# PRÁCTICA FINAL ANDROID

## INTRODUCCIÓN

Esta aplicación nace de la necesidad de tener una manera rápida y fácil de obtener los datos de un sensor de temperatura y humedad que tengo conectado a una RaspberryPi, de ahí el nombre de la app: “Temperaberry”. La idea es mediante una aplicación sencilla y con un solo click, saber la temperatura y humedad que recoge el sensor de la rpi que, en este caso, está físicamente en un dormitorio de la vivienda. De esta forma podemos monitorizar bajo demanda y de manera sencilla los cambios de temperatura que va sufriendo la habitación.

El sensor está conectado a un puerto serie de la RaspberryPi, y a su vez la rpi corre un servidor web Apache sobre el puerto 80 sobre el que escucha las peticiones y devuelve los datos de temperatura y humedad en formato JSON. Por ejemplo, una petición HTTP al puerto 80 de la rpi podría devolver:

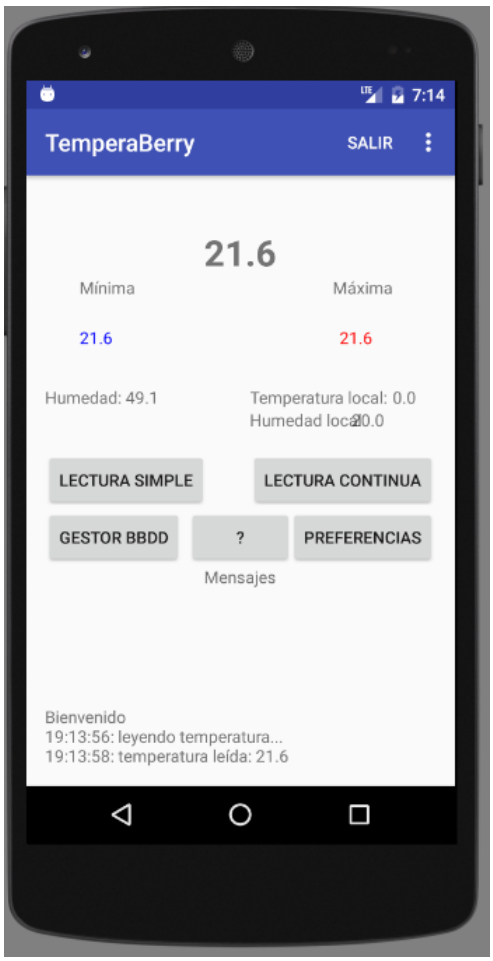
{"res":"OK","temp":"24.5","humi":"45.9"}

Esta resultado indica una lectura correcta del sensor que reporta 24.5 grados de temperatura y 45.9% de humedad relativa. La aplicación interpretará estos datos y los mostrará en pantalla. Adicionalmente, podrá guardar esta información en una base de datos interna.

## DISEÑO Y DESARROLLO

La idea principal se trata de tener una interfaz limpia y simple. Aquí, los datos más importantes a destacar son los valores que se recogen del sensor, por lo tanto esta información debe ser la más visible. Adicionalmente, es buena la idea de tener una pequeña porción de la pantalla donde se indique qué está haciendo la aplicación en un determinado momento, a modo de log. Tampoco deben faltar botones adicionales para settings, ayuda, etc.

El desarrollo de la app parte de una actividad principal MainActivity. Esta es la pantalla principal que muestra los datos del sensor en un instante dado del tiempo. Esta pantalla o actividad principal es la que muestra la información más importante: último valor de temperatura y humedad leídos, temperatura ambiente del dispositivo (si es compatible), acceso a las actividades de “Ayuda, Gestión de BBDD, Acerca de, y preferencias”:




Esta actividad presenta las siguientes vistas y botones

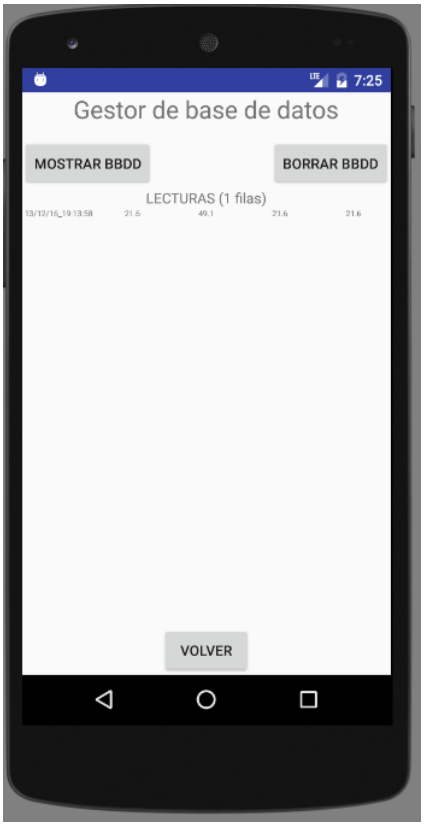
- Último valor leído del sensor (o doble guión si no se ha leído aún nada). Es un textview en la parte superior de la actividad, centrado y con la fuente en gris
- Valores mínimos y máximos de temperatura de las lecturas leídas hasta el momento. Son dos textview que se sitúan a la izquierda y derecha con colores azul y rojo, respectivamente. Si se detecta una nueva máxima o mínima se registrará en el textview correspondiente. Adicionalmente, una notificación toast aparacerá para notificar de dicho evento (se puede desactivar)
- En la fila central de esta actividad, se muestran el valor de humedad relativa leída del sensor, y la temperatura y humedad local del dispositivo (si éste posee dicho sensor)
- Justo debajo está la botonera principal. Un botón de “Lectura simple” se conectará al dispositivo y leerá los datos del sensor. La URL de conexión al dispositivo es configurable. El botón “Lectura continua” iniciará un proceso de lecturas consecutivas y automáticas cada cierto intervalo de tiempo (configurable). El botón “Gestor de BBDD” nos lleva a una nueva actividad donde podremos ver los valores que hemos leído en otras ocasiones con la app y/o borrar la bbdd para poder así limpiar y empezar lecturas nuevas. Cada vez que se lee un dato del sensor, éste se guarda automáticamente en la bbdd, aunque este comportamiento se

puede deshabilitar. El botón “?” nos llevará a la actividad de ayuda, que nos mostrará el texto pertinente. El último botón es el de “Preferencias”, donde podremos configurar varios aspectos de la app. Para terminar, un textview en la parte inferior refleja las acciones que va realizando la aplicación, a modo de log.

Una vez descrita la actividad principal, pasamos a ver las distintas actividades restantes:

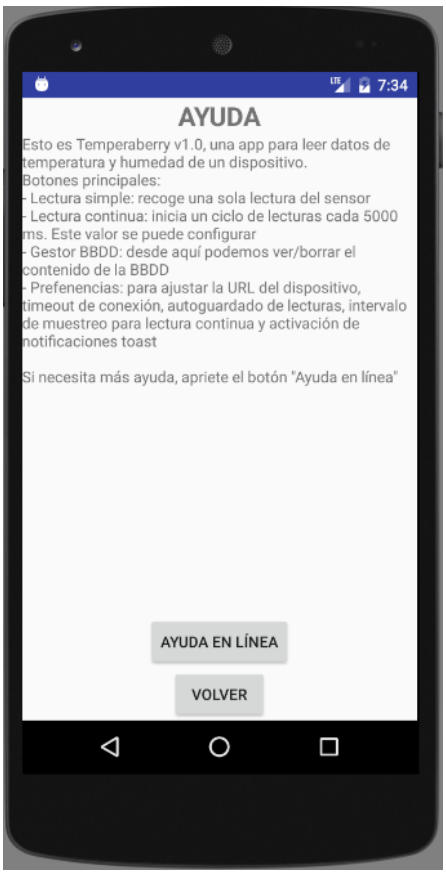
- Gestor BBDD.

  Podemos acceder a esta actividad desde la actividad principal presionando el botón “Gestor BBDD”, y tiene el siguiente aspecto:



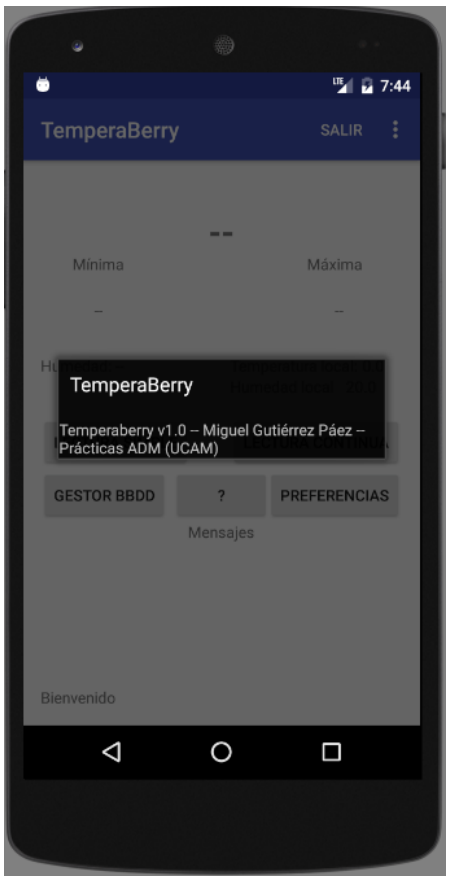

  Desde esta actividad podemos ver el contenido de la base de datos con las lecturas que se han guardado apretando el botón “Mostrar BBDD”, que volcará toda la información en el textview “Lecturas”, indicando el número de lecturas entre paréntesis. Otra opción que tenemos es la de borrar por completo toda la información de la bbdd para empezar lecturas nuevas si así lo deseamos. Esto lo haremos presionando el botón “Borrar BBDD” que nos una caja de diálogo para confirmar o cancelar dicha operación. Con el botón “Volver” podemos regresar a la actividad principal.

- “?” (botón de ayuda). Accesible desde la actividad principal con el botón “?”, esta activity mostrará un texto con la ayuda principal de la app, y un botón para acceder a más ayuda en línea:




- Preferencias. Podemos acceder a esta actividad con el botón “Preferencias”. Desde aquí podemos configurar los siguientes parámetros:

    - URL completa para conectar con el servidor que tiene conectado el sensor. Se especificará la URL con protocolo (normalmente http), ruta y puerto. Por defecto, la url es “http://quintoblog.com/temperaberry/temperaBerry.sh”
    - Timeout para la conexión hacia el servidor
    - Guardar lecturas automáticamente. Si se desactiva, las lecturas no se guardarán en la base de datos
    - Intervalo de polling: intervalo en ms para la lectura continua (por defecto 5000 ms)
    - Aviso de temperatura máx/min: activa/desactiva las notificaciones toast que avisan de nuevas temperaturas máximas y/o mínimas
- About. Esta actividad está accesible desde el actionbar extendido y botón “Acerca de”. Muestra una breve información de la aplicación.




## PROBLEMAS ENCONTRADOS

A continuación enumero los principales problemas que encontré durante el desarrollo de la aplicación:

- Librerías externas

  He necesitado usar librerías externas para determinados cometidos. Por ejemplo, necesitaba interpretar la salida JSON que devuelve el servidor al que se conecta la aplicación. De esto se deriva también que necesitaba funciones que de manaera asíncrona hicieran peticiones HTTP.

  Esto lo resolví con las librerías org.json, android-async-http y httpclient. Estas dos últimas las descargué de internet en forma de jar y las importé a Android Studio en forma de módulo. La librería de JSON es estándar y solo bastaba con hacer el import correspondiente.

- También necesitaba una manera de solucionar las peticiones HTTP periódicas en segundo plano cada cierto tiempo (botón “Lectura continua”). Leyendo y buscando por internet, averigué que para eso existía los “Handler” de Android. Básicamente es un framework de Android para manejar hilos de ejecución. Definiendo un objeto de la clase Handler y haciendo un override del método handleMessage dentro del propio objecto, puedo ya usar la función sendMessageDelayed para ejecutar una determinada operación cada cierto tiempo. Esto es lo que uso para hacer lecturas cada ciertos milisegundos (configurable, 5000 por defecto)

- Icono personalizado. Quería ponerle a la app un icono personalizado. Esto no fue difícil y buscando un poco de información por internet vi que solo se trataba de poner el icono deseado en la carpeta drawable, y declararlo en el xml de manifest.

## PUNTOS FUERTES Y DÉBILES

Como punto fuerte considero el hecho de usar características no aprendidas durante el curso, como el uso de la clase Handler para mandar mensajes y ejecutar operaciones periódicas, o el uso e importación de librerías externas buscando info de terceros. La practicidad de la aplicación también la considero un buen punto a favor, puesto que es una aplicación realmente usable y no solo sirve de práctica para la asignatura. Cierto es también que está diseñada para un entorno bastante específico, pero resulta realmente útil. El uso de sharedpreferences, lectura de ficheros raw y guardado de información en una base de datos también veo que son características imporantes.

Como puntos débiles, hay poca validación de datos y manejo de errores (lecturas vacías, URL no válida, parámetros de configuración no válidos, etc.). Debería hacer un filtrado de datos y valores introducidos por el usuario, pero poco tiempo me ha dado. Por el mismo motivo, no se ha implementado el multilenguaje.

## PRUEBAS DE FUNCIONAMIENTO

Como se ha explicado, la aplicación funciona conectándose a un dispositivo (indicando su URL) para recoger los datos JSON del sensor. Para hacer pruebas reales desde fuera de mi intranet, he implementado un script cgi que simula el comportamiento y devuelve datos JSON con valores de temperatura y humedad con cierta aleatoriedad. Para probar pues la app desde internet, es conveniente dejar la URL por defecto: “http://quintoblog.com/temperaberry/temperaBerry.sh”

## CONCLUSIONES

Esta práctica y el curso en general me ha servido realmente para introducirme en el inmenso mundo de la programación Android. El ámbito es enorme y las posibilidades ahora me doy cuenta que son realmente amplias. El hecho de haber desarrollado una aplicación que realmente voy a usar a diario, es una motivación para seguir perfeccionando la aplicación (tengo algunas ideas) y adentrándome más en el mundo de programación Android, que me parece muy interesante. La falta de tiempo va a ser el mayor inconveniente, pero al menos la base ya la tengo. Este tema (desarrollo de aplicaciones Android) es algo que me pasaba por la cabeza desde hace tiempo. De hecho la idea de la aplicación no es nueva y lo pensé hace un tiempo, y me puse a buscar tutoriales e información para intentar empezarla pero, una vez más, la falta de tiempo me lo impidió. Ahora, por “obligación”, he podido aprender al menos las bases del desarrollo de aplicaciones. Muy interesante.

## WEBGRAFÍA


- http://developer.android.com

El punto de partida para cualquier desarrollador Android

- http://www.stackoverflow.com

  Imprescindible, con todas las letras. Un MUST para resolver casi cualquier duda tecnológica, desarrollo de apps Android incluida. Muchísimo soporte por parte de la comunidad de usuarios

- http://www.experts-exchange.com

  Muy importante también. En la línea de stackoverflow, donde puedes encontrar info que no has encontrado en la primera