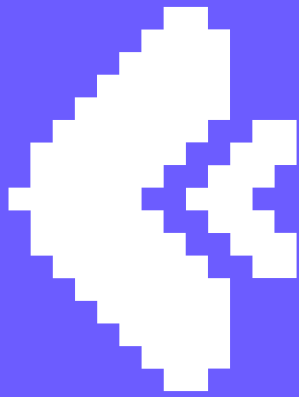


LEVEL UP!



CODING LAB



**Politecnico
di Torino**

Politecnico di Torino

DAD - Dipartimento di Architettura e Design

Corso di Laurea Triennale in Design e

Comunicazione Visiva

A. A. 2022/2023

Tesi di Laurea

Coding Lab

Sviluppo dell'identità visiva e della
strategia di comunicazione per un
corso di front end, creative coding e
game development dedicato a designer

RELATORE:

Prof. Luca Cattaneo

CANDIDATO:

Gabriele Giovan Maria Basile

Giulio Borgese

Indice

	Premessa	10
1.0	Scenario	12
	1.1 Coding e Design	14
	1.2 User Interface Design	17
	Storia dello UI design	20
	Good UI Design	28
	1.3 Coding	32
	Web Development	34
	Creative Coding	36
	Linguaggi di sviluppo	40
	Game Development	44
	Storia del game dev	45
	Sviluppo open-source	50
	Roblox	52
	1.4 Figure Professionali	54
	UI designer	55
	Web Developer	58
	UI designer vs. Web developer	60
	Game developer	64

2.0	Casi Studio	68
2.1	UI design	70
	Gumroad	71
	Spotify	72
	Pinterest	73
	Figma	74
	Apple	75
2.2	Creative Coding	76
	Platonic Solid	77
	Mechanical plotter drawings	78
	Sprawl	79
	Plasm 4K shot	80
	We Are Poems	81
2.3	Game Development	82
	Roblox	83
	Unreal Engine	84
	Unity	85
	Minecraft	86
	Fortnite	87
2.4	Code academy	88
	Boolean	89
	Aulab	90
	Udemy	91
	Codecademy	92
	Coursera	93
3.0	Personas	94
3.1	Del settore	
	Trixie - L'anima della festa	96
	Pietro - L'alternativo	98
	Valentina - The Boss	100
	Giacomo - Il Creativo	102

3.2	Curiosi	
	Edoardo - Il Tecnico	104
	Valentina - La Secchiona	106
	Silvio - Il Brontolone	108
	Mario - Apache99X_x	110
4.0	Progetto	112
4.1	Coding Lab	114
4.2	Sviluppo del concept	116
	Linee Guida	117
	Concept	118
4.3	Visual Identity	120
	Naming e logo	122
	Claim	124
	Tipografia	126
	Palette cromatica	128
	Visual	132
	Comunicazione	134
	Discord	136
	Linkedin	138
	YouTube e GitHub	140
	Instagram	144
	Google Form	150
	Supporti	154
	Gadget	168
5.0	Conclusioni	170
6.0	Ringraziamenti	174

Premessa

Internet è ormai un player fondamentale nella nostra quotidianità. Vivere in questa era altamente digitalizzata, vuol dire essere costantemente connessi. Le nuove abitudini digitali offrono alle aziende nuove e numerose opportunità per aumentare i propri profitti, moltiplicando i touchpoint con users e customers. Va da sé che le competenze digitali si trovino ad essere molto importanti nel panorama lavorativo odierno e lo dimostra la costante crescita della richiesta in tal senso. Ancor di più, le aziende, risultano essere attratte da nuove figure professionali ibride che sanno mettere in relazione le competenze progettuali di design con quelle tecniche di sviluppo.

Realizzare un sito web non implica solo la definizione di uno stile grafico. Per far sì che un sito web venga posto in essere, bisogna tradurre quella che è un'interfaccia fatta di testi, immagini, colori, forme e tasti interattivi in un linguaggio che i browser sui nostri computer possano leggere e codificare.

Dunque conoscere i linguaggi di programmazione diventa indispensabile. Per questo motivo si è soliti pensare agli informatici come agli esperti per definizione in questo campo. Questo è corretto, ma solo in parte. Se da un lato i designers (e nella fattispecie gli UI designers) detengono un ampio ventaglio di competenze circa la progettazione

grafica (software, background di formazione) e l'ergonomia cognitiva (abitudini antropologiche, affordance), dall'altro gli informatici (web developers) hanno la capacità e le competenze di padroneggiare una serie di linguaggi indispensabili per lo sviluppo web.

I linguaggi di programmazione si fanno strada nel panorama odierno, così come in passato, in svariati ambiti. Tramite essi è possibile oggi diventare dei veri e propri artisti digitali. Grazie alle incredibili potenzialità di questi linguaggi e l'utilizzo di funzioni e varianti si può dar vita a quello che viene definito codice generativo. Un'interazione uomo-macchina che tramite il principio di randomicità, dettato dalle regole che impartiamo a un computer, è in grado di realizzare opere digitali in quasi completa autonomia. L'artista generativo utilizza appunto come strumento di lavoro i computer, e le applicazioni sono le più svariate. Da un pattern a un solido di forme inorganiche fino ad arrivare a intere installazioni sensoriali capaci di generare esperienze immersive e intriganti.

Un'altra applicazione di questi ultimi è lo sviluppo di videogiochi che vede la sua storia iniziata già negli anni 50 del 1900. Tramite essi e gli odierni motori grafici di gioco è possibile realizzare piccoli e grandi capolavori anche indipendentemente. Le moderne tecnologie permettono inoltre di cimentarsi in questo ambito senza avere conoscenze pregresse in materia. I cosiddetti motori di gioco nocode ampliano le possibilità di sviluppo e di successo a un panorama di persone ampio e variegato.

1.0

Scenario

1.1 Coding e Design

Punti di contatto

Luoghi comuni Quando si parla di coding e design, pare che il senso comune tenda ad associarli a due ambiti completamente avulsi l'uno dall'altro. Da una parte ritroviamo tecnici informatici, ingegneri testardi e nerd del computer, mentre dall'altra, distinti ed eleganti disegnatori con dolcevita alla Steve Jobs e occhiali alla moda. Leggende metropolitane a parte, è corretto considerarli due mondi che in molte occasioni hanno visto, vedono e vedranno collaborare i propri professionisti su tantissimi progetti aventi come minimo comune denominatore tecnologia e digitalizzazione.

Dove avviene il confronto Il momento in cui vengono realizzati progetti per interfacce grafiche (UI) di qualsiasi mezzo digitale (siti web, app, cellulari, tv, computer, software, ecc.) rappresenta quello in cui queste due sfere professionali cooperano alla realizzazione finale. I designer ne progettano l'usabilità e le funzioni che poi i developer provvederanno ad implementare con l'utilizzo del codice. Si tratta di un processo che alterna fasi di ricerca e produzione creativa tipiche del design thinking, a fasi di applicazione tecnica e pensiero computazionale per risolvere con semplicità le sfide progettuali.

Discipline integrate Al giorno d'oggi è importantissimo che design e sviluppo si trovino ad essere due discipline ben integrate¹, al fine di realizzare prodotti digitali che le persone possano utilizzare in maniera

soddisfacente e senza difficoltà. Ciò nonostante, per quanto complementari, il confronto e la collaborazione fra il ruolo dello sviluppatore e quello del designer risultano essere tutt'altro che agevoli.

Le difficoltà operative

Gli sviluppatori riscontrano grosse difficoltà quando i designer pensano a interfacce, sistemi e funzionalità difficili da implementare, aumentando le possibilità di bug dovuti alla complessità dei progetti. Sull'altro versante, invece, i designer si battono strenuamente affinché i loro prodotti arrivino agli utenti esattamente come sono stati pensati, indicando ogni caratteristica nel dettaglio (font, colori, interazioni, architettura dell'informazione, ecc.) e sperando poi che il development team segua pedissequamente ogni loro indicazione².

Nuovi metodi

Questo approccio, simile ad una catena di montaggio, fortunatamente tende ormai ad essere superato a vantaggio di processi più agevoli e inclusivi. In essi designer e sviluppatori condividono il contesto, le tempistiche, il linguaggio e gli

Don't just learn to code, learn to create

Justin Richards | TEDxYouth



obiettivi lungo tutte le fasi del progetto, procedono su percorsi paralleli e scambiandosi costantemente dei feedback. Ciò consente un vicendevole scambio di conoscenze, tale da permettere approcci più consapevoli alla progettazione in termini di fattibilità.

Nuove
opportunità

I dipartimenti HR rivolgono l'attenzione a figure professionali ibride di questo genere e la domanda è in costante crescita. Si tratta di professionisti capaci di integrare una formazione sullo UX/UI design con una più tecnico-informatica, come front end e back end development, permettendo ai teams un nuovo modus operandi, facendo da spalla agli sviluppatori, per progettare e realizzare interfacce di applicazioni o siti web, in modo più semplice e veloce.

-
- 1 <https://medium.com/uxtales/quello-che-designer-e-developer-non-dicono-197a66f967d9>
 - 2 <https://www.xn--mtaweb-3ua.it/web-designer-o-web-developer-differenze>

User Interface Design

Funzionalità a misura di user

PC e dispositivi
mobile

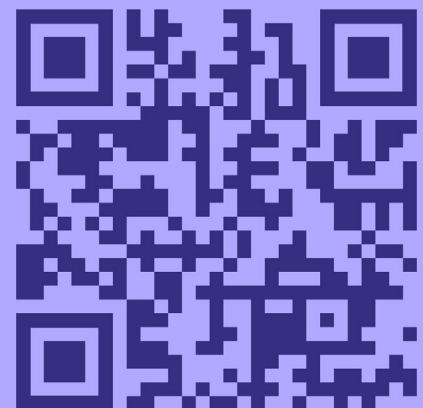
Parlando di computer od informatica, pensiamo subito ad un personal computer con uno schermo, una tastiera ed un mouse, uno strumento rivoluzionario ed utilissimo per una miriade di utilizzi. Diverse attività, proprie dei pc, oggi possono essere svolte sui dispositivi mobile (smartphone, tablet, ecc.), attraverso le app. Queste vengono gestite non con il mouse, ma mediante l'utilizzo dei touchscreen. Tale interazione però non è sempre semplice e immediata, poiché l'interfaccia tra noi e l'oggetto può risultare molto complessa. Ma cosa è un'interfaccia?

L'interfaccia

Un'interfaccia si può definire come il punto di incontro tra due oggetti o due sistemi differenti ed è legata anche al concetto di affordance cognitiva. Se prendiamo come esempio un paio di forbici, i due

Psychology behind UI/UX Design

Harrish Murugesan | TEDxUTA



anelli nei quali inseriamo le dita della mano per utilizzarle, rappresentano l'interfaccia tra noi e l'oggetto. La stessa cosa avviene tra i dispositivi digitali e noi (user) che utilizziamo le interfacce grafiche (graphic user interface/GUI) per interagire con essi.

UI Design

La branca del design che progetta queste interazioni è lo User Interface Design, al quale spesso si fa riferimento con la famigerata sigla UI. Questo settore si occupa proprio della progettazione grafica dell'interfaccia di un qualsiasi prodotto digitale, come per esempio un'app o un sito web. È il processo con cui si progettano le proprietà grafiche e interattive di un'esperienza visuale, assicurando un'interfaccia chiara per la navigazione dell'utente ed anche esteticamente piacevole³.

UX/UI

Lo UI design è un ambito multidisciplinare che coinvolge elementi diversi provenienti dal visual design, lo UX design ed il graphic design. Infatti è facile imbattersi nell'accostamento proprio con lo UX design (user experience design) la cui ricerca precede lo UI che, invece, ne è la diretta conseguenza⁴⁻⁵.

User al centro

Gli user, i loro bisogni e la loro esperienza sono il vero fulcro della progettazione. Nel corso del tempo UX e UI sono andati incontro alle loro esigenze, talvolta educandoli all'interazione con determinati layout e pattern verso i quali oggi hanno acquisito una certa familiarità. Dalla tipografia al colore, ogni aspetto di un'interfaccia digitale ne influenza la soddisfazione. Lo UI design dunque è quella complessa arte di mettere ordine tra tutti questi elementi per guidare l'utente, mantenendo al minimo il suo carico cognitivo e creando al contempo un'esperienza finale unica.

Lo UI design è molto più del semplice rendere tutto bello e piacevole, specialmente all'interno del panorama digitale odierno. Molte aziende riconoscono che uno UI design ben studiato è vitale per costruire la riconoscibilità e la fiducia del brand. I clienti non vogliono imbattersi in siti web poco user-friendly senza un'esperienza digitale ben progettata; se l'aspettano, la pretendono. In assenza di un buon design, gli utenti si rivolgeranno inevitabilmente ad un competitor.

3 <https://careerfoundry.com/en/blog/ui-design/>

4 <https://www.neting.it/blog/differenza-ui-ux-design.html>

5 <https://careerfoundry.com/en/blog/ux-design/the-difference-between-ux-and-ui-design-a-laymans-guide/>

Storia dello UI Design⁶

Una disciplina innovativa

Batch computing

La storia dello UI design comincia dalla metà del secolo scorso, quando vennero alla luce i primi rudimentali computer caratterizzati dalle *batch interface*⁷. Si tratta di UI non interattive, dove lo user specifica tutti i dettagli prima che questi vengano processati e riceve l'output quando il processo è concluso⁸. Queste tecnologie informatiche erano un fenomeno rivoluzionario ma impiegato solo per scopi aziendali o per le tecnologie di difesa militare.

CLI e GUI

Le *GUI* (graphic user interface) erano cosa ancora lontana dall'essere implementate, così i primi computer venivano impiegati da persone appositamente addestrate al loro utilizzo. I primi computer ricevevano le informazioni attraverso le cosiddette *schede perforate* e successivamente le istruzioni venivano impartite attraverso il prompt della *riga di comando* (command line interface o *CLI*). La prima GUI venne implementata nel 1973 sul computer Alto della Xerox Corporation e vi si interagiva con il primo mouse (a 3 tasti)⁹.

Nasce la UX

È durante gli anni 80, quando i Macintosh della Apple ed i Windows della Microsoft furono lanciati sul mercato, che le aziende cominciarono a comprendere il valore di un'interfaccia user-friendly per creare una memorabile user experience. Ed è proprio alla fine di questo decennio che l'ingegnere e psicologo cognitivo **Donald Norman**, unitosi al team della Apple, conierà il termine *user experience* per racchiudere ciò al quale ci si riferiva come *Human Interface Research*¹⁰. Norman capì come gli sforzi



© MITRE Corp.

1956
IBM SAGE
COMMAND LINE INTERFACE

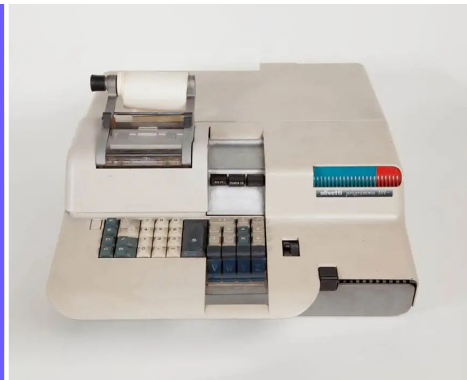
Computer per la difesa aerea militare



© Joi Ito

1962
PDP-1
COMMAND LINE INTERFACE

Esegui Spacewar, uno dei primi videogiochi



© Museo NST, Milano

1965
PROGRAMMA 101
COMMAND LINE INTERFACE

Primo "personal computer"



© Maksym Kozlenko

1973
IBM SAGE
WIMP GRAPHIC UI

Nuovo standard di GUI

dell'azienda dovevano concentrarsi verso una progettazione antropocentrica, unendo due ambiti che ancora non comunicavano: tecnologia e psicologia.

L'avvento del colore

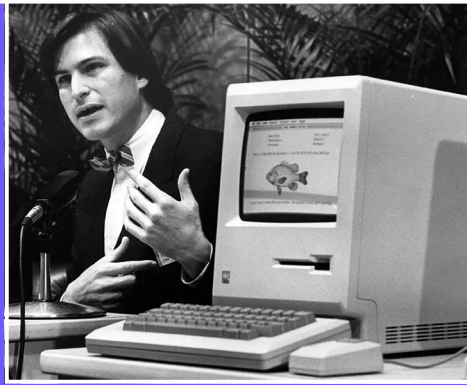
Le interfacce venivano così aggiornate costantemente insieme ai nuovi prodotti lanciati sul mercato in rapida successione, partendo da GUI monocromatiche, fino ad approdare al mondo del colore, momento che la Apple ha celebrato con l'adozione dell'iconico logo della mela colorata.



Il paradigma WIMP

Nel 1980 **Merzouga Wilberts** conia l'acronimo **WIMP**¹¹ per indicare un nuovo standard di interfaccia caratterizzato da:

- **finestre** (*windows*) che eseguono programmi isolati dagli altri eseguiti su altre finestre;
- **icone** (*icons*) come shortcut per i programmi;
- **menù** (*menu*) come sistemi testuali di selezione di task;
- **puntatore** (*pointer*) che descrive su display i movimenti dell'utente mentre interagisce con un device di controllo.

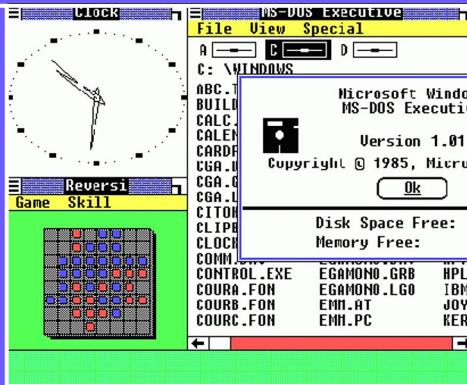


© Terry Schmitt

1984 MACINTOSH

WIMP GRAPHIC UI

Funzione finder e nuova icona del cestino.

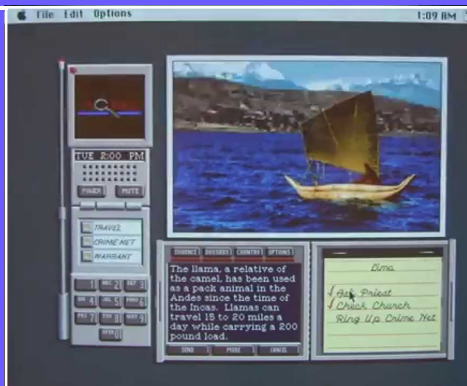


© Microsoft

1985 WINDOWS 1.01

WIMP GRAPHIC UI

Primo display 8-bit e barra delle applicazioni

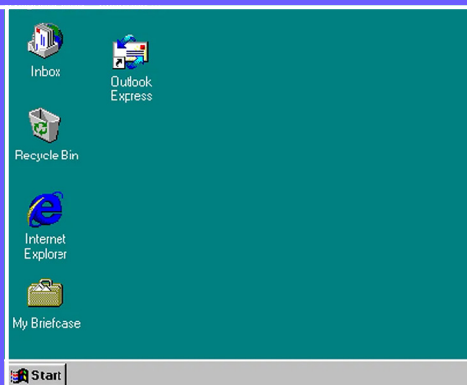


© Apple

1987 MACINTOSH II

WIMP GRAPHIC UI

Primo Apple a colori



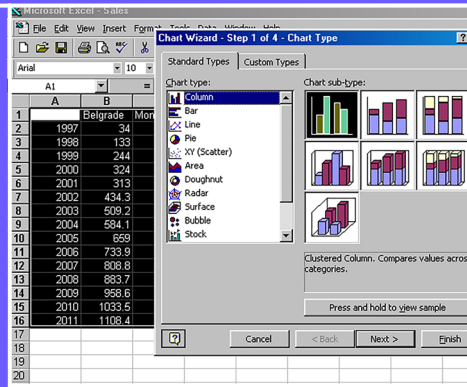
© Microsoft

1995 WINDOWS 95

WIMP GRAPHIC UI

Aggiunto il tasto start ed internet explorer

- Sviluppo web Nel 1992 va online il primo sito web, caratterizzato da un semplice testo occasionalmente colorato col blu predefinito dell'ipertesto. Altri elementi dell'interfaccia utente sono stati gradualmente introdotti nelle esperienze digitali, fino a quando il plug-in *Flash* ha cambiato le regole del gioco. Per la prima volta gli sviluppatori potevano creare qualsiasi forma e animazione desiderassero e tutto con un unico strumento, rendendo i siti web molto più coinvolgenti.
- Microsoft Office Con il lancio di Microsoft Windows 98 venne rilasciata la prima versione del pacchetto software Office, ciò permise la condivisione di file tra utenti e dunque un'esternalità di rete che ha avvantaggiato l'azienda sul mercato.
- Game development Importantissimo menzionare anche il ruolo dei videogames che hanno costituito un perfetto campo di prova e sperimentazione per le UI grafiche e digitali, come ad esempio le Playstation console e l'innovativo Nintendo Wii.
- Nuova interazione: i touchscreen Il lancio del il primo iPhone nel 2007, ha inaugurato un'età d'oro per lo UI design. Le funzionalità *touchscreen* sempre più sofisticate, hanno fatto sì che gli utenti non guardassero più i loro cellulari (adesso *smartphone*) allo stesso modo. Si tratta di un cambiamento di prospettiva che portò i progettisti delle interfacce ad assumere improvvisamente un nuovo status.
- Interfacce responsive Nel 2010, il web designer **Ethan Marcotte** introduce per la prima volta un concetto che oggi pare assolutamente scontato: la *responsiveness*¹² dei siti web. Marcotte propose che un contenuto fruibile da desktop potesse essere visualizzato su uno schermo diverso con un layout adatto ed ottimizzato,



© Microsoft

1998

WINDOWS 98

WIMP GRAPHIC UI

Primo pacchetto di Office



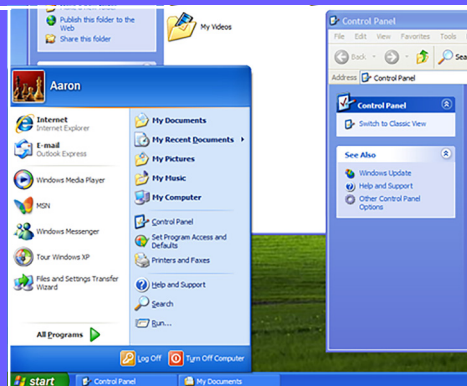
© Apple

2000

MAC OS X

WIMP GRAPHIC UI

Nuova GUI con ombre sui bordi ed effetti animati interattivi



© Microsoft

2001

WINDOWS XP

WIMP GRAPHIC UI

GUI aggiornata ai livelli di Mac OS X



© Nintendo

2006

NINTENDO WII

WIRELESS DEVICE E GUI

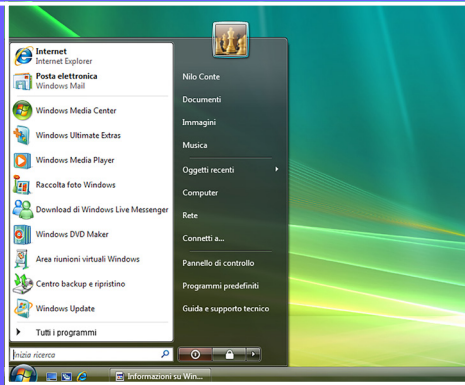
Nuova interazione di wireless control

piuttosto che avere siti web esclusivamente pensati per dispositivi mobile. Realizzare un sito web che sia responsive, che ora è una parte vitale ed esigenza primaria del design dell'interfaccia utente.

Disciplina in
espansione

Oggi lo UI design è un campo in continua espansione ed evoluzione verso pattern sempre più semplificati, che interessa i progetti più diversi designer più esperti dovrebbero avere alcune competenze di UI nel loro arsenale.

-
- 6 <https://careerfoundry.com/en/blog/ui-design/>
 - 7 <https://medium.theuxblog.com/a-short-history-of-computer-user-interface-design-29a916e5c2f5>
 - 8 https://www.freedesktop.org/wiki/User_interface/
 - 9 <https://youtu.be/U10y4X5Ni8Y>
 - 10 <https://promosweb22.com/user-experience-i-6-principi-fondamentali-di-donald-norman/>
 - 11 <https://www.techopedia.com/definition/4687/windows-icons-menus-and-pointing-device-wimp>
 - 12 <https://alistapart.com/article/responsive-web-design/>



© Microsoft

2007
WINDOWS VISTA
WIMP GRAPHIC UI

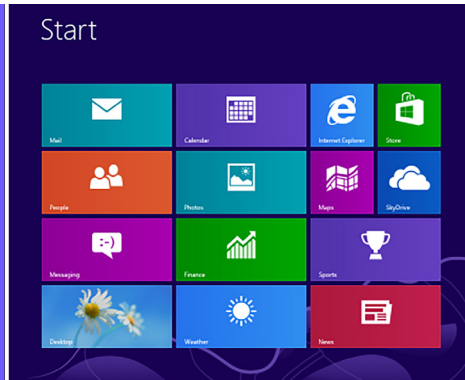
Vetromorfismo e rendering 3D della GUI



© Apple

2007
iPHONE 1° GENERAZIONE
TOUCHSCREEN GRAPHIC UI

Primo smartphone con GUI scheumorfica



© Microsoft

2012
WINDOWS 8
WIMP GRAPHIC UI

GUI minimale, poi ottimizzata su Windows 10



© Apple

2013
IPHONE iOS 7
TOUCHSCREEN GRAPHIC UI

Nuova GUI flat

Good UI Design

Le 10 euristiche di Nielsen

Nel tempo sono state diverse le ricerche teoriche che si sono susseguite per individuare e definire le regole precise per progettare l'interazione utente. A questo scopo, l'informatico ed imprenditore danese, **Jakob Nielsen** (co-founder del NN/group insieme a D. Norman) ha definito dieci *euristiche*¹³ per una corretta progettazione di interaction design. Si chiamano così perché si pongono come delle scelte convenienti e raccomandabili e non come regole assolute¹⁴.

1 STATO DELLA NAVIGAZIONE

La zona di un sito o app che si sta visitando, deve essere sempre chiara e riconoscibile. Allo user vanno forniti i giusti riferimenti grafici ed una chiara gerarchia dell'informazione.

2 NAVIGAZIONE LIBERA

La navigazione dell'utente deve essere fluida, agevole e svincolata da limitazioni non necessarie. Bisogna che lo user sia abilitato a muoversi avanti e indietro, annullare e ripetere azioni senza difficoltà.

3 SISTEMA VS. MONDO REALE

Lo UI design deve avvalersi di uno stile di comunicazione congruo a quello che il target designato utilizza nel mondo reale, in modo da evitare incomprensioni ed ambiguità, e modulato sul tone of voice che apprezza maggiormente.

4 COERENZA E STANDARDIZZAZIONE

L'adeguamento a certi standard grafici e terminologici devono essere assolutamente contemplati nella definizione dello UI design. Ciò permette l'utilizzo di elementi condivisi per una più facile ed intuitiva comprensione, anche su dispositivi e OS diversi.

5 PREVENZIONE DEGLI ERRORI

È obbligo della UI, prevenire ambiguità e tenere lo user lontano dagli errori ancora prima di risolverli. Mettere lo user davanti alle scelte corrette agevola l'esperienza abbassando il suo carico cognitivo.

6 MESSAGGI USER-FRIENDLY

Al contrario di quanto detto, se l'errore dovesse verificarsi, ogni messaggio per lo user (es.: pop-up) dovrebbe sempre evitare tecnicismi di difficile comprensione. Se prendessimo ad esempio un messaggio di errore, oltre ad essere espresso chiaramente, questo dovrebbe sempre indicare una soluzione alla portata dello user.

Nielsen's 10 heuristics of usability

Matt Borchert



Jakob Nielsen's 10 Heuristics for User Interface Design

<https://www.nngroup.com/articles/ten-usability-heuristics/>



7 FLESSIBILITÀ ED EFFICENZA

Rendere più veloci e immediate certe azioni sulla UI, agevola la navigazione. Una certa efficienza è raggiungibile mediante l'uso di scorciatoie, tasti funzione e comandi nascosti, oltre alla possibilità di adattare la UI in base all'utente in questione.

8 RICONOSCERE VS. RICORDARE

Come già detto in precedenza, lo UI design ha tra i suoi obiettivi primari quello di ridurre al minimo il carico cognitivo dello user. Dunque lo UI designer dovrà definire informazioni di intuitiva riconoscibilità, essenziali e rilevanti. Si tratta di un pattern cognitivo più rapido, rispetto ad un macchinoso processo di richiamo alla memoria. Un esempio può essere quello di prediligere una scelta da un elenco di risposte, in alternativa alla richiesta di compilazione libera di una casella di testo.

9 MINIMALISMO

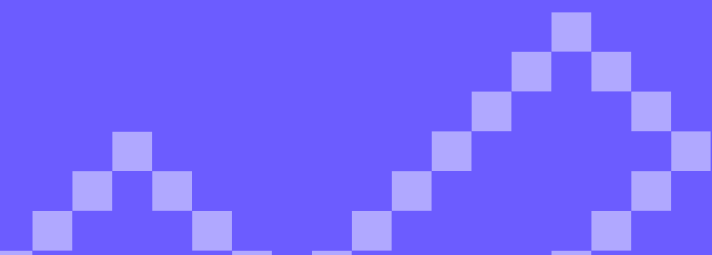
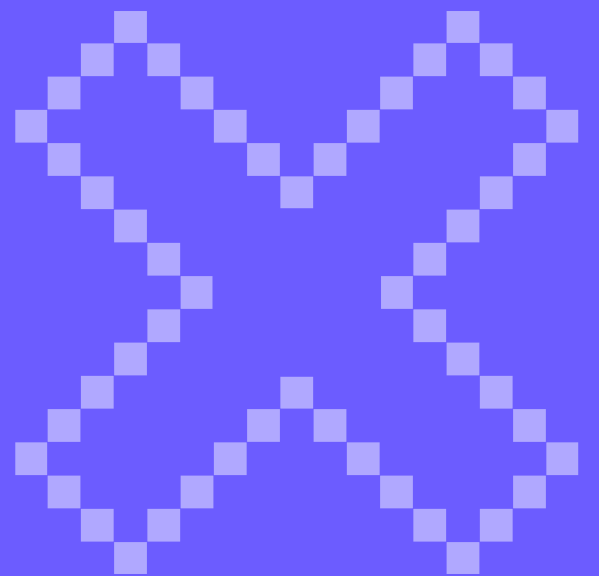
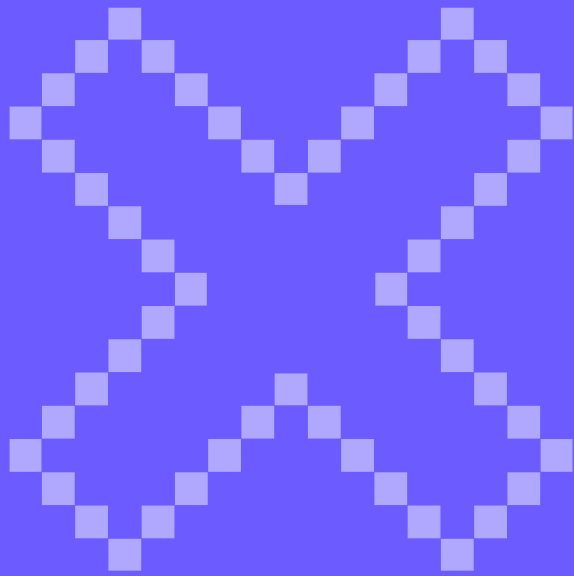
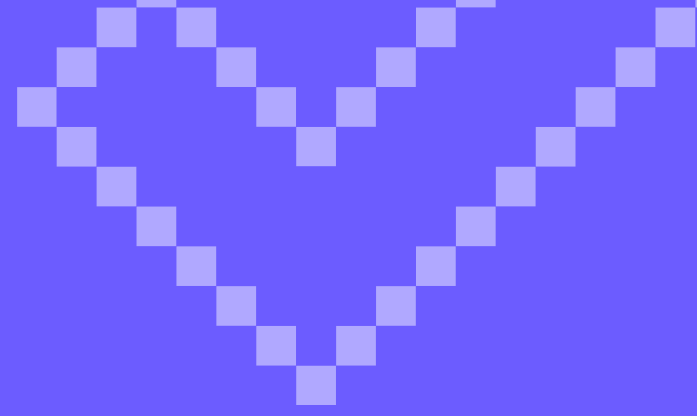
La fisiologica ridotta capacità di attenzione dell'essere umano, rende necessario evitare tutte le informazioni superflue per non confondere l'utente. Una UI essenziale, priva di ambiguità, permetterà l'agevole svolgimento di tutte le attività più frequenti tipiche della navigazione.

10 AIUTO ALLA NAVIGAZIONE

Un 'esperienza di navigazione, anche se chiara e ben strutturata, dovrebbe essere sempre provvista di una guida a margine che elenchi una serie spiegazioni utili alla corretta fruizione (es.: FAQ).

13 https://www.researchgate.net/profile/Niccolo-Donati/publication/350813865_La_filosofia_dell'usabilita/links/60743ed9a6fdcc5f779d0bdc/La-filosofia-dellusabilita.pdf

14 <https://www.nngroup.com/articles/ten-usability-heuristics/>



Coding

La lingua dei computer

Definizione

La parola *coding* indica letteralmente l'attività dello scrivere codice, ed ha un significato quasi assimilabile al programmare. "Quasi, perché programmare può significare anche analizzare, progettare, verificare, integrare un codice sorgente, mentre *coding* fa riferimento solo alla scrittura del codice" (Dietro il Coding, Stefano Penge, 2016¹⁵).

A cosa serve fare coding

Lo scopo del coding è quello di impartire ad una macchina di calcolo una serie di comandi in sequenza. Occorre però conoscere una grammatica adatta perché questi vengano ben interpretati. Questa è rappresentata dai linguaggi di programmazione, i quali vengono utilizzati all'interno di software comunemente chiamati *editor* testuali, ma che il linguaggio tecnico definisce *Ambienti Integrati per lo Sviluppo* (IDE).

Computational thinking

Alla base di questa attività vi è il *pensiero computazionale*, diretto contraltare del *design thinking* per la progettazione creativa, che è definito come "un processo di formulazione di problemi e di soluzioni in una forma che sia eseguibile da un agente che processa informazioni" (Jeannette Wing, Communications of the ACM, vol. 49/3, Marzo 2006¹⁶). Il coding rappresenta anche un valido strumento di apprendimento che, infatti, sta prendendo sempre più spazio nella definizione di innovativi piani didattici sin dalla scuola primaria. Ai bambini non insegnano certo la complessità della

programmazione testuale degli editor, bensì di altri sistemi di programmazione visiva (es.: scratch¹⁷) che attraverso blocchi, colori, forme ed un'interazione tipicamente drag-and-drop, permettono di rendere concreti i concetti del pensiero¹⁸.

Competenze
del futuro

Oltre ad evolvere il computational thinking, tra gli scopi impliciti della didattica del coding vi è quello di educare e preparare le nuove generazioni ai lavori dell'oggi e del domani. Scrivere il codice è la competenza base di qualsiasi lavoro nel settore del tech e dell'informatica ed oltre ad una preparazione scolastica ed accademica, molti professionisti riescono a costruire le proprie competenze da autodidatti. Molti web developer ne sono un esempio.

15 https://it.wikibooks.org/wiki/Dietro_il_coding

16 <https://dl.acm.org/doi/fullHtml/10.1145/1118178.1118215>

17 www.scratch.mit.edu

18 it.pearson.com/docenti/primaria/classe-dinamica/che-cosa-pensiero-computazionale.html

Learn to create art using code

Yuhan Liu | TEDxRansomEvergladesSchool



Web Development

Casa c'è dietro l'interfaccia?

Il web development

Con web development ci si riferisce in generale alle competenze di codice per lo sviluppo di siti web ed app sia su internet sia su network privati (intranet). Tra i processi del web development, i principali sono il web design, lo sviluppo lato client (*client-side*) e lato server (*server-side*)¹⁹. In senso più ampio comprende tutte le azioni volte alla costruzione, aggiornamento, manutenzione e gestione di un sito o applicazione web per assicurarne le giuste performance e UX. Può anche includere una serie di strategie per assicurare una buona indicizzazione tra i risultati dei motori di ricerca, che fanno capo alla cosiddetta *search engine optimization*²⁰ (SEO).

Le due componenti del web

I siti web ed app, si compongono di due sezioni principali e complementari: il *front end* ed il *back end*. La prima descrive la parte che è visibile all'utente, l'interfaccia (UI) con la quale egli può interagire, mentre la seconda indica invece quella che permette l'effettivo funzionamento di queste interazioni. In estrema analisi è possibile affermare che un prodotto digitale per il web non può fare a meno dell'una o dell'altra componente.

I professionisti

I professionisti del settore sono gli sviluppatori web, o *web developers* (web devs), che si suddividono in tre categorie: front end devs, back end devs e full stack devs.

¹⁹ <https://www.techopedia.com/definition/23889/web-development>

²⁰ <https://www.studiosamo.it/seo/>

SVILUPPO FRONT END

Nel campo informatico dello sviluppo web/app client-side, lo sviluppo front end serve a gestire il sistema a cui lo user accede e con il quale interagisce direttamente durante la fruizione di un prodotto digitale²¹ e corrisponde all'interfaccia utente.

Funzionamento Per mezzo dell'interazione degli utenti, il front end invia richieste input, ricevendo indietro data output elaborati dal sistema back end oppure informazioni dell'host, e di nuovo disponibili tramite il sistema front end.

I linguaggi La struttura di questo sistema viene sviluppata tipicamente attraverso una parte HTML che ne definisce i contenuti, una parte CSS che ne definisce l'aspetto estetico ed una parte JavaScript che aggiunge interazione al complesso.

SVILUPPO BACK END

Il back end development comprende tutta una serie di operazioni volte alla creazione di un sistema per la ricezione, la raccolta e gestione di dati in entrata e uscita di un sistema front end²².

I linguaggi La struttura di questo sistema viene gestita tipicamente attraverso linguaggi come PHP e Java.

Progettazione agevolata Il fatto che i sistemi di front end e back end siano separati, soprattutto quando si lavora a sviluppo e manutenzione su più livelli, permette processi computazionali e di sviluppo facilitati.

²¹ <https://www.techopedia.com/definition/3799/front-end-system>

²² <https://www.techopedia.com/definition/1405/back-end-system>

Creative Coding

Un nuovo modo di generare arte

Creazioni non convenzionali

Tra le modalità che l'uomo nei secoli ha utilizzato per la propria produzione espressiva ritroviamo la scultura, la pittura e la musica, tutte forme proprie dell'arte. L'era digitale dei computer ha portato alla nascita di una nuova forma espressiva "sviluppata" in modo non convenzionale: il *creative coding*.

Connubio uomo-macchina

Si tratta proprio di una realizzazione espressiva e non di una funzionale, a metà strada fra una produzione creativa grafico-artistica e lo sviluppo informatico. In origine viene pensata dall'uomo, ma viene poi eseguita da un computer che ha elaborato le sue richieste programmate²³.

Arte digitale generativa

Volendo dunque parlare di creative coding, possiamo riferirci a questa pratica con generative art, in cui le componenti coinvolte sono tre²⁴:

- **l'artista** generativo, l'uomo, che scrive il codice e giudica il risultato finale;
- **l'algoritmo**, che codifica le regole per generare l'opera;
- **il sistema autonomo**, che esegue e crea l'opera.

Il procedimento

Dunque serve un software che viene istruito e guidato da alcuni vincoli artistici, in questo caso variabili, dettati da un codice scritto da uno sviluppatore. Sulla base di quelle indicazioni, il software genererà opere artistiche in completa autonomia e in modo imprevedibile. Allo sviluppatore è demandato il compito di valutare l'output dell'esecuzione ed a seconda del feedback deciderà

se approvare o modificare i parametri per ricercare un risultato più soddisfacente.

Una definizione d'autore

L'artista **Philip Galanter**²⁵ descrive la generative art come di seguito:

"Il termine arte generativa si riferisce a qualsiasi pratica artistica nella quale l'artista crea un processo, come insieme di regole di un linguaggio naturale, un programma informatico, una macchina, o un meccanismo di altro tipo, che viene poi avviato con un certo grado di autonomia contribuendo a creare o avendo come effetto un'opera finita."

(What is Generative Art? Complexity Theory as a Context for Art Theory²⁶)

Output imprevedibili

Queste creazioni restituiscono output imprevedibili: forme semplici o complesse, statiche o dinamiche, a partire dall'esecuzione di un codice, definito appunto generativo, per ottenere risultati imprevedibili. Rappresenta un campo inesplorato, le cui applicazioni sono tutte da scoprire.

Creative Coding = unexplored territories

Tim Rodenbröker | TEDxUniPaderborn



Un altro artista, **Michael Hansmeyer**²⁷, dice così:

*“Il processo di progettazione raggiunge un equilibrio tra l’atteso e l’inaspettato, tra controllo e rinuncia. Sebbene i processi siano deterministici, i risultati non sono prevedibili. Il computer acquisisce il potere di sorprenderci.”*²⁸

Con la Generative Art le possibilità sono pressoché infinite. La combinazione di arte, design e programmazione dà spazio a possibilità e soluzioni inedite ed originali per i contesti più vari, sia che essi siano puramente estetici ed espressivi, sia che essi siano a scopo interattivo per gli utenti di un mercato di consumo. Può essere utilizzata per creare pattern dalle innumerevoli fantasie, per generare oggetti di design dalle forme astratte o ancora per creare esperienze sensoriali, visive e uditive. Diversi brand, infatti, potrebbero usare questa opportunità per regalare inconsuete e memorabili esperienze ai propri customers. Può altresì essere impiegato in mostre ed installazioni artistiche, caratterizzate da innovative interazioni con il pubblico. Altro esempio è la possibilità di impiegarlo in eventi musicali, generando in tempo reale visual a partire da suoni e tracce audio, un po’ come avveniva con i frattali animati delle vecchie edizioni di windows media player.

23 <https://www.disko-agency.it/riflessioni/technical/creative-coding-una-tendenza/>

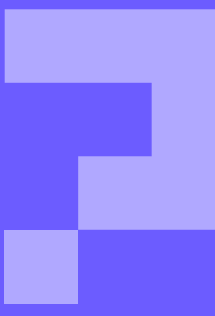
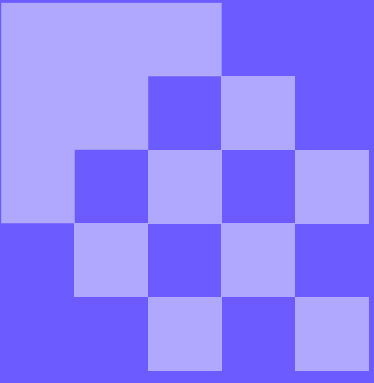
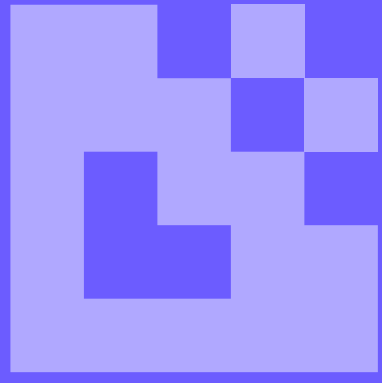
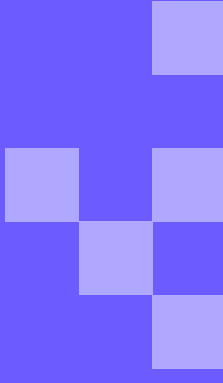
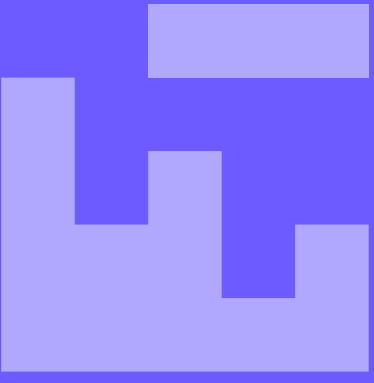
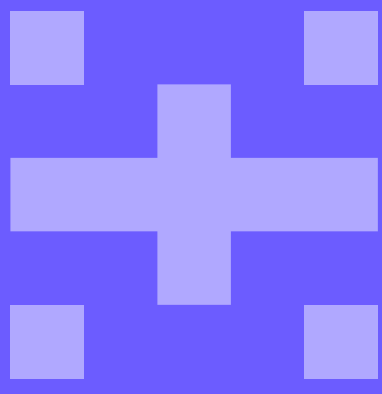
24 <https://www.yatta.xyz/generative-art-e-ai-creativa/>

25 <http://philipgalanter.com/about>

26 http://philipgalanter.com/downloads/ga2003_what_is_genart.pdf

27 <https://www.michael-hansmeyer.com/press>

28 <https://aiartists.org/generative-art-design>



Linguaggi di sviluppo

Parlare con i computer

Lessico digitale Giunti a questo punto, è tempo di spiegare la “grammatica” delle parole che i computer sono in grado di comprendere per dettare loro le informazioni necessarie. Quella grammatica è costituita dai linguaggi di programmazione che, così come un linguaggio umano, possiedono un lessico, una sintassi e una semantica ben definita.

Cosa sono? I *linguaggi di sviluppo*, o *di programmazione*, sono strumenti informatici utilizzati per impartire ad un dispositivo elettronico, istruzioni e ordini (*input*) da trasformare in dati e azioni (*output*) . In generale, tutti i linguaggi condividono alcuni elementi comuni: istruzioni, variabili e costanti, espressioni, tipi di dati, strutture dati, strutture di controllo e sottoprogrammi²⁹.

Perché utilizzarli? I linguaggi permettono di trasformare idee e concetti umani in *codici macchina*, comprensibili da computer e device elettronici. Dato che il linguaggio macchina è basato sul sistema binario, i linguaggi si frappongono in questa comunicazione uomo-macchina che altrimenti sarebbe troppo lunga e complicata. L'assenza di questi determinerebbe processi progettuali per nulla scalabili, se pensiamo a lunghi script ed alla necessità di doverli analizzare per scovare eventuali *bug* al loro interno.

Come utilizzarli? Partendo da un problema, per arrivare a una soluzione è necessario analizzare dettagliatamente la situazione per trovare una strategia risolutiva. Quando questa strategia è definita, bisogna scrivere

le istruzioni da impartire all'esecutore (computer) che deve compierle. L'insieme di queste istruzioni, che permettono di risolvere il problema, si chiama algoritmo.

Gli IDE

A supporto delle attività di coding, i developer utilizzano software chiamati *IDE (Integrated Development Environment)* i quali sono composti da:

- un **editor**;
- un **interprete**;
- un **debugger**.

Software come Atom, Vs Code, Dreamweaver sono editor che facilitano le azioni di coding con suggerimenti visivi suggeriti durante la scrittura. Gli interpreti, o compilatori, sono dotati di debugger che aiutano tantissimo, eseguendo localmente le istruzioni e testandone le funzioni alla ricerca di errori da dover risolvere³⁰.

Diversi linguaggi

Oggi esistono moltissimi linguaggi per lo sviluppo che è possibile distinguere in base all'utilizzo, ma principalmente si dividono in alcune categorie: linguaggi di basso livello, di alto livello, per la programmazione front end e quella back end.

I linguaggi di basso livello

I *linguaggi assembly* sono quelli comunemente definiti di basso livello, molto vicini al linguaggio macchina, che tipicamente contengono le istruzioni che il processore può eseguire, ma espresse con nomi comprensibili agli umani³¹. Queste indicazioni sono stringhe di bit (solitamente della lunghezza di 1 o 2 ma anche 4 bytes) che vengono tradotte in linguaggio macchina da programmi "assemblatori" chiamati *assembler*. In generale si tratta di linguaggi le cui funzioni (poco integrate) vanno inserite bit per bit.

I linguaggi di alto livello	I linguaggi definiti di alto livello possiedono funzioni già integrate senza doverle ogni volta specificare nel codice sorgente. Tra questi possiamo ricordare i più importanti come Python , JavaScript , Java e C# che definiscono lo standard maggiormente diffuso in quanto a programmazione software.
I linguaggi di front end	Nello standard del front-end development rientrano i linguaggi HTML , CSS ed il già citato JavaScript .
HTML	L' HTML (<i>HyperText Markup Language</i>) viene utilizzato per disporre i contenuti testuali e visivi di un'interfaccia front end mediante tag di formattazione, che comunicano alla macchina la tipologia del contenuto da codificare e visualizzare (titolo, paragrafo, immagine, link, ecc.).
CSS	Il CSS (<i>Cascading Style Sheet</i>) serve ad indicare come debbano essere visualizzati i contenuti html, applicando delle regole ai tag (colore, dimensione, allineamento, contorni, ecc.).
JavaScript	Il linguaggio JavaScript (abbr. <i>js</i>), da non confondere con Java, aggiunge caratteristiche interattive all'interfaccia front end in base a determinati eventi (es.: click dell'utente, caricamento, ecc.) e variabili. JavaScript serve ad eseguire logicamente le azioni di un copione, uno script appunto.
I linguaggi di back end	Per quanto riguarda i linguaggi di sviluppo back end, possiamo annoverare nuovamente Java e C# , oltre a PHP e .NET ³² .
PHP	Il PHP (da <i>Personal Home Page</i>) è il più conosciuto ed utilizzato durante la creazione di pagine web dinamiche.

Java	Java è un linguaggio <i>object-oriented</i> , studiato per contenere strumenti e librerie per il networking ed essere indipendente dalla piattaforma hardware di esecuzione.
Microsoft .NET	Microsoft .NET contiene una serie di strumenti e funzionalità così integrate da avere una sintassi riconosciuta come standard ISO ed in cui rientra anche C# (object-oriented).
L'aiuto dei framework	Per alcuni linguaggi esistono i <i>framework</i> , vale a dire delle strutture di base ready-to-use che, se utilizzate, velocizzano gli sviluppi più complessi ed agevolano il lavoro dei developer.
Informatica in evoluzione	Dalla fabbricazione del primo computer ad oggi l'informatica ha fatto passi avanti che in passato mai nessuno avrebbe immaginato, arrivando a programmare le tecnologie alla base dei nuovi sistemi AI (<i>artificial intelligence</i>) di <i>machine learning</i> e <i>deep learning</i> . Questi nuovi paradigmi aprono le porte a tantissime nuove possibilità per applicazioni e calcoli complessi che porteranno ad un ulteriore sviluppo nel campo della programmazione.

29 https://it.wikipedia.org/wiki/Linguaggio_di_programmazione

30 <https://www.redhat.com/it/topics/middleware/what-is-ide>

31 https://it.wikipedia.org/wiki/Linguaggio_assembly

32 <https://www.techyon.it/articoli/tool-back-end-developer.html#idx0>

Game Development

Arte giocabile

Cos'è?

Il *game development* è quell'ambito che, appunto, si occupa dello sviluppo di videogiochi. La ricerca è rivolta alle meccaniche di gioco, al design dei livelli, alle ricompense ed ovviamente all'esperienza e al coinvolgimento del giocatore.

Passato e presente

Da molti considerato come un'arte, il game development coinvolge aree d'interesse e figure professionali diverse fra loro. Una volta la figura dello sviluppatore era l'unica ad occuparsi del progetto videoludico, ricoprendo i ruoli di programmatore, designer e animatore. Oggi a fare parte del team di sviluppo sono designers, artisti illustratori, programmatori, ingegneri sonori, tester e molti altri.

Pubblico esigente

Oggi l'approccio è differente per vari motivi e primo fra tutti il fatto che il pubblico è diventato parecchio esigente. Per non parlare del fatto che i videogiochi di oggi dispongono di dinamiche parecchio complesse. Ciononostante continuano ad avvicinarsi alcuni successi di sviluppatori indipendenti, che fanno capo alla scena *indie* (indipendente) dei videogame.

33 <https://dbggameacademy.it/cose-il-game-design/>

Storia del game dev³⁴⁻³⁵

Dal tubo catodico al VR

Le origini

L'attribuzione dell'origine dei videogiochi è ancora oggi oggetto di dibattito. Alcune fonti la attribuiscono ai due fisici **T. T. Goldsmith Jr.** ed **E. R. Mann** che nel 1947 depositarono un brevetto per la creazione di un *dispositivo a tubo catodico*³⁶. Secondo altre fonti, l'origine dei videogiochi è da attribuire invece all'inventore **Josef Kates** che nel 1950 creò Bertie the Brain, un enorme macchinario con il quale si poteva giocare al comunissimo tris. Questi però non possono essere davvero considerati videogiochi³⁷. Per questo motivo le origini vanno cercate altrove.

PRIMA GENERAZIONE

Il primo videogioco

L'invenzione dei videogiochi si attribuisce a **William Higinbotham**³⁸, fisico che nel 1958 all'interno dei suoi laboratori inventò Tennis for Two e che contribuì alla creazione della prima bomba nucleare. Ma è nel 1961 che l'informatico statunitense **Steve Russel** realizzò *Spacewar*³⁶, un videogioco pensato per essere utilizzato da chiunque fosse in possesso di un computer. Questo gioco, infatti, fu fatto girare sul *modello PDP-1*.

Spacewar

SECONDA GENERAZIONE - SISTEMI 8-BIT

Il primo cabinato

Una volta lanciati sul mercato, i videogiochi divennero un vero e proprio fenomeno culturale popolare. Nel 1971 venne alla luce *Galaxy Game*, il primo videogioco cabinato da sala giochi ispirato a Spacewar. Ideata nel 1968, e lanciata nel 1972 fu messa sul mercato la prima *Magnavox Odissey* dell'ingegnere **Ralph Baer**. Una console collegata a un televisore che, tramite manopole rudimentali,

Magnavox

Channel F
programmabile

permetteva di controllare due blocchi che andavano da una parte all'altra dello schermo giocando così a tennis. In seguito nel 1976 le console sbarcarono nelle case dei videogiocatori console come l'*Atari 2600*, la *Bally Astrocade* e la *Channel F* di **Fairchild** (programmabile), che avevano nuovi controller più ergonomici e i cui giochi erano contenuti nelle innovative cartucce rimovibili.

TERZA GENERAZIONE

Il Giappone

La terza generazione fa capo alla produzione giapponese. Nel 1978 la **Taito** lanciò sul mercato il primo videogioco ad introdurre una trama, *Space Invaders*, e due anni dopo la **Namco** rilasciò *Pac-Man*. Due pietre miliari del videogame e la cultura pop vintage. Nel 1983 fecero la loro comparsa alcune tra le più iconiche console di sempre come *NES*, *Atari 7800* ed il *Commodore 64*. Anni d'oro della **Nintendo** che costruì un impero con capolavori come *Zelda*, *Mario Bros* e *Metroid*.

Nintendo



FIG. 19 ▲
Pac-Man
(©Namco)

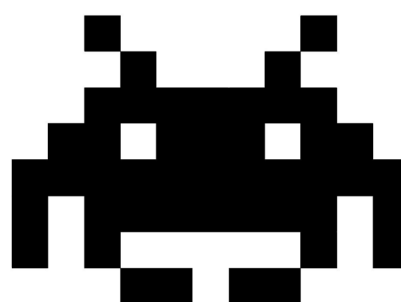


FIG. 20 ▲
Space
Invaders
(©Taito)

QUARTA GENERAZIONE - 16 BIT

Le console distribuite a partire dal 1987 diedero inizio alla quarta generazione dei 16-bit, aumentando capacità e velocità dei processori e migliorando le sezioni grafiche e audio. Queste console tra cui il *Game-boy*, *Master System* e *Atari Lynx* divennero oggetto di vero e proprio culto. Sono anni in cui si consolidano famose case di produzione come **Konami**, **Capcom**, **SEGA**, **Ubisoft** ed **Electronic Arts**.

QUINTA GENERAZIONE - 32 BIT

II 3D

La quinta generazione è caratterizzata dall'introduzione dei primi giochi di avventura 3D, che grazie al sistema a 32-bit ed ai *CD-ROM*, permettevano una maggiore durata, oltre alla possibilità di salvare i progressi di gioco. Sono gli anni di uscite come *Wolfenstein 3D* e *Quake*, primo videogame giocabile on-line, entrambi della **ID Software**. Altre importanti uscite furono *Final Fantasy*, *Metal Gear Solid*, *Resident Evil* e *Tekken*. Protagonista indiscussa di questi anni fu la *PlayStation one* del 1994, seguita a fatica dal *Nintendo 64*.

PS One



◀ FIG. 21
Tekken 1
(©Namco)

SESTA GENERAZIONE - 128 BIT

Nel 1998 venne rilasciata la console di casa **SEGA**, *Dreamcast*. Nasce la sesta generazione dei videogames con sistema a 128-bit. Nascono i primi giochi *MMORPG* (*Massive Multiplayer Online Role-Playing Game*), in cui migliaia di utenti si incontrano online. Nel 2000 viene lanciata la longeva *PlayStation 2*, prodotta per 12 anni, mentre nel 2001 **Microsoft** fece il suo ingresso in campo con *Xbox*. Entrambe le console supportavano il *DVD*, e il collegamento Internet. Un anno dopo furono raggiunte dal *Nintendo GameCube* che però non riscosse molta popolarità.

PS2 vs. Xbox

SETTIMA GENERAZIONE

Il 2004 è l'anno di rivalsea per Nintendo che introduce una console portatile di grande successo, il *Nintendo DS*: questo disponeva di due schermi LCD di cui uno tattile. L'anno seguente la **Sony** risponde al competitor con la *PSP*, ma il mercato rimane a favore della Nintendo che nel 2006 lancia la console *Nintendo Wii*, una delle più famose di sempre per l'innovativa modalità di controllo remoto. Così Microsoft e Sony tentano di eguagliare l'impresa con *Kinect* e *PS Move*, ma non riscuotono lo stesso successo. Nel 2005 Microsoft mette in commercio

Device portatili

Nintendo Wii

PS3 vs.
Xbox 360



◀ FIG. 22
PlayStation2
(©Sony)

l'Xbox 360, competitiva ed economica, e l'anno seguente Sony la PlayStation 3, economicamente meno accessibile.

OTTAVA GENERAZIONE

PS4 vs. Xbox ONE

Nel 2011 Sony rilascia *PS Vita* con una potenza hardware a metà tra PS2 e PS3, mentre Nintendo opta per una console 3D con *Nintendo 3DS*. Nel 2013 Microsoft sorprende il mercato con *Xbox ONE*, ma contemporaneamente Sony rilascia la PS4 e la PS4 Pro nel 2016, tecnicamente superiori alla rivale. Vengono rilasciati giochi ormai cult come *GTA 5*, *Red Dead Redemption 2* e *Animal Crossing: New Horizon*.

NONA GENERAZIONE

Nintendo Switch

Ancora oggi le tre maggiori case di produzione rimangono le stesse. La Nintendo in particolar modo ha visto i propri profitti schizzare alle stelle con il rilascio di *Nintendo Switch*. Le nuove console di ultima generazione promettono esperienze sempre più realistiche migliorando l'hardware e grazie a tecnologie come il *ray tracing*. Al momento gli unici contendenti del mercato sono Sony con *PS5* e Microsoft con *Xbox Serie S*. La nuova frontiera dei videogiochi è il *cross-play*, che permette un'esperienza interattiva fra utenti che utilizzano console differenti. Il futuro si prospetta prospero anche se ancora in fase embrionale per le realtà aumentate, la VR, le AI e i giochi blockchain come Roblox.

PS5 vs. Xbox S

34 <https://plarium.com/it/blog/console-generations/>

35 <https://itigic.com/it/all-generations-of-consoles-history-evolution-and-models/>

36 <https://patents.google.com/patent/US2455992A/en>

37 <https://plarium.com/it/blog/primo-videogioco/>

38 http://www.giuseppecostanza.it/research_videogames_storia.html

Sviluppo open-source

Procedimenti facilitati

Motori grafici
nocode

Rispetto a quanto succedeva in passato, al giorno d'oggi è possibile sviluppare un videogioco indipendente, anche senza saper scrivere codice, con i cosiddetti *software nocode*.

Primi
framework

Inizialmente, per dare una mano agli sviluppatori, furono creati strumenti che risolvevano il problema con alcuni framework di gioco come *libGDX* e *OpenGL* che fornivano chiamate di funzioni e caratteristiche standard. Tuttavia era ancora difficile cimentarsi e comprendere per chiunque non avesse quelle conoscenze minime in materia.

Motori grafici
open-source

A tal proposito sono stati sviluppati *motori di gioco nocode open-source* come **Unity**, **Unreal**, **Construct** e **Game Maker Studio** che andarono a semplificare di gran lunga la comprensione ed il lavoro grazie all'utilizzo delle GUI e di indicazioni più invitanti³⁹.



◀ FIG. 23
SUPERHOT
(©Superhot
Team)

Motori per principianti

Alcuni software semplificano notevolmente l'esperienza creativa, grazie a semplici funzioni *drag-and-drop* che permettono di applicare script predefiniti, fisiche di gioco, parametri fisici, grafiche, suoni ed altro ancora. Tra questi troviamo **Game Maker Studio**, **Construct** e **Construct 2**.

Motori intermedi

Troviamo anche una fascia a difficoltà intermedia che include **Unity** e **Unreal**. Unity utilizza C# e JavaScript in un IDE dedicato alla creazione di simulazioni bidimensionali e tridimensionali, che ben si presta sia ad un uso per principianti che da più esperti. Questo IDE è stato capace di dare alla luce piccoli capolavori indie come *SUPERHOT* e *Overcooked*⁴⁰.

Motori per esperti

Altri software richiedono competenze maggiori e l'utilizzo del codice per implementare alcune funzioni, come per esempio **Unreal Engine** e **CryEngine**. Unreal Engine per esempio utilizza il linguaggio C++ e supporta anche il *Blueprints Visual Scripting*, non molto consigliato per sviluppatori indipendenti.

39 <https://www.aranzulla.it/programmi-per-creare-giochi-25717.html>

40 <https://www.freecodecamp.org/news/what-is-game-development/>

Roblox

Spazio alla creazione

La piattaforma *Roblox* è una piattaforma, o meglio un ambiente di programmazione, per il gioco e lo sviluppo di giochi, rilasciata nel 2006 come videogioco del genere MMO. La casa di produzione ha sviluppato la piattaforma perché fosse compatibile con qualsiasi dispositivo pc o mobile, cosa che lo ha reso subito popolare in tutto il mondo.

Doppio ruolo Una volta che l'utente si è registrato ed ha creato il suo avatar, ha la possibilità di loggarsi sulla piattaforma sia da *giocatore* che da *creator*. Grazie infatti ad una cassetta degli attrezzi virtuale, la *Roblox Studio*, diversi strumenti per lo sviluppo e il linguaggio di programmazione *Lua*, chiunque in Roblox può creare il proprio mondo virtuale e dare vita ad un gioco indipendente basato sulla piattaforma.



◀ FIG. 24
ROBLOX
(©Roblox)

La distribuzione I giochi poi possono essere distribuiti in forma gratuita, oppure a pagamento tramite varie modalità. Tutte le transazioni sulla piattaforma avvengono attraverso una moneta proprietaria, il *Robux*, che può essere scambiato in valuta reale.

User generated content L'approccio *UGC (User Generated Content)* che per molte compagnie è un elemento secondario, per Roblox ha costituito una carta vincente che ha permesso la diffusione del progetto, creando una piattaforma in grado di dare libertà alla creazione individuale in uno spazio di socialità online.

41 <https://www.rollingstone.it/cultura/gaming/il-curioso-caso-di-roblox/452252/>

42 <https://www.economyup.it/innovazione/roblox-cose-e-come-funziona-la-piattaforma-che-da-accesso-al-metaverso/>

43 <https://www.ninjamarketing.it/2021/06/14/roblox-come-funziona-perche-vale-cosi-tanto/>



◀ FIG. 25
NIKELAND
(©Roblox)

Figure Professionali

Skill diverse, obiettivo comune

Stessa area
operativa

Le professioni legate agli ambiti fin qui analizzati hanno la particolarità di essere sì variamente specifiche ma, al contempo, complementari ed in alcuni casi sovrapponibili. Gli ambiti di applicazione, infatti, sono tali da poter essere ricondotti alla stessa macroarea digitale ed informatica. Qui è indispensabile porre l'accento sia sulle caratteristiche estetiche, di ergonomia cognitiva e legate alla UX, sia su quelle di ordine tecnico e tecnologico volte alla realizzazione finale di qualsivoglia progetto.

Fine comune

Lo scopo di UI designer, web e game developer dunque è il medesimo: assicurare allo user una buona usabilità attraverso prodotti digitali tecnicamente efficienti e che sappiano ben comunicare la propria funzione senza ambiguità.

Why you should learn to code

Matthew Reynolds | TEDxBasel



UI Designer

Il progettista dell'usabilità

UX e UI: scopi
congiunti

Lungo il percorso progettuale, lo UI designer lavora a valle di una consistente ricerca degli UX designer. Infatti, nonostante molti pensino il contrario, UX e UI sono due discipline diverse ma complementari. La sostanziale differenza risiede nel fatto che lo UX designer si preoccupa dell'esperienza globale, analizzando l'intero percorso che lo user fa per risolvere un problema e come questo possa essere facilitato, mentre lo UI designer concentra i suoi sforzi sull'estetica e sulla funzionalità dell'interfaccia digitale di un prodotto come conseguenza della UX research⁴⁴. Dunque UX e UI design condividono il processo di design thinking, la consapevolezza dei bisogni dell'utente e l'obiettivo finale di realizzare un prodotto che lo soddisfi.

Non solo
estetica

Il suo lavoro non è solo focalizzato sul design estetico, ma anche e soprattutto sulla comprensione psicologica dell'*affordance cognitiva* delle componenti disposte all'interno di un'interfaccia. Per evitare errori e garantire una progettazione oggettivamente funzionale che faccia il suo dovere, dovrà altresì condurre ricerche e test. Per questo motivo è tipico che gli UI designer offrano i loro servizi anche in fasi post-lancio, per monitorare le prestazioni ed eventualmente aggiornare i contenuti.

Ruolo primario

Quindi la priorità dello UI design è di fornire all'utente i giusti mezzi per trovare ciò che cerca, in modo veloce, gradevole e intuitivo, preoccupandosi dell'estetica e della navigabilità di siti, e GUI in genere, assicurandosi anche che

queste siano responsive in modo da essere visualizzate correttamente su dispositivi diversi.

Hard skill

Le hard skill che non devono mancare nel suo bagaglio sono:

- conoscenza del **graphic design** e della gestione degli elementi di **visual branding**;
- gestione dei **software** per la produzione grafica e la **prototipazione mock-up**;
- conoscenza degli **standard** dei device per i quali progetta;
- competenza nel campo dell'**ergonomia cognitiva**.

Soft skill

Per ciò che riguarda le soft skill, queste includono:

- buone **capacità creative** e di **problem solving**;
- una spiccata **empatia** e capacità di **comunicazione** per comprendere al meglio i bisogni degli utenti;
- una certa **flessibilità** di adattamento ai vari progetti;
- una buona capacità di **lavorare in team**.

Team work

Quest'ultimo punto riveste un ruolo fondamentale ed il motivo pare essere ovvio. Poiché le strutture di siti e app creati dallo UI designer sono mockup, egli dovrà sempre considerare come e quanto il design possa essere codificato poi dallo sviluppatore che dovrà implementarlo⁴⁵. Il designer dovrebbe costantemente scambiarsi feedback con gli sviluppatori in modo da rendersi consapevoli dell'effettiva fattibilità di un progetto.

44 <https://careerfoundry.com/en/blog/ux-design/the-difference-between-ux-and-ui-design-a-laymans-guide/>

45 <https://www.francescoazzaro.it/differenza-tra-web-designer-e-sviluppatore/>

UX

DESIGN THINKING

BISOGNI UTENTE

SODDISFAZIONE
UTENTE

UI

Web Developer

Il traduttore dell'usabilità

Sviluppatore
front end

Il principale responsabile della seconda fase di progettazione di un sito web o app è lo sviluppatore front end. Il suo compito infatti è quello di tradurre in codice i mockup dei designer nella maniera più fedele possibile.

Front end
hard skill

Per la gestione dei loro progetti, gli sviluppatori front end devono disporre di alcune hard skill come:

- un'ottima conoscenza dei linguaggi **HTML**, **CSS** e **JavaScript** indispensabili per la progettazione;
- una perfetta padronanza delle funzioni dei browser e dei software di **text editing**;
- una buona preparazione nell'utilizzo dei **framework** più comuni⁴⁶.

Sviluppatore
back end

Allo sviluppatore back end è invece demandato il compito di progettare la struttura invisibile dei siti sui server web e sui database, non sul browser. In definitiva, il back end developer svolge tutte le operazioni di coding per la definizione del sistema che sostengono il front end. La sua conoscenza si estende anche sui *database*, ovvero quei sistemi che immagazzinano i dati di navigazione processati dal back end e forniti al front end.

Back end
hard skill

Le hard skill dello sviluppatore back end comprendono:

- ottima conoscenza dei maggiori linguaggi server-side come **Java**, **Python**, **PHP**, **C**, **C#**;
- conoscenza di metodologie **problem solving**;
- conoscenza dei **protocolli di rete**;

- conoscenza dei principali **framework**;
- padronanza delle funzionalità dei **sistemi di gestione** (database)⁴⁷.

Sviluppatore
full stack

Quando le mansioni del front end e del back end developer sono svolte dalla stessa persona, allora si parla di *full stack developer*. Si tratta di una figura professionale molto ricercata perchè agevola molto le procedure progettuali, sapendo gestire il codice sia da un lato che dall'altro.

Web developer
soft skill

Le soft skill dei web developer, comuni a tutte e tre le categorie, sono:

- buona attitudine al problem solving;
- una minima dose di creatività;
- capacità di lavorare in team.

Team work

Anche per queste professioni vale il discorso della collaborazione in team lungo tutte le fasi progettuali. I colleghi designer vanno ascoltati nelle richieste che fanno e accompagnati verso le scelte più fattibili in termini di coding, per gestire il tutto più facilmente.

⁴⁶ <https://www.techyon.it/articoli/front-end-developer-cosa-fa-quali-strumenti.html>

⁴⁷ <https://www.techyon.it/articoli/web-developer-chi-e-cosa-fa.html>

UI Designer vs. Web Dev

Un'occasione di scambio

Obiettivi
condivisi

Una volta compreso i compiti che UI designers e web devs devono svolgere, è facile accorgersi di quanto sia opportuno che queste due figure professionali lavorino fianco a fianco per una causa comune: creare prodotti e UX che soddisfino i desideri ed i bisogni degli user. Gli utenti, infatti, rimangono sempre al centro di questa grande ruota che è la progettazione.

L'incontro è
occasione

I due versanti, dovendo cooperare, devono anche sforzarsi di comprendere la disciplina dell'altro. I progettisti avrebbero la possibilità di imparare le basi dello sviluppo, mentre i developer potrebbero applicare al loro lavoro un approccio un po' più di design. Si tratta di uno scambio in cui sia gli uni che gli altri non avrebbero nulla da perdere, ma tutto da guadagnare.

Un incentivo
per i designer

Ai designer che si avvicinano al coding, nonostante possano scoprire una nuova passione ed un nuovo lavoro, non viene chiesto di diventare degli sviluppatori full stack. Tuttavia avere delle competenze di base può dimostrarsi di grande impatto quando si lavora all'interno di un team multidisciplinare, perché aiuta ad essere più flessibili ed a comunicare con i developer in maniera significativamente migliore⁴⁸.

Evitare gli errori

Questo tipo di comunicazione risulta essere di grande aiuto nei passaggi di consegne, spesso momento ostico, tra progettisti e sviluppatori. Parlando due lingue diverse, i progettisti sono spesso ignari dei vincoli e delle limitazioni del

codice. Ecco che anche in questo caso, possedendo le nozioni di base, si eviterebbero perdite di tempo. I designer, sapendo cosa è possibile e cosa invece non lo è, risolverebbero eventuali problemi a monte del percorso progettuale.

Agevole
prototipazione

Sovente i designer si accorgono tardi delle limitazioni insite nei tool di progettazione e prototipazione che utilizzano, come Adobe XD, Figma o Sketch che non offrono la stessa fedeltà di un prototipo regolarmente codificato. Conoscere i linguaggi di sviluppo consente la consegna al dev team di mockup e prototipi privi di sorprese. Ciò permetterebbe anche di ispezionare e modificare il codice direttamente sul browser durante le fasi di testing.

Competenze di
base

Questa conoscenza di base che i progettisti dovrebbero possedere consiste nei linguaggi di programmazione front end. I linguaggi fondamentali come sappiamo sono HTML, che rappresenta il *wireframing*, e CSS. La combinazione dei due equivale ad un mockup privo di certe funzionalità. Per accedere a questa conoscenza più approfondita dei meccanismi dello sviluppo, i designer possono anche acquisire le nozioni di JavaScript, il che darebbe ancora una marcia sul piano della comunicazione con i team di sviluppo descritta fin qui.

Valore aggiunto

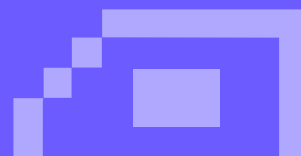
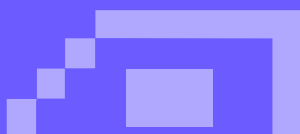
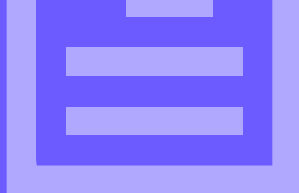
Il vantaggio di queste competenze diviene ancora più evidente in un'ottica di ricerca lavorativa. Arricchire la propria figura di UI designer permette di dare lustro al proprio profilo professionale. Soprattutto all'interno di un panorama lavorativo florido come quello dell'industria tecnologica, in cui nuovi ruoli ed opportunità nascono costantemente. Le professioni ibride si trovano ad essere ricercatissime dai dipartimenti HR, poiché

rappresentate da persone dotate di grande flessibilità e agevole gestione lungo tutto l'iter progettuale.

Vantaggi per i
web developer

Sul versante opposto, i web developer possono agevolare i propri progetti adottando una serie di espedienti, tratti dal design thinking, per velocizzare e facilitare determinati processi. Ciò può avvenire ad esempio definendo una metodologia di progettazione per componenti. Gli elementi all'interno di un progetto, ma anche tra progetti diversi, sono spesso ripetuti in più punti, dunque è qui che si possono definire degli elementi modulari, dei pezzi di codice, ricorrenti ed intercambiabili che accorciano i tempi di un progetto. Questo approccio per componenti modulari consente di accedere ad un variegato corredo di elementi di codice che un team di sviluppatori può utilizzare, potremmo dire, come un framework. Per i web devs significherebbe una comunicazione efficace con i progettisti su cosa è possibile e cosa no, una più veloce consegna degli esecutivi al cliente ed un approccio flessibile e scalabile a più progetti.

48 <https://www.dencreative.com/im-a-designer-how-has-learning-to-code-benefited-my-career>



Game Developer

Un artista computazionale

Chi lavora allo sviluppo

Nel mondo della creazione e sviluppo dei videogame, le figure professionali maggiormente richieste sono essenzialmente quattro⁴⁹:

- il **game designer** che cura l'aspetto generale del videogioco, gli storyboard degli eventi, le particolarità dei personaggi, la difficoltà dei livelli e le meccaniche di gioco, lavorando in stretto contatto con il Game Programmer per assicurare prestazioni adeguate.
- il **game programmer** o developer che utilizza una programmazione orientata ad oggetti, struttura i dati, conosce i motori di gioco più utilizzati del momento e garantisce performance ottimali.
- **l'art illustrator** e **l'animator 2D/3D**: utilizzando il lavoro del game designer, mette in pratica le idee di quest'ultimo, animando personaggi e oggetti.
- il **videogame tester**: collauda per ore i videogame, per assicurarsi che non ci siano bug e che l'esperienza finale sia perfetta.

Hard skill

Le sue hard skill comprendono:

- la conoscenza dei principali linguaggi di programmazione object-oriented;
- approfondite competenze IT, grafica e design;
- nozioni di matematica e fisica, AI;
- perfetta padronanza della lingua inglese, sia parlata che scritta.

Soft skill

Per ciò che riguarda le soft skill, invece, queste devono comprendere:

- un buon senso estetico;
- una forte capacità di team working;
- spiccata attitudine al problem solving;
- saper gestire al meglio il suo tempo in relazione ai progetti.

I compiti del ruolo

Il principale compito del game developer è quello di programmare e sviluppare codici e script su cui si verrà basato il videogioco. Definisce tutte le linee guida a cui il gioco dovrà fare riferimento lungo l'intero iter di progetto e, per questo motivo, dovrà prendere parte attiva anche alla fase di game design⁵⁰. Dopo aver preso parte alla definizione del concept, integra le grafiche 2D/3D dei suoi colleghi e traduce in codici il design e tramite l'ausilio dei tester individua e risolve i bug presenti⁵¹. Una volta definita ogni caratteristica generale, divide il lavoro in base alla priorità dei moduli di gioco, scegliendo accuratamente il giusto ambiente di sviluppo integrato.

49 <https://www.classup.it/blog/sviluppatore-di-videogiochi-come-si-crea-un-videogame-wseg7>

50 <https://4news.it/che-differenze-ci-sono-tra-un-game-designer-e-un-game-developer/>

51 <https://www.wgu.edu/blog/video-game-industry-jobs-designer-developer1910.html#close>

2.0

Casi Studio

2.1 Casi Studio

Introduzione

Scelta dei casi studio

Per i casi studio sono stati utilizzati esempi per i tre ambiti diversi più quelli nello specifico per il progetto. Esempi di “great UI design”, progetti realizzati tramite creative coding, i più famosi motori di gioco e piattaforme con cui poter realizzare i propri progetti e casi studio sulle più diffuse accademie online.

UI Design

Per lo sviluppo Front End sono stati presi in considerazione quei casi di UI Design in cui le funzioni e gli elementi svolgono perfettamente il loro lavoro, quello di tradurre visivamente in modo diretto e intuitivo le informazioni rilevanti, utilizzando ognuna tecniche diverse o complementari.

Creative Coding

Per il Creative Coding sono stati presi in considerazione progetti realizzati da vari artisti in diversi ambiti di arte generativa. Ognuno a dimostrare le innumerevoli potenzialità di questi strumenti.

Game
Development

Per il Game Development sono stati analizzati i casi dei motori grafici di gioco per lo sviluppo tramite codici e nocode. Oltre questi sono stati analizzati due casi in cui i videogiochi permettono agli utenti di poter creare qualcosa di personale al loro interno.

Academy

Per le academy online, infine, si è deciso di prendere in considerazione quelle del panorama italiano, in modo tale da analizzarne l'impostazione grafica e comunicativa.

UI

Design

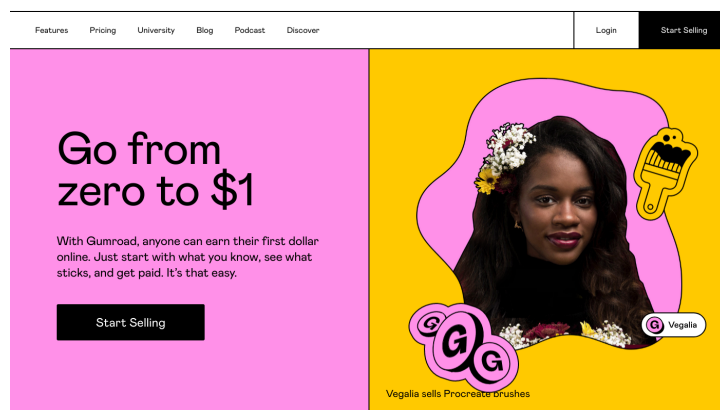
Gumroad

2011 | E-COMMERCE

1 <https://gumroad.com/>

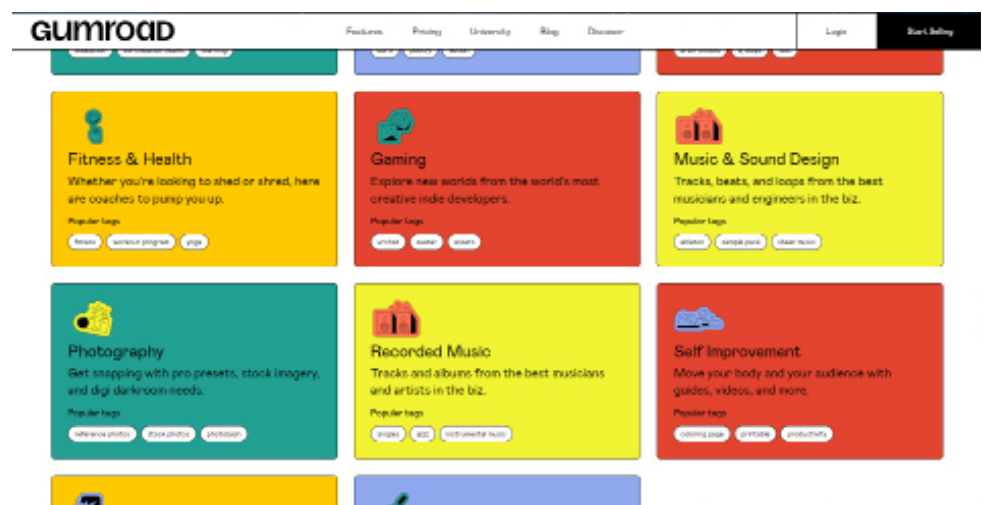
Un travolgente insieme di illustrazioni dal gusto retro anni '80, con colori ad alto contrasto e forme astratte, di derivazione Memphis design, pongono perfettamente l'attenzione sugli gli elementi desiderati nel momento adatto.

GUMROAD



◀ FIG. 26
GUMROAD
WEBSITE
(@Gumroad)

FIG. 27 ▶
GUMROAD
WEBSITE
(@Gumroad)

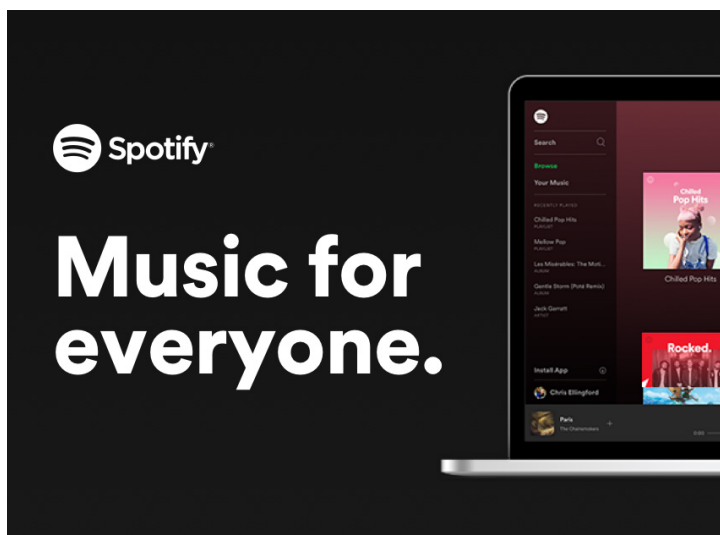


Spotify

2006 | PIATTAFORMA DI STREAMING

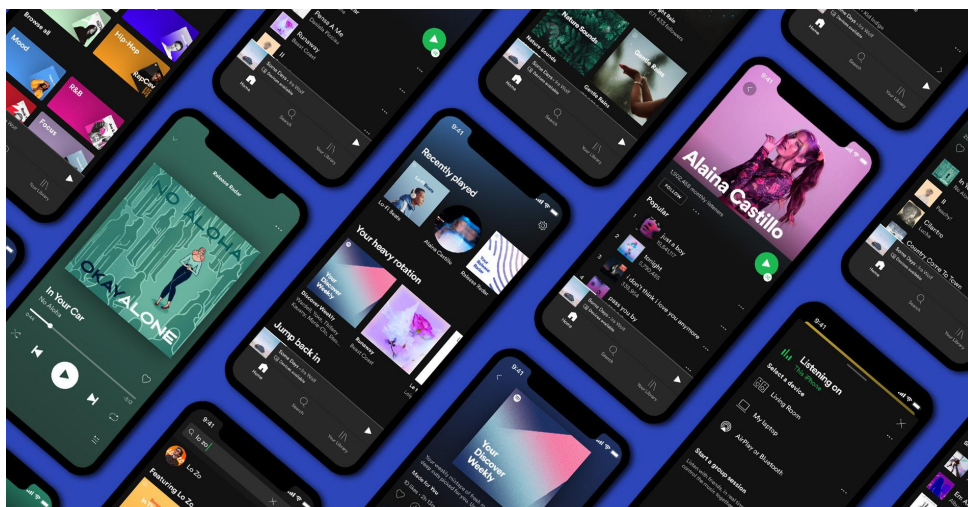
2 <https://open.spotify.com/>

La UI di Spotify predilige i gradienti ai colori piatti. Ciò permette sia di trasmettere le giuste emozioni associate a determinati brani sia di virare l'attenzione su determinati elementi rispetto ad altri.



◀ FIG. 28
SPOTIFY WEB
(©Spotify)

FIG. 29 ▶
SPOTIFY APP
(©Spotify)

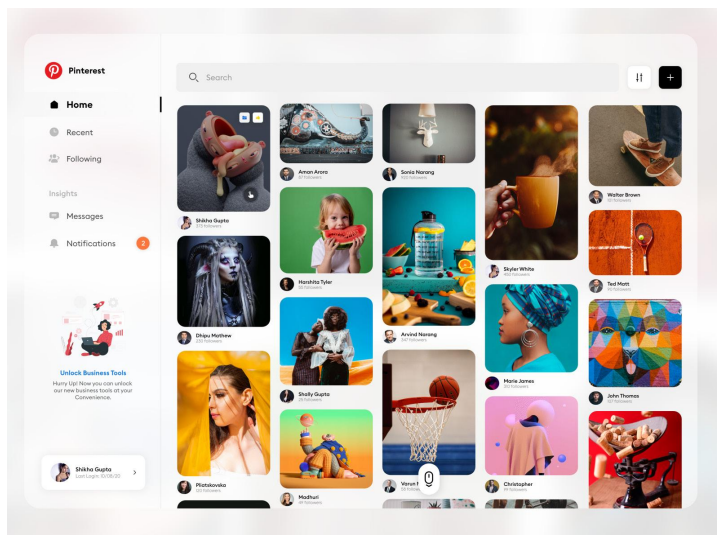


Pinterest

2010 | SOCIAL

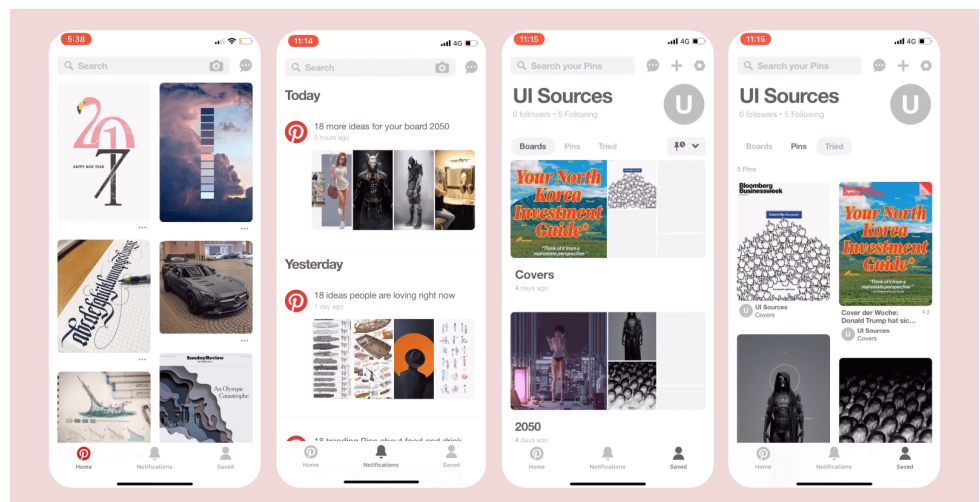
3 <https://www.pinterest.it/>

Pioniere per eccellenza del “card design”, Pinterest, unisce la tecnica del flusso a cascata con leggere sfumature che permettono di mettere in risalto le diverse card quando il cursore si trova sopra di esse, offrendo agli utenti un’esperienza fluida.



◀ FIG. 30
PINTEREST
WEBSITE
(©Pinterest)

FIG. 31 ▶
PINTEREST APP
(©Pinterest)

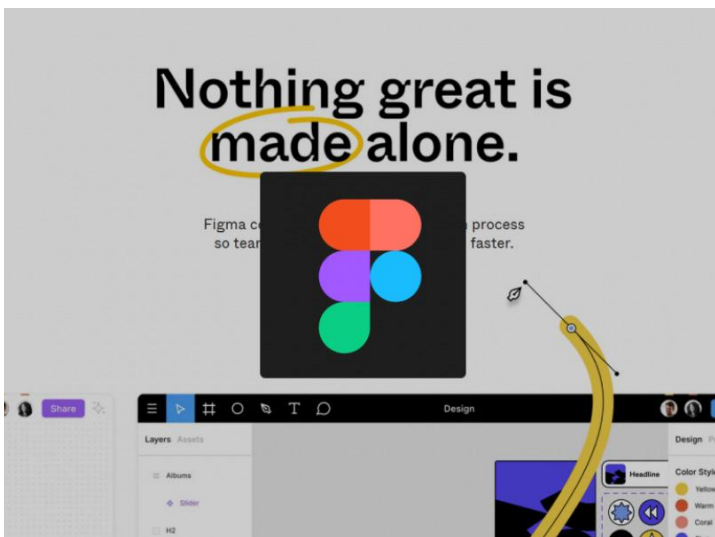


Figma

2016 | SOFTWARE

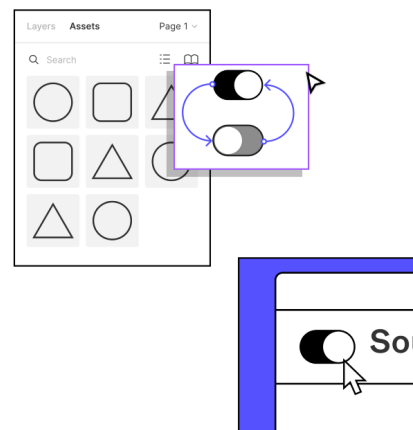
4 <https://www.figma.com/>

Figma è un software di progettazione in grado di fornire istruzioni precise circa le attività essenziali per il funzionamento dell'applicazione. Per aumentare l'interazione, il software mostra i suggerimenti tramite pop-up che non scompaiono finché l'utente non vi clicca sopra per chiuderli.



◀ FIG. 32
FIGMA
(©Figma)

FIG. 33 ▶
Figma
(©Figma)

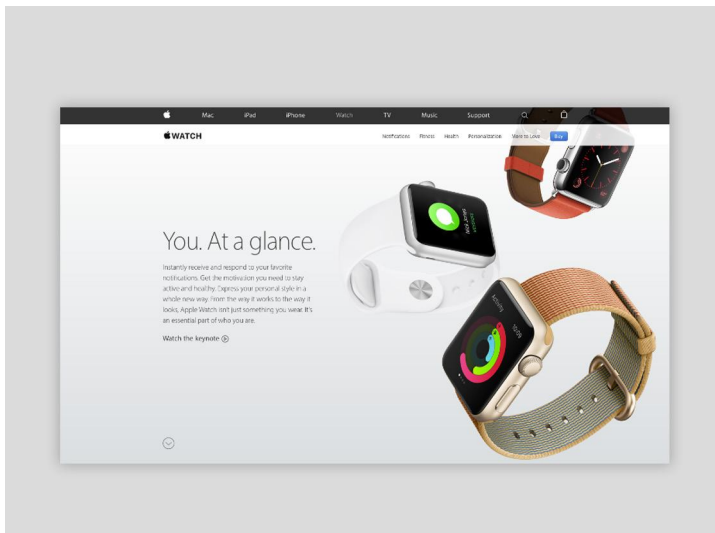


Apple

2011 | E-COMMERCE

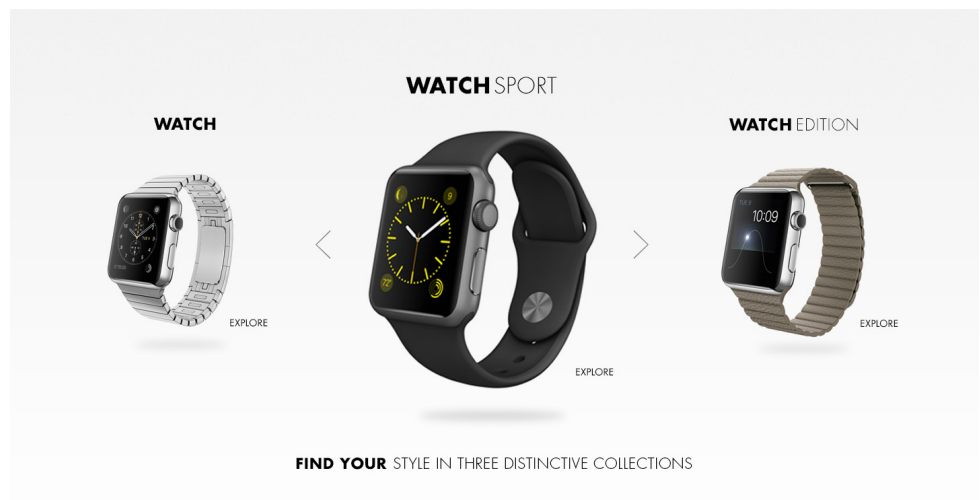
5 <https://www.apple.com/>

Il sito della Apple, utilizza da una parte colori monocromatici, quali il bianco e il nero, per enfatizzare i prodotti. Dall'altra garantisce un'esperienza immersiva durante la navigazione, attraverso la tecnica dello zoom.



◀ FIG. 34
APPLE E COMMERCE
(©Apple)

FIG. 35 ▶
APPLE E
COMMERCE
(©Apple)



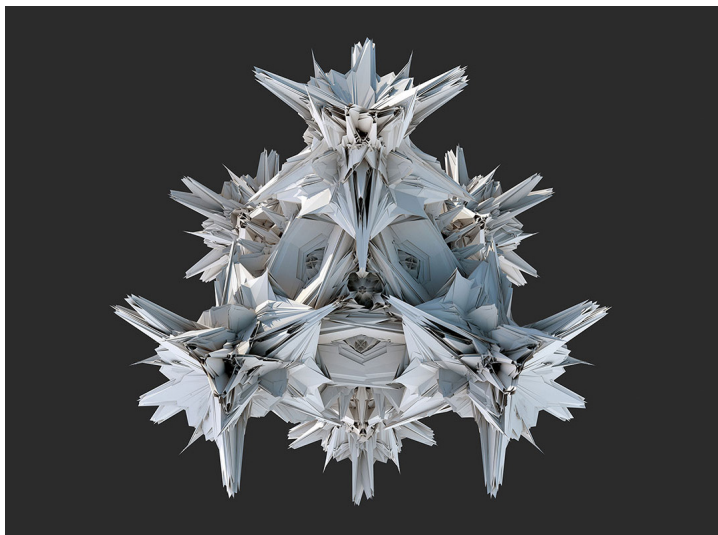
Creative Coding

Platonic Solid

2008 | MICHAEL HANSMEYER

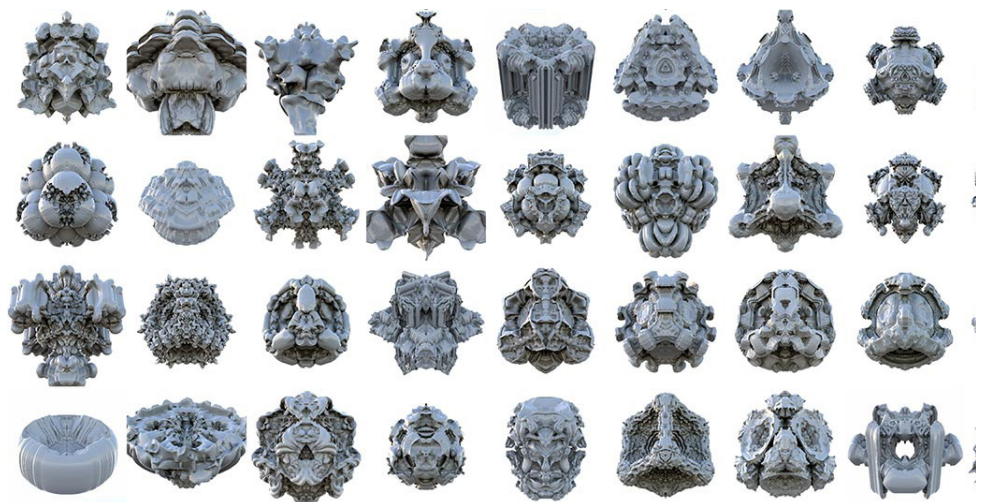
6 <https://www.michael-hansmeyer.com/platonic-solids>

Il progetto esplora come da un processo geometrico puramente operativo si possano creare forme complesse. L'opera di generative Art è interamente realizzata attraverso processi di creative coding.



◀ FIG. 36
PLATONIC
SOLID
(©M.Hansmeyer)

FIG. 37 ▶
PLATONIC
SOLID
(©M.Hansmeyer)



Mechanical Plotter Drawings

2016 | ANDERS HOFF

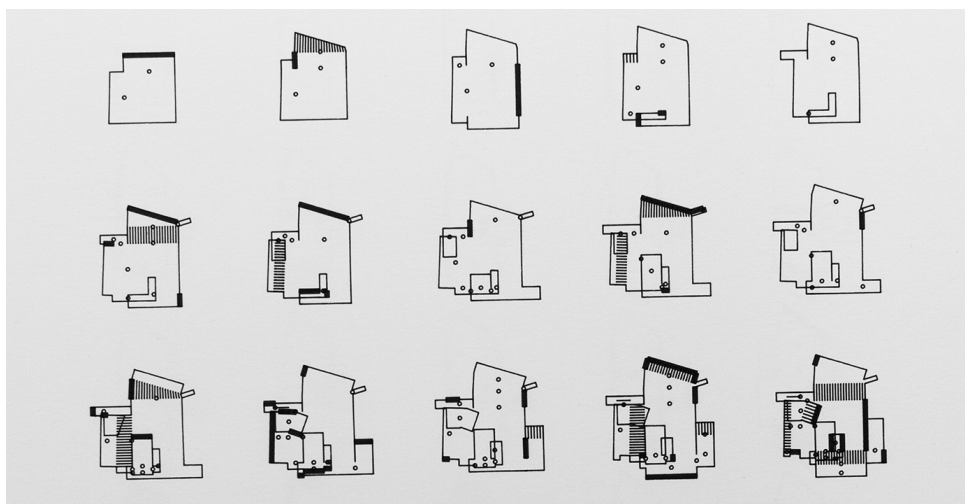
7 <https://inconvergent.net/>

Partendo da strutture organizzate, l'artista si avvale di algoritmi per scinderne la struttura, creando composizioni estetiche e texture caotiche.



◀ FIG. 38
MECHANICAL
PLOTTER
DRAWINGS
(©A.Hoff)

FIG. 39 ▶
MECHANICAL
PLOTTER
DRAWINGS
(©A.Hoff)

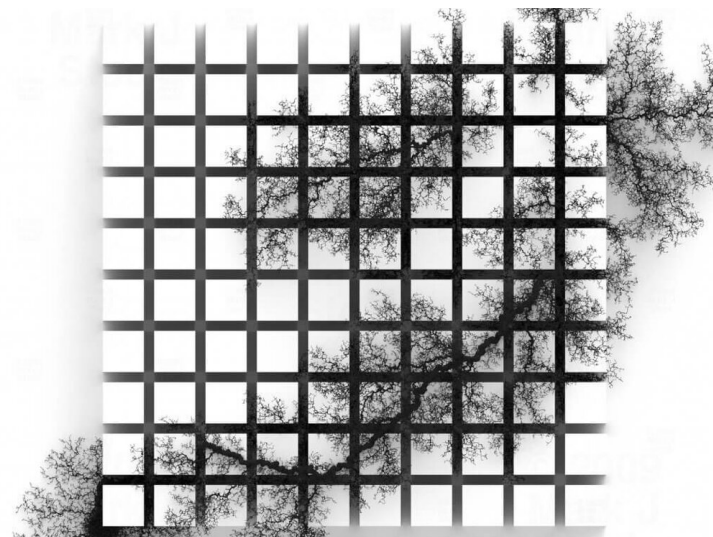


Dendrite

2009 | MARK J. STOCK

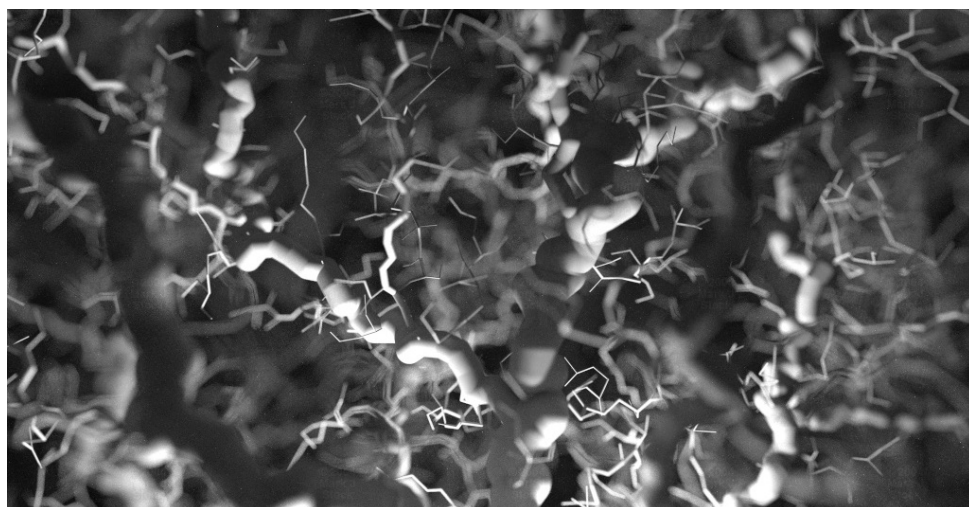
8 <http://markjstock.com/#/dendrite/>

Attraverso processi algoritmici l'artista studia la connessione tra forme organiche e inorganiche, ricreando così strutture digitali che rimandano al mondo naturale.



◀ FIG. 40
SPRAWL
(©M.J.Stock)

FIG. 41 ▶
TWIG
(©M.J.Stock)

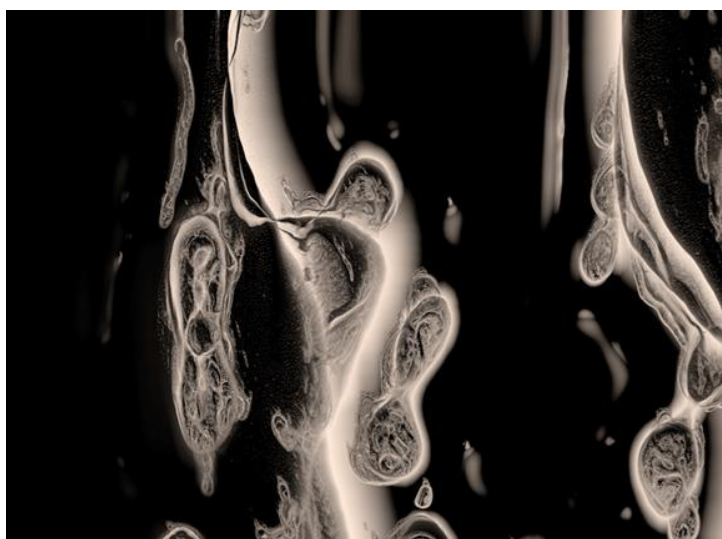


Plasm 4K shot

2014 | MAURIZIO MARTINUCCI AKA TEZ

9 https://tez.it/index_2011.html

Attraverso la Generative Art, l'artista crea delle installazioni percettive a livello visivo e sonoro, in grado di far immergere lo spettatore generando così esperienze emotive.



◀ FIG. 42
PLASM 4K SHOT
(©Tez)

FIG. 43 ▶
DISPLACE
(©Tez)

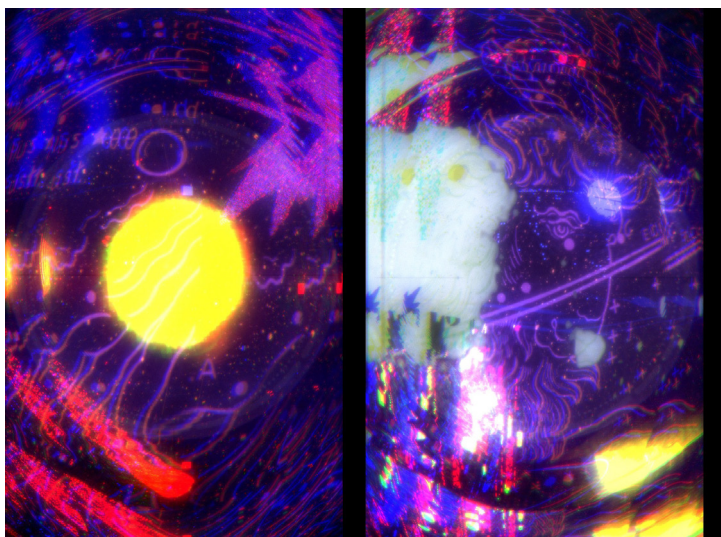


We are Poems

2023 | FERN GOLDFRAB-RAMALLO

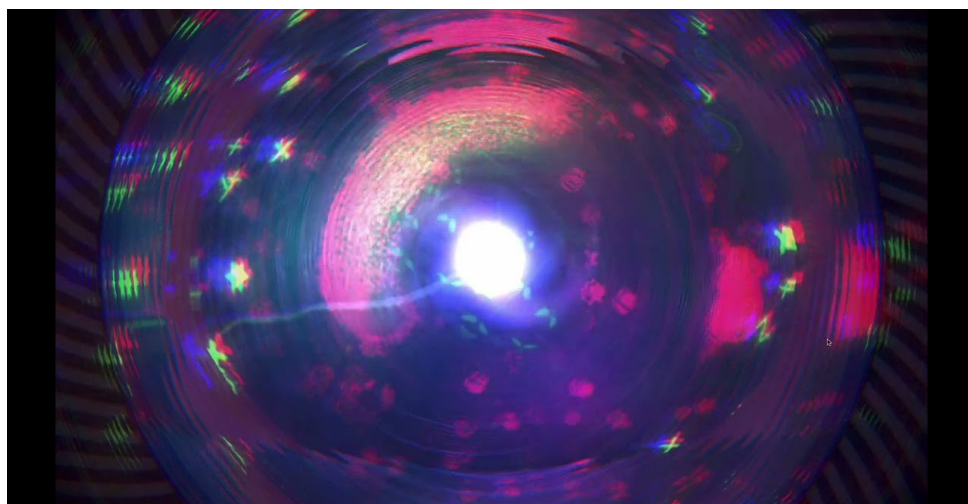
10 <https://triennale.org/en/magazine/unknown-play-pleasure-poems-ramallo>

L'opera, realizzata per la 23^a Esposizione Internazionale di Triennale a Milano, si presenta come un'installazione interattiva in grado di trasmettere le sensazioni di spazio profondo.



◀ FIG. 44
WE ARE POEMS
(©F.Goldfarb-
Ramallo)

FIG. 45 ▶
WE ARE POEMS
(©F.Goldfarb-
Ramallo)



Game

Dev

Roblox

2006 | VIDEOGAME MMO

11 <https://www.roblox.com/>

Roblox è una piattaforma sandbox che permette di sviluppare i propri giochi e condividerli con altri utenti, in modo gratuito o a pagamento. Il software utilizza il linguaggio di programmazione Lua.



◀ FIG. 46
ROBLOX
(©Roblox)

FIG. 47 ▶
ANIME FIGHTER
SIMULATOR
(©Roblox)



Unreal Engine

1998 | SOFTWARE

12 <https://www.unrealengine.com/en-US>

Unreal Engine è un motore grafico sviluppato da Epic Games in continuo sviluppo nel software per adattarsi ad hardware sempre più potenti. Tramite il linguaggio C++ permette lo sviluppo di videogiochi, grafiche e animazioni 2D/3D in tempo reale.



◀ FIG. 48
UNREAL
(©Unreal)

FIG. 49 ▶
BLACK MYTH
WUKONG
(©Games
Science)



Unity

2005 | MOTORE GRAFICO MULTIPIATTAFORMA

13 <https://unity.com/>

Motore grafico multipiattaforma, Unity, permette di creare videogiochi, animazioni 3D in tempo reale e altri contenuti interattivi. Tale strumento è maggiormente utilizzato per applicazioni mobile e utilizza il linguaggio C#.



◀ FIG. 50
UNITY
(©Unity)

FIG. 51▶
THE FOREST
(©Endnight
Games)

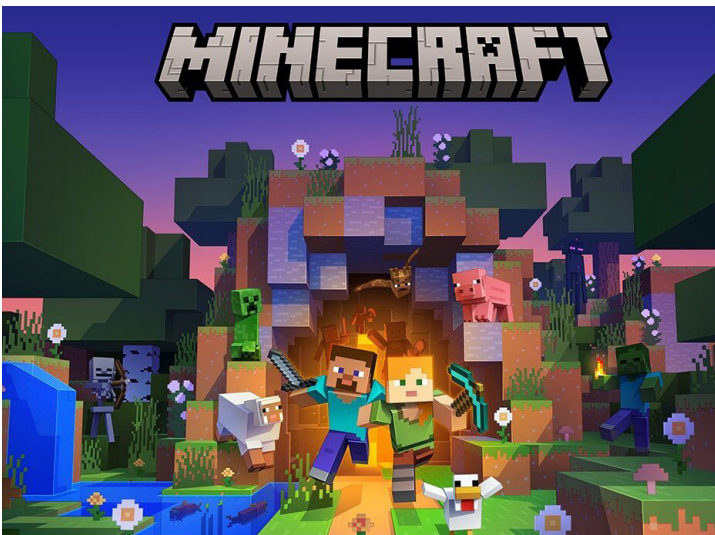


Minecraft

2011 | VIDEOGIOCO SANDBOX OPEN WORLD

14 <https://www.minecraft.net/it-it>

Minecraft è un videogioco sandbox open world sviluppato in Java. Il gioco presenta un ambiente generativo nel quale gli utenti possono giocare in solitaria o in modalità multiplayer. Ogni utente può, inoltre, creare i propri mondi e godere di risorse infinite nella modalità creativa.



◀ FIG. 52
MINECRAFT
(©Minecraft)

FIG. 53 ▶
MINECRAFT
(©Minecraft)



Fortnite

2017 | BATTLE ROYALE GAME

15 <https://www.epicgames.com/fortnite/it/home>

Sviluppato da Epic Games, Fortnite fenomeno mondiale dei battle royale, possiede una modalità creativa che permette agli utenti di creare sessioni di gioco personalizzate e condividere i codici ad altri giocatori permettendo loro di giocare con le modalità create.



◀ FIG. 54
FORTNITE
(©Fortnite)

FIG. 55 ▶
FORTNITE
(©Fortnite)



Code Academy

Boolean

2021 | E-COMMERCE

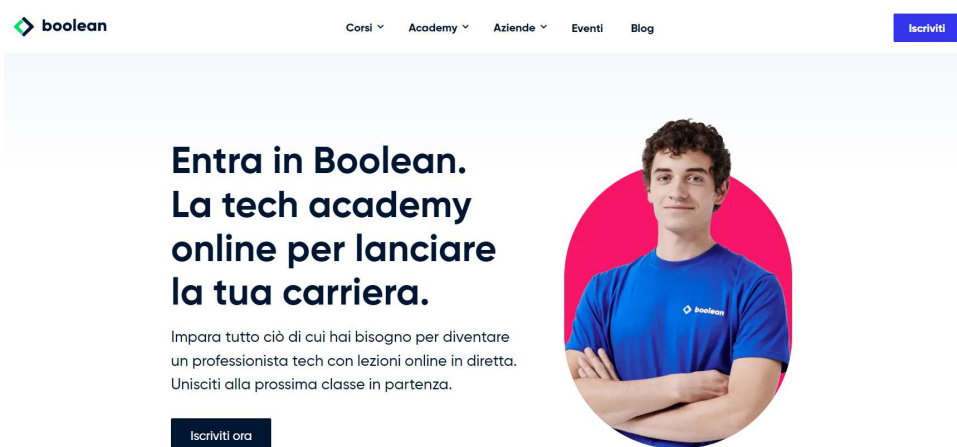
16 <https://boolean.careers/>

Boolean è un'accademia online che utilizza una grafica caratterizzata da colori accesi e illustrazioni in stile pixel affiancate da foto. Il tone of voice è ottimista, informale e assertivo. Nel complesso la strategia utilizzata rende la comunicazione intuitiva.



◀ FIG. 56
BOOLEAN LOGO
(©Boolean)

FIG. 57 ▶
BOOLEAN
WEBSITE
(©Boolean)



17 <https://aulab.it/>

L'accademia online Aulab utilizza una grafica caratterizzata da illustrazioni e elementi inorganici. I colori prevalenti sono il grigio e il giallo. Il tone of voice è motivazionale e ottimista. Nel complesso la strategia utilizzata rende la comunicazione chiara ed efficace.

◀ FIG. 58
AULAB LOGO
(©Aulab)



FIG. 59 ▶
AULAB WEBSITE
(©Aulab)



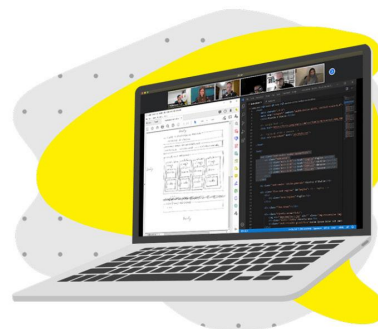
CORSI ▾ GUIDE ▾ WEBINAR ▾ BLOG CHI SIAMO ▾ SEI UN'AZIENDA? 🏢

DIVENTA SVILUPPATORE WEB CORSO DI CODING

Vuoi imparare a programmare?
Scopri come diventare sviluppatore web

- ▶ 3 mesi di corso, lezioni online in diretta + on demand
- ▶ Scopri i segreti di HTML, CSS, Javascript e PHP con Framework Laravel
- ▶ Soluzioni di pagamento ultra personalizzate!

📞 RICHIEDI INFO



Udemy

2021 | E-COMMERCE

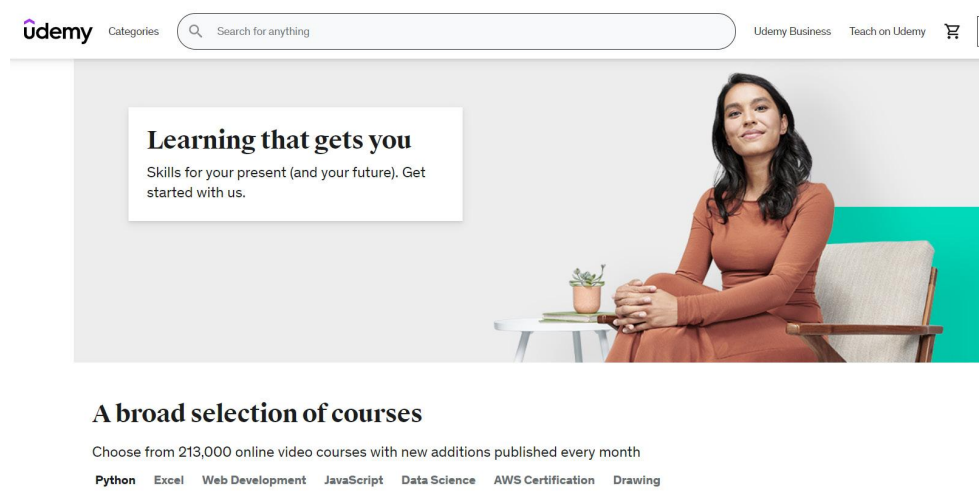
18 <https://www.udemy.com/>

Udemy è un'accademia online che utilizza una grafica caratterizzata da colori piatti affiancati ad un uso prevalente di foto. Il tone of voice è colloquiale e motivazionale. Tuttavia, nel complesso la strategia utilizzata rende la comunicazione confusionale.

◀ FIG. 60
UDEMY
LOGO
(©Udemy)



FIG. 61 ▶
UDEMY
WEBSITE
(©Udemy)



Codecademy

2021 | E-COMMERCE

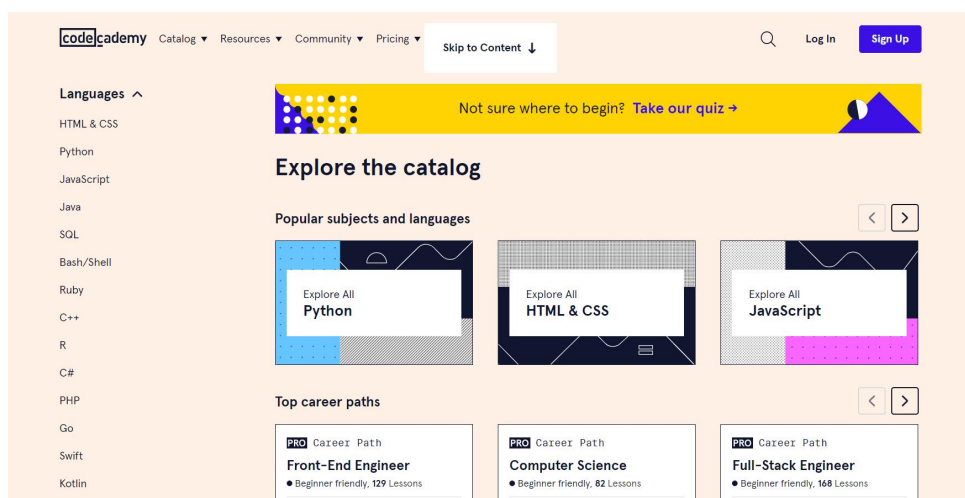
19 <https://www.codecademy.com/>

L'accademia online Codecademy utilizza una grafica caratterizzata da un contrasto di colori e illustrazioni basilari geometriche. Il tone of voice è serio e assertivo. Nel complesso la strategia utilizzata rende la comunicazione intuitiva.

◀ FIG. 62
CODECADEMY
LOGO
(©CodeCademy)



FIG. 63 ▶
CODECADEMY
WEBSITE
(©CodeCademy)



Coursera

2021 | E-COMMERCE

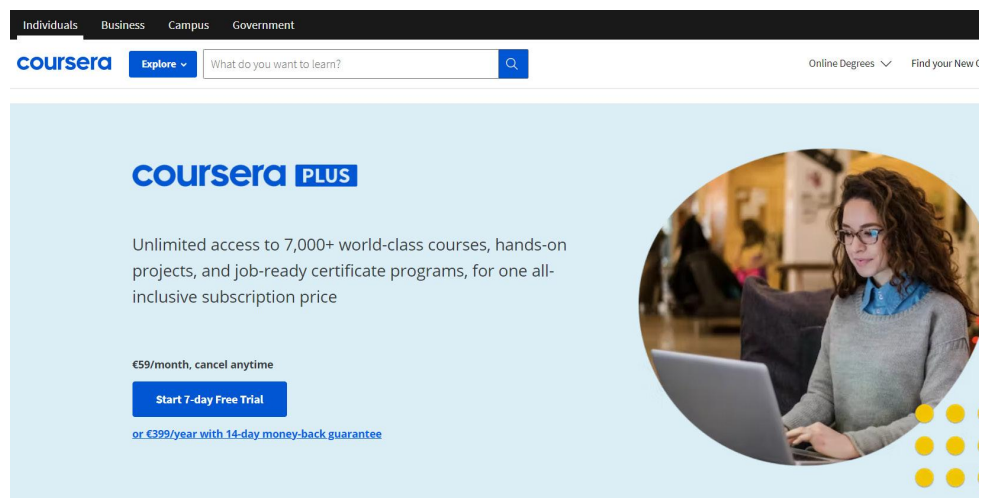
20 <https://www.coursera.org/>

Coursera è un'accademia online che utilizza una grafica caratterizzata da colori piatti e un uso prevalente di foto. Il tone of voice è formale e motivazionale. Nel complesso la comunicazione risulta semplice e di facile comprensione.



◀ FIG. 64
COURSERA
LOGO
(©Coursera)

FIG. 65 ▶
COURSERA
WEBSITE
(©Coursera)



3.0

Personas



“Vorrei capire perchè i dev hanno
sempre da ridere sui miei prototipi!”

“VABBÈ BALLIAMO!”



L'ANIMA DELLA FESTA

TRIXIE RINGO | DESIGNER

Età 27 Anni

Città Milano - L'Avana

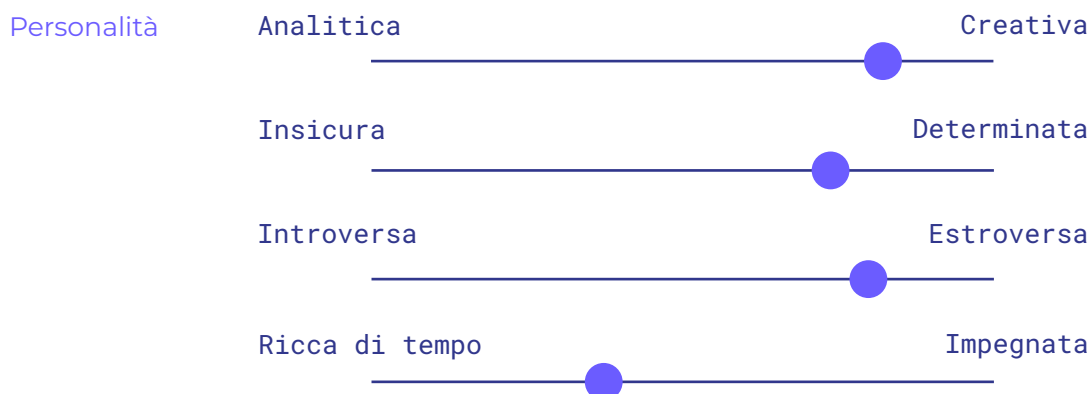
Ruolo UX/UI Designer

Interessi Graphic trends
Musica
Moda

Hobby Clubbing
Scrittura creativa
Fotografia

Social   

Tools Figma
SurveyMonkey
KeyNote



L'ALTERNATIVO

PIETRO NASCA | WEB DEVELOPER

Età 51 anni

Città Losanna - Milano

Ruolo Sviluppatore Back End

Interessi
Natura
Innovazione
Associazioni Umanitarie

Hobby
Camping
Yoga
Mindfulness

Social   

Tools
Eclipse (Linux)
TeamViewer
PowerPoint





“Mia figlia dice che non ho gusto per questo voglio approfondire UI e Front End”

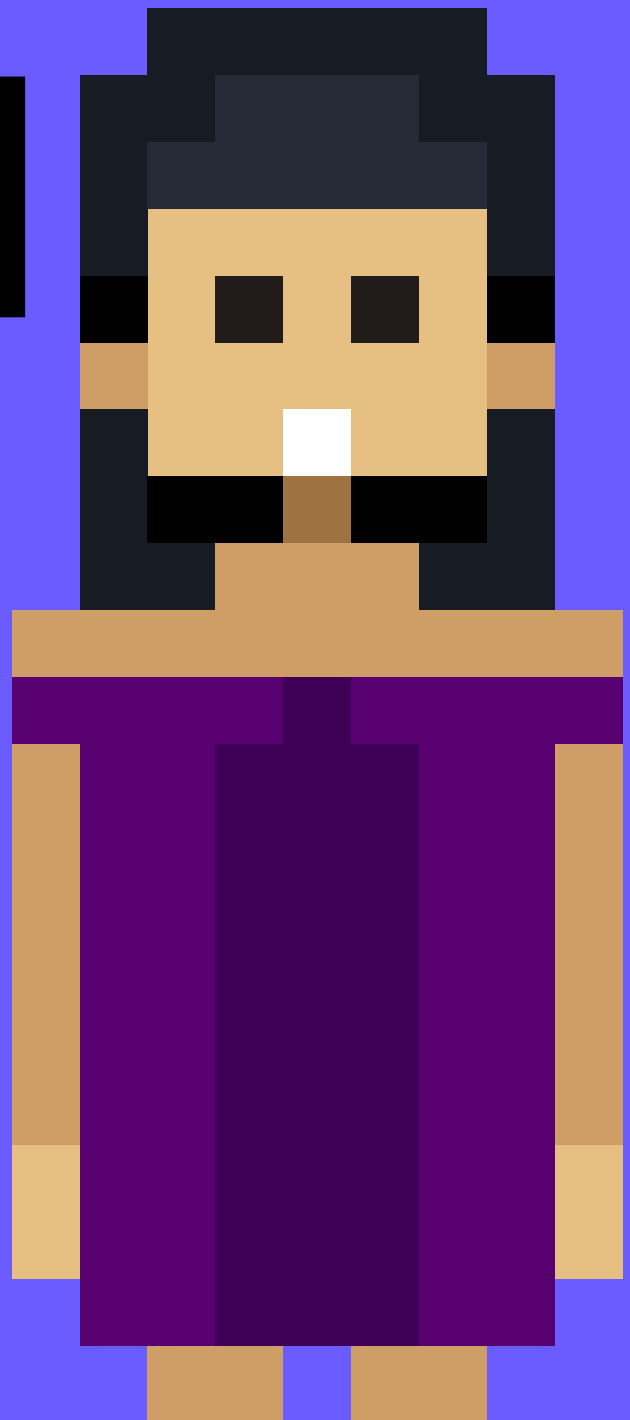
**“TROVA IL CORAGGIO,
NO UNA SCUSA!”**





“Voglio formarmi nel coding e trovare
un lavoro 100% da remoto”

“METTICI UNA CTA!”



THE BOSS

VALENTINA RIZZO | DESIGNER

Età 29 anni

Città Roma

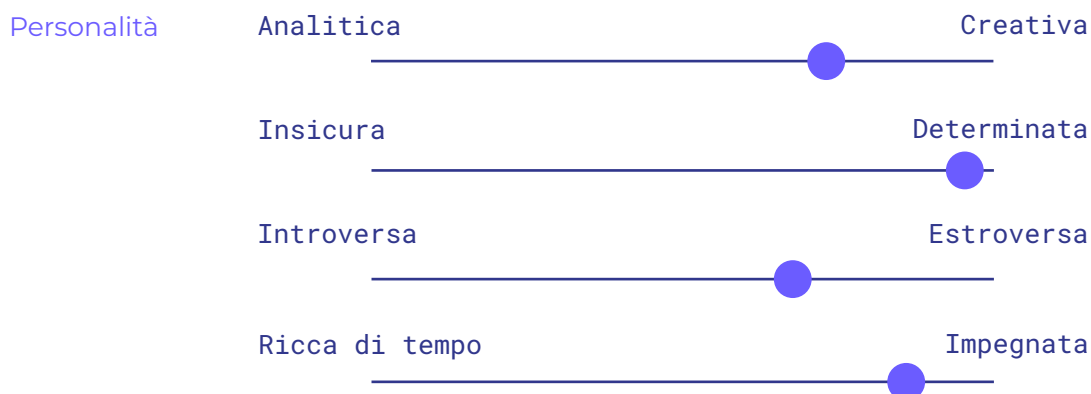
Ruolo UI Designer

Interessi Innovazione digitale
Arte
UX design

Hobby Chitarra
Pilates
Cucina

Social   

Tools Miro
Figma
Sketch



IL CREATIVO

GIACOMO PRINCIPINO | WEB DEVELOPER

Età 33 anni

Città Bari

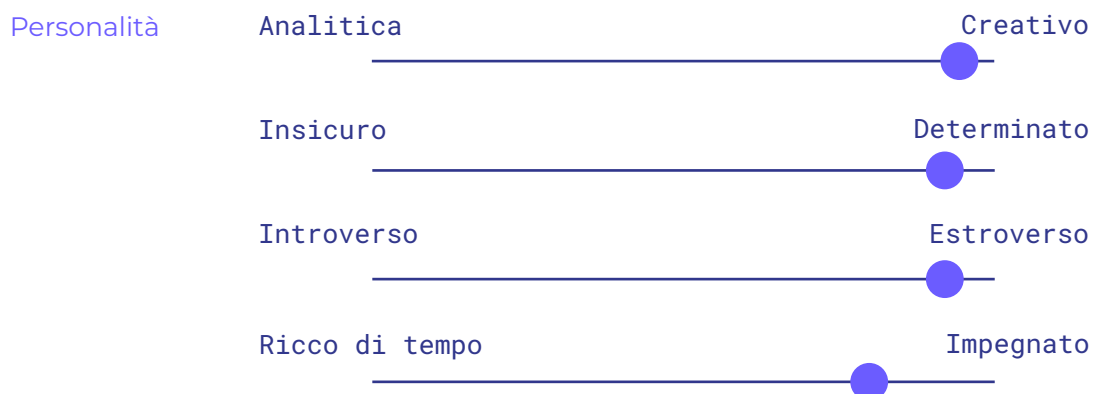
Ruolo Sviluppatore Front End

Interessi Serie TV
Generative Art
Innovazione Digitale

Hobby Disegno
Videomaking
Nuoto

Social   

Tools DaVinci Resolve
Adobe DreamWeaver
Figma





“Vorrei approfondire JS ed il Creative Coding potrebbe essere un ottimo esercizio”

**“L’IMPORTANTE NON È
LA LIBRERIA, MA
COME LA USI!”**





Two horizontal lines representing the address bar and search bar of a browser window.



“So un po programmare, ma volgio migliorare le mia conoscenze Front End”

“CE LA FAI?
SEI CONNESSO?”



IL TECNICO

EDOARDO NOVO / TECNICO INFORMATICO

Età 28 anni

Città Asti

Ruolo Colladautore Ripetitori

Interessi UX/UI Design
Tecnologia
Fumetti

Hobby Elettronica
Stampa 3D
Anime

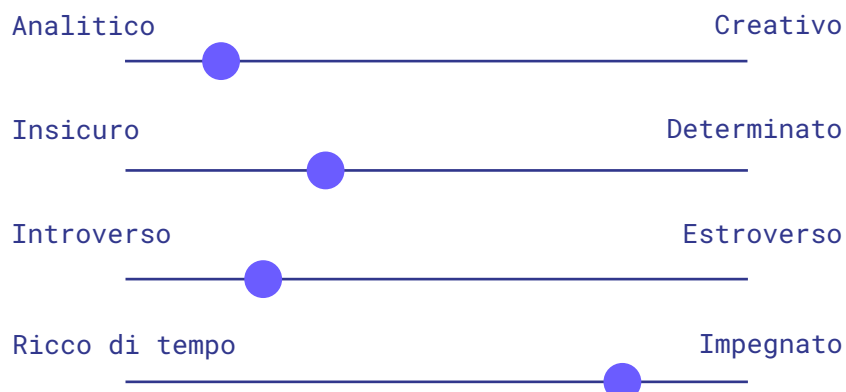
Social



Tools

Cercafase
GitHub
Atom

Personalità



LA SECCHIONA

MARIALUCIA STALLONE / STUDENTESSA

Età 24 anni

Città Firenze

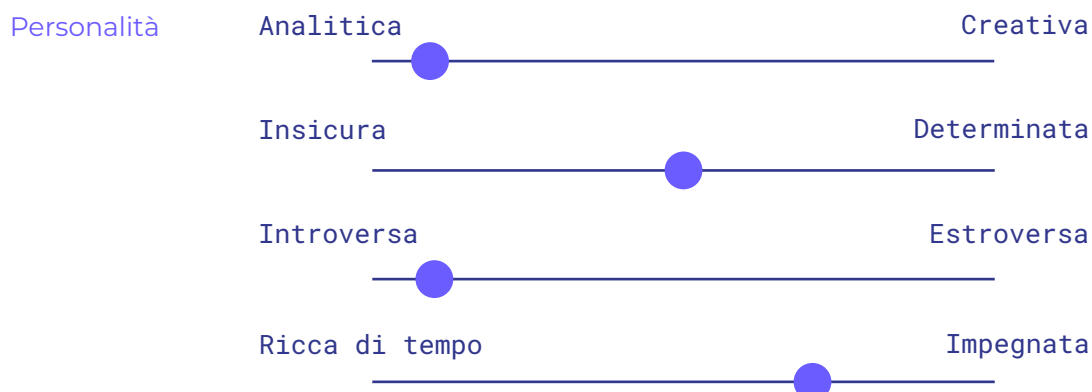
Ruolo Studentessa di Design

Interessi
Arte Digitale
Metaverso
LOTR

Hobby
Gaming
Lettura
Disegno

Social   

Tools
Pages
Adobe Illustrator
ProCreate





Two horizontal lines representing a browser address bar.



“Mi attira molto lo UI/UX Design, ma
devo capire meglio le mie attitudini”

“L’HO LETTO IN
UN LIBRO”





“Voglio creare un sito web per darmi
il giusto valore con la concorrenza”

“ASSOLUTAMENTE NO!”



IL BRONTOLONE

SILVIO BETUMERA / ARCHITETTO

Età 47 anni

Città Bologna

Ruolo Architetto Progettista

Interessi Spaghetti Western
Musica Classica
UX/UI Design

Hobby Modellismo
Ciclismo
Trekking

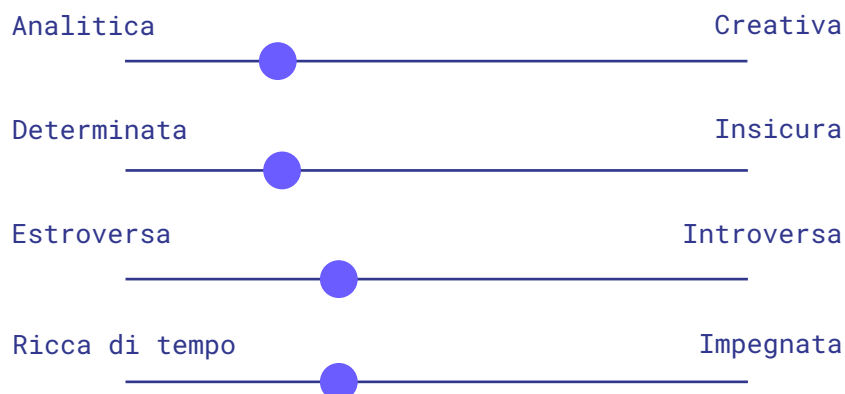
Social



Tools

AutoCAD
SketchUp
Revit

Personalità



APACHE99X_X

MARIO FALCE / STREAMER

Età 23 anni

Città Venezia

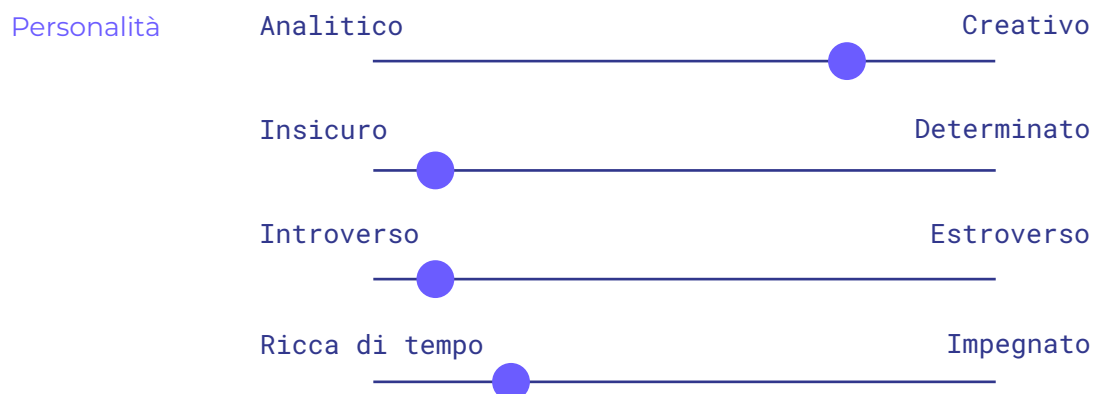
Ruolo Gamer online

Interessi
Techno
Gaming
Film

Hobby
Motion Graphic
Beat Production
Rugby

Social   

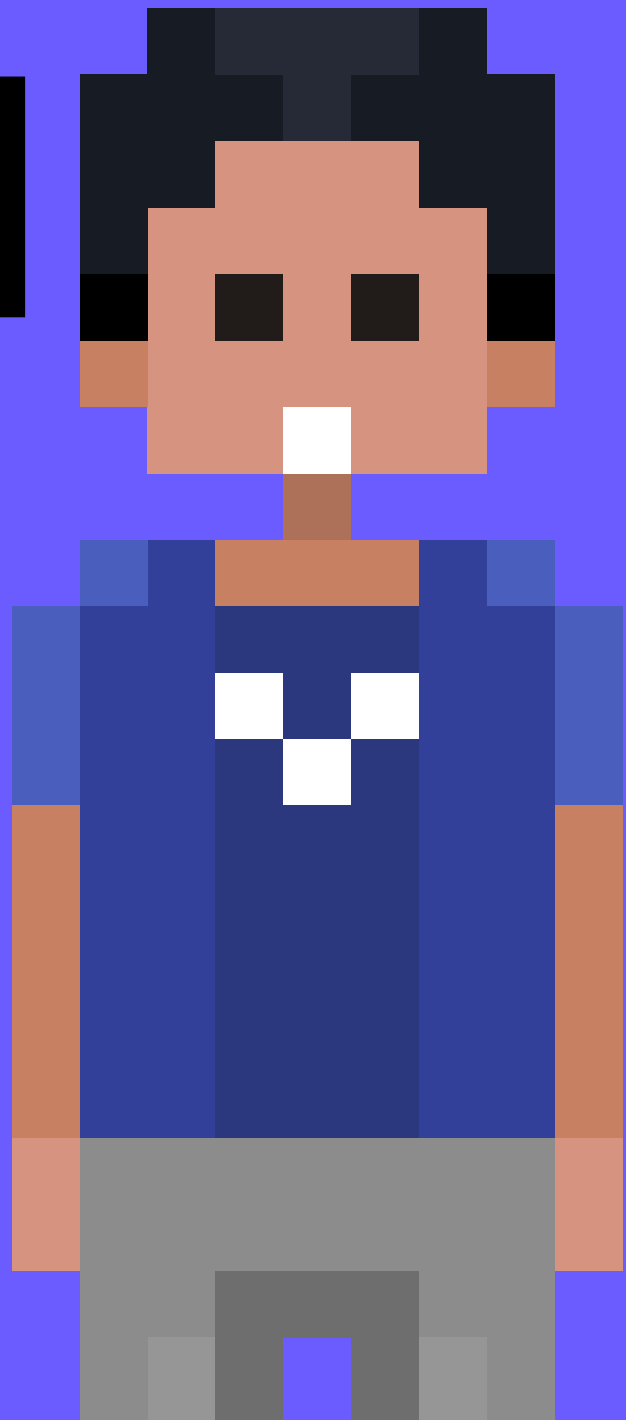
Tools
OBS Studio
Blender
Adobe AfterEffects





“Guadagno giocando ai videogames, ma vorrei poterne sviluppare uno tutto mio”

**“NON SONO
CICCIOGAMER89!”**



4.0

Progetto

4.1 CODING LAB

Introduzione

Sviluppato da Mekit Studio, Coding Lab è un progetto dedicato ai designer che vogliono sperimentare e avvicinarsi al mondo del coding. Il progetto consiste in un corso di apprendimento che nasce dall'idea principale di introdurre gli utenti ad argomenti di connessione tra la sfera del design e quella dello sviluppo, quali il Front End Web Development, il Creative Coding ed il Game Development.

Competenze in questi ambiti hanno ricadute dirette sull'esperienza che ogni user può avere nell'approccio con un prodotto digitale: si può progettare in modo più efficace se si comprende come funziona la tecnologia che sta alla base del prodotto a cui si sta lavorando.

Il progetto si svilupperà attraverso laboratori teorici e pratici, in cui gli studenti potranno mettersi in gioco attraverso l'aiuto dei tutor.



4.2 SVILUPPO DEL CONCEPT

Introduzione

Terminata la fase di ricerca si è passati quindi allo sviluppo del concept. Obiettivo di questa tesi è quello di sviluppare un concept legato allo sviluppo di un'identità visiva adatta alla comunicazione online e offline del progetto Coding Lab.

A tal proposito è stato necessario stilare delle linee guida da rispettare durante lo sviluppo del concept.

Linee Guida

Criteria fondamentali

Dall'analisi dei caso studio e delle personas sono emerse alcune considerazioni importanti.

In primo luogo, dall'analisi delle Personas è emerso come per molti designer sia fondamentale comprendere ciò che si cela alla base dei loro progetti. Tale esigenza è rispettata dal progetto stesso che vuole fornire gli strumenti adatti a tutti coloro che vogliono approfondire il mondo del coding.

L'analisi dei casi studio invece è stata fondamentale per comprendere l'impostazione visiva da adottare nella comunicazione e per analizzare i punti di forza e debolezza delle attuali academy online a livello comunicativo.

Da ciò è stato possibile stilare tre criteri fondamentali da rispettare durante lo sviluppo del concept.

Semplicità e
chiarezza

Il concept deve risultare visivamente semplice e non ridondante in modo da attirare l'attenzione e guidare l'utente nella comprensione del progetto che vi sta alla base.

Intuitività

Il concept deve rimandare visivamente agli argomenti trattati durante il corso e allo scopo primario del progetto: imparare l'arte del creative coding.

Motivazione

Essendo il creative coding una disciplina spesso percepita come estremamente difficile, il concept deve adottare un tone of voice motivazionale ed ottimista.

Concept

Motivazione, ricerca e sperimentazione

Il concept consiste nello sviluppo di un'identità visiva per il laboratorio di creative coding, Coding Lab, sviluppato da Mekit Studio.

Sulla base delle linee guida il progetto fa riferimento alla celebre frase "si può fare!" pronunciata dal Dottor Frederick Von Frankenstein, l'attore Gene Wilder, nell'omonimo film Frankenstein Junior. Un' esortazione a dare il massimo, un incito motivazionale con l'intento di trasmettere il classico senso di appagatezza quando stupiti ci rendiamo conto di essere riusciti a completare un obiettivo che credevamo impossibile. Questo perché il mondo dei codici è un mondo vasto e a primo impatto anche complesso. Si favorisce inoltre la ricerca, come avviene nei laboratori in cui con perseveranza e buona volontà si ottengono incredibili risultati.

FIG. 64 ►
FRANKENSTEIN
JUNIOR
(©Gruskoff
Venture
Film)





4.3 VISUAL IDENTITY

Introduzione

Pixel

La scelta dei visual è stata dettata dalla necessità di rappresentare visivamente i tre ambiti di progetto in modo efficace e intuitivo. Lo stile pixelato serve per consolidare ancora di più un'immagine coordinata che ha come studio i linguaggi di programmazione e quindi l'informatica stessa. Andando a richiamare le prime grafiche storiche generalmente in 8bit e 16bit.

Front End

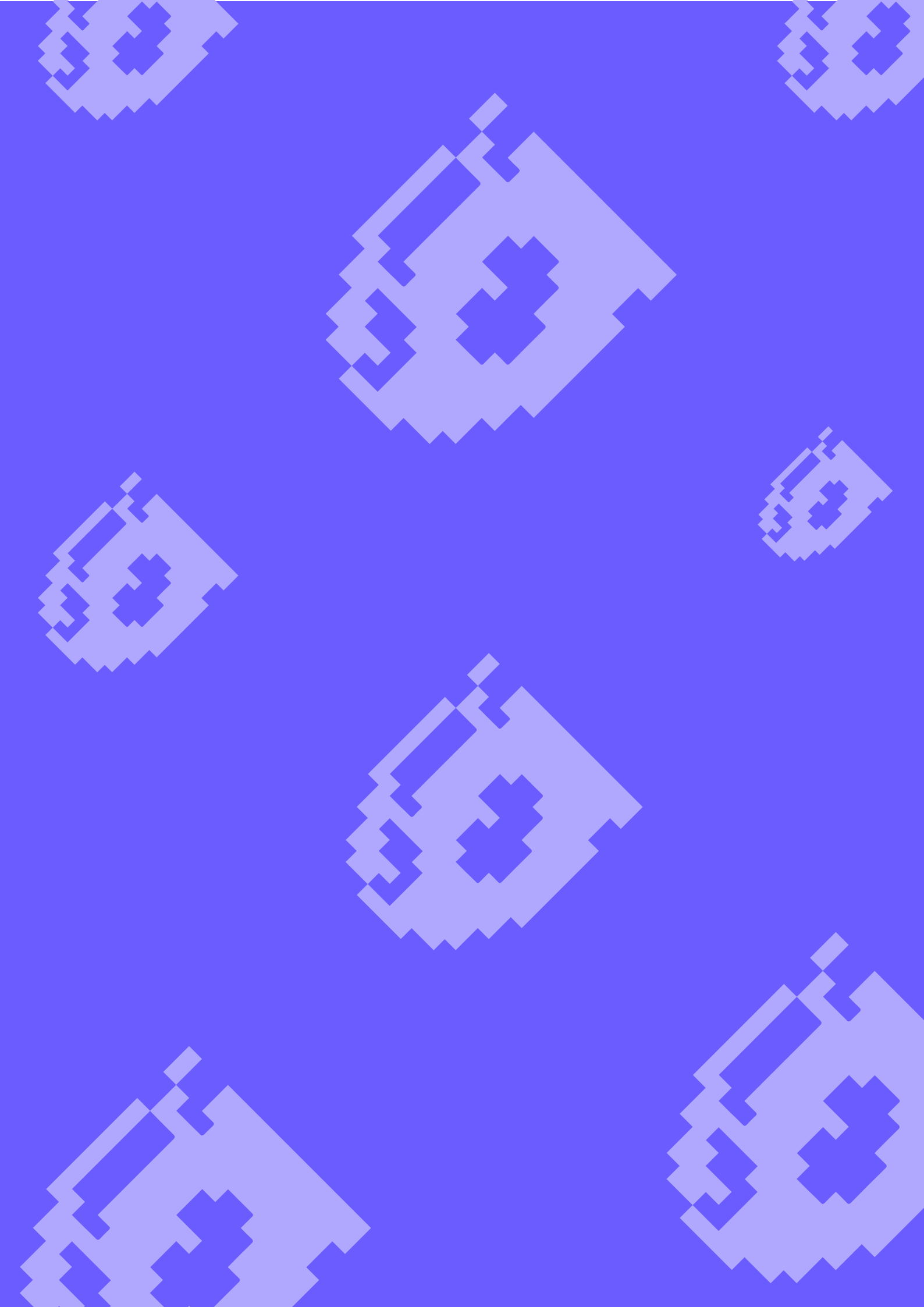
Il Front End fonde elementi concreti come fogli di testo, simboli ed elementi di statistica, con elementi più concettuali che richiamano al calcolo e l'informatica.

Creative Coding

Il Creative Coding è caratterizzato da visual astratti che ricordano elementi generativi in continua evoluzione e sviluppo. Dimostrando di fatto le infinite possibilità che questo ambiente garantisce.

Game Development

Il Game Development richiama elementi iconici dei videogiochi più famosi come Bomberman e Super Mario e altri elementi che hanno segnato la cultura pop degli ultimi decenni.



Logo

Ideogramma e Naming

Ideogramma

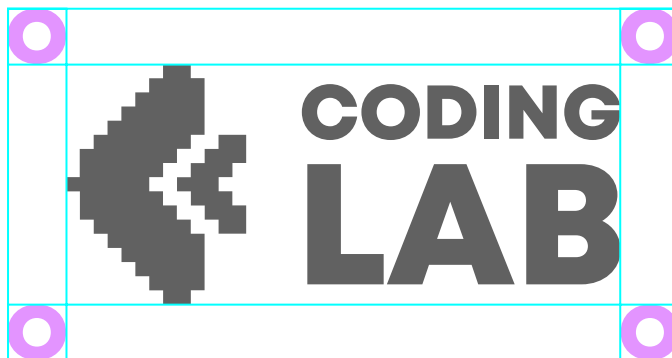
L'ideogramma è realizzato tramite l'utilizzo di due simboli di minore. Questi simboli sono diffusissimi nell'ambito della sviluppo e stesura dei codici in quanto contenitori delle righe di testo di tutti i linguaggi di programmazione. Nello specifico il simbolo del minore, funge da contenitore per i blocchi di testo sotto la forma di apertura.

Il simbolo viene quindi sdoppiato e affiancato uno all'altro ridimensionando il secondo creando così un senso di movimento e transitorietà. Leggermente diversi nella forma i simboli ricordano anche le iniziali del naming del progetto: Coding Lab

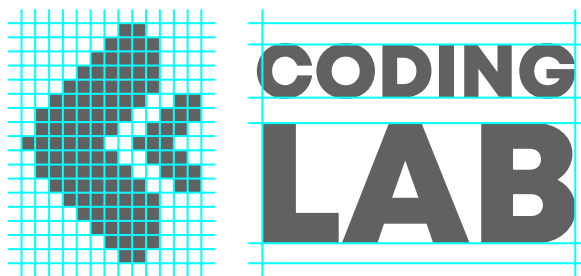
Naming

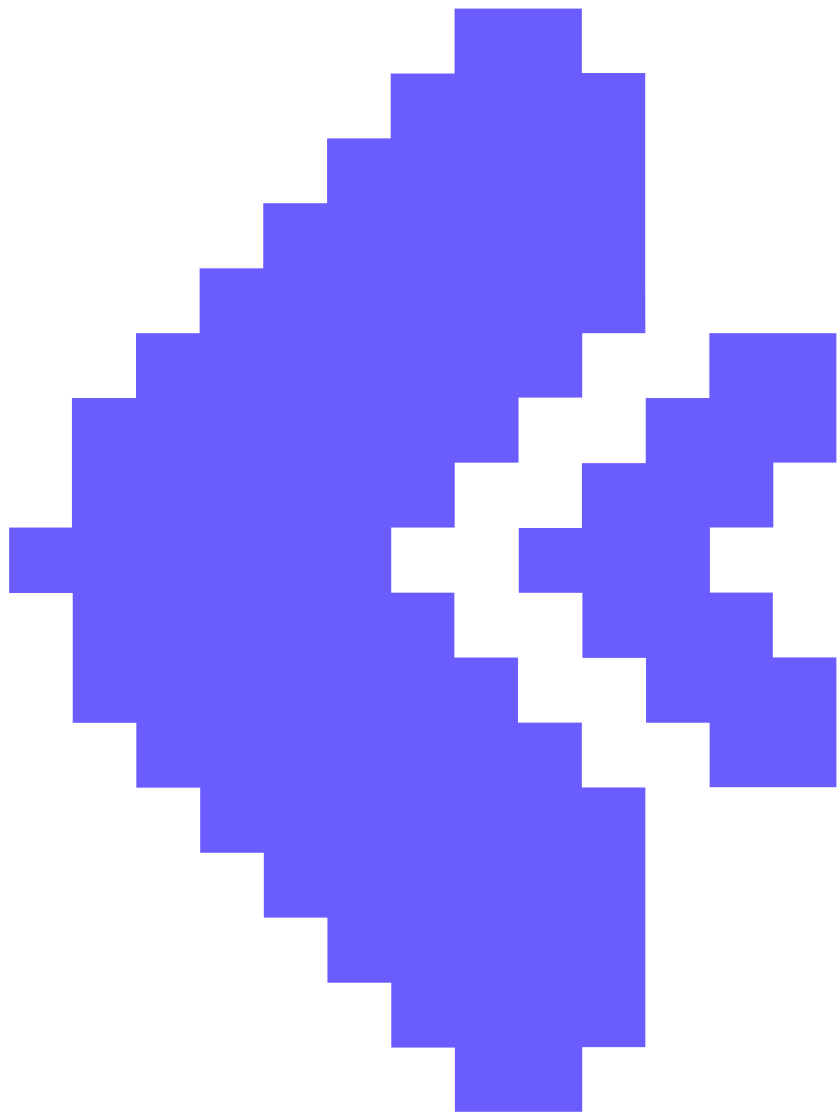
Il naming è un semplice e diretto rimando all'ambito del progetto: il mondo del codice, visto in un ambiente da laboratorio.

Area di rispetto



Costruzione





CODING LAB

Claim

*“Si può fare!, Impossibile?Provaci!,
Un gioco da ragazzi!”*

“3 Claim”

I claim fanno diretto riferimento al concept, invogliare la gente creando stupore. “Si può fare!” ti fa chiedere prima di tutto “cosa? Cosa posso fare?”. Una volta incuriosito la frase assume un valore motivazionale e di supporto incentivando così la decisione di entrarne a far parte, perché comunque sai che se qualcuno ha detto che si può, allora si potrà veramente. Stesso discorso vale per gli altri due claim Impossibile?Provaci! e Un Gioco da Ragazzi! L’invito anche qui è sempre quello di favorire il mettersi in gioco, sotto forma però di sfida: “Se pensi che sia impossibile perché non ci provi?”

Se è un gioco da ragazzi, può farlo chiunque, posso farlo anch’io. Questo specifico claim è stato pensato per l’ambito del Game Development in quanto contenente la parola stessa gioco.

Front End

SI PUÒ FARE!

Creative
Coding

***IMPOSSIBILE?
PROVACI!***

Game Dev

***UN GIOCO
DA RAGAZZI!***

Tipografia

Scelta dei font

I font utilizzati per l'immagine coordinata sono tre, specifici per le applicazioni.

Mon Heavy
Demo

Il Mont Heavy Demo è utilizzato per il naming del logo.

Montserrat

Il Montserrat utilizzato è da usare in primis per i claim all'interno delle sponsorizzazioni e in generale per quei testi che necessitano un impatto visivo maggiore essendo questo un font sans-serif elegante e maturo che riesce a trasmettere modernità e solidità. La lunghezza di linea permette inoltre una disposizione non centrale ma comunque impattante.

Roboto Mono

Il Roboto Mono invece, variante della famiglia Roboto, è un font graziato monospace in grado di mantenere una chiara leggibilità anche nel caso in cui i testi descrittivi sono piccoli e lunghi. Un font chiaro e leggibile della famiglia dei TypeWriter che venivano montati sulle macchine da scrivere in cui i singoli caratteri erano tutti di egual misura. Ideale nella rappresentazione di calcoli e linguaggi di programmazione.

Mont Heavy
DEMO

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
1234567890.,:;!?

Montserrat

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
1234567890.,:;!?

Roboto Mono

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
1234567890., : ; ? !

Palette cromatica

Ispirazione e colori

Ispirazione

La palette cromatica prende ispirazione dai Code Editor come Vs Code e Atom, in cui a gli elementi di testo dei linguaggi di programmazione viene spesso accostato un colore. Questa tecnica permette una decifrazione più intuitiva e immediata, semplificando il caos che generano documenti di testo interamente formati da lettere, simboli e numeri. Questi colori sono spesso modificabili e non univoci. La scelta è stata dettata dal gusto personale e dal rapporto che questi colori hanno con l'ambito che rappresentano.

Celeste

I celesti per il front end in quanto i coloriti chiari e freddi della palette ricordano la semplicità e la formalità dei fogli di testo. Non generano troppo contrasto e mettono in risalto i testi che sono gli elementi fondamentali senza cui i linguaggi avrebbero senso.

Verde

I verdi per il creative coding, poiché il verde è un chiarissimo rimando alle stringhe di codice con 0 e 1 che siamo sempre stati abituati a vedere in informatica, e anche nei film (Matrix). Con 0 e 1 possono essere rappresentati tutti i numeri, di conseguenza tutti i caratteri. Un chiaro rimando a cosa può fare il codice creativo.

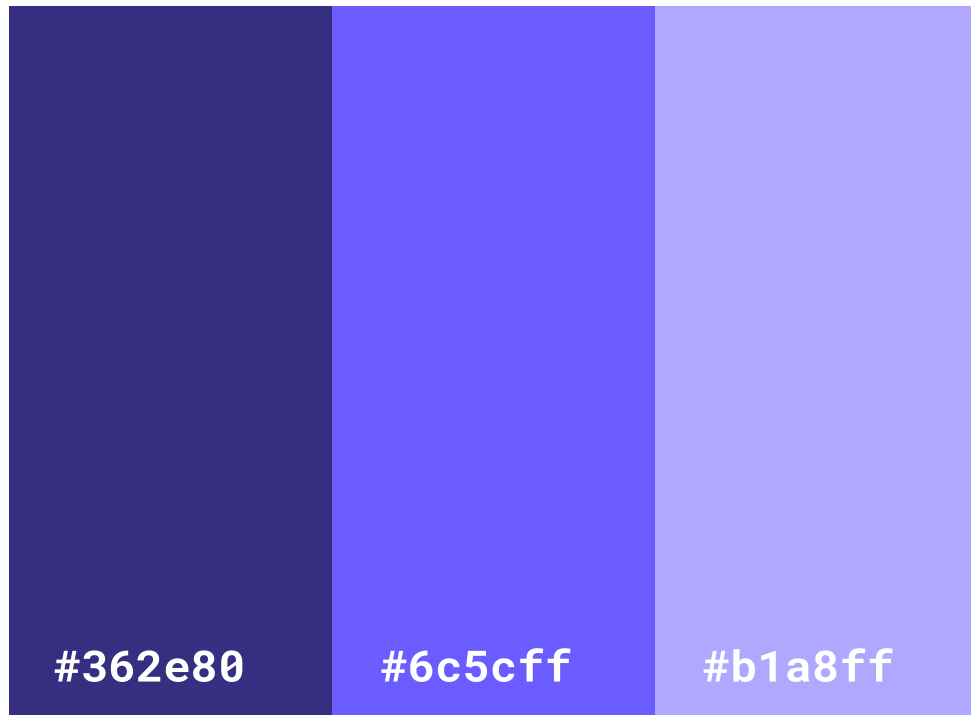
Giallo

I gialli per il game development in quanto colori più accesi e giocosi creano maggiore contrasto dando così quel senso di ambiente luminoso e divertente quale quello del gioco. Ispirato a elementi diffusi e iconici del mondo dei videogame come il celebre Pac Man o le monete collezionabili di Super Mario.

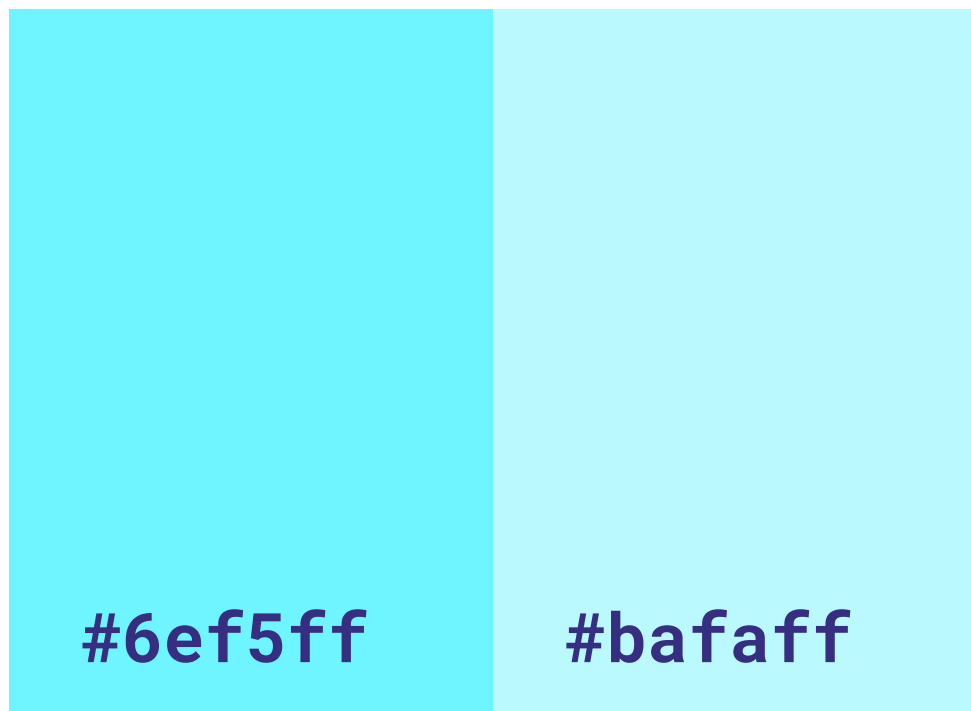
Viola

I viola infine, più corposi e decisi, sono colori che racchiudono in sé gli altri tre e vengono utilizzati quindi per l'immagine coordinata generale del progetto.

Generale



Front End



Creative
Coding

#8cff6d

#c9ffba

Game
Development

#ffd20a

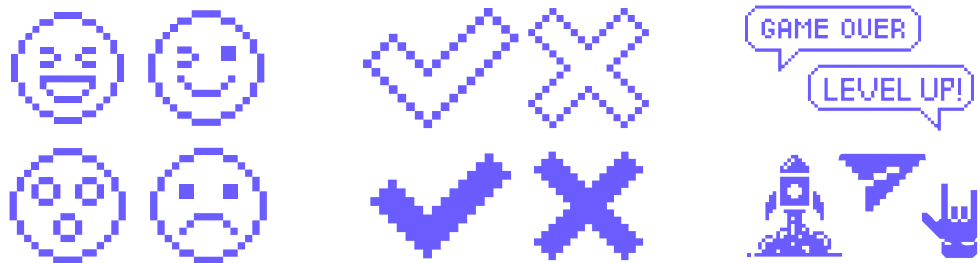
#ffe057

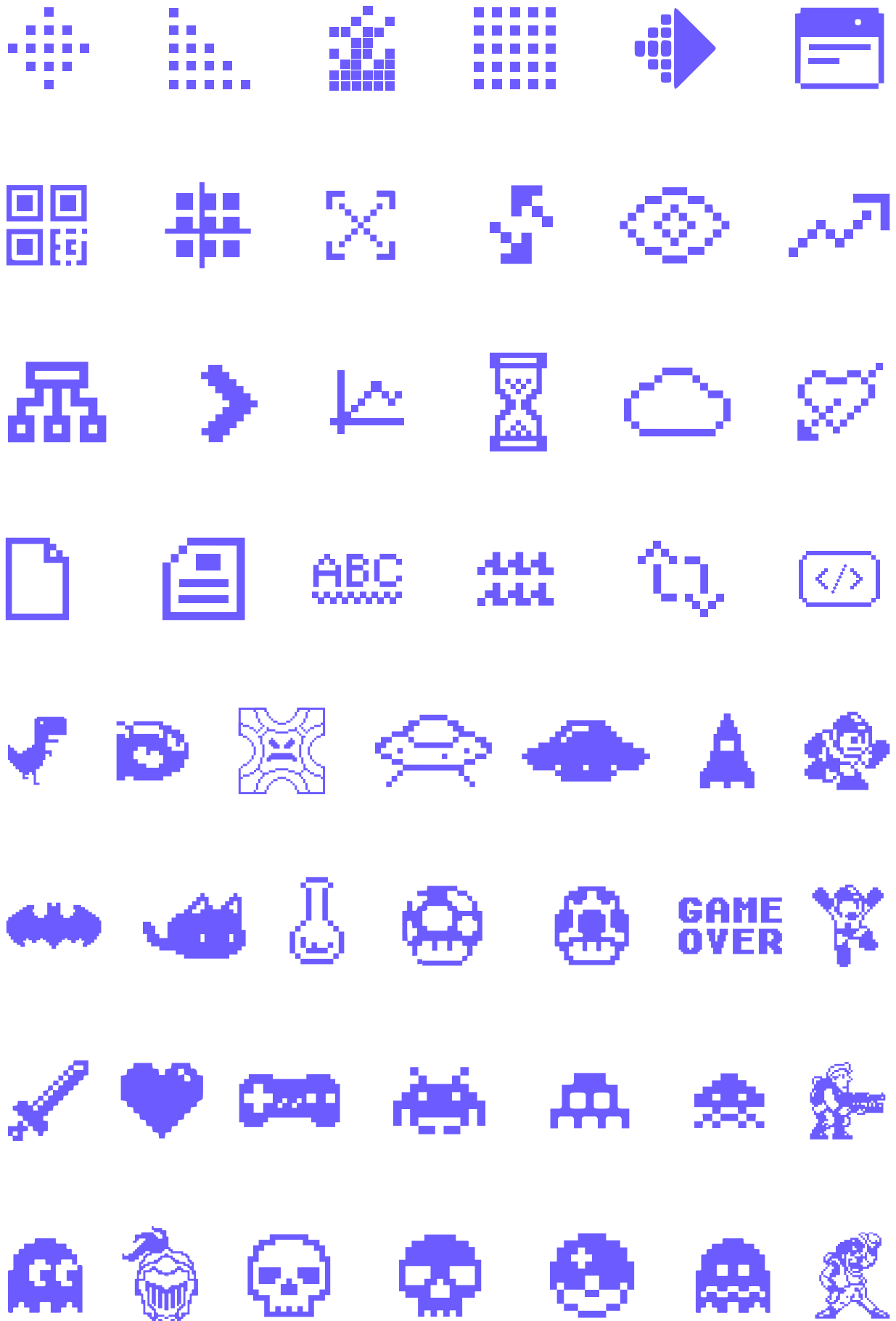
Visual

Icone

Icone

Per completare la comunicazione sono state sviluppate delle icone in stile pixel che rimandano ai tre ambiti legati al Front End, al Creative Coding e al Game Development.





4.4 COMUNICAZIONE

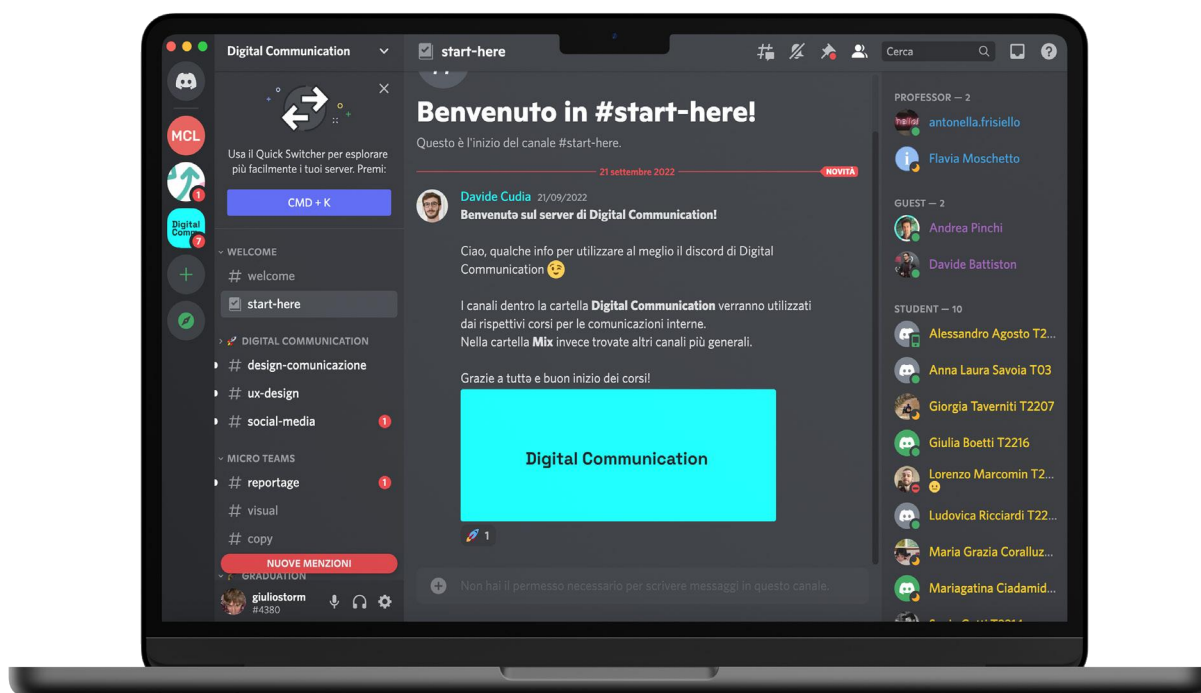
Strategia e canali

Per la strategia comunicativa si è optato prevalentemente per l'utilizzo di canali online. Creare un passaparola tra i più curiosi all'interno di community già solide e conosciute fa sì che la notizia possa arrivare a un amico, e a un amico di un amico. Per questo motivo i canali di comunicazione scelti sono: **Discord**, al cui interno i canali collaborativi di Sherpa Community e Professione Creativo generano quel passaparola. **LinkedIn**, tramite i canali di Mekit Studio, Professione Creativo e Sherpa. **Instagram** con le sponsorizzate di @mekit.studio e @professione.creativo. **Youtube** e **GitHub** come repository e tutorial. Si è anche optato per canali offline da utilizzare come **supporti**.



Discord

Discord è un canale dedicato in cui i partecipanti ai corsi e utenti esterni possono **comunicare** scambiandosi consigli, organizzando eventi e tenersi aggiornati **in tempo reale** favorendo così la nascita della community.





Sherpa Community

- MCL
- Info
 - welcome
 - annunci
 - #conosciamoci
 - arrivi
- Community
 - generale
 - suggerimenti-e-aiuto
 - self-promo-portfolio

#conosciamoci
Agostino Toriello 03/11/2022
Ciao a tutt*, sono Agostino, UX/UI Designer @Revolve
Felice di essere qui! 🍷🍷

Alcuni miei link:
linkedin <https://www.linkedin.com/in/agostinatoriello/>
behance <https://www.behance.net/agostinatoriello>
instagram <https://www.instagram.com/mos4ico/>

Linda Raley 04/11/2022
Ciao a tutti, grazie di avermi accolto in questa community!
Leggo [tanti] libri [narrativa + UX], scrivo racconti e articoli, cerco lavoro nel settore UX/UI da remoto. Da pochi mesi mi sono avvicinata a questo "mondo" ma l'ho apprezzato da subito e cerco di fare quanta più formazione possibile

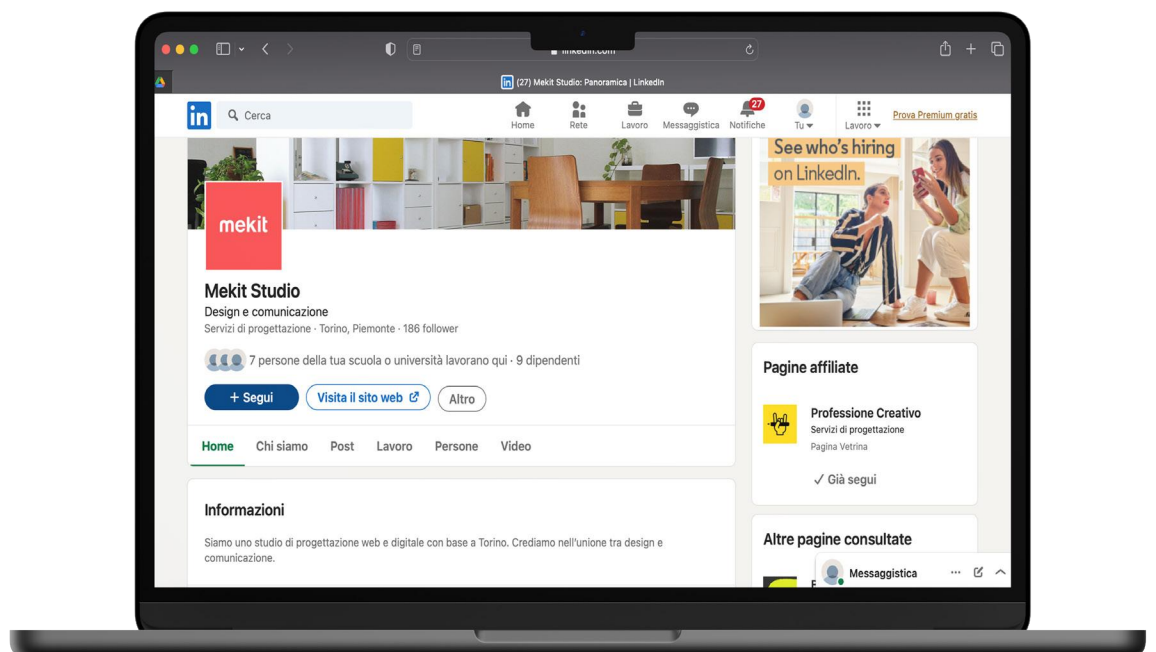
Matteo Muscas 10/11/2022
Ciao a tutti e tutti, sto cambiando carriera passando dalla [sic] su questo gruppo. Ci

- MENTOR - 1
 - Andrea Pinchi
- ONLINE - 83
 - //gioele//
 - 00. Simone Mariano
 - 03. Andrea Bellizzi
 - 10r3n20
 - Ambivert
 - Adzoine
 - Alessandro Demaria
 - Alessandro Lagasca
 - Alessandro98
 - Remote
 - Alessio
 - Alessio La Forgia
 - Enigmatic Thunder
 - Alessio Romualdo
 - Pensando a cose

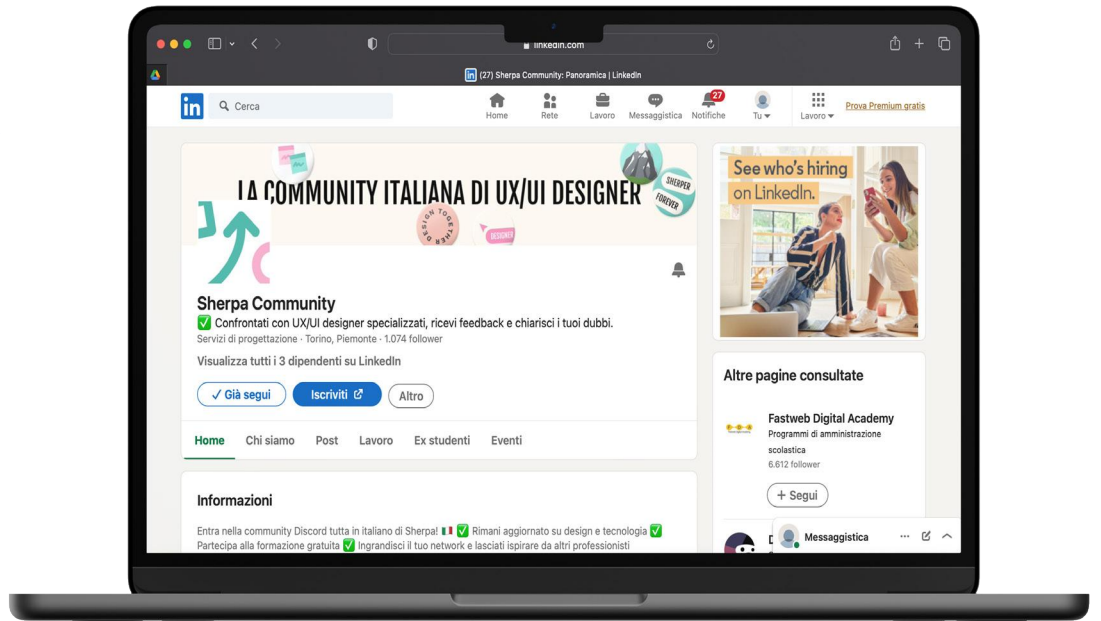
LinkedIn

LinkedIn, tramite le pagine di **Mekit**, **Professione Creativo** e **Sherpa** verrà utilizzato come forum di accesso all'ambiente, luoghi in cui esplorare tramite esperienze già vissute, professionisti e neofiti del settore cosa vuol dire avvicinarsi al mondo del coding ed entrare a far parte di corsi di formazione e introduzione all'ambiente come Coding Lab.

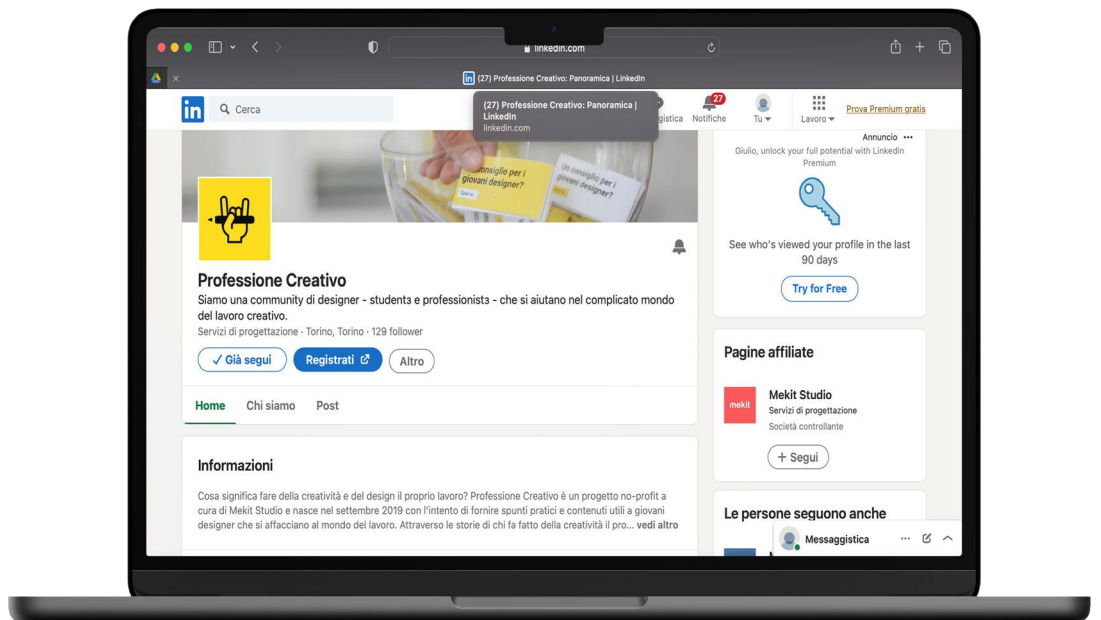
Mekit
Studio



Sherpa
Community



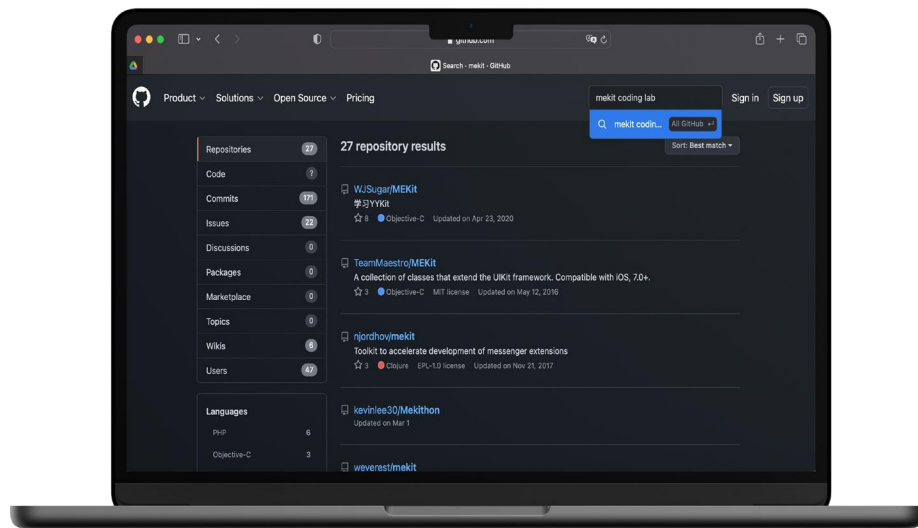
Professione
creativo



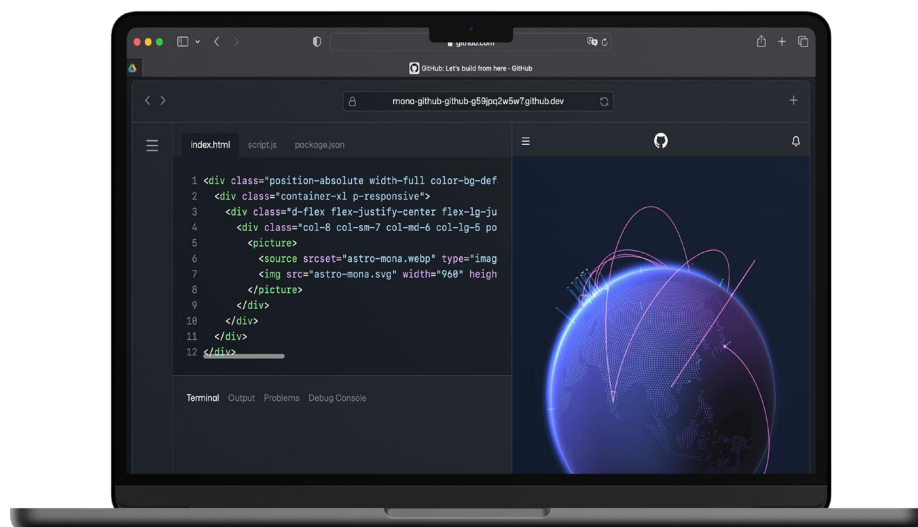
Youtube e GitHub

YouTube e Github fungeranno invece da **repository** e **tutorial** per micro tematiche. Un catalogo di informazioni sempre disponibile da poter consultare e utilizzare per i propri progetti.

GitHub repository



GitHub tutorial









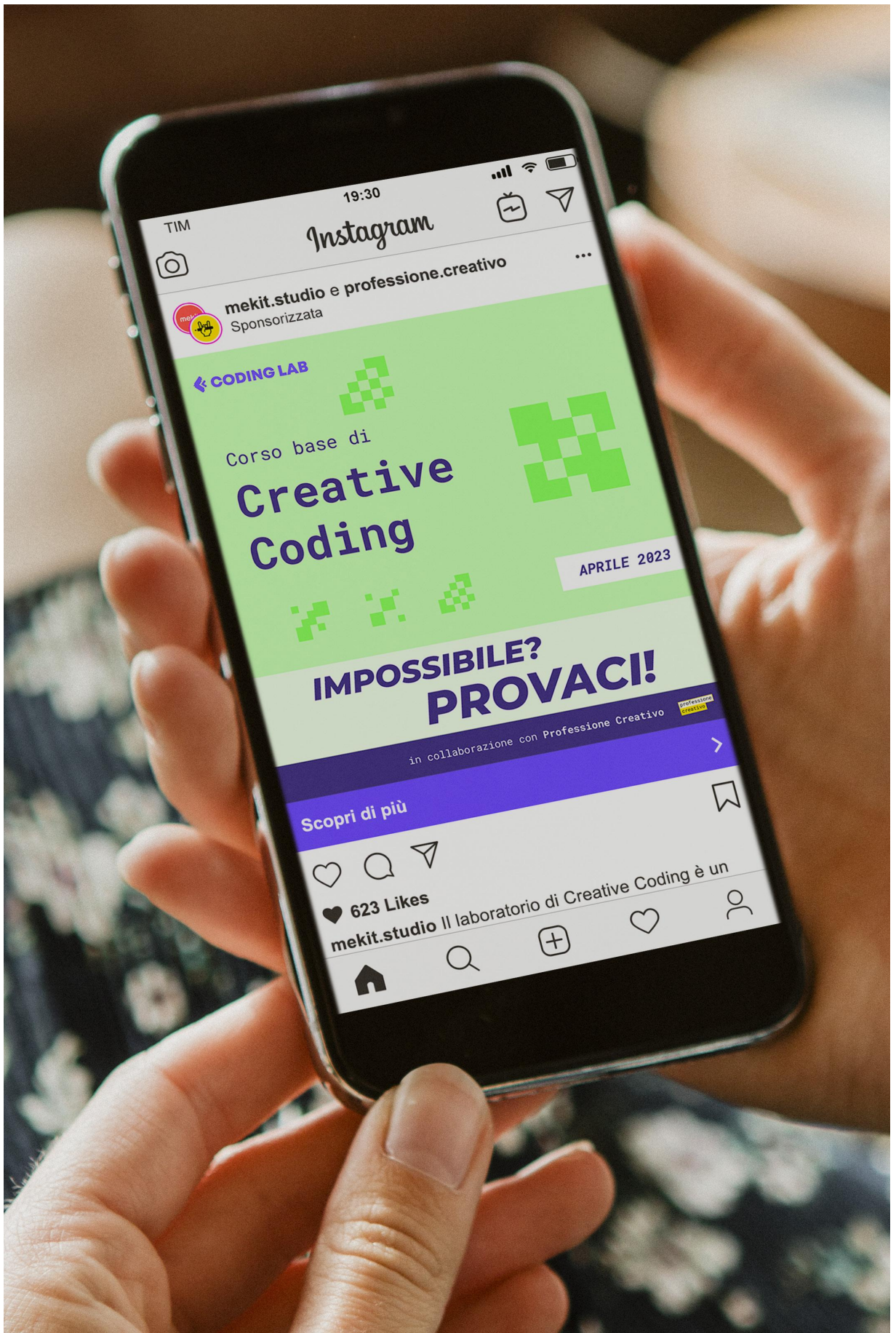
Instagram

Per comunicare il progetto su Instagram non si è creata una pagina dedicata in quanto progetto in via di sviluppo. Si utilizzeranno invece le sponsorizzate delle pagine **mekit.studio** e **professione.creativo** per pubblicizzare il progetto in uno dei canali social più influenti del momento. Tramite queste, si potrà accedere al **Google Form** con cui poter inviare la richiesta di iscrizione.







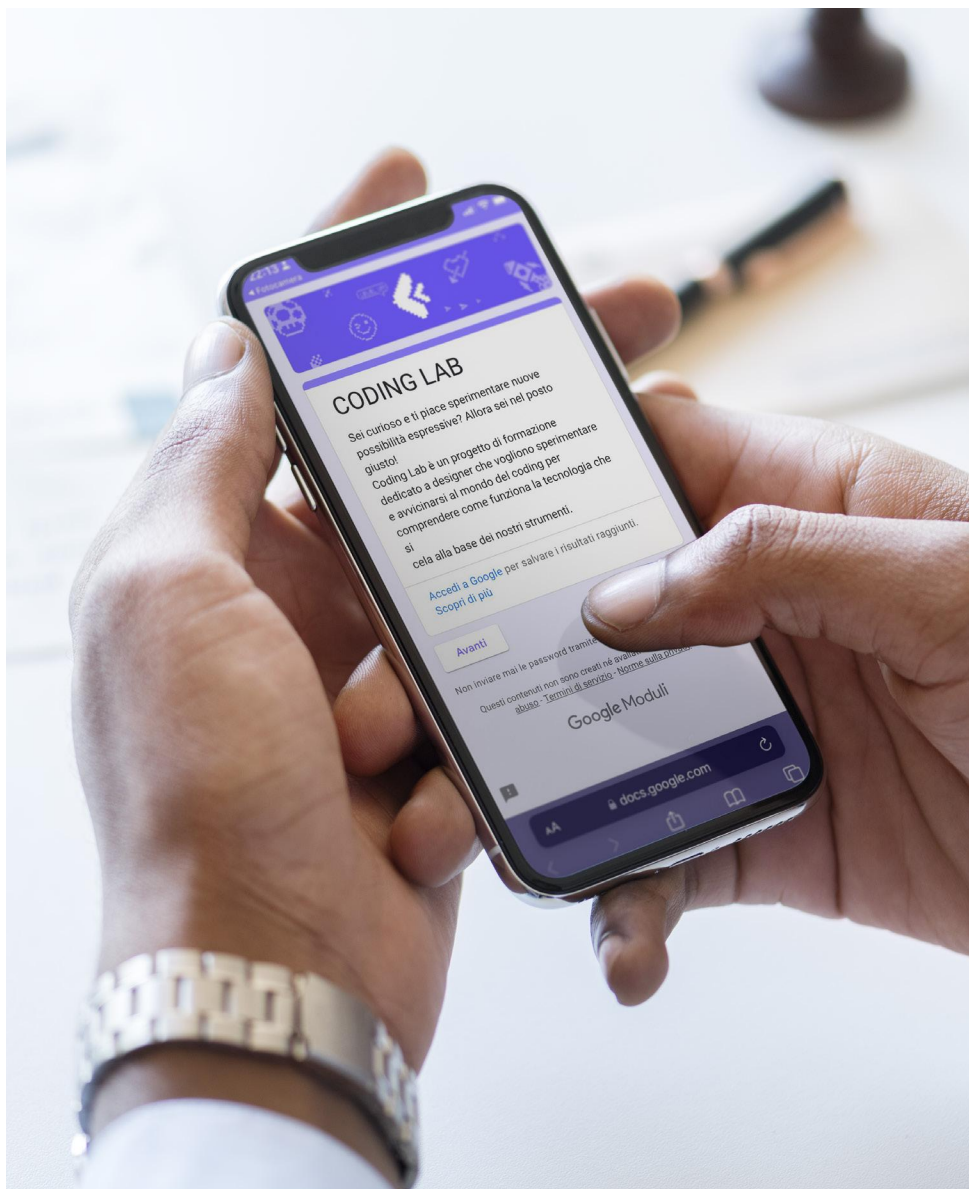




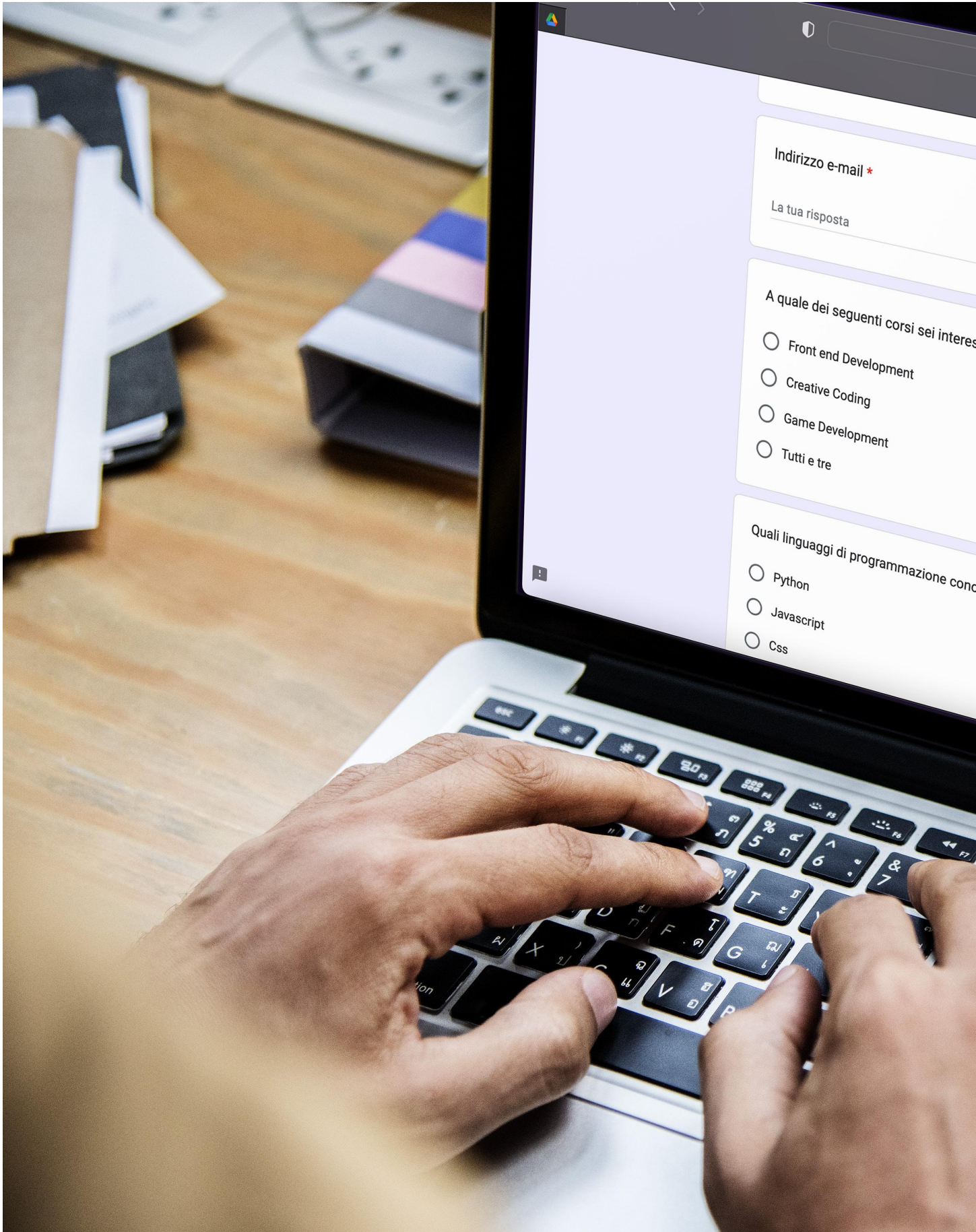


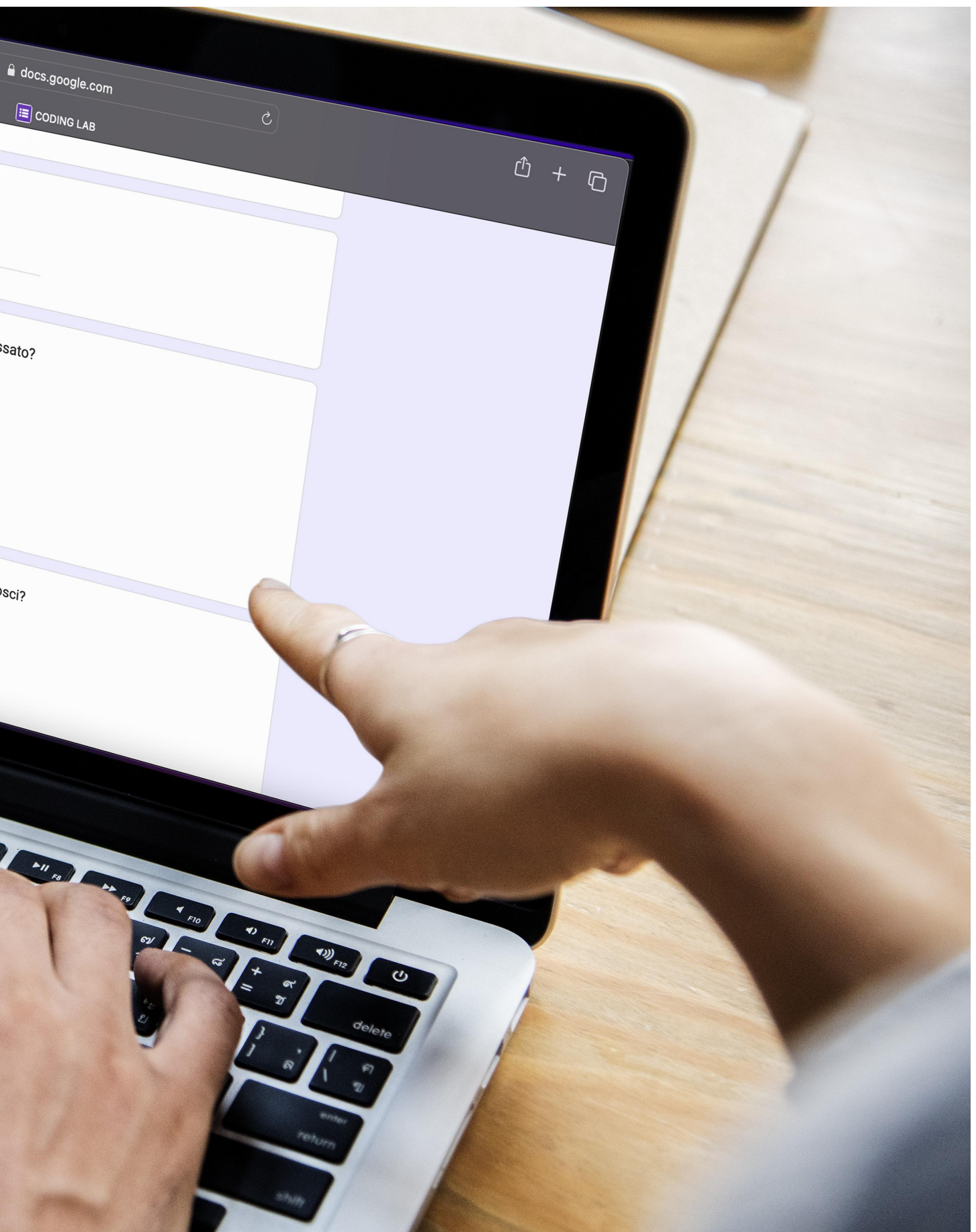
Google Form

La modalità di accesso al corso avverrà tramite un **Google Form** che, grazie alle sponsorizzate o tramite il passaparola all'interno delle community **LinkedIn** e **Discord** di Sherpa Community e Professione Creativo, garantirà, attraverso un semplice questionario, la richiesta di partecipazione per i più curiosi.









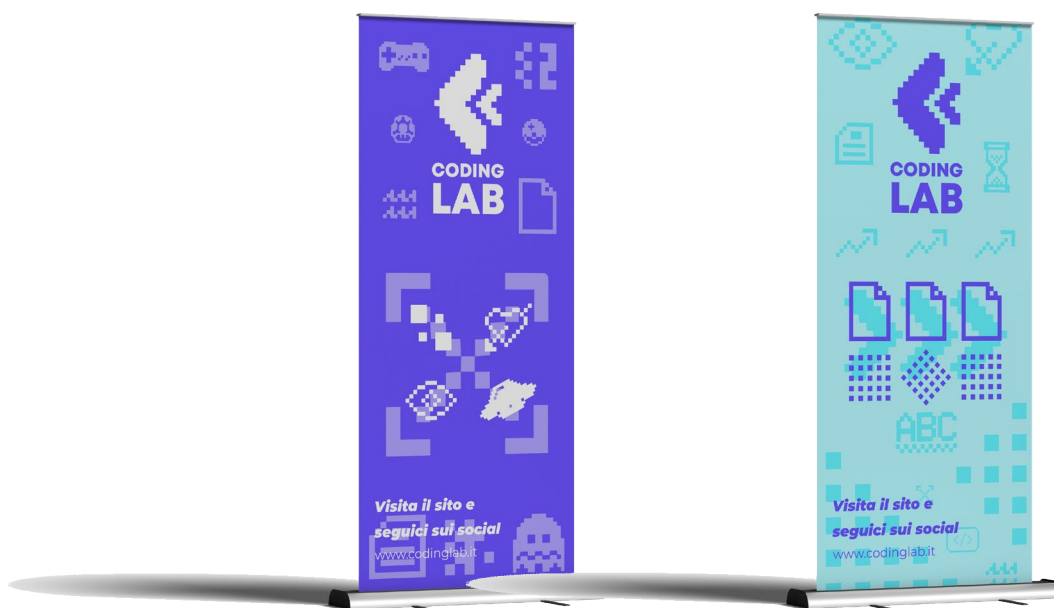
Supporti

Roll up, Badge, Trifold

Banner Per i supporti, **indicativi** specifici dei tre mini corsi, vengono utilizzati dei roll-up da presentare durante il periodo di laboratorio e bandiere, indirizzando così più facilmente i partecipanti nelle aule di studio e dando **un maggiore contesto** all'ambiente lavorativo.

Badge I **badge** vengono utilizzati dai **tutor** per indicare la loro posizione ed essere sempre un punto di riferimento importantissimo per gli studenti.

Trifold Il **trifold**, infine, è un **supporto esplicativo** e indirizzante per uno specifico corso, compreso di spiegazione dell'itinerario, date e modalità di iscrizione.





**CODING
LAB**

**GAME
OVER**

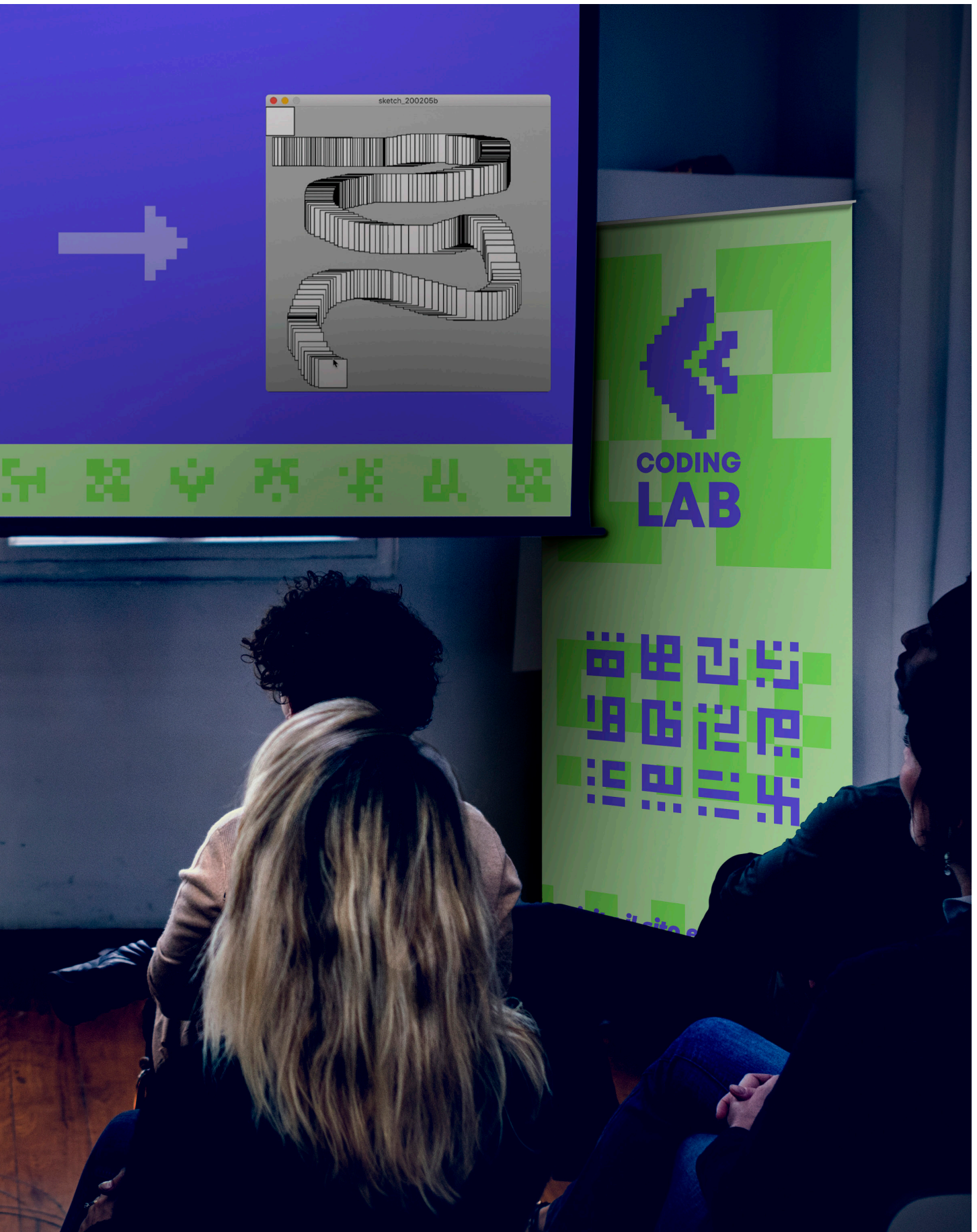
Seguici sui social

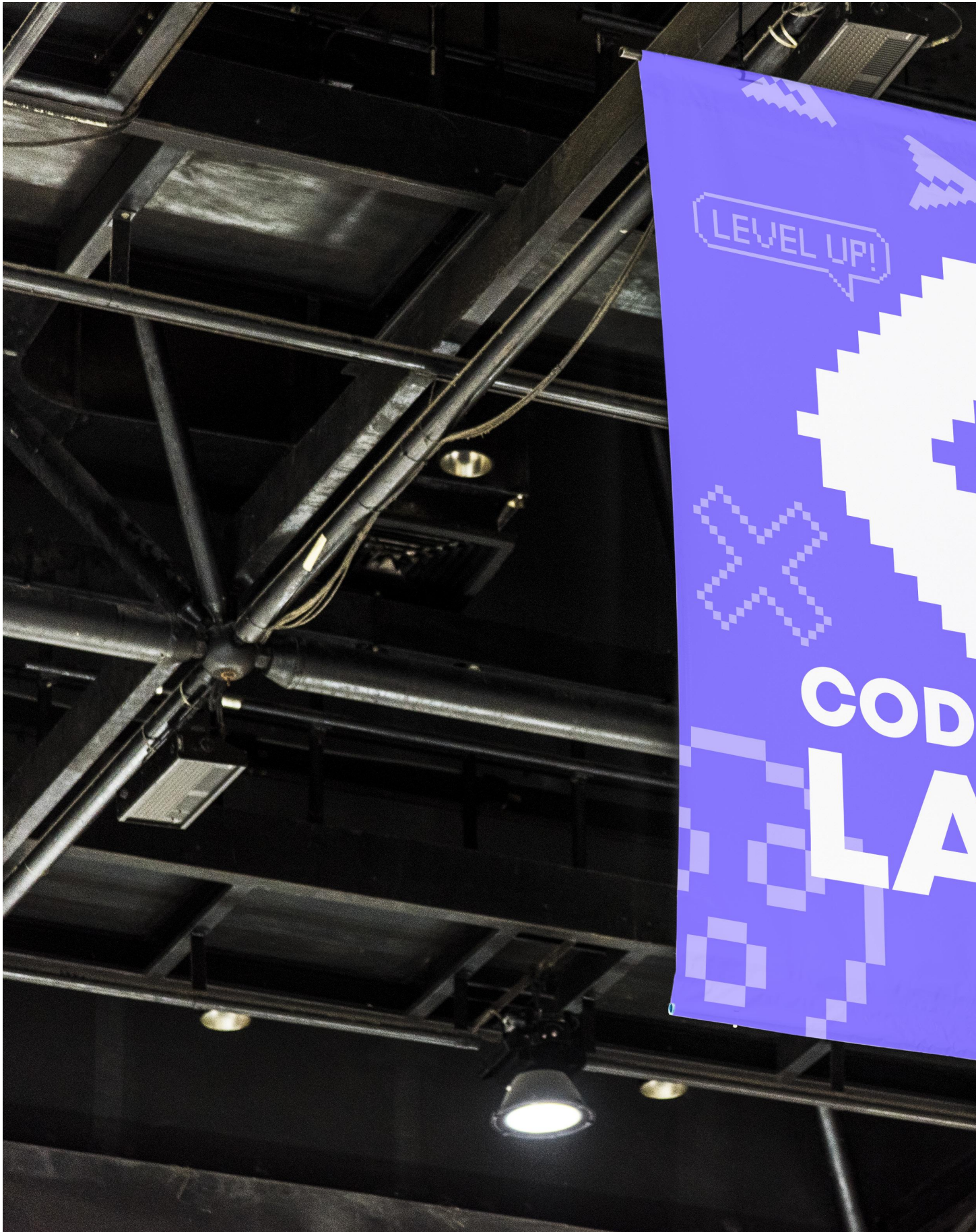
www.codinglab.it



```
void setup() {  
  size(500, 500)  
}  
  
void draw() {  
  rect(mouseX, mouseY, 50, 50)  
}
```

◀ CODING LAB [Icons]









Il laboratorio di **CREATIVE CODING** si usa il computer per creare diversi **VISUAL GENERATIVI** e/o interattivi; è un percorso progressivo di sperimentazione che combina elementi formali (punti, rette, forme, colori...), interazioni uomo-macchina e funzioni randomiche.

Servirsi di un computer per disegnare attraverso l'uso del codice spesso produce risultati inaspettati e sorprendenti.

Il laboratorio è rivolto ai designer che vogliono esplorare nuove possibilità e a persone che hanno una predisposizione (anche se piccola) per la matematica o altre materie scientifiche.

Esplora i fondamenti della programmazione, realizza la tua serie personale di sketch generativi con JavaScript e p5.js

1° Laboratorio - 7 Aprile 2023
Introduzione al creative coding e P5.js

2° Laboratorio - 14 Aprile 2023
Loop, griglie e funzioni random

3° Laboratorio - 14 Aprile 2023
Processare le immagini

4° Laboratorio - 14 Aprile 2023
Animazione e Interattività



ISCRIVITI SUBITO!
I posti sono limitati, affrettati a prendere il tuo scansionando il codice QR



professione
creativo



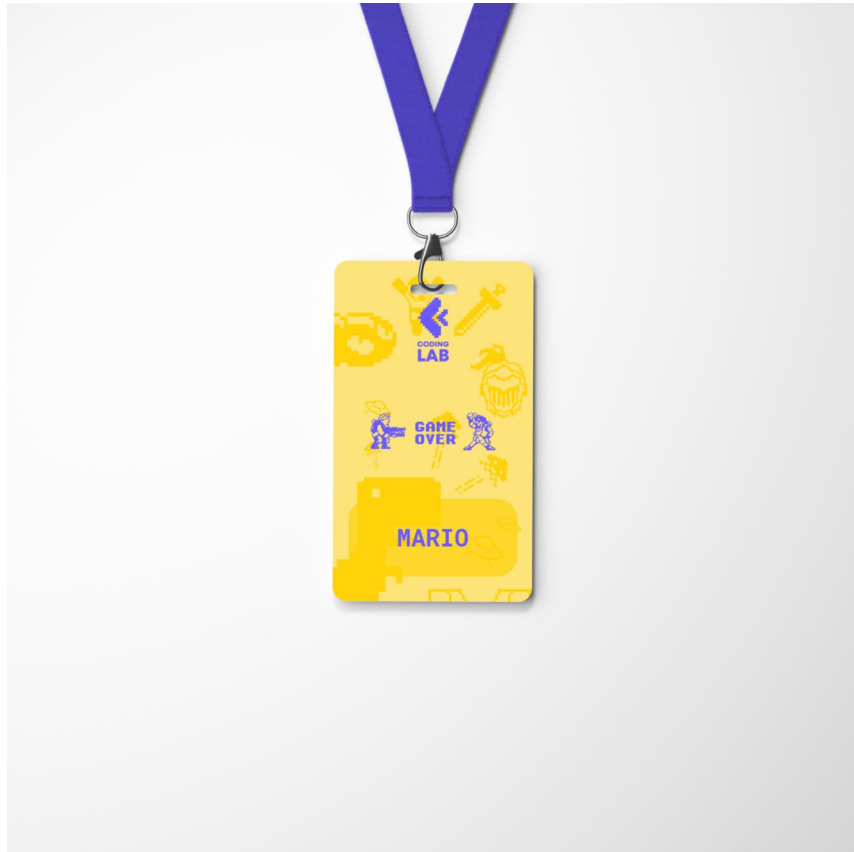
CONTATTI

hello@mekit.it
Sherpa Community
in Professione Creativo



Sherpa
Com





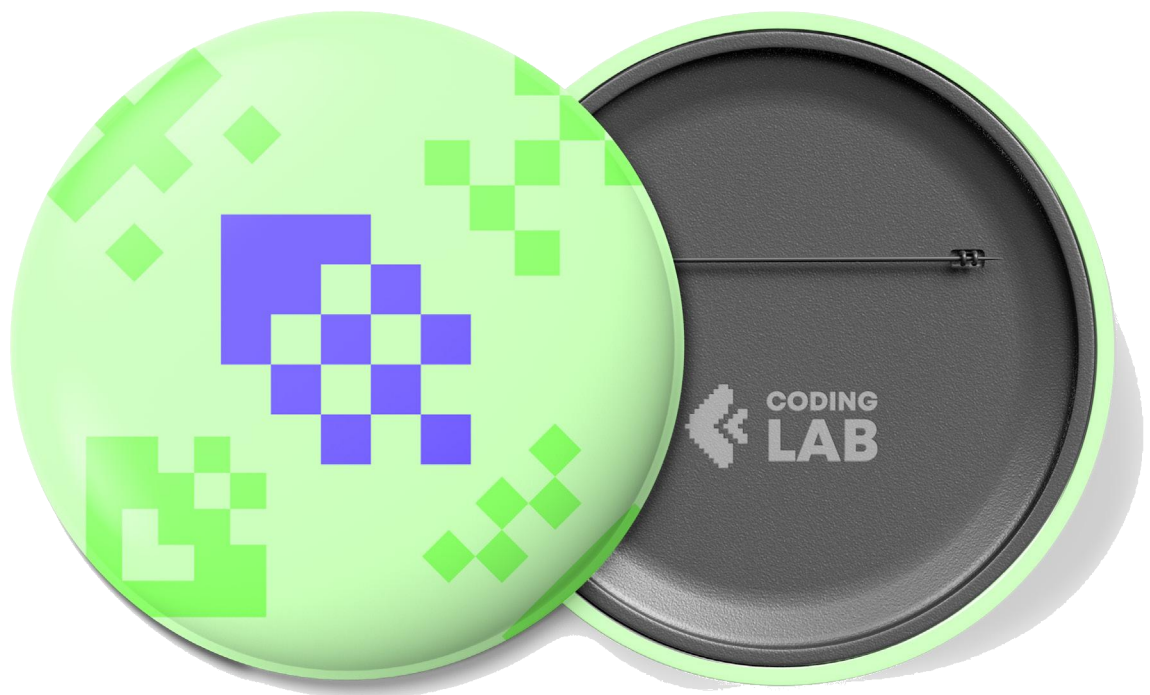


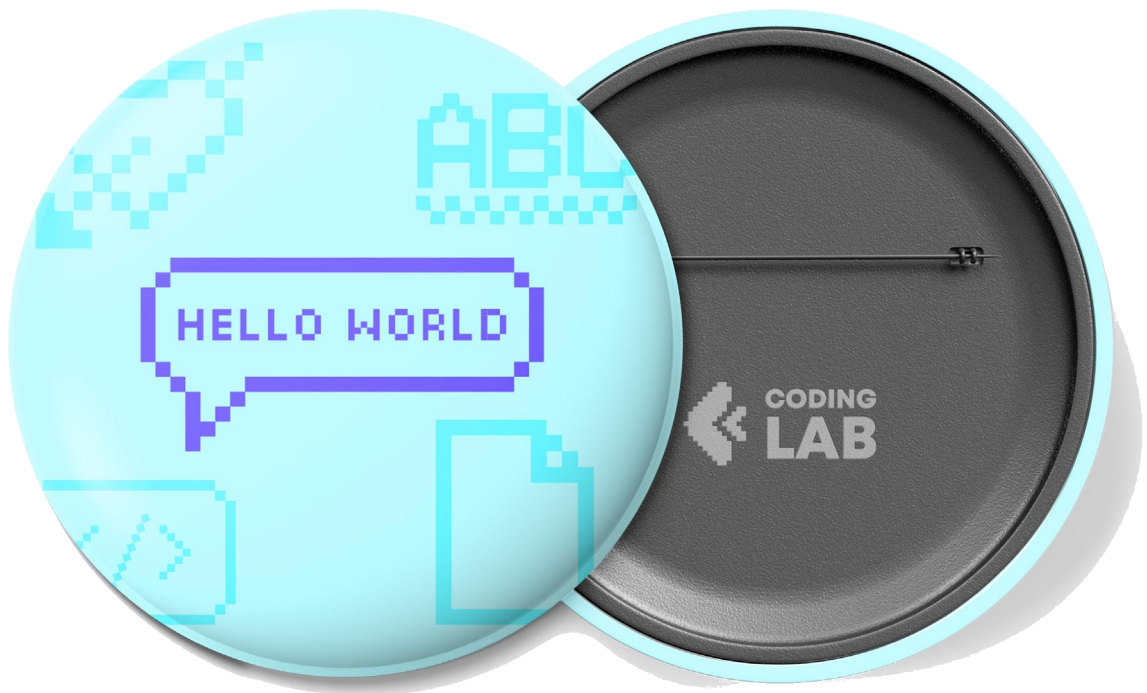


Gadget

Spillette

Infine, completato il corso, saranno regalate delle spillette caratterizzate da grafiche e colori rimandanti ai corsi. Queste indirettamente contribuiscono alla familiarizzazione con il logo e la brand identity, e rimangono a gli studenti come merito per aver completato il corso.









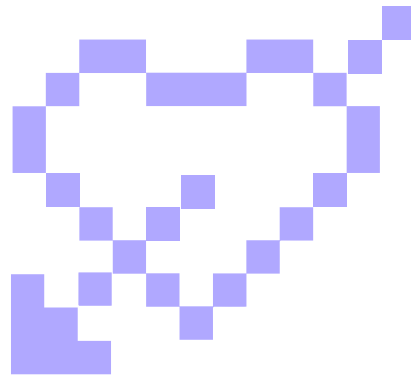
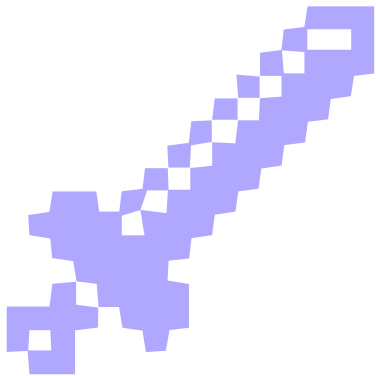
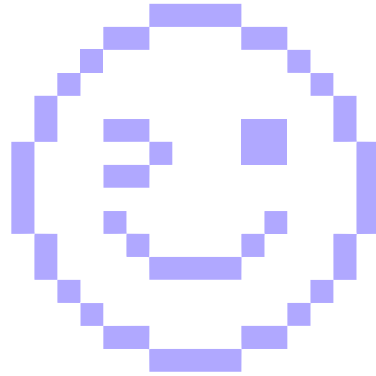
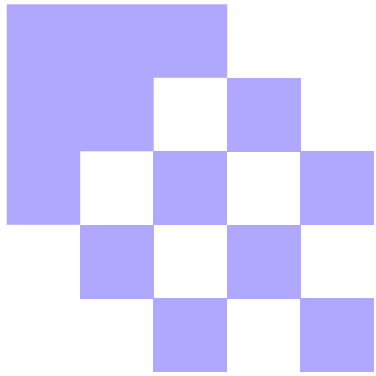
5.0

Conclusioni

In conclusione, la realizzazione del progetto è stato un lungo percorso di studio che ci ha permesso di comprendere meglio quali elementi siano fondamentali per una comunicazione intuitiva e diretta. La scelta dell'immagine coordinata rispetta le linee guida imposte di Chiarezza e Semplicità in quanto questa risulta semplice e guida l'attenzione dell'utente verso gli aspetti fondamentali che il progetto vuole comunicare. Grazie all'utilizzo di elementi visivi che rimandano ai tre ambiti e all'uso dei font scelti, il progetto rispetta le linee guida precedentemente menzionate. Infatti, questo non si presenta come confusionario, ma al contrario il suo aspetto è semplice ed impattante a livello visivo, attirando così l'attenzione delle persone che lo vedono.

Il criterio di intuitività è, invece, rispettato nell'impiego di una comunicazione, che proprio grazie al suo stile e alle icone pixelate rimanda al mondo del creative coding.

Infine, è possibile affermare che il linguaggio adottato rispetta la necessità del progetto di suscitare un sentimento di sicurezza, motivando l'utente ad intraprendere tale esperienza e generando curiosità e non timore nei confronti della programmazione.



6.0

Ringraziamenti

UN RINGRAZIAMENTO SENTITO VA A

La mia famiglia

Gabriele

Aurora

Luca Cattaneo

Davide Cudia

Nala

7.0

Sitografia

Sitografia

Indirizzi Web

- 1 <https://medium.com/uxtales/quello-che-designer-e-developer-non-dicono-197a66f967d9>
- 2 <https://www.xn--mtaweb-3ua.it/web-designer-o-web-developer-differenze>
- 3 <https://careerfoundry.com/en/blog/ui-design/>
- 4 <https://www.neting.it/blog/differenza-ui-ux-design.html>
- 5 <https://careerfoundry.com/en/blog/ux-design/the-difference-between-ux-and-ui-design-a-laymans-guide/>
- 6 <https://careerfoundry.com/en/blog/ui-design/>
- 7 <https://medium.theuxblog.com/a-short-history-of-computer-user-interface-design-29a916e5c2f5>
- 8 https://www.freedesktop.org/wiki/User_interface/
- 9 <https://youtu.be/U10y4X5Ni8Y>
- 10 <https://promosweb22.com/user-experience-i-6-principi-fondamentali-di-donald-norman/>
- 11 <https://www.techopedia.com/definition/4687/windows-icons-menus-and-pointing-device-wimp>
- 12 <https://alistapart.com/article/responsive-web-design/>
- 13 https://www.researchgate.net/profile/Niccolo-Donati/publication/350813865_La_filosofia_dell'usabilita/links/60743ed9a6fdcc5f779d0bdc/La-filosofia-dellusabilita.pdf

- 14 <https://www.nngroup.com/articles/ten-usability-heuristics/>
- 15 https://it.wikibooks.org/wiki/Dietro_il_coding
- 16 <https://dl.acm.org/doi/fullHtml/10.1145/1118178.1118215>
- 17 www.scratch.mit.edu
- 18 it.pearson.com/docenti/primaria/classe-dinamica/che-cosa-pensiero-computazionale.html
- 19 <https://www.techopedia.com/definition/23889/web-development>
- 20 <https://www.studiosamo.it/seo/>
- 21 <https://www.techopedia.com/definition/3799/front-end-system>
- 22 <https://www.techopedia.com/definition/1405/back-end-system>
- 23 <https://www.disko-agency.it/riflessioni/technical/creative-coding-una-tendenza/>
- 24 <https://www.yatta.xyz/generative-art-e-ai-creativa/>
- 25 <http://philipgalanter.com/about>
- 26 http://philipgalanter.com/downloads/ga2003_what_is_genart.pdf
- 27 <https://www.michael-hansmeyer.com/press>
- 28 <https://aiartists.org/generative-art-design>
- 29 https://it.wikipedia.org/wiki/Linguaggio_di_programmazione
- 30 <https://www.redhat.com/it/topics/middleware/what-is-ide>

- 31 https://it.wikipedia.org/wiki/Linguaggio_assembly
- 32 <https://www.techyon.it/articoli/tool-back-end-developer.html#idx0>
- 33 <https://dbgameacademy.it/cose-il-game-design/>
- 34 <https://plarium.com/it/blog/console-generations/>
- 35 <https://itigic.com/it/all-generations-of-consoles-history-evolution-and-models/>
- 36 <https://patents.google.com/patent/US2455992A/en>
- 37 <https://plarium.com/it/blog/primo-videogioco/>
- 38 http://www.giuseppecostanza.it/research_videogames_storia.html
- 39 <https://www.aranzulla.it/programmi-per-creare-giochi-25717.html>
- 40 <https://www.freecodecamp.org/news/what-is-game-development/>
- 41 <https://www.rollingstone.it/cultura/gaming/il-curioso-caso-di-roblox/452252/>
- 42 <https://www.economyup.it/innovazione/roblox-cose-e-come-funziona-la-piattaforma-che-da-accesso-al-metaverso/>
- 43 <https://www.ninjamarketing.it/2021/06/14/roblox-come-funziona-perche-vale-cosi-tanto/>
- 44 <https://careerfoundry.com/en/blog/ux-design/the-difference-between-ux-and-ui-design-a-laymans-guide/>
- 45 <https://www.francescoazzaro.it/differenza-tra-web-designer-e-sviluppatore/>
- 46 <https://www.techyon.it/articoli/front-end-developer-cosa-fa-quali-strumenti.html>
- 47 <https://www.techyon.it/articoli/web-developer-chi-e-cosa-fa.html>
- 48 <https://www.dencreative.com/im-a-designer-how-has-learning-to-code-benefited-my-career>

- 49 <https://www.classup.it/blog/sviluppatore-di-videogiochi-come-si-crea-un-videogame-wseg7>
- 50 <https://4news.it/che-differenze-ci-sono-tra-un-game-designer-e-un-game-developer/>
- 51 <https://www.wgu.edu/blog/video-game-industry-jobs-designer-developer1910.html#close>

APPROFONDIMENTI VIDEO

- Video 1** <https://youtu.be/6rxWc-TNIJI>
- Video 2** <https://youtu.be/fdXI9yznzz8>
- Video 3** https://youtu.be/VMIP_2srCSk
- Video 4** https://youtu.be/n_PUOC-M3JA
- Video 5** <https://youtu.be/JW7oAbLVNJE>
- Video 6** <https://youtu.be/C6WJiUNZf7U>

CASI STUDIO

- 1 <https://gumroad.com/>
- 2 <https://open.spotify.com/>
- 3 <https://www.pinterest.it/>
- 4 <https://www.figma.com/>
- 5 <https://www.apple.com/>
- 6 <https://www.michael-hansmeyer.com/platonic-solids>
- 7 <https://inconvergent.net/>
- 8 <http://markjstock.com/#/dendrite/>
- 9 https://tez.it/index_2011.html
- 10 <https://triennale.org/en/magazine/unknown-play-pleasure-poems-ramallo>

- 11 <https://www.roblox.com/>
- 12 <https://www.unrealengine.com/en-US>
- 13 <https://unity.com/>
- 14 <https://www.minecraft.net/it-it>
- 15 <https://www.epicgames.com/fortnite/it/home>
- 16 <https://boolean.careers/>
- 17 <https://aulab.it/>
- 18 <https://www.udemy.com/>
- 19 <https://www.codecademy.com/>
- 20 <https://www.coursera.org/>

Sviluppo dell'identità visiva e della strategia di comunicazione per un corso di front end, creative coding e game development dedicato a designer.

**Laurea Triennale in Design
e Comunicazione Visiva,
Politecnico di Torino**

**Candidato
Giulio Borgese**

**Relatore
Luca Cattaneo**