

Aplicación para la Gestión de Pozos de Riego - MisPozosDeRiego

Proyecto final de curso: Fundamentos de Programación

(septiembre-diciembre 2016)

Nombre de la aplicación: *MisPozosDeRiego*

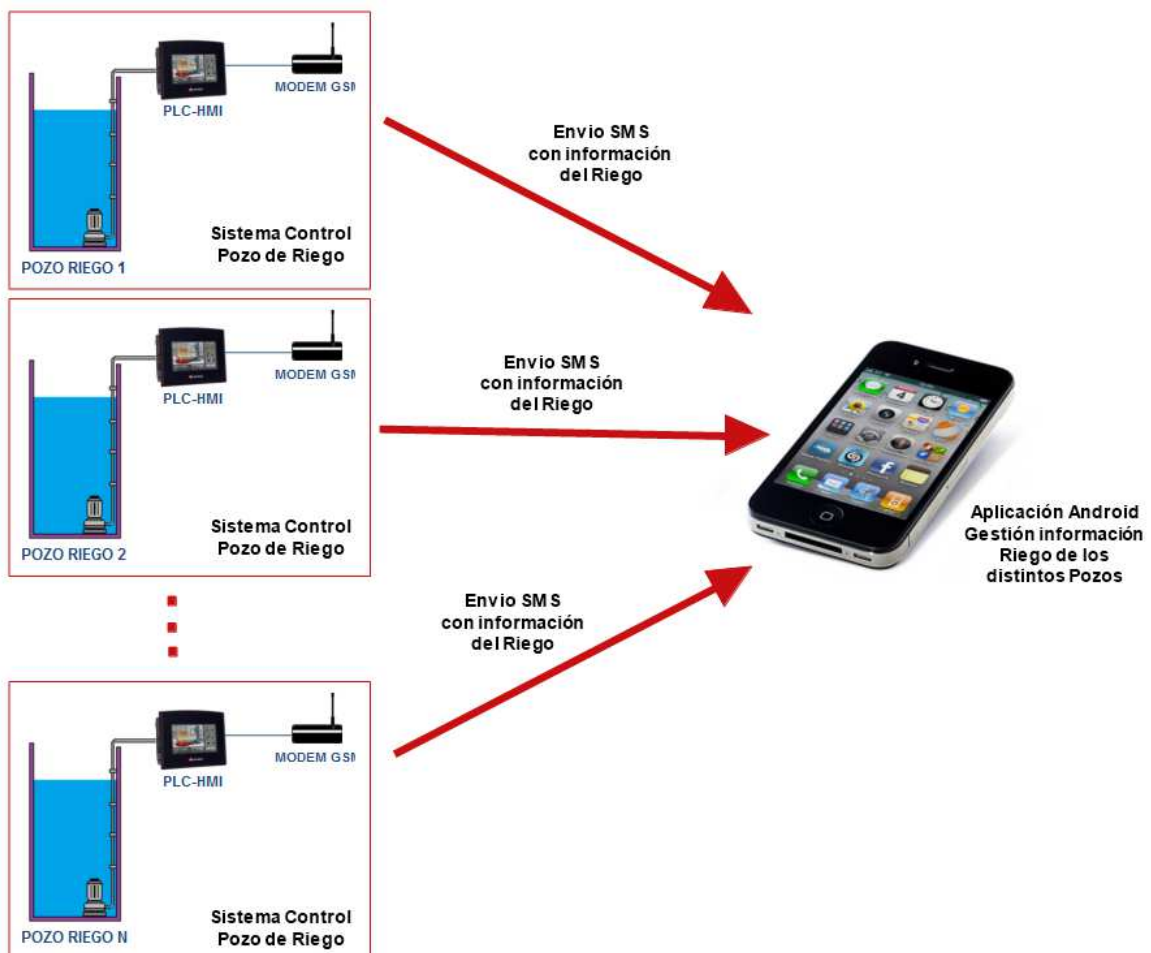
Autor: *Ramos Ramón, Vte. Raúl*

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](#)".

Qué hace la aplicación:

Se trata de implementar una aplicación en Android que gestione la información acerca del estado del proceso de riego, que es recibida vía mensajes SMS desde los distintos Pozos de Riego, y mostrarnos esta información en un dispositivo móvil.



Por ejemplo: Indicación del inicio y fin del riego, sectores que han sido regados, caudal de agua consumido, potencia eléctrica consumida. Además de alarmas y avisos de avería y/o mal funcionamiento del sistema.

La aplicación se plantea más como un **proyecto a largo plazo**, en vistas de ser presentado como un de Proyecto Fin de Curso. Pero se presentarán las funcionalidades que se puedan para el presente Proyecto de cierre de este primer módulo del curso.

Alguna de las funcionalidades que tendrá la aplicación final:

- Se crearía una interfaz similar a la de *Mis Lugares*, en lo que respecta a la introducción y muestra de los datos de los diferentes Pozos de Riego (Nombre del Pozo, localización geográfica, número teléfono módem SMS foto de pozo, ubicación y otras características de la instalación) **[Implementado en este proyecto]**
- Implementaría toda la gestión de los SMS recibidos desde los distintos Pozos de Riego, para mostrar la información del estado del pozo obtenida a partir de estos mensajes en forma de un listado (mediante un Recycler View) **[Implementado parcialmente en este proyecto]**
- Toda esta información sera almacenada en diferentes tablas de una BD Relacional SQLite. **[Implementado parcialmente en este proyecto]**
- Como alternativa de visualización, se mostraría en un mapa de Google la ubicación de los distintos pozos haciendo uso de marcadores, mostrándose también la información asociada al último riego efectuado y/o si están en alarma.
- Mostrar al usuario históricos de los riego y/o históricos de las alarmas de los distintos pozos, a partir la información recopilada a través de los SMS recibidos.
- .Posibilidad de interactuar con los controladores de los pozos mediante envío de SMS, para poder modificar la Programaciones, o bien para solicitar información sobre el estado actual del pozo, sin tene que esperar a que se produzca un evento para obtenerla.
- Posibilidad de almacenar los datos en un servidor externo con Apache, PHP y MySQL, implementando un servicio para obtener y/o enviar esta información
- Posibilidad de gestionar operarios para la resolución de averías en los pozos.
- Además de otras posibilidades que se irán comentando en su momento.

Evidentemente, para este Proyecto que nos atañe se han abordado solo unos pocos puntos, y se dejan su mayoría para ser tratados y/o completados en el Proyecto Final.

Para el Proyecto que actualmente se presenta, pasaré a continuación a redactar lo que se ha podido implementar.

A destacar:

- Gestión de permisos sobre los SMS (RECEIVE_SMS, READ_SMS) y permiso de lectura en el almacenamiento externo (para poder incluir fotos de los pozos obtenidas de la Galería)
- Almacenamiento de los datos de los Pozos y de la información de su estado actual en sendas tablas de una *BD Relacional SQLite*. Se han implementado los métodos necesarios para la inserción, borrado, modificación y manipulación de registros en esta tablas.
- Uso de receptor de anuncios para los SMS recibidos, de un Servicio para el tratamiento del mensaje SMS, y de una notificación para indicar la obtención de la información del estado del pozo.
- Uso de *Recycler Views*: uno para mostrar la lista con la información de estado actual de los pozos y acceder al detalle de la información, y otro para mostrar la lista de los pozos para su posterior visualización y edición de sus datos.
- Uso de menús, uso de iconos externos, uso de recurso de *strings* y *colors* para referenciar las cadenas o colores de texto utilizados en la aplicación.
- Se utiliza para la mayoría de las vistas un diseño basado en *material design*, con barra de acciones y un botón flotante, y cuyo contenido podrá desplazarse, a la vez que la barra de estado cambia de tamaño.

Cómo lo hace:

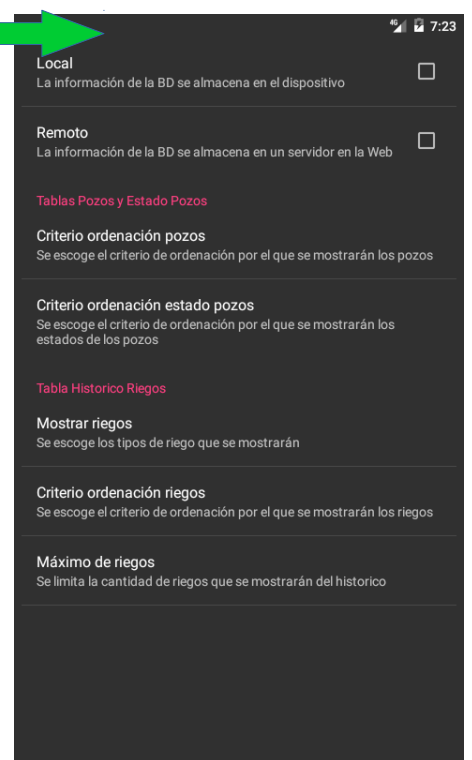
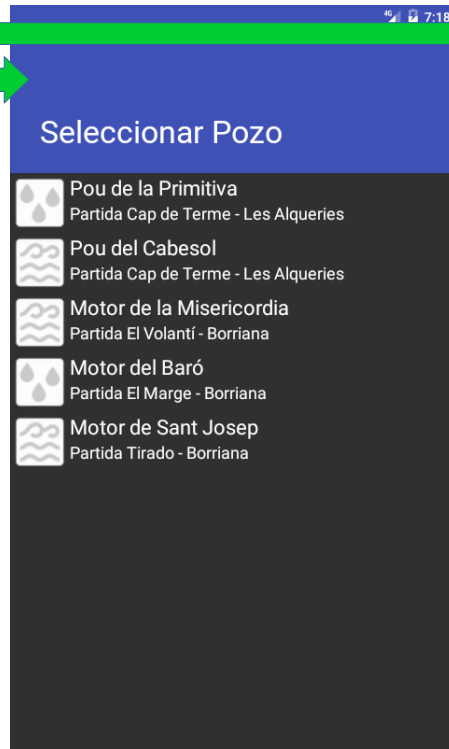
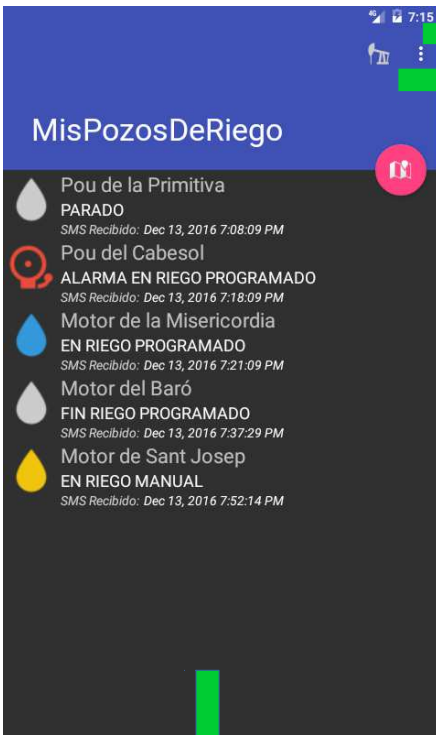
- Se ha implementado un **Receptor de Anuncios** que se encarga de recibir los SMS desde los distintos Pozos de Riego. Este receptor se encarga de extraer el número de teléfono y el cuerpo del mensaje con la información del estado actual del pozo, y almacenarlos en sendas cadenas de texto. Para posteriormente llamar a un **Servicio en segundo plano**, que se encargará de procesar estas cadenas de texto. En primer lugar, a partir del teléfono averiguará, consultando en la tabla correspondiente de la BD, bien de qué pozo se trata, o bien, si se trataba de un SMS que no tenía nada que ver con la gestión de los pozos. En segundo lugar, si el mensaje era de un pozo, a partir del cuerpo del mensaje extraerá la información relacionada con el estado actual de dicho pozo y procederá actualizar esta información. Por desgracia, por falta de tiempo, no he podido implementar el procesamiento del teléfono y del cuerpo del mensaje. Por lo tanto, sólo podemos ver como el receptor de anuncios extrae la información y llama al servicio, que se limita a mostrarla mediante un *Toast* por pantalla, y luego lanza una Notificación asociada a la recepción del SMS, también con esta información.
- En la clase **BDPozosDeRiegos** se ha implementado la BD Relaciona SQLite, y se han creado dos tablas de momento: **tb_pozos** y **tb_estadopozos**, que almacenan la información de los pozos y del estado actual de los pozos, respectivamente. Se han añadido registros a estas dos tablas para poder probar la aplicación. Las clases AlmacenPozosBD y AlmacenInfoEstadoBD implementan todas las operaciones a efectuar sobre los registros de estas tablas (añadir, eliminar, actualizar)
- Para el contenido de la **actividad principal (MainActivity)**, se ha utilizado un *Recycler View*, incrustado dentro de un *NestedScrollView*, donde se muestra un listado con la información actual del estado de todos los pozos. Para su creación se ha implementado la clase **AdaptadorEstadoPozosBD** que extrae la información desde las dos tablas. Se puede ver su aspecto en la figura 1.
- Volviendo a la actividad principal, desde su menú asociado, seleccionando el icono del pozo, podremos acceder a una actividad que mediante un *RecyclerView* nos da un listado de los pozos, para poder acceder a su visualización en detalle y a su posible edición. Todas estas actividades en concreto, toman como referente, el ejemplo de la aplicación **Mis Lugares**, ofreciéndonos casi todas sus posibilidades:
 - Se pueden añadir fotos del pozo desde la Galería, desde la Cámara.
 - Se puede mostrar la posición del pozo llamando a una aplicación externa de visualización de Mapas.
 - Se pueden modificar los datos de los pozos, hacer altas y/o eliminar pozos.
- Para hacerse una mejor idea, mirar el apartado final de las capturas de pantalla.

Evidentemente, se trata de un primer acercamiento al problema, y no he podido alcanzar alguno de los objetivos que me había propuesto en un principio. Sobre todo, el procesamiento del teléfono y el cuerpo del mensaje. Esta información es la que se utilizaría para actualizar los registros de la tabla **tb_estadopozos** que mostramos en este *Recycler View* de la actividad principal y en la actividad que nos muestra su detalle. También me hubiera gustado haber podido implementar la vista alternativa de la información del estado actual de los pozos sobre un mapa de Google. Por motivos de disponibilidad por mi trabajo, por falta de soltura en la programación en Java, que me han hecho avanzar muy lentamente, y por otras circunstancias, no he podido avanzar en su desarrollo todo lo que hubiera deseado. Esperemos que lo que he podido hacer sea suficiente.

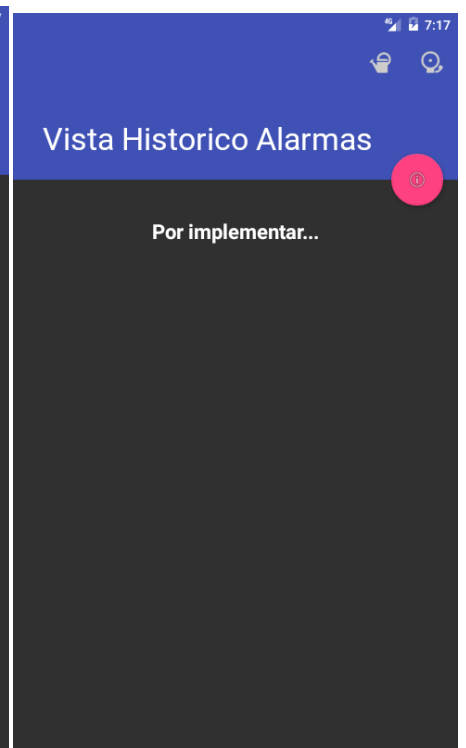
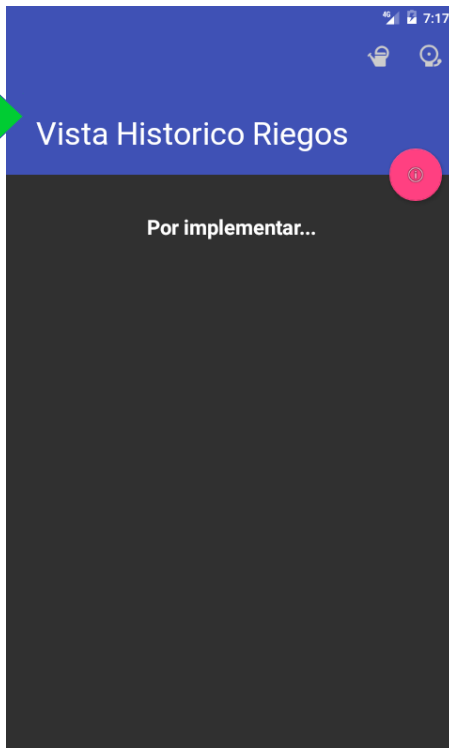
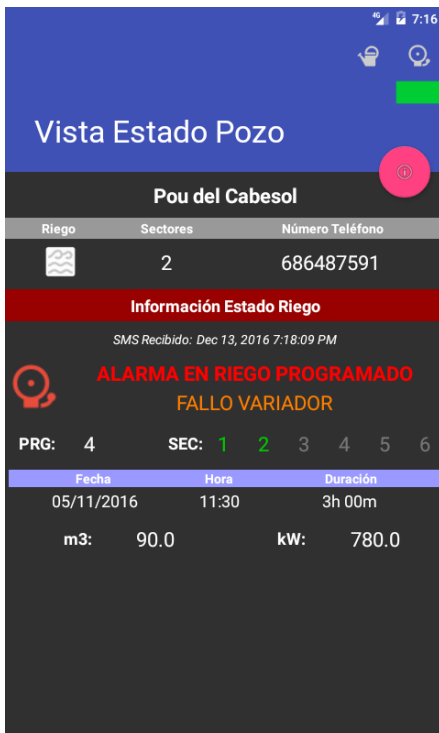
Durante el año se irá completando la aplicación pues como ya he dicho, mi intención es que sirva de Proyecto Fin de Curso.

Captura de pantallas:

Informacion estado actual pozo

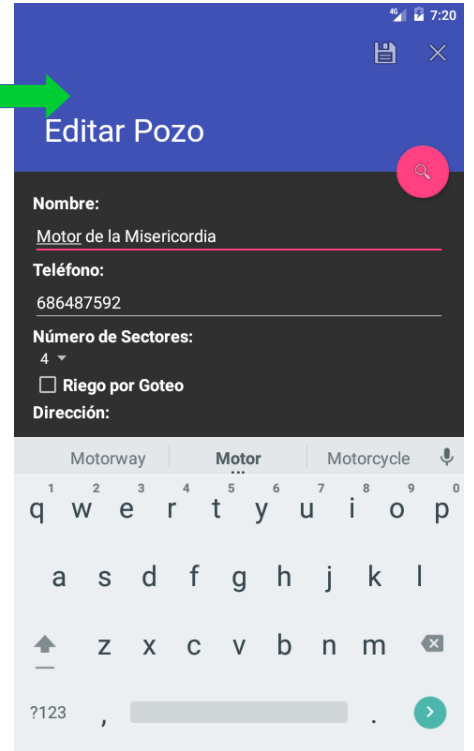
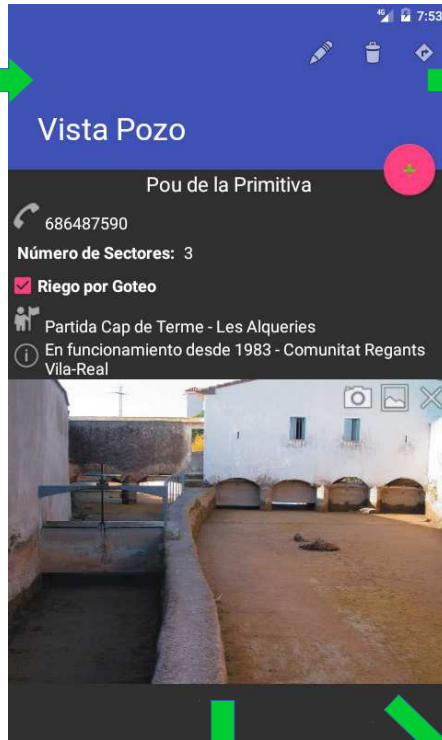
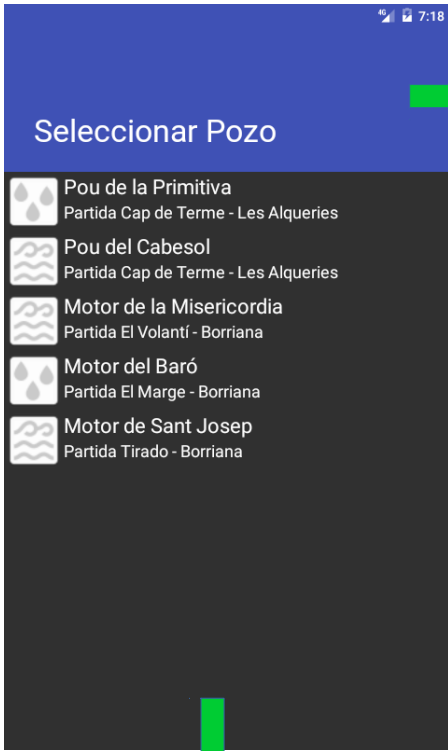


Informacion en detalle del estado actual pozo



Vista datos pozo

Modificar datos pozo



Alta pozo nuevo

