# Memoria del proyecto.

A continuación se van a detallar los siguientes puntos a tratar.

Contenido

[Memoria del proyecto. 1](#_Toc380544341)

[Enunciado del problema 2](#_Toc380544342)

[Realización y explicación de la implementación punto por punto. 2](#_Toc380544343)

[Punto 1 2](#_Toc380544344)

[Punto 2 3](#_Toc380544345)

[Punto 3 3](#_Toc380544346)

[Punto 4 3](#_Toc380544347)

[Cosas a tener en cuenta 4](#_Toc380544348)

[Objetivos conseguidos y no conseguidos 4](#_Toc380544349)

[Conseguidos 4](#_Toc380544350)

[No conseguidos 4](#_Toc380544351)

[Valoración personal 5](#_Toc380544352)

[Glosario de enlaces. 6](#_Toc380544353)

## Enunciado del problema

Desarrollar una aplicación Android que trate la cuenta de usuario de dropbox como una biblioteca de libros remota, de forma que:

1. Exista una pantalla inicial de login, donde el usuario pueda introducir sus credenciales para tener acceso a su cuenta Dropbox. (https://www.dropbox.com/)
2. Una vez introducidos los credenciales, se mostrarán en pantalla todos los libros que el usuario disponga en su cuenta Dropbox en forma de biblioteca. Esto es, todos los ficheros que tengan extensión .epub Para representar cada libro, puede usarse un icono genérico).
3. Dicha pantalla tendrá un menú desplegable con dos opciones, una de ellas ordenará la lista de libros por el nombre del archivo y la otra por fecha de creación. (Idealmente nos gustaría ordenar por el título del libro en vez del nombre del fichero)
4. Si el usuario clickea dos veces sobre el icono genérico que representa cada libro, la aplicación mostrará la portada del libro clickeado.

## Realización y explicación de la implementación punto por punto.

### Punto 1

“Exista una pantalla inicial de login, donde el usuario pueda introducir sus credenciales para tener acceso a su cuenta Dropbox.”

Para la realización de este punto tuve que realizar unos ajustes debido al problema que detalle en el fichero de problemas encontrados (problema 1). Este problema me obligó a dejar un único botón que permita enlazar con Dropbox. Al pulsarlo se abre el navegador de internet del móvil y se conecta con Dropbox, donde se piden los credenciales de su cuenta y también una serie de permisos para que la aplicación pueda acceder a la cuenta de Dropbox. En nuestro caso será un permiso para tratar con los ficheros de tipo libro electrónico, entre los que se encuentra nuestra extensión “.epub”

Para poder tener acceso a Dropbox, hubo que crear una aplicación en la página de Dropbox (enlace 1) para poder crear los permisos para la aplicación (enlace 2) y de este modo obtener un usuario y una clave que hay que introducir en la aplicación, para que al conectar con Dropbox no haya problemas.

Una vez se tiene acceso a la cuenta de Dropbox se procederá a conseguir el sistema de ficheros (DbxFileInfo). Con este sistema tendremos acceso a todos los directorios y ficheros de esa cuenta.

Hay un detalle importante a tener en cuenta en este punto. Se ha agregado una actualización automática (por medio del método syncNowAndWait) que actualiza y carga todo el sistema de ficheros cuando se produce un cambio en dicho sistema (se comprueba mediante el método hasSynced, para más información consultar la documentación de la Sync API for Android, enlace 3).

Esta actualización ocurre por dos motivos, has hecho un login (ya sea por primera instalación o tras un logout) o se ha producido un cambio en el sistema de ficheros de Dropbox. En ambos casos se lleva a cabo esta actualización y carga del sistema de ficheros.

Esto ocasiona un retardo en la aplicación debido a que sin ese sistema de ficheros no se puede realizar la búsqueda y, por ende, la aplicación no funciona. El retardo puede llegar a ser de varios minutos, aunque nunca ha superado los 3 en mis pruebas.

Por último en este puntos se han implementado dos botones mas, uno para hacer logout y otro para acceder a la siguiente activity, que es la que tiene la mayor parte de la funcionalidad.

### Punto 2

“Una vez introducidos los credenciales, se mostrarán en pantalla todos los libros que el usuario disponga en su cuenta Dropbox en forma de biblioteca. Esto es, todos los ficheros que tengan extensión .epub (Para representar cada libro, puede usarse un icono genérico).”

Se ha implementado una búsqueda iterativa que recorre todos los directorios y agrega a una lista de objetos “BdBook”, creada por mí para contener la información necesaria.

Tras esta búsqueda se procede a insertar los elementos en una ListView. He elegido una ExpandableListView para ello. Una vez rellena la lista se podrá ver en el orden el que fueron encontrados.

### Punto 3

“Dicha pantalla tendrá un menú desplegable con dos opciones, una de ellas ordenará la lista de libros por el nombre del archivo y la otra por fecha de creación. (Idealmente nos gustaría ordenar por el título del libro en vez del nombre del fichero)”

Como se pide se ha implementado un menú desplegable con dos opciones, ordenar por fecha de subida a Dropbox o por Nombre del fichero.

Para la ordenación se ha decidido usar el método “sort” de la clase “Collections” presente en la librería de java.util (más información en enlace 4). Como este método requiere el compareTo para realizar el ordenamiento se ha implementado un compareTo en la clase BdBook para cuando se realice la ordenación por fecha y se ha creado un nuevo comparator en el propio método sort para cuando se haga por nombre de fichero.

### Punto 4

“Si el usuario clickea dos veces sobre el icono genérico que representa cada libro, la aplicación mostrará la portada del libro clickeado.”

Una vez mostrada la lista, si se selecciona un elemento se procede a la descarga y exposición de la portada del libro, en caso de que tenga. Para ello se ha elegido un “Pop up” o ventana emergente que contiene la imagen y un botón para cuando se quiera cerrar dicha ventana. Mientras se carga la imagen se muestra otro mensaje, mediante otra ventana para informar de lo que está pasando.

## Cosas a tener en cuenta

Se ha decidido, por modularización y comodidad, tener todos los métodos y clases que traten con el API de Dropbox y con la librería de “epublib” en un mismo fichero llamado “TestManager”.

En él se pueden encontrar varias clases privadas, dentro de la clase principal, así como intefaces para poder implementar métodos en una activity y evitar tener que usar variables estáticas.

Además cada una de las clases privadas mencionadas anteriormente se han implementado con la clase “AyncTask” (enlace 5) para poder realizar tareas que paralizarían la ejecución del hilo principal, y hasta podrían llegar a ocasionar un fallo por sobre explotación del mismo.

Estas clases se corresponden con las tareas de:

1. Comprobar posibles actualizaciones y recuperar el sistema de ficheros (punto 1).
2. Realizar la búsqueda de los libros en el sistema de ficheros (punto 2)
3. Descargar el icono con la portada y mostrarlo en pantalla (punto 4).

## Objetivos conseguidos y no conseguidos

### Conseguidos

1. El acceso al sistema Dropbox.
2. Búsqueda de los ficheros con extensión .epub
3. Menú desplegable con opciones de ordenación por fecha y por nombre de archivo.
4. Descargar y exposición de la portada del libro, en caso de tenerla.

### No conseguidos

1. Login con credenciales. Estos se ponen directamente en Dropbox vía navegador web.
2. Ordenación según el título del libro. No he conseguido recuperar el título del libro.
3. Doble click para mostrar la imagen de la portada. No he conseguido detectar el doble clickeo.

## Valoración personal

En cuanto a mi opinión personal sobre esta aplicación, estoy bastante satisfecho con el trabajo realizado. He conseguido implementar toda la funcionalidad principal de la aplicación y casi toda la secundaria a pesar de los hándicaps que tenía, como el hecho de desconocer por completo el API de Dropbox así como la librería epublib y la falta de tiempo.

Hay cosas que mejorar, y que me hubiera gustado mejorar como una mejor interfaz, un método para actualizar el directorio de ficheros que no consuma tanto tiempo cuando se hace por primera vez o cuando hay un cambio en dicho sistema o la consecución de los objetivos marcados como no conseguidos.

A pesar de todo pienso que la aplicación refleja con exactitud mis conocimientos en Android y mi forma de programar.

## Glosario de enlaces.

1. https://www.dropbox.com/developers/apps
2. https://www.dropbox.com/developers/apps/info/4rsxxjd1d0sczfu
3. https://www.dropbox.com/developers/sync/docs/android#com.dropbox.sync.android.DbxAccountManager
4. http://developer.android.com/reference/java/util/Collections.html
5. http://developer.android.com/reference/android/os/AsyncTask.html