

Proyecto Fin de Grado

Ingeniería de las Tecnologías de Telecomunicación

Herramienta de autocontrol y seguimiento por parte
del paciente de dolor lumbar

Autor: Antonio Ariza Llamas

Tutor: José Manuel Fornés Rumbao

Dpto. Ingeniería Telemática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2019



Proyecto Fin de Carrera
Ingeniería de Telecomunicación

Herramienta de autocontrol y seguimiento por parte del paciente de dolor lumbar

Autor:
Antonio Ariza Llamas

Tutor:
José Manuel Fornés Rumbao
Profesor titular

Dpto. de Ingeniería Telemática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla
Sevilla, 2019

Proyecto Fin de Carrera: Herramienta de autocontrol y seguimiento por parte del paciente de dolor lumbar

Autor: Antonio Ariza Llamas

Tutor: José Manuel Fornés Rumbao

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2019

El Secretario del Tribunal

A mis padres

A mis abuelos

A mis hermanos

Agradecimientos

Con este trabajo pongo la guinda final a mi época universitaria. Ha sido un duro camino y como siempre digo, la persona que entró aquí y la que sale, son muy distintas. Esta carrera me ha enseñado a valorar, aún más, el trabajo duro, la perseverancia y la dedicación. Entré aquí sin saber qué eran las telecomunicaciones, y me voy sin seguir sabiéndolo, porque las telecomunicaciones, sobre todo en esta era, lo son todo. Esta etapa, a base de palos, me ha demostrado lo dura que es la vida. Y es que, para llegar a la meta, te tienes que tropezar muchas veces antes.

Por eso a los primeros que tengo que dar las gracias son a mis principales baluartes, mis padres, que han hecho un esfuerzo enorme porque yo esté aquí y me han demostrado el potencial que dentro de mí encierro. Muchos han sido los consejos que me han dado, algunos los he escuchado, otros menos, pero siempre entendiendo que lo han hecho por mi bien y por el futuro que me espera. Perdonadme por haber sido un poco cabezota y travieso, pero hay cosas que me va a costar trabajo corregir. A mis hermanos, a mis abuelos y a mi familia, también quiero dar las gracias, por estar ahí cuando los he necesitado y por ayudarme mientras estudiaba.

No todo ha sido sufrimiento en esta carrera, es obvio, y quién diga lo contrario, miente. Admito que tal vez haya faltado a alguna que otra clasecilla para irme con mis amigos, que he tenido alguna que otra escapada nocturna y que he visto amanecer en el puente de Triana más de una vez. ¿Pero quién no lo ha hecho? Hasta usted, que estás leyendo esto alguna vez has sido joven. Una de las cosas que he aprendido en este tiempo, es que la vida se mide en vivencias y no en años, y de Sevilla me llevo como para escribir un libro.

He tenido la suerte de coincidir con un grupo de personas excelentes, compañeros que se han convertido en hermanos, que me han servido de apoyo cuando el tiempo traía tormenta y me han enseñado que lo importante no es el lugar, sino la compañía. Por ello tengo que dar las gracias a mis machos, por sacar mi mejor versión, a mis tabernícolas, porque juntos hemos descubierto que mucha solera y buena ensaladilla es signo de confianza, a mis amigos de toda la vida, porque no somos muy numerosos, pero lo nuestro es amistad de verdad, a pesar de nuestros pequeños conflictos, al Perdón, por recordarme que las tradiciones no deben de morir nunca y al Betis, porque pasará el tiempo y me seguirá demostrando que el “manquepierda” siempre será un lema de vida, una forma de sentir y de vivir.

Por último quisiera dar las gracias a mi tutor y a los médicos que me han ayudado a que esto sea posible, porque la verdad, la medicina nunca fue lo mío.

Y para acabar, lo haré con una cita de Alberto García Reyes (pregonero de la Semana Santa 2017), que dice así:

“Y ante Dios, cuando Dios quiera, mis dos palabras postreras serán Sevilla y amén”

*Antonio Ariza Llamas
Sevilla, 2019*

Resumen

En la actualidad son muchas las personas entre 30 y 60 años que sufren de un dolor crónico lumbar, y cada vez con más frecuencia, personas relativamente jóvenes. Esto en muchas ocasiones provoca la aglomeración de personas que sufren de dicho dolor en las consultas de centros médicos y hospitales de nuestro país, reduciendo la eficiencia de los profesionales que trabajan en los mismos y por lo tanto, ofreciendo un servicio de peor calidad.

¿Qué solución puede aportar la ingeniería para darle remedio a esta situación? Es bien sabido que la irrupción de las nuevas tecnologías y, sobre todo, de los teléfonos móviles está cambiando por completo la forma que tienen los habitantes de afrontar el día a día tanto en sus tareas diarias, como en su ocio, como en la forma de relacionarse.

La idea general es, usar los denominados “smartphones” para que, a través de una aplicación móvil, el usuario (paciente) sea capaz de controlar su dolor lumbar y por otra, el médico, a través de un servicio web, sea capaz de llevar un seguimiento de la evolución del dolor de su paciente durante un intervalo de tiempo determinado.

Para favorecer a dicha labor, se hace uso del programa Android Studio (desarrollo de la app móvil), el framework Yii2 (parte web), de MySQL y phpMyAdmin para la creación de la base de datos y del servicio Firebase para el envío de notificaciones.

| | |
|--|-------------|
| Agradecimientos | ix |
| Resumen | xi |
| Índice | xiii |
| Índice de Tablas | xv |
| Índice de Figuras | xvii |
| 1 Introducción | 1 |
| 1.1 <i>Objetivos</i> | 1 |
| 1.2 <i>Estructura del trabajo fin de grado</i> | 2 |
| 2 Estado del arte | 3 |
| 2.1 <i>Dolor lumbar</i> | 3 |
| 2.2 <i>Medida del dolor lumbar (EVA)</i> | 5 |
| 2.3 <i>Salud móvil</i> | 6 |
| 2.4 <i>Android Studio</i> | 7 |
| 2.5 <i>MySQL + phpMyAdmin</i> | 8 |
| 2.6 <i>Yii2</i> | 10 |
| 2.7 <i>REST</i> | 11 |
| 2.8 <i>Firebase</i> | 12 |
| 3 Escenario de funcionamiento | 15 |
| 4 Aplicación Móvil | 17 |
| 4.1 <i>Algoritmo</i> | 17 |
| 4.1.1 <i>Grado de dolor bajo</i> | 19 |
| 4.1.2 <i>Grado de dolor medio</i> | 20 |
| 4.1.3 <i>Grado de dolor alto</i> | 21 |
| 4.1.4 <i>Grado de dolor muy alto</i> | 22 |
| 4.2 <i>Estructura de ficheros</i> | 22 |
| 5 Servidor Web | 34 |
| 5.1 <i>Base de datos</i> | 34 |
| 5.2 <i>Configuración de Yii2</i> | 41 |
| 5.3 <i>Servicios REST</i> | 49 |
| 5.3.1 <i>Registro EVA</i> | 51 |
| 5.3.2 <i>Registra Ejercicio</i> | 51 |
| 5.3.3 <i>Consulta nombre</i> | 52 |
| 5.3.4 <i>Consulta histórico</i> | 52 |
| 5.3.5 <i>Consulta ejercicio</i> | 53 |
| 5.3.6 <i>Consulta consejos</i> | 53 |
| 5.3.7 <i>Consulta medicación</i> | 54 |
| 5.3.8 <i>Comprueba variables</i> | 54 |
| 5.3.9 <i>Actualiza variables</i> | 55 |
| 5.3.10 <i>Mensaje formativo</i> | 55 |
| 5.3.11 <i>Mensaje informativo</i> | 56 |

ÍNDICE DE TABLAS

| | |
|---|----|
| Tabla 5-1. Relaciones Modelo-Vista-Controlador (CRUD) | 48 |
| Tabla 5-2. Relaciones Modelo-Vista-Controlador (REST) | 49 |
| Tabla 5-3. Servicio “Registro EVA” | 51 |
| Tabla 5-4. Servicio “Registra Ejercicio” | 51 |
| Tabla 5-5. Servicio “Consulta nombre” | 52 |
| Tabla 5-6. Servicio “Consulta histórico” | 52 |
| Tabla 5-7. Servicio “Consulta ejercicio” | 53 |
| Tabla 5-8. Servicio “Consulta consejos” | 53 |
| Tabla 5-9. Servicio “Consulta medicación” | 54 |
| Tabla 5-10. Servicio “Comprueba variables” | 54 |
| Tabla 5-11. Servicio “Actualiza variables” | 55 |
| Tabla 5-12. Servicio “Mensaje formativo” | 55 |
| Tabla 5-13. Servicio “Mensaje informativo” | 56 |

ÍNDICE DE FIGURAS

| | |
|--|----|
| Ilustración 2-1. Causas primarias y secundarias del dolor lumbar | 4 |
| Ilustración 2-2. Algoritmo de la evolución natural del dolor lumbar | 5 |
| Ilustración 2-3. Escala Visual Analógica (EVA) | 6 |
| Ilustración 2-4. Distribución de aplicaciones móviles de salud. Año 2015 | 7 |
| Ilustración 2-5. Icono Android Studio | 7 |
| Ilustración 2-6. Interfaz gráfica de Android Studio | 8 |
| Ilustración 2-7. Creación de un usuario MySQL | 9 |
| Ilustración 2-8. Pantalla de inicio de sesión en phpMyAdmin | 9 |
| Ilustración 2-9. Pantalla principal de phpMyAdmin | 10 |
| Ilustración 2-10. Esquema tipo del modelo MVC | 10 |
| Ilustración 2-11. Vista general de Gii | 11 |
| Ilustración 2-12. Esquema general de funcionamiento de una API REST | 12 |
| Ilustración 2-13. Controlador tipo REST | 12 |
| Ilustración 2-14. Icono de Google Firebase | 13 |
| Ilustración 3-1. Escenario de funcionamiento | 15 |
| Ilustración 4-1. Funcionamiento general del algoritmo | 18 |
| Ilustración 4-2. Algoritmo para el grado de dolor bajo | 19 |
| Ilustración 4-3. Algoritmo para el grado de dolor medio | 20 |
| Ilustración 4-4. Algoritmo para el grado de dolor alto | 21 |
| Ilustración 4-5. Algoritmo para el grado de dolor muy alto | 22 |
| Ilustración 4-6. Lista de ficheros de la aplicación móvil | 23 |
| Ilustración 4-7: Pantalla Autenticacion.xml | 24 |
| Ilustración 4-8: Pantalla SplashScreen.xml | 24 |
| Ilustración 4-9: Pantalla MainActivity.xml | 25 |
| Ilustración 4-10: Pantalla PreguntaEjercicio.xml | 26 |
| Ilustración 4-11: Pantalla MensajeFormativo.xml | 26 |
| Ilustración 4-12: Pantalla EscuelaEspalda.xml | 27 |
| Ilustración 4-13: Pantalla PautaAnalgesica.xml | 28 |
| Ilustración 4-14: Pantalla MoriskyGreen.xml | 29 |
| Ilustración 4-15: Pantalla PautaRescate.xml | 29 |
| Ilustración 4-16: Ejemplo de notificación creada | 30 |
| Ilustración 4-17: Pantalla MenuInicial.xml | 30 |
| Ilustración 4-18: Pantalla MuestraHistorico.xml | 31 |

| | |
|--|----|
| Ilustración 4-19: Pantalla MuestraConsejos.xml | 31 |
| Ilustración 4-20: Pantalla MuestraMedicacion.xml | 32 |
| Ilustración 4-21: Pantalla MuestraEjercicio.xml | 32 |
| Ilustración 5-1. Estructura de la base de datos | 35 |
| Ilustración 5-2. Valores de la tabla “n_actividad_física” | 35 |
| Ilustración 5-3. Valores de la tabla “n_estudios” | 36 |
| Ilustración 5-4. Valores de la tabla “n_renta” | 36 |
| Ilustración 5-5. Valores de la tabla “sexo” | 37 |
| Ilustración 5-6. Valores de la tabla “valores_var” | 38 |
| Ilustración 5-7. Valores de la tabla “valores_var” | 39 |
| Ilustración 5-8. Valores de la tabla “nivel_ejercicio” | 39 |
| Ilustración 5-9. Valores de la tabla “valores_medic” | 40 |
| Ilustración 5-10. Directorios dentro de la carpeta basic | 41 |
| Ilustración 5-11. Ficheros que componen el directorio config | 42 |
| Ilustración 5-12. Configuración del fichero db.php | 43 |
| Ilustración 5-13. Activación del módulo gii | 43 |
| Ilustración 5-14. Creación de un modelo usando gii | 44 |
| Ilustración 5-15. Generación de código CRUD usando gii | 44 |
| Ilustración 5-16. Vista del modelo Paciente | 45 |
| Ilustración 5-17. Ejemplo de uso | 45 |
| Ilustración 5-18. Controlador REST | 46 |
| Ilustración 5-19. Activación JSONParser | 46 |
| Ilustración 5-20. Configuración de web.php | 46 |
| Ilustración 5-21. Ejemplo de servicio REST | 47 |
| Ilustración 5-22. Modelos creados. | 47 |
| Ilustración 5-23. Controladores creados | 48 |
| Ilustración 5-24. Vistas creadas | 48 |
| Ilustración 5-25: Apariencia de la pantalla inicial | 57 |
| Ilustración 5-26: Lista de pacientes | 57 |
| Ilustración 5-27: Histórico de dolor | 58 |
| Ilustración 5-28: Medicación de los pacientes | 58 |
| Ilustración 5-29: Credenciales de autenticación | 59 |
| Ilustración 5-30: Listado de tokens | 59 |
| Ilustración 5-31: Aspecto general de Firebase | 60 |
| Ilustración 5-32: Herramienta Cloud Messaging | 60 |
| Ilustración 5-33: Ejemplo de notificación | 61 |

1 INTRODUCCIÓN

1.1 Objetivos

La lumbalgia es el motivo de consulta más frecuente para un médico de Atención Primaria en relación con la patología del aparato locomotor y está considerada la principal causa de limitación de la actividad en personas menores de 45 años. Es la segunda causa de incapacidad laboral transitoria y una de las causas más frecuentes de invalidez permanente. Por todo ello, la lumbalgia constituye una de las principales causa de consumo de recursos sanitarios y sociales.

Estudios demuestran que un 80% de la población tendrá dolor lumbar en algún momento de su vida, lo que, trasladándolo a cifras, en el caso de un país de España con más de 46 millones de habitantes supone que más de 36 millones de personas lo sufrirán al menos una vez. Pero esto no acaba aquí. Hablando del impacto económico que tiene la mencionada patología, esta cuesta a la sanidad española unos 16000 millones de euros, lo que supone un 1,7% del PIB. En resumen, hay pocas patologías con la trascendencia individual, social y económica del dolor lumbar.

A esto hay que añadir ya no solo el crecimiento exponencial en el número de usuarios que hacen uso de terminales móviles para realizar sus labores más cotidianas, sino también de aplicaciones móviles del entorno sanitario, conocido como salud móvil. Dentro de la medicina, una de las cosas más importantes es la relación que existe entre médico y paciente, pero también es cierto que, alrededor del paciente, se ponen miles de opciones que refuerzan la labor de los especialistas. Y es aquí donde entran las aplicaciones para el móvil. Las nuevas tecnologías ayudan al paciente, pero también ayuda al médico a diagnosticar y a informarse

Teniendo en cuenta todo lo referido, el objetivo principal es crear una aplicación móvil que de manera interactiva y dinámica permita al usuario tener un mayor grado de autonomía en cuanto a la gestión de esta enfermedad se refiere, pero con la seguridad de que la misma le va a aportar formación para el automanejo, conocimiento sobre la enfermedad, recordatorios para disminuir el incumplimiento del tratamiento recomendado y contacto con el profesional sanitario de referencia para abordar adecuadamente las exacerbaciones o crisis. La propia aplicación, le permite al usuario identificar de manera objetiva el grado de dolor mediante una escala analógica visual (EVA) y compararla con los días previos, permitiendo un juicio evolutivo del síntoma principal de este proceso, el dolor. Según el grado de dolor se aconsejarán medidas higiénico-dietéticas, tratamiento farmacológico o incluso gestionar una cita con el profesional responsable si el dolor alcanza un cierto nivel de severidad o se incrementa de forma importante sin causa aparente.

En resumen, esta aplicación móvil permite que el paciente aprenda a convivir con el dolor lumbar, mejorando su percepción de su calidad de vida y adecuando el uso de recursos farmacológicos, sanitarios y sociales; a la par que disminuir los eventos adversos relacionados con el consumo de fármacos para controlar el dolor.

1.2 Estructura del trabajo fin de grado

En los siguientes puntos se pone de manifiesto de manera resumida el contenido de los capítulos y anexos que componen este trabajo:

- Capítulo 1 – Introducción

Capítulo actual, en el mismo se ponen de manifiesto los motivos y causas que han llevado a la realización del mismo así como los objetivos que se quieren conseguir.

- Capítulo 2 – Estado del Arte

En este capítulo se describen y desarrollan todos los conceptos y entornos necesarios para llevar a cabo la comprensión del documento y el trabajo que en el mismo se desarrolla.

- Capítulo 3 – Escenario de funcionamiento

Se muestra una visión global y una breve descripción del funcionamiento del conjunto de elementos que componen el proyecto.

- Capítulo 4 - Aplicación móvil

Dentro de este capítulo se explica desde el algoritmo de funcionamiento de la aplicación móvil hasta el aspecto final de la misma, pasando por la estructura de los ficheros que conforman el proyecto.

- Capítulo 5 – Servidor Web

Aquí se habla de la configuración de la base de datos así como su estructura y de la configuración y los ficheros que conforman el entorno de trabajo Yii2 así como la aplicación web creada para facilitar al médico las labores de administración de la información del paciente.

2 ESTADO DEL ARTE

Antes de empezar con el desarrollo de todo lo que incumbe al trabajo fin de grado propiamente, es necesario situarse en un punto de partida para conocer el entorno en el que se va a desarrollar el mismo.

Es por ello que es necesario explicar algunos de los conceptos que serán repetidos con asiduidad a lo largo de este documento, así como las distintas herramientas de trabajo para facilitar al lector la total comprensión de este trabajo fin de grado.

En este capítulo se explica por una parte todos aquellos conceptos del ámbito sanitario que se van a tratar debido a que el objetivo principal del mismo es controlar el dolor lumbar, luego es competencia del proyectante hacer una descripción del dolor lumbar, explicar cómo se mide y como la tecnología puede ayudar a través de la telefonía móvil.

Por otra parte se explican cuáles serán los entornos de desarrollo de la aplicación y que herramientas se usarán para permitir una conectividad entre la aplicación y la base de datos dónde se almacenan todos los datos de pacientes que se usarán para llevar a cabo un estudio que mida la eficacia e impacto de dicha aplicación, aunque esto último está fuera de la competencia de dicho trabajo fin de grado.

Luego los conceptos que se van a desarrollar en este capítulo son los siguientes:

- Dolor lumbar
- Medida del dolor lumbar (EVA)
- Salud móvil
- Android Studio
- MySQL + phpMyAdmin
- Yii2
- REST
- Firebase

2.1 Dolor lumbar

Se conoce lumbalgia como la sensación dolorosa circunscrita a la columna que impide su movilidad con normalidad. En función de la duración puede ser aguda si dura al menos 3 meses y crónica una vez superado ese límite temporal y además se acompaña de intolerancia al esfuerzo. Esta última será objeto de estudio durante este trabajo.

En el caso de pacientes con dolor lumbar, la sensación dolorosa puede bien ceñirse únicamente a la región lumbar, o incluir una o ambas extremidades inferiores. Esto recibe el nombre de síndrome lumbar. Si la lumbalgia se acompaña de dolor genital, perianal, anestesia en silla de montar, afección de la micción y/o defecación, recibe el nombre de cola de caballo.

Generalmente en el dolor lumbar la sintomatología es mecánica, es decir, aparece con la movilización de la columna y desaparece con el reposo. La segunda sintomatología más frecuente es la inflamatoria, es decir, un dolor continuo, persistente e intenso que a pesar de la inmovilidad no mengua. En la siguiente tabla podemos

ver una tabla con las causas primarias y secundarias del dolor lumbar:

| |
|--|
| Mecánicas |
| Inespecíficas |
| Alteraciones degenerativas |
| Prolapso del disco intervertebral |
| Artrosis de las articulaciones interapofisarias |
| Malformaciones adquiridas |
| Estenosis vertebral |
| Espondilolistesis |
| Espondilólisis |
| Malformaciones congénitas |
| Espina bifida |
| Anomalías de transición (lumbarización S1, sacralización L5) |
| Sobrecarga funcional |
| Dismetrías pélvicas |
| Insuficiencia vertebral/trastornos de la estática |
| De origen coxofemoral |
| Inflamatorias |
| Espondiloartritis anquilopoyética |
| Artritis reumatoide |
| Neoplasias |
| Tumores vertebrales primarios o secundarios |
| Metabólicas |
| Osteoporosis (si aplastamiento vertebral) |
| Osteomalacia |
| Infecciones |
| Osteomielitis vertebral |
| Discitis |
| Sacroileitis |
| Enfermedad ósea de Paget |
| Funcionales y psicógenas |

Ilustración 2-1. Causas primarias y secundarias del dolor lumbar

Referido al dolor lumbar, se tienen 3 grandes síndromes:

- **Lumbalgia mecánica aguda:** también conocida como lumbar, es aquella que se da en periodos inferiores a 3 meses con crisis que pueden ser aisladas o repetidas. Aparece en individuos mayores de 18 años y menores de 50 años.
- **Lumbociática aguda:** es la sensación dolorosa radicular que sigue el trayecto del nervio ciático que puede tener afección motora (ciática) y/o sensitiva (ciatalgia).
- **Estenosis lumbar:** presenta síntomas crónicos, como retención urinaria, anestesia perianal, progresiva debilidad en extremidades inferiores, impotencia sexual (varones) o incontinencia fecal.

Como se menciona en el anterior capítulo, el dolor lumbar es una de las principales causa de consulta en atención primaria [1] y una de las causas más frecuentes de absentismo laboral. En España representa la primera causa de baja por incapacidad temporal (IT). Según estudios, se ha relacionado esa incapacidad laboral debido al dolor lumbar con insatisfacción laboral, por tipo de trabajo (rutinario), el lugar de trabajo y el ambiente laboral.

La duración de incapacidad laboral se retroalimenta. Es decir, conforme crece el tiempo que una persona está de baja, menos son las posibilidades de que el mismo vuelva en un futuro al puesto laboral que ocupaba. En el siguiente esquema se aprecia la evolución natural del dolor lumbar:

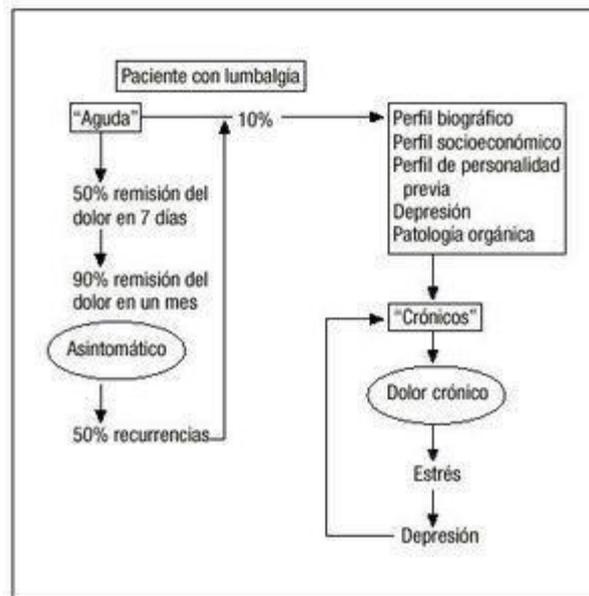


Ilustración 2-2. Algoritmo de la evolución natural del dolor lumbar

Para tratar el dolor lumbar se puede recurrir al tratamiento médico o al tratamiento quirúrgico. Este último no es recomendable pues es extremadamente delicado y sólo está recomendado para procesos que afectan de manera irreversible a la médula espinal o raíces de la misma. En este proyecto el tratamiento que usan los pacientes es médico (excepto aquellos pacientes que puedan tener severas complicaciones).

En cuanto al tratamiento médico recomendado se basa en analgesia mediante fármacos orales, que generalmente son antiinflamatorios no esteroideos (AINE) y métodos físicos, y realizar ejercicio físico mientras sea posible. No es recomendable superar los 2 días de reposo absoluto, pues además de no mejorar los problemas lumbares puede empeorar la situación.

La medicación más prescrita son los AINE (por ejemplo, Ibuprofeno) y pueden combinarse junto con relajantes musculares durante 7-14 días. Los narcóticos u opiáceos pueden usarse durante un escaso intervalo de tiempo debido a los efectos secundarios de los mismos, tales como debilidad, somnolencia o riesgo de adicción.

Se ha demostrado de la eficacia de las escuelas de espalda en la reducción de paciente con lumbalgias temporales y crónicas.

Para finalizar, hay que decir que en el caso de pacientes que sufren alguna de las manifestaciones de dolor lumbar, lo más importante devolverlos a su vida habitual con la mayor rapidez posible para evitar que la incapacidad lumbar se vuelva crónica.

En este trabajo se van a estudiar pacientes que ya tienen un dolor lumbar crónico.

2.2 Medida del dolor lumbar (EVA)

Dado que en general, los pacientes que se tratan no tienen el conocimiento necesario para conocer cuáles son los síntomas reales que sufre en su dolor lumbar, es necesario buscar una manera de medir el grado de dolor lumbar. Para valorar la intensidad del dolor lumbar, se pueden bien recurrir a dibujos visuales que puedan expresar el grado de dolor o bien (como es el caso) a una Escala Visual Analógica (EVA).

La EVA, permite medir la intensidad del dolor que describe el paciente con la máxima reproducibilidad entre observadores. Se trata de una línea en cuyos extremos se encuentran las expresiones extremas de un síntoma. El extremo izquierdo es el grado de menor intensidad y el extremo derecho el de máxima intensidad. En el caso de este proyecto se usa una EVA modificada. Donde se tienen 4 grados de dolor:

- Dolor bajo: 0-3

- Dolor medio: 4-6
- Dolor alto: 6-7
- Dolor muy alto: 8-10

En la siguiente figura podemos apreciar un tipo de EVA:

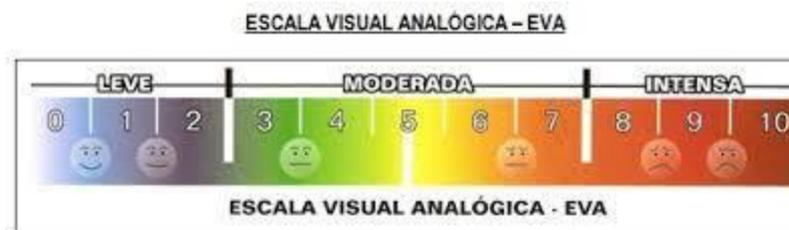


Ilustración 2-3. Escala Visual Analógica (EVA)

De esta forma el paciente podrá identificar de manera más intuitiva que nivel de dolor tiene un día concreto. Será tarea del médico informar de manera adecuada al mismo de que síntomas puede mostrar en cada uno de los niveles que conforman la EVA.

2.3 Salud móvil

La salud móvil o en inglés, mHealth, es un término que se utiliza para la práctica de medicina y salud pública con el apoyo de dispositivos móviles. Actualmente es uno de los sectores en pleno crecimiento, debido en primer lugar crecimiento exponencial en el uso de smartphones en países desarrollados y en segundo lugar por las oportunidades que estas ofrecen. Las aplicaciones móviles se han convertido en un instrumento muy importante a la hora de impulsar hábitos saludables, de detectar de manera rápida algunas enfermedades así como el seguimiento de enfermedades crónicas (cómo es el caso de este trabajo).

El IMS Health Institute [2] explica que, del total de aplicaciones móviles que existen en el mercado, al menos un 30% pertenecen al sector sanitario. Otros datos que avalan la importancia de la salud móvil es que según el Observatorio Nacional de las Telecomunicaciones y de la Sociedad de la Información (ONTSI), el 84% de la población española conoce la existencia de dispositivos y aplicaciones para medir variables relacionadas con su salud y el 27% los utiliza habitualmente en su vida cotidiana. Luego estos indicadores muestran el impacto que está teniendo la tecnología en la salud de las personas.

Los principales usos [3] que se suelen dar a este tipo de aplicaciones son los siguientes:

- Recopilación de datos de Salud: esos datos son usados por organizaciones de salud para planificar y asignar recursos de manera que se apliquen de manera eficiente e inteligente.
- Suministrar información sobre Salud: para permitir al usuario tener a mano una consulta médica.
- Monitoreo en tiempo real de algunas constantes de pacientes: puede ser el caso de personas con diabetes. Ya existen aplicaciones que permiten llevar un control y un seguimiento de los niveles de glucosa que una persona tiene para evitar complicaciones en su enfermedad.
- Telemedicina: permite la gestión de procesos sanitarios a distancia, mejorando la accesibilidad sin aumentar el riesgo de efectos adversos y registrando la protección de datos personales.

En resumen, la salud móvil ha abierto las puertas a nuevas formas de relación entre médico-paciente, aportando numerosos beneficios para ambas partes, tanto en lo que respecta a la optimización del tiempo en consulta como en la mejora de la adherencia terapéutica, elaboración de diagnósticos precoces o la monitorización de pacientes crónicos, dependientes o de riesgo.

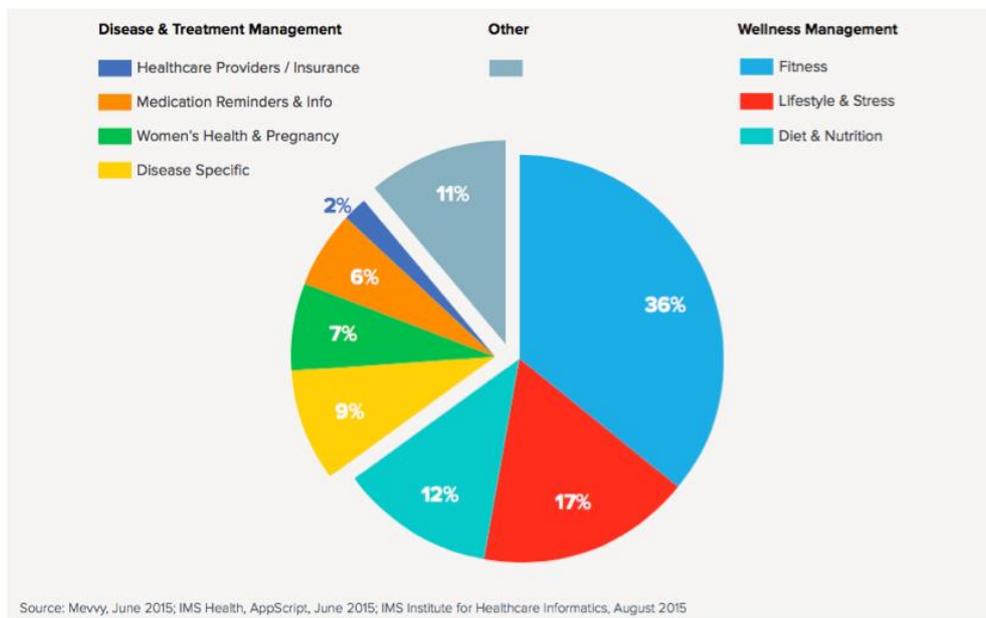


Ilustración 2-4. Distribución de aplicaciones móviles de salud. Año 2015

2.4 Android Studio

Android Studio [4] es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones para Android. Dicho entorno está desarrollado en lenguaje Java. Fue anunciado el 16 de mayo de 2013 en la conferencia Google I/O, y reemplazó a Eclipse como el IDE oficial para el desarrollo de aplicaciones para Android. La primera versión estable fue publicada en diciembre de 2014. Hoy en día es una de las herramientas más utilizadas para programar aplicaciones móviles.



Ilustración 2-5. Icono Android Studio

Dado que es una herramienta con muchas utilidades y sería excesivamente extenso hablar en profundidad sobre la misma, se explicará cual es el aspecto general de la interfaz gráfica que el mismo ofrece:

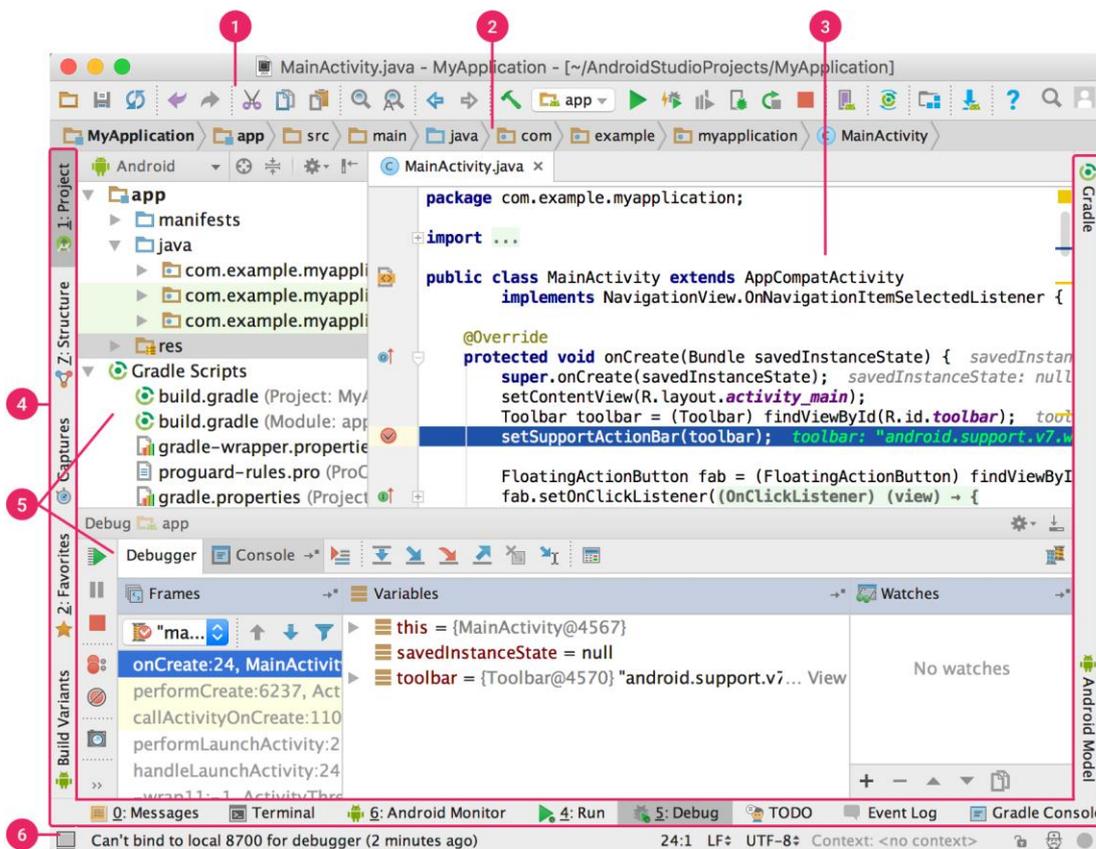


Ilustración 2-6. Interfaz gráfica de Android Studio

- 1) Barra de herramientas: permite entre otras cosas ejecutar la aplicación e inicio de herramientas de Android.
- 2) Barra de navegación: permite abrir archivos para editar así como explorar el proyecto.
- 3) Ventana del editor: área donde crear y modificar el código. Según el tipo de archivo, puede cambiar.
- 4) Barra de la ventana de herramientas: contiene los botones que permiten expandir o contraer ventanas de herramientas individuales.
- 5) Ventanas de herramientas: dan la posibilidad de acceder a tareas específicas, como administración de proyectos, búsquedas o controles de versión.
- 6) Barra de estado: muestra estado del IDE y del proyecto así como posibles avisos.

Aunque no se ha tratado en profundidad, para finalizar hay que decir que Android Studio permite la fácil creación de apps así como probarlas en emuladores, para ver un funcionamiento real de las mismas. Además su compatibilidad con Windows, Mac OS y Linux, la hacen una herramienta perfecta para desempeñar dicha tarea.

2.5 MySQL + phpMyAdmin

Dado que se necesitan guardar datos de los pacientes ya sean personales como generados por la aplicación móvil se hace uso de una base de datos relacional basada en SQL (Structured Query Language). Para facilitar la labor de gestión de dicha base de datos se hace uso de MySQL.

MySQL [5] es un sistema de gestión de bases de datos relacional. Actualmente está considerada como la base de datos de código abierto más extendida y utilizada en el mundo. La mayor parte del mismo está desarrollado en los lenguajes de programación C y C++. Su función general es guardar todos los datos que se desee en bases de datos relacionales. Para acceder al mismo se necesita de un nombre de usuario y contraseña que es

configurada en el servidor dónde se hace uso del mismo. En la siguiente figura se muestra un ejemplo de creación de un usuario MySQL:

```
[root@vps248529 ~]# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 14
Server version: 5.6.29 MySQL Community Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> grant all on *.* to root@'%' identified by 'lo que sea ';
Query OK, 0 rows affected (0,00 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0,00 sec)

mysql> quit
Bye
[root@vps248529 ~]#
```

Ilustración 2-7. Creación de un usuario MySQL

Para ayudara a la tarea de administración de la base de dataos se va a hacer uso del programa phpMyAdmin. phpMyAdmin es una herramienta escrita en PHP cuyo objetivo principal es que a través de su interfaz visual y a todas las opciones y herramientas de las que dispone, se pueda administrar de manera intuitiva la base de datos MySQL creada previamente. El aspecto que muestra el mismo es el siguiente:



phpMyAdmin

Bienvenido a phpMyAdmin

Idioma - Language

Español - Spanish

Iniciar sesión

Usuario: root

Contraseña: *****

Continuar

Ilustración 2-8. Pantalla de inicio de sesión en phpMyAdmin

Tras acceder al mismo, se muestra el aspecto general que tiene el mismo. Como se puede apreciar en la siguiente ilustración, es sencillo pues permite añadir, modificar y borrar tanto tablas como filas dentro de las propias tablas de manera muy intuitiva.

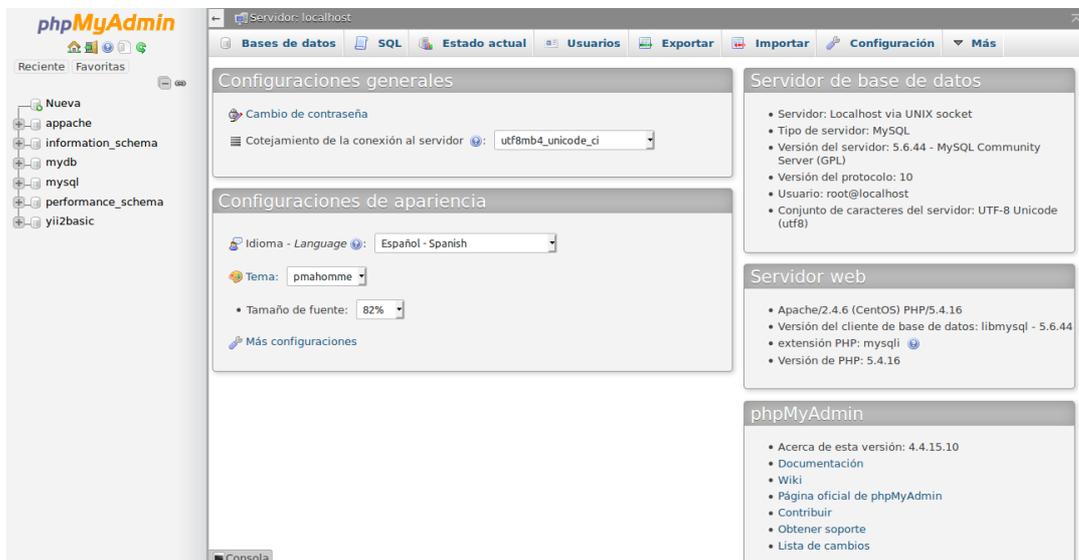


Ilustración 2-9. Pantalla principal de phpMyAdmin

2.6 Yii2

Para evitar poner en peligro la integridad de la base de datos y no tener que realizar tediosas consultas en lenguaje SQL desde la aplicación a la base de datos así como facilitar del trabajo de creación de una aplicación web, se hace uso de un entorno de trabajo llamado Yii.

Yii [6] es un entorno de trabajo (framework, en inglés) de PHP basado en componentes para desarrollar aplicaciones web de manera dinámica. Está basado en el modelo MVC (Modelo-Vista-Controlador), cuyo filosofía es separar por una parte lo de datos de la aplicación (Modelo), la interfaz de usuario (Vista) y la lógica de control (Controlador). El controlador es el encargado poner en contacto la vista con el modelo.

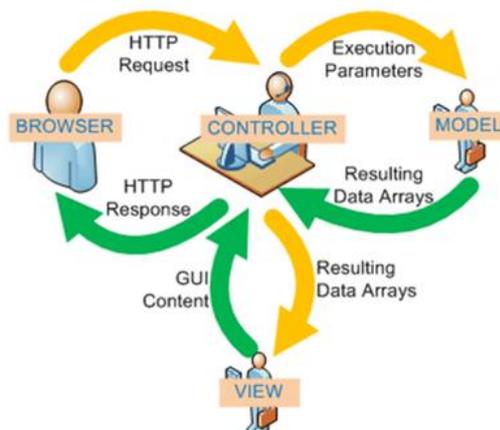


Ilustración 2-10. Esquema tipo del modelo MVC

Yii contiene un módulo llamado Gii, que dispone de una interfaz gráfica que permite crear de manera intuitiva tanto el modelo como el controlador de las distintas tablas que aparecen en la base de datos que se va a usar

para desarrollar el proyecto. Además permite crear CRUD es el acrónimo de Crear, Obtener, Actualizar y Borrar (del inglés: Create, Read, Update y Delete) representando la cuatro funciones con datos más comunes en la mayoría de sitios Web, lo que hace de Yii una herramienta muy completa, potente y flexible.

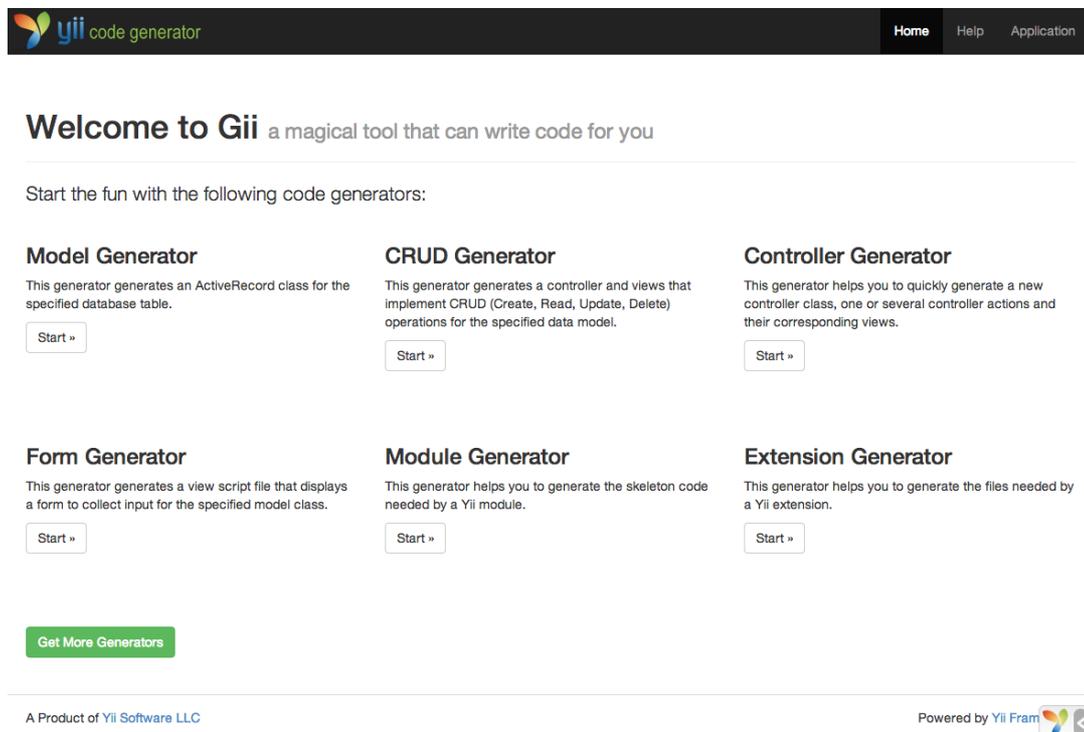


Ilustración 2-11. Vista general de Gii

En el caso de esta aplicación es de gran importancia el uso de la clase ActiveRecord para las consultas de base de datos relacionales y la X para el uso de servicios REST. Yii tiene dos versiones, la 1.0 y la 2.0. En el caso de este trabajo se hará uso de la última al estar más actualizada y añadir más funcionalidades.

Por último resaltar que es este entorno de trabajo es muy utilizado para el desarrollo de aplicaciones de una envergadura considerable como pueden ser páginas webs o servicios web compatibles con arquitectura REST. Esta última característica es indispensable para llevar a cabo la aplicación.

En el siguiente apartado se explica exclusivamente qué es un servicio REST, cómo se usa y cómo puede incluirse un componente en Yii que ayude a realizar dicha labor.

2.7 REST

REST (Representational State Transfer) [7] es cualquier interfaz entre sistemas HTTP para obtener datos o generar operaciones sobre esos datos en formatos como XML o JSON.

El modo de funcionamiento de REST o en su defecto, de una API REST, es el siguiente:

- El cliente hace una petición HTTP que contiene toda la información necesaria para ejecutar la misma. Esto es una ventaja pues no es necesario que se almacenen estados previos.
- En general el conjunto de operaciones más usadas son POST (crear recurso), GET (lectura de un recurso), PUT (editar/actualizar recurso) y DELETE (eliminar un recurso).
- Dicho recurso es direccionable a través de su URI (cadena de caracteres que identifica los recursos de una red de forma unívoca).

En el siguiente esquema se puede apreciar el funcionamiento general de una API REST:

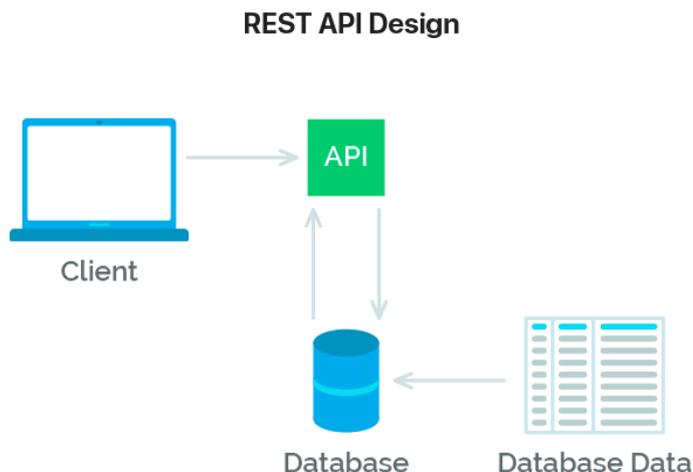


Ilustración 2-12. Esquema general de funcionamiento de una API REST

Yii ofrece todo un conjunto de herramientas para simplificar la tarea de implementar un servicio web APIs RESTful. Esto es de ayuda no sólo para facilitar la labor de obtener, crear, modificar o borrar recursos, sino también porque aumenta la seguridad e integridad de todos los datos contenidos en la base de datos. Dicho controlador tiene la siguiente estructura:

```
namespace app\controllers;  
  
use yii\rest\ActiveController;  
  
class UserController extends ActiveController  
{  
    public $modelClass = 'app\models\User';  
}
```

Ilustración 2-13. Controlador tipo REST

En el capítulo X se explica con más profundidad los tipos de controladores, modelos y vistas que van a ser usados en Yii.

2.8 Firebase

La idea de la aplicación móvil es que el paciente pueda interactuar hasta cierto grado con el médico. Una de las medidas que se implantan son las notificaciones. Para llevar a cabo dicha tarea se hace uso de Firebase.

Firebase [8] es una potente plataforma de Google cuya finalidad es ayudar a desarrollar apps para que cualquier persona, a pesar de no ser un desarrollador, pueda crear sus aplicaciones móviles de manera sencilla. De entre todas las herramientas que posee, en este trabajo únicamente se va a hacer uso de una de ellas, la Cloud Messaging.



Ilustración 2-14. Icono de Google Firebase

Cloud Messaging permite enviar mensajes a cualquier dispositivo móvil. Estos mensajes pueden enviarse bien personalizados únicamente a una persona o bien a un grupo de usuarios que se han suscrito a un determinado tema. También puede programarse de manera que se envíen mensajes a modo recordatorio. En el caso de esta aplicación, por ejemplo, se programan envíos de mensajes para recordar al paciente que tome su medicación.

Lo mejor de todo es que permite todo esto sin necesidad de programar absolutamente nada y de manera bastante interactiva para facilitar la labor al médico.

En el próximo capítulo se detalla cómo se produce este envío desde la plataforma a la aplicación móvil.

3 ESCENARIO DE FUNCIONAMIENTO

Para empezar con el desarrollo del trabajo, en este capítulo se va a mostrar un esquema general de todo lo que el trabajo implica para que el lector pueda tener una visión en conjunto de los distintos componentes que engloban el escenario de trabajo.

En el siguiente esquema se puede tener una visión global del escenario de funcionamiento que en los próximos capítulos se detalla con más profundidad:

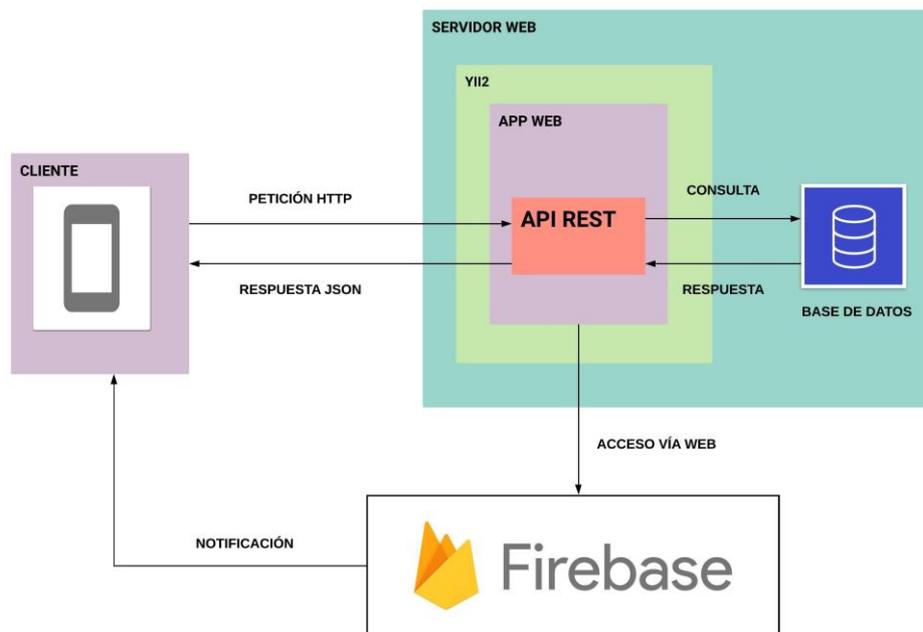


Ilustración 3-1. Escenario de funcionamiento

En el esquema se distinguen dos partes, cliente y servidor web:

- Cliente: está formado por el dispositivo móvil, que contiene la aplicación móvil.
- Servidor web: (en el que corre CentOS 7) contiene por una parte el entorno de trabajo Yii2 que contiene una aplicación web y dentro de esta una API REST y la base de datos dónde se guardan todos los datos de los pacientes.

El funcionamiento es el siguiente:

- 1) La aplicación móvil que ha sido desarrollada en Android Studio y posteriormente instalada en el terminal móvil, lanza una petición HTTP con el método que se desee (GET, POST, PUT, DELETE) junto con los parámetros que sean necesarios para completar dicha petición a una dirección URL que depende del servicio REST del que se vaya a hacer uso.

- 2) Como vimos en el capítulo anterior, Yii2 tenía un componente que permitía con relativa facilidad montar un servicio REST. Cuando se recibe una petición por parte de la aplicación móvil, dicho servicio REST contiene la lógica necesaria para hacer la consulta que usuario ha pedido a la base de datos (MySQL).
- 3) La base de datos recibe una petición por parte del controlador, la procesa y manda la respuesta competente.
- 4) El controlador recibe esa respuesta de la base de datos. Para que el cliente (aplicación) pueda leer e interpretar el contenido, la misma necesitar pasarse a formato JSON.
- 5) El cliente recibe una respuesta con formato JSON, que interpreta y actúa en situación.

Cómo se puede apreciar, a priori, parece muy sencillo el funcionamiento básico. Hay que tener en cuenta que cuando se hace una petición HTTP, la aplicación móvil se olvida por completo de la lógica de consultas directas a la base de datos, eso es tarea del servicio REST, lo que simplifica el trabajo de manera importante.

Es por esto último por lo que actualmente la gran mayoría de empresas de ámbito internacional (Twitter, Amazon, Facebook), implementan servicios REST en sus aplicaciones.

Además de esto, se dispone de una app web (cuya apariencia aparece en el capítulo 5), que es usada por el médico para disponer de las funcionalidades básicas que son necesarias para administrar tanto datos como el funcionamiento de sus pacientes. Esta se comunica con Firebase (a través de un enlace web), cuya funcionalidad “Cloud Messasing” es usada, cómo vimos anteriormente, para enviar notificaciones al paciente.

Para conseguir enviar una notificación a un determinado paciente se hace uso del token. Un token es un identificador único que, en este caso, da Google para identificar a un usuario que usa una determinada aplicación. Este puede ir cambiando bien cuando se instala o desinstala o cuando pasa un período de tiempo determinado. La aplicación móvil registra cuando ese token cambia y, a través de un servicio REST, lo modifica en la base de datos a fin de que el médico disponga de la información actualizada.

En los próximos capítulos se va a mostrar todo lo relacionado con el algoritmo, funcionamiento y visualización de la aplicación móvil así como la configuración del servidor web ya sea tanto de la base de datos como del entorno de trabajo Yii2.

4 APLICACIÓN MÓVIL

En este capítulo se va a desarrollar en profundidad todo lo que envuelve a la aplicación móvil, desde el algoritmo, pasando por la estructura de los ficheros que conforman el proyecto hasta el aspecto final de que tiene la aplicación una vez está en ejecución.

Para organizar este capítulo a fin de clarificar todos los conceptos que se van a explicar a lo largo del mismo y que de esa forma el lector pueda tener una mejor comprensión del mismo, se va a dividir el mismo en 2 apartados:

- Algoritmo: dónde se va a explicar con profundidad cual es el algoritmo general que está detrás de la aplicación.
- Estructura de ficheros: dónde se muestra cuáles son y cuál es la estructura de los ficheros más relevantes que conforman la aplicación móvil. Además se muestran capturas de las distintas pantallas que conforman la aplicación móvil a fin de que se pueda ver visualmente lo que en los dos apartados anteriores se explica.

4.1 Algoritmo

Cómo se ha dicho en capítulos anteriores el objetivo principal del trabajo es crear una aplicación para que los pacientes con dolor lumbar crónico diagnosticado tengan un mejor autoconocimiento de su dolencia y de esa manera puedan llevar un seguimiento diario de la misma.

La base de todo el algoritmo se apoya sobre la Escala Visual Analógica (EVA) modificada, es decir, a diario se pregunta al paciente cuál es su grado de dolencia en ese mismo día. En función de su dolor, la aplicación va a actuar en consecuencia a fin de conseguir que la dolencia sea la mínima posible. Dado que el grado de dolor del paciente (usuario), puede cambiar de un día a otro, es necesario establecer una lógica de actuación.

En primer lugar hay que resaltar que existen 4 grados (grupos) de dolor:

- Dolor bajo: Desde el valor 0 hasta el valor 3
- Dolor medio: Desde el valor 4 hasta el valor 5
- Dolor alto: Desde el valor 6 hasta el valor 7
- Dolor muy alto: Desde el valor 8 hasta el valor 10

Los valores son números enteros e indican el nivel de dolor que el paciente tiene ese día.

Establecido los grados de dolor, se necesita también una serie de variables generales que nos permitan definir un comportamiento, que se encuentran guardadas en la base de datos. Ya que todavía no se ha explicado la estructura que tiene la base de datos debido a que forma parte del servidor, es posible que al principio exista algo de confusión, es por ello que se van a usar nombres para las variables muy indicativos, evitando que exista algún tipo de malentendido. Las variables que se van a usar dentro del programa son las siguientes:

- Día: es una variable que indica los días que una persona se ha encontrado en el nivel de dolor más bajo. Está estrechamente relacionada con la siguiente variable.
- Ejercicio: indica los días que el paciente (durante una semana) ha realizado los ejercicios físicos. Cómo se explicó en el apartado 3.1, el ejercicio físico es clave en la recuperación del paciente.

- Nivel ejercicio: dado que existen distintos programas de entrenamiento, el paciente puede tener una serie de ejercicios con mayor o menor intensidad. Esta variable indica dicho nivel de entrenamiento.
- Pauta analgésica: el médico generalmente va a tener 3 tipos de medicaciones en función del grado de dolor que la persona sienta a fin de calmar su dolor. Esos 3 tipos de medicaciones son una basal (para tomar día a día), otra analgésica cuando el dolor crece y por última una pauta de rescate, cuando el dolor es alto. Con esta variable se indica los días que un paciente lleva tomando dicha pauta analgésica.
- Pauta rescate: similar a la variable anterior, pero en este caso se indican los días que un paciente lleva tomando una pauta de rescate.
- Nivel alto: aunque ahora no tiene sentido, cuando se explique el siguiente apartado sí que lo tendrá. Es una bandera que indica si el paciente está en el nivel de dolor alto.

Sabiendo esto, la estructura general del algoritmo es la siguiente:

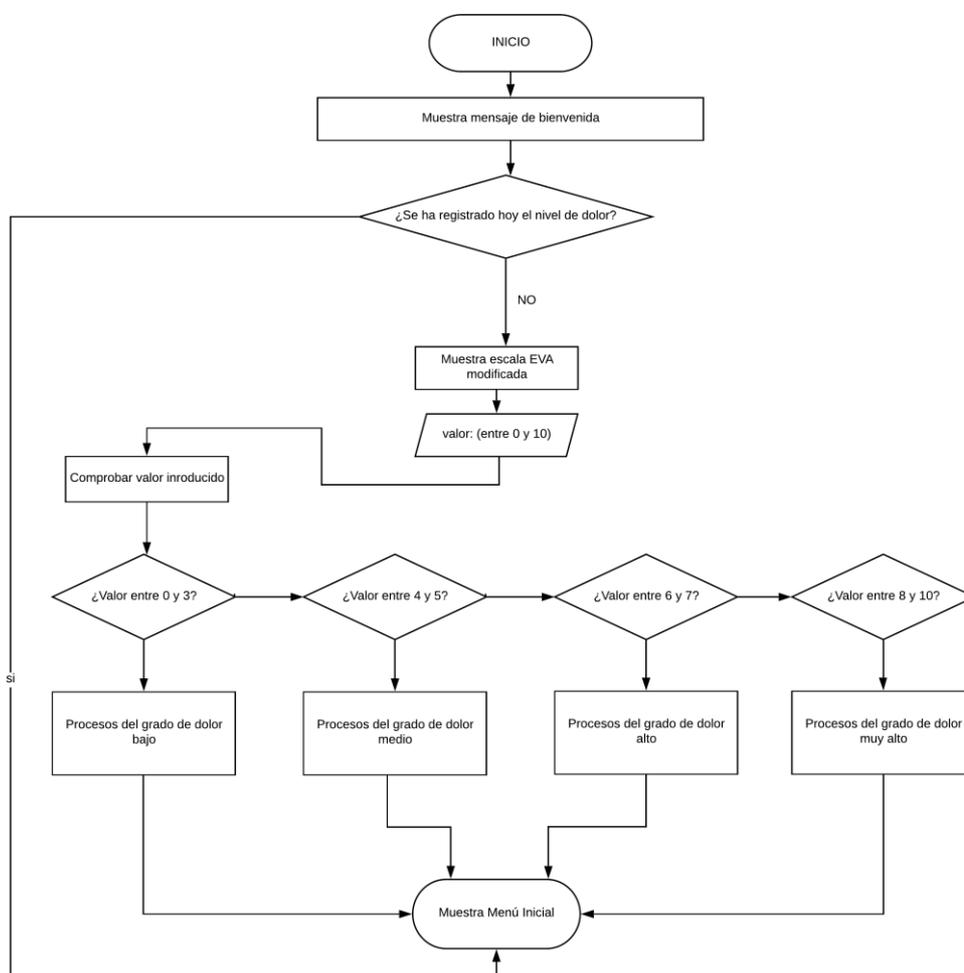


Ilustración 4-1. Funcionamiento general del algoritmo

El esquema anterior muestra a grande rasgos cual es el funcionamiento que va a tener el algoritmo en general. En los siguientes sub-apartados se explica detallado todos los procesos que involucran cada uno de los 4 grados de dolor.

Hay que tener una cosa en cuenta y es que cada vez que se entra en un grado de dolor, las variables de los otros niveles se ponen a cero. Estas son las siguientes:

- Grado de dolor bajo: día, ejercicio y nivel de ejercicio.
- Grado de dolor medio: pauta analgésica y pauta rescate.
- Grado de dolor alto: nivel alto.

4.1.1 Grado de dolor bajo

En el grado de dolor bajo, el paciente está en condiciones de realizar ejercicio físico, esto es un factor fundamental pues sobre el mismo se va a definir el comportamiento del algoritmo en este nivel. El comportamiento para dicho grado de dolor es el siguiente:

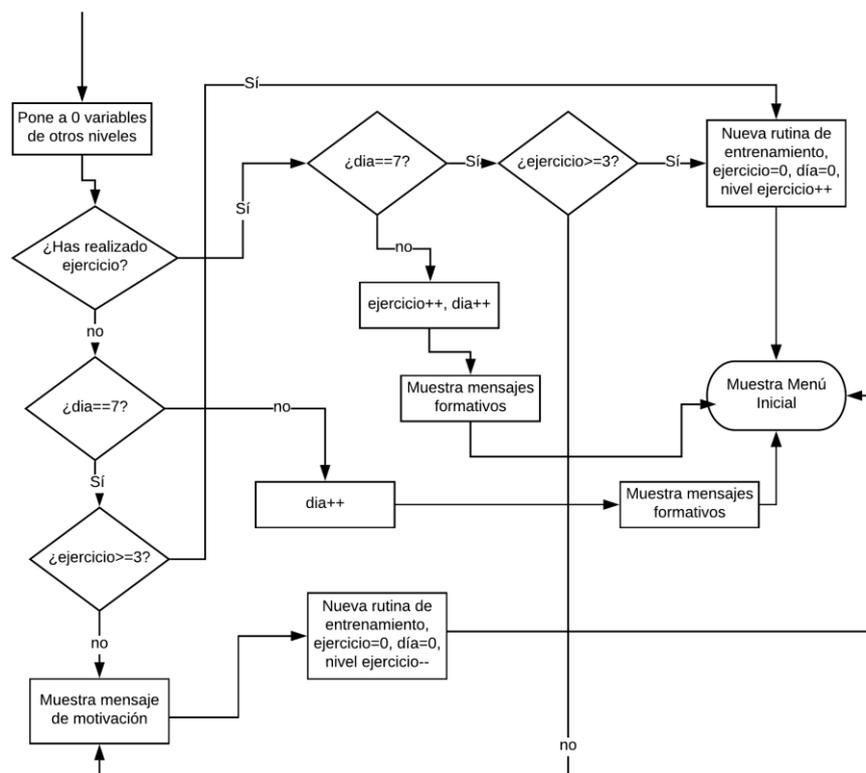


Ilustración 4-2. Algoritmo para el grado de dolor bajo

Como se ha mencionado anteriormente, el ejercicio físico es determinante para mejorar la dolencia lumbar. En este grado de dolor, se da por supuesto que el paciente está en condiciones de realizar ejercicio físico. Luego siempre que se entra en este nivel lo primero que se hace es preguntar por el ejercicio físico. En función de si lo ha realizado o no, la variable ejercicio aumentará o no. La variable día aumenta cada vez que se pregunta sobre la realización del ejercicio físico hasta llegar a 7, es decir, una semana. Pasado una semana pueden pasar dos cosas:

- El paciente ha realizado ejercicio físico al menos 3 días durante esa semana, lo que implica que se puede aumentar la intensidad de los ejercicios para la próxima semana.
- El paciente no ha realizado ejercicio físico al menos 3 días. En este caso se muestra un mensaje motivador para incitar al paciente a llevarlos a cabo. Para favorecer dicha labor, se baja la intensidad de los ejercicios para la próxima semana.

Además de esto, todas las veces que se entra en este nivel se muestra un mensaje formativo para, valga la redundancia, formar al paciente en todo lo relacionado con su dolencia.

Para finalizar, y es una constante que se va a repetir en todos los niveles, es que tras realizar todo lo anterior

mencionado, se muestra un menú inicial dónde el paciente puede:

- Consultar un histórico de su dolor.
- Consultar su mediación actual.
- Consultar consejos sobre medicación, alimentación y ejercicio físico.
- Consultar su entrenamiento actual.

4.1.2 Grado de dolor medio

En este grado, la persona ha visto aumentado el dolor. Para paliar dicho dolor, se recurre a subir la analgesia, ya sea a través de una pauta analgésica o bien, si pasados unos días la sensación dolorosa no aminora, con una pauta de rescate. En el siguiente diagrama de flujo se muestra dicho comportamiento:

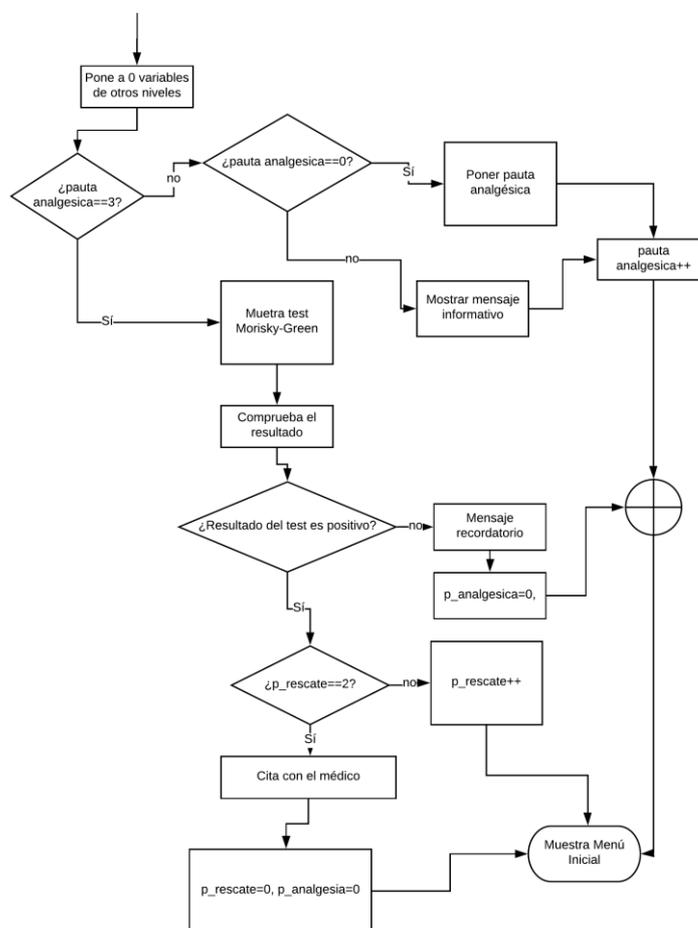


Ilustración 4-3. Algoritmo para el grado de dolor medio

El funcionamiento en este grado de dolor es el siguiente.

Lo primero que se hace es comprobar si el paciente tiene ya puesta una pauta analgésica, es decir, si no es la primera vez que el paciente ha entrado a este grado de dolor. Pueden pasar tres cosas:

- El paciente no tiene una pauta analgésica, luego hay que poner una nueva pauta. Cabe recordar que en este nivel aparece una variable llamada “Pauta analgésica” cuyo valor es el número de días que un paciente lleva con la misma. Cada día que pasa con la misma, se incrementa el valor de la misma.
- El paciente tiene ya puesta una pauta pero lleva menos de 3 días con ella. En este caso se muestra por

pantalla un mensaje informativo para un mejor conocimiento de su dolencia.

- El paciente tiene ya puesta una pauta y lleva 3 días con la misma. En este caso aparece el conocido test de Morisky-Green.

El test de Morisky-Green es una serie de 4 preguntas para comprobar que el paciente cumple con la toma de medicación indicada. Las preguntas que se hacen son las siguientes:

- ¿Olvida alguna vez tomar los medicamentos para tratar su enfermedad?
- ¿Toma los medicamentos a las horas indicadas?
- Cuando se encuentra bien, ¿deja de tomar la medicación?
- Si alguna vez le sienta mal, ¿deja usted de tomarla?

Las respuestas posibles son “SI” o “NO”. Se consideran pacientes que cumplen el tratamiento a quienes contestan NO a las cuatro preguntas y no cumplidores a quienes contestan SI a una o más.

Tras hacer dicho test, el resultado puede arrojar dos resultados:

- Positivo. Si esto es así se considera que el paciente es cumplidor y que por lo tanto la medicación no está funcionando adecuadamente. En este caso se procede a poner una pauta de rescate (si no la tenía ya puesta). Al igual que con el caso de la variable “Pauta analgésica”, la variable “Pauta rescate” indica el número de días que el paciente lleva con su pauta de rescate.
- Negativo. En este caso el paciente se considera no cumplidor y por lo tanto se muestra un mensaje recordatorio para que tome la medicación y se pone a cero la variable “Pauta analgésica”, pues esto quiere indicar que no está haciendo caso de las recomendaciones médicas.

Si el paciente lleva 2 días con la pauta de rescate sin encontrar mejora, se procede a dar cita con el médico que lleve el historial de dicho paciente. Tras esto se ponen las dos variables que se han mencionado a cero.

4.1.3 Grado de dolor alto

En este nivel, el dolor ya es bastante severo. El funcionamiento es el siguiente:

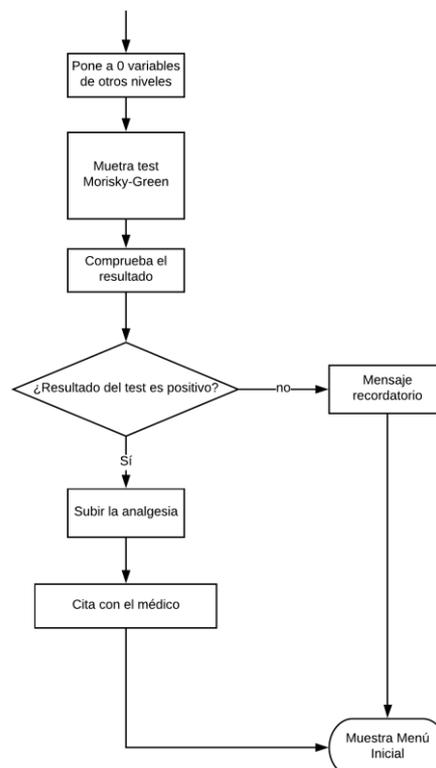


Ilustración 4-4. Algoritmo para el grado de dolor alto

Aquí directamente se realiza el test Morisky-Green. Puede arrojar dos resultados:

- Positivo. En este caso se procede a subir la analgesia para paliar el dolor y se da cita con el médico para que el mismo pueda hablar en persona con el paciente.
- Negativo. Se muestra un mensaje recordatorio para que el paciente tome la medicación. Esto al igual que en el caso del grado de dolor anterior quiere indicar que el paciente no está siguiendo con las indicaciones médicas.

4.1.4 Grado de dolor muy alto

A este nivel sólo se accede cuando el dolor es extremo y el paciente no encuentra manera de soportarlo. El funcionamiento en este caso es el más simple de todos:

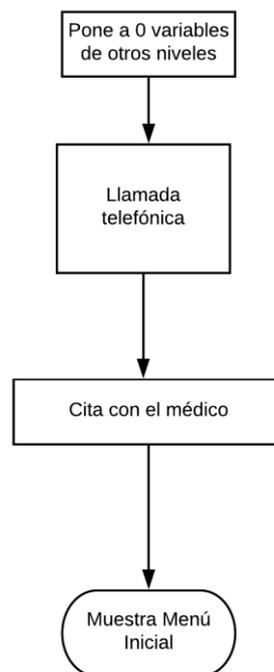


Ilustración 4-5. Algoritmo para el grado de dolor muy alto

Se realiza una llamada directamente con el médico para que el mismo pueda comprobar los síntomas y si verdaderamente el paciente muestra ese dolor que ha indicado. Tras esto se le da cita con el médico para que pueda comprobar en persona si esa dolencia era real o no, y qué solución puede buscar.

Es importante volver a recordar que es competencia del médico explicar correctamente al paciente cuales son los síntomas en cada uno de los grados de dolor para evitar la subjetividad de la sensación dolorosa.

4.2 Estructura de ficheros

En el apartado anterior se ha explicado con detenimiento el algoritmo que gobierna la aplicación móvil. En este apartado se va a explicar todo lo referente a estructura, organización y contenido de los ficheros que contiene la aplicación móvil. Antes de empezar es importante saber que en Android Studio tenemos dos tipos de ficheros de código:

- Ficheros Java: contienen las distintas clases, funciones y objetos que van a definir la aplicación móvil. Contienen el código fuente.
- Layouts: escritos en XML, son los archivos que contienen el diseño de cada una de las distintas pantallas que se van a mostrar en el terminal móvil.

En general cada fichero Java tiene asociado un layout. Ambos ficheros se pueden relacionar a través de identificadores únicos en los ficheros XML (1 identificador por cada objeto que aparezca en el fichero layout).

Además de estos ficheros hay otros muy importantes:

- AndroidManifest.xml: proporciona información esencial sobre tu aplicación al sistema Android, información que el sistema debe tener para poder ejecutar el código de la app.
- Directorio res: que contiene recursos que pueden ser de ayuda para presentar la aplicación móvil tales como iconos, imágenes, cadenas de texto o colores. Contiene también los ficheros layout.

Una vez sabido esto, se va a proceder a explicar cada uno de los ficheros que componen la aplicación móvil. En la siguiente imagen se puede apreciar una captura de Android Studio con la lista de dichos ficheros:

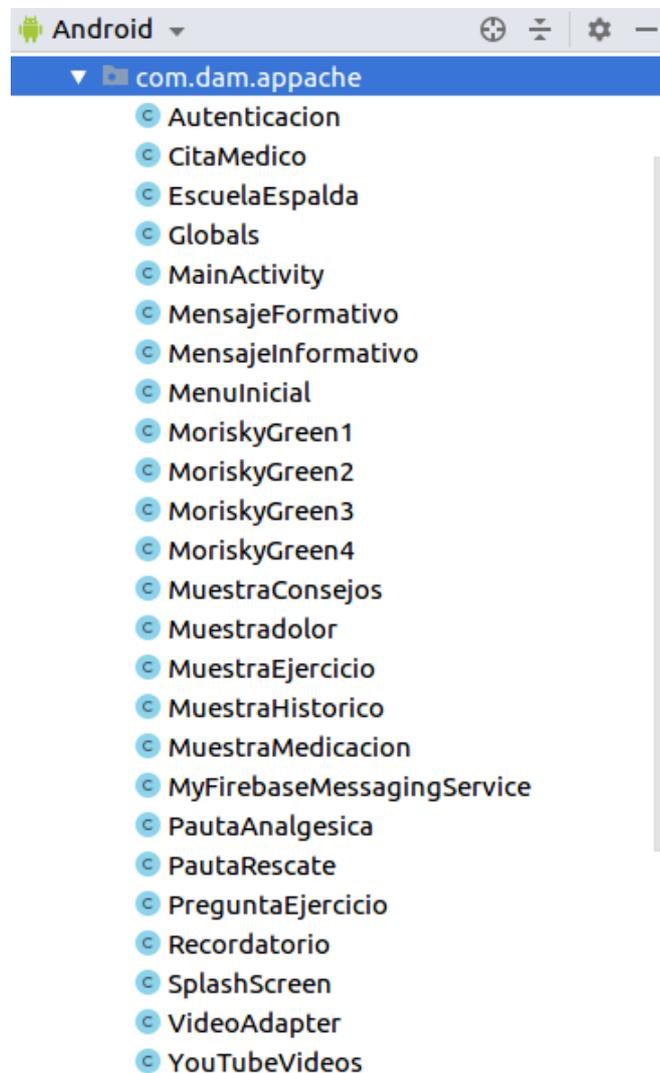


Ilustración 4-6. Lista de ficheros de la aplicación móvil

En este apartado se van a explicar las funciones principales de cada uno de los ficheros java y una imagen con la apariencia del layout asociado a dicha clase:

- **Autenticación:** este sólo se va a ejecutar una única vez. El mismo contiene la lógica necesaria para registrar dicho dispositivo. Para que el mismo se registre se necesita que el paciente introduzca, su NID, número de teléfono y una clave que el médico facilitará al mismo. Es una manera de controlar el acceso a la aplicación de personas externas a la misma.



Ilustración 4-7: Pantalla Autenticacion.xml

- **SplashScreen:** este fichero es el inicial, es decir, es el primero que se abre cuando se pone a funcionar la aplicación. El mismo muestra en primer lugar un mensaje de bienvenida. Tras esto tiene que hacer una comprobación fundamental y es ver si se ha registrado el nivel de dolor en el día actual. Pueden suceder dos cosas:
 - A. El nivel de dolor no ha sido registrado en el día actual, con lo cual se pasa a MainActivity dónde se va a preguntar por el nivel de dolor.
 - B. El nivel de dolor ya ha sido registrado en el día actual. En este caso se pasa a MenuInicial, dónde se muestran una serie de acciones que se pueden realizar.

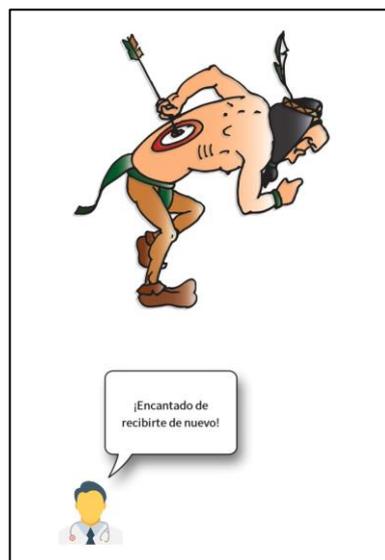


Ilustración 4-8: Pantalla SplashScreen.xml

- **MainActivity**: aquí básicamente se pregunta por el nivel de dolor que el paciente tiene en ese día. Se registra en la base de datos, creando una nueva fila con dicho valor y la fecha del día. Por último se pasa ese valor a Muestradolor.

The screenshot shows a mobile application interface. At the top, it says "¡Hola Antonio!". Below that is a speech bubble asking "¿Qué nivel de dolor tienes hoy?". Underneath is a small icon of a doctor. The main part of the screen is a "Escala de dolor" (Pain Scale) ranging from 0 to 10. The scale is color-coded: 0 is blue (Sin dolor), 1-2 are green (Poco dolor), 3-5 are yellow (Dolor moderado), 6-7 are orange (Dolor fuerte), 8-9 are red (Dolor muy fuerte), and 10 is dark red (Dolor insoponible). Below the scale is a text input field with the placeholder "Introduce un valor entre 0 y 10" and a button labeled "ENVIAR".

Ilustración 4-9: Pantalla MainActivity.xml

- **Muestradolor**: este fichero posiblemente sea el más importante de todo el programa. Su función principal es elegir qué camino de los 4 posibles (explicado en el apartado 4.1) debe de elegir. En función del dolor introducido puede elegir el camino:
 - Grado de dolor bajo: apartado 4.1.1.
 - Grado de dolor medio: apartado 4.1.2.
 - Grado de dolor alto: apartado 4.1.3.
 - Grado de dolor muy alto: apartado 4.1.4.

Si el dolor introducido es erróneo, se vuelve a MainActivity de nuevo.
- **PreguntaEjercicio**: se pregunta al usuario sobre la realización o no realización del ejercicio físico. Pueden darse dos situaciones:
 - El usuario ha realizado ejercicio físico. Entonces la variable "Ejercicio" incrementa su valor en una unidad.
 - El usuario no ha realizado ejercicio físico. Entonces la variable "Ejercicio" no cambia su valor.

Tras esto se pasa a MensajeFormativo.



Ilustración 4-10: Pantalla PreguntaEjercicio.xml

- MensajeFormativo: muestra por pantalla un mensaje formativo que sea de ayuda para el paciente. Este sólo aparece cuando se está en el grado de dolor bajo.



Ilustración 4-11: Pantalla MensajeFormativo.xml

- MensajeInformativo: muestra por pantalla un mensaje informativo que sea de ayuda para el paciente. Este sólo aparece cuando se está en el grado de dolor medio.
- EscuelaEspalda: este es otro de los ficheros más importantes. Como se vio en el apartado 2.1, una de las maneras más efectivas de combatir las dolencias lumbares era acudir a escuelas de espalda. España dispone de su propia escuela de espalda. Este fichero sólo se ejecuta cuando se viene de los grados de

dolor medio, alto y muy alto. Esto se basa en la idea de que una de las posibles causas que han podido incrementar el dolor puede haber sido un sobreesfuerzo o una mal postura de espalda. Es por ello que en este fichero se muestra un enlace al portal web de la Escuela de Espalda Española. En función del nivel de dolor que ha sido pasado de Muestrador a este fichero, pueden darse tres situaciones:

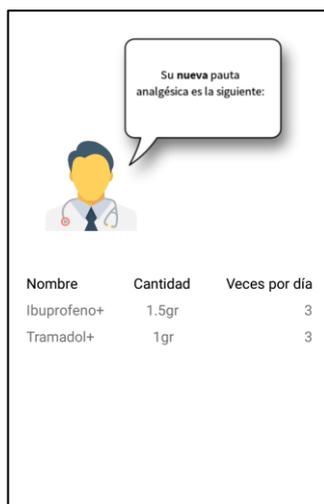
- A. Grado de dolor medio: se comprueba el valor de la variable “Pauta Analgésica” y como se explica en 4.1.2, se procede bien a poner una pauta analgésica (PautaAnalgésica), bien a mostrar un mensaje formativo (MensajeFormativo) o a realizar el test Morisky-Green (MoriskyGreen1).
- B. Grado de dolor alto: como se explica en 4.1.3 se procede a realizar directamente Morisky-Green (Morisky-Green1).
- C. Grado de dolor muy alto: se procede a realizar una llamada con el médico. Tras esto se pasa al menú inicial (MenuInicial).



Ilustración 4-12: Pantalla EscuelaEspalda.xml

- PautaAnalgésica: muestra por pantalla la pauta analgésica que el paciente tiene asignada. Se consulta el valor de la variable “Nivel alto”. Dependiendo del valor:
 - A. Si el valor es 0, se muestra un mensaje informativo (MensajeInformativo).
 - B. Si el valor es 1, se da una cita con el médico (Cita Medico).

Todo esto se hace para que de alguna manera se puedan reutilizar ficheros, ya que la pauta analgésica se puede poner tanto en el grado de dolor medio como en el grado de dolor alto.



| Nombre | Cantidad | Veces por día |
|-------------|----------|---------------|
| Ibuprofeno+ | 1.5gr | 3 |
| Tramadol+ | 1gr | 3 |

Ilustración 4-13: Pantalla PautaAnalgésica.xml

- MoriskyGreenX: hay 4 ficheros Morisky-Green, 1 por cada pregunta que se va a realizar. El fichero MoriskyGreen1, MoriskyGreen2 y MoriskyGreen3, muestran una estructura prácticamente similar. Muestra por pantalla una pregunta y el paciente puede elegir entre seleccionar “Sí” o “No”. Si el paciente en alguna de las preguntas responde “Sí” significa que es no cumplidor y por lo tanto se muestra por pantalla un mensaje recordatorio (MensajeRecordatorio). Si responde “No” se pasa a la pantalla inmediatamente superior.

El fichero MoriskyGreen4, además de tener integrado este funcionamiento, añade otro más. Se consulta el valor de “Pauta Rescate” y de “Nivel alto”. En función de los valores pueden pasar tres cosas:

- Si “Pauta Rescate” es igual a 0 y “Nivel alto” es igual a 0, significa que el paciente necesita una nueva pauta de rescate (Pauta Rescate).
- Si “Pauta Rescate” esta entre 0 y 3 y además “Nivel alto” es igual a 0, significa que el paciente está siguiendo el tratamiento que le ha sido impuesto, luego se muestra por pantalla un mensaje informativo (Mensaje Informativo).
- Si “Pauta Rescate” es igual a 3 y además “Nivel alto” es igual a 0, significa que la pauta de rescate no está siendo efectiva y el paciente necesita ser visto por el médico, luego se da cita con el mismo (Cita Médico).
- Si “Nivel alto” es igual a 1, significa que estamos en el grado de dolor alto y por lo tanto hay que proceder a subir la analgesia al paciente (PautaAnalgésica).

Todo esto se hace para que de alguna manera se puedan reutilizar ficheros, ya que el test de Morisky-Green se puede preguntar tanto en el grado de dolor medio como en el grado de dolor alto.



Ilustración 4-14: Pantalla MoriskyGreen.xml

- PautaRescate: muestra por pantalla la pauta de rescate que el paciente tiene asignada.

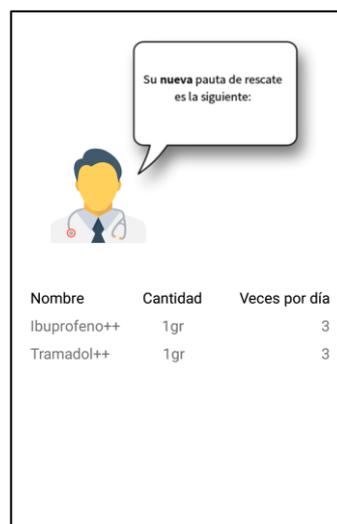


Ilustración 4-15: Pantalla PautaRescate.xml

- Recordatorio: cuando el resultado de Morisky-Green es negativo, es decir, el paciente no cumple con la toma de su medicación y por lo tanto se muestra por pantalla un mensaje recordando de la importancia de cumplir con las indicaciones médicas.
- MyFireBaseMessasingService: la aplicación móvil puede recibir notificaciones que el médico puede mandar desde el servicio de Google Firebase. Para controlar dichas notificaciones y que se puedan mostrar por pantalla, se crea esta clase Java.



Ilustración 4-16: Ejemplo de notificación creada

- MenuInicial: este fichero muestra un menú donde se dan 4 posibles opciones:
 - A. Consultar un histórico de su dolor (MuestraHistorico).
 - B. Consultar su mediación actual (MuestraMedicacion).
 - C. Consultar consejos sobre medicación, alimentación y ejercicio físico (MuestraEjercicio).
 - D. Consultar su entrenamiento actual (MuestraConsejos).

Este es el punto final de todo el algoritmo, es decir, cuando se acaban todos los procesos de los distintos grados de dolor, la pantalla final a la que se va es a la que muestra este fichero.



Ilustración 4-17: Pantalla MenuInicial.xml

- MuestraHistorico: su función es tomar los datos que han sido registrados en la base de datos del histórico de dolor de un paciente para un determinado mes (que el paciente puede elegir) y mostrar por pantalla dos gráficas:
 - A. Una gráfica lineal en la parte superior con todos los niveles (valores) de dolor registrados para ese mes.
 - B. Una PieChart que contiene el tanto por ciento que representa cada uno de los 4 grados de

dolor durante el mes seleccionado.

En caso de que el mes seleccionado no se encuentre registrado, se lanza un aviso por pantalla avisando de lo mismo.

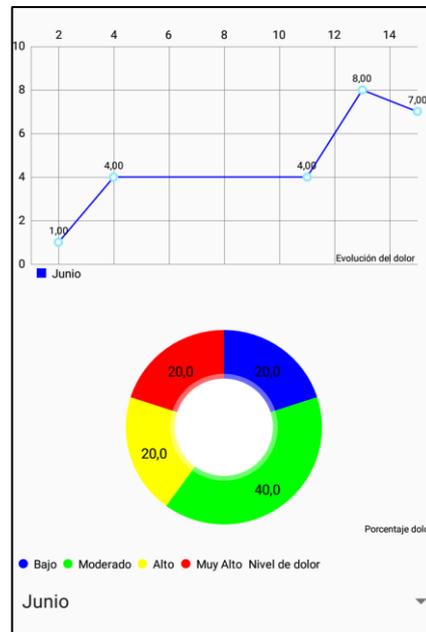


Ilustración 4-18: Pantalla MuestraHistorico.xml

- MuestraConsejos: la idea es que el paciente al acceder a este fichero, pueda consultar de manera aleatoria uno de los múltiples consejos que han sido introducido en la base de datos.



Ilustración 4-19: Pantalla MuestraConsejos.xml

- MuestraMedicacion: con este fichero el paciente puede consultar cual es a día de hoy su medicación actual para no perder nunca referencia.



Ilustración 4-20: Pantalla MuestraMedicacion.xml

- MuestraEjercicio: el ejercicio es fundamental en la recuperación del dolor de espalda. En esta pantalla se muestra una lista con enlaces a vídeos (que generalmente serán de Youtube) que muestran una serie de ejercicios que el paciente puede realizar (siempre que esté en condiciones de ello). Por cada video se le asigna un objeto de la clase YoutubeVideos.

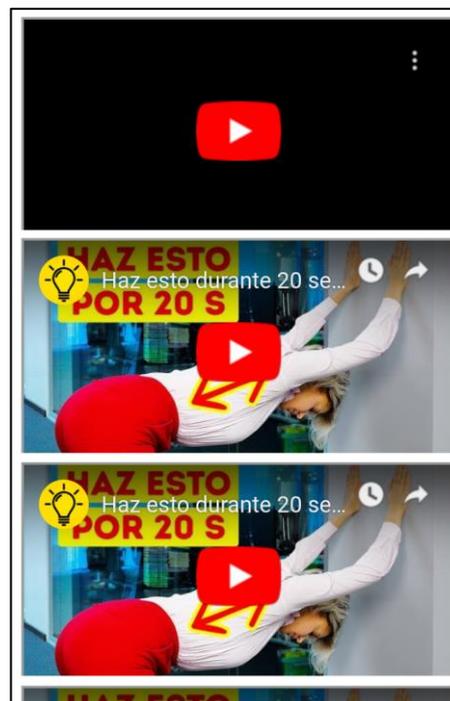


Ilustración 4-21: Pantalla MuestraEjercicio.xml

- VideoAdapter: contiene toda la lógica necesaria para mostrar por pantalla una lista con todos los vídeos que han sido tomado de la base de datos.
- YoutubeVideos: contiene la clase YoutubeVideos que se usa en MuestraEjercicio para asignar a cada vídeo un objeto del tipo anterior.

Estos son todos los ficheros .java que se usan en el programa, en el próximo capítulo se muestra cada uno de los layout asociados a cada clase y cuál es el aspecto final de los mismos una vez ejecutada la aplicación.

5 SERVIDOR WEB

En el capítulo anterior se explicó el algoritmo que hay detrás de la aplicación móvil, contenido de los ficheros que la componen así como la vista final de la misma. No se hizo mención aún de como la aplicación móvil se comunica con el servidor web y de cómo y dónde este guarda toda la información que recoge de la misma.

En este capítulo se explica con detalle todo aquello relacionado con el servidor web, es decir, la estructura, descripción y campos de cada una de las tablas que componen la base de datos, la configuración del entorno de trabajo Yii2 y una descripción de cada uno de los servicios REST que la aplicación móvil usa a fin de obtener los datos necesarios para su correcto funcionamiento.

El capítulo se divide en los siguientes apartados:

- Base de datos
- Configuración de Yii2
- Servicios REST
- Aplicación web

Antes de empezar cabe mencionar que el servidor corre en una máquina virtual (VMWare Workstation 15 Player) de CentOS7. Tras arrancarlo por primera vez se ha instalado el servidor Apache, el programa MySQL y phpMyAdmin, y por último el entorno de trabajo Yii2.

Una vez todo está configurado, se comprueba desde un ordenador que se encuentre en la misma red privada (esto es debido a que la versión de CentOS 7 utilizada no posee interfaz gráfica) que tanto Yii2, MySQL y phpMyAdmin corren correctamente. A partir de este punto, ya se está en disposición de configurar todo lo necesario para conseguir el objetivo principal del trabajo.

5.1 Base de datos

Cómo se explicó en el apartado 2.5, para facilitar la labor de creación de la base de datos, que recibe el nombre de appache (del inglés, app + ache, en español, aplicación + dolor) se ha hecho uso de MySQL y de phpMyAdmin. La siguiente figura muestra las tablas y relaciones de dicha base de datos:

| Tabla | Filas | Tipo | Tamaño | Comentarios |
|--------------------|-------|--------|--------|-------------------------------------|
| antecedentes | 0 | InnoDB | 16 KB | Creación: 18-05-2019 a las 23:34:12 |
| autenticacion | 1 | InnoDB | 16 KB | Creación: 11-07-2019 a las 19:35:29 |
| consejos | 2 | InnoDB | 16 KB | Creación: 26-06-2019 a las 12:10:51 |
| ejercicio | 15 | InnoDB | 32 KB | Creación: 26-06-2019 a las 05:00:46 |
| hist_eva | 8 | InnoDB | 32 KB | Creación: 18-05-2019 a las 23:17:12 |
| medicacion | 6 | InnoDB | 32 KB | Creación: 25-06-2019 a las 18:12:43 |
| mensaje_for | 0 | InnoDB | 16 KB | Creación: 25-06-2019 a las 21:20:11 |
| mensaje_infor | 1 | InnoDB | 16 KB | Creación: 25-06-2019 a las 21:22:41 |
| municipios | 2 | InnoDB | 16 KB | Creación: 18-05-2019 a las 03:37:37 |
| nivel_ejercicio | 4 | InnoDB | 16 KB | Creación: 26-06-2019 a las 03:24:27 |
| n_actividad_fisica | 4 | InnoDB | 16 KB | Creación: 18-05-2019 a las 03:38:13 |
| n_estudios | 3 | InnoDB | 16 KB | Creación: 18-05-2019 a las 03:38:25 |
| n_renta | 6 | InnoDB | 16 KB | Creación: 18-05-2019 a las 03:38:36 |
| paciente | 2 | InnoDB | 112 KB | Creación: 18-05-2019 a las 22:36:14 |
| register | 1 | InnoDB | 32 KB | Creación: 11-07-2019 a las 19:28:37 |
| rest | 120 | InnoDB | 32 KB | Creación: 18-05-2019 a las 23:54:59 |
| sexo | 2 | InnoDB | 16 KB | Creación: 18-05-2019 a las 03:38:48 |
| toma_medic | 2 | InnoDB | 16 KB | Creación: 18-05-2019 a las 23:21:23 |
| valores_medic | 3 | InnoDB | 16 KB | Creación: 25-06-2019 a las 18:08:34 |
| valores_rest | 11 | InnoDB | 16 KB | Creación: 18-05-2019 a las 23:44:48 |
| valores_var | 7 | InnoDB | 16 KB | Creación: 19-05-2019 a las 00:14:14 |
| variables | 7 | InnoDB | 32 KB | Creación: 25-06-2019 a las 02:12:37 |
| v_clinicas | 0 | InnoDB | 16 KB | Creación: 18-05-2019 a las 03:39:11 |
| 23 tablas | 207 | — | 560 KB | |

Ilustración 5-1. Estructura de la base de datos

Las tablas que componen la base de datos son las siguientes:

- **municipios**: contiene los municipios a los que los pacientes pertenecen:
 1. id_municipio (int, 11, clave primaria): identificador único de municipio.
 2. Nombre_municipio (varchar, 30): contiene el nombre del municipio.
- **n_actividad_fisica**: dado que cada paciente puede realizar con mayor o con menor frecuencia ejercicio físico ya sea por cuestiones meramente físicas o bien por cuestiones laborales, se hace una clasificación del nivel de actividad física al que se somete cada paciente:
 1. id_actividad (int, 11, clave primaria): identificador único de nivel de actividad física.
 2. nivel (varchar, 20): contiene una descripción de dicho nivel.

+ Opciones

| | id_actividad | nivel |
|---|--------------|---|
| <input type="checkbox"/> Editar Copiar Borrar | 1 | Sentado/a la mayor parte de la jornada |
| <input type="checkbox"/> Editar Copiar Borrar | 2 | De pie la mayor parte de la jornada sin esfuerzos |
| <input type="checkbox"/> Editar Copiar Borrar | 3 | Caminando, llevando peso o con desplaz frec |
| <input type="checkbox"/> Editar Copiar Borrar | 4 | Realizando tareas que requieren gran esfuerzo |

Ilustración 5-2. Valores de la tabla “n_actividad_fisica”

- n_estudios: cada paciente puede tener un menor o mayor formación académica. Es interesante recoger dicha información, es por ello que se hace una distinción por niveles de estudio. Sus campos son los siguientes:
 1. id_estudio (int, 11, clave primaria): identificador único de nivel de actividad física.
 2. nivel_estudios (varchar, 20): contiene una descripción para dicho nivel de estudios.

+ Opciones

| | | id_estudio | nivel_estudios |
|--------------------------|----------------------|------------|------------------|
| <input type="checkbox"/> | Editar Copiar Borrar | 1 | Sin estudios |
| <input type="checkbox"/> | Editar Copiar Borrar | 2 | E primarios |
| <input type="checkbox"/> | Editar Copiar Borrar | 3 | E secundarios |
| <input type="checkbox"/> | Editar Copiar Borrar | 4 | E universitarios |

Ilustración 5-3. Valores de la tabla “n_estudios”

- n_renta: el nivel de renta también puede influir en el dolor de espalda, pues se asocia a trabajos más físicos con mayores probabilidades de aparición de dolores lumbares. Los campos que componen esta tabla son los siguientes:
 1. id_renta (int, 11, clave primaria): identificador único de nivel de renta.
 2. nivel_renta (varchar, 15): descripción de dicho nivel de renta.

+ Opciones

| | | id_renta | nivel_renta |
|--------------------------|----------------------|----------|-----------------|
| <input type="checkbox"/> | Editar Copiar Borrar | 1 | < 10.000 € |
| <input type="checkbox"/> | Editar Copiar Borrar | 2 | 10.000-18.000 € |
| <input type="checkbox"/> | Editar Copiar Borrar | 3 | 18.000-24.000 € |
| <input type="checkbox"/> | Editar Copiar Borrar | 4 | 24.000-36.000 € |
| <input type="checkbox"/> | Editar Copiar Borrar | 5 | 36.000-50.000 € |
| <input type="checkbox"/> | Editar Copiar Borrar | 6 | > 50.000 € |

Ilustración 5-4. Valores de la tabla “n_renta”

- sexo: contiene los sexos de los pacientes.
 1. idsexo (int, 11, clave primaria): identificador único del sexo.
 2. sexo (varchar, 6): nombre del sexo.

| + Opciones | | id_sexo | sexo |
|--------------------------|--------|---------|--------|
| <input type="checkbox"/> | Editar | 1 | Mujer |
| <input type="checkbox"/> | Editar | 2 | Hombre |

Ilustración 5-5. Valores de la tabla “sexo”

- paciente: esta tabla es fundamental, pues contiene todos los datos de relevancia de un paciente. Los campos que lo componen son los siguientes:
 1. NID (int, 11, clave primaria): es un identificador único de paciente. Es fundamental en muchas otras tablas, pues identifica al paciente de manera unívoca.
 2. NUHSA (varchar, 14): dado que un principio, este proyecto está pensado para Andalucía, es importante el Número Único de Historia de Salud de Andalucía.
 3. Apellido1 (varchar, 20): contiene el primer apellido del paciente.
 4. Apellido2 (varchar, 20): contiene el segundo apellido del paciente.
 5. Nombre (varchar, 20): contiene el nombre del paciente.
 6. Fech_nacimiento (date): fecha de nacimiento del paciente.
 7. sexo (int, 11): sexo del paciente. Su valor depende de la tabla “sexo”.
 8. municipio (int, 11): municipio del paciente. Su valor depende de la tabla “municipio”.
 9. a_fisica (int, 11): nivel de actividad física. Su valor depende de la tabla “n_actividad_fisica”.
 10. n_renta (int, 11): nivel de renta. Su valor depende de la tabla “n_renta”.
 11. num_telef (varchar, 45): número de teléfono del paciente.
 12. correo_E (varchar, 45): correo electrónico del paciente.
 13. n_estudios (int, 11): nivel de estudios. Su valor depende de la tabla “n_estudios”.
- hist_eva: una de las tablas más importantes, pues contiene el nivel de dolor de un paciente para un determinado día. Sus campos son:
 1. NID (int, 11, clave primaria): identificador único de paciente.
 2. Valor (tinyint, 4): nivel de dolor del paciente.
 3. Fecha (date): fecha de registro de dicho nivel de dolor.
 4. Num_hist (int, 11, clave primaria): número que representa dicho registro.
- v_clinicas: contiene variable clínicas propias del paciente que es interesante registrar para tener en cuenta. Está formada por:
 1. NID (int, 11, clave primaria): identificador único del paciente.
 2. Peso (varchar, 45): contiene el peso del paciente.
 3. Talla (varchar, 45): contiene la altura del paciente.
 4. IMC (varchar, 45): contiene el valor Índice de Masa Corporal del paciente.
 5. T_art_sistolica (varchar, 45): valor de la tensión arterial sistólica del paciente.
 6. T_art_diastolica (varchar, 45): valor de la tensión arterial diastólica del paciente.
- antecedentes: para sacar conclusiones finales es importante conocer los antecedentes médicos del

paciente como si sufre de hipertensión o si tiene alguna enfermedad renal crónica. Los campos que componen dicha tabla son los siguientes:

1. NID (int, 11, clave primaria): identificador único de paciente.
 2. Hipertensión (tinyint, 1): indica si el paciente sufre hipertensión. 1 Sí, 0 No.
 3. E_renal_crónica (tinyint, 1): indica si el paciente sufre de una enfermedad renal crónica. 1 Sí, 0 No.
- valores_var: esta tabla se crea con el fin de identificar con un número las posibles variables que se van a usar para lograr cumplir con el algoritmo de la aplicación móvil:
1. id (int, 11, clave primaria): identificador único de la variable.
 2. valor (varchar, 30): nombre identificativo de la variable.

+ Opciones

| | | | | | id | valor | | |
|--------------------------|---|--------|---|--------|---|--------|---|-----------------|
| <input type="checkbox"/> |  | Editar |  | Copiar |  | Borrar | 1 | EVA |
| <input type="checkbox"/> |  | Editar |  | Copiar |  | Borrar | 2 | D_ALTO |
| <input type="checkbox"/> |  | Editar |  | Copiar |  | Borrar | 3 | DIA |
| <input type="checkbox"/> |  | Editar |  | Copiar |  | Borrar | 4 | EJERCICIO |
| <input type="checkbox"/> |  | Editar |  | Copiar |  | Borrar | 5 | PANALGESICA |
| <input type="checkbox"/> |  | Editar |  | Copiar |  | Borrar | 6 | PRESCATE |
| <input type="checkbox"/> |  | Editar |  | Copiar |  | Borrar | 7 | NIVEL EJERCICIO |

Ilustración 5-6. Valores de la tabla “valores_var”

- variables: a fin de poder hacer uso de las variables para que la aplicación funcione correctamente, se crea esta tabla que contiene los valores actualizado de dichas variables. Sus campos son:
1. NID (int, 11, clave primaria): identificador único del paciente.
 2. variable (int, 11, clave primaria): indica a que variable se refiere dicha fila. Su valor depende de la tabla “valores_var”.
 3. valor (int, 11): valor de la variable.
 4. instante (timestamp): último instante en el que se modificó el valor de la variable.
- valores_rest: esta tabla se crea con el fin de identificar con un número los posibles servicios REST de los que la aplicación móvil puede hacer uso:
1. id (int, 11, clave primaria): identificador único del servicio REST.
 2. significado (varchar, 30): nombre identificativo del servicio.

| + Opciones | | | | id | significado | | | |
|--------------------------|--|--------|--|--------|-------------|--------|----|---------------------|
| <input type="checkbox"/> | | Editar | | Copiar | | Borrar | 1 | Registra EVA |
| <input type="checkbox"/> | | Editar | | Copiar | | Borrar | 2 | Registra Ejercicio |
| <input type="checkbox"/> | | Editar | | Copiar | | Borrar | 3 | Consulta nombre |
| <input type="checkbox"/> | | Editar | | Copiar | | Borrar | 4 | Consulta Histórico |
| <input type="checkbox"/> | | Editar | | Copiar | | Borrar | 5 | Consulta Ejercicio |
| <input type="checkbox"/> | | Editar | | Copiar | | Borrar | 6 | Consulta Consejos |
| <input type="checkbox"/> | | Editar | | Copiar | | Borrar | 7 | Consulta Medicacion |
| <input type="checkbox"/> | | Editar | | Copiar | | Borrar | 8 | Comprueba variables |
| <input type="checkbox"/> | | Editar | | Copiar | | Borrar | 9 | Actualiza variables |
| <input type="checkbox"/> | | Editar | | Copiar | | Borrar | 10 | Mensaje Formativo |
| <input type="checkbox"/> | | Editar | | Copiar | | Borrar | 11 | Mensaje Informativo |

Ilustración 5-7. Valores de la tabla “valores_var”

- rest: a fin de conseguir estadísticas que puedan ser de interés, cada vez que se accede a un servicio REST, se crea una línea en esta tabla.
 1. NID (int, 11, clave primaria): identificador único del paciente.
 2. Valor (int, 11, clave primaria): este campo indica a qué servicio REST se ha accedido. Su valor depende de la tabla “valores_rest”.
 3. Instante (timestamp, clave primaria): instante en el que se accede al servicio REST.
- nivel_ejercicio: establece los posibles niveles de ejercicio físico recomendados al paciente. Sus campos son:
 1. id (int, 11, clave primaria): identificador único de nivel de ejercicio.
 2. nivel (varchar, 60): contiene un nombre identificativo de dicho nivel.

| + Opciones | | | | id | nivel | | | |
|--------------------------|--|--------|--|--------|-------|--------|---|--------------|
| <input type="checkbox"/> | | Editar | | Copiar | | Borrar | 1 | Principiante |
| <input type="checkbox"/> | | Editar | | Copiar | | Borrar | 2 | Intermedio |
| <input type="checkbox"/> | | Editar | | Copiar | | Borrar | 3 | Profesional |
| <input type="checkbox"/> | | Editar | | Copiar | | Borrar | 4 | Experto |

Ilustración 5-8. Valores de la tabla “nivel_ejercicio”

- ejercicio: esta tabla contiene todos los ejercicios que se van a recomendar al paciente. Su estructura es la siguiente:
 1. id (int, 11, clave primaria): identificador único del ejercicio para un nivel determinado.
 2. nivel (int, 3, clave primaria): indica el nivel del ejercicio. Dicho valor está limitado por los de la tabla “nivel_ejercicio”.
 3. dirección (text, 80): contiene la url del vídeo en el que se muestra el ejercicio.
- valores_medic: esta tabla se crea con el fin de identificar con un número alguno de los 3 posibles “tipos” de medicación que un paciente puede tener recetado:
 1. id (int, 11, clave primaria): identificador único del tipo de medicación.
 2. valor (varchar, 30): nombre identificativo del tipo de medicación.

| + Opciones | | | | | | |
|--------------------------|---|---|---|----|--------------------|--|
| | | | | id | valor | |
| <input type="checkbox"/> |  Editar |  Copiar |  Borrar | 1 | Medicacion Inicial | |
| <input type="checkbox"/> |  Editar |  Copiar |  Borrar | 2 | Pauta Analgésica | |
| <input type="checkbox"/> |  Editar |  Copiar |  Borrar | 3 | Pauta Rescate | |

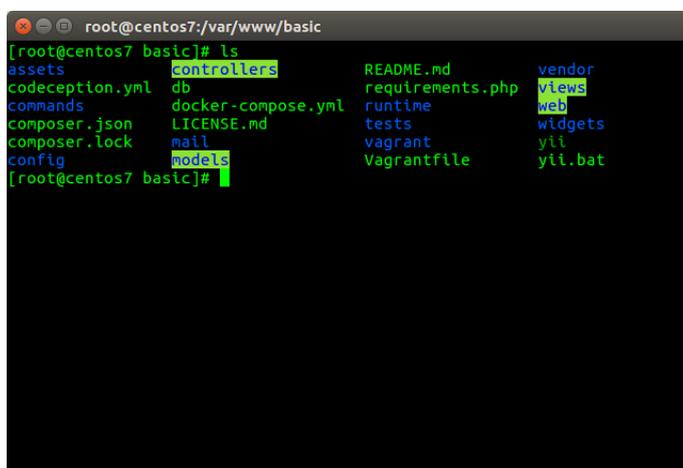
Ilustración 5-9. Valores de la tabla “valores_medic”

- medicacion: contiene la medicación de un determinado paciente. Cada paciente tiene asignado tres tipos, una basal, una pauta analgésica y una pauta de rescate. En función del grado de dolor, tal y como se ha visto en capítulos anteriores, se le asigna una u otra. Sus campos son los siguientes:
 1. NID (int, 11, clave primaria): identificador único de paciente.
 2. Nombre (varchar, 30, clave primaria): nombre de la medicación.
 3. Tipo (int, 11, clave primaria): tipo de medicación a la que pertenece. Depende de los valores de “valores_medic”.
 4. Cantidad (varchar, 30): cantidad de medicamento que se debe de tomar.
 5. Horario (varchar, 30): número de tomas por día.
 6. Fecha (date): fecha en la que se ha recetado dicha medicación.
- toma_medic: esta tabla contiene un registro de la toma de medicinas por parte del paciente. Su campos son:
 1. NID (int, 11, clave primaria): identificador único del paciente.
 2. Valor (tinyint, 1): indica si el paciente ha tomado la medicación. 1 Sí, 0 No.
 3. Fecha (date): fecha en la que se ha registrado dicho valor.
 4. Num_toma (tinyint, 4, clave primaria): número de toma que representa dicho valor.
- mensaje_for: contiene los mensajes formativos que se van a mostrar al paciente.
 1. id (int, 11, clave primaria): identificador único del mensaje formativo.
 2. mensaje (text, 80): contiene el texto del mensaje que se va a mostrar.

3. url (text, 80): contiene la url de la imagen que se va a mostrar.
- **mensaje_infor**: contiene los mensajes informativos que se van a mostrar al paciente.
 1. id (int, 11, clave primaria): identificador único del mensaje informativo.
 2. mensaje (text, 80): contiene el texto del mensaje que se va a mostrar.
 3. url (text, 80): contiene la url de la imagen que se va a mostrar.
 - **consejos**: contiene los consejos que se van a mostrar al paciente. Además contiene las url de cada una de las fotografías que van a acompañar al consejo. Sus campos son:
 1. id (int, 11, clave primaria): identificador único del consejo.
 2. consejo (text, 80): contiene el texto del consejo que se va a mostrar.
 3. url (text, 80): contiene la dirección de la imagen que se va a mostrar.
 - **autenticación**: su función es registrar la dirección MAC de un paciente cuando este accede por primera vez. Para ello el cliente necesita introducir su NID, clave y número de teléfono. Sus campos son:
 1. NID (int, 20, clave primaria): identificador único del paciente.
 2. clave (varchar, 20): clave que el médico tiene que suministrar al paciente para su correcta identificación.
 3. teléfono (int, 20): número de teléfono del paciente.
 4. MAC (varchar, 20): dirección MAC del teléfono del paciente.
 - **register**: contiene el token asociado a un paciente. Su valor puede ir cambiando en el tiempo. Sus campos son:
 1. NID (int, 20, clave primaria): identificador único del paciente.
 2. Token (varchar, 200): contiene el identificar único que Google suministra al usuario.

5.2 Configuración de Yii2

Una vez ha sido creada la base de datos con todas las tablas y campos correctamente creados, el siguiente paso es poner en contacto el entorno Yii2 con dicha base de datos. Antes de empezar a explicar es importante conocer que dicho entorno se ha instalado en el directorio “/www/var/basic”. En la siguiente imagen podemos observar la lista de directorios que se hallan dentro de dicha carpeta:

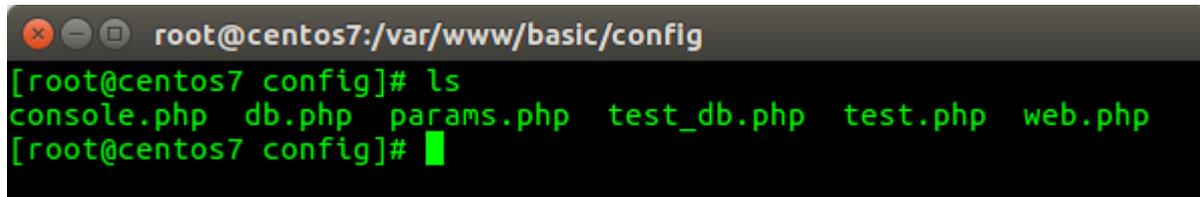


```
root@centos7:/var/www/basic
[root@centos7 basic]# ls
assets          controllers     README.md     vendor
codeception.yml db              requirements.php views
commands       docker-compose.yml runtime        web
composer.json  LICENSE.md    tests         widgets
composer.lock  mail          vagrant       yii
config         models        Vagrantfile   yii.bat
[root@centos7 basic]#
```

Ilustración 5-10. Directorios dentro de la carpeta basic

De entre todos los directorios que aparecen en la fotografía anterior, los que son más usados y que tienen más importancia en el desarrollo del proyecto son los siguientes:

- config: contiene todos aquellos archivos que tienen que ver con la configuración de yii. Entre ellos destacan dos archivos:

A terminal window screenshot showing the command 'ls' being executed in the directory '/var/www/basic/config'. The output lists several files: console.php, db.php, params.php, test_db.php, test.php, and web.php. The terminal title is 'root@centos7:/var/www/basic/config' and the prompt is '[root@centos7 config]#'.

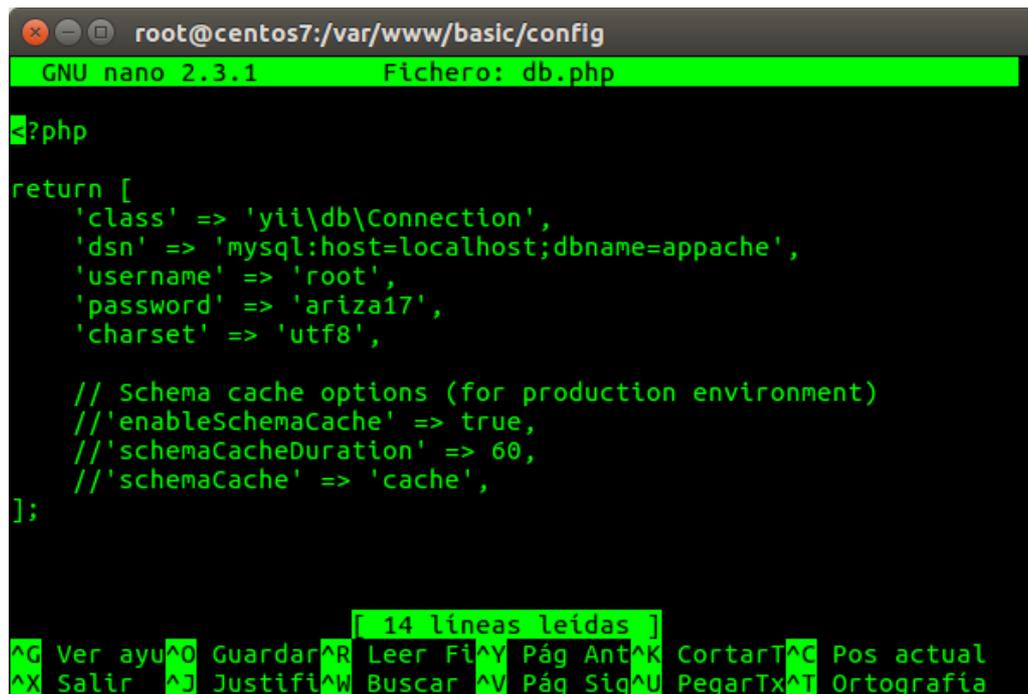
```
root@centos7:/var/www/basic/config
[root@centos7 config]# ls
console.php db.php params.php test_db.php test.php web.php
[root@centos7 config]#
```

Ilustración 5-11. Ficheros que componen el directorio config

- A. db.php: este archivo tiene como función crear la conexión de Yii2 con la base de datos MySQL.
 - B. web.php: contiene todo aquello relacionado con nuestra aplicación web y su administración, desde permisos, hasta reglas de ejecución cómo de des/activación de módulos Yii.
- controllers: cómo se explicó en el apartado 2.6, Yii2 estaba pensado según el Modelo-Vista-Controlador. En el interior del directorio controllers, se tienen todo los controladores que son usados en nuestra aplicación. Cómo se explica más adelante, en nuestro caso existe más de un controlador por modelo.
 - models: contiene todos los modelos de cada una de las tablas que se han declarado en la base de datos.
 - views: este modelo contiene todas las vistas que son usadas por Yii2 para mostrar el contenido de los distintos modelos.

Tanto los controladores, como modelos y vistas (a excepción de los controladores para servicios REST), son creados a través de Gii, que de manera rápida y eficaz permite crear los ficheros PHP que sean necesarios.

Cómo se ha mencionado anteriormente, el primer paso que hay que realizar es poner en contacto la base de datos con Yii [9]. Dicha conexión se hace a través del fichero db.php como se muestra en la siguiente imagen:



```

root@centos7:/var/www/basic/config
GNU nano 2.3.1 Fichero: db.php
<?php
return [
    'class' => 'yii\db\Connection',
    'dsn' => 'mysql:host=localhost;dbname=apache',
    'username' => 'root',
    'password' => 'ariza17',
    'charset' => 'utf8',

    // Schema cache options (for production environment)
    //'enableSchemaCache' => true,
    //'schemaCacheDuration' => 60,
    //'schemaCache' => 'cache',
];
14 líneas leídas
Ver ayuda Guardar Leer Fichero Pág Ant Cortar Pos actual
Salir Justificar Buscar Pág Sig Pegar Ortografía

```

Ilustración 5-12. Configuración del fichero db.php

Sólo es necesario modificar el contenido de 3 líneas:

- Dsn: dónde se tiene que especificar la dirección y nombre de la base de datos. En este caso, la dirección es local y el nombre de la base de datos como vimos en el anterior apartado es apache.
- Username: nombre de usuario MySQL.
- Password: contraseña de usuario MySQL.

Tras esto ya existe conexión directa entre Yii2 y la base de datos. El siguiente paso es crear los modelos, controladores y vistas que vayan a ser usados. Gii [10] da la opción de generar automáticamente cada uno de los anteriores. Para activar Gii, es necesario activar dicho módulo en el fichero web.php:

```

$config = [ ... ];

if (YII_ENV_DEV) {
    $config['bootstrap'][] = 'gii';
    $config['modules']['gii'] = [
        'class' => 'yii\gii\Module',
    ];
}

```

Ilustración 5-13. Activación del módulo gii

La creación de un modelo usando gii es bastante intuitiva como se puede ver en la siguiente imagen:

Ilustración 5-14. Creación de un modelo usando gii

Además de modelos, gii permite generar código CRUD (Create, Read, Update, View) para qué, directamente, a través del navegador web se pueda crear una vista dónde se puedan ver, crear, actualizar y leer, todos los registros de una determinada tabla de la base de datos creada:

Ilustración 5-15. Generación de código CRUD usando gii

- Model Class: se introduce la ruta del modelo que se va a usar app\models\Country
- Search Model Class: el módulo de búsqueda se genera automáticamente si no ha sido creado previamente app\models\CountrySearch
- Controller Class: se introduce la ruta del controlador de dicha vista, en este caso app\controllers\CountryController
- View Path: se introduce la dirección de la vista que se va a usar para visualizar dicho modelo.

Si se introduce en el navegador “http://hostname/paciente” se muestra de la siguiente forma:

My Application Home About Contact Login

Home / Pacientes

Pacientes

Create Paciente

Showing 1-2 of 2 items.

| # | Nid | Nuhsa | Apellido1 | Apellido2 | Nombre | |
|---|-----|----------------|-----------|-----------|---------|---|
| 1 | 1 | 1234536356678A | Ariza | Llamas | Antonio |    |
| 2 | 2 | 1234536356678B | Aguilar | Jimenez | Paco |    |

Ilustración 5-16. Vista del modelo Paciente

Haciendo click en “Create Nombre_de_la_vista” o en algunos de los iconos que aparecen al final de cada fila se puede crear, ver, editar o borrar un determinado elemento (en este caso, un registro de dolor):

Crear un registro de dolor

Nid

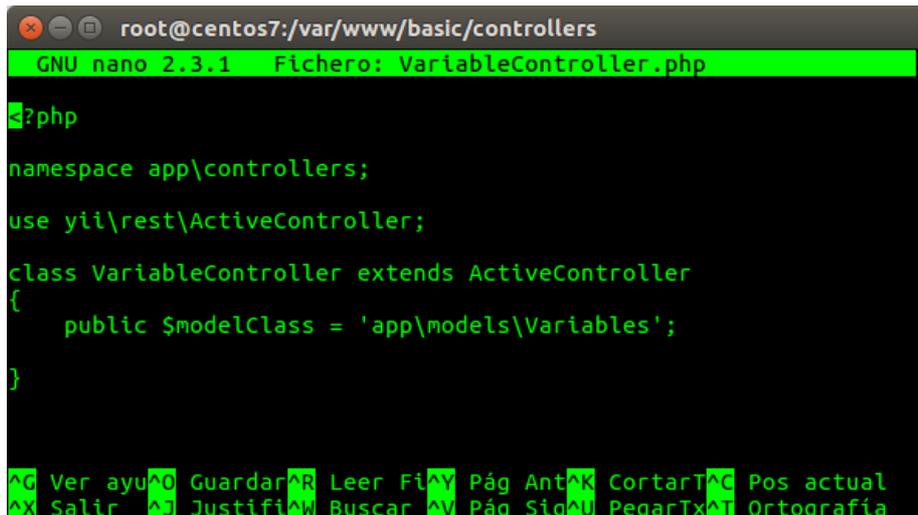
Valor

Fecha

Aceptar

Ilustración 5-17. Ejemplo de uso

Además de esto, en el apartado 2.7 se mencionó y explicó que la aplicación móvil iba a hacer uso de servicios REST. Yii2 permite crear controladores REST [11] de tal manera que, desde la aplicación móvil no sea necesario programar cada una de las consultas a la base de datos. El contenido de dicho controlador (1 por modelo que vaya a ser usado) es el siguiente:



```

root@centos7:/var/www/basic/controllers
GNU nano 2.3.1  Fichero: VariableController.php

<?php

namespace app\controllers;

use yii\rest\ActiveController;

class VariableController extends ActiveController
{
    public $modelClass = 'app\models\Variables';
}

^G Ver ayuda ^O Guardar ^R Leer Fich ^Y Pág Ant ^K CortarTx ^C Pos actual
^X Salir ^J Justificar ^W Buscar ^V Pág Sig ^U PegarTx ^T Ortografía

```

Ilustración 5-18. Controlador REST

Además de crear cada uno de estos controladores, es necesario modificar (de nuevo) el fichero de configuración web.php. En este caso lo primero que se hace es habilitar el “JSONParser”, ya que la aplicación móvil está programada en Java y va a manejar objetos tipo JSON:

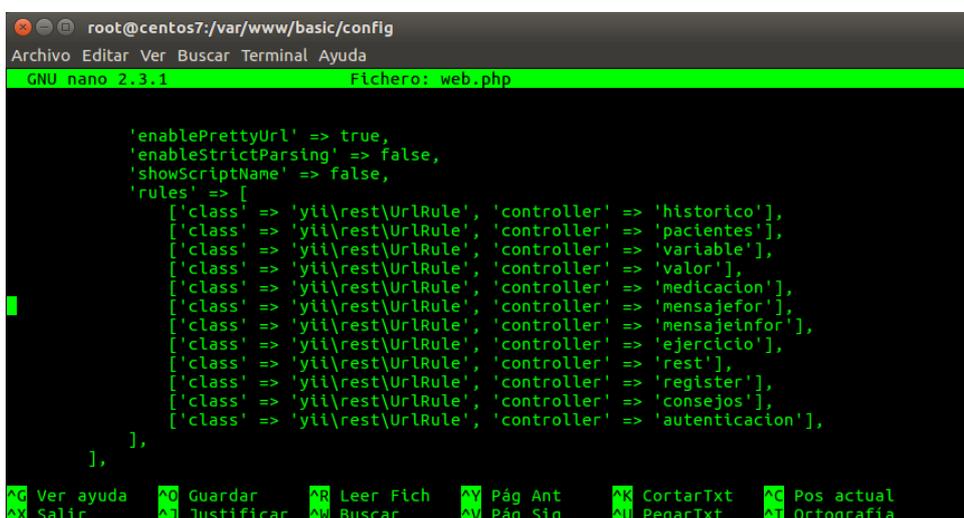
```

'request' => [
    'parsers' => [
        'application/json' => 'yii\web\JsonParser',
    ]
]

```

Ilustración 5-19. Activación JSONParser

Por último, es necesario activar el urlManager para que a través del navegador web se pueda obtener una vista de lo que el servicio REST muestra. Hay que añadir una línea por cada controlador REST que vaya a ser usado:



```

root@centos7:/var/www/basic/config
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.3.1  Fichero: web.php

'enablePrettyUrl' => true,
'enableStrictParsing' => false,
'showScriptName' => false,
'rules' => [
    ['class' => 'yii\rest\UrlRule', 'controller' => 'historico'],
    ['class' => 'yii\rest\UrlRule', 'controller' => 'pacientes'],
    ['class' => 'yii\rest\UrlRule', 'controller' => 'variable'],
    ['class' => 'yii\rest\UrlRule', 'controller' => 'valor'],
    ['class' => 'yii\rest\UrlRule', 'controller' => 'medicacion'],
    ['class' => 'yii\rest\UrlRule', 'controller' => 'mensajefor'],
    ['class' => 'yii\rest\UrlRule', 'controller' => 'mensajeinfor'],
    ['class' => 'yii\rest\UrlRule', 'controller' => 'ejercicio'],
    ['class' => 'yii\rest\UrlRule', 'controller' => 'rest'],
    ['class' => 'yii\rest\UrlRule', 'controller' => 'register'],
    ['class' => 'yii\rest\UrlRule', 'controller' => 'consejos'],
    ['class' => 'yii\rest\UrlRule', 'controller' => 'autenticacion'],
],
],
^G Ver ayuda ^O Guardar ^R Leer Fich ^Y Pág Ant ^K CortarTx ^C Pos actual
^X Salir ^J Justificar ^W Buscar ^V Pág Sig ^U PegarTx ^T Ortografía

```

Ilustración 5-20. Configuración de web.php

Desde el navegador ya se puede acceder a ver el contenido de alguno de los servicios que ofrece las APIs REST creadas:

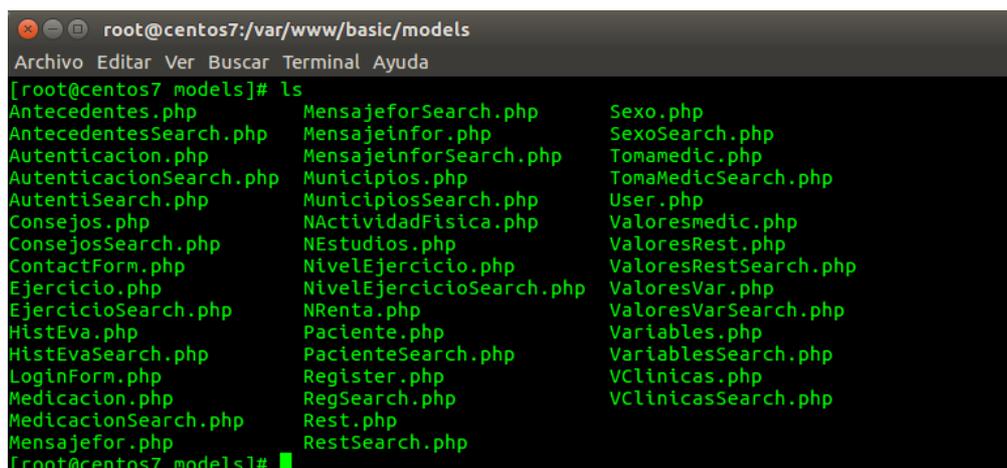
Este fichero XML no parece tener ninguna información de estilo asociada. Se muestra debajo el árbol del documento.

```
--<response>
- <item>
  <NID>1</NID>
  <NUHSA>1234536356678A</NUHSA>
  <Apellido1>Ariza</Apellido1>
  <Apellido2>Llamas</Apellido2>
  <Nombre>Antonio</Nombre>
  <Fech_nacimiento>1996-03-04</Fech_nacimiento>
  <sexo>2</sexo>
  <municipio>3</municipio>
  <a_fisica>3</a_fisica>
  <n_renta>1</n_renta>
  <num_telef>636554267</num_telef>
  <correo_e>antonioariza96@gmail.com</correo_e>
  <n_estudios>4</n_estudios>
</item>
- <item>
  <NID>2</NID>
  <NUHSA>1234536356678B</NUHSA>
  <Apellido1>Aguilar</Apellido1>
  <Apellido2>Jimenez</Apellido2>
  <Nombre>Paco</Nombre>
  <Fech_nacimiento>1996-08-05</Fech_nacimiento>
  <sexo>1</sexo>
  <municipio>1</municipio>
  <a_fisica>1</a_fisica>
  <n_renta>1</n_renta>
  <num_telef>615284192</num_telef>
  <correo_e>pacoag96@gmail.com</correo_e>
  <n_estudios>3</n_estudios>
</item>
</response>
```

Ilustración 5-21. Ejemplo de servicio REST

Todos los modelos, controladores y vistas que se han creado se muestran en las siguientes capturas:

- Modelos:



```
root@centos7:/var/www/basic/models
Archivo Editar Ver Buscar Terminal Ayuda
[root@centos7 models]# ls
Antecedentes.php      MensajeforSearch.php  Sexo.php
AntecedentesSearch.php Mensajeinfor.php      SexoSearch.php
Autenticacion.php     MensajeinforSearch.php Tomamedic.php
AutenticacionSearch.php Municipios.php         TomaMedicSearch.php
AutentiSearch.php     MunicipiosSearch.php  User.php
Consejos.php          NActividadFisica.php  Valoresmedic.php
ConsejosSearch.php   NEstudios.php         ValoresRest.php
ContactForm.php       NivelEjercicio.php    ValoresRestSearch.php
Ejercicio.php        NivelEjercicioSearch.php ValoresVar.php
EjercicioSearch.php  NRenta.php            ValoresVarSearch.php
HistEva.php          Paciente.php          Variables.php
HistEvaSearch.php    PacienteSearch.php   VariablesSearch.php
LoginForm.php        Register.php          VClinicas.php
Medicacion.php       RegSearch.php         VClinicasSearch.php
MedicacionSearch.php Rest.php
Mensajefor.php       RestSearch.php
[root@centos7 models]#
```

Ilustración 5-22. Modelos creados.

- Controladores:

```

root@centos7:/var/www/basic/controllers
Archivo Editar Ver Buscar Terminal Ayuda
[root@centos7 controllers]# ls
AntecedentesController.php  MenforController.php      SexoController.php
AutenticacionController.php  MeninforController.php   SiteController.php
AutentiController.php       MensajeforController.php  SrestController.php
ConsejController.php        MensajeinforController.php TomamedicController.php
ConsejosController.php      MunicipiosController.php  ValorController.php
EjercicioController.php     NejercicioController.php  ValrestController.php
EjerController.php          PacienteController.php   ValvarController.php
HistController.php          PacientesController.php  VarController.php
HistoricoController.php     RegController.php        VariableController.php
MedicacionController.php    RegisterController.php   VclinicasController.php
MedicController.php         RestController.php
[root@centos7 controllers]#

```

Ilustración 5-23. Controladores creados

- Vistas:

```

root@centos7:/var/www/basic/views
Archivo Editar Ver Buscar Terminal Ayuda
[root@centos7 views]# ls
antecedentes  ejer  medic  municipios  reg  site  valrest  vclinicas
autenti      hist  menfor  nejerccio  rest  srest  valvar
consej       layouts  meninfor  paciente  sexo  tomamedic  var
[root@centos7 views]#

```

Ilustración 5-24. Vistas creadas

Para tener una visión general de la relación de cada modelo con sus controladores y vista, se suministra al lector la siguiente tabla (en la misma no aparecen los controladores REST, puesto que se le va a dedicar un apartado entero para su explicación en profundidad):

Tabla 5-1. Relaciones Modelo-Vista-Controlador (CRUD)

| Modelo | Vista | Controlador |
|------------------|--------------|----------------------------|
| Medicacion.php | medic | MedicController.php |
| HistEva.php | hist | HistController.php |
| Rest.php | srest | SrestController.php |
| Paciente.php | paciente | PacienteController.php |
| Sexo.php | sexo | SexoController.php |
| Consejos.php | consej | ConsejController.php |
| Antecedentes.php | antecedentes | AntecedentesController.php |
| Ejercicio.php | ejer | EjerController.php |
| Mensajefor.php | menfor | MenforController.php |
| Mensajeinfor.php | meninfor | MeninforController.php |
| Municipios.php | municipios | MunicipiosController.php |

| | | |
|--------------------|------------|--------------------------|
| Tomamedic.php | tomamedic | TomamedicController.php |
| ValoresRest.php | valrest | ValrestController.php |
| ValoresVar.php | valvar | ValvarController.php |
| NivelEjercicio.php | nejercicio | NejercicioController.php |
| VClinicas.php | vclinicas | VclinicasController.php |
| Autenticacion.php | autenti | AutentiController.php |
| Register.php | reg | RegController.php |

Todas las vistas anteriores han sido generadas con Gii haciendo uso de CRUD. No se ha mencionado nada de REST, pues se va a realizar una descripción con más detenimiento en el siguiente apartado cada uno de los servicios REST que la app móvil va a usar.

5.3 Servicios REST

En este apartado se va a mostrar una descripción detallada de cada uno de los servicios REST que son usados. Cada vez que es usado alguno de estos servicios REST, en la tabla “rest” que aparece en el apartado 5.1, se crea una nueva entrada en la que aparece un número identificativo del servicio REST junto con el NID (identificador único del paciente) y el momento en el que se ha accedido a él (Timestamp). Esto se hace por el mero hecho de sacar estadísticas o analizar comportamientos que puedan ser de utilidad en un futuro.

Al igual que en el apartado anterior, se va a mostrar una tabla. En este caso con los controladores REST que son usados para dar todos los servicios REST (que se describen más adelante):

Tabla 5-2. Relaciones Modelo-Vista-Controlador (REST)

| Modelo | Vista | Controlador |
|-------------------|---------------|-----------------------------|
| Medicacion.php | medicacion | MedicacionController.php |
| HistEva.php | historico | HistoricoController.php |
| Rest.php | rest | RestController.php |
| Paciente.php | pacientes | PacientesController.php |
| Consejos.php | consejos | ConsejosController.php |
| Ejercicio.php | ejercicio | EjercicioController.php |
| Mensajefor.php | mensajefor | MensajeforController.php |
| Mensajeinfor.php | mensajeinfor | MensajeinforController.php |
| Variable.php | variable | VariableController.php |
| Autenticacion.php | autenticación | AutenticacionController.php |
| Register.php | register | RegisterController.php |

Cada uno de estos controladores se puede usarse en más de una ocasión para dar distintos servicios en función del objetivo que se quiere conseguir. En el caso de Yii, las operaciones que se permiten son las siguientes (se pone como ejemplo el controlador users):

- GET: /users: una lista de todos los usuarios página por página;
- HEAD: /users: muestra la información general de la lista de usuarios;
- POST: /users: crea un nuevo usuario;

- GET: /users/123: devuelve los detalles del usuario 123;
- HEAD: /users/123: muestra la información general del usuario 123;
- PATCH: /users/123 y PUT /users/123: actualiza el usuario 123;
- DELETE: /users/123: elimina el usuario 123;
- OPTIONS: /users: muestra los verbos compatibles respecto al punto final /users;
- OPTIONS: /users/123: muestra los verbos compatibles respecto al punto final /users/123.

Es importante saber que Yii2 tiende a pluralizar (añadir una -s) de manera automática los controladores para usarlo en los puntos finales.

En los siguientes sub-apartados se hace una descripción de cada uno de los servicios REST que van a ser usados. En la figura 5-6 se puede apreciar un listado en el que aparecen los mismos.

5.3.1 Registro EVA

Tabla 5-3. Servicio “Registro EVA”

| Identificativo del servicio | <i>REGISTRO EVA</i> |
|-----------------------------|---|
| Descripción breve | Mediante este servicio se registra el nivel de dolor de un paciente en un determinado día |
| URL | http://localhost/historicos |
| Método | POST |
| Parámetros de entrada | <ul style="list-style-type: none"> • NID (identificador único de paciente) • Valor (nivel de dolor del paciente) • Fecha (fecha actual) |
| Formato respuesta | JSON |
| Notas | Se añade una –s al final de la URL debido a que Yii pluraliza los controladores. Este servicio es fundamental, pues al final uno de los objetivos principales que se quiere conseguir es llevar un seguimiento de los niveles de dolor durante un determinado tiempo. |

5.3.2 Registra Ejercicio

Tabla 5-4. Servicio “Registra Ejercicio”

| Identificativo del servicio | <i>REGISTRA EJERCICIO</i> |
|-----------------------------|---|
| Descripción breve | Gracias a este servicio, la respuesta a |
| URL | http://localhost/variables |
| Método | POST |
| Parámetros de entrada | <ul style="list-style-type: none"> • NID (identificador único de paciente) • Valor (nivel de dolor del paciente) • Fecha (fecha actual) |
| Formato respuesta | JSON |
| Notas | Se añade una –s al final de la URL debido a que Yii pluraliza los controladores. Este servicio es fundamental, pues al final uno de los objetivos principales que se quiere conseguir es llevar un seguimiento de los niveles de dolor durante un determinado tiempo. |

5.3.3 Consulta nombre

Tabla 5-5. Servicio “Consulta nombre”

| Identificativo del servicio | <i>CONSULTA NOMBRE</i> |
|-----------------------------|---|
| Descripción breve | Con este servicio se puede obtener el nombre del paciente para que sea mostrado en el layout MainActivity.xml |
| URL | http://localhost/pacientes/ |
| Método | GET |
| Parámetros de entrada | <ul style="list-style-type: none"> • NID (identificador único de usuario) |
| Formato respuesta | JSON |
| Notas | Únicamente se obtiene el nombre de pila del Paciente |

5.3.4 Consulta histórico

Tabla 5-6. Servicio “Consulta histórico”

| Identificativo del servicio | <i>CONSULTA HISTÓRICO</i> |
|-----------------------------|---|
| Descripción breve | <p>Mediante este servicio se obtienen todos los registros de dolor de un determinado paciente en un determinado mes. Con estos datos la aplicación móvil crea dos gráficas:</p> <ul style="list-style-type: none"> - Una con todos los niveles de dolor registrados en ese mes. - Otra con el tanto por ciento que representa cada grado de dolor |
| URL | http://localhost/historico/ |
| Método | GET |
| Parámetros de entrada | <ul style="list-style-type: none"> • NID (identificador único de paciente) • Mes (mes para el que se quiere obtener el histórico de dolor) |
| Formato respuesta | JSON |
| Notas | Este servicio es fundamental, puesto que cumple con el objetivo principal del trabajo que es que el paciente lleve un control (seguimiento) sobre la evolución de su dolor |

5.3.5 Consulta ejercicio

Tabla 5-7. Servicio “Consulta ejercicio”

| Identificativo del servicio | <i>CONSULTA EJERCICIO</i> |
|-----------------------------|---|
| Descripción breve | Dado que el paciente puede tener asignado un tipo de entrenamiento en función de su estado actual y del cumplimiento del mismo, se debe consultar el entrenamiento actual del mismo. Se devuelve una serie de URL que son mostradas en la aplicación móvil gracias a una Recycled View. |
| URL | http://localhost/ejercicio/ |
| Método | GET |
| Parámetros de entrada | <ul style="list-style-type: none"> Nivel (dificultad de los ejercicios recomendados) |
| Formato respuesta | JSON |
| Notas | Para conocer con más exactitud cuáles son los grados de dificultad que se han tenido en cuenta, consultad el apartado 5.1. |

5.3.6 Consulta consejos

Tabla 5-8. Servicio “Consulta consejos”

| Identificativo del servicio | <i>CONSULTA CONSEJOS</i> |
|-----------------------------|---|
| Descripción breve | Mediante este servicio se pueden consultar los consejos disponibles en la base de datos “apache”. |
| URL | http://localhost/ consejos |
| Método | GET |
| Parámetros de entrada | - |
| Formato respuesta | JSON |
| Notas | Los consejos se muestran de manera aleatoria cada vez que se pincha en el botón “Consejos” del menú inicial de la aplicación móvil. |

5.3.7 Consulta medicación

Tabla 5-9. Servicio “Consulta medicación”

| Identificativo del servicio | <i>CONSULTA MEDICACIÓN</i> |
|-----------------------------|---|
| Descripción breve | Dado que cada paciente tiene asignado tres tipos de medicación (apartado 5.1), en función de la que en ese momento tenga activa, se debe mostrar la más restrictiva (más potente). El orden de restricción es de mayor a menor, pauta rescate, pauta analgésica y básica. |
| URL | http://localhost/medicacion/ |
| Método | GET |
| Parámetros de entrada | <ul style="list-style-type: none"> • NID (identificador único de paciente) • Tipo (número que identifica al tipo de medicación) |
| Formato respuesta | JSON |
| Notas | Dicha medicación que se consulta está compuesta por el nombre del medicamento, cantidad y número de tomas por día. |

5.3.8 Comprueba variables

Tabla 5-10. Servicio “Comprueba variables”

| Identificativo del servicio | <i>COMPRUEBA VARIABLES</i> |
|-----------------------------|---|
| Descripción breve | Mediante este servicio, se puede acceder al valor de una determinada variable para que el algoritmo de la aplicación móvil tome la decisión competente en el momento adecuado |
| URL | http://localhost/variable/ |
| Método | GET |
| Parámetros de entrada | <ul style="list-style-type: none"> • NID (identificador único de paciente) • Num_variable (número que representa a la variable consultada) |
| Formato respuesta | JSON |
| Notas | Para ver el listado de variables disponibles recurrir al apartado 5.1 |

5.3.9 Actualiza variables

Tabla 5-11. Servicio “Actualiza variables”

| Identificativo del servicio | <i>ACTUALIZA VARIABLES</i> |
|-----------------------------|--|
| Descripción breve | Con este servicio se puede actualizar el valor de alguna de las variables que han sido definidas en la base de datos cuando se produce un evento de interés |
| URL | http://localhost/variables/NID,num_variable |
| Método | PUT |
| Parámetros de entrada | <ul style="list-style-type: none"> • NID (identificador único de paciente). • Num_variable (número correspondiente a la variable cuyo contenido se va a actualizar). • Valor (valor que se va introducir en el campo) |
| Formato respuesta | JSON |
| Notas | El hecho de necesitar dos parámetros en la URL para actualizar dicha variable es debido a que la tabla que se quiere actualizar contiene dos claves primarias. |

5.3.10 Mensaje formativo

Tabla 5-12. Servicio “Mensaje formativo”

| Identificativo del servicio | <i>MENSAJE FORMATIVO</i> |
|-----------------------------|--|
| Descripción breve | Mediante este servicio se consulta un determinado mensaje formativo a fin de ser mostrado por pantalla al paciente. |
| URL | http://localhost/mensajefor/ |
| Método | GET |
| Parámetros de entrada | - |
| Formato respuesta | JSON |
| Notas | El mensaje formativo está compuesto por un mensaje escrito y la URL de una determinada foto acorde al mensaje que se muestra |

5.3.11 Mensaje informativo

Tabla 5-13. Servicio “Mensaje informativo”

| Identificativo del servicio | <i>MENSAJE INFORMATIVO</i> |
|-----------------------------|--|
| Descripción breve | Mediante este servicio se consulta un determinado mensaje informativo a fin de ser mostrado por pantalla al paciente. |
| URL | http://localhost/mensajeinfor/ |
| Método | GET |
| Parámetros de entrada | - |
| Formato respuesta | JSON |
| Notas | El mensaje informativo está compuesto por un mensaje escrito y la URL de una determinada foto acorde al mensaje que se muestra |

5.4 Aplicación WEB

Para culminar con todo el conglomerado de todo lo que esta aplicación permite hacer, se ha creado una aplicación web. La misma está pensada para que el médico pueda llevar un control de todos los datos que la aplicación móvil va recopilando. Para esto, se hace uso de nuevo de Yii2, pues permite de manera fácil crear la misma sin necesidad de un complejo código.

En la siguiente imagen se puede apreciar cual es la apariencia de la página principal.

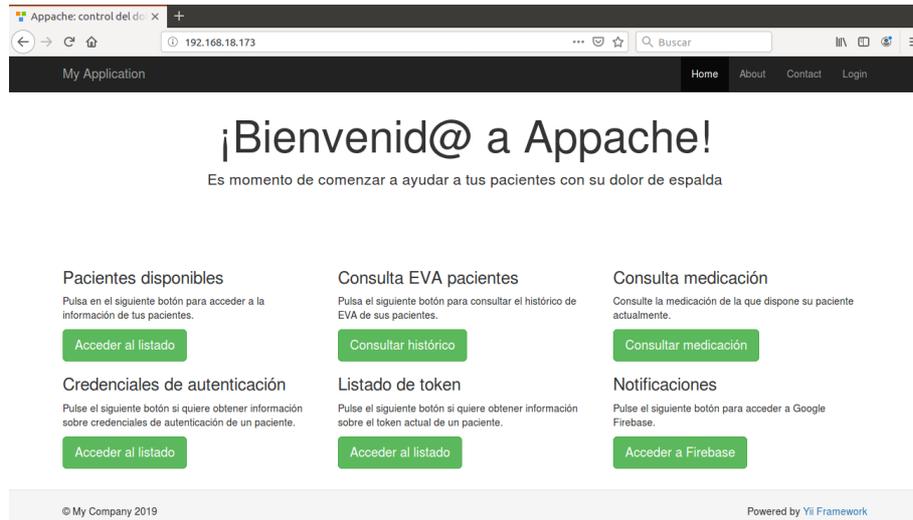


Ilustración 5-25: Apariencia de la pantalla inicial

Se puede apreciar que se aparecen 6 botones, cada uno de ellos llevan a una vista que contienen una funcionalidad de importancia para el médico. No se han incluido todas las vistas que se han llegado a crear, con la finalidad de que el médico sólo acceda a lo necesario y de esa manera pueda ahorrar tiempo.

Las funcionalidades disponibles son las siguientes:

- Listado de pacientes: el médico puede acceder a ver toda la información de los pacientes que actualmente están a su cargo.

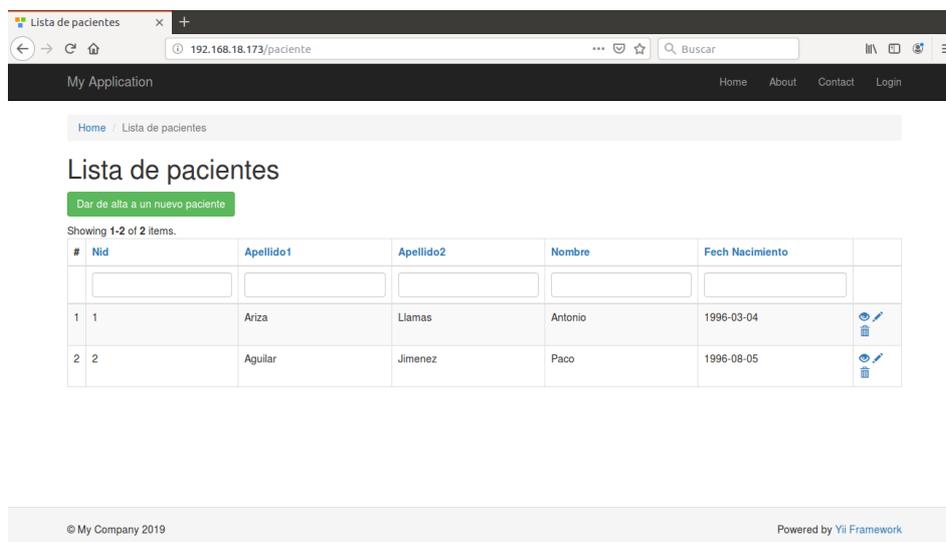


Ilustración 5-26: Lista de pacientes

- Consulta EVA: con el fin de llevar un control del histórico de dolor de cada uno de sus pacientes.

| # | Nid | Valor | Fecha | |
|---|-----|-------|------------|--|
| 1 | 1 | 1 | 2019-06-02 | |
| 2 | 1 | 4 | 2019-06-04 | |
| 3 | 1 | 4 | 2019-06-11 | |
| 4 | 1 | 8 | 2019-06-13 | |
| 5 | 1 | 7 | 2019-06-15 | |
| 6 | 1 | 2 | 2019-07-01 | |
| 7 | 1 | 2 | 2019-07-02 | |
| 8 | 1 | 6 | 2019-08-12 | |

Ilustración 5-27: Histórico de dolor

- Consultar medicación: dado que la medicación de un determinado paciente puede ir cambiando, el médico puede acceder a ver la medicación actual del mismo o recetar una nueva si es necesario.

| # | Nid | Nombre | Tipo | Cantidad | Horario | |
|---|-----|--------------|------|----------|---------|--|
| 1 | 1 | Ibuprofeno | 1 | 700mgr | 3 | |
| 2 | 1 | Ibuprofeno+ | 2 | 1.5gr | 3 | |
| 3 | 1 | Ibuprofeno++ | 3 | 1gr | 3 | |
| 4 | 1 | Tramadol | 1 | 1gr | 3 | |
| 5 | 1 | Tramadol+ | 2 | 1gr | 3 | |
| 6 | 1 | Tramadol++ | 3 | 1gr | 3 | |

Ilustración 5-28: Medicación de los pacientes

- Credenciales de autenticación: para ver todas las personas que se han registrado en el sistema, el doctor puede acceder a ver las mismas.

My Application Home About Contact Login

Home / Credenciales de autenticación

Credenciales de autenticación

Crear credenciales

Showing 1-1 of 1 item.

| # | Nid | Clave | Telefono | Mac | |
|---|-----|------------|-----------|-------------------|--|
| 1 | 1 | 123456789A | 636554267 | 2C:59:8A:40:DD:82 | |

Ilustración 5-29: Credenciales de autenticación

- Listado de token: como se explicó en el capítulo 3, un token es un identificador único que google da a cada dispositivo por utilizar sus servicios. Dado que este se puede ir actualizando, el médico necesita ese token cuando quiere mandar una notificación personal a dicha persona.

My Application Home About Contact Login

Home / Listado de tokens

Listado de tokens

Crear registro

Showing 1-1 of 1 item.

| # | Nid | Token | |
|---|-----|--|--|
| 1 | 1 | faQFoi4Kqjw:APA91bHv9D26tvPzhyoL73dw9ys00QsjUPufHQQVUoMVOreHGptsFBINAxakeq3XCbF7SchLyIV-O9LKTyr4sHtV9eFqtSsOOJPNxHvhfeQdNvNblvSWgEfx6Nn3PE7dRsLzuiqfsor2 | |

Ilustración 5-30: Listado de tokens

- **Notificaciones:** está relacionada con la funcionalidad anterior. Es un enlace a Firebase, donde el médico va a poder hacer uso de la herramienta Cloud Messaging, para mandar notificaciones a un paciente o grupos de pacientes. En las siguientes ilustraciones se puede ver cuál es el aspecto del mismo.

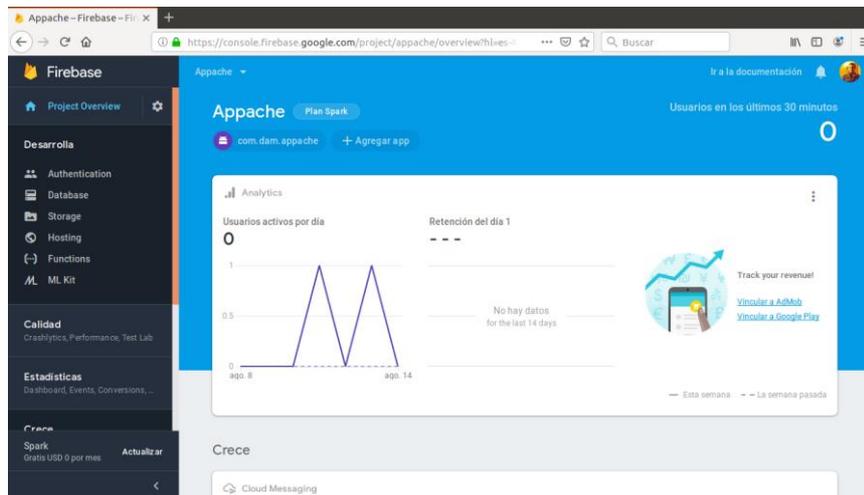


Ilustración 5-31: Aspecto general de Firebase

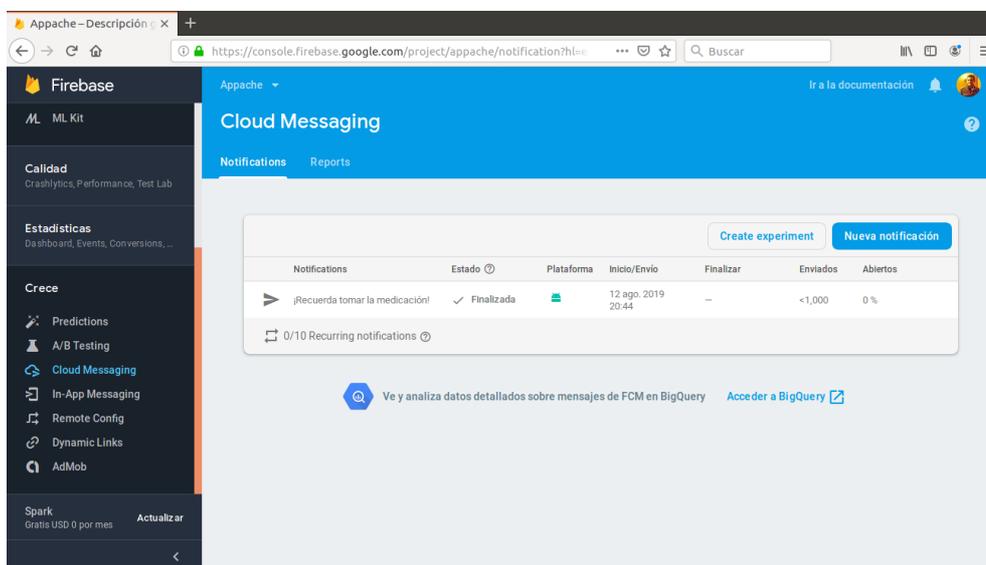


Ilustración 5-32: Herramienta Cloud Messaging

1 Notificación

Título de la notificación (solo para Android y watchOS) ⓘ

Texto de la notificación

Imagen de notificación (opcional) ⓘ

Ejemplo: <https://tuapp.com/imagen.png> 

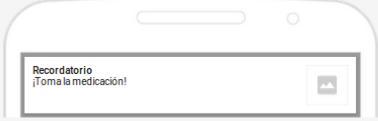
Nombre de la notificación (opcional) ⓘ

Vista previa del dispositivo

En esta vista previa, se ofrece una idea general de cómo se mostrará tu mensaje en un dispositivo móvil. La apariencia real del mensaje varía en función del dispositivo. Para obtener resultados precisos, prueba con un dispositivo real.

[Enviar mensaje de prueba](#)

[Estado inicial](#) [Vista expandida](#)



Recordatorio
¡Toma la medicación!

Android

Ilustración 5-33: Ejemplo de notificación

REFERENCIAS

- [1] *Artículo sobre el dolor lumbar*, <https://www.elsevier.es/es-revista-medicina-familia-semergen-40-articulo-el-dolor-lumbar-13025464>
- [2] *Ventajas de la salud móvil*, <https://www.sivsa.com/site/salud-movil-mhealth-ventajas/>
- [3] *¿Qué es mhealth?*, <http://blog.diagnostrum.com/2015/08/21/que-es-mhealth-2/>
- [4] *Página principal de Android Studio*, <https://developer.android.com/studio/intro?hl=es-419>
- [5] *Información sobre MySQL*, <https://neoattack.com/neowiki/mysql/>
- [6] *Introducción a Yii2*, <https://www.yiiframework.com/doc/guide/2.0/es/intro-yii>
- [7] *Ventajas de API REST*, <https://bbvaopen4u.com/es/actualidad/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos>
- [8] *Página principal de Firebase*, <https://firebase.google.com/?hl=es-419>
- [9] *Configuración de base de datos en Yii2*, <https://www.yiiframework.com/doc/guide/2.0/es/start-databases>
- [10] *Generador de código en Yii2*, <https://www.yiiframework.com/doc/guide/2.0/es/start-gii>
- [11] *Servicio RESTFUL en Yii2*, <https://www.yiiframework.com/doc/guide/2.0/es/rest-quick-start>

