



**Politecnico
di Torino**

Politecni 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

Progetto di identità visiva e strategia
di comunicazione per un corso di coding
rivolto a designer e creativi

RELATORE

Luca Cattaneo

CANDIDATI

Gabriele Giovan Maria Basile

Giulio Borgese

Indice

00	Premessa	12
01	Scenario	14
	Coding e Design	16
	User interface design	18
	Quando nasce l'UI?	20
	Good UI design	24
	Coding	26
	Web development	28
	Creative coding	30
	Linguaggi di Sviluppo	32
	Game development	34
	Cenni storici	35
	Sviluppo opens source	38
	Roblox	39
	Figure professionali	40
	UI designer	41
	Web developer	44
	UI designer vs. Web dev	46
	Game developer	48
02	Casi Studio	50
	UI design	54
	Creative coding	60
	Game dev	66

Code academy	72
---------------------	-----------

03	Personas	78
	Antonio Sanna	82
	Federica Tosi	84
	Giulio Gualtieri	86
	Yvonne Maltese	88
	Pietro Fassa	90
	Sara De Luca	92
	Giacomo Principino	94
	Marco Bianchi	96
	Giulia Conti	98
	Alessio Caridi	100

04	Progetto	102
	Concept	104
	Linee guida	104
	Logo	106
	Palette	110
	Typeface	112
	Visual	114
	Strategia	132
	Server Discord	134
	Categorie e canali	136
	Ruoli del server	142
	Google Form	144
	Instagram	146

Lezioni offline/online 148

Canale Youtube 150

LinkedIn 152

05 Conclusioni 154

06 Sitografia 158

07 Ringraziamenti 160

OO

PREMESSA

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.

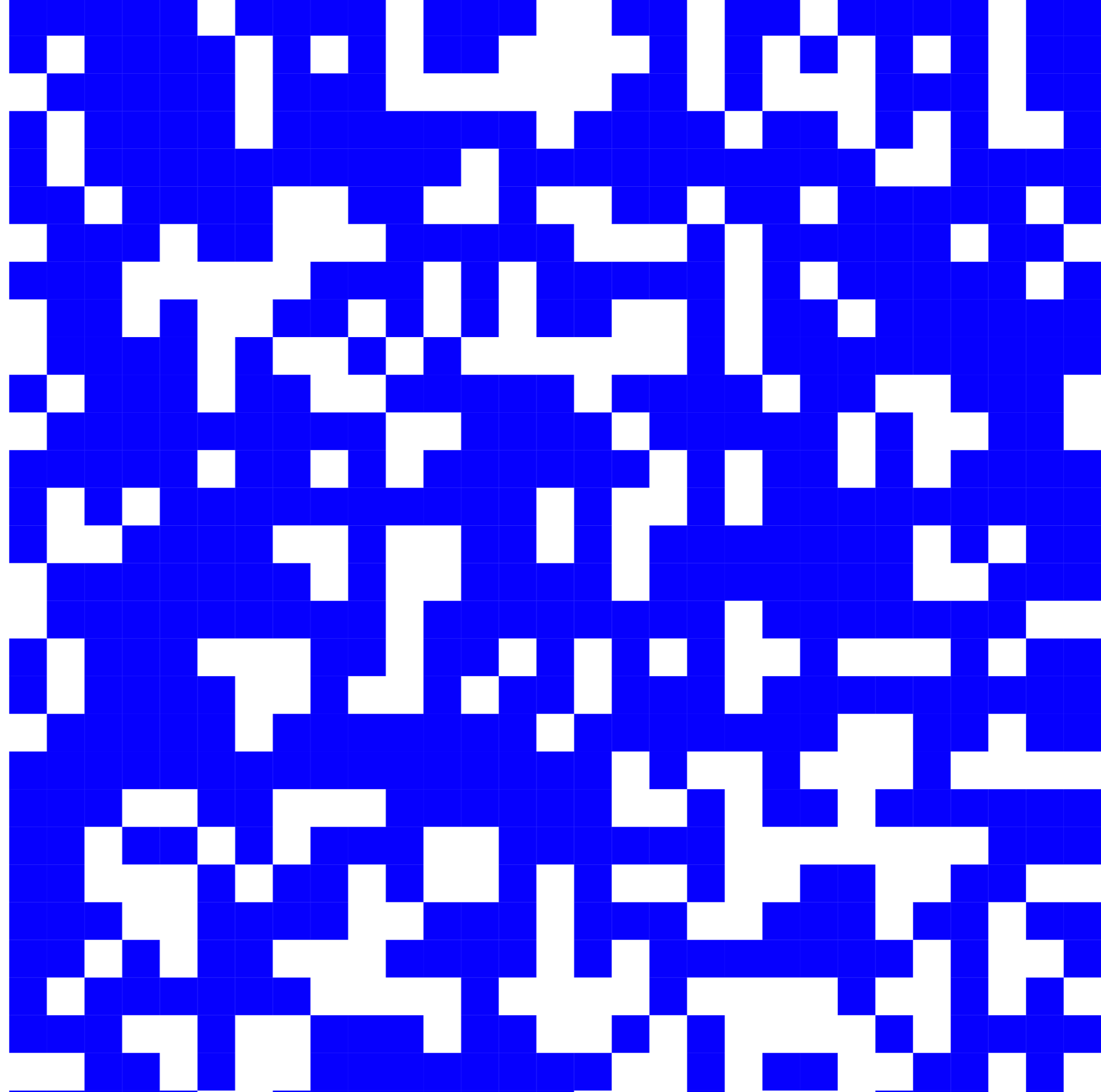
Ancor di più, le aziende, risultano essere attratte da nuove figure professionali ibride che sanno mettere in relazione le competenze progettuali di design con quelle tecniche di sviluppo.

In questo mondo in continua evoluzione dunque, l'importanza dell'apprendimento continuo e della formazione non deve essere sottovalutata.

Questa tesi esplora in profondità il progetto "Coding Lab", un'iniziativa ambiziosa progettata per esercitare in maniera sinergica il design thinking e il pensiero computazionale.

Coding Lab è un laboratorio di formazione per designer e creativi su tre aree chiave: front-end development, creative coding e game development con Roblox.

Attraverso l'analisi delle linee guida, la definizione del concept, della strategia di comunicazione e della metodologia pedagogica, questa tesi esplorerà come Coding Lab intenda creare un ecosistema di apprendimento e networking, dove gli studenti possono veramente "alzare il livello" delle proprie competenze nel mondo digitale e creativo.



01

SCENARIO

Coding e Design

Punti di contatto

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 integrate⁴, 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 indicazione⁵.

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 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-dicono197a66f967d9>
2. <https://www.xn--mtaweb-3ua.it/web-designer-o-web-developer-differenze>



DON'T JUST LEARN TO CODE, LEARN TO CREATE

Justin Richards
TEDxYouth



User interface design

Funzionalità a misura di user

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 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.

UI NEL BUSINESS

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/>



PSYCHOLOGY BEHIND UI/UX DESIGN

Harrish Murugesan
TEDxUTA



Quando nasce l' UI?

Cenni storici⁶

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.

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 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.



▲ FIG. 1
©Apple

IL PARADIGMA WIMP

Nel 1980 **Merzouga Wilberts** conia l'acronimo WIMP¹¹ 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.

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 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, 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/>

1956
IBM SAGE
 COMMAND LINE INTERFACE
 Computer per la difesa aerea militare



FIG. 2 ▶
 © MITRE Corp

1965
PROGRAMMA 101
 COMMAND LINE INTERFACE
 Primo "personal computer"



FIG. 4 ▶
 © Museo NST, Milano

1984
MACINTOSH
 WIMP GRAPHIC UI
 Funzione finder e nuova icona del cestino.

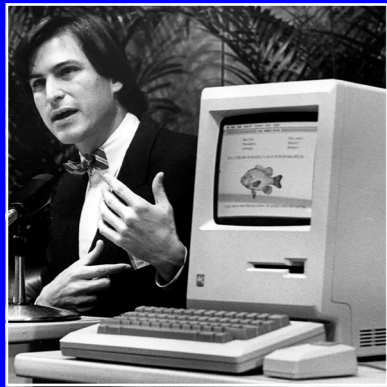


FIG. 6 ▶
 © Terry Schmitt

1987
MACINTOSH II
 WIMP GRAPHIC UI
 Primo Apple a colori

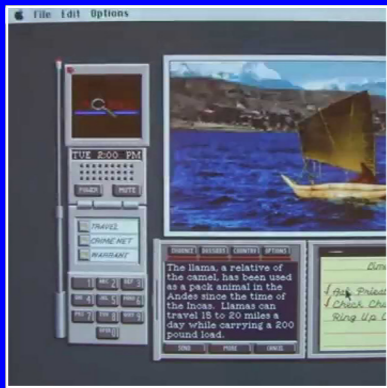


FIG. 8 ▶
 © Apple

1962
PDP-1
 COMMAND LINE INTERFACE
 Esegui Spacewar, uno dei primi videogiochi



FIG. 3
 © Joi Ito

1973
XEROX ALTO
 WIMP GRAPHIC UI
 Nuovo standard di GUI



FIG. 5
 © Maksym Kozlenko

1985
WINDOWS 1.01
 WIMP GRAPHIC UI
 Primo display 8-bit e barra delle applicazioni

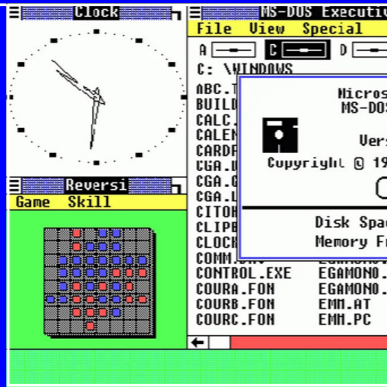


FIG. 7
 © Microsoft

1995
WINDOWS 95
 WIMP GRAPHIC UI
 Aggiunto il tasto start ed internet explorer

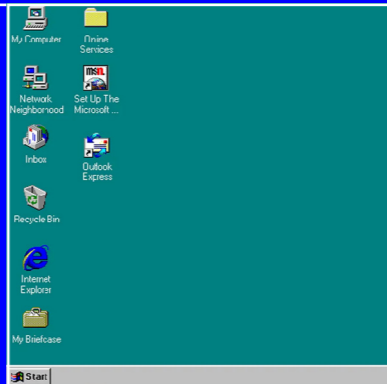


FIG. 9
 © Microsoft

1998
WINDOWS 98
 WIMP GRAPHIC UI
 Primo pacchetto di Office

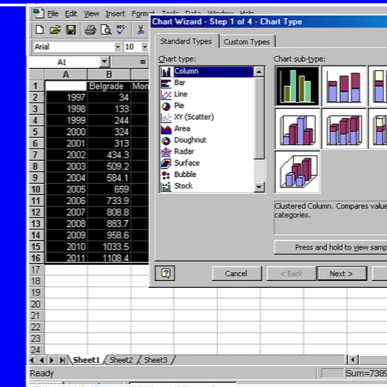


FIG. 10
 © Microsoft

2000
MAC OS X
 WIMP GRAPHIC UI
 Nuova GUI con ombre sui bordi ed effetti animati interattivi

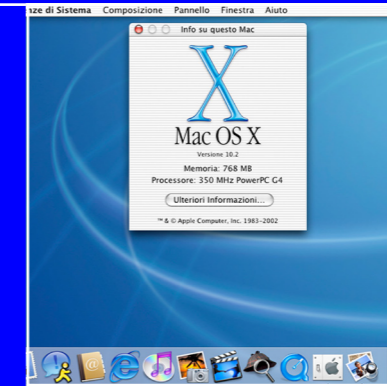


FIG. 11
 © Apple

2001
WINDOWS XP
 WIMP GRAPHIC UI
 GUI aggiornata ai livelli di MAC OS X

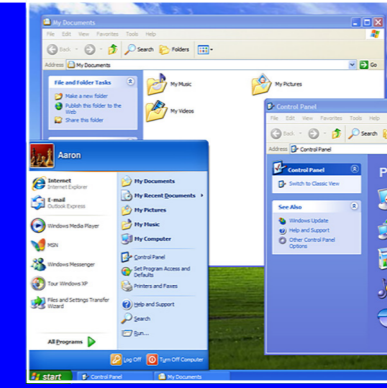


FIG. 12
 © Microsoft

2006
NINTENDO WII
 WIRELESS DEVICE E GUI
 Nuova interazione di wireless control



FIG. 13
 © Nintendo

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 (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.

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 di spiegazioni utili alla corretta fruizione (es.: FAQ).

13. https://www.researchgate.net/profile/Niccolo-Donati/publication/350813865_

[La filosofia dell'usabilità/links/60743ed9a6fdcc5f779d0bdc/La-filosofia-dellusabilita.pdf](https://www.researchgate.net/publication/350813865_)

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



Jakob Nielsen's 10 Heuristics for User Interface Design

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

NIELSEN'S 10 HEURISTICS OF USABILITY

Matt Borchert



Coding

La lingua dei computer

La parola *coding* indica letteralmente l'attività dello scrivere codice, ed ha un significato quasi assimilabile al programmare.

"Quasi, perchè 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, 2016⁴⁵).

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 2006⁴⁶).

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.: *scratch*⁴⁷) che attraverso blocchi, colori, forme ed un'interazione tipicamente drag-and-drop, permettono di rendere concreti i concetti del pensiero⁴⁸.

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

Cosa c'è dietro l'interfaccia?

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.

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. 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.

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.

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

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.

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>

Creative coding

Un nuovo modo di generare arte

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*.

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.

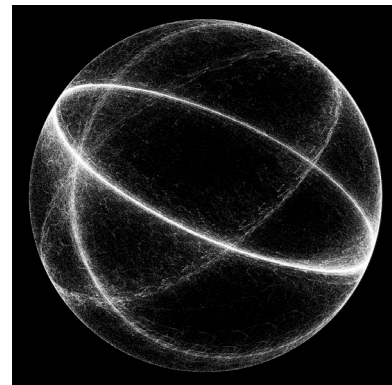
ARTE DIGITALE GENERATIVA

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,



◀ FIG. 14
N.A.O.N.
(©Hayk Zakoyan)

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.

Un altro artista, Michael Hansmeyer²⁷, dice così:

*"Il processo di progettazione raggiunge un equilibrio tra l'atteso e l'inatteso, 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. Un 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/ri flessioni/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>



CREATIVE CODING = UNEXPLORED TERRITORIES

Tim Rodenbröker
TEDxUniPaderborn



Linguaggi di sviluppo

Parlare con i computer

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.

Questa grammatica è costituita dai linguaggi di programmazione che, così come un linguaggio umano, possiedono un lessico, una sintassi e una semantica ben definita.

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 USARLI?

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.

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.

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. L'**HTML** (*HyperText Markup Language*) 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.). Il **CSS** (*Cascading Style Sheet*) serve ad indicare come debbano essere visualizzati i contenuti html, applicando delle regole ai tag (colore, dimensione, allineamento, contorni, ecc.).

Il linguaggio **JavaScript** (abbr. *js*), 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 .NET³².

Il **PHP** (da *Personal Home Page*) è il più conosciuto ed utilizzato durante la creazione di pagine web dinamiche.

Java è un linguaggio *object-oriented*, studiato per contenere strumenti e librerie per il networking ed essere indipendente dalla piattaforma hardware di esecuzione.

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).

I FRAMEWORK

Per alcuni linguaggi esistono i *framework*, 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 (*artificial intelligence*) di *machine learning* e *deep learning*. 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

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://dbgameacademy.it/cose-il-game-design/>

Cenni storici³⁴⁻³⁵

Dal tubo catodico al VR

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

L'invenzione dei videogiochi si attribuisce a **William Higinbotham**³⁸, fisico che contribuì alla creazione della prima bomba nucleare, il quale nel 1958 all'interno dei suoi laboratori inventò Tennis for Two.

Ma è nel 1961 che l'informatico statunitense **Steve Russe** 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*.

SECONDA GENERAZIONE - SISTEMI 8-BIT

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 Odyssey* dell'ingegnere **Ralph Baer**.

Una console collegata a un televisore che, tramite manopole rudimentali, 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

La terza generazione fa capo alla produzione giapponese.



◀ FIG. 15
Space Invaders
(©Taito)

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*.

Questi furono anni d'oro per la **Nintendo**, che riuscì a costruire un impero con capolavori come *Zelda*, *Mario Bros* e *Metroid*.

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

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**.

▼ FIG. 16 - Pac-Man (©Namco)



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*.

SESTA GENERAZIONE - 128 BIT

Nel 1998 venne rilasciata la console di casa **SEGA**, *Dreamcast* e 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.

SETTIMA GENERAZIONE

Il 2004 è l'anno di rivalse 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

▼ FIG. 17 - Tekken 1 (©Namco)



con *Kinect* e *PS Move*, ma non riscuotono lo stesso successo.

Nel 2005 Microsoft mette in commercio l'*Xbox 360*, competitiva ed economica, e l'anno seguente Sony la *PlayStation 3*, economicamente meno accessibile.

OTTAVA GENERAZIONE

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

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 per le realtà aumentate.

▼ FIG. 18 - Playstation 2 (©Sony)



la VR, le AI e i giochi blockchain come Roblox, anche se ancora in fase embrionale.

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/US245592A/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 NO-CODE

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 no-code*.

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.

A tal proposito sono stati sviluppati motori di gioco no-code 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.

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/>

▼ FIG. 19 - SUPERHOT (©Superhot Team)



Roblox

Spazio alla creazione

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.

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

▼ FIG. 20 - NIKELAND (©Roblox)



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. 21 - ROBLOX (©Roblox)



Figure professionali

Skill diverse, obiettivo comune

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.

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

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.

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.

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.

Quest'ultimo punto riveste un ruolo fondamentale, 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.francescoazzero.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. 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.

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.

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.

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

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.

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 migliore⁴⁶.

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.

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.

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.

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

Game developer

Un artista computazionale

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.

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.

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.

IL 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>

02

CASI STUDIO

Casi studio

Introduzione

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 ognuno 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 no-code.

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

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.

52. www.gumroad.com



◀ FIG. 22
GUMROAD logo
(©Gumroad)

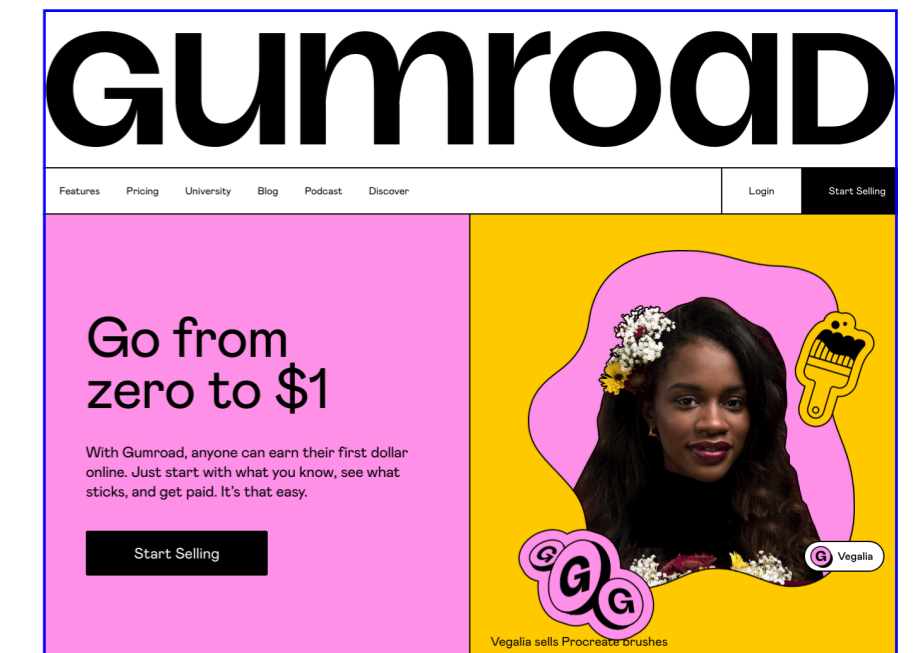


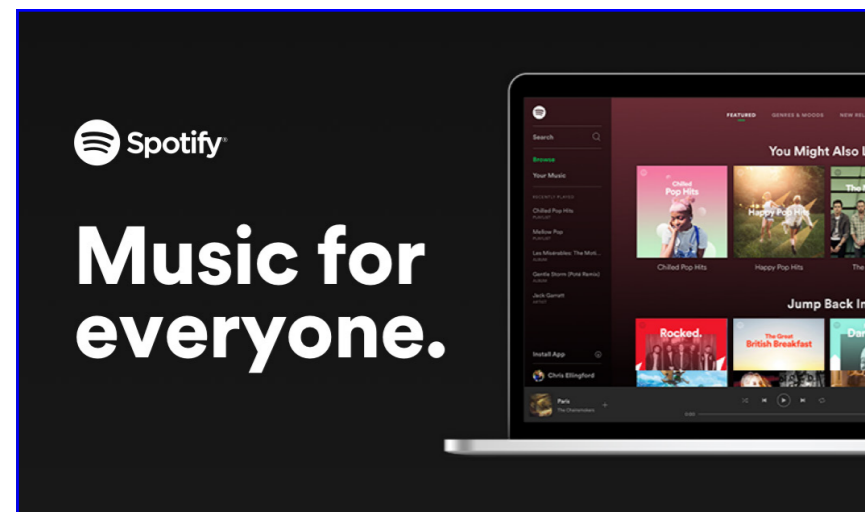
FIG. 23 ▶
GUMROAD logo
(©Gumroad)

Spotify

2006 / PIATTAFORMA DI STREAMING

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.

53. open.spotify.com



◀ FIG. 24
Spotify Web
(©Spotify)

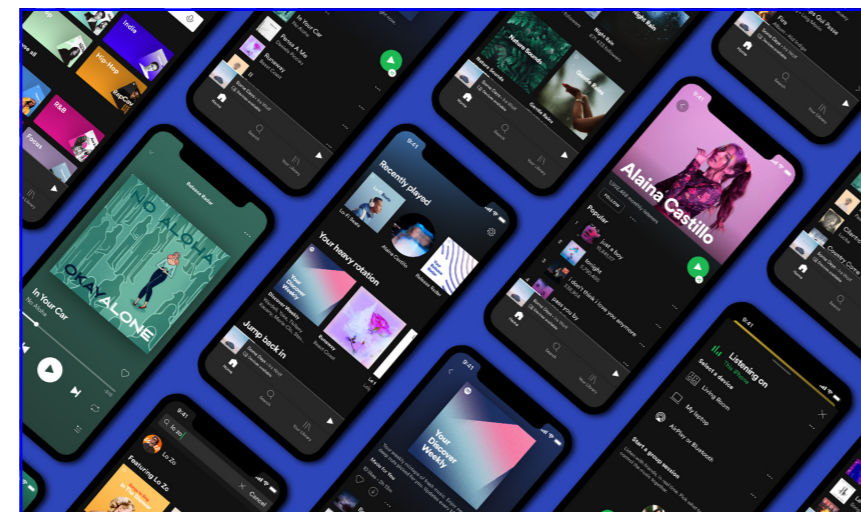


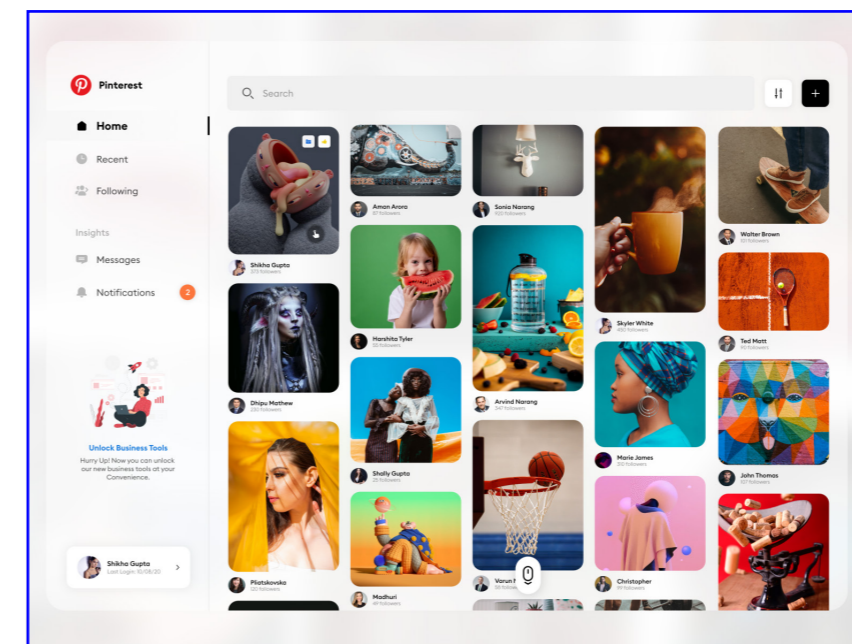
FIG. 25 ▶
Spotify App
(©Spotify)

Pinterest

2010 / SOCIAL

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.

54. www.pinterest.it



◀ FIG. 26
Pinterest Web
(©Pinterest)

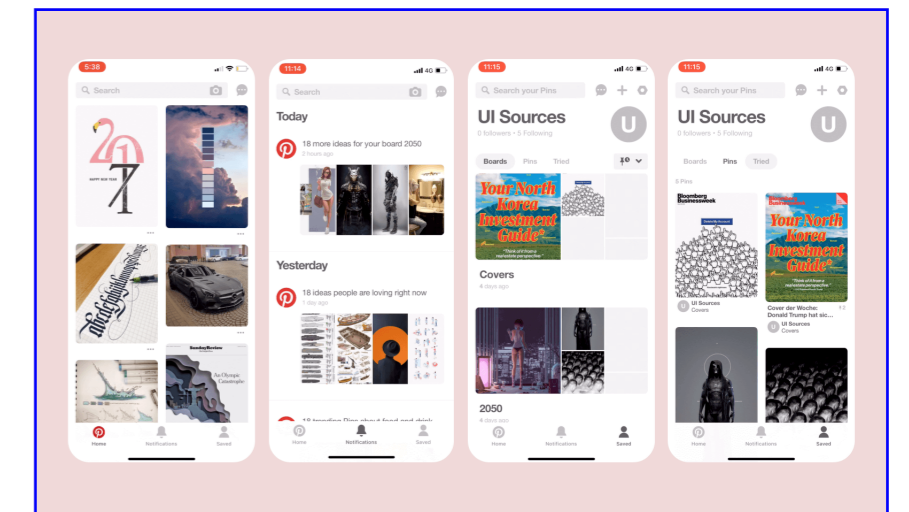


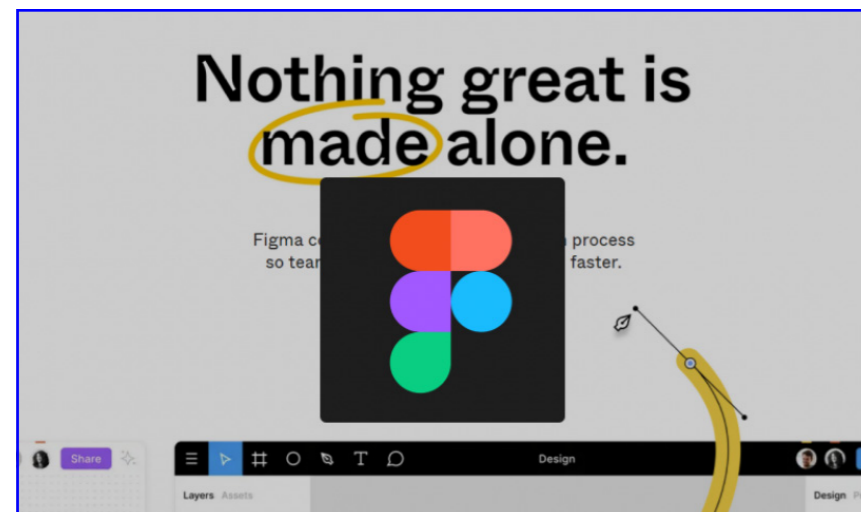
FIG. 27 ▶
Pinterest App
(©Pinterest)

Figma

2016 | SOFTWARE

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.

55. www.figma.com



◀ FIG. 28
FIGMA
(©Figma)

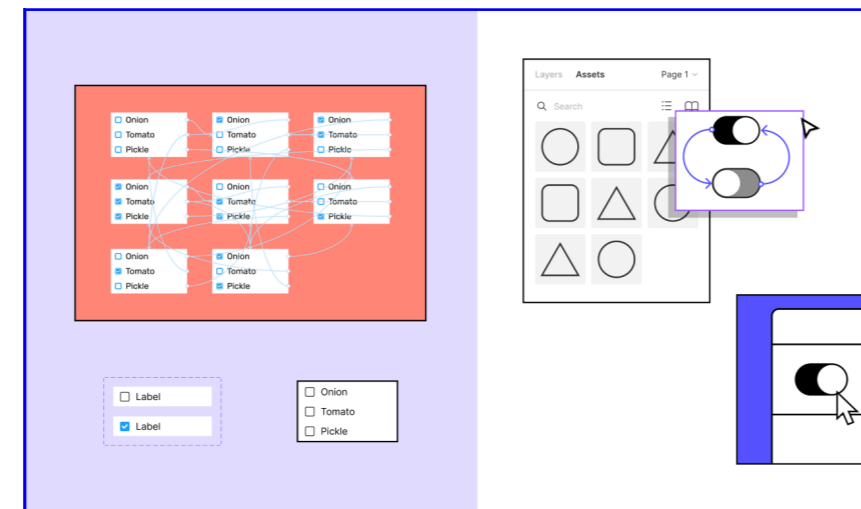


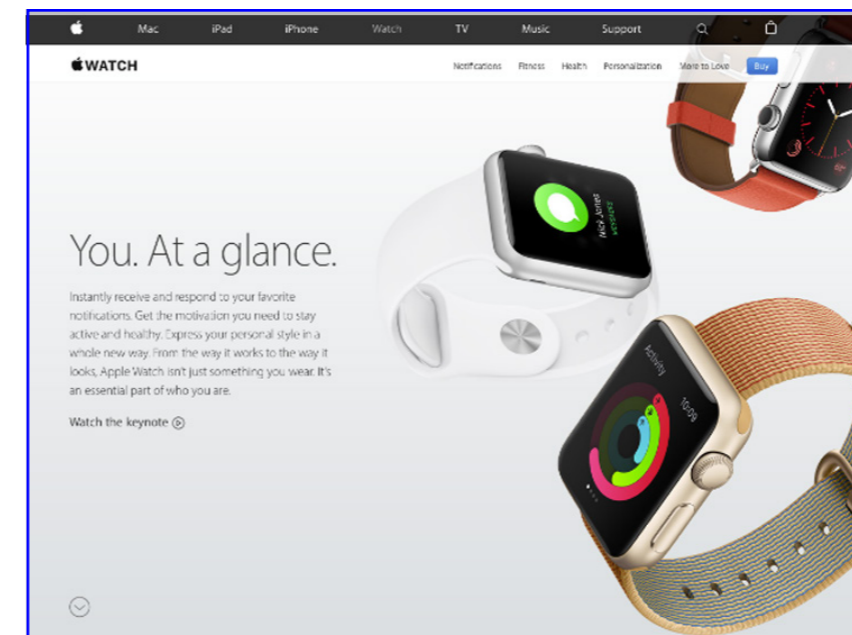
FIG. 29 ▶
FIGMA
(©Figma)

Apple

2011 | E-COMMERCE

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.

56. www.apple.com



◀ FIG. 30
Apple Website
(©Apple)

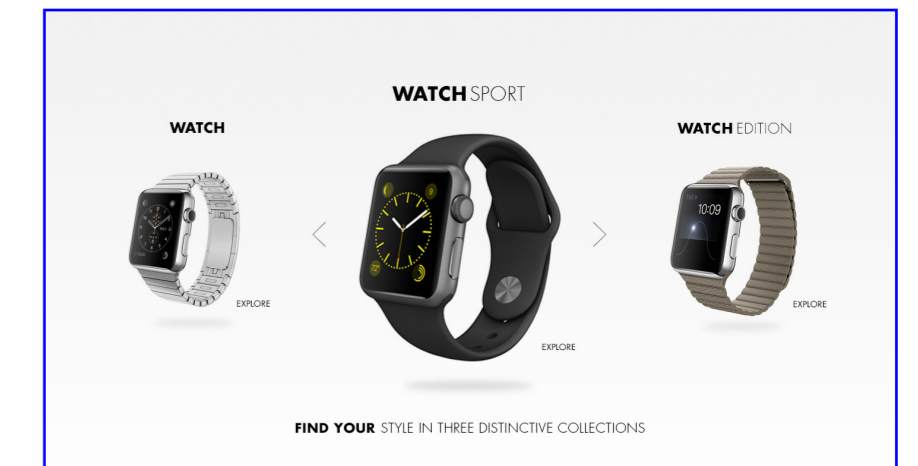


FIG. 31 ▶
Apple Web
(©Apple)

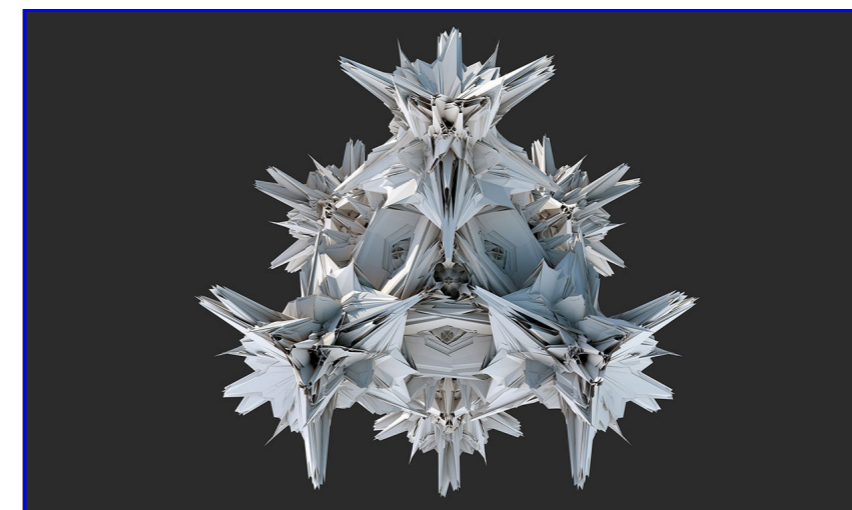
CREATIVE CODING

Platonic Solid

2008 | MICHEAL HANSMEYER

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.

57. www.michael-hansmeyer.com/platonic-solids



◀ FIG. 32
PLATONIC SOLID
(©M.Hansmeyer)

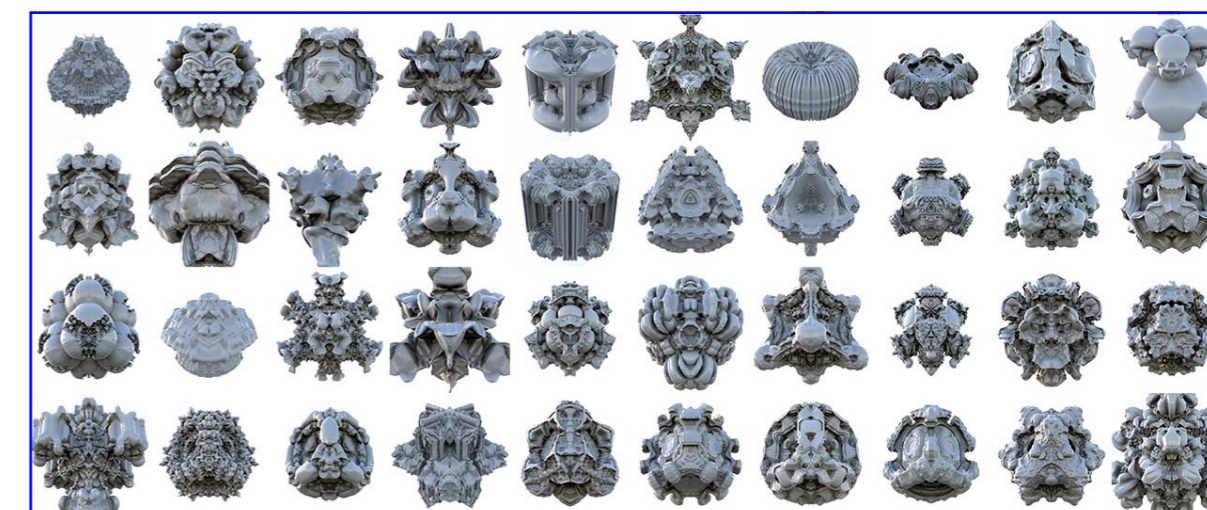


FIG. 33 ▶
PLATONIC SOLID
(©M.Hansmeyer)

Tech Plotter Drawings

2016 | ANDERS HOFF

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

58. www.inconvergent.net

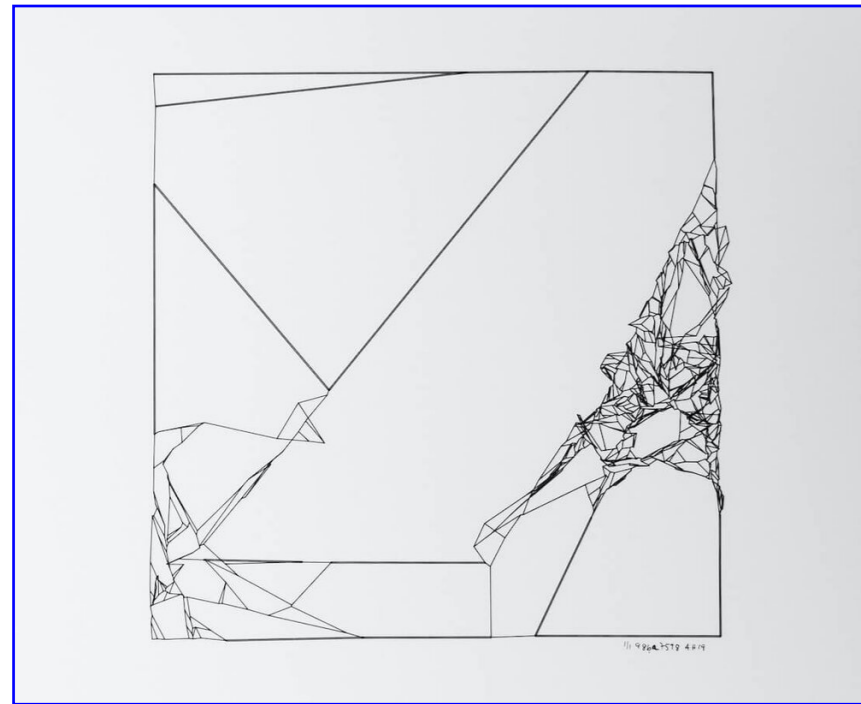


FIG. 35 ►
MECHANICAL PLOTTER DRAWINGS
(©A. Hoff)

◀ FIG. 34
MECHANICAL PLOTTER DRAWINGS
(©A. Hoff)

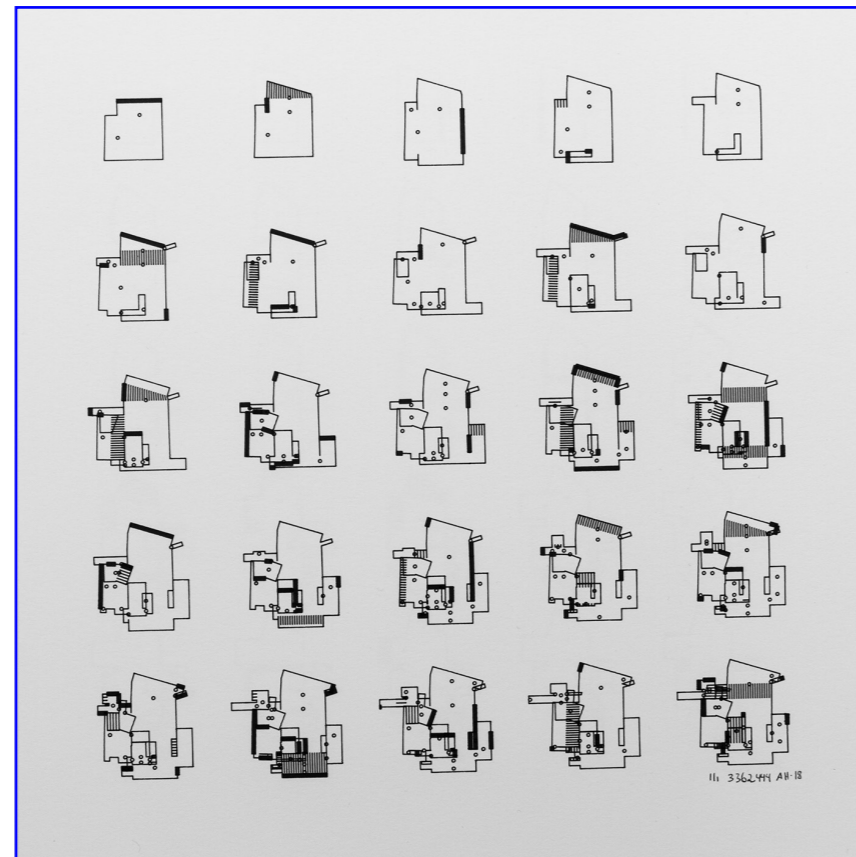


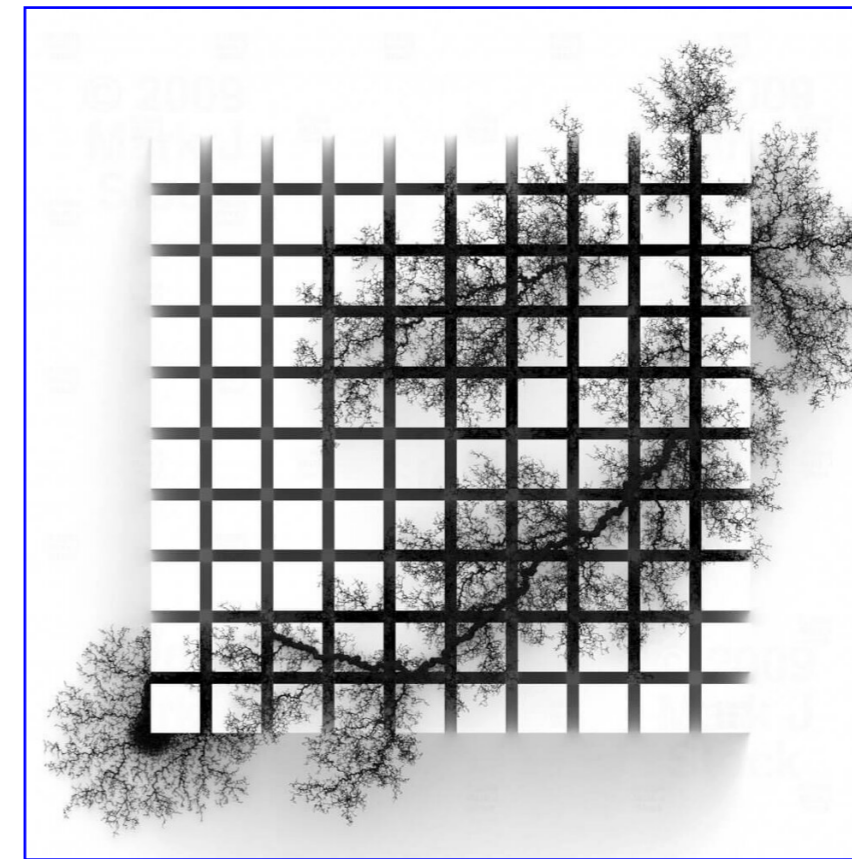
FIG. 37 ►
TWIG
(©M. J. Stock)

Dendrite

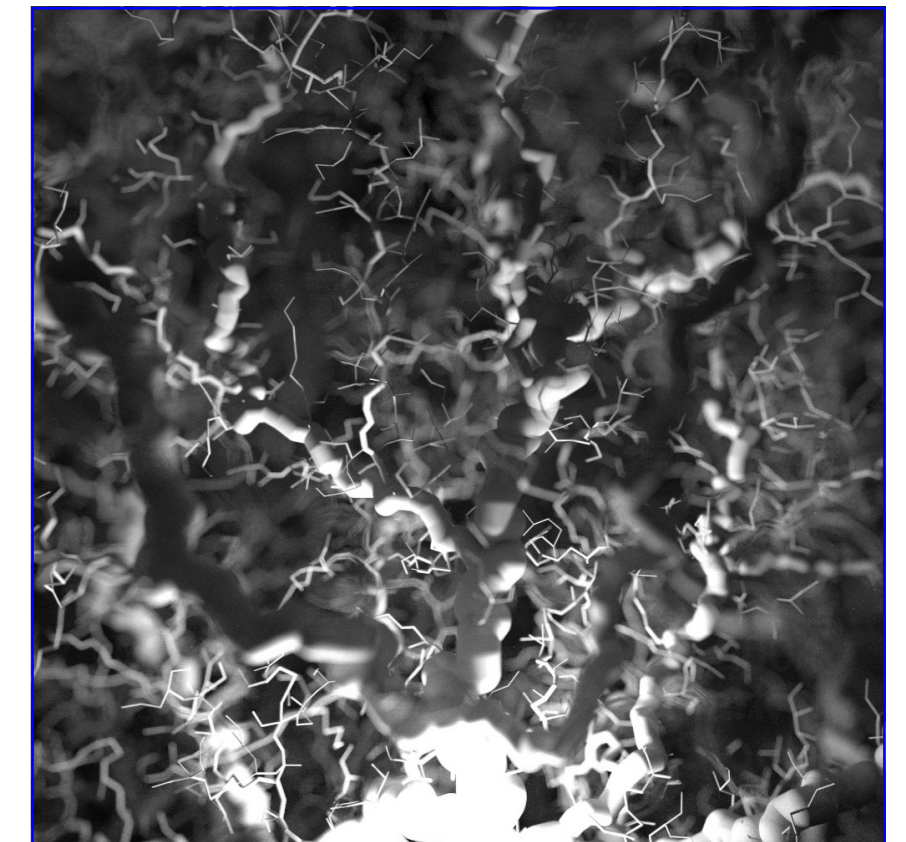
2009 | MARK J. STOCK

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

59. www.markjstock.com/#/dendrite



◀ FIG. 36
SPRALW
(©M. J. Stock)

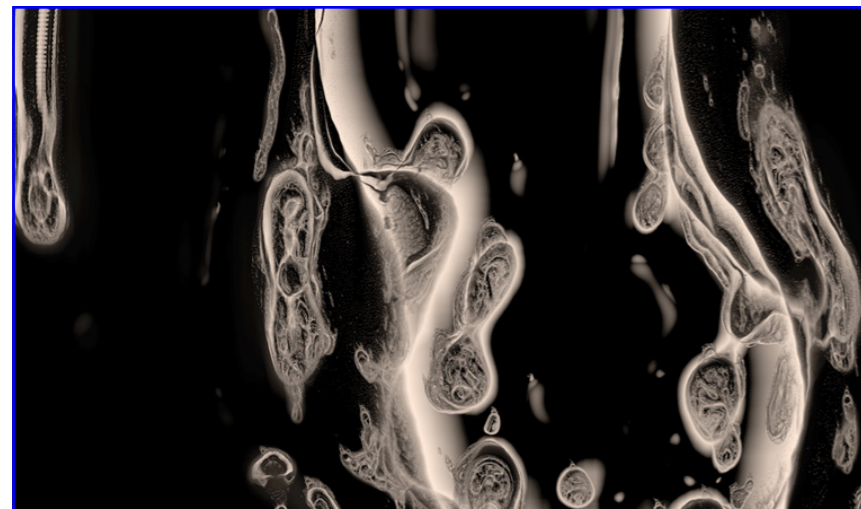


Plasm 4K Shot

2014 | MAURIZIO MARTINUCCI aka TEZ

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.

60. www.tez.it/index_2011.html



◀ FIG. 38
PLASM 4K SHOT
(©Tez)



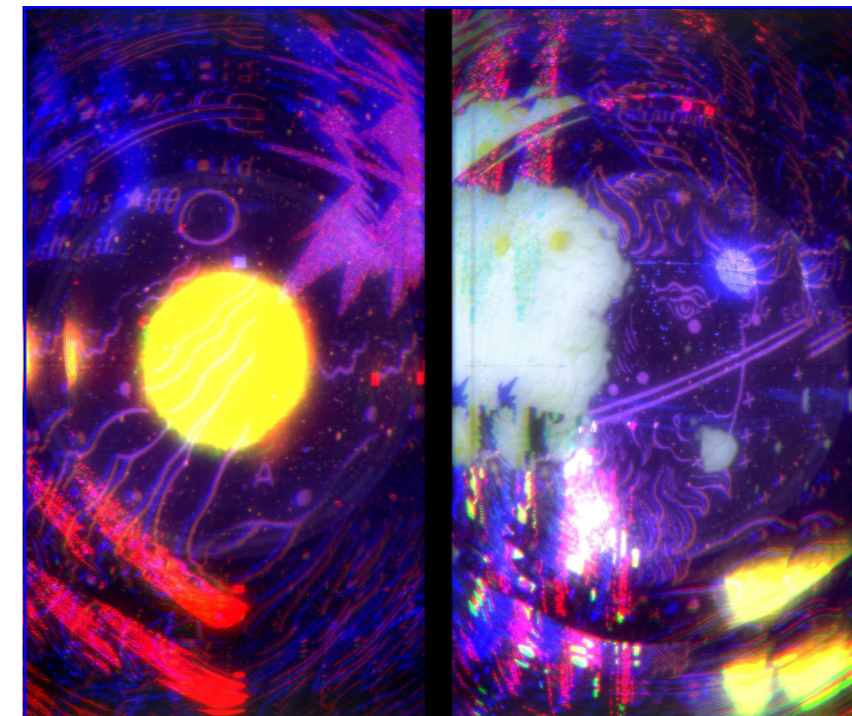
FIG. 39 ▶
DISPLACE
(©Tez)

We Are Poems

2023 | FERN GOLDFRAB-RAMALLO

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

61. www.triennale.org/en/magazine/unknown-play-pleasure-poems-ramallo



◀ FIG. 40
WE ARE POEMS
(©F. Goldfarb-Ramallo)

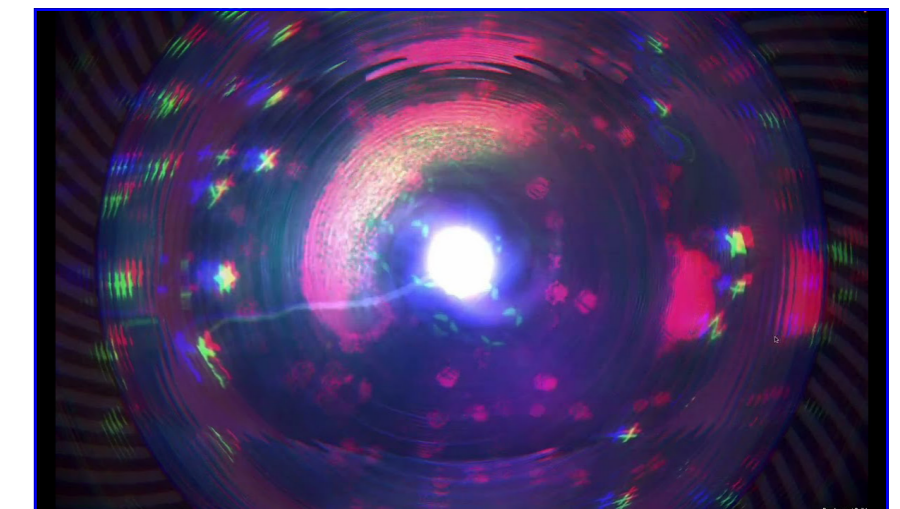


FIG. 41 ▶
WE ARE POEMS
(©F. Goldfarb-Ramallo)

GAME DEV

Roblox

2006 | VIDEOGAME MMO

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.

62. www.roblox.com



◀ FIG. 42
ROBLOX
(©Roblox)



FIG. 43 ▶
ANIME FIGHTER SIMULATOR
(©Roblox)

Unreal Engine

2006 | SOFTWARE

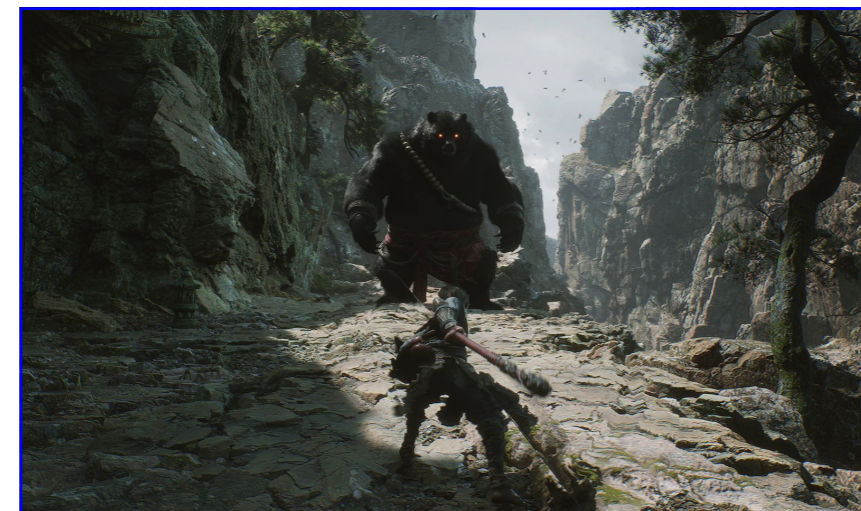
Unreal Engine è un motore grafico sviluppato da Epic Games in continuo sviluppo nel software per adattarsi ad hardware sempre più potenti.

Tramite il linguaggio C++ permette lo sviluppo di videogiochi, grafiche e animazioni 2D/3D in tempo reale.

63. www.unrealengine.com



FIG. 45 ►
BLACK MYTH WUKONG
(©Games Science)



◀ FIG. 44
UNREAL
(©Unreal)

Unity

2005 | MOTORE GRAFICO MULTIPIATTAFORMA

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#.

64. www.unity.com



◀ FIG. 46
UNITY
(©Unity)



FIG. 47 ►
THE FOREST
(©Endnight Games)

Minecraft

2011 | SANDBOX OPEN WORLD VIDEOGAME

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.

65. www.minecraft.net



FIG. 49 ►
MINECRAFT
(©Minecraft)

◀ FIG. 48
MINECRAFT
(©Minecraft)



Fortnite

2017 | BATTLE ROYALE GAME

Motore grafico multiplatforma, 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#.

66. www.fortnite.com



◀ FIG. 50
FORTNITE
(©Fortnite)



FIG. 51 ►
FORTNITE
(©Fortnite)

CODE ACADEMY

Boolean

2021

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.

67. www.boolean.careers



◀ FIG. 52
BOOLEAN LOGO
(©Boolean)

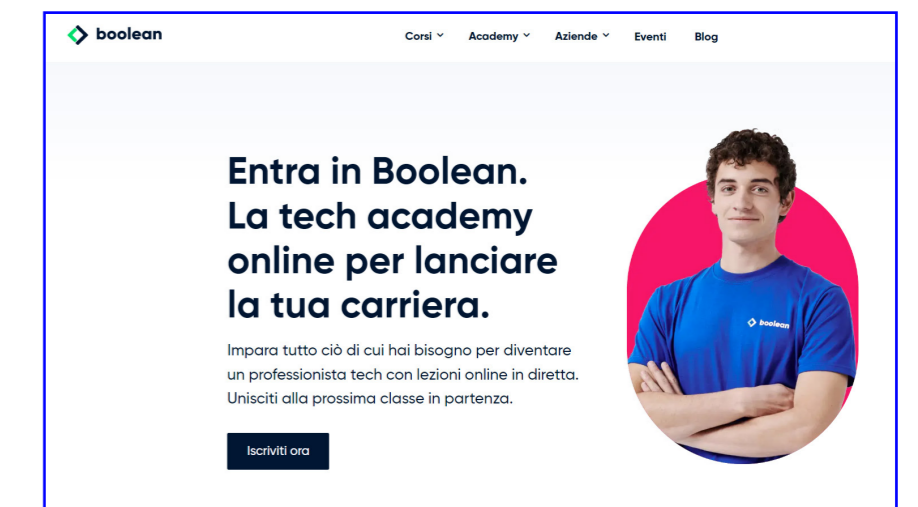


FIG. 53 ▶
BOOLEAN WEBSITE
(©Boolean)

Aulab

2021

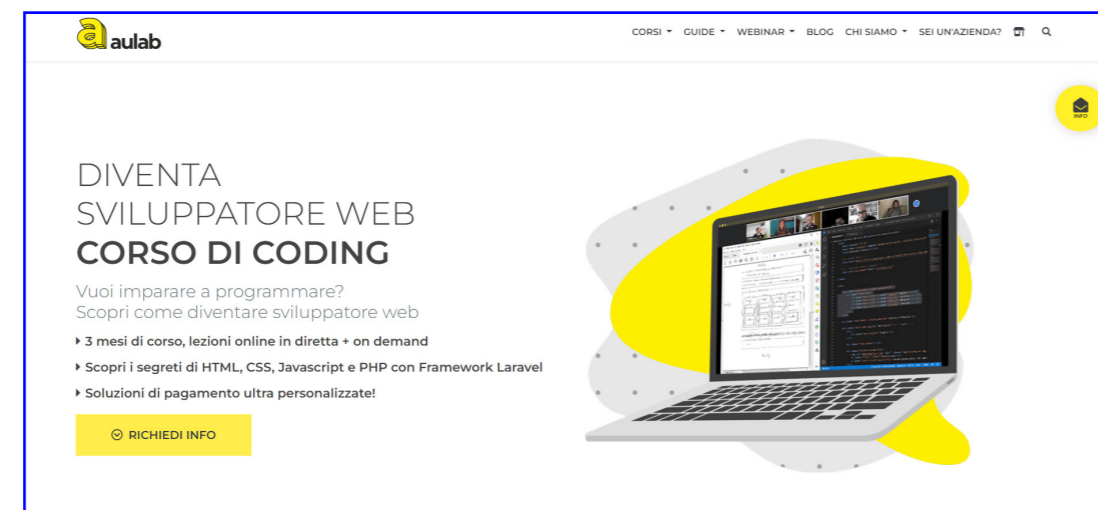
L'accademia online Aulab utilizza una grafica caratterizzata da illustrazioni e elementi inorganici. I colori prevalenti sono il grigio e il giallo. Il tone of voice è motivazionale e ottimista. Nel complesso la strategia utilizzata rende la comunicazione chiara ed efficace.

68. www.aulab.it



◀ FIG. 54
AULAB LOGO
(©Aulab)

FIG. 55 ▶
AULAB WEBSITE
(©Aulab)



Udemy

2021

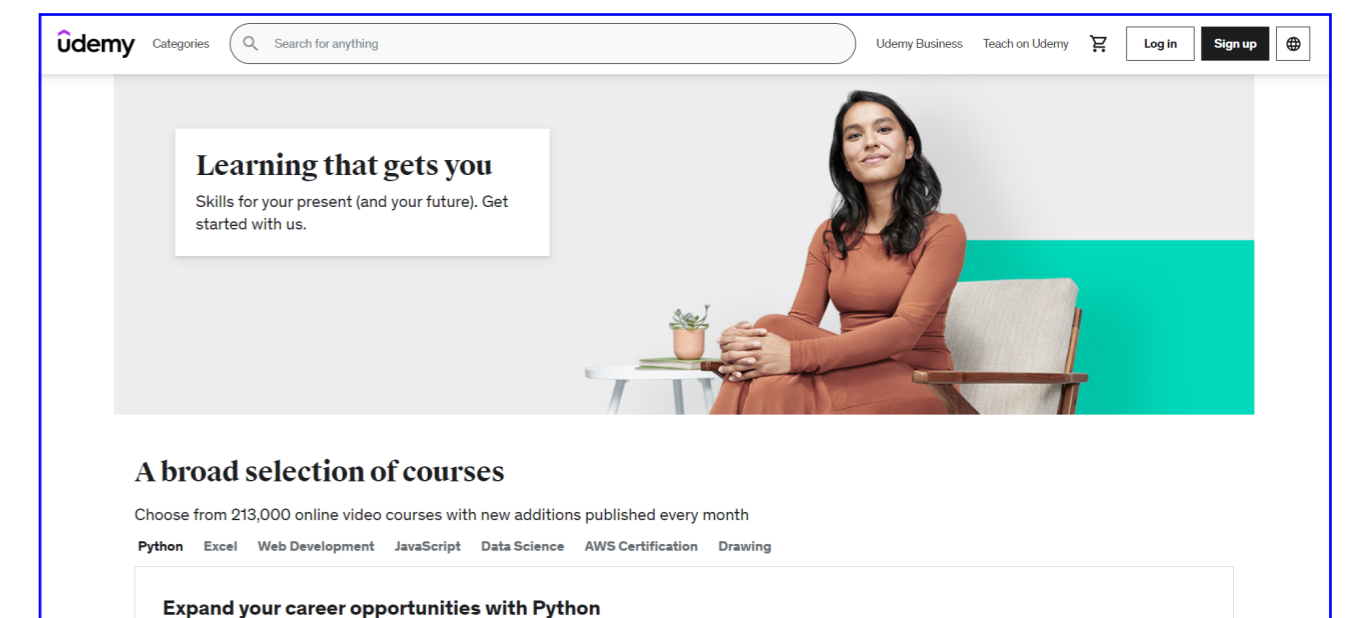
Udemy è un'accademia online che utilizza una grafica caratterizzata da colori piatti affiancati ad un uso prevalente di foto. Il tone of voice è colloquiale e motivazionale. Tuttavia, nel complesso la strategia utilizzata rende la comunicazione confusionale.

69. www.udemy.com



◀ FIG. 56
UDEMY LOGO
(©Udemy)

FIG. 57 ▶
UDEMY WEBSITE
(©Udemy)



Codecademy

2021

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.

70. www.codecademy.com



◀ FIG. 58
CODECADEMY LOGO
(©Codecademy)

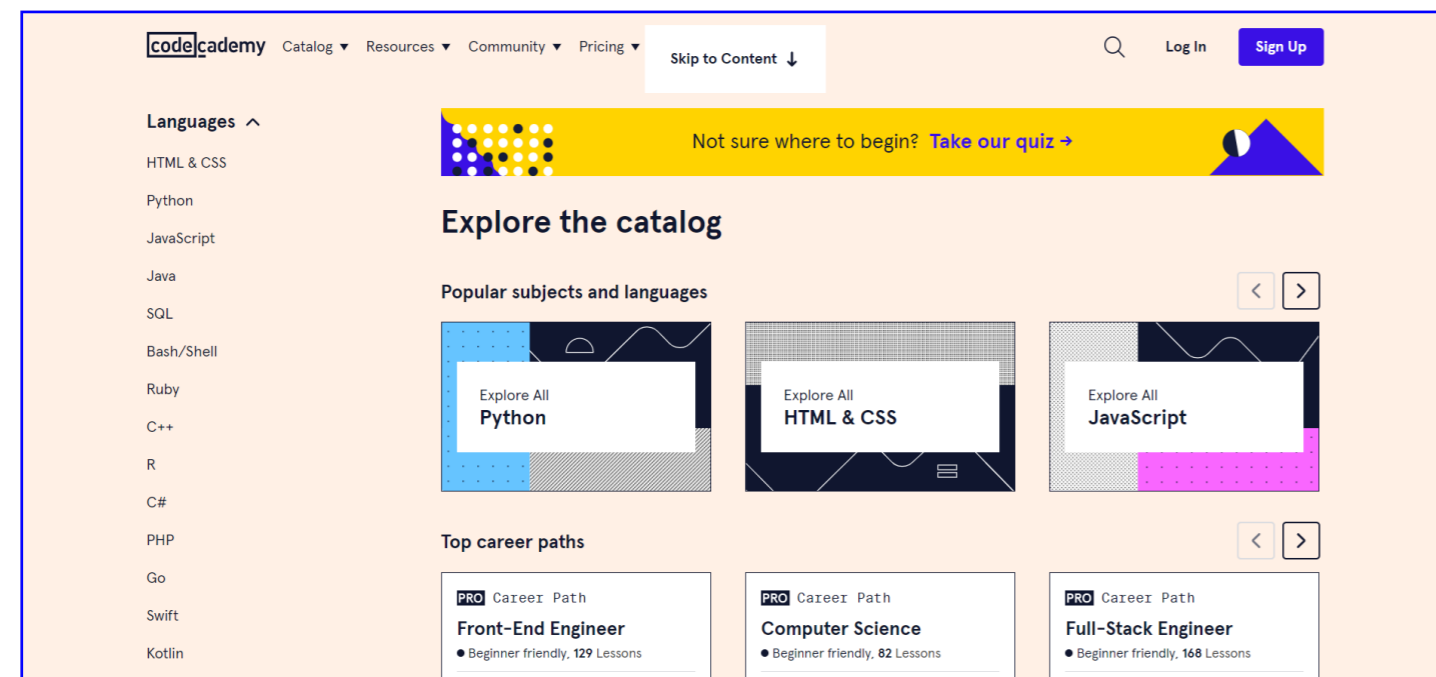


FIG. 59 ▶
CODECADEMY WEBSITE
(©Codecademy)

Coursera

2021

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.

71. www.coursera.org



◀ FIG. 60
COURSERA LOGO
(©Coursera)

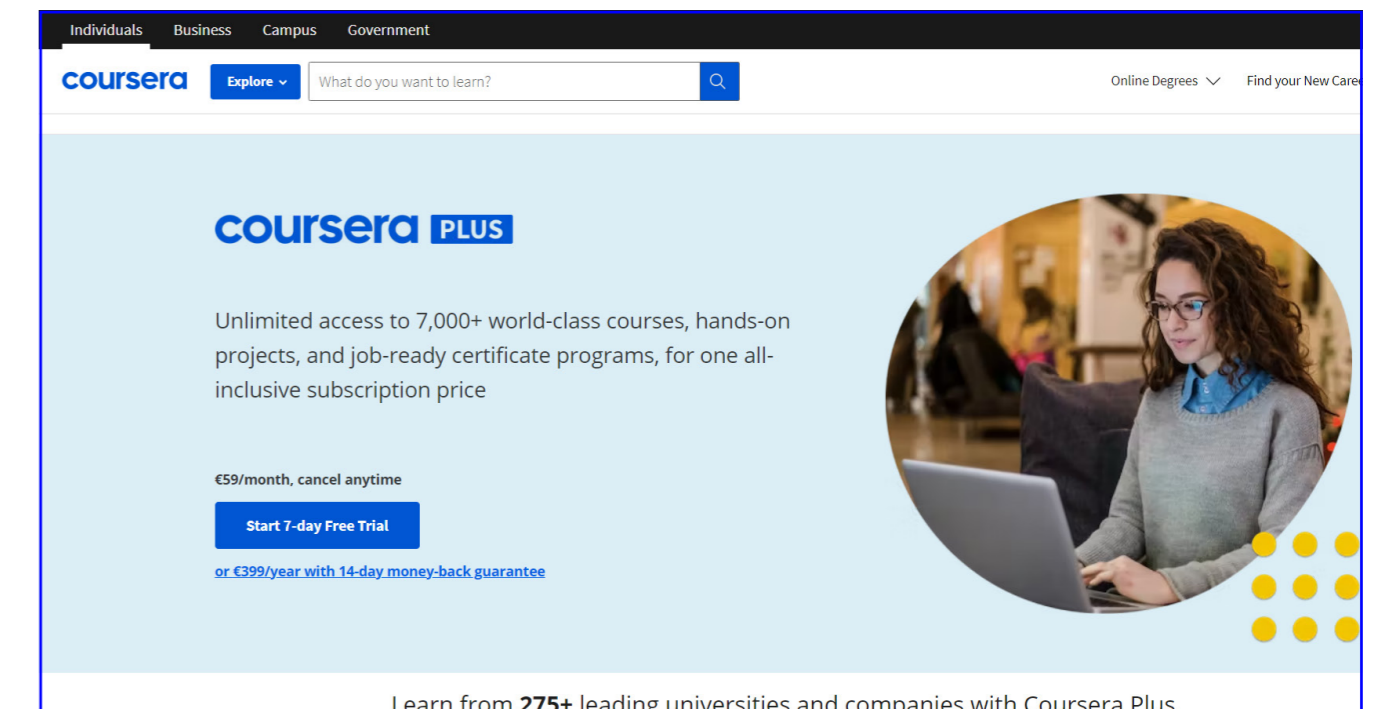


FIG. 61 ▶
COURSERA WEBSITE
(©Coursera)

03

PERSONAS

Personas

2021

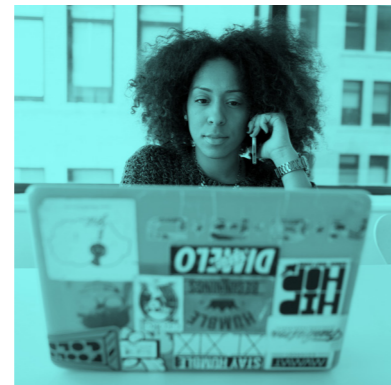
Le personas individuate possono essere suddivise in categorie, a cominciare dalla prima, più istituzionale, del professore di università.

Si passa poi a tre gruppi di personas interessate alle tre aree di sviluppo di Coding Lab, tutti provenienti da ambienti creativi o da una qualche pregressa esperienza nello sviluppo, ognuno con i propri pain points, obiettivi ed esigenze.

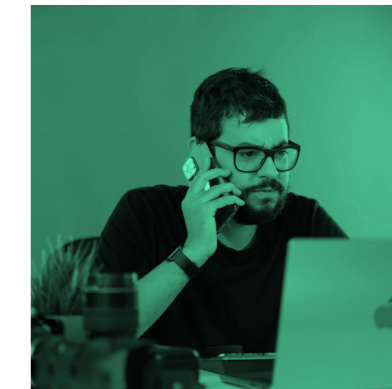
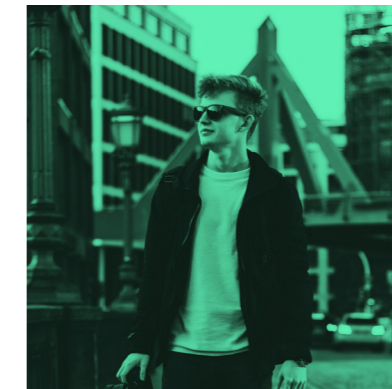
UNIVERSITÀ



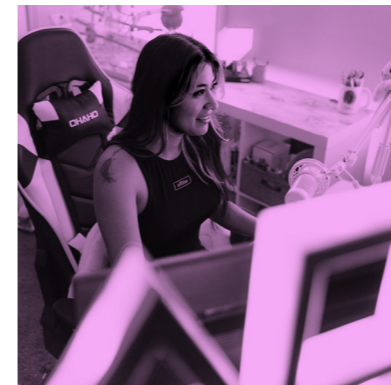
FRONT-END DEVELOPMENT



CREATIVE CODING



GAME DEVELOPMENT





Amedeo Sanna

Professore Universitario
39 anni

“L’università dovrebbe impegnarsi a fornire tutte le competenze che l’attuale mondo del lavoro richiede.”

DESCRIZIONE

Amedeo Sanna è un professore e coordinatore didattico che crede nel valore del proprio lavoro. Cerca di costruire una didattica dinamica e mutevole, che sia coerente con i tempi moderni e che sappia uniformarsi al reale mondo del lavoro. Il professor Sanna è molto curioso e si tiene aggiornato su mostre, conferenze e spesso legge con piacere i saggi tematici più diversi.

HOBBY & INTERESSI

Eventi
Mostre
Saggistica
Innovazione digitale

SOCIAL NETWORK



TOOLS



PAIN POINTS

A volte si scontra con visioni un po' antiquate del sistema universitario, che avvicinano gli studenti verso tematiche e competenze più teoriche che empiriche.

GOALS

Vorrebbe formare i suoi studenti fornendo loro competenze utili nel mondo del lavoro, attraverso una didattica dinamica e coinvolgente.



Federica TOSI

UI Designer
27 anni

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

DESCRIZIONE

Federica dopo una laurea in interaction design desidera approfondire le sue conoscenze di codice per lavorare da remoto e avere più tempo da dedicare alle sue passioni.

HOBBY & INTERESSI

Data Visualization
Fotografia
Viaggiare
Trekking
Natura

SOCIAL NETWORK



TOOLS



PAIN POINTS

Non dispone di abbastanza tempo per viaggiare e non è disposta a rinunciarci. Non trova corsi scalabili alle sue esigenze.

GOALS

Vuole acquisire competenze di front end in modo flessibile e trovare un lavoro da remoto che le consenta di viaggiare.



Giulio Gualtieri

Graphic Designer freelance
26 anni

“Oggi essere un creativo non può prescindere da una presenza sul web. E qui bisogna presentarsi al meglio!”

DESCRIZIONE

Giulio è un graphic designer freelance che desidera ampliare i servizi che offre. Desidera acquisire competenze tecniche di sviluppo web per implementare i suoi progetti di design in modo autonomo.

HOBBY & INTERESSI

Design
Fotografia
Viaggiare
Street art
Vita Notturna

SOCIAL NETWORK



TOOLS



PAIN POINTS

I suoi progetti dipendono sempre da sviluppatori esterni che devono implementarli.

GOALS

Vuole acquisire skill di front end development per implementare i suoi progetti in autonomia, offrire servizi completi ai suoi clienti e realizzare un sito personale per darsi il giusto valore con la concorrenza.



Yvonne Maitese

Web Designer
27 anni

“Vorrei capire perché i dev hanno sempre da ridire sui miei prototipi!”

DESCRIZIONE

Yvonne è una giovane professionista nel campo del web design. Ha una solida base di design e vorrebbe sviluppare le sue competenze nel front-end development per progettare in maniera più consapevole e possibilmente realizzare le sue idee in modo autonomo.

HOBBY & INTERESSI

Design Trends
Musica
Serie TV
Musica
Fotografia

SOCIAL NETWORK



TOOLS



PAIN POINTS

Manca di competenze tecniche per implementare le sue idee di design con il coding, creando spesso incomprensioni con i team di sviluppo.

GOALS

Vuole imparare le basi del front end development per sviluppare siti web interattivi e migliorare le sue abilità di design.



Pietro Fasssa

Back-end Developer
48 anni

“Se la parte più complicata è quella di back-end, perchè non diventare uno sviluppatore full-stack?”

DESCRIZIONE

Pietro è un developer con esperienza nel codice di server e nell'architettura back-end. Desidera acquisire competenze di front-end development per sviluppare applicazioni web autonomamente dalla A alla Z.

HOBBY & INTERESSI

AI
Viaggiare
Trekking
Natura
Yoga

SOCIAL NETWORK



TOOLS



PAIN POINTS

Pietro si è sempre limitato alle competenze di backend, adesso vorrebbe rinnovare il suo curriculum.

GOALS

Ha intenzione di acquisire competenze front-end per dare un input alla sua creatività e migliorare le sue opportunità di lavoro.

Sara De Luca

Studentessa di Design
23 anni

“Il lavoro del designer è correlato all’innovazione, vale a dire che non bisogna precludersi nuovi sentieri da percorrere ed esplorare.”

DESCRIZIONE

Sara è una studentessa di design con una forte passione per l’interazione digitale. Desidera imparare il creative coding per creare opere d’arte e design interattivi.

HOBBY & INTERESSI

Design
Arte
Disegno
Danza contemporanea
Nuoto

SOCIAL NETWORK



TOOLS



PAIN POINTS

Sara non ha conoscenze di programmazione per trasformare le sue idee di design in opere d’arte interattive.

GOALS

Vuole sviluppare le competenze di creative coding per creare design e opere d’arte interattive ed arricchire così il suo portfolio.

Giacomo Principino

Motio Designer freelance
24 anni

“I linguaggi di programmazione come JS sono tool utilissimi per agevolare il proprio workflow.”

DESCRIZIONE

Giacomo è un giovane motion designer neo laureato. È un appassionato del mondo della creatività visiva con alcune esperienze alle spalle. Per imparare si è spesso affidato a tutorial online, venendo così a conoscenza di utili strumenti come i linguaggi di scripting come JS, rimanendo affascinato dalla versatilità e dalle numerose possibilità creative che offre.

HOBBY & INTERESSI

Film d'animazione
Disegno
Videogiochi
Serie TV
Musica

SOCIAL NETWORK



TOOLS



PAIN POINTS

Non conosce molte funzioni di JS e quando ne trova di interessanti non sa come applicarle nel concreto per potersi allenare e ricordarle successivamente.

GOALS

Vuole acquisire dimestichezza con la lettura e la strutturazione di script per i propri progetti, attraverso metodologie che permettano di sperimentare il suo pensiero computazionale.



Marco Bianchi

Grafico Pubblicitario
32 anni

“Esistono nuove metodologie capaci di ampliare le nostre capacità creative attraverso la programmazione.”

DESCRIZIONE

Marco è un grafico pubblicitario con esperienza nel design visivo. Desidera espandere le sue competenze imparando il creative coding per creare esperienze digitali interattive.

HOBBY & INTERESSI

Arte digitale
Innovazione digitale
Musica Elettronica
Calcio
Fotografia

SOCIAL NETWORK



TOOLS



PAIN POINTS

Si sente limitato dalla sua conoscenza di grafica tradizionale, e vorrebbe integrare la programmazione nel suo processo creativo.

GOALS

Vuole imparare il creative coding per creare opere d'arte digitali interattive e ampliare le sue opportunità professionali.



Giulia Conti

UX / UI
28 anni

“Da grandi interazioni derivano grandi responsabilità!”

DESCRIZIONE

Giulia è una UX/UI Designer che ha sempre nutrito un certo interesse per i videogiochi. Desidera imparare a sviluppare giochi con Roblox per creare esperienze interattive coinvolgenti e user-friendly.

HOBBY & INTERESSI

Videogiochi
Innovazione Digitale
Musica
Social Media
Disegno

SOCIAL NETWORK



TOOLS



PAIN POINTS

Vorrebbe aggiungere interazioni ai suoi progetti ma non dispone delle competenze necessarie.

GOALS

Vuole imparare ad usare Roblox per creare esperienze di gioco coinvolgenti e utilizzabili in un'ottica di design.



Alessio Cariddi

Studente Liceale
19 anni

“Chi pensa che i videogiochi siano solo roba da bambini, non sa che impegno e competenze servono per creare queste piccole opere d’arte.”

DESCRIZIONE

Alessio è uno studente liceale neofita della programmazione che ha sempre sognato di creare un videogioco tutto suo. Ha conosciuto Roblox e adesso vuole utilizzarlo per il suo scopo. Si sta affacciando sul panorama universitario cercando un indirizzo di studi che asseconi sia le sue capacità creative sia quelle computazionali.

HOBBY & INTERESSI

Videogiochi
Streaming
Programmazione
Disegno
Editing

SOCIAL NETWORK



TOOLS



PAIN POINTS

Non ha familiarità con la programmazione avanzata, dunque non sa da dove partire per ideare un suo gioco.

GOALS

Vuole imparare a creare giochi su Roblox e condividere le sue creazioni con la community.

04.

PROGETTO

Concept

Level Up

“Level up” vuol dire fornire un ambiente inclusivo e collaborativo in cui designer e creativi possano sviluppare le proprie abilità, creare connessioni significative e aggiungere un nuovo tassello nella loro carriera.

L'aspirazione è quella di essere il catalizzatore che aiuta i corsisti ed i membri della community a raggiungere il massimo del proprio potenziale e fare un “level up” come in un videogioco.

VISION

Il progetto Coding Lab vuole creare un ambiente dinamico e ispirante dedicato alla formazione e allo sviluppo di designer e creativi.

Vorremmo trasformare Coding Lab in un faro di opportunità, dove i partecipanti possono crescere sia personalmente che professionalmente, attraverso l'acquisizione di competenze di punta nel campo del design grafico e dello sviluppo creativo.

Coding Lab vuole essere un punto di riferimento per chi desidera “alzare il livello” delle proprie abilità e connettersi con altri professionisti con la stessa passione.

MISSION

La missione di Coding Lab è quella di promuovere la crescita individuale e collettiva dei nostri corsisti, fornendo loro le risorse necessarie per eccellere nel mondo in continua evoluzione del design grafico e dello sviluppo creativo ad esso collegato.

Linee Guida

COMMUNITY E NETWORKING

Un'atmosfera di collaborazione e condivisione all'interno della comunità di corsisti è un punto chiave.

Le connessioni tra individui che condividono esperienze, valori e interessi comuni sono fondamentali per la crescita professionale e personale.

La community sarà un luogo in cui i partecipanti potranno imparare dagli altri, ispirarsi reciprocamente e costruire relazioni durature.

POTENZIARE

Ai corsisti devono essere fornite le conoscenze e le competenze necessarie per eccellere nel proprio campo. Bisogna aiutare ogni partecipante a migliorare le proprie abilità creative, professionali e lavorative attraverso corsi e workshop garantendo il tutoring.

CREARE VALORE

Va riconosciuta l'importanza di applicare concretamente le competenze acquisite. Per questo bisogna offrire opportunità pratiche per costruire un portfolio di valore.

I partecipanti dovranno non solo imparare, ma anche creare opere e progetti significativi che li distinguano nel mercato del lavoro.

FLESSIBILITÀ

Vanno tenuti in considerazione l'accessibilità e la flessibilità nell'apprendimento. Vanno garantiti contenuti online così da permettere l'accesso a risorse e tutoring quando e dove più conviene.

L'apprendimento deve potersi adattare alle esigenze dei partecipanti.



Logo

SIMBOLO

Il simbolo "<" è universalmente riconosciuto nel mondo della programmazione come l'operatore "minore di" o "inferiore a".

La sua inclusione nel logo suggerisce immediatamente una connessione con il coding ed il development.

PIXEL ART

L'uso dello stile *pixel art* per rappresentare il simbolo ha diverse implicazioni.

In primo luogo, richiama il concetto di **pixel**, che è l'unità di base di un'immagine su uno schermo.

Questo è significativo perché i programmatori lavorano spesso a livello di dettaglio, pixel per pixel, quando creano grafica o interfacce utente.

Inoltre, lo stile *pixel art* è profondamente radicato nella cultura dei videogiochi retro a 8 e 16 bit, che possono essere associati all'origine della programmazione stessa, richiamando un senso di nostalgia e appartenenza alla cultura del coding.

GRIGLIA ORTOGONALE

La griglia di costruzione è ortogonale e trasmette un senso di **ordine**, **precisione** e **organizzazione**, tutti concetti cruciali della programmazione, che si basa su strutture dati e algoritmi logici.

QUADRATO

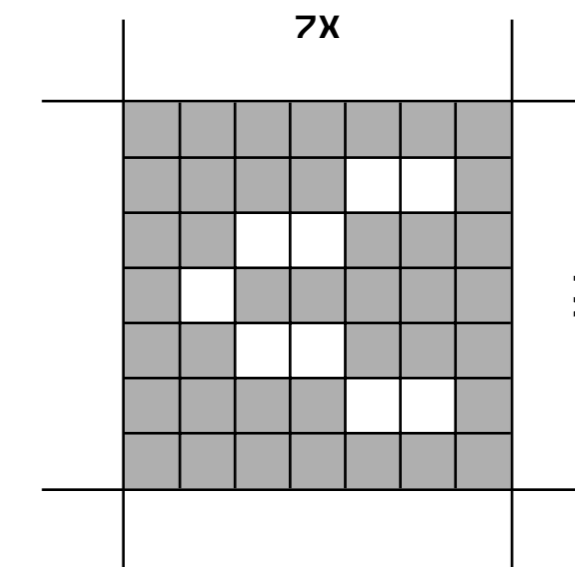
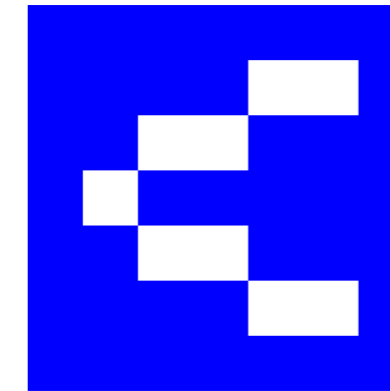
La forma quadrata del logo rappresenta l'**equilibrio** e la **stabilità**, qualità desiderabili in un ambiente di sviluppo. Inoltre suggerisce l'idea di "blocchi di codice", un insieme ben definito di istruzioni.

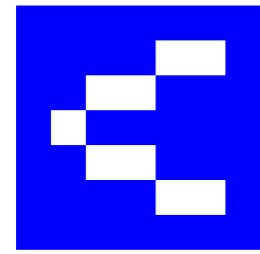
SINTESI VISIVA

Il logo è stato costruito con una profonda comprensione delle suggestioni visive e concettuali legate alla programmazione, creando una **sintesi visiva** unica.

CARATTERE

L'accostamento di un carattere monospaziato al logo è una scelta attentamente ponderata che enfatizza l'essenza della programmazione, la chiarezza e la leggibilità del codice e fornisce un aspetto coerente e adatto al contesto educativo del corso.

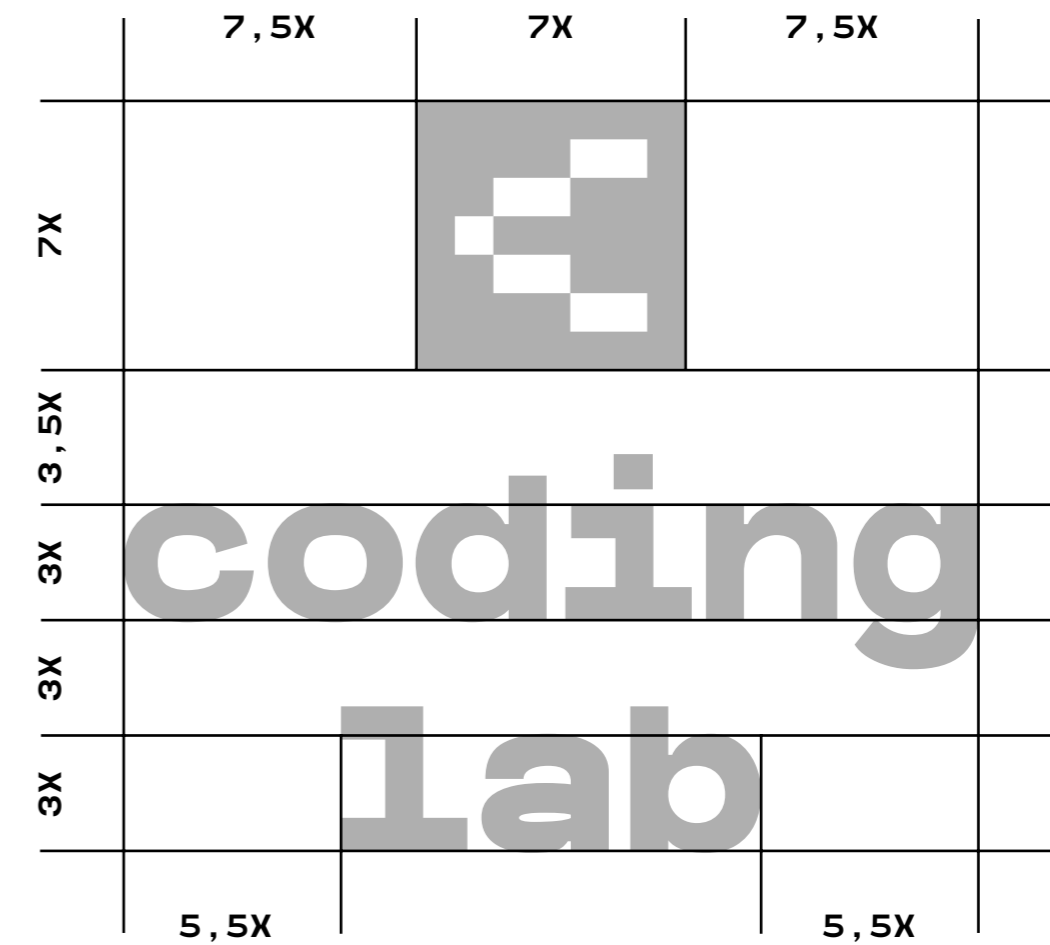




coding Lab

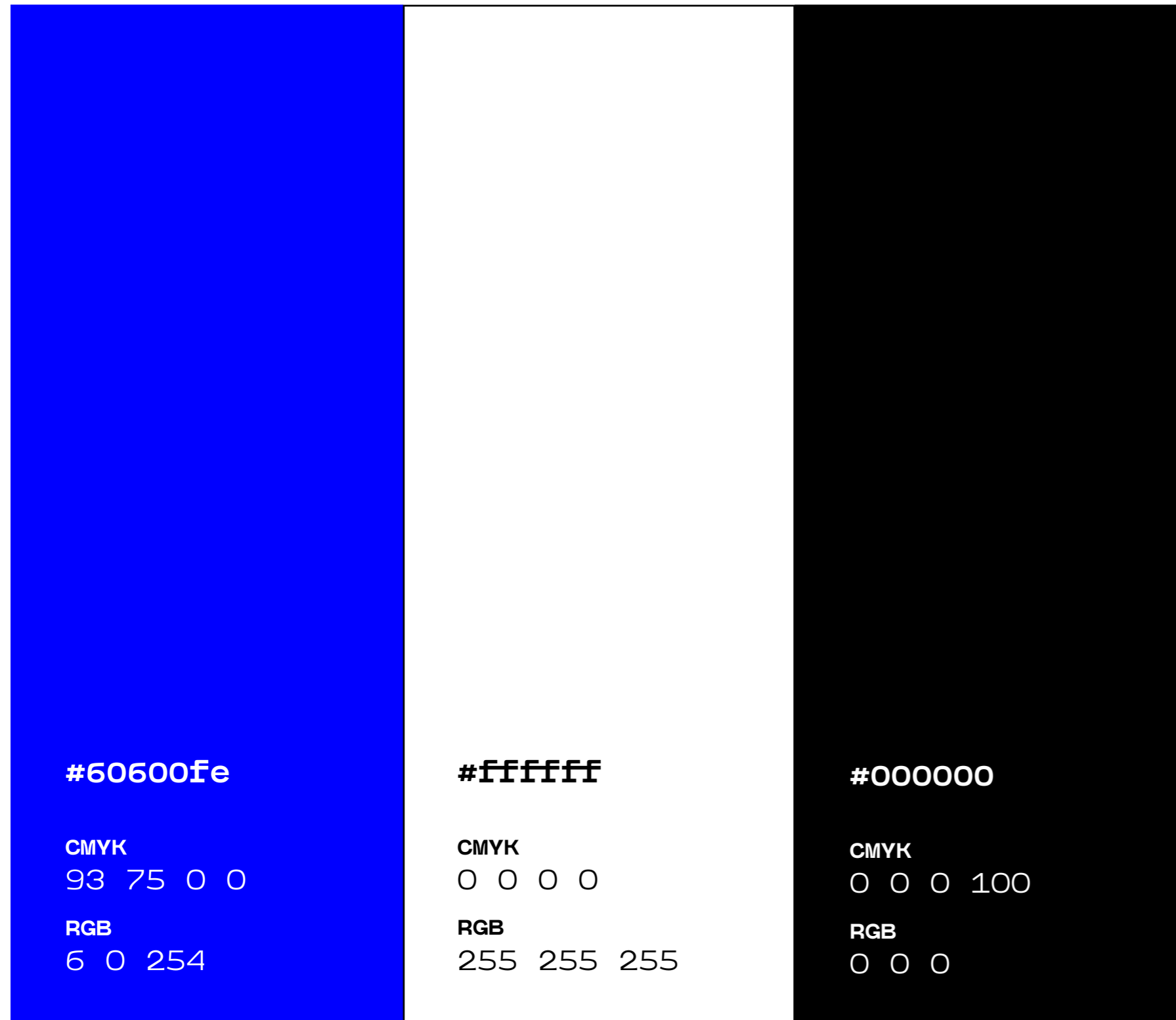


coding Lab



Palette principale

Colori generali



Palette secondaria

Colori dei corsi



Typeface

Il typeface scelto per l'identità del progetto è un originale e moderno monospaziato, al quale non ne viene accostato un secondo, poiché esso dispone di molti font dai pesi diversi, dal thin fino al black.

A motivo di questa scelta bisogna osservare diversi aspetti.

RIFLESSIONE DELLA PROGRAMMAZIONE

I caratteri monospaziati sono comunemente utilizzati in programmazione per scrivere codice.

Questa associazione è particolarmente significativa per un corso di coding, poiché sottolinea immediatamente il contesto di riferimento.

LEGGIBILITÀ

I caratteri monospaziati sono noti per la loro chiarezza e leggibilità, poiché progettati per facilitare la comprensione del testo, fondamentale nella programmazione, dove una struttura confusa può causare problemi significativi.

CONSISTENZA ESTETICA

Questo monospaziato si adatta bene all'ortogonalità del logo.

Tale accostamento crea infatti coerenza visiva tra il logo e il testo circostante, contribuendo a un aspetto complessivo armonioso e coerente.

Extra Light

Caleb Mono

Light

Caleb Mono

Regular

Caleb Mono

Medium

Caleb Mono

Semi Bold

Caleb Mono

Bold

Caleb Mono

Extra Bold

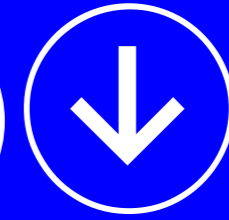
Caleb Mono

Black

Caleb Mono

Caleb

Mono



A a B b C c D d E 1 2 3 4 5 6 7 8 9 0

e F f G g H h I i j ① ② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨ ⑩

K k L l M m N n O ← ↑ → ↓ ↔ ↕ ↖ ↗ ↘ ↙

o P p Q q R r S s T t ← ↑ → ↓ ↔ ↕ ↖ ↗ ↘ ↙

U u V v W w X x Y y Z z ← ↑ → ↓ ↔ ↕ ↖ ↗ ↘ ↙

! # " \$ ' % & () * + - / < = > : ? @

{ } | ¢ ¤ § « » ‹ › — — — € № ™ © ® ∅ √ ∞ ∫ ≈ ≠ ≤

Visual

Pattern generativi dinamici

I visual di Coding Lab sono pattern generativi basati sulla palette dell'identità e realizzati utilizzando il linguaggio di programmazione JavaScript insieme alla libreria p5.js.

LA GRIGLIA

Il visual è composto da una griglia di quadrati colorati, ognuno dei quali rappresenta un'unità visiva fondamentale. Questi quadrati sono organizzati in modo preciso sulla griglia, creando una struttura visiva coerente.

Ogni quadrato è riempito con uno dei colori della palette in modo casuale, aggiungendo un elemento di creatività e varietà al visual. Questa variabilità riflette l'approccio creativo e l'aspetto giocoso che può essere associato al processo di sviluppo.

VERSATILITÀ

Le sue variazioni di colore e la sua struttura organizzata lo rendono un simbolo versatile che può essere utilizzato in una varietà di contesti per promuovere il corso e comunicare il suo carattere unico. Può essere utilizzato come sfondo grafico in materiali promozionali, presentazioni, siti web od all'interno dei materiali didattici del corso.

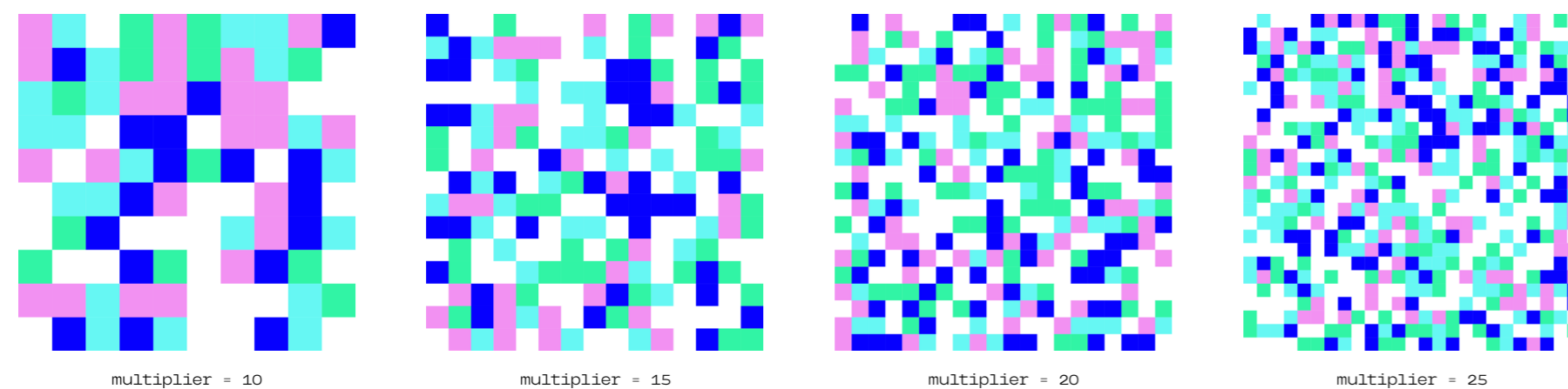


Fig. 62 - Modifica variabile

Semplicemente modificando la sola variabile "multiplier", è possibile aumentare il numero di elementi nella griglia.

DEFINIZIONE DEI COLORI

I colori sono definiti in un array denominato "colors". Ogni colore è rappresentato da un codice esadecimale. Sono i sei colori dell'identità visiva, che verranno poi richiamati con numeri da 0 a 5.

SPECIFICHE GRIGLIA

La variabile "multiplier" serve ad aumentare il numero di quadrati nella griglia generativa. Le variabili "sizeW" e "sizeH" definiscono larghezza e altezza del canvas in cm, mentre "inch" e "dpi" serviranno a definire la risoluzione.

DEFINIZIONE MCD

La funzione "MCD" (Massimo Comun Divisore) è utilizzata per calcolare il numero da dividere per la larghezza ("sizeW") e l'altezza ("sizeH") del canvas. Questo assicura che vengano sempre generati quadrati anche quando il canvas ha dimensioni rettangolari.

GENERAZIONE PATTERN

La funzione "drawGrid" viene chiamata per disegnare i quadrati nella griglia. Viene quindi assegnato un colore ai quadrati in base alle condizioni ai valori della variabile "rnd" specificati nel codice. Ogni condizione "if (rnd == ...)" contiene un metodo "fill()" che corrisponde a uno dei colori dell'array "colors", garantendo il rispetto della palette. Questi colori possono essere cambiati a seconda del pattern che bisogna generare.

CODICE SORGENTE INTEGRALE

<https://editor.p5js.org/gabriele.bass/sketches/DQuKbfmOE>

```
let colors = [
  "#0600fe",
  "#ffffff",
  "#000000",
  "#31f4a5",
  "#66f7f3",
  "#f291f2",];

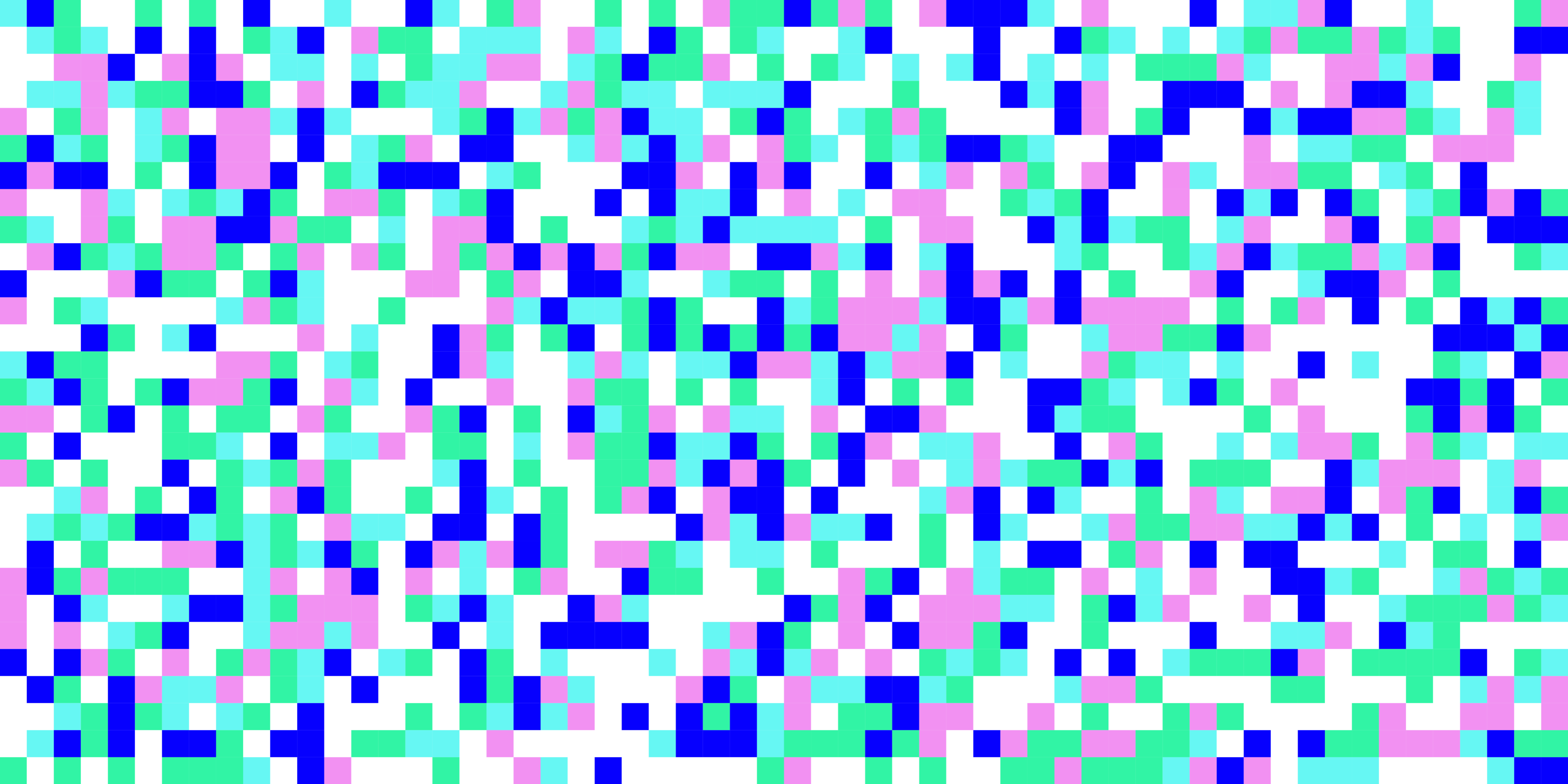
let multiplier = 1;

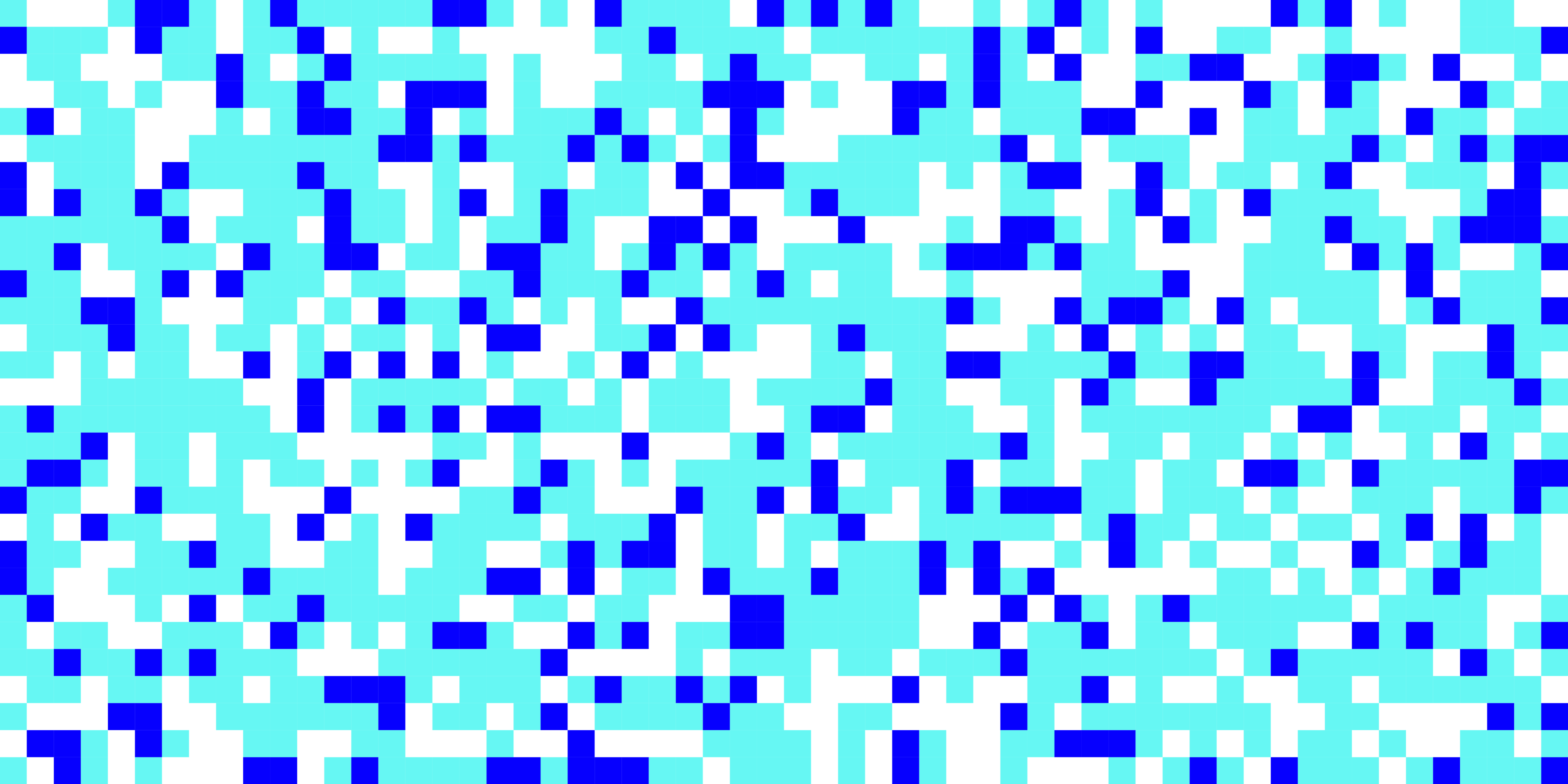
let sizeW = 29;
let sizeH = 29;
let inch = 2.54;
let dpi = 300;

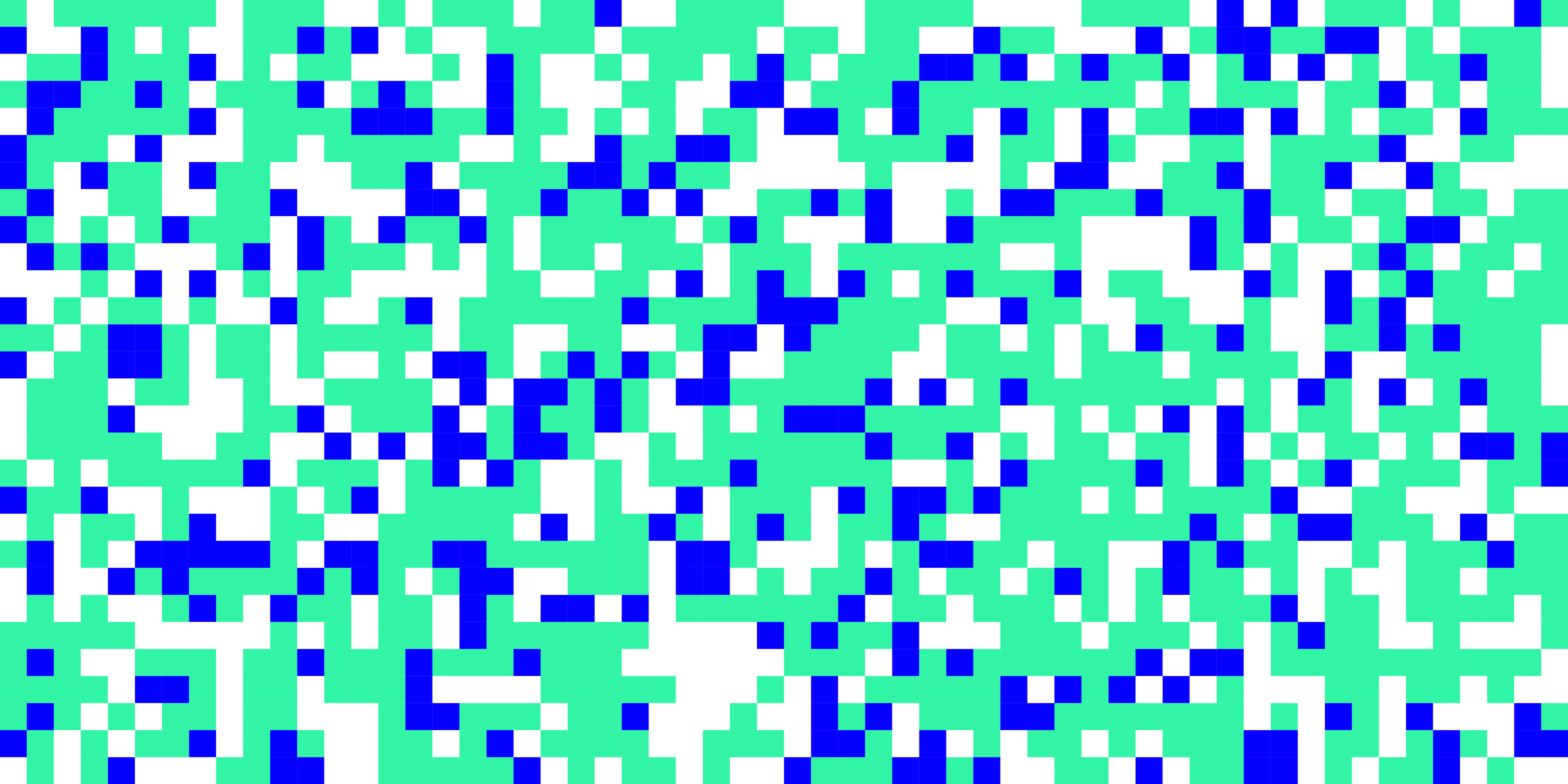
function MCD(K, n) {
  return K ? MCD(n % K, K) : n;
}

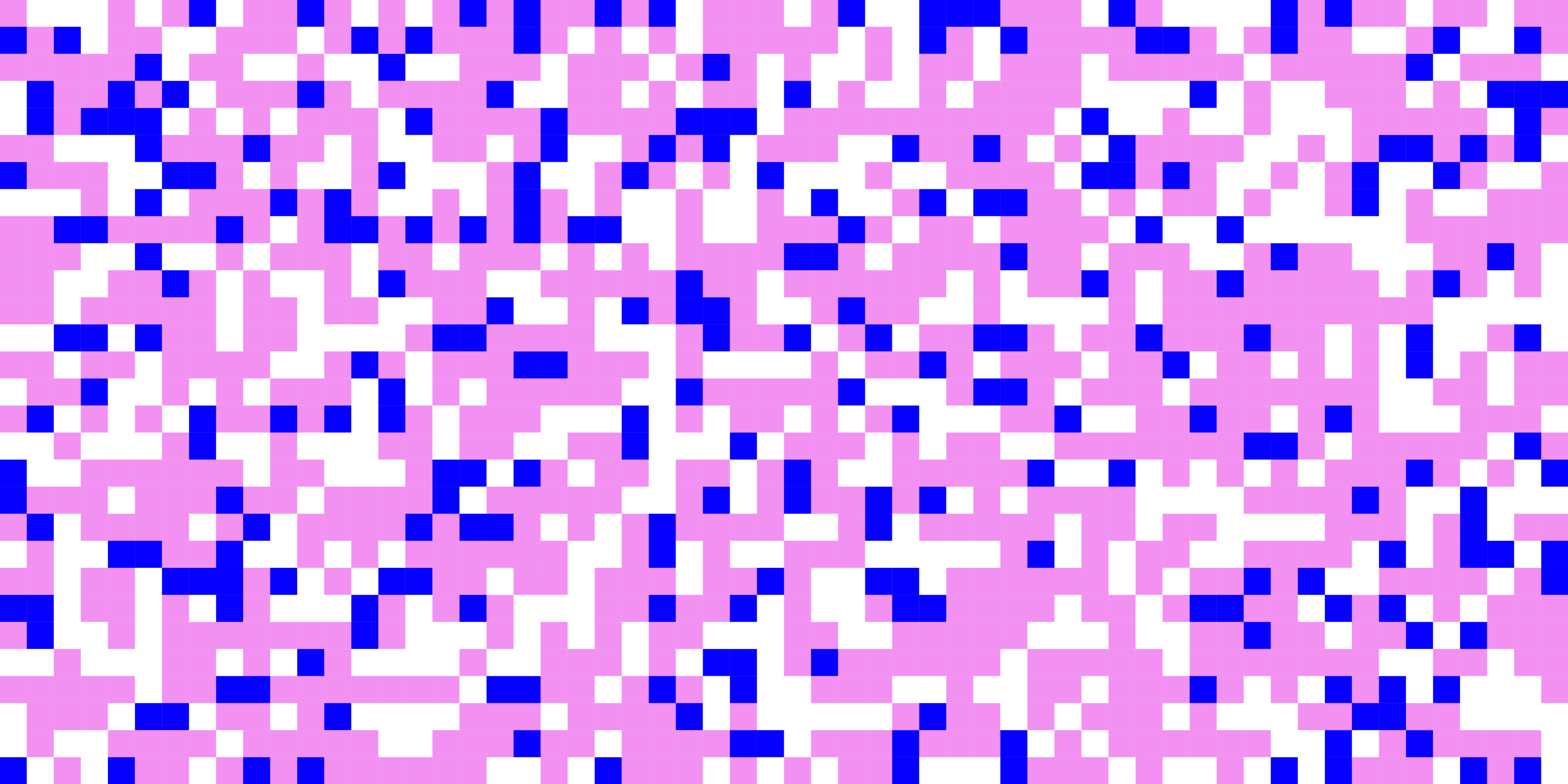
let nItemsW = (sizeW / MCD(sizeW, sizeH)) * multiplier;
let nItemsH = (sizeH / MCD(sizeW, sizeH)) * multiplier;
...

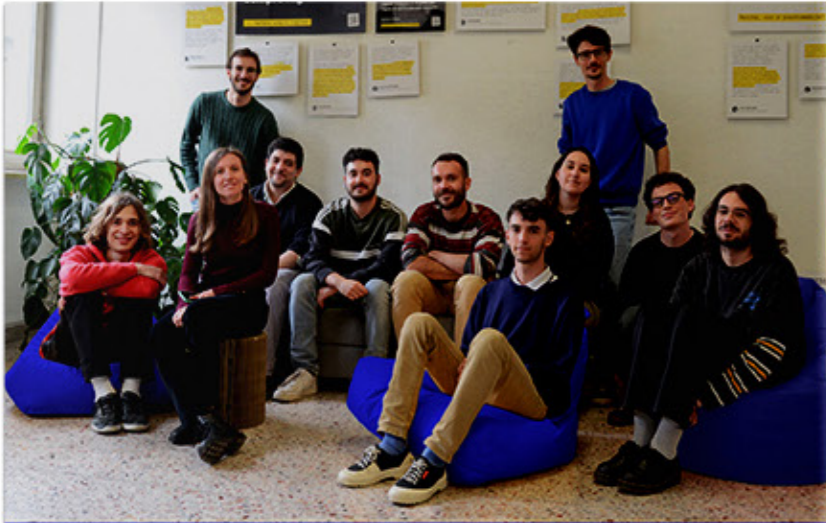
function drawGrid() {...
  let rnd = int(random(0, 6));
  noStroke();
  if (rnd == 0) {
    fill(colors[0]);
    rect(0, 0, itemW, itemH);
  } else if (rnd == 1) {
    fill(colors[1]);
    rect(0, 0, itemW, itemH);
  } else if (rnd == 2) {
    fill(colors[1]);
    rect(0, 0, itemW, itemH);
  } else if (rnd == 3) {
    fill(colors[3]);
    rect(0, 0, itemW, itemH);
  } else if (rnd == 4) {
    fill(colors[4]);
    rect(0, 0, itemW, itemH);
  } else {
    fill(colors[5]);
    rect(0, 0, itemW, itemH);
  }
}
```











Coding Lab è molto più di un semplice corso di programmazione.

Siamo una community aperta e inclusiva di creativi e progettisti che crede nel superpotere del coding applicato al design thinking.

Immagina di poter trasformare audaci idee di design in realtà digitale. Con il coding possiamo essere creatori di esperienze uniche.

È ciò che facciamo qui a Coding Lab.

Non importa qual è il tuo background, la nostra community è aperta a chiunque abbia il desiderio di apprendere e creare.

Unisciti su Discord, scopri il tuo potenziale e sali di livello con noi!



Coding Lab Community



Pronti per un level up?



Via A. di Bernezzo 34,
Torino



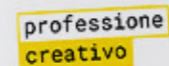
hello@meKit.it



+39 367 66 49 874



In collaborazione con



The Creativity Upgrade



coding Lab

Creative Coding

Codifica la creatività

Il creative coding è la chiave per dar vita a nuove visioni digitali.

Entrando nel corso imparerai a unire il tuo talento artistico con il coding per creare opere interattive.

Scoprirai p5.js e javascript per realizzare coinvolgenti ambienti virtuali.



start
Ottobre 2023



Coding Lab è molto più di un semplice corso di programmazione.

Siamo una community aperta e inclusiva di creativi e progettisti che crede nel superpotere del coding applicato al design thinking.

Immagina di poter trasformare audaci idee di design in realtà digitale. Con il coding possiamo essere creatori di esperienze uniche.

È ciò che facciamo qui a Coding Lab.

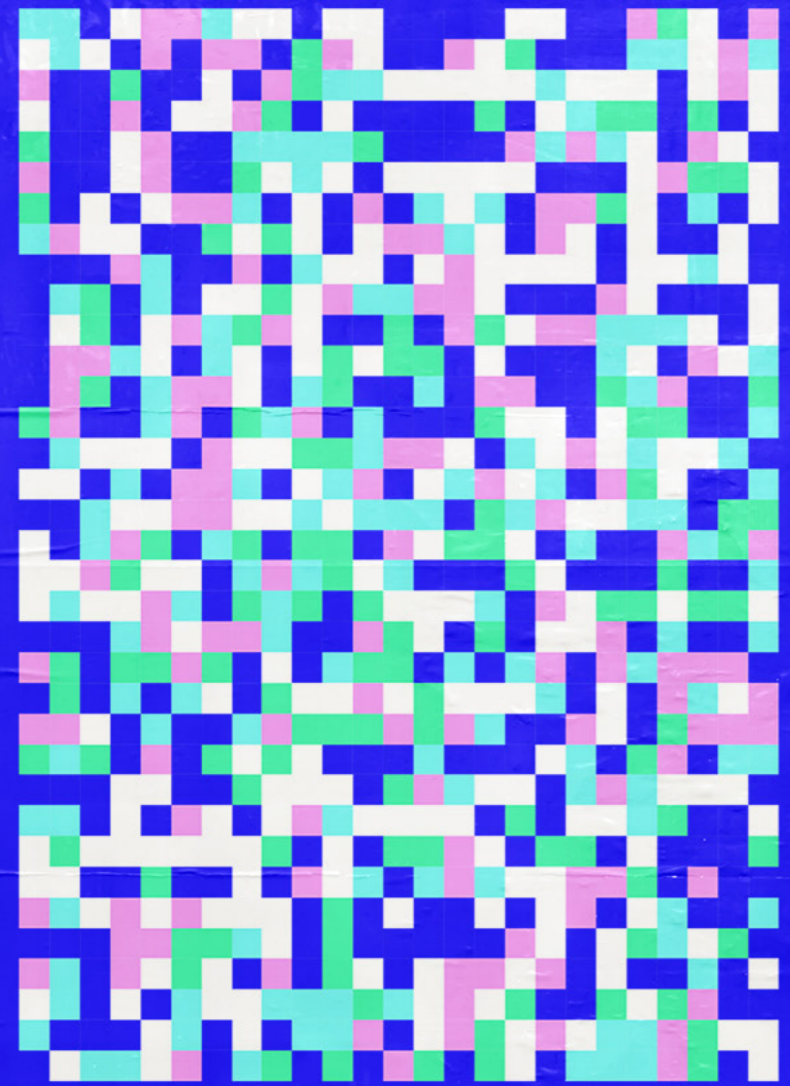
Non importa qual è il tuo background, la nostra community è aperta a chiunque abbia il desiderio di apprendere e creare.

Unisciti su Discord, scopri il tuo potenziale e sali di livello con noi!



Coding Lab Community

CREATIVE CODING ↘

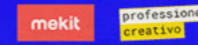
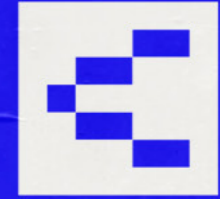


Il codice è uno strumento, una stringa di numeri, glifi e lettere che, se disposti in un ordine particolare, possono essere branditi come un cacciavite. Ma il codice è anche un mezzo visivo, come l'illustrazione o la scultura, che nelle mani giuste può creare qualcosa di visivamente evocativo. Poetico, anche.

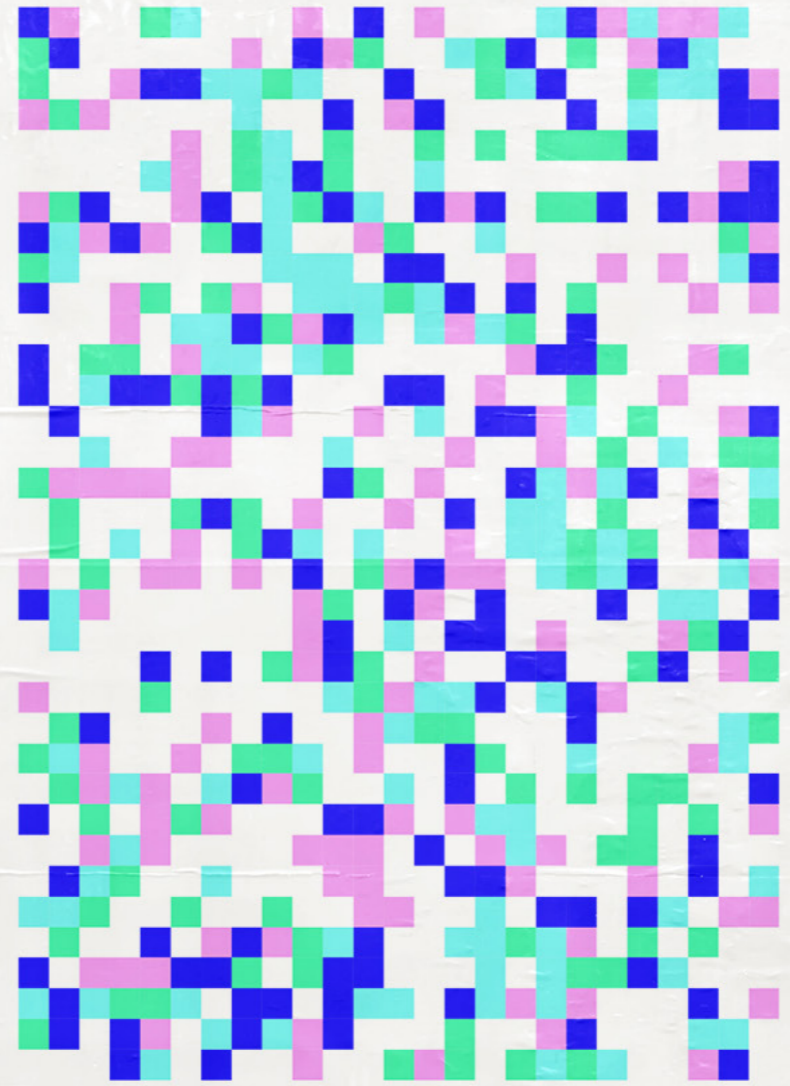
MEKIT SO
VIA ASPHET 05, MERRAZZO 04, TORINO
WWW.MEKIT.IT
WWW.MEKIT.CODINGLAB.IT

Coding Lab è un progetto di formazione a cura di mekit so e con la collaborazione di Professione Creativa.

Prodotto per un livello 4/5 di progetti correlati a guide gratuite per cominciare nel mondo del coding. INDIRIZZI AL COMUNITA' NEL SERVICIO CODING LAB SU DISCORD!



CREATIVE CODING ↘

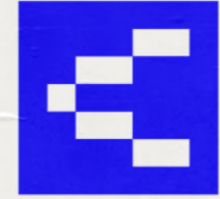


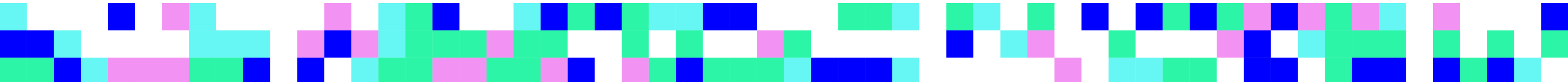
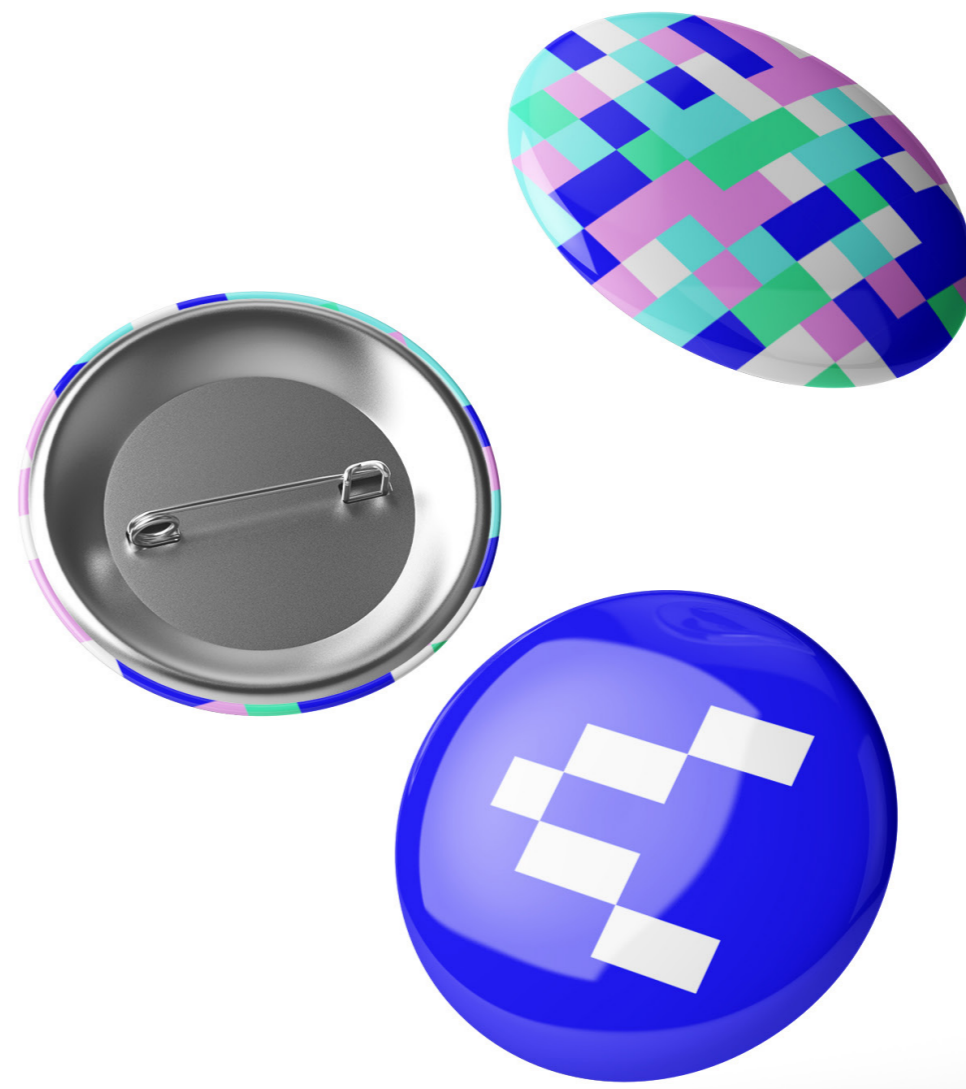
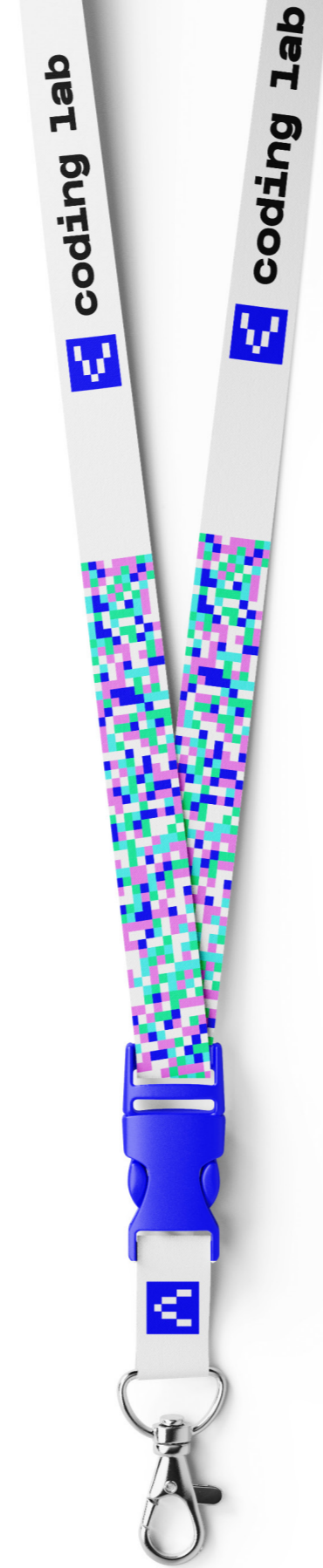
Il codice è uno strumento, una stringa di numeri, glifi e lettere che, se disposti in un ordine particolare, possono essere branditi come un cacciavite. Ma il codice è anche un mezzo visivo, come l'illustrazione o la scultura, che nelle mani giuste può creare qualcosa di visivamente evocativo. Poetico, anche.

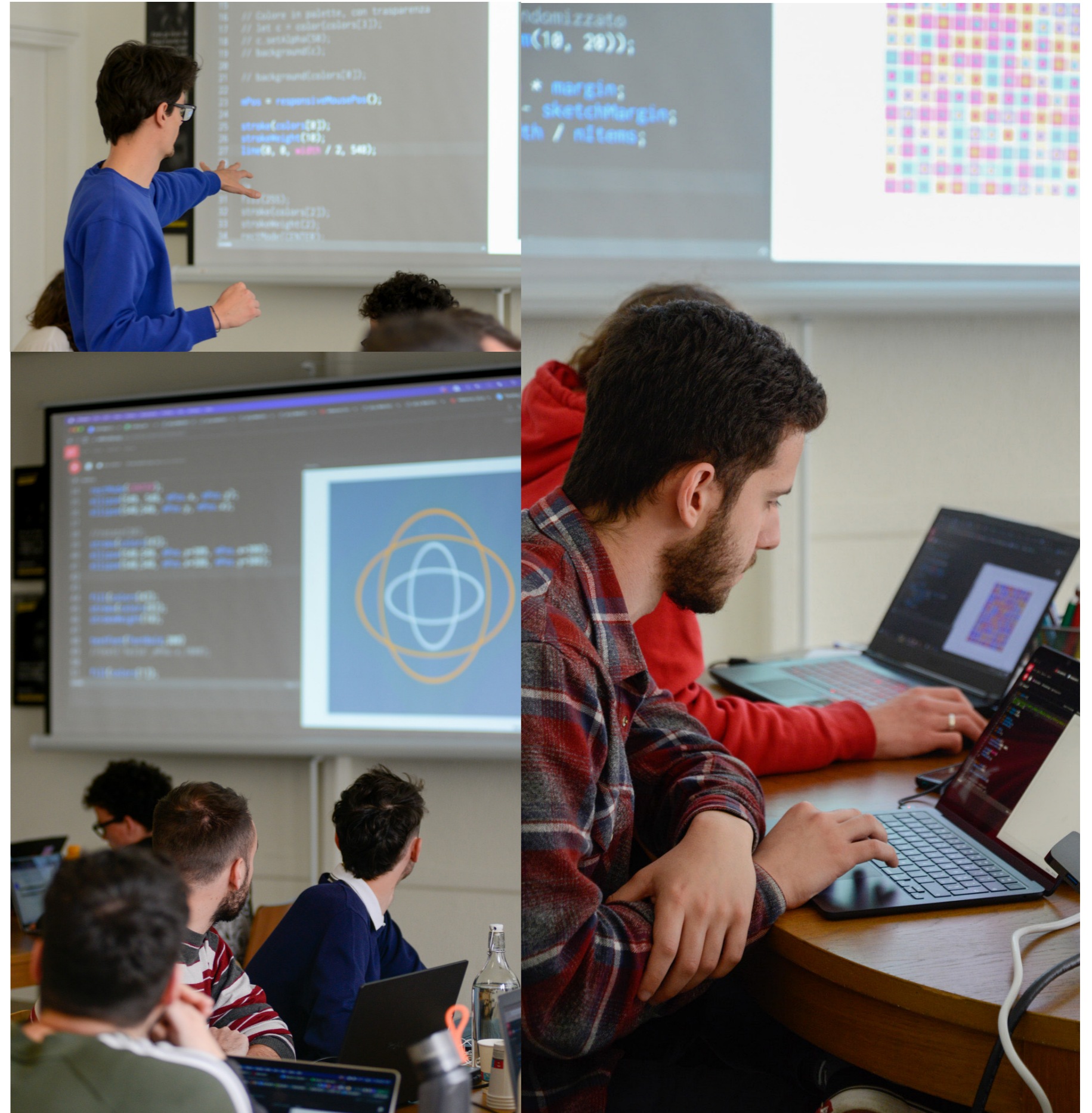
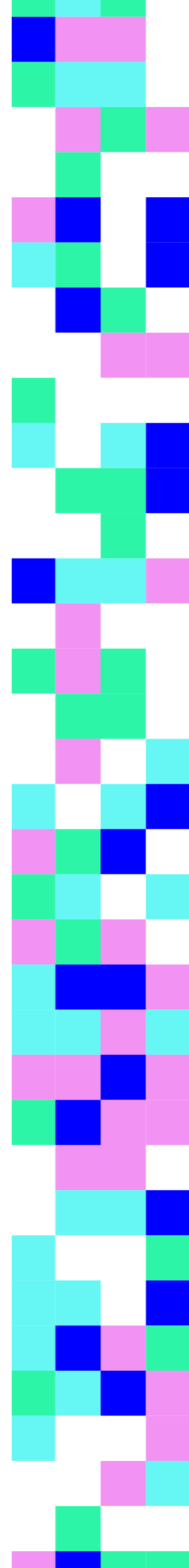
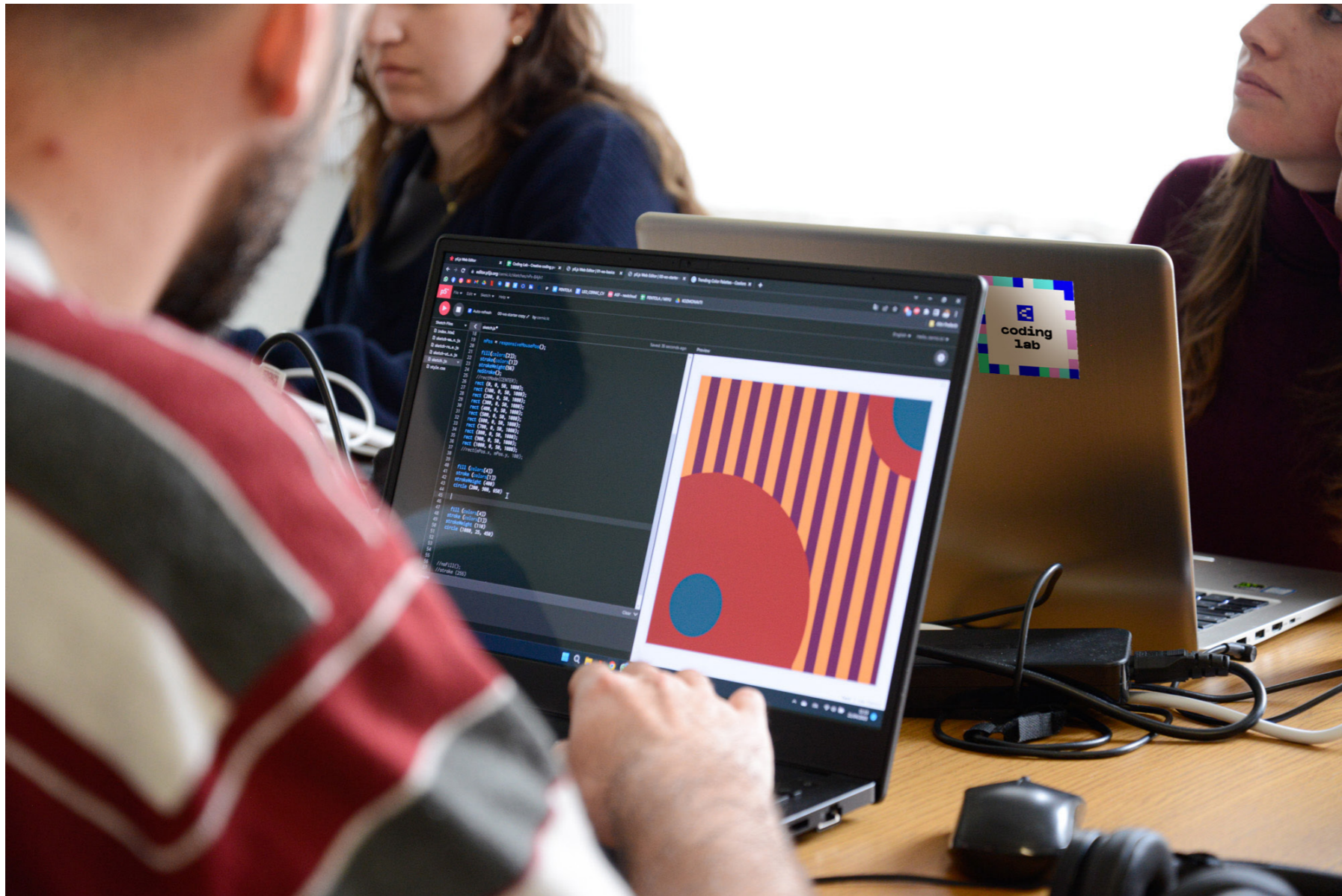
MEKIT SO
VIA ASPHET 05, MERRAZZO 04, TORINO
WWW.MEKIT.IT
WWW.MEKIT.CODINGLAB.IT

Coding Lab è un progetto di formazione a cura di mekit so e con la collaborazione di Professione Creativa.

Prodotto per un livello 4/5 di progetti correlati a guide gratuite per cominciare nel mondo del coding. INDIRIZZI AL COMUNITA' NEL SERVICIO CODING LAB SU DISCORD!







Strategia

La strategia di comunicazione è stata concepita come distinta in due fasi: pre-lancio e post-lancio.

La fase di pre-lancio si concentra sulla creazione di consapevolezza e sulla raccolta di iscritti, mentre la fase di post-lancio mira a mantenere il coinvolgimento dei corsisti attraverso la condivisione di contenuti di valore e la promozione di attività interne alla community.

In entrambe le fasi, l'obiettivo principale è fornire un ambiente di apprendimento e crescita professionale di alta qualità per i designer e i creativi interessati.

FASE 1: PRE-LANCIO

1. SERVER DISCORD DEDICATO

Verrà creato un **server Discord**, strutturato in modo da avere canali pubblici aperti a tutta la community e canali privati riservati esclusivamente agli iscritti ai corsi.

Questa suddivisione consentirà di mantenere una forte **interazione tra gli iscritti** e di promuovere l'accesso ai corsi.

2. COSTRUZIONE GOOGLE FORM

L'accesso ai corsi sarà possibile successivamente alla compilazione di un Google Form, alla fine del quale i partecipanti riceveranno un **link d'invito** per unirsi al server Discord.

Tale approccio permette di raccogliere non solo iscritti ma anche informazioni generali sull'utenza interessata, che saranno utili per le future **strategie di targetizzazione**.

3. POST INSTAGRAM SPONSORIZZATI

I profili Instagram di *MeKit* e *Professione Creativo* saranno utilizzati per pubblicare post sponsorizzati ed adeguatamente targetizzati che conterranno **link al Google Form** di iscrizione.

Questa strategia permetterà di raggiungere un pubblico ampio e interessato al progetto.

4. CONDIVISIONE CONTENUTI

Sfrutteremo i server **Discord** e i profili **LinkedIn** di *Professione Creativo*, *Digital Communication*, *MeKit* e *Sherpa Community* per condividere il **link al Google Form** e promuovere Coding Lab.

Questo consentirà di raggiungere una rete di professionisti e appassionati del settore.

FASE 2: POST LANCIO

1. LEZIONI

Le lezioni potranno svolgersi sia in presenza che online, utilizzando il server Discord come piattaforma per lo streaming.

Questa flessibilità consentirà ai corsisti di partecipare secondo le proprie preferenze e disponibilità.

2. UTILIZZO CANALI DISCORD

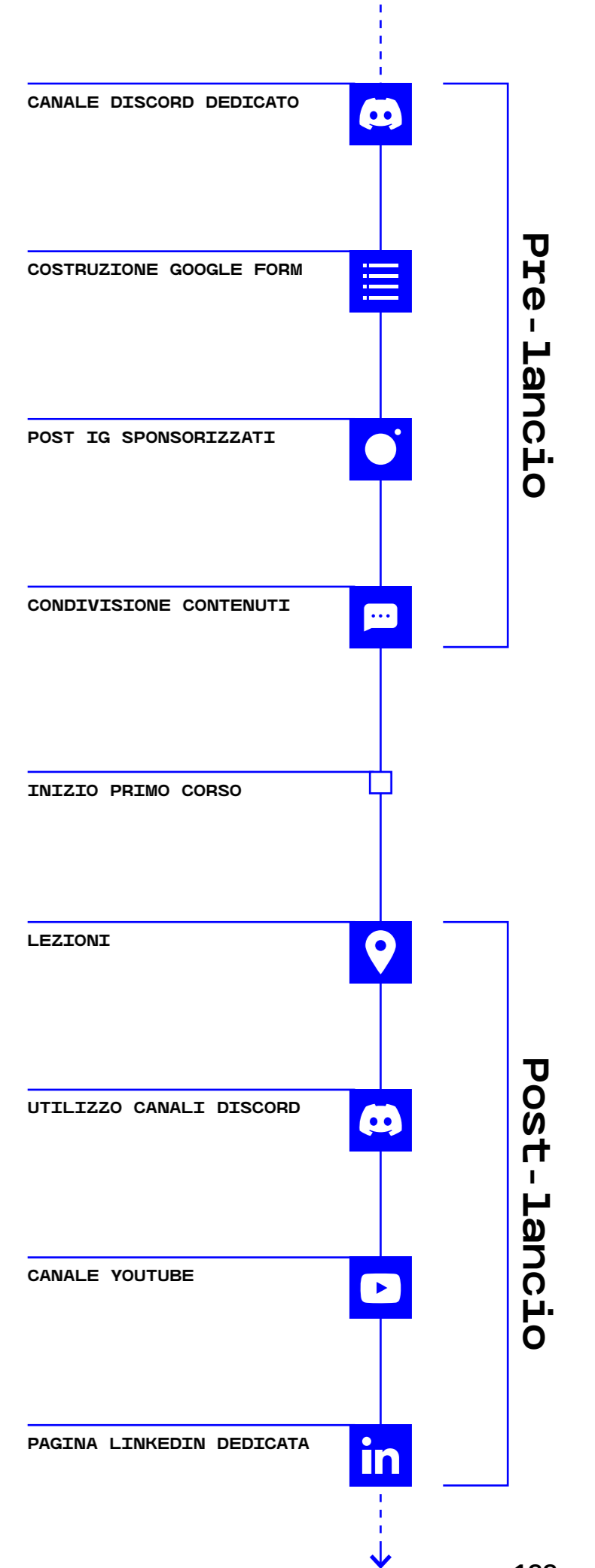
I canali privati del server Discord di Coding Lab verranno sfruttati per servizi dedicati ai corsisti. I canali pubblici saranno invece dedicati a temi di carattere generale, risorse e tutorial gratuiti fruibili dall'intera community, promuovendo anche attività di networking.

3. CANALE YOUTUBE

Il canale YouTube è pensato per la pubblicazione di tutorial sullo sviluppo ed al tempo stesso per promuovere Coding Lab e la sua community. Questo materiale fornirà valore aggiunto al progetto e contribuirà a posizionare Coding Lab come una risorsa affidabile per l'apprendimento.

4. PAGINA LINKEDIN DEDICATA

A questo punto verrà creata una pagina LinkedIn istituzionale per Coding Lab. Qui si potranno pubblicare post e articoli su argomenti legati al mondo tech, al design e alle loro correlazioni. La presenza su LinkedIn permetterà di stabilire una presenza nel settore della formazione, condividere informazioni e di creare awareness attorno al progetto.



Server Discord

Ingresso

La community sarà raccolta principalmente attorno al server Discord.

Una volta fornito il link di invito, l'utente dovrà inserire una e-mail di verifica ed accettare le regole del server, dopodiché verrà reindirizzato al canale "#rules", dove gli verranno fornite le istruzioni per usare al meglio il server e muovere i primi passi nella community.

I canali sono raccolti all'interno di categorie distinte.

The screenshot shows the Discord onboarding interface for the 'Coding Lab' server. It includes a description of the community, a verification step for the user's email, and a list of server rules. A blue line connects the 'Accetta le regole' section to the Discord chat window on the right.

Coding Lab

Questa è la community di Coding Lab, un laboratorio rivolto a designer e creativi per apprendere e approfondire nuove skills su tre ambiti legati al coding: front-end development, creative coding e game development. Che aspetti? Unisciti a noi!

Per parlare qui...
Devi completare i passaggi necessari qui sotto.

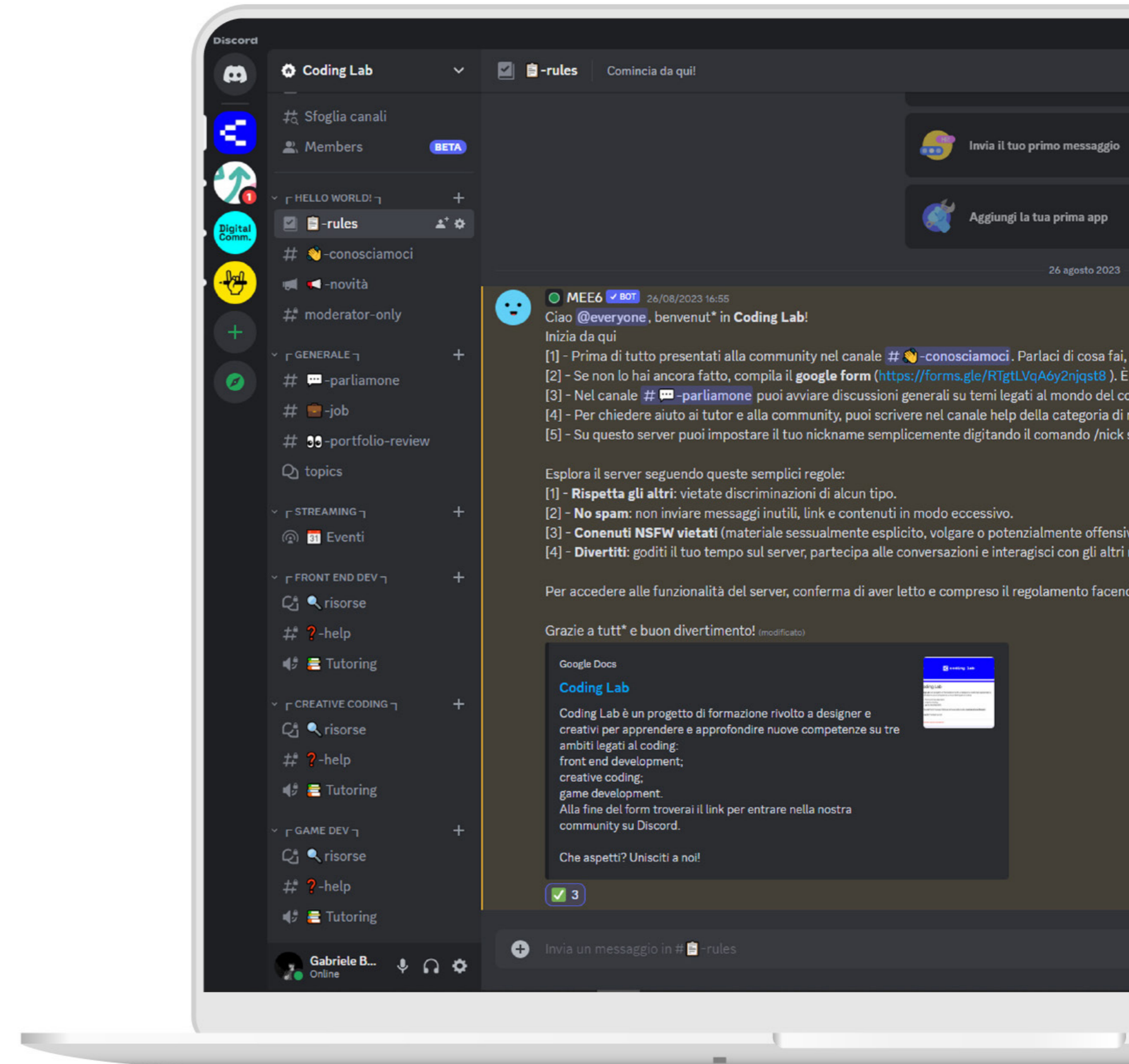
Verifica il tuo indirizzo e-mail.

La tua e-mail e il tuo numero di telefono non verranno condivisi con lo staff della community.

Accetta le regole

- Se non lo hai ancora fatto, compila il **google form** (<https://forms.gle/RTgtLVqA6y2njqt8>). È **obbligatorio** per iscriversi ai corsi.
- Rispetta gli altri.** Non sarà tollerato alcun tipo di molestia, persecuzione, sessismo, razzismo o incitamento all'odio.
- No spam** o autopromozione (inviti a server, pubblicità, ecc.) senza il permesso di un membro dello staff. Ciò include inviare messaggi diretti ai membri.
- Niente contenuti NSFW.** Ciò include testi, immagini o link contenenti nudità, sesso, violenza

5 membri



Server Discord

Categorie e canali

HELLO WORLD! - PUBBLICA

#rules	Regole ed informazioni sui primi passi da compiere sul server (presentarsi, ecc.).
#conosciamoci	Messaggi di benvenuto automatici e presentazioni dei membri.
#novità	Annunci inerenti Coding Lab.
#moderator-only	Canale privato degli admin del server.

GENERALE - PUBBLICA

#parliamone	Chat generale per i membri della community.
#job	Chat per diffondere annunci di lavoro.
#portfolio-review	Chat per mostrare i propri progetti e chiedere feedback.
#topics	Canale forum dove condividere, e arricchire risorse.

STREAMING - PUBBLICA

#eventi	Canale eventi per il live streaming di conferenze e simili.
---------	---

FRONT-END DEV - PRIVATA

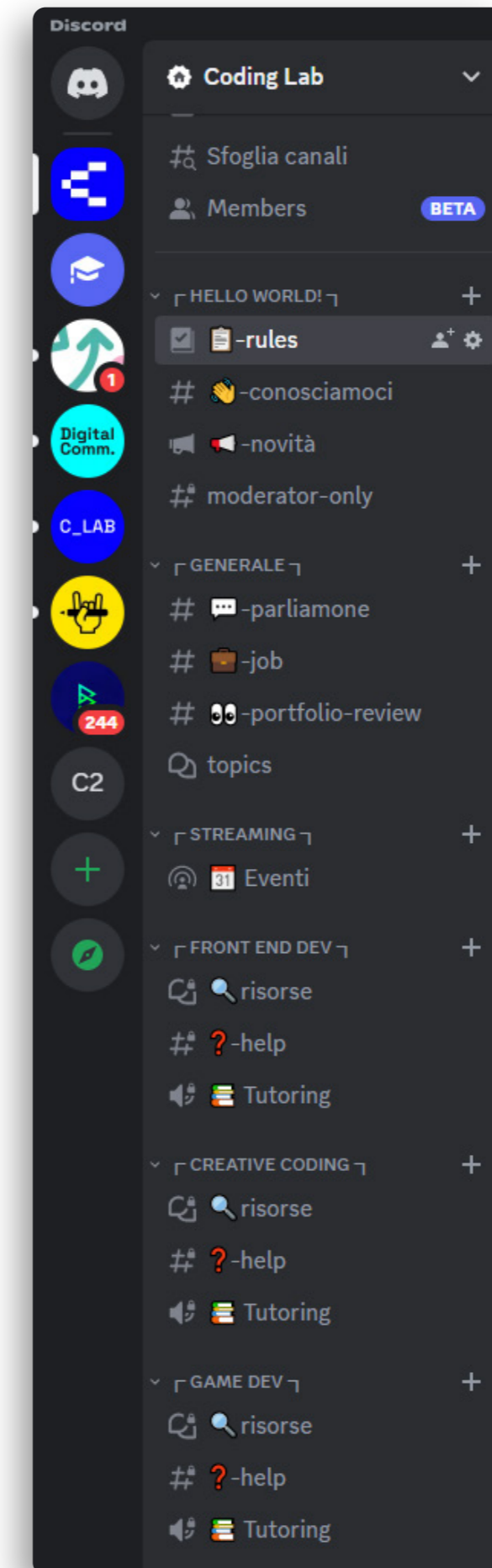
#risorse	Canale forum dove raccogliere risorse per lo sviluppo.
#help	Canale per chiedere aiuto su sviluppo, programmi, esercitazioni, ecc..
#tutoring	Chat vocale/video per lezioni e tutoring.

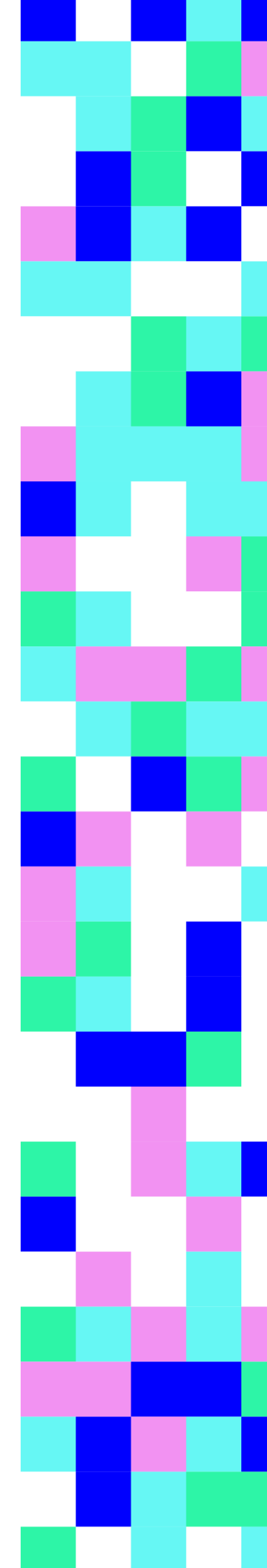
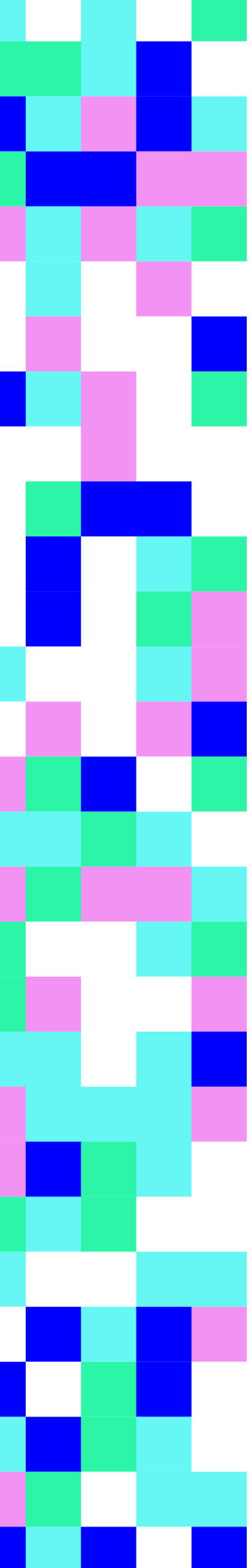
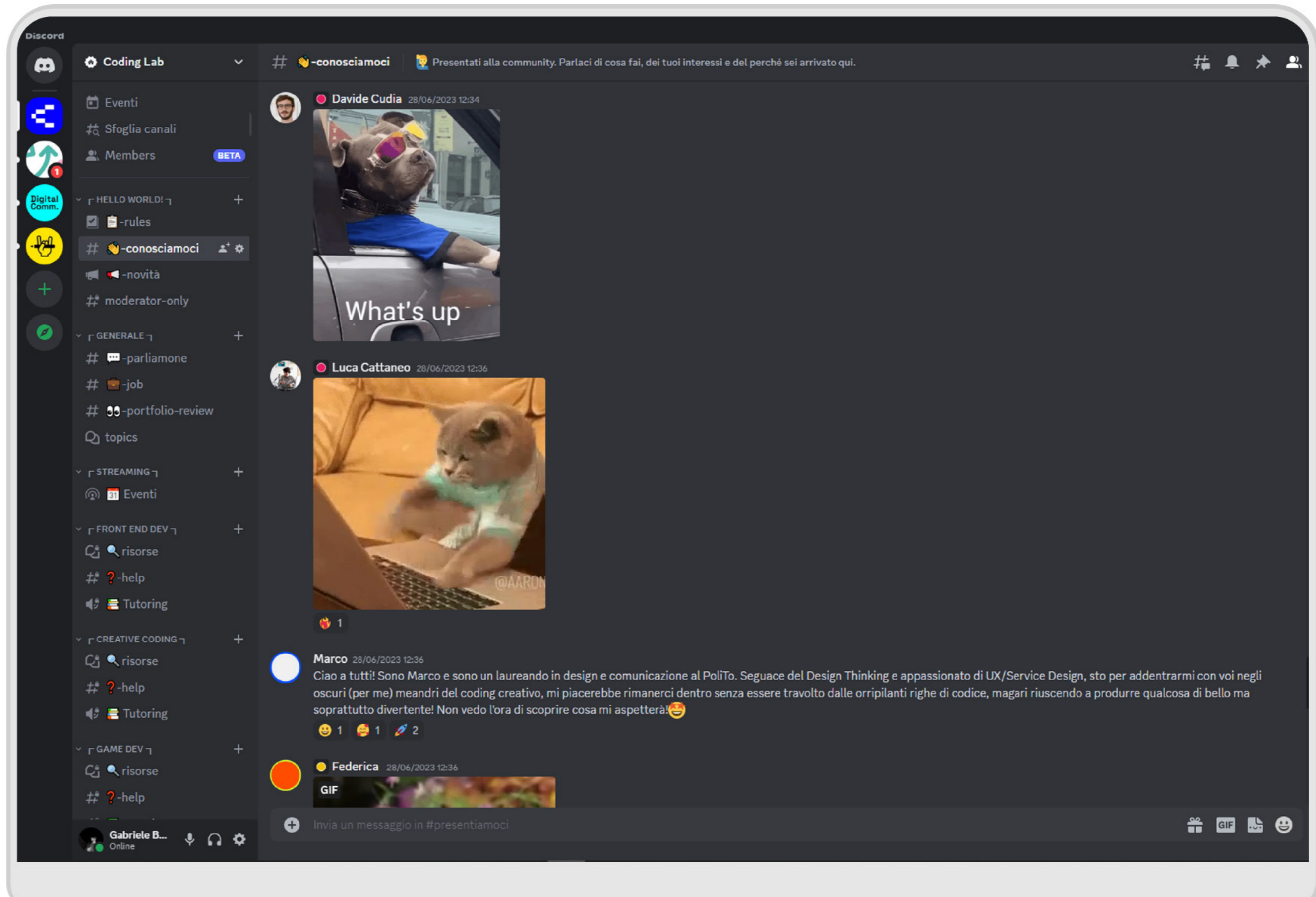
CREATIVE CODING - PRIVATA

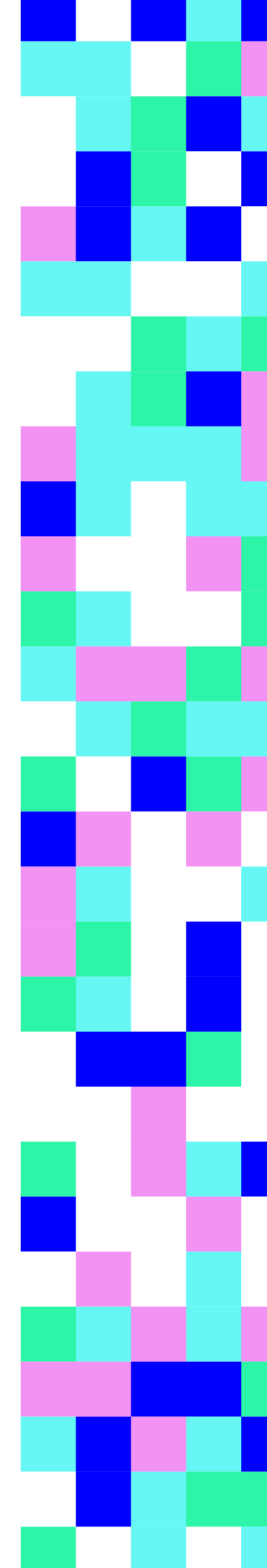
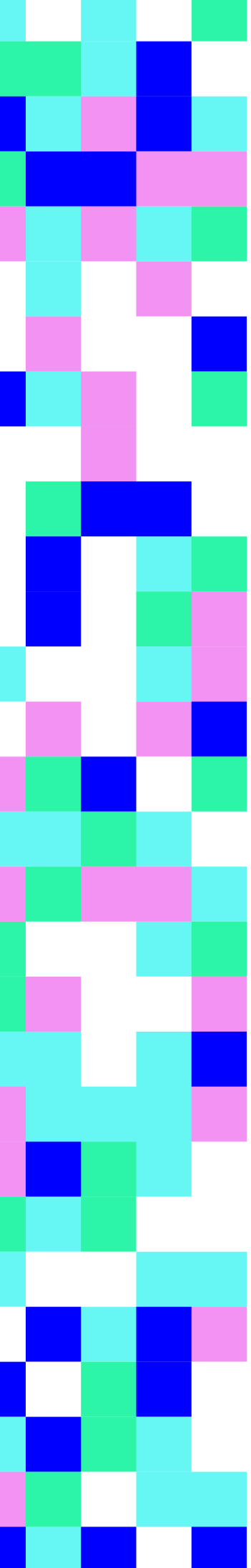
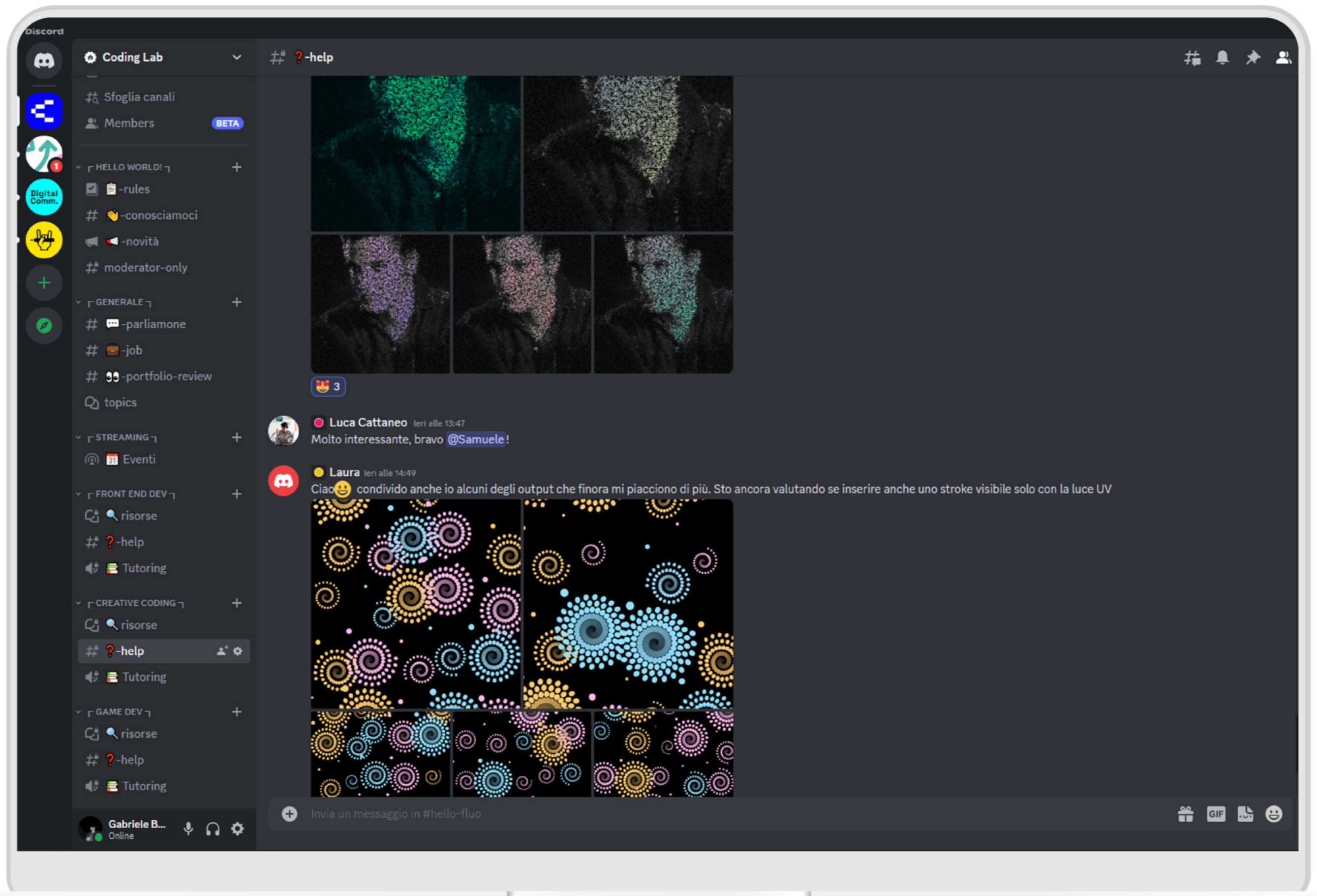
...

GAME DEV - PRIVATA

...





















Server Discord

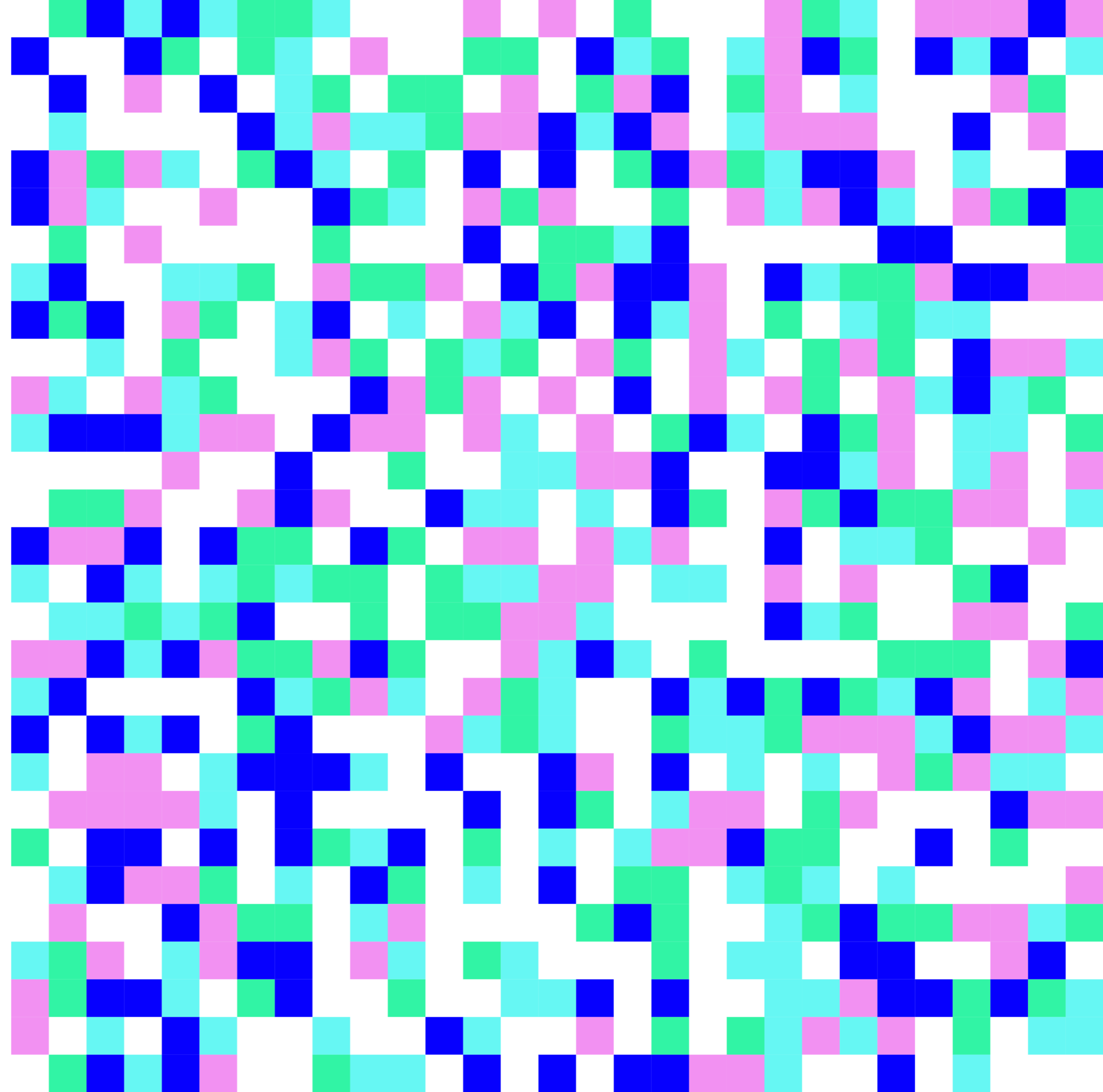
Ruoli del server

Ad ognuno dei membri verrà assegnato il ruolo di utente verificato nel momento in cui accetterà le regole della community.

Il ruolo di studente di uno dei corsi verrà attribuito da uno dei Tutor o dei membri dello staff.

Nell'immagine a fronte sono riportati i primi ruoli inseriti alla creazione del server.

RUOLI - 7	MEMBRI
 MEE6	1 
 Verificato	4 
 Tutor	2 
 Studente front-end dev	0 
 Studente creative coding	0 
 Studente game dev	0 
 Staff	3 



Google form

Questionario d'iscrizione

Come detto, per iscriversi ai corsi con posti limitati, sarà indispensabile compilare un Google Form, alla fine del quale i partecipanti riceveranno il **link d'invito** al server Discord.

In questo modo sarà possibile raccogliere le iscrizioni ed anche alcune informazioni generali sul background dell'utente interessata, in maniera tale da fornire un dataset utile per future **strategie di targetizzazione**.

CODING LAB - SEZIONE 1

Introduzione	Presentazione del corso e claim con CTA "Sei pronto per un level up? Unisciti a noi!".
E-mail	Inserire un indirizzo e-mail valido.

CONOSCIAMOCI! - SEZIONE 2

Domanda 1	Inserire nome.
Domanda 2	Inserire Cognome
Domanda 3	Selezionare Età
Domanda 4	Selezionare professione.
Domanda 5	Linguaggi di programmazione conosciuti
Domanda 6	Corsi d'interesse.
Domanda 7	Come ha conosciuto Coding Lab.

LINK DISCORD - SEZIONE 3

Link	Link alla Community.
-------------	----------------------

The image displays a Google Form titled "Coding Lab" on both a desktop monitor and a smartphone. The desktop view shows the full form with a header, a list of course topics, a Discord link, and a CTA. The smartphone view shows a zoomed-in portion of the form, including the title, course topics, and the CTA.

coding lab

Project by: mekit & profession creation

Coding Lab

Coding Lab è un progetto di formazione rivolto a designer e creativi per apprendere ed elevare nuove competenze su tre ambiti legati al coding:

- front end development;
- creative coding;
- game development.

Alla fine del form troverai il link per entrare nella nostra **community su Discord**.

Potrai accedere ai corsi, conoscere altri coders e fare networking, trovare guide e tutorial gratuiti.

- Form obbligatorio per l'iscrizione ai corsi (posti limitati). -

Sei pronto per un level up? Unisciti a noi!

basile.graphics@gmail.com [Cambia account](#)

* Indica una domanda obbligatoria

Email *

Accedi a Google per salvare i risultati raggiunti. [Scopri di più](#)

Instagram

Post sponsorizzati

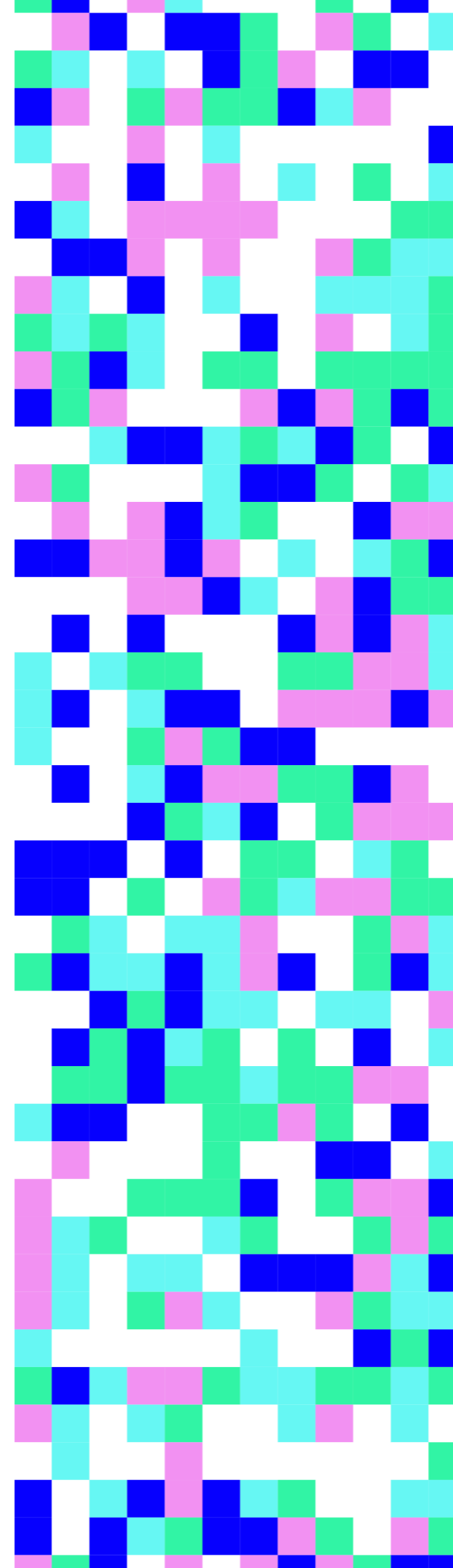
Non è previsto, almeno per il momento, di creare un profilo Instagram esclusivamente di Coding Lab. Per questo motivo, la condivisione di post sponsorizzati passa attraverso i profili Instagram dei partner MeKit e Professione Creativo.

Gli utenti avranno modo di raggiungere il **Google Form** tramite link.

Il pubblico target verrà filtrato sulla base di alcuni parametri demografici ed interessi:

Luogo:	Torino e Piemonte
Età:	da 18 a 35 anni
Generi:	tutti
Interessi:	Web design Sviluppo web Javascript Front-end e back-end Videogame development Game design Creative technology Art & graphic design Progettazione grafica Corsi online

Questa strategia permetterà di raggiungere un pubblico ampio e interessato al progetto.



Lezioni offline / online

Slides

Consentire lezioni sia in presenza che online significa fornire corsi accessibili a studenti da diverse aree geografiche.

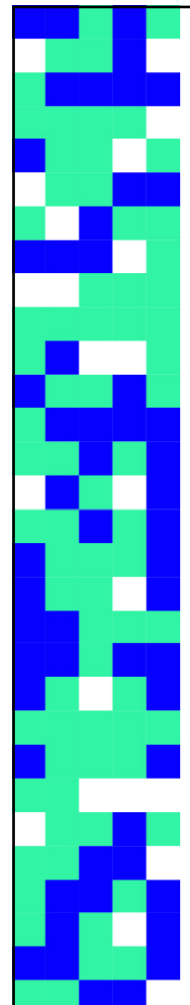
Un approccio ibrido tra lezioni in presenza e in live streaming, consente di adattare i corsi alle diverse esigenze, oltre ad offrire numerosi vantaggi in termini di partecipazione e apprendimento personalizzato.


A sostegno di tale approccio, agli studenti verranno

anche fornite risorse didattiche, come dispense e slides graficamente coerenti con l'immagine coordinata.

Le slides aggiungono un supporto visivo alle lezioni, aiutando gli studenti a comprendere e assimilare meglio gli argomenti trattati.

Le dispense potranno essere scaricate e utilizzate per la revisione e lo studio autonomo.



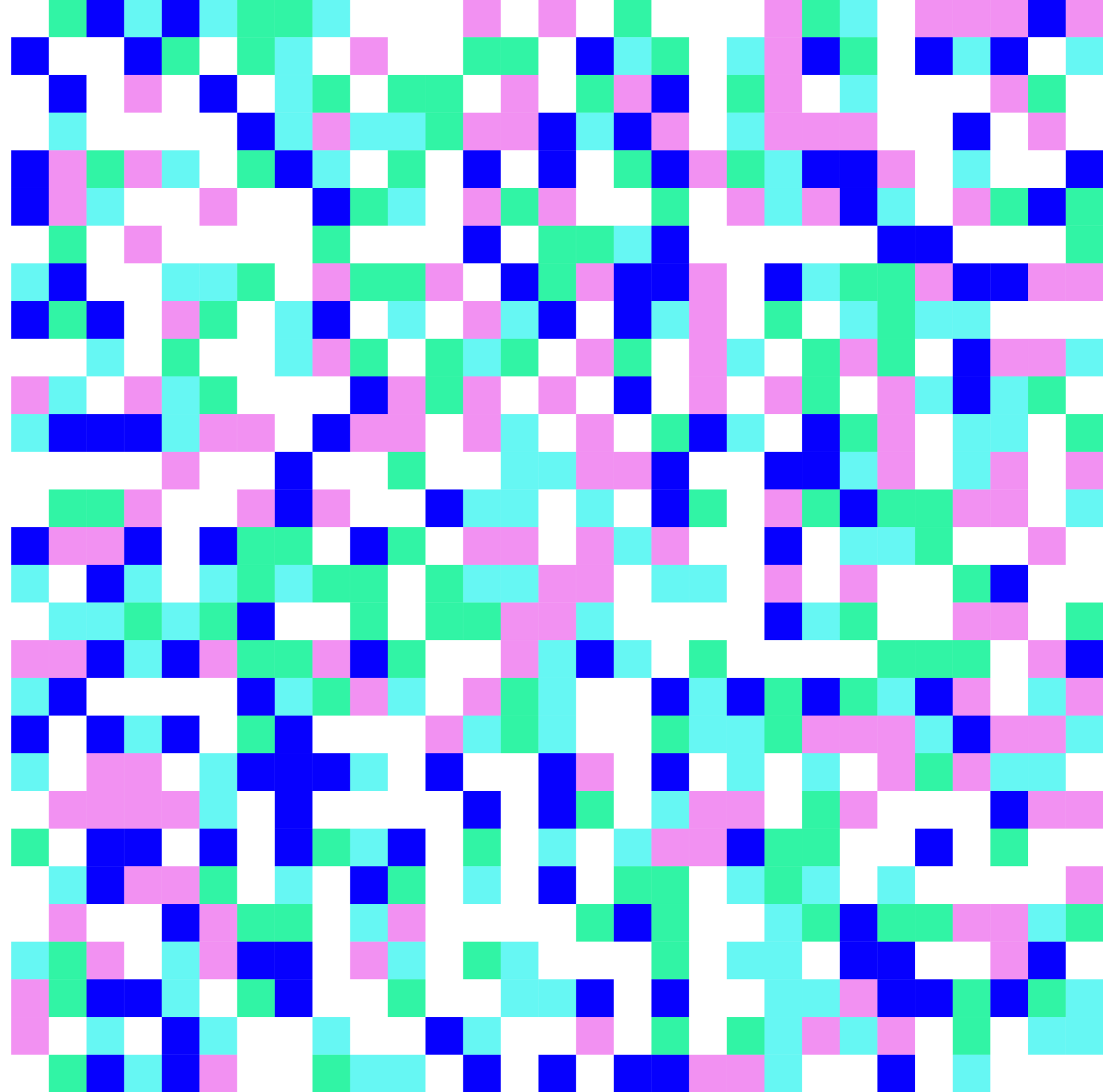
 **coding lab**

Lezione 1

**Loop, griglia
e random()**

Tutor
Luca Cattaneo
Devide Cudia

CODING LAB
Creative Coding
05/10/23



Canale YouTube

Guide e tutorial

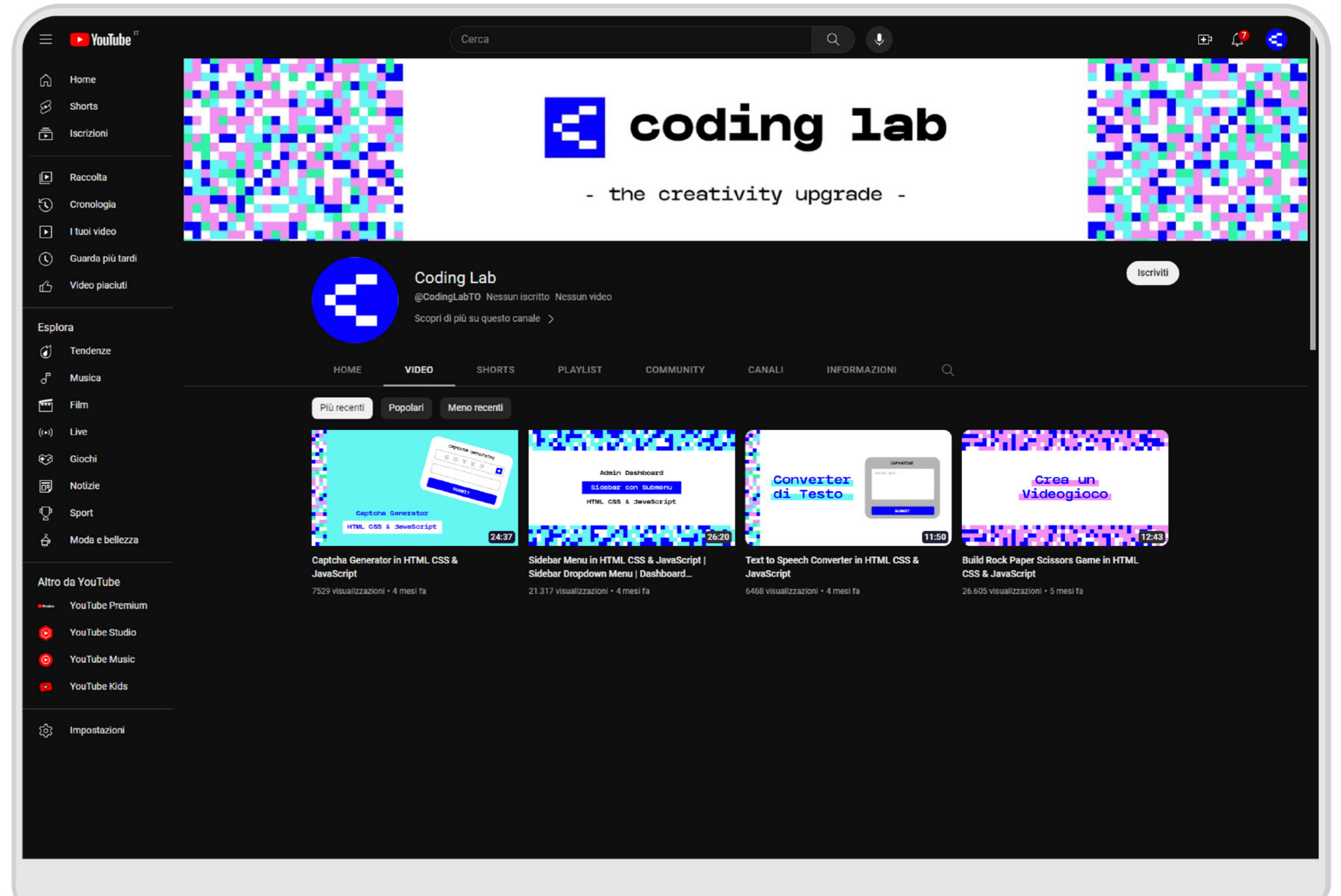
Il canale YouTube di Coding Lab svolge un ruolo importante all'interno del progetto.

Non è solo un luogo dove pubblicare tutorial, ma anche un pilastro fondamentale per promuovere l'intera esperienza di community.

Il canale YouTube è infatti un centro comune di formazione virtuale.

Qui verranno condivisi tutorial, guide e risorse di qualità per aiutare, corsisti e non, a sviluppare le loro abilità digitali e creative.

Ogni video dovrà essere pensato come educativo, informativo e facilmente accessibile.



Pagina istituzionale

Creare una pagina istituzionale su LinkedIn rappresenta un passo significativo della strategia, poiché consentirà di stabilire una presenza autorevole nel settore della formazione e di interagire con una audience di professionisti, studenti e appassionati.

AWARENESS

Su LinkedIn si potranno condividere informazioni rilevanti, articoli approfonditi e analisi sulle tendenze nel mondo del tech e del design. Questo contribuirà a far conoscere Coding Lab nel settore e aiuterà a costruire credibilità tra coloro che cercano opportunità di apprendimento di qualità.

VALORIZZARE L'APPRENDIMENTO

La pagina LinkedIn verrà utilizzata per condividere storie di successo degli studenti, evidenziando i risultati che hanno ottenuto grazie ai corsi e promuovendo il valore dell'apprendimento nel migliorare le competenze professionali.

CONNESSIONI

La rete di connessioni su LinkedIn è un tesoro di opportunità. Stabilire, ad esempio, relazioni con altre organizzazioni, istituzioni educative e professionisti del settore può consentire la nascita di partnership, eventi congiunti e progetti speciali che possono arricchire ulteriormente l'esperienza di Coding Lab.

The screenshot displays the LinkedIn profile of Coding Lab, a company based in Torino. The profile header includes the company logo, name, tagline 'The Creativity Upgrade', and location. Navigation tabs for 'Home', 'Chi siamo', 'Post', 'Lavoro', and 'Persone' are visible. A post from Coding Lab is featured, with the text: 'Il Creative Coding è un mondo affascinante, e noi di Coding Lab lo esploriamo insieme ai nostri studenti. La nostra libreria preferita è p5.js che offre infinite possibilità'. The post includes a graphic with the text 'coding lab IMPARA P5.JS Sblocca nuove possibilità col Creative Coding'. The right sidebar shows a list of companies followed by the user, including Pentagram, Graphic Days, and Figma. The bottom of the page contains navigation links such as 'Consiglia', 'Commenta', 'Diffondi il post', and 'Invia'.

05

CONCLUSIONI

Conclusioni

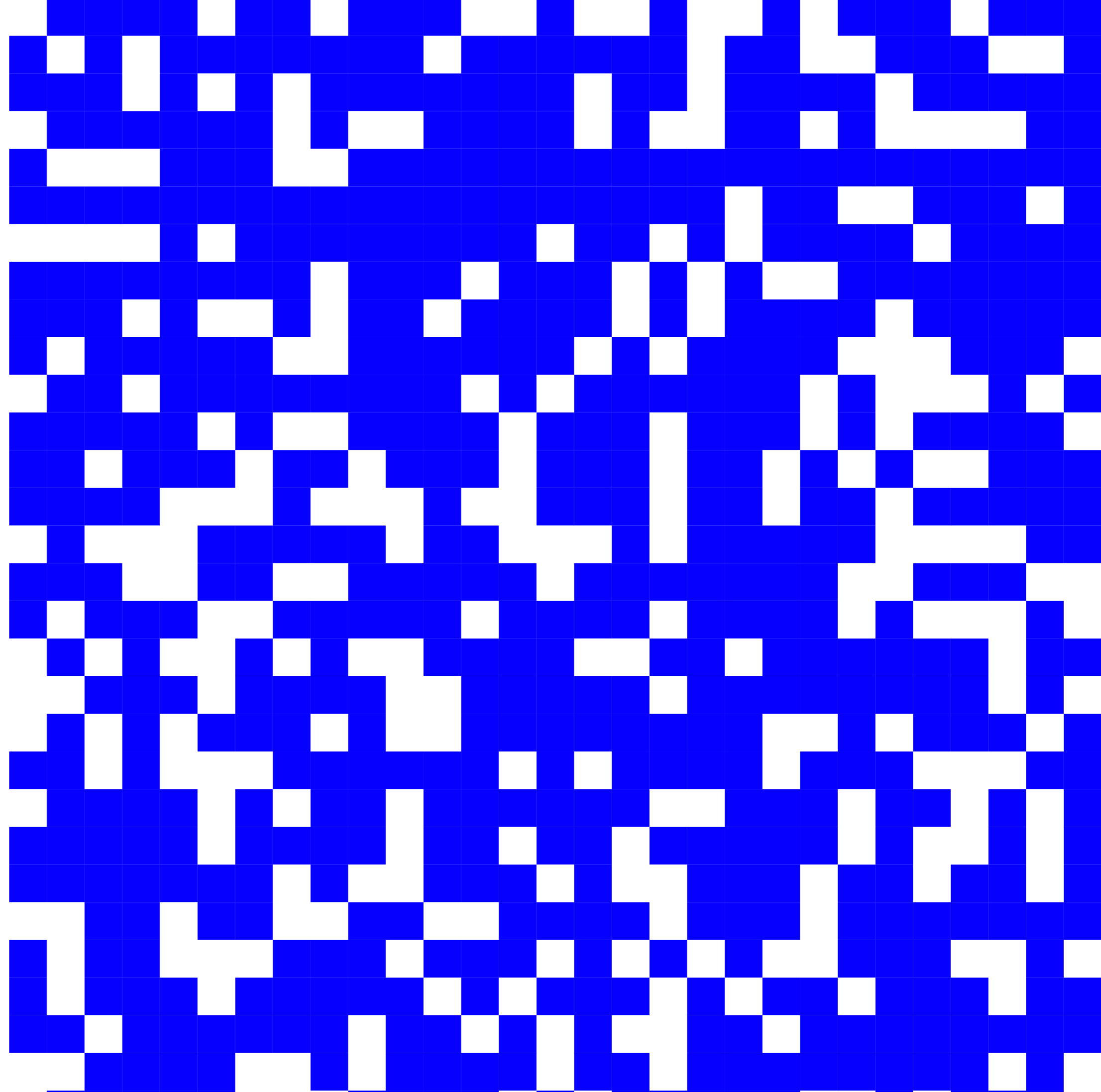
In conclusione, Coding Lab è un progetto che riconosce l'importanza delle competenze trasversali tra sviluppo e design.

L'approccio di Coding Lab, con la sua vision e la sua mission mirano ad allenare una mentalità flessibile e interdisciplinare negli utenti che prendono parte al progetto.

La strategia di comunicazione ha infatti lo scopo principale di creare un ambiente proattivo, stimolante e ricco di connessioni.

Questo progetto non vuole essere solo una risorsa per l'apprendimento, ma una comunità di appassionati, esperti e studenti che condividono un interesse comune.

Guardando al futuro, il progetto dovrà impegnarsi ad adattare l'offerta alle esigenze degli utenti e mantenersi in costante evoluzione, esattamente come l'ambito dinamico e innovativo nel quale si trova ad operare.



06

SITOGRAFIA

Sitografia

- 1 <https://medium.com/uxtales/quello-che-designer-e-developer-nondicono-197a66f967d9>
- 2 <https://www.xn--mtaweb-3ua.it/web-designer-o-web-developerdifferenze>
- 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-differencebetween-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-userinterface-design-29a916e5c2f5>
- 8 https://www.freedesktop.org/wiki/User_interface/
- 9 <https://youtu.be/U10y4X5Ni8Y>
- 10 <https://promosweb22.com/user-experience-i-6-principifondamentali-di-donald-norman/>
- 11 <https://www.techopedia.com/definition/4687/windows-icons-menusand-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-cosapensiero-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/creativecoding-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-historyevolution-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-diroblox/452252/>

42 [\https://www.economyup.it/innovazione/roblox-cose-e-comefunziona-la-piattaforma-che-da-accesso-al-metaverso/](https://www.economyup.it/innovazione/roblox-cose-e-comefunziona-la-piattaforma-che-da-accesso-al-metaverso/)

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

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

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

46 <https://www.techyon.it/articoli/front-end-developer-cosa-faquali-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-sicrea-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-designerdeveloper1910.html#close>

CASI STUDIO

52 <https://gumroad.com/>

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

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

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

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

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

58 <https://inconvergent.net/>

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

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

61 <https://triennale.org/en/magazine/unknown-play-pleasure-poemsramallo>

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

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

65 <https://unity.com/>

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

66 <https://www.epicgames.com/Fortnite/it/home>

67 <https://boolean.careers/>

68 <https://aulab.it/>

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

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

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

APPROFONDIMENTI VIDEO

Video 1 <https://youtu.be/6rxWc-TN1JI>

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>

07

RINGRAZIAMENTI

Riongraziamenti

Al mio Relatore,
desidero esprimere la mia sincera gratitudine per aver
accettato di guidarmi lungo il percorso di questa tesi.
I suoi consigli e rassicurazioni sono stati
fondamentali.
Grazie per aver condiviso la sua conoscenza e per
essere stato un mentore eccezionale.

Ad Ambra,
Il tuo amore, il tuo sostegno e la tua comprensione
durante questo periodo sono stati fondamentali.
Grazie per avermi sempre sostenuto, anche nei momenti
più impegnativi.
La tua presenza è stata una fonte di ispirazione
costante.

Ai miei genitori,
nessuna parola può esprimere adeguatamente la mia
gratitudine per tutto quello che avete fatto per me.
Il vostro amore, il vostro sacrificio e il vostro
incoraggiamento mi hanno portato fino a questo momento.
Siete la mia roccia, vi amo profondamente.

Grazie ancora a tutti per essere stati al mio fianco in
questo capitolo della mia vita.

Gabriele

