

images/UniPD\_DEI\_logo.png

UNIVERSITÀ DEGLI STUDI DI PADOVA

CORSO DI PROGRAMMAZIONE DI SISTEMI EMBEDDED

# Material Design 3

Principi di base e presentazione di un'applicazione che ne fa uso



images/logoOption1.png

*Gruppo:*

**Bolzonello Enrico**

**Moschetta Daniele**

**Zanini Fabio**

Anno Accademico 2021/22

# Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
<b>2</b>	<b>La nascita di Material Design</b>	<b>2</b>
2.1	Skeumorfismo . . . . .	3
2.2	Flat Design . . . . .	3
2.3	Material Design . . . . .	3
<b>3</b>	<b>Material Design 1</b>	<b>4</b>
3.1	Ambiente e materiali . . . . .	5
3.2	Movimento . . . . .	8
3.3	Stile . . . . .	11
3.4	Layout . . . . .	17
3.5	Componenti . . . . .	20
3.6	Pattern per l'interfaccia utente . . . . .	33
3.7	Istruzione dell'utente . . . . .	38
3.8	Usabilità . . . . .	39
3.9	Piattaforme . . . . .	41
<b>4</b>	<b>Material Design 2</b>	<b>42</b>
4.1	Forme . . . . .	43
4.2	Temi . . . . .	44
4.3	Suono . . . . .	46
4.4	Componenti . . . . .	47
<b>5</b>	<b>Material Design 3</b>	<b>51</b>
5.1	Material You . . . . .	52
5.2	Colori dinamici . . . . .	54
5.3	Material Theme Builder . . . . .	56
5.4	Tipografia . . . . .	57
5.5	Material Symbols . . . . .	61
5.6	Componenti . . . . .	63
5.7	Adattamento a dispositivi diversi . . . . .	72
<b>6</b>	<b>BookTracker</b>	<b>77</b>
6.1	Struttura dell'applicazione . . . . .	78
6.2	Colori Dinamici . . . . .	83
6.3	Tema scuro . . . . .	85

# 1 Introduzione

Material Design è un sistema adattabile di linee guida, componenti e strumenti che supportano le migliori pratiche per il design di interfacce utente. Material permette di rendere più agevole e proficua la collaborazione tra sviluppatori e designer, fornendo la possibilità di creare velocemente prodotti esteticamente piacevoli. L'idea fondamentale di Material Design è quella di fornire agli sviluppatori una base da cui partire per poi sviluppare in modo unico le loro idee e le loro visioni.

In questo testo abbiamo analizzato le funzionalità e le potenzialità di questo strumento, in particolare della sua versione più recente e potente: Material Design 3.

Infine, abbiamo analizzato e descritto il funzionamento di Book Tracker, un'applicazione da noi sviluppata che fa uso dei principi di Material Design.

## 2 La nascita di Material Design

**Material Design** è nato dall'idea di un gruppo di designer di rendere più agevole lo sviluppo di applicazioni con un design gradevole e al tempo stesso funzionale.

Material Design è stato lanciato nel 2014 sul sito *material.io* ed in breve tempo è stato adottato dalla maggior parte delle applicazioni web. Sul sito ufficiale è tutt'ora possibile consultare informazioni riguardo i componenti e linee guida necessarie ad un designer per progettare siti e applicazioni piacevoli all'utilizzo.

A differenziare Material Design da molti altri sistemi di design è il non essere stato progettato per un brand o una piattaforma specifica, rendendo possibile il suo utilizzo a chiunque stia sviluppando un sito web o un'applicazione, ma è stato inizialmente usato per lo **sviluppo di applicazioni Android** con l'obiettivo di renderle più intuitive e user-friendly su diversi dispositivi.

Prima dell'avvento di Material, le applicazioni presentavano interfacce inconsistenti. Il metodo di lavorare sul design delle applicazioni non funzionava perché non era stata studiata la fisica alla base dello sviluppo di interfacce. L'idea, quindi, è stata quella di analizzare le proprietà concrete delle applicazioni e di capire "di che materiale potesse essere fatto il software".

Si è giunti così all'idea di una carta smart ("*Quantum Paper*") e quindi di un'interfaccia utente con una forte caratteristica concreta e tattile.

È stato approfondito lo studio delle **ombre** che oggetti di carta producono in differenti condizioni di illuminazione, portando allo sviluppo dei concetti di spazio e di strati. L'idea alla base delle **palette cromatiche** è stata quella di scegliere un colore principale e costruire attorno ad esso una gamma di colori secondari che possano rendere lo spettro dei colori coerente e naturale. Le **gesture**, inoltre, sono state pensate per essere familiari all'utente e sono state correlate ad animazioni che mimano oggetti ed interazioni del mondo reale.

Il risultato è stato lo sviluppo di un concetto di interfaccia utente che ricorda fortemente le azioni concrete e che dona all'utente un senso di immediatezza e allo stesso di potenza che con altri metodi non era stato possibile raggiungere.

Negli anni successivi all'introduzione di Material Design, quasi ogni piattaforma software ha iniziato a sfruttare i principi e gli strumenti forniti da esso. Qual è stato, quindi, il motivo del successo così ampio di Material Design?

Precedentemente all'introduzione del nuovo paradigma, esistevano due principali approcci al design delle applicazioni:

- **skeumorfismo** (skeumorphism)
- **flat design**

## 2.1 Skeumorfismo

Lo **skeumorfismo**[15] (skeumorphism) è un approccio di design che ha come obiettivo quello di creare elementi di interfaccia utente che assomiglino il più possibile agli **oggetti reali** su cui essi sono basati. Questo tipo di design era usato principalmente per sfondi e icone.

Il principio di base era quello di rendere l'utente in grado di riconoscere facilmente gli oggetti con cui interagiva, perché molto simili a quelli usati nel mondo reale. Questo rendeva le applicazioni piacevoli da vedere perché già familiari all'utente, ma di difficile utilizzo perché **troppo complesse** e talvolta anche sgradevoli alla vista. Inoltre, questo tipo di design non offriva alcun vantaggio all'utente: era solamente volto ad essere bello.

## 2.2 Flat Design

L'eccessivo realismo e la complessità dello skeumorfismo hanno portato all'adozione del **flat design**[14], la cui idea di fondo era quella di **rimuovere gli eccessi** del precedente approccio e di essere comunque in grado di rappresentare fedelmente gli oggetti reali. L'efficienza e la semplicità di questo nuovo approccio lo hanno reso in breve tempo la scelta adottata dalla maggior parte dei designer di tutto il mondo. Lo scopo di questo approccio minimalista era quello di rendere l'**interfaccia efficiente, pulita e funzionale**: obiettivo comune tra sviluppatori e designer. Tuttavia questo tipo di design presentava spesso incertezza e ciò portava l'utente ad essere disorientato e a non usufruire in modo gradevole delle funzionalità del software.

## 2.3 Material Design

**Material Design** ha unito le caratteristiche di successo dei due precedenti approcci e le ha fuse insieme per renderle ancora più utili. Material **prende ispirazione dagli oggetti reali** ma non li copia in modo preciso: ne prende solamente le caratteristiche che li rendono riconoscibili e che gli donano un senso di forma e concretezza (come ombre, luci, profondità, spigoli, variazioni di colore, etc.). Di conseguenza, esso è molto **basato sul flat design** ma gli elementi integrano caratteristiche fisiche reali e si comportano esattamente come fa la carta nella realtà. Questo aspetto di concretezza permette agli sviluppatori di creare delle applicazioni che rispondano in modo naturale alle interazioni con l'utente e che sfruttino gesture da esso prevedibili.

Material Design, quindi, ha permesso di risolvere molti dei problemi riscontrati nei due approcci precedenti e ciò lo ha reso il design system più usato su internet.

## 3 Material Design 1

Sin dalla prima versione di Material Design [5], gli obiettivi sono i seguenti:

- creare un linguaggio visivo che sintetizzi i principi di buon design, integrando le possibilità che la tecnologia moderna offre;
- sviluppare un sistema di progettazione universale che possa unificare l'esperienza utente anche su dispositivi molto diversi.

I principi sono già delineati nella presentazione al Google I/O del 2014[10] e sono i seguenti:

- *Material is the Metaphor*, per Google la metafora è il retroscena per il design, lo unifica e funziona sia per il pubblico, presentando una metafora che possono capire, sia per i designer e i developer;
- *Surfaces are Intuitive and Natural*, la base è un materiale simile alla carta ma così complesso da essere *indistinguibile dalla magica*. Usando la carta, un materiale che esiste nel mondo reale, le superfici diventano una metafora di facile comprensione, addirittura più naturale del linguaggio;
- *Dimensionality affords Interaction*, gli elementi cliccabili salgono in superficie in risposta al tocco. L'idea che dà è che quella superficie sale ed incontra il dito dove tocca;
- *Content is Bold, Graphic and Intentional*;
- *Color, Surface and Iconography emphasize Actions*, fornisce un set di icone base e una grid universale che supporta le forme geometriche primarie per creare icone custom;
- *Users initiate Change*, l'app risponde all'utente con un feedback;
- *Animation is Choreographed on a Shared Stage*, deve essere il materiale che si muove verso l'utente;
- *Motion provides Meaning*, il modo in cui le cose si muovono forniscono indizi su come funzionano e sulle loro proprietà oltre a mostrare cos'è importante;
- *One Adaptive Design*, ogni dispositivo mostra una versione diversa dello stesso sistema, modificato alla dimensione e alle interazioni appropriate per il dispositivo;
- *A Unified Creation Process*, unisce ingegneria e grafica.

### 3.1 Ambiente e materiali

L'ambiente in cui vengono sviluppate le applicazioni è un **ambiente tridimensionale**. Ogni oggetto, quindi, ha dimensioni x, y e z, con l'asse z impostato perpendicolarmente rispetto allo schermo del dispositivo. Ogni oggetto occupa una posizione sull'asse z ed ha uno spessore standard di 1dp.

Nell'ambiente sono impostate varie tipologie di luce che interagiscono in modi diversi con gli oggetti per formare ombre più o meno marcate. Le luci possono essere ambientali, che formano ombre leggere, e direzionali, che danno vita ad ombre marcate e incisive.

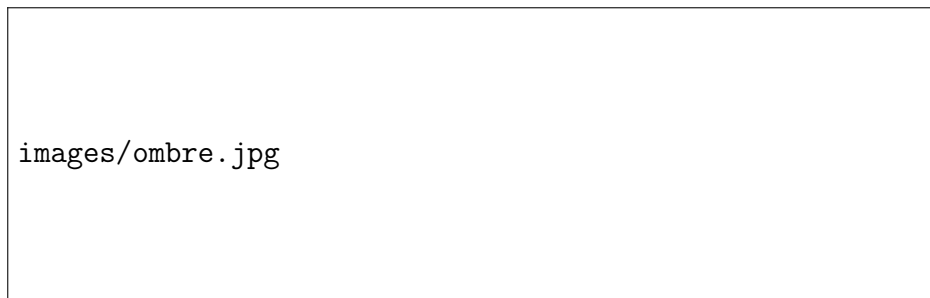


Figura 1: Varie tipologie di ombre

I materiali e gli oggetti che formano l'interfaccia utente hanno le seguenti caratteristiche:

- sono **solidi** e **impenetrabili**
- occupano **posizioni specifiche** nello spazio (più materiali non possono occupare la stessa posizione nello spazio)
- hanno **forme mutabili** ma non possono essere piegati
- cambiano dimensione solo nel loro piano
- possono essere **uniti** ad altri oggetti, **divisi o riuniti**
- possono essere **creati e distrutti ovunque** nell'ambiente
- possono **muoversi su tutti gli assi**

I materiali hanno ma uno spessore costante di 1dp ed i contenuti sono mostrati su questi materiali, con qualsiasi colore e forma. I contenuti non aumentano lo spessore dei materiali e possono comportarsi in modo indipendente dai materiali, senza però uscire dai loro bordi.

Il movimento dei materiali sull'asse z è tipicamente dovuto alle interazioni con l'utente.

L'**elevazione**[2] degli oggetti viene misurata tra il fronte di uno oggetto e il fronte del successivo e determina la distanza (sull'asse z) tra le superfici e, di conseguenza, le ombre che si formano.

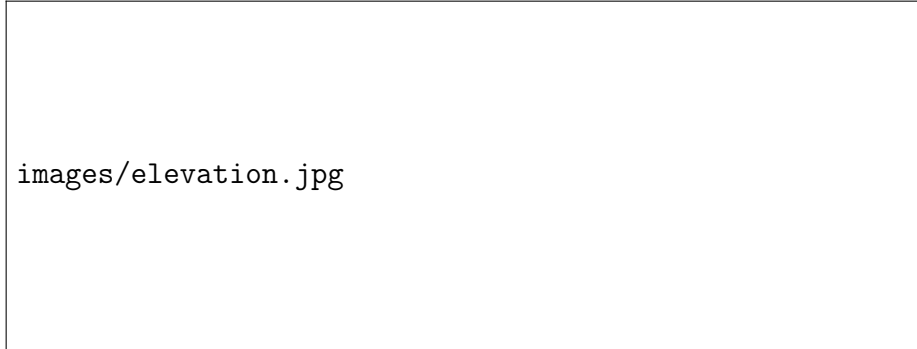


Figura 2: Elevazione

Ogni componente ha un'elevazione a riposo, che è consistente tra diverse applicazioni, ma può cambiare tra dispositivi e piattaforme differenti. L'elevazione di un componente figlio è relativa all'elevazione del componente padre.

L'elevazione di alcuni oggetti cambia in relazione alle interazioni con l'utente o agli eventi di sistema. Questi cambiamenti sono definiti da **offset** (*Dynamic elevation offsets*) che permettono di rendere le variazioni di elevazione consistenti attraverso diverse azioni e tipologie di componenti.

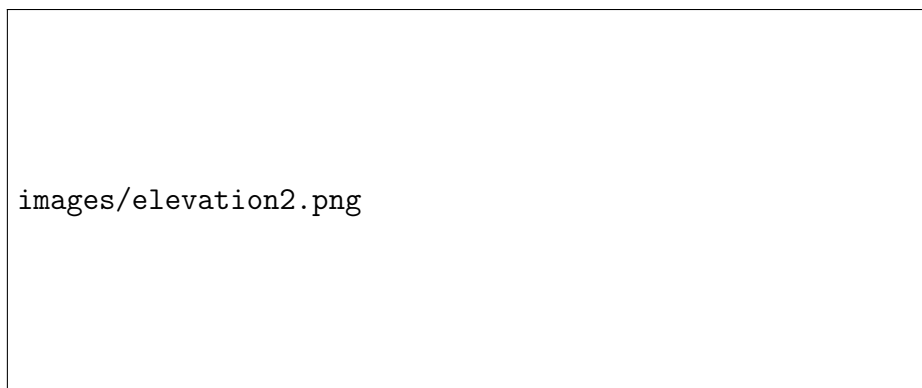


Figura 3: Un esempio pratico di elevazione e strati

I componenti sono impenetrabili, quindi non possono interferire e passare attraverso altre componenti, di conseguenza possono sparire e ricomparire per evitare interazioni oppure muoversi all'interno dell'interfaccia. Questa è quindi progettata per minimizzare le possibilità di interferenze tra oggetti diversi.

Le **ombre** permettono di amplificare la tridimensionalità dell'ambiente e mostrano la separazione tra le varie superfici. Esse infatti sono determinate dall'elevazione degli oggetti. Inoltre, quando gli oggetti sono in movimento, permettono di identificare con maggiore facilità la direzione di tale movimento e capire se la distanza tra superfici sta cambiando o meno (come si vede in figura 41).



Le ombre di alcuni componenti seguono degli standard. Per esempio, il FAB (Floating Action Button), di cui si parlerà in seguito, deve avere un'elevazione a riposo di 6dp ed un'elevazione in seguito a pressione di 12dp.



Figura 4: A sinistra il cambio di elevazione, a destra il cambio di forma

Tutti gli oggetti fanno parte di una **gerarchia**. In ognuna di queste relazioni, il figlio è riferito ad un elemento più importante, che è l'elemento genitore. Ogni oggetto può essere figlio di un altro oggetto, oppure del sistema.

Gli oggetti possono muoversi indipendentemente gli uni dagli altri oppure avere movimenti relazionati a quelli dei componenti più alti nella gerarchia.

Le regole della relazione figlio-genitore sono le seguenti:

- ogni oggetto ha solo un genitore
- ogni oggetto può avere un numero qualsiasi di figli. i figli ereditano le proprietà di trasformazione del genitore, come per esempio la posizione, la rotazione, la scala e l'elevazione
- i fratelli sono oggetti con la stessa posizione nella gerarchia

Alcuni elementi, come il FAB, sono autonomi e si muovono indipendentemente dagli altri.

La relazione determina anche come gli elementi interagiscono tra di loro. Per esempio, i figli hanno una separazione sull'asse z minima dai genitori. Di conseguenza, altri oggetti non possono inserirsi tra di loro.

## 3.2 Movimento

Il **movimento**[13] è essenziale per mostrare ciò che un'applicazione può fare e come essa è organizzata.

Il movimento permette di:

- **guidare l'attenzione** attraverso diverse viste e interfacce
- indicare all'utente **ciò che accadrebbe se completasse una gesture**
- **evidenziare le gerarchie e le relazioni** tra gli oggetti
- **togliere l'attenzione** da ciò che accade in background e che non si vuole mostrare
- rendere l'applicazione **più ordinata e gradevole** alla vista

L'ambiente di Material Design prende ispirazione dalla **fisica del mondo reale**. Di conseguenza, anche per quanto riguarda il movimento, esso è dovuto ai **rapporti tra forze** dovute all'interazione tra l'utente e gli elementi dell'interfaccia.

Il materiale di cui sono formati gli oggetti è estremamente responsivo, si comporta in modo naturale, è consapevole dell'ambiente circostante ed è preciso nell'obiettivo e nella sincronizzazione del movimento.

Un buon movimento, inoltre, deve essere veloce, chiaro, semplice e coerente con l'azione che si vuole rappresentare e con gli altri movimenti dell'interfaccia.

Il movimento degli elementi deve avere la **giusta durata**[11] in modo da poter essere compreso, ma anche non essere troppo lungo. La durata delle animazioni non deve essere standard, ma deve essere tarata per il tipo di movimento e la distanza percorsa dai vari elementi, l'entità del cambiamento della loro forma e la loro velocità. Inoltre, animazioni che richiedono meno attenzione dovrebbero essere più veloci.

La durata delle animazioni è da impostare considerando anche la grandezza dello schermo: schermi grandi necessiteranno di animazioni più lunghe, mentre schermi piccoli di animazioni più corte.

Le accelerazioni e decelerazioni devono essere **fluide** durante la durata dell'animazione e, inoltre, esse appaiono più naturali se sono asimmetriche. Ci sono quattro comuni tipi di curve di accelerazione:

1. **standard** (*standard curve*), in cui gli elementi accelerano velocemente e decelerano lentamente (applicate solitamente ad elementi che cambiano di dimensione o le cui caratteristiche variano in altri modi)
2. **di decelerazione** (*deceleration curve*), in cui gli elementi entrano nello schermo a velocità massima e decelerano lentamente fino a fermarsi (talvolta accompagnati da cambiamenti di dimensione e/o opacità)

3. **di accelerazione** (*acceleration curve*), in cui gli elementi partendo da fermi, accelerano e lasciano lo schermo a velocità massima (anch'essi talvolta accompagnati da variazioni di opacità e/o dimensione)
4. **spigolose** (*sharp curve*), in cui gli elementi accelerano e decelerano velocemente (usate per elementi che escono temporaneamente dallo schermo e che potrebbero rientrare in qualsiasi momento)

Per ricordare la fisica reale e l'effetto della forza di gravità, gli elementi che salgono (in diagonale) nello schermo seguono traiettorie ad arco concavo dal basso verso l'alto, mentre elementi che scendono nello schermo seguono un arco concavo dall'alto verso il basso. Invece, gli elementi che si muovono solo su un singolo asse non seguono traiettorie ad arco, ma traiettorie semplicemente rettilinee.

Material Design fa sembrare vive le superfici, dividendole, moltiplicandole e cambiando la loro forma e dimensione. Esistono due tipi di trasformazioni:

1. **rettangolare**, quando un oggetto cambia forma e dimensione, la sua larghezza e la sua altezza cambiano in modo asincrono. Tuttavia, il contenuto presente nell'oggetto viene modificato in modo costante, per evitare deformazioni.
2. **Trasformazione radiale**, è una trasformazione simmetrica e circolare, che ha origine dal punto di tocco dell'utente. Spesso viene usata su superfici circolari che si trasformano in altre forme e anche per confermare l'input dell'utente.

In caso di unione di materiali, i loro margini vengono sovrapposti prima della fine del movimento.

Durante le transizioni, alcuni elementi (solo i più importanti) devono rimanere **visibili** per guidare l'occhio dell'utente. Se invece non ci sono elementi comuni tra l'oggetto di partenza e quello di arrivo, allora bisogna ancorare gli elementi al margine superiore della superficie e lasciare che il nuovo contenuto oscuri il contenuto precedente. Se non ci sono superfici in comune, allora una nuova superficie deve entrare nello schermo da un'altra posizione e diventare il nuovo punto di focus per l'attenzione dell'utente.

Quando devono essere create delle nuove superfici, queste solitamente nascono per trasformazione **radiale** o **rettangolare** dal punto di tocco dell'utente. Quando più superfici devono essere create contemporaneamente, esse devono entrare nello schermo in **veloce successione** e secondo una **direzione unica**. Se delle superfici vengono create senza l'input dell'utente, queste devono comparire in modo naturale attraverso l'uso di sfocatura, movimento e dimensionamento.

Le animazioni hanno varie applicazioni e possono essere usate per unire bellezza e funzionalità.

Le icone e i pulsanti possono avere duplici scopi e trasformarsi a seconda delle esigenze. Per esempio, l'icona di un menù può essere trasformata in un pulsante per tornare alla schermata precedente e viceversa. Inoltre, per puro scopo di bellezza e rifinitura, anche le icone delle applicazioni possono essere animate.

Si può inoltre integrare l'interfaccia con illustrazioni e immagini animate che rendano più leggera e giocosa l'esperienza utente.

## 3.3 Stile

### 3.3.1 Colore

Con Material Design vengono creati schemi colore comprensivi di **variazioni scure e chiare dei colori primari e secondari**, in cui viene tenuto conto dell'accessibilità del testo applicato a sfondi di colore diverso. Tutto questo viene scelto, deciso e previsualizzato attraverso lo strumento *color tool*[1].

La scelta dei colori si basa su palette che partono da un **colore primario** (il 500) e lo espandono integrando tutte le sfumature necessarie per la formazione dell'interfaccia utente. Con queste ricche palette è quindi possibile rappresentare luci e ombre di varie tipologie e intensità.

Il colore primario è il colore che appare più frequentemente nell'applicazione, il **secondario** viene utilizzato per evidenziare le parti importanti dell'interfaccia utente. Se si sceglie di non utilizzare un colore secondario, è possibile evidenziare gli elementi importanti con il colore primario.

Per creare del contrasto tra gli elementi, si sfruttano le tonalità chiare o scure del colore primario.

Il colore secondario può essere complementare o meno del colore primario, ma non deve essere una sua tonalità chiara o scura: dovrebbe contrastare con gli elementi che lo circondano, in modo da attirare l'attenzione dell'utente.

I colori secondari sono principalmente usati per:

- **bottoni**, FAB e bottoni con testo
- campi **testuali**, cursori e selezioni testuali
- **barre** di progresso
- controlli di selezione e **slider**
- **link**
- **Titoli**



images/palette.jpg

Figura 5: esempio di palette

Il colore può essere usato per evidenziare le **gerarchie** tra gli elementi dell'interfaccia utente: con combinazioni di colori particolari si possono far risaltare maggiormente i componenti più importanti.

Il colore può anche essere utilizzato per esprimere il **significato** di vari elementi sullo schermo. Si può, inoltre, esprimere lo stato di alcuni oggetti con tonalità ben precise e che possano essere notate facilmente in caso di cambi di stato.

Il testo deve essere sempre **leggibile**, quindi bisogna settarne il colore dipendentemente dal colore dello sfondo su cui è posizionato. Per esempio, su sfondi chiari va utilizzato un testo di colore grigio scuro (o nero), mentre su sfondi scuri va utilizzato un testo di colore grigio chiaro (o bianco). Il testo di importanza maggiore, inoltre, deve avere opacità maggiore, mentre testi di minore importanza (suggerimenti) devono avere opacità bassa. L'utilizzo di testo colorato su sfondi colorati va limitato agli elementi importanti, e non utilizzato sempre.

Le scelte dei toni e dei colori per l'applicazione vanno a formare un **tema**. Devono sempre disponibili sia il tema chiaro che il tema scuro.

### 3.3.2 Icone

Le **icone**[3] di Material Design utilizzano le forme geometriche per rappresentare idee, potenzialità o argomenti.

L'icona di un prodotto[9] è l'espressione grafica del suo scopo e dei servizi forniti da esso. Tutte le icone di uno stesso brand dovrebbero essere diverse, ma avere un tratto distintivo che le unifichi.

Ogni icona, secondo i principi di Material Design, deve essere concepita avendo come riferimento un oggetto fisico fatto di carta, ma essere molto più semplice della controparte reale. L'icona, quindi, deve presentare ombre e caratteristiche che mimino come l'oggetto fisico interagisce con la luce.

È stata definita una **griglia di riferimento**, la *product icon grid*, che possa facilitare la coerenza tra le icone e che indica alcune regole per posizionare i vari elementi in essa.



Figura 6: Product icon grid

Le linee guida presenti nella griglia aiutano a mantenere delle proporzioni visuali coerenti con altre icone di prodotti correlati.

Le icone sono formate da cinque elementi principali:

1. **finitura** (ne determina il riflesso superficiale)
2. materiale di **sfondo**
3. materiale **principale** (rialzato rispetto allo sfondo)
4. **colore** (risponde in modo diverso alle varie ombre e luci)
5. **ombre** (più intense nella parte inferiore destra)

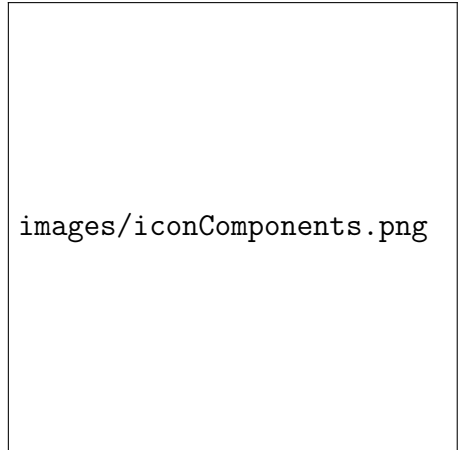


Figura 7: Componenti icone

Alcune linee guida comuni:

- gli elementi colorati non presentano ombre aggizionali
- le superfici sovrapposte formano spigoli e ombre
- elevare un oggetto ne aumenta l'importanza
- evitare di "piegare" gli elementi principali
- non sovrapporre più di due elementi
- non distorcere mai le icone
- nelle icone con silhouette umane, usare sfondi semplici

Material Design fornisce una serie di icone di sistema standard, che simboleggiano azioni, file, dispositivi e directory comuni. Per queste icone valgono la maggior parte dei principi delle icone di prodotto (griglie, forme geometriche, etc.).

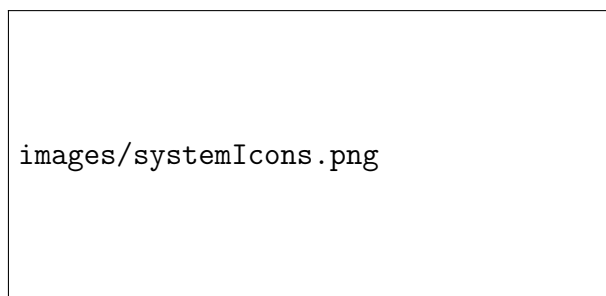


Figura 8: Icone di sistema di Material Design

### 3.3.3 Immagini

Le **immagini**[4] non servono solo per abbellire, ma permettono di comunicare, di differenziare il prodotto e di coinvolgere l'utente.

Possono comunicare informazioni specifiche che rendano il contenuto più **accessibile** e immediato.

Devono essere diverse, dinamiche e soprattutto **coerenti** con il contesto in cui vengono presentate. È possibile nascondere delle parti per rendere l'interfaccia più immersiva.

È consigliabile sfruttare sia fotografie che illustrazioni grafiche. Le **fotografie** permettono di riferirsi ad entità e storie specifiche, mentre le **illustrazioni** vengono utilizzate per rappresentare concetti più generali. Inoltre, le fotografie devono essere specifiche e personali: è meglio evitare di utilizzare immagini stock.

Le immagini utilizzate devono avere uno **scopo** ed essere chiare nell'esprimere il loro soggetto principale. Questa chiarezza è ottenibile attraverso l'uso dei colori e dello sfocato.

Attraverso immagini correlate è possibile creare una storia che renda l'applicazione meno meccanica e più naturale e umana.

Le **dimensioni** delle immagini devono essere adeguate a diverse tipologie e grandezze di schermo. Sono da preferire immagini di grandi dimensioni, in modo che esse non risultino sgranate o di scarsa qualità.

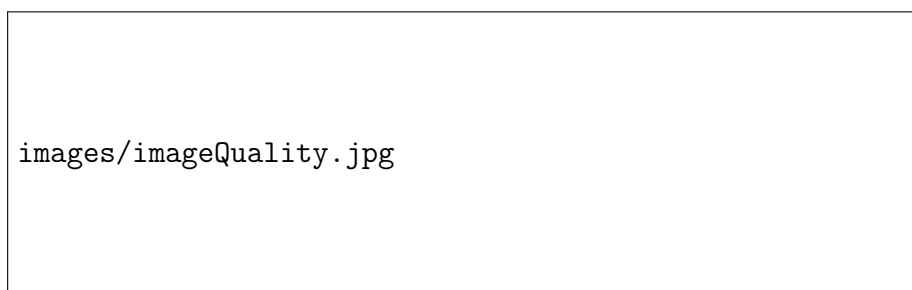


Figura 9: La qualità delle immagini è fondamentale

È utile dimensionare diversamente le immagini per creare diversi livelli di importanza.

Per permettere di inserire del testo sulle immagini è utile sfruttare gli *scrim*, degli strati di materiale traslucido posti sopra le foto, che scuriscono lievemente lo sfondo aumentando la leggibilità del testo.

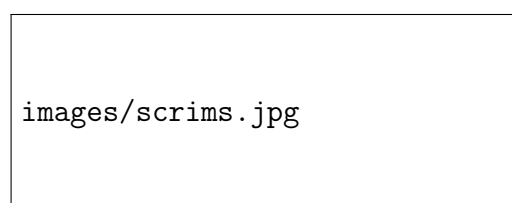


Figura 10: Esempio di scrim



### 3.3.4 Tipografia

È possibile e utile dividere le tipologie di **testi**[12] in tre categorie:

- *inglesi e simili*, che non necessitano di dimensioni maggiori di quelle standard
- *alti* (*Tall*), che necessitano di un'altezza maggiore per rappresentare simboli particolari tipici dei linguaggi del sud e sud-est asiatico
- *densi* (*Dense*), che necessitano di un'altezza maggiore per rappresentare simboli più grandi, come quelli giapponesi, cinesi e coreani.

Ogni lingua è stata inserita in una delle tre categorie.

Il carattere standard di Android è *Roboto*. Esso è caratterizzato dall'essere leggermente più largo e tondeggiante. Queste caratteristiche gli donano una maggiore chiarezza e lo rendono più ottimistico.

Il carattere standard di Chrome, invece, è *Noto*. Questo ha delle metriche compatibili con Roboto, e non differisce particolarmente da esso. Esso è utilizzato per rappresentare i simboli particolari dei testi alti e dei testi densi (caratteri cinesi, hindi, etc.).



Figura 11: Roboto e Noto

Esistono delle versioni di Roboto e Noto che implementano dei suggerimenti all'interno dei caratteri (*hint*), utili per modificare il testo quando bisogna rappresentarlo su dispositivi con scarsa risoluzione. Questo tipo di testo è più pesante e, per questo, viene adottato solo sui dispositivi che implementano i testi con suggerimenti (Chrome OS, Windows e Linux).

Le dimensioni e gli stili dei testi sono definiti **a priori** per evitare una grande confusione derivante dalle infinite possibilità disponibili. Ne derivano delle regole specifiche su come applicare i vari stili in contesti differenti.

Sono inoltre definiti i livelli di **contrasto** dei testi: non deve essere troppo chiaro da impedirne la lettura e nemmeno troppo contrastato. Il contrasto può anche derivare dalle varie scelte di colore adottate. Di conseguenza, esso va settato anche considerando tali aspetti.

È consigliato usare i **testi dinamici**: dei testi che modificano il proprio stile in base allo spazio disponibile.

È definita anche la **lunghezza** consigliata delle righe di testo: una riga corretta dovrebbe contenere circa 60 caratteri. Questo aumenta la leggibilità del testo e riduce la stanchezza percepita dall'utente nella lettura di righe troppo lunghe.



Figura 12: Scala di stili

### 3.3.5 Scrittura

I testi devono essere **comprensibili** da chiunque, a prescindere dalla loro cultura e dalla loro lingua.

Testi **brevi e chiari** rendono l'interfaccia più comprensibile e stabile. Devono riferirsi all'utente in un modo consistente: o in seconda persona o in prima persona, non un mix delle due. Il linguaggio deve essere **diretto e semplice** e i testi devono essere al presente. I verbi e i vocaboli devono essere usati in modo **consistente** attraverso tutta l'applicazione. La punteggiatura deve essere essenziale, per non appesantire eccessivamente la lettura.

L'utilizzo del **tono** con cui ci si riferisce all'utente non è casuale. Il testo deve essere rispettoso, amichevole e incentrato sull'utente.

Queste ed altre regole di scrittura ci danno la possibilità di aumentare la fruibilità delle nostre applicazioni e ci permettono di **coinvolgere** maggiormente l'utente.

## 3.4 Layout

In Material Design, il layout è guidato e definito attraverso l'uso degli elementi tipografici (testi, immagini, colori, etc.) e si basa sulla creazione di una gerarchia che possa definire il senso del layout e che possa guidare l'utente attraverso esso. Per fare questo, vengono sfruttati gli elementi del mondo della stampa, ripetendo le stesse griglie e gli stessi oggetti per ottenere un prodotto consistente anche attraverso schermi diversi.

L'idea di base, quindi, è quella di relazionarsi all'interfaccia come se essa fosse un foglio di carta e come se le azioni che l'interfaccia compie seguissero il comportamento che la carta ha quando viene piegata, tagliata e modificata in generale.

Le superfici sono costituite da fogli di materiale. Quando due fogli sono hanno un lato in comune, essi sono "cuciti" (seams) insieme e si muovono insieme. Quando, invece, due fogli si sovrappongono, essi danno vita ad uno scalino (step) e si muovono indipendentemente l'uno dall'altro.

Il Floating Action Button (FAB) è un foglio circolare di materiale diviso dalla barra dei comandi che rappresenta una singola azione. Esso può essere posizionato su cuciture o su gradini, se il suo contenuto è relativo a tali, oppure indipendentemente dagli elementi circostanti.

Alla base del layout dell'interfaccia ci sono le misure del nostro schermo e la sua densità di pixel.

La densità di pixel è il numero di pixel che sono contenuti in un pollice del nostro schermo. Gli elementi dell'interfaccia appariranno quindi più piccoli su schermi ad alta densità di pixel e più grandi su schermi a bassa densità.

Per ovviare al problema delle differenze di densità, si utilizzano i "Density-independent pixels" (dips), ossia unità flessibili che hanno dimensione costante su qualsiasi schermo di qualsiasi grandezza e definizione.

Quando si sviluppa un'applicazione è essenziale usare i dp per dimensionare gli elementi dell'interfaccia. Per i font, invece, si usano gli sp (Scaleable Pixels), che hanno la stessa funzione dei dp.

Per quanto riguarda le immagini, si utilizzano i seguenti rapporti (ratio):

Risoluzione schermo	dpi	Rapporto pixel (pixel ratio)	Dimensioni immagine (in pixel)
xxxhdpi	640	4.0	400 x 400
xxhdpi	480	3.0	300 x 300
xhdpi	320	2.0	200 x 200
hdpi	240	1.5	150 x 150
mdpi	160	1.0	100 x 100

Tabella 1: Rapporti per dimensionare le immagini

Tutti gli elementi sono disposti su una riga formata da quadrati con lato 8dp. I caratteri, invece, sono allineati su una griglia di quadrati di lato 4dp.

Material Design fornisce dei template per disporre gli elementi in liste, in viste con dettaglio e in menu di navigazione. Questo template sono differenziati a seconda della tipologia di dispositivo.

Le dimensioni degli elementi possono essere definite in relazione alle dimensioni di un elemento chiave, attraverso gli increment. Tale pratica, tuttavia, è utilizzata principalmente per dispositivi desktop e non per smartphone.

La dimensione di oggetti toccabili dall'utente dovrebbe essere di almeno 48 x 48 dp.

Sono definite le strutture di base delle applicazioni nei casi esse siano per tablet, smartphone o desktop. Per esempio, la struttura di un'applicazione per dispositivi mobili deve presentare un floating action button e una barra di applicazioni costantemente presenti. Inoltre può presentare altri pulsanti opzionali.

Per distribuire il contenuto, è consigliato dividere l'interfaccia in regioni e usare le card.

La barra dell'applicazione (app bar) è una toolbar speciale utilizzata per mostrare il brand e delle azioni specifiche. Le toolbar possono essere di diversi tipi e avere diversi utilizzi.

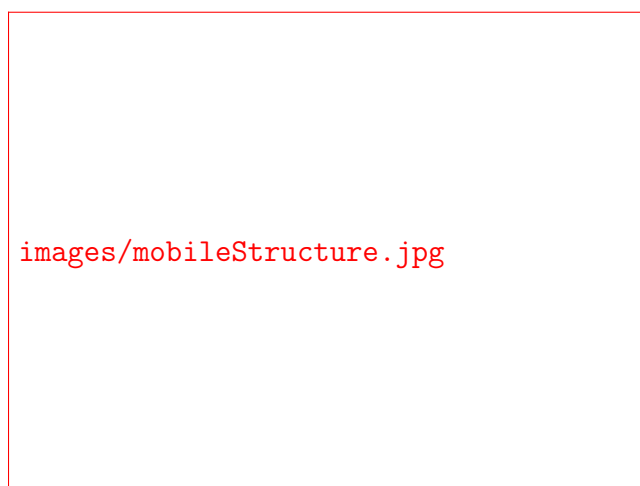


Figura 13: Struttura per mobile



Figura 14: App bar

Nella barra dell'applicazione, le icone dovrebbero essere tutte dello stesso colore. Il titolo può avere un colore diverso.

È utile integrare la barra dell'applicazione con dei menu a tendina, che si sovrappongono alla barra.

La barra di stato, in Android, contiene le icone delle notifiche e altre icone di sistema. È possibile nascondere tale barra implementando l'utilizzo della modalità a tutto schermo dell'applicazione. È anche possibile settare il colore di questa barra, per renderla più o meno evidente.

Le barre laterali possono essere visibili permanentemente oppure sovrapporsi all'interfaccia quando necessarie. Le barre di sinistra contengono spesso contenuti relativi all'identità dell'utente o alla navigazione, mentre quelle di destra contengono elementi secondari rispetto al contenuto principale della pagina.

Material Design sfrutta i layout reattivi, che adattano l'interfaccia a qualsiasi dimensione di schermo e ne garantiscono la consistenza su dispositivi differenti.

Le interfacce vengono adattate per delle risoluzioni specifiche (breakpoints), per le quali sono definiti una serie di parametri utili a dimensionare l'interfaccia in modo corretto.

Tutto questo sistema è fondato sull'utilizzo di una griglia a 12 colonne.

Il dimensionamento dell'interfaccia avviene attraverso l'uso di pattern:

- rivela (reveal)
- trasforma (transform)
- dividi (divide)
- ridisponi (reflow)
- espandi (expand)
- posiziona (position)

La modalità di schermo condiviso (split screen) permette di visualizzare sullo schermo due activity contemporaneamente. Essa deve essere attivata dall'utente: le applicazioni non possono attivarla autonomamente. Inoltre, tale modalità rimane attiva finché l'utente non la chiude o prova ad attivarla con un'applicazione che non la supporta.

La porzione di schermo attribuita a ciascuna delle due attività può essere ridimensionata dall'utente.

Il contenuto dell'interfaccia delle due activity viene scalato e riposizionato grazie all'utilizzo delle interfacce adattabili.

## 3.5 Componenti

Nel seguito verranno descritti i principali componenti [6] di Material Design 1.

### 3.5.1 Barra di navigazione inferiore

La **barra di navigazione inferiore** permette di cambiare rapidamente tra varie viste e attività all'interno dell'applicazione. È utilizzata principalmente sui dispositivi mobili, mentre su dispositivi con schermi più grandi lo stesso effetto è ottenibile con una barra di navigazione laterale.

Nella barra vanno inserite da 3 a 5 icone che portano ad alternative di simile importanza (**le più importanti** dell'applicazione).

Quando una schermata è attiva, nella barra va mostrato anche il nome dell'attività. Nel caso la barra presenti 3 icone, allora va mostrato anche per le viste inattive il nome dell'attività. Nel caso, invece, le icone siano più di 3, allora vanno mostrate solo le icone. I nomi delle attività devono essere corti. Le icone dovrebbero occupare la stessa quantità di spazio ciascuna. La divisione della barra può essere cambiata in modo che l'icona corrispondente alla vista attualmente attiva occupi più spazio.

La barra di navigazione non dovrebbe essere utilizzata per raggiungere la schermata delle impostazioni o quella delle preferenze.

La barra può scomparire o comparire durante lo scrolling della schermata, oppure essere costantemente visibile.

images/boottomNavigation.png

Figura 15: Barra di navigazione inferiore

### 3.5.2 Bottom sheets

Le **bottom sheets** sono delle pagine inferiori che salgono dalla parte bassa dello schermo per mostrare contenuto aggiuntivo. Esse compaiono solo dopo un'azione specifica dell'utente e possono essere di due tipi:

- **modali**, che sono dei menu aggiuntivi ce possono presentare contenuti relativi ad altre applicazioni (la loro elevazione è maggiore di quella del resto dell'interfaccia dell'applicazione)
- **persistenti**, che presentano esclusivamente contenuto appartenente all'applicazione (hanno la stessa elevazione dell'interfaccia dell'applicazione)

Le bottom sheet persistenti sono perennemente visibili, anche quando non in uso, e integrano la vista principale con del contenuto aggiuntivo.

Le bottom sheet modali sono **alternative ai menu** e si sovrappongono a parte della vista principale, oscurando leggermente quest'ultima. Inoltre, per proseguire nell'utilizzo dell'applicazioni, esse vanno chiuse esplicitamente.

Esse possono contenere dei link che portano ad altre applicazioni, che possono essere aperte in una schermata provvisoria, che si sovrappone a quella dell'applicazione principale.

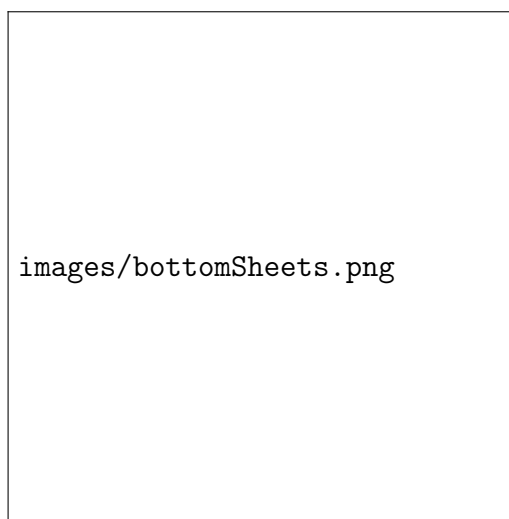


Figura 16: Bottom sheet

### 3.5.3 Pulsanti

I **pulsanti**, al loro tocco, attivano l'attività o l'azione ad essi associata. Essi possono presentare del testo o un'icona che descriva l'azione che svolgono.

I pulsanti possono essere piatti e avere la stessa elevazione del testo, oppure essere rialzati.

Ci sono 3 tipologie standard di pulsanti:

- ***Floating action button***: un pulsante circolare rialzato che si alza e mostra un'azione quando viene premuto
- ***Pulsante rialzato (raised button)***: un pulsante rialzato, di solito rettangolare, che si alza e mostra un'azione quando viene premuto
- ***Bottone piatto (flat button)***: un pulsante costituito da testo che, quando viene premuto, non si alza ma mostra un'azione e si colora (da usare nelle barre di navigazione e nei testi)

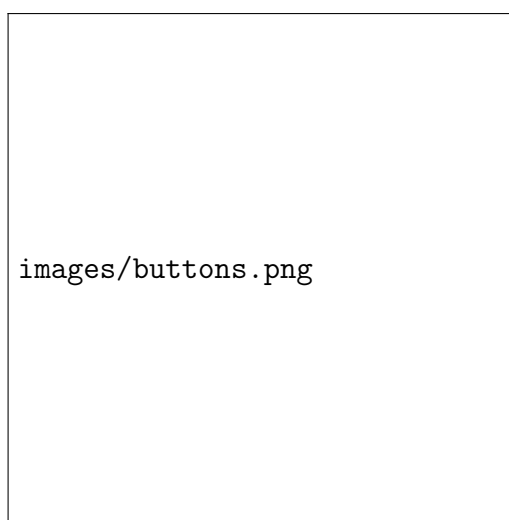


Figura 17: Pulsanti

I floating action buttons sono i pulsanti con il livello di importanza maggiore, e per questo sono da utilizzare con discrezione.

Il testo presente sui pulsanti dovrebbe essere maiuscolo mentre, per i linguaggi che non presentano lettere maiuscole, si distinguono i pulsanti piatti dal testo normale attraverso l'uso del colore.

I pulsanti possono anche mostrare, alla loro pressione, un menu a tendina con diverse opzioni. Un'altro esempio sono i toggle buttons, ossia dei pulsanti che una volta premuti, cambiano stato e vi rimangono fino a che l'utente non li premerà nuovamente. Questi sono utilizzati per settare delle impostazioni o dei parametri, per aggiungere degli elementi a delle liste, o per altri scopi.

### 3.5.4 Floating Action Button

Il **floating action button** (FAB) rappresenta l'azione principale della nostra applicazione. Quando premuto, cambia colore e si eleva e può mostrare diverse azioni correlate.

È consigliato inserire **un solo FAB per schermata** per rappresentare l'azione più comune. Esso può essere di due diverse dimensioni: la dimensione standard e una dimensione più piccola (per creare un senso di continuità con altri elementi dell'interfaccia).

Non tutte le schermate devono presentare un floating action button: solo le schermate che ne hanno necessità.

I FAB dovrebbero rappresentare azioni positive e creazionali più che distruttive. Essi, inoltre, devono essere **circolari e monodimensionali**.

Se un'azione è presente in più schermate successive, il floating action button deve rimanere visibile.

Il pulsante, quando viene premuto, può dare vita a diverse animazioni o trasformarsi in materiali diversi, come una toolbar. La trasformazione del FAB in toolbar, e viceversa, è particolarmente utile in determinati meccanismi di scrolling, per evitare di togliere spazio necessario alla fruizione del contenuto.

Un FAB, una volta premuto, può dividersi in altri FAB più piccoli (minimo 3 e massimo 6).

### 3.5.5 Cards

Le **card** sono fogli di materiale, utili per fornire informazioni specifiche su determinati argomenti. Possono contenere testo, immagini e/o link ad altri oggetti e possono mostrare contenuti di dimensioni diverse e variabili.

Le card sono consigliate, appunto, per mostrare del contenuto composto da più elementi di tipo e dimensione diversi. Esse sono utili per mostrare oggetti che

images/Fab2.png

Figura 18: Floating Action Button

images/cards.png



fanno parte di collezioni e che contengono bottoni o testi interattivi, di lunghezza o dimensioni non standard e altamente variabili.

Ogni card può essere un punto di accesso per del contenuto più ampio e specifico.

Le card che fanno parte di collezioni possono essere spostate e riposizionate dall'utente attraverso delle gesture specifiche. Lo scrolling tra card avviene in direzione verticale e i pulsanti sono solitamente disposti nella parte inferiore della card.

### 3.5.6 Chips

Le **chips** permettono di mostrare contenuti brevi che, al tocco, forniscono contenuto più dettagliato o danno vita ad un'azione particolare. Sono molto utili per **riassumere** brevemente informazioni di contatto.

Una chip può contenere una piccola immagine, o un'icona, un titolo e una breve informazione.

Le chips possono essere cancellabili al tocco. In tal caso deve essere presente un'icona di cancellazione.



Figura 20: Chips

### 3.5.7 Tabelle di dati

Le tabelle sono utili per mostrare dati semplici e grezzi. Esse possono essere integrate in una card o in altri componenti e possono fornire la possibilità di manipolare i dati mostrati.

Le tabelle sono formate da una prima riga di intestazione e da una serie di righe contenenti i dati. Queste righe possono contenere dei check-box nel caso in cui l'utente possa dover selezionare i dati.

Le tabelle dovrebbero contenere tre o più righe, non meno.

Può inoltre essere implementato il column sorting, ossia l'ordinamento definito dai valori assunti dalle righe per una determinata colonna.

È possibile permettere all'utente di inserire del testo all'interno delle righe della tabella. Inoltre si possono integrare le tabelle con delle icone di modifica o dei menu a tendina



Figura 21: Tabella di dati

### 3.5.8 Finestre di dialogo

Le **finestre di dialogo** permettono di mostrare all'utente delle informazioni importanti, delle scelte necessarie, o di dividere varie attività minori.

Tali finestre raccolgono tutto il focus dell'interfaccia su di esse e lo mantengono fino a che non vengono chiuse. Per questo sono da utilizzare con parsimonia.

Le finestre di dialogo **interrompono la normale attività** sovrapponendosi a parte del contenuto mostrato precedentemente e oscurandolo. I dialoghi non dovrebbero dare vita ad altri dialoghi, ma potrebbero portare a finestre successive che occupano l'intero spazio dello schermo disponibile.

Alcuni dialoghi possono contenere liste scrollabili.

Gli **alert** sono tipi di dialoghi che interrompono urgentemente l'interfaccia e che informano l'utente di una determinata situazione, mostrando una decisione da prendere.

Le finestre possono essere utili ad effettuare delle scelte oppure a confermare delle scelte predefinite.



Figura 22: Finestre di dialogo

### 3.5.9 Divisori

Un **divisore** è una linea che divide il contenuto in liste e in layout di pagina. Essi sono alternative più forti e incisive all'utilizzo dello spazio bianco.

I divisori sono utili per creare delle **gerarchie**, ma vanno utilizzati con criterio perché possono appesantire l'interfaccia e renderla meno efficiente. Essi possono essere utilizzati anche per dividere varie sezioni dell'interfaccia utente.

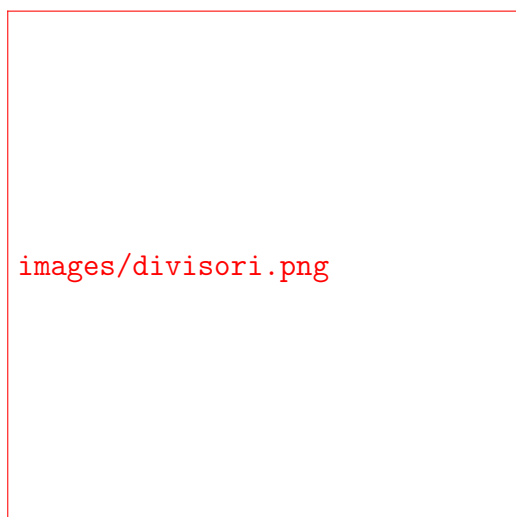


Figura 23: Divisori

### 3.5.10 Pannelli espandibili

Un pannello espandibile è un tipo di contenitore che può essere espanso o collegato ad altri tipi di oggetti, tipo una card. Questo tipo di pannello può essere utile per modificare delle impostazioni o selezionare delle opzioni, oppure per creare degli oggetti per campagne pubblicitarie che mostrano dei dettagli o delle informazioni più specifiche.

Quando il pannello è contratto, esso mostra un riassunto del contenuto. Quando invece esso viene espanso, permette di aggiungere o modificare delle preferenze, oppure di visualizzare più informazioni.



Figura 24: Pannelli espandibili

### 3.5.11 Griglie

Le griglie a griglia sono alternative alle normali liste e consistono in una serie di celle disposte in righe orizzontali e in colonne verticali. Esse aumentano la comprensione visiva del contenuto e sono da prediligere nel caso in cui le celle contengano tipi di contenuto simili e omogenei.

Nelle celle della griglia possono essere inserite immagini o righe di testo.

Il testo della griglia non deve essere contenuto in una cella, ma in un contenitore differente, in modo da evidenziarlo e da distinguerlo dal contenuto.

Il contenuto delle celle può presentare un titolo e un'azione primaria e una secondaria. Le azioni possono aprire oggetti secondari, come una card.

L'azione primaria occupa l'intera cella e non presenta griglie, mentre l'azione secondaria è solitamente posta ai margini della cella e presenta un testo e un'icona.

Lo scrolling avviene in direzione verticale. Lo scrolling orizzontale, infatti, è da evitare perché interferisce con le meccaniche di lettura e, di conseguenza, limita la fruibilità del contenuto. Questa osservazione non vale per le griglie di immagini a singola riga, per le quali è possibile sfruttare lo scrolling orizzontale.

È consigliabile fornire all'utente la possibilità di ridisporre il contenuto secondo vari parametri (data, ordine alfabetico, etc.).

I testi troppo lunghi vanno troncati: è da evitare la modifica della dimensione della cella.




Figura 25: Griglie

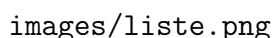


Figura 26: Liste

### 3.5.12 Liste

Le **liste** sono composte da elementi disposti per righe, che vanno a formare un unico oggetto. Ogni riga presenta un testo, che contiene l'azione principale, e può presentare delle icone o del testo per le azioni secondarie.

Le liste sono ottime per visualizzare dati simili.

Una lista è composta da una singola colonna e le righe hanno uguali dimensioni, sconsigliata nel caso il titolo fosse composto da più di tre righe (usare le card) e nel caso in cui si vogliano mostrare immagini (usare le griglie).

Le liste si scrollano in **verticale**, perché hanno una forte componente testuale e di lettura. È consigliabile permettere all'utente di spostare le righe con delle semplici gesture. Le righe, inoltre, possono essere filtrate e riordinate secondo determinati parametri.

In ogni riga, la maggior parte dello spazio deve essere dedicata all'attività principale, per attirare il focus dell'utente su tale contenuto.

Le **azioni**, primarie o secondarie, possono aprire altri oggetti, come delle card.

Per le liste, sono disponibili dei **controlli** che mostrano informazioni o azioni riguardando le righe delle liste, oppure indicano lo stato della riga. Questi controlli sono mostrati attraverso icone posizionate nella parte destra, per le azioni secondarie o per le informazioni aggiuntive, o sinistra della riga, per l'azione primaria o per lo stato.

Questi controlli possono sfruttare dei check-box, degli switch, delle icone per riordinare, delle icone per espandere il contenuto della riga e delle icone che suggeriscono all'utente l'effetto di uno swipe laterale.

Per i menu, speciali tipi di liste, sono utili le icone di spunta, le informazioni aggiuntive in riga e gli indicatori per raggiungere i menu interni a tale riga.

### 3.5.13 Menu

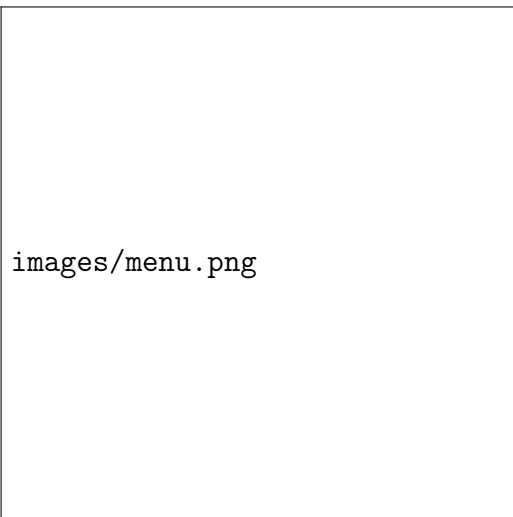
I **menu** sono delle finestre che presentano una lista di scelte e compaiono solo dopo una determinata azione dell'utente, come la pressione di un pulsante. Al momento della comparsa, la finestra del menu si sovrappone temporaneamente all'interfaccia e viene visualizzata in una posizione legata al pulsante che ne ha scaturito la visualizzazione. La finestra rimane visibile fino a quando l'utente non effettua una scelta oppure chiude il menu.

Ogni elemento del menu, una volta selezionato, modifica il comportamento di altri oggetti dell'applicazione. Un elemento può contenere, oltre al testo, un'icona o delle immagini di ridotte dimensioni e può essere disattivato in base allo stato dell'applicazione nel momento della sua creazione. Nel caso un elemento non sia disponibile è necessario renderlo noto all'utente, ad esempio schiarendone il contenuto.



images/listControls.png

Figura 27: Controlli per le liste



images/menu.png

Figura 28: Menu

In base alla loro dimensione i menu possono essere scorribili verticalmente. Inoltre è possibile creare dei menu a cascata, ma questa opzione è preferibile solo su desktop.

I menu non devono essere utilizzati come metodo di navigazione principale all'interno di un'applicazione.

### 3.5.14 Picker



Figura 29: Picker

I *picker* sono delle interfacce utili a permettere all'utente di selezionare un valore da un determinato insieme e possono essere di due tipi:

- **data picker**, utilizzati per selezionare una data da un calendario;
- **time picker**, utilizzati per selezionare un orario da un orologio.

### 3.5.15 Barre di avanzamento

Le **barre di avanzamento** indicano all'utente che l'applicazione sta caricando del contenuto che non è immediatamente disponibile. In base all'operazione che si sta eseguendo, la barra può avere uno stato determinato, ovvero è possibile visualizzare a quale livello di avanzamento ci si trovi, oppure indeterminato, cioè

non è possibile sapere quanto tempo manca al termine del processo.

La barra può essere lineare oppure circolare, in entrambi i casi l'indicatore deve sempre progredire dallo 0% al 100%.

Nel caso siano in atto processi multipli, è opportuno mostrare una sola barra di avanzamento che li rappresenti tutti.

### 3.5.16 Slider

Gli **slider**, permettendo all'utente di selezionare un valore in un determinato range, come ad esempio luminosità dello schermo e volume dell'audio. Gli slider possono essere continui, ovvero permettere all'utente di selezionare un qualsiasi valore nel range prestabilito, oppure discreti, cioè dare la possibilità all'utente di

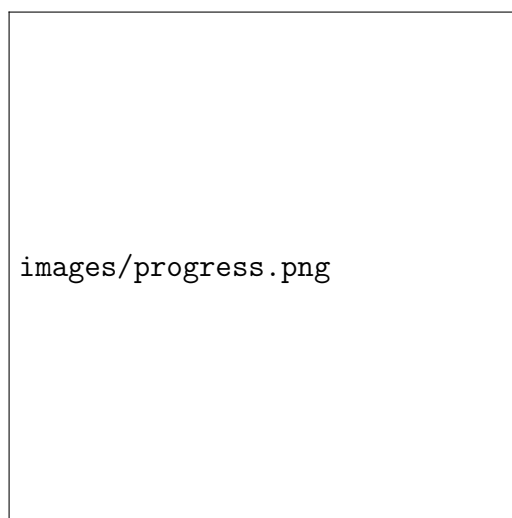


Figura 30: Barre di avanzamento

selezionare solo dei valori specifici. Inoltre, è possibile far digitare all'utente un valore numerico specifico.

### 3.5.17 Snackbar



Figura 31: Slider



Figura 32: Snackbar

Le *snackbar* visualizzano sulla parte inferiore dello schermo dei brevi messaggi temporanei. A differenza dei *toast*, le *snackbar* forniscono all'utente la possibilità di interagire con il messaggio, ad esempio permettendo di annullare l'operazione appena eseguita. Questo tipo di avvisi scompare automaticamente dopo un determinato intervallo di tempo e ne può essere mostrato solo uno alla volta.

### 3.5.18 Stepper

Gli **stepper** sono utili a mostrare all'utente il progresso attraverso una serie di passaggi numerati, in modo da rendere più chiaro il flusso di esecuzione. Questi possono essere verticali/orizzontali, modificabili/non modificabili dall'utente e lineari/non lineari, nel caso i passaggi siano o meno vincolati da un ordine specifico. Determinati punti possono essere resi opzionali.

È consigliato evitare l'utilizzo di più serie di stepper all'interno della stessa pagina.

### 3.5.19 Sottotitoli

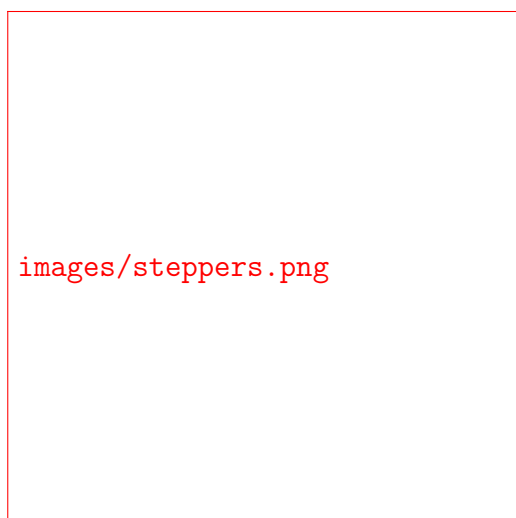


Figura 33: Stepper

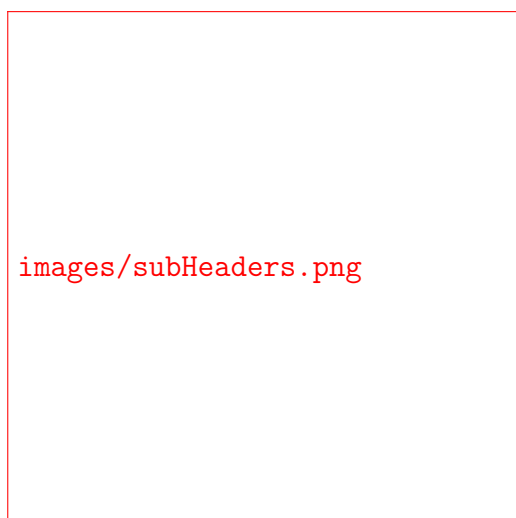


Figura 34: Sottotitoli

I **sottotitoli** sono utili a definire varie sezioni all'interno di contenitori come liste o griglie. Essi possono essere posti in linea con il titolo principale, oppure essere allo stesso livello del contenuto.

Durante lo scorrimento del contenitore i sottotitoli restano ancorati alla parte superiore dello schermo finché non si passa al titolo successivo.

### 3.5.20 Tab

Le **tab** sono utili di differenziare la visualizzazione di determinati insiemi di dati, ad esempio in categorie, permettendo una più precisa organizzazione interna dell'applicazione. L'obiettivo è fornire una navigazione di alto livello che permetta di suddividere gruppi di dati della stessa tipologia in sezioni differenti. Perché

la struttura delle suddivisione sia comprensibile all'utente le tab devono essere ordinate in base al contenuto e il gli elementi al loro interno non devono essere ulteriormente suddiviso.

Le tab non devono contenere oggetti che implementano l'azione di swipe (scorrimento), in quanto si tratta dell'azione che permette la navigazione tra di esse.

Se il numero di tab da visualizzare è ridotto, queste possono essere fisse, altrimenti possono essere scrollabili orizzontalmente. Utilizzare una visualizzazione fissa delle tab aiuta l'utente a prendere più facilmente confidenza con l'applicazione e sviluppare una memoria muscolare che gli permetta di interagire in modo più familiare con l'interfaccia.



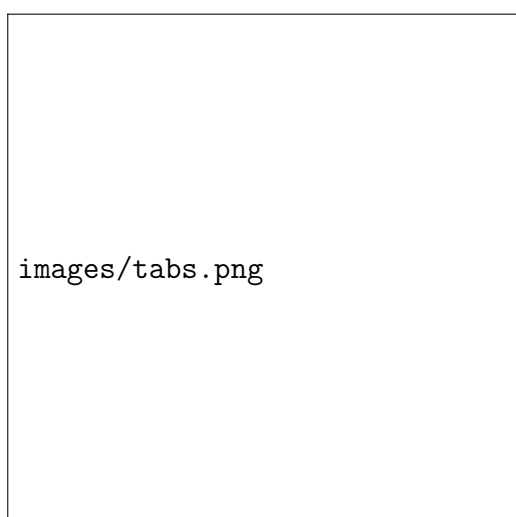


Figura 35: Tab

### 3.5.21 Campi testuali

I **campi testuali** permettono all'utente di inserire, modificare e selezionare del testo. L'utente deve essere in grado di intuire facilmente che questi campi siano modificabili e quale sia l'input richiesto. Durante l'inserimento, il colore del campo deve cambiare e dev'essere visualizzato un cursore per indicare uno stato attivo della digitazione.

È consigliato visualizzare un testo d'aiuto che indirizzi l'utente nel comprendere correttamente la tipologia di contenuto richiesto. Questo testo può essere sostituito da un messaggio di errore nel caso in cui l'utente inserisca un valore non valido.

Inoltre è possibile inserire un asterisco che indichi l'opzionalità del campo, un contatore di caratteri nel caso ne siano richiesti un numero limitato o delle icone, ad esempio quella di cancellazione di un carattere.

Il testo inserito può essere disposto su un numero di righe predefinito oppure su un'area di testo adattabili alla dimensione di quest'ultimo. È inoltre possibile formattare in automatico il testo, in modo che durante l'inserimento esso si disponga in modo da aumentare la leggibilità e la comprensione del contenuto.



Figura 36: Campi testuali

### 3.5.22 Toolbar

La **toolbar** è una barra contenente le principali azioni che possono essere eseguite nella schermata visualizzata. Questa può contenere un titolo e un insieme di azioni.

### 3.5.23 Suggerimenti

I *suggerimenti* sono caselle di testo che compaiono quando l'utente tiene premuto su un elemento dell'interfaccia. Questi sono utili a fornire all'utente delle informazioni sull'elemento che è stato cliccato, ad esempio per un pulsante di cui viene visualizzata solo l'icona o per un'immagine cliccabile.

### 3.5.24 Widget

I **widget** permettono ad un'applicazione di visualizzare i dati e le funzionalità principali direttamente nella schermata di Home del dispositivo. Questi possono essere spostati e ridimensionati a piacimento dall'utente. Quando un widget viene premuto, la relativa applicazione viene lanciata e il contenuto selezionato può essere approfondito.

In particolare, i widget possono mostrare:

- informazioni, viene visualizzato un'insieme ridotto di dati importanti e la loro variazione nel tempo;
- collezioni, viene visualizzata una lista di elementi della stessa tipologia. Al tocco di un elemento è possibile approfondirne il contenuto all'interno dell'applicazione;
- controlli, vengono visualizzati i pulsanti e le azioni utilizzati più frequentemente, così da essere subito a disposizione dell'utente.

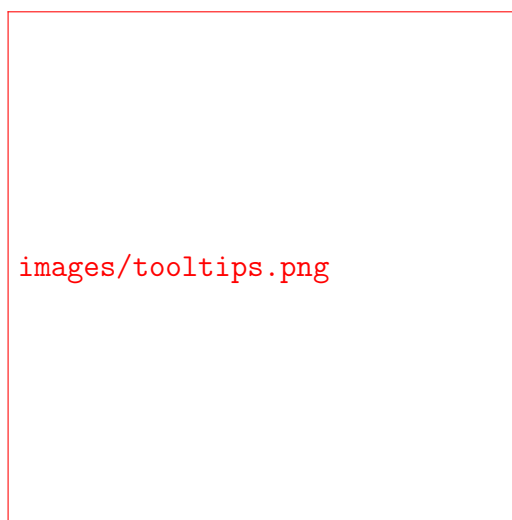


Figura 37: Suggerimenti

## 3.6 Pattern per l'interfaccia utente

### 3.6.1 Conferme per le azioni

Quando l'utente invoca un'azione irreversibile, è opportuno chiedere conferma che questa fosse sua effettiva intenzione, esplicitando le effettive conseguenze di questa, così da evitare che questa venga svolta nel caso di attivazione involontaria. Inoltre, può essere utile visualizzare è utile mostrare un messaggio anche al completamento dell'azione.

### 3.6.2 Formati di data e ora

Le date e le ore hanno formati diversi a seconda della zona del mondo e della lingua adottata. Per questo motivo, quando si visualizza una data, è necessario esplicitare anche il formato in cui viene visualizzata, il fuso orario di riferimento e tutte le informazioni utili all'utente per leggere correttamente la data.

### 3.6.3 Contenitori vuoti

Quando un oggetto non può mostrare del contenuto assume uno stato vuoto, ad esempio nel caso di una lista senza elementi. In questa situazione è opportuno visualizzare un messaggio o una semplice immagine che descriva la situazione e che permetta all'utente di intuire la funzionalità di tale oggetto, nonostante questo sia al momento privo di contenuto.

### 3.6.4 Messaggi di errore

Un messaggio di errore appare quando un'applicazione fallisce nello svolgere o nel completare un'azione. Gli errori devono essere ridotti al minimo, aumentando il più possibile la flessibilità della risposta agli input dell'utente e la robustezza del codice dell'applicazione.

I messaggi di errore devono essere facilmente distinguibili da quelli informativi, utilizzando colori riconoscibili (principalmente tonalità di rosso) e devono mostrare un testo che indichi la tipologia di errore che si è verificato e la motivazione che lo ha scatenato.

### 3.6.5 Impronta digitale

La scansione dell'impronta digitale è utilizzabile per facilitare gli accessi alle applicazioni, per agevolare i pagamenti e per sbloccare lo smartphone.

Al momento della scansione dell'impronta, deve essere mostrata una finestra di dialogo con la relativa icona.

È opportuno consentire ugualmente di accedere tramite metodi di backup come PIN o password, nel caso in cui non fosse possibile effettuare la scansione in un determinato momento.

### 3.6.6 Gesture

Le *gesture* sono delle combinazioni di tocco che attivano delle funzionalità specifiche. Le più comuni sono tocco semplice, doppio tocco, striscio e tocco prolungato. Esistono anche delle gesture più articolate, come ad esempio tocco a pinza ad aprire/chudere (zoom in/out), tocco con due dita, rotazione con due dita, ecc.

Lo striscio è differenziato in tre tipologie:

- drag, uno striscio lento e controllato;
- swipe, uno striscio veloce e senza un preciso obiettivo sullo schermo;
- fling, uno striscio impreciso.

### 3.6.7 Aiuto e feedback

La sezione di aiuto e feedback ha l'obiettivo di assistere l'utente nell'utilizzo della applicazione, spiegando in modo semplice le funzionalità di quest'ultima. Tale sezione deve essere facilmente individuabile dall'utente e accessibile da più punti all'interno dell'applicazione. Nel caso si tratti di un'applicazione articolata, è possibile inserire un pulsante di aiuto all'interno della barra di navigazione.


### 3.6.8 Schermata di avvio

La schermata di avvio è il primo contenuto che l'utente visualizza al lancio dell'applicazione. Essa serve a mostrare il logo e gli elementi principali del brand dell'applicazione, nell'attesa che il contenuto di questa venga caricato completamente.

In alternativa, nel caso di applicazioni che non necessitano di un tempo di caricamento elevato, è possibile mostrare un'interfaccia vuota con pochi e semplici oggetti.

### 3.6.9 Caricamento di immagini

Le immagini possono essere contenuti complessi e pesanti, per questo motivo il relativo caricamento avviene in tre fasi, con una variazione progressiva dei parametri di opacità, esposizione e infine saturazione, come illustrato nella Figura 38.



images/loadingimages1.png

Figura 38: Caricamento in tre fasi delle immagini

### 3.6.10 Navigazione

La navigazione guida l'utente attraverso i vari componenti dell'applicazione, per questo motivo è necessario che la sua realizzazione sia estremamente intuitiva. Il contenuto deve essere presentato in modo chiaro e semplice, guidando l'attenzione dell'utente sugli elementi più importanti dell'interfaccia. Le destinazioni più importanti dell'applicazione devono essere facilmente e velocemente raggiungibili e l'interfaccia deve tenere presente delle azioni più frequenti.

Le activity vanno differenziate e organizzate secondo una chiara gerarchia, che sarà la base della struttura della navigazione. Inoltre, è necessario fornire sempre all'utente la possibilità di tornare indietro alla schermata precedente o a quella principale.

Per fornire una navigazione intuitiva è possibile utilizzare un *navigation drawer* (cassetto di navigazione), un pannello che scorre da sinistra verso l'interno dell'interfaccia e contiene tutte le destinazioni principali dell'applicazione. Quando il drawer viene mostrato, la restante parte dell'interfaccia viene lievemente oscurata, in modo da concentrare l'attenzione sul pannello appena aperto.



images/navigation.png

Figura 39: Navigation drawer

### 3.6.11 Transizioni di navigazione

Le transizioni di navigazione permettono di evidenziare il passaggio da uno stato ad un altro all'interno dell'applicazione.

La maggior parte delle transizioni è gerarchica e tiene quindi conto delle relazioni padre-figlio tra le diverse viste dell'applicazione. Il passaggio tra oggetto padre e oggetto figlio viene

evidenziato tramite cambi di elevazione e/o con l'espansione del nuovo oggetto a partire dall'oggetto padre. Quando invece la navigazione avviene tra oggetti di pari importanza la transizione può essere direzionale, lungo un asse che varia in base alla posizione reciproca delle viste.

### 3.6.12 Notifiche

Le notifiche forniscono brevi informazioni riguardanti l'applicazione quando essa non è visualizzata sullo schermo. Le notifiche possono avere principalmente due funzionalità:

- comunicare all'utente il verificarsi di un determinato evento, come ad esempio la ricezione di un messaggio;
- visualizzare informazioni riguardanti processi in corso in background o che devono essere avviati (*reminder*);

La parte principale di una notifica è il contenuto, oltre a questo possono essere presenti anche icone e informazioni aggiuntive. Nelle notifiche espandibili è possibile permettere all'utente di interagire ed eseguire delle azioni.

Le notifiche vengono visualizzate nel *notification drawer* (cassetto di notifica) e la loro icona può essere presente nella barra di stato. Inoltre, in base alle preferenze dell'utente, queste possono comparire anche nella schermata di blocco del dispositivo.

Le notifiche non devono essere utilizzate come canale principale di comunicazione tra l'applicazione e l'utente, in modo da evitare che queste vengano visualizzate con troppa frequenza. Nel caso vengano inviate più notifiche da una stessa applicazione, queste vengono raggruppate, ma possono essere comunque espanse e visualizzate separatamente dall'utente.

### 3.6.13 Stati offline

Lo stato offline di un'applicazione permette all'utente di utilizzarne le funzionalità senza disporre di una connessione di rete. Se il contenuto visualizzabile offline è differente, l'applicazione deve visualizzare lo stato in cui si trova. È necessario evitare la totale assenza di contenuti offline. Inoltre, è bene fornire all'utente la possibilità di scaricare dei contenuti nel momento in cui dispone di una connessione di rete, così da poterne usufruire quando la connessione viene a mancare.

### 3.6.14 Permessi

Le richieste di permesso per l'utilizzo di determinate funzionalità del dispositivo devono essere messaggi brevi e facilmente comprensibili. È bene spiegare in modo chiaro all'utente il motivo per cui viene chiesto il permesso e dev'essere dato un feedback sulle conseguenze dell'accettazione o negazione di quest'ultimo.

### 3.6.15 Ricerca

La funzione di ricerca permette all'utente di individuare del contenuto specifico all'interno dell'applicazione. Questo strumento è particolarmente utile nel caso in cui l'applicazione contenga molte informazioni che l'utente può visualizzare e analizzare.

La ricerca può avvenire attraverso l'inserimento di testo oppure attraverso comandi vocali. Inoltre, l'utente può essere aiutato tramite sistemi di auto-completamento o di cronologia.

Se la ricerca è un'azione fondamentale dell'applicazione, la barra ad essa adibita deve essere sempre presente e ben visibile.

### 3.6.16 Impostazioni

La sezione delle impostazioni permette all'utente di modificare il modo in cui l'applicazione appare e si comporta. Tale sezione deve essere facilmente raggiungibile attraverso una specifica icona posta nella barra di navigazione dell'applicazione o nel navigation drawer, se presente.

Le impostazioni devono essere raggruppate in categorie e presentare la loro funzione in modo chiaro, nonché un'indicazione dello stato in cui si trova quello specifico parametro al momento della modifica.

L'obiettivo delle impostazioni è quello di permettere all'utente di configurare l'applicazione secondo le proprie preferenze. Al primo avvio dell'applicazione, i valori di default devono essere quelli più comuni e prevedibili.

### 3.6.17 Swipe di aggiornamento

Lo swipe di aggiornamento permette all'utente di aggiornare manualmente l'interfaccia, utilizzando una gestire che consiste nello scorrimento dall'alto verso il basso. L'applicazione dovrebbe essere aggiornata in automatico per visualizzare sempre lo stato attuale delle informazioni, ma è bene fornire la possibilità all'utente di effettuare tale aggiornamento in modo manuale.

Lo swipe di aggiornamento può essere utilizzato con liste, griglie e collezioni ma non va assolutamente utilizzato nei navigation drawer, nei contenuti scorribili (ad esempio le mappe) e nei widget della schermata Home.

## 3.7 Istruzione dell'utente

### 3.7.1 Fase iniziale

Nel momento del primo accesso nell'applicazione è necessario aumentare l'interesse dell'utente nei confronti del prodotto e motivarlo ad utilizzare frequentemente l'applicazione, facendo capire quali sono le funzionalità principali di quest'ultima e come ne può integrarne l'utilizzo nella propria quotidianità.

La fase iniziale può essere di tre tipi, in base alla tipologia dell'applicazione:

- si porta l'utente direttamente all'interno dell'applicazione, indirizzandolo chiaramente verso la prima azione per iniziare ad utilizzare quest'ultima;
- si invita l'utente a selezionare una breve serie di preferenze, così da permettergli di configurare l'applicazione a proprio piacimento. Le preferenze a disposizione dell'utente devono agire su aspetti basilari, ad esempio l'aspetto dell'applicazione, dato che l'utente non ha ancora acquisito familiarità con le funzionalità di quest'ultima;
- si mostrano in un carosello le principali funzionalità dell'applicazione, illustrando l'obiettivo di quest'ultima e le principali feature utilizzabili. Le grafiche visualizzate devono essere correlate tra loro e dare un senso di continuità.

### 3.7.2 Scoperta delle funzionalità

L'applicazione deve istruire l'utente sulle nuove feature in momenti strategici che lo involino ad approfondirne la conoscenza e non lo infastiscano durante l'utilizzo. In base alla funzionalità, questa scoperta può avvenire in un semplice passo oppure in una serie guidata di step successivi.

L'indicazione sulle azioni da svolgere può essere correlata da animazioni e deve avere un colore ben visibile, solitamente il colore primario dell'applicazione.

### 3.7.3 Insegnamento delle gesture

L'insegnamento delle gesture dell'applicazione è essenziale per far capire all'utente come relazionarsi a specifiche funzionalità e come accedere più agevolmente a determinati contenuti.

Questo insegnamento avviene attraverso delle animazioni che mimano le gesture da compiere, corredate da una breve descrizione dell'azione che viene compiuta. Questi suggerimenti devono comparire solo quando l'utente ne ha bisogno, non necessariamente al primo avvio dell'applicazione. Inoltre, la quantità di suggerimenti di questo tipo deve essere limitata agli elementi essenziali, per evitare di appesantire eccessivamente l'esperienza utente e ogni consiglio deve essere visualizzato solo una volta.



## 3.8 Usabilità

### 3.8.1 Accessibilità

Un prodotto ben progettato deve essere utilizzabile e accessibile da tutte le tipologie di utenti, indipendentemente dal grado di esperienza e dalle abilità tecniche e fisiche. Material Design presenta delle linee guida che permettono agli sviluppatori di effettuare i giusti accorgimenti per rendere le proprie applicazioni accessibili a chiunque.

L'interfaccia deve presentare un layout chiaro, con delle azioni ben visibili e distinguibili. I bottoni, le immagini e le linee di testo devono essere utilizzate con parsimonia, in quanto appesantiscono l'interfaccia e l'esperienza utente. L'interfaccia può essere resa chiara e comprensibile attraverso l'uso di:

- elementi ben visibili e distinguibili, attraverso il contrasto dei colori e la dimensione degli oggetti;
- gerarchie ben definite tra gli elementi dell'interfaccia, con le informazioni cruciali facilmente individuabili.

I colori dell'interfaccia devono essere chiari, funzionali e non devono generare confusione. Il contrasto del testo deve essere sufficientemente elevato da permetterne una facile lettura. È inoltre raccomandato l'utilizzo di messaggi testuali che diano consigli all'utente quando è necessario fare una scelta o quando si verifica un errore. I font e i campi testuali devono essere definiti in modo da accomodare con facilità il cambio della dimensione del testo.

Anche l'utilizzo di suoni comprensibili e distinguibili è molto utile per rendere l'esperienza utente universale.

L'obiettivo dello sviluppatore deve essere quello di rendere immediatamente comprensibile la navigazione all'interno dell'applicazione, la comprensione delle informazioni presentate e l'accesso ai contenuti anche agli utenti che usufruiscono di un'esperienza utente solamente testuale o vocale. È molto importante supportare le funzionalità di assistenza specifiche della piattaforma su cui si sta lavorando, queste aiutano ad aumentare la scorrevolezza e la piacevolezza dell'utilizzo dell'applicazione da parte di utenti con disabilità sfruttando dispositivi fisici, come gli apparecchi acustici, oppure software, come la lettura dello schermo.

### 3.8.2 Bidirezionalità

Nel caso di interfacce pensate per linguaggi la cui lettura avviene da destra a sinistra, come arabo ed ebraico, i contenuti devono essere specchiati, per essere più facilmente comprensibili. Di conseguenza, il modo in cui il contenuto viene mostrato e le gerarchie e le sequenze vengono visualizzate deve essere differente. Il processo di cambiamento a cui è soggetta un'interfaccia che implementa entrambe le modalità di lettura e visualizzazione, è chiamato *mirroring*.



Figura 40: Mirroring

## 3.9 Piattaforme

Material Design supporta pratiche per progettazione e usabilità su dispositivi di tipo diverso, in modo da permettere la creazione di esperienze utente coerenti e piacevoli. Le interfacce necessitano di un adattamento nel caso in cui si passi a piattaforme che implementano differenti funzionalità di design oppure nel caso di dimensioni dello schermo notevolmente differenti.

Nell'implementazioni di un'applicazione su una determinata piattaforma è bene utilizzare l'iconografia standard del dispositivo, per evitare di creare confusione nell'utente. Ad esempio per le toolbar si consiglia di utilizzare l'allineamento testuale standard della piattaforma su cui viene utilizzata l'applicazione, come illustrato nella Figura 41.



Figura 41: Differenze tra piattaforme diverse

## 4 Material Design 2

L'introduzione della seconda versione di Material Design [7] ha apportato le seguenti aggiunte e modifiche:

- nuovo set di icone colorate;
- carattere Google Sans;
- maggiore utilizzo degli spazi vuoti, che diventano parte integrante nel design di un'interfaccia semplice ed intuitiva;
- utilizzo di **forme con spigoli smussati**, tra cui gli indicatori pop-up e di selezione dei menu;
- ampliamento delle funzionalità legate ai temi, compresa la definizione delle **linee guida per il design del tema scuro**;
- definizione delle **linee guida per l'utilizzo del suono** all'interno dell'applicazione;
- **nuovi componenti**: navigation rail, bottom app bar, backdrop, banner, liste di immagini;

Nel seguito analizziamo nel dettaglio le novità principali.

## 4.1 Forme

La forma di default dei componenti è rettangolare con angoli smussati di 4dp. È possibile modificarne l'aspetto modificando le linee degli spigoli e la loro inclinazione.

L'utilizzo di forme particolari aiuta a **focalizzare l'attenzione dell'utente** sugli elementi importanti dell'interfaccia, ad esempio attraverso l'uso di una app bar rettangolare con una rientranza volta a contenere un FAB rotondo (Figura 42 in basso a sinistra).



Figura 42: Forme standard e forme modificate

Inoltre, cambiare forma ad un oggetto può essere utile ad indicarne il **cambiamento di stato**.

Le forme personalizzate vanno utilizzate in modo consistente, insieme all'uso dei colori, per rafforzare il branding dell'applicazione. Esse devono però risultare **familiari ed intuitive** per l'utente, è quindi sconsigliata la creazione di forme troppo particolari che creino confusione nell'utente ed interferiscano con gli altri elementi dell'interfaccia, rendendone magari illeggibile il contenuto.

Material Design fornisce degli appositi strumenti per la creazione e personalizzazione di forme.

## 4.2 Temi

Nonostante l'utilizzo dei temi fosse già presente nella prima versione di Material Design, nella seconda versione sono state ampliate le relative funzionalità, permettendo di **rendere consistenti i colori** all'interno dell'applicazione ma anche di personalizzare le forme e i testi.

La creazione di un tema avviene tramite la definizione dei seguenti elementi:

1. colore **primario** e colore **secondario**;
2. colori delle superfici, dello sfondo e delle schermate di errore;
3. colori per gli elementi che si sovrappongono ad altri elementi, come ad esempio i testi che appaiono su elementi del colore principale;
4. tipografia, attraverso la definizione di una *type scale* tramite l'apposito strumento fornito da Google;
5. forme degli oggetti;
6. icone, che possono essere standard o personalizzate.

### 4.2.1 Tema scuro

Il tema scuro è una tipologia di interfaccia che presenta principalmente superfici e oggetti di colore scuro. Esso è pensato per essere un supplemento al tema chiaro di default. L'utilizzo del tema scuro diminuisce la luminanza emessa dallo schermo del dispositivo, riducendo la stanchezza visiva dell'utente e rendendo più piacevole l'utilizzo del dispositivo in contesti bui. Inoltre, su determinati dispositivi, permette di risparmiare batteria.

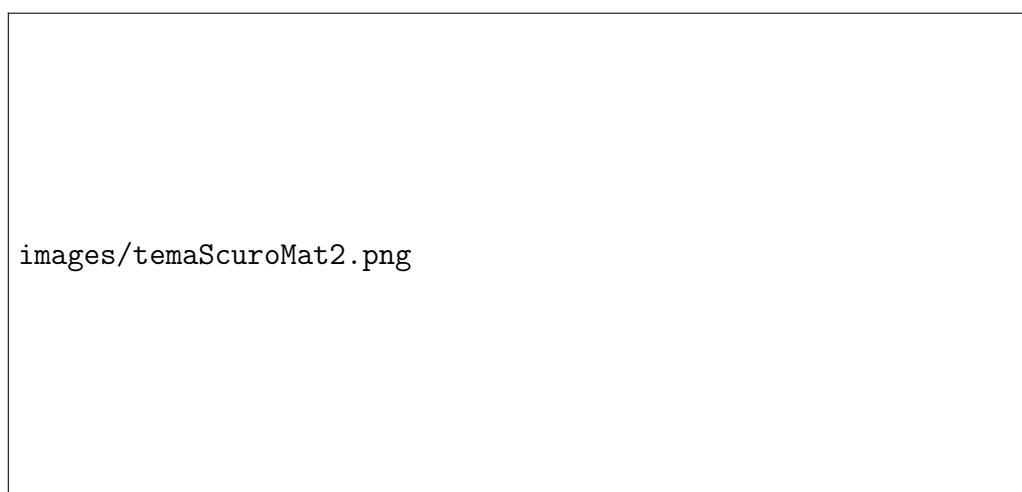


Figura 43: Tema scuro

I principi alla base del design di un tema scuro sono i seguenti:

- utilizzare le **tonalità di grigio per esprimere l'elevazione e lo spazio** degli elementi nell'interfaccia, come mostrato in Figura 44;
- sfruttare con parsimonia gli elementi con colori vivaci, per lasciare la maggior parte dell'interfaccia di colore scuro;
- aumentare l'accessibilità attraverso un **utilizzo corretto del contrasto**.

Come per il tema chiaro, Material Design fornisce i parametri standard per il contrasto, la saturazione, la profondità e lo spettro dei colori.

Il tema scuro può essere attivato o disattivato attraverso un controllo posto nella barra dell'applicazione, nel caso l'utilizzo sia frequente, oppure in un menu separato altrimenti.

Il colore principale del tema scuro non è il nero, bensì il grigio scuro. Questo permette di visualizzare un maggior range di colori e ombre e di ridurre l'affaticamento della vista, il testo bianco su sfondo nero ha un contrasto più elevato che tenderebbe ad affaticare maggiormente il lettore. È necessario che le superfici del tema siano però abbastanza scure da far sì che il testo bianco sia leggibile su di esse con un contrasto adeguato. Per differenziare l'importanza del testo, si utilizzano diversi gradi di opacità, ma il colore rimane sempre il bianco.

I colori utilizzati all'interno dell'interfaccia devono essere **desaturati**, per superare gli standard WCAG<sup>1</sup> di accessibilità. Inoltre, i colori saturi creano delle vibrazioni ottiche su superfici scure, che possono causare affaticamento visivo all'utente.



Figura 44: Elevazione nel tema scuro

<sup>1</sup><https://www.w3.org/WAI/standards-guidelines/wcag/>

### 4.3 Suono

Il suono ha la capacità di migliorare l'esperienza utente, fornendo del **feedback utile all'utente**, che può anche esprimere la personalità dell'applicazione.

Il suono deve essere:

- informativo, e quindi intuitivo, funzionale e facilmente riconoscibile;
- contraddistinguibile, e quindi esprimere il carattere del brand e l'estetica del prodotto;
- piacevole, dando un senso di comfort all'utente.

All'interno dell'interfaccia utente possono essere presenti tre tipi di suono:

- di design, permette di fornire informazioni, esprimere emozioni e educare l'utente riguardo ad un'interazione con l'interfaccia. Questo permette ad esempio di comunicare all'utente se un'azione si è conclusa correttamente o meno. Nel caso il suono sia particolarmente fedele alla realtà, ed è detto **skeumorfico**, altrimenti astratto;
- musica, utilizzata principalmente per raccontare storie e per esprimere degli stati d'animo. Quando è correlata con immagini, essa **rende molto più immersiva l'esperienza** dell'utente con l'applicazione;
- voce e lettura del testo, esprimono informazioni nelle situazioni in cui l'informazione da trasmettere è complessa e i suoni di design e la musica non riescono a farlo. Ad esempio quando si tratta di dialoghi, conversazioni o si vuole esprimere un tono e una personalità ben precise.

Ognuno di questi fornisce informazioni di tipo diverso e il loro insieme, se utilizzati correttamente, può creare un **effetto unico e riconoscibile**.



## 4.4 Componenti

### 4.4.1 Navigation rail

Il **navigation rail** fornisce un'alternativa all'utilizzo di una barra di navigazione inferiore, preferibile nel caso di **dispositivi di grandi dimensioni**. Si tratta di un componente di navigazione laterale in grado di mostrare da tre a sette destinazioni all'interno, ognuna rappresentata da un'icona e una label, e opzionalmente un floating action button. Ad ogni icona può a sua volta essere associato un badge che fornisce informazioni su quella destinazione, come ad esempio il numero di nuove notifiche.

Il navigation rail deve quindi essere compatto, facilmente raggiungibile e consistente, aparendo sempre nella stessa parte dello schermo. Esso va posizionato in verticale nel lato sinistro, in caso di linguaggi la cui lettura va da sinistra a destra, e nel lato destro altrimenti.

Durante lo scrolling verticale del contenuto il rail rimane fisso, mentre con quello orizzontale è possibile decidere se lasciarlo all'interno dell'interfaccia o se farlo uscire temporaneamente.

### 4.4.2 Bottom app bar

La **bottom app bar** è una barra disposta nella parte inferiore dell'interfaccia, utile a contenere un gruppo di azioni chiave e opzionalmente anche un floating action button (vedi Figura 46). La bottom app bar va utilizzata solo sui dispositivi mobili, quando è necessario visualizzare più di un'azione chiave e non è presente una barra di navigazione inferiore.

Questa barra deve essere sempre **facilmente raggiungibile**, ma la struttura può cambiare in base alle necessità e alla schermata in cui ci si trova, per evitare di interferire con altri elementi.

Quando è presente, è possibile elevare il FAB oppure inserirlo alla stessa elevazione della barra, modificandone la forma rettangolare standard.



Figura 45: Navigation Rail

Quando si effettua lo scrolling verso il basso, la barra scompare, mentre scrollando verso l'alto, ricompare. Alcuni elementi dell'interfaccia, come la tastiera, si sovrappongono alla barra e la nascondono.

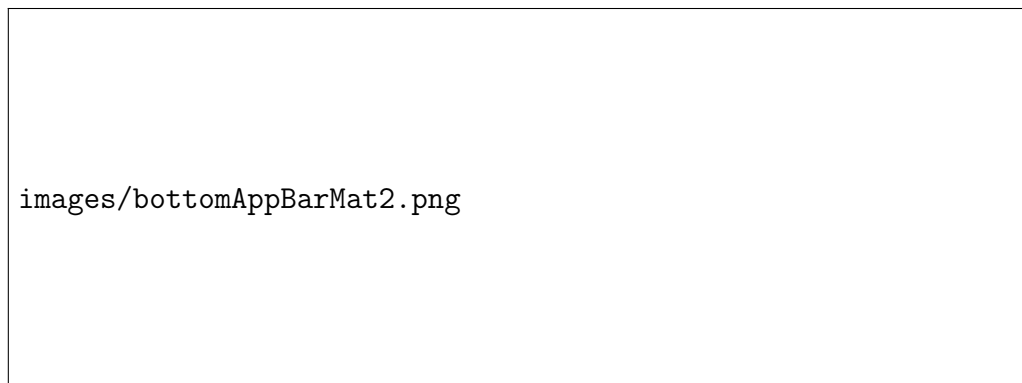


Figura 46: Bottom app bar

#### 4.4.3 Backdrop

Il **backdrop** è un foglio di materiale che appare dietro a tutte gli altri elementi dell'interfaccia, presentando del contenuto informativo anche azionabile. Esso è composto da due elementi (vedi Figura 47):

- **strato retrostante**, mostra il contesto attuale e le azioni che controllano il contenuto dello strato frontale. All'occorrenza questo può essere espanso per mostrare ulteriori opzioni;
- **strato frontale**, si sovrappone parzialmente a quello retrostante e mostra i contenuti primari dell'interfaccia. Quando questo è attivo, lo strato retrostante presenta un titolo e un'azione che permette di attivare il backdrop.

Il backdrop, se implementato, deve essere sempre presente, concentrare il suo focus su una pagina alla volta ed essere facilmente accessibile.

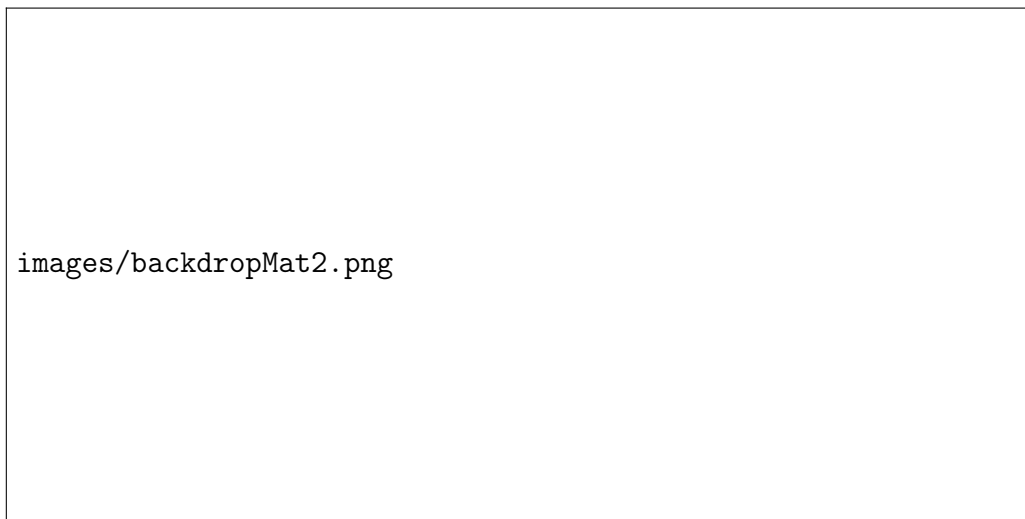


Figura 47: Backdrop (in blu lo strato retrostante, in bianco lo strato frontale)

#### 4.4.4 Banner

Un **banner** mostra un messaggio importante, opzionalmente accompagnato da azioni ad esso legate. Questo è persistente e necessita quindi di un'azione dell'utente per essere chiuso e cancellato. Il banner deve essere visualizzato nella parte superiore dello schermo ed è bene visualizzarne solamente uno alla volta.

Il contenitore del banner è rettangolare e può contenere: un messaggio breve e chiaro, un'icona opzionale e fino a due pulsanti testuali, che presentano a sinistra l'azione di cancellazione del banner e a destra quella di esecuzione dell'azione.

Il banner interrompe l'attività attuale dell'utente, la sua visualizzazione deve essere opportuna e coerente con l'importanza del messaggio che deve essere mostrato.

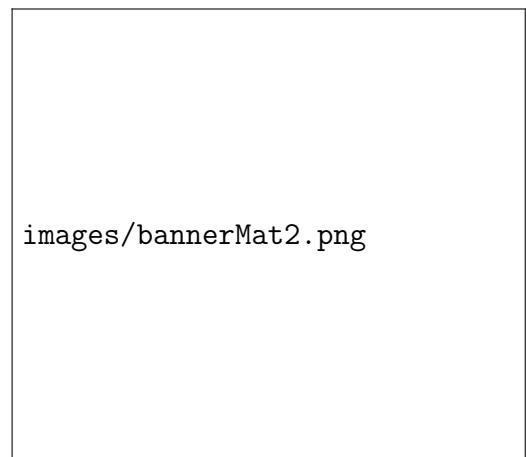


Figura 48: Banner

#### 4.4.5 Liste di immagini

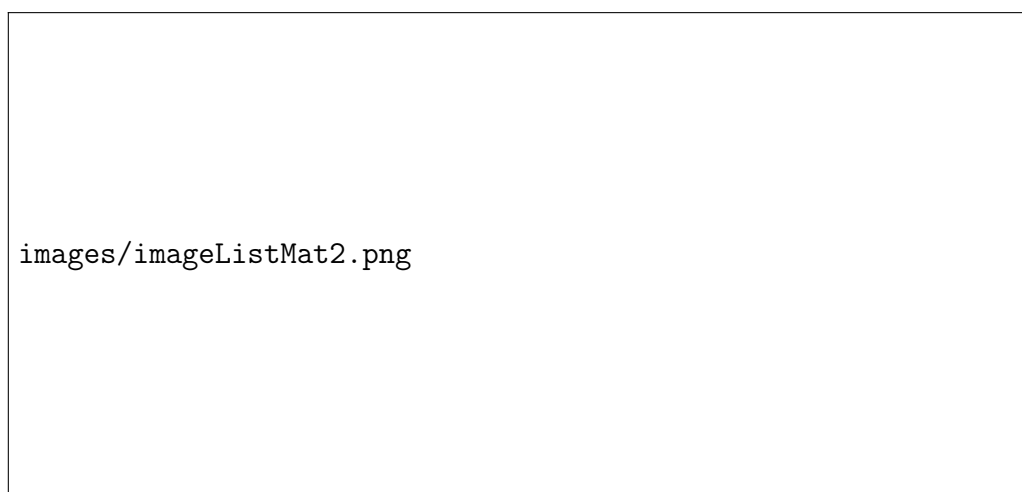


Figura 49: Lista di immagini

Le **liste di immagini** sono utili a disporre collezioni di immagini in una griglia ordinata secondo determinati pattern ripetuti, in modo da aumentare la comprensione visiva del contenuto e facilitare il confronto tra elementi vicini.

La lista può essere di 4 tipologie:

- **standard**, tutti gli elementi di uguale dimensione e importanza;
- **quilted** (trapuntata), enfatizza gli elementi più importanti, aumentando lo spazio da essi occupato;
- **woven** (intrecciata), dove le immagini possono assumere diverse forme e rapporti, con dei pattern di ordinamento ripetuti;
- **masonry** (a muratura), dove le immagini hanno rapporti e dimensioni diversi tra loro.

Ogni oggetto della lista, oltre all'immagine, può contenere opzionalmente del testo e delle icone. Inoltre, lo spazio tra le varie immagini della lista è variabile in base alle necessità organizzative e stilistiche.

È anche possibile implementare delle funzionalità di tocco, scorrimento, filtraggio e ordinamento per gli elementi delle liste.

## 5 Material Design 3

**Material Design 3** [8] è la terza e più sviluppata versione del sistema di design di Google.

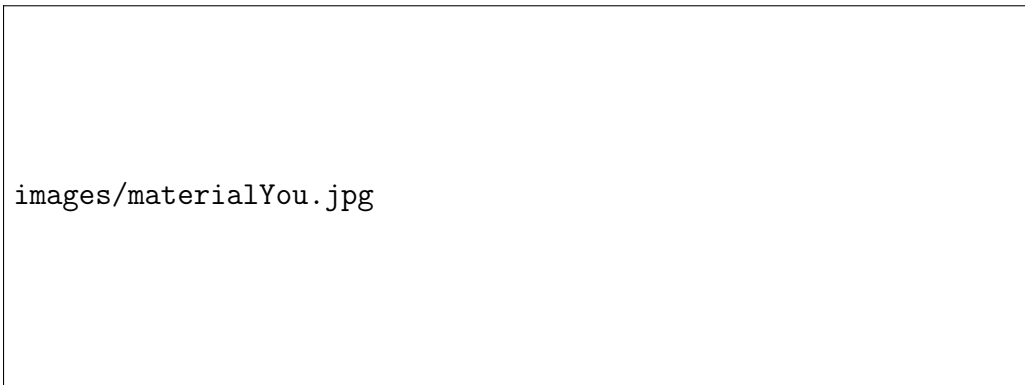
In questa versione è stato introdotto il nuovo **Material You** ossia un modo completamente nuovo di pensare al design dell'applicazione che si sta sviluppando.

Con l'introduzione di Material Design 3 e di Material You sono state sviluppate le seguenti novità:

- i temi di Material You hanno **palette cromatiche espanse** rispetto ai temi delle versioni precedenti, comprendendo un colore terziario e tutti i derivanti dalle combinazioni di questo con il colore primario, il secondario e il colore di errore.
- sono stati introdotti i **colori dinamici**, con la classe relativa
- è stato sviluppato e introdotto lo strumento **Material Theme Builder**, che permette di migrare facilmente dai temi delle versioni passate ai nuovi temi di material You
- la **tipografia** è stata **semplificata**
- sono stati introdotti i **Material Symbols**
- sono stati **aggiornati alcuni componenti**
- sono state aggiunte **linee guida per l'adattamento a diverse tipologie di dispositivi**

Nel seguito analizziamo nel dettaglio le novità principali.

## 5.1 Material You



images/materialYou.jpg

L'obiettivo di Material You è quello di esprimere al meglio **emozioni e movimento**. Esso, quindi, esplora un atteggiamento molto più umano nei confronti del design, analizzando e celebrando il rapporto tra la sensibilità del design e le preferenze personali dell'utente, senza tralasciare le emozioni.

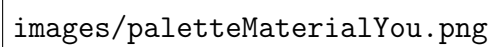
Material You punta a permettere di dare vita a design specifici e adatti ad ogni stile, accessibili per ogni necessità e che si adattino a qualsiasi schermo o dispositivo e a qualsiasi situazione. Per fare questo, il nuovo metodo di design si basa sullo **sfondo del dispositivo** selezionato dall'utente per costruire e pensare una palette cromatica relazionata ad esso e basata sui suoi colori principali.

Material You, oltre a sfruttare le palette cromatiche per rendere personale l'esperienza utente, utilizza il **movimento degli elementi** per esprimere il brand e per aiutare l'utente a comprendere la spazialità del sistema. L'interfaccia, di conseguenza, reagisce ai cambiamenti di dispositivo o di contesto, in un modo coerente ma anche quasi umano, integrando piccole sorprese che rendono l'esperienza utente più spontanea e coinvolgente.

L'**accessibilità** viene migliorata, consentendo all'utente di modificare contrasti, dimensioni e spessori ma allo stesso tempo affiancandogli un sistema che controlla e adatta al meglio i valori selezionati con la tipologia di interfaccia in esame.

Material You è disponibile dalla versione 12 di Android.

Le palette di Material You sono, come accennato in precedenza, state ampliate del colore terziario e dei colori derivanti. Queste palette vengono generate, sempre in modo opzionale e consentendo all'utente di scegliere, partendo dal colore primario, secondario e terziario dello sfondo selezionato dall'utente.

The image area is mostly blank, with the text 'images/paletteMaterialYou.png' located on the left side. This text likely refers to a file path for a color palette image associated with the Material You design system.

images/paletteMaterialYou.png

Figura 50: Esempio di palette di Material You

## 5.2 Colori dinamici



Figura 51: Colori dinamici

Con l'utilizzo dei colori dinamici, l'applicazione mantiene i **colori distintivi** del brand e lo stile che la caratterizza, considerando allo stesso tempo i settaggi e le **preferenze** dell'utente, fornendo a quest'ultimo la possibilità di adattare a proprio piacimento il prodotto.

Material Design 3 supporta applicazioni sistematiche dei parametri personalizzabili per aiutare a definire uno stile che esprima il brand e le caratteristiche principali del prodotto.

Il sistema colore agisce automaticamente nel caso di situazioni in cui, in seguito ad accostamenti particolari di colori, i parametri di accessibilità vengono a mancare.

I colori dinamici, inoltre, funzionano e sono applicabili anche nel caso di componenti che non fanno parte di Material Design.

Per creare il profilo colore, le applicazioni possono utilizzare sia colori personalizzati, che colori generati dinamicamente.

Per sfruttare questa feature, è necessario definire un **tema di default** su cui le preferenze dell'utente possano essere mappate e che venga utilizzato nel caso esso scelga di non utilizzare i colori dinamici.

È possibile **integrare** i colori dinamici all'interno di un tema esistente e già in uso, selezionando i componenti che li sfruttano, oppure **partire da zero** con la definizione di un tema che ne implementi già le funzioni.

Attraverso l'uso del Material Theme Builder, di cui si parla nella sezione successiva, è possibile impostare i **colori del brand**, inserendo un solo colore oppure anche un'intera palette cromatica. Nel caso di un colore singolo, è consigliato impostarlo come il colore principale del



tema, in modo che il sistema crei una combinazione di colori adeguata, fondata su di esso e sulle scelte dell'utente.

I colori dinamici possono essere applicati anche alle **illustrazioni**, o a parte di esse. Per fare questo, il formato SVG è il più consigliato e quello che meglio si adatta alle variazioni di colore.

Alcuni elementi dell'illustrazione, tuttavia, potrebbero dover rimanere invariati, per creare continuità tra i vari temi scelti dall'utente.

### 5.2.1 Schemi generati dall'utente

Gli schemi generati dall'utente derivano dalla scelta che esso fa dello **sfondo del dispositivo**. L'interfaccia di sistema ne utilizzerà, quindi, il colore principale in alcune tipologie di componenti e icone.

Da singolo colore da cui partire, derivano 5 palette tonali, che sono sufficienti a dare forma ai temi chiari e scuri necessari. Anche i colori di errore vengono settati automaticamente, ma non fanno parte dell'algoritmo con cui il sistema estrae le palette dallo sfondo.

### 5.2.2 Tema di base e token

Il tema di base è una serie di colori utili a creare i temi chiaro e scuro dell'applicazione. Esso è progettato **impostando** i diversi **ruoli** tra i colori e le **relazioni** tonali in modo da accettare un'eventuale palette personalizzata dall'utente che lo sostituisca.

Il colore di default del tema di Material 3 presenta un violetto (6750A4) come colore principale e le differenze di tono che ne derivano.

### 5.2.3 Colori personalizzati

L'utilizzo di colori personalizzati permette l'integrazione con i colori dinamici, mantenendo **standard** i colori di alcuni **componenti principali** e caratteristici del brand dell'applicazione.

Quando vengono inseriti i colori personalizzati, un algoritmo ne deriva automaticamente le palette relative, per andare a definire tutto lo spettro dei colori dell'applicazione.

Una volta fatto questo, l'utente può scegliere quali colori applicare all'interfaccia.

L'algoritmo, inoltre, aggiusta leggermente la saturazione e la tonalità dei colori personali, per renderli più bilanciati con quelli di sistema.

### 5.3 Material Theme Builder

Lo strumento **Material Theme Builder** offerto da Google permette di costruire un **tema personalizzato**, con colori dinamici, e di esportarlo facilmente in codice. Esso è disponibile sul web o come plugin di Figma.

Con Material Design 3 vengono introdotti i **design token**, ossia piccole decisioni di design riutilizzabili, che aiutano a mantenere una **coerenza** di base all'interno dell'applicazione. Material Theme Builder costruisce il design utilizzando questi token.

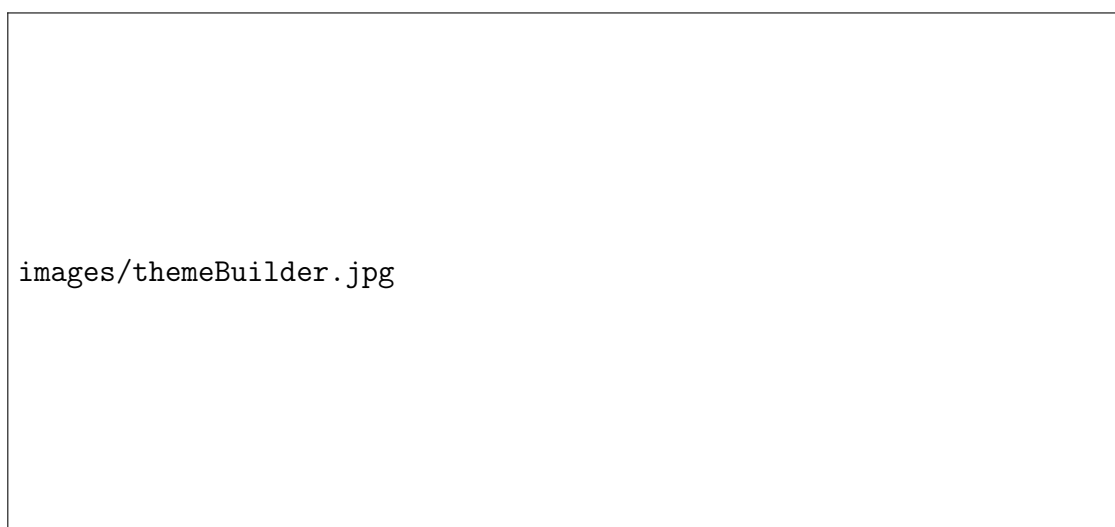


Figura 52: Esempio di tema e palette cromatica relativa

I temi definiti con questo strumento utilizzano i colori dinamici, in modo da accettare gli input del sistema. Utilizzando Figma, è possibile visualizzare come il tema si comporta con le impostazioni del nostro sistema. Il **tema scuro**, inoltre, viene automaticamente creato dallo strumento ed è, ovviamente, modificabile se ce n'è il bisogno.

Material Theme Builder, infine, permette di esportare il codice necessario ad applicare il tema alla nostra applicazione. Tale codice può essere scaricato in diversi formati: formato XML (per le viste Android), formato Kotlin (per l'utilizzo con Jetpack Compose) e in formato DSP (per il Design System Package e i token).

## 5.4 Tipografia

Con l'introduzione di Material Design 3, lo **stile tipografico** delle versioni precedenti è stato **semplificato**. Precedentemente, infatti, ogni stile conteneva 6 tipi di titolo, 2 tipi di sottotitolo, 2 tipi di carattere, e le variazioni relative ai bottoni, alle caption e alle overline. Ora, invece, si trovano solo **3 varianti per ogni tipologia di testo**: piccola, media e grande. Ogni testo appartiene ad uno dei seguenti **cinque tipi**: display, headline, title, body e label.

**Display** Il tipo di testo display è il **più grande** tipo di testo dell'interfaccia. Esso è quindi utilizzato per mostrare **testi brevi e importanti**, ed è consigliato su **dispositivi di grandi dimensioni**. Per questo tipo di testo, è consigliabile selezionare un **font più espressivo** e, se possibile, settare la dimensione ottica in modo appropriato. Nella type scale di default ci sono tre stili display: grande, medio e piccolo.

**Headline** Il tipo di testo headline è pensato per **testi brevi e di grande importanza**, da mostrare su **dispositivi di piccole dimensioni**. Questo è indicato per evidenziare i contenuti di maggiore importanza e per dare importanza ai **passaggi principali** del testo.

**Title** Il testo di tipo title è più piccolo di quello di tipo headline, e dovrebbe essere utilizzato per i **testi di media importanza e relativamente corti**. Esso è utilizzato per dividere i **passaggi secondari** del testo. Quando si usa questo tipo, bisogna fare attenzione ad utilizzare font particolarmente espressivi, come quelli utilizzati per i testi di tipo display.

**Body** Il testo di tipo body è raccomandato per essere utilizzato nei **passaggi testuali più lunghi** dell'interfaccia. In questo caso, sono consigliati **font leggibili** di dimensioni pensate per la lettura. Sono da evitare i font troppo espressivi e sono da preferire quelli **semplici e lineari**.

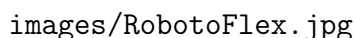
**Label** I testi label sono piccoli e utilizzati per **citazioni, note, didascalie o testi dentro ad oggetti**. I bottoni, per esempio, utilizzano i testi di tipo label grande.

### 5.4.1 Font variabili

La tipografia dell'ultima versione fa uso anche dei **font variabili**, per offrire un **controllo maggiore** sul testo e sull'accessibilità. Inoltre, vengono sfruttati i **design token**, per aiutare a trasportare le decisioni stilistiche attraverso dispositivi di tipo diverso. Essi, infatti, impostano lo stile del testo attraverso una variabile configurabile, non attraverso un valore definito e non modificabile.

Il font standard di Android è **Roboto**, e le sue versioni variabili sono chiamate **Roboto Flex** e **Roboto Serif**. Esse possono assumere un ampio spettro di dimensioni e spessore.

Roboto Flex è, come dice il nome, indicato per una **maggiore flessibilità** rispetto al predecessore Roboto.



images/RobotoFlex.jpg

Figura 53: Roboto Flex

Roboto Serif, invece, è particolarmente indicato per un'**esperienza di lettura rilassante**. Si tratta di un carattere **minimal** e **funzionale** e, di conseguenza, utilizzabile in qualsiasi situazione, anche grazie al suo essere variabile in molti aspetti.



images/RobotoSerif.jpg

Figura 54: Roboto Serif

L'ultimo carattere è **Noto**, che come nelle versioni precedenti, è utilizzato per i **linguaggi non supportati** da Roboto.



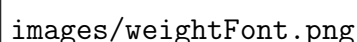
images/NotoMat3.jpg

Figura 55: Noto

È possibile **modificare e adattare** alle proprie esigenze **la type scale** in uso, inserendo o rimuovendo determinati stili, oppure cambiando Roboto con un font a nostra scelta.

Ogni carattere variabile è modificabile attraverso **4 parametri**: peso, grado, larghezza e dimensione ottica.

**Peso** Il peso definisce lo **spessore generale** di un font. I pesi più comuni sono il regolare e il grassetto, ma con i caratteri variabili è possibile impostare molti valori di peso differenti.



images/weightFont.png

Figura 56: Peso

**Grado** Il grado è un altro parametro che modifica lo **spessore di un carattere**, ma lo fa in modo più **granulare** e non cambiando la larghezza delle lettere oppure la divisione tra le linee.

Il grado è anche utile per **contrastare il cambiamento di peso** dello stesso carattere passando tra un tema chiaro e uno scuro.

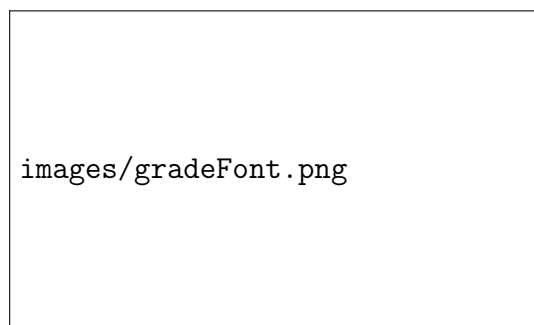


Figura 57: Grado

**Larghezza** La larghezza indica quanto **spazio orizzontale** è occupato dai caratteri. Una larghezza minore permette di stipare più caratteri all'interno di una riga mentre una larghezza maggiore può esprimere una personalità più spiccata.

**Dimensione ottica** La dimensione ottica è utilizzata per **adattare la typeface** a diverse dimensioni. Design con **dimensione ottica ridotta** si concentrano sulla **leggibilità** del testo, mentre testi con **dimensione ottica maggiore** mostrano maggiormente le **particolarità** dei caratteri e offrono più possibilità di modifica degli altri parametri.



Figura 58: Larghezza (a sinistra) e dimensione ottica (a destra)

### 5.4.2 Token

I **token** sono formati indipendentemente dalla piattaforma su cui si sviluppa, ma possono anche essere usati nel codice specifico per una piattaforma.

Esistono **3 tipi** di token: **reference**, **system** e **component**. I token component sono ancora in fase di sviluppo, mentre le altre due tipologie sono già utilizzate da Material Design 3. I

token sono utili a **mantenere** l'**allineamento** e la **coerenza** dell'interfaccia, anche attraverso dispositivi molto diversi.

I token rimpiazzano i valori statici, usando dei nomi specifici e definiti dal sistema di design. Essi formano una specie di repository contenente tutte le scelte stilistiche e i cambiamenti relativi.

## 5.5 Material Symbols

Con l'introduzione delle **Material Icons**, le icone erano disegnate e pensate per essere **minimale** e **universalmente riconoscibili**, anche attraverso piattaforme diverse e culture differenti. Questo le ha rese estremamente popolari e versatili. Ora, con il lancio e lo sviluppo di Material Symbols, esse sono diventate modificabili.

Il font delle icone di Material Symbols è la **versione variabile** del font delle icone di Material Icons. Analogamente ai caratteri, questo permette all'utente di **modificare** a proprio piacimento le proprie **icone**, impostando i valori di **4 parametri**: peso, riempimento, grado e dimensione ottica.

**Peso** Il peso, come per i caratteri, permette di impostare quanto è pesante o in **grassetto** la nostra icona, senza modificare la sua dimensione o il suo grado.

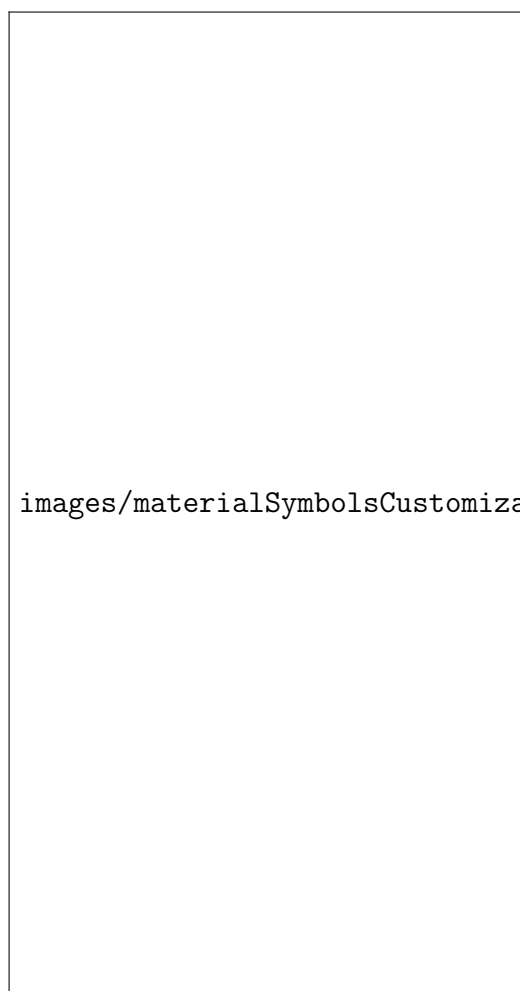
**Riempimento** Il riempimento permette di **riempire** le **zone vuote** e trasparenti dell'icona, per renderle opache. Questo può essere utile nel caso di animazioni o transizioni, o per personalizzare lo stile della propria applicazione.

**Grado** Come per i caratteri, il grado modifica lo **spessore** dell'icona, ma lo fa in modo più granulare rispetto al peso e avendo un minore impatto sulle sue dimensioni. È consigliato impostare in modo analogo il grado delle icone e quello del testo, per rendere più armoniosa l'interfaccia.

**Dimensione ottica** La dimensione ottica permette di **aggiustare** il peso dell'icona quando se ne modificano le dimensioni, in modo che essa risulti sempre proporzionata e naturale alla vista.

Le icone di Material Symbols sono disponibili in **3 stili**: outlined, rounded o sharp.

**Outlined** Le icone outlined sono **chiare** e **pulite** e sono ottime per lavorare in interfacce dense di contenuto e oggetti.



images/materialSymbolsCustomization.jpg

Figura 59: Parametri di modifica delle icone

**Rounded** Le icone rounded sono consigliate per essere accoppiate a stili che utilizzano tipografie pesanti, loghi rotondeggianti ed elementi circolari.

**Sharp** Le icone sharp presentano bordi e angoli ben definiti e dritti, per stili **definiti** e rettangolari, che rimangono leggibili e comprensibili anche a dimensioni più piccole.

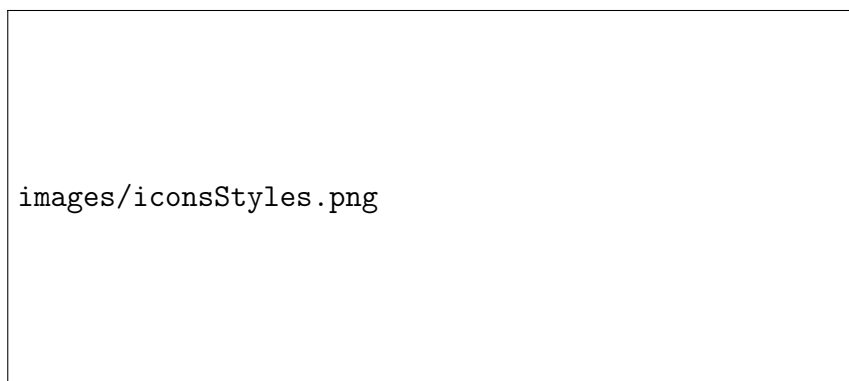


Figura 60: 1.Outlined 2.Rounded 3.Sharp



## 5.6 Componenti

### 5.6.1 Top app bar

La **top app bar** mostra informazioni e azioni nella parte alta dello schermo. Il contenuto mostrato è relativo alla pagina dell'applicazione in cui ci si trova, ma possono anche essere mostrate delle azioni universali.

Ci sono **4 tipi** di top app bar: allineata centralmente, piccola, media e grande.

Per differenziare la barra dal contenuto non si usa una differenza di elevazione, come nelle versioni precedenti di Material Design, ma si utilizza il **colore**. La larghezza della barra, inoltre, è la stessa dello schermo del dispositivo.



Figura 61: Differenze tra Material 2 e Material 3

**Top app bar allineata centralmente** La top app bar allineata centralmente è composta da:

1. un **contenitore rettangolare**, con elevazione 0 quando è statico, mentre elevazione 2 durante lo scrolling verticale
2. un **titolo centrale** di tipo title **large**
3. un'**icona di navigazione principale** (a sinistra)
4. un'altra **icona opzionale** (a destra)

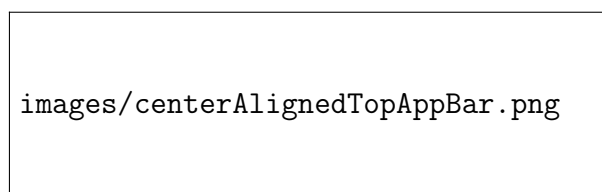


Figura 62: Top app bar allineata centralmente

**Top app bar piccola** La top app bar piccola presenta i seguenti elementi:

1. un contenitore rettangolare, analogo a quello della top app bar allineata centralmente
2. un titolo di tipo title **large**, allineato a **sinistra**
3. un'icona di navigazione principale (a sinistra)
4. altre icon opzionali (a destra)

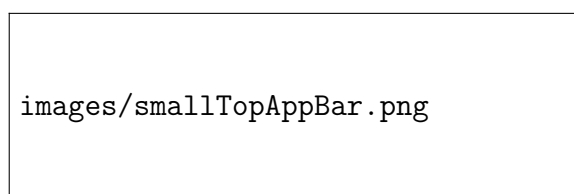


Figura 63: Top app bar piccola

**Top app bar media** La top app bar media è formata dai seguenti elementi:

1. un contenitore rettangolare, analogo a quello delle altre top app bar
2. un titolo di tipo **headline small**, allineato a sinistra, **sotto le icone**
3. un'icona di navigazione principale (a sinistra)
4. altre icon opzionali (a destra)

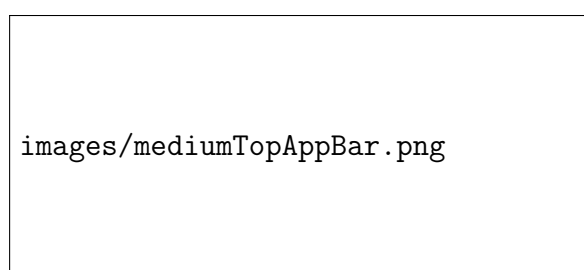


Figura 64: Top app bar media

**Top app bar grande** La top app bar grande è analoga a quella media, ma presenta un titolo più grande, di tipo **headline medium**.

L'icona di navigazione può essere:

- un'icona di **menù**, che apre una navigazione

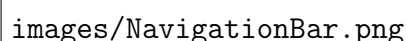
- una **freccia verso l'alto**, che ci permette di spostarsi in alto nella gerarchia dell'applicazione
- una **freccia rivolta a sinistra**, che ci consente di ritornare alla schermata precedente

Il titolo, invece, può descrivere la schermata o la sezione in cui ci si trova, oppure il nome dell'app che si sta utilizzando.

Le icone che si possono inserire nella parte destra della barra possono essere al **massimo 3** e devono essere disposte in modo che l'icona più utilizzata sia quella più a sinistra. Quando sono presenti più di 3 attività da mostrare, quelle di minore importanza possono essere inserite all'interno di un menù, la cui icona occupa la posizione della terza azione più a destra.

### 5.6.2 Barra di navigazione

La **barra di navigazione** presenta un modo veloce e stabile di muoversi all'interno delle **destinazioni principali** dell'applicazione Essa, dovendo essere il facile accesso sui dispositivi mobili, è posizionata nella parte inferiore dell'interfaccia.



images/NavigationBar.png

Figura 65: Barra di navigazione

La disposizione del contenuto della barra non cambia spostandosi all'interno dell'applicazione. Inoltre, le destinazioni presenti dovrebbero essere di importanza simile.

Come per la top app bar, la barra di navigazione non presenta una differenza di elevazione rispetto al contenuto della schermata, ma è distinta da esso attraverso l'uso del colore.



images/navBarComponents.png

Figura 66: Componenti della barra di navigazione

La barra di navigazione è formata dai seguenti componenti:

1. un **contenitore rettangolare**, diviso equamente in colonne
2. una **serie di icone** (minimo 3, massimo 5), ognuna occupante una delle colonne del contenitore
3. le **label** relative alle icone, che devono essere corte e significative
4. un **indicatore** che indica quale delle icone è attualmente **attiva**, che consiste in una forma di colore diverso che circonda l'icona

5. un **badge piccolo** relativo all'icona (opzionale)
6. un **badge grande** relativo all'icona (opzionale)
7. la **label del badge grande** (opzionale)

Le icone delle destinazioni possono essere in uno dei seguenti **stati**: attivata, con il puntatore posizionato su di essa, in focus (il puntatore sta premendo), premuta.

La barra di navigazione deve essere utilizzata solamente su **dispositivi mobili** e deve contenere solo le destinazioni dell'app che devono essere accessibili da qualsiasi punto di essa. Essa non deve essere usata per mostrare singoli task oppure la pagina delle impostazioni.

Quando un'icona è attiva, si utilizza la sua versione riempita, mentre quando non lo è, la sua versione delineata.

L'indicatore di selezione compare dall'icona stessa e si espande partendo da essa. Questa espansione avviene su un singolo asse.

È possibile scegliere di mostrare solamente la label dell'icona attiva, nel caso non sia possibile utilizzare una versione riempita dell'icona.

I badge, che sono opzionali, contengono informazioni dinamiche, come il numero di nuovi messaggi, etc.

La barra di navigazione può essere corredata da un **FAB**, che può essere posizionato superiormente alla barra, nella parte destra dello schermo.

Quando si passa a dispositivi con schermi più grandi e, quindi, un maggiore spazio orizzontale, la barra di navigazione va sostituita con un altro componente, come un cassetto di navigazione laterale.

Quando si seleziona l'icona corrispondente ad una destinazione, l'applicazione naviga automaticamente a tale destinazione e successivamente possono succedere due cose: si può tornare alla posizione in cui l'utente aveva precedentemente lasciato tale destinazione, oppure si può azzerarne lo stato e partire da zero.

### 5.6.3 Floating Action Button

Il **floating action button (FAB)** rappresenta l'**azione principale** della nostra applicazione.

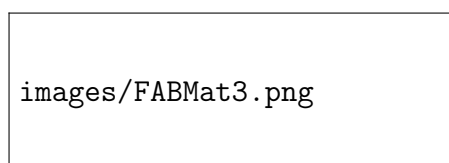


Figura 67: FAB

Il FAB è composto da **due componenti**:

1. il **contenitore**, che presenta un'elevazione di livello 3
2. l'**icona**

Nelle versioni precedenti alla terza, il FAB era circolare. Ora ha una forma più **quadrata**, con gli angoli smussati.

Un FAB può essere negli stessi **4 stati** dell'icona della barra di navigazione (e di ogni pulsante): attivato, con il puntatore posizionato su di esso (hovered), in focus (il puntatore sta premendo), premuto.

Un FAB, una volta premuto, può dividersi in altri FAB più piccoli (minimo 3 e massimo 6).

Il FAB può essere di **tre diverse dimensioni**: FAB (56x56), FAB piccolo (40x40) e FAB grande (96x96). Il **FAB piccolo** è utilizzato per **azioni di supporto** o **secondarie** (quando il FAB si divide, si divide in FAB piccoli). Esso è anche utilizzato per **dispositivi di piccole dimensioni**. Il **FAB grande**, viceversa, è utilizzato per **dispositivi di grandi dimensioni**.

L'icona dentro al FAB deve essere particolarmente chiara, perché il FAB non presenta alcun tipo di label. Tuttavia esso può presentare un suggerimento che compaia quando il FAB è in focus. Anche il colore del contenitore deve essere tale da distinguersi sopra il resto del contenuto.

Il FAB appare sopra a tutti gli altri elementi dell'interfaccia e può essere posizionato centralmente, a destra, a sinistra, o essere incorporato alla barra di navigazione. La sua **posizione** deve essere **consistente** e stabile all'interno dell'interfaccia.

È consigliato inserire un solo FAB per schermata per rappresentare l'azione più comune. Non tutte le schermate devono presentare un floating action button: solo le schermate che ne hanno necessità.

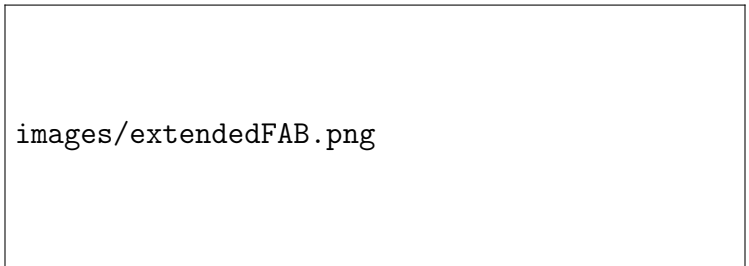
Se un'azione è presente in più schermate successive, il floating action button scompare durante la transizione, per poi ricomparire. Anche durante lo scrolling esso deve essere sempre presente.

Il pulsante, quando viene premuto, può dare vita a diverse animazioni o **trasformarsi** in oggetti diversi, come una toolbar. La trasformazione del FAB in toolbar, e viceversa, è particolarmente utile in determinati meccanismi di scrolling, per evitare di togliere spazio necessario alla fruizione del contenuto. Inoltre, quando viene premuto, il FAB può dare vita ad un'azione nello schermo corrente o aprire direttamente una nuova schermata.

Il FAB dà vita a importanti **azioni costruttive**, come creazione, inserimento tra i preferiti, condivisione, inizio di un processo, e mai distruttive.

#### 5.6.4 FAB esteso

Il **FAB esteso** è un FAB analogo a quello standard, ma di **dimensioni maggiori**, in modo da contenere anche una **label testuale** (1) e avere un'area target più ampia. Esso può anche non contenere nessuna icona e presentare solo la label (2).



images/extendedFAB.png

Figura 68: FAB esteso

Avendo la possibilità di presentare anche una label, questo tipo di FAB è utile nel caso in cui l'uso della sola icona risulterebbe ambiguo.

Il FAB esteso non va utilizzato in pagine che non contengono **contenuto scorribile**. Le altre linee guida sono analoghe a quelle del FAB standard.


### 5.6.5 Pulsanti

I **pulsanti** sono utili all'utente per iniziare diversi tipi di azione.

I pulsanti possono essere collocati **ovunque** all'interno dell'interfaccia utente. Essi, tuttavia, non sono l'unico modo di iniziare azioni e, per questo, non vanno utilizzati in modo eccessivo. Troppi pulsanti all'interno dello schermo danneggiano la gerarchia dei componenti.

Ci sono **5 tipi** di pulsanti:

1. pulsanti **rialzati**
2. pulsanti **riempiti**
3. pulsanti **riempiti tonalmente**
4. pulsanti **delineati**
5. pulsanti **testuali**



images/buttonTypes.png

Figura 69: Tipologie di pulsanti

Il tipo di pulsante va scelto in base all'importanza dell'azione che esso rappresenta: va data **più enfasi alle azioni più importanti**.

A differenza delle versioni precedenti di Material, tutti i tipi di pulsanti hanno **angoli arrotondati**.

**Pulsante rialzato** Il pulsante rialzato presenta un contenitore con **elevazione di livello 1**, all'interno del quale è presente sicuramente una **label testuale** e può essere presente anche un'icona. Questo tipo di pulsante può trovarsi in uno dei seguenti **5 stati**: abilitato, disabilitato, hovered (con il cursore sopra di esso), in focus e premuto.

**Pulsante riempito** Il pulsante riempito presenta un contenitore colorato del **colore primario** con **elevazione di livello 0**. Per le altre caratteristiche è analogo al pulsante rialzato. Questo tipo di pulsante è **il più importante dopo il FAB** e, per questo, va utilizzato per azioni importanti o che finalizzano una serie di azioni o scelte.

**Pulsante riempito tonalmente** Questo pulsante è analogo a quello riempito, ma il contenitore è colorato con il **colore secondario**. L'importanza delle azioni espresse da questo pulsante è minore di quella dei pulsanti riempiti, ma maggiore di quella dei pulsanti delineati.

**Pulsante delineato** Il pulsante delineato presenta un contenitore di cui è ben visibile **sola-mente il contorno**. Per le altre caratteristiche, è analogo ai precedenti. Questo pulsante è di media enfasi ed esprime azioni importanti, ma non primarie nell'applicazione.

**Pulsante testuale** Questo tipo di pulsante presenta un **contenitore totalmente invisibile** (è solo utile per individuare l'area di azione del pulsante, non ha alcuna funzione estetica). Di conseguenza, gli unici elementi visibili sono la label testuale e l'eventuale icona. Quando questo tipo di pulsante si trova negli stati hovered, in focus e premuto, il contenitore si colora. Il pulsante testuale è **il pulsante con minor importanza** ed è spesso integrato all'interno di altri componenti.

La **label testuale** è il componente più importante del pulsante. Essa deve essere **breve e chiara** e deve descrivere l'azione svolta.

L'icona deve essere posizionata alla sinistra del testo (o alla destra di esso nel caso di linguaggi la cui lettura avviene da destra a sinistra). Inoltre, un pulsante non deve presentare più di un'icona.

Le dimensioni del contenitore e la posizione del pulsante si adattano automaticamente ai cambiamenti di dispositivo e di schermo. Il testo e il contenuto vanno quindi definiti in relazione alle dimensioni variabili del contenitore.

### 5.6.6 Chips

Le **chips** permettono all'utente di inserire informazioni, effettuare selezioni, filtrare contenuti e iniziare azioni. Esse possono permettere di mostrare più elementi interattivi nella stessa area.

Ci sono **4 tipi** di chips:

1. assist
2. filtro
3. input
4. suggerimento

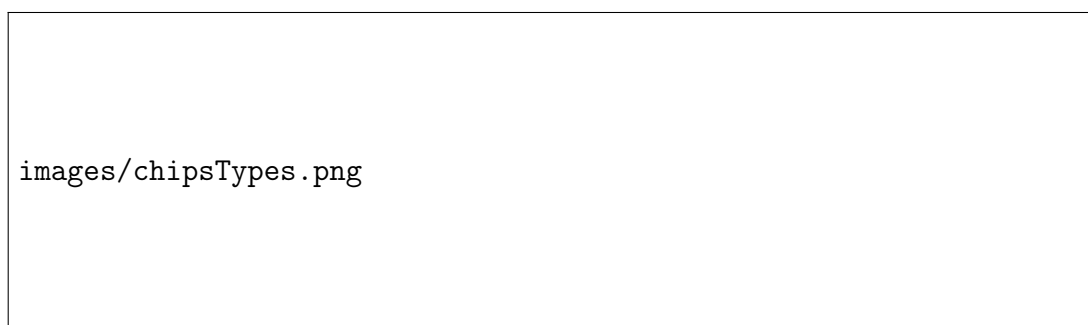


Figura 70: Tipi di chips

Le chip **rappresentano** delle **opzioni** relative ad un determinato contesto: **non** sono **persistenti**, come i pulsanti.

Ogni chip può trovarsi in uno dei seguenti **stati**: abilitato, disabilitato, hovered (con il cursore sopra di esso), in focus, premuto e trascinato.

**Chip assist** La chip di tipo assist è composta da:

- un **contenitore rettangolare** con gli angoli smussati ed elevazione di livello 0 (o di livello 1, nel caso sia necessario differenziare maggiormente il contenitore dal contesto)
- una **label testuale**
- un'**icona** o un'**immagine** (opzionale)

Queste chip rappresentano **azioni automatiche**, che possono anche interagire con altre applicazioni.



**Chip filtro** La chip di tipi filtro presenta gli stessi componenti di quella di tipo assist, in cui l'**icona** è una **spunta**, che compare quando il filtro è attivato. La label deve essere **chiara** ma allo stesso tempo **breve** (si possono tralasciare le regole grammaticali per donare brevità al testo). Queste chip sono utilizzate per **filtrare**, come suggerisce il nome, il contenuto, descrivendo il filtro attraverso l'uso di parole chiave. Esse possono essere utilizzate insieme ad altri componenti (tipo una barra di ricerca).

**Chip input** La chip di input è analoga alle precedenti, ma può presentare un'**icona aggiuntiva** (X) alla destra della label. Queste chip rappresentano **informazioni inserite dall'utente**, trasformando il testo in chip, appunto. In questo modo l'utente può controllare che l'input inserito sia corretto. Esse possono, quindi, supportare la modifica del loro contenuto, ritornando momentaneamente ad essere testo. Le chip di input possono essere espanse per mostrare il proprio contenuto in una forma più estesa.

**Chip suggerimento** La chip di tipo suggerimento **non contiene icone**, ma è composta solamente dal contenitore e dalla label. I suggerimenti permettono di **consigliare all'utente** degli input prestabiliti.

Le chips **non sono pulsanti**. A differenza di essi, infatti, le chips vanno mostrate **insieme in gruppo**. Esse sono reattive e non predeterminate e dovrebbero mostrare **azioni** che **dipendono dal tipo di contenuto** che supportano.

I gruppi di chip possono essere scorribili orizzontalmente. Le chips, infatti, vanno disposte in riga, non in colonna.

## 5.7 Adattamento a dispositivi diversi



Figura 71: Adattamento dispositivi

Quando si sviluppa un'applicazione, è bene sviluppare l'interfaccia sia per l'orientamento **verticale**, che per quello **orizzontale** (landscape). Per fare questo, è utile sviluppare la **UI in colonne multiple di contenuti**, che in base alla larghezza del dispositivo, o all'orientazione di esso, verranno mostrate affiancate o separatamente. Si crea così una **griglia adattabile**, che permette di creare un **layout consistente e coerente**. La larghezza delle colonne e il numero di quelle mostrate contemporaneamente, varia in base alle dimensioni dello schermo su cui si visualizza la griglia.

Per definire le dimensioni dello schermo a cui l'interfaccia deve adattarsi, vengono definiti dei **breakpoint range** (in dp), che, appunto, definiscono il numero delle colonne e i valori per i vari parametri per tutte le varie casistiche di dimensioni.

I dispositivi di grandi dimensioni sono divisi in 3 **regioni di layout**:

1. **navigazione**, che contiene gli elementi di navigazione
2. **app bar**
3. **corpo** (body), che contiene la maggior parte del contenuto dell'applicazione

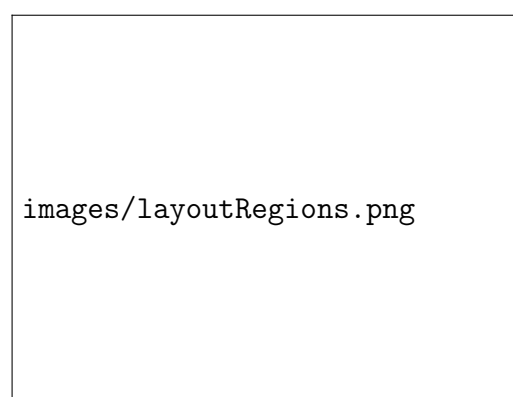


Figura 72: Regioni di layout

Per relazionare elementi che condividono un contesto, viene utilizzato un metodo chiamato **contenimen-**

to. Esso differenzia in modi di raggruppare in **esplicito** e uno **implicito**.

**Contenimento implicito** Il contenimento implicito è un metodo per raggruppare contenuti che non sembrano avere bordi definiti, ma che indicano dei **confini sottintesi**.

**Contenimento esplicito** Il contenimento esplicito **mostra chiaramente i bordi del contenitore**, e può enfatizzare maggiormente la regione interessata con l'uso dell'elevazione o di altri delineamenti.



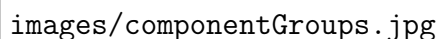
Figura 73: Contenimento implicito (1) ed esplicito (2)

Quando i contenitori vengono scalati e cambiano dimensioni, il contenuto può essere gestito in vari modi: può allinearsi a sinistra, a destra, centralmente o avere posizione fissa.

Quando i componenti o i contenitori vengono scalati, va considerato anche come la loro posizione e il loro allineamento va scalato.

Il metodo più comune per l'adattamento del contenuto è la **rappresentazione visiva**. Questo tipo di adattamento modifica la **scala** e il **posizionamento** del contenuto e degli oggetti nello schermo, tenendo conto delle relazioni che ci sono tra essi.

I componenti dell'interfaccia, inoltre, possono essere scambiati, per rendere l'interfaccia più pulita e funzionale. Essi sono divisi in **gruppi funzionali**, che ne definiscono le variazioni in base alle varie tipologie di dispositivo.



images/componentGroups.jpg

Figura 74: Gruppi funzionali di componenti

Quando occorrono cambiamenti di dimensioni dello schermo, alcuni elementi che erano nascosti, possono essere mostrati, grazie al maggior spazio disponibile. Altri elementi, invece, possono subire riposizionamenti, per fare in modo che tutto lo spazio disponibile venga sfruttato nel miglior modo possibile.

È inoltre consigliabile sviluppare l'applicazione in modo che sia in grado di ricevere **input diversi dal solo tocco**, come un mouse, un trackpad o una tastiera.

Nel caso del mouse, un click del tasto destro del mouse corrisponde ad un tocco, mentre un click del tasto sinistro dovrebbe attivare un menu che mostri opzioni aggiuntive per l'oggetto su cui si è cliccato. Il cursore, inoltre, può assumere diverse forme a seconda dell'oggetto su cui si trova (diventare un indicatore di posizione testuale quando si trova sul del testo, per esempio).

Con l'utilizzo della rotella del mouse, o del trackpad, si può effettuare lo scrolling verticale (ma anche orizzontale) dei contenuti.

L'utilizzo di una tastiera fisica sostituisce semplicemente la tastiera virtuale.

Le linee guida di Material Design 3 definiscono dei **layout canonici**, ossia delle composizioni pronte all'uso, che aiutano ad adattare i layout più comuni ai cambiamenti di dimensioni dello schermo.

Sono disponibili **tre layout**:

- list-detail
- pannello di supporto
- feed

**List-detail** Il layout list-detail mostra a **sinistra** una **lista** e a **destra** il **dettaglio del contenuto** selezionato dalla lista. È consigliato nei casi in cui si vuole scorrere e analizzare contenuti e vederne velocemente i dettagli, per esempio nelle applicazioni di messaggistica

(appaiano il messaggio e la conversazione) o di ascolto di musica (appaiano il dettaglio dell'album e l'artista).



Figura 75: List-detail

**Pannello di supporto** Il pannello di supporto è un modo per espandere lo schermo, in modo diverso dal list-view. In questo, infatti, la **vista principale** e quella **secondaria**, sono **equamente importanti** e integrate l'una nell'altra. Esso, quindi, crea una relazione di **focus** e **supporto**. Il layout sfrutta un contenitore che occupa i due terzi dello schermo ed è accoppiato con un pannello, appunto, di supporto che occupa lo spazio rimanente. È sfruttato tipicamente per la produttività, per la redazione dei documenti e il commento degli stessi e per la fruizione di contenuti e media.

**Feed** I feed sono tipicamente utilizzati per notizie e contenuti social. Esso, infatti, permette di mostrare **più contenuti simultaneamente**, attraverso l'uso di card e liste. Il feed è tipicamente utilizzato per un tipo di esperienza guidata dal contenuto, come per le notizie o le foto.

Material Design, inoltre, fornisce le linee guida per adattare l'interfaccia ai dispositivi piegabili (foldable).



Figura 76: Pannello di supporto

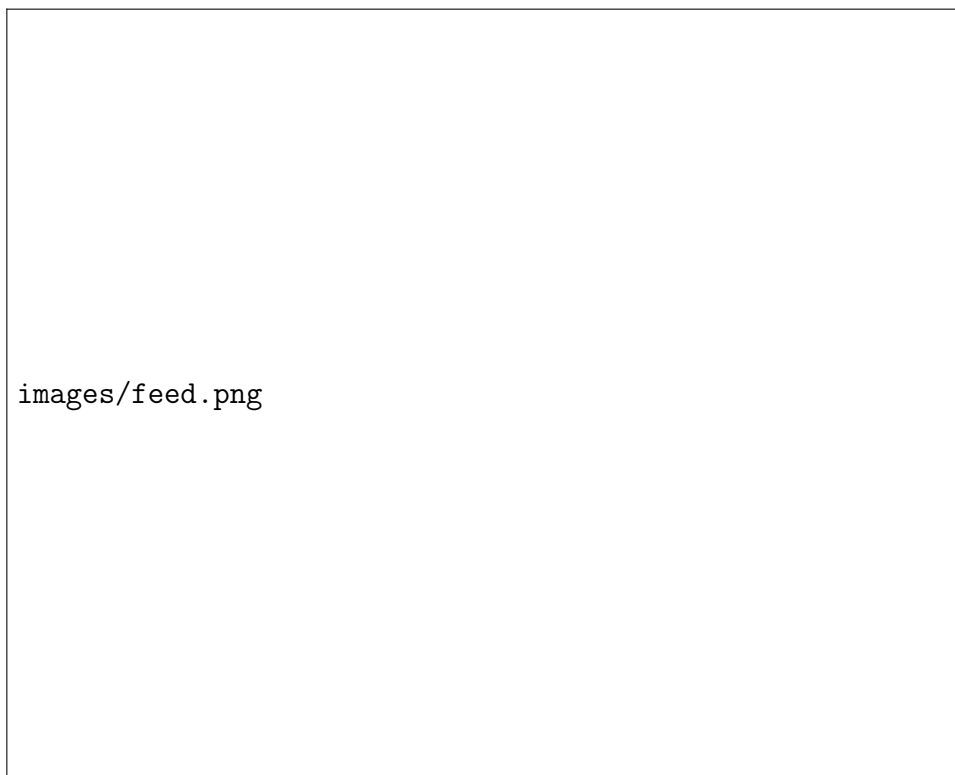


Figura 77: Feed

## 6 BookTracker

Per dimostrare l'utilizzo dei componenti chiave e delle linee guida di Material Design 3, abbiamo sviluppato un'applicazione che ne fa uso: **BookTracker**.

Si tratta, sostanzialmente, di una **libreria virtuale** che permette all'utente di tenere traccia dei libri letti, del progresso di quelli in lettura, dei libri che si possiedono e di salvare delle idee di libri per le letture future.

Nello sviluppare tale applicazione, abbiamo fatto particolarmente attenzione all'aspetto estetico dell'interfaccia e all'utilizzo delle linee guida chiave di Material Design, cercando di mettere in mostra il quantitativo più alto possibile di elementi significativi del tema del progetto.

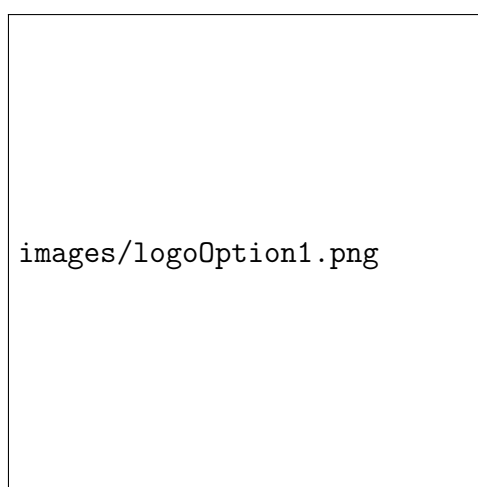


Figura 78: Logo dell'applicazione

## 6.1 Struttura dell'applicazione

BookTracker è un'applicazione che presenta 3 sezioni principali: library (libreria), wishlist (lista dei desideri) e stats (statistiche). Queste sezioni sono accessibili dalla **barra di navigazione**.

La sezione **library** è la sezione in cui possiamo inserire, attraverso l'uso del **FAB**, i libri che fanno parte della nostra libreria, ossia i libri in nostro possesso ma che dobbiamo ancora leggere, i libri che stiamo leggendo e quelli che abbiamo già letto. Nella sezione **wishlist**, invece, inseriamo i libri che vogliamo leggere ma che ancora non abbiamo letto e non possediamo nella nostra libreria. Infine, nella pagina **stats** troviamo le statistiche relative alle nostre letture.

### 6.1.1 Library



Figura 79: Library e dettaglio del libro

La pagina relativa alla libreria presenta, oltre alla **top app bar** e alla barra di navigazione che sono sempre presenti all'interno dell'applicazione, tre **chips** filtro che ci permettono di selezionare quali libri mostrare. Al centro della schermata viene mostrata una lista di **card** relative ai libri della nostra libreria. Ogni card presenta un'immagine della copertina del libro, il titolo, l'autore e un indicatore percentuale dell'avanzamento della lettura. In basso a destra troviamo il FAB che, se premuto, ci permette di aprire la ricerca dei libri e di aggiungere un libro alla libreria, o alla lista dei desideri.



Quando si seleziona uno dei libri, viene mostrata la pagina di dettaglio del libro, che fornisce la possibilità di selezionare se il libro appartiene alla libreria o alla lista dei desideri. Nel caso della libreria, viene mostrato, oltre a copertina, titolo, autore e dettagli del libro, un selettore di pagine, che ci permetterà di impostare il nostro avanzamento nella lettura.

### 6.1.2 Wishlist

La sezione relativa alla lista dei desideri contiene semplicemente una **lista di card**, ciascuna contenente la copertina del libro, il suo titolo e l'autore.

Quando si seleziona un libro, viene aperto il dettaglio relativo a tale libro e relativo alla wishlist: non viene mostrato il selettore delle pagine.



Figura 80: Wishlist e dettaglio del libro

### 6.1.3 Ricerca e aggiunta libro



Figura 81: Ricerca dei libri

La funzione di ricerca e di aggiunta di un libro alla library o alla wishlist viene attivata attraverso il **FAB** posto in basso a destra. Viene così aperto un pannello che presenta un campo di **inserimento testuale**. Inserendo il titolo del libro, lo si può cercare e, una volta selezionato, verrà aperto il dettaglio del libro e si potrà scegliere se aggiungerlo alla libreria o alla lista dei desideri.

### 6.1.4 Stats



Figura 82: Stats

La pagina delle statistiche presenta diverse **card**, ognuna delle quali contenente informazioni relative alle letture effettuate.

Sono presenti due indicatori relativi alla quantità di pagine lette nel giorno di utilizzo corrente e nell'anno corrente. Questi sono corredati con un indicatore percentuale che mette in relazione il valore giornaliero (o annuale) con i valori degli altri giorni (o anni) in cui è stata effettuata una lettura.

Poi sono presenti informazioni riguardanti l'autore più letto e il totale di autori letti, la categoria più letta e il totale di categorie lette e il totale di pagine e libri letti.

### 6.1.5 Impostazioni

La sezione delle impostazioni, raggiungibile tramite l'icona a destra nella top app bar, presenta una **serie di opzioni** che l'utente può utilizzare per **personalizzare** l'applicazione e per **gestire i dati** salvati.

Nella sezione **books** sono presenti le voci per esportare o importare dati relativi ai libri. Inoltre è fornita l'opzione per cancellare tutti i libri salvati nella libreria e nella lista dei desideri.

Nella sezione **app appearance**, invece, è possibile scegliere se utilizzare il tema chiaro, scuro o quello del sistema. Inoltre, è possibile scegliere se utilizzare i colori dinamici o il tema standard dell'app.

Sono poi riportate alcune informazioni relative al prodotto, quali la versione e il nome degli sviluppatori.

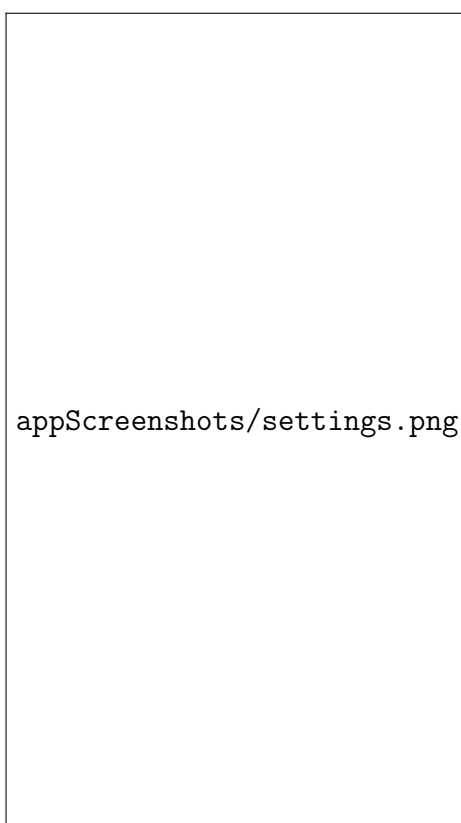


Figura 83: Impostazioni

## 6.2 Colori Dinamici

Seguendo le linee guida di Material Design 3, sono stati implementati i **colori dinamici**. Questi sono attivabili dal menù delle impostazioni e si basano sullo **sfondo** del dispositivo per creare una palette cromatica da applicare all'interfaccia dell'applicazione (e all'icona).

Seguono alcune immagini di esempio.



Figura 84: Colori dinamici con sfondo blu



Figura 85: Colori dinamici con sfondo rosso

I colori dinamici, ovviamente, vengono applicati a tutte le schermate dell'applicazione e anche allo splash screen.



Figura 86: Pagina delle impostazioni e splash screen con colori dinamici

### 6.3 Tema scuro

Tramite le impostazioni è possibile attivare il **tema scuro**.

L'impostazione di default è quella che il tema dell'applicazione è analogo a quello del sistema. Tuttavia il tema scuro può essere impostato anche manualmente all'interno della sola applicazione, senza alterare le impostazioni di sistema.

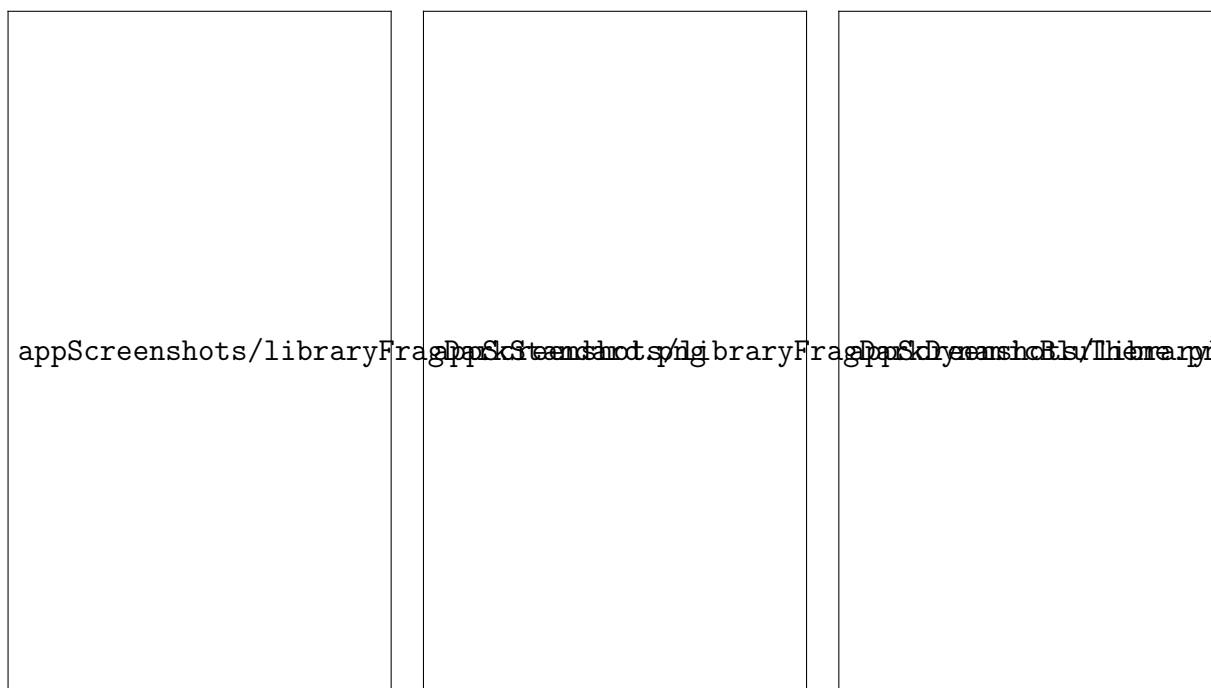


Figura 87: Tema scuro nei vari temi

## Riferimenti bibliografici

- [1] *Color Tool*. URL: <https://material.io/resources/color>.
- [2] *Elevation*. URL: <https://material.io/design/environment/elevation.html>.
- [3] *Iconography*. URL: <https://material.io/design/iconography/>.
- [4] *Imagery*. URL: <https://material.io/design/communication/imagery.html>.
- [5] *Material Design 1*. URL: <https://material.io/archive/guidelines/material-design/introduction.html>.
- [6] *Material Design 1 - Components*. URL: <https://material.io/components>.
- [7] *Material Design 2*. URL: <https://material.io/>.
- [8] *Material Design 3*. URL: <https://m3.material.io/>.
- [9] *Product Icons*. URL: <https://material.io/design/iconography/product-icons.html#design-principles>.
- [10] Christian Robertson, Jon Wiley, Jonathan Lee, Matias Duarte e Nicholas Jitkoff. *Google I/O 2014 - Material design principles*. URL: <https://www.youtube.com/watch?v=isYZXwaP3Q4>.
- [11] *Speed*. URL: <https://material.io/design/motion/speed.html#controlling-speed>.
- [12] *Typography*. URL: <https://material.io/design/typography/the-type-system.html#type-scale>.
- [13] *Understanding Motion*. URL: <https://material.io/design/motion/understanding-motion.html>.
- [14] *What is Flat Design?* URL: <https://www.interaction-design.org/literature/topics/flat-design>.
- [15] *What is Skeuomorphism?* URL: <https://www.interaction-design.org/literature/topics/skeuomorphism>.