

ACTIVIDADES - EJERCICIOS

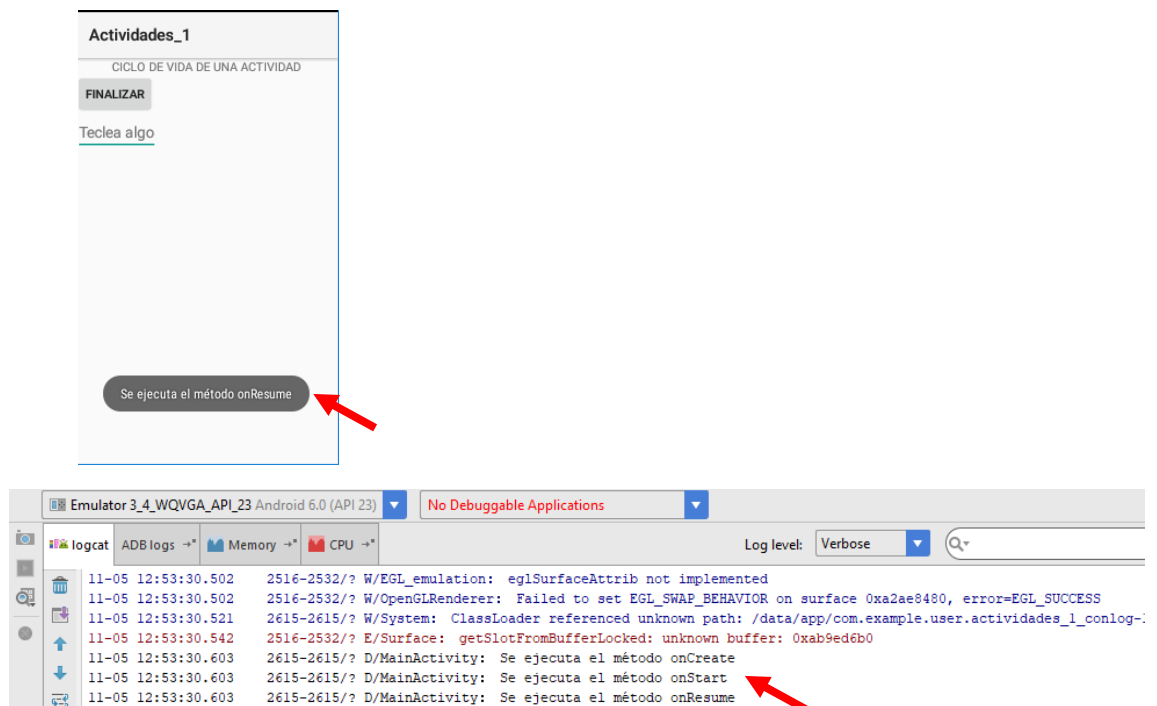
EJERCICIO 1

Proyecto: Ciclo_de_Vida

Características:

- El proyecto supone visualizar las diferentes fases del ciclo de vida de una actividad.
- Podemos hacerlo de dos formas: con mensajes mediante objetos de la clase Toast, o usando la clase Log.

Ejecución en emulador:

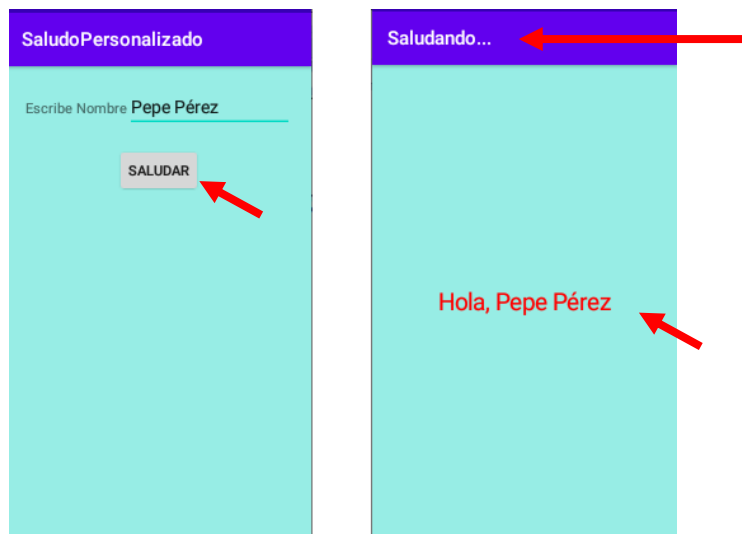


EJERCICIO 2

Proyecto: **SaludoPersonalizado**

Características:

- El proyecto supone una modificación del que ya hemos realizado en la unidad de eventos, con el nombre de **SaludoPersonalizado**.
- La variante está en que el saludo se va a mostrar en una nueva pantalla, según se ve en las capturas siguientes



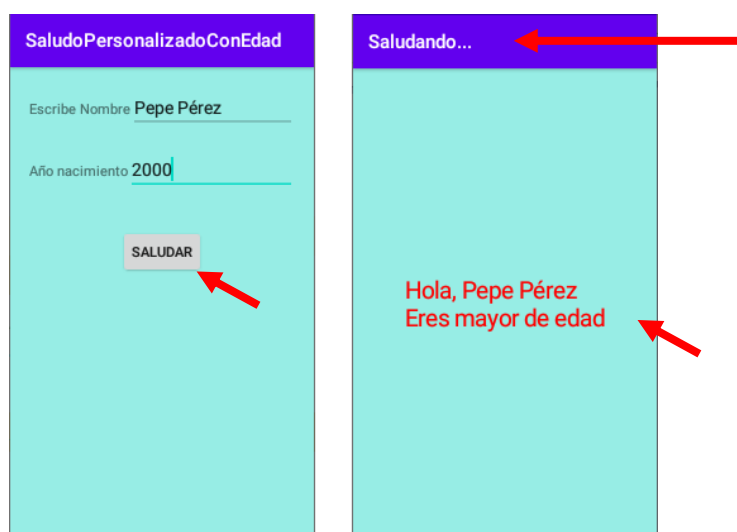
EJERCICIO 3

Proyecto: SaludoPersonalizado

Características:

- El proyecto supone una pequeña modificación del anterior.
- Como se ve en las siguientes capturas de pantalla, se trata de enviar **dos datos** a la segunda actividad: el **String** con el nombre y un **valor numérico** con la edad, que servirá para realizar el **test de mayoría de edad en la segunda actividad**.

Ejecución en emulador:



Propuesta:

Realizar el mismo proyecto empleando un **Bundle** para el envío/recepción de los datos.

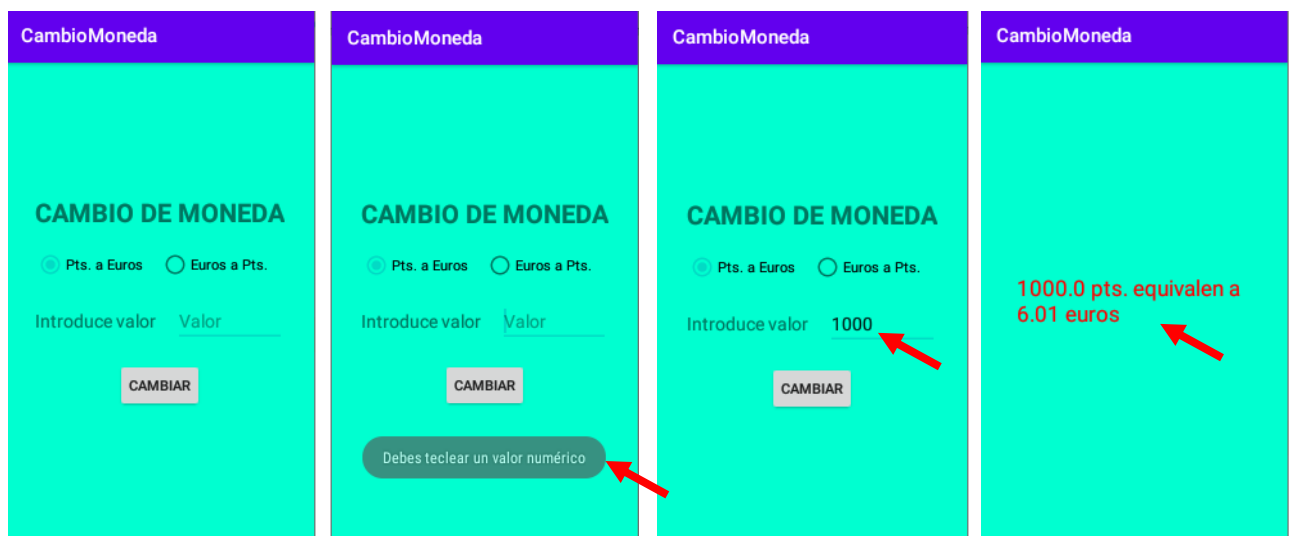
EJERCICIO 4

Proyecto: CambioDeMoneda

Características:

- Este proyecto supone una modificación del proyecto que habíamos realizado, con el mismo nombre, en el apartado de “Eventos”.
- Ahora se trata de utilizar dos pantallas:
 - **Pantalla inicial:** Se solicita el tipo de cambio y el valor a cambiar. Para aprovechar en lo posible el código original, también aquí se efectuará el cambio.
 - **Pantalla segunda:** Se muestra el resultado del cambio.

Ejecución en emulador:



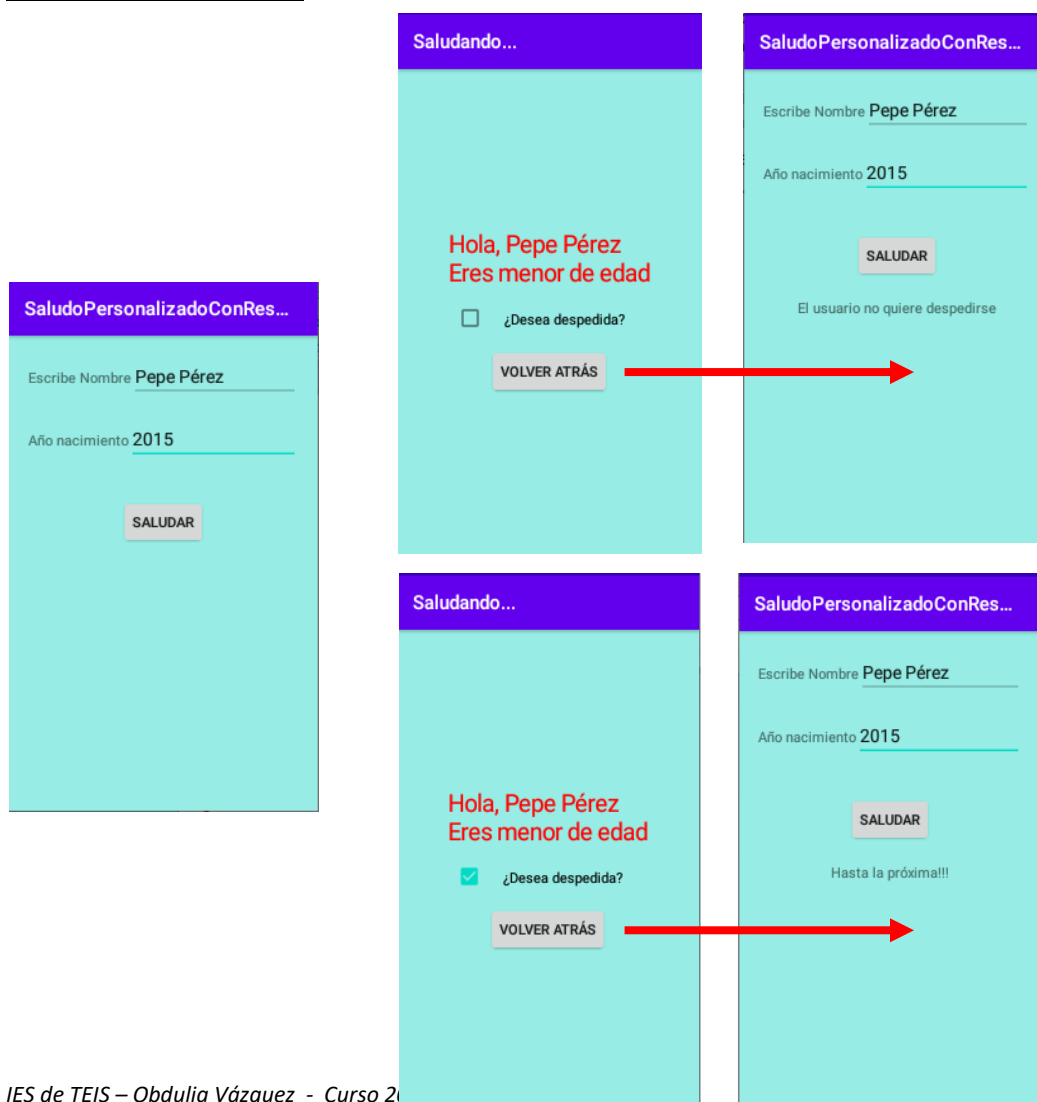
EJERCICIO 5

Proyecto: SaludoPersonalizado

Características:

- El proyecto supone una pequeña modificación del anterior **SaludoPersonalizado**, para añadir el caso de **llamada con respuesta**.
- Se siguen utilizando dos pantallas:
 - **Pantalla inicial:** Se solicitan los datos del usuario y se muestra un mensaje despedida.
 - **Pantalla segunda:** Se saluda de forma personalizada y se permite al usuario elegir si quiere o no una despedida.
- En la segunda pantalla:
 - Si el usuario pulsa el botón **Volver atrás** sin haber seleccionado la checkbox, se retorna a la pantalla inicial y se muestra un mensaje significativo (p.ej., “El usuario no quiso despedirse”)
 - Con la checkbox seleccionada, debe aparecer un mensaje diferente.

Ejecución en emulador:



Propuesta:

- Realizar una segunda versión más compleja.
- En la segunda pantalla:
 - Si el usuario selecciona la checkbox, se visualizará un grupo de botones de radio para que pueda elegir qué mensaje de despedida desea. El mensaje de la despedida debe ser adecuado a la selección del usuario.
 - El grupo de botones desaparece si el usuario desmarca la checkbox (como habíamos hecho en el ejercicio de eventos).

Ejecución en emulador:



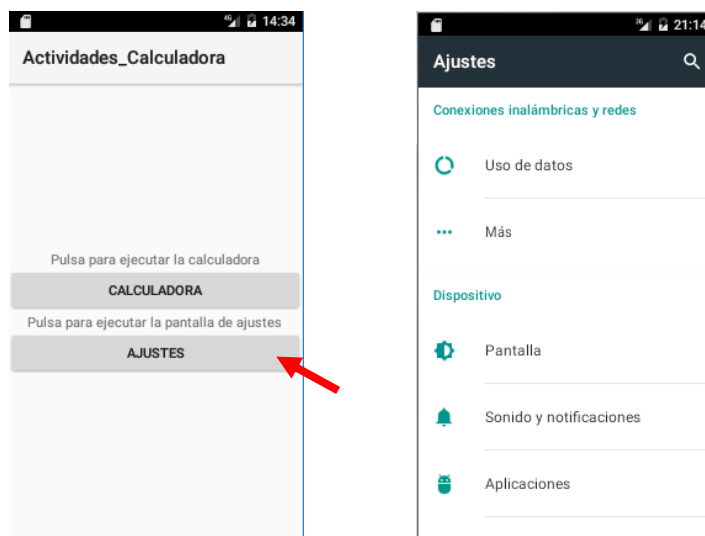
EJERCICIO 6

Proyecto: Actividades_Calculadora

Características:

- El proyecto supone una pequeña **modificación del propuesto en el pdf de los apuntes**, para añadir la posibilidad de **lanzar la ventana de Ajustes** de un dispositivo Android.
- Simplemente debe añadir un mensaje significativo y un botón que permita lanzar dicha aplicación.

Ejecución en emulador:



EJERCICIO 7

Proyecto: TresActividades

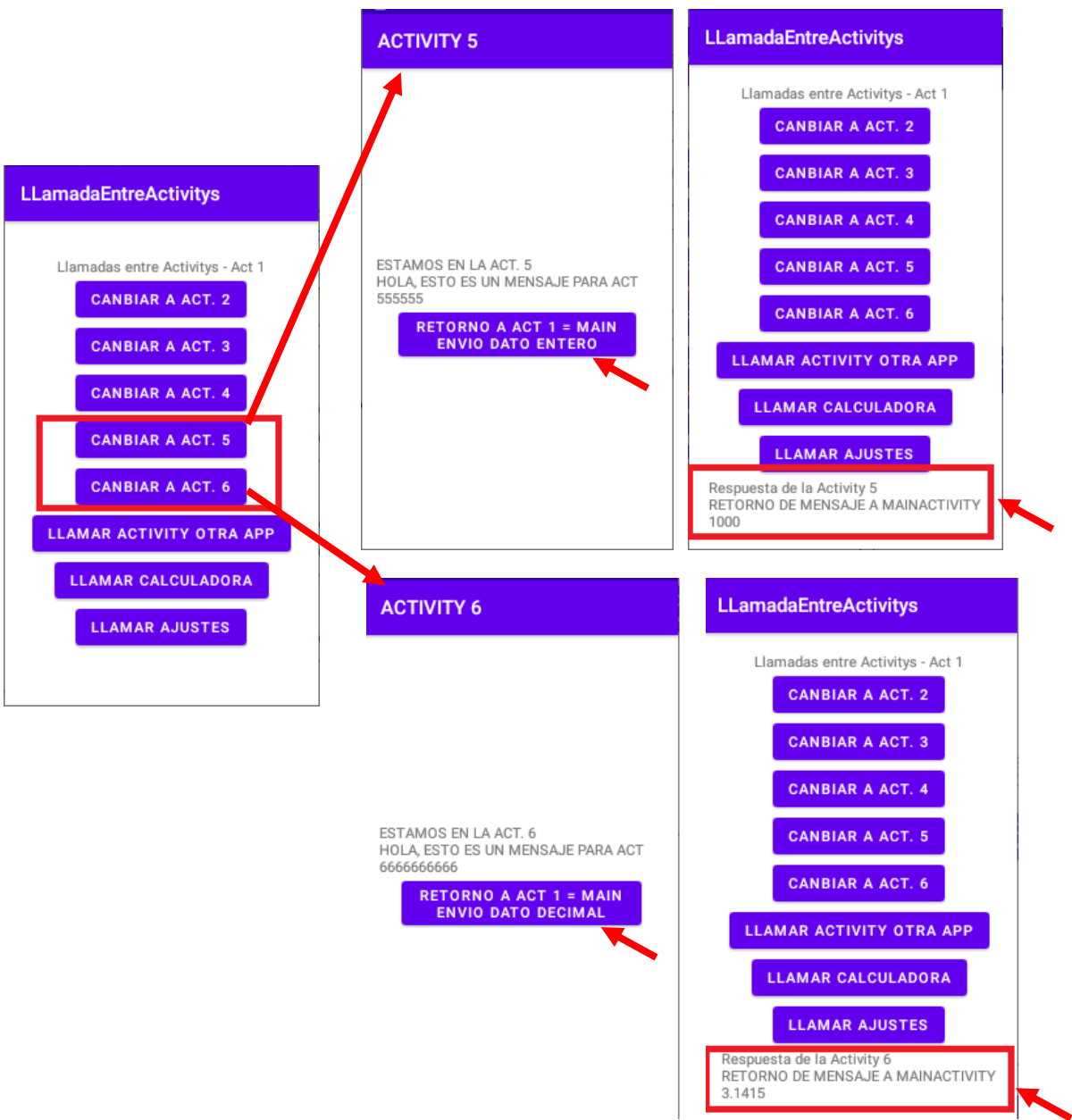
Características:

- Proyecto con una actividad inicial desde la cual se **llama a dos subactividades esperando respuesta** (actividad 2 y actividad 3). ***También podemos aprovechar el último proyecto sobre activitys realizado en clase.***
- En cada llamada se enviará un mensaje significativo, diferente en cada caso.
- Cada una de las subactividades retornará un dato diferente a la actividad inicial:
 - La **actividad 2** devolverá un **valor entero**.
 - La **actividad 3** devolverá un **valor con decimales**.
 - La actividad inicial mostrará en un Toast el número recibido en cada caso, **diciendo, además, de qué subactividad lo ha recibido.**

Ejecución en emulador:



Ejemplo de capturas desde el proyecto hecho en clase:



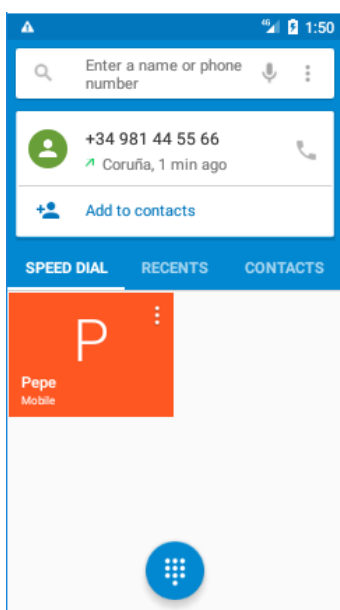
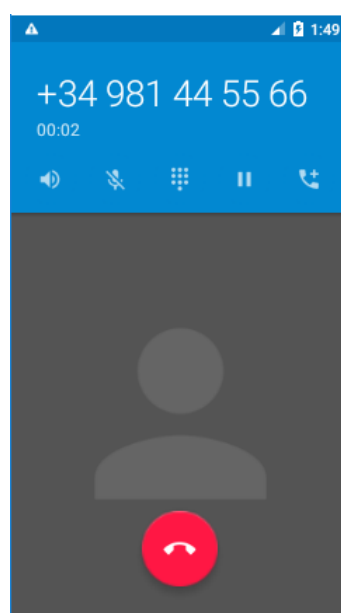
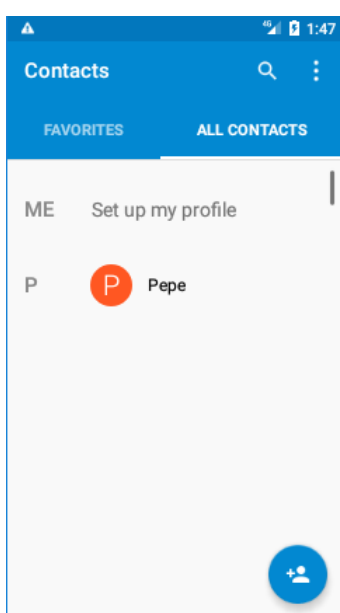
EJERCICIO 8

Proyecto: IntentsImplicitos

Características:

- Vamos a utilizar varios intents implícitos con diferentes acciones: visualizar los contactos almacenados en el teléfono, llamar a un número determinado, mostrar el dial para efectuar una llamada y cargar la página inicial de la Xunta.
- Inicialmente, lo haremos con un terminal de API inferior a 23 (por el tema de los permisos).

Ejecución en emulador:



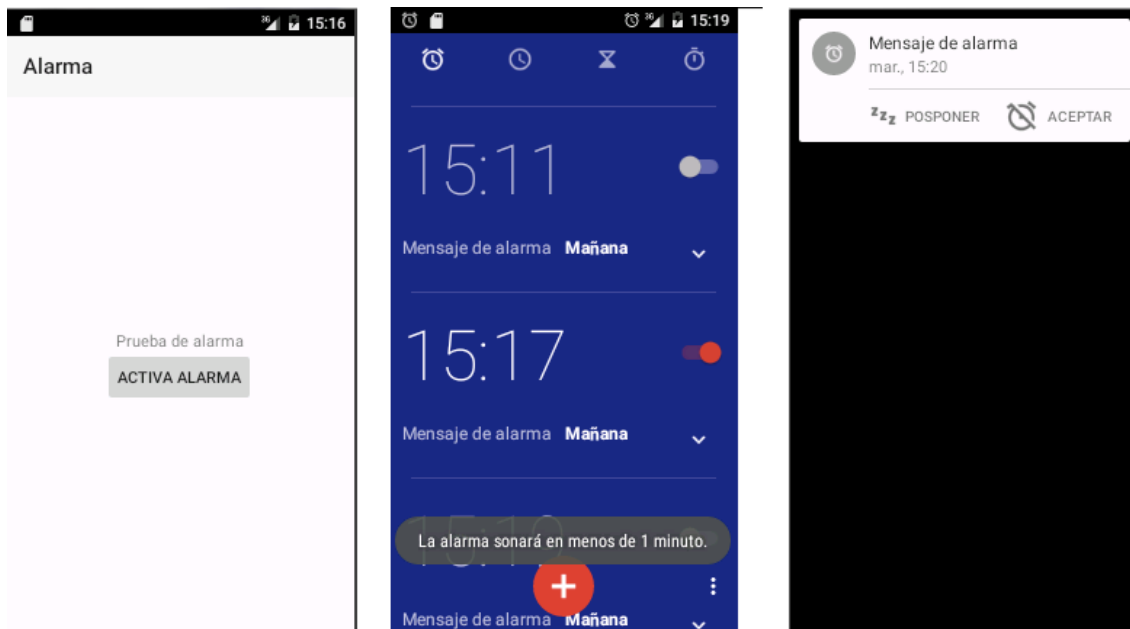
EJERCICIO 9

Proyecto: Alarma

Características:

- Proyecto que establece una alarma para una hora determinada.
- Cuando suene la alarma, se visualiza un mensaje.
- Utiliza un intent implícito: ***AlarmClock.ACTION_SET_ALARM***
- Para poder probarla, en esta primera versión debemos fijarnos en la hora del terminal, y establecer la hora de la alarma unos minutos después.

Ejecución en emulador:



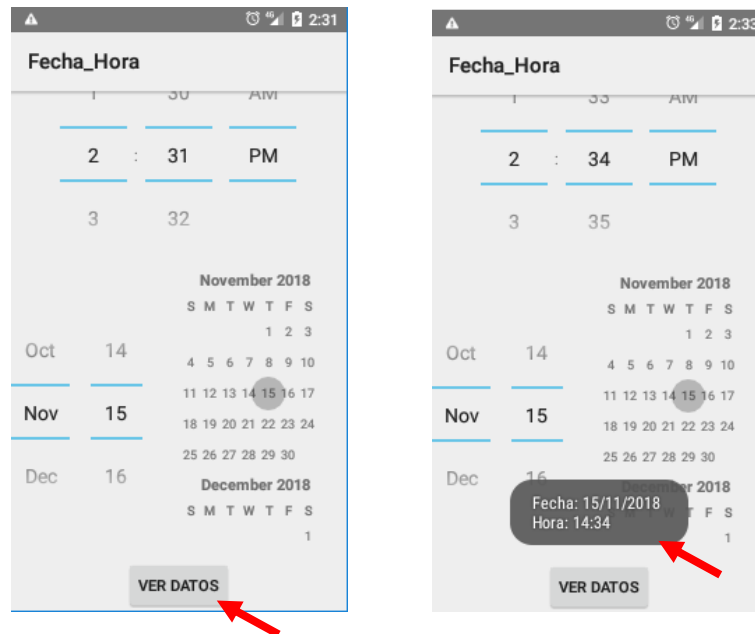
EJERCICIO 10

Proyecto: **Fecha_Hora**

Características:

- Proyecto que visualice en un toast la fecha y la hora seleccionadas por el usuario en sendas vistas **DatePicker** y **TimePicker**.

Ejecución en emulador:



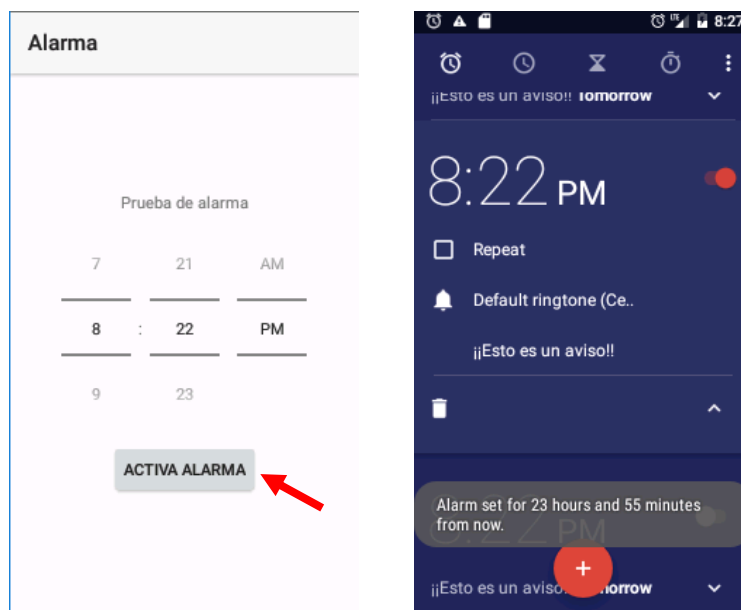
EJERCICIO 11

Proyecto: **Alarma_v2**

Características:

- Modificaremos el código del proyecto “**Alarma**” de forma que la hora de la alarma no se indique como parte del código, sino que la elija el usuario mediante la vista **TimePicker**.

Ejecución en emulador:



EJERCICIO 12

Proyecto: Intent-filter

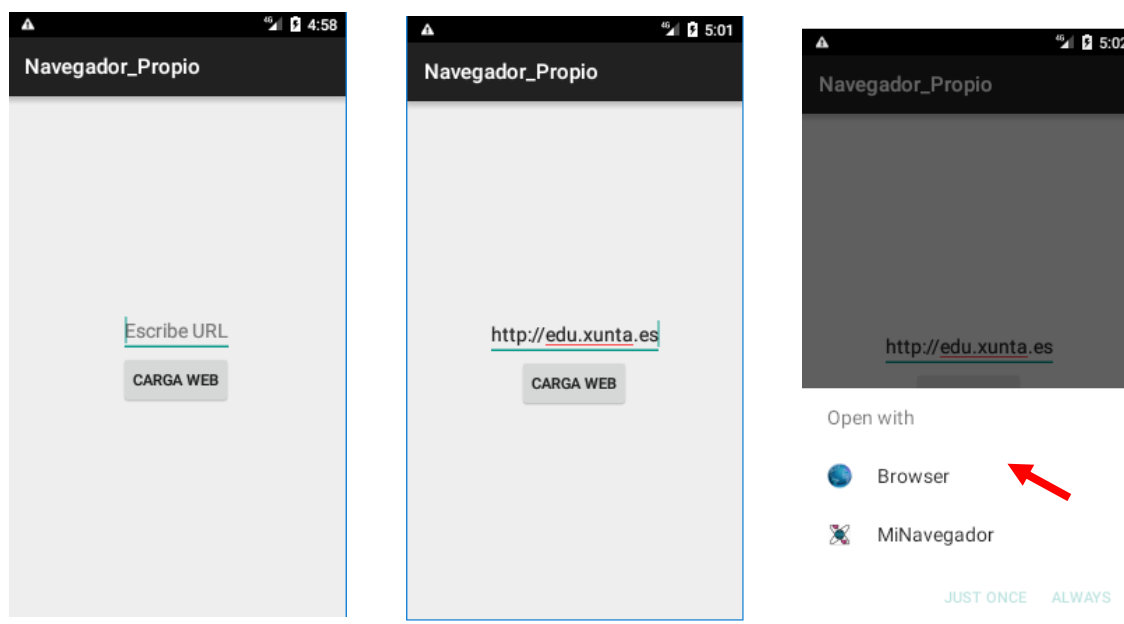
Características:

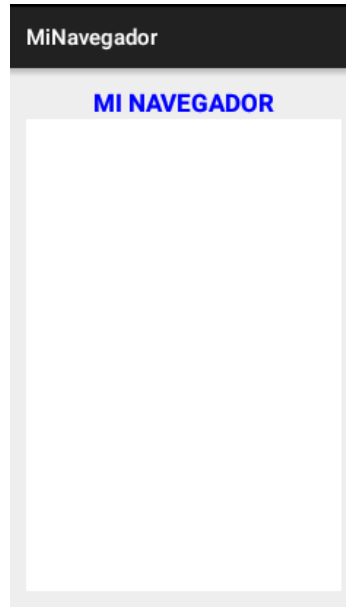
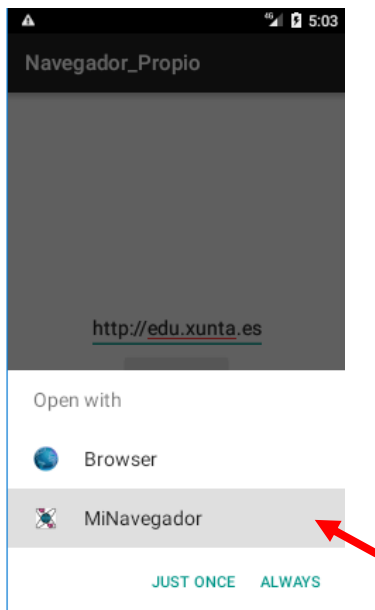
- Se trata de un proyecto en el que probar los filtros de intención de una actividad creada por nosotros.
- La **actividad principal** permitirá la entrada de una URL en una caja de texto. Damos por supuesto que se tratará de una URL correcta.
- Un botón permite activar la carga de la página web correspondiente a dicha URL.
- La carga de dicha página web se va a efectuar en una **segunda actividad**, dentro de una vista de tipo **<WebView>** definida a tal fin.
- Veremos como el sistema nos da a elegir entre las diferentes activitys que pueden resolver nuestra petición.

Documentación WebView: <https://developer.android.com/guide/webapps/webview.html>

Ejecución en emulador:

(Las capturas corresponden a un terminal de API 22)





(Captura en terminal con API 25)

