

# Material Design 3

Ferin Eli

Nardi Marco

Trincanato Marco

6 giugno 2022

# Indice

<b>1</b>	<b>Esempi e template</b>	<b>3</b>
1.1	Citazioni . . . . .	3
1.2	Blocco di codice . . . . .	3
<b>2</b>	<b>Principi del Material Design</b>	<b>5</b>
2.1	Ambiente . . . . .	5
2.1.1	Superfici . . . . .	5
2.1.2	Elevazione . . . . .	5
2.1.3	Luci e Ombre . . . . .	6
2.2	Layout . . . . .	6
2.2.1	Anatomia del layout . . . . .	6
2.2.2	Organizzazione dei contenuti . . . . .	7
2.2.3	Misure . . . . .	7
2.2.4	Responsive layout grid . . . . .	7
2.2.5	Metodi di spaziatura . . . . .	8
2.2.6	Contenitori e Aspect Ratios . . . . .	8
2.2.7	Densità . . . . .	8
2.3	Navigazione . . . . .	8
2.3.1	Navigazione laterale . . . . .	8
2.3.2	Navigazione in avanti . . . . .	9
2.3.3	Navigation transitions - da tenere se implementiamo nell'app . . . . .	9
2.4	Colori . . . . .	10
2.4.1	Temì . . . . .	11
2.5	Suono . . . . .	11
2.5.1	Applicare il suono all'interfaccia utente . . . . .	12
2.5.2	Tipi di suono . . . . .	13
2.5.3	Sound attributes . . . . .	15
2.5.4	Sound choreography . . . . .	17
2.5.5	Ottimizzazione del suono . . . . .	18
2.5.6	Formato dei file e ottimizzazione della memoria . . . . .	18

2.6	Tipografia . . . . .	19
2.6.1	Titoli . . . . .	19
2.6.2	Proprietà di un typeface . . . . .	20
2.6.3	Leggibilità . . . . .	20
2.6.4	Linguaggi . . . . .	21
2.7	Iconografia . . . . .	21
2.7.1	Icone di sistema . . . . .	21
2.7.2	Icone animate . . . . .	22
2.8	Forme . . . . .	22
2.8.1	Gruppi di forme . . . . .	23
2.8.2	Uso efficace delle forme . . . . .	24
2.8.3	Collegare superfici attraverso le forme . . . . .	28
2.9	Interaction . . . . .	28
2.9.1	Gesture . . . . .	28
2.9.2	Selezione . . . . .	29
2.9.3	Stato . . . . .	30
2.9.4	Tipi di stato . . . . .	30
2.10	Motion . . . . .	31
2.10.1	Transition patterns . . . . .	31
2.10.2	Velocità . . . . .	32
2.11	Comunicazione . . . . .	33
2.11.1	Confirmation and acknowledgement . . . . .	33
2.11.2	Visualizzazione dei dati . . . . .	34
2.11.3	Stati vuoti . . . . .	38
2.11.4	Aiuto e feedback . . . . .	39
2.11.5	Immagini . . . . .	40
2.11.6	Launch screen . . . . .	41
2.11.7	Onboarding . . . . .	41
2.11.8	Scrittura . . . . .	42

# 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 `cite{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`

# 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.

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 divisori

### 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

## 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 sepra 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'azione singola (*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*.

### 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 quest'ottica, il suono all'interno dell'app deve essere:

- *Informativo*: intuitivo, funzionale e comprensibile;
- *Pulito*: una rappresentazione autentica dell'identità e dell'estetica del marchio del prodotto;
- *Rassicurante*: dovrebbe creare un senso di comfort e sicurezza, invitando all'azione solo quando necessario.

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. Differenti tipi di suono possono essere utilizzati per creare un effetto particolare e più suoni possono presentarsi separatamente o insieme nell'interfaccia utente. Sebbene la musica e la voce facciano parte dell'insieme di suoni di un prodotto, questo report si concentra sul suono UX. L'interfaccia utente è composta principalmente da elementi visivi. Il suono può contribuire ad esprimere le informazioni, attirando il focus dell'utente, e fornire un altro modo per connettersi con esso.

Il **sound design** può essere utilizzato per:

- Associare un elemento dell'interfaccia utente a un suono specifico;
- Esprimere emozioni o personalità;
- Far capire qual è la struttura gerarchica, associando le interazioni con determinati suoni;
- Fornire feedback sensoriale.

La **musica** viene utilizzata principalmente per la narrazione ed esprime 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 musica può essere utilizzata in un'interfaccia utente per:

- Contesti emotivamente risonanti;
- Pubblicità;
- Fornire momenti di gioia;
- Raccontare una storia.

La **voce** e la sintesi vocale utilizzano entrambe il linguaggio parlato per comunicare informazioni dove il sound design e la musica non sono abbastanza. La voce può essere utilizzata in un'interfaccia utente per:

- Comunicare informazioni complesse;
- Fornire conversazione e dialogo;
- Fornire momenti di gioia;
- Migliorare il tono e la personalità.

### 2.5.1 Applicare il suono all'interfaccia utente

Il suono svolge il compito di dare espressione alle interazioni e rafforzare funzionalità specifiche. Il suono può fornire **feedback** o aggiungere **decorazioni** a un'esperienza utente quando applicato in momenti strategici.

I suoni sono collegati alle interazioni dell'utente con l'app sono chiamati *earcon* e rappresentano informazioni, azioni o eventi. Rafforzano il significato di un'interazione, l'estetica, l'emozione e la personalità di un prodotto. Gli earcon possono essere ispirati al mondo reale (e vengono chiamati *skeuomorphic*) oppure possono essere inventati appositamente per esprimere un concetto astratto.

I suoni utilizzati per la **decorazione** amplificano un momento espressivo o giocoso. Di solito sono usati per comunicare uno stato emotivo. Questo tipo di suoni dovrebbero essere usati con giudizio e riservati a momenti con alta risonanza emotiva.

## **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 effettivamente abbassare l'attenzione e il comfort dell'utente. Il suono, di solito, non è adatto per:

- UI che richiedono privacy o discrezione;
- Utenti che non hanno richiesto interruzioni;
- Azioni eseguite di frequente.

In qualsiasi contesto, il suono dovrebbe elevare l'esperienza visiva piuttosto che sminuirla.

## **2.5.2 Tipi di suono**

### **Hero sound**

Gli hero sound sono audio che evidenziano un momento importante, evocano uno stato emotivo o vogliono celebrare un certo tipo di eventi. Si verificano in interazioni fondamentali, come quelle che:

- Festeggiano un'azione significativa intrapresa dall'utente;
- Accolgono agli utenti in una nuova app o esperienza;
- Confermano un momento chiave dell'esperienza di un prodotto.

Gli hero sound si verificano raramente e, data la loro importanza, dovrebbero essere applicati in modo coerente.

### **Notifiche**

Il ruolo principale delle notifiche è aiutare a dirigere l'attenzione dell'utente. Esse offrono l'opportunità di dare un'identità sonora alle interazioni. Poiché si verificano più frequentemente in un'interfaccia utente rispetto ad altri suoni, dovrebbero essere più brevi degli hero sound e realizzate affinché possano essere riprodotte più volte. Le notifiche dovrebbero essere progettate per essere udibili da distanze diverse e in ambienti rumorosi. Il suono stesso può variare per attirare il focus, usando variazioni ritmiche, tonali e di frequenza. Si consiglia di fornire opzioni che consentano agli utenti di personalizzare le notifiche, con opzioni che vanno dal suono di base, non decorativo, al suono ricco e decorativo.

## Suonerie, sveglie e timer

Suonerie, sveglie e timer sono avvisi che spesso hanno uno stile giocoso. Possono essere personalizzati in base alle preferenze dell'utente e alla personalità del marchio.

## Suoni di sistema

I suoni di sistema si dividono in **suoni UX primari** e **suoni UX secondari**.

I **suoni UX primari** sono generati da un sistema operativo (o dispositivo) per fornire feedback agli utenti. Possono aggiungere suoni a qualsiasi interazione, come ad esempio:

- Navigazione nel menu;
- Conferma dell'azione diretta di un utente;
- Inserimento dati.

Dovrebbero essere in armonia con l'estetica del suono di un prodotto, pur rimanendo semplici e discreti. Questi suoni si verificano più frequentemente rispetto ad altri suoni nell'interfaccia utente. Per questo, dovrebbero essere adatti per essere riprodotti spesso, evitando che vengano percepiti come fastidiosi o ridondanti.

I **suoni UX secondari** vengono riprodotti meno frequentemente in un'interfaccia utente. Dovrebbero riflettere la personalità di un prodotto, ma sono utilizzati principalmente per scopi funzionali, per informare gli utenti di cambiamenti di stato o per indicare azioni poco frequenti.

## Suoni ambientali

Il suono ambientale è uno strato decorativo del suono che esprime emozioni comunicando contemporaneamente la personalità o il marchio di un prodotto. Può essere posizionato ovunque ci si aspetti un forte elemento sonoro, come ad esempio:

- Un flusso di start-up, per accogliere l'ascoltatore nell'esperienza;
- Un accompagnamento a una schermata iniziale, per esprimere un tono emotivo e la posizione dell'utente nell'interfaccia.

Il suono ambientale dovrebbe creare atmosfera e non distogliere l'utente dall'esecuzione di un'attività. Può anche fornire spunti, cambiando il tono della musica per indicare un cambiamento di stato. I suoni ambientali possono essere basati su musica, suoni ambientali o altri suoni che generano atmosfera.

### 2.5.3 Sound attributes

Le caratteristiche del suono possono essere regolate in modi diversi per creare un effetto specifico. Queste sono misurabili e, quindi, si può capire quanto sarà efficace l'effetto che si vuole attribuire ad un suono. La tecnica che raffigura queste caratteristiche è chiamata *visualizzazione del suono*. In un grafico, l'asse x rappresenta una caratteristica (come il tempo) e l'asse y rappresenta un'altra caratteristica (come l'ampiezza). In alcuni casi, una caratteristica viene rappresentata graficamente utilizzando un solo asse. In questo modo, si può visualizzare il suono in funzione del **tempo**, della **frequenza** (quanto è acuto o basso un suono) o del **timbro** (la qualità di un suono).

Per rappresentare il suono in funzione del **tempo**, si può visualizzare il tempo sull'asse x e l'ampiezza sull'asse y. Ciò rende evidente la nitidezza (o morbidezza) dei suoi picchi.

#### Timbro

Per rappresentare il suono in funzione del **timbro**, si può visualizzare l'ampiezza sull'asse y e il timbro sull'asse x. Molti aspetti del timbro possono essere rappresentati visivamente ed è necessario scegliere quali aspetti sono importanti da visualizzare. Il timbro di un suono ne descrive la qualità e il carattere (indipendentemente dall'altezza o dal volume). Ci sono molte caratteristiche da considerare quando si studiano timbri diversi. Un aspetto del timbro che può essere adattato all'interfaccia è la luminosità (la quantità di suono ad alta frequenza):

- Un suono brillante ha più contenuti ad alta frequenza, dandogli una presenza più forte;
- Un suono silenzioso ha meno contenuti ad alta frequenza, rendendo il suo suono sottile e più silenzioso, specialmente in ambienti rumorosi;

Ogni tipo di suono è appropriato a contesti diversi. I timbri più luminosi si sentono più ricchi e giocosi. I timbri smorzati appaiono più pesanti e seri. Quando si sceglie il timbro di un prodotto, è necessario conoscere la tipologia di utente e il contesto in cui ogni suono viene riprodotto.

#### Tonalità

La tonalità si riferisce a due tipi di suoni:

- Suoni **tonali**, di natura più musicale, tra cui melodia, motivo e armonia;



- Suoni **atonali** (chiamati anche suono non armonici), che assomigliano a suoni o rumori quotidiani, non conformi alle composizioni musicali tradizionali.

I suoni possono essere progettati utilizzando elementi tonali, atonali o entrambi. I suoni **tonali** sono l'ideale per comunicare personalità, emozioni e cambiamenti di stato, mentre i suoni **atonali** supportano meglio le transizioni di movimento ed esprimono un senso di feedback tattile, di movimento o di skeuomorfismo.

Una *melodia* è un suono tonale che ha una breve successione ritmica di singole altezze. Melodie semplici e brevi possono essere ripetute in un'interfaccia utente che non richiede distrazioni.

Un *motivo* è una melodia che si ripete, rendendola uno schema riconoscibile. I motivi comuni possono comunicare in modo rapido ed efficace semplici segnali emotivi e informativi.

Il suono atonale *skeuomorphic* dovrebbe essere utilizzato quando un'interazione ha una forte associazione con il mondo reale, come un clic dell'otturatore della fotocamera.

## Dinamica

Nel sound design, la **dinamica** si riferisce a variazioni di volume su una scala micro (il volume cambia in un singolo suono) o una scala macro (il volume cambia in un insieme di suoni). La variazione dinamica consente ai suoni di sembrare naturali e realistici. I cambiamenti dinamici possono evidenziare momenti sonori di interesse, creare un senso di progressione attraverso una narrazione musicale e conferire al suono una qualità naturale.

## Inviluppo

L'**inviluppo** di un suono si riferisce all'andamento dell'ampiezza di un suono dal momento in cui viene generato a quando si estingue. Un inviluppo è composto da molte componenti tecniche, ma le due fasi basilari sono l'ascesa (*attacck*) e la caduta (*decay*) di un suono. L'*attacck* si riferisce alla velocità iniziale di un suono, prima che raggiunga l'ampiezza massima. Il *decay* si riferisce alla quantità e alla velocità con cui l'ampiezza di un suono diminuisce dopo il suo *attacck* (fino a quando la sua ampiezza è zero). Maggiore è il valore di *decay*, più lungo e lento sarà il suono.

## Riverbero e ritardo

Riverbero e ritardo sono effetti che possono aggiungere un senso di spazio e profondità. Per evitare che gli effetti sonori diventino una parte troppo prominente del suono di un earcon, dovrebbero essere usati con parsimonia.

Il **riverbero**, o effetto di riverbero, è un effetto applicato al suono che gli conferisce più presenza. Se applicato in modo lieve, si ha l'effetto di sentire un suono in un ambiente reale.

Simile al riverbero, il **ritardo** utilizza gli echi del suono originale mescolati con il suono stesso per creare un effetto decorativo. Il ritardo dovrebbe essere usato in modo mirato e con parsimonia.

### 2.5.4 Sound choreography

Il suono può essere coreografato per ottimizzare l'esperienza del prodotto e far capire all'utente le relazioni di gerarchia. Nella coreografia, infatti, ogni suono dovrebbe riflettere il suo livello di importanza nella gerarchia dell'interfaccia utente. I suoni dello stesso tipo (come gli hero sounds) dovrebbero condividere lo stesso livello di gerarchia. Inoltre, i suoni che sono più in alto nella gerarchia sono le rappresentazioni importanti di un marchio o di un prodotto. In un'esperienza utente, i suoni che si susseguono o precedono l'un l'altro dovrebbero avere caratteristiche correlate (come timbro, melodia o inviluppo). I suoni che condividono delle caratteristiche vengono unificati come *gruppo*.

Le tonalità aiutano a costruire relazioni armoniche tra le interazioni. I suoni che vengono riprodotti in stretta vicinanza l'uno all'altro dovrebbero utilizzare le stesse tonalità o complementari, a meno di casi specifici.

Per mostrare come gli stati sono correlati tra loro si possono usare i motivi. Ad esempio, il suono per uno stato "on" può essere correlato al suono per uno stato "off".

I suoni di interazione che si verificano regolarmente, come i suoni associati alla digitazione, allo scorrimento o alla navigazione, possono essere resi con piccole modifiche a un suono di base, come variazioni nel timbro del suono, per imitare la variazione dei suoni nel mondo reale.

## Missaggio

Il missaggio è l'arte di combinare diverse sorgenti sonore in un unico flusso audio. Implica la regolazione del volume, della frequenza, del posizionamento spaziale di ogni suono e altre caratteristiche per creare un suono ricco e coeso. Diverse sorgenti sonore possono essere mescolate per variare l'emo-

zione, lo scopo o il carattere del suono finale. I suoni UX dovrebbero essere bilanciati per accogliere altri suoni nell'interfaccia utente. Trattamenti che isolano, abbassano, mescolano e bilanciano alcuni suoni in momenti specifici possono aiutare a focalizzare correttamente l'attenzione dell'utente, in modo che l'intento dietro un suono si manifesti.

### **2.5.5 Ottimizzazione del suono**

Per ottimizzare il suono, il sound designer può ascoltare un suono utilizzando dispositivi del mondo reale in diversi ambienti. Ascoltando i suoni in condizioni reali (usando il software, l'hardware, il rumore ambientale, l'acustica e altri fattori di un ambiente), il suono può essere regolato meglio per essere riprodotto in una gamma più ampia di condizioni. È anche possibile apportare modifiche alle caratteristiche di un suono (come il timbro) utilizzando i seguenti processi: riscritture della composizione, riorchestrazione, variazioni melodiche, equalizzazione e altre modifiche.

#### **Equalizzazione**

L'equalizzazione (EQ) è un effetto che migliora o riduce frequenze specifiche. L'equalizzazione deve essere regolata in base ai dispositivi per cui è progettata l'applicazione.

#### **Rumorosità**

I suoni dovrebbero essere riprodotti ad un livello di volume coerente a seconda della loro posizione nella gerarchia del suono (determinata dal livello di priorità e dalla categoria di un suono). Ad esempio, il suono di una suoneria può essere più forte del suono del feedback dell'interfaccia utente, poiché ha una priorità maggiore nel momento in cui si verifica.

Quando si misura il volume tramite hardware specifico, bisogna tenere in considerazione il "volume percepito" (misurato in decibel ponderati A o dB(A)), piuttosto che basarsi esclusivamente sul livello del misuratore di picco diretto. Inoltre, gli aumenti del livello del volume dovrebbero utilizzare una scala logaritmica, piuttosto che lineare, per riflettere il modo in cui le persone sentono il suono.

### **2.5.6 Formato dei file e ottimizzazione della memoria**

La riproduzione del file audio finale può variare a seconda delle limitazioni hardware e software di un dispositivo. Per ridurre le dimensioni del file (con un peggioramento minimo della qualità):

- Applicare la compressione con perdita (come mp3 o ogg) fino a quando non si sentono gli artefatti;
- Ridurre il numero di bit e la frequenza di campionamento finché non si sentono degli artefatti;
- Eliminare qualsiasi silenzio non necessario all'inizio o alla fine del file.

Il formato finale dell'audio dipende dall'implementazione e dalle restrizioni a livello di sistema. E' consigliabile provare a scegliere il formato migliore (più lossless) consentito dal sistema, specialmente per i suoni chiave dell'esperienza utente.

## 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 Titoli

Per i titoli ci sono 6 scale e sono il testo più grande a schermo in quanto importanti. Per i titoli si possono usare font meno convenzionali in quanto possono avere dettagli che catturano l'occhio dell'utente.

#### Sottotitoli

Sottotitoli sono generalmente più piccoli dei titoli in quanto hanno meno importanza, ma rappresentano comunque blocchi di testo di piccola lunghezza. Bisogna stare attenti a non usare font troppo espressivi dato che maggior parte delle volte typefaces come Serif o sans funzionano bene. Hanno 2 scale disponibili.

#### Corpo di testo

Per la parte del corpo di testo sono disponibili 2 scale e sono usate per blocchi di testo lunghi, non bisogna usare font particolari tenendo il tutto più semplice possibile per facilitare la lettura

## **Didascalie e overline**

Didascalie e overline sono i type scale più piccoli. Didascalie spesso servono per annotare degli elementi mentre overlines a introdurre dei titoli. Come il corpo di testo devono avere dei font molto semplici

## **Testo per pulsanti**

Il testo per pulsanti viene usato per indicare azioni, è consigliato avere tutto in maiuscolo e con font sans serif

## **2.6.2 Proprietà di un typeface**

Questa sezione comprende le proprietà che caratterizzano un typeface o permettono di modificarne le misure

### **Baseline**

La baseline è la linea invisibile su cui stanno le lettere e nel material design si allinea sulla griglia da 4dp. E' importante per misure come:

- Cap Height, la misura in altezza delle lettere piatte maiuscole (come la M)
- X Height, la misura in altezza della lettera x minuscola, fornisce un'indicazione di leggibilità di un determinato typeface
- Ascenders e Descenders, sono linee appartenenti a lettere come y e d che in caso di distanza tra linee troppo piccola rischiano di interferire con altre righe

### **Peso**

Il peso rappresenta lo spessore del tratto di un typeface, ogni typeface ne ha un determinato numero che spesso sta tra 1 (Light) e 4 (Bold)

## **2.6.3 Leggibilità**

### **Letter spacing**

Letter spacing (detto anche tracking) 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. Per esempio un titolo avrà degli spazi più piccoli

in quanto il testo è già grande. Per misure più piccole, è consigliato tenere uno spazio abbastanza grande in quanto migliora la leggibilità.

### **Tabular figures**

Le Tabular figures sono cifre che occupano sempre lo stesso spazio (mono-spaced), ciò aiuta a leggere cifre lunghe quando sono vicine ad altre, come per esempio in colonna in una tabella.

### **Line length e height**

Ci sono delle misure consigliate sia per la lunghezza in caratteri di una riga che della sua altezza (distanza dalla baseline). La line length consigliata dipende dal linguaggio utilizzato: in inglese è tra 40 e 60 caratteri, ma altri linguaggi hanno in media parole più lunghe o più corte. La line height è proporzionale al type size e controlla l'altezza della riga dalla baseline

## **2.6.4 Linguaggi**

Il material design supporta diversi linguaggi compresi i linguaggi classificati come *dense* (per esempio il cinese), oppure linguaggi *tall* come l'arabo. A seconda del linguaggio cambiano tutte le caratteristiche descritte precedentemente.

## **2.7 Iconografia**

Il material design fornisce strumenti, griglie e specifiche per creare l'icona per il proprio brand o app. Sono presenti templates e molti esempi, il tema comune è lo stesso: semplicità e complicare il design solo in certi casi.

### **2.7.1 Icone di sistema**

Le icone di sistema sono create per essere semplici e moderne, sono ridotte alla forma minima, esprimendo solo le caratteristiche più importanti; le forme sono geometriche e consistenti, e consistenti. Questa semplicità permette alle icone di rimanere leggibili e distinguibili anche a dimensioni piccole. Le icone di sistema sono 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, questo è una misura per prevenire sovrapposizioni o oscuramenti in layout densi o nel caso appaiano altri elementi a schermo.

## Linee guida per icone di sistema

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. 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. Nel caso l'icona non sia attiva è consigliato ridurre la sua opacità

### 2.7.2 Icone animate

Le animazioni delle icone rappresentano l'azione che devono compiere aggiungendo dettagli alla propria interfaccia utente

#### 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.

Si può decidere di modificare la forma dei componenti di un'applicazione per vari motivi. Le forme con un design unico (*forme uniche*) differiscono dalle forme che le circondano, facendole risaltare. Infatti, si distinguono dagli

altri componenti, dal contenuto che li circonda e dall'interfaccia utente nel suo insieme. Questa caratteristica aiuta a dirigere l'attenzione dell'utente.

Inoltre, si può comunicare il cambiamento di stato di una componente utilizzando una forma diversa rispetto alle altre componenti di quel gruppo.

Ancora, per esprimere il linguaggio visivo di un **marchio** in un'app, si usa la forma insieme ad altre personalizzazioni, come colore e tipografia, in modo coerente. Lo stile di una famiglia di forme, come le forme organiche o geometriche, dovrebbe corrispondere al marchio. Queste piccole modifiche alla forma, applicate strategicamente, contribuiscono all'impressione generale che un brand dà all'utente.

In breve, la forma dà modo agli utenti di riconoscere gli elementi e identificare parti diverse delle superfici. Le forme sono utilizzate per riflettere uno scopo o un significato specifico. Il testo o le icone possono aiutare a rafforzare questo significato quando la forma di una sola superficie potrebbe essere ambigua.

### 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;
- Componenti medi;
- Componenti grandi.

I seguenti componenti permettono di personalizzare i loro angoli:

- Componenti piccoli: Button, Chip, Foglio inferiore espandibile (compreso), Floating action button esteso, Floating action button, campo di testo, Snackbar, tooltip;
- Componenti medi: Card, Dialog, Image list Item, menu;
- Componenti di grandi dimensioni: Sfondo, Tabella dati, Foglio inferiore espandibile, Nav Drawer, Foglio laterale, Foglio inferiore standard.

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

- Forma di un'intera famiglia;



- Dimensioni della forma;
- Simmetria.

I componenti possono avere uno tra questi due stili per gli angoli:

- Angolo arrotondato;
- Angolo tagliato.

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.

La *dimensione della forma* degli angoli può essere determinata utilizzando un valore **assoluto** o **percentuale**.

La dimensione *assoluta* consiste in un valore specifico (ad esempio 2dp). Quando un raggio di un angolo o una lunghezza di taglio ha una dimensione assoluta, questa rimane la stessa indipendentemente dall'altezza del componente. Ad esempio, se un componente cambia dinamicamente la sua altezza, la forma dell'angolo mantiene lo stesso raggio dell'angolo o la stessa dimensione della lunghezza di taglio.

I piccoli componenti possono impostare la dimensione della forma dell'angolo utilizzando una *percentuale* (valutata rispetto all'altezza totale del componente). Ciò significa che la forma dell'angolo cambierà al variare dell'altezza del componente.

Ai componenti si possono applicare forme angolari in modo **simmetrico** o **asimmetrico**. I componenti simmetrici applicano gli stessi valori a tutti gli angoli, mentre i componenti asimmetrici possono avere angoli con valori diversi l'uno dall'altro.

Per un componente dalla forma *simmetrica*, bisogna specificare una singola famiglia di forme e un unico valore di dimensione che verrà utilizzato da tutti gli angoli.

Per un componente dalla forma *asimmetrica*, bisogna specificare la famiglia di forme e i valori di dimensione di ogni angolo. I valori della famiglia e della dimensione della forma vengono applicati in senso orario, a partire dall'angolo in alto a sinistra.

## 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. Il grado con cui i componenti possono cambiare forma dipende da:

- facilità di identificazione di un componente, da parte dell'utente, in base alla forma;
- requisiti ergonomici (come una dimensione minima del componente cliccabile).

Lo *strumento di personalizzazione della forma* può essere utilizzato per generare forme diverse per vari singoli componenti. Una tabella indica gli intervalli di valori consigliati per ciascun componente.

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 punto da toccare, se non lo è.

## Angoli

Bisogna evitare di utilizzare valori percentuali per i componenti che modificano dinamicamente la propria altezza, altrimenti la forma potrebbe modificarsi in modi inaspettati.

Gli angoli ancorati ai bordi dello schermo non possono essere personalizzati, altrimenti si creerebbero spazi vuoti che mostrano il contenuto dietro un componente.

## Marchio

Quando si esprime il marchio con la forma, si evitano forme che:

- Implicano interattività;
- Esprimono stati con poca accuratezza;
- Interferiscono con l'usabilità.

La combinazione di stili diversi e forme completamente diverse può rendere difficile l'associazione di forme particolari a un marchio. Quando si applica la forma in un'interfaccia utente, si identifica la forma distintiva del marchio, ad esempio una forma organica o geometrica che riflette gli attributi del marchio. Sulla base di questa forma, viene poi sviluppata una serie di forme simili e correlate 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.

### **Override delle forme**

Un componente può richiedere una forma diversa da quella definita da un gruppo. In questo caso, si può utilizzare una sostituzione della sua famiglia di forme, dimensioni o entrambe. Le sostituzioni si usano per branding, usabilità, layout, gerarchia o altri fattori. Quando a un componente specifico viene assegnato un valore, questo sovrascrive il valore ereditato dal gruppo di forme.

### **Componenti a schermo intero**

Non utilizzare forme sugli angoli dei componenti a schermo intero o espansi. Le forme rettangolari offrono più spazio per il contenuto scorrevole. Inoltre, impediscono che il contenuto dietro i componenti a schermo intero venga visualizzato attraverso gli spazi tra la forma e il bordo dello 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:

- Testo nel componente tagliato a causa della sua forma;
- Modifica di altezza e forma del componente, se le dimensioni sono definite come percentuali;
- Modifiche al contenuto.

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. Inoltre, si usa il morphing per rimanere coerenti con un linguaggio visivo al variare delle dimensioni della superficie o per indicare quando gli elementi vengono aggiunti a un set selezionato. Il morphing può causare modifiche a:

- Risalto di un elemento;
- Ergonomia degli elementi;
- Significato di una forma particolare;
- Espressione di un marchio;
- Relazioni tra elementi.

Quando una superficie cambia dimensione, la sua forma può mantenere la posizione o le dimensioni originali, può allungarsi o ridursi. Una forma può anche trasformarsi tra due forme diverse. Una forma può allungarsi o restringersi in risposta ai cambiamenti nelle dimensioni della superficie.

Il morphing di una forma dovrebbe preservare le proporzioni delle forme distintive, per evitare la distorsione. Inoltre, tutto il contenuto su una superficie dovrebbe essere visibile mentre la superficie si trasforma, senza ritagliare il contenuto.

Per impostazione predefinita, le superfici sono rettangolari. Possono trasformarsi da forme rettangolari in altre forme e viceversa. Ad esempio, un pulsante di azione mobile rotondo può diventare un menu rettangolare e quindi tornare alla sua forma rotonda originale. Le forme rettangolari massimizzano lo spazio disponibile per il contenuto scorrevole e si fondono con altre superfici rettangolari. Di conseguenza, il contenuto riceve enfasi.

Quando si utilizza la forma rettangolare predefinita, è possibile aggiungere un segnale (come un'icona di compressione o espansione) per indicare come trasformare la superficie.

Per quanto riguarda i componenti che trasformano la loro forma, si dovrebbe considerare in che modo tali trasformazioni possono influenzare l'identificabilità del componente. I componenti sagomati che si espandono per riempire l'intero schermo non devono utilizzare la sagomatura quando diventano a schermo intero. I componenti possono utilizzare i cambiamenti di forma per indicare i cambiamenti di stato. Ad esempio, lo stato selezionato di una card può utilizzare una forma unica che la distingue dalle card non selezionate. Le forme utilizzate per indicare lo stato devono essere distinte dalla forma tipica del componente.

### 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 Interaction

### 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 gesture da eseguire e il suo effetto nello schermo

#### Gesture di navigazione

Le gesture 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 gestures 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 menù).

#### 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 perchè 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. Così, si può ridurre l'incertezza su un'azione che un utente ha intrapreso o sta per intraprendere e si evita di commettere errori.

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. Ad esempio, se un segno di spunta indica che un'immagine è stata selezionata, non sono necessarie ulteriori conferme. La richiesta di conferma viene resa 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. Un acknowledgement notifica all'utente le azioni di sistema che si verificano in background. 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. I criteri per la scelta del giusto componente includono:

- Livello di urgenza;
- Inclusione di un'azione per correggere un problema;
- Durata sullo schermo (transitoria, ignorabile o entrambi).

Gli acknowledgement temporanei indicano che il componente sparirà da solo entro pochi secondi dalla sua comparsa. Gli acknowledgement ignorabili possono essere eliminabili istantaneamente dall'utente. Tipi di componenti che si possono usare:

- *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;
- *Empty state*. Quando un'interfaccia utente è disponibile solo online e il contenuto non è stato caricato o sincronizzato, utilizzare uno stato vuoto. L'utente dovrebbe essere in grado di interagire con la maggior parte possibile del resto dell'app. È possibile presentare un collegamento per ricaricare il contenuto.

## 2.11.2 Visualizzazione dei dati

La visualizzazione dei dati rappresenta le informazioni in forma grafica. E' una forma di comunicazione che ritrae informazioni dense e complesse attraverso elementi visivi, per semplificare il confronto dei dati. La visualizzazione dei dati può esprimere dati di diversi tipi e dimensioni: da pochi data point a grandi set di 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.

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.

### Stile

La visualizzazione dei dati utilizza stili e forme personalizzate per semplificare la comprensione dei dati a colpo d'occhio, in modi che si adattano alle esigenze e al contesto dell'utente. I grafici possono trarre vantaggio dalla personalizzazione di quanto segue:

- Elementi grafici;
- Tipografia;
- Iconografia;
- Assi ed etichette;
- Leggende e annotazioni.

#### *Elementi e grafici.*

La codifica visiva è il processo di traduzione dei dati in forma visiva. È possibile applicare attributi grafici univoci sia a dati quantitativi (come temperatura, prezzo o velocità) sia a dati qualitativi (come categorie, aromi

o espressioni). Questi attributi includono una grande varietà di elementi, tra cui forma, colore, volume, lunghezza, angolo, posizione, direzione e densità.

È possibile applicare più trattamenti visivi a più di un aspetto di un elemento dei dati. Ad esempio, il colore di una barra può rappresentare una categoria, mentre la lunghezza di una barra può esprimere un valore (come la dimensione della popolazione).

I grafici possono utilizzare le forme per visualizzare i dati in diversi modi. Una forma può essere disegnata come giocosa e curvilinea o precisa e ad alta fedeltà, o in modi ibridi tra questi due estremi. I dati destinati all'esplorazione ravvicinata dovrebbero essere rappresentati da forme adatte all'interazione (in termini di dimensioni del target touch e relative affordances). Mentre i dati che hanno lo scopo di esprimere un'idea o una tendenza generale possono utilizzare forme con meno dettagli.

Le **linee** del grafico possono esprimere delle qualità dei dati, come gerarchia, evidenziazioni e confronti. Gli stili di linea possono essere stilizzati in diversi modi, ad esempio utilizzando trattini o opacità diverse

Il **colore** può essere utilizzato per differenziare i dati del grafico in quattro modi principali:

- Distinguere le categorie l'una dall'altra;
- Rappresentare la quantità;
- Evidenziare dati specifici;
- Esprimere significato.

Per soddisfare gli utenti che non vedono differenze di colore, si possono utilizzare altri metodi per accentuare i dati, come l'ombreggiatura, la forma o la trama ad alto contrasto. L'applicazione di etichette di testo ai dati aiuta anche a chiarirne il significato, eliminando la necessità di una legenda.

#### *Tipografia*

Il testo può essere utilizzato per etichettare diversi elementi del grafico, tra cui:

- Titoli dei grafici;
- Etichette dei dati;
- Etichette dell'asse;
- Legenda.

Il testo con il livello di gerarchia più alto è solitamente il titolo del grafico. Le intestazioni e i diversi pesi dei caratteri possono comunicare quale contenuto è più (o meno) importante di altri contenuti nella gerarchia. Tuttavia, questi trattamenti dovrebbero essere usati con parsimonia, con un numero limitato di stili tipografici.

### *Iconografia*

L'iconografia può rappresentare diversi tipi di dati in un grafico e migliorarne l'usabilità complessiva. L'iconografia può essere utilizzata per:

- Dati categoriali per differenziare gruppi o categorie;
- Controlli e azioni dell'interfaccia utente, come filtro, zoom, salvataggio e download;
- Stati, come errori, assenza di dati, stati completati e pericolo.

Quando si posizionano le icone in un grafico, si consiglia di utilizzare simboli universalmente riconoscibili, in particolare quando si rappresentano azioni o stati, come: salvataggio, download, completamento, errore e pericolo.

## **Comportamento**

I grafici forniscono modelli di interazione che danno agli utenti il controllo sui dati visualizzati. Questi modelli consentono agli utenti di concentrarsi su valori o intervalli specifici di un grafico. I seguenti modelli di interazione, stili ed effetti consigliati (come il feedback tattile) possono migliorare la comprensione da parte dell'utente dei dati del grafico:

- La *divulgazione progressiva* fornisce un percorso chiaro per svelare i dettagli, accessibile su richiesta;
- La *manipolazione diretta* consente agli utenti di agire direttamente sugli elementi dell'interfaccia utente, riducendo al minimo il numero di azioni necessarie su uno schermo, tra cui: zoom e panoramica, impaginazione e controlli dei dati;
- La *modifica della prospettiva* consente a un progetto di funzionare per utenti e tipi di dati diversi, come controlli dati e movimento.

Lo **zoom** e la **panoramica** sono interazioni popolari tra i grafici che influiscono sulla precisione con cui gli utenti possono studiare i dati ed esplorare l'interfaccia utente del grafico. Lo *zoom* cambia se l'interfaccia utente viene mostrata da più vicino o da più lontano. Il tipo di dispositivo determina come viene eseguito lo zoom e, quando non è l'azione principale, può

essere implementato facendo clic e trascinando (sul desktop) o toccando due volte (sul dispositivo mobile). La *panoramica* consente all'utente di esplorare l'interfaccia utente che si espande oltre lo schermo. Dovrebbe essere vincolata in modi che abbiano senso per i dati visualizzati. Ad esempio, se una dimensione di un grafico è più importante di un'altra, la direzione della panoramica può essere limitata solo a quella direzione. L'atto della panoramica è spesso associato allo zoom. Sui dispositivi mobili, la panoramica viene spesso implementata tramite gesti, come lo scorrimento con un dito.

Il **movimento** può migliorare e rafforzare la relazione tra i dati e il modo in cui gli utenti interagiscono con essi. Il movimento dovrebbe essere usato intenzionalmente (non in modo decorativo), esprimendo la connessione tra stati e spazi diversi, dovrebbe risultare logico, fluido e reattivo, senza ostacolare il percorso dell'utente.

Nei **grafici vuoti** si possono visualizzare contenuti che suggeriscono cosa aspettarsi quando i dati sono disponibili. Ove applicabile, le animazioni possono essere mostrate per fornire gioia e intrattenimento.

## Dashboards

La visualizzazione dei dati può essere visualizzata su una serie di più grafici, in interfacce utente denominate dashboard. Grafici multipli e separati a volte possono comunicare meglio una storia, rispetto ad un grafico complesso.

Lo scopo di una dashboard dovrebbe riflettersi nel layout, nello stile e nei modelli di interazione. Il suo design dovrebbe adattarsi al modo in cui verrà utilizzato, 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 dashboard:

- *Dashboard analitiche*

Le dashboard analitiche consentono agli utenti di esplorare più set di dati e scoprire le tendenze. Di solito queste dashboard 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*

Le Operations dashboard sono progettati per rispondere a una serie predefinita di domande. In genere vengono utilizzati per completare attività relative al monitoraggio. Nella maggior parte dei casi, questi tipi di dashboard 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*

Le Presentation dashboards forniscono un'istantanea curata su un argomento di interesse. Queste dashboard in genere 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. Gli stati vuoti possono visualizzare un'ampia varietà di contenuti. Ad esempio, possono includere un elenco senza voci o una ricerca che non restituisce risultati. Sebbene questi stati non siano tipici, dovrebbero essere progettati per evitare confusione.

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.

#### Alternative

Per aiutare gli utenti che non conoscono un'app o una sezione, 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. Raccomandazioni:

- I contenuti iniziali sono ideali per le app che archiviano contenuti (come libri o musica) o creano contenuti basati su modelli (come note o documenti);
- Usa contenuti che abbiano un ampio appeal e mostrino le caratteristiche principali;
- Offri agli utenti la possibilità di eliminare e sostituire il contenuto iniziale;
- Se possibile, fornisci contenuti personalizzati.

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

- Consenti di ignorare o saltare questo contenuto;
- Sii breve;
- Mantieni il contenuto contestuale allo schermo. Questo non dovrebbe essere un luogo in cui eseguire l'onboarding dell'utente nell'intera app.

## 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". È più facile trovare il contenuto della Guida se è sia nel drawer di navigazione che nella barra delle app o solo nella barra delle app. È meno facile trovare il contenuto della Guida quando è solo nel drawer di navigazione o nel menu di overflow.

Se l'app è complessa e dispone di un drawer, si inserisce un link alla "Guida" sia nel drawer di navigazione che nella barra dell'app.

Se l'app è complessa e non dispone di un drawer di navigazione, si inserisce un link alla "Guida" nella barra dell'app.



Se l'app non è complessa, inserisci "Aiuto" in un drawer di navigazione o solo nel menu di overflow.

"Aiuto" dovrebbe essere l'ultimo elemento nel drawer di navigazione, con "Invia feedback" direttamente sopra di esso. Inoltre, ogni volta che viene visualizzato "Esci" in un drawer di navigazione, questo dovrebbe essere l'ultimo elemento nell'elenco.

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. Le immagini devono essere selezionate in base alla loro capacità di esprimere il messaggio dell'app e riflettere lo stile del prodotto. Indipendentemente dal fatto che si utilizzino delle fotografie generate dagli utenti, delle fotografie professionali o diversi stili di illustrazione, ogni modalità dovrebbe conferire un aspetto che rifletta il prodotto. Le immagini dovrebbero essere correlate tra loro condividendo una funzione, uno stile e un'intenzione comuni. Per garantire l'accessibilità, le immagini dovrebbero includere un testo alternativo (o una didascalia) che gli screen reader leggeranno agli utenti con disabilità visive.

Avere un punto focale iconico nelle immagini, è importante, poiché influisce sul modo in cui poi l'immagine viene tagliata per schermi con dimensioni diverse. Un punto focale può essere qualsiasi cosa, da una singola entità a una composizione complessiva. Per visualizzare al meglio immagini di diverse dimensioni e tipi, bisogna ridimensionare le immagini in modo appropriato per diversi display e piattaforme. È fondamentale testare le dimensioni di risoluzione appropriate per rapporti e dispositivi specifici, assicurandosi che le risorse non appaiano pixelate. Tuttavia, la risoluzione è anche il fattore più importante per la velocità di caricamento delle immagini. Per preservare la larghezza di banda della rete, sarà necessario mantenere bassa la risoluzione ove possibile.

## Miniature

Le miniature sono piccole immagini che rappresentano informazioni in spazi ristretti. In genere fungono da target di tocco che portano al contenuto principale, aparendo 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.

Le miniature possono essere ritagliate in diverse forme ed elementi, come testo o azioni, possono essere posizionati sopra di esse. Inoltre, la dimensione delle miniature riflette la loro importanza.

### 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. Le schermate di avvio non dovrebbero essere visualizzate se un'app è in esecuzione. Esistono due tipi di schermate di avvio:

- Schermate che mostrano un'anteprima non interattiva dell'interfaccia utente effettiva dell'app, chiamate interfacce segnaposto. Questa schermata di avvio è appropriata sia per l'avvio di app che per le transizioni di attività all'interno di un'app. All'avvio, gli elementi strutturali principali, come la barra di stato, la barra dell'app e il foglio inferiore, vengono visualizzati senza contenuto fino al caricamento dell'app. Gli elementi segnaposto, con un'animazione semplice, dovrebbero essere visualizzati nelle posizioni in cui verranno caricati i contenuti.
- Schermate che forniscono un'esposizione momentanea del marchio, visualizzando un logo o altri elementi che migliorano il riconoscimento del marchio. In questa schermata, si evita di utilizzare il testo tranne quello del logo e, se applicabile, uno slogan.

### 2.11.7 Onboarding

L'onboarding è un'esperienza di unboxing virtuale che aiuta gli utenti a iniziare con un'app. È un punto di un percorso più lungo che inizia nell'App

Store e termina con l'esecuzione della prima vera azione da parte dell'utente sull'app. Quando si progetta l'onboarding, bisogna considerare le schermate precedenti e quelle che verranno dopo. E' fondamentale mostrare l'onboarding solo ai nuovi utenti, per evitare di peggiorare l'esperienza. 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.

## Stati offline

Gli stati offline consentono agli utenti di interagire con un'app senza accesso a Internet. Si adatta il comportamento dell'app quando l'utente ha una connessione lenta, intermittente o assente. Quando la connessione è limitata, si carica ugualmente il contenuto disponibile, anziché non caricare nulla. Per comunicare che una funzione dell'app sta funzionando offline, si mostra l'icona pin offline con l'etichetta di testo "offline". Quando le funzionalità non sono disponibili offline, si indicano utilizzando l'icona nuvola off, possibilmente utilizzando anche l'etichetta di testo "offline" insieme all'icona.

### 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. Inoltre, non è necessario scrivere subito tutte le informazioni, ma risulta meno dispersivo rivelarle progressivamente e secondo necessità, man mano che l'utente le esplora e necessita di maggiori informazioni.

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 ( "1, 2, 3", non "uno, due, tre") a meno di casi particolari, come "Inserisci due 3". 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. Ad esempio, se il resto del flusso dell'app non usa i periodi, introdurli in alcuni punti potrebbe sembrare incoerente. Inoltre, ci si rivolge agli utenti in seconda persona (tu o tuo) o in prima persona (io, io o mio), a seconda di quale sia adatto e più chiaro 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.

# 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>.