**LEZIONE CBL – DESIGN 19/06/2025**

Il **design** riguarda diversi ambiti (interior, fashion, image, product) ma ha sempre al centro il **progetto intenzionale** volto a creare soluzioni con uno scopo preciso. A differenza del disegno, che è un’attività visiva, il design è un **processo strutturato e metodologico** di progettazione, non necessariamente visivo.

Nel contesto digitale, il design si declina in **UI (User Interface)** e **UX (User Experience)**, ovvero come l’utente interagisce con interfacce come app o siti web. La crescente diffusione delle tecnologie ha creato la necessità di specialisti capaci di progettare interfacce intuitive e gradevoli.

Un buon design digitale deve garantire un’**interazione naturale e immediata**, senza far sentire l’utente “costretto” a imparare come usare l’app, ma rendendo l’esperienza fluida e piacevole. Fondamentale è rispettare i **pattern cognitivi e comportamentali preesistenti** nelle persone, che fungono da riferimenti per creare esperienze coerenti e intuitive, evitando di confondere o allontanare l’utente.

In sintesi, il design mira a costruire esperienze che guidano l’utente in modo quasi invisibile, facilitando l’uso e incentivando il ritorno.

**APP ICONS**

Nel contesto delle app, l’**icona** è il primo elemento visivo che l’utente vede e rappresenta un punto di contatto cruciale, quindi non va sottovalutata. Dal punto di vista della **User Interface (UI)**, l’icona deve essere unica, memorabile e coerente con l’identità visiva dell’app, senza includere testi lunghi o elementi dell’interfaccia interna. Deve attirare l’attenzione senza risultare fuorviante.

Tecnicamente, Apple richiede che l’icona sia di 1024x1024 pixel per poi generare automaticamente versioni scalate (1x, 2x, 3x) adatte ai vari dispositivi e alle diverse densità di pixel, garantendo un aspetto nitido su ogni schermo.

Per creare queste varianti si possono usare software grafici o strumenti di intelligenza artificiale, assicurandosi che ogni versione sia chiara anche a dimensioni ridotte.

Infine, è importante **semplificare il design dell’icona**, evitando troppi dettagli che diventano illeggibili in formato piccolo, come dimostra l’esempio del logo Safari, che mantiene la sua efficacia grazie alla semplicità visiva nelle versioni ridotte.

**COLORS**

Ecco un riassunto del testo:

Nel design di un’app, la **scelta dei colori** è cruciale perché suscita emozioni, comunica significati culturali e influenza l’esperienza utente. I colori possono avere significati diversi a seconda delle culture (es. il rosso simboleggia pericolo in Occidente ma fortuna in Asia, il bianco purezza in Occidente e lutto in alcune culture orientali). Oltre agli aspetti culturali, i colori evocano risposte emotive universali: il blu calma, il giallo gioia, il verde natura, il nero eleganza o mistero. È importante scegliere i colori considerando non solo l’estetica ma anche l’impatto psicologico. L’**accessibilità visiva** è un altro aspetto fondamentale: bisogna garantire un adeguato contrasto tra colori per facilitare la lettura anche a persone con difficoltà visive. Esistono linee guida internazionali e strumenti come **whocanuse.com** che permettono di testare la leggibilità delle combinazioni di colori, assegnando punteggi (A, AA, AAA) basati sul contrasto. Questo strumento aiuta a scegliere palette di colori che assicurino buona visibilità e migliorino l’esperienza per tutti gli utenti. Il sito **whocanuse.com** è uno strumento gratuito e facile da usare che permette di testare combinazioni di colori inserendo codici esadecimali. Fornisce informazioni sulla resa dei colori su vari sfondi, sulla leggibilità per persone con disabilità visive e assegna punteggi di accessibilità (A, AA, AAA). È utile per verificare in anticipo se i colori scelti sono adeguati, migliorando l’esperienza utente.

**ICONS**

Quando parliamo di icone all’interno di un’app, non ci riferiamo solo all’icona principale visibile nella schermata Home (di cui abbiamo già parlato), ma anche a tutte quelle icone secondarie che popolano l’interfaccia dell’applicazione vera e propria. Queste icone, per quanto più piccole e spesso "in secondo piano", sono fondamentali per l’esperienza utente: guidano, indicano funzioni, danno coerenza visiva e influenzano la percezione della qualità generale dell'app.

**1. SF Symbols: lo standard Apple**

Apple mette a disposizione un ampio set di **icone standard chiamate SF Symbols**. Queste icone sono già **perfettamente integrate con il sistema**, il che significa che:

* sono progettate con una **dimensione coerente**;
* rispettano le **linee guida di design Apple**;
* si adattano automaticamente a diverse dimensioni, pesi e stili;
* supportano funzioni avanzate come l’adattamento automatico alla **modalità scura**, alla **dimensione dinamica del testo** o alla **direzione dell’interfaccia (es. RTL)**.

Utilizzare gli SF Symbols è **altamente consigliato** perché permette di **risparmiare tempo**, garantisce **coerenza visiva** e rende l’app immediatamente familiare agli utenti, che riconoscono simboli come il cestino, la lente, il "+" o il simbolo della condivisione.

**2. Personalizzare le icone: attenzione alla coerenza**

Tuttavia, ci sono casi in cui potrebbe essere utile o necessario **creare un set di icone personalizzate**, ad esempio:

* quando si vuole dare all’app un’identità visiva unica (pensiamo a giochi o app creative);
* quando si vogliono evocare elementi non presenti negli SF Symbols;
* quando si desidera applicare uno **stile grafico specifico o distintivo**.

In questi casi, però, bisogna procedere con cautela. Il rischio maggiore non è tanto legato alle **dimensioni** delle icone (che si possono sempre adattare), quanto alla **coerenza grafica** tra tutte le icone personalizzate. Se ogni simbolo ha uno stile leggermente diverso — in spessore, proporzione, livello di dettaglio o colore — l’interfaccia risulterà **disomogenea e meno professionale**.

Per mantenere coerenza stilistica, può essere utile **utilizzare l’intelligenza artificiale**, ma è fondamentale scrivere **prompt ben strutturati**, specificando le proporzioni, il contesto d’uso e il tipo di tratto desiderato. Ricordiamoci che meno vincoli mettiamo, più è difficile ottenere risultati coerenti.

**3. Le varianti degli SF Symbols**

Gli SF Symbols sono forniti **in più versioni** per adattarsi a diversi contesti d’uso. Le quattro principali varianti grafiche sono:

* **Outline (vuota)** – solo il contorno dell’icona;
* **Filled (piena)** – l’icona completamente riempita;
* **Circle + outline** – l’icona inserita in un cerchio, in versione contorno;
* **Circle + filled** – icona in un cerchio pieno.

Queste varianti sono pensate per garantire **flessibilità visiva**: ad esempio, un’icona “vuota” può essere usata in uno stato inattivo, mentre la versione “piena” segnala un’azione completata o uno stato attivo.

**4. Colorazione: monocromatico, gerarchico e multicolore**

Anche la **gestione del colore** nelle icone è un aspetto cruciale. Esistono diversi stili di colorazione:

* **Monocromatico**: un unico colore applicato uniformemente a tutta l’icona. È semplice, chiaro e funziona bene nella maggior parte dei casi.
* **Gerarchico**: sempre monocromatico, ma con **diverse opacità** applicate ai livelli dell’icona, creando una **gerarchia visiva** interna. Ad esempio, il bordo di un cerchio può essere più opaco dell’elemento centrale, mettendo in evidenza ciò che conta di più.
* **Multicolor**: alcune icone, come il simbolo del cestino, possono essere rappresentate in **più colori** per enfatizzare il significato. Ad esempio, un **cestino rosso** comunica immediatamente un’azione distruttiva o definitiva (come l’eliminazione di dati). Tuttavia, il colore rosso deve essere **usato con attenzione**: se lo applichiamo a troppe icone, perde la sua forza comunicativa e rischia di generare confusione o ansia.

La **scelta del colore** (e del numero di colori) dovrebbe riflettere sempre una **gerarchia visiva ben precisa**: evidenziare solo ciò che davvero deve emergere, e lasciare il resto in secondo piano. Troppi colori o colori mal calibrati possono ridurre la **leggibilità** e l’**impatto comunicativo**.

Un esempio classico di cattiva leggibilità è l’uso di **arancione e viola insieme**: sono colori che, pur stando bene dal punto di vista estetico, possono risultare **difficili da leggere se usati con scarso contrasto**. In questi casi è preferibile **limitare la palette** e giocare con il **livello di opacità** per distinguere i livelli senza sacrificare la leggibilità.

**IMAGES**

Nelle app moderne, l’uso delle **immagini** è decisamente meno centrale rispetto al passato. Questo non significa che siano scomparse, ma che oggi il design privilegia **interfacce più essenziali, chiare e leggere**, soprattutto per garantire prestazioni elevate, semplicità d’uso e una migliore esperienza utente. Tuttavia, quando decidiamo di includere immagini all’interno della nostra app, dobbiamo farlo in modo **ragionato e consapevole**, seguendo alcune buone pratiche fondamentali.

**1. Qualità dell’immagine: mai trascurarla**

Prima di tutto, è essenziale garantire che le immagini utilizzate siano di **alta qualità**. Un’immagine **sgranata, sfocata o con una risoluzione troppo bassa** comunica scarsa cura, può apparire poco professionale e abbassa il livello qualitativo percepito dell’intera app. Questo vale sia per le immagini principali (es. copertine, sfondi, anteprime) che per quelle secondarie o decorative.

Inoltre, bisogna assicurarsi che **le immagini siano coerenti tra loro**. Questo significa che dovrebbero appartenere allo **stesso stile visivo**: se alcune immagini sono fotografiche e altre invece sono illustrazioni piatte, il risultato sarà disomogeneo. Lo stesso vale per immagini con tonalità cromatiche molto diverse o con stili troppo distanti (es. immagini vintage mischiate a elementi ultramoderni). La coerenza visiva crea un'interfaccia **ordinata, armonica e professionale**.

**2. 1x, 2x e 3x: il concetto di scala nei dispositivi Apple**

Nel mondo iOS, le immagini vengono fornite in diversi **fattori di scala** per adattarsi ai vari dispositivi. In particolare, dobbiamo sapere che:

* **@1x** è la dimensione base (standard, usata raramente oggi);
* **@2x** è la dimensione per schermi Retina (la più comune);
* **@3x** è la dimensione per schermi con densità di pixel ancora più alta (es. iPhone di fascia alta).

Anche se nella pratica spesso ci concentriamo su immagini in **@2x** — perché sono il compromesso ideale in termini di qualità e dimensione del file — è importante capire che **questo concetto di scala può essere applicato a qualsiasi asset grafico**, non solo alle immagini. Quindi anche le icone, i loghi e gli altri elementi visivi possono (e spesso devono) essere esportati in formato 1x, 2x e 3x per garantire **nitidezza e resa ottimale su ogni dispositivo Apple**.

**3. Dove trovare immagini di qualità: risorse gratuite e con filtri**

Quando ci servono immagini da inserire nel nostro progetto, non dobbiamo necessariamente crearle da zero. Esistono infatti **molti siti gratuiti** che mettono a disposizione **immagini libere da diritti**, spesso organizzate per tema, composizione, colore e orientamento. Questi strumenti sono utilissimi soprattutto quando:

* si sta progettando un layout e serve **una base visiva coerente**;
* si vogliono testare varie versioni di design prima di arrivare alla scelta definitiva;
* si cercano **immagini illustrative o ispirazionali** da integrare nei mockup o nei prototipi.

Molti di questi siti permettono anche di **applicare filtri** avanzati per raffinare la ricerca (es. trovare immagini con uno sfondo neutro, con persone, oggetti singoli o palette specifiche), facilitando così la **coerenza visiva del progetto**.

**TYPOGRAPHY**

Quando parliamo di **tipografia** nel contesto della progettazione di interfacce utente (UI), ci riferiamo all’uso dei **caratteri tipografici (font)** e al modo in cui essi vengono impiegati per trasmettere informazioni in modo chiaro, efficace e leggibile. La scelta del font, delle dimensioni, dello stile e della gerarchia visiva ha un impatto diretto sull’esperienza utente, perché influenza la **leggibilità**, la **scansione del contenuto** e la percezione generale dell’app.

**1. Le due famiglie principali: Serif e Sans Serif**

Il primo concetto base da comprendere è che i font si suddividono in **due grandi famiglie**:

* **Font Serif (graziati):** Sono caratterizzati dalla presenza di piccole **"grazie"** o **"terminazioni"** decorative alle estremità delle lettere. Queste piccole curve o tratti aggiuntivi aiutano l’occhio nella lettura fluida di **blocchi di testo lunghi**, come quelli che troviamo nei libri o negli articoli. Un esempio classico è **Times New Roman**, uno dei font serif più noti e utilizzati, in cui possiamo notare le grazie sulla **"r"**, sulla **"g"** e su molte altre lettere.
* **Font Sans Serif (a bastoni):** Come suggerisce il nome (dal francese “sans”, cioè “senza”), questi font **non presentano grazie**, ma si distinguono per la loro linea **pulita, essenziale e moderna**. Sono più adatti per testi brevi, interfacce digitali, titoli e lettura su schermo, perché risultano **più chiari e leggibili a piccole dimensioni**. Un esempio pratico è il font **San Francisco**, il font ufficiale utilizzato da Apple nei propri sistemi operativi.

**2. I font Apple: San Francisco e New York**

Apple utilizza principalmente due font di sistema:

* **San Francisco:** È il **font di default** di iOS, macOS, watchOS e tvOS. È un **sans serif** estremamente versatile, con molte varianti e pesi (light, regular, bold, ecc.) ed è stato progettato appositamente per garantire **alta leggibilità su tutti i dispositivi Apple**, sia in condizioni normali sia in modalità accessibilità. È il font ideale per la quasi totalità dell’interfaccia, compresi pulsanti, testi brevi, etichette e contenuti di piccole dimensioni.
* **New York:** È un **font serif** sviluppato da Apple, utilizzato in contesti molto specifici, per esempio **nei titoli di libri o riviste** all’interno delle app. La sua funzione è quindi più “decorativa” o editoriale. Nonostante sia un font elegante, **non è consigliato per la lettura su schermo di testi brevi o piccoli**, perché la presenza delle grazie può rendere più faticosa la lettura, soprattutto su schermi piccoli o ad alta densità di pixel.

**3. Scelte tipografiche corrette: dimensioni, leggibilità, coerenza**

Quando scegliamo un font per la nostra app, dobbiamo anche considerare:

* La **dimensione minima consigliata** è **11pt**, per garantire una buona leggibilità anche in condizioni standard. Tuttavia, nelle linee guida di Apple è spesso suggerito di usare **dimensioni che siano multipli di 4** (es. 12, 16, 20, 24), in modo da mantenere coerenza e armonia tra testi e spaziature.
* **I font serif sono preferibili per testi lunghi**, ma solo quando lo schermo lo permette e le dimensioni del font sono adatte. Su dispositivi mobili, dove lo spazio è ridotto e la lettura è più veloce, è meglio optare per un sans serif come San Francisco, che facilita la **scansione visiva** dei contenuti.
* Al contrario, l’uso di font serif per **piccoli testi o interfacce complesse è sconsigliato**, perché può rendere la lettura più difficile, richiedendo all’occhio uno **sforzo cognitivo maggiore**. Quindi, anche se si vuole adottare uno stile “editoriale” o “classico”, bisogna farlo con equilibrio e sempre pensando all’utente finale.

**LAYOUTS E COMPONENTI DI INTERFACCIA**

Nel design di un’interfaccia utente, il concetto di layout è fondamentale. Quando parliamo di layout ci riferiamo alla struttura visiva che organizza il contenuto sullo schermo. Un layout ben progettato permette di distribuire gli elementi in modo ordinato, coerente e accessibile, migliorando l’esperienza utente.

In modo molto pratico, possiamo immaginare il layout come una griglia composta da rettangoli, pensati per ospitare i contenuti (testi, immagini, pulsanti, ecc.) nella maniera più efficiente e leggibile possibile. Questi rettangoli seguono regole precise di allineamento e proporzione, spesso basate su colonne.

Apple consiglia di utilizzare griglie tra le 8 e le 12 colonne, ma di default si considerano 10 colonne. Queste colonne aiutano a mantenere una disposizione coerente e responsiva, adattabile ai diversi dispositivi e dimensioni di schermo.

**Safe Area**

Quando si progettano interfacce su iOS, bisogna tenere conto delle cosiddette safe area, ovvero le aree visibili e sicure dello schermo, dove il contenuto non rischia di essere coperto da elementi di sistema come la barra inferiore, il notch, o la Dynamic Island. Xcode ci fornisce dei riferimenti chiari per sapere dove si trova la safe area, e dobbiamo sempre posizionare gli elementi interattivi all’interno di essa per garantire una buona esperienza utente, anche dal punto di vista dell’accessibilità.

**LABELS**

Una Label è un componente statico di interfaccia che serve per visualizzare un testo, ma non è modificabile dall’utente. Può essere selezionata e copiata, ma non modificata. Le label sono fondamentali per mostrare nomi, titoli, descrizioni o istruzioni all’interno dell’interfaccia. iOS fornisce quattro colori di default per le label (come il colore primario, secondario, ecc.), e l’uso corretto di questi colori aiuta a costruire una gerarchia visiva. Questo significa che possiamo dare più o meno importanza al testo in base al colore utilizzato, aiutando così l’utente a distinguere a colpo d’occhio ciò che è rilevante.

**TEXT FIELDS**

I Text Field, a differenza delle Label, sono elementi interattivi: si tratta di campi di testo editabili che permettono all’utente di inserire informazioni, come nel caso di un form. Un esempio classico è un campo per inserire email e password in un modulo di login. Tuttavia, una buona progettazione suggerisce di non inserire un login se non è strettamente necessario. Chiedere dati personali all’utente richiede responsabilità, e se non è essenziale per la funzione dell’app, è meglio evitarlo. Quando utilizziamo text field, è importante guidare l’utente nel loro utilizzo. Per esempio, possiamo mostrare un placeholder all’interno del campo per spiegare cosa ci si aspetta venga inserito, oppure aggiungere un clear button per permettere di cancellare il testo rapidamente, migliorando così l’esperienza d’uso. Infine, è utile ripetere il tipo di contenuto previsto, ad esempio “Inserisci la tua email”, per essere chiari e accessibili.

**TAB BAR**

La Tab Bar è un componente di navigazione che si trova in basso nello schermo e consente di passare da una schermata all’altra all’interno dell’app. Ogni tab rappresenta una sezione distinta dell’applicazione, ma tutte le sezioni devono avere un tema comune che le accomuna. In una buona progettazione, la Tab Bar dovrebbe contenere da un minimo di 3 a un massimo di 5 opzioni. Meno di 3 opzioni non giustifica l’uso della tab bar; in tal caso, possiamo optare per una singola view oppure per un altro tipo di navigazione. Se abbiamo solo 2 schermate, può capitare di “chiudere un occhio” e usarla comunque, ma è una soluzione da valutare con attenzione. Importante: la tab bar non è obbligatoria, ma può essere molto utile quando l’applicazione ha sezioni chiare e separate tra loro.

**SEGMENTED CONTROL**

Il Segmented Control è un componente che funziona da filtro ed è posizionato solitamente nella parte alta di una schermata. Serve per suddividere e selezionare visivamente diverse categorie o stati di un contenuto. Un esempio classico è l’app Telefono di iOS: nella sezione “Recenti”, il segmented control permette di scegliere se vedere tutte le chiamate, solo quelle perse, o quelle in uscita. Ciascun segmento funziona quasi come un pulsante che aggiorna il contenuto visibile, oppure può portare a una view differente con criteri specifici. È buona norma modificare il colore o il peso del testo dei segmenti per rendere chiaro quale sia selezionato, anche se spesso questo comportamento viene gestito automaticamente da Xcode per garantire coerenza visiva e accessibilità.

**LISTS E TABLES**

Le **Liste** e le **Tabelle** sono componenti che **rappresentano gruppi o gerarchie di dati**, spesso in forma testuale. Questi componenti permettono **interazioni dirette**: l’utente può selezionare un elemento, eliminarlo, modificarlo o approfondirne i dettagli. Ad esempio, una lista può contenere delle voci cliccabili che portano a **nuove schermate**, in un comportamento chiamato **navigazione a profondità**. In questi contesti, è importante **non sovraccaricare la lista** con immagini o troppi dettagli. Meglio essere **succinti** e mantenere l’interfaccia semplice e ordinata.

**CONTENTS: TextView e Istogrammi**

Nel mostrare **contenuti complessi**, possiamo usare componenti come:

* **Istogrammi** o grafici per **comunicare dati visivi**, ad esempio dati statistici o misurazioni.
* **TextView** per inserire **grandi blocchi di testo**, come in un'app Note. A differenza dei **TextField**, che sono più adatti per campi brevi (es. email, nome), le **TextView** permettono all’utente di **scrivere contenuti più estesi**.

**MENUS AND ACTIONS: Activity View**

Un altro componente importante è l’**Activity View**, conosciuto anche come **schermata di condivisione**. Viene utilizzato per **mostrare un insieme di azioni** contestuali che l’utente può eseguire, come condividere un file, inviare un link, salvare un contenuto, ecc.

Questa vista si adatta al contesto e offre **solo le opzioni rilevanti** in quel momento, migliorando la semplicità e l’efficienza dell’interazione.

**SLIDERS**

Uno **Slider** è un componente grafico che **permette all’utente di selezionare un valore all’interno di un intervallo**(minimo - massimo) spostando un indicatore su una barra orizzontale. È utile, ad esempio, per regolare il volume, la luminosità o un livello. Importante: ogni elemento che può essere **toccato con il pollice** deve rispettare le linee guida Apple in termini di dimensioni, ovvero avere **almeno 44x44 punti**. Questo garantisce **accessibilità e usabilità**, specialmente su dispositivi mobili.

**SELECTION AND INPUT**

I **pull-down buttons** sono bottoni che, una volta premuti, **mostrano un menu a discesa** con più opzioni tra cui scegliere. Dopo aver selezionato un’opzione, il menu si **richiude automaticamente**, migliorando l’efficienza.

Quando abbiamo **tre o più opzioni** per una stessa azione, è consigliato usare questo tipo di componente per **semplificare l’interfaccia** e non occupare troppo spazio visivo con bottoni separati.

**TOGGLES**

Un **toggle** è un **interruttore** che permette di **attivare o disattivare una funzione**. Rappresenta una scelta binaria (acceso/spento, sì/no, abilitato/disabilitato). Quando ci troviamo di fronte a **due opzioni contrapposte**, è **preferibile usare un toggle** piuttosto che due pulsanti distinti, perché è più chiaro e intuitivo.

User Flow

Speiga come il fliusso dell’utente dovrebbe mostrare come l’utente si muove attraverso il prodotto o il servizio per conseguire lo scopo. Se il mio goal è loggarmi nell’app, dovrei costrutire un Login flow, aprea app, tocca login, inserisce mail e password, e vede la dashboard. Lìimprotate è avere un idea dello userFlow dell’utente, spesso l’utente fa al massimo 7 tab, poi si scoccia. Per scrivere un flow, occorre partire dal goal che intediamo conseguire. Per costruire lo user Flow è possibile usare freeForm

Esercizio: dato

**Apri applicazione**

Clicca su “Prenota Volo”

Selezione “Punto di partenza e Destinazione”

Seleziona “Data di andata” e opzionale “Data ritorno”

Seleziona” Posto a sedere”

Inserisci “Informazioni personali”

Seleziona “Metodo di pagamento”

Inserisci “Informazioni di pagamento”

Conferma “pagamento”

**Visualizza ”Conferma sullo Schermo”**

Scaricare Sketch per mockup dell’applicazione