

Material Design 3

Ferin Eli

Nardi Marco

Trincanato Marco

13 maggio 2022

Indice

1	Esempi e template	3
1.1	Citazioni	3
1.2	Blocco di codice	3
2	Principi del Material Design	5
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	Principi delle linee guida	6
2.2.2	Organizzazione dei contenuti	7
2.2.3	Misure	8
2.2.4	Responsive layout grid	8
2.2.5	Spacing methods	9
2.2.6	Containers e Aspect Ratios	9
2.2.7	Density	9
2.3	Navigazione	10
2.3.1	Direzioni di navigazione	10
2.3.2	Navigation transitions	12
2.3.3	Ricerca	13
2.4	Colori	14
2.4.1	Temi	15
2.5	Suono	15
2.5.1	Applicare il suono all'interfaccia utente	17
2.5.2	Tipi di suono	17
2.5.3	Sound attributes	19
2.5.4	Sound choreography	21
2.5.5	Ottimizzazione del suono	22
2.5.6	Formato dei file e ottimizzazione della memoria	23
2.6	Tipografia	23

2.6.1	Titoli	23
2.6.2	Proprietà di un typeface	24
2.6.3	Leggibilità	25
2.6.4	Linguaggi	25
2.7	Iconografia	25
2.7.1	Icone di sistema	26
2.7.2	Icone animate	26
2.8	Interaction	27
2.8.1	Gesture	27
2.8.2	Selezione	28
2.8.3	Stato	28
2.8.4	Tipi di stato	28
2.9	Motion	29
2.9.1	Transition patterns	29
2.9.2	Velocità	30

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 Principi delle linee guida

Il material Design fornisce delle linee guida per ottenere dai layout che sono intuitivi, consistenti, e responsive. Ciò viene ottenuto attraverso l'uso di griglie, metodi di spaziamento degli elementi, vincoli e altro.[4]

Il layout è diviso in 3 zone:

- App bars
- Navigation
- Body

Body

Il body è la regione usata per mostrare la maggior parte dei contenuti in un'app.

Esistono diversi layout per gestire i contenuti presenti nel body come per esempio:

- Responsive column grid
- Responsive layout grid

Per esempio per una dimensione dello schermo tra 0 e 599dp vengono consigliate 4 colonne per il layout e un margine di 16dp

Navigation

La parte della navigazione aiuta l'utente a raggiungere azioni, componenti o impostazioni importanti utilizzando componenti come il navigation drawer

La regione di navigazione ha sempre una grandezza di 72dp quando chiusa (collapsed) e 256 dp quando aperta

Esistono casi in cui su certe grandezze dello schermo per fare spazio si diminuisce la grandezza del body, o su schermi ancora più piccoli la barra di navigazione può apparire "sopra" il body oscurandolo quasi completamente

App bar

L'app bar viene usata per mostrare o raggruppare azioni e componenti che possono servire all'utente

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: per esempio una notizia avrà un'immagine, un titolo e magari un piccolo paragrafo introduttivo. Questi elementi possono essere contenuti assieme in diversi modi:

Contenimento implicito

Si ottiene riducendo lo spazio tra gli elementi e aumentando lo spazio verso gli elementi estranei

Contenimento esplicito

Si ottiene creando una divisione più marcata con gli elementi estranei come per esempio con una cornice, utilizzare delle cards o altri elementi che esplicitano la presenza di un gruppo [5]

2.2.3 Misure

Nel material design tutte le misure sono allineate ad incrementi di 8dp per mantenere consistenza visiva. Nel caso di elementi più piccoli come icone sono allineate invece a 4dp [6]

2.2.4 Responsive layout grid

Il responsive layout grid è un layout fatto per adattarsi a diverse grandezze schermo e orientamenti per garantire consistenza [7]

E' principalmente composto da:

- Columns
- Gutters
- Margins

Columns

Per un layout responsive la grandezza delle colonne viene espressa in percentuale al posto di valori fissi per permettere al contenuto di adattarsi a diverse grandezze schermo. Il numero di colonne è definito in base a dei range già preimpostati: per esempio su uno schermo mobile (360dp) ci possono essere fino a 4 colonne

Gutters

Il gutter è lo spazio tra colonne ed è anche questo un valore predefinito in base alla grandezza dello schermo: per esempio su uno schermo mobile è da 16dp

Margins

I Margins sono lo spazio dai bordi laterali dello schermo, possono essere definiti con valori fissi o percentuali

2.2.5 Spacing methods

Per manipolare lo spazio in modo più granulare del responsive layout grid esistono funzionalità come:

Padding

Il padding è lo spazio tra elementi dell'interfaccia, sia verticale che orizzontale

Alignment

E' l'allineamento di un elemento all'interno di un componente

Keylines

Keylines sono uno strumento per allineare in modo consistente elementi al di fuori dal layout grid. Sono delle linee verticali misurate dalla distanza dai lati dello schermo ad incrementi di 8dp[8]

2.2.6 Containers e Aspect Ratios

Un container rappresenta un'area che contiene elementi di interfaccia utente. Possono essere rigidi e quindi tagliare il contenuto al suo interno nel caso vada oltre i limiti oppure flessibili e cambiare 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[9]

2.2.7 Density

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, può essere utile rendere le informazioni più compatte diminuendo lo spazio tra di loro, rendendole più facili da "scannare" o confrontare. Vengono date delle indicazioni [10] su quando non usare l'aumento di densità:

Focused tasks

Componenti come un *Date picker*, cioè componenti che si concentrano su svolgere un'azione (scegliere un giorno), soffrono di un aumento della densità in quando rende più difficileappare la data corrente per l'utente

Alerts e messages

Componenti che si occupano di avvisare l'utente di errori, cambiamenti o altro devono essere facilmente visibili, e quindi renderli più compatti rischia di renderli meno leggibili per l'utente

Density nel layout

Per bilanciare un layout che ha componenti ad alta densità è consigliato avere margins e gutters a bassa densità. Avere sia componenti che valori del layout a bassa densità, rende l'esperienza dell'utente confusa a causa della difficoltà nel distinguere raggruppamenti.

Density nei components

E' possibile cambiare anche la densità di componenti come buttons o simili, ma bisogna stare attenti in quanto sono già dei componenti piccoli e ridurne troppo la densità può *romperli*. Bisogna stare attenti anche al fatto che cambiando la densità di un componente le proprie dimensioni cambiano ma non padding e allineamento. Inoltre se sono presenti più elementi all'interno di un componente è consigliato usare 4dp per separarli e centrarli all'interno del component

2.3 Navigazione

La maggior parte delle applicazioni è composta da più schermate che mostrano all'utente diversi contenuti. E', quindi, fondamentale che l'utente possa muoversi agevolmente tra queste schermate. Quest'operazione si chiama navigazione. Vista l'importanza dello spostamento tra schermate, nella documentazione Material Design sono state devinite delle relazioni gerarchiche tra le diverse schermate. Quindi, sono state definite delle direzioni di navigazione.

2.3.1 Direzioni di navigazione

Lateral Navigation

La lateral navigation (navigazione laterale) si usa per muoversi tra schermate allo stesso livello di gerarchia. Consente l'accesso a diverse destinazioni e funzionalità dell'app o il passaggio tra elementi correlati. Secondo le linee guida del material design, il componente di navigazione più importante di un'app dovrebbe fornire l'accesso a tutte le destinazioni al livello più alto

della sua gerarchia. Per le app con due o più destinazioni di primo livello, si consiglia di fornire la navigazione laterale tramite:

- *Navigation drawers*, appropriati per cinque o più destinazioni di primo livello. Possono essere utilizzati su dispositivi di tutte le dimensioni per un'esperienza di navigazione coerente.
- *Bottom navigation bars*, che forniscono l'accesso a 3-5 destinazioni di primo livello sui dispositivi mobili. La loro posizione, visibilità e persistenza sugli schermi consentono un rapido spostamento tra le destinazioni;
- *Tabs*, che possono essere utilizzate a qualsiasi livello della gerarchia di un'app per presentare due o più set di dati di uguale importanza su tutte le dimensioni dello schermo.

Forward Navigation

La forward navigation (navigazione in avanti) si riferisce allo spostamento tra le schermate a livelli consecutivi di gerarchia, passaggi in un flusso o attraverso un'app. La navigazione in avanti comprende il comportamento di navigazione in contenitori (come schede, elenchi o immagini), pulsanti, collegamenti o utilizzando la ricerca. Le linee guida specificano 3 tipi di movimento tra le schermate per completare un'attività:

- verso il basso nella gerarchia di un'app per accedere da un livello gerarchico superiore (padre) a un livello inferiore (figlio);
- in sequenza, attraverso un flusso o una sequenza ordinata di schermate. Ad esempio, un processo di iscrizione);
- direttamente da una schermata all'altra nell'app. Ad esempio, da una schermata iniziale a una schermata nella gerarchia dell'app.

Mentre la navigazione laterale utilizza componenti di navigazione dedicati, la navigazione in avanti è spesso incorporata nel contenuto di una schermata attraverso diversi componenti. La navigazione in avanti può essere implementata utilizzando contenitori di contenuti come schede, elenchi o elenchi di immagini, pulsanti che avanzano a un'altra schermata, ricerca in-app su una o più schermate, collegamenti all'interno del contenuto.

Reverse Navigation

La reverse navigation (navigazione all'indietro) si riferisce allo spostamento attraverso le schermate in ordine cronologico inverso. Le convenzioni della piattaforma determinano il comportamento esatto della navigazione inversa all'interno di un'app. Ci sono 2 tipi di reverse navigation:

- *Reverse chronological navigation*, fornita dal sistema operativo o dalla piattaforma. Consentono all'utente di spostarsi tra le schermate all'interno di un'app o tra più app. Le singole piattaforme definiscono come si comporta e come gli utenti possono accedere a tale funzionalità.
- *Upward navigation*, che consente agli utenti di spostarsi di un livello verso l'alto all'interno della gerarchia di una singola app, finché non viene raggiunta la schermata iniziale o la schermata di livello superiore dell'app. Ad esempio, la freccia in alto nella barra di un'app è una forma di navigazione inversa verso l'alto. La navigazione verso l'alto dovrebbe essere implementata per tutti gli schermi figli in un'app e seguire le indicazioni della piattaforma. Le guidelines consigliano di utilizzare l'elemento Material Up.

Il design e la funzionalità dell'app dovrebbero tenere conto di entrambi i tipi di navigazione inversa. Per ottimizzare l'esperienza di un utente durante la navigazione inversa, vengono date queste indicazioni:

- riportare gli utenti alla posizione e allo stato precedenti dello schermo, per accelerare il ricaricamento delle informazioni e la ripresa delle attività;
- fornire messaggi chiari se lo stato precedente di una schermata non è più disponibile;
- indicare chiaramente il rapporto delle schermate figlio con le schermate che le precedono nella gerarchia.

2.3.2 Navigation transitions

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.

2.3.3 Ricerca

La ricerca consente agli utenti di trovare rapidamente un contenuto specifico presente nell'app. La ricerca di base consente agli utenti di inserire una query in un campo di testo di ricerca per visualizzare i risultati correlati. I metodi di input delle query di ricerca possono essere estesi per includere cronologia, completamento automatico delle query e input vocale.

Persistent Search

La Persistent search si usa quando la ricerca è l'obiettivo principale dell'app. Il campo di testo di ricerca viene presentato all'interno di una barra di ricerca, pronta per ricevere il *focus* (il tocco da parte dell'utente che vuole utilizzarla). Quando è in fuoco, l'esperienza di ricerca si espande per riempire l'intero schermo. Eventualmente, la cronologia di ricerca può essere visualizzata

sotto il campo di testo. Un utente può digitare una query o selezionare da un suggerimento (se disponibile). Premendo Invio, la ricerca viene eseguita. I risultati della ricerca vengono visualizzati sotto la barra di ricerca. Per rimuovere lo stato attivo dalla casella di ricerca e ignorare i suggerimenti di ricerca, l'utente deve toccare la freccia su.

Expandable search

L'expandable search si usa quando la ricerca non è l'obiettivo principale dell'app. Si mostra un'icona di ricerca nella barra degli strumenti, invece di una casella di testo di ricerca aperta. Un utente può toccare l'icona di ricerca (nella barra degli strumenti) per trasformarla in una barra di ricerca mirata. Mentre l'utente sta digitando, la cronologia può essere visualizzata sotto la barra di ricerca. Dopo aver inserito una query o aver scelto un suggerimento, premendo Invio la ricerca viene eseguita. I risultati vengono visualizzati sotto la barra di ricerca e scorrere al di sotto di essa. Durante la visualizzazione dei risultati, il campo di testo della ricerca rimane visibile, ma non a fuoco. Toccando la freccia su si libera lo stato attivo dalla ricerca, si ignorano i suggerimenti e la tastiera su schermo e si riporta la barra degli strumenti al suo stato originale.

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](#) [11]

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 silenziato 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'emozione, 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 (monospaced), 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 Interaction

2.8.1 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

2.8.2 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.

2.8.3 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

2.8.4 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

2.9 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.

2.9.1 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](#).

2.9.2 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](#)

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 misura*. URL: <https://material.io/design/layout/understanding-layout.html#composition>.
- [7] *Material Design layout grid*. URL: <https://material.io/design/layout/responsive-layout-grid.html#columns-gutters-and-margins>.
- [8] *Material Design layout spacing*. URL: <https://material.io/design/layout/spacing-methods.html#spacing>.
- [9] *Material Design layout containers*. URL: <https://material.io/design/layout/spacing-methods.html#containers-and-ratios>.
- [10] *Material Design layout density*. URL: <https://material.io/design/layout/applying-density.html#usage>.
- [11] *Material Color Tool*. URL: <https://material.io/resources/color/#!/view.left=0&view.right=0&primary.color=6002ee>.