# D.1 – Activity e Intents

## Activity

Finora abbiamo visto come modificare una singola classe, la Main Activity.

L’activity rappresenta una funzionalità dell’app, e spesso un Entry point per l’utente.

Quando la nostra app viene lanciata dal sistema, Android inizializza l’activity che si occupa di disegnare l’interfaccia. Durante il suo ciclo di vita, l’activity ci offre delle callback come *onCreated* con le quali la programmiamo.

Tutte le activity devono essere dichiarate all’interno del manifest, tramite l’unico parametro obbligatorio *name*. L’activity che verrà lanciata quando l’utente clicca sull’icona dal sistema, sarà impostata tramite i parametri *action-Main* e *category-Launcher* dentro *intent-filter.*

Text

Description automatically generated

L’activity possiede un *lifecycle* – implementa l’interfaccia *lifecycleOwner* – che rappresenta i vari stati in cui si puo’ trovare durante la sua esistenza. È infatti sottoposta a varie interazioni con l’utente, che puo’ decidere da un momento all’altro di farla partire, tornare indietro, metterla in background o chiuderla definitivamente.

Diagram

Description automatically generated

Lifecycle dell’activity

Possiamo sfruttare le varie callback per gestire le features che intendiamo sviluppare in base allo stato dell’activity. onCreate richiede di bindare il layout, e il setup delle view viene impostato. onResume avviene quando l’activity diventa visibile dopo il binding, o ritorna visibile dopo essere stata messa in background.

La documentazione che spiega a fondo questi stati è disponibile qui:

[https://developer.android.com/guide/components/activities/intro-activities#mtal](https://developer.android.com/guide/components/activities/intro-activities%23mtal)

## Controllo delle risorse da parte del sistema

È importante chiarificare il concetto che Android si occupa molto aggressivamente della gestione delle risorse. Ci sono delle limitazioni nelle attività che possono essere effettuate, e molte richiedono che l’utente stia attivamente visualizzando l’app.

Questo implica che se la nostra app ha delle schermate che non sono visibili all’utente in quel momento, magari perché in un punto precedente dell’app, queste possono essere soggette alla distruzione da parte del sistema nel caso ci sia una necessità di risorse. Ad esempio, se l’utente riceve una telefonata o decide di scattare una foto, la nostra app potrebbe essere distrutta.

## Intents

Le activity sono in grado di comunicare tra di loro, non solo all’interno della stessa app, ma anche tra le app installate nel sistema operativo. Pensiamo alla funzionalità di share, che ci permette di mandare un’immagine o un testo da un’app all’altra.

Eppure, le activity non sono legate strettamente tra di loro, non avendo modo diretto di comunicare.   
Per questo si utilizzano gli **Intents**

Un Intent è un oggetto messaggero, che puoi utilizzare per richiedere un'azione da un altro componente.

Ci sono 3 casi fondamentali per l’utilizzo di Intenti:

* **Starting an activity**
  + Far partire e inviare dati ad altre activity, tramite *startActivity()*
* **Starting a service**
  + Un Service è un componente che effettua operazioni in background senza UI. ([Daemon](https://en.wikipedia.org/wiki/Daemon_(computing)))
* **Delivering a broadcast**
  + Un Broadcast è un messaggio intercettabile da tutte le app

Vediamo come è possibile utilizzarli:

Naviga verso un’altra activity

Crea una nuova app con una Empty Activity

* Aggiungi un FAB un basso a dx

Text

Description automatically generated

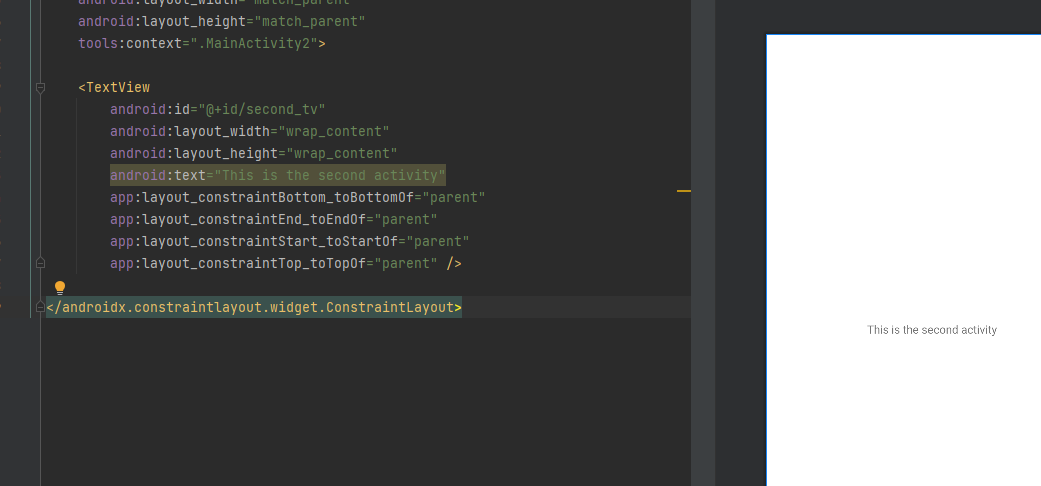
Crea una nuova Empty activity con il wizard:

A screenshot of a computer

Description automatically generated with medium confidence

Questo aggiungerà anche i necessari campi nel manifest.

Nel layout della nuova activity, aggiungi un testo che specifica che siamo in un’altra activity:



Vai in Main Activity, effettua il binding. Aggiungi un clicklistener al fab

Text

Description automatically generated

Ora vediamo come è possibile, alla pressione del fab, cambiare activity.

* Crea un Intent che prende come input un contesto (partenza) e una classe (destinazione)
* Lancialo tramite la funzione *startActivity,* disponibile qui perché siamo all’interno di una Activity:

Graphical user interface, text

Description automatically generated

Ora cliccando il FAB navigheremo verso la nuova activity.

Decidiamo di impostare il testo della seconda activity inviando dati dalla prima activity

Aggiungi all’Intent il testo che vuoi inviare alla seconda activity:

Graphical user interface, text, website

Description automatically generated

Nell’activity di destinazione ora è possibile recuperare questo extra tramite il *name* da noi inserito – gli extra sono una mappa chiave-valore

* Effettua il binding nella seconda activity

Activity ha un campo *intent* dove c’è il riferimento all’intent che ha lanciato l’activity.

Il dato che abbiamo messo negli extra è una stringa: utilizziamo la funzione intent.getStringExtra(name) per recuperarlo.

Graphical user interface, application

Description automatically generated

La mappa degli extra non assicura che abbia la chiave *custom\_text* che gli stiamo fornendo, quindi il risultato sarà nullable. Per prendere il risultato in caso di non-null, utilizziamo ?.let e sfruttiamo subito la nostra stringa per popolare il testo della view *secondTv.*

Graphical user interface, text, application

Description automatically generated

Condividi un contenuto

Vediamo ora come è possibile inviare dati ad un’altra app tramite la funzionalità di *share* di Android.

Nella seconda activity

* Aggiungi un FAB con l’icona @android:drawable/ic\_menu\_share

Ora condivideremo il testo del messaggio ad altre app, come Whatsapp o Gmail.

* Recupera il testo del messaggio tramite *textview.text.toString()*

L’intent da creare avrà una *action* che definisce l’azione da effettuare, e *type* che definirà le app che sono in grado di ricevere la condivisione

La funzione *Intent.createChooser* specifica al sistema che dovrà esporre il contenuto tramite picker. Conterrà l’intent con i dati a le azioni, e un titolo da dare al picker.

Graphical user interface, text, application

Description automatically generated

## Esercizio: Condividi Card Collection Power

Aggiungi queste features a Card collection

* La main activity contiene anche un campo che somma tutti i valori della lista per mostrare il totale.
* Condividi il valore di potenza del tuo mazzo con gli altri!