

BRICKS | TEMA

Numeri che suonano, forme che calcolano: un percorso tra arte, musica e coding con Ada Lovelace e Pitagora di Samo

a cura di:

Francesco Picca



Coding, Arte, Musica, Frattali, Fibonacci, AI

Questo testo è stato realizzato anche con l'ausilio di strumenti di intelligenza artificiale. L'autore ha orientato le richieste, selezionato e rielaborato i contenuti prodotti, verificandone accuratezza e coerenza, e ne è pertanto l'unico responsabile

Inizio del cammino

Mi chiamo Augusta Ada King¹, contessa di Lovelace.

Sono nata a Londra nel 1815, in un'epoca in cui le macchine cominciavano a moltiplicare la forza delle braccia, ma non ancora quella del pensiero. Mio padre scriveva versi; mia madre volle che io imparassi a governare i numeri. Tra poesia e calcolo ho trovato il mio linguaggio.

Ho lavorato sulle macchine analitiche di Charles Babbage², immaginando ciò che esse avrebbero potuto fare prima ancora che esistessero davvero. Scrissi istruzioni destinate a una macchina capace di operare non solo sui numeri, ma su qualunque entità potesse essere rappresentata simbolicamente. Oggi le chiamereste algoritmi.

La mia idea più radicale non fu tecnica, ma concettuale: se la musica obbedisce a rapporti numerici, se le forme seguono regole, allora una macchina può comporre, disegnare, trasformare. Non per ispirazione, ma per esecuzione. L'immaginazione, compresi, non è opposta alla logica: ne è una conseguenza disciplinata.

Non in un luogo fisico, ma in uno spazio dove le idee che hanno attraversato i secoli si incontrano. Dove numero, forma e ritmo non sono separati. Dove il codice non è ancora scritto, ma già presente come possibilità.

Camminavo lungo una galleria fatta di relazioni, in attesa di riconoscere chi, prima di me, aveva già ascoltato l'armonia nascosta delle cose.

Lo vidi e mi fermai per capire cosa stesse osservando.

Dove il codice diventa trama

Al centro della galleria, era in funzione un telaio Jacquard³. Le schede perforate scorrevano con regolarità, producendo un suono secco e ritmico. I fili si tendevano e si rilasciavano come corde musicali. Il tessuto che nasceva davanti a me mutava motivo a ogni passaggio, pur obbedendo a una legge costante.

¹ Augusta Ada King, contessa di Lovelace (1815-1852), fu la prima a comprendere che la Macchina Analitica di Babbage poteva fare molto più che calcoli numerici. Tradusse e commentò un articolo di Luigi Menabrea sul progetto della macchina, aggiungendo le sue note originali, in cui descrisse il primo algoritmo destinato a una macchina: il calcolo dei numeri di Bernoulli. Ada intuì che qualsiasi entità rappresentabile simbolicamente poteva essere manipolata da una macchina, anticipando concetti di programmazione generale, algoritmi e persino arte generativa. La sua visione collega logica, musica e pattern, fondando le basi del coding creativo

² Charles Babbage (1791-1871), matematico e ingegnere inglese, progettò la Macchina Analitica, considerata il primo concetto di computer programmabile. La macchina era pensata per eseguire calcoli complessi tramite ruote e ingranaggi, seguendo sequenze di istruzioni codificate su schede perforate, ispirate al telaio Jacquard. Babbage intuì che una macchina poteva essere generale, non limitata a un calcolo specifico, anticipando i principi fondamentali della programmazione e del software. Sebbene la macchina non fu mai completata, le sue idee costituirono la base teorica dell'informatica moderna

³ Il telaio Jacquard, introdotto all'inizio del XIX secolo da Joseph-Marie Jacquard, utilizza schede perforate per controllare automaticamente il disegno dei tessuti. Ogni scheda contiene una serie di istruzioni: il foro permette il passaggio di un filo, l'assenza del foro lo impedisce. Per la prima volta, un processo artigianale complesso viene governato da una sequenza discreta di comandi, separando la progettazione dell'opera dalla sua esecuzione materiale. Questo principio influenzerà direttamente la macchina analitica di Babbage e, più tardi, l'informatica moderna. Il telaio Jacquard dimostra che una regola può generare una forma, e che l'estetica può emergere dall'esecuzione automatica di un codice

Davanti al telaio c'era un uomo immobile.

Indossava una tunica semplice, priva di ornamenti. Il suo abbigliamento non apparteneva alla mia epoca, né a quella della macchina che avevo davanti. Era fermo, il capo leggermente inclinato, come se stesse ascoltando più che guardando. Non osservava il movimento dei fili, ma cercava la relazione che li governava.

Mi avvicinai piano, per non disturbare quel momento che, di certo, per l'uomo doveva essere magico.

«Osserva il disegno o il ritmo?» chiesi, per rompere il silenzio.

L'uomo non si voltò subito.

«Non tesse stoffa», disse. «Tesse rapporti.»

Sorrisi. Era una risposta che riconoscevo. «Esegue istruzioni», replicai.

Il telaio continuava a lavorare, indifferente alla nostra presenza. Ogni colpo era identico al precedente, eppure il motivo che lentamente prendeva forma non lo era mai del tutto.

L'uomo si voltò verso di me.

«Quando dimezzavo una corda, nasceva l'ottava. Quando ne prendevo i due terzi, la quinta. La bellezza non era nel suono, ma nel rapporto.»

«Qui accade lo stesso», risposi. «Ogni foro è una scelta, ogni scheda è una regola. La bellezza non è nel filo, ma nella sequenza.»

Osservò le schede perforate con attenzione, come se fossero numeri diventati materia. Ne sfiorò una con le dita.

Solo allora lo riconobbi.

Era Pitagora di Samo⁴, nato intorno al 570 avanti Cristo, su un'isola dell'Egeo.

Aveva attraversato il Mediterraneo in un'epoca in cui conoscere significava muoversi: dall'Asia Minore all'Egitto, forse fino a Babilonia, raccogliendo numeri, ritmi, geometrie.

Le sue scoperte non nacquero in un solo luogo, nacquero lungo il cammino.

Fu a Crotone, nella Magna Grecia, che diede forma compiuta al suo pensiero. Lì fondò una comunità in cui matematica, musica e osservazione della natura non erano discipline separate, ma linguaggi diversi per descrivere lo stesso ordine. Lì comprese che l'armonia musicale obbedisce a rapporti numerici semplici e che quelle stesse **relazioni potevano spiegare le forme, i movimenti, persino il cosmo.**

Di fronte al telaio, non vedeva una macchina moderna. Vedeva il compimento di un'intuizione antica: che il mondo, se ascoltato con attenzione, si comporta come una sequenza regolata.

L'armonia che aveva misurato tendendo corde, ora scorreva davanti a lui sotto forma di schede perforate.

«Dunque l'armonia può essere scritta», disse.

«E letta» aggiunsi. «E soprattutto... eseguita.»

⁴ A Pitagora è tradizionalmente attribuito il celebre **teorema sul triangolo rettangolo**, anche se era probabilmente noto in forme diverse già ad altre civiltà. Il suo contributo più originale fu però l'intuizione del legame profondo tra **numeri e musica**: studiando le corde vibranti, scoprì che gli intervalli musicali consonanti corrispondono a rapporti numerici semplici (2:1, 3:2, 4:3). Da qui nacque l'idea dell'**armonia delle sfere**, secondo cui anche i movimenti dei corpi celesti seguono proporzioni matematiche, producendo una musica cosmica invisibile. Pitagora rappresenta una figura simbolica del dialogo tra **matematica, arte, musica e filosofia**, anticipando l'idea che regole astratte e strutture numeriche possano spiegare e generare bellezza, ordine e conoscenza: un'intuizione che risuona ancora oggi nel coding, nella scienza dei dati e nell'arte generativa.

Un numero prende forma



Il tessuto cresceva davanti a noi: una figura prendeva forma.

«Una legge genera infinite forme», disse lentamente. «Lo avevo intuito nella musica.»

«Io l'ho immaginato per le macchine», risposi. «Se le relazioni sono esprimibili, allora sono programmabili.»

Per un istante tacemmo. Il telaio scandiva il tempo, come un metronomo meccanico.

«Allora la macchina non crea», osservò. «Obbedisce.»

«Sì», risposi. «Ma nell'obbedienza nasce qualcosa che nessuna singola mano potrebbe tessere.»

Annui e continuò, con un filo di voce: «Numero, ritmo, forma. Non sono mai stati separati.»

«No», affermai. «Ora semplicemente... scorrono.»

Il tessuto si staccò dal telaio. Non era un'opera finita, ma qualcosa di nuovo, pronto per essere osservato.

Ci fermammo a guardarlo.

Tra i motivi intrecciati, tra fili e schede perforate, riconoscemmo una figura precisa: era **il numero π , che spiccava nella geometria del tessuto.**

Pitagora lo indicò con un gesto. «Guarda,» disse, «la proporzione appare anche qui.»

Annuii, meravigliata.

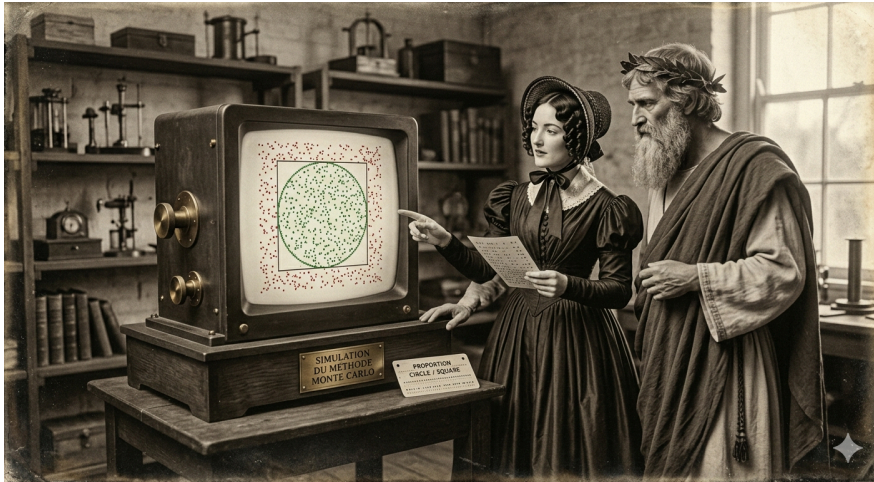
Quel piccolo tessuto, generato dal telaio, era stato ottenuto trasformando numeri e regole: un nuovo numero in un'immagine concreta.

Lasciammo il telaio alle nostre spalle e avanzammo lungo la galleria.

La luce cambiò: ora era fredda, metallica, riflessa su superfici lisce. Al centro di una pedana, uno schermo: schermi, luci che lampeggiavano come piccole stelle.

Mi avvicinai, incuriosita, il Maestro mi seguì.

Una pioggia di punti



Sullo schermo compariva una simulazione in tempo reale: migliaia di punti colorati che venivano lanciati dentro un quadrato, mentre quelli che cadevano all'interno di un cerchio venivano contati. Spiegai che la proporzione tra i due insieme approssimava il numero π ⁵.

Pitagora si fermò accanto a me, osservando con attenzione.

«Interessante,» disse, «un metodo basato sulla casualità, eppure calcola lo stesso numero che abbiamo visto nel tessuto.

Annuì.

«È come se il filo del telaio si fosse trasformato in punti digitali,» spiegai. «E questa macchina li esegue così rapidamente che possiamo **vedere l'approssimazione di π crescere davanti ai nostri occhi.**»

L'uomo osservava la simulazione. I punti si accumulavano, formando un cerchio perfetto all'interno del quadrato, e il numero π si avvicinava lentamente al valore corretto.⁶

«Regole semplici, risultati complessi,» mormorò Pitagora.

«Esatto,» risposi. «Proprio come nel telaio, solo che ora i fili sono dati, e la macchina li interpreta come istruzioni.»

Ci fermammo a osservare, affascinati: **la stessa armonia matematica di sempre**, ora incarnata in una danza digitale di punti e numeri.

«Maestro, mi segua,» dissi. Era assorto in cupi pensieri che costruivano idee e proporzioni. «Possiamo calcolare π con un semplice algoritmo: generiamo numeri casuali per le coordinate dei punti, verifichiamo se cadono all'interno del cerchio, e poi calcoliamo la proporzione. È un procedimento semplice, ma potente.»

Seguimmo un lungo corridoio e, giunti fuori, proiettata su uno schermo che levitava senza avere un punto d'appoggio, indicai la sequenza sullo schermo: ogni ciclo di calcoli è un passo dell'algoritmo, ogni risultato un piccolo tassello della conoscenza.

⁵ Il **metodo Montecarlo** stima π usando **punti casuali**. Si considera un quadrato con un cerchio inscritto: generando molti punti casuali nel quadrato, la proporzione di punti caduti nel cerchio rispetto al totale approssima il rapporto tra le aree. Moltiplicando per 4 si ottiene un'approssimazione di π

⁶ Una simulazione con Scratch è disponibile al link: [Pi-greco con Scratch](#); simulazione in python disponibile al link: [Pi-greco con Python](#)

Dalla regola al numero: armonie senza fine

«Eppure,» continuai, « π non è solo un numero. È infinito nelle sue cifre, e in quelle cifre possiamo trovare tutto: date di nascita⁷, poesie, citazioni intere. È un universo compresso in un simbolo. Ogni cifra contiene possibilità immense, esattamente come ogni punto di questa simulazione.»⁸

Pitagora mi osservava attentamente.

«Numero, ritmo, forma...» disse lentamente. «Ora vedo come le leggi che governano l'armonia musicale possano trasformarsi in regole eseguibili. Anche il caso, se ordinato, diventa struttura.»

«Proprio così», confermai. «La macchina obbedisce, ma nella sua obbedienza rivela bellezza. Come il telaio, come la musica, come l'universo stesso.»

«Maestro» ripresi, «c'è un altro modo di calcolare π , più iterativo, e molto simile a come funziona il telaio: ogni passo aggiunge dettagli alla figura.»

Per me fu un'emozione spiegare al maestro i dettagli della formula, nei suoi occhi si accese una nuova luce:

$$\pi = 4 \cdot \left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots \right)_9$$

Mentre avanzavamo lungo la galleria, il suono ci raggiunse prima dell'immagine.



Era una **melodia per pianoforte**¹⁰, lenta, regolare, ma difficile da prevedere. Non seguiva una struttura classica: nessuna ripetizione evidente, nessun tema che tornasse identico. Eppure non era caotica.

Aveva una coerenza interna, come se obbedisse a una regola silenziosa.

⁷ Cerca la tua data di nascita in π : <https://www.mypiday.com/>

⁸ La bellezza della Matematica: π (<https://www.youtube.com/watch?v=rjwOauMx4d4>)

⁹ Gottfried Wilhelm Leibniz (1646-1716), matematico e filosofo tedesco, scoprì una serie infinita per calcolare il numero π . Ogni termine della serie aggiunge precisione al calcolo. Questo metodo, iterativo e semplice da programmare, mostra come un numero irrazionale e infinito possa essere approssimato tramite ripetizioni regolari di un procedimento. È un esempio storico di algoritmo prima che esistessero i computer, anticipando concetti di programmazione e calcolo numerico; una simulazione è disponibile al link: [sviluppo di Leibniz](#)

¹⁰ David MacDonald – [Pi in La minore](#). La composizione traduce le cifre del numero π in note musicali, associando ogni cifra da 0 a 9 a un grado della scala. La tonalità di La minore garantisce coerenza armonica, mentre la melodia segue rigorosamente l'ordine delle cifre di π , senza ripetizioni o interventi compositivi aggiuntivi. Il risultato è una musica potenzialmente infinita, in cui una regola matematica diventa direttamente struttura sonora

Pitagora si fermò di colpo.

«Non è improvvisazione» dissi. «E non è nemmeno una composizione tradizionale.»

Seguimmo il suono, poco distante un pianoforte, suonava, senza mani umane. Una sequenza scorreva lentamente ogni cifra una nota, ogni nota una cifra:



Il direttore sollevò lo sguardo, incuriosito.

«Ogni strumento entra al momento giusto,» continuai. «Nessuno suona isolato, eppure ognuno segue una linea propria.»

Indicai l'orchestra nel suo insieme.

Il direttore è come un **programma centrale**. Non produce il suono direttamente, ma **coordina**.
Dà il tempo, gestisce gli ingressi, sincronizza le parti.

Pitagora ascoltava in silenzio.

«Gli archi, i fiati, le percussioni» proseguì, «sono **procedure diverse**.

Ogni sezione ha il suo compito, le sue regole, il suo metodo di esecuzione.»

Feci un passo indietro, osservando l'insieme.

Quando suonano insieme, operano in parallelo. Come **thread** che condividono lo stesso tempo, lo stesso ritmo, ma svolgono funzioni differenti.»

Il maestro accennò un sorriso. «E se uno sbaglia?» chiese.

«L'intero sistema se ne accorge», risposi. «La musica, come il codice, è tollerante solo fino a un certo punto. La bellezza nasce dall'equilibrio tra libertà e regola.»

Il gesto del direttore scattò: l'orchestra prese vita, ogni parte esatta, coordinata, necessaria.

Non stavamo assistendo ad una semplice esecuzione.

Stavamo osservando un **programma in esecuzione**, scritto in suoni invece che in simboli.

Più linee musicali emersero insieme, intrecciandosi con precisione. Non era più una sola voce: erano molte, indipendenti eppure perfettamente coordinate.

Solo quando il brano giunse alla fine, mi voltai verso Pitagora e cominciai a parlare, cercando di tradurre in parole il legame tra musica di Bach che avevamo appena ascoltato ed il codice.

«Sa, maestro» dissi, «la musica di Bach non nasce dall'improvvisazione, ma da **regole precise**. Ogni composizione parte da un tema iniziale, che possiamo pensare come un input: da quel tema derivano trasformazioni rigorose che garantiscono un risultato coerente. Nelle fughe, per esempio, un tema viene eseguito, poi ripetuto trasposto, invertito, rallentato e sovrapposto a sé stesso. Ogni passaggio è calcolato, un vero **processo iterativo** che costruisce complessità a partire da regole semplici.»

Pitagora annuì, interessato, così continuai.

«Anche il contrappunto funziona come un **algoritmo**. Ogni voce musicale è indipendente, ma deve rispettare vincoli rigorosi: nessuna deviazione è casuale. Possiamo paragonarlo a **funzioni che operano in**

parallelo o a **thread che seguono regole comuni**¹¹ all'interno di un sistema deterministico. Bach non scrive ogni nota "a caso": scrive regole di combinazione, che determinano come le note interagiscono tra loro.»

Feci un passo indietro, indicando alcune partiture.

«E poi ci sono le simmetrie e le trasformazioni complesse: inversione, la melodia specchiata, retrogradazione, la melodia al contrario, e canoni matematici. Opere come *L'Arte della Fuga* o *L'Offerta Musicale* contengono strutture che oggi definiremmo **trasformazioni formali**, anticipando in musica concetti simili a quelli che ritroviamo nel **coding e negli algoritmi**.»

Pitagora rimase a guardare il leggio e poi il pianoforte.

«Numero, ritmo, regola... tutto si trasforma in suono,» disse infine, quasi tra sé e sé. «Come il codice.»

Lo guardai, per sostenere le sue ultime affermazioni.

«Esatto. La bellezza nasce quando le regole vengono rispettate e messe in esecuzione», conclusi.

Riprendemmo a camminare: sapevamo che la galleria aveva ancora molto da mostrare. Davanti a noi apparve una nuova sala.

L'armonia dei cerchi



Alle pareti erano appesi diversi quadri, ma il cerchio era la figura dominante: ogni cerchio sembrava muoversi dentro un altro, come se l'artista avesse applicato una regola invisibile, una successione precisa di proporzioni e rotazioni.

«Maestro» dissi, «qui π è silenzioso ma presente. Ogni cerchio obbedisce al rapporto circonferenza-diametro, ogni rotazione segue angoli precisi.

L'autore¹² lavora con forme che si generano in modo **iterativo**, come se ogni cerchio fosse calcolato passo dopo passo.»

Indicai i cerchi concentrici e le loro variazioni di colore.

«È esattamente la logica del **coding creativo**: stabilisci regole semplici e le applichi ripetutamente, creando **pattern complessi**. Il seme numerico può essere π o una proporzione, le regole sono i cicli del programma, e il risultato è un'opera unica e armonica.»

¹¹ In un programma, più attività possono procedere **insieme**: la **programmazione concorrente** le alterna rapidamente; la **programmazione parallela** le esegue contemporaneamente. I **thread** sono le singole attività, i **processi** i contenitori principali. È come un'orchestra: ogni musicista suona la propria parte, ma insieme creano armonia sotto la guida del direttore

¹² In *Cerchi in un cerchio* (1923), Kandinsky organizza cerchi concentrici secondo proporzioni precise. Il numero π guida silenziosamente le forme, rendendo visibile l'armonia geometrica. Ogni cerchio è un passo iterativo: regole semplici applicate ripetutamente creano pattern complessi, anticipando i concetti di **arte generativa e coding creativo**

Pitagora osservava attentamente.

«Allora anche la pittura può essere costruita come un algoritmo,» disse.

«Ogni iterazione aggiunge un dettaglio, ogni cerchio contribuisce alla totalità, proprio come ogni filo del telaio o ogni cifra di π in un calcolo iterativo.»

Davanti a noi, i cerchi continuavano a catturare lo sguardo del Maestro: semplici nelle singole parti, **completi e armoniosi nell'insieme**, un esempio visibile di come **matematica, arte e codice** possano incontrarsi.

Pitagora restava ad osservare, gli occhi fissi sui cerchi più piccoli. Sembrava cercare, all'interno di ciascuno, **cerchi ancora più piccoli**, come se la struttura potesse continuare all'infinito.

«Sembra che ogni cerchio contenga un mondo intero» disse, a bassa voce «e, dentro quel mondo, altri mondi ancora, uno dentro l'altro...».

Annuì, sorridendo, appena in tempo, prima di essere trascinati ancora una volta da una forza invisibile. Restammo al buio solo per alcuni secondi, poi all'improvviso una luce ci accecò e un nuovo mondo si manifestò ai nostri occhi.

Ordine nel caos



«Che splendide figure!», esclamò Pitagora, continuando a volteggiare e cercando ogni singolo particolare del mondo che appariva ai nostri occhi.

«Maestro, stiamo nel mondo dei frattali¹³. Forme che si ripetono a scale diverse, dove ogni dettaglio contiene un micro-mondo simile al tutto. La regola è semplice, l'applicazione ricorsiva crea complessità infinita.»

I suoi occhi scintillavano di curiosità.

«Allora anche l'arte può essere infinita» disse, «basta seguire la **regola giusta**, e l'infinito si manifesta nella forma.»

Davanti a noi, i cerchi di Kandinsky non erano più solo cerchi: erano l'eco di strutture **auto-simili**, pattern che si moltiplicano all'infinito, anticipando la matematica dei frattali e la bellezza dei sistemi iterativi e ricorsivi.

¹³ I **frattali** sono strutture geometriche generate da **processi iterativi o ricorsivi**, in cui una stessa regola viene applicata ripetutamente. Presentano **autosimilarità**, con parti che riproducono la forma del tutto a scale diverse, e descrivono strutture potenzialmente **infinite** in uno spazio finito. L'insieme di **Mandelbrot**, definito da Benoît Mandelbrot, è l'esempio più noto: nasce dall'iterazione di una semplice funzione matematica sul piano complesso e mostra una ricchezza di forme senza fine. In arte e coding creativo, i frattali rendono visibile il legame tra **algoritmo, numero e forma**. [Approfondimento video](#)

Il maestro distolse lo sguardo dalle pareti e disse: «Hai parlato di forme che si ripetono e di regole che si applicano più volte. Ma dimmi, Ada: cosa intendi esattamente per **ricorsione**?»

Sorrisi, cercando un esempio che potesse parlargli.

«La ricorsione» spiegai, «è quando una regola richiama **sé stessa**. Non procede in linea retta, ma torna indietro, ripetendo lo stesso gesto a una scala diversa.»

Indicai le figure più piccole in una delle pareti.

«Immagina un cerchio che genera un altro cerchio al suo interno, e poi un altro ancora, e così via. Ogni volta la regola è la stessa, cambia solo la dimensione. In coding, una funzione ricorsiva fa proprio questo: risolve un problema applicando lo stesso procedimento a una versione più semplice di quel problema.»

Pitagora annuì lentamente.

«Come una dimostrazione che contiene un'altra dimostrazione» disse, «fino a raggiungere il Teorema.»

«Esatto!» risposi.

«La ricorsione è ordine che si ripete. È così che nascono i frattali, ed è così che l'infinito può essere rappresentato in una forma finita.»

Tornammo a osservare le pareti, consapevoli che in quelle magiche figure non c'era solo colore e forma, ma una **regola che continuava a pensare a sé stessa**.

All'improvviso, la galleria scomparve. La luce cambiò, l'aria divenne aperta, viva.

Proporzioni armoniche



Ci trovammo all'esterno, circondati da conigli che si muovevano rapidi nell'erba, moltiplicandosi sotto i nostri occhi. Pitagora osservò la scena con stupore.

«Si riproducono... velocemente» disse. «Ma non in modo casuale.»

Sorrisi. «È l'ultimo passo del nostro cammino,» risposi. «La **successione di Fibonacci**.»¹⁴

¹⁴La **successione di Fibonacci**, introdotta da **Leonardo di Pisa (Fibonacci)** nel *Liber Abaci* (1202), nasce da un problema sulla crescita dei conigli e i rapporti tra termini consecutivi tendono al **numero aureo** $\phi \approx 1,618$, percepito come armonioso. In **natura** ricorre in spirali di conchiglie, petali e girasoli. In **arte e architettura**, guida proporzioni e composizioni: dall'**Uomo Vitruviano** di Leonardo al **Partenone**, fino a pittori come Botticelli e Mondrian. Regole matematiche semplici generano **armonia e bellezza**

Mi chinai e tracciai una sequenza sul terreno:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, ...

«Ogni numero nasce dalla somma dei due precedenti.

È una regola semplice, applicata iterativamente. Fibonacci la descrisse studiando proprio la crescita ideale di una popolazione di conigli.»

I conigli continuavano a muoversi, come se seguissero quella legge invisibile.

«Quando questa successione cresce,» continuai, «il rapporto tra due numeri consecutivi tende a un valore speciale: il **numero aureo**, indicato con ϕ .»

Pitagora sollevò lo sguardo. «Armonia» disse, lentamente.

«Esatto. Il numero aureo compare in natura: nelle spirali delle conchiglie, nella disposizione dei petali, nelle foglie lungo un ramo, ma anche nella forma delle galassie.

E compare anche nell'arte: nella composizione, nelle proporzioni, nell'equilibrio visivo.»

Indicai il paesaggio intorno a noi.

«È la stessa logica che abbiamo visto finora: una **regola semplice**, ripetuta nel tempo, genera forme complesse e armoniche. In coding la chiameremmo iterazione. In natura, crescita. In arte, bellezza.»

Pitagora osservava i conigli, il campo, le spirali invisibili che sembravano attraversare ogni cosa.

«Numero» disse, «che diventa forma. Forma che diventa vita. Ed è qui che matematica, arte e natura si incontrano definitivamente.»

«E ricorsivamente!!», aggiunsi.

Il laboratorio delle meraviglie



Pitagora rallentò il passo e si soffermò davanti a **pareti luminose**, coperte di immagini in movimento, forme mai viste prima, come se appartenessero a un tempo lontano o che forse deve ancora essere. Si avvicinò, attratto, desideroso di comprenderle, di entrarvi, di osservare da vicino quel linguaggio sconosciuto.

Mi fermai accanto a lui e scossi lievemente il capo. Gli spiegai che ci sarebbe stato un tempo futuro per varcare quella soglia: **la stanza dell'Intelligenza Artificiale**, il luogo in cui nuove idee prendono forma, dove nascono dipinti e musiche che ancora non esistono.

E aggiunsi che forse quella stanza non crea solo opere, ma anche incontri.

Come questo nostro: soltanto virtuale, mai avvenuto davvero, nato tra righe di testo generate in parte da un'intelligenza artificiale; suggerite, in silenzio, da una mente umana che ha indicato la direzione e guidato la creazione.

Un dialogo possibile, apparso per un istante oltre una soglia che, in fondo, non esiste.

Per il momento, era sufficiente sapere che quel linguaggio stava già parlando al mondo, in attesa di essere ascoltato.

Proseguimmo il nostro cammino e ci fermammo su una collina, il vento leggero tra i capelli, il paesaggio intorno pieno di forme, spirali e cerchi concentrici. Mi voltai verso di lui e sorrisi.

Il viaggio continua

Il nostro tempo era giunto alla fine, sapevamo di essere arrivati alla fine del viaggio, o forse era solo l'inizio. Ogni passo compiuto aveva aperto la porta al successivo, e ogni volta che quel percorso si fosse ripetuto avrebbe aggiunto **magia e stupore** a ciò che avevamo già scoperto: come un algoritmo che chiama sé stesso, ogni iterazione genera nuovi dettagli, nuovi orizzonti, nuove armonie, procedimenti di calcolo elementari, sequenza di istruzioni che applicate con ordine danno vita a risultati complessi e armonici, regole semplici, ripetute con cura, possono generare forme intricate e sorprendenti.

Il numero, la regola, il codice: tutto può produrre **musica, arte e forme naturali**¹⁵. Ogni spirale, ogni cerchio, ogni sequenza di Fibonacci che osserviamo non è mai isolata, ma parte di un disegno più grande, iterativo e infinito. Il viaggio continua, pronto a sorprendere ancora, perché la scoperta è infinita e le regole, quando si combinavano con la fantasia, diventavano pura armonia.

Il Maestro si allontanò ed io fui attraversata da una lieve malinconia: la consapevolezza che ogni fine è solo una pausa, e che molte idee, come quelle stanze non ancora visitate, avrebbero atteso un altro tempo per essere comprese.

*FINE*¹⁶

¹⁵ La **Nota G**, parte delle Note di Ada Lovelace sul **Motore Analitico di Charles Babbage** (1843), descrive il primo **algoritmo pubblicato** della storia: un procedimento dettagliato per calcolare i numeri di Bernoulli. Mostra come **istruzioni semplici e sequenziali**, applicate con ordine e iterazione, possano produrre risultati complessi. Ada anticipa così il concetto moderno di **programma per computer**, sottolineando che la macchina può elaborare **numeri e simboli**, aprendo la strada a applicazioni creative oltre l'aritmetica

¹⁶ Tutte le immagini sono state generate da Google Gemini

Algoritmi tra matematica e didattica: numeri e metodo

Premessa

Le avventure di Ada Lovelace e Pitagora di Samo nel mondo del coding, dei numeri, dell'arte e della musica ci hanno guidato attraverso un percorso fatto di scoperte iterative, pattern nascosti e regole universali. Dai cerchi concentrici di Kandinsky alle spirali di Fibonacci nella natura, dai frattali alle composizioni musicali basate su π , abbiamo visto come matematica, codice e creatività possano fondersi in un unico linguaggio.

Viene ora proposta un'attività di apprendimento che permette di mettere in pratica i concetti esplorati durante il viaggio, sperimentando programmazione, calcolo e arte generativa. Gli studenti saranno invitati a osservare, progettare e creare, trasformando numeri e regole in forme, suoni e pattern visivi, proprio come Ada e Pitagora hanno fatto nel loro mondo immaginario.

Obiettivi dell'attività (riferimento DigComp 3.0)

In particolare, l'attività mira a sviluppare competenze digitali riconducibili alle aree del **DigComp 3.0**, attraverso un approccio interdisciplinare e creativo.

- **Comprendere il concetto di pattern, regola e iterazione** come elementi fondamentali del **pensiero computazionale** (*DigComp 3.0 – Area 5: Problem solving*).
- **Riconoscere il ruolo dei numeri e delle proporzioni** (π , successione di Fibonacci, numero aureo) nella rappresentazione della realtà naturale e artistica, interpretando dati e strutture matematiche in contesti diversi (*Area 1: Ricerca, valutazione e gestione delle informazioni*).
- **Sperimentare la programmazione come strumento creativo**, capace di generare forme, immagini e suoni, producendo contenuti digitali originali (*Area 3: Creazione di contenuti*).
- **Applicare processi di modellizzazione e simulazione** per descrivere fenomeni matematici e artistici, utilizzando il digitale per esplorare e rappresentare sistemi complessi (*Area 5: Problem solving*).
- **Sviluppare capacità di osservazione, progettazione e problem solving**, trasformando un'idea astratta in un artefatto concreto attraverso processi iterativi di progettazione e verifica (*Aree 3 e 5*).
- **Favorire un approccio interdisciplinare**, integrando conoscenze di matematica, informatica, arte e musica e comunicando in modo efficace i processi e i risultati (*Area 2: Comunicazione e collaborazione*).
- **Stimolare una creatività consapevole**, fondata su regole, strutture e vincoli, e non sull'improvvisazione casuale, promuovendo un uso responsabile e riflessivo degli strumenti digitali (*Aree 3 e 4: Sicurezza, benessere e uso responsabile*).

Obiettivo formativo finale

L'obiettivo finale è rendere gli studenti consapevoli che **dietro ogni forma armonica esistono regole semplici**, e che il **coding**, in linea con il DigComp 3.0, può diventare un **linguaggio espressivo** capace di unire logica, creatività e immaginazione.

Competenze sviluppate

- **Pensiero computazionale:** riconoscere pattern, iterazioni, regole e ricorsione.
- **Creatività digitale:** progettare e realizzare contenuti originali visivi o sonori tramite codice.
- **Problem solving:** scomporre problemi complessi in passaggi logici e iterativi.
- **Collaborazione e comunicazione:** presentare e condividere i risultati, spiegando scelte progettuali e algoritmi.
- **Consapevolezza interdisciplinare:** comprendere le connessioni tra numeri, arte e musica.

Metodologia

- **Approccio graduale:** dalla comprensione concettuale alla sperimentazione pratica.
- **Didattica attiva:** laboratori hands-on, esperimenti di coding, produzione di arte generativa.
- **Narrazione e contesto:** utilizzo del viaggio di Ada Lovelace e Pitagora come filo conduttore per collegare concetti teorici e attività pratiche.
- **Interdisciplinarietà:** integrazione di matematica, informatica, arte e musica.

A chi si rivolge

L'Unità di Apprendimento è pensata per studenti di diverse fasce d'età e competenze, proponendo due percorsi differenziati:

Percorso A – Coding artistico base

- Destinatari: studenti di **scuola secondaria di primo grado** o **biennio della secondaria di secondo grado**.
- Obiettivo: avvicinare gli studenti al **coding creativo** utilizzando strumenti **block-based** come Scratch o Pictoblox.
- Contenuti: realizzazione di semplici animazioni, pattern grafici, sequenze e interazioni visive, ispirate ai concetti matematici e artistici esplorati nel viaggio di Ada e Pitagora.

Percorso B – Coding creativo avanzato

- Destinatari: studenti del **triennio della secondaria di secondo grado**, in particolare ad **indirizzo informatico**.
- Obiettivo: approfondire il **coding creativo** attraverso il linguaggio **Python**, sviluppando progetti più complessi.
- Contenuti: generazione di **frattali**, programmazione di sequenze musicali basate su π , implementazione della **successione di Fibonacci** e esplorazione di pattern matematici e artistici, integrando matematica, musica e arte digitale.

Tempi dei percorsi

Entrambi i percorsi (A e B) sono progettati per essere completati in circa **12 ore**, articolate in due momenti principali:

1. Prima parte – Teorica (circa 4 ore)

- Introduzione ai concetti di pattern, regole, iterazione e coding creativo.
- Presentazione del viaggio di Ada Lovelace e Pitagora di Samo come filo narrativo per spiegare matematica, arte e musica.
- Analisi dei numeri, proporzioni e sequenze (π , Fibonacci, frattali) e loro applicazioni artistiche e musicali.

2. Seconda parte – Applicativa (circa 8 ore)

- Percorso A: realizzazione di animazioni, pattern grafici e interazioni in Scratch o Pictoblox.
- Percorso B: sviluppo di progetti complessi in Python, come generazione di frattali, sequenze musicali basate su π e implementazione della successione di Fibonacci.
- Verifica e condivisione dei risultati con un approccio collaborativo e creativo.

Materiali e strumenti

- **Percorso A:** computer, Scratch o Pictoblox, proiettore per mostrare esempi, materiali di supporto visivi (immagini dei dipinti, schede riassuntive dei concetti matematici e musicali).
- **Percorso B:** computer con Python (installazione di librerie per grafica e musica, ad esempio turtle, matplotlib, pygame), materiali di supporto sui concetti matematici (frattali, Fibonacci, π), esempi di codice di riferimento.

Sviluppo delle attività

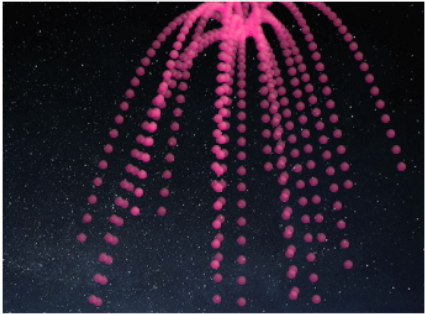
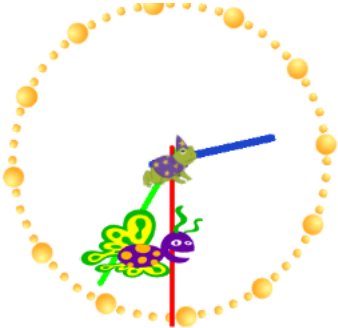
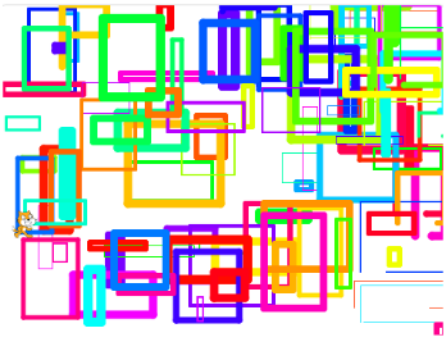
Lo sviluppo delle attività è stato pensato come un percorso graduale che accompagna gli studenti dalla comprensione concettuale alla sperimentazione pratica, mantenendo costante il legame tra matematica, coding, arte e musica.

La prima fase introduce i temi attraverso esempi narrativi e visivi – anche richiamando il viaggio di Ada Lovelace e Pitagora – per stimolare curiosità, immaginazione e consapevolezza delle regole che stanno dietro alle forme e ai suoni. La seconda fase è invece laboratoriale: gli studenti applicano quanto appreso realizzando semplici programmi e produzioni creative, trasformando numeri, algoritmi e iterazioni in immagini, animazioni e musica.

L'obiettivo non è solo "scrivere codice", ma usare il coding come linguaggio espressivo, sperimentando, osservando gli effetti delle regole e riflettendo sui risultati ottenuti.

Esempi di applicazioni e attività pratiche che gli studenti potrebbero realizzare includono:

Percorso A – Coding artistico (artefatti digitali realizzati con Scratch)

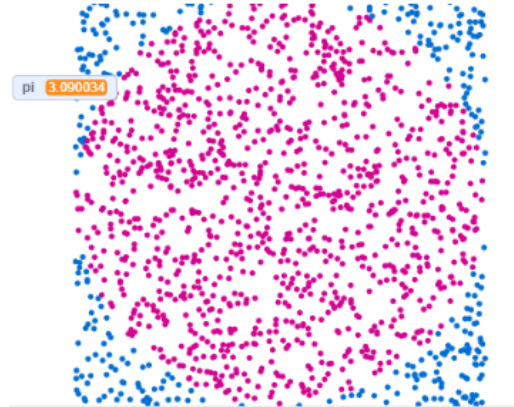
Animazione di fuochi d'artificio: utilizzo di sprite e movimenti casuali per creare sequenze visive dinamiche¹⁷	
Orologio digitale: programmazione di numeri, cicli e aggiornamento temporale in tempo reale¹⁸	
Opera in stile moderno: generazione casuale di rettangoli e quadrati sovrapposti, con variazioni di dimensione e colore, ispirata all'arte geometrica moderna¹⁹	

¹⁷ <https://scratch.mit.edu/projects/1147958268>

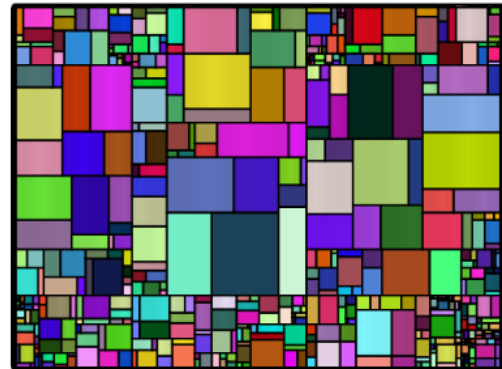
¹⁸ <https://scratch.mit.edu/projects/1094757379>

¹⁹ <https://scratch.mit.edu/projects/1094817296>

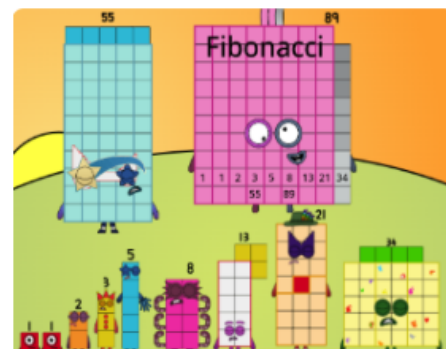
Simulazione del metodo Montecarlo:
utilizzo di punti casuali per stimare il valore
di π , collegando coding e matematica²⁰



Arte stile pittore olandese Piet Mondrian:
della comunità di Scratch una **ricca palette
di colori** e una **automazione iterativa
casuale**²¹



Animazione musicale collaborativa: della
comunità di Scratch un'animazione in
musica per la sequenza di Fibonacci²²



Albero iterativo: della comunità di Scratch
la simulazione per la crescita di un albero²³



²⁰ <https://scratch.mit.edu/projects/1147119203>

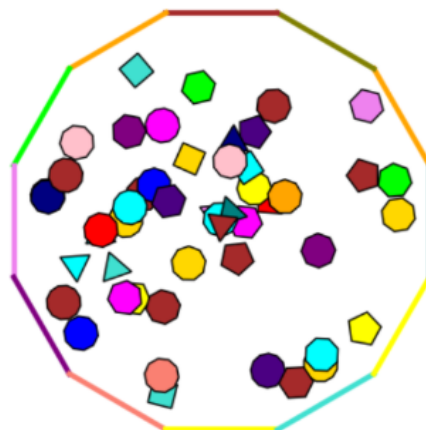
²¹ <https://scratch.mit.edu/projects/1259815919>

²² <https://scratch.mit.edu/projects/1101711805>

²³ <https://scratch.mit.edu/projects/1259810716>

Percorso B – Coding creativo avanzato (artefatti digitali realizzati con Python)

Figure in Figura: simulazione in python, figure regolari sono generate casualmente nelle forme e nei colori²⁴

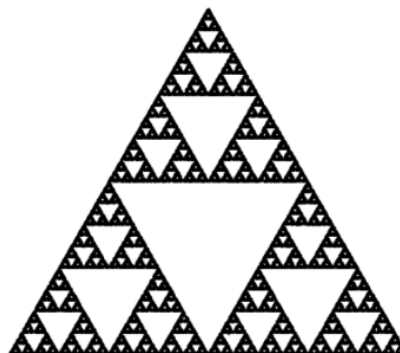


Disegni stile Mondrian: simulazione in python, rettangoli colorati generati casualmente; possibilità di variare i colori: `color = random.choice(["red", "blue", "yellow", "black"])`²⁵



Il triangolo di Sierpiński: simulazione in python, cambiando il numero di iterazioni si raggiunge un maggiore dettaglio: simulazione con Numero di iterazioni $N = 50000$ ²⁶

Triangolo di Sierpiński

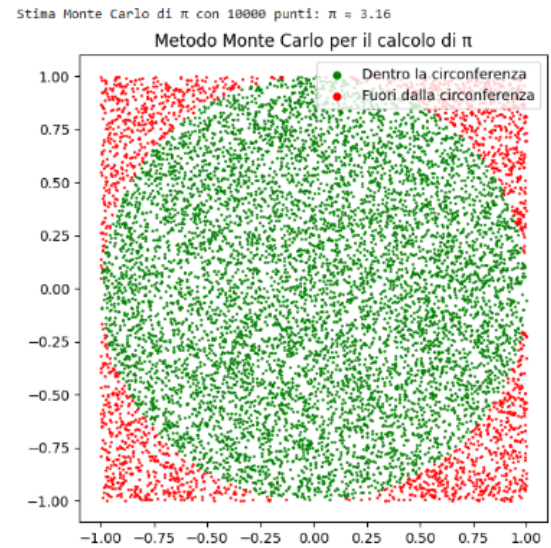


²⁴ [figure in figura](#)

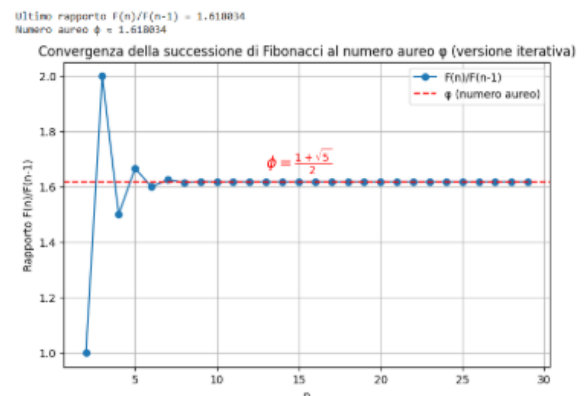
²⁵ [Disegni alla Mondrian.ipynb](#)

²⁶ [Il triangolo di Sierpiński.ipynb](#)

Simulazione del metodo Montecarlo: versione in python per stimare il valore di π ²⁷



Successione di Fibonacci: simulazione della convergenza della successione²⁸



Spirale Aurea – Arte, Natura e Fibonacci: la simulazione mostra la costruzione della spirale aurea²⁹

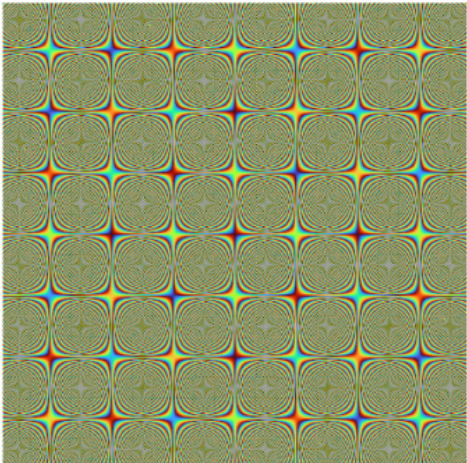
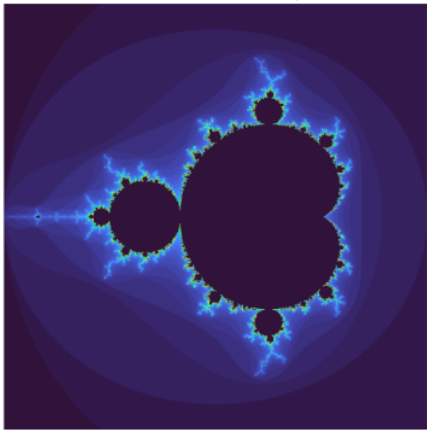
Spirale Aurea – Arte, Natura e Fibonacci



²⁷ [metodo Montecarlo.ipynb](#)

²⁸ [successione di Fibonacci.ipynb](#)

²⁹ [Spirale Aurea – Arte, Natura e Fibonacci.ipynb](#)

I suoni di Fibonacci: simulazione python, musica a 2 voci della successione:³⁰	Numeri di Fibonacci generati: [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597] Sequenza di cifre suonate: [1, 1, 2, 3, 5, 8, 1, 3, 2, 1, 3, 4, 5, 5, 8, 9, 1, 4, 4, 2, 3, 3, 7] 
Pixel Art Matematica Interattiva: l'immagine non viene disegnata, ma nasce da formule matematiche i cui parametri (tipo, intensità e colori) trasformano numeri in arte visiva³¹	Tipo: ☒ Griglia ☐ Radiale (Mandala) ☐ Onde (Op Art) ☐ Moiré Intensità: <input type="range"/> 15 Atmosfera: turbo Griglia Intensità = 15 
Insieme di Mandelbrot: simulazione in python con un menù a tendina che permette di cambiare la palette dei colori³²	Palette: turbo Insieme di Mandelbrot - cmap: turbo 

Verifica e valutazione

- **Osservazione diretta:** valutazione della partecipazione, della collaborazione e dell'autonomia nell'applicazione dei concetti.

³⁰  suona Fibonacci.ipynb

³¹  Pixel Art Matematica Interattiva.ipynb

³²  Insieme di Mandelbrot.ipynb

- **Prodotti concreti:** animazioni, pattern visivi, generazioni musicali, simulazioni matematiche.
- **Riflessione finale:** breve relazione o discussione su cosa è stato appreso e come il coding ha permesso di esplorare arte, musica e matematica.



Francesco Picca

f.picca@righitaranto.edu.it

I.I.S.S. "Augusto Righi" – Taranto

Francesco Picca, dopo la laurea in Scienze dell'Informazione conseguita nel 1993 presso l'Università degli Studi di Bari, ha insegnato Matematica e Matematica Applicata. Attualmente è un Docente a T.I. presso l'I.I.S.S. Augusto Righi di Taranto, dove insegna Scienze e Tecnologie Informatiche.

Si occupa, tra l'altro, di Formazione dei Docenti, collaborando con l'Ambito 21 di Taranto, in veste di Formatore dei Docenti Neo Immessi in Ruolo e dei Progetti PNRR Scuola Futura.

Dal 2015, collabora con la Casa Editrice Zanichelli, nella progettazione dei Corsi di Formazione per i Docenti e come Tutor.

Nel 2026 ha conseguito il Master di 2° livello in Applied Artificial Intelligence, Ingegneria dell'Innovazione presso l'Università del Salento.