UNIVERSITAT POLITÈCNICA DE VALÈNCIA

etsinf

Escola Tècnica Superior d'Enginyeria Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Diseño e implementación de una aplicación multiplataforma para la presentación y seguimiento de programas médicos

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

Autor: Alejandro Melcón Álvarez

Tutor: Jorge Hortelano Otero
David de Andrés Martínez

Curso 2016-2017

Resum

El present projecte exposa l'anàlisi, el disseny i la implementació d'una solució software per a centres mèdics. Aquesta atorga als centres els medis per a millorar la comunicació i realitzar el seguiment del progrés dels seus pacients. Els pacients fan ús d'aquest software per a proporcionar la informació de seguiment d'un programa del centre al qual estan subscrits i per a visualitzar-lo.

Es du a terme en col·laboració amb l'empresa de desenvolupament de software BiiT Sourcing solutions mitjançant la implementació d'una aplicació multiplataforma. Aquesta compta amb una sèrie de servicis i sistemes associats també explicats en aquesta memòria.

Paraules clau: aplicació, multiplataforma, empresa, centre mèdic

Resumen

El presente proyecto expone el análisis, diseño e implementación de una solución software para centros médicos. Esta otorga a los centros los medios para mejorar la comunicación y realizar el seguimiento del progreso de sus pacientes. Los pacientes hacen uso de este software para proporcionar la información de seguimiento de un programa del centro al que están suscritos y para visualizarlo.

Se lleva a cabo en colaboración con la empresa de desarrollo de software BiiT Sourcing Solutions mediante la implementación de una aplicación multiplataforma. Esta cuenta con una serie de servicios y sistemas asociados también explicados en esta memoria.

Palabras clave: aplicación, multiplataforma, empresa, centro médico

Abstract

The present project presents the analysis, design and implementation of a software solution for medical centres. It provides the centres the means to improve the communication and carry out the monitoring of their patients progress. The patients make use of this software to provide their tracking information of a program from the centre which they are subscribed to. Also, for visualizing this information.

It is carried out in collaboration with the software development company BiiT Sourcing Solutions by the implementation of a multi platform application. The application also has a series of associated systems and services which are explained in this memory.

Key words: application, multiplatform, enterprise, medical center

Índice general

CAPÍTULO 1
Introducción

La salud forma parte de todos los seres vivos, se define como el estado en que un ser orgánico ejerce todas sus funciones con normalidad, o el conjunto de condiciones físicas en las que un organismo se encuentra en un momento determinado [1]. La salud es aquello con lo que contamos por el hecho de estar vivos y por lo tanto es nuestra responsabilidad cuidar la propia y la de aquellos que nos rodean.

A la hora de solucionar un problema, no hay mejor manera que el prevenir su aparición. Esto se consigue, por un lado, estudiando sus causas y, por otro, reconociendo estas mismas en nuestro entorno y nuestro organismo. Así, somos capaces de paliar o eliminar los problemas antes de que se manifiesten. El estudio de estas causas y sus efectos, junto con la aplicación práctica de estos conocimientos, es el objetivo de las ciencias de la salud, las cuales han desarrollado gran cantidad de métodos y pruebas a la hora de analizar las condiciones de un organismo para detectar así indicios de la aparición de problemas.

Motivación

La forma más común de realizar el seguimiento de la salud, es la realización de reconocimientos médicos periódicos [2]. Un reconocimiento médico es un examen sobre la condición física y/o psicológica de una persona realizado con la intención de prevenir y detectar posibles amenazas para su salud. Normalmente se realizan concertando una cita previa con la clínica o el centro donde vaya a llevarse a cabo. Los reconocimientos se separan en diferentes examinaciones y pruebas específicas que deben ser llevadas a cabo por personal especializado en la materia.

Estos reconocimientos, requieren de una gran cantidad de tiempo para su realización ya que obligan al usuario a presentarse en el centro médico y llevar a cabo todos los exámenes que sean necesarios. Dado que este proceso requiere tanto tiempo y es por lo tanto muy costoso, no es usual el llevar un seguimiento continuado de la progresión de cada individuo. La interacción entre doctor y paciente acaba con la finalización de la cita. Tras ella, los especialistas proporcionan una serie de consejos y medidas a tomar para corregir las deficiencias en el estilo de vida que puedan tener los pacientes.

La cantidad y frecuencia con la que se practica ejercicio, la alimentación o los hábitos de sueño son factores que hay que cuidar [3]. Y pese a ser conscientes de la importancia que tienen, quizá por falta de esfuerzo o por falta de herramientas, no todos los pacientes son capaces de actuar de manera consecuente con sus circunstancias. Se desvinculan de los consejos proporcionados ya que no tienen maneras de continuar en contacto con su situación. Así se perpetúan los viejos hábitos que pueden derivar en un deterioro de la calidad de vida.

Con todo esto en mente, surge la idea de mejorar el repertorio de herramientas de cada individuo, de dotarlo de un sistema que recuerde y refuerce las ventajas que aporta el esfuerzo continuo que requiere el cuidar de su salud.

Problemática

El modelo actual de realizar un reconocimiento y recibir unos resultados y recomendaciones vemos que no es perfecto. Carece de implicación por parte de los pacientes y por lo tanto de su voluntad de llevar a cabo los cambios necesarios para mejorar su salud. Algunos de los problemas que presenta este modelo son:

- La localidad temporal del evento, no se distribuye ni se sigue el progreso de manera continua, sino que se realizan mediciones puntuales.
- La información proporcionada resulta muchas veces confusa para una persona carente de conocimientos técnicos.
- No se genera un programa basado en objetivos y los métodos para conseguirlos y por lo tanto se pierde el sentido del por qué.
- La comunicación entre paciente y doctor tras la finalización de la cita es escasa si no nula.

Todo esto nos deja un paciente que durante la mayor parte del año no tiene una fuerte concienciación acerca de su estado de salud y cómo la desarrolla.

Objetivos

Queremos conseguir la implicación de los pacientes a la hora de desarrollar hábitos saludables, que sean capaces de adquirir estos hábitos y sustituir los viejos sin tener que invertir una cantidad de esfuerzo, tiempo o dinero desmesurados. Para ello se propone la creación de un sistema que permita acortar las distancias entre los conocimientos de los profesionales médicos y las necesidades de sus pacientes. Que proporcione a los pacientes razones y objetivos para trabajar todos los días en mejorar su salud. Un sistema al que cualquier paciente de una clínica sea capaz de acceder sin complicaciones extras y con el cual se pueda motivar el desarrollo de cada individuo de manera personalizada. Este sistema debe estar disponible para cualquier usuario y tiene que poder acceder a él sin supervisión.

Estructura de la memoria

La memoria del trabajo se estructura como sigue:

- Comenzaremos haciendo una explicación del contexto en el que se desarrolla este proyecto, el entorno y la tecnología de la que disponemos.
- A continuación haremos un desarrollo de la arquitectura del sistema incluyendo la especificación de los requisitos del software, las herramientas y tecnologías con las que vamos a trabajar y el diseño inicial del sistema que vamos a implementar.
- Proseguiremos con los detalles de la implementación del sistema.

- Explicaremos los mecanismos de verificación utilizados para asegurar la integridad y robustez del software desarrollado.
- Como conclusión haremos un análisis del cumplimiento de los objetivos planteados en esta introducción.
- Para finalizar, indicaremos la dirección que se pretende tomar a la hora de realizar mejoras sobre el sistema.

CAPÍTULO 2
Contexto

La empresa

Este trabajo se realiza en colaboración con la empresa BiiT Sourcing Solutions (BiiT) como continuación a una estancia de prácticas de empresa que ha finalizado en un contrato laboral.

BiiT es una empresa que proporciona diseño, desarrollo y soporte de aplicaciones y servicios software. En el momento en el que comienza a desarrollarse este trabajo, la empresa cuenta con un sistema de gestión para centros médicos, Sport Medi Score (SMS), que facilita la gestión del personal del centro así como la lógica de negocio del mismo. Actualmente el sistema se encuentra desplegado para el centro Orbis Sport en Sittard-Geleen (Países Bajos). Se utiliza también para el registro de datos de los pacientes a la hora de realizar un examen y para generar informes a partir de estos y las reglas de conocimiento diseñadas para proporcionar la información más relevante.

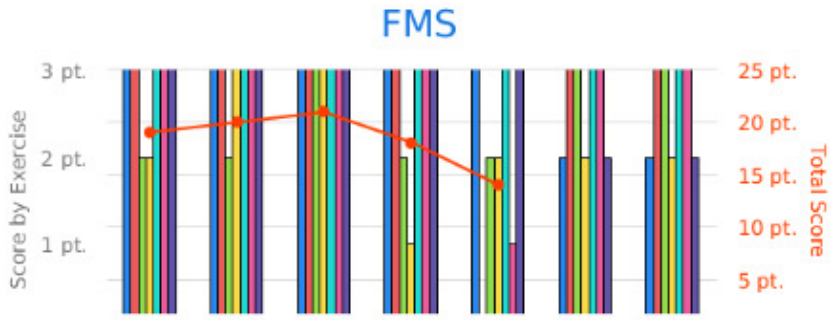
Este sistema está basado en el modelo estándar de gestión similar al expuesto en la introducción. Se crea una cita y al finalizarla, se entrega un informe al usuario. Como podemos ver en la Figura 2.1, la información ofrecida por el informe generado es compleja y no aporta demasiado valor para un usuario sin cualificación médica ni comprensión de las métricas establecidas para generarlo. Junto con este informe se adjuntan unas conclusiones escritas donde el especialista puede hacer recomendaciones a los pacientes. Pero no existe una manera estandarizada de generar este tipo de sugerencias ni una manera de realizar un seguimiento de su progreso. Además, estas recomendaciones no se proporcionan como medio para conseguir un objetivo sino como un objetivo en si mismo haciendo desaparecer el sentido del porqué para los pacientes. No existen canales extra para llevar a cabo la comunicación entre doctores y pacientes.

Queremos mejorar el sistema de generación de reportes y ofrecer más opciones a doctores y pacientes, una mayor flexibilidad a la hora de comunicarse y motivarse. Por eso se plantea la creación de una aplicación móvil accesible para todos los pacientes que se sincronice con el sistema SMS, obtenga los datos para generar los informes, permita hacer un seguimiento de las recomendaciones y ejercicios propuestos, recibir mensajes de los doctores, mostrar los objetivos a alcanzar y el progreso realizado. Esto requerirá también la modificación del sistema actual para incluir todas estas nuevas características centradas en los los clientes.

Antropometrie

Lengte	180.0 cm
Gewicht	85.0 kg
Som 4 huidplooien	40.0 mm
Tailleomtrek	80.0 cm
Bloeddruk	120 / 75 mmHg

Functional Movement Screen



Opmerkingen

Functional Movement Screen

Test		Raw Score	Final Score	Comments
Deep Squat		3	3	
Hurdle Step	**L**	3	3	
	R	3		
Inline Lunge	**L**	2	2	
	R	2		
Shoulder Mobility	**L**	3	3	
	R	3		
Impingement Clearing Test	**L**	pass		
	R	pass		
Active Straight-leg Raise	**L**	2	2	
	R	2		
Trunk Stability Pushup		2	2	
Press-up Clearing Test		pass		
Rotary Stability	**L**	3	3	
	R	3		
Posterior rocking Clearing Test		pass		
Total		18		

Figura 2.1: Sección de informe médico (en neerlandés)

Sistemas móviles y salud

Para crear una aplicación que cubra todas nuestras necesidades tenemos primero que mirar la oferta existente en el mercado. Debido a que todos nos podemos beneficiar de los avances en la salud, han surgido una variedad de aplicaciones móviles que proporcionan desde entrenadores personales, hasta consultas médicas por video llamada para realizar diagnósticos rápidos. En esta sección vamos a hacer una lista de algunas de ellas y a comparar sus características para hacernos una idea de las funcionalidades deseables para el sistema que vamos a diseñar.

Health Tap

Proporciona consejos de salud, consulta con profesionales mediante mensajería o videoconferencia y listas de tareas sugeridas por doctores afiliados a la aplicación.

Apple Health

Recopila información de otras aplicaciones de salud para hacer un sumario, proporcionar estadísticas y consejos de salud.

Samsung Health

Incluye un sistema de desafíos entre usuarios junto con artículos de salud, seguimiento de los ejercicios realizados y sincronización con dispositivos werables.

Google Fit

Como el Apple Health, reúne los datos de distintas aplicaciones, permite planear objetivos y ver el progreso realizado.

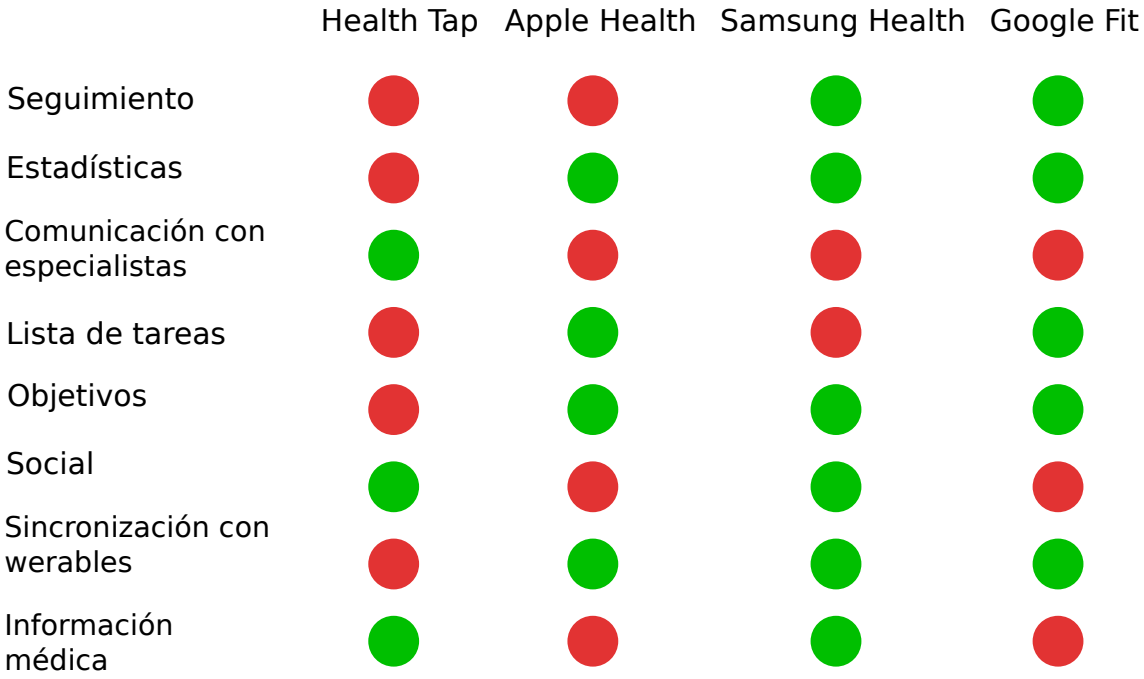


Figura 2.2: Tabla comparativa de funcionalidades

No podemos hacer uso de las aplicaciones existentes ya que ninguna satisface nuestra necesidad de enlazarla con el sistema Sport Medi Score. Y aunque esto fuese posible, no podríamos confiar los datos sensibles de los pacientes a una aplicación externa. Por lo tanto, tendremos que realizar nosotros el desarrollo de la aplicación.

De las características que podemos ver en la Figura 2.2, las más deseables para nuestra aplicación son el seguimiento de los datos proporcionados por el usuario, la generación de estadísticas a partir de estos datos, la posibilidad de ofrecer comunicación con los doctores y especialistas, el planteamiento de unos objetivos a seguir, una lista de tareas para realizar y el aporte de información médica relevante para cada caso concreto.

CAPÍTULO 3
Arquitectura

Siguiendo los objetivos marcados, dividiremos la arquitectura del proyecto en dos sistemas, una aplicación web de gestión y una aplicación móvil para la toma y visualización de datos. Con respecto al sistema de gestión Sport Medi Score habrá que realizar las modificaciones necesarias para que sea compatible con la aplicación móvil.

Especificación de requisitos

Realizaremos una especificación de los requisitos del software en conformidad con el estándar ISO/IEC/IEEE 29148:2011[4].

Propósito

El software a desarrollar tiene como propósito el motivar y mejorar el nivel de implicación de los pacientes a la hora de llevar a cabo un programa médico de seguimiento contínuo para mejorar su salud.

Alcance

Se debe desarrollar una aplicación móvil *Ik Ga voor GezondheidsWinst*, a partir de ahora IGOW e implementar nuevas funcionalidades sobre una aplicación web Sport Medi Score ya existente. IGOW debe ser accesible por cualquier usuario y proporcionar los medios para comunicarse con una clínica médica, realizar tareas programadas y visualizar los resultados obtenidos en los exámenes realizados. SMS debe generar los informes de las clínicas, gestionar la información de los pacientes y facilitar la comunicación entre doctores y pacientes.

Perspectiva del producto

El sistema se estructura en dos aplicaciones interdependientes, además, Sport Medi Score, cuenta con varias dependencias de proyectos anteriores de la empresa.

Funciones del producto

1. **Generación de informes médicos:** Los doctores analizan los resultados de los exámenes y pruebas realizadas y a partir de ellos generan un informe con los datos más relevantes.

2. **Visualización de informes:** La información seleccionada por el personal médico se muestra de manera sencilla a los pacientes facilitando la comprensión de los datos.

3. **Definición de objetivos y ejercicios:** El personal médico marca unos objetivos y define unas actividades a realizar durante los meses siguientes con el fin de alcanzarlos. Estos objetivos se deciden con los pacientes aumentando así su nivel de implicación.

4. **Procurar información sobre ejercicios realizados:** A lo largo de los meses los pacientes registran el desarrollo de las actividades propuestas y expresan la dificultad que les ha supuesto realizarlas.

5. **Visualizar la información obtenida:** Los doctores pueden visualizar esta información y, si lo creen necesario, actuar conforme a ella y comunicarse con los pacientes para ofrecer consejos o sugerencias.

Usuarios

El tipo de usuario vendrá dado por el sistema utilizado. Los usuarios de la aplicación web serán los miembros de la clínica, entre los que se encuentran el personal médico con formación, que se encarga de la toma de datos y la redacción de informes y el personal administrativo que se encarga de la gestión de las citas y el correcto funcionamiento de la clínica. Los usuarios de la aplicación móvil serán pacientes de la clínica, que acudirán para realizarse examinaciones y suscribirse a un programa de salud personalizado, por lo tanto pueden ser usuarios de cualquier tipo.

Restricciones

La aplicación para móviles debe ser desplegable para múltiples sistemas operativos. Queremos llegar a todo el público posible, por esta razón, la aplicación móvil debe estar disponible para Android a partir de la versión 4.4 abarcando así el 91 % de sus usuarios [5].

Con respecto a iOS, debido a la agresiva política de actualización del SO, y la ausencia de compatibilidad hacia atrás, podemos desarrollar para las últimas versiones en este caso a partir de iOS 8.

La aplicación móvil debe ser capaz de enviar y recibir datos de la aplicación web de manera segura mediante el uso de servicios web. Debe también poder utilizarse durante periodos sin conexión, persistiendo los datos obtenidos y luego ser capaz de sincronizarse con la aplicación web.

Al gestionar datos de carácter médico, debemos tener en cuenta el nivel de seguridad necesario para su transmisión y guardado. Además de pedir el consentimiento explícito para su utilización a los usuarios del sistema[6].

El sistema de alta del cliente en la aplicación móvil debe llevarse a cabo de manera segura, para ello debe gestionarse la creación de los usuarios de manera automática.

La comunicación mediante mensajes debe ser unidireccional del doctor al usuario ya que los doctores tienen un horario determinado y no pueden atender todas las peticiones de sus clientes.

Suposiciones y dependencias

Se realizan las siguientes asunciones sin las cuales los requisitos especificados podrían no llegar a cumplirse:

1. Existe un proyecto previo sobre el que crear la parte del servidor y de gestión.
2. Este sistema se puede desplegar en un servidor y sus servicios web son accesibles por cualquier dispositivo con acceso a Internet.
3. Se puede generar una aplicación móvil exportable a múltiples sistemas operativos.

Reparto de requisitos

La aplicación web debe ser capaz de: generar informes, visualizar informes, definir objetivos para cada usuario, definir ejercicios para cada usuario, visualizar el progreso registrado en la aplicación móvil, enviar mensajes a la aplicación móvil.

Por su parte la aplicación móvil debe también visualizar informes, registrar los datos de los ejercicios realizados, visualizar el progreso realizado, recibir y mostrar mensajes enviados desde la aplicación web.

Interfaces

Interfaces con el usuario

- Debe haber una vista por cada funcionalidad principal de la aplicación móvil.
- Para la aplicación web, se generarán nuevas opciones en la ventana de gestión de informes ya existente.

Interfaces con el software

- La aplicación web debe ofrecer de manera segura los datos mediante una estructura JSON.
- La aplicación móvil debe poder recibir mensajes.

Interfaces de comunicación

- La comunicación entre las aplicaciones será iniciada por la aplicación móvil.
- Como caso extraordinario, a la hora de recibir mensajes, la comunicación se iniciará por parte de la aplicación web.
- Todas las comunicaciones de datos críticos, deben ser seguras.
- Se utilizará el protocolo https[21] para realizar la comunicación.

Requisitos funcionales

Estos definen el comportamiento del sistema a implementar. El sistema debe permitir:

IGOW

- Mostrar informes.
- Mostrar tareas realizadas y por realizar.
- Completar tareas.
- Informar del esfuerzo empleado.
- Mostrar información de las tareas.
- Mostrar información del progreso realizado.
- Mostrar mensajes recibidos de la aplicación web.

Sport Medi Score

- Mostrar informes.
- Definir informes.
- Definir objetivos.
- Definir ejercicios a realizar.
- Mostrar el progreso de los clientes.
- Enviar mensajes a los clientes.

Requisitos de usabilidad

La aplicación móvil debe abrirse en menos de 5s.

Requisitos de rendimiento

El sistema debe ser capaz de soportar varios usuarios concurrentes utilizando la aplicación móvil sin afectar al rendimiento. La aplicación móvil no debe bloquear la interacción de los usuarios al realizar comunicaciones con el servidor.

Requisitos lógicos de la base de datos

Con respecto a la base de datos de la aplicación web, construiremos sobre la ya existente, añadiendo nuevas entidades: los objetivos, los ejercicios correctivos, el registro de progreso de los clientes y los mensajes enviados. La base de datos de la aplicación móvil se accederá una vez cuando haya que cargar los datos al iniciar la aplicación, debe mantenerse actualizada a medida que el usuario interactúa con el sistema y sincronizarse cuando se encuentre obsoleta. Contaremos con las entidades para las citas, los informes médicos, las tareas a realizar y los mensajes recibidos.

Atributos del sistema

Fiabilidad

El sistema debe tener una buena fiabilidad que tienda a 1. Aún así, dado que requiere de comunicación constante entre varias aplicaciones, debe hacerse un diseño tolerante a fallos. Se deben contemplar posibles faltas de disponibilidad en los servicios web o en la propia señal. Para asegurar la fiabilidad del sistema deben realizarse tests.

Disponibilidad

Se establecerá un acuerdo de nivel de servicio (SLA) superior al 99 %. Si el servicio no esté disponible se responderá con un estado HTTP 503 (service unavailable).

Seguridad

Se debe utilizar comunicación cifrada para todos aquellos intercambios que contengan información médica o personal.

Se deberá impedir la posible suplantación de identidad de los usuarios.

Portabilidad

La aplicación móvil debe ser desplegable para varios sistemas operativos, como mínimo Android e iOS, y si es posible otros. Por lo tanto debe elegirse para su desarrollo un lenguaje de programación y un entorno adecuados.

Verificación

Para verificar la calidad del software se deben realizar pruebas automáticas sobre los componentes del sistema.

IGOW

Pruebas unitarias para la lógica. Pruebas de integración o end-to-end si es necesario.

Sport Medi Score

Pruebas unitarias para la lógica. Pruebas de integración para mostrar el funcionamiento completo.

Información de soporte

No es necesario añadir información de soporte adicional.

Herramientas y tecnologías

De BiiT

BiiT cuenta con una serie de herramientas que se utilizan a la hora de desarrollar cualquier proyecto y se hará uso de ellas durante la implementación del sistema.

Metodología de trabajo

Los objetivos de la metodología de trabajo que utiliza la empresa son la agilidad y adaptabilidad a la hora de desarrollar los proyectos. Haremos uso de las partes de Scrum que nos son aplicables a la hora de implementar este proyecto. Se realizarán varios sprints para ir implementando las funcionalidades del sistema. Estos, son secciones temporales de entre dos y cuatro semanas durante las cuales se debe desarrollar un producto completamente funcional y desplegable en un entorno de producción para ser presentado al usuario final.

Cada sprint tiene cuatro etapas principales:

- **Reunión de planificación:** Al inicio de cada sprint se planifican las funcionalidades a implementar y los objetivos a conseguir durante el mismo. En esta reunión se hace una estimación del tiempo que va a requerir el desarrollo de cada funcionalidad y cada miembro del equipo escoge la carga de trabajo que crea adecuada.
- **Reuniones diarias:** los miembros del equipo dedican 15 minutos a informar del trabajo realizado hasta el momento, el trabajo que van a realizar y los impedimentos existentes o que puedan existir en el desarrollo de sus tareas. Aquí entra en juego el uso de la herramienta Taiga[7] que nos ayuda a seguir el progreso de las tareas realizadas.
- **Revisión:** al acabar el sprint, el equipo se reúne para demostrar un producto funcional a todos los interesados en el mismo.
- **Retrospectiva:** para finalizar se realiza una retrospectiva para observar los resultados del sprint y anotar los errores cometidos para subsanarlos en futuros sprints.

Control de versiones

Se hará uso de Git como herramienta de control de versiones. El uso de esta tecnología está bien expandido en el ámbito del desarrollo de software y nos ofrece posibilidades para unificar el trabajo de varias personas sin que este sea solapado. Haremos uso de Git mediante la línea de comandos, los más usados son.

- `git pull` incorpora cambios de un repositorio remoto en la rama actual.
- `git add` actualiza el índice de contenido preparado para el siguiente commit.
- `git commit` guarda el contenido actual del índice en la pila de cambios junto con un mensaje descriptivo del usuario.
- `git push` actualiza los repositorios remotos utilizando los cambios locales.

A la hora de implementar una nueva funcionalidad, utilizaremos una rama específica para ella y luego la uniremos a la rama principal de desarrollo.

Gestión del ciclo de vida del software

El ciclo del software que sigue la empresa es el de la entrega continua para lo cual utilizaremos el software Jenkins[8]. Se trata de un software de integración continua que nos permite el automatizado de la compilación, realización de tests y el despliegue de los proyectos desarrollados.

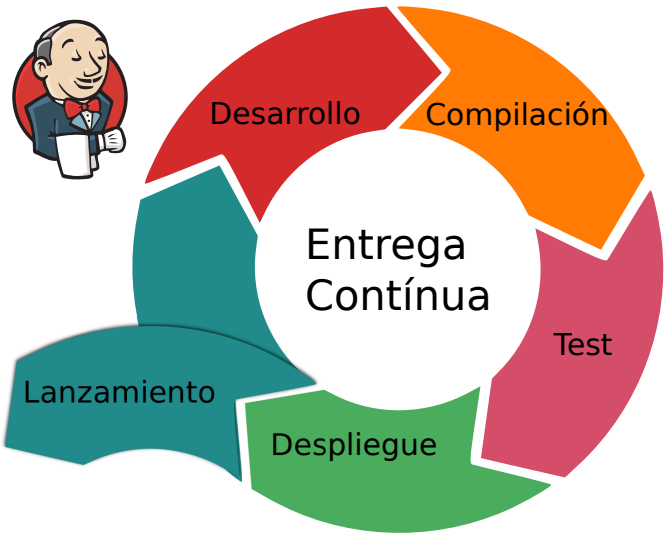


Figura 3.1: Diagrama de entraga contínua

El ciclo de vida será como se muestra en la Figura 3.1, desarrollaremos en la rama de Git `development`, subiremos los cambios realizados al repositorio remoto y Jenkins realizará automáticamente todos los pasos para optimizar el proyecto y asegurar que los tests se completan de manera satisfactoria. Al finalizar el proceso y si todo ha salido bien, tendremos un producto listo para ser desplegado. Este proceso se describe en detalle en el trabajo final de grado de Cristian Todea[25].

Generación de Informes

A la hora de generar los informes médicos y ya que queremos dotarlos de un aspecto más ameno e intuitivo, utilizaremos una mezcla de datos y elementos visuales en forma de infografías que generaremos haciendo uso de la librería Infographic-js. Esta librería ha sido desarrollada por Anna Llorens y su funcionamiento y configuración se explican en su memoria de trabajo de fin de grado[26].

Desarrollo de Sport Medi Score

La aplicación está desarrollada en Java y por lo tanto haremos uso de este lenguaje a la hora de implementar las nuevas funcionalidades del sistema.

Gestión de dependencias

Los proyectos java de la empresa se gestionan haciendo uso de Maven. Este hace uso de un Project Object Model (POM) para describir el software a construir, las dependencias a otros proyectos y componentes, el orden de construcción y los perfiles a utilizar a la hora de construir el proyecto.

Spring

Ofrece un sistema de programación orientada a aspectos. Su punto más fuerte es el contenedor de inversión de control (IoC por sus siglas en inglés) que nos proporciona una forma consistente de configuración y administración de los objetos mediante el uso de la reflexión. Es decir que puede razonar sobre su propia computación, esto se traduce en la inyección de dependencias y la gestión del ciclo de vida de los objetos. Spring nos permite también configurar servicios web mediante anotaciones. Y ejecutar los tests unitarios.

Hibernate y MySql

Se trata de una herramienta de mapeo objeto-relacional(ORM). Mediante anotaciones se mapea el sistema de objetos de java a la base de datos, en este caso MySQL, proporcionando así una solución para la persistencia.

Vaadin

Vaadin es un framework para el desarrollo de interfaces de aplicaciones web. La lógica se ejecuta en el servidor y se comunica al navegador mediante la tecnología Ajax. Esto lo gestiona Vaadin y por lo tanto sólo se tiene que desarrollar en java haciendo uso de los objetos específicos de la librería.

Sistema de conocimiento

Sport Medi Score hace uso de un sistema de conocimiento [27] para obtener información a partir de las examinaciones realizadas y los resultados obtenidos, para ello se utiliza Drools [10], un sistema de inferencia basado en reglas. Se le proporcionan datos de los tests médicos y mediante la ejecución de unas reglas que nos permiten extraer la información más relevante. Además completa esa información con artículos publicados en una base de conocimiento para la cual hacemos uso de Liferay [11].

Testeo

Para la generación de tests unitarios se hace uso del TestNG [12], un framework para realizar tests inspirado en JUnit.

Se usa Vaadin Testbench [13] para implementar los tests unitarios, este ofrece gran cantidad de funcionalidades para simular las interacciones de un usuario sobre una interfaz gráfica. Ya que vamos a realizar los tests mediante un proceso automatizado en un servidor externo, los tests se ejecutarán sobre un navegador con interfaz simulada, PhantomJS [14].

Desarrollo IGOW

En la actualidad imperan varios sistemas operativos para móviles entre los cuales los más utilizados son Android e iOS pero también existen Windows Phone, Firefox OS, BlackBerry OS, etc. El tener tan amplio abanico de sistemas provoca que a la hora de desarrollar una aplicación, se necesite elegir los sistemas operativos para los cuales se realizará la implementación teniendo que desarrollar una aplicación distinta para cada uno de ellos y multiplicando así el esfuerzo de desarrollo de las aplicaciones y por lo tanto su coste. En respuesta a este exceso de posibilidades, han surgido varios sistemas y

entornos que pretenden crear un puente entre los sistemas operativos y el desarrollo de aplicaciones permitiendo la portabilidad entre distintos sistemas a un coste muy bajo o incluso nulo.

Aplicaciones móviles multiplataforma

Podemos encontrar sistemas que permiten el desarrollo en el lenguaje de preferencia del usuario. Pero estos suelen necesitar bastantes ajustes a la hora de cambiar de un sistema a otro y no siempre nos dotan de las herramientas que necesitamos. Por otro lado, con la explosión del desarrollo web y sus tecnologías ha surgido una tendencia a desarrollar aplicaciones web auto-contenidas. Estas aplicaciones se insertan en un navegador implementado en código nativo del dispositivo. Haremos uso de esta tecnología ya que la comunidad alrededor de estos sistemas es mucho más amplia y la familiaridad con las tecnologías de la web nos proporciona una curva de adaptación menor.

Cordova

Apache Cordova [15] es un framework que utiliza las tecnologías de la web como HTML, CSS y JavaScript para desarrollar aplicaciones multiplataforma, esto nos evita la utilización del lenguaje nativo de la plataforma para la cual estemos desarrollando. Las aplicaciones se ejecutan dentro de una capa contenedora desarrollada para cada plataforma y dependen de APIs estándar para acceder a las características específicas de cada dispositivo como los sensores, datos internos, uso de red, etc. Cuenta con gran cantidad de plugins que sirven como interfaz entre la aplicación y los componentes nativos del sistema.

Angular

Angular [16] es una plataforma que proporciona asistencia a los desarrolladores a la hora de construir aplicaciones en JavaScript. Presenta una serie de características que simplifican la implementación de los complejos requisitos de las aplicaciones modernas. Entre ellos el uso de animaciones, la inyección de dependencias o el enlazado de la vista con el modelo de negocio. Angular aporta también una serie de convenciones[20] que sirven de guía a la hora de enfrentarse al desarrollo de una aplicación. Se seguirán estas convenciones de una manera laxa durante la implementación.

TypeScript

A la hora de desarrollar en Angular, utilizaremos TypeScript [17], un lenguaje de programación que nos proporciona un superconjunto de JavaScript y añade tipado estático de datos y un modelo orientado a objetos.Haciendo uso de un compilador podremos traducirlo a JavaScript. Este es un lenguaje libre desarrollado por Miscrosoft.

Ionic

Ionic [18] se cimienta sobre Angular y Cordova para generar a su vez un entorno específico para el desarrollo de aplicaciones móviles. Cuenta con una gran cantidad de componentes implementados que se pueden utilizar a la hora de crear una aplicación, cuenta también con un repertorio de iconos y con una interfaz por línea de comandos (ionic-cli) que nos reduce a unas pocas líneas todas las tareas necesarias para realizar el

empaquetado y la optimización de una aplicación. Proporciona también un entorno de testeo que simula el dispositivo móvil en el navegador sin tener que hacer uso de un dispositivo real o virtualizado. Este entorno se ejecuta mediante la orden `ionic serve`.

Npm

Se trata de un gestor de paquetes de javascript y lo emplearemos para gestionar todas las dependencias de la aplicación móvil. Se declaran las dependencias en el archivo package.json, se ejecuta el comando `npm install` y tras una espera, tendremos todas nuestras dependencias disponibles para ser utilizadas. El gestor de paquetes permite también publicar los productos desarrollados en caso de que se quieran utilizar como dependencias de otros proyectos [19].

Testeo

Para el testeo de la aplicación móvil utilizaremos Jasmine para definir los tests, Y Karma para ejecutarlos.

Diseño

Sport Medi Score

El incremento de las funcionalidades del sistema se centra en la última etapa del reconocimiento médico, la generación del informe y el periodo posterior entre citas. Las funcionalidades se integrarán en las vistas actuales del sistema.

En la vista del informe se sustituirá el informe actual por las infografías generadas, se deben añadir tres nuevos paneles desplegables, uno para diseñar el informe en función de los datos más relevantes, otro para planificar los objetivos para los valores de rendimiento deseados, y otro para definir los ejercicios para cumplir esos objetivos.

En la vista de gestión de los pacientes se podrá visualizar el feedback proporcionado por los usuarios y enviar mensajes urgentes.

IGOW

Casos de uso

Los escenarios en los que se utiliza la aplicación tienen cierto contexto en común y es el hecho de que se ha realizado como mínimo un primer reconocimiento médico y se ha decidido seguir un programa para obtener mejoras en la salud. Se han definido objetivos y ejercicios para alcanzarlos.

- **Registrarse en la aplicación:**
  El usuario se descarga y accede por primera vez a la aplicación, el sistema muestra un formulario de acceso, el usuario lo rellena con los datos necesarios y accede así a la aplicación. Esto solo ocurrirá una vez por instalación.
- **Ver reports:**
  El usuario quiere visualizar los resultados de los reconocimientos realizados. Abre la aplicación, se desplaza a la vista de de los informes y el sistema los muestra cronológicamente del más reciente al más antiguo.

- **Ver tareas:**
  El usuario quiere realizar los ejercicios y tareas que tiene planeadas para el día. Accede a la aplicación para ver la información relativa a estos, selecciona la sección de información de la tarea que le interesa y el sistema muestra cómo realizar el ejercicio o qué precauciones se deben tomar al llevarlo a cabo.

- **Realizar tareas:**
  Una vez sabe como realizar el ejercicio, el usuario lo ejecuta y registra el esfuerzo realizado en la aplicación. Para ello, en la ventana de los ejercicios, selecciona el ejercicio que ha realizado y proporciona un indicador del esfuerzo empleado. Estos datos se sincronizarán después con el servidor Sport Medi Score.

- **Ver estadísticas de progreso:**
  El usuario quiere ver el progreso del trabajo realizado hasta la fecha. Accede a la aplicación móvil y se desplaza a la ventana de estadísticas de progreso, el sistema muestra de forma gráfica los ejercicios realizados y el feedback proporcionado hasta el momento.

- **Leer mensajes:**
  El usuario puede recibir mensajes de los doctores (Notificaciones en la aplicación) y visualizar una lista de ellos en la aplicación.

Estructura del software

Ya que vamos a seguir el modelo de desarrollo propuesto por SCRUM, tendremos que diseñar la aplicación de manera que sea un producto incremental, añadiendo nuevas funcionalidades a medida que se conviertan en prioritarias. Esto significa que debemos independizar en la medida de lo posible todas las capas del software de manera que podamos hacer cambios sobre secciones sin afectar a la estructura completa.

La aplicación en sí no realiza tareas que requiera un alto nivel de procesamiento, es más bien un sistema de gestión de comunicaciones. Recibe y envía datos, y los muestra haciendo uso de la interfaz de usuario y las librerías específicas para ello. Se distinguen tres capas principales:

La vista o capa de presentación, crea representaciones visuales con las que el usuario es capaz de interactuar. Obtiene los datos de la capa de negocio y le transmite las acciones del usuario cuando es necesario. Esta capa se construye mediante Components de Angular que definen bloques de la interfaz de manera independiente. Estos componentes constan de una plantilla en HTML que proporciona la vista, un controlador en TypeScript que gestiona la presentación y el flujo de datos hacia la vista, y una hoja de estilos SCSS que define el acabado visual y las animaciones. También formarán parte de esta capa los componentes que añadan funcionalidad a la vista estos se implementarán como Directives.

La capa del modelo o capa de negocio, lleva a cabo la lógica básica de la aplicación y mantiene el modelo en memoria para su uso por parte de la vista. Gestiona la sincronización de los datos y recibe instrucciones de la capa de presentación. Se construye mediante Providers de Angular, estos son elementos inyectables que mantienen una sola instancia en toda la aplicación.

La capa de datos y servicios ofrece acceso a todas las herramientas externas a la aplicación mediante el uso de objetos que implementan APIs específicas. Dentro de esta capa se encuentran los servicios web, los sistemas de acceso a la persistencia, y todos los servicios requeridos por la aplicación para realizar funciones específicas como la traducción o el acceso a utilidades del sistema. Se implementan también mediante Providers.

Estructura del almacenamiento

La aplicación IGOW hará uso de varios objetos provenientes de Sport Medi Score:

- **Appointments:** representan los reconocimientos realizados.
- **Reports:** representan los resultados de estos reconocimientos.
- **CorrectiveExercises:** son las tareas y ejercicios planeados para realizar para alcanzar los objetivos.
- **PerformedExercises:** Son los ejercicios que se han realizado, con su fecha de realización y su puntuación proporcionada por el usuario.
- **Messages:** Son todos aquellos mensajes que el doctor tenga la necesidad de hacer llegar a sus pacientes de manera urgente.
- **User:** Representa al usuario de la aplicación.

Interfaz Gráfica

Estructuraremos la interfaz en base a las funcionalidades principales de la aplicación. Habrá cuatro páginas y cada una desempeñará una función, el mostrado de informes y la lista de tareas pendientes como se muestra en la Figura 3.2, y el seguimiento del progreso y los mensajes urgentes enviados por los doctores como aparecen en la Figura 3.3.

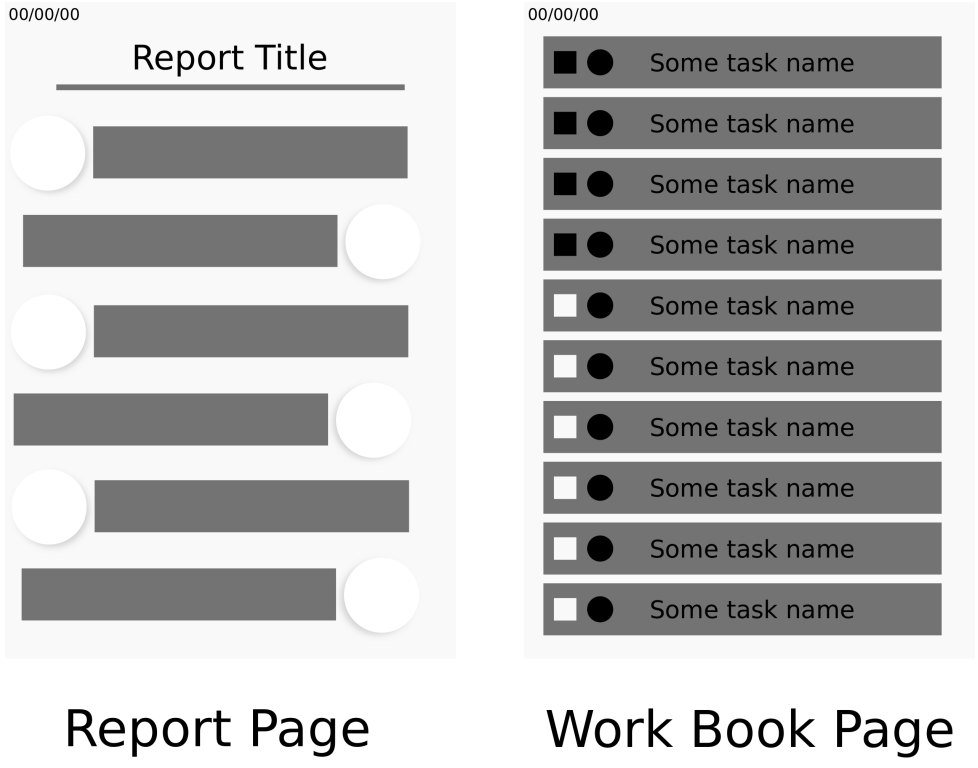


Figura 3.2: Esquema de la interfaz informes y lista de tareas

Figura 3.3: Esquema de la interfaz seguimiento y mensajes

Estructura

A un alto nivel, Sport Medi Score se encarga de recoger los datos iniciales de los usuarios haciendo uso de las examinaciones y más tarde utiliza el sistema de conocimiento para generar información relevante para el centro médico, a partir de esta información se plantean los objetivos y ejercicios a seguir. Por otro lado, IGOW se encarga de la presentación de los datos generados a los usuarios y de recoger su feedback y enviarlo de vuelta al servidor. Mediante la monitorización del feedback proporcionado, los especialistas del centro médico pueden tomar decisiones a la hora de intervenir en el programa propuesto.

En la Figura 3.4 se puede apreciar un diagrama de las relaciones entre las herramientas y los usuarios del sistema. La relación bidireccional entre el centro y el cliente o paciente expresa la interacción humana que se realiza durante todo el proceso desde que se concierta la cita y se realiza el reconocimiento médico, hasta que el paciente deja la clínica. Podemos ver cómo el centro interactúa con Sport Medi Score a la hora de generar las citas e insertar los datos del paciente y de visualizar estos datos.

Sport Medi Score genera unos resultados a partir de las examinaciones realizadas que envía al sistema de conocimiento (Drools) el cual ejecuta las reglas que se hayan definido para seleccionar los datos necesarios de la base de conocimiento(Liferay) y producir los datos que se emplearán para la generación de informes.

El centro toma las decisiones necesarias para crear un report significativo y define toda la información extra que el usuario vaya a necesitar para seguir el programa. Toda esta información se envía a la aplicación móvil IGOW mediante servicios web en formato JSON y se presentará al cliente mediante gráficas e infografías. IGOW porporciona al usuario las opciones para llevar a cabo su seguimiento que será registrado y comunicado a Sport Medi Score para que los doctores puedan acceder a él.

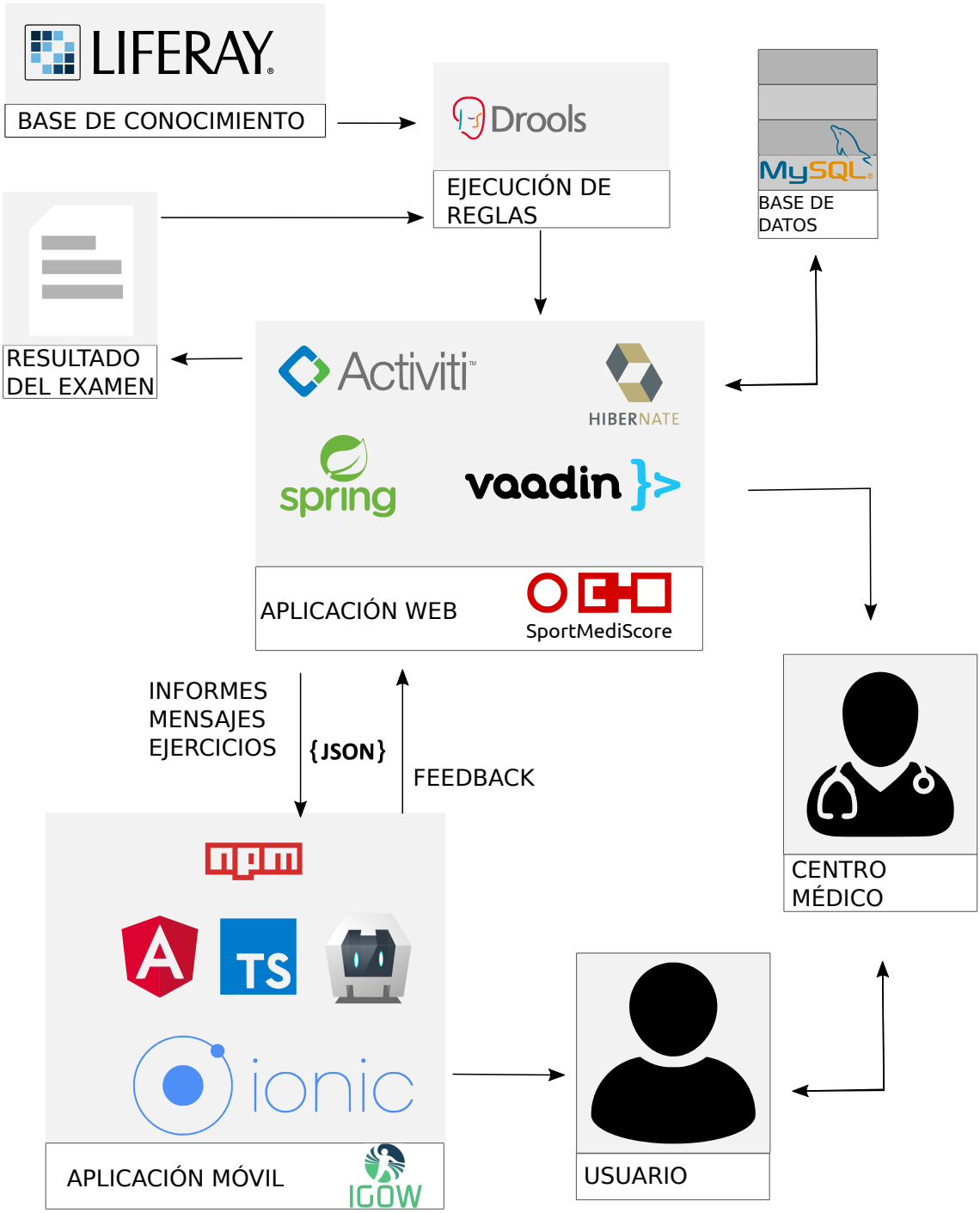


Figura 3.4: Esquema del sistema con sus herramientas

CAPÍTULO 4
Implementación

Siguiendo con el proceso de trabajo de Scrum, hemos dividido el desarrollo de la aplicación en sprints de modo que cada funcionalidad se implemente de manera independiente.

Sprint 1: Creación del proyecto

En este sprint, se planea la familiarización con las nuevas herramientas y la generación del proyecto con sus componentes más básicos.

Formación

Al inicio se realiza una etapa de formación para familiarizarse con las tecnologías de desarrollo de la aplicación móvil. Esto constituye un desafío ya que estas tecnologías no tienen antecedentes de uso en la empresa. Durante esta etapa se llevan a cabo una serie de cursos y pruebas para obtener un conocimiento general de las nuevas tecnologías con las que se va a tratar: Cordova, Angular, TypeScript, Ionic.

Inicio

Finalizado el tiempo de formación se inicia la preparación del proyecto para integrarlo con el sistema de la empresa. Se genera la estructura básica de la aplicación móvil 4.1, se instalan las dependencias iniciales y se comprueba el funcionamiento inicial del sistema.

```
alex@alex-pc:~/workspace/BiiTProjects$ ionic start IGOW

? What starter would you like to use: tabs
✔ Creating directory ./IGOW - done!
[INFO] Fetching app base (https://github.com/ionic-team/ionic2-app-base/archive/master.tar.gz)
✔ Downloading - done!
[INFO] Fetching starter template tabs (https://github.com/ionic-team/ionic2-starter-tabs/archive/master.tar.gz)
✔ Downloading - done!
✔ Updating package.json with app details - done!
✔ Creating configuration file ionic.config.json - done!
```

Figura 4.1: Inicialización de la aplicación con ionic-cli

Integración con el ciclo de BiiT

Comprobado que la versión inicial vacía funciona, se genera un repositorio remoto y se sube el proyecto. Se genera una tarea en Jenkins para la gestión del ciclo del software.

Así se puede empezar con un sistema integrado en el ciclo del despliegue continuo. La configuración de las tareas de Jenkins son competencia del departamento de sistemas de la empresa.

Servicios básicos

Se procede a generar los servicios básicos necesarios para una aplicación de este tipo que serán reutilizables para cualquier otra: servicio de traducción e internacionalización (i19n), servicio de configuración del sistema.

Gestión de usuarios

Para hacer la gestión de usuarios del sistema, primero se rellena un formulario de ingreso con los datos del paciente que va a inscribirse en el programa. Tras el envío de este formulario, Sport Medi Score genera de manera automática un nuevo objeto `Patient` en la base de datos con los datos de la inscripción. El personal del centro médico genera entonces una cita para este usuario.

A la hora de registrar el usuario en la aplicación IGOW, este introduce su número de identificación, después recibe un mensaje con un código de verificación que debe introducir en la aplicación. Después de esta confirmación inicial, no será necesario volver a proporcionar estos datos. El sistema se comunica mediante un token de autenticación[22] generado por el servidor con una clave propia, los datos del usuario, y el número de identificación o uuid de su teléfono móvil.

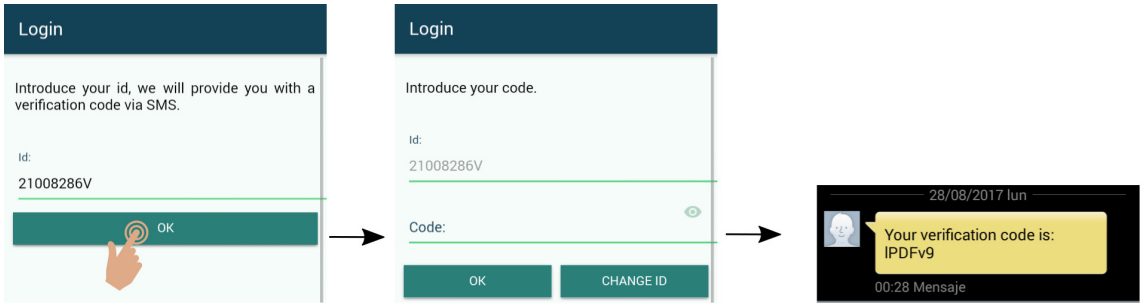


Figura 4.2: Página de login

Sprint 2: Reports

Los reportes son documentos que ofrecen información clave sobre la situación actual de un individuo tras la realización de un reconocimiento médico. Nuestro objetivo es eliminar los reportes antiguos que resultan demasiado técnicos y sustituirlos por unos más dinámicos e intuitivos.

Generación de informes

Para conseguirlo, vamos a transformar los conjuntos de valores obtenidos en los exámenes mediante reglas de conocimiento para obtener los indicadores clave de cada uno. Esto reduce la complejidad del reporte en gran medida ya que convertimos una gran cantidad de datos inconexos en un valor único. En función a estos valores compuestos, podemos marcar objetivos globales para el programa de ejercicios que se va a diseñar.

Los objetivos junto con los valores clave obtenidos y el seguimiento de estos, generan Indicadores Clave de Rendimiento (KPIs) que nos ayudan a visualizar el progreso para un área determinada.

Un ejemplo de KPI es el índice de masa corporal. Para obtenerlo, generamos unas reglas en Drools que procesan los datos obtenidos mediante la examinación de antropometría. Esta examinación contempla los aspectos básicos de la biometría de una persona entre ellos le peso y la altura. Para obtener el indicador aplicamos la fórmula:

$$BMI = peso/altura^2 \tag{4.1}$$

Que como regla de Drools se representa de la forma:

```
rule "CalculateBMI_a0d4079222ba4a4f8b4c4fa5574b4e7b"
when
        $droolsForm: DroolsForm()
        $form1Antropometrie : DroolsSubmittedForm() from $droolsForm.getDroolsSubmittedForm()
        $form1Antropometrie_Category1Formulas : DroolsSubmittedCategory( getText() == 'Formulas') from
$form1Antropometrie.getChildren(ISubmittedCategory.class)
        $form1Antropometrie_Category1Formulas_question0BMI : DroolsSubmittedQuestion( getText() == 'BMI') from
$form1Antropometrie_Category1Formulas.getChildren(ISubmittedQuestion.class)
        $form1Antropometrie_Category0Anthropometry : DroolsSubmittedCategory( getText() == 'Anthropometry') from
$form1Antropometrie.getChildren(ISubmittedCategory.class)
        $form1Antropometrie_Category0Anthropometry_question4Weight : DroolsSubmittedQuestion( getText() == 'Weight') from
$form1Antropometrie_Category0Anthropometry.getChildren(ISubmittedQuestion.class)
        $form1Antropometrie_Category0Anthropometry_question3Height : DroolsSubmittedQuestion( getText() == 'Height') from
$form1Antropometrie_Category0Anthropometry.getChildren(ISubmittedQuestion.class)
then
        $form1Antropometrie_Category1Formulas_question0BMI.setVariableValue('Value', ((Double)
$form1Antropometrie_Category0Anthropometry_question4Weight.getAnswer('NUMBER') / (((Double)
$form1Antropometrie_Category0Anthropometry_question3Height.getAnswer('NUMBER') / 100.0) * ((Double)
$form1Antropometrie_Category0Anthropometry_question3Height.getAnswer('NUMBER') / 100.0))));
end
```

Figura 4.3: Regla para calcular el BMI

Infografías

Los KPIs se representan en el nuevo reporte haciendo uso del sistema de generación de infografías. Para cada uno de los indicadores, mostraremos el valor actual obtenido junto con cierta información explicativa de lo que representa, indicamos también los objetivos marcados para cada indicador y unas gráficas de estimación de la progresión del indicador a lo largo del programa. Por último, incluiremos una conclusión con todos los detalles que el doctor encargado del reconocimiento quiera aportar.

A la hora de mostrar el reporte en la aplicación móvil, creamos una vista que consta de diferentes páginas y permite hacer zoom sobre los elementos expuestos en ellas. Vamos a obtener los informes en formato SVG mediante el uso de la librería Infographic-js. Estos se generan de manera modular. Cada objeto del reporte está definido de manera unitaria y está expresado como una plantilla que describe su estructura, y un contenido, que determina los valores concretos con los que se va a rellenar.

Caché

Ya que la generación de las infografías de los informes médicos es uno de los procesos más pesados de los que se ejecutan en la aplicación móvil, implementamos un sistema de cacheo para mantenerlas en memoria, esto disminuirá el consumo de batería del dispositivo y ayudará con la carga del servidor que no tendrá que servir los informes continuamente. Cada vez que se inicia la aplicación, se consulta a Sport Medi Score con la fecha de modificación de los datos, en caso de estar desactualizados, este envía los nuevos que sobrescribirán los ya guardados.

Figura 4.4: Información extra de un ejercicio (IGOW)

Para esto hacemos uso de un servicio web para la recuperación de datos y un servicio de persistencia. Ya que implementamos este sistema para el cacheo de los informes, lo haremos también para el resto de los datos de la aplicación.

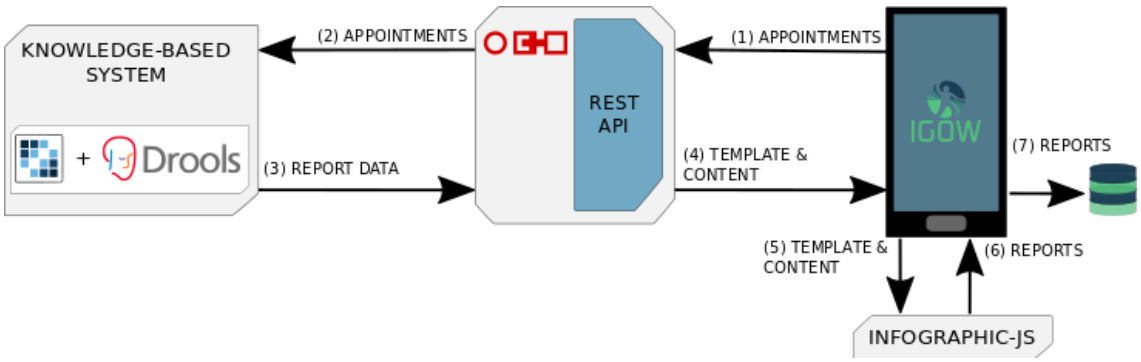


Figura 4.5: Diagrama de flujo de los reports

Figura 4.5: (1) Se envía la ID de los appointments registrados la fecha de modificación registrada. En caso de haberse modificado o existir nuevos appointments, (2) SMS pide al sistema de conocimiento (3) una síntesis de la información para hacer el informe. SMS transforma estos datos en una estructura propia para definir infografías en JSON que contiene una plantilla y el contenido de la misma. (4) Envía la estructura a IGOW. (5) Esta la procesa con la librería Infographic-js generando (6) un report en SVG que se mostrará en la vista. (6) Este report se guardará en la memoria del dispositivo hasta que se modifique en el servidor.

Sprint 3: Workbook

El objetivo de este sprint es implementar las funcionalidades de definición de ejercicios correctivos, la visualización de las explicaciones y el registro de los ejercicios realizados. Esta será la función principal de IGOW ya que el uso más común de la aplicación será la visualización y el registro de los ejercicios.

Ejercicios propuestos

Definiremos, nuevamente en el sistema de conocimiento de la empresa, una serie de reglas para asignar todos los ejercicios que pueden ser recomendados en función a los exámenes médicos realizados. Para simplificar el trabajo del centro médico, en base a los resultados de estos exámenes, se sugerirán unos ejercicios los cuales pueden ser cambiados en el panel de control del doctor a la hora de terminar el informe. Una vez finalizada la cita, los ejercicios se envían a la aplicación del usuario para que pueda comenzara a llevar su seguimiento.

Lista de ejercicios

Para mostrar los ejercicios en IGOW, se genera un objeto de páginas deslizables, un slider, este debe mostrar los ejercicios del día actual y todas las funcionalidades relacionadas con ellos. Para desarrollar este objeto, se ha decidido implementar una interfaz que mantenga cargada la vista de las tareas de un día, el anterior, y el siguiente, se visualiza la vista del día actual y al realizarse un desplazamiento en la pantalla, se inicia una transición en la dirección desplazada. Al finalizar la transición con la vista del día siguiente o anterior visible, se fija el nuevo día actual y se sustituye la vista del siguiente día adyacente.

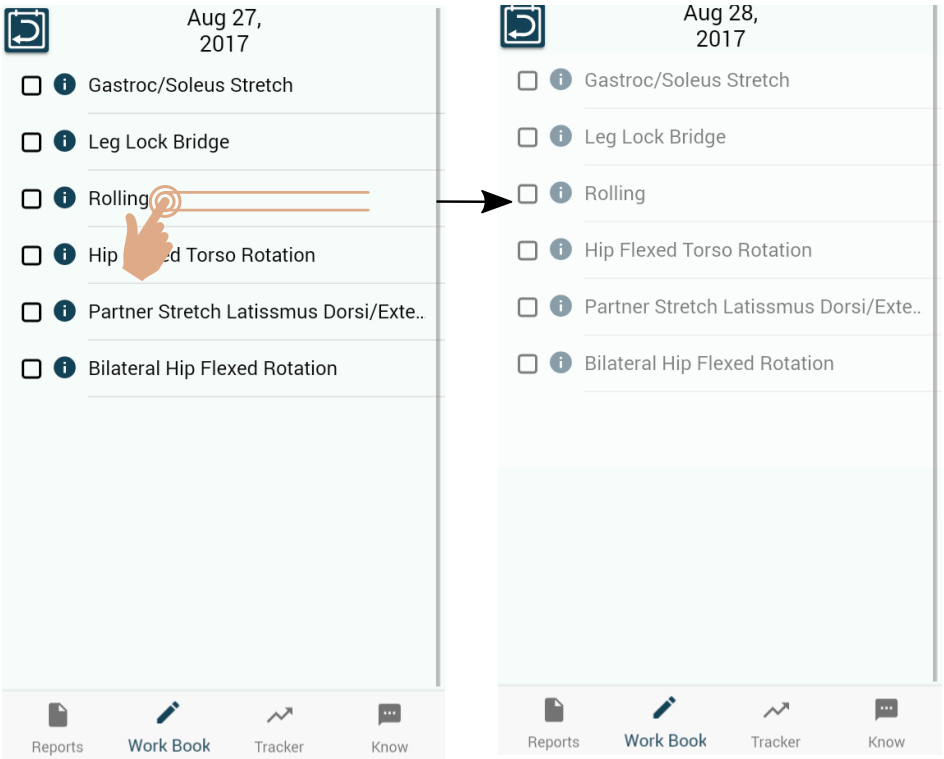


Figura 4.6: Cambio de día en la vista de tareas (IGOW)

Explicación de los ejercicios

Mediante el uso de reglas del sistema de conocimiento, se obtiene la información de los ejercicios que se van a realizar, esta se encuentra en forma de artículo en HTML y está definida en la base de conocimiento, se pueden generar también reglas para incluir vídeos a la sección de información. En IGOW podemos acceder a esta información pulsando la opción de información de cada ejercicio lo que nos lleva a una nueva vista donde se muestra la información recuperada por el servidor.

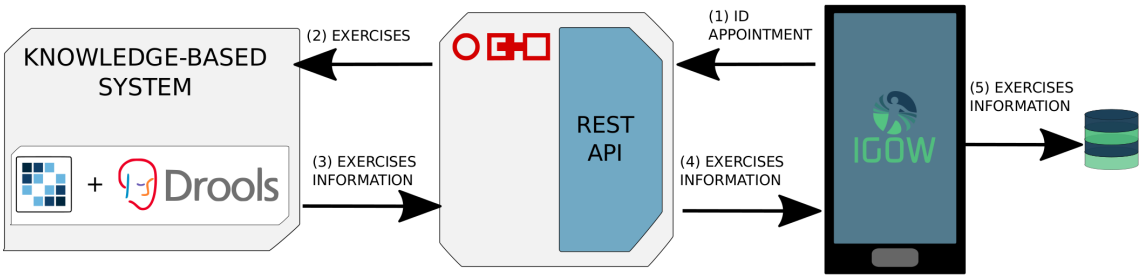


Figura 4.7: Información extra de un ejercicio (IGOW)

Figura 4.7(1) IGOW envía la id de la última cita y la fecha de modificación, si está desactualizado, (2) y (3) se pide la información extra de los ejercicios al sistema de conocimiento, tras recuperarla, (4) se envían los ejercicios actualizados con su información a la aplicación móvil. (5) La aplicación los persiste para su posterior presentación.

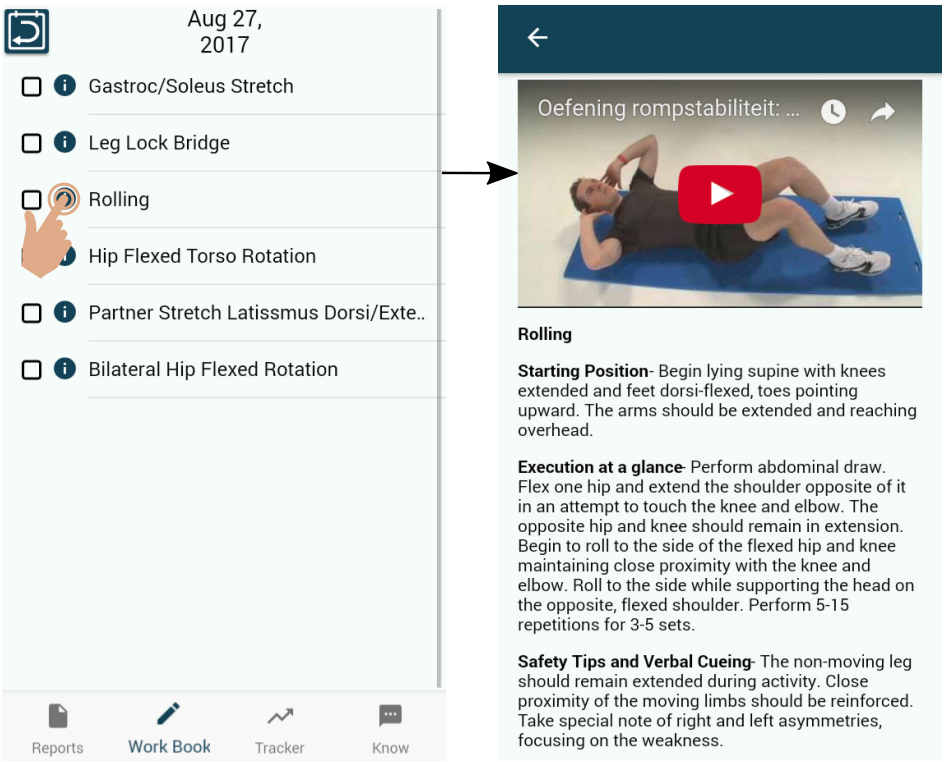


Figura 4.8: Información extra de un ejercicio (IGOW)

Feedback

Por último se crea un componente para la introducción del feedback de los ejercicios realizados. Este se muestra al pulsar una tarea y ofrece una serie de opciones que el

usuario puede seleccionar para completarla. El feedback se envía a Sport Medi Score mediante el uso de un servicio web y se genera una vista para su visualización desde la ventana de gestión de los pacientes.

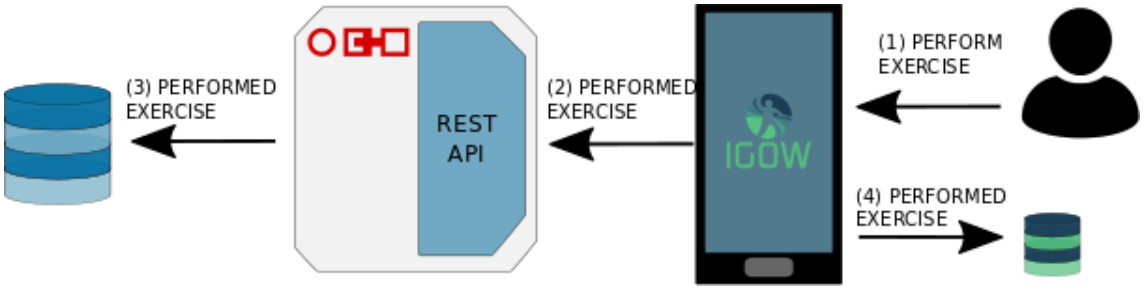


Figura 4.9: Diagrama de flujo completar y eliminar ejercicios

Figura 4.9: (1) El usuario proporciona el feedback del ejercicio realizado, (2) este se envía al sistema de backend de Sport Medi Score, (3) que lo persiste en su base de datos. (4) Cuando recibe mensaje de confirmación, IGOW guarda el ejercicio en su base de datos.

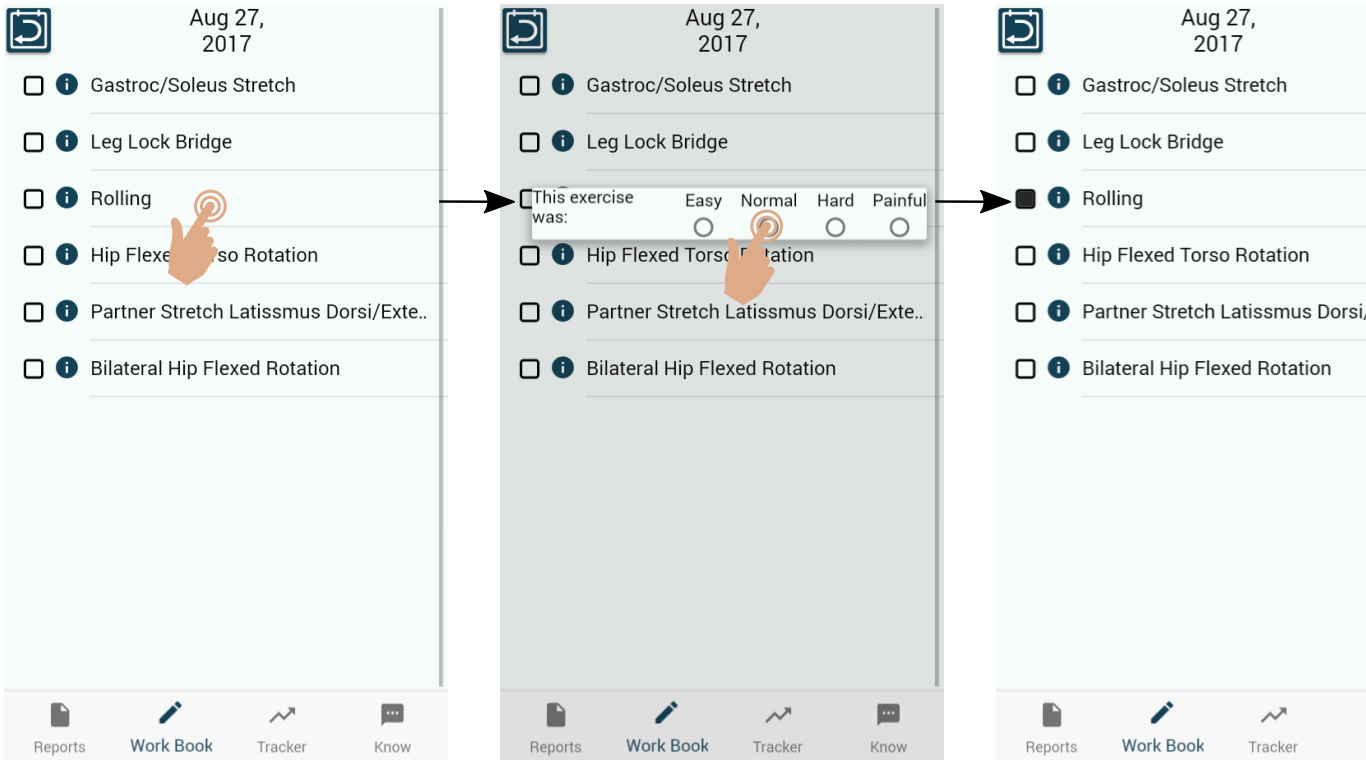


Figura 4.10: Realización de un ejercicio (IGOW)

Sprint 4: Tracker

La vista del tracker tiene como finalidad otorgar a los usuarios los medios para seguir el progreso que realizan a lo largo del programa. Esto se consigue mediante un pequeño proyecto implementado haciendo uso de vaadin-charts web.

Este proyecto se inyecta en la aplicación IGOW haciendo uso de un marco incorporado o iframe que apunta al archivo que lo contiene dentro del directorio local. La aplicación proporciona los datos al tracker haciendo uso de eventos a los que este está suscrito.

El tracker muestra estadísticas de progreso en función de la cantidad de ejercicios planificados para una semana concreta y los realizados a lo largo de ella. En una vista desglosada de los ejercicios realizados, se puede comprobar el feedback proporcionado para cada ejercicio pudiendo así tener una comprensión más amplia de la mejora obtenida durante el seguimiento del programa.

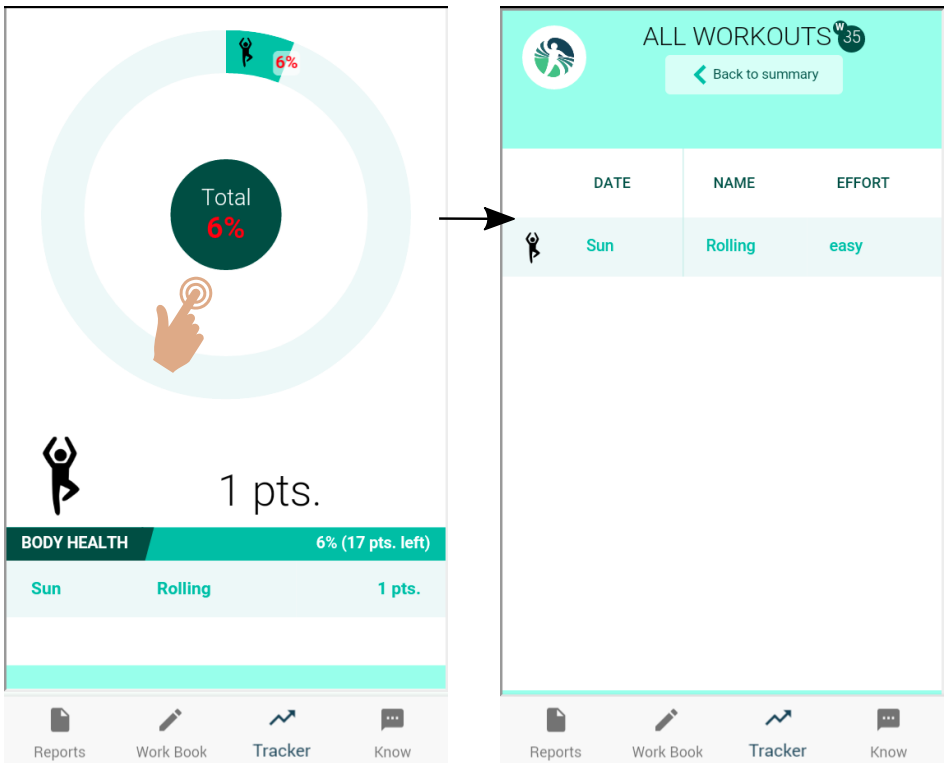


Figura 4.11: Página de seguimiento (IGOW)

Sprint 5: Messages

Ya que ahora es posible visualizar las tareas realizadas por los clientes, los doctores deben ser capaces de comunicarse con ellos en caso de observar anomalías en el desempeño de los ejercicios ya sea por dolor en la realización de alguno de ellos o la falta de progreso tras un largo período de tiempo.

Para ello necesitamos ser capaces de realizar envíos de mensajes desde el servidor hasta el cliente. Esto se consigue haciendo uso de las notificaciones push.

Las notificaciones push son un servicio que facilitan los principales sistemas operativos de dispositivos móviles (Android, iOS, Windows Phone y Blackberry) y permiten el envío de mensajes desde un servidor a un dispositivo determinado mediante el uso de una API de confianza. Existen varios servicios de este tipo pero nosotros utilizamos el de Google, Firebase Cloud Messaging (FCM) [23] y lo hacemos a través de un servidor proxy que es con el que se comunica Sport Medi Score.

Al iniciarse la aplicación móvil por primera vez, se genera un token para establecer la comunicación, un token que se envía a Sport Medi Score que lo persiste junto con los datos del usuario y accede a él cuando es necesario para el envío de un mensaje.

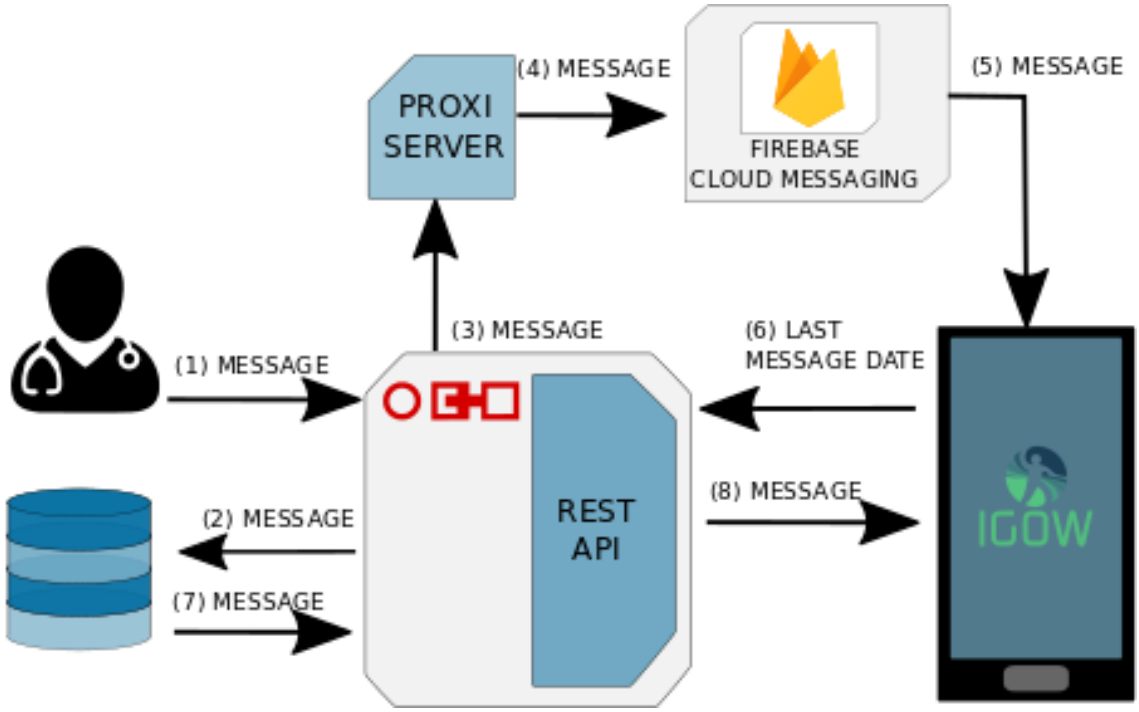


Figura 4.12: Diagrama de flujo envio de mensajes

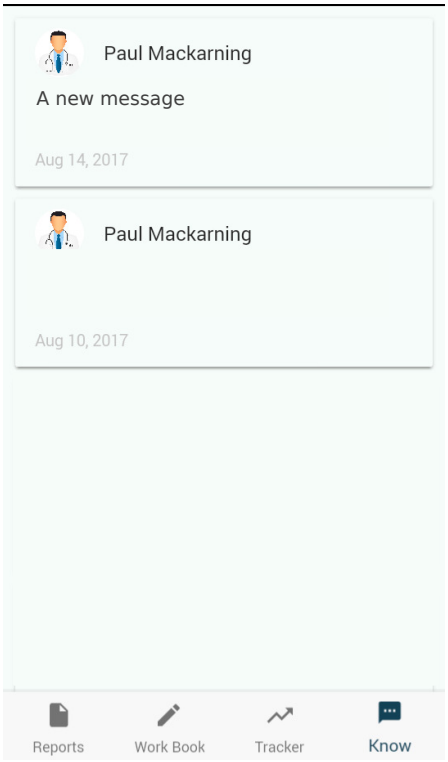


Figura 4.13: Página de mensajes (IGOW)

Figura 4.12: (1) el doctor escribe un mensaje, (2) este se guarda en la base de datos y (3) se envía al servidor proxy con el token del usuario, en el proxy se le añaden los atributos necesarios y (4) se envía a la API de FCM, desde allí se redirecciona al dispositivo generando un evento, este evento (7) dispara en IGOW la comprobación de mensajes en el servidor (6) que obtiene los recién guardados y (8) los envía a la aplicación móvil. Al iniciarse la aplicación se llevan a cabo estos tres últimos pasos independientemente de si

han llegado o no notificaciones ya que Google no asegura que las notificaciones lleguen a su destino [24].

Los mensajes se mostrarán en la pestaña correspondiente 4.13 marcadas con la fecha y el emisor del mensaje.

CAPÍTULO 5
Testeo

El uso de tests automáticos nos permite verificar la calidad y el correcto funcionamiento de las aplicaciones y sistemas desarrollados. A la hora de automatizar los tests, tendremos que desarrollar los casos de prueba para cada nueva funcionalidad. El mejor momento para hacer esto es tras la finalización de la implementación de cada una de ellas.

Al definir los casos de prueba, el objetivo del tester es conseguir que el código desarrollado falle, es por eso que junto con los casos de uso más básicos, hay que hacer combinaciones que, aunque improbables, puedan llegar a ser introducidas en el sistema. El código que solo es capaz de tratar como entrada los casos lógicos y simples, será vulnerable a valores de entrada anómalos y por lo tanto fallará con facilidad. Si el programa permanece estable ante entradas excepcionales que pretenden provocar errores, podemos decir que se trata de un software robusto.

Sport Medi Score

El sistema de testeo de la aplicación web ya existía y está automatizado, los tests los lleva a cabo Jenkins tras la compilación del sistema como ya se ha presentado en la Figura 3.1. Por ello en vez de crear todo el sistema de testeo, sólo hay que añadir los nuevos tests para las funcionalidades implementadas. Se desarrollan dos tipos de tests en función de su alcance:

Test unitarios

Los test unitarios se realizan sobre componentes individuales, proporcionan la seguridad de que cada uno desempeña su función correctamente. A la hora de definir los tests unitarios, se define el entorno y los resultados que se espera obtener y se somete a ellos el componente, si con el entorno planteado no se obtienen los resultados predichos, el test falla. Cuando un test falla, se revisa la implementación hasta que se cumplan los requisitos definidos.

Un ejemplo básico de la definición de un test es el que utilizamos para demostrar el funcionamiento de el Objeto DAO que se encarga de gestionar la interacción con la base de datos para las notificaciones enviadas a los clientes. Esta es una de las funciones o tests que forma parte de la batería de pruebas de este objeto.

```
1 // ...
2   @Autowired
3   private IPushNotificationDao pushNotificationDao;
4
```

```
  public PushNotification createPushNotification() {
    // ...
  }

  /*Check that the object is persisted into the database*/
  @Test
  public void storePushNotification() {
    Assert.assertEquals(pushNotificationDao.getRowCount(), 0);
    PushNotification testPushNotification = createPushNotification();
    pushNotificationDao.makePersistent(testPushNotification);
    Assert.assertEquals(pushNotificationDao.getRowCount(), 1);
  }
// ...
```

La etiqueta @Test define la función como una que debe ser ejecutada al pasar los tests. En este ejemplo se muestra cómo se comprueba que la realización de una inserción en la base de datos al usar el método `makePersistent(...)`. El test se activa en el archivo de configuración `testng.xml`.

```
<suite thread-count="1" verbose="1" name="BiitArchetypePersistence" parallel="none" group-by-instances="true">
  <test name="BiitPersistenceTest" enabled="true" preserve-order="true">
    <groups>
      <run>
        ...
        <include name="pushNotificationDao" />
      </run>
    </groups>
    <classes>
      ...
      <class name="com.biit.usmo.persistence.dao.jpa.PushNotificationDaoTest" />
    </classes>
  </test>
</suite>
```

Figura 5.1: Sección del archivo testng.xml

Para comprobar su funcionamiento ejecutamos el comando `mvn test` desde la carpeta del proyecto.

```
Tests run: 73, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 91.562 sec - in TestSuite

Results :

Tests run: 73, Failures: 0, Errors: 0, Skipped: 0
```

Figura 5.2: Resultado de ejecutar comando mvn test

Test de integración

Estos tests nos sirven para comprobar la integridad del sistema, cómo funcionan todas las piezas del software unidas. Se hace una simulación de la interacción de un usuario con la aplicación mediante Vaadin Testbench, este nos aporta unos métodos específicos para interactuar con los objetos de la interfaz entre ellos: leer textos, rellenar campos, pulsar botones y buscar elementos por nombre o tipo.

Estos tests requieren de un navegador para poder ejecutarse. Para lanzarlos en el servidor de Jenkins ya que no contamos con una interfaz gráfica, hacemos uso del navegador headless PhantomJS, que emula un navegador en segundo plano permitiendo la interacción con los tests sin necesidad de una interfaz gráfica. A la hora de comprender por qué falla un test, puede ejecutarse haciendo uso de un navegador web normal como Firefox y así se puede ver la secuencia de interacciones que estos realizan.

La metodología de definición de los tests es parecida a la utilizada en los tests unitarios pero en este caso contaremos con maneras de interactuar con la interfaz en vez de hacerlo con los propios componentes del software.

IGOW

Los tests para la aplicación web deben integrarse con el sistema de entrega continua de Jenkins, para ello se definen las tareas necesarias para pasar los tests tras la compilación de la aplicación. Los tests se pasan haciendo uso del comando `npm test`, esto lanza karma con el navegador que hayamos elegido, en el caso de Jenkins, debido a que de nuevo no necesitamos ver como pasan los tests, utilizaremos PhantomJS, pero para su uso en desarrollo utilizaremos Chrome. Para definir los tests con Jasmine, crearemos un nuevo archivo `component.spec.ts` por cada componente que queramos probar. Dentro de este archivo definiremos los casos de prueba del componente, a la hora de definir los tests, tendremos que implementar servicios simulados o mocks, que sustituirán a los servicios reales utilizados por la aplicación y devolverán valores arbitrarios definidos por el desarrollador. Así probamos cada componente en un entorno completamente controlado y en el que podemos someter el software a situaciones específicas que puedan llevar a errores.

Una vez pasados los tests 5.3 y si no hay fallos, se realiza la versión de producción de la aplicación y estará lista para desplegarse en la plataforma que queramos.

```
Chrome 58.0.3029 (Linux 0.0.0): Executed 27 of 27 SUCCESS (6.911 secs / 6.901 secs)
TOTAL: 27 SUCCESS
```

Figura 5.3: Resultado de ejecutar el comando npm test

CAPÍTULO 6
Conclusiones y casos de éxito

Nuestro sistema proporciona a los usuarios las herramientas necesarias para implicarse de manera completa en el desarrollo del programa de salud que propone el centro. Gracias a los informes, tiene acceso a toda la información necesaria para comprender la situación en la que se encuentra creando así un punto de partida. Se definen unos objetivos con el centro, creando la línea de meta del programa, estos objetivos son valores mesurables y por lo tanto se puede llevar un registro del progreso en cada uno. El centro define el camino a seguir por los clientes mediante la selección de ejercicios que se ven reflejados como tareas en la aplicación móvil. Gracias al feedback proporcionado por los clientes y el sistema de mensajes, el centro puede ofrecer correcciones y sugerencias a lo largo del programa en caso de que surja una situación no prevista. A la hora de motivarse para continuar con el programa, los clientes puede consultar el progreso realizado a lo largo del tiempo y ver cómo ha avanzado desde el inicio.

Pese a tratarse de un proyecto complejo ya que cuenta con una gran cantidad de herramientas y tecnologías de las que se hace un uso continuo, se ha conseguido una solución sencilla para la experiencia del usuario que no exige al más que la confirmación de la realización de los ejercicios que le son propuestos por el centro. Gracias al empleo de infografías a la hora de mostrar los informes, es muy fácil comprender los indicadores mostrados y toda la información que se requiere para alcanzar las metas está siempre disponible.

Sport Medi Score está desplegado en el centro Orbis Sport en la ciudad de Sittard-Geleen, en estas instalaciones cuenta con más de 5500 clientes registrados, algunos de los cuales ya se están realizando pruebas con la aplicación móvil. Para poder instalar la aplicación sin supervisión, se ha desplegado en la plataforma de Google, Play Store, desde donde se la descargan e instalan los usuarios de pruebas actuales. Por el momento se encuentra en versión beta y el lanzamiento final de la aplicación está programado para octubre.

CAPÍTULO 7
Trabajo futuro

Uno de los objetivos finales de la aplicación móvil es que sea multiplataforma, ya tenemos la aplicación desplegada para Android en la Play Store y esperamos extender este soporte a la App Store de Apple en septiembre, aún no se ha podido hacer ya que se está a la espera de la entrega de distintos modelos de dispositivos para probar los sistemas iOS.

Para próximas iteraciones sobre el proyecto, se quiere ofrecer la posibilidad de rellenar los formularios de inscripción haciendo uso de IGOW. Esto podría ahorrar tiempo a los clientes al ser capaces de rellenarlos entre examinaciones. Esta funcionalidad además podría extenderse a la realización de nuevos ejercicios, como tests de seguimiento para tratamientos psicológicos.

Se planea también añadir integración con dispositivos werables para obtener medidas precisas del estrés producido por los ejercicios en los clientes, el ritmo cardíaco o la distancia recorrida en un ejercicio concreto son algunas de las aplicaciones proyectadas para estos dispositivos.

A largo plazo, se pretende realizar un estudio sobre la aceptación del sistema y la fidelización de los usuarios de las clínicas para comprobar que mejora la implicación de los clientes con el programa y que al ver resultados en su desarrollo vuelven al centro para continuar creando nuevos objetivos.

Bibliografía

[1] Definición de salud, RAE. Consultado el 13 de Septiembre de 2017, en http://dle.rae.es

[2] Convenios de la OIT referentes a prevención. Consultado el 13 de Septiembre de 2017, en http://www.insht.es/InshtWeb/Contenidos/Documentacion/TextosOnline/ErgaFP/2006/ErFP52_06.pdf

[3] Mejorar las condiciones de vida cotidianas. Consultado el 13 de Septiembre de 2017, en http://www.who.int/social_determinants/thecommission/finalreport/closethegap_how/es/index1.html

[4] 29148-2011 - ISO/IEC/IEEE International Standard. Systems and software engeneering. Consultado el 13 de Septiembre de 2017, en https://standards.ieee.org/findstds/standard/29148-2011.html.

[5] Android dashboards. Consultado el 13 de Septiembre de 2017, en https://developer.android.com/about/dashboards/index.html.

[6] Artículo 7 de la *Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal* Consultado el 13 de Septiembre de 2017, en https://www.boe.es/buscar/doc.php?id=BOE-A-1999-23750

[7] Taiga. Consultado el 13 de Septiembre de 2017, en https://taiga.io/

[8] Jenkins. Consultado el 13 de Septiembre de 2017, en https://jenkins.io/

[9] Vaadin. Consultado el 13 de Septiembre de 2017, en https://vaadin.com

[10] Drools. Consultado el 13 de Septiembre de 2017, en https://www.drools.org/

[11] Liferay. Consultado el 13 de Septiembre de 2017, en https://www.liferay.com

[12] TestNG. Consultado el 13 de Septiembre de 2017, en http://testng.org/doc/

[13] Vaadin TestBench. Consultado el 13 de Septiembre de 2017, en https://vaadin.com/testbench

[14] PhantomJS. Consultado el 13 de Septiembre de 2017, en http://phantomjs.org/

[15] Apache Cordova. Consultado el 13 de Septiembre de 2017, en https://cordova.apache.org

[16] Angular. Consultado el 13 de Septiembre de 2017, en https://angular.io/

[17] TypeScript. Consultado el 13 de Septiembre de 2017, en https://www.typescriptlang.org/

[18] Ionic. Consultado el 13 de Septiembre de 2017, en https://ionicframework.com/

[19] Npm. Consultado el 13 de Septiembre de 2017, en https://www.npmjs.com/

[20] Angular Style Guide. Consultado el 13 de Septiembre de 2017, en https://angular.io/guide/styleguide

[21] Network Working Group (May 2000). *"HTTP Over TLS"* Consultado el 13 de Septiembre de 2017, en https://tools.ietf.org/html/rfc2818

[22] JSON Web Token (JWT). https://tools.ietf.org/html/rfc7519

[23] Firebase Cloud Messaging. Consultado el 13 de Septiembre de 2017, en https://firebase.google.com/docs/cloud-messaging

[24] Messaging Concepts and Options Consultado el 13 de Septiembre de 2017, en https://developers.google.com/cloud-messaging/concept-options

[25] Todea, Virgil Cristian (Septiembre, 2016) *Diseño e implementación de un sistema de entrega continua para aplicaciones web sobre contenedores Docker*.

[26] Llorens Roig, Anna (Julio, 2017) *Diseño e implementación de un sistema multiplataforma para la generación automática de infografías*

[27] Smith, Reid (Mayo 8, 1985) *Knowledge-Based Systems Concepts, Techniques, Examples* Consultado el 13 de Septiembre de 2017, en http://www.reidgsmith.com/Knowledge-Based_Systems_-_Concepts_Techniques_Examples_08-May-1985.pdf