elimina todos los registros relacionados dependientes (independencia lógica).
- Favorece la normalización por ser más comprensible y aplicable.
- Integridad de los datos.
- Uniformidad en cuanto a la manipulación de la base de datos por parte de los usuarios.
- Sencilla a la hora de ser utilizada por el usuario.

También cabe comentar sus puntos débiles, aunque no sean tan relevantes como sus puntos fuertes.

- Presentan deficiencias con datos gráficos, multimedia y sistemas de información geográfica.
- No se manejan igual los bloques de texto que los tipos de dato. Las bases de datos orientadas a objetos se propusieron para satisfacer esa necesidad y complementar a las bases de datos relacionales.
- Pueden promover problemas de islas de información al ser un modelo tan fácil de utilizar. Muchas personas crean subconjuntos y aplicaciones de la base de datos.

3.2.2.2.1 MySQL



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA



Figura 21. Logo MySQL

MySQL es un sistema de gestión de bases de datos relacionales o SGBD. Es un sistema multiusuario y multihilo, lo que le permite ser utilizado por varias personas a la vez realizando diferentes consultas, un sistema muy versátil (MySQL, 2017).

Nació como un software libre y se sigue usando como tal para usos particulares. No obstante, si se quiere utilizar para promover datos en una empresa, se puede comprar una licencia, como software de propietario, que es una autoría de la empresa patrocinante, Oracle Corporation.

La mayor parte del código está escrito en lenguaje C/C++ y la sintaxis de su uso es bastante simple, lo que permite crear bases de datos con facilidad. Además de esto, es compatible con la mayoría de las plataformas informáticas y ofrece una gran cantidad de aplicaciones que permiten acceder rápidamente a las sentencias del gestor de bases de datos.

Su uso mayoritario es en desarrollo web, puesto que permite a los programadores realizar cambios con relativa facilidad sin necesidad de cambiar el código web. Es debido a que MySQL trabaja con un sistema centralizado de gestión de bases de datos que permite realizar cambios en un solo fichero y que se ejecuta en toda la estructura de datos que se comparte en la red.

Pero la forma más potente de utilizar este SGBD es en combinación con PHP, ya que se aprovecha para realizar aplicaciones cliente/servidor (como la realizada en este TFG) que requieren de una base de datos rápida y segura. Entre los competidores principales de MySQL, se puede citar a PostgreSQL,

Microsoft SQL Server y Oracle, aunque MySQL es mucho más consistente por las razones expuestas previamente.

3.2.2.2 Bases de datos orientadas a objetos

Es un modelo más reciente, trata de almacenar en la base de datos los objetos completos (estado y comportamiento). La información que contienen se organiza en atributos y el comportamiento en operaciones. Es un modelo complementario al modelo relacional.

CÁPITULO 3: ESTUDIO DE LAS DIFERENTES TECNOLOGÍAS



Las bases de datos orientadas a objetos manipulan datos complejos de manera rápida, además de tener una gran capacidad en el modelado, puesto que un objeto puede encapsular un estado como un comportamiento, puede almacenar todas las relaciones que tenga y puede agruparse para formar objetos complejos. Esto ofrece la posibilidad de construir datos a partir de los que ya existen y reutilizar las clases existentes.

El lenguaje de consulta es más expresivo que SQL, ya que SGDBOO es una forma común para acceder a los datos, mientras en las SQL es un acceso asociativo. Además, frente a las bases de datos relacionales, presentan mejoras de rendimiento significativas.

3.2.3 Servicios Web

Los servicios Web se definen como sistemas software diseñados para soportar una interacción interoperable máquina a máquina sobre una red. Los servicios Web suelen ser APIs Web que pueden ser accedidas dentro de una red y son ejecutados en el sistema que los aloja (W3C, 2016). Su principal característica es su gran interoperabilidad y extensibilidad, así como por proporcionar información fácilmente procesable por las máquinas gracias al uso de mensajes. Los servicios Web pueden combinarse con muy bajo acoplamiento para conseguir la realización de operaciones complejas. De esta forma, las aplicaciones que proporcionan servicios simples pueden interactuar con otras para conceder servicios sofisticados añadidos.

Un Servicio Web deben tener las siguientes características:

- Debe poder ser accesible a través de la Web. Para ello debe utilizar protocolos de transporte estándares como HTTP, y codificar los mensajes en un lenguaje estándar que pueda conocer cualquier cliente que quiera utilizar el servicio.
- Debe contener una descripción de sí mismo. De esta forma, una aplicación podrá saber cuál es la función de un determinado servicio Web, y cuál es su interfaz, de manera que pueda ser utilizado de forma automática por cualquier aplicación, sin la intervención del usuario.
- Debe poder ser localizado. Debemos tener algún mecanismo que nos permita encontrar un servicio Web que realice una determinada función. De esta forma tendremos la posibilidad de que una aplicación localice el servicio que necesite de forma automática, sin tener que conocerlo previamente el usuario.



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

A nivel técnico, los servicios pueden implementarse de varias formas. Se van a distinguir dos grandes tipos de servicios Web, SOAP y REST, que se describirán con detalle a continuación.

3.2.3.1 SOAP

Los servicios Web SOAP utilizan mensajes XML para comunicarse siguiendo el estándar SOAP (*Simple Object Access Protocol*), un lenguaje XML que define la arquitectura y formato de los mensajes. Dichos sistemas normalmente contienen una descripción legible por la máquina de la descripción de las operaciones ofrecidas por el servicio, escrita en WSDL (*Web Service Description Language*), un lenguaje basado en XML para definir interfaces sintácticamente.

El formato de mensaje SOAP y el lenguaje de definición de interfaces WSDL se ha extendido bastante, y muchas herramientas de desarrollo, por ejemplo, Netbeans, pueden reducir la complejidad de desarrollar aplicaciones de servicios Web.

El diseño de un servicio basado en SOAP debe establecer un contrato formal para describir la interfaz que ofrece el servicio Web. WSDL puede utilizarse para describir los detalles del contrato, que pueden incluir mensajes, operaciones, *bindings*, y la localización del servicio Web.

A continuación, se describe con más detalle las funcionalidades de este servicio Web y sus ventajas y desventajas:

Características	<ul style="list-style-type: none">• Las operaciones son definidas como puertos WSDL.• Dirección única para todas las operaciones.• Múltiples instancias del proceso comparten la misma operación.• Componentes fuertemente acoplados.
Ventajas	<ul style="list-style-type: none">• Fácil de utilizar normalmente.• Depuración posible.• Las operaciones complejas pueden ser ocultas.• Envolver APIs existentes es sencillo.• Incrementa la privacidad.• Tiene herramientas de desarrollo.

CÁPITULO 3: ESTUDIO DE LAS DIFERENTES TECNOLOGÍAS



Desventajas	<ul style="list-style-type: none">• Los clientes necesitan saber las operaciones y la semántica antes del uso.• Los clientes necesitan puertos dedicados para diferentes tipos de notificaciones.• Las instancias del proceso son creadas implícitamente.
-------------	---

Tabla 4. Información sobre el servicio Web SOAP.

3.2.3.2 REST

REST o Representational State Transfer es un estilo de arquitectura a la hora de realizar una comunicación entre un cliente y un servidor.

Habitualmente cuando se realiza una comunicación cliente y servidor, se accede al servidor en un punto de acceso, se envía una información y se espera una respuesta. Esta comunicación es una gran necesidad, por lo que es necesario que sea abierta y se pueda acceder desde cualquier sitio, por lo que lo ideal es una comunicación HTTP.

Una vez se sabe el protocolo de comunicación el siguiente paso es decidir que tipología de mensajes se envía. Como punto de partida, se podrá mandar a un servicio un mensaje en formato XML o JSON (usado en nuestra aplicación). El servicio lo recibirá y nos devolverá una respuesta. Esto es lo que habitualmente en arquitecturas REST se denomina el nivel 0, en una estructura de 4 niveles. No hay ningún tipo de organización. Es el caos (ArquitecturaJava, 2016).

El siguiente paso, el nivel 1, en vez de tener servicios con métodos diversos se declaran recursos. Un recurso hace referencia a un concepto importante del negocio en que se está trabajando. Este estilo permite aportar al nivel 0, el nivel del caos, una cierta organización permitiendo acceder a cada uno de los recursos de forma independiente, favoreciendo la reutilización y aumentando la flexibilidad.

Por el momento, se pueden hacer peticiones sin ningún tipo de norma. En el nivel 2 las operaciones pasan a ser categorizadas de forma más estricta. Es el nivel en el que están muchas de las arquitecturas REST. Dependiendo de cada tipo de operación se utilizará un método diferente de envío.

- GET: Se usa para solicitar consultar a los recursos
- POST: Se usa para insertar nuevos recursos
- PUT: Se usa para actualizar recursos
- DELETE: Se usa para borrar recursos



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

Por último, tenemos el nivel 3, HATEOAS (Hypertext As The Engine Of Application State). Este nivel es más complicado, por lo que se hará uso de un ejemplo como apoyo.

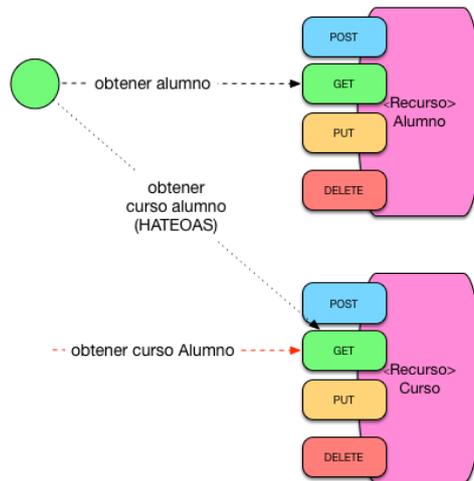


Figura 22. Servicio REST, nivel 3.

Supongamos que queremos acceder a un recurso Alumno vía REST. Si tenemos una arquitectura a nivel 2, primero accederemos a ese recurso utilizando GET. En segundo lugar, deberemos acceder al recurso de Cursos para añadir al alumno al curso (línea roja).

Esto implica que el cliente que accede a los servicios REST asume un acoplamiento muy alto, debe conocer la url del Alumno y la del Curso. Sin embargo, si el recurso del Alumno contiene un link al recurso de Curso esto no hará falta.

Podríamos tener una estructura JSON, como la utilizada en la aplicación, con la siguiente estructura:

```
{nombre:pedro, apellidos:"gomez", cursos:"http://miurl/cursos"}
```

De esta manera, se podrá acceder directamente al recurso de "Curso" utilizando las propiedades del "Alumno". De esta forma se aumenta la flexibilidad y se reduce el acoplamiento.

Para finalizar este apartado, se comentarán las principales características de este servicio Web además de sus ventajas y sus desventajas.

Características	<ul style="list-style-type: none">• Las operaciones se definen en los mensajes.• Una dirección única para cada instancia del proceso.
-----------------	--

CÁPITULO 3: ESTUDIO DE LAS DIFERENTES TECNOLOGÍAS



	<ul style="list-style-type: none"> • Cada objeto soporta las operaciones estándares definidas. • Componentes débilmente acoplados
Ventajas	<ul style="list-style-type: none"> • Bajo consumo de recursos. • Las instancias del proceso son creadas explícitamente. • El cliente no necesita información de enrutamiento a partir de la URI inicial. • Los clientes pueden tener una interfaz “listener” (escuchadora) genérica para las notificaciones • Generalmente fácil de construir y adoptar.
Desventajas	<ul style="list-style-type: none"> • Gran número de objetos. • Manejar el espacio de nombres (URIs) puede ser engorroso. • La descripción sintáctica/semántica muy informal (orientada al usuario). • Pocas herramientas de desarrollo

Tabla 5. Información sobre el servicio Web REST.

3.2.3.3 Comparativa de servicios Web

A continuación, se hace una comparativa de las diferencias entre REST Y SOAP desde distintos puntos de vista.

	REST	SOAP
Tecnología	<ol style="list-style-type: none"> 1. Interacción dirigida por el usuario por medio de formularios. 2. Pocas operaciones con muchos recursos. 	<ol style="list-style-type: none"> 1. Flujo de eventos orquestados. 2. Muchas operaciones para pocos recursos.



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

	<ol style="list-style-type: none"> 3. Mecanismo consistente de nombrado de recursos. 4. Se centra en la escalabilidad y el rendimiento para sistemas distribuidos. 	<ol style="list-style-type: none"> 3. Falta de mecanismo de nombrado. 4. Se centra en el diseño de aplicaciones distribuidas.
Protocolo	<ol style="list-style-type: none"> 1. XML, JSON. 2. HTTP/HTTPS. 3. HTTP es un protocolo de aplicación. 4. Síncrono. 	<ol style="list-style-type: none"> 1. Tipado fuerte, XML Schema. 2. Independiente del transporte. 3. HTTP es un protocolo de transporte. 4. Síncrono y asíncrono.
Descripción del servicio	<ol style="list-style-type: none"> 1. Confía en documentos orientados al usuario que define las direcciones de las peticiones y las respuestas. 2. Interactuar con el servicio supone horas de testado y depuración. 3. No es necesario el tipado fuerte, si ambos extremos están de acuerdo en el contenido. 	<ol style="list-style-type: none"> 1. WSDL. 2. Se pueden construir automáticamente clientes por medio de WDSL. 3. Tipado fuerte.
Gestión del estado	<ol style="list-style-type: none"> 1. Los servidores no tienen estado. 2. Los recursos contienen datos y enlaces representando transiciones a estados válidos. 3. Los clientes mantienen el estado siguiendo los enlaces. 4. Técnicas para añadir sesiones: Cookies. 	<ol style="list-style-type: none"> 1. El servidor puede mantener el estado de la conversación. 2. Los mensajes solo tienen datos. 3. Los clientes mantienen el estado suponiendo el estado del servicio. 4. Técnicas para añadir sesiones: Cabecera de sesión (no estándar).
	<ol style="list-style-type: none"> 1. HTTPS. 	<ol style="list-style-type: none"> 1. WS-Security.

CÁPITULO 3: ESTUDIO DE LAS DIFERENTES TECNOLOGÍAS



Seguridad	<ol style="list-style-type: none"> 2. Comunicaciones punto a punto seguras. 	<ol style="list-style-type: none"> 2. Comunicación origen a destino seguras.
Metodología del diseño	<ol style="list-style-type: none"> 1. Identificar recursos a ser expuestos como servicios. 2. Definir URLs para direccionarlos. 3. Distinguir los recursos de solo lectura (GET) de los modificables (POST, PUT, DELETE). 4. Implementar e implantar el servicio Web. 	<ol style="list-style-type: none"> 1. Listar las operaciones del servicio en el documento WDSL. 2. Definir un modelo de datos para el contenido de los mensajes. 3. Elegir un protocolo de transporte apropiado y definir las correspondientes políticas de QoS y seguridad. 4. Implementar e implantar el contenedor del servicio Web.

Tabla 6. Comparativa de servicios Web.

3.3 Lenguajes en el cliente

En este apartado se analizan los diferentes lenguajes de programación que es posible utilizar para desarrollar una aplicación nativa para un dispositivo con un sistema operativo iOS. En concreto, se analizará una versión global de Swift, el lenguaje de cliente utilizado, comentando ciertas comparaciones con el otro lenguaje de cliente que se permite utilizar para este sistema operativo, Objective-C, lenguaje original y predecesor de Swift.

3.3.1 Swift



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA



Figura 23. Logo Swift

Apple define Swift como un lenguaje que une lo mejor de C y Objective-C, pero sin las restricciones de la compatibilidad con C. Un lenguaje moderno y flexible que introduce nuevos conceptos de programación aparecidos en los últimos años con multitud de lenguajes de scripting como Ruby o Python u otros más complejos como C# o Haskell (Swift, 2017).

Se podría decir que Swift es a Apple lo que C# es a Microsoft, su propia versión de C que es el lenguaje que en mayor medida ha definido los actuales flujos y modelos de algoritmos. Aunque en este caso Apple ha añadido una buena porción de lo más destacado de otros lenguajes de gran éxito entre los desarrolladores como Python o Haskell.

Los tres pilares fundamentales que definen Swift en su esencia son que es potente, moderno y seguro.

Es potente porque su compilación a código intermedio está mucho más optimizada que Objective-C y requiere de menos capas de lenguaje para llegar hasta este estado, consiguiendo también una transición a código máquina más rápida y efectiva para los mismos flujos que Objective-C. Swift, como tal, es compilado con menor peso de capas de interpretación a código intermedio y dicho resultado a su vez conecta de manera más óptima al hardware que soporta OS X, iOS, watchOS o tvOS. Cuanto menor es el recorrido entre programa y hardware, mejor es el rendimiento y mayor la potencia.

Uno de los factores de la mejor optimización de Swift frente a Objective-C es que, como súper clase de C, depende de este en su estructura y eso resta rendimiento y aumenta la cantidad de código que se genera y ha de ejecutarse a nivel más bajo de la máquina o al generar el código intermedio.

Los lenguajes de programación, por lo general, suelen situarse en dos posibles polos. Uno de ellos es el de la facilidad: cualquier lenguaje de script es sencillo y con muchas posibilidades, pero se sitúa a mucha distancia del hardware y el código máquina, debido a que su facilidad repercute en que la traducción de este requiere de muchas capas de compilación, traducción, ejecución o generación de código intermedio, según el caso.

Son lenguajes más sencillos a la hora de ser entendidos y manejados, pero requieren de mucho proceso y pasos para llegar al código máquina: un ejemplo de esto son lenguajes como Ruby, Python, Javascript o Lua, productivos y eficientes, pero a su

CÁPITULO 3: ESTUDIO DE LAS DIFERENTES TECNOLOGÍAS



vez con el problema de ser poco potentes en cuanto a rendimiento por la gran cantidad de capas de compilación que requieren hasta llegar al hardware que los ejecuta.

El otro polo es el que ocupan lenguajes como Objective-C, C, C# o Java que son potentes, pero requieren de mucho más código, componentes, estructuras y son más complejos a la hora de conseguir los resultados, derivado todo ello de la potencia que tienen como tal. Se pegan más al código máquina, son más eficientes, pero más pesados y complejos.



Figura 24. Situación de Swift con respecto al resto de lenguajes (Fernández Muñoz, J.C, 2015).

Swift, sin embargo, se sitúa en una posición privilegiada con respecto al resto de lenguajes con todas las ventajas de un lenguaje de scripting en cuanto a su facilidad, versatilidad y posibilidades, pero a la vez es un lenguaje con la potencia de los más potentes y muy pegado al código máquina. Tiene todas las ventajas de ambos polos y ninguno de sus inconvenientes (Fernández Muñoz, J.C, 2015).

Para finalizar, hay que comentar Swift se encuentra en su tercera versión actualmente, Swift 3.

3.4 Elección de la tecnología utilizada

A la hora de afrontar este proyecto se estudiaron las necesidades en la sociedad y en el mercado actual de aplicaciones relacionadas con el sector de la salud. Tras comprobar que este tipo de aplicaciones estaban en demanda, se buscó participar en un proyecto relacionado con este ámbito comentado.

Como ya se ha mencionado al principio, la fisioterapia es una rama de la salud que necesita un empujón. Por tanto, se optó por ser participe en un Trabajo Fin de Grado relacionado.

Surge un proyecto que se ajusta perfectamente con la idea que se tenía. Se trata de completar un proyecto que se había puesto en marcha hace unos años. Este proyecto se basa en proporcionar una herramienta para los centros de fisioterapia que sirva para



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

complementar las labores de los profesionales además de mejorar la intercomunicación con sus pacientes. Para este proyecto la aplicación en Android ya estaba implementada por lo que definitivamente se decidió por hacer una adaptación para el sistema operativo iOS.

A la hora de elegir la tecnología utilizada para implementar la aplicación, se tuvo en cuenta el reparto actual del mercado de la telefonía móvil y la consistencia que requería la aplicación. Después de comprobar que los líderes en el sector son Android e iOS y que la aplicación necesitaba un gran potencial, con un buen rendimiento, una buena experiencia de usuario, ya que se trata de complementar las labores de diarias de los usuarios, la posibilidad de usar las funciones del dispositivo y una mínima frecuencia de fallo, se opta por desarrollar una aplicación nativa que permite todos estos requisitos.

Por otra parte, una vez ya elegido el sistema operativo, se decidió implementar la aplicación en Swift en vez de Objective-C. La elección fue simple, además de que Swift es el sucesor de Objective-C, es un lenguaje con una estructura parecida a otros lenguajes muy extendidos y estudiados, por lo que, presumiblemente, el aprendizaje y la adaptación serían más rápidos.

Para comunicarse con la base de datos, se decidió utilizar el paquete MAMP 4.0.6. Usa una recopilación de servicios, formada por “Apache”, servidor Web, “MySQL”, servidor de bases de datos y, “PHP”. La interacción con el usuario es muy dinámica y mediante la herramienta web “phpMyAdmin”, permite realizar diferentes operaciones de una manera sencilla.

Para finalizar, hay que mencionar que la respuesta de la base de datos se hace utilizando servicios web REST, en formato JSON que, posteriormente será interpretado de manera eficiente por el cliente.



Capítulo 4: Descripción técnica de la aplicación

En este capítulo se realizará una descripción técnica de la aplicación, en la que se comentarán los aspectos fundamentales de la base de datos con la que se ha trabajado y la función que desempeña cada uno de los campos de las distintas tablas. Para dicha descripción técnica, se hará uso de diferentes imágenes y tablas de la base de datos, además de diagramas de flujo que tratarán de maximizar la explicación de cada una de las funcionalidades de la aplicación.

A continuación, se explicarán los aspectos relacionados con la base de datos externa, así como la relación entre tablas, tomando como guion el seguido por una de las personas que participó en el diseño de dicha base de datos (Gómez Tejedor, David, 2015).

4.1 Estructura de la base de datos

En este apartado, se muestra la estructura de la base de datos utilizada, tanto para las aplicaciones móviles desarrolladas en los diferentes lenguajes como para la Web, además de describir de forma detallada cada una de las tablas con las que se ha trabajado. Esta base de datos sigue una estructura relacional y las tablas se vinculan entre sí.



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

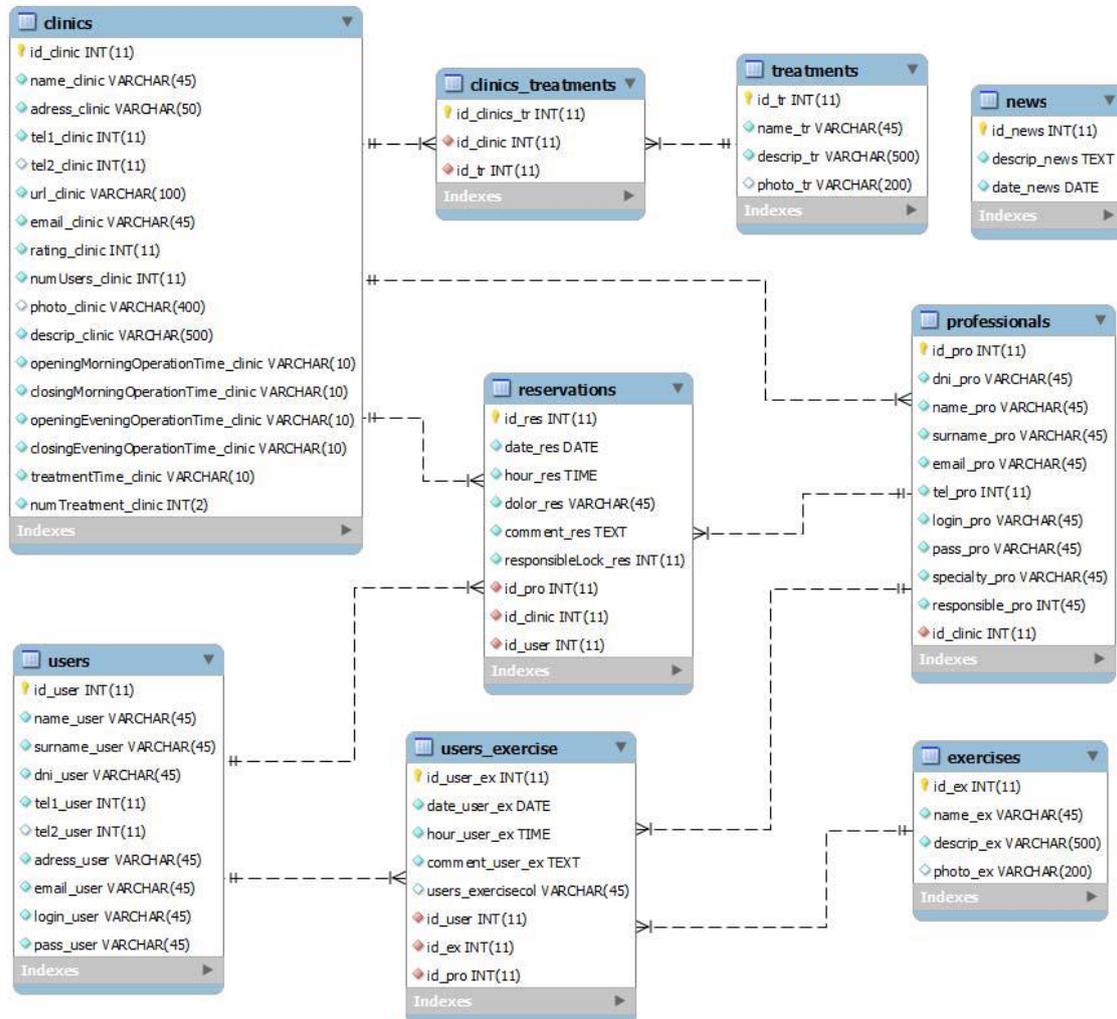


Tabla 7. Estructura de la base de datos.

4.1.1 Tabla clinics

Esta tabla contiene toda la información necesaria relacionada con una clínica, como puede ser, por ejemplo, un número de contacto o una dirección de correo electrónico, datos de interés para un paciente, o bien, el número de tratamientos que tiene la clínica o el número de profesionales, datos de interés para el funcionamiento propio de una clínica.

A continuación, se describe cada uno de los campos de dicha tabla:

- id_clinic: identificador único de cada clínica, auto-incremental.
- name_clinic: indica el nombre de la clínica.
- adress_clinic: indica la dirección de la clínica.
- tel1_clinic: indica el teléfono principal de la clínica.
- tel2_clinic: indica un teléfono secundario de la clínica.



CÁPITULO 4: DESCRIPCIÓN TÉCNICA DE LA APLICACIÓN

- url_clinic: indica la página Web de la clínica
- email_clinic: indica el correo electrónico de la empresa.
- rating_clinic: indica, en las aplicaciones móviles en forma de RatingBar, la valoración de cada clínica.
- numUsers_clinics: indica el número de usuarios que han reservado en la clínica, necesario para establecer el rating.
- photo_clinic: indica la “url” donde se ubica la imagen principal de la clínica, no se almacena dicha imagen en la base de datos.
- descrip_clinic: indica una breve descripción de los servicios que ofrecen las clínicas a los pacientes.
- opening(Morning, Evening)OperationTime_clinic: indica la hora en la que la clínica inicia la apertura, tanto por la mañana (Morning), como por la tarde (Evening).
- closing(Morning, Evening)OperationTime_clinic: indica la hora en la que la clínica cierra, tanto por la mañana (Morning), como por la tarde (Evening).
- treatmentTime_clinic: indica el tiempo aproximado que tarda un profesional en realizar un tratamiento a un cliente.
- numTreatment_clinic: indica el número de tratamientos que realiza una clínica en una misma franja horaria.



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

Tabla Clínicas				
Campo	Tipo	Nulo	Predeterminada	Extra
id_clinic	int (11)	No	ninguna	auto_increment
name_clinic	varchar (45)	No	ninguna	
adress_clinic	varchar (50)	No	ninguna	
tel1_clinic	int (11)	No	ninguna	
tel2_clinic	int (11)	No	<i>Null</i>	
url_clinic	varchar (100)	No	ninguna	
email_clinic	varchar (45)	No	ninguna	
rating_clinic	int (11)	No	ninguna	
numUsers_clinic	int (11)	No	ninguna	
photo_clinic	varchar (400)	No	<i>Null</i>	
descrip_clinic	varchar (500)	No	ninguna	
openingMorningOperationTime_clinic	varchar (10)	No	ninguna	
closingMorningOperationTime_clinic	varchar (10)	No	ninguna	
openingEveningOperationTime_clinic	varchar (10)	No	ninguna	
ClosingEveningOperationTime_clinic	varchar (10)	No	ninguna	
treatmentTime_clinic	varchar (10)	No	ninguna	
numTreatment_clinic	int (11)	No	ninguna	

Tabla 8. Tabla clinics.

4.1.2 Tabla treatments

Esta tabla contiene toda la información de un tratamiento que se realiza en cada una de las diferentes clínicas.

A continuación, se describe cada uno de los campos de dicha tabla:

- id_tr: identificador único de cada tratamiento, auto-incremental
- name_tr: indica el nombre del tratamiento.
- descrip_tr: indica una breve descripción del tratamiento.
- photo_tr: indica la “url” donde se ubica la imagen del tratamiento, no se almacena dicha imagen en la base de datos.

CÁPITULO 4: DESCRIPCIÓN TÉCNICA DE LA APLICACIÓN



Tabla treatments				
Campo	Tipo	Nulo	Predeterminada	Extra
id_tr	int (11)	No	ninguna	auto_increment
name_tr	varchar (45)	No	ninguna	
descrip_tr	varchar (500)	No	ninguna	
photo_tr	varchar (200)	No	<i>Null</i>	

Tabla 9. Tabla treatments.

4.1.3 Tabla clinics_treatments

Esta tabla se crea porque es necesario una relación de varios a varios entre las tablas clinics y treatments, es decir, una clínica puede tener varios tratamientos, y un tratamiento puede estar implantado en varias clínicas.

A continuación, se describe cada uno de los campos de dicha tabla:

- id_clinic_tr: identificador único de cada relación entre una clínica y un tratamiento que se ofrece en ella, auto-incremental
- id_clinic: identificador de la clínica al que pertenece.
- id_tr: identificador del tratamiento.

Tabla clinics_treatments				
Campo	Tipo	Nulo	Predeterminada	Extra
id_clinic_tr	int (11)	No	ninguna	auto_increment
id_clinic	int (11)	No	ninguna	
id_tr	int (11)	No	ninguna	

Tabla 10. Tabla clinics_treatments.

4.1.4 Tabla users

Esta tabla contiene toda la información necesaria sobre un paciente para que éstos puedan realizar todas las funcionalidades que la aplicación le ofrece además de contener información relevante para los encargados de administrar la aplicación.

A continuación, se describe cada uno de los campos de dicha tabla:

- id_user: identificador único de cada paciente, auto-incremental.
- name_user: indica el nombre del paciente.
- surname_user: indica los apellidos del paciente.



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

- dni_user: indica el documento nacional de identidad del paciente.
- tel1_user: indica el teléfono principal de contacto del paciente.
- tel2_user: indica el segundo teléfono de contacto del paciente.
- adress_user: indica la dirección del domicilio del paciente.
- email_user: indica el correo electrónico del paciente.
- login_user: indica el login de cada paciente, diferente para cada paciente y no puede haber un login_user igual a ningún login_pro(login de los profesionales de las clínicas).
- pass_user: indica la clave de acceso de cada paciente a la aplicación.

Tabla users				
Campo	Tipo	Nulo	Predeterminada	Extra
id_user	int (11)	No	ninguna	auto_increment
name_user	varchar (45)	No	ninguna	
sumame_user	varchar (45)	No	ninguna	
dni_user	varchar (45)	No	ninguna	
tel1_user	int (11)	No	ninguna	
tel2_user	int (11)	No	Null	
adress_user	varchar (45)	No	ninguna	
email_user	varchar (45)	No	ninguna	
login_user	varchar (45)	No	ninguna	
pass_user	varchar (45)	No	ninguna	

Tabla 11. Tabla users

4.1.5 Tabla exercises

Esta tabla contiene los datos necesarios para describir cada uno de los ejercicios que tienen que realizar los pacientes de la aplicación o asignar los profesionales de las clínicas.

A continuación, se describe cada uno de los campos de dicha tabla:

- id_ex: identificador único de cada ejercicio, auto-incremental.
- name_ex: indica el nombre del ejercicio.
- descrip_ex: indica una breve descripción del ejercicio.
- photo_ex: indica la “url” donde se ubica la imagen del ejercicio, no se almacena dicha imagen en la base de datos.

CÁPITULO 4: DESCRIPCIÓN TÉCNICA DE LA APLICACIÓN



Tabla exercises				
Campo	Tipo	Nulo	Predeterminada	Extra
id_ex	int (11)	No	ninguna	auto_increment
name_ex	varchar (45)	No	ninguna	
descrip_ex	varchar (500)	No	ninguna	
photo_ex	varchar (200)	No	<i>Null</i>	

Tabla 12. Tabla exercises.

4.1.6 Tabla users_exercises

Esta tabla se ha creado por la necesidad de relacionar de varios a varios las tablas users y exercises, es decir un paciente puede tener varios ejercicios asignados, y cada ejercicio puede estar asignado a diferentes pacientes, por este motivo, es necesario la creación de esta tabla que los relacione.

A continuación, se describe cada uno de los campos de dicha tabla:

- id_user_ex: identificador de relación entre usuario y ejercicio, único y auto-incremental
- id_user: identificador de usuario.
- id_ex: identificador de ejercicio.
- date_user_ex: indica la fecha del ejercicio asignado.
- hour_user_ex: indica la hora del ejercicio asignado.
- comment_user_ex: indica información que el profesional puede considerar interesante sobre cómo realizar un ejercicio, o algún comentario que crea hacer oportuno.
- users_exerciseCol: indica si el ejercicio está realizado.



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

Tabla users_exercise				
Campo	Tipo	Nulo	Predeterminada	Extra
id_user_ex	int (11)	No	ninguna	auto_increment
id_user	int (11)	No	ninguna	
id_ex	int (11)	No	ninguna	
date_users_ex	date	No	ninguna	
hour_users_ex	time	No	ninguna	
id_pro	int (11)	No	ninguna	
comment_user_ex	text	No	ninguna	
users_exercisecol	varchar (45)	No	ninguna	

Tabla 13. Tabla users_exercises.

4.1.7 Tabla professionals

Esta tabla contiene la información necesaria de un profesional de una clínica para el desarrollo de todas las funcionalidades que propone la aplicación.

A continuación, se describe cada uno de los campos de dicha tabla:

- id_pro: identificador único del profesional, auto-incremental.
- name_pro: indica el nombre del profesional.
- surmane_pro: indica el apellido del profesional.
- email_pro: indica el correo electrónico de contacto del profesional.
- tel_pro: indica el teléfono de contacto del profesional.
- login_pro: indica el login de cada profesional, diferente para cada profesional y no puede haber un login_pro igual a ningún login_user(login de los pacientes de las clínicas).
- pass_pro: almacena la clase de acceso a los servicios de la parte de profesional o encargado.
- specialty_pro: indica la especialidad de cada profesional.
- id_clinic: indica el identificador de la clínica en la que trabaja el profesional.
- resposable_pro: indica si el profesional ejerce también la labor de encargado (1) o no (0).

CÁPITULO 4: DESCRIPCIÓN TÉCNICA DE LA APLICACIÓN



Tabla professionals				
Campo	Tipo	Nulo	Predeterminada	Extra
id_pro	int (11)	No	ninguna	auto_increment
dni_pro	varchar (45)	No	ninguna	
name_pro	varchar (45)	No	ninguna	
sumame_pro	varchar (45)	No	ninguna	
email_pro	varchar (45)	No	ninguna	
tel_pro	int (11)	No	ninguna	
login_pro	varchar (45)	No	ninguna	
pass_pro	varchar (45)	No	ninguna	
specialty_pro	varchar (45)	No	ninguna	
id_clinic	int (11)	No	ninguna	
responsible_pro	varchar (45)	No	ninguna	

Tabla 14. Tabla professionals.

4.1.8 Tabla reservations

Esta tabla contiene información de las reservas que realizan los pacientes. Cada paciente podrá hacer reservas en las diferentes clínicas, dependiendo de las necesidades que tenga.

A continuación, se describe cada uno de los campos de dicha tabla:

- id_res: identificador único de cada reserva realizada, auto-incremental.
- date_res: indica la fecha de la reserva.
- hour_res: indica hora de la reserva.
- id_clinic: indica el identificador de la clínica en la que se hace la reserva.
- id_user: indica el identificador de paciente que hace la reserva.
- dolor_res: indica el lugar del dolor que tiene el paciente.
- comment_res: indica un breve comentario que el paciente le escribe al paciente a la hora de la realización de la reserva.
- id_pro: indica el identificador del profesional que se le ha asignado a la reserva
- responsibleLock_res: indica que el encargado pueda bloquear ciertas horas o días enteros para impedir que los pacientes puedan reservar (1).



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

Tabla reservations				
Campo	Tipo	Nulo	Predeterminada	Extra
id_res	int (11)	No	ninguna	auto_increment
date_res	date	No	ninguna	
hour_res	time	No	ninguna	
id_clinic	int (11)	No	ninguna	
id_user	int (11)	No	ninguna	
dolor_res	varchar (45)	No	ninguna	
comment_res	text	No	ninguna	
id_pro	int (11)	No	ninguna	
responsibleLock_res	int (11)	No	ninguna	

Tabla 15. Tabla reservations

4.1.9 Tabla news

Esta tabla es utilizada en el sitio web para ofrecer las últimas noticias a los pacientes.

A continuación, se describe cada uno de los campos de dicha tabla:

- id_news: identificador único de cada noticia, auto-incremental.
- descrip_news: indica la descripción de cada noticia.
- date_news: indica la fecha de cada noticia.

Tabla news				
Campo	Tipo	Nulo	Predeterminada	Extra
id_news	int (11)	No	ninguna	auto_increment
descrip_news	text	No	ninguna	
date_news	date	No	ninguna	

Tabla 16. Tabla news

4.2 Relación entre las tablas de la base de datos

En este apartado se muestran las distintas relaciones que hay entre todas las tablas de la base de datos que utiliza la aplicación para su perfecto funcionamiento. Para su mejor comprensión, se hace uso de unas figuras ilustrando estas relaciones.



CÁPITULO 4: DESCRIPCIÓN TÉCNICA DE LA APLICACIÓN

4.2.1 Clinics and treatments

La siguiente figura ilustra la relación de varios a varios que existe entre las tablas clinics y treatments (verde). Una clínica puede tener varios tratamientos y de la misma manera, un tratamiento puede ser proporcionado en varias clínicas. Por este motivo, es necesario que se tenga que crear una tercera tabla (clinics_treatments) que relacione las tablas que había inicialmente.

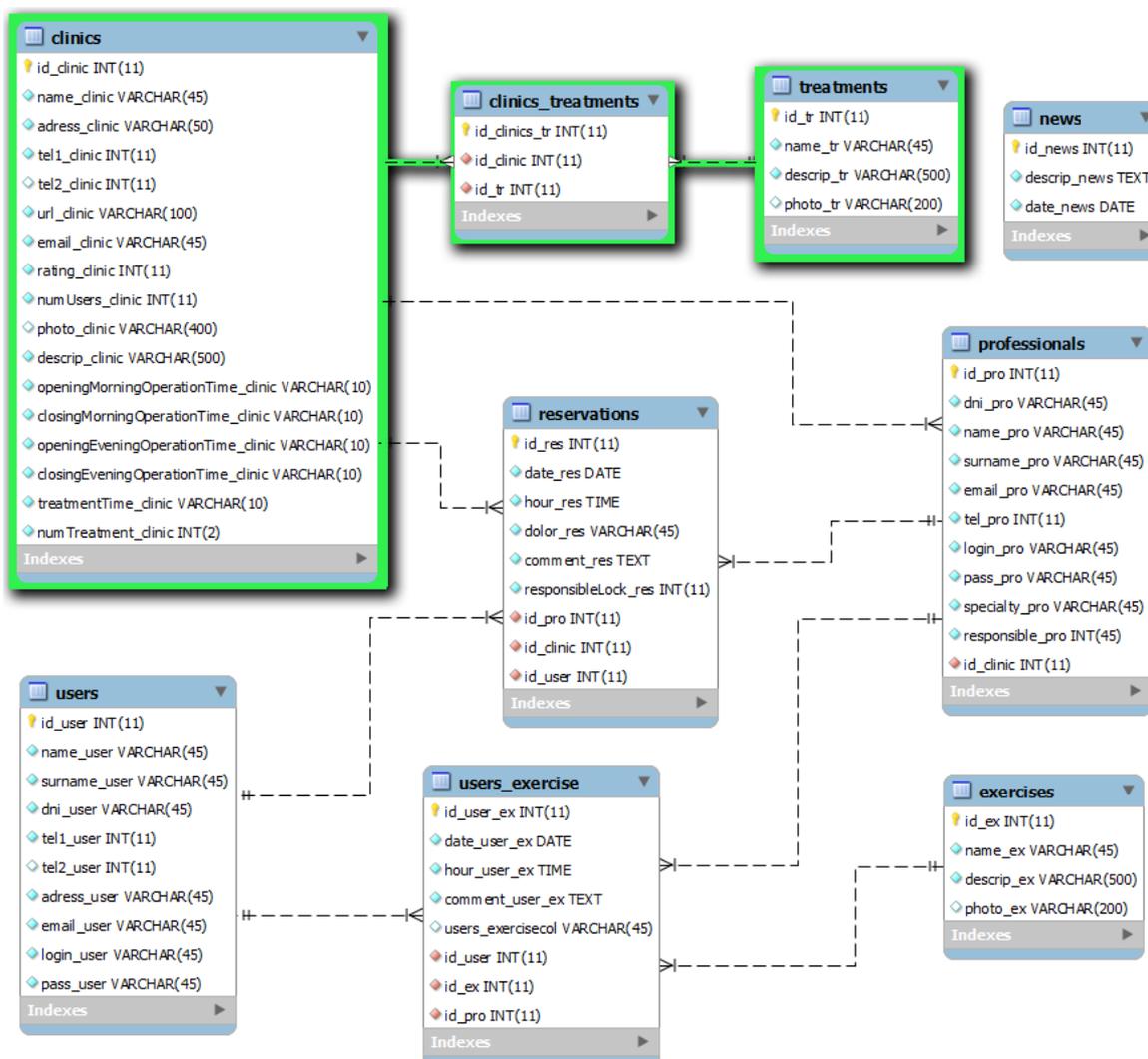


Figura 25. Relación entre tablas clinics y treatments.

4.2.2 Professionals, exercises and users



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

La siguiente figura ilustra la relación de varios a varios que existe entre exercises y users (morado). Un ejercicio puede ser asignado a varios usuarios y un usuario puede tener asignado varios ejercicios. Por este motivo, es necesario que se tenga que crear una tercera tabla (users_exercises) que relacione las tablas que había inicialmente.

Por otra parte, la figura también ilustra la relación de uno a varios entre professionals y exercises (morado). Un ejercicio puede ser tratado y asignado por un profesional, pero cada profesional puede tener asignado varios ejercicios.

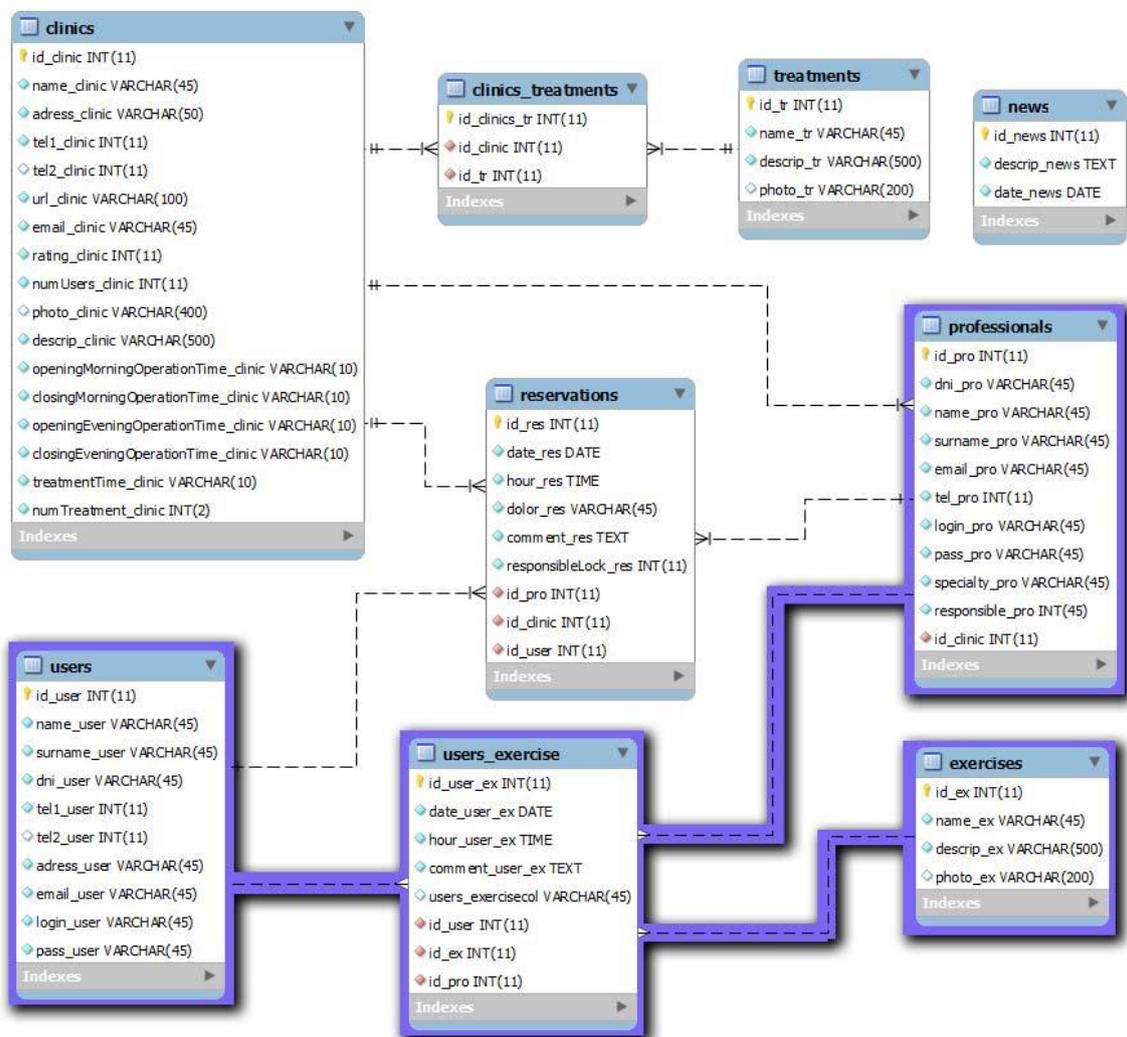


Figura 26. Relación entre tablas users, professionals y exercises

4.2.3 Clinics, treatments, reservations and professionals



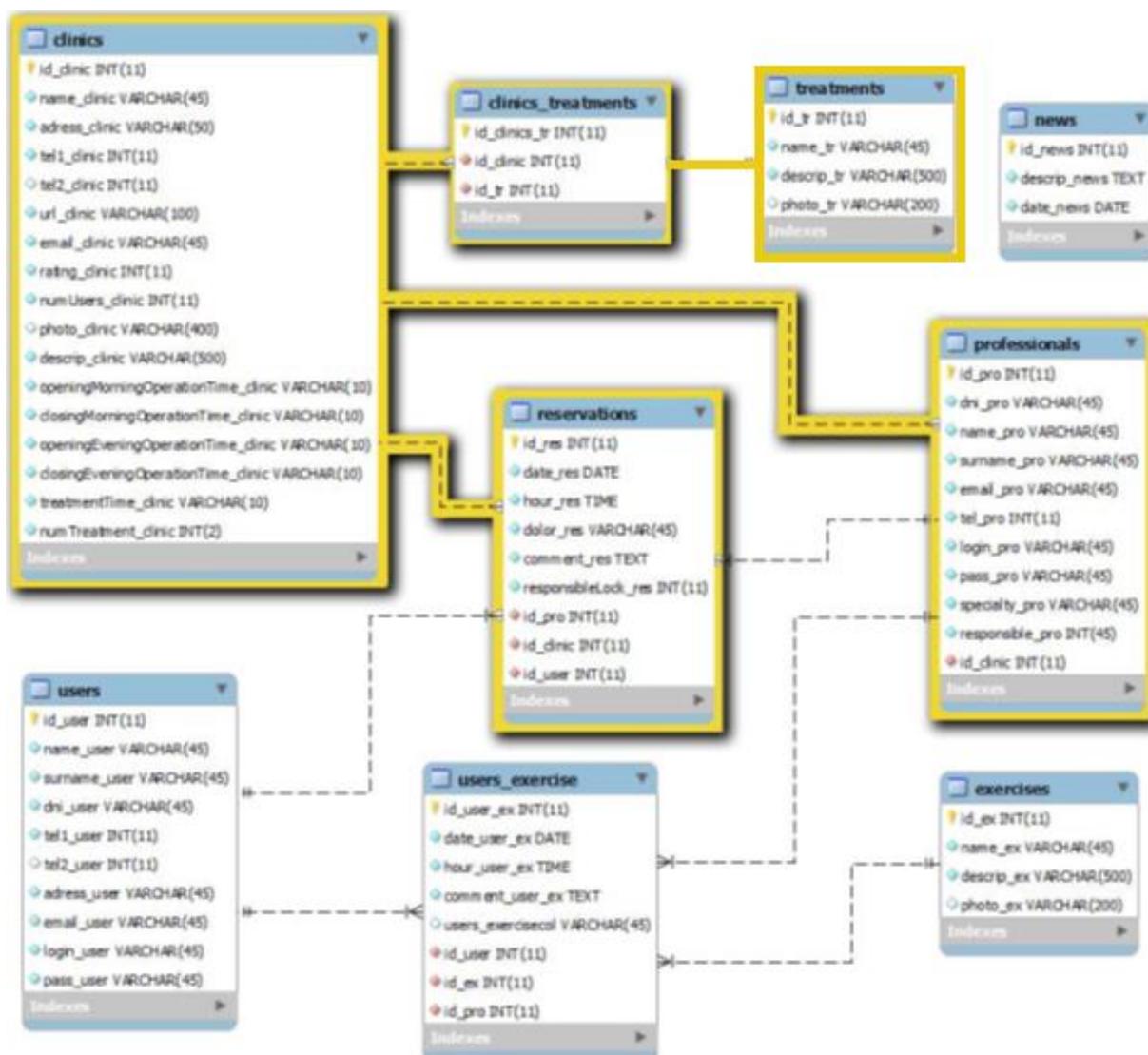
CÁPITULO 4: DESCRIPCIÓN TÉCNICA DE LA APLICACIÓN

La siguiente figura ilustra dos relaciones de uno a varios entre la tabla clinics y las tablas professionals y reservations y, una relación de varios a varios entre las ya anteriormente comentadas tablas clinics y treatments (amarillo).

La relación de uno a varios entre las tablas clinics y professionals viene marcada por el identificador de la clínica en la tabla professionals ya que, en cada clínica puede haber varios profesionales.

La relación de uno a varios entre las tablas clinics y reservetions, de la misma manera, viene marcada por el identificador de la clínica en la tabla reservations ya que, en cada clínica puede haber varias reservas.

La relación de varios a varios entre las tablas clinics y treatments ya ha sido descrita anteriormente.





APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

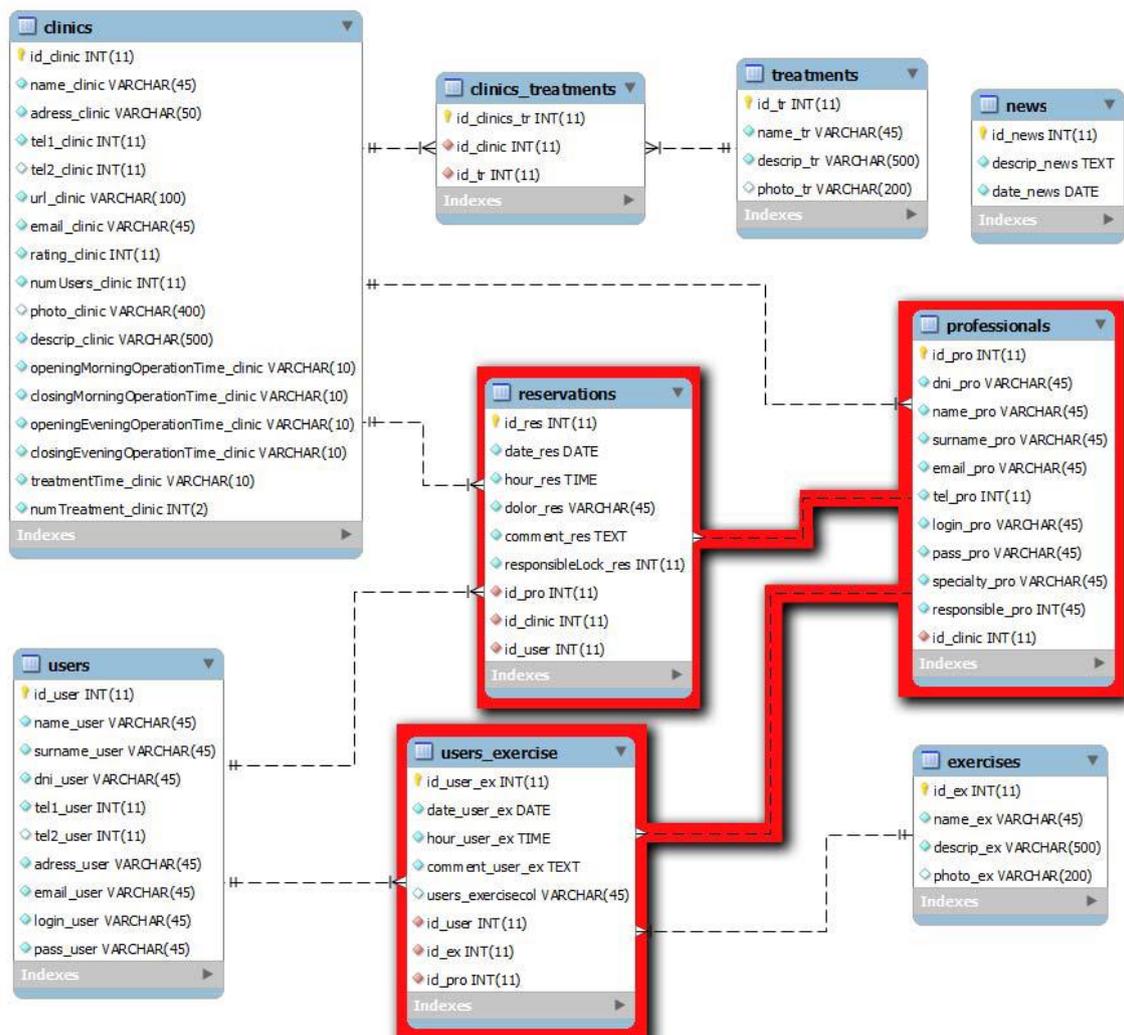
Figura 27. Relación entre tablas clinics, reservations, professionals and treatments.

4.2.4 Professionals, reservations y users_Exercise

La siguiente figura ilustra las dos relaciones de uno a varios entre la tabla professionals y las tablas reservations y users_exercises (rojo).

La relación de uno a varios de las tablas professionals y reservations viene marcada por identificador de profesional en la tabla reservations de manera que solo ese profesional puede gestionar dicha reserva, pero ese profesional puede tener varias reservas.

La relación de uno a varios entre las tablas profesionales y users_exercises viene marcada porque los ejercicios que se le asignan a los pacientes solo puede ser asignados por un profesional, pero este profesional puede asignar varios ejercicios a sus pacientes.



CÁPITULO 4: DESCRIPCIÓN TÉCNICA DE LA APLICACIÓN



Figura 28. Relación entre las tablas profesionales, reservations y users_exercise.

4.2.5 Professionals, reservations, clinics y users

La siguiente figura ilustra la relación entre las tablas professionals, users, reservations y clinics. En este caso tenemos relaciones uno a varios, dado que una reserva solo puede tener una clínica, un profesional y un paciente asignado, pero cada paciente, profesional o clínica pueden tener varias reservas adjudicadas (naranja).

Las reservas, es decir, en la tabla reservations, se necesita información de las demás tablas. Cuando un paciente realiza su reserva, en ésta queda guardado el identificador del paciente (id_user) y el identificador de la clínica donde ha reservado (id_clinic). Una vez que la cita se ha reservado, el encargado la gestionará y le adjudicará un profesional dependiendo de las patologías del paciente, por lo que, de nuevo, tendremos que guardar otro campo, el identificador del profesional (id_pro).



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

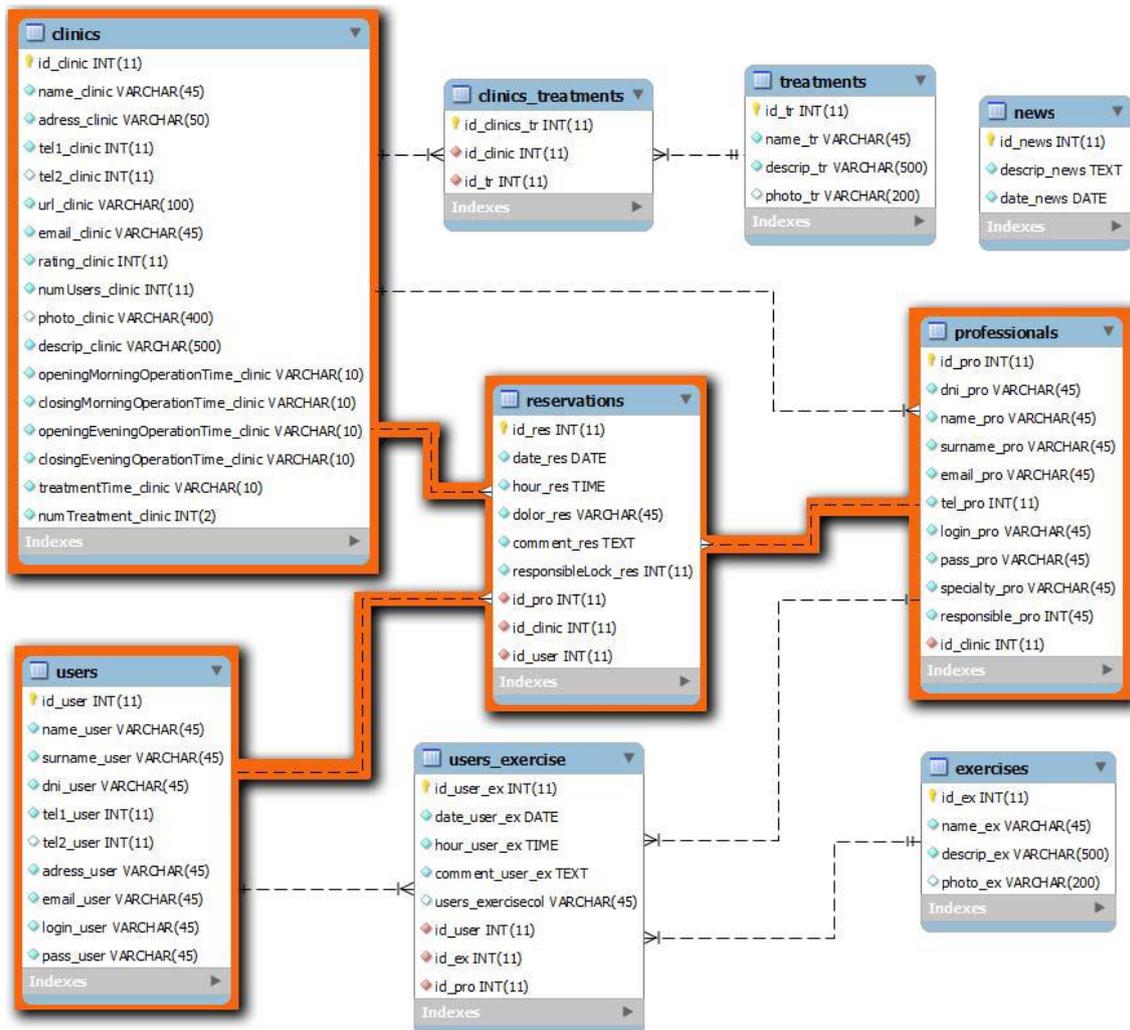


Figura 29. Relación entre las tablas reservations, clinics, professionals y users.

4.3 Descripción de las funcionalidades y diagramas de flujo

En este apartado se van a describir detalladamente las funcionalidades internas de la aplicación. La aplicación se ha desarrollado tratando de adoptar los mejores criterios y siguiendo en todo momento las funcionalidades para maximizar el rendimiento del usuario y su interactividad con el profesional.

Para ello, se toman como apoyo diferentes diagramas de flujo de la aplicación. Cada una de estas funcionalidades partirá desde un estado inicial, desde donde la aplicación dividirá en diferentes ramas, dependiendo del tipo de usuario que inicie sesión y de si el usuario de la aplicación ya está logueado o cerró la sesión.



CÁPITULO 4: DESCRIPCIÓN TÉCNICA DE LA APLICACIÓN

Los usuarios que pueden usar esta aplicación pueden ser de los tres tipos mencionados con anterioridad: pacientes (*users*) y profesionales y encargados (*professionals*), diferenciados por el valor del campo *responsable_pro*.

Los encargados tienen como principales funcionalidades:

- Adjudicar paciente al profesional.
- Asignar ejercicio al paciente.
- Ver tratamientos.
- Ver todas las reservas de los pacientes.

Los profesionales tienen como principales funcionalidades:

- Ver tratamientos.
- Asignar ejercicio al paciente.
- Contactar con la clínica.
- Lista de citas.

Los pacientes tienen como principales funcionalidades:

- Clínicas.
- Reservar una cita.
- Lista de ejercicios del paciente.
- Mis reservas.

4.3.1 Clase LoginVC y clase ViewController

El siguiente diagrama de flujo ilustra las clases a las que se puede acceder cuando se inicia la aplicación. Al iniciar la aplicación, se accede a la clase *ViewController* en la que se comprueba si el usuario cerró la sesión antes de dejar de usar la aplicación o no. En caso de que no la cerrara, esta clase accederá a la pantalla del menú principal del usuario logueado. En caso de haber cerrado la sesión, se accede a la pantalla de “Login”.

En esta pantalla, se tienen dos opciones, o bien registrarse y pasar a la pantalla de registro, en la que se hará uso de la clase *RegisterPageViewController*, o bien, introducir los datos del usuario para loguearse en la aplicación y hacer uso de la clase *LoginViewController*.

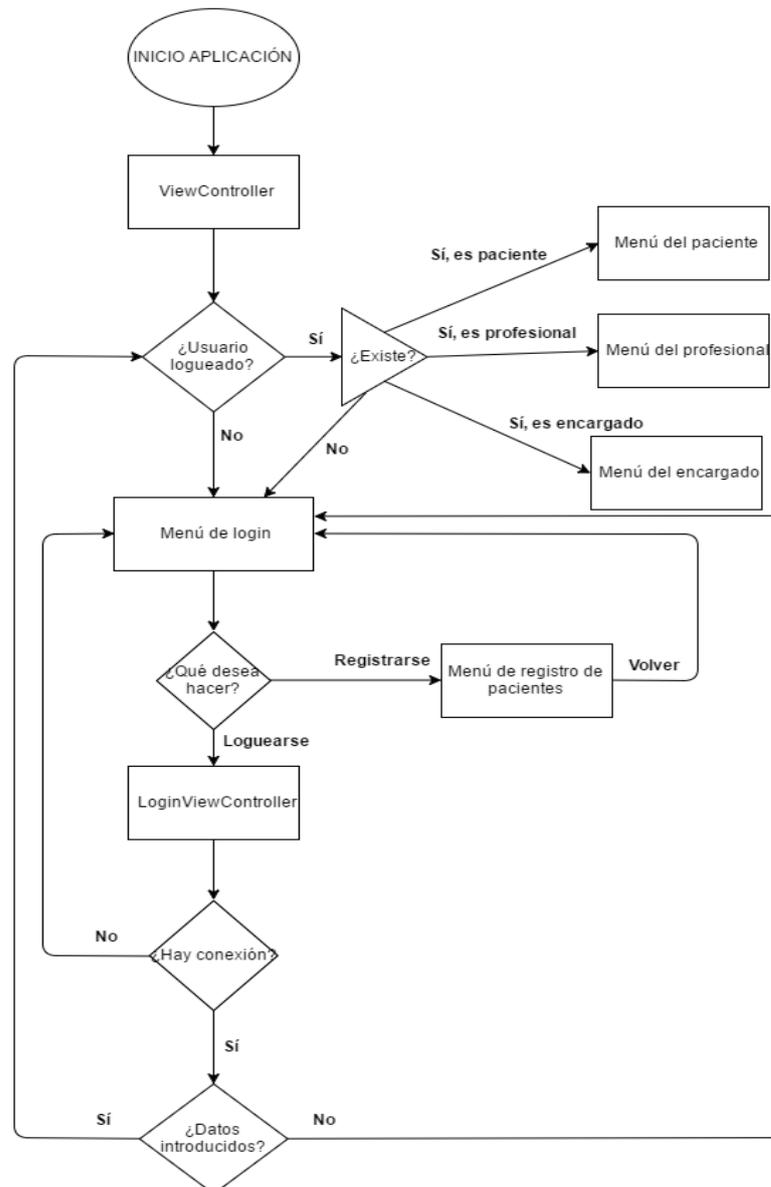
Tras introducir los datos de login y contraseña, esta clase primero comprueba que hay conexión a internet (esta comprobación se realiza cada vez que se intenta acceder a un fichero php en cualquier funcionalidad de la aplicación, por lo que no se



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

repetirá en próximas secciones), y que se han introducido todos los datos. Después envía los datos mediante un POST a un php cotejando los datos con la base de datos externa MySQL para ver si el usuario está registrado en la base de datos. Se comprobará si los datos se corresponden con los almacenados en la tabla *users* o en la tabla *professionals* de la base de datos. Después de esta comprobación, se vuelve a la clase *LoginViewController*, con la respuesta en formato JSON. Esta acción se realiza cada vez que se accede a un fichero php en cada una de las funcionalidades de la aplicación.

Si se ha logueado un paciente, pasaremos al menú principal del usuario y si se ha logueado un profesional, dependiendo de si es encargado o no, accederá al menú principal del profesional o del encargado.



CÁPITULO 4: DESCRIPCIÓN TÉCNICA DE LA APLICACIÓN



Figura 30. Diagrama de flujo de “Login”.

4.3.2 Clase RegisterPageVC

El siguiente diagrama ilustra la clase *RegisterPageViewController*. Comenzando en la pantalla de “Registro” de pacientes de la aplicación, tras rellenar toda la información que se requiere, pulsando el botón de “Registrarse”, se hace uso de [ésta esta](#) clase. Se envían los datos mediante POST a un php (RegistroUsuario.php). En este fichero, tras recibir los datos del paciente, comprueba, en este orden, si los campos con asterisco, los campos obligatorios, están rellenos, si el nombre y los apellidos están bien escritos, si el DNI es real, si los teléfonos cumplen con sus características, si la dirección y el correo electrónico están bien escritos y si los identificadores de login y password cumplen con las condiciones impuestas. Por último, si se cumplen todos los requisitos, se comprueba que el login no esté usado ni por paciente ya registrado ni por un profesional y, que el DNI no se ha usado en un paciente, puesto que un mismo profesional si puede ser paciente a la vez. Ante cualquier error en los datos, se muestra la advertencia oportuna. Si se cumple todo lo anterior, el paciente se registrará en la aplicación, almacenándose dicho paciente en la tabla *users* de la base de datos y, devolviendo la respuesta en formato JSON a la clase mencionada. Una vez registrado el paciente, se volverá a la pantalla de “Login”.



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

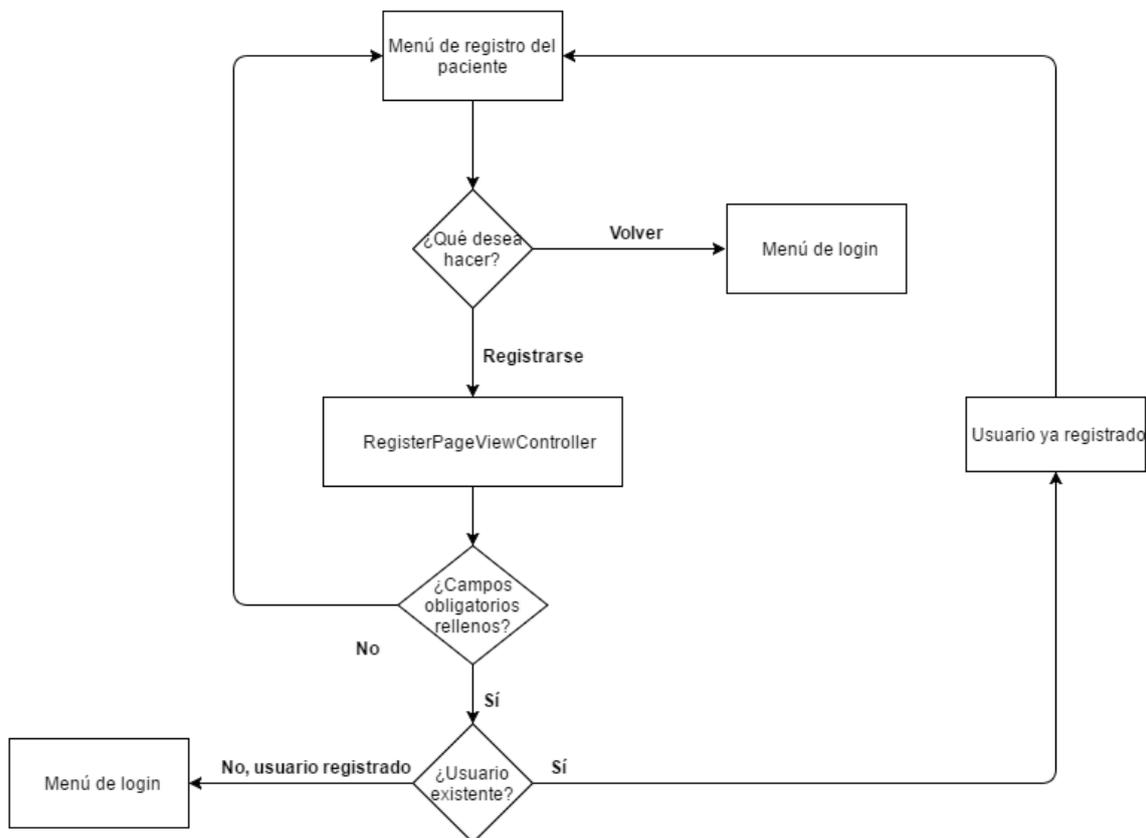


Figura 31. Diagrama de flujo de “Registro”.

4.3.3 Clase UsuarioVC

El diagrama siguiente describe las funcionalidades de un paciente una vez que logueado. Accede a la pantalla principal del paciente, en la que podrá realizar las diferentes funciones anteriormente enumeradas.

A continuación, si el paciente desea realizar alguna función, se hará uso de la clase *UsuarioViewController* dónde dependiendo de la funcionalidad deseada se ejecutará la función correspondiente accediendo al fichero php asignado para, dependiendo de los datos recibidos de la base de datos, realizar la tarea correspondiente. Puede mostrarse un pop-up si no se encuentran resultados a su consulta, o bien, pasar a la pantalla siguiente.



CÁPITULO 4: DESCRIPCIÓN TÉCNICA DE LA APLICACIÓN

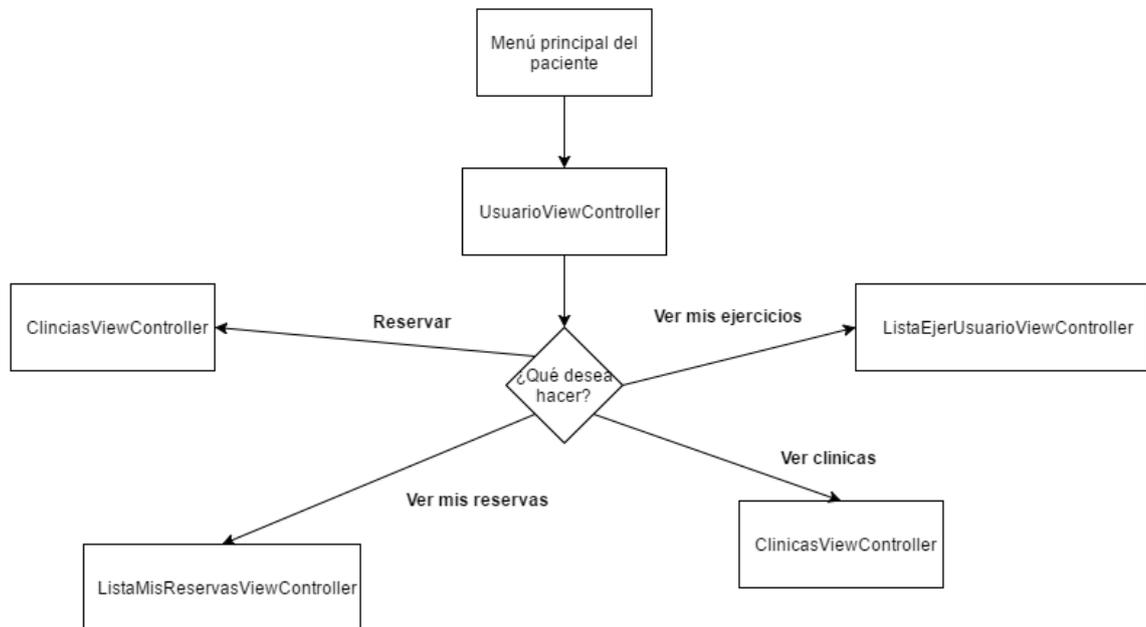


Figura 32. Diagrama de flujo menú principal del usuario/paciente.

4.3.3.1 Clases *ClinicasVC*, *EspecificacionesClinicasVC*, *ComoLlegarVC* y *TratamientosVC*

El diagrama siguiente describe las funcionalidades de un paciente cuando pulsa en el menú principal del paciente el botón “*Clinicas*”.

Una vez pulsado el botón, en caso de haber clínicas disponibles, se hace uso de la clase *ClinicasViewController*, donde se muestra una lista con todas las clínicas, en caso contrario no se accederá a dicha clase y se mostrará un pop-up con la información pertinente.

Si se selecciona una clínica de la lista, se hará uso de la clase *EspecificacionesClinicasViewController* donde se muestra con detalle las características de la clínica elegida y se ofrecen diferentes funcionalidades adicionales como poder compartir la clínica, localizarla en un mapa y ver los tratamientos con los que trabaja.

En el caso de localización de la clínica en el mapa, se hace uso de la clase *ComoLlegarViewController* y en el caso de querer ver los tratamientos, se hace uso de la clase *TratamientosViewController*, que muestra una lista de tratamientos. Si no hay tratamientos disponibles en ese momento por la causa que sea, se muestra un pop-up informativo y se devolverá al menú principal.



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

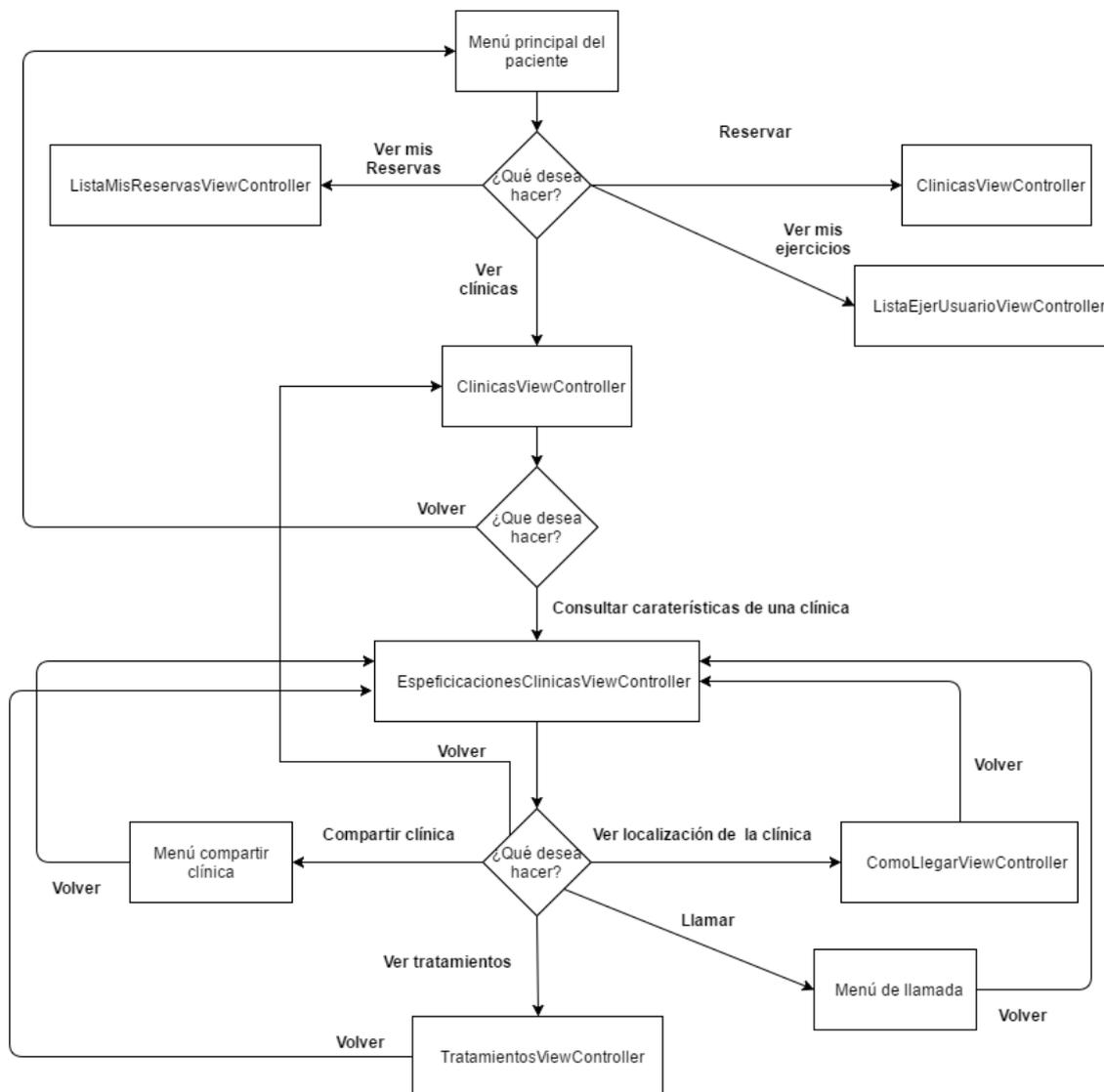


Figura 33. Diagrama de flujo “Clínicas”.

4.3.3.2 Clases *ClinicasVC*, *EspecificacionesClinicaReservaVC*, *DatosParaReservarViewController*, *CalendarioVC* y *ConsultarCitaVC*

El diagrama siguiente describe las funcionalidades de un paciente cuando pulsa en el menú principal del paciente el botón “*Clinicas*”.

Una vez pulsado el botón, en caso de haber clínicas disponibles, se hace uso de la clase *ClinicasViewController*, donde se muestra una lista con todas las clínicas en las que se puede reservar cita, en caso contrario no se accederá a dicha clase y se mostrará un pop-up con la información pertinente.

Si se selecciona una clínica de la lista, se hará uso de la clase *EsfeficacionesClinicaReservasViewController*, donde se muestra con detalle las

CÁPITULO 4: DESCRIPCIÓN TÉCNICA DE LA APLICACIÓN

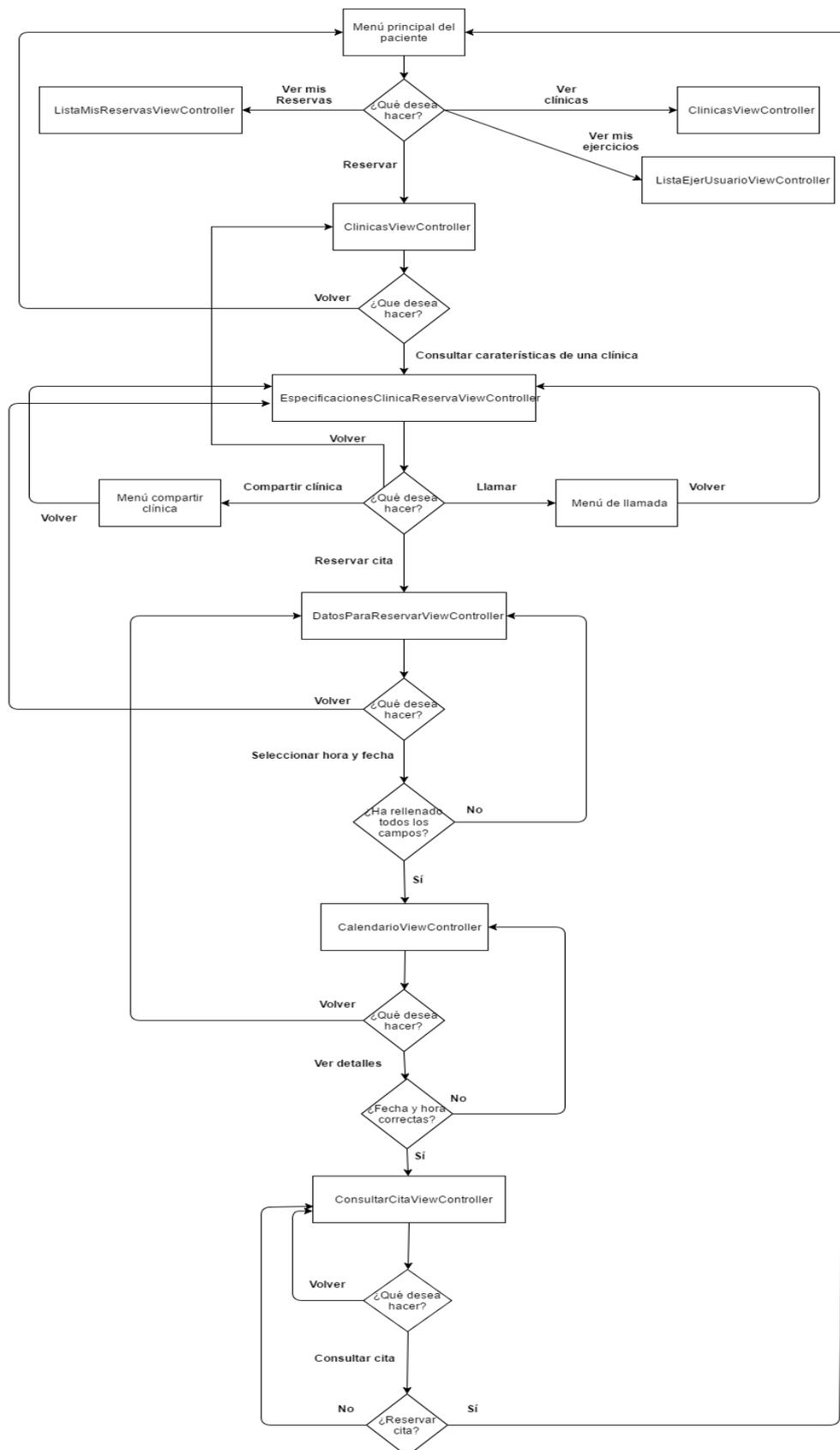


características de la clínica elegida y se ofrecen diferentes funcionalidades adicionales como poder compartir la clínica o hacer una reserva en esa clínica.

Para reservar una cita, en primer lugar, se hará uso de la clase *DatosParaReservarViewController*, donde se introducirán datos relevantes para realizar la reserva. Si se han rellenado todos los campos, mediante el botón “*Seleccionar fecha y hora*”, se hará uso de la clase *CalendarioViewController*, donde se selecciona la hora y fecha deseada de la consulta. Después, mediante el botón “*Ver detalles*”, se hará uso de la clase *ConsultarCitaViewController*, donde se mostrará un resumen de nuestra reserva y un botón “*Consultar cita*”, en el que pulsándolo se muestra un pop-up con la hora más próxima a deseada por el paciente, tras acceder al fichero php correspondiente enviando todos los requisitos de la cita deseada por el paciente. Éste podrá decidir si la hora ofrecida le conviene y podrá reservar la cita o cancelarla.



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA





CÁPITULO 4: DESCRIPCIÓN TÉCNICA DE LA APLICACIÓN

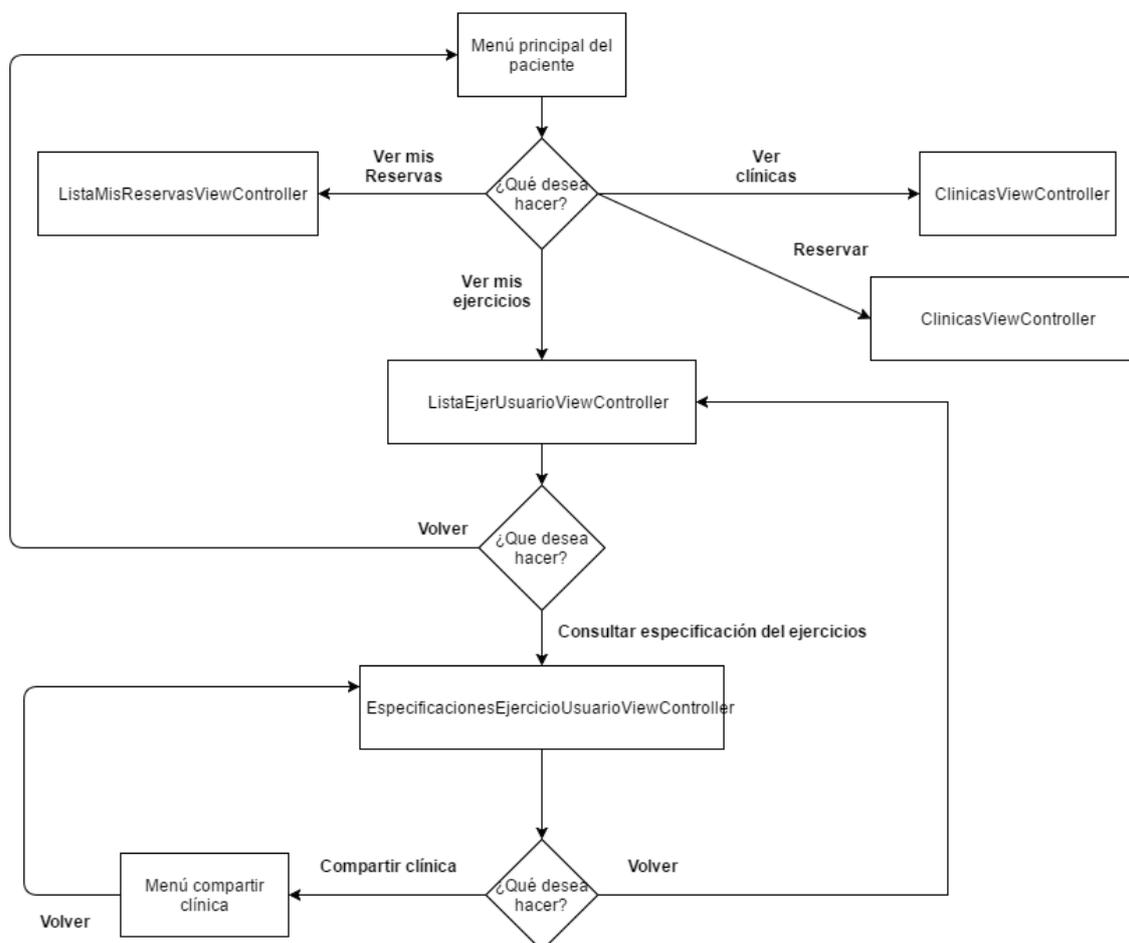
Figura 34. Diagrama de flujo “Reservar cita”.

4.3.3.3 Clases *ListaEjerUsuarioVC* y *EspecificacionesEjercicioUsuarioVC*

El diagrama siguiente describe las funcionalidades de un paciente cuando pulsa en el menú principal del paciente el botón “Ejercicios”.

Una vez pulsado el botón, en caso de que el paciente tenga ejercicios disponibles para realizar, se hace uso de la clase *ListaEjerUsuarioViewController*, donde se muestra una lista con todos los ejercicios que el paciente tiene que realizar, junto con la fecha del día con el que se asignó por el profesional, en caso contrario no se accederá a dicha clase y se mostrará un pop-up con la información pertinente.

Si se selecciona un ejercicio de la lista, se hará uso de la clase *EspecificacionesEjercicioUsuarioViewController*, donde se muestra con detalle las características del ejercicio seleccionado y se ofrece la opción de compartir el ejercicio.





APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

Figura 35. Diagrama de flujo “Lista de ejercicios del paciente”.

4.3.3.4 Clases *ListaMisReservasVC* y *AnularReservaVC*

El diagrama siguiente describe las funcionalidades de un paciente cuando pulsa en el menú principal del paciente el botón “*Mis reservas*”.

Una vez pulsado el botón, en caso de haber el paciente tenga reservas, se hace uso de la clase *ListaMisReservasViewController*, donde se muestra una lista con todas las reservas que tiene disponible el paciente, en caso contrario no se accederá a dicha clase y se mostrará un pop-up con la información pertinente.

Si se selecciona una clínica de la lista, se hará uso de la clase *AnularCitaViewController*, donde se muestra con detalle las características de la reserva seleccionada y se ofrece la opción de anular cualquier cita. En el caso de querer realizar alguna esta opción, se presionará el botón “*Anular cita*” y, tras consultar el fichero php correspondiente se procede a advertir al usuario si realmente quiere anular su cita.



CÁPITULO 4: DESCRIPCIÓN TÉCNICA DE LA APLICACIÓN

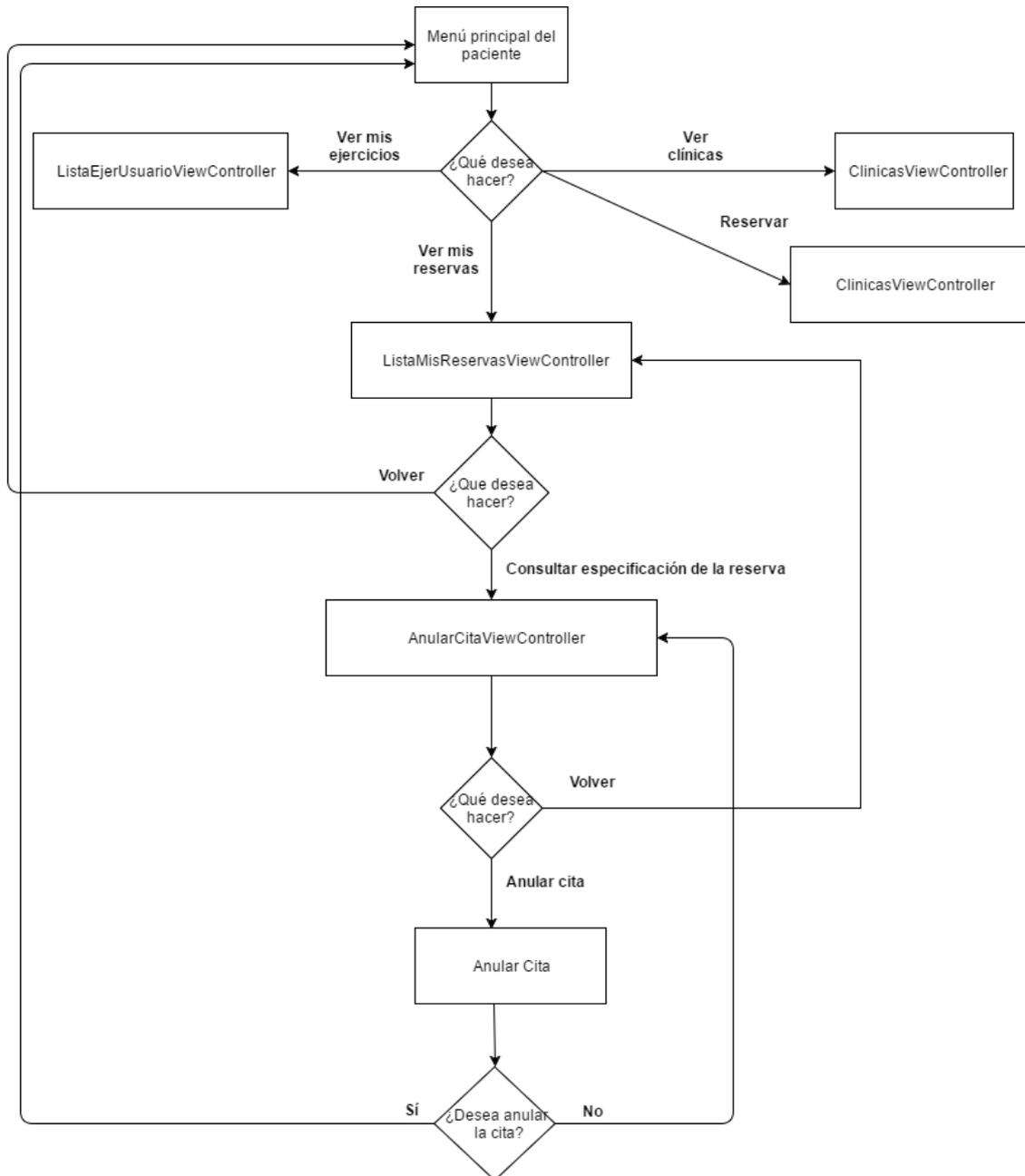


Figura 36. Diagrama de flujo “Ver reservas del paciente”.

4.3.4 Clase ProfesionalVC

El diagrama siguiente describe las funcionalidades de un profesional una vez que logueado. Accede a la pantalla principal del profesional, en la que podrá realizar las diferentes funciones anteriormente enumeradas.



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

A continuación, si el paciente desea realizar alguna función, se hará uso de la clase *ProfesionalViewController* dónde dependiendo de la funcionalidad deseada se ejecutará la función correspondiente accediendo al fichero php asignado para, dependiendo de los datos recibidos de la base de datos, realizar la tarea correspondiente. Puede mostrarse un pop-up si no se encuentran resultados a su consulta, o bien, pasar a la pantalla siguiente.

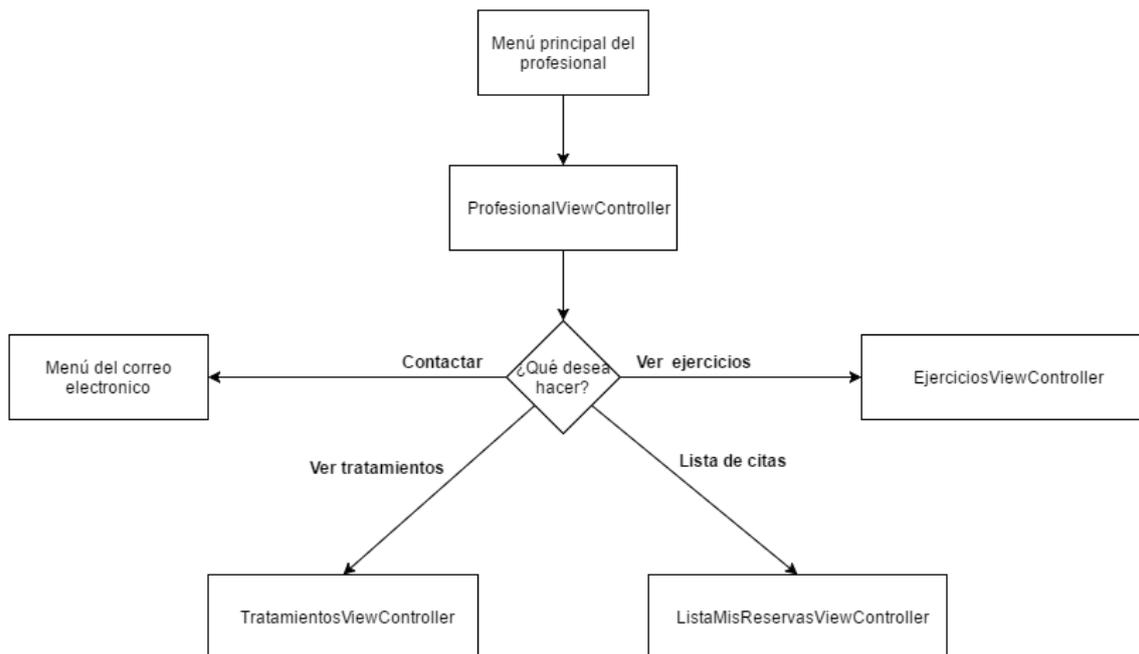


Figura 37. Diagrama de flujo del menú principal del profesional.

4.3.4.1 Clases *TratamientosVC* y *EspecificacionesTratamientoVC*

El diagrama siguiente describe las funcionalidades de un profesional cuando pulsa en el menú principal del paciente el botón “*Tratamientos*”.

Una vez pulsado el botón, en caso de que haya tratamientos disponibles, se hace uso de la clase *TratamientosViewController*, donde se muestra una lista con todos los tratamientos disponibles, en caso contrario no se accederá a dicha clase y se mostrará un pop-up con la información pertinente.

Si se selecciona un tratamiento de la lista, se hará uso de la clase *EspecificacionesTratamientoViewController*, donde se muestra con detalle las características del tratamiento seleccionado y se ofrece la opción de compartir dicho tratamiento.



CÁPITULO 4: DESCRIPCIÓN TÉCNICA DE LA APLICACIÓN

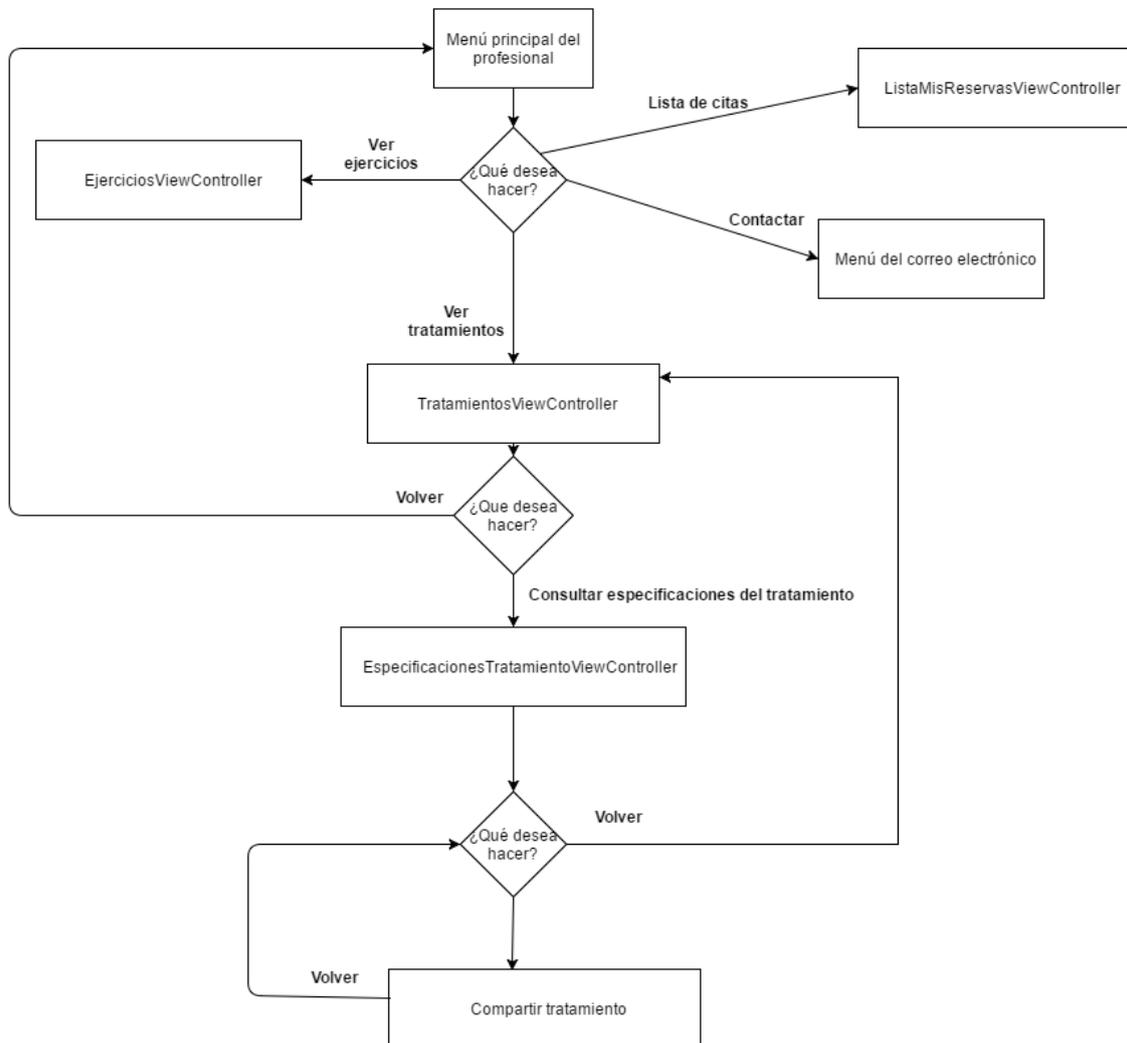


Figura 38. Diagrama de flujo “Ver tratamientos”.

4.3.4.2 Clases EjerciciosVC, EspecificacionesEjercicioVC, ListaPacProVC y AsignarEjercicioVC.

El diagrama siguiente describe las funcionalidades de un profesional cuando pulsa en el menú principal del paciente el botón “Ejercicios”.

Una vez pulsado el botón, en caso de que haya ejercicios disponibles que se puedan asignar, se hace uso de la clase *EjerciciosViewController*, donde se muestra una lista con todos los ejercicios disponibles, en caso contrario no se accederá a dicha clase y se mostrará un pop-up con la información pertinente.

Si se selecciona un ejercicio de la lista, se hará uso de la clase *EspecificacionesEjerciciosViewController*, donde se muestra con detalle las



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

características del tratamiento seleccionado y se ofrece la opción de compartir dicho ejercicio y que el profesional pueda asignarlo a uno de sus pacientes.

En el caso de querer asignar el ejercicio a un paciente, una vez el profesional pulse en el botón “*Asignar ejercicio*”, se hará uso de la clase *ListaPacProViewController*, donde se mostrará una lista con todos los pacientes que tiene asignados el profesional para que pueda asignar el ejercicio al paciente que seleccione. En caso de que no se tenga pacientes asignados, se podrá volver al menú anterior.

Una vez que el profesional selecciona el paciente al que quiere asignar el ejercicio, se hará uso de la clase *AsignarEjercicioViewController*, en la que podrá insertar un breve comentario para la realización del ejercicio y pulsando en el botón “*Asignar Ejercicio*”, finalmente, el profesional asigna el ejercicio deseado.

CÁPITULO 4: DESCRIPCIÓN TÉCNICA DE LA APLICACIÓN

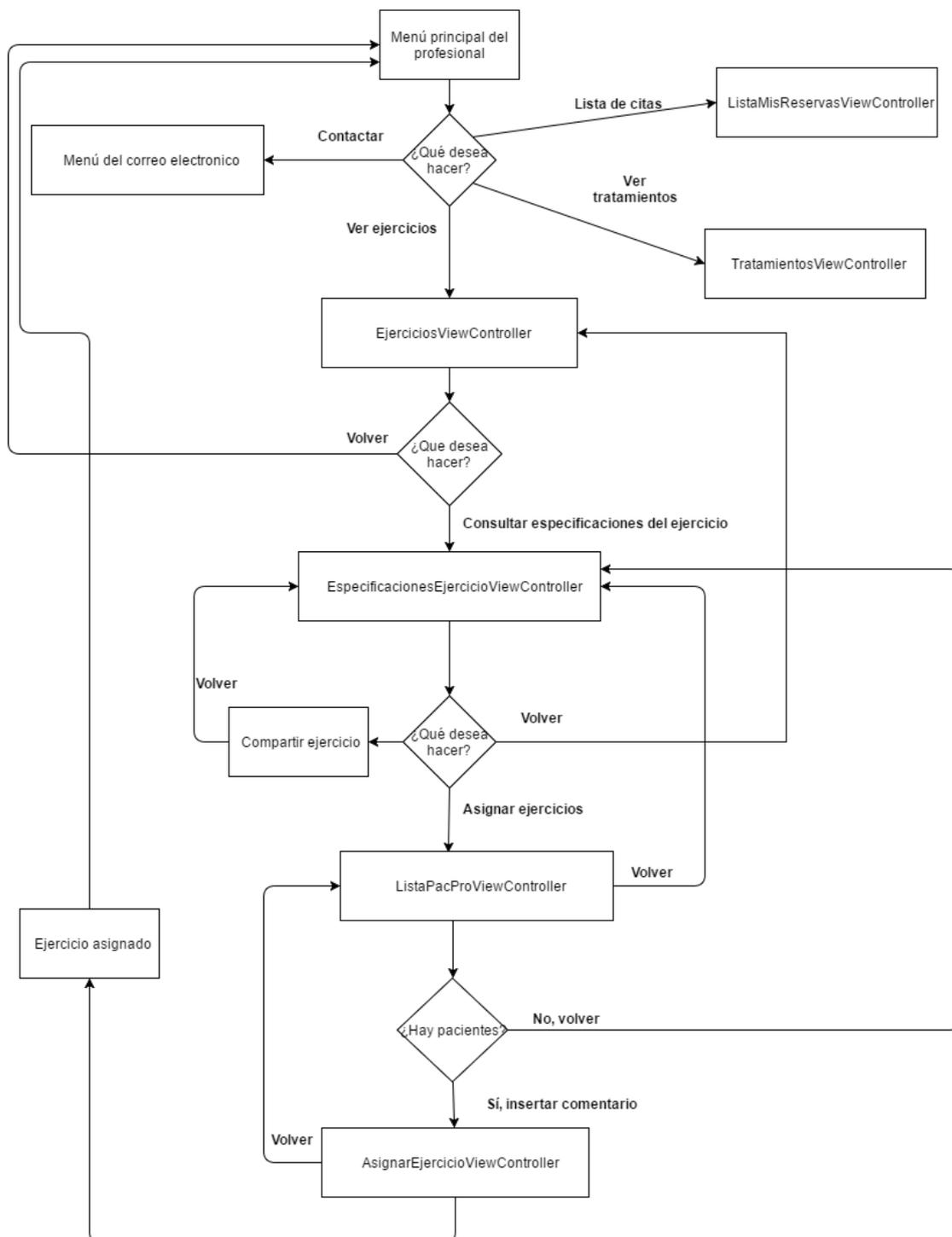


Figura 39. Diagrama de flujo “Asignar ejercicio al paciente”.



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

4.3.4.3 Clase *ListaMisReservasVC* y *EspecificacionesCitaProfesionalVC*

El diagrama siguiente describe las funcionalidades de un profesional cuando pulsa en el menú principal del paciente el botón “*Lista de citas*”.

Una vez pulsado el botón, en caso de que el profesional tenga citas asignadas, se hace uso de la clase *ListaMisReservasViewController*, donde se muestra una lista con todos los tratamientos disponibles, en caso contrario no se accederá a dicha clase y se mostrará un pop-up con la información pertinente.

Si se selecciona una cita de la lista, se hará uso de la clase *EspecificacionesCitasProfesionalViewController*, donde se muestra con detalle los datos de la cita seleccionada.

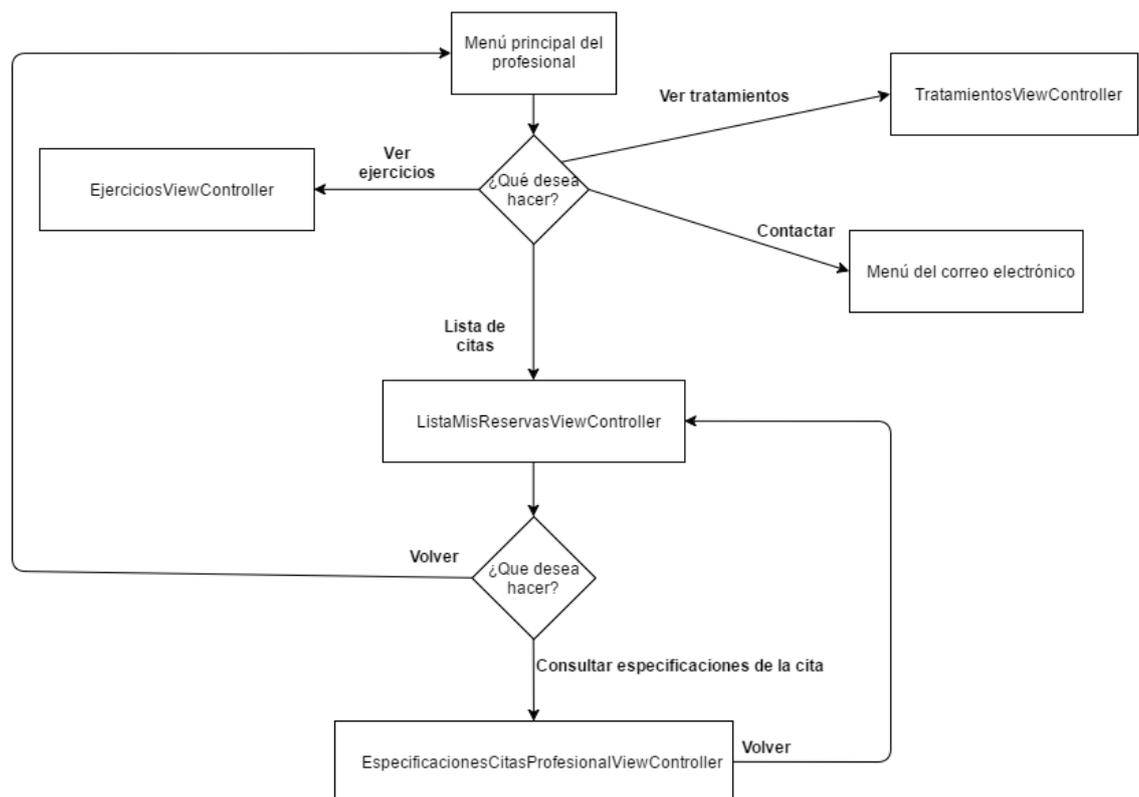


Figura 40. Diagrama de flujo “Lista de citas”.

4.3.5 Clase *EncargadoViewController*



CÁPITULO 4: DESCRIPCIÓN TÉCNICA DE LA APLICACIÓN

El diagrama siguiente describe las funcionalidades de un encargado una vez que logueado. Accede a la pantalla principal del encargado, en la que podrá realizar las diferentes funciones anteriormente enumeradas.

A continuación, si el paciente desea realizar alguna función, se hará uso de la clase *EncargadoViewController* dónde dependiendo de la funcionalidad deseada se ejecutará la función correspondiente accediendo al fichero php asignado para, dependiendo de los datos recibidos de la base de datos, realizar la tarea correspondiente. Puede mostrarse un pop-up si no se encuentran resultados a su consulta, o bien, pasar a la pantalla siguiente.

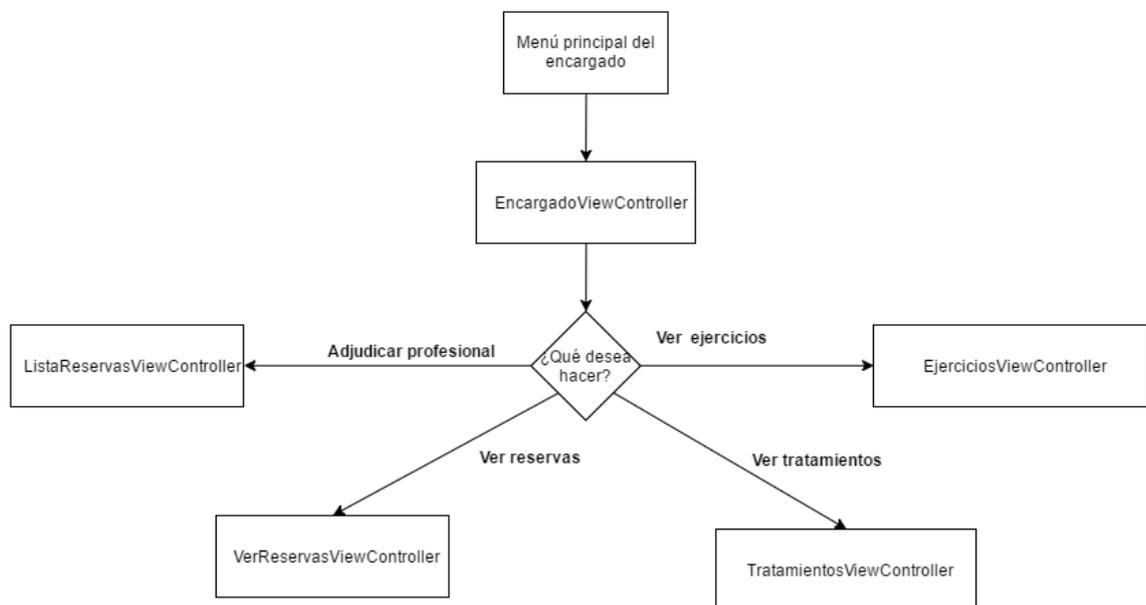


Figura 41. Diagrama de flujo del menú principal del encargado.

4.3.5.1 Clases *ListaReservasVC*, *EspecificacionesReservasVC* y *ListaProfesionalesVC*

El diagrama siguiente describe las funcionalidades de un profesional cuando pulsa en el menú principal del paciente el botón “*Adjudicar Paciente*”.

Una vez pulsado el botón, en caso de que haya reservas disponibles en las que el profesional aún no ha sido adjudicado, se hace uso de la clase *ListaReservasViewController*, donde se muestra una lista con todas las reservas de este tipo disponibles, en caso contrario no se accederá a dicha clase y se mostrará un pop-up con la información pertinente.

Si se selecciona un ejercicio de la lista, se hará uso de la clase *EspecificacionesReservasViewController*, donde se muestra con detalle las

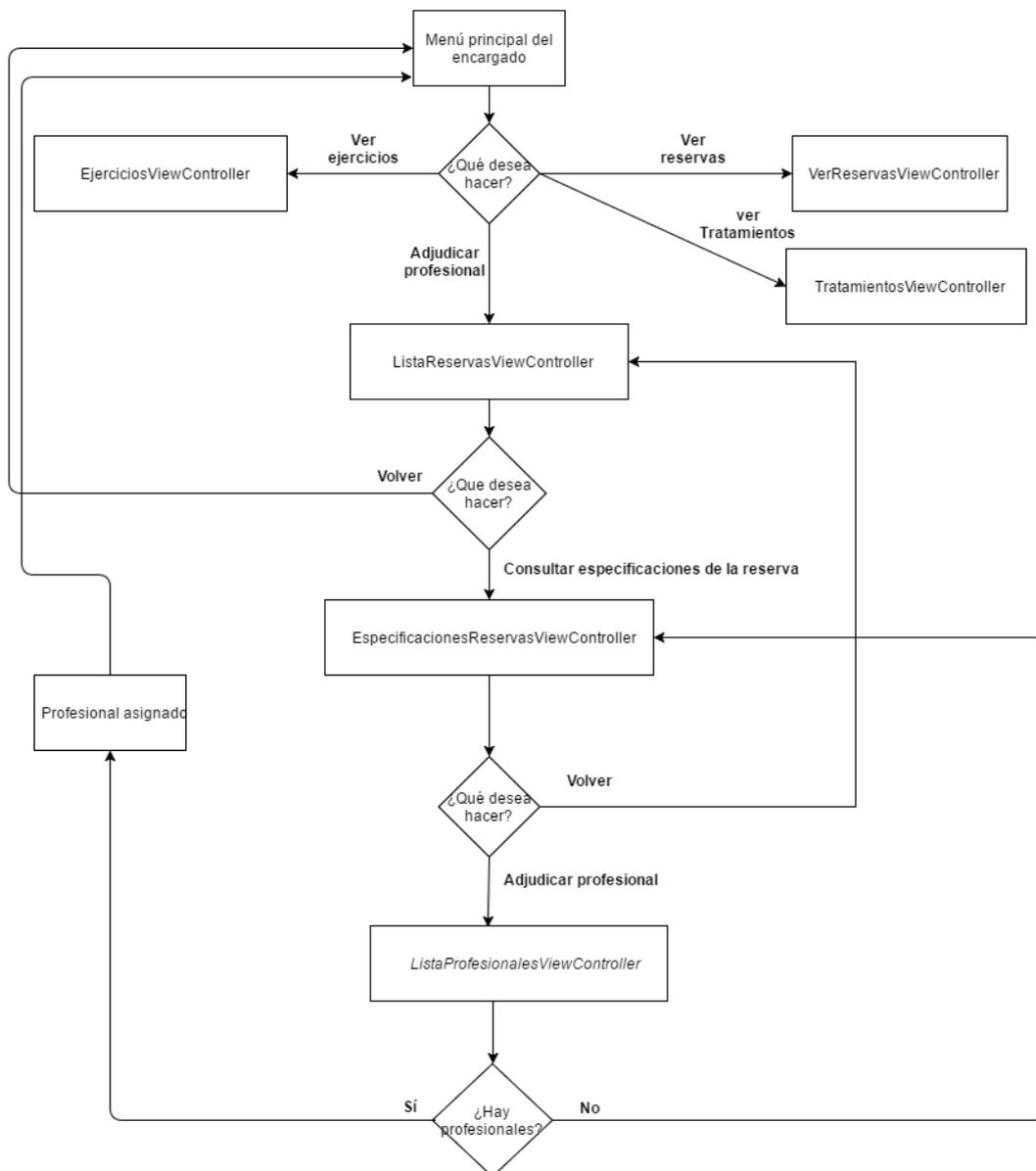


APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

características de la reserva seleccionada y se ofrece la opción de que el encargado pueda adjudicar profesional

En el caso de querer asignar el ejercicio a un paciente, una vez el profesional pulse en el botón “*Adjudicar Profesional*”, se hará uso de la clase *ListaProfesionalesViewController*, donde se mostrará una lista con todos los profesionales que tiene asignados la clínica donde se ha hecho la reserva. En caso de que no se tengan profesionales en esa clínica, se podrá volver al menú anterior.

Una vez que el profesional selecciona el paciente al que quiere asignar el ejercicio, se insertará el profesional en la propia reserva.



CÁPITULO 4: DESCRIPCIÓN TÉCNICA DE LA APLICACIÓN



Figura 42. Diagrama de flujo “Adjudicar paciente”.

4.3.5.2 Clases *TratamientosVC* y *EspecificacionesTratamientoVC*

Estas clases realiza las mismas funciones que las explicadas anteriormente al pulsar el botón “*Tratamientos*” en la clase *ProfesionalViewControllor* pero, en este caso, pulsando en el botón “*Tratamientos*” en el menú principal del encargado.

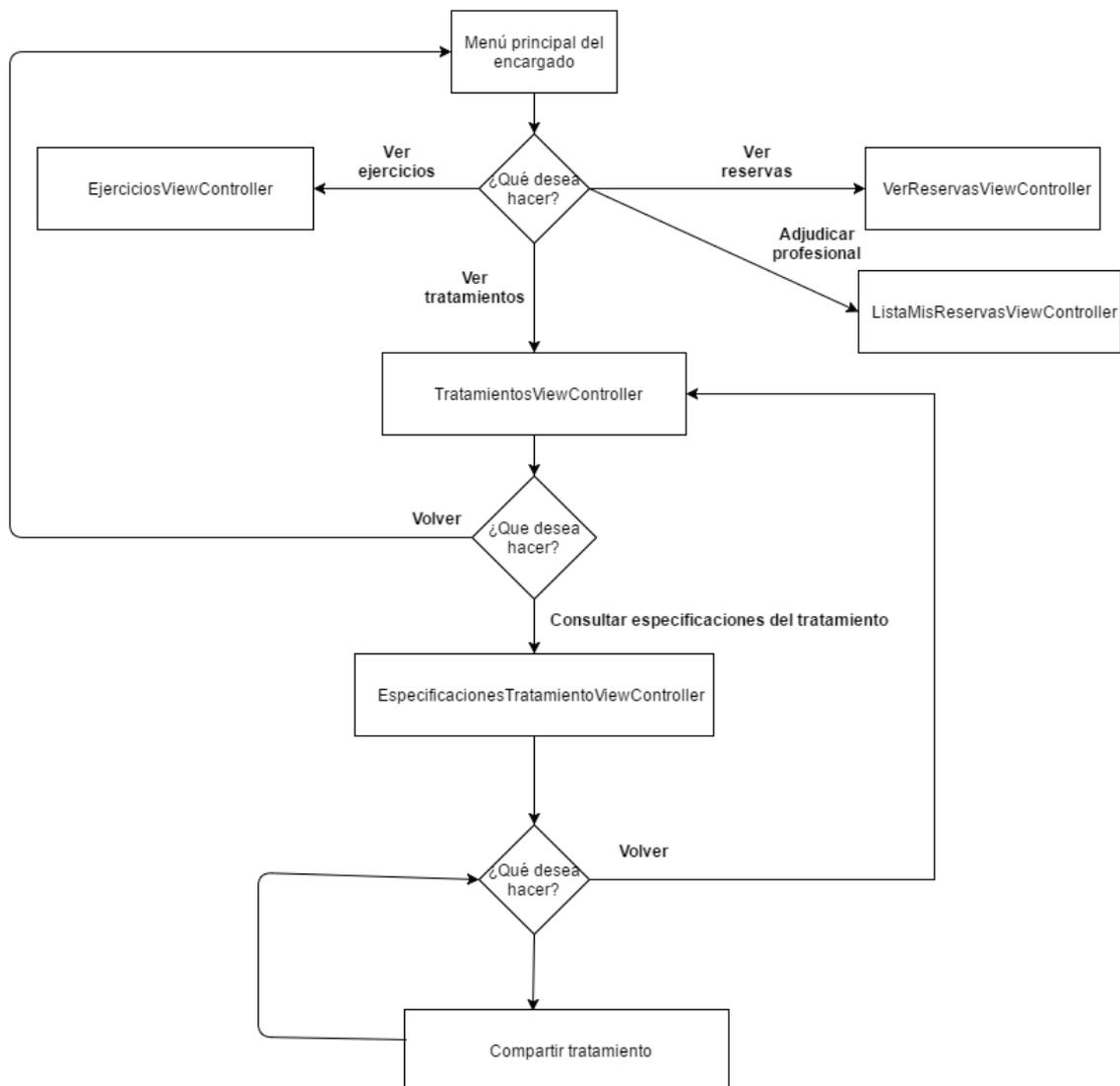


Figura 43. Diagrama de flujo “Ver tratamientos (encargado)”.



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

4.3.5.3 Clases EjerciciosVC, EspecificacionesEjercicioVC, ListaPacProVC y AsignarEjercicioVC

Estas clases realiza las mismas funciones que las explicadas anteriormente al pulsar el botón “Ejercicios” en la clase *ProfesionalViewController* pero, en este caso, pulsando en el botón “Ejercicios” en el menú principal del encargado.

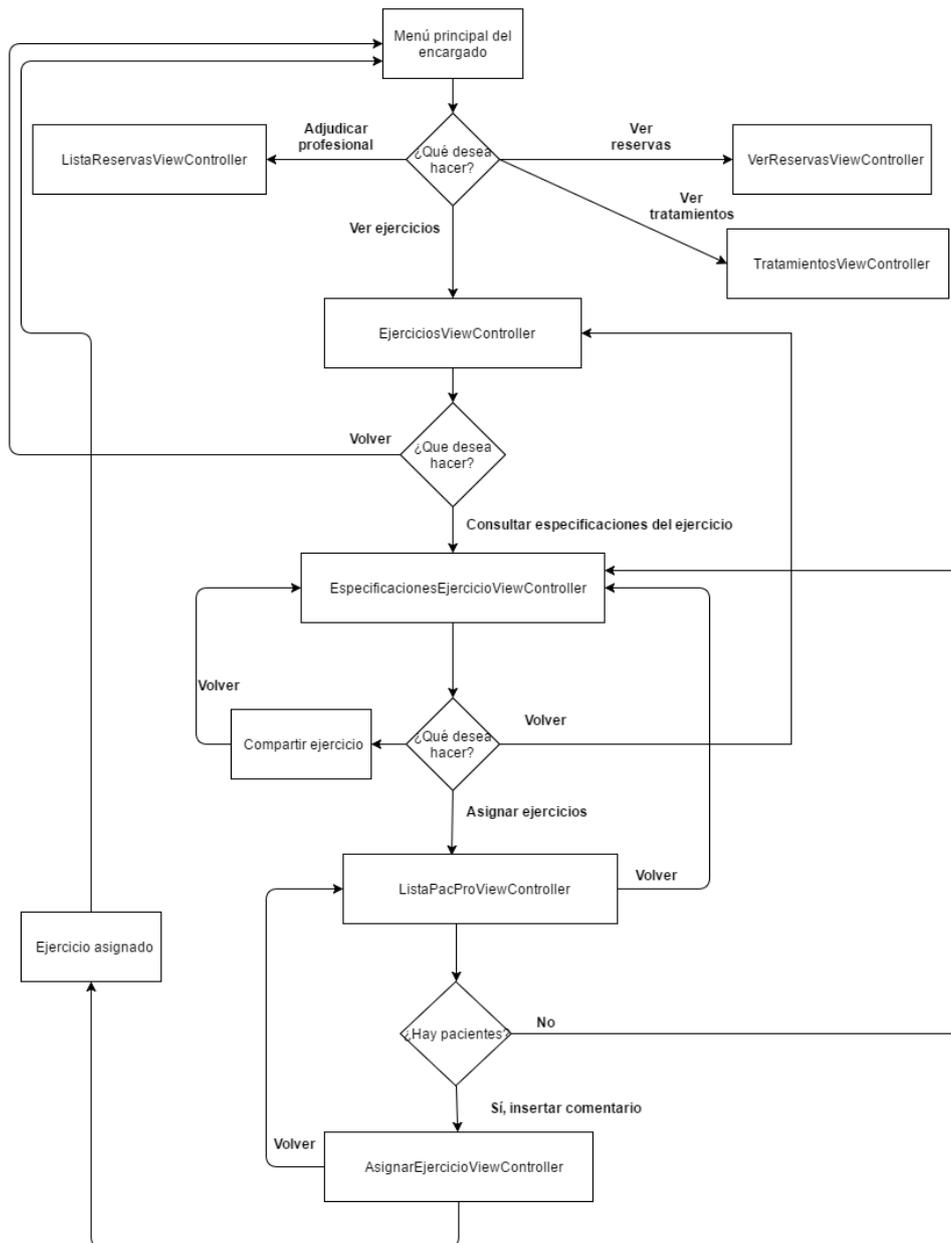


Figura 44. Diagrama de flujo “Asignar ejercicio (encargado)”.

CÁPITULO 4: DESCRIPCIÓN TÉCNICA DE LA APLICACIÓN



4.3.5.4 Clases *VerReservasVC* y *AnularCitaVC*

El diagrama siguiente describe las funcionalidades de un encargado cuando pulsa en el menú principal del paciente el botón “*Ver reservas*”.

Una vez pulsado el botón, en caso de haber el paciente tenga reservas, se hace uso de la clase *ListaReservasViewController*, donde se muestra una lista de todas las reservas realizadas, en caso contrario no se accederá a dicha clase y se mostrará un pop-up con la información pertinente.

Si se selecciona una clínica de la lista, se hará uso de la clase *AnularCitaViewController*, donde se muestra con detalle las características de la reserva seleccionada y donde el encargado tiene la opción de anular cualquier cita. En el caso de querer realizar esta opción, se pulsará el botón “*Anular cita*”.



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

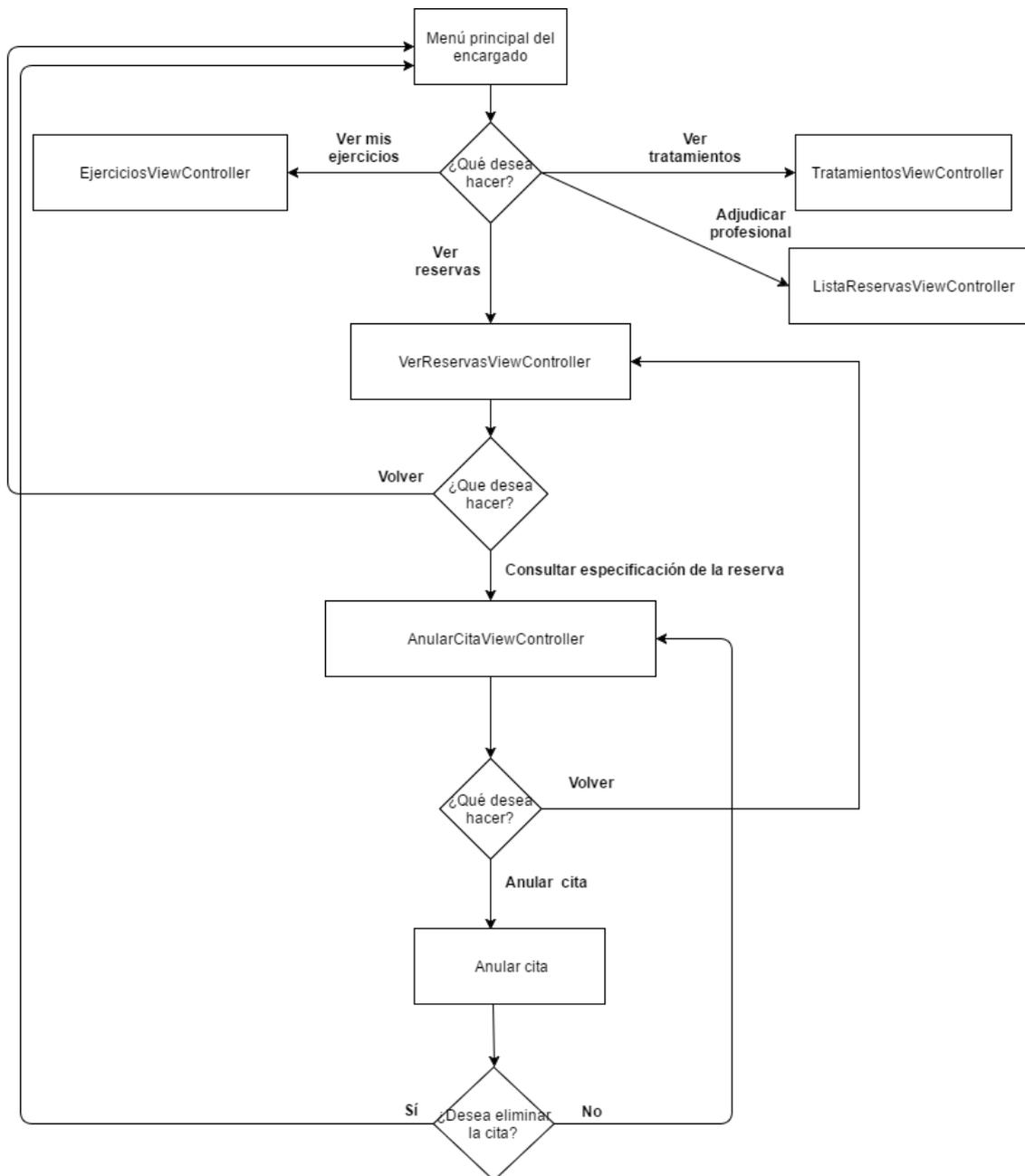


Figura 45. Diagrama de flujo “Ver reservas clínicas”.



Capítulo 5: Manual del usuario

En este capítulo se trata de elaborar una guía útil para el uso de la aplicación de una manera sencilla, rápida y eficaz. Se atenderán las necesidades que requerirán los diferentes usuarios de la aplicación, profesionales y clientes, de manera que se explique de manera clara la forma de intercomunicación entre ambos.

Para ello, se ha tratado que cada función de la aplicación quede explicada de forma que al usarla resulte totalmente intuitivo cada una de estas funciones, al igual que si se produjese algún error interno en alguna de estas funciones, se notificaría al usuario de manera que éste supiese con certeza si es fallo al realizar el mismo alguna función o es fallo de la propia aplicación.

A continuación, se mostrarán cada una de las partes de las que consta la aplicación junto con una descripción de cada una de ella, sus posibles errores y sus requisitos para el correcto funcionamiento.

5.1 Login

La pantalla de “Login” es la pantalla inicial de la aplicación. Es dónde, tanto pacientes como profesionales introducen sus datos para acceder a su menú correspondiente de la aplicación. Si el usuario no está registrado deberá pasar a hacerlo pulsando el botón de registrarse (esta funcionalidad se detallará más adelante).

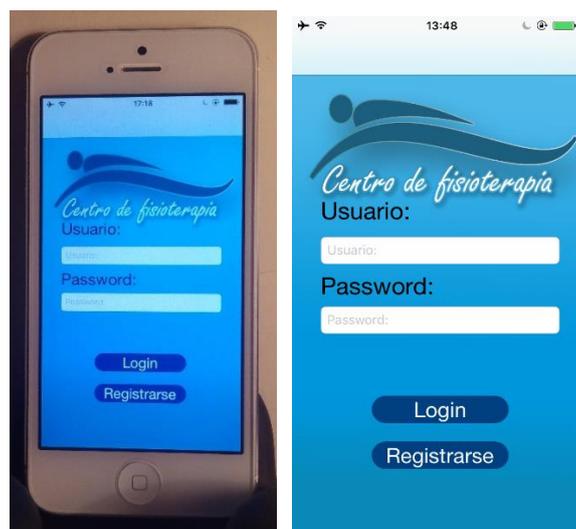


Figura 46. Pantalla inicial de la aplicación.



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

El proceso de “Login” puede dar lugar a una serie de errores si los datos no se introducen correctamente o bien por la ausencia de conexión a internet.

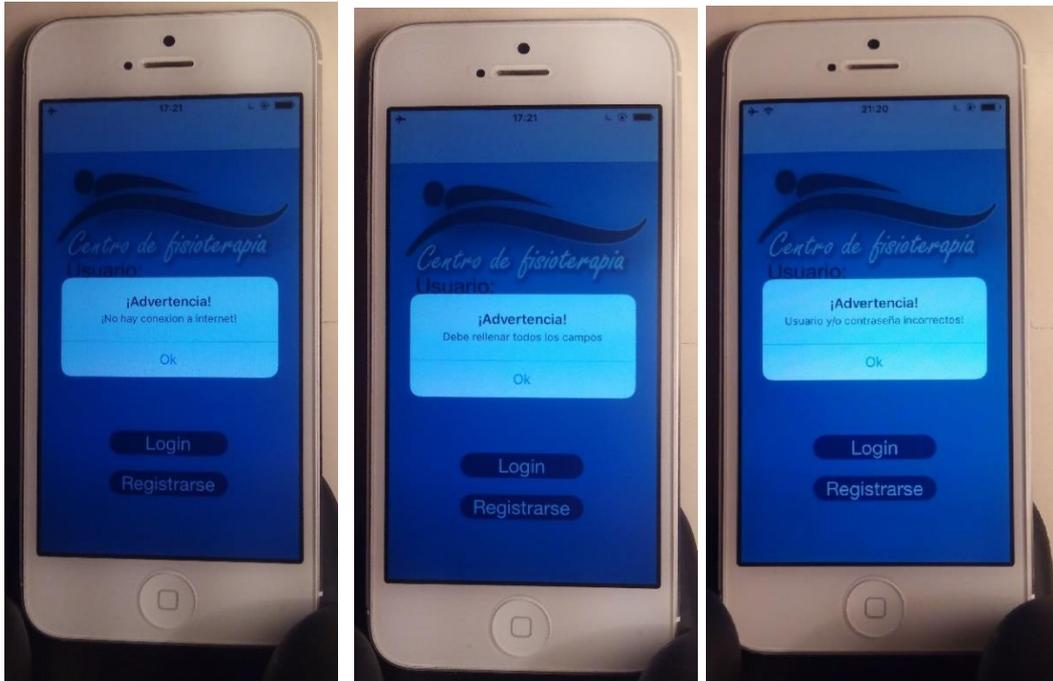


Figura 47. Errores a la hora de producirse el logeo en la aplicación (móvil).

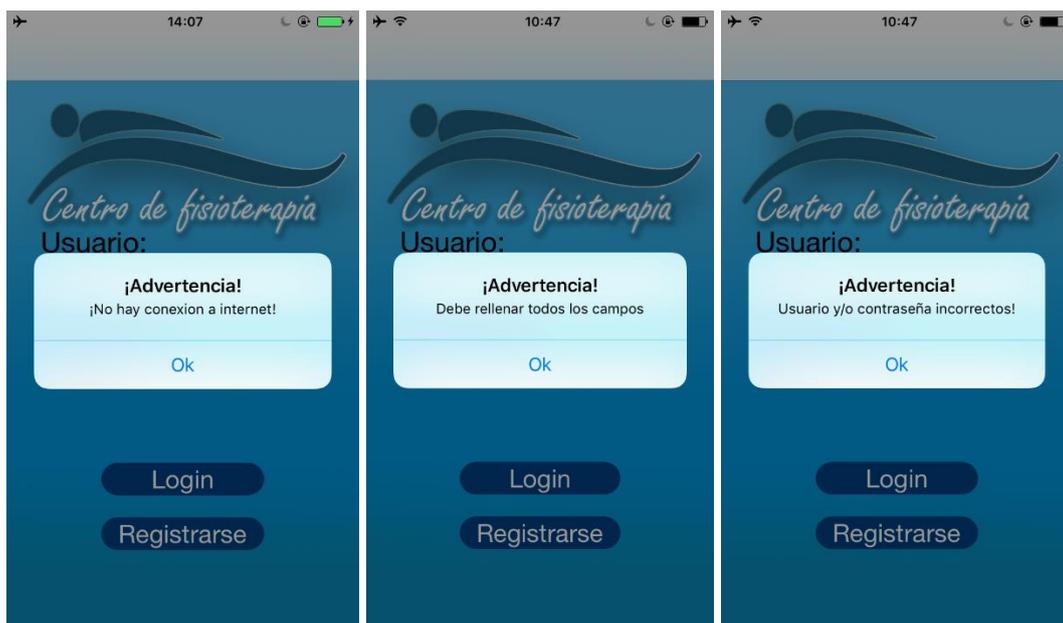


Figura 48. Errores a la hora de producirse el logeo en la aplicación.



5.2 Registro de pacientes

Cuando un usuario utiliza por primera vez la aplicación, debe acceder desde la pantalla inicial de “Login” al menú de registro de pacientes pulsando sobre el botón “Registrarse”.

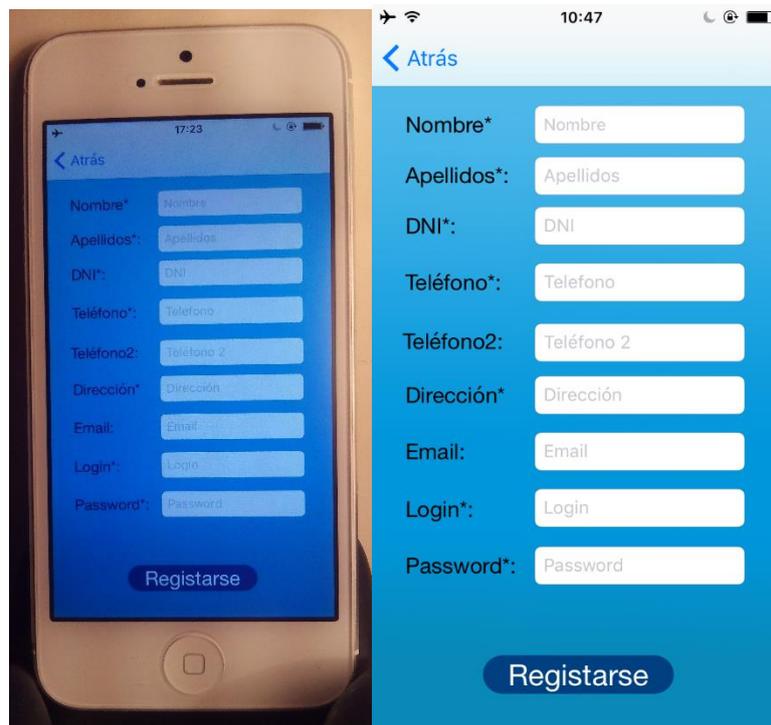
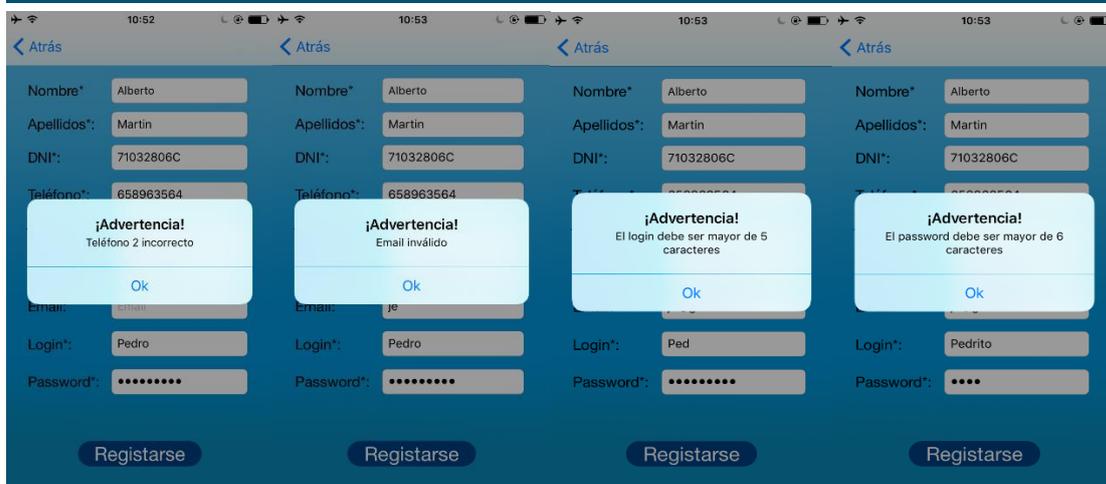
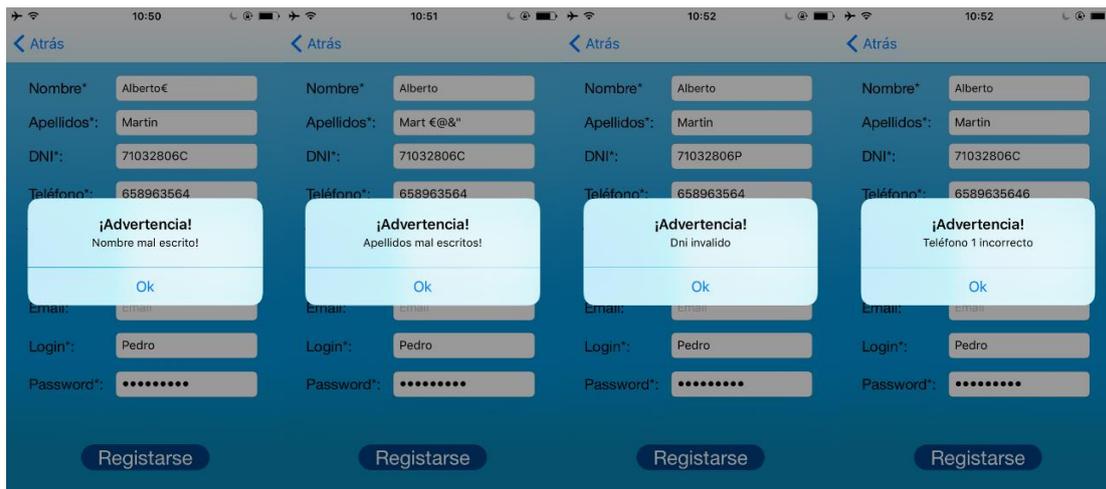


Figura 49. Pantalla de registro de la aplicación.

Una vez el usuario está en la pantalla de “Registro de pacientes”, deberá rellenar correctamente cada uno de los campos obligatorios (marcados con *) para registrarse adecuadamente. De no ser así, se notificarán al usuario el campo introducido de forma errónea. A continuación, se muestran estas notificaciones:



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA





CÁPITULO 5: MANUAL DEL USUARIO

Figura 50. Errores en el proceso de registro del paciente.

Si el usuario introduce bien todos los datos, aún puede que el proceso de registro sea erróneo, dado que el login o el DNI puede que ya estén en uso. En el caso del DNI, salvo que el usuario ya esté registrado o hayan suplantado su identidad, no ofrecerá mayor problema. Para el caso del campo del login, puede estar en uso por cualquier otro usuario que se haya registrado en la aplicación. A continuación, se muestran dichos errores:

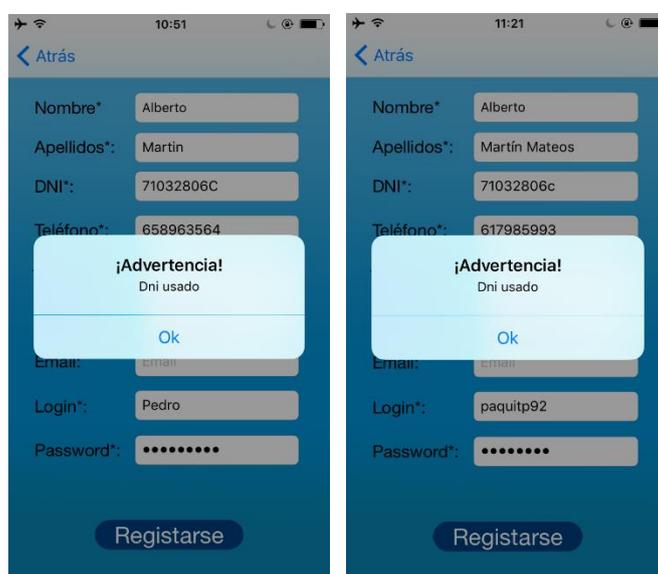


Figura 51. Errores producidos al introducir datos correctos.

Finalmente, si el usuario ha logrado realizar todos los pasos anteriores con éxito, se dará lugar al registro del paciente y se retornará a la pantalla de “Login”.



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

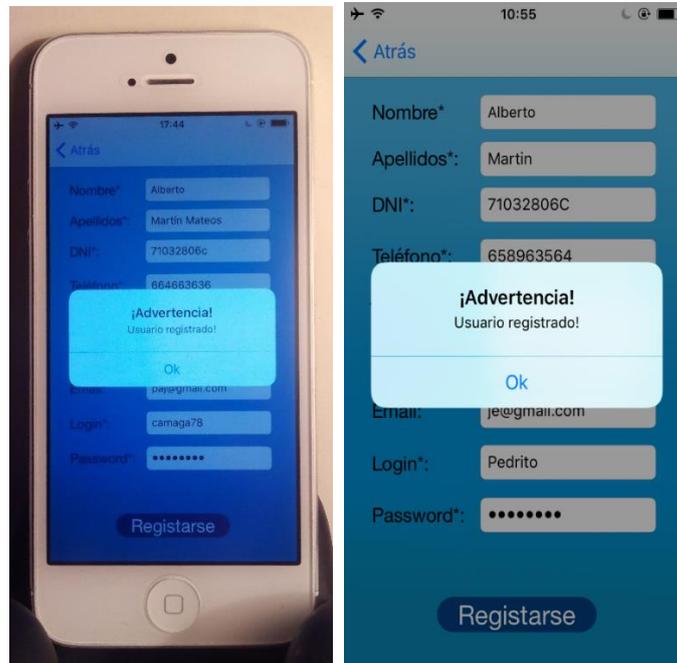


Figura 52. Registro del paciente.

5.3 Menú principal del encargado

Una vez un usuario se ha logueado en la aplicación, puede acceder a uno de los tres menús principales de los que consta la aplicación, encargado (además de ser profesional), profesional o usuario/paciente.

En este apartado se van a abordar las funcionalidades cuando el usuario que se loguea es el encargado de la aplicación. Una vez se produce este logueo, se accede al menú principal del encargado.



Figura 53. Pantalla de menú principal del encargado.

Este menú principal ofrece 4 opciones:

- Adjudicar paciente al profesional.
- Asignar ejercicio al paciente.
- Ver tratamientos.
- Ver todas las reservas de los pacientes.

5.3.1 Adjudicar paciente

Adjudicar un paciente a un profesional es una de las funcionalidades clave de la aplicación. Consiste en recoger todas aquellas reservas que se han producido y asignarles un profesional adecuado dependiendo de los requerimientos que tenga cada una de ellas.

Una vez se pulsa en el botón “Adjudicar paciente”, pueden ocurrir dos cosas: en caso de que haya reservas disponibles sin tener un profesional adjudicado, se mostrarán tantas como haya en una lista, enumeradas por el lugar del dolor en el que el paciente sufre la lesión. Por el contrario, si no hay reservas disponibles, se notifica de ello.



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA



Figura 54. No hay reservas pendientes de adjudicar.

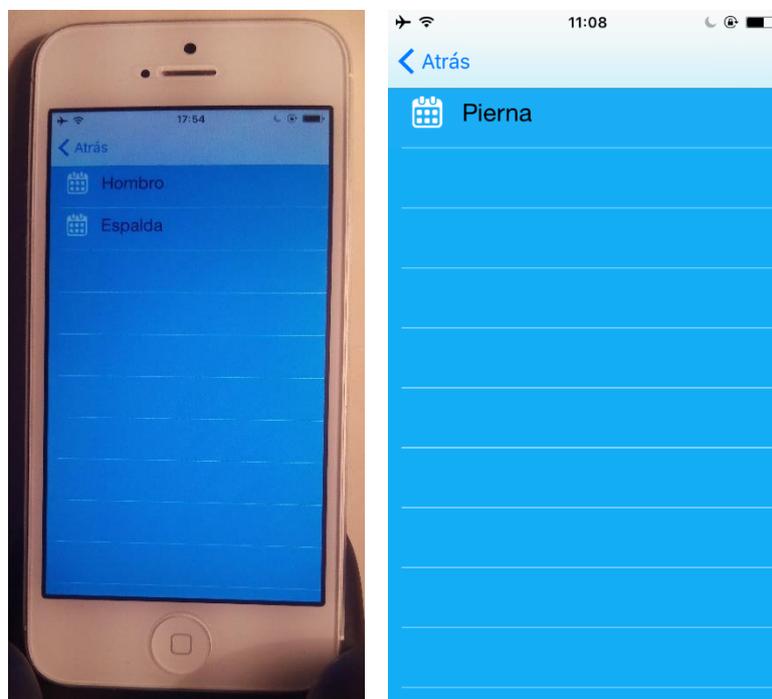


Figura 55. Lista de reservas sin adjudicar profesional.

CÁPITULO 5: MANUAL DEL USUARIO



Cuando se muestra la lista de reservas, se selecciona una para poder ver sus detalles y poder asignar el profesional mas adecuado a las necesidades que se tienen.



Figura 56. Detalles de la reserva.

Una vez se han analizado los detalles de la reserva a la que se le quiere adjudicar un profesional, pulsando en el botón “Adjudicar profesional”, se accederá a ver la lista de profesionales que el encargado puede adjudicar a la reserva. En caso de que no haya en ese momento profesionales, en la lista solamente se indicara que no hay profesionales.



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

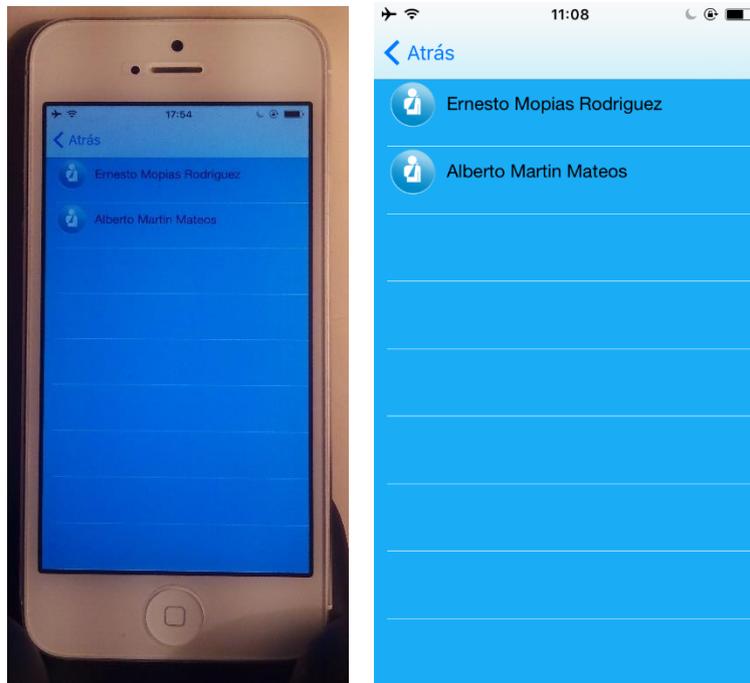


Figura 57. Lista de profesionales disponibles para adjudicar.

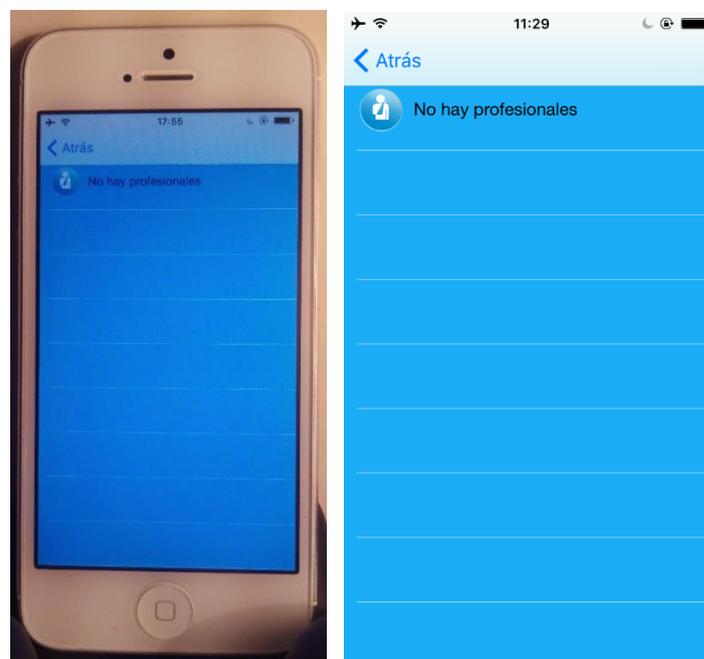


Figura 58. No hay profesionales disponibles para adjudicar.

CÁPITULO 5: MANUAL DEL USUARIO



Finalmente, cuando se selecciona un profesional, este queda adjudicado a la reserva seleccionada y se retorna al menú principal notificando que la acción se ha realizado correctamente.



Figura 59. Paciente adjudicado al profesional.

5.3.2 Asignar ejercicio al paciente

En este apartado, el encargado de la aplicación, ahora cumpliendo la función de profesional en la clínica, podrá consultar la lista de ejercicios que tiene disponibles, en caso de que haya, y tendrá la posibilidad de asignar alguno de ellos a sus pacientes junto con un breve comentario.

Primeramente, el encargado podrá ver la lista de ejercicios que tiene disponibles para asignar a sus pacientes.



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA



Figura 60. Lista de ejercicios disponibles.

Cuando el encargado presiona cualquier ejercicio de la lista, se introducirá detalles de dicho ejercicio y se ofrecen las opciones de compartirlo o de asignarlo a uno de sus pacientes.



CÁPITULO 5: MANUAL DEL USUARIO



Figura 61. Detalles del ejercicio

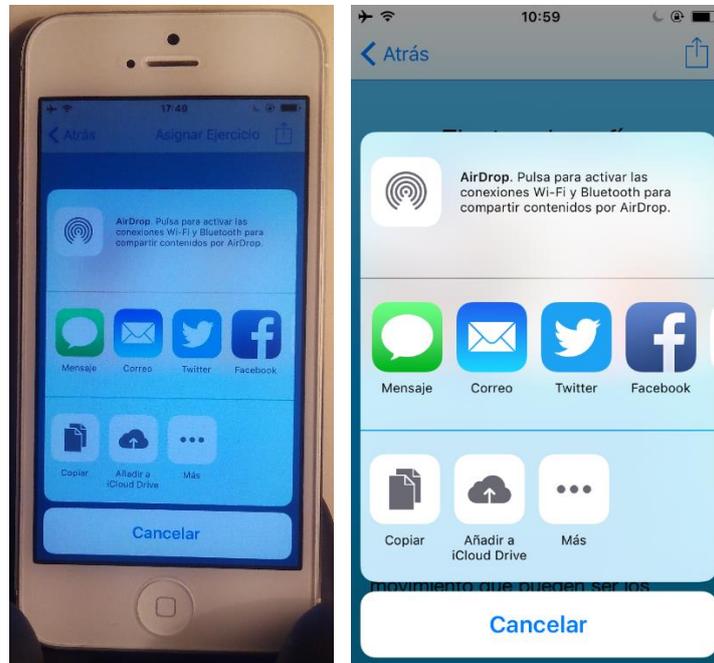


Figura 62. Compartir ejercicio

Si el encargado quiere asignar un ejercicio a uno de sus pacientes, deberá pulsar el botón “Asignar ejercicio”. A continuación, en caso de contar con pacientes, se mostrará un listado con todos sus pacientes. En caso contrario, se indicará que no hay pacientes.



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

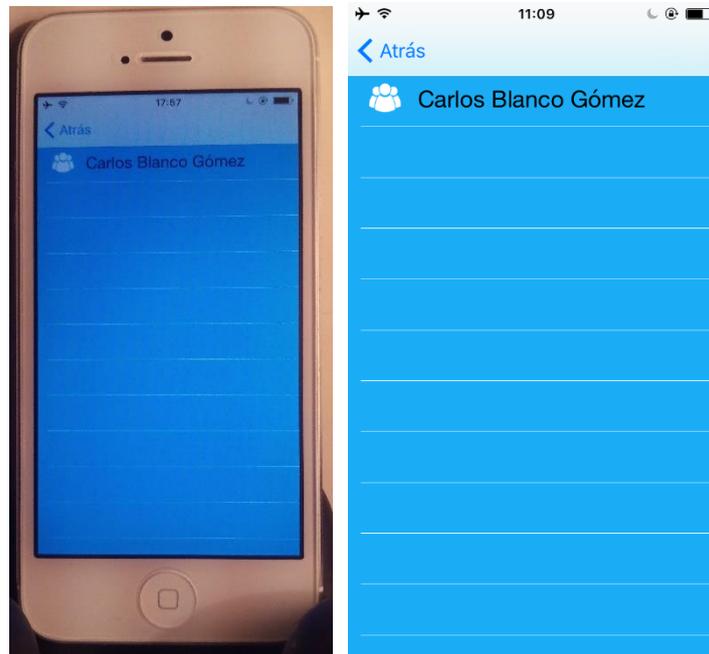


Figura 63. Pacientes a los que asignar ejercicios.



Figura 64. Profesional sin pacientes.

Una vez que el encargado asignar un paciente al que asignarle el ejercicio, podrá recomendar mediante un comentario especificaciones sobre el ejercicio que ha asignado.

CÁPITULO 5: MANUAL DEL USUARIO

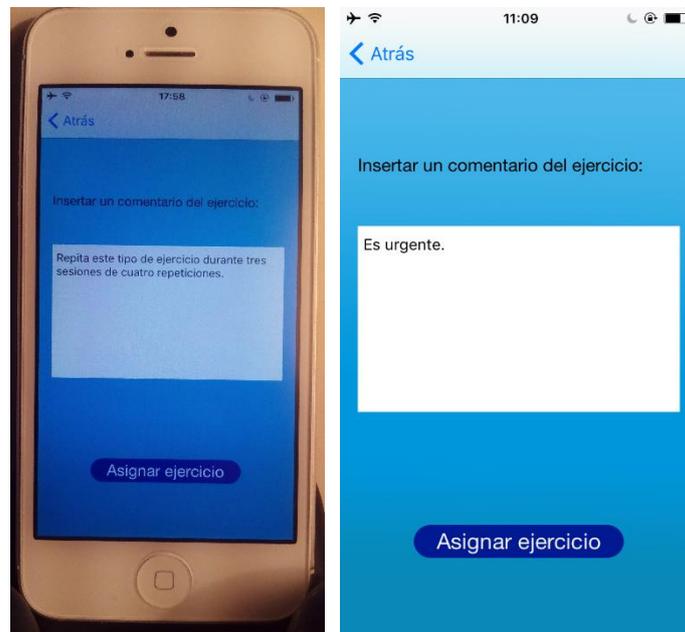


Figura 65. Comentario del profesional sobre el ejercicio.

Finalmente, después de que el profesional pulse el botón “Asignar ejercicio”, se volverá al menú principal del encargado notificando que la acción se ha realizado correctamente.





APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

Figura 66. Ejercicio asignado correctamente.

5.3.3 Ver tratamientos

Otra opción que ofrece el menú principal del encargado es la de poder ver todos los tratamientos que se realizan en sus clínicas. Primero se mostrará una lista con todos ellos y después, tras seleccionar uno de ellos, se detallarán sus características y se ofrecerá la opción de compartirlo.



Figura 67. Lista de tratamientos.

Tras pulsar en un tratamiento, se mostrarán los detalles de este. De la misma manera se oferta la opción de compartir, al igual que ocurre con los ejercicios.



Figura 68. Detalles del tratamiento.

5.3.4 Ver reservas de las clínicas

La última opción que ofrece el menú principal del encargado es poder ver todas las reservas que se han producido en sus clínicas y, si se ha producido algún contratiempo, poder eliminar la reserva. En caso de no haber reservas, se notificará.



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA



Figura 69. No hay reservas en ninguna clínica.

En el caso de que si se hayan producido reservas en alguna de las clinicas, se mostrará una lista con todas ellas indicando su fecha y su hora.

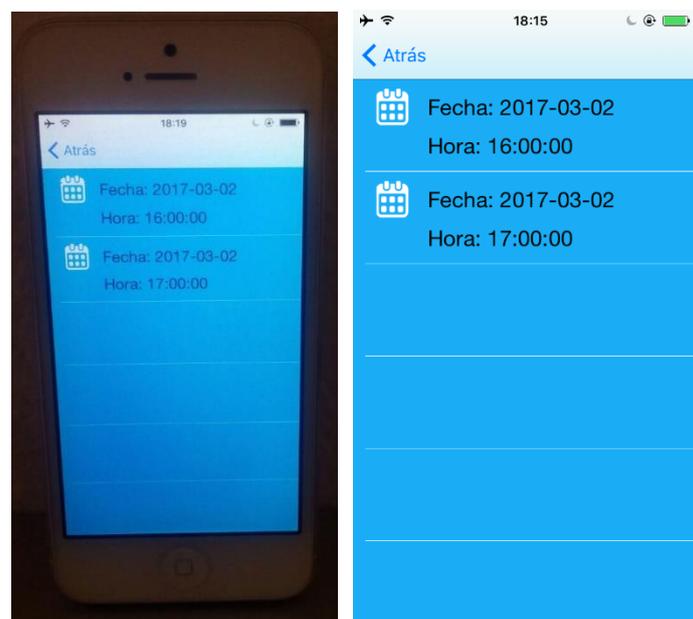


Figura 70. Lista de reservas.

CÁPITULO 5: MANUAL DEL USUARIO

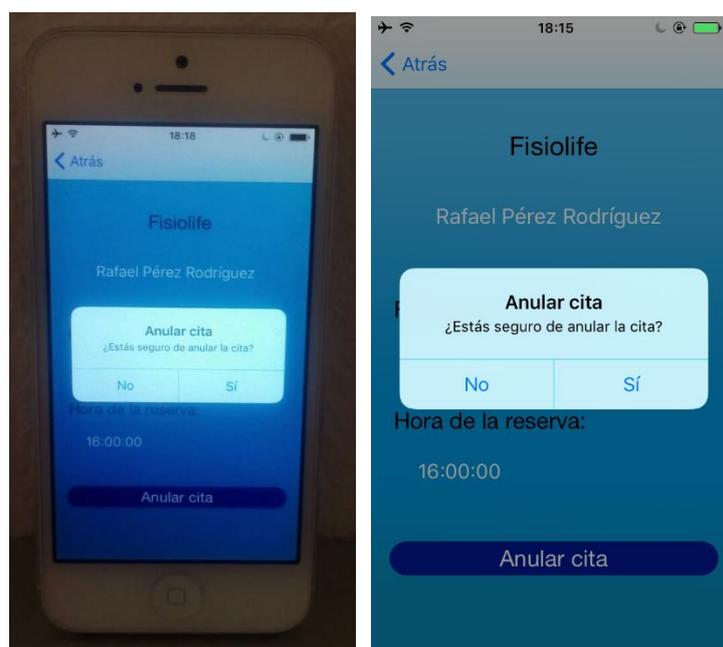


Si se pulsa en alguna de las reservas, se mostrarán los detalles de la misma y se ofrece al encargado la posibilidad de anular la reserva en caso de que haya habido algún tipo de problema.



Figura 71. Detalles de la reserva

Al anular la cita, el encargado podrá anularla. En caso de presionar “Anular Cita”, se le pedirá al usuario que confirme la anulación de la cita.





APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

Figura 72. Anular cita

Después de anular la cita, se volverá al menú principal del encargado notificando que la acción se ha realizado correctamente.



Figura 73. Cita anulada

5.4 Menú principal del profesional

En este apartado se van a abordar las funcionalidades cuando el usuario que se loguea es un profesional de alguna clínica. Una vez se produce este logueo, se accede al menú principal del profesional.



Figura 74. Pantalla del menú principal del profesional.

Este menú principal ofrece 4 opciones:

- Ver tratamientos.
- Asignar ejercicio al paciente.
- Contactar con la clínica.
- Lista de citas.

5.4.1 Ver tratamientos

Esta opción es similar a la vista anteriormente en el apartado 5.3.3 para el usuario logueado como encargado de la aplicación.

5.4.2 Asignar ejercicio

Esta opción es similar a la vista anteriormente en el apartado 5.3.2 para el usuario logueado como encargado de la aplicación.



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

5.4.3 Contactar con la clínica.

La opción contactar permite a un profesional contactar con la clínica en la que trabaja para hacer cualquier tipo de consulta sobre cualquiera de sus pacientes. Pulsando el botón “Contactar”, se abrirá el menú del correo electrónico del teléfono junto con la dirección de correo de la clínica en la que trabaja el profesional.

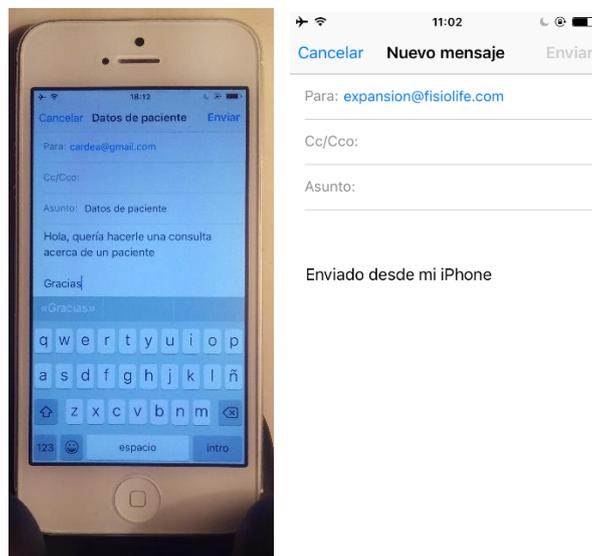


Figura 75. Menú de correo del teléfono.

Una vez que el correo electrónico se ha enviado, se volverá al menú principal del profesional y se notificará que la acción de ha realizado correctamente.



Figura 76. Mensaje de e-mail enviado.

5.4.4 Lista de citas

La última opción que ofrece el menú principal del profesional es la que permite a éste saber el total de citas que tiene concertadas. En caso de no tener ninguna se notificará.





APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

Figura 77. No hay citas para el profesional.

En caso de que el profesional si tenga citas concertadas, se mostrará una lista con todas ellas.

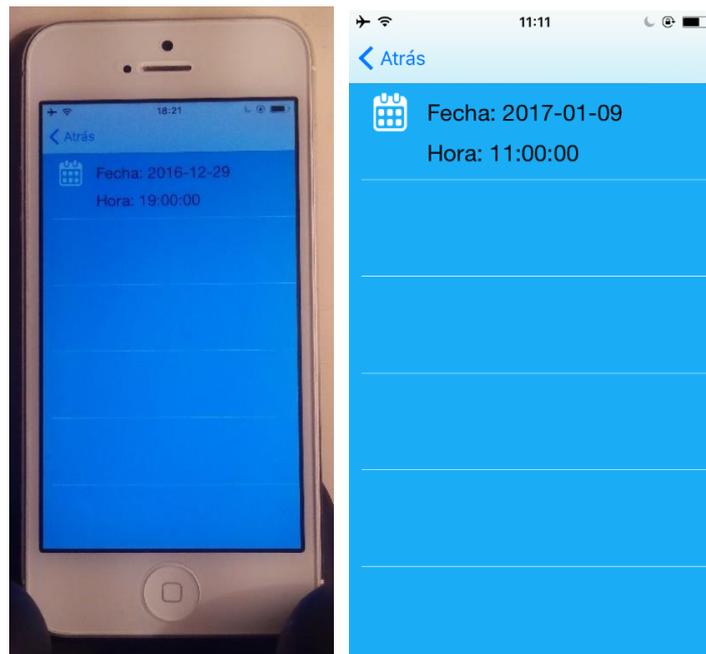


Figura 78. Lista de citas concertadas.

Si el profesional quiere saber más detalles sobre una cita en concreto tendrá que pulsa sobre ella.

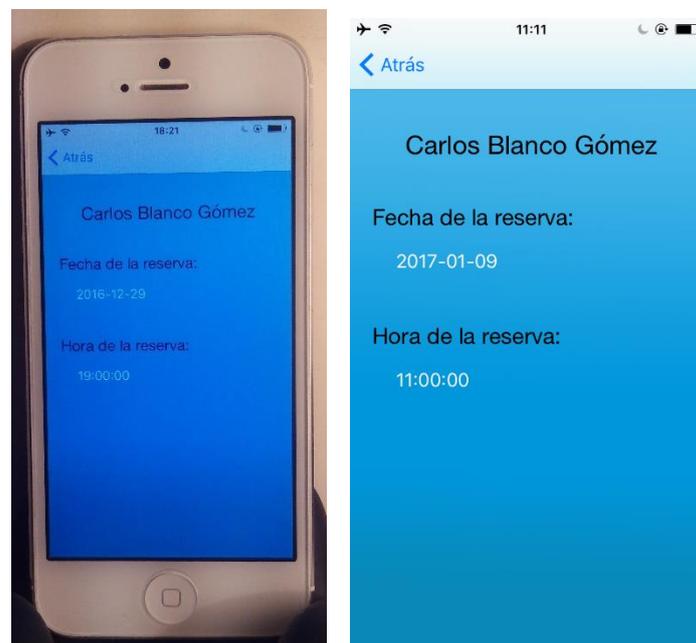


Figura 79. Detalles de la cita.

5.5 Menú principal del usuario/paciente.

En este apartado se van a abordar las funcionalidades cuando el usuario que se loguea es un paciente. Una vez se produce este logueo, se accede al menú principal del usuario o paciente.





APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

Figura 80. Pantalla de menú principal del paciente.

Este menú ofrece cuatro opciones:

- Clínicas.
- Reservar una cita.
- Lista de ejercicios del paciente.
- Mis reservas.

5.5.1 Clínicas

Una de las funcionalidades que ofrece el menú principal del paciente es la de consultar todas las clínicas que hay disponibles para realizar cualquier consulta u obtener información sobre ellas. En el caso de no haber ninguna disponible se notificará.

Si hay clínicas disponibles, se mostrará una lista con todas ellas para que luego el paciente pueda consultar detalles sobre la clínica elegida.



Figura 81. Lista de clínicas

CÁPITULO 5: MANUAL DEL USUARIO

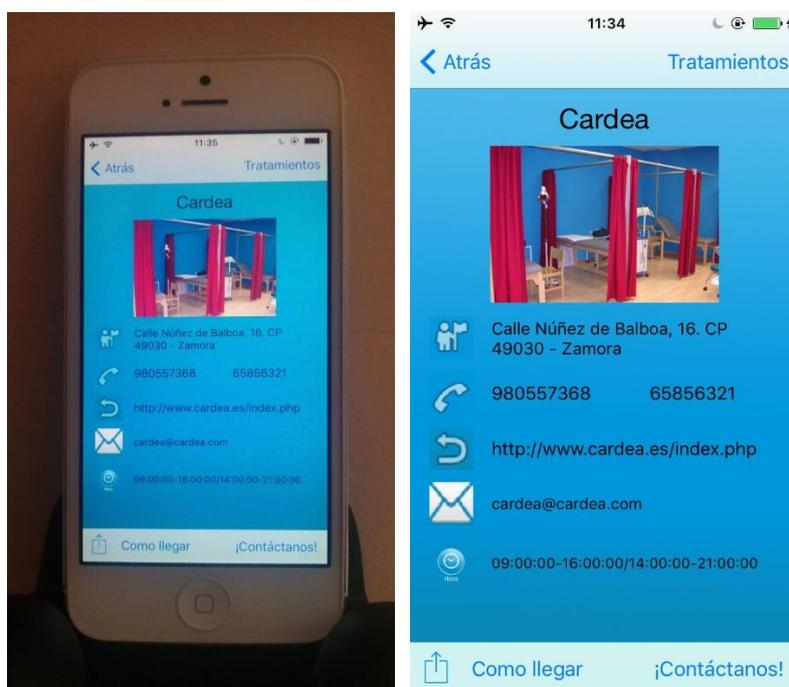


Figura 82. Detalles de la clínica seleccionada.

Al consultar los detalles de la clínica, se podrán realizar diversas acciones:

- Llamada a la clínica. Presionando el botón “Teléfono”.
- Contactar con la clínica via e-mail. Presionando el botón “¡Contáctanos!”. Similar a la comentada anteriormente en el menú del profesional.
- Compartir la clínica. Similar a la función de compartir un tratamiento o un ejercicio.
- Saber la situación de la clínica.
- Ver la lista de tratamientos de la clínica. Presionando el botón “Tratamientos”. Se obtendrá un listado con todos los tratamientos de la clínica y sus detalles, similar a la acción de ver tratamientos en los menús principales del encargado y del profesional.

Si el paciente quiere saber donde está la clínica, deberá pulsar el botón “Como llegar” y se abrirá un mapa con la ubicación de la clínica.



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

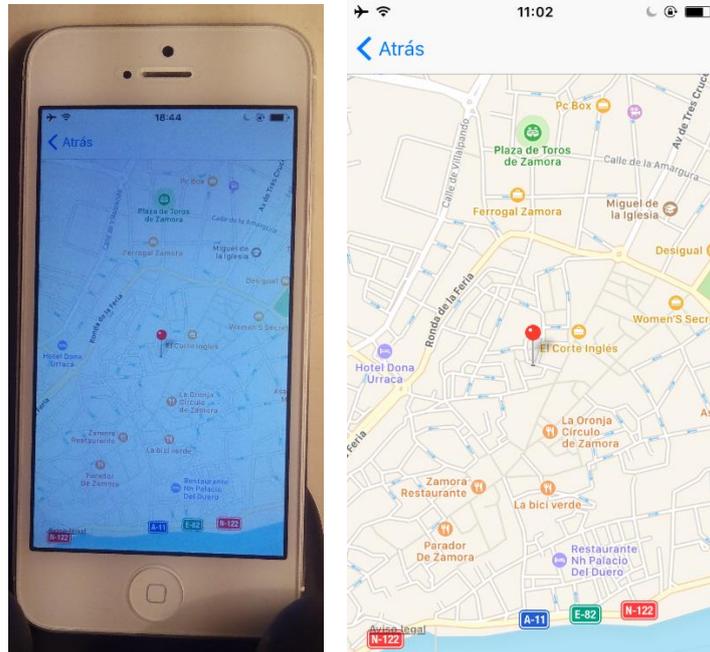


Figura 83. Cómo llegar a la clínica.

5.5.2 Reservar cita

Otra opción que ofrece el menú del paciente es reservar una cita en alguna de las clínicas disponibles. En caso de no haber clínicas disponibles se notificará al usuario.

Al igual que en la funcionalidad anterior, se mostrará un listado con todas las clínicas disponibles y tras seleccionar una de ellas, se mostrarán los detalles de ésta.

CÁPITULO 5: MANUAL DEL USUARIO

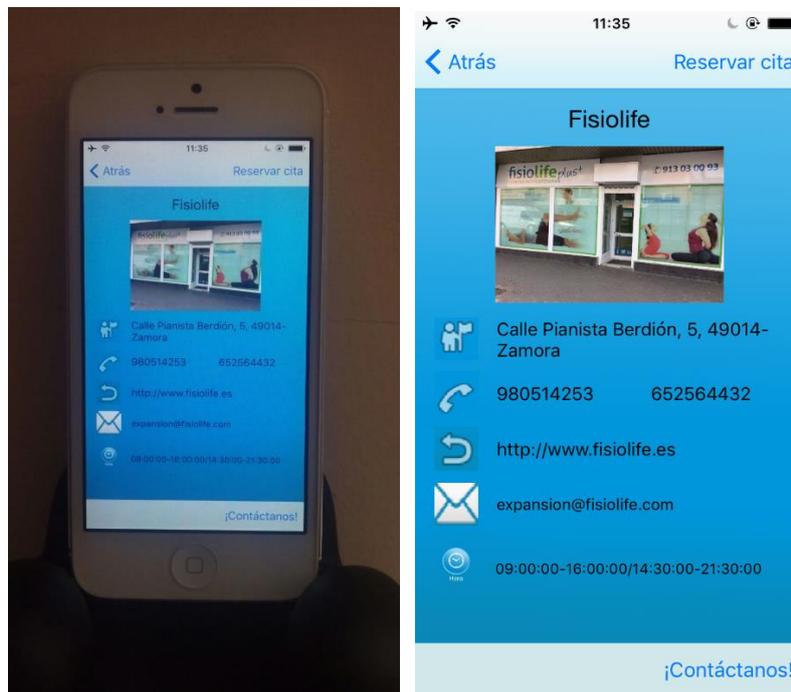


Figura 84. Detallas de la clínica para reservar cita.

Al consultar los detalles de la clínica, se podrán realizar varias acciones:

- Llamar a la clínica. Pulsando el botón del “teléfono”.
- Contactar vía e-mail con la clínica. Pulsando el botón “¡Contáctanos!”.
- Reservar cita.

Si el paciente desea reservar una cita en la clínica, deberá pulsar el botón “Reservar cita”. Posteriormente, se pasarán a rellenar los datos necesarios para ello.



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

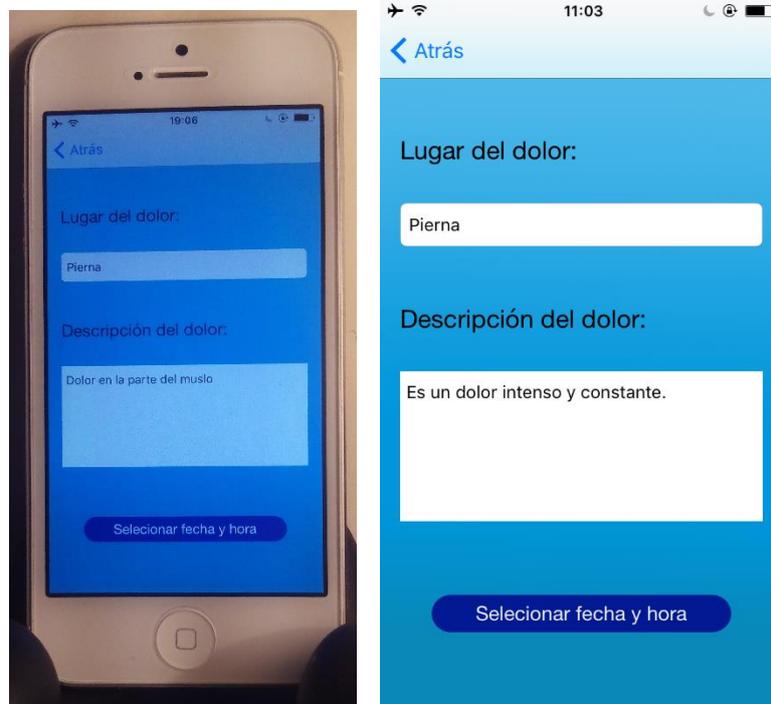


Figura 85. Datos para reservar.



Figura 86. Fecha y hora de la reserva.

CÁPITULO 5: MANUAL DEL USUARIO



Después de introducir todos los datos necesarios para realizar la reserva, se mostrará con detalle todos ellos y se podrá consultar la hora más próxima disponible con los datos previamente introducidos.



Figura 87. Detalles de la reserva.



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

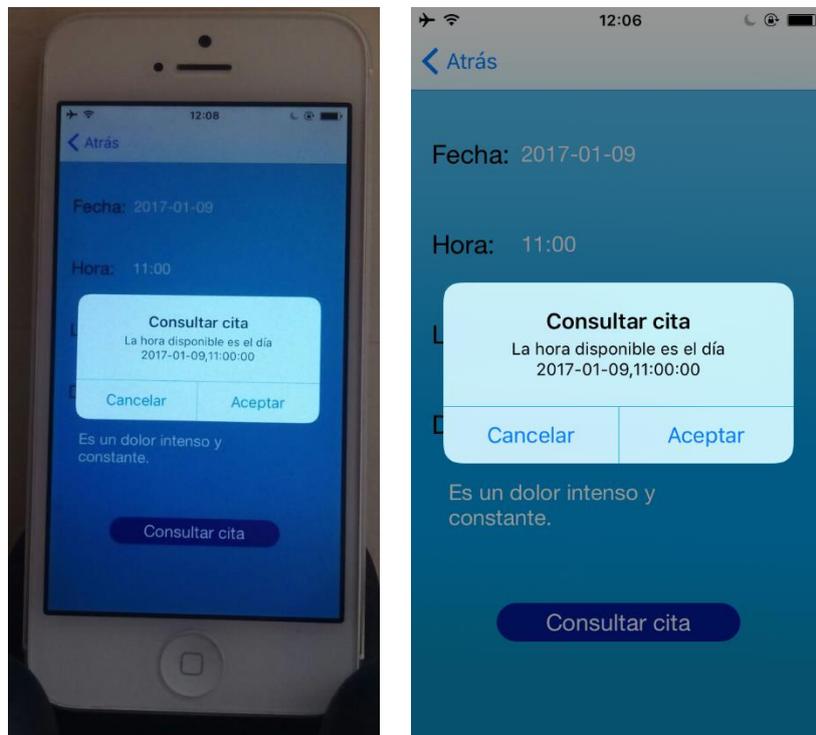


Figura 88. Fecha y hora de la posible reserva.

Si el usuario acepta la consulta, se volverá al menú principal del paciente y se notificará que la acción se ha realizado correctamente.

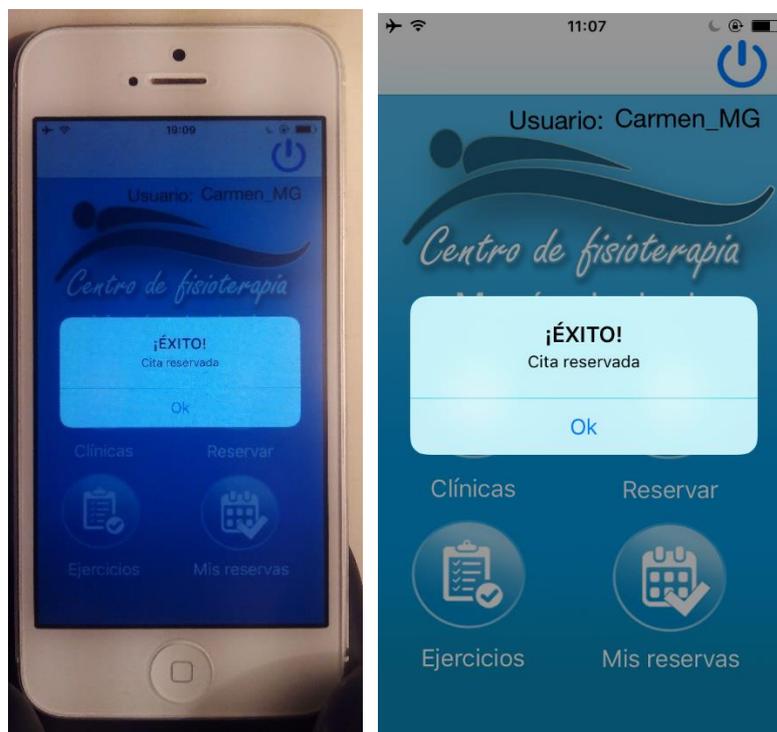


Figura 89. Cita reservada.

5.5.3 Lista de ejercicios del paciente

Esta opción permite al paciente ver que ejercicios se le han sido asignados y tener un breve comentario de su profesional sobre cómo realizarlo. Si no hay ejercicios asignados al paciente, se notificará. Por último, se le permitirá compartirlo de la misma manera que al encargado o al profesional.



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA



Figura 90. No hay ejercicios para el paciente.

Primero, el paciente obtendrá un listado con todos sus ejercicios asignados y pulsando sobre el que desee visualizará la demás información.

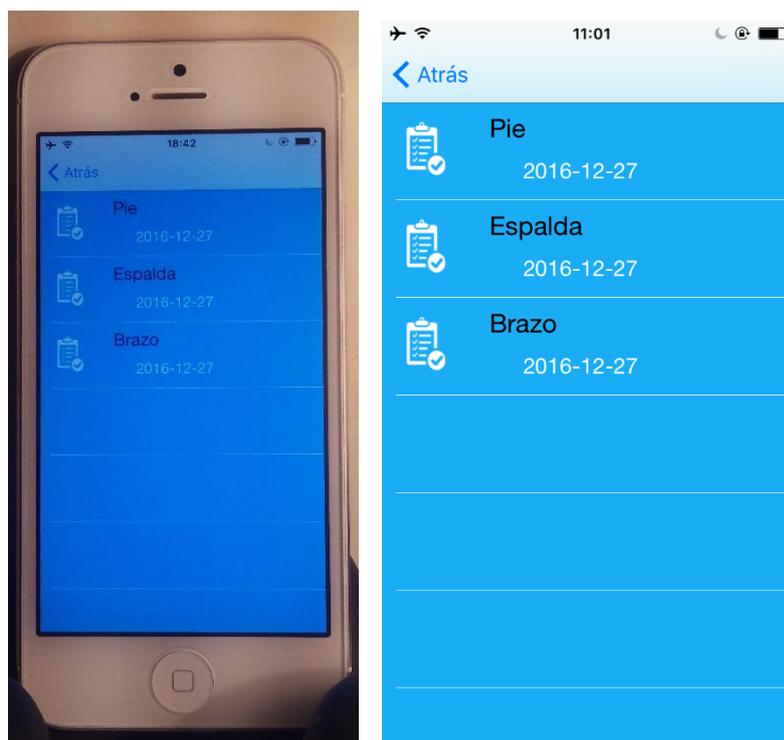




Figura 91. Listado de ejercicios del paciente.

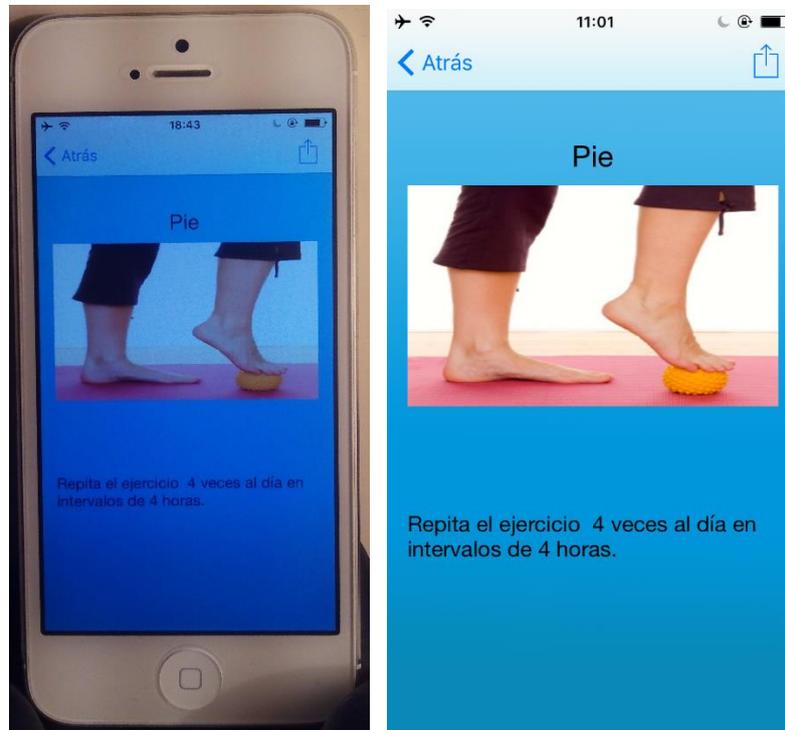


Figura 92. Detalles del ejercicio.

5.5.4 Mis reservas

La última opción permite al paciente consultar todas las reservas que tiene en las clínicas de la aplicación. De no tener ninguna, se notificará. Es una opción idéntica a “Ver reservas” (5.3.4) del menú del encargado, con la diferencia de que, en el encargado, la lista es de todas las reservas de las clínicas y en el paciente, de sus reservas.

5.5 Desconexión

Por último, el usuario de la aplicación independientemente de su rol en ella puede optar por:

- Desconectarse.
- Seguir logueado.



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

En el caso de que el usuario se desconecte, se volverá a la pantalla inicial de “Login” notificándose que la acción se ha realizado correctamente.

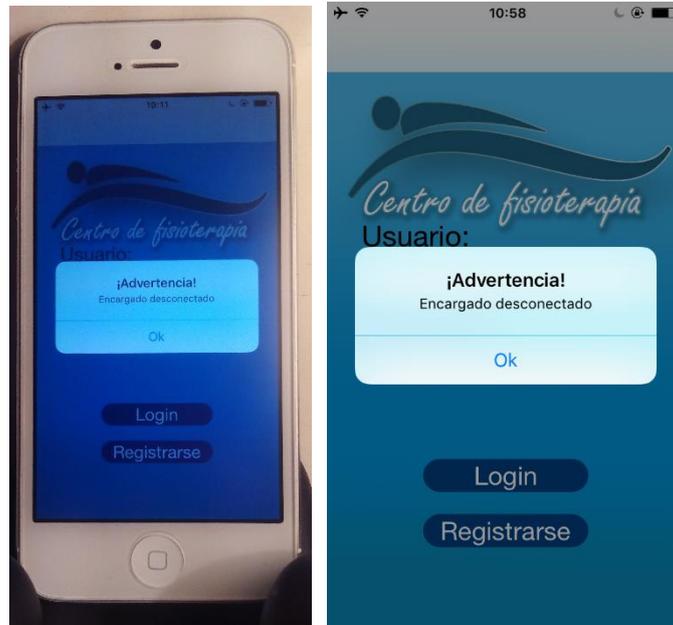


Figura 93. Encargado desconectado.

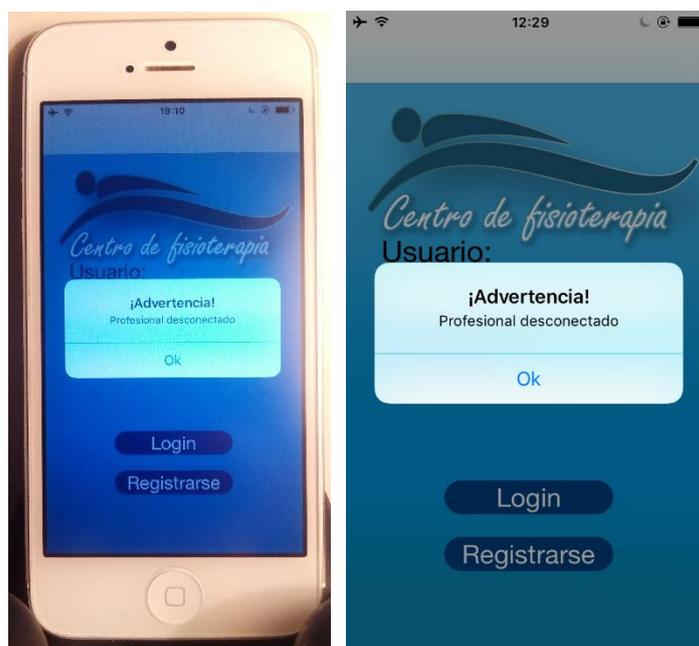


Figura 94. Profesional desconectado.

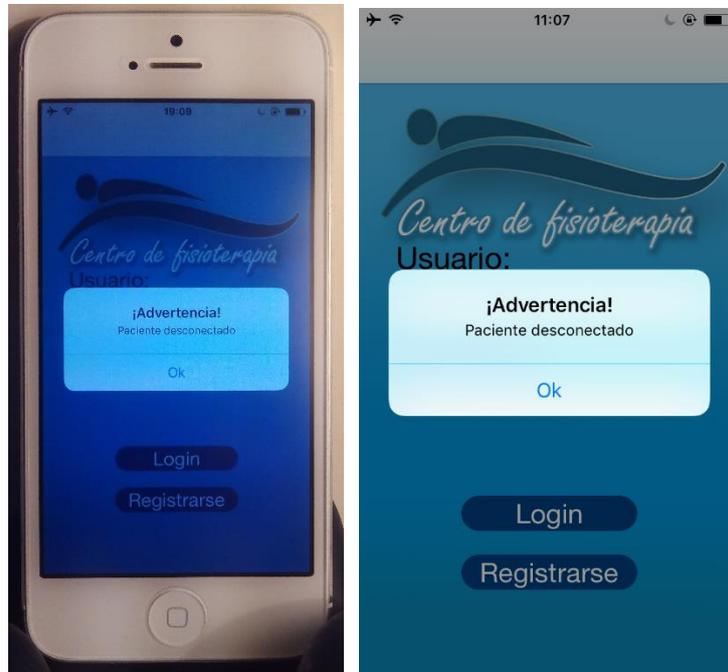


Figura 95. Paciente desconectado.

Finalmente, si el usuario decide seguir logueado, cuando vuelva a utilizar la aplicación de nuevo aparecerá en su menú correspondiente de manera directa sin pasar por la pantalla de “Login”.

5.6 Vista global de la aplicación

Para final este capítulo, se muestra una visión global del “Main Storyboard” de la aplicación en la que se recogen todos “View Controller” por los que pasa la aplicación al realizar cada una de sus funcionalidades.



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

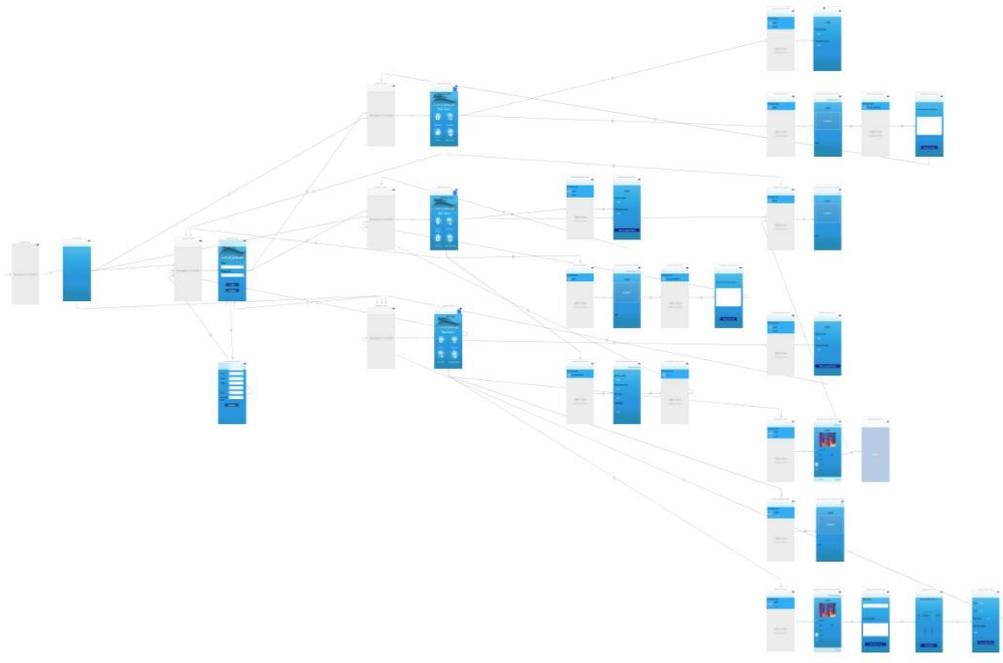


Figura 96. “Main StoryBoard” de la aplicación.



Capítulo 6: Presupuesto económico.

Para alcanzar los objetivos propuestos, ha sido necesario el uso de varios dispositivos detallados a continuación:

- Un ordenador MAC mini para el desarrollo software con 8GB de RAM y doble núcleo valorado en aproximadamente 800€.
- Una pantalla LG de 21'' valorado en 150€.
- Un teclado Magic keyboard compatible con iOS valorado en 120€.
- Un teléfono iPhone con soporte para iOS 10, equivalente a un iPhone 5 o superior. Estos teléfonos están valorados alrededor de 450€.

El equipo utilizado no se utiliza única y exclusivamente para este proyecto y, en algunos casos es prestado con lo que se ha asumido que la sexta parte son gastos asociados al proyecto.

Por otra parte, no hay costes vinculados a ningún tipo de software, ya que se trata de un software libre y gratuito si disponemos del equipo descrito. La cuenta de desarrollador básica también es gratuita.

Por último, se ha estimado que para la realización del proyecto han sido necesarias unas 240 horas, excluyendo horas de aprendizaje al nuevo lenguaje. Ya que la experiencia que tengo no es mucha, valoró mis horas como programador junior en 12€ la hora, que sumado al valor del equipo hardware, valoró mi proyecto en 3100€.



Capítulo 7: Conclusiones, líneas futuras y experiencia personal

En este capítulo se describe las conclusiones a las que se ha llegado después del desarrollo de este Trabajo Fin de Grado, nuevas líneas futuras que podrían mejorar la aplicación.

7.1 Conclusiones

Este Trabajo Fin de Grado surge de la necesidad de complementar a la plataforma Web y a la aplicación Android de apoyo a centros de fisioterapia. Hoy en día, la población dirigida a este tipo de aplicaciones es muy variada y, por lo tanto, es preciso la implementación de esta herramienta en los sistemas operativos más utilizados.

Las aplicaciones móviles a día de hoy están instaladas en la mayor parte de los hogares a los que se dirige esta aplicación, directa o indirectamente, es decir, poniendo en ejemplo a una pareja de ancianos, ellos seguramente no hagan uso de este producto, pero es muy posible que sus hijos si lo hagan, de manera que evitaremos, por ejemplo, que esta pareja tenga que acudir al centro elegido para reservar una cita. Este es un ejemplo de los múltiples que puede haber dentro de las herramientas similares a estas en cualquier área de la salud y, que puede resolver, bien por una razón o por otra, un problema.

Ya hablando de la aplicación en sí, se ha tenido en cuenta que cuente con sistema de control de acceso por login y contraseña cifrada que incrementa la seguridad y confidencialidad del usuario, y si el usuario no se desconecta al dejar de usar la aplicación, quedará logueada para próximos usos. Además, se ha logrado implementar cada una de las funcionalidades requeridas y se ha tratado, pensando en el usuario final, que tenga una interacción con la aplicación sencilla y clara que le permita el uso sin ningún tipo de problemas.

En líneas generales, se ha cumplido con los objetivos que se presentaron, se ha logrado adaptar en la medida de lo posible a las aplicaciones ya desarrolladas, y se ha obtenido un resultado satisfactorio una vez ya ha sido probada en un terminal real. Además, se ha conseguido adquirir una serie de conocimientos y habilidades técnicas sobre toda la tecnología utilizada que han dado lugar a realizar los objetivos expuestos.

7.2 Líneas futuras

CÁPITULO 7: CONCLUSIONES, LÍNEAS FUTURAS Y EXPERIENCIA PERSONAL



Como surge en este tipo de aplicaciones, siempre hay funcionalidades que van surgiendo mientras se va desarrollando el producto o incluso después de acabarlo. Estas nuevas funciones siempre son de agradecer en un futuro para un usuario, que el cliente sienta que hay preocupación por su experiencia de navegación y que se preocupan por él, que la aplicación está siendo revisada y no se ha quedado obsoleta.

Se proponen las siguientes mejoras para la aplicación en futuras versiones:

- **Notificaciones:** En el proyecto actual no hay desarrolladas ningún tipo de notificaciones, pero si es verdad que una buena mejora sería incluirlas, además de varios tipos:
 - **Profesional asignado:** Una vez que el encargado haya asignado el profesional a la reserva hecha por paciente, una notificación para avisar de que ya ha sido atendida.
 - **Ejercicio asignado:** Una vez que el profesional haya asignado un ejercicio a un paciente, notificación para comunicar al paciente que tiene un nuevo ejercicio por realizar.
 - **Recordatorio de cita:** Aviso mediante una notificación para que unas horas se recuerde que hay una cita reservada.
- **Base de datos interna:** Volcado de la información de un usuario de la base de datos externa a la interna una vez que se loguea por primera vez en la aplicación por internet.
- **Nuevas funcionalidades:** Desde las aplicaciones móviles se pueden realizar las funcionalidades más importantes, pero si es cierto que existen varias más que realiza la plataforma Web que podían ser implementadas. Algunas de estas funcionalidades son:
 - **Borrar ejercicios:** Los profesionales desde la aplicación pueden adjudicar ejercicios, pero no los pueden borrar en caso de equivocación y lo deben realizar desde la Web. De esta manera se proporcionaría una forma más cómoda de hacer esta acción.
 - **Incluir nuevos ejercicios y tratamientos:** En la Web es posible hacerlo, pero en la aplicación no para que resultara más sencillo, pero si se podría incluir para que al encargado le resultara más cómodo.
 - **Dar de alta/baja profesionales:** La clínica cuenta con una plantilla de profesionales, pero esta puede incrementarse o puede disminuir. El encargado de la clínica sería la



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

- persona que realizaría las bajas o las altas de nuevos profesionales.
- Dar de baja un usuario: Al igual que un usuario puede registrarse, podría tener una opción donde pudiese borrar su cuenta.
- Localización: Una posible mejora en esta funcionalidad sería marcar la ruta que tiene que hacer cada usuario para llegar a la clínica en la que ha reservado cita.
- Idiomas: Actualmente, la aplicación esta implementada en castellano e inglés, pero podría ser una buena mejora su traducción a algún otro idioma.

7.3 Experiencia personal

Este proyecto me ha producido una gran satisfacción a nivel personal, me ha hecho desarrollar habilidades que no tenía tan destacadas como pueda ser una planificación semanal y mensual de las tareas que tenía que ir realizando, comunicación con antiguos compañeros que podrían saber de ciertos temas en los que estaba trabajando, mejorar mis capacidades de síntesis y análisis de información recopiladas, adaptación a nuevos lenguajes de programación, etc...

El inicio fue bastante complicado al no tener apenas idea de cómo usar el entorno en el que iba a trabajar, tanto el entorno propio de *OS X* como en el entorno de desarrollo "*Xcode*". Poco a poco fui descubriendo cosas nuevas a través de videos, artículos y libros que me hacían progresar cada vez más rápido. Además, mi experiencia con el lenguaje de programación utilizado, *Swift*, era nula por lo que también supuso un gran reto aprender a dar los primeros pasos.

Una vez comprendido y aprendido algunas de las nociones básicas sobre todos los aspectos anteriormente comentados, inicié el proyecto en el que cada vez iba aprendiendo cosas nuevas además de las ya aprendidas. A veces, era necesario volver a versiones anteriores para mejorar el desarrollo y que fuese más eficiente. Múltiples errores me acompañaron por el camino, pero solo eran piedras en el camino a la meta que se iban solventando con el paso del tiempo y, estos errores son los que me permitían luego avanzar mucho más rápido y aprender a desarrollar nuevas habilidades, tanto de código como de análisis de éste.

Puedo concluir que, aunque en algunas fases del proyecto pensaba que no iba salir, mi afán de superación hizo que saliese adelante, encontrará siempre una nueva solución y lograr finalizar este Trabajo Fin de Grado que, aunque haya sido laborioso y a veces complicado, puedo decir que ha sido una de las grandes experiencias de mi vida



Capítulo 8: Bibliografía

- Android Developers. (2017). *Arquitectura de Android*
<https://developer.android.com/guide/index.html> (02-2017).
- Antón Rodríguez, M. (2015). *Introducción a iOS*. Valladolid: Universidad de Valladolid (02-2017).
- Antón Rodríguez, M., & Pérez Juárez, M. Á. (2015). *Introducción a PHP*. Valladolid: Universidad de Valladolid (02-2017).
- Apple (2016). *iOS Technology Overview*.
https://developer.apple.com/library/content/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/Introduction/Introduction.html#//apple_ref/doc/uid/TP40007898-CH1-SW1 (02-2017).
- Apple5x1. (2016). *Adaptación a las nuevas versiones iOS*.
<https://apple5x1.com/ios-10-es-la-version-con-la-tasa-de-adopcion-mas-rapida-en-la-historia-de-ios/> (02-2017).
- Apple developer. (2017). API Guide.
<https://developer.apple.com/library/prerelease/content/navigation/> (02-2017).
- ArquitecturaJava. (2016). *Servicios Rest*. <http://www.arquitecturajava.com/que-es-rest/> (02-2017).
- Barreno, D. (2012). *Solución de acentos para bases de datos*.
<http://www.danibarreno.com/la-solucion-a-los-problemas-con-los-acentos-en-php-mysql-y-html/> (02-2017).
- El Mundo (2015). *Negocio de las aplicaciones móviles de la salud*.
<http://www.elmundo.es/salud/2014/03/19/5329df6f22601dff5e8b457b.html> (02-2017).
- Expansión (2015). *Uso sistemas operativos a nivel mundial*.
<http://www.expansion.com/economia-digital/companias/2015/12/09/56684be1ca474151018b4590.html> (02-2017).
- Fernández Muñoz, J.C. (2015). *Aprendiendo Swift 2*. Gabel Studios S.L. Madrid. España. (02-2017).
- Fundación Telefónica (2016). *La sociedad de la información en España 2016*.
http://www.fundaciontelefonica.com/arte_cultura/publicaciones-listado/pagina-item-publicaciones/itempubli/483/ (02-2017).
- Gómez Tejedor, David. (2015). *Web de apoyo a una aplicación Android de fisioterapia*. Valladolid: Universidad de Valladolid. (02-2017).
- Inmobi. (2016). *Estado de los desarrolladores de aplicaciones móviles*.
<http://www.inmobi.com/blog/2016/02/08/2016-state-of-mobile-app-developers> (02-2017).



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

- Lancetalent. (2014). *Tipos de aplicaciones móviles*.
<https://www.lancetalent.com/blog/tipos-de-aplicaciones-moviles-ventajas-inconvenientes/> (02-2017).
- Marketshare. (2016). *Comparación de mercados de sistemas operativos móviles*.
<http://marketshare.hitslink.com/operating-system-market-share.aspx?> (02-2017).
- Mysql. (2017). *MySQL*. <https://www.mysql.com/> (02-2017).
- Pwc. (2016). *Top health industry issues of 2016*.
<https://www.pwc.com/us/en/health-industries/top-health-industry-issues/assets/2016-us-hri-top-issues.pdf> (02-2017).
- Mixpanel. (2016). *Comparativa de versiones iOS*.
https://mixpanel.com/trends/#report/ios_10/from_date:-280,report_unit:week,to_date:0 (02-2017).
- OMS. (2000). *Informe sobre la salud en el año 2000*.
http://www.who.int/whr/2000/en/whr00_es.pdf?ua=1 (02-2017).
- Rodríguez Cayetano, M. (2016). *Introducción al sistema operativo Android*.
Valladolid: Universidad de Valladolid. (02-2017).
- Statista. (2016). *Reparto del mercado de las aplicaciones móviles*.
<https://www.statista.com/statistics/263795/number-of-available-apps-in-the-apple-app-store/> (02-2017).
- Swift. (2017). *El lenguaje Swift*. <https://developer.apple.com/swift/> (02-2017).
- TAD. (2016). *Mejores 50 apps de la salud*.
<http://boletines.prisadigital.com/Informe-TAD-50-Mejores-Apps-de-Salud.pdf> (02-2017).
- W3C. (2017). *W3C Consortium*. <http://www.w3c.es/> (02-2017).
- Windows Phone. (2016). *Introducción a Windows Phone*.
<http://conceptodefinicion.de/windows-phone/> (02-2017).



ANEXO: Solución a los problemas con acentos en PHP, MySQL y SWIFT

Uno de los problemas que se encontró cuando se estaba desarrollando la aplicación fue que, a la hora de introducir acentos, “ñ” o cualquier carácter extraño, PHP no los identificaba y daba error y, como consecuencia, el resultado de la consulta realizada por Swift no era el esperado (Barreno D, 2012).

En este caso, se va a utilizar la codificación UTF-8, que es la codificación que tiene por defecto “*Xcode*”. Si se quisiera utilizar otra, se tendría que cambiar la codificación también en este programa. Por lo tanto, la parte de Swift ya está codificada correctamente.

Acto seguido se procede a realizar la codificación del servidor de base de datos “*MySQL*”. Cuando se crean las tablas con *phpMyAdmin* (recordar que se trabaja con la recopilación de servicios “*MAMP*”), hay que tener especial cuidado con el cotejamiento, que indica el tipo de codificación que se le asigna al elemento que se está tratando. Todos deben tener la misma codificación, UTF-8. De esta manera, se debe elegir en cada cotejamiento la opción “*utf8_general_ci*”.

Hay cuatro tipos de cotejamiento:

- Para las conexiones al servidor. En este caso se debe utilizar “*utf8mb4_general_ci*”.



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

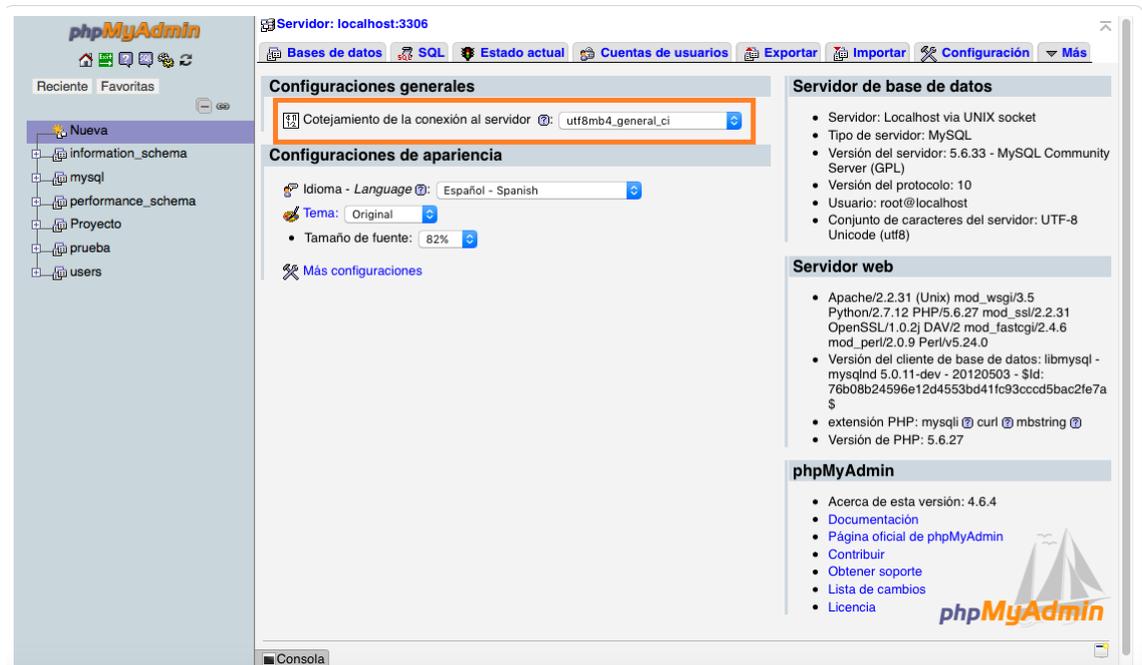


Figura 97. Configuración lenguaje para la conexión al servidor.

- Para la base de datos.

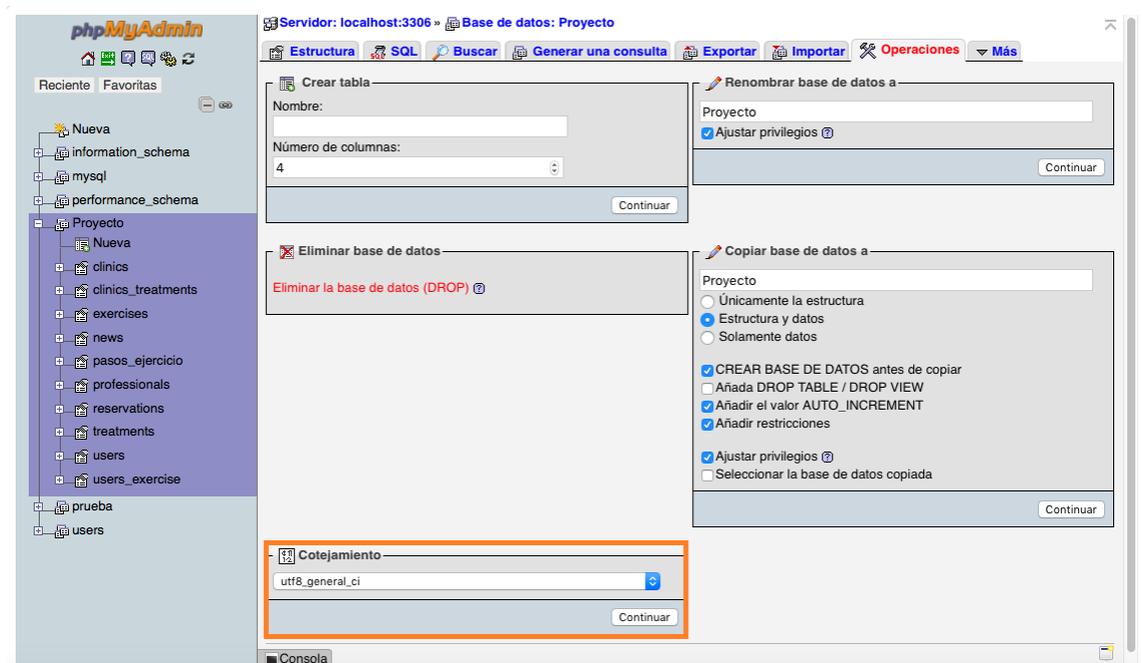


Figura 98. Configuración lenguaje para la base de datos.

ANEXO



- Para las tablas.

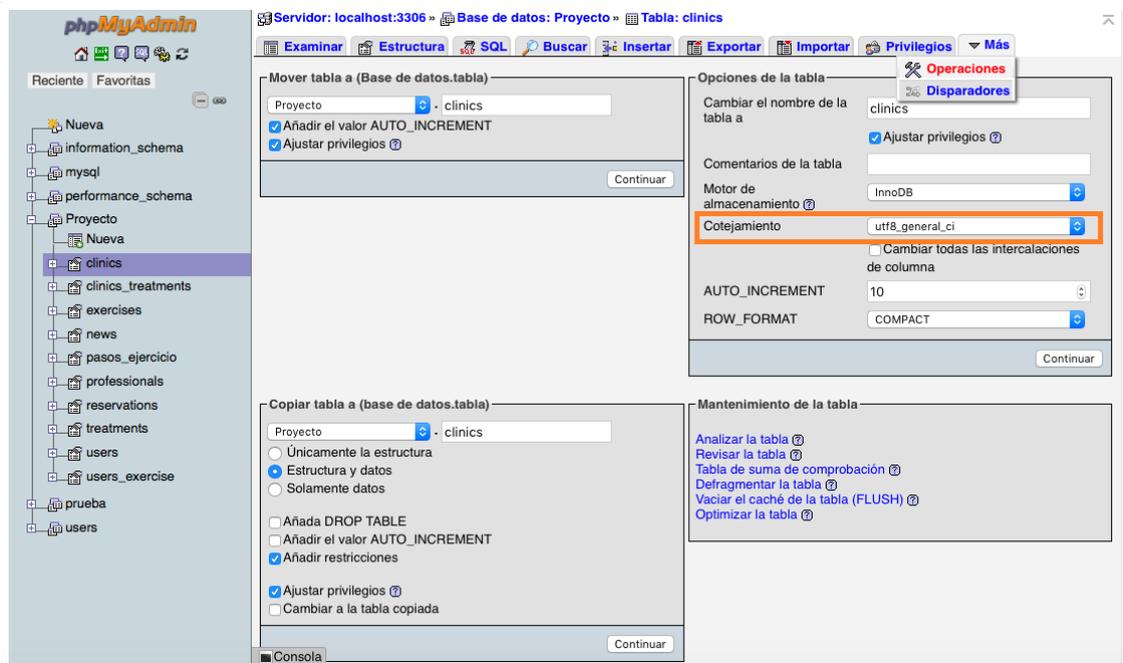


Figura 99. Configuración lenguaje para las tablas (1).

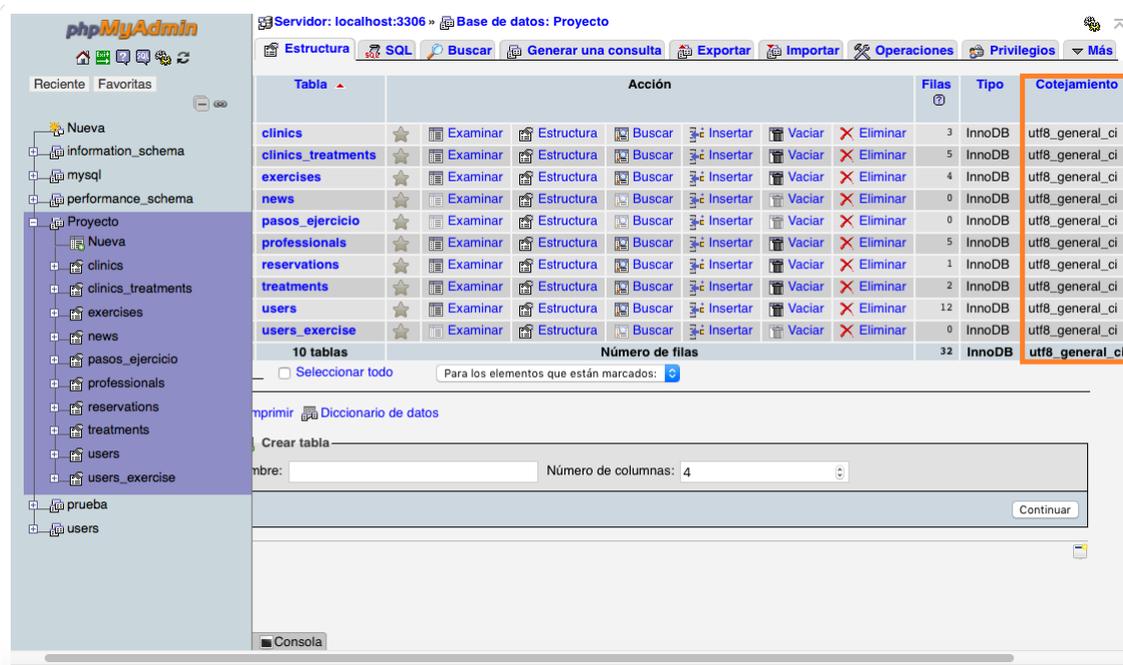


Figura 100. Configuración lenguaje para las tablas (2).

- Para los campos o columnas.



APLICACIÓN iOS DE APOYO A PACIENTES Y PROFESIONALES DE FISIOTERAPIA

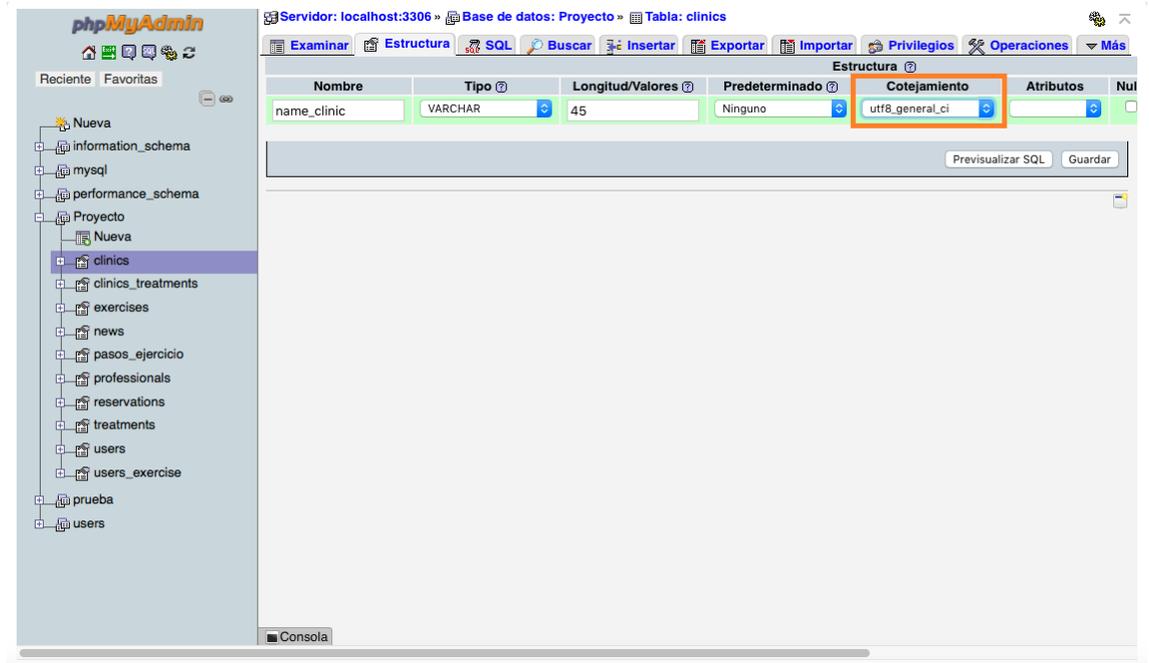


Figura 101. Configuración lenguaje para los campos.

Una vez realizados todos estos pasos, ya estaría correctamente codificada “MySQL”.

Por último, solo quedaría codificar el fichero “PHP” que va a comunicarse con la base de datos. Simplemente introduciendo a continuación de la apertura de la conexión a la base de datos la sentencia “`mysqli_set_charset('utf8');`”, se tendría codificada esta comunicación.