

Aplicaciones Móviles-WeLit

Caballero Alexis

Mayer Javier

Presentación

La aplicación, bautizada como WeLit, tiene como propósito servir a los lectores, fungiendo como intermediario entre ellos para el intercambio de libros (en formato físico).

Cada usuario contiene una lista personalizable de los libros a su disposición que esté dispuesto a intercambiar con usuarios de su zona. Para esto, en el buscador, el usuario puede indicar la distancia máxima a buscar entre él y los demás (medido en kilómetros).

Por último, la aplicación ofrece un chat entre los usuarios que estén interesados en el intercambio. Desde allí se podrá fijar el punto de encuentro para el intercambio exitoso.

Alcance

Más allá de aprobar la cursada, se ha diseñado la aplicación con dos objetivos aparte.

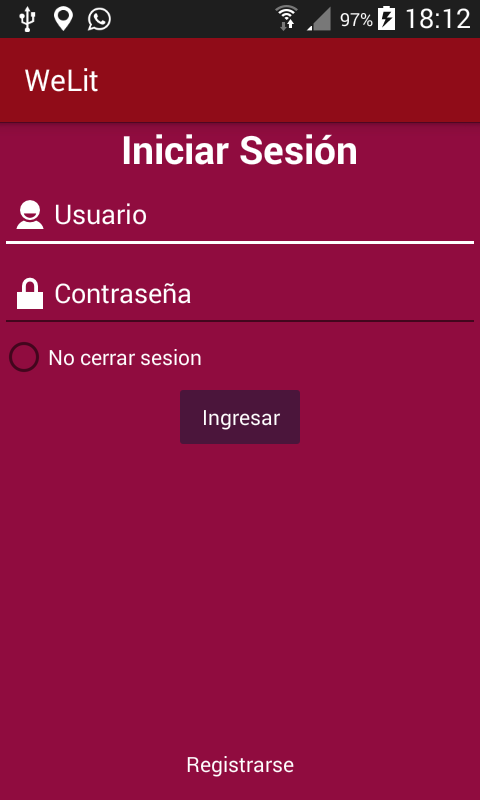
El primero es la creación de una aplicación totalmente funcional y que encaje con el nivel de calidad de un producto de la Store. Esto significa que, finalizado los retoques necesarios, la intención será el de publicar esta aplicación.

El segundo objetivo; crear una base para futuras aplicaciones. Es por esto que se ha recurrido a funcionalidades tan variadas como lo es la mensajería, cámara, uso del gps y la implementación de una base de datos. Con la reutilización de códigos se busca a futuro desarrollar nuevas aplicaciones por nuestra propia cuenta.

Desarrollo

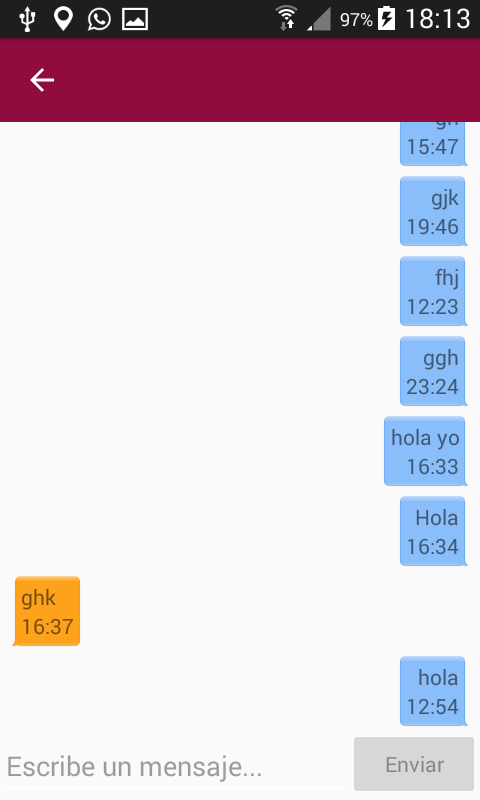
A la hora del desarrollo, hemos optado por construir un sistema de mensajería y construir alrededor de este el resto de las funciones necesarias.

Por esto, hemos iniciado diseñando el Login, la primera actividad que verá el usuario. El primero paso fue comprobar que los campos “Usuario” y “Contraseña” recibieran los valores apropiadamente (como pudimos comprobar imprimiéndolos en pantalla). Una vez allí, utilizamos SharedPreferences para guardar la cuenta y evitar que tenga que loguearse cada vez que inicie la aplicación.

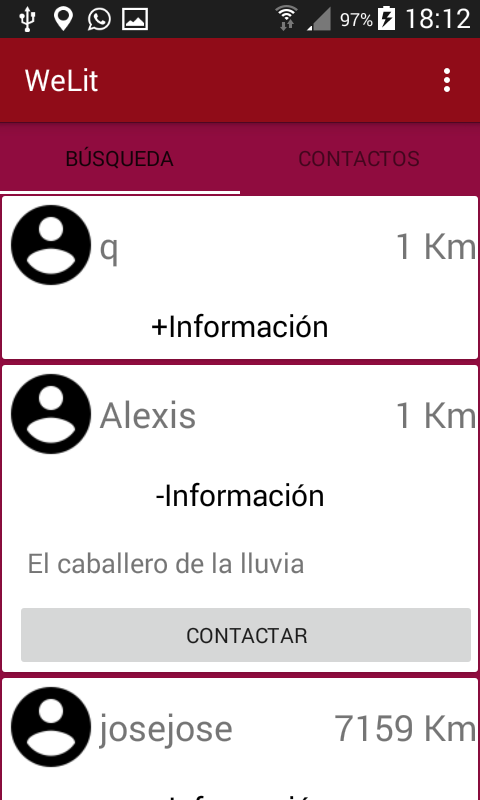


Para terminar con el inicio de sesión, nos hemos hecho con un servidor gratuito (<https://www.000webhost.com/>) para acceder a una base de datos remota en el que comprobar si los usuarios existen. Para lograr esto, archivos php retornan archivos json que son accedidos por la aplicación para comprobar la existencia de dichos usuarios. Este sistema es implementado cada vez que se busca acceder a la base de datos en el resto de las actividades.

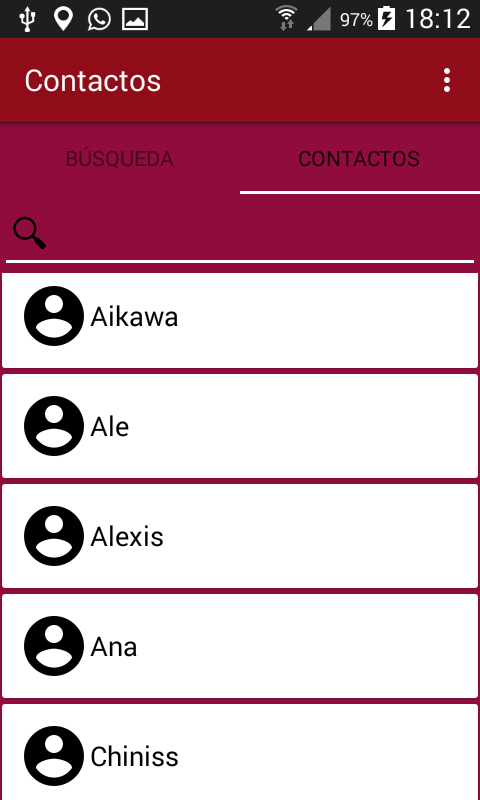
Ya con usuarios funcionales, partimos hacia el diseño de la mensajería, el trabajo más pesado de la aplicación. No solo debíamos retornar todos los mensajes en la ventana de mensajería, sino que debían ser ordenados por fecha, distinguir usuario de emisor y actualizarse cada nuevo mensaje. Además, notificarle al usuario ante el contacto (realizado más tarde con FireBase). Fue el sector que más tiempo ocupó durante el desarrollo y, a la vez, el más satisfactorio.



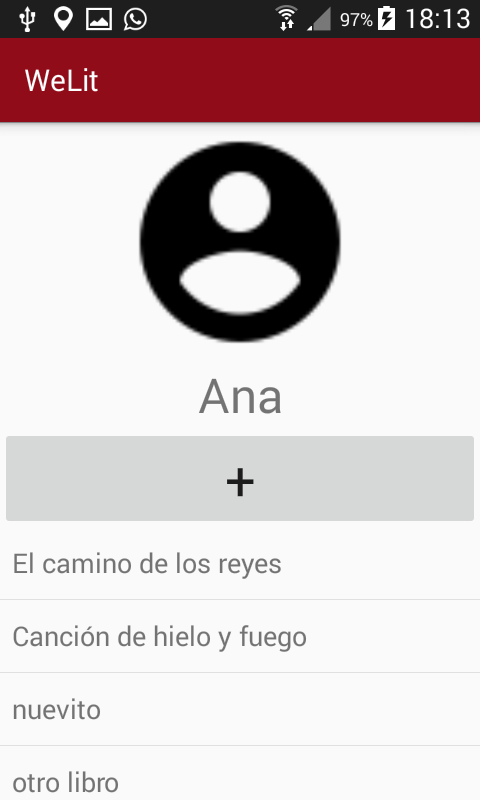
Hecho la mensajería, el usuario debía de poder localizar a los demás usuarios para iniciar de forma orgánica la toma de contacto. Al principio visualizamos todos los usuarios sin ningún tipo de filtro, listándolos en un ReclylcerView para implementar el maestro detalle. Ya funcionando, comenzamos a retocar para una mayor versatilidad. Desde mostrar solo aquellos usuarios con libros y omitir al propio usuario en la búsqueda, llegamos a la implementación del GPS para calcular la distancia entre el usuario y los demás. Para esto, hemos retocado la actividad Login para que, una vez accedido, se guarde/envíe datos de la longitud y latitud. Android ofrece una función para calcular la distancia con estos datos por lo que no hubo mayor inconveniente.



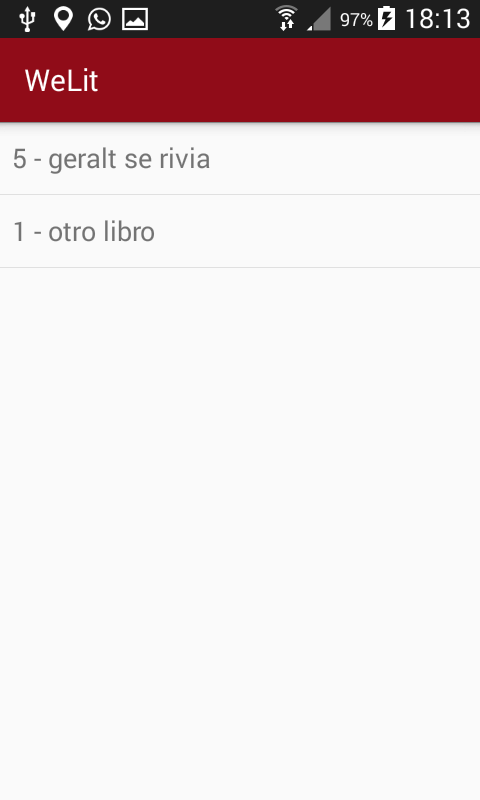
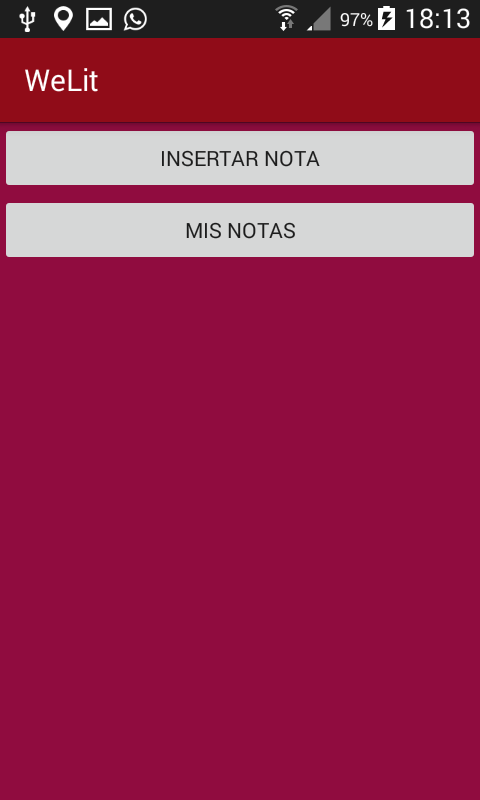
En la actividad principal, utilizamos dos fragments, uno para la búsqueda mencionada anteriormente (Búsqueda), la otra para guardar aquellos usuarios con los que se contactó (Contactos). En este fragment en particular, hemos provisto de un filtro para facilitar la búsqueda de los contactos.



Por último, la actividad perfil (anteriormente un tercer fragment) fue creada con la intención de editar la lista de libros disponibles y modificar el avatar por medio de la cámara.

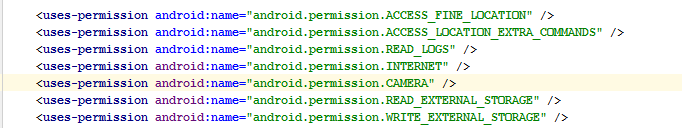


A modo de cumplir todas las demandas del trabajo, hemos integrado también implementando SQLite, un anotador para los en el que uno esté interesado en poseer a modo de recordatorio. Se puede anotar estos libros en orden de prioridad con un segundo valor.

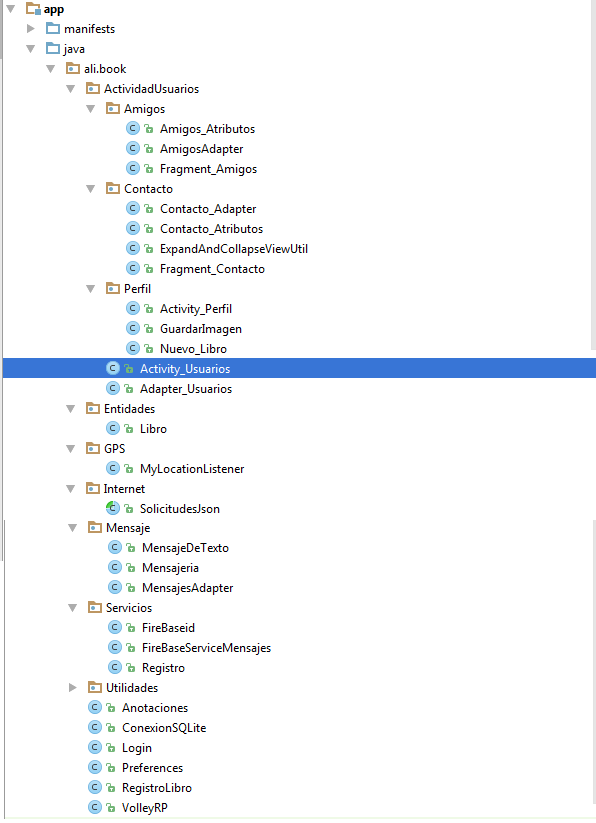


Ya todo funcional, hemos retocado el diseño y modificado el ícono para dar un aspecto más profesional y agradable a la vista.

Permisos



Estructura



Mejoras a futuro

Por mucho que se haya cubiertos las expectativas del trabajo, como aplicación independiente aún cuenta con factores a mejores (que serán tratadas en un futuro cercano).Estos son, divididos por apartados:

Login:

* Recuperar contraseña: Un botón que, como tal, de paso a un proceso para recuperar la cuenta.

Búsqueda:

* Filtro: Poder modificar el máximo de distancia en el que buscará a los demás usuarios.
* Orden: Listar los cardview en orden, de menor a mayor distancia.

Contactos:

* Orden de mensaje: Deberían de ordenarse acorde al último mensaje enviado/recibido.
* Último mensaje: Junto al avatar y nombre, debería ilustrarse el último mensaje de la conversación.

Perfil:

* Avatar dinámico: Además de modificarse vía cámara, se deberá poder insertar imágenes ya almacenadas en el dispositivo.
* Listado de libros: Además de agregar, aún no se implementó la eliminación o la edición. Opcionalmente, podríamos incluir descripción y recomendación personal del usuario.
* Editar: La posibilidad de cambiar datos personales como el nombre.

Conclusión

En conclusión, hemos avanzado gratamente en la aplicación de los conocimientos acordes al trabajo. En pocos meses hemos adquirido experiencia suficiente desde un conocimiento casi nulo a la creación de la aplicación aquí descrita.

La versatilidad que permiten las aplicaciones móviles así como el ingenio detrás de las ideas es un atractivo para sumergirse aún más en la materia.