

# Material Design 3

Ferin Eli      Nardi Marco      Trincanato Marco

14 giugno 2022

# Indice

<b>1 Esempi e template</b>	<b>4</b>
1.1 Citazioni . . . . .	4
1.2 Blocco di codice . . . . .	4
1.3 Immagini . . . . .	5
<b>2 Principi del Material Design</b>	<b>6</b>
2.1 Ambiente . . . . .	6
2.1.1 Superfici . . . . .	6
2.1.2 Elevazione . . . . .	6
2.1.3 Luci e Ombre . . . . .	7
2.2 Layout . . . . .	7
2.2.1 Anatomia del layout . . . . .	7
2.2.2 Organizzazione dei contenuti . . . . .	8
2.2.3 Misure . . . . .	9
2.2.4 Responsive layout grid . . . . .	9
2.2.5 Metodi di spaziatura . . . . .	9
2.2.6 Contenitori e Aspect Ratios . . . . .	9
2.2.7 Densità . . . . .	10
2.3 Navigazione . . . . .	10
2.3.1 Navigazione laterale . . . . .	10
2.3.2 Navigazione in avanti . . . . .	10
2.3.3 Navigation transitions - da tenere se implementiamo nell'app . . . . .	11
2.4 Colori . . . . .	12
2.4.1 Temi . . . . .	12
2.5 Suono . . . . .	13
2.5.1 Formato dei file e ottimizzazione della memoria . . . . .	14
2.6 Tipografia . . . . .	14
2.6.1 Dimensioni . . . . .	14
2.6.2 Proprietà di un typeface . . . . .	14
2.6.3 Leggibilità . . . . .	15

2.7	Iconografia . . . . .	15
2.7.1	Design delle icone . . . . .	15
2.8	Forme . . . . .	16
2.8.1	Gruppi di forme . . . . .	17
2.8.2	Uso efficace delle forme . . . . .	18
2.8.3	Collegare superfici attraverso le forme . . . . .	20
2.9	Interazione . . . . .	20
2.9.1	Gestture . . . . .	20
2.9.2	Selezione . . . . .	21
2.9.3	Stato . . . . .	22
2.9.4	Tipi di stato . . . . .	22
2.10	Motion . . . . .	23
2.10.1	Transition patterns . . . . .	23
2.10.2	Velocità . . . . .	24
2.11	Comunicazione . . . . .	25
2.11.1	Confirmation and acknowledgement . . . . .	25
2.11.2	Visualizzazione dei dati . . . . .	25
2.11.3	Stati vuoti . . . . .	27
2.11.4	Aiuto e feedback . . . . .	28
2.11.5	Immagini . . . . .	29
2.11.6	Launch screen . . . . .	29
2.11.7	Onboarding . . . . .	30
2.11.8	Scrittura . . . . .	30
<b>3</b>	<b>Novità del material design 3</b>	<b>32</b>
3.1	Accessibilità . . . . .	32
3.1.1	Gerarchia . . . . .	32
3.1.2	Touch targets . . . . .	32
3.1.3	Pointer Targets . . . . .	33
3.1.4	Testo . . . . .	33
3.1.5	Contrasto . . . . .	33
3.1.6	Best practice . . . . .	33
3.2	Personalizzazione . . . . .	33
3.3	Design tokens . . . . .	34
3.4	Interaction states . . . . .	34
3.5	Colori . . . . .	34
3.5.1	Novità nel sistema di colori . . . . .	35
3.5.2	Ottenerne uno schema di colori . . . . .	36
3.5.3	Gerarchia . . . . .	36
3.5.4	Colori dinamici . . . . .	36
3.6	Forme . . . . .	38

3.7	Iconografia material 3 . . . . .	38
3.8	Tipografia material 3 . . . . .	39
<b>4</b>	<b>Material Reviews</b>	<b>40</b>
4.1	Funzionamento e struttura . . . . .	40
4.1.1	Dettagli ristoranti . . . . .	41
4.1.2	Le mie recensioni . . . . .	43
4.1.3	Profilo . . . . .	44

# Capitolo 1

## Esempi e template

### 1.1 Citazioni

Ho seguito delle guide per riuscire a far andare la bibliografia, e in sostanza bisogna:

- avere installato il package biblatex
- quando si compila, eseguire:
  - PdfLatex
  - bibtex
  - PdfLatex
  - PdfLatex
  - ViewPdf

Per fare una citazione bisogna:

- aggiungere la risorsa nel file `resources.bib`
- usare il comando `cit{id_citazione}`

Le citazioni vengono mostrate automaticamente alla fine del documento con il comando `printbibliography`

Qui faremo una citazione di esempio: citazione [1], [1]

### 1.2 Blocco di codice

Creo un blocco di codice con il syntax highlighting [2]:

```
1 // this is a simple code listing:  
2 println("hello kotlin from latex")
```

Se invece voglio semplicemente una parola in monospace: monospace

## 1.3 Immagini

Per inserire immagini usare il template qui definito



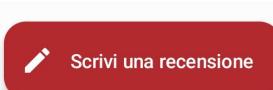
Figura 1.1: Floating Action Button

Nel Material Design, gli elementi dell’interfaccia utente sono costituiti da *superfici*, ossia delle forme geometriche piatte e dai contorni ben definiti.

Una superficie può mostrare un contenuto (testo, immagini, icone, ...), che deve essere anch’esso piatto come la superficie su cui si appoggia. Il contenuto può essere indipendente dalla superficie e può cambiare, ma deve sempre rimanere entro i limiti geometrici imposti dalla superficie.

Le superfici si comportano come dei solidi, che possono cambiare forma e opacità, possono unirsi e dividersi, spostarsi e ruotare. Tutti gli elementi della UI del Material Design sono il risultato delle modifiche degli attributi di una superficie di base, che è bianca opaca, di spessore 1dp, con un’ombra.

Riferimento alla figura 1.1: 2.3



# Capitolo 2

## Principi del Material Design

### 2.1 Ambiente

L’ambiente del Material Design [3] specifica come sono costruiti e come si comportano gli elementi dell’interfaccia utente.

#### 2.1.1 Superfici

Nel Material Design, gli elementi dell’interfaccia utente sono costituiti da *superfici*, ossia delle forme geometriche piatte e dai contorni ben definiti.

Una superficie può mostrare un contenuto (testo, immagini, icone, ...), che deve essere anch’esso piatto come la superficie su cui si appoggia. Il contenuto può essere indipendente dalla superficie e può cambiare, ma deve sempre rimanere entro i limiti geometrici imposti dalla superficie.

Le superfici si comportano come dei solidi, che possono cambiare forma e opacità, possono unirsi e dividersi, spostarsi e ruotare. Tutti gli elementi della UI del Material Design sono il risultato delle modifiche degli attributi di una superficie di base, che è bianca opaca, di spessore 1dp, con un’ombra.

#### 2.1.2 Elevazione

L’elevazione rappresenta la distanza relativa delle superfici lungo l’asse z.

Lo sfondo ha elevazione 0. Tutte le superfici hanno una elevazione definita in dp. L’elevazione viene visualizzata mediante delle ombre, tanto più intense quanto più grande è il distacco tra la superficie in primo piano e la superficie in secondo piano. Altri modi per mostrare l’elevazione sono l’utilizzo di colori e opacità diversi.



Figura 2.1: Elevazione in un Floating Action Button

Quando l’utente interagisce con una superficie, la superficie aumenta la sua elevazione, per poi tornare al suo posto quando l’interazione termina. Tutti gli elementi standard hanno una elevazione di default.

Per evidenziare una superficie in maniera marcata (ad esempio per i messaggi di errore), è possibile oscurare tutto ciò che sta al di sotto della superficie

Le superfici del Material Design non possono occupare la stessa posizione contemporaneamente. Se più elementi devono risiedere nella stessa porzione dello schermo, allora devono avere altezza differente, generando una gerarchia di visualizzazione. Nelle animazioni, le superfici che cambiano elevazione non possono attraversare altre superfici.

### 2.1.3 Luci e Ombre

Nel Material Design delle fonti di luce virtuali illuminano la scena.

Le ombre vengono generate dalla luce che colpisce gli elementi dell’interfaccia utente. L’ombra è tanto più estesa quanto la superficie che colpisce è elevata. Per questo motivo le ombre sono il mezzo con cui si rappresenta la differenza di elevazione tra superfici.

## 2.2 Layout

### 2.2.1 Anatomia del layout

Il Material Design fornisce delle linee guida per ottenere layout che siano intuitivi, consistenti, e reattivi. Ciò è ottenuto attraverso l’uso di griglie, metodi di spaziamento degli elementi, vincoli e altro.[4]

Il layout è diviso in 3 zone:

- *Body*: mostra la maggior parte del contenuto
- *Barra del titolo*: viene usata per mostrare e raggruppare azioni e componenti che possono servire all’utente, in relazione al contenuto del *body*

- *Barra di navigazione*: permette all'utente di spostarsi tra le varie schermate dell'app

### 2.2.2 Organizzazione dei contenuti

#### Raggruppamento visivo

Il raggruppamento è uno dei passi fondamentali per creare ordine in un layout: elementi con contenuti o funzionalità simili devono essere raggruppati assieme e separati da altri elementi diversi utilizzando spazi, tipografia o divisorì

#### Contenimento

Oltre al raggruppamento visivo un altro metodo è contenere assieme elementi che condividono determinate features [5]: per esempio una notizia avrà un'immagine, un titolo e magari un piccolo paragrafo introduttivo.

Il contenimento si può ottenere:

- *implicitamente*: lo spazio tra elementi correlati viene ridotto
- *esplicitamente*: gli elementi vengono racchiusi graficamente, ad esempio con una cornice



Figura 2.2: Le informazioni del ristorante sono racchiuse in una *Card*

### 2.2.3 Misure

Le dimensioni degli elementi della UI possono essere misurate in molti modi.

L'unità di misura più utilizzata è il *density-independent pixel (dp)*, che permette di mantenere le stesse dimensioni in schermi anche molto diversi tra loro.

Nel material design in genere si usano multipli di 8 dp per la maggior parte degli elementi della UI. Per elementi molto piccoli si può usare anche 4 dp come unità di base.

### 2.2.4 Responsive layout grid

Il responsive layout grid è un layout in grado di adattarsi a diverse grandezze schermo e orientamenti per garantire consistenza e armonia [6]

È composto da:

- *Colonne*: in un dispositivo mobile ce ne sono 4
- *Grondaie*: definisco lo spazio che separa le colonne
- *Margini*: definiscono la distanza dal bordo dello schermo

### 2.2.5 Metodi di spaziatura

Per manipolare lo spazio in modo più granulare il layout possono essere utilizzati altri elementi grafici quali:

- *padding*: spazio vuoto che descrive la distanza tra elementi
- *allineamenti*: come gli elementi sono allineati nelle righe/colonne

### 2.2.6 Contenitori e Aspect Ratios

Un container rappresenta un'area che contiene elementi di interfaccia utente. Possono essere rigidi, che quindi tagliano il contenuto al loro interno nel caso vada oltre i limiti, oppure flessibili, che cambiano la propria dimensione in base al contenuto.

Per mantenere consistenza nel proprio layout è consigliato l'utilizzo di Aspect Ratios consistenti per elementi come immagini o superfici. Quelli consigliati sono: 16:9; 3:2; 4:3; 1:1; 3:4; 2:3 [7]

## 2.2.7 Densità

Ci sono casi dove aumentare la densità di informazioni mostrate a schermo può essere utile e rende migliore l'esperienza dell'utente. Se l'utente deve interagire con molte informazioni, si possono rendere le informazioni più compatte diminuendo lo spazio tra loro, rendendole più facili da consultare e confrontare.

È invece sconsigliato usare una alta densità di contenuti nei componenti che si concentrano nello svolgere un a singola azione (*focused task*) o nei messaggi d'errore

## 2.3 Navigazione

La maggior parte delle applicazioni è composta da più schermate che mostrano all'utente diversi contenuti. È quindi fondamentale che l'utente possa muoversi agevolmente tra queste schermate in maniera rapida e intuitiva.

Solitamente esiste una gerarchia tra le schermate di un'applicazione. Questa gerarchia si riflette nell'interfaccia che permette all'utente di navigare.

### 2.3.1 Navigazione laterale

La navigazione laterale viene utilizzata per spostarsi tra pagine allo stesso livello nella gerarchia.

Solitamente viene implementata con una *bottom navigation bar* (preferita nei dispositivi mobili) o con un *navigation drawer*.

Per schermate di gerarchia più bassa possono essere utilizzate le *tabs*.



Figura 2.3: Barra di navigazione

### 2.3.2 Navigazione in avanti

La navigazione in avanti viene utilizzata quando l'utente si sposta tra schermate a livelli consecutivi di gerarchia. Si utilizza la navigazione in avanti ogni volta che si utilizzano pulsanti, collegamenti, ricerche.

A differenza della navigazione laterale, quella in avanti non prevede elementi dell'interfaccia utente dedicati.

Associata alla navigazione in avanti è la navigazione all'indietro, che permette all'utente di ritornare alla schermata cronologicamente precedente oppure ad un livello gerarchico superiore.

### 2.3.3 Navigation transitions - da tenere se implementiamo nell'app

Le transizioni di navigazione si verificano quando gli utenti si spostano tra le schermate e utilizzano il movimento per guidare gli utenti tra due schermate nell'app. Aiutano gli utenti a orientarsi esprimendo la gerarchia dell'app, usando il movimento per indicare in che modo gli elementi sono correlati tra loro. Il material design definisce 2 macro tipi di transizioni:

- *Hierarchical transitions.* Si usano quando gli utenti salgono o scendono di un livello in un'app. Gli schermi a livelli adiacenti tra loro hanno una relazione genitore e figlio l'uno con l'altro, in cui il genitore si trova a un livello gerarchico più alto rispetto al figlio. Nelle transizioni padre-figlio, un elemento figlio presente nello schermo padre si solleva al tocco e si espande sul posto, utilizzando un modello di transizione di trasformazione del contenitore. Il movimento attira l'attenzione sullo schermo figlio (che è la destinazione dell'interazione), mentre rafforza la relazione tra gli schermi genitore e figlio.
- *Peer transitions.* Si verificano tra le schermate allo stesso livello di gerarchia. Le transizioni tra pari si verificano tra schermate che condividono un genitore, mentre le transizioni tra pari di livello superiore vengono utilizzate solo per passare da una destinazione primaria all'altra. Si distinguono 2 tipi di peer transition:
  - *Sibling transitions.* Gli schermi che condividono lo stesso genitore (come le foto in un album, sezioni di un profilo o passaggi in un flusso) si muovono all'unisono per rafforzare la loro relazione reciproca. Lo schermo del peer scorre da un lato, mentre il suo fratello si sposta fuori dallo schermo nella direzione opposta.
  - *Top-level transitions.* Al livello superiore di un'app, le destinazioni sono spesso raggruppate in attività principali (e le attività potrebbero non essere correlate tra loro). Queste schermate si aprono utilizzando uno schema di transizione di dissolvenza.

## 2.4 Colori

Il Material Design definisce delle *palette* di colori, con lo scopo di armonizzare e rendere piacevole alla vista ciascun elemento delle app.

Per ciascuna palette sono definiti dei colori specifici per ciascun ruolo degli elementi della UI:

- *primary*: indica il colore principale degli elementi dell'app, viene utilizzato per gli elementi della UI come la barra di navigazione
- *secondary*: indica un colore alternativo, per distinguere gli elementi della UI vicini tra loro (uno di colore primario e uno di colore secondario)
- *primaryVariant* e *secondaryVariant*: sono delle versioni leggermente alterate da utilizzare quando più elementi si sovrappongono o hanno importanza diversa
- *background*: colore dello sfondo
- *surface*: colore di default delle superfici di menu, cards, ...
- *error*: colore per i messaggi d'errore
- Colori *on[...]*: da applicare al testo e alle icone che si trovano su un determinato colore di sfondo. Questi colori garantiscono un adeguato contrasto con lo sfondo stesso
  - *onPrimary*
  - *onSecondary*
  - *onBackground*
  - *onSurface*
  - *onError*

### 2.4.1 Temi

Dato un colore *primary*, può essere definita una palette di colori coerente con il colore scelto.

In particolare il Material Design indica delle linee guida per generare una palette per il *tema chiaro* e una per il *tema scuro*. Quest'ultima è generata desaturando i colori per il tema chiaro, così da renderli più leggibili su superfici scure.

I colori particolarmente adatti alle linee guida del Material Design sono forniti dal [Color Tool](#) [8]

## 2.5 Suono

Il suono viene usato per migliorare l'esperienza dell'utente: comunica un feedback utile, assegnando personalità e perfezionando l'estetica del prodotto.

In un'interfaccia utente, si distinguono tre tipologie di suono: **sound design**, **musica** e **voce**. Ogni tipologia comunica informazioni e *brand identity* in modi diversi.

Il **sound design** viene utilizzato principalmente per associare un elemento dell'interfaccia utente a un suono specifico e far capire qual è la struttura gerarchica. Tuttavia, si usa anche per esprimere personalità o fornire feedback.

La **musica** viene utilizzata principalmente per la narrazione ed esprimere uno "stato d'animo" dell'interfaccia utente. Se abbinata a immagini o transizioni, la musica può elevare la narrativa e le sensazioni generali di un prodotto.

La **voce** e la sintesi vocale utilizzano entrambe il linguaggio parlato per comunicare informazioni dove il sound design e la musica non sono abbastanza.

### Quando non usare il suono

Il silenzio nel design dell'interfaccia utente è importante tanto quanto il suono. In molti casi, l'audio non è necessario e può, ridurre il comfort dell'utente. Il suono, di solito, non è adatto per UI che richiedono privacy e per azioni eseguite di frequente.

### Tipi di suono

- **Hero sound:** si verificano in interazioni fondamentali, evocando uno stato emotivo o celebrando un certo tipo di eventi;
- **Notifiche:** aiutano a dirigere l'attenzione dell'utente, più brevi degli hero sound e realizzate affinché possano essere riprodotte più volte;
- **Suoni di sistema:** divisi in:
  - *suoni UX primari*: generati da un sistema operativo (o dispositivo) per fornire feedback agli utenti;
  - *suoni UX secondari*: riprodotti meno frequentemente e utilizzati principalmente per scopi funzionali.
- **Suoni ambientali:** creano atmosfera senza distogliere l'utente dall'attività, in genere basati su musica o suoni ambientali

### 2.5.1 Formato dei file e ottimizzazione della memoria

La riproduzione del file audio può variare a causa delle limitazioni hardware e software di un dispositivo. Per ridurre le dimensioni del file si usano diversi metodi, come la compressione con perdita o la riduzione della frequenza di campionamento.

Il formato finale dell'audio dipende dall'implementazione e dalle restrizioni a livello di sistema. Quindi, di solito si sceglie il formato più lossless consentito dal sistema.

## 2.6 Tipografia

Il material design fornisce molti strumenti per poter scegliere il font adatto con le misure giuste per ogni tipo di testo, forniscono pure un [Type scale generator](#) che permette di creare il proprio type scale. Vengono fornite misure e conversioni per Android, iOS e Web; e misure per titoli, sottotitoli e corpi di testo.

### 2.6.1 Dimensioni

Le categorie di testo individuate dal Material Design sono

- *headline*: con 6 dimensioni disponibili, headline viene utilizzata per i titoli
- *sottotitoli*: più piccoli dei titoli, sono sempre di breve lunghezza
- *corpo*: viene utilizzato per il contenuto testuale dell'app, che può essere anche molto lungo
- *testo per i pulsanti*: in genere con sole lettere maiuscole, è dedicato ai pulsanti
- *didascalie e overline*: sono i type scale più piccoli. Didascalie spesso servono per annotare degli elementi mentre overlines a introdurre dei titoli.

### 2.6.2 Proprietà di un typeface

Le proprietà che caratterizzano un typeface o permettono di modificarne le misure sono

- *baseline*: è una linea invisibile che definisce come i singoli caratteri si allineano sulla griglia (ad esempio definisce quanto sono grandi le lettere maiuscole rispetto alle minuscole)
- *peso*: rappresenta lo spessore del tratto dei caratteri. Assume valori da 1 (*light*) a 4 (*bold*)

### 2.6.3 Leggibilità

La leggibilità di un testo è determinata da molteplici fattori quali

- *letter spacing*: si riferisce allo spazio tra le lettere che compongono una parola, a seconda della funzione del testo si possono usare spazi più piccoli o grandi
- *tabulazione*: aiuta a leggere cifre lunghe quando sono vicine ad altre, come per esempio in colonna in una tabella
- *lunghezza e altezza delle righe*

## 2.7 Iconografia

Il material design fornisce strumenti, griglie e specifiche per utilizzare e creare le proprie icone, con l'obiettivo di mantenere un certo livello di semplicità.

### 2.7.1 Design delle icone

Nel Material Design, le icone sono realizzate per essere semplici e moderne, ridotte alla forma minima, esprimendo solo concetti essenziali, con forme geometriche semplici e consistenti. Questa semplicità permette alle icone di rimanere leggibili e distinguibili anche a dimensioni piccole.



Figura 2.4: Icona di MaterialReviews.

## Dimensioni

Le icone sono normalmente 24dp x 24dp, nel caso di layout più densi come su un desktop possono essere scalate a 20dp x 20dp. Per ogni icona vi è un perimetro di padding di 2dp, per prevenire sovrapposizioni o oscuramenti in layout densi o nel caso appaiano altri elementi a schermo.

Vengono fornite delle keylines e griglie per il design di icone: principalmente vengono usati quadrati, cerchi e rettangoli. Il tratto che si usa quando si crea la propria icona deve essere sempre spesso 2dp, questo mantiene consistenti le icone, i tratti finali di un'icona, per esempio la punta di una freccia, sono sempre dritti e non tondi.

## Orientamento

Le icone devono sempre guardare avanti, non vanno girate o piegate, e devono sempre apparire piatte e non tridimensionali. L'opacità di un'icona dipende dallo sfondo e dal suo stato: attiva o inattiva.

## Transizioni

Le transizioni sono effetti tra due icone, quando si preme un'icona viene eseguita l'animazione e l'altra icona diventa visibile o attiva, per esempio cliccando l'icona *pausa* in un lettore di musica, la musica si ferma e l'icona diventa *riprendi*. Le animazioni possono essere semplici (una semplice rotazione), o complesse (trasformazioni e altri effetti), spesso se la funzionalità è più importante dell'aspetto visivo si scelgono animazioni semplici. La durata di un'animazione semplice (come un pulsante on/off) è di 100ms, animazioni un po' più complesse 200ms (per esempio una rotazione), per animazioni che fanno uso di molte trasformazioni si usano 500ms così da poter vedere bene l'animazione.

## 2.8 Forme

Gli elementi di un'applicazione possono essere visualizzati attraverso diverse forme. Le forme dirigono l'attenzione, identificano i componenti, mostrano come le superfici si relazionano tra loro ed esprimono il brand.

Le componenti hanno una forma rettangolare per impostazione predefinita, con angoli arrotondati di 4 dp (density pixel). La loro forma può essere personalizzata regolando dimensione e forma degli angoli.

La forma di alcuni componenti di un'applicazione viene modificata per vari motivi: le forme con un design unico (*forme uniche*) differiscono dalle

forme che le circondano e dall’interfaccia utente nel suo insieme, aiutando a dirigere l’attenzione dell’utente. Inoltre, una forma diversa rispetto alle altre componenti di quel gruppo può comunicare il cambiamento di stato di una componente utilizzando .

Ancora, per esprimere il linguaggio visivo di un **marchio** in un’app, si usano le forme insieme ad altre personalizzazioni, come colore e tipografia, in modo coerente. Queste piccole modifiche alla forma, applicate strategicamente, contribuiscono all’impressione generale che un brand da all’utente.

### 2.8.1 Gruppi di forme

I componenti sono raggruppati in gruppi di forme in base alle loro dimensioni. In questo modo, si possono modificare contemporaneamente più valori di un gruppo di componenti. I gruppi di forme includono:

- *Componenti piccoli*, come pulsanti, chip, campi di testo, snackbar;
- *Componenti medi*, come menu, card, dialog;
- *Componenti grandi*, come sfondi, tabelle, drawer

I gruppi di forme utilizzano degli attributi per definire la forma da assegnare agli angoli dei componenti. È possibile personalizzare i seguenti attributi:

- *tipo di angolo*: appuntito, arrotondato, tagliato;
- *dimensioni*: il raggio dell’angolo. Può essere assoluto (in dp) oppure in funzione delle dimensioni della forma stessa;
- *simmetria*: è possibile applicare modifiche solo ad angoli specifici.

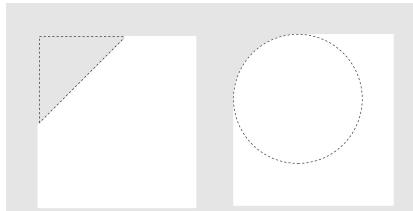


Figura 2.5: A sinistra angolo tagliato. A destra angolo arrotondato.

Per applicare una famiglia di forme e una dimensione a tutti i componenti di un gruppo, bisogna impostare i valori per il gruppo di forme. Quando

vengono apportate modifiche a un gruppo, le modifiche influiscono su tutti i componenti in quel gruppo di forme, ad eccezione di quelli con una sostituzione.

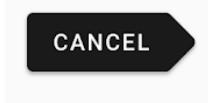


Figura 2.6: Forma asimmetrica.

### 2.8.2 Uso efficace delle forme

Le forme sono visibili quando i bordi della superficie hanno un contrasto sufficiente rispetto allo sfondo. Per impostazione predefinita, il Material Design rende visibili le forme utilizzando le ombre per visualizzare i bordi delle superfici. Tuttavia, esistono altri metodi per rendere visibili le forme, come riempimenti di colore e opacità, che possono essere utilizzati in combinazione con le ombre o da soli.

Le forme aiutano gli utenti a identificare i componenti e influiscono sulla loro usabilità.

La forma può comunicare molte informazioni su un elemento, incluso il suo stato corrente, il risultato di un’interazione dell’utente o altre modifiche in un’app. Se utilizzata in questi modi, la forma dovrebbe essere presentata in modo coerente nello stesso stato e nelle stesse interazioni, in modo che una forma specifica abbia lo stesso significato ogni volta che viene incontrata.

Se una forma non è interattiva, è necessario evitare l’utilizzo di forme con dimensioni sufficientemente grandi da apparire interattive. Ad esempio, una piccola forma triangolare su una card non dovrebbe essere abbastanza grande da essere scambiata per un elemento da toccare, se non lo è.

### Branding

La combinazione di stili e forme diverse può rendere difficile l’associazione di forme particolari a un marchio. Per esprimere l’identità di un brand, si identifica la forma distintiva del marchio, ad esempio una forma organica o geometrica che riflette gli attributi del logo. Sulla base di questa forma, viene poi sviluppata una serie di forme simili da applicare al prodotto che contribuiscono a unificare l’espressione del marchio in tutta l’applicazione. Tuttavia, l’uso eccessivo della forma per esprimere il marchio può portare alla diluizione dello stesso: la forma perde la sua connessione con il marchio e diventa un luogo comune. Inoltre, troppe forme uniche possono portare

a nessuna forma particolarmente prominente, nonché a una mancanza di coesione visiva tra i componenti.

### Angoli

E' necessario evitare di utilizzare valori percentuali per i componenti che modificano dinamicamente la propria altezza, altrimenti la forma potrebbe modificarsi in modi inaspettati. Inoltre, gli angoli ancorati ai bordi dello schermo non possono essere personalizzati, altrimenti si creerebbero spazi vuoti che lasciano trasparire il contenuto dietro ad un componente.

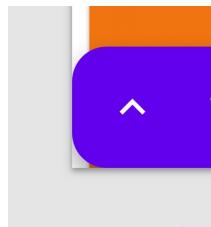


Figura 2.7: Forma inappropriata arrotondata a bordo schermo.

### Modifiche ad altri sistemi

Le personalizzazioni del sistema possono influire sull'uso della forma da parte di un componente. Ad esempio, l'aumento della dimensione del carattere potrebbe causare il taglio di parte di un testo e la modifica di altezza e forma del componente, se le dimensioni sono definite come percentuali. Le modifiche al contenuto possono anche influire sulla forma di un componente. Ad esempio, se la lunghezza del testo aumenta, il testo potrebbe andare a capo su una riga aggiuntiva, aumentando l'altezza del componente e influenzando le forme definite come percentuale.

### Morphing

Gli elementi possono cambiare forma in risposta alle modifiche al contenuto o all'interazione dell'utente. Questa trasformazione si chiama *morphing*. Le forme possono trasformarsi per vari motivi: in risposta a cambiamenti di stato o contenuto o come risultato dell'interazione dell'utente. Ad esempio, quando si ruota un dispositivo mobile da verticale a orizzontale, le superfici possono rispondere modificando le dimensioni, il che può causare la trasformazione delle forme. Il morphing di una forma dovrebbe preservare le proporzioni delle forme distintive, per evitare la distorsione. Inoltre, tutto



Figura 2.8: Forma che assume un comportamento inaspettato.

il contenuto su una superficie dovrebbe essere visibile mentre la superficie si trasforma, senza ritagliare il contenuto.

### 2.8.3 Collegare superfici attraverso le forme

Le forme possono aiutare gli utenti a capire in che modo le superfici sono correlate tra loro. Forme simili possono indicare che le superfici sono dello stesso tipo, come le carte in una raccolta con dimensioni e angoli corrispondenti.

Le superfici correlate tra loro possono essere indicate utilizzando forme che ricordano le frecce, in modo tale da "puntare" verso altre superfici. Ad esempio, un angolo a forma di freccia di un menu può puntare a una superficie correlata.

Le forme possono enfatizzare le superfici separate l'una dall'altra. Ad esempio, quando una forma unica appare a un'altezza maggiore rispetto a un'altra superficie, si sottolinea che le due superfici sono separate.

## 2.9 Interazione

### 2.9.1 Gesture

Le *gestures* permettono all'utente di interagire con gli elementi dello schermo in maniera rapida e intuitiva.

Gli elementi con cui è possibile interagire hanno degli accorgimenti grafici che permettono all'utente di intuire la gestione da eseguire e il suo effetto nello schermo

#### Gestione di navigazione

Le gestioni di navigazione permettono all'utente di spostarsi tra le schermate dell'app:

- *tap*: la schermata cambia quando viene toccato un elemento
- *scroll and pan*: la schermata scorre seguendo il tocco dell'utente

- *drag*: permette all’utente di spostare elementi per visualizzarli o nasconderli
- *swipe*: permette la navigazione orizzontale tra schermate
- *pinch*: modifica le dimensioni delle superfici della UI

### Gesture di azione

Le gesture di azione permettono all’utente di eseguire azioni e avere accesso a ulteriori funzionalità

- *tap*: esegue azioni di base, come la navigazione
- *long press*: permette di accedere a azioni e modalità generalmente nascoste e poco accessibili
- *swipe*: permette di eseguire velocemente azioni quando viene superato un certo *threshold*. Viene generalmente utilizzata nelle liste

### Gesture di trasformazione

Le gesture di trasformazione permettono di spostare, ridimensionare, ruotare gli elementi della UI

- *double tap*: permette di ingrandire e rimpicciolire il contenuto della schermata. Ha una funzionalità molto simile al *pinch*
- *compound gestures*: sono delle gesti che permettono di combinare spostamenti, zoom e rotazioni. Vengono spesso utilizzati nelle mappe
- *pick up and move*: combina un *long press* con un *drag*. Ha lo scopo di riordinare gli elementi di un insieme ordinato

### 2.9.2 Selezione

La selezione permette di specificare gli elementi della schermata su cui si vuole eseguire una determinata azione.

Nei dispositivi touch, la selezione viene effettuata con un *long press*, oppure con delle *selection mode* da cui è possibile selezionare gli elementi con un *tap*.

Quando un elemento viene selezionato, l’utente riceve un feedback, sotto forma di spunta o di un cambiamento di colore dell’elemento stesso.

### 2.9.3 Stato

Lo stato di un componente viene visualizzato mediante una variazione del suo aspetto.

Le variazioni associate a ciascuno stato devono essere:

- *distinguibili*: ogni stato deve essere riconoscibile
- *additive*: se più stati sono presenti nello stesso momento, tutti devono essere rappresentati
- *consistenti*: devono essere le stesse indipendentemente dal componente su cui sono applicate

Inoltre, le variazioni devono sempre garantire la leggibilità e l'armonia generale delle componenti

### 2.9.4 Tipi di stato

I tipi di stato che generalmente vengono rappresentati sono:

- *enabled/disabled*: indica se è possibile interagire con il componente
- *hover*: indica se il cursore è posizionato sopra il componente. Suggerisce che è possibile un'interazione
- *focused*: indica che il componente è evidenziato da un sistema di input diverso dal tocco (ad esempio un assistente vocale o una tastiera esterna)
- *selezionato*: indica se l'utente ha selezionato il componente
- *attivato*: indica, generalmente in una lista, con quale componente si sta interagendo
- *premuto*: tipico dei pulsanti, indica che l'utente ha premuto con un *tap* il componente
- *trascinato*: indica se l'utente sta trascinando il componente con un *drag*
- *on/off*; tipico di *switch* e *checkbox*, indica lo stato booleano del componente
- *error*: indica che qualcosa non va

## 2.10 Motion

Le transizioni aiutano l'utente a capire quali elementi sono correlati, e rendono l'interfaccia utente più bella da vedere. Animazioni in loghi, icone, immagini possono dare feedback all'utente per delle azioni, per esempio premere un pulsante fa partire un'animazione che mostra che sia stato effettivamente premuto. Le animazioni possono anche insegnare all'utente come fare una certa azione: l'azione *swipe to unlock* è animata mostrando la schermata che effettivamente scorre verso l'alto.

### 2.10.1 Transition patterns

Material design fornisce delle patterns per transitions e sono:

- Container transform
- Shared axis
- Fade through
- Fade

Per scegliere quale pattern usare vengono fornite delle linee guida:

#### Container transform

Nel caso siano presenti dei container che rimangono visibili a schermo durante tutta la transizione (sono appunto chiamati persistent), è consigliato l'utilizzo di una container transform. Come mostrato [qui](#) il container transform è molto utile per mostrare relazioni tra elementi, come per esempio una scheda e i suoi dettagli. Ne esistono diverse varianti ma tutte hanno in comune il container che aprendosi si allarga a determinate dimensioni (schermo intero o anche un piccolo menu).

#### Shared axis

Gli elementi collegati sia dal punto di vista semantico che spaziale, possono essere animati attraverso una *shared transformation*: gli elementi che escono ed entrano si muovono allo stesso modo attraverso un asse deciso come mostrato [qua](#). A seconda dell'asse utilizzato si possono avere diversi significati, per esempio procedere avanti attraverso un form da compilare può essere visto come un'animazione orizzontale, che rappresenta lo scorrimento di una pagina.

## Fade Through

Elementi che non sono collegati possono essere animati con un semplice fade through. Questa animazione è spesso usata per elementi come destinazioni di navigation bars.

## Fade

Il fade è utilizzato per elementi che entrano lo schermo utilizzandone solo una parte, come un [menù delle opzioni](#).

## 2.10.2 Velocità

La velocità di una transizione è molto importante perché se è troppo veloce l'utente non capisce cosa sta accadendo, mentre se è troppo lenta rallenta anche l'esperienza dell'utente. Per modificare la velocità si possono cambiare durata e *easing* (accelerazione). Vengono fornite anche qua delle linee guida in base al tipo di transizione:

### Chiusura

Transizioni che chiudono elementi, o tornano indietro devono essere corte in quanto richiedono meno attenzione dall'utente. Elementi piccoli o di poca importanza hanno anche loro transizioni che durano poco.

### Elementi grandi

Elementi che occupano grande parte dello schermo sono quelli che devono avere animazioni più lunghe

### Easing

L'easing permette alle transizioni di accelerare o rallentare rendendole più naturali e meno rigide. Come si vede [qua](#) l'easing rende il movimento del puntino molto più naturale. Ne esistono diversi tipi:

- Standard
- Emphasized
- Decelerated
- Accelerated

## 2.11 Comunicazione

### 2.11.1 Confirmation and acknowledgement

Le comunicazioni di **conferma** e **acknowledgement** chiedono l'autorizzazione all'utente prima di intraprendere un'azione e gli confermano la riuscita delle stesse.

Le azioni di *conferma* chiedono all'utente se desidera procedere con l'azione appena eseguita. Possono essere associate a un avviso o ad informazioni critiche relative a tale azione. La conferma non è necessaria quando le conseguenze di un'azione sono reversibili o trascurabili. La richiesta viene rese al meglio utilizzando una finestra di avviso (alert dialog).

Le azioni di *acknowledgement* forniscono un testo per far sapere all'utente se un'azione che ha scelto è stata completata. Viene visualizzato per un breve lasso di tempo e può includere un'opzione per annullare l'azione. Gli acknowledgement possono essere resi da una varietà di componenti:

- *Alert*. Utilizzare gli avvisi per inviare un messaggio in-app persistente che informa gli utenti di un particolare stato di modifica;
- *Snackbar*. Utilizzare uno snack bar per fornire un breve feedback su un'operazione;
- *Stati vuoti*. Quando un'interfaccia utente è disponibile solo online e il contenuto non è stato caricato o sincronizzato, utilizzare uno stato vuoto.

### 2.11.2 Visualizzazione dei dati

La visualizzazione dei dati rappresenta le informazioni in forma grafica. È una forma di comunicazione che ritrae informazioni dense e complesse attraverso elementi visivi, per semplificare il confronto dei dati. Principi della visualizzazione grafica:

- *Accuratezza*: priorità all'accuratezza, alla chiarezza e all'integrità dei dati, presentando le informazioni in modo da non distorcerle;
- *Utilità*: aiuta gli utenti a navigare tra i dati con contesto e convenienza che enfatizzano l'esplorazione e il confronto;
- *Scalabilità*: adatta le visualizzazioni per dispositivi di dimensioni diverse, anticipando le esigenze degli utenti in termini di profondità, complessità e modalità dei dati.



Figura 2.9: Alert dialog.

La visualizzazione dei dati può essere espressa in diverse forme. I grafici sono un modo comune di esprimere i dati, in quanto ne rappresentano diversi tipi e consentono il confronto tra essi. Il tipo di grafico che si usa dipende principalmente da due cose: i dati che si vogliono comunicare e ciò che si vuole trasmettere attraverso quei dati.

## Dashboards

La visualizzazione dei dati può essere visualizzata su una serie di più grafici, in interfacce utente denominate dashboard.

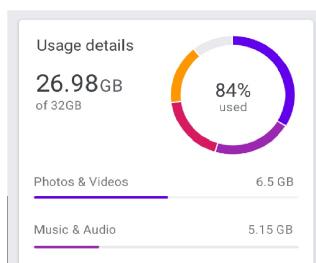


Figura 2.10: Operation dashboard.

Il design di una dashboard dovrebbe adattarsi al modo in cui verrà utilizzata, sia che si tratti di uno strumento per fare una presentazione o di esplorare a fondo i dati. Una dashboard dovrebbe:

- Dare la priorità alle informazioni più importanti (usando il layout);
- Visualizzare un punto focale che dia priorità alle informazioni in base alla gerarchia (utilizzando colore, posizione, dimensione e peso visivo);

Esistono diversi tipi di dashboards:

- *Dashboard analitiche*

Consentono agli utenti di esplorare più set di dati e scoprire le tendenze. Includono grafici complessi che consentono l'individuazione di informazioni dettagliate sui dati. I casi d'uso includono:

- Evidenziare le tendenze nel tempo;
- Rispondere alle domande "perché" e "e se";
- Previsione;
- Creazione di report approfonditi.

- *Operation dashboard*

Sono progettate per rispondere a una serie predefinita di domande. In genere vengono utilizzati per completare attività relative al monitoraggio. Presentano informazioni aggiornate disposte in una serie di semplici grafici. I casi d'uso includono:

- Monitoraggio dei progressi attuali rispetto a un obiettivo;
- Monitoraggio delle prestazioni del sistema in tempo reale.

- *Presentation dashboards*

Forniscono un'istantanea curata su un argomento di interesse. Includono alcuni piccoli grafici o una scorecard, con titoli dinamici che spiegano le tendenze e le informazioni dettagliate fornite in ciascun grafico di supporto. I casi d'uso includono:

- Fornire una panoramica degli indicatori chiave di prestazione;
- Creazione di un sommario esecutivo di alto livello.

### **2.11.3 Stati vuoti**

Gli stati vuoti si verificano quando il contenuto di un elemento non può essere mostrato.

Lo stato vuoto più elementare consiste in un'immagine non interattiva e uno slogan di testo. Si usa un'immagine che ha un tono neutro o umoristico

ed è coerente con il marchio. Lo slogan dovrebbe, invece, avere un messaggio utile, essere coerente con il marchio e trasmettere lo scopo dell'app, senza apparire cliccabile.



Figura 2.11: Stato vuoto.

## Alternative

Le schermate che altrimenti sarebbero vuote possono essere popolate con **contenuti iniziali**. Ciò consente agli utenti di iniziare a utilizzare un'app immediatamente, rendendo più facile per loro conoscere cosa un'app ha da offrire.

Se lo scopo dello schermo non è facilmente veicolato attraverso un'immagine e uno slogan, si mostrano **contenuti didattici**. I contenuti didattici aiutano gli utenti a capire cosa sarà in grado di fare un'app una volta che avrà dei contenuti.

### 2.11.4 Aiuto e feedback

Il contenuto della guida fornisce risposte alle domande e ai dubbi degli utenti. Gli utenti possono inviare commenti, segnalare bug e porre domande a cui non è già stata data risposta. La guida dovrebbe essere facile da trovare. Può essere posizionata in vari punti della navigazione. Di solito appare in un drawer di navigazione o in un menu di overflow sotto l'etichetta "Aiuto" o "Invia feedback".

Per fornire assistenza per problemi urgenti, come pagamenti e rimborsi, posizionare un’icona della Guida nella barra dell’app. Le app desktop possono anche inserire un’icona della Guida nella barra delle app, poiché c’è più spazio nell’interfaccia utente del desktop.

### 2.11.5 Immagini

Le immagini comunicano e differenziano un prodotto. Possono sia migliorare l’esperienza dell’utente che esprimere il linguaggio visivo di un marchio. Aiutano a raccontare una storia, a chiarire messaggi complessi difficili da esprimere con le parole e a mostrare agli utenti come eseguire un’azione.

Per garantire l’accessibilità, le immagini dovrebbero includere un testo alternativo (o una didascalia) che gli screen reader leggeranno agli utenti con disabilità visive. In compose, per descrivere l’immagine si usa l’attributo content description:

```
1 Image(  
2     bitmap = imageData!!.asImageBitmap(),  
3     contentDescription = "Immagine del ristorante",  
4     contentScale = ContentScale.Crop  
5 )
```

Avere un punto focale iconico nelle immagini, è importante, poiché influenza sul modo in cui poi l’immagine viene tagliata per schermi con dimensioni diverse.

### Miniature

Le miniature sono piccole immagini che rappresentano informazioni in spazi ristretti. In genere fungono da target di tocco che portano al contenuto principale, apparendo all’interno di componenti come schede o elenchi. Le miniature vengono utilizzate per:

- Alludere a maggiori informazioni;
- Fornire un’anteprima di contenuti su altri schermi;
- Assistere nella navigazione.

### 2.11.6 Launch screen

La schermata di avvio è la prima esperienza che un utente ha con l’app. Può essere visualizzata all’avvio, mentre l’app viene caricata, al posto di visualizzare una schermata vuota. Questo può ridurre l’impressione di un lungo

tempo di caricamento e ha il potenziale di migliorare l’esperienza dell’utente. Esistono due tipi di schermate di avvio:

- Schermate che mostrano un’anteprima non interattiva dell’interfaccia utente effettiva dell’app, chiamate interfacce segnaposto.
- Schermate che forniscono un’esposizione momentanea del marchio, visualizzando un logo o altri elementi che migliorano il riconoscimento del marchio.

### 2.11.7 Onboarding

L’onboarding è un’esperienza di unboxing virtuale che aiuta gli utenti a iniziare con un’app. Sono disponibili tre modelli di onboarding:

- **selezione automatica:** consente agli utenti di personalizzare le proprie esperienze;
- **avvio rapido:** avvia l’utente direttamente nell’app;
- **illustrazione dei principali vantaggi per l’utente:** mostra un carosello o una breve animazione che evidenzia i vantaggi dell’utilizzo dell’app.

Nel modello **Quickstart**, gli utenti atterrano direttamente nell’interfaccia utente senza alcun modello di onboarding mostrato (a parte l’accesso e la configurazione). Il modello Quickstart:

- Consente agli utenti di iniziare rapidamente con le funzionalità principali dell’app;
- Spesso dà priorità alla prima azione chiave;
- Può anche fornire un modo opzionale per saperne di più o chiedere aiuto.

### 2.11.8 Scrittura

Il testo dovrebbe essere comprensibile da chiunque, ovunque, indipendentemente dalla cultura o lingua. Per facilitare la navigazione e l’individuazione, si usano brevi segmenti che si focalizzano su un numero limitato di concetti alla volta. E’ fondamentale comunicare solo i dettagli essenziali in modo che gli utenti possano concentrarsi sulle proprie attività. A volte l’interfaccia utente più efficace non contiene alcun testo. Inoltre, è bene evitare la

terminologia specifica del settore o i nomi inventati per le funzionalità dell’interfaccia utente. Se una frase descrive un obiettivo e l’azione necessaria per raggiungerlo, è opportuno iniziare la frase con l’obiettivo.

Per descrivere il comportamento dell’app, si utilizza il tempo presente, non il futuro. Se è necessario scrivere al passato o al futuro, è raccomandato usare forme verbali semplici. Per quanto riguarda i numeri, per facilitare la lettura, vanno scritti in cifre. La punteggiatura va evitata dove non è necessaria, per aiutare i lettori a scansionare il testo a colpo d’occhio. Si evita di usare punti su frasi singole in questi elementi dell’interfaccia utente:

- Etichette;
- Testo al passaggio del mouse;
- Elenchi puntati;
- Testo del corpo della finestra di dialogo.

Frasi più lunghe o complesse possono usare i punti se farlo si adatta meglio al contesto. Per rivolgersi agli utenti si usa la prima o seconda persona, in

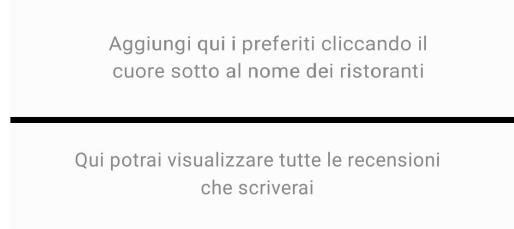


Figura 2.12: Testi al tempo presente e futuro.

base a quale sia adatta e più chiara per la situazione:

- Seconda persona, ”tu” o ”tuo”: questo stile di conversazione è appropriato nella maggior parte delle situazioni; come se l’interfaccia utente parlasse direttamente all’utente;
- Prima persona, ”io” o ”mio”: in alcuni casi, questa forma di indirizzo sottolinea la proprietà dell’utente di contenuti o azioni;

Per evitare di confondere l’utente, si evita di usare ”io” o ”mio” e ”tu” o ”tuo” nella stessa frase.

# Capitolo 3

## Novità del material design 3

### 3.1 Accessibilità

Al giorno d'oggi la tecnologia aiuta incredibilmente le vite di persone con disabilità, rendendole più indipendenti grazie a certi strumenti tra cui tastiere o lettori di schermo. Si può rendere l'interfaccia utente più semplice e accessibile grazie a:

- elementi chiari e visibili
- contrasto elevato
- una gerarchia di importanza chiara

#### 3.1.1 Gerarchia

Azioni, informazioni e pulsanti importanti dovrebbero stare nel lato superiore o inferiore dello schermo. Elementi di importanza simile invece devono stare vicini. E' importante seguire convenzioni o regole logiche come l'ordine di lettura (da sinistra a destra e dall'alto al basso nel nostro caso) per stabilire e capire l'ordine di gerarchia che si sta formando.

#### 3.1.2 Touch targets

I touch targets sono parti dello schermo che rispondono ad eventi di tocco, vengono consigliate delle misure adatte a renderli facili da premere. Nella nostra applicazione i touch targets automaticamente implementati da compose, come switches o IconButton, sono già creati con una grandezza minima, mentre i touch targets custom gli rendiamo conformi allo standard con il Modifier

### **3.1.3 Pointer Targets**

I pointer targets sono l'equivalente dei touch targets ma che rispondono a puntatori che tracciano il movimento come mouse o penne

### **3.1.4 Testo**

Uno dei modi migliori per aumentare l'accessibilità è l'aggiunta di didascalie, etichette e testo alternativo, in quanto sono facilmente leggibili da lettori di schermo, i quali possono avere delle difficoltà a "leggere" un'immagine.

Il testo alternativo aiuta a tradurre un elemento visivo, come un'immagine, in un elemento testuale.

### **3.1.5 Contrasto**

Per rendere gli elementi più accessibili nel proprio tema è consigliato di avere un rapporto 3:1 per testo grande rispetto al background, e di 4.5:1 per il testo piccolo.

### **3.1.6 Best practice**

La pratica migliore consiste però nell'utilizzo di elementi nativi, in quanto sono garantiti ad avere un certo livello di accessibilità, mentre elementi custom possono richiedere del testing per verificarne l'accessibilità

## **3.2 Personalizzazione**

Nel material design 3 sono stati aggiunti i colori dinamici, i quali permettono ad un'app di mantere un suo stile e accessibilità ma allo stesso tempo permettere all'applicazione di adattarsi al tema dell'utente. Questo è possibile grazie alla definizione di 5 ruoli principali per i colori:

- Primary
- Secondary
- Tertiary
- Neutral
- Neutral Variant

Inoltre google ha creato dei tools che permettono di generare tutti i colori di un tema dando un input uno o più colori Il tema dell'app è stato appunto generato grazie al tool di google

### 3.3 Design tokens

Un token salva delle informazioni: un colore, un certo stile o un certo componente, per permetterne l'utilizzo in più prodotti. Oltre ad essere molto utili per il design, i tokens permettono di condividere i design attraverso un formato chiamato [Design System Package](#), che permette di salvare assets e json che descrivono un determinato design. Al momento sono disponibili solo i design per colore e tipografia, con il supporto di tokens per componenti e misure in sviluppo. Si può scaricare il dsp del material design da [qua](#), modificarlo con dei tools, e poi esportarli per android.

Nella nostra applicazione non ne abbiamo fatto uso in quanto sono ancora in sviluppo, e i loro utilizzi non ci sono utili, come creare nuovi prodotti che usano lo stesso stile.

### 3.4 Interaction states

Gli interaction states sono indicatori visivi utilizzati per comunicare lo stato di un componente o di un elemento interattivo. Si dividono in livelli che si presentano come una copertura semitrasparente su un elemento che rende quest'ultimo più o meno opaco, mantenendo lo stesso colore del contenuto. Ciò non accade ai componenti che utilizzano gli stati "attivo" o "selezionato" che, invece, cambiano il contenitore e il colore del contenuto . Per impostazione predefinita, il colore del livello di stato di un componente deriva dal suo contenuto, dal colore di un'icona o dal testo dell'etichetta se non è presente alcuna icona. Un livello può essere applicato a un intero elemento o in una forma circolare e può essere applicato solo un livello alla volta.

### 3.5 Colori

Una delle principali novità introdotte dal Material Design 3 è il nuovo sistema dei colori, in particolare il sistema di colori *dinamici* (disponibile solo in Android 12).

### 3.5.1 Novità nel sistema di colori

Mentre nel Material Design 2 la palette di colori era costituita da 12 valori, con due colori principali, nel Material Design 3 il numero di tinte è più che raddoppiato. Infatti una palette (in Jetpack Compose rappresentata dalla classe `ColorScheme`) è costituita da ben 25 elementi, suddivisi come segue:

- *accent colors*: sono i tre colori principali dell'app e ne definiscono il tema
- *container colors*: sono delle varianti degli accent colors, da utilizzare per le superfici
- *background* e *surface*: colori neutri, generalmente bianco o nero, da utilizzare per lo sfondo dell'app
- *colori neutrali*: ulteriori colori per elementi di contorno o di scarsa importanza
- *colori per gli errori*: vengono usati per i messaggi d'errore
- *colori on[...]*: garantiscono un contrasto adeguato tra il testo e il colore della superficie



Figura 3.1: Schema di colori della nostra app

### 3.5.2 Ottenere uno schema di colori

Per ottenere uno schema di colori è messo a disposizione degli sviluppatori il [Material Theme Builder](#), uno strumento che permette di scegliere gli *accent colors* desiderati e che genera una paletta che rispetta le linee guida del Material Design 3. La paletta può essere esportata, sotto forma di codice, e implementata nell'app.

L'utilizzo di questo strumento garantisce una certa armonia nei colori e un adeguato contrasto tra i colori di sfondo e i colori per il testo, assicurando la leggibilità sia nel tema chiaro che nel tema scuro.

Per lo schema di colori della nostra app, abbiamo selezionato come *primary color* il colore rosso scuro caratteristico dell'Università di Padova, lasciando che il sistema generasse automaticamente tutti gli altri colori.

### 3.5.3 Gerarchia

All'interno dello schema di colori esiste una gerarchia, con lo scopo di guidare l'utente nell'interfaccia grafica dell'applicazione.

Il colore primario deve essere utilizzato per gli elementi più importanti, come pulsanti, o per indicare che un certo elemento di UI è selezionato. In altre parole, il colore primario viene utilizzato per attirare l'attenzione dell'utente e suggerirgli le azioni da compiere.

Il colore secondario viene utilizzato sulle componenti di priorità minore, come la barra di navigazione e altri elementi per svolgere azioni di bassa priorità.

Il colore terziario viene utilizzato per bilanciare i primi due in tutte le componenti che non rientrano nelle categorie descritte precedentemente.

### 3.5.4 Colori dinamici

Se si dispone di Android 12, il sistema mette a disposizione una paletta di colori determinata dallo sfondo selezionato dall'utente. È possibile utilizzare questa paletta anche nelle app, così da ottenere una armonia di colori a livello di sistema e aumentare l'effetto di personalizzazione del dispositivo.

In Jetpack Compose si può ottenere la paletta dinamica con due funzioni dedicate, `dynamicLightColorScheme()` e `dynamicDarkColorScheme()`, rispettivamente per il tema chiaro e scuro.



Figura 3.2: Gerarchia di colori implementata nella nostra app

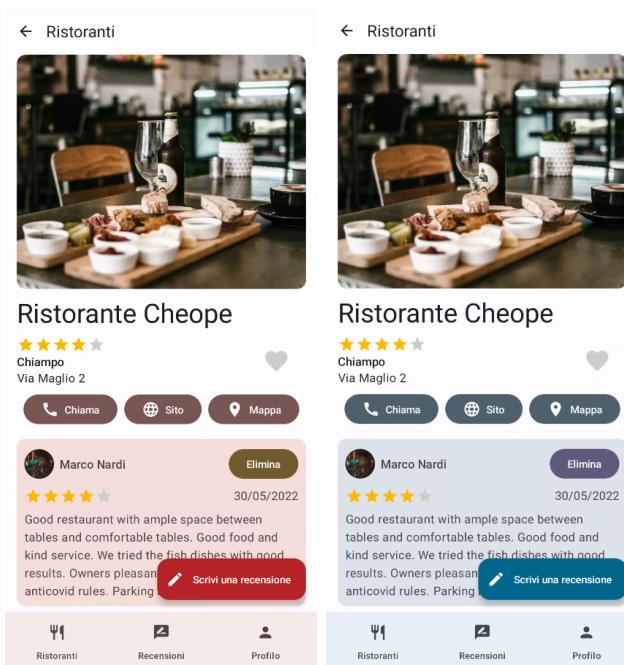


Figura 3.3: Colori di default e colori dinamici

## 3.6 Forme

Il Material Design 3 introduce un cambiamento ai gruppi di forme, sempre basati sulle dimensioni, ma con sette stili degli angoli, assegnati in base alla quantità di arrotondamento desiderata. Le forme con angoli squadrati sono chiamate "none", le forme leggermente arrotondate sono "extra-small", mentre le forme completamente arrotondate sono "full". Ciò consente di conferire maggior espressività all'interfaccia utente. Viene, inoltre, mantenuta la possibilità avere forme degli angoli simmetriche o asimmetriche, con angoli tagliati o arrotondati. Inoltre, è sempre modificare lo stile di un'intera categoria di forme.



Figura 3.4: Gruppi di forme in M3.

## 3.7 Iconografia material 3

Nel material design 3 le icone hanno ricevuto 3 nuovi stili: outlined, rounded, sharp, e sono scaricabili dal [catalogo di google fonts](#). Hanno un leggero livello di personalizzazione con 4 slider che ne cambiano le caratteristiche. Nel catalogo molte icone vecche sono state ridisegnate, dando ora un'impressione più moderna. Le linee guida nel design di icone nuove rimangono molto

simili a quelle del material design 2. Sono state aggiunte delle linee guida nell'utilizzo delle icone, e nelle caratteristiche da usare nella personalizzazione: per esempio nella decisione del weight (spessore dell'icona) è bene che non sia troppo fino (sconsigliati valori sotto al 200), e che non sia troppo grosso; soprattutto bisogna stare attenti ad usare lo stesso weight in modo consistente e non mischiare icone di weight diversi

Una caratteristica molto importante è la grandezza ottica, ci permette di cambiare la grandezza di un'icona da 24dp fino a 48dp aggiustando però il weight, infatti scalando un'icona scalerebbe anche il suo spessore il che può non essere desiderato a seconda del proprio stile. Per quanto riguarda l'accessibilità è importante che icone che eseguono delle azioni abbiano un etichetta che descriva cosa facciano, nella nostra applicazione le icone che eseguono azioni come scrivere una recensione, oppure cambiare tab, hanno tutte un testo che indica il loro ruolo.

## 3.8 Tipografia material 3

Nel material design 3 sono stati rinnovati i ruoli che avevano blocchi di testo, ora sono 5 e sono:

- display per le parti corti e importanti
- headline per le parti corti e importanti in schermi piccoli
- title per parti che sono mediamente importanti ma comunque corte
- body per le parti più lunghe
- label per il testo che accompagna componenti come pulsanti o didascalie

Ciò è stato fatto per rendere più semplice applicare le linee guida e rendere i ruoli più facili da capire. Sono stati introdotti i tokens per il testo per permettere facile modifica di valori come type scale o weight. Il resto delle linee guida invece rimane simile a quelle del material design 2

# Capitolo 4

## Material Reviews

La nostra applicazione si chiama Material reviews, e come suggerisce il nome si occupa di recensioni per ristoranti. L'idea è nata dalla necessità di creare un'applicazione complessa per poter mettere in pratica più linee guida possibili. Abbiamo usato Jetpack Compose [9] e Room[10] per l'implementazione, anche se alcune features utilizzate sono ancora in via di sviluppo.

### 4.1 Funzionamento e struttura

L'applicazione è divisa in 3 parti principali: i ristoranti, le proprie recensioni, il proprio profilo:



Figura 4.1: Home dell'applicazione

#### 4.1.1 Dettagli ristoranti

Nella parte Ristoranti si può scorrere la lista di ristoranti già presente nell'app, aggiungerne ai preferiti cliccando l'icona a forma di cuore, e accedere ai dettagli cliccando sulla card del ristorante. E' anche possibile filtrare per solo i ristoranti preferiti premendo lo switch in alto a destra. Tappando un ristorante apre la seguente schermata dei dettagli e recensioni:



Figura 4.2: Dettagli di un ristorante

Tappando il fab per aggiungere la recensione fa comparire un dialog per l'aggiunta, dove si possono premere il numero di stelle da dare e scrivere una recensione come mostrate in Figura 4.3



Figura 4.3: Dialog per l'aggiunta di una recensione

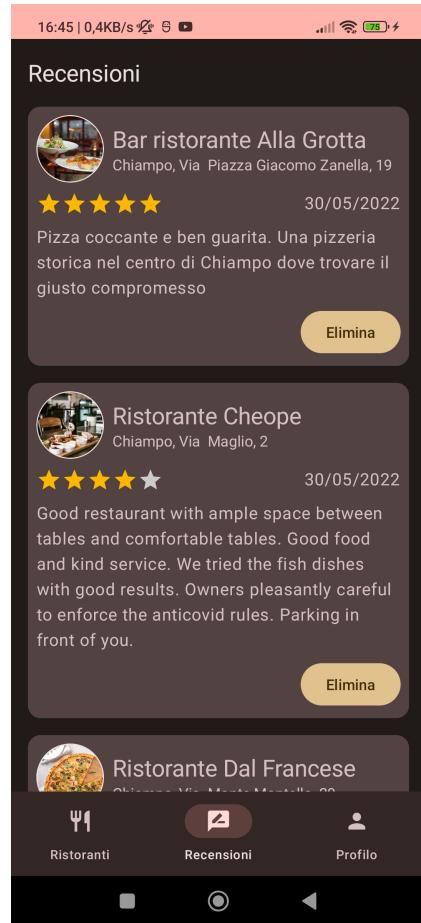


Figura 4.4: Schermata con le recensioni dell'utente

#### 4.1.2 Le mie recensioni

Nella parte recensioni è possibile vedere le proprie recensioni, cliccare sulla recensione porta alla schermata di dettagli del ristorante stesso. Si può anche rimuovere recensioni con il pulsante dedicato, che fa apparire un dialog di conferma.

### 4.1.3 Profilo

Nella parte del profilo l'utente può modificare le proprie informazioni e le impostazioni, anche se al momento l'unica impostazione disponibile è la scelta del tema.

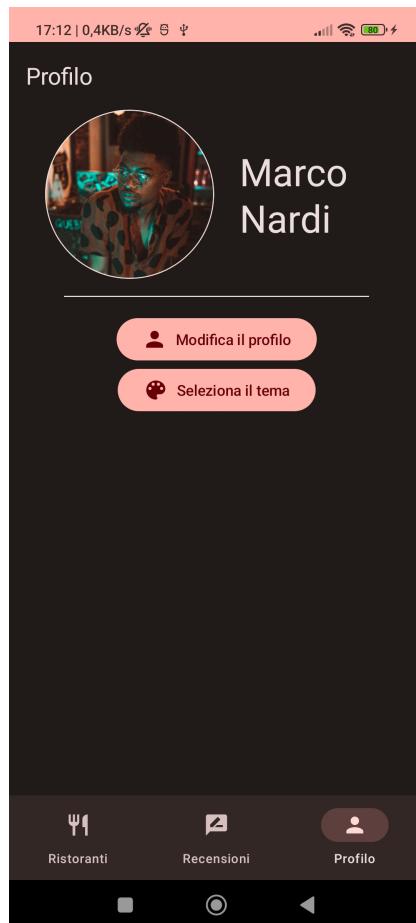


Figura 4.5: Schermata con il profilo dell'utente



Figura 4.6: Dialog per modificare il profilo

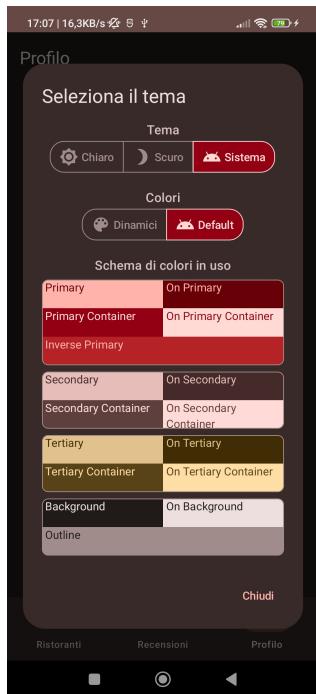


Figura 4.7: Dialog per modificare il tema

# Bibliografia

- [1] *Material Design*. URL: <https://material.io/design>.
- [2] *Kotlin for LaTeX Listings on github*. URL: <https://github.com/cansik/kotlin-latex-listing>.
- [3] *Material Design environment*. URL: <https://material.io/design/environment/surfaces.html#material-environment>.
- [4] *Material Design layout Principi*. URL: <https://material.io/design/layout/understanding-layout.html#principles>.
- [5] *Material Design layout organizzazione*. URL: <https://material.io/design/layout/understanding-layout.html#material-measurements>.
- [6] *Material Design layout grid*. URL: <https://material.io/design/layout/responsive-layout-grid.html#columns-gutters-and-margins>.
- [7] *Material Design layout containers*. URL: <https://material.io/design/layout/spacing-methods.html#containers-and-ratios>.
- [8] *Material Color Tool*. URL: <https://material.io/resources/color/?view.left=0&view.right=0&primary.color=6002ee>.
- [9] *Jetpack Compose*. URL: <https://developer.android.com/jetpack/compose/documentation>.
- [10] *Room Database*. URL: <https://developer.android.com/jetpack/androidx/releases/room>.