



Scimago App



UNIVERSIDAD POLITÉCNICA DE MADRID
Escuela Técnica Superior de Ingeniería de Sistemas Informáticos

MÁSTER EN INGENIERÍA WEB

Proyecto Fin de Máster

SCIMAGO APP

Autor

Adrián Rodríguez Caballero

Tutor

Luis Fernando de Mingo López.



INDICE GENERAL

Contenido	
1 RESUMEN	3
2 ABSTRACT.....	4
3 INTRODUCCIÓN	5
4.OBJETIVOS.....	6
5 DESCRIPCIÓN DEL ENTORNO.	8
6. METODOLOGÍA DE DESARROLLO	12
7- DESCRIPCIÓN DE PROBLEMA.....	15
8 INCEPTION.	17
9 DIAGRAMA DE LA BASE DE DATOS.....	34
10- DIAGRAMA DE PAQUETES.....	35
11 PANTALLAS PRINCIPALES DE USO	37
12 LINEAS DE FUTURO.....	40
13 LECCIONES APRENDIDAS Y CONCLUSIONES.....	41
14 BIBLIOGRAFIA	43



1. RESUMEN

Tras acabar las clases del master y tener que decidir qué proyecto realizar como PFM opté por el desarrollo de una aplicación móvil Android, plataforma sobre la que nunca había trabajado y cuyo estudio en una de las asignaturas despertó mi interés.

Tras decidir esto el segundo paso era decidir en qué iba a basar el desarrollo, después de estudiar varias posibilidades y analizar una de las propuestas del foro despertó mi atención.

Existe una plataforma en la que se muestran las revistas de ciencia más influyentes del planeta en las cuales exponen artículos los científicos más importantes. Así me decidí a poder gestionar esta información desde una app móvil.

Las inquietudes de cada persona son muy dispares. Sin embargo, a todos nos gusta obtener la última información disponible acerca de nuestros intereses, así que las personas interesadas en las revistas de ciencia les gustará obtener cuales de sus artículos preferidos son los más leídos en un ranking expuesto por Scimago en su web. Con un simple vistazo obtendremos la información necesaria de un vistazo.

PLABRAS CLAVE

App, Android , Sqlite , Scimago , Web Scrapping , Jsoup , Revista , Reporting.



2. ABSTRACT

After finishing the master classes and having to decide which project I am going to carry out as PFM, I chose the development of an Android mobile application, platform which I had never worked. When I studied this topic in one of my subjects awakened my own interest.

Secondly, I had to decide what I am going to develop. After studying different options, I analyzed one of the forum's proposal, which finally I chose.

There is a web site where you are able to check the most important science magazines in the world. Thus, I decided to manage this information from a mobile app.

People want to check information about the updates of their inquisitiveness. In addition, people who like science articles are able to check the scimago ranking. In this way, just with a click, you will check the information that you need.

KEYWORDS

App, Android , Sqlite , Scimago , Web Scrapping , Jsoup , Revista , Reporting.



3. INTRODUCCIÓN

Para poder obtener la información acerca de las revistas de ciencia más influyentes, publicadas en la web Scimago, es necesario acceder desde un ordenador portátil, o bien desde el móvil abriendo un navegador web y buscando la url de la web.

Esta tarea es algo tediosa y puede simplificarse eliminando navegador web o el tener que arrancar un pc para acceder.

La aplicación resultada de este proyecto nos permitirá obtener la información de las 50 mejores y más influyentes revistas con un solo toque de dedo, de forma rápida y simple.

Para almacenar estos datos y poder consultarlos cuando uno este fuera de cobertura usaremos una base de datos que almacena información sobre las revistas sus títulos su índice de impacto tipología...

La aplicación permite obtener de forma sencilla los datos, además de consultar cada uno de ellos y ver al detalle, o bien generar un report en formato pdf exportable a otras ubicaciones.



4. OBJETIVOS

El objetivo principal que se persigue en la elaboración de este proyecto fin de master de basa en conseguir crear una app que facilite la obtención de las revistas más influyentes publicadas en Scimago.

El sistema se basará en una aplicación móvil para dispositivos Android, que ofrecerá una interface amigable y conocida para los usuarios de otras aplicaciones de la misma plataforma, lo cual le proporcionará un rápido y fácil anejo al alcance de la mayoría de los usuarios.

Como objetivo de fondo a la hora del desarrollo del proyecto estará siempre la aplicación de los distintos conocimientos adquiridos en las asignaturas cursadas a lo largo de la impartición del master en ingeniería web, así como los conocimientos adquiridos en el terreno laboral, con ánimo de profundizar y ampliar en la medida de lo posible.

En resumen, los objetivos a alcanzar son:

- Analizar, diseñar e implementar una aplicación para la obtención y gestión de datos de scimago.
- Obtención de los datos de una web sin Api, mediante la técnica de obtención de datos denominada Webscrapping.



- Permitir al usuario introducir anotaciones propias, visualizar los datos en forma de lista, visualizar los datos en forma de detalle, obtención de listado en formato Pdf.
- Implementar la capa de persistencia de capa de datos permitiendo al usuario almacenar, modificar y consultar la información en la aplicación.
- Diseñar la aplicación de forma que cualquier usuario de un dispositivo móvil sea capaz de realiza las tara apoyándose en una presentación de contenidos con una interfaz amigable y de fácil manejo.
- Vistas de futuro para que a posteriori se puedan añadir funcionalidades o realizar mejoras para dar respuesta a las necesidades de los usuarios.
- Por supuesto, una obtención del título de master en ingeniería web en la universidad politécnica de Madrid.



5. DESCRIPCIÓN DEL ENTORNO

- Entorno de desarrollo

El desarrollo de aplicaciones par Android se puede realizar con diversos lenguajes de programación, C#, C , .Net, java, VisuaBasic..., para la implementación de este proyecto se ha optado por Java y XML.

- Java

Es un lenguaje de programación de propósito general, concurrente y orientado a objetos que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban programa una vez y lo ejecuten en cualquier dispositivo (WORA o write once, run anywhere”), lo que quiere decir que el código es ejecutado en una plataforma no tiene que ser compilado para correr en otros.

Android no utiliza los estándares establecidos de Java, Java Se y ME lo cual es un problema de desarrollo. Esto impide la compatibilidad entre aplicaciones JavaScript en otras plataformas. Android solo utiliza la sintaxis y la semántica de Java, pero no incorpora en su totalidad las bibliotecas de Java y Apis que acompañan a Java SE o ME. Sin embargo, hay diversas herramientas en el mercado de empresas como Myriad grop que dan un servicio entre J2Me y Android.



- XML

Siglas en inglés de eXtensible Markup Language, traducido como “lenguaje de marcado extensible” o “lenguaje de marcas extensible”, es un meta – lenguaje que permite definir lenguajes de marcas desarrollador por el World Wide Web consortium (W3C) utilizado para almacenar datos en forma legible. Proviene del lenguaje SGML y permite definir la gramática de lenguajes específicos (de la misma forma que HTML es a su vez un lenguaje definido por SMGL) para estructurar documentos grandes. A diferencia de otros lenguajes, XM da soporte a bases de datos, siendo útil cuando varias aplicaciones deben comunicarse entre sí o integrar información.

- Jsoup

Esta es la librería que se ha elegido para realizar la captura de datos de la web Scimago. Esta librería nos permite realizar el scrapping.

El **Web Scraping** (o **Scraping**) son un conjunto de técnicas que se utilizan para obtener de forma automática el contenido que hay en páginas web a través de su código HTML. El uso de estas técnicas tiene como finalidad recopilar grandes cantidades de datos de diferentes páginas web cuyo uso posterior puede ser muy variado: homogenización de datos, tratamiento de contenido para la extracción de conocimiento, complementar datos en una web, etc. Decir por último en esta introducción que las técnicas de Scraping se pueden enmarcar dentro de los campos de la Inteligencia Artificial y del Big Data en la primera fase de



recolección de datos para su posterior almacenamiento, tratamiento y visualización.

- [Sqlite](#)

Se ha optado por esta solución para la persistencia de datos. Sqlit es un sistema de gestión de bases de datos relacional con ACID, contenida en una pequeña biblioteca escrita en C. Sqlite es un proyecto publico creado por Richard hipp.

A diferencia de los sistemas de gestión de bbdd client – servidor, el motor de sqlite no es un proceso independiente con el que el programa principal e comunica. En lugar de eso sqlite se lanza con el programa apando a ser parte del mismo. El programa usa la funcionalidad e sqlite a través de llamadas simples a subrutinas y funciones. Esto reduce la latencia en el acceso a la base de datos, debido a que las llamadas a funciones son más eficientes que la comunicación entre procesos. El conjunto e base de datos son guardado como un solo fichero estándar en la maquina host. Este diseño simple se logra bloqueando todo el fichero de base de datos al principio de cada transacción.

- [Android studio](#)

Como entorno de programación integrado se ha utilizado Android studio 3.1.3 en su versión para la plataforma Windows, Anadroid studio reemplazo a Eclipse como el IDE oficial para el desarrollo de aplicaciones Android. La primera versión estable fue publicada en diciembre de 2014.



Está basado en el software IntelliJ IDEA de JetBrains y ha sido publicado de forma gratuita a través de licencia Apache 2.0. Está disponible para plataformas Windows, macOS, GNU/LINZ. Ha sido diseñado para el desarrollo de Android Específicamente.

- Entorno de ejecución

La herramienta resultante de este proyecto va a consistir en una app móvil que correrá en dispositivo bajo arquitectura Android.

Los requerimientos mínimos del dispositivo para su correcta instalación son contar con versión mínima de Android 7.1 y aceptar los permisos de:

- (Seguridad) Acceso a la red
- Ver conexiones de Red
- acceder al sistema de archivos de SO.



6. METODOLOGÍA DE DESARROLLO

La forma de trabajo seguida durante el proyecto corresponde Agile con SCRUM. Se ha optado por esta metodología ya que el equipo de trabajo es de una sola persona y el tiempo disponible para su desarrollo, así como la necesidad de compaginar con el horario laboral no permite espacio para la elaboración de documentación que requiere RUP.

Scrum se utiliza para el desarrollo de software, aunque otros sectores se aprovechan de sus beneficios implantando esta metodología en sus modelos organizativo

Scrum es una evolución del agile management. Por un lado, el mercado demanda calidad, rapidez en la entrega y bajos costes, para lo cual una empresa debe ser ágil y flexible en el desarrollo de los productos para conseguir ciclos de desarrollo cortos que satisfagan la demanda de los clientes sin quitar por ello calidad en el resultado final. Es una metodología fácil de implantar y popular por los rápidos resultados que consigue. Evita burocracia y documentación de forma que los resultados lleguen rápidamente.



- Funcionamiento

Scrum diferencia dos elementos, actores y acciones.

Los actores ejecutan las acciones y se establecen cuatro tipologías

Diseño del producto: Normalmente el cliente, que marca los requisitos del proyecto.

Experto SCRUM : El gestor de proyecto, que vela porque la metodología se cumpla y guíe al resto del equipo.

Equipo SCRUM: Los desarrolladores que ejecutarán el proyecto.

Usuarios: Los beneficiarios finales del producto y a los que también se debe implicar desde un primer momento para que aporten opiniones y permitan mejorar el producto durante su desarrollo, antes incluso de haberse finalizado una primera versión del mismo.

Para el desarrollo de la aplicación deberé tomar el papel de cada uno de los actores.

Las acciones se dividen en categorías y han sido pensadas para minimizar el esfuerzo y maximizar el resultado.

Product backlog : tareas a realizar y objetivos que se deben conseguir, estos son marcados por el cliente y el experto SCRUM.

Sprint Backlog: tareas que se realizarán en un plazo muy corto, entre dos y cuatro semanas. Al finalizar se debe obtener un entregable.



Sprint Planning Meeteng : reunión que sirve para decidir y planificar que tareas pasaran del product backlog al sprint backlog.

Daily SCRUM Meeteng: reunión operativa que se realiza cada día mientras dura el Sprint backlog en la que cada miembro del equipo comenta que áreas ha realizado, cuales va a realizar durante el día y que riesgos percibe.

Una vez terminado el sprint backlog se revisa y se extraen las lecciones aprendidas para el próximo sprint backlog.



7. DESCRIPCIÓN DE PROBLEMA

Debemos tener claro el motivo por el cual crear una solución y por tanto la aplicación, cuales son las utilidades y beneficios que ofrecerá a los usuarios.

Para ello se ha realizado un análisis de las posibilidades que existen para consultar los datos acerca del dato de la web scimago, detectando la necesidad de poder consultarlos de forma rápida y sencilla sin necesidad de navegadores y/o equipos informáticos.

Con esto podemos pasar a describir las líneas del proyecto y plantear las necesidades que se desean abordar para construir el backlog del proyecto.

- Definición de líneas generales

Esta definición servirá para mejorar consistencia y claridad de los procedimientos y reducir las tareas de ejecución y tiempo de aprendizaje, así como el número y tipo de errores que puedan ocurrir.

Las líneas generales sobre las que nos moveremos se basan en los objetivos enumerados en la justificación:

- Conseguir crear una app que facilite la obtención de las revistas más influyentes publicadas en scimago.
- El sistema debe mostrar el ranking de revistas publicadas más influyentes en tiempo real.



Scimago App

- El sistema debe poder tener acceso a los datos sin conexión a internet.
- El sistema debe mostrar el detalle de cualquiera de las revistas.
- El sistema debe poder obtener un reporte en formato Pdf.



8. INCEPTION

En este apartado se describe el proceso previo al desarrollo seguido para generar una hoja de ruta.

- Product backlog

Funcionalidades planteadas inicialmente durante la fase de inception.

- Consultar ranking
- Actualizar ranking
- Almacenar datos
- Búsqueda de revistas
- Crear nota
- Editar nota
- Gestión de reporting.



- Hoja de ruta

En este apartado se muestra la asignación de costes y el valor de cada una de las funcionalidades para calcular su ROI y poder así componer un ahoja de ruta.

Los valores de los costes se usan en puntos de historia (story points) tomando como unidad de referencia que un punto de historia equivale al coste de desarrollar una funcionalidad de crear una nota.

Las cifras del valor son las resultantes de repartir 300 puntos de valor entre todas las funcionalidades, según el valor que aportaran al usuario.

El ROI que se muestra es un simple cociente Valor/coste con fines comparativos

FUNCIONALIDAD	COSTE	VALOR	ROI
1Consultar ranking	5	60	12
2Actulizar Ranking	5	40	8
1Almacenar Datos	3	50	16.6
2Busqueda revistas	2	50	25
1Crear notas	1	15	15
3Editar notas	1	15	15
3Gestion reporting	4	70	17.5



Según los datos obtenidos en la estimación, vamos a elaborar la siguiente hoja de ruta, con cuidado que cada entrega aporte menos valor que la anterior y a su vez no haya grandes oscilaciones en los costes de una entrega a la siguiente:

➤ Entrega 1: Desarrollo del Scrapping y almacenaje en Base de datos

- Consulta ranking
- Almacenar Datos
- Crear notas

➤ Entrega 2: Actualización del ranking y motor de búsqueda.

- Actualizar Ranking
- Búsqueda de revistas

➤ Entrega 3: Edición de notas y Reporting

- Edición de notas
- Gestión del reporting



#	Entrega	Coste	Valor
1	Desarrollo del Scrapping y almacenaje en Base de datos	9	43.5
2	Actualización del ranking y motor de búsqueda.	7	33
3	Edición de notas y Reporting	5	32.5

- [Sprint 1: Desarrollo Scrapping y almacenaje en Base de datos](#)

Describimos los pasos seguidos para analizar diseñar y probar las funcionalidades del primer sprint.

Sprint backlog

Se describen sistemáticamente las historias de usuarios asociadas a las funcionalidades de este sprint.



Consulta Ranking

Descripción	Como usuario deseo poder listar las revistas de scimago.
Conversación	Usuario solicita consultar los datos de scimago -> El sistema maestra los datos en pantalla.
Criterios de aceptación	La aplicación deberá mostrar un botón para obtener y o actualizar los datos de la web.

Almacenaje en base de datos

Descripción	Como usuario deseo poder tener a mano los datos sin conexión
Conversación	Usuario solicita consultar los datos-> el sistema obtiene los datos , los almacena y los muestra en pantalla
Criterios de Aceptación	El usuario sin necesidad de actuar debe poder obtener los datos y listarlos.



Planificación del sprint

Describimos las actividades llevadas a cabo durante la planificación.

Tareas

Consulta de ranking

- [Análisis de la librería a usar.](#)

Después de investigar sobre una API y ver que de forma gratuita no se puede tener acceso al api disponible, nos decidimos a buscar otra solución para obtener los datos de la web.

Esta solución encontrada es la técnica denominada Scraping que no es más que buscar patrones de texto hasta poder obtener la información deseada.

Se analizan las distintas posibilidades que se usarán para parsear los datos de la web.

Optamos por usar la librería JSOUP.

Estudiamos el funcionamiento de la librería y sus funciones, mediante ejemplos de distintas páginas de internet.

Analizamos la estructura de la web de scimago y buscamos un patrón de búsqueda donde podamos encontrar los datos para capturarlos.



Una vez encontrado el patrón lo que hacemos es recorrerlo tantas veces como elementos encontramos. El paso posterior será introducirlo en base de datos.

Se integra en el sistema y creamos la clase MagazineAsync que se encargara de realizar la captación de los datos de forma asíncrona sin interceder al sistema.

- Almacenaje en la base de datos

Tras poder obtener los datos y saber exactamente que datos obtenemos de la web, creamos la clase Magazine con la estructura necesaria para el almacenaje de los datos necesarios

Campo	Tipo	Descripción
_ID	Int	Identificador único
Title	String	Título de la revista
Type	String	Tipo de la revista
Sjr	Int	Índice de impacto
Refs	Int	Referencias
Notes	String	Notas del usuario



Se crea la clase MagazineRepositoy encargada de crear, editar, listar y eliminar los datos de magazines. Para usar esta clase nos apoyamos en la clase magazineContract donde almacenamos los datos referentes a nombrado de los campos en base de datos. Esta clase nos facilitara la labor a la hora de realizar peticiones a la base de datos. Además, un cambio en la estructura de los datos, tiene un coste muy pequeño y nos permite monitorear cualquier error en el código cuando se produzca un cambio en los requisitos de la base de datos.

- Creación del Activity Main

Clase encargada de iniciar el programa. En ella se mostrarán las distintas opciones con las que obtener los datos. Mostramos esta imagen como referencia:





1- Implementación del Adaptador :MagazineAdapter

Esta clase será la encargada de obtener todos los elementos mediante el id en la vista, necesarios para mostrar los datos en pantalla.

2- Diseño del fichero Elementos.xml y ListView_magazine.xml

El fichero elementos .xml es el encargado de definir las cabeceras del listado a mostrar por pantalla.

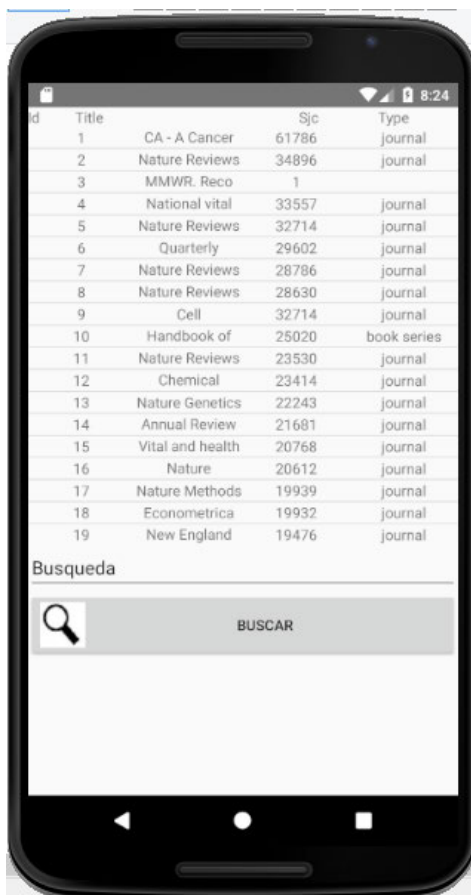
Por otro lado, el fichero ListView_magazine es el encargado del almacenaje de los datos a mostrar en pantalla al usuario.

3- Por último. se crea la clase ShowMagazine encargada de gestionar los datos junto con MagazineAdapter y lanzar el intent que mostrara finalmente el detalle de los datos.



Tareas (Pruebas de aceptación)

Tras finalizar el primer sprint se realizan las pruebas de aceptación para comprobar que el código cumple con lo descrito en el sprint Backlog



En la imagen mostrada se ven los datos consultados y guardados en la base de datos, que posteriormente se plasman en la vista al usuario.

Observamos que se han alcanzado todos los criterios de aceptación definidos al inicio del sprint por lo que está lista para ser entregada al cliente.



- [Sprint 2: Actualización del ranking y motor de búsqueda](#)

Aquí se describen los pasos seguidos para analizar diseñar y probar las funcionalidades del segundo sprint.

- [Sprint Backlog](#)

Aquí se describen sistemáticamente las historias de usuario asociadas a sprint.

Actualizar Ranking

Descripción	Como usuario deseo poder actualizar los datos de scimago.
Conversación	El usuario solicita al sistema la actualización de los últimos datos. El sistema realiza el borrado de la base de datos y la actualización de los datos.
Criterio de aceptación	El usuario podrá visualizar los nuevo datos obtenidos del proceso de consulta a la wb.



- [Modificación de la clase MagazineAsync](#)

Dado que debemos restablecer los datos ya guardados y substituirlos por los nuevos actualizados, realizamos la modificación necesaria en la clase MagazineAsync para que realice el borrado previo de la base de datos.

- [Modificación de la clase MagazineRepository](#)

Para que esto se realice, modificamos la clase de bbdd para que realice el borrado de los magazines almacenados en base de datos.

Es decir, creamos la función deleteall para vaciar la base de datos.

Búsqueda de datos

Descripción	Como usuario deseo buscar una revista por su nombre
Conversación	El usuario solicita el nombre del magazine, Sistema realiza la búsqueda por el contenido
Criterios de aceptación	Al pulsar el botón de buscar, el listado de magazines se actualiza.



- [Modificación de ListView_magazine.xml](#)

Agregamos al diseño de la pantalla edit text y un botón para controlar el evento de búsqueda del usuario.

Después lo introducido por el usuario se manda al método `getFromContent()` de la clase `Magazine repository` para que realice la acción necesaria de búsqueda.

Una vez adquiridos los datos de la consulta recargamos el listado para hacer efectiva la búsqueda y mostrar los datos filtrados por pantalla.

- [Modificación de la clase Magazine report](#)

Agregamos el edit text como private para poder usarlo, además agregamos la gestión del list box para realizar la actualización de los datos tras la búsqueda.

- [Modificación de la clase Magazine repository](#)

Agregamos la funcionalidad de búsqueda por un texto que coincida en contenido con alguno de los elementos de la base de datos.

Para implementarlo, creamos el método `getFromContent` y nos apoyamos en las funcionalidades que nos presta `sqlite`, creando una query y pasándosela como parámetro a una función de ejecución de queries cuyo resultado devuelve en forma de array list.



- [Sprint 3: Edición de notas y Reporting](#)

Aquí se describen los pasos seguidos para analizar diseñar y probar las funcionalidades del tercer sprint.

- [Sprint backlog](#)

Aquí se describen sistemáticamente las historias de usuario asociadas a las funcionalidades de este sprint

Modificar la nota agregada.

Descripción	Yo como usuario deseo poder editar a nota introducida en el sistema
Conversación	El usuario selecciona un elemento de la lista, accede a él y modifica el dato. El sistema realiza las operaciones necesarias para actualizar el dato.
Criterios de aceptación	El usuario al finalizar la ejecución, y acceder a dat lo vera cambiado.



- [Creación del fichero show_magazine.xml](#)

En este fichero agregamos los textview y el edit text junto con el botón que nos permitirá la edición de los datos necesarios.

- [Edición del fichero Show Magazine](#)

En esta clase se obtienen todos los elementos del layout a tratar.

Como modificación agregamos el edit text para las notas y a su vez, mediante el elemento pasado por parámetro, obtenemos las notas almacenadas en el para después traspasarlas al text del edit text y poder así mostrarlas en el fichero show magazine.xml.

- [Modificación del fichero magazine repository](#)

Asignamos al botón la función alojada en showmagazine class para guardar el dato modificado de notas.

El método sabe, realiza una llamada a la clase magazine repository para que esta con ayuda de una query realice el guardado del dato.

Como paramentos recibe el id del magazine a actualizar y el dato a introducir.



Se realiza la búsqueda del elemento en la base de datos y en caso de ser encontrado actualiza el dato de “nota”.

Módulo de reporting

Descripción	Yo como usuario deseo listar los datos e formato pdf
Conversación	El usuario solicita al sistema la obtención de un fichero pdf.
Criterios de aceptación	El usuario obtiene un fichero pdf con los datos del sistema.

- Creación de la clase MagazineReport

Mediante la funcionalidad Document, se crea un objeto para ir rellenando los datos al pdf.

Tras crear el document y configurarlo, se lo pasamos al objeto instanciado PdfWriter que será el encargado de ir escribiendo los datos.

Se desarrolla la carga de los datos al fichero mediante un recorrido de la base de datos.



El fichero se almacenará en el almacenamiento externo del teléfono, así no cargamos de ficheros al sistema.

Para obtener el listado, hemos creado en la clase Magazine un método de toString propio el cual nos proporciona el dato de cada magazine almacenado en la base de datos ya preparado para trasladarlo al fichero sin necesidad de tratarlo.

- [Edición del android mainfest](#)

Es necesario indicar en el fichero android mainfest, que queremos hacer uso del almacenamiento externo del teléfono, sino no será posible crear el fichero y nos reportará un error.



9. DIAGRAMA DE LA BASE DE DATOS

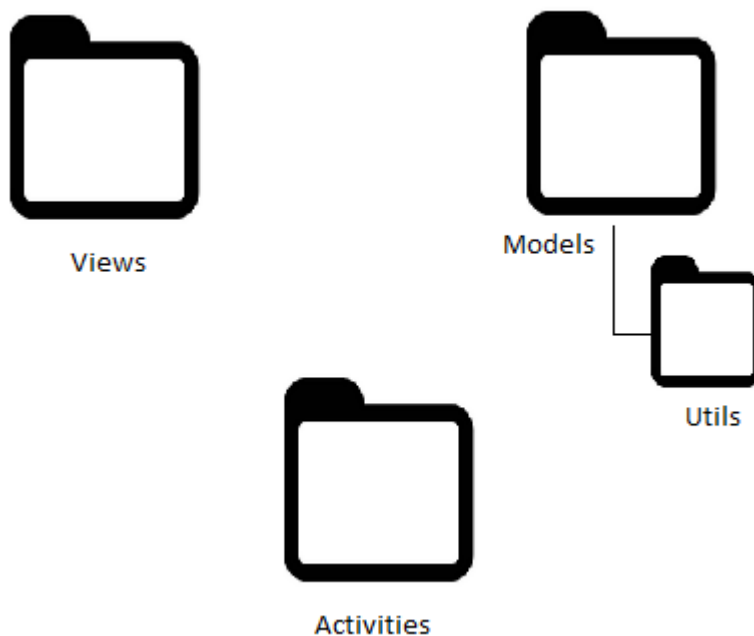
El sistema se compone de una única base de datos en la que se almacenan los siguientes datos:

Identificador	Detalle
Title	Título del magazine
Type	Tipo de revista en la que la propia web ha clasificado el título
Sjc	Índice de impacto proporcionado por la web.
Refs	Numero de referencias, proporcionado por la web.
Notes	Notas introducidas por el usuario para el título seleccionado.



10. DIAGRAMA DE PAQUETES

Una vez concluido los sprints y antes de realizar la entrega de trabajo se realiza una refactorización del código dando como resultado el siguiente diagrama de paquetes.



- Activities : Contiene la actividad que gestiona la aplicación.
- Model : Contiene las clases referentes a la gestión de los datos, creación, consulta, actualización. Incluye la gestión de base de datos mediante contract y un communicator.

+ Utils : Dado que JSoup procesa datos y agazineAsync captura los datos, los hemos introducido en este subpaquete.

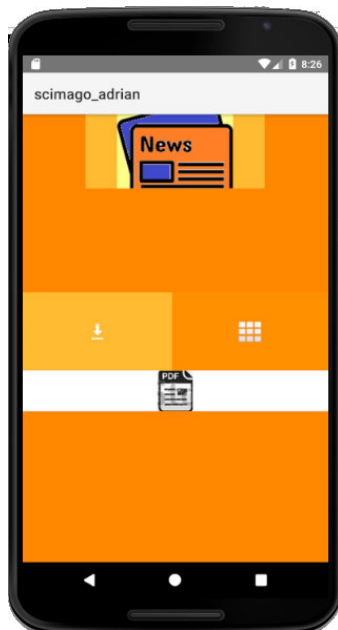


Scimago App

- Views ; Contiene todo lo referente a la visualización de los datos por pantalla. Además de por pantalla incluye el reporting en formato pdf.



11. PANTALLAS PRINCIPALES DE USO



Pantalla principal desde la que podemos realizar la gestión de los datos.

Obtención de los datos actualizados.

Visualización del listado de datos obtenidos.

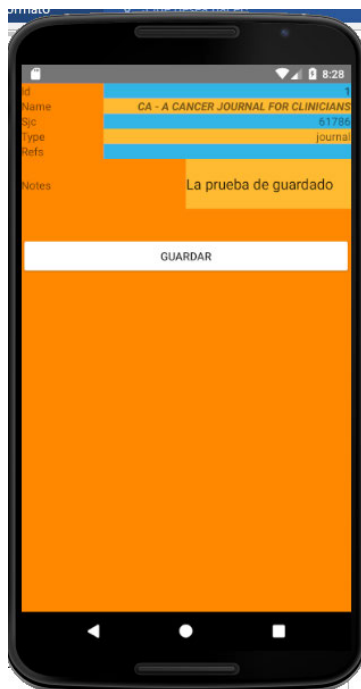
Generación de Reportings.



Scimago App



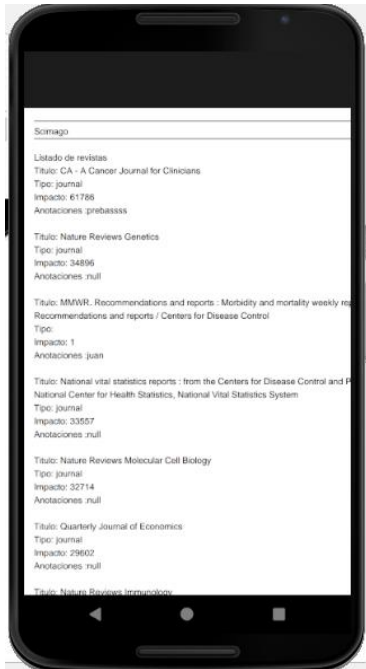
Listado de datos obtenidos desde la web Scimago y que están almacenados en nuestra base de datos.



Consulta al detalle de uno de los elementos obtenidos de la web con opción a realizar una anotación personal del elemento.



Scimago App



Reporting obtenido de los datos almacenados en la base de datos del terminal tras haber obtenido los datos de la web scimago.



12. LÍNEAS DE FUTURO

Como líneas de trabajo futuras para la mejora de la aplicación realizada expongo las siguientes líneas:

- Primero de todo cambiar el método para obtener los datos.
- La técnica de webscrapping tiene el grave problema que va acoplado a una vista.
- Si la visualización de la web de la que obtenemos los datos cambia, estamos expuestos a la obtención de dato erróneos.
- Es por eso completamente necesario adaptar esto a una consulta de ApiRest. Con ello finalizarían los problemas si hubiera cambios en la web.
- Posibilidad de incorporar imágenes a cada uno de los registros y almacenarlos en la base de datos.
- Posibilidad de consulta y visualización de la revista completa y almacenaje en el terminal.
- Posibilidad de detectar la ubicación del usuario y en función de la ubicación (por continentes) mostrar el listado de revistas del continente en el que se ubica el usuario.



13. LECCIONES APRENDIDAS Y CONCLUSIONES

Con la realización de este TFM se han cumplido todos los objetivos generales y particulares que se marcaron al inicio:

- Se han implementado herramientas necesarias para simplificar la labor de consulta de revistas de la web scimago.
- Se han obtenido los datos de la web sin la tecnología API REST usando otra técnica distinta.
- Permitir al usuario introducir anotaciones propias, visualizar los datos en forma de lista, visualizar los datos en forma de detalle, obtención de listado en formato Pdf.
- Interfaz amigable y fácil uso para cualquier tipo de usuario.

El proyecto ha dado como resultado una aplicación que ayuda a la gestión y consulta de las revistas de la web scimago.

El uso de la aplicación se ha conseguido para que sea sencillo con una interfaz simple. Por ello es una herramienta básica para la consulta de los datos de la web scimago.

A nivel personal, la realización de este Trabajo fin de Master me ha permitido conocer como planificar, gestionar y desarrollar una aplicación real, incluyendo todas sus fases. Desde la especificación del Sprint BackLog, diseño de la base de datos y la interfaz de usuario, como realizar las estimaciones de tiempo para el desarrollado de cada una de las tareas realizadas.



Ha servido para profundizar los conocimientos en el campo de la programación y aprender a usar y utilizar entornos de programación como es Android Studio.

También se ha ahondado en la investigación sobre técnicas distintas a las actuales (APIS) para la obtención de datos, lo que nos ha hecho darnos cuenta de la importancia de una API REST y la potencia que supone.



14. BIBLIOGRAFIA

- Cómo programar en JAVA
 - Javier Ditel
- Diseño apps para móviles
 - Javier Sion, José Vittoe
- Introducción a SQLite
 - Daniel Ponsoda Moniel.
- Metodos Ágiles y Scrum
 - Alonso Álvarez Garcia, Rafael de las Heras del Dedo
- WebScrapping With Java
 - Ryan Mitchell

ENLACES WEB

- Scrpping con java

<https://jarroba.com/scrapping-java-jsoup-ejemplos/>

- JSoup

<https://jsoup.org>

- *Document Android developers*

<https://developer.android.com/reference/org/w3c/dom/Document>

- *StackOverFlow*

<https://es.StackOverFlow.com/>

- *Tutorial de base de datos Sqlite para aplicaciones Anroid*

<http://www.hermosaprogramacion.com/2014/10/android-sqlite-bases-de-datos/>

- SqLiet Docuentaton

<https://www.sqlite.org/doclist.html>