

Proyecto final de curso

Android: Fundamentos de Programación

(julio-noviembre 2014)

Nombre de la aplicación: *Trackers*

Autor: *Garcés Solís* , Pablo

Qué hace la aplicación:

Aplicación donde se almacenan varios trackers de seguimiento de paquetes de correos, para comprobar si el estado del paquete ha tenido alguna variación desde la última vez que lo comprobamos.

Licencia:

Autorizo la difusión del código fuente con fines educativos siempre que se haga referencia al autor bajo los términos generales de la licencia "[Academic Free License v.3.0](#)".

A destacar:

- Hay aplicaciones que también te dan esta información, pero no usan notificaciones para el cambio de estado ni se pueden comprobar más de un tracker a la vez.
- Uso de tabs y fragments para poder usar la aplicación, además usamos listViews para los trackers
- Guarda en sqlLite los diferentes estados de los envíos asociados al tracker para compararlos en cada comprobación.
- Mejoras: Notificaciones y mejor gestión de la información de los trackers, además de poder ver todos los estados de un tracker.
- La aplicación es muy útil para usuarios que compren por ebay, mercado asiático o cualquier tipo de compra que use correos para el envío
- Los estados de cada tracker se obtienen con una llamada a un web services php que recoge los datos de un xml y los guarda en bd si es necesario obteniendo la información mediante DOM.

Cómo lo hace:

- La aplicación tiene un entorno visual con 2 tabs, una para comprobar los estados y otra para añadir el tracker con una descripción. Ambas tabs usan un listView, para crear estos listView hemos usado 2 adapters, para poder trabajar mejor con la información que verá al usuario. Al final encontramos un botón en cada pestaña para Buscar Cambios o Guardar Tracker.
- En la primera de ellas se muestran todos los trackers almacenados en base de datos, con su último estado conocido y su fecha, además del tracker y la descripción que se le dio al añadirlo. Al final del todo tiene un botón para

actualizar la información, esta información es obtenida mediante llamadas a un web services en PHP y recogiendo los datos en formato XML mediante DOM, una vez tenemos la información en el formato deseado, procedemos a guardarlo en una base de datos SQLite si ese estado no está ya almacenado en nuestra tabla.

- En la segunda de las tabs, tenemos un listView con los trackers almacenados y el resto de campos para añadir nuevos, para añadir un tracker solo tenemos que saber el identificador que nos han dado de correos y la descripción que queremos darle. Al final del todo está situado el botón para guardar esta información. Una vez guardemos esta información podemos pasar a la primera tab para comprobar cual es la situación del nuevo tracker introducido.
- En las preferencias tenemos varias opciones:

1) Cuantos trackers queremos seguir

Por defecto está puesto a 5, pero se pueden elegir más, el usuario debe tener en cuenta que cuanto más trackers haya almacenados más se tardará en cada llamada de actualización y/o salvado de datos.

Siguiente versión:

2) Activar notificaciones

Se realiza lanzando un Service cuando se enciende el teléfono, este service va comprobando si en los trackers almacenados hay algún nuevo estado que notificar al usuario. Esta empezado pero preferí dedicar tiempo en dejar bien lo que tenía.

- 3) Cada cuanto tiempo queremos buscar actualizaciones

El usuario escoge cada cuantas horas debe la aplicación de buscar actualizaciones en sus trackers. Esto consistiría en el uso de Bind y Timer, lanzando al iniciar el teléfono un servicio, si el usuario cambia en preferencias el tiempo, recuperar el servicio que está corriendo y modificar el tiempo de comprobación. Esto lo hará siempre que las notificaciones estén activas, también comprobado.

- Base de datos: Tiene una sola tabla muy sencilla donde se almacenan los trackers, al añadir el usuario se guarda una primera línea para saber cuando fue añadida a la aplicación, cuando ya tiene otros estados se agregan en la tabla para poder monitorizarlos. En un principio quería asociarlo al teléfono, pero he leído que ya no se puede obtener el teléfono en los modelos más nuevos, así que lo asocio por imei asociado al teléfono.
- **Web services:** El webServices es una página en PHP que devuelve un xml a leer, en este caso he usado DOM. He decidido hacerlo uno a uno, a pesar de ser más costoso pero daba mejor resultado que todos en una petición. He usado un dominio que tenía sin usar, la dirección es: http://architecturefirm.nixiweb.com/autonomos/proyecto_correos.php?trackers=XXXXXXXX
- **Trackers de ejemplo:** Estos están entregados pero no tengo ninguno en curso

RD118421735CN

RC865658266HK

RF236056259SG

Captura de pantallas:

