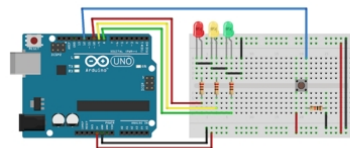
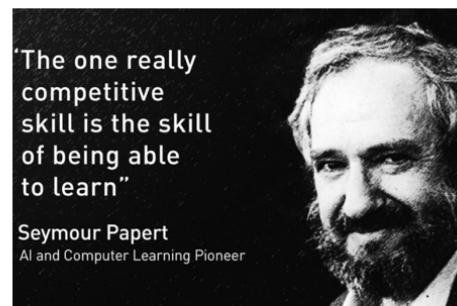


ESCURSIONI ovvero SPUNTI DI RIFLESSIONE VARI



**L'unica abilità
veramente
competitiva è quella
di saper imparare!**



INTRODUZIONE	4
LE “MACCHINETTE”	4
IL LINGUAGGIO LOGO	8
I CREATORI DI LOGO	9
I LINGUAGGI DI PROGRAMMAZIONE	12
LA SPERIMENTAZIONE CON LE SCUOLE	12
LA VERA DIFFUSIONE DI LOGO	12
VERSO I ROBOT PROGRAMMABILI	13
LA PROGRAMMAZIONE VISUALE	14
LA FIGURA CENTRALE: SEYMOUR PAPERT E IL LOGO	15
SEYMOUR PAPERT E LE CHILDREN MACHINES	16
LA VITA	17
ADDIO A SEYMOUR PAPERT, PAPÀ DEL LOGO E PIONIERE DELLA TECNOLOGIA NELL’EDUCAZIONE	17
LA VITA	17
ADDIO A SEYMOUR PAPERT, PAPÀ DEL LOGO E PIONIERE DELLA TECNOLOGIA NELL’EDUCAZIONE	17
CONSIDERAZIONI PERSONALI	21
IL LINGUAGGIO LOGO	22
GLI AMBIENTI MULTIMEDIALI	23
ARDUINO	24
IDE	26
ARTICOLO “MISCONCETTI & ARDUINO”	28
1. INTRODUZIONE	28
2 FASE INIZIALE	29
3 CONCEPTUAL CHANGE MODEL	29
4 ARDUINO E PROGETTAZIONE SKETCH	30
5 L’APPRENDIMENTO	30
6 CONCLUSIONE	31
7 BIBLIOGRAFIA	31
DARK	34

LA GENESI	34
IL CONCEPTUAL CHANGE	36
I FASE UNFREEZE	36
II FASE CHANGE	38
III FASE REFREEZE O RINFORZO	38

Escursioni

Introduzione

Ho voglia di fare un excursus su vari e diversi argomenti che mi hanno affascinato in tutti questi anni, sempre con il mio *modus operandi*: casuale e bizzarro! Sempre alla ricerca di un fil rouge che colleghi tutte le esperienze informatiche e relativi contenuti affrontati nel corso di mezzo secolo!

Pur essendo un laureato in CHIMICA PURA, indirizzo chimico inorganico, l'informatica ha sempre avuto un posto privilegiato nei miei studi. Ho sempre amato moltissimo la chimica e le scienze in generale, ma l'informatica è sempre stata al primo posto.....

La svolta avvenne al terzo anno d'Università (Padova) quando con i miei compagni d'appartamento, dopo varie e animate discussioni, decidemmo di inserire nel piano di studi un esame che si chiamava "*Elaborazione automatica dei dati*" presso la facoltà di Statistica. È stato l'esame che ci ha aperto le porte verso il mondo dell'informatica!!

Le "macchinette"

Facendo due passi indietro prima dei grandi computer e lo studio dell'informatica eravamo affascinati dalle prime "macchinette" arrivate dagli USA.....

L'inizio è stato nel 1972 (avevo 22 anni) con le prime macchine calcolatrici HP a notazione inversa polacca detta RPN. Questa macchinetta ha sostituito gli strumenti adoperati fino ad allora: il regolo calcolatore e/o le tavole logaritmiche.

Il sistema RPN all'inizio può sembrare ostico, ma una volta imparato semplifica di molto lo sviluppo delle operazioni, specialmente quelle con le parentesi. Successivamente comparvero le calcolatrici algebriche SHARP e CASIO con le quali si poteva immettere le espressioni algebriche usando la stessa notazione con la quale le risolvevamo con carta e penna, in pratica direttamente dalla carta alla calcolatrice.

Le calcolatrici hanno segnato il passaggio epocale dal regolo calcolatore ad uno strumento elettronico che dava risposte precise. Il regolo calcolatore era basato sui logaritmi e quindi bisognava leggere attentamente i risultati e c'era sempre un margine d'incertezza e errore.

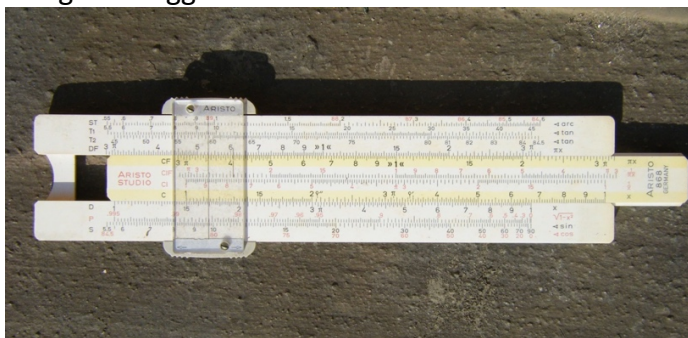


Figura 1 Il regolo calcolatore



Figura 2 prima calcolatrice RPN della HP

Attualmente tutte le calcolatrici si possono trovare in forma virtuale in tutti gli ambienti operativi iOS, Windows e in ambiente INTERNET.

In quegli anni per noi studenti era importante superare le difficoltà che si incontravano in matematica e quando apparvero queste macchinette ne fummo subito affascinati e ingolositi. Finalmente avevamo uno strumento (costoso al tempo) che poteva alleviarci nel calcolo e si poteva iniziare ad elaborare una primitiva programmazione.

Gli anni 70 possono essere considerati gli albori dell'informatica per tutti, ma l'impatto era sempre di natura prettamente matematica. Si aveva un linguaggio di programmazione semplice come il BASIC per creare quelle oggi chiamiamo semplici "app".

Agli inizi degli anni 80 riuscii a comprare un home computer il Texas Instruments TI-99/4A.

Texas Instruments TI-99/4A computer	
	
Tipo	Home computer
Paese d'origine	 Stati Uniti
Produttore	Texas Instruments Incorporated
Presentazione	giugno 1981
Fine vendita	ottobre 1983
CPU	Texas Instruments TMS9900
Frequenza	3,0 MHz
ROM	26 KB
RAM di serie	16 KB
RAM massima	52 KB
SO di serie	TI BASIC

Questa è la scheda reperibile su Wikipedia, ma quello che più mi ha colpito è che nel 2023 questo computer ha ancora un mercato(retro) dove si può acquistarlo per circa 300€. Macchina meravigliosa e primo amore, dove mi sono esercitato a programmare con il linguaggio TI BASIC. Di seguito riporto i titoli di alcuni programmi redatti dal manuale d'istruzione fornito dalla Texas e personali:

1. INTERESSE SEMPLICE
2. TROVARE IL PIU' PICCOLO NUMERO UGUALE ALLA SOMMA DEI CUBI DELLE SUE CIFRE
3. CONTO DA 1 A 10
4. CONTO ALLA ROVESCIA
5. STAMPA NOME ROVESCiato
6. ACCOPPAMENTO LISTE
7. STAMPA TABELLA VALORI E TABELLA VALORI TRIPLI
8. DA BASE N A BASE 10
9. TEOREMA DI PITAGORA
10. MCD DI TRE NUMERI
11. CALCOLARE LA MEDIA ARITMETICA E LA MEDIANA DI N NUMERI
12. ERRORE ASSOLUTO E RELATIVO

Lo strumento di Input/Output collegato con il computer era un semplice e comune registratore a cassette audio dove venivano registrati o acquisiti i programmi da noi elaborati. Queste cassette erano la nostra memoria (oggi sarebbe un Hard Disk) dove si poteva registrare o leggere i programmi.

In quel periodo comparve anche un'altra macchina leggendaria come il Commodore 64, anche questo si apriva in ambiente BASIC e permetteva la realizzazione di programmi a vocazione matematica. L'unità di I/O e la possibilità di stampare fecero fare a questo sistema un balzo in avanti rispetto al TEXAS TI-99/4:

1. Avere una unità esterna Disk drive che permetteva la scrittura e lettura dei programmi su floppy-disk da 5" ¼ (cinque pollici e un quarto).
2. Inoltre c'era la possibilità di avere una stampantina ad aghi con la quale stampare i listati dei programmi e altro.



Con questi home computer abbiamo imparato i fondamentali della programmazione reperibili al seguente indirizzo: <https://www.adalovelace.it/due-domande-cosa-servono-come-sono-fatti/>
 Un documento tratto dal mio libro "Fil rouge".

Queste conoscenze e competenze mi hanno permesso di tenere **n** corsi di formazione per Insegnanti, studenti e persone, al di fuori dell'ambiente scolastico, interessate all'argomento. Dopo tanti anni posso esprimere un bilancio e le sensazioni personali. I meno interessati sicuramente i docenti, mentre gli studenti e persone esterne hanno avuto migliori performance.

Inizialmente tutti dimostravano un interesse e una voglia di conoscere questo nuovo mondo, ma appena la matematica e la programmazione si presentavano ecco scomparire l'interesse. L'informatica si presenta con una faccia sorridente e compiacente, ma sotto c'è un'anima complessa e complicata che affascina e coinvolge i giovani come una sfida e le persone interessate, ma non i docenti che vedono fatica in più e suscita loro una paura di non padroneggiarla facilmente e in breve tempo. Lasciatemi questa piccola cattiveria: quelli più riottosi sono stati proprio i docenti di matematica. La paura era quella di non poter sostenere la loro immagine verso gli studenti. Sono considerazioni amare, ma reali e comprovate.

Ho anche gestito corsi TIC A TIC B. Sono le **Tecnologie dell'Informazione e della Comunicazione** (in inglese ICT), cioè tutti quei processi e strumenti tecnologici che servono a produrre e migliorare le conoscenze e gli strumenti di apprendimento. Erano gestiti dal Ministero della Pubblica Istruzione, ma anche questi corsi per quanto ben pensati e strutturati non hanno portato alla diffusione dell'informatica nella scuola. Ho cercato di pensarla anche come uno strumento trasversale alle discipline, ma solo pochi hanno avuto il coraggio di impegnarsi e cercare la potenzialità derivante da questa nuova disciplina.

Le tre seguenti slide aiutano a definire le TIC e loro importanza.

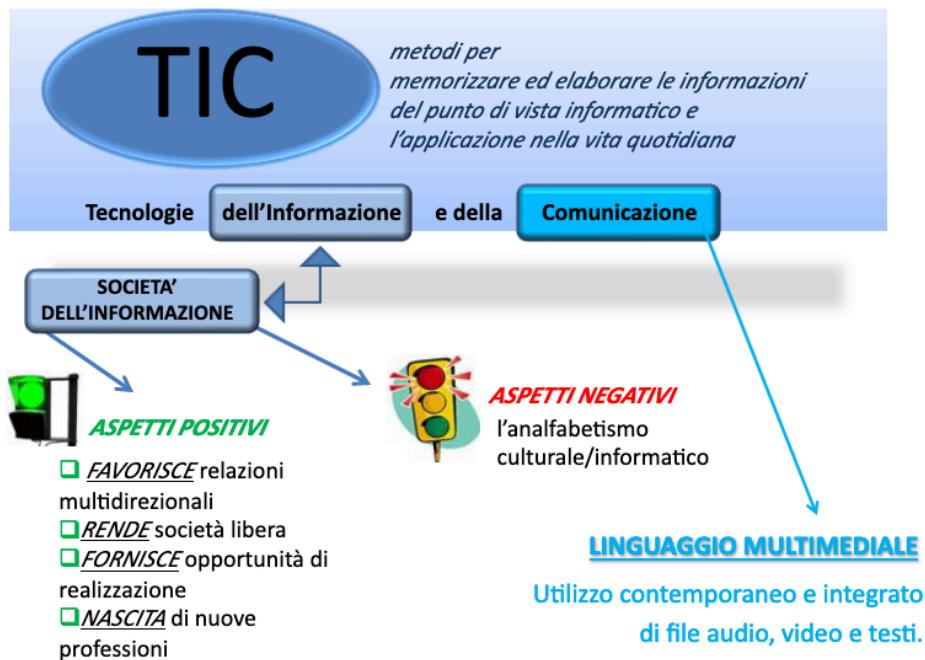


Figura 3 Slide TIC n°1

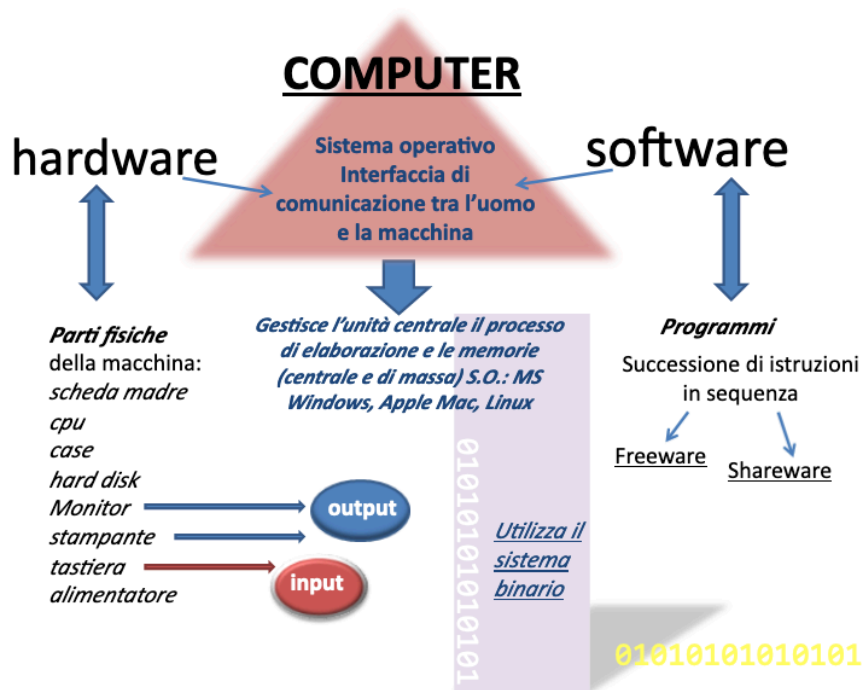


Figura 4 Slide TIC n° 2

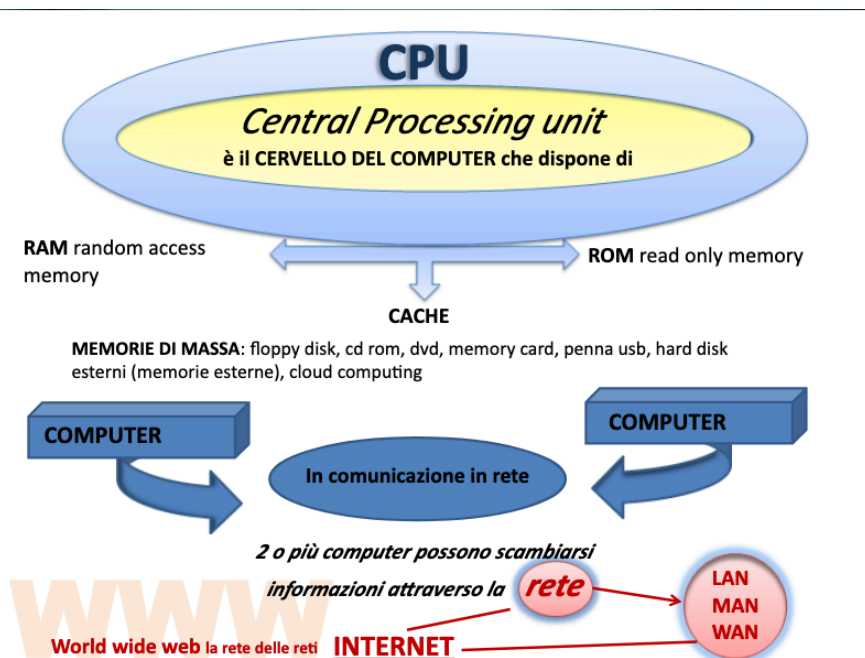


Figura 5 Slide TIC n° 3

Il linguaggio LOGO

Un ricordo particolare è legato alla tartaruga del linguaggio LOGO con la quale ho affascinato tanti ragazzi avvicinandoli alla programmazione.

LOGO, è un linguaggio di programmazione a scopo didattico, il primo pensato eminentemente per scopi educativi e orientato all'infanzia.

Creatore: [Wally Feurzeig](#), [Seymour Papert](#), [Cynthia Solomon](#)

Data di origine: 1967

Papert pensò alla tartaruga come a “un oggetto con il quale pensare” semplici comandi e una grafica interattiva sono fondamentali per invogliare a provare e a pensare: il “**far divertire a pensare**” è un obiettivo fondamentale della scuola

Prima del computer e dopo

Prima del computer è utile imparare a dare i comandi, nel modo e nella sequenza giusta, per es. ad un amico. Dopo aver introdotto i primi strumenti di base del Logo, è sicuramente utile analizzare qualche problema usando i diagrammi di flusso. Essi costringono a valutare la logica intrinseca di un problema, da un punto di vista fortemente operativo e costringono a valutare la differenza fra dati, variabili, operazioni, alternative.

Informatica matematica e lingua

La precisione ed il rigore necessari per comunicare con il computer inducono i ragazzi ad un uso più preciso e puntuale della lingua:

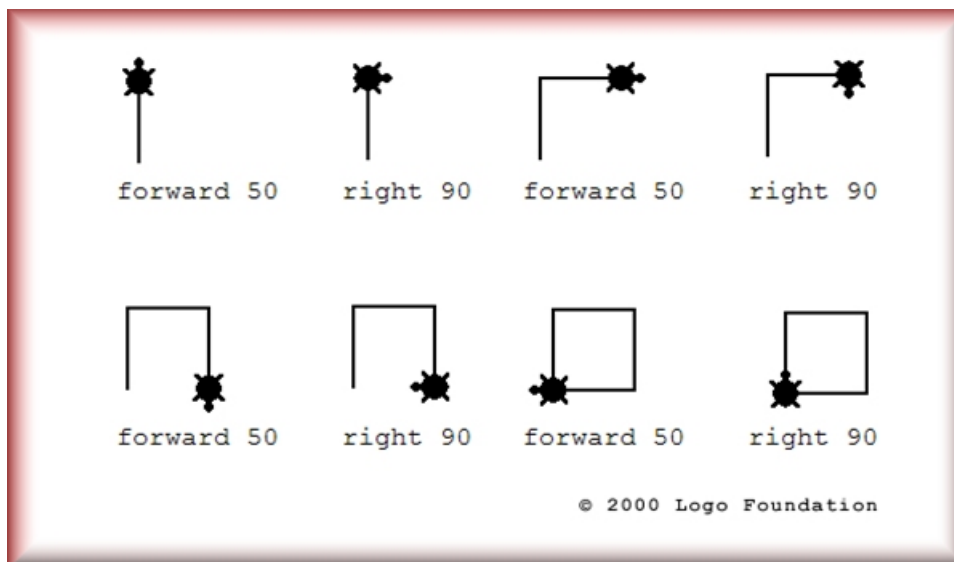


Figura 6 I comandi LOGO per costruire un quadrato

Oggi, nelle scuole italiane, molti alunni utilizzano **Scratch** per esprimere la propria creatività costruendo storie o piccoli videogiochi.

Sono sempre più numerose anche le scuole che propongono la **robotica educativa** come **attività interdisciplinare** per trasmettere conoscenze e per sviluppare abilità e competenze. L'obiettivo è quello di formare ciò che chiamiamo **il pensiero computazionale**: *in molte scuole la robotica educativa viene proposta come attività interdisciplinare per trasmettere conoscenze e sviluppare abilità e competenze. In entrambi i casi tra gli obiettivi vi è quello di formare il cosiddetto pensiero computazionale.*

Uno dei fattori chiave della diffusione di questi strumenti è la facilità con cui avviene il **processo di apprendimento**.

Un risultato al quale si è arrivati dopo decenni di studi e che ha un punto di inizio preciso: la creazione del **linguaggio di programmazione Logo**.

I creatori di Logo

Secondo il documento [LOGO – A Project History](#), nel 1966 gli scienziati [Seymour Papert](#), [Wallace Feurzeig](#), e [Daniel Bobrow](#) si incontrarono per discutere sulle modalità con cui **insegnare ai bambini a usare il computer**.

Feurzeig guidava allora il gruppo di lavoro dedicato all'educazione della [Bolt Beranek and Newman Inc.](#) (BNN), una società tecnologica di ricerca e sviluppo con sede a Cambridge, negli Stati Uniti d'America.

La BNN ebbe un ruolo importante anche nello sviluppo della rete Internet. L'atmosfera pionieristica di quell'epoca è ben descritta in [A Culture of Innovation, Insider Accounts of Computing and Life at BBN](#).

Dalle idee di Papert, Feurzeig e Bodrow nacque Logo, che rappresentava un metodo completamente nuovo per supportare l'**apprendimento del pensiero formale** e sviluppare capacità di **problem solving**.

Un'altra figura molto importante per lo sviluppo di questo linguaggio fu la ricercatrice [Cynthia Solomon](#), all'epoca collaboratrice di BBN e che anni dopo curerà la versione di Logo per i computer della Apple.

Presentato nel 1967, **Logo** era influenzato da tre aree di ricerca:

- intelligenza artificiale;
- psicologia dello sviluppo;
- logica matematica.

Gli informatici coinvolti nello sviluppo di Logo erano esperti di intelligenza artificiale, molto in voga in quegli anni (la celebre [Conferenza di Dartmouth](#) si era tenuta da poco).

Non deve stupire quindi che Logo nacque come “dialetto” del [LISP](#), un linguaggio di programmazione per l'intelligenza artificiale ideato nel 1958 da [John McCarthy](#).

Inserisco come immagine il frontespizio del lavoro di Seymour Papert del 1971.

Notare la data che è luglio 1971, l'intestazione relativa al MIT, il riferimento è nell'ambito della AI (Artificial Intelligence) e traduciamo il titolo:

TEACHING CHILDREN TO BE MATHEMATICIANS

VS.

TEACHING ABOUT MATHETICS

insegnare ai bambini a essere matematici

contro

insegnamento della matematica

Questo report introduce alla ricerca fatta per progettare il LOGO che tanta parte ha avuto nell'introdurre i principali concetti di informatica ai bambini.

È un momento storico!!!

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
A.I. LABORATORY

July 1971

Artificial Intelligence
Memo No. 249

LOGO
Memo No. 4

TEACHING CHILDREN TO BE MATHEMATICIANS
VS.
TEACHING ABOUT MATHEMATICS¹

Seymour Papert

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's education research is provided in part by the National Science Foundation under grant GJ-1049.

¹To be published in the International Journal of Mathematical Education in Science and Technology (New York: John Wiley & Sons, 1972) and in Proceedings of 1970 CEMREL Conference on Algebra (Carbondale, Illinois: CEMREL, Spring, 1971).

I linguaggi di programmazione

Un linguaggio di programmazione è un [linguaggio formale](#) che consente di realizzare **algoritmi**. Ma non viene “compreso” direttamente dal computer.

Le istruzioni di qualsiasi linguaggio di programmazione, infatti, vengono tradotte in **linguaggio macchina**, che è l'unico linguaggio direttamente eseguibile da un computer.

Già nel 1964 era stato introdotto un linguaggio di programmazione per principianti denominato BASIC, molto facile da imparare, ma poco “strutturato”.

Nel linguaggio Logo, invece, una delle idee di fondo è che il codice viene suddiviso in **procedure**.

Logo doveva rappresentare uno **strumento di apprendimento con “pavimento basso e nessun soffitto”**, adatto ai neofiti, ma in grado di appassionare gli utilizzatori esperti.

La sperimentazione con le scuole

Nel biennio 1968-69 Logo venne sperimentato in alcune scuole statunitensi. Nonostante i limiti della tecnologia dell'epoca, l'esperienza ebbe successo e si notarono miglioramenti in abilità non matematiche come la lettura.

I ricercatori si stupirono del fatto che bambini tra i 7 e 9 anni al termine della sperimentazione mostrassero uno sviluppo nel pensiero formale e matematico.

Successivamente, lo sviluppo di Logo venne influenzato dal fatto che non tutti gli ideatori di questo linguaggio avevano i medesimi obiettivi.

Da una parte c'era Seymour Papert che voleva incidere profondamente sul sistema scolastico, dall'altra Wallace Feurzeig che ne vedeva le potenzialità commerciali.

Intorno al 1972 BBN presentò **Irving**, la prima tartaruga robot. Mentre Papert spostò lo sviluppo del “suo” Logo al [Massachusetts Institute of Technology](#) (MIT).

Contemporaneamente Papert creò un gruppo di ricerca su al MIT, dove continuò le sue sperimentazioni didattiche osservando come i bambini usavano questo linguaggio.

L'informatico [Hal Abelson](#) trovò la maniera di visualizzare un vettore grafico sullo schermo. Ora i bambini potevano disegnare con il computer!

Purtroppo per gran parte degli anni '70 non era facile accedere a un computer. Questo fatto frenò la diffusione di Logo nelle scuole, almeno per i primi anni.

La vera diffusione di Logo

Dalla fine degli anni '70 il linguaggio Logo venne reso disponibile per gli **home computer** prodotti da Texas Instruments, Apple, Atari e Commodore.

Giuseppe Laricca, ricercatore del CNR tra i primi in Italia a interessarsi di informatica cognitiva, curò la traduzione in italiano della versione per Commodore, presentata nel 1985.

Nel 1981 Seymour Papert fondò la società Logo Computer Systems Inc. (LCSI), con l'obiettivo di creare software per aiutare i bambini a sviluppare la capacità di risolvere problemi e il pensiero critico.

Negli anni '80 Logo venne quindi utilizzato in decine di migliaia di classi negli Stati Uniti e in tutto il mondo.

Il successo del libro *Mindstorms* di Papert incentivò migliaia di docenti a lasciare che i propri alunni sperimentassero l'uso del computer. Nel Regno Unito Logo entrò nei curricula scolastici nazionali.

Concetti complicati come feedback, ricorsione e astrazione potevano essere acquisiti in maniera naturale.

Questo successo ebbe però anche **implicazioni negative**:

- si crearono troppe aspettative sulla facilità di apprendimento della matematica;
- venne data troppa enfasi all'aspetto grafico di Logo (la "geometria della tartaruga");
- c'erano troppi versioni di Logo che differivano tra di loro, impedendo la formazione di un'unica grande comunità.

Inoltre, non ancora stato sviluppato un sistema di valutazione per questo tipo di attività. Ciò rappresentava un problema per gli insegnanti.

Verso i robot programmabili

Secondo Papert insegnare ai bambini la matematica era differente dall'insegnare a essere matematici (si veda ad esempio [questo suo lavoro](#)).

Pertanto, con il suo gruppo di lavoro si mosse in molte direzioni. Una di queste fu la **robotica educativa**.

Grazie al lavoro dei ricercatori del MIT [Mitchel Resnick](#) and Steve Ock, nella seconda metà degli anni '80 una speciale versione di Logo riusciva ad interfacciarsi con i mattoncini della LEGO dotati di motori, luci sensori ecc. (**LEGO/Logo**).

Sempre dalla collaborazione tra il MIT e la LEGO nacque il [Programmable Brick](#) (il mattoncino "programmabile"), in cui la capacità di calcolo era inserita direttamente all'interno di un apposito mattoncino.

Ora i bambini potevano costruire con i mattoncini e programmare quello che avevano costruito.

L'evoluzione del Programmable Brick rappresenterà il "cuore" della serie robotica Mindstorms della LEGO (lanciata sul mercato nel 1998).

La programmazione visuale

Nel 1993 la LCS di Papert presentò la prima versione di [MicroWorlds](#), un programma che utilizzava il linguaggio Logo per insegnare lingua, matematica, robotica attraverso il coding. In Italia ebbe una certa diffusione con il nome di “MicroMondi”.

Ma nonostante l’evoluzione dei computer e l’introduzione delle interfacce grafiche, la programmazione di Logo avveniva sempre in maniera **testuale**, cioè scrivendo le istruzioni un carattere alla volta.

Serviva un’idea per semplificare il passaggio tra l’ideazione di un algoritmo e la sua **implementazione** in Logo.

Intorno al 1995 venne sviluppata la prima versione di [LogoBlocks](#), un Logo visuale per il Programmable Brick.

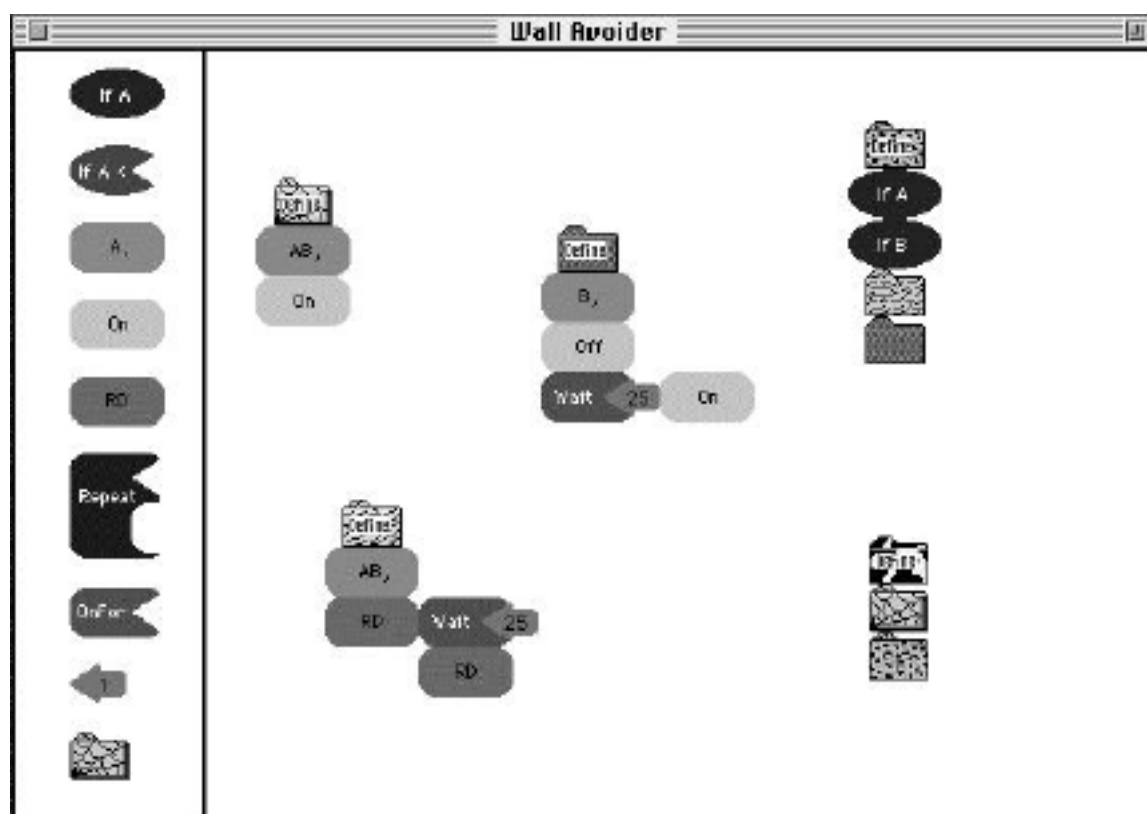


Figura 8 Schermata di LogoBlocks (Andrew Biegel) molto vecchia

Per **programmazione visuale** si intende l’uso di simboli grafici nel processo di programmazione.

In questo senso, le istruzioni possono avvenire sotto forma di blocchi e l’interfaccia grafica dell’ambiente di programmazione può essere costruita del tutto o in parte da icone.

Dall’idea di Logo Blocks si arrivò, nel 2004* alla prima versione di **Scratch**, un ambiente di programmazione che utilizza il paradigma di programmazione a blocchi originariamente implementato come LogoBlocks.

* Lo Scratch Team indica il 2007 come anno di “nascita” di Scratch.

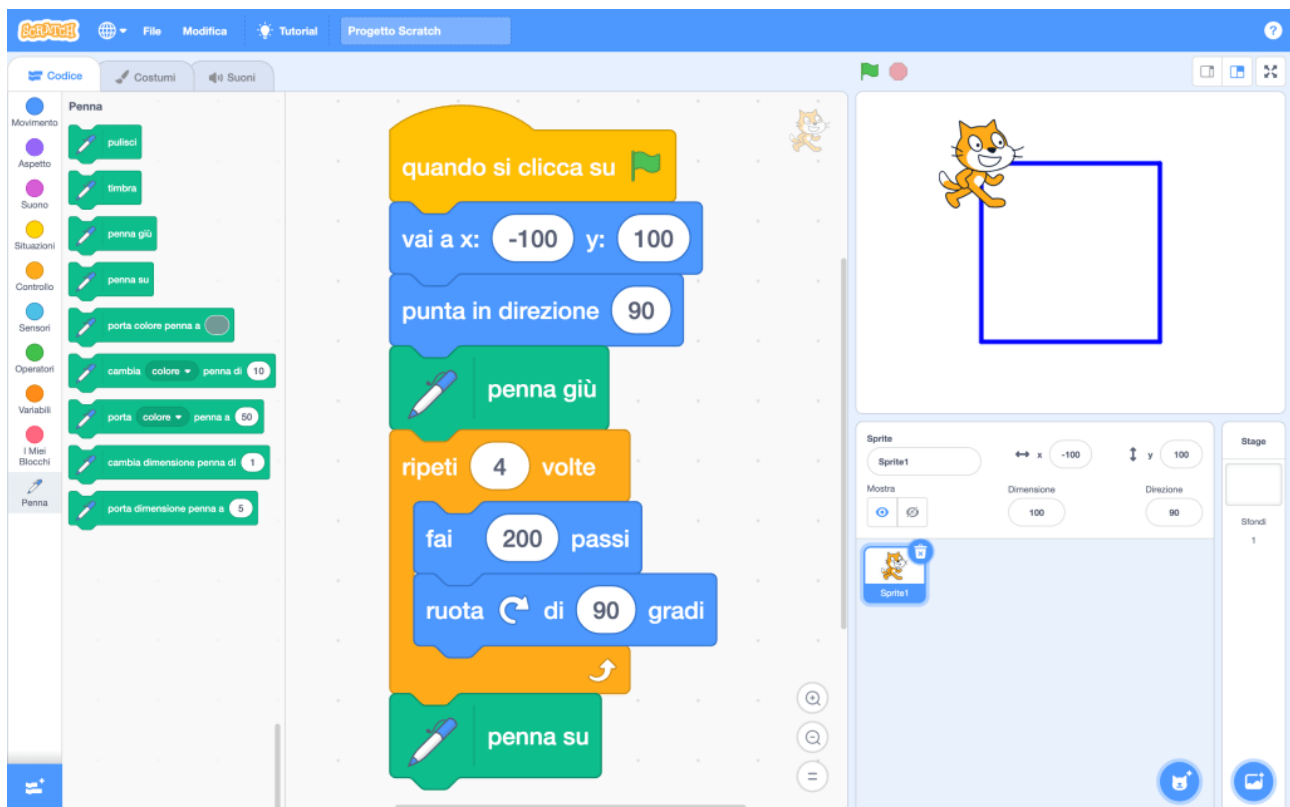


Figura 9 Scratch 3

Nel 2012 Google presentò la libreria [Blockly](#), scritta in JavaScript e pensata appositamente per supportare la programmazione visuale basata su blocchi.

Grazie a questa risorsa è stato possibile realizzare la piattaforma [Code.org](#), la terza versione di [Scratch](#), e molte altre applicazioni di coding e robotica educativa.

E il resto, come si usa dire, è storia...

La figura centrale: Seymour Papert e il Logo

Seymour Papert (1928 – 2016), dopo la laurea in Sudafrica, è stato ricercatore presso l'Università di Cambridge, nel Regno Unito, dal 1954 al 1958.

Quando il celebre psicologo e pedagogista [Jean Piaget](#), noto per il suo lavoro nel campo dello sviluppo cognitivo, lo invitò all'Università di Ginevra, Papert accolse l'invito e rimase in Svizzera da 1958 al 1963.

Nel 1963 Papert lasciò Ginevra per andare negli Stati Uniti, al [Massachusetts Institute of Technology](#) (MIT).

Il ricercatore sudafricano era enormemente ispirato dall'immagine che Piaget aveva del bambino e dall'idea di quanto i bambini potessero imparare **senza un insegnamento diretto**.

In particolare, voleva studiare quali fossero le migliori condizioni per agevolare lo sviluppo cognitivo nei più piccoli, specie per quel che riguardava il pensiero formale.

Alcune sue considerazioni in merito sono presenti nel paper [*Teaching Children Thinking*](#).

Nel 1980 diede alle stampe [*Mindstorms: Children, Computers, and Powerful Ideas*](#), in cui riassume le sue esperienze didattiche con il linguaggio Logo.

Dalla metà degli anni '80, sempre attraverso il MIT, cominciò a collaborare con la **LEGO**.

L'evoluzione della collaborazione tra la LEGO e i MIT è mostrata in questo bellissimo video.

MIT e LEGO

A Papert viene attribuita l'introduzione di concetti quali **pensiero computazionale**, **oggetto per pensare** e **micromondo**, nonché lo sviluppo della teoria dell'apprendimento denominata [costruzionismo](#).

Propongo un articolo del 2 agosto 2016 di Antonio Dini scritto per commemorare la scomparsa di Seymour Papert.

Seymour Papert e le Children Machines

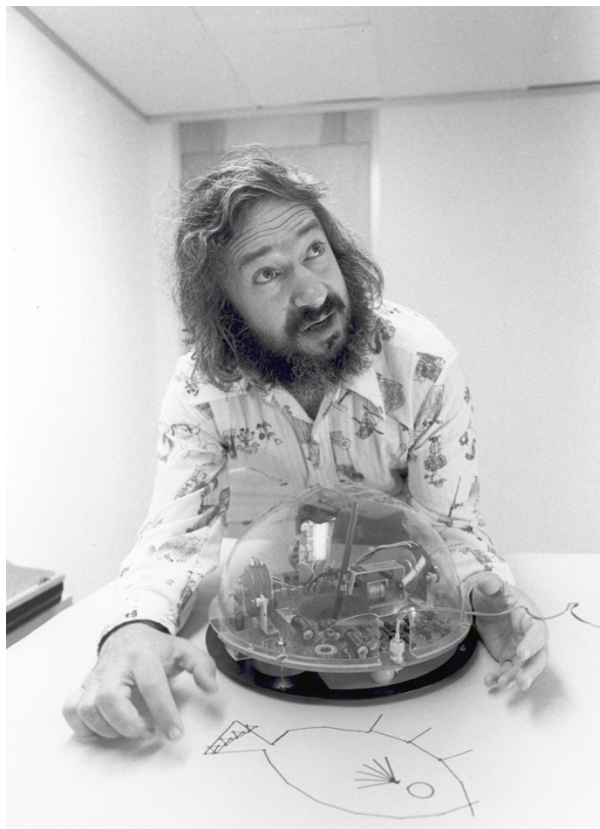


Figura 10 Papert e la prima turtle

È una foto molto simbolica e affascinante che mostra il matematico Papert all'opera al MIT di Boston con la sua prima tartaruga.

Alla turtle poteva far eseguire movimenti e tracciare figure con semplici comandi impartiti dal computer con il linguaggio LOGO.

Ha introdotto con la geometria del LOGO i primi rudimenti di informatica nel complesso e variegato mondo dei bambini.

Personaggio controverso, ha avuto il merito di far cambiare approccio all'insegnamento della matematica alle giovani generazioni. Si è fidato dei bambini e della loro capacità di apprendere e di "costruire da soli" le loro competenze. Ha fornito loro lo strumento per "costruire" tramite righe di codice le prime "learning machine".

I ricordi vanno ai primi momenti con i quali ho affrontato assieme ai miei studenti le prime programmazioni e alle loro facce sorridenti e piene di entusiasmo di fronte ai primi stupefacenti risultati. Ho capito che si poteva insegnare la matematica in un nuovo modo più accattivante e efficiente.

Questo ha permesso l'attivazione del processo di "insight". L'insight consiste nella **comprensione improvvisa e subitanea della strategia utile ad arrivare alla soluzione di un problema o della soluzione stessa** - colloquialmente conosciuto come lampo di genio o con l'espressione inglese:

"Aha! Experience".

La vita

Addio a Seymour Papert, papà del Logo e pioniere della tecnologia nell'educazione

Di [Antonio Dini](#)

2 Ago 2016

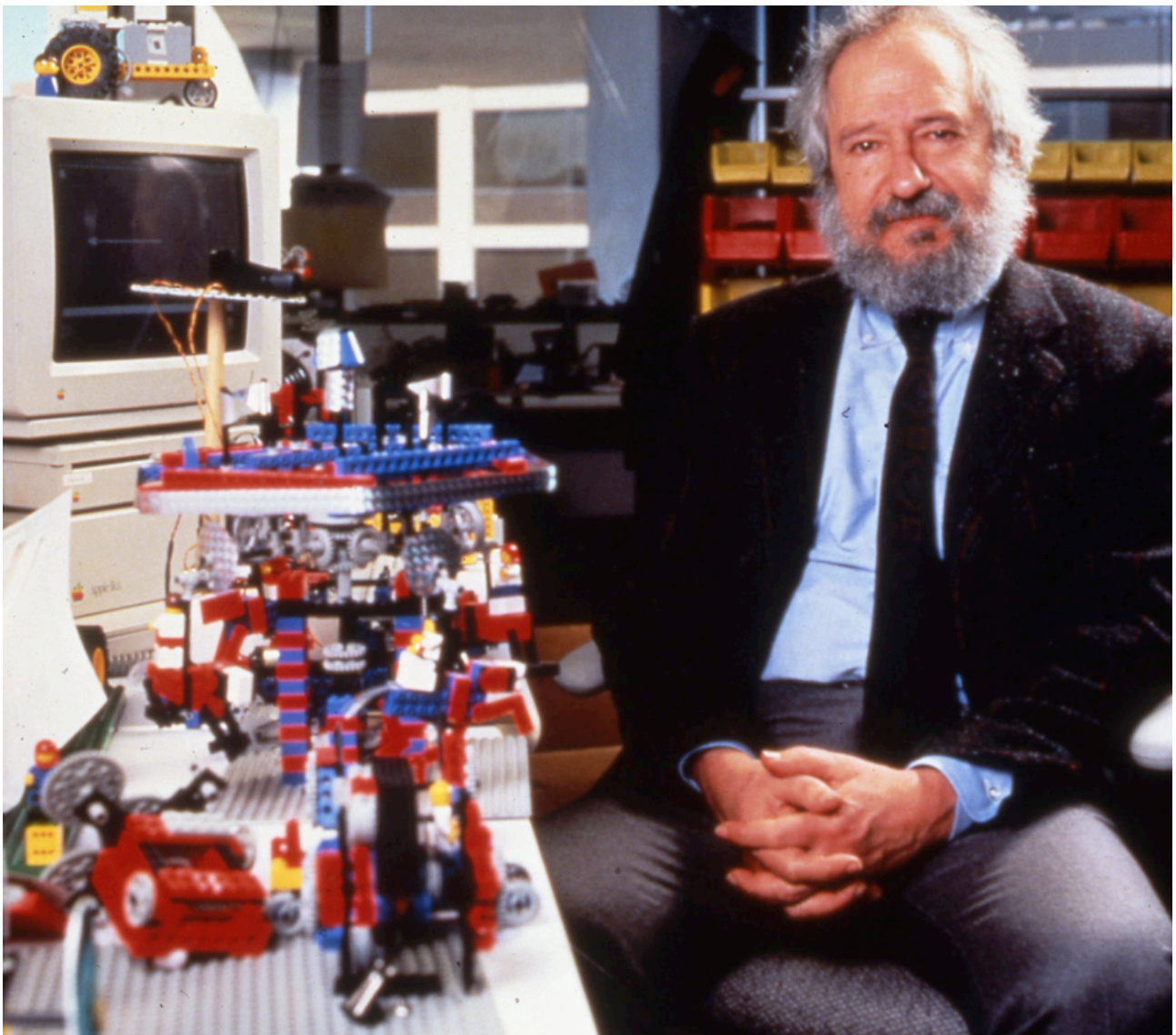


Figura 11 Seymour Papert e i suoi mattoncini Lego con i quali costruiva i primi Robot

La vita

Addio a Seymour Papert, papà del Logo e pioniere della tecnologia nell'educazione

Di [Antonio Dini](#)

2 Ago 2016

La vita di Seymour Papert basterebbe per riempire la carriera di tre o quattro ricercatori di informatica. La sua scomparsa per questo è un momento triste per tutta l'umanità, che vede spegnersi una delle luci che hanno brillato con maggior fulgore nel settore critico dell'impatto delle tecnologie digitali nell'insegnamento a scuola. La chiave per la società del futuro è il modo con il quale educeremo i nostri figli e Papert è stata una delle figure cardine del nostro tempo. Non c'è che dire!!

Nato nel 1928 il sudafricano naturalizzato americano è scomparso il 31 luglio scorso nella sua casa di Blue Hill nel Maine, dopo una carriera che lo ha portato ad attraversare il mondo, diventare uno dei pionieri degli studi sull'intelligenza artificiale, poi creare uno dei più importanti linguaggi di programmazione – il Logo – che hanno aperto la strada a nuovi modi di insegnare l'informatica ai bambini, e infine collaborare a numerosi progetti tra i quali l'OLPC, il computer portatile a bassissimo costo immaginato dal Media Lab del MIT di Boston, dove Papert era professore e direttore dell'Artificial Intelligence Laboratory (1967-1981) assieme a Marvin Minsky.



Figura 12 Foto del 1971: Marvin Minsky, padre dell'AI, assieme a Seymour Papert

Papert aveva studiato matematica prima nel Sudafrica della segregazione razziale (si è laureato nel 1949), aveva poi ottenuto un PhD sempre in Sudafrica nel 1952, seguito da un altro alla Cambridge University nel 1959. Come tutti gli accademici di razza la sua carriera lo aveva portato a spostarsi tra tutti i principali centri di ricerca e di studio del mondo da Cambridge all'istituto Henri Poincaré di Parigi, dall'università di Londra a quella di Ginevra. Dopo tanto girare, era arrivato negli USA nel 1967 per fermarsi al MIT, dove aveva incontrato uno dei suoi grandi mentori, Marvin Minsky, padre degli studi sull'intelligenza artificiale.

Papert aveva lavorato sulle teorie dell'apprendimento, cercato di attivare nuovi modelli di studio e di insegnamento avvicinandosi al pensiero di Jean Piaget, altro grande della ricerca sull'educazione e il modo di apprendere. Papert peraltro aveva lavorato a Ginevra con Piaget diventando uno dei suoi allievi prediletti. Dal Costruttivismo di Piaget (nel settore delle teorie sull'apprendimento) Papert aveva ricavato il suo "Costruzionismo", una teoria dell'apprendimento che si appoggiava su un approccio fortemente multidisciplinare.

Il suo Gruppo di Ricerca sull'Epistemologia e l'Apprendimento all'interno del famoso e importantissimo gruppo del MIT chiamato semplicemente Architecture Machine (che poi sarebbe diventato il Media Lab reso popolare soprattutto dalla visione di Nicholas Negroponte) ebbe una importanza planetaria. Le sue teorie su come la scuola dovrebbe funzionare per consentire

l'apprendimento migliore degli studenti sono tutt'ora la frontiera più avanzata del nostro pensiero in questo ambito.

La relativa giovinezza dell'informatica, la "computer science" degli anglosassoni, il "pensiero computazionale" per i ricercatori come Papert, ci sta permettendo di vedere ancora per un po' di tempo la prima e la seconda generazione dei padri fondatori. Non c'è più Alan Turing (oggi avrebbe 102 anni) ma ci sono ancora molti altri grandi teorici e pratici delle varie discipline che chiamiamo genericamente "informatica". Peccato accorgersene solo quando scompaiono.



Figura 13 Children con la LOGO turtle di Papert

Papert, che per buona parte della sua vita insegnò all'università matematica applicata, anche se poi la sua ricerca si svolse a un livello di astrazione completamente diverso, si trovò a fare anche molte altre cose. A partire dal Logo, il linguaggio di programmazione pensato per essere insegnato ai bambini come vero e proprio linguaggio espressivo per insegnare a pensare in modo computazionale e risolvere problemi.

Il robot mosso dal Logo, visibile manifestazione del lavoro di astrazione e ricerca algoritmica, era la "Logo Turtle", la tartarughina del Logo, che consentiva di eseguire e quindi validare i programmi creati dai bambini. Papert ha sempre pensato che il Logo sarebbe stato un linguaggio semplice per

bambini ma capace di costrutti complessi adatti anche ai programmatori “adulti”. Tanto da lavorare poi nella fondazione che ha prodotto una serie di software basati e programmati in Logo.

Non c’era solo quella ricerca puramente accademica: la Lego aveva chiesto a Papert di lavorare alla realizzazione di un middleware per far funzionare i suoi giocattoli digitali intelligenti e Papert collaborò portando il Logo per la realizzazione del sistema di programmazione dei [Lego Mindstorms](#). È stato uno dei successi più grandi di sempre della Lego: il nome è un omaggio al [lavoro dello scienziato](#).

In più, Papert lavorò (assieme ad altri) alla realizzazione del visionario ma sfortunato (e mal gestito) programma OLPC, One Laptop Per Child, che doveva all’inizio portare i computer nelle mani di bambini dei paesi in via di sviluppo. Era la “Knowledge Machine”, lo strumento digitale di conoscenza che in molti tra i più visionari pionieri del computational thinking avevano visto come necessario per riuscire a smuovere miliardi di bambini senza altre possibilità di realizzare il loro potenziale.

La Children’s Machine avrebbe dovuto consentire un salto di ben più che non di una generazione, ma il programma non riuscì mai ad arrivare a maturità per vari motivi, nonostante le intuizioni alla base (una macchina personale per ciascun bambino e non in un laboratorio, con un sistema operativo alternativo, open e programmabile, fatto meglio di quello di Microsoft) ci fossero tutte.

Soprattutto, un costruttore asiatico, la taiwanese Asus, vide negli [OLPC](#) un fantastico laptop sottopotenziato da vendere a basso o bassissimo costo ai bambini e soprattutto agli adulti ricchi (o relativamente ricchi, rispetto a indiani, africani, brasiliani) dell’Occidente.

Fu la rivoluzione del NetBook, computer super economici che invasero l’Europa e gli USA all’inizio del 2007 e furono gli antesignani sia degli strumenti Post PC di Apple che dei Chromebook di Google, oltre che – per le dimensioni – della categoria degli Ultrabook derivati dai MacBook Air (anche se con un fattore di prezzo completamente diverso). La storia avrebbe potuto essere molto diversa, invece fu solo una alternativa agli allora costosissimi ma potenzialmente sempre più popolari computer portatili.



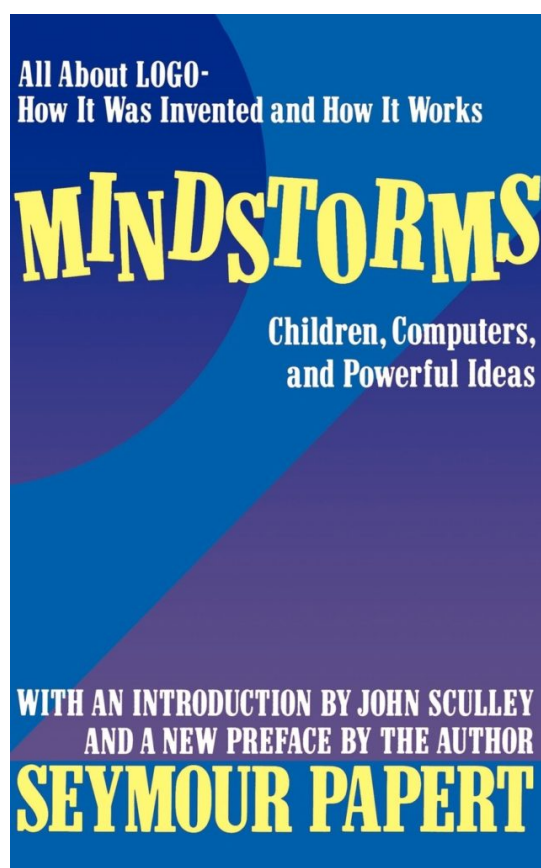
Figura 14 I primi laptop per il III mondo. La carica era a manovella

Tornando a Papert, il suo impatto fu enorme, la sua eredità è gigantesca. Fu lui, per dire, a influenzare il giovane Alan Kay che ragionava sul concetto di Dynabook, che poi è l’antesignano dell’idea di computer portatile ripresa da Steve Jobs e madre di tutti i portatili che oggi abbiamo in mano. Minsky

ha definito Papert uno dei “più grandi educatori per la matematica” mentre con la sua terza moglie, la professoressa del MIT Sherry Turkle, Papert ha scritto uno dei paper di ricerca più influenti, cioè “Epistemological Pluralism and the Revaluation of the Concrete”.

A seguito di un incidente avuto quando aveva già 78 anni, che ha portato a una operazione per eliminare un ematoma nella scatola cranica, e poi della setticemia derivante da questa operazione che ha portato alla sostituzione di una valvola cardiaca, Papert ha dovuto passare un lungo periodo di riabilitazione, aiutato da un team che ha lavorato su di lui con alcune delle tecniche di apprendimento esperienziale che lo studioso aveva concettualizzato decenni prima.

La cosa più importante da ricordare di Papert però è l’impatto che il suo pensiero e la sua attività di ricerca hanno avuto per le teorie dell’apprendimento, dove ha potuto mettere al centro l’uso delle nuove tecnologie per l’apprendimento in generale, soprattutto all’interno delle strutture scolastiche. Da giovane, negli anni Cinquanta, quando viveva a Londra, fu anche una delle figure chiave attorno alla Socialist Review, la rivista nata attorno a un collettivo di ricercatori e intellettuali di sinistra.



Considerazioni personali

Ho iniziato con il LOGO installato nei computer Commodore C64. I ragazzini hanno subito dimostrato un interesse e una capacità di coding e quindi di pensiero computazionale veramente notevoli. Ho “sacrificato” tante ore di matematica e scienze alla scuola media per insegnare loro il LOGO. Questo accadeva negli anni 80!

In quel periodo non c’era ancora INTERNET e l’approccio ai computer era quasi esclusivamente matematico e videogiochi. Ho sempre lavorato con ragazzini delle medie di età compresa tra i 11 anni e 14 anni, la scoperta più notevole dal punto di vista didattico-pedagogico è stata quella preconizzata dal “costruzionismo” di Papert: i ragazzi avviati con poche istruzioni e messi in gruppi

di quattro nel brevissimo tempo erano in grado di realizzare veri e propri capolavori di programmi. Non vi annoio con le ricadute didattiche, ma vi accenno alla teoria del "costruzionismo" così come la tratta Wikipedia.

Il costruzionismo è un processo di apprendimento grazie al quale si tenta la costruzione di rappresentazioni, più o meno corrette e funzionali, del mondo in cui si interagisce. Tale teoria sostiene che l'apprendimento è più efficace quando le persone sono occupate attivamente e hanno a che fare con oggetti tangibili del mondo reale; questo è il collegamento maggiore con l'apprendimento esperienziale sviluppato dalle idee piagetiane. Il costruzionismo di Papert, al contrario però, si focalizza sull'arte dell'apprendimento, o di **"imparare ad imparare"**; ci spinge a riflettere su come si formano le idee, come vengono trasformate ed espresse attraverso i media. Sposta l'attenzione sulla conversazione con i singoli allievi, tanto che l'insegnante costruzionista assume il ruolo di mediatore. Come Piaget, Papert crede che i bambini siano i costruttori dei loro strumenti cognitivi e della loro realtà esterna. Ma a differenza del suo collega afferma che "immergersi" nelle situazioni, piuttosto che guardarle da lontano e diventare un tutt'uno con il fenomeno oggetto di studio sia un potente mezzo di comprensione e un'ottima chiave per l'apprendimento. La conoscenza e il mondo vengono costruiti e cambiati continuamente attraverso l'esperienza personale. L'arte dell'apprendimento si può capire più semplicemente usando un aforisma dello stesso Papert: "...Se un uomo ha fame gli puoi dare un pesce, ma meglio ancora è dargli una lenza e insegnargli a pescare..."; certo è che l'uomo dovrà conoscere la pesca e magari avere buone lenze, dovrà cioè conoscere lo "sfondo", la realtà in cui vive e gli strumenti (i saperi) per soddisfare i propri bisogni. Lo studente è, infatti, il vero protagonista del suo processo di apprendimento, deve essere interessato e motivato alle nuove conoscenze: gli studenti che hanno successo sono quelli che trovano da soli la propria strada. Questo non è un autoapprendimento, perché è l'insegnante che deve disporre un ambiente disponibile, materiali adatti e delle attività didattiche interessanti che stimolino l'alunno a mettere in pratica e modificare la propria esperienza. È per questo, che vi è la convinzione che i processi di insegnamento/apprendimento di tipo costruzionistico risultino facilitati se chi apprende è coinvolto nella soluzione di problemi riscontrabili nel mondo reale; se la conoscenza già esistente è attivata come base di partenza per la produzione di ulteriori e approfonditi stadi di conoscenza e se la conoscenza che ne emerge non è soltanto detta, ma è usata operativamente dall'allievo e integrata alle altre già presenti. Questa forma di insegnamento collaborativo deve essere sostenuta da buone "lenze", cioè dal computer e dalle nuove tecnologie considerati strumenti per realizzare tecnologie della mente/corpo. Il contributo di Papert è proprio in questo, riconoscere nelle nuove tecnologie lo strumento capace di rifondare la scuola, in quanto attraverso il computer è possibile realizzare ambienti inesplorabili concretamente ed ottenere un apprendimento più libero, invece, delle rigide regole imposte dai sistemi scolastici vigenti.

Il linguaggio LOGO

In passato gli unici mezzi di insegnamento, informazione e comunicazione erano il discorso interattivo o il testo passivo con il quale non si poteva avere nemmeno un margine di scambio e/o confronto. Oggi, invece, attraverso i computer e le nuove tecnologie abbiamo un diverso accesso alla conoscenza e possiamo modificare il modo di informare, ma anche di formare. L'uso delle nuove tecnologie permette di rendere concreti concetti astratti, di esplorare aree non conosciute, di poter accedere a diversi saperi in qualsiasi momento senza limiti. L'utilità del computer è quella di permettere agli studenti di impegnarsi contemporaneamente in teoria e pratica, in una prassi complementare. Secondo la teoria di Papert i bambini devono essere aiutati ad imparare e per questo ha coniato una frase: "...il costruttivismo e l'istruzionismo sono i nomi di due approcci che porteranno all'innovazione didattica..." e continua: "L'istruzionismo è la teoria che dice che per ottenere una migliore istruzione, dobbiamo migliorare l'istruzione, e se per far questo useremo i computer questi dovranno fare istruzione...". L'istruzione che

Papert ha dato al computer il linguaggio LOGO, il più noto linguaggio di programmazione per fini didattici. Questa è versatile e facilmente accessibile. È nata nel 1967, come strumento per agevolare e migliorare l'apprendimento; le sue potenzialità sono esplicitate in "Mindstorms", libro pubblicato negli anni '80. Inizialmente venne usato per muovere un robot, al quale venivano dati comandi base come camminare in avanti o indietro e girare a destra o sinistra; questi robot avevano una corazza simile a quella di una tartaruga, da lì il nome del cursore (che nelle prime versioni era un piccolo triangolo). In seguito divenne uno strumento grafico, trasferito sul computer e ad oggi è usato per disegnare e creare immagini. Negli anni '80 il LOGO divenne più sviluppato e iniziò ad essere utilizzato per scopi didattici, specie nei laboratori di geometria. Venne scoperto il carattere pedagogico di questo linguaggio: disegnare attraverso il logo è importante per l'esperienza degli alunni perché rispecchia il modo in cui ci si orienta nello spazio. Per questo il suo utilizzo, nelle scuole di ogni grado, è stato vantaggioso, perché permette anche ad un principiante di avere buoni risultati. Oggi un bambino che usa il LOGO costruisce automi più complessi, della simpatica tartaruga triangolare, ne sono l'esempio diversi laboratori come i LEGO-ROBOT o i LEGO-LOGO che sono fondati sull'idea di costruzione e costruzionismo. A scuola, dunque, le nuove tecnologie sono un passo importante, non solo come nuove fonti di sapere e finestre sempre aperte sul mondo reale, ma anche per migliorare le qualità dell'integrazione scolastica delle persone in situazione di handicap. Purtroppo questi nuovi mezzi non sempre riescono ad entrare nelle scuole, e quando riescono, molto spesso non sono usati come si dovrebbe, ma dovrebbe farci riflettere quanto affermato dallo stesso Papert a conclusione del suo libro "I bambini e il computer. Nuove idee per i nuovi strumenti dell'educazione": "... Sarebbe straziante guardare nel futuro e vedere meravigliose reti telematiche di accesso al sapere a disposizione solo di alcuni, mentre altri ne sono esclusi o vedere che l'istruzione si è trasformata, ancora più che in passato, in un terreno di coltura dell'intolleranza e dell'odio... L'unica scelta razionale... consiste nel lavorare intensamente per incoraggiare il pluralismo educativo con l'impegno non solo di allargare i benefici a tutti coloro che ne hanno bisogno, ma di assicurarsi che coloro che decidono di non avvalersene lo facciano in base a una scelta consapevole."

Gli ambienti multimediali

La multimedialità è la disponibilità di più mezzi, la produzione di diversi media gestita da un computer. Papert in linea con questo, ha sviluppato il concetto di "micromondo", un modo per apprendere in maniera attiva e costruttiva. Il primo micromondo creato fu quello della "geometria della tartaruga", in cui attraverso dei comandi era possibile tracciare figure e disegni e scoprire le specifiche proprietà geometriche in esse presenti. Questo concetto si sviluppa dall'idea che l'autore ha dei bambini, ossia individui che affrontano il mondo che li circonda attraverso l'immaginazione, l'avventura, la fantasia. Questi ambienti virtuali di apprendimento non pretendono di sostituirsi a quelli concreti ma cercano di diventarne parte integrante. I temi affrontanti devo, quindi, rispecchiare problematiche e attività di vita quotidiana; suscitando curiosità, motivazione e interesse nei bambini. L'ambiente per l'apprendimento di Papert, serve agli alunni per concepire progetti e costruire saperi utili e sviluppare l'autonomia, in quanto, tali progetti non sono imposti ma orientati dall'insegnante e aggiustati dall'allievo all'emergere di ogni tipo di difficoltà. Si può quindi affermare che i micromondi papertiani siano vere e proprie "palestre cognitive" in cui si procede per problem finding (individuazione di problemi) e problem solving (risoluzione di problemi).

Propongo anche un lavoro molto interessante fatto in EXCEL che riflette il pensiero di Papert e che è possibile reperire nel sito: www.adalovelace.it

<https://www.adalovelace.it/jenni/>

ARDUINO

Arduino è una piattaforma hardware composta da una serie di schede elettroniche dotate di un microcontrollore. È stata ideata e sviluppata nel 2005 da alcuni membri dell'Interaction Design Institute di Ivrea come strumento per la prototipazione rapida e per scopi hobbistici, didattici e professionali.



Figura 15 la scheda ARDUINO

Prima di parlare dell'esperienza realizzata con i ragazzi vediamo in sintesi cos'è ARDUINO.

È una scheda open source¹ che ci permette di entrare nel mondo del *Physical Computing*. Il Physical Computing è l'atto di creare sistemi interattivi in grado di percepire e reagire al mondo intorno a noi, consentendo agli esseri umani di comunicare attraverso i computer. Richiede un pensiero computazionale, "un modo di risolvere i problemi, progettare i sistemi e comprendere il comportamento umano attingendo dai principi fondamentali dell'informatica". - Jeanette Wing, una competenza fondamentale per tutti, non solo per gli informatici.

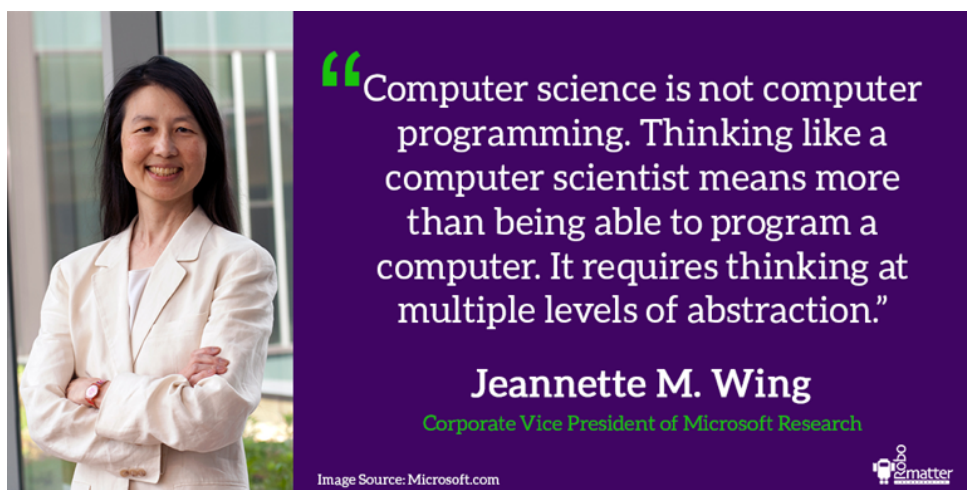


Figura 16 Jeanette M. Wing e il pensiero computazionale

¹ In informatica, di software non protetto da copyright e liberamente modificabile dagli utenti. Nel nostro caso oltre al sw anche la scheda fisica. È il primo caso nel campo dell'elettronica di consumo. In genere tutte le schede elettroniche associabili a sensori e attuatori si servivano di sw proprietari e non erano facilmente modificabili.

L'informatica non è la programmazione del computer. Pensare da informatico significa molto più che saper programmare un computer. Ciò richiede di pensare a più livelli di astrazione.

Con la “Physical Computing” significa costruire sistemi fisici interattivi con l'uso di software e hardware in grado di rilevare e rispondere al mondo reale o analogico.

In senso lato, physical computing è un framework (struttura) creativo per la comprensione del rapporto degli esseri umani con il mondo digitale.

Nell'uso pratico, il termine più spesso descrive handmade (fatto a mano) arte, design o fai da te per progetti o hobby che utilizzano sensori e microcontroller² per tradurre un ingresso analogico (calore, luce, suono) per un sistema software, e/o il controllo elettromeccanico di dispositivi quali ad esempio motori elettrici o altro hardware.

Qual è la differenza tra microprocessore e microcontrollore?

*Il **microprocessore** non dispone di periferiche nel circuito integrato. Le periferiche vengono invece collegate esternamente per espandere i casi d'uso del **microprocessore** oltre l'elaborazione matematica e logica. Al contrario, i **microcontroller** si collegano a periferiche su chip attraverso un bus di controllo interno.*

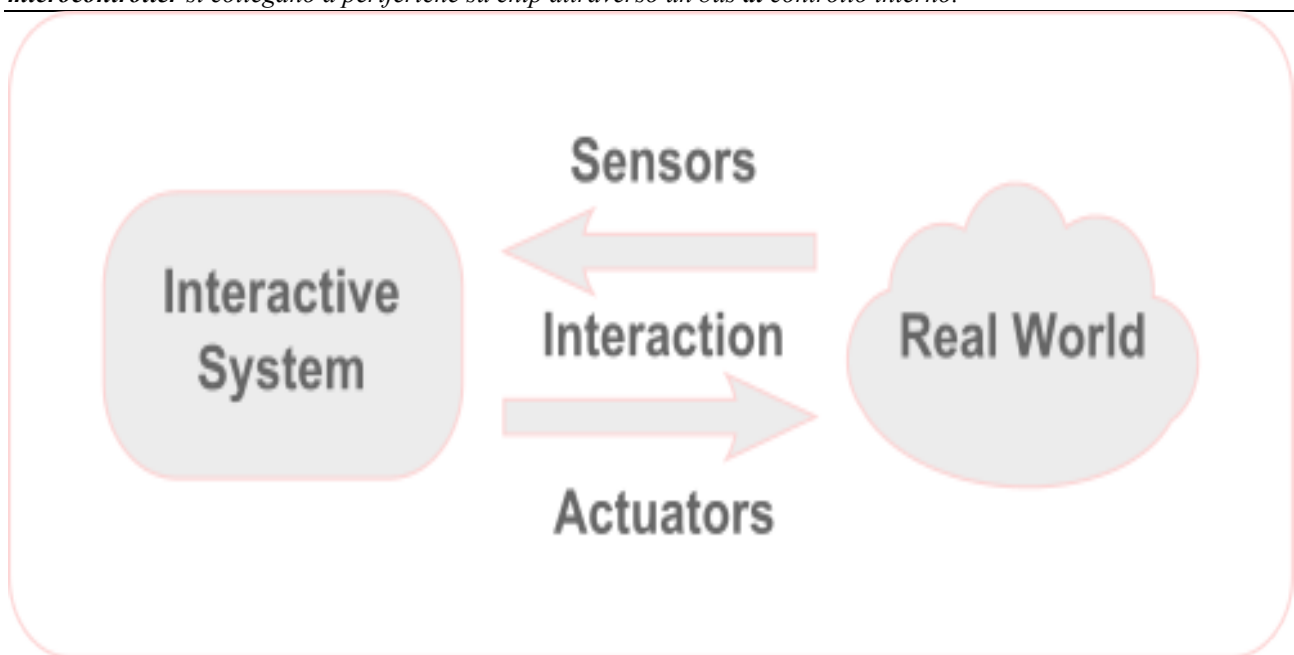


Figura 17 mappa Physical Computing

Si propone un esempio di sensore e di attuttore, ne esistono un'infinità, ma è importante capire la diversa natura. Il sensore interagisce con l'ambiente, luce, suono, vento ecc. L'attuttore trasforma il segnale che riceve dal sensore e tramite un programma compilato al computer esegue le istruzioni ricevute.

² Un **microcontrollore** (in inglese **microcontroller** in acronimo **MCU** ovvero *MicroController Unit*), in elettronica digitale, è un dispositivo elettronico integrato su singolo circuito elettronico, nato come evoluzione alternativa al microprocessore e utilizzato generalmente in sistemi embedded ovvero per applicazioni specifiche di controllo digitale

Sensori & attuatori

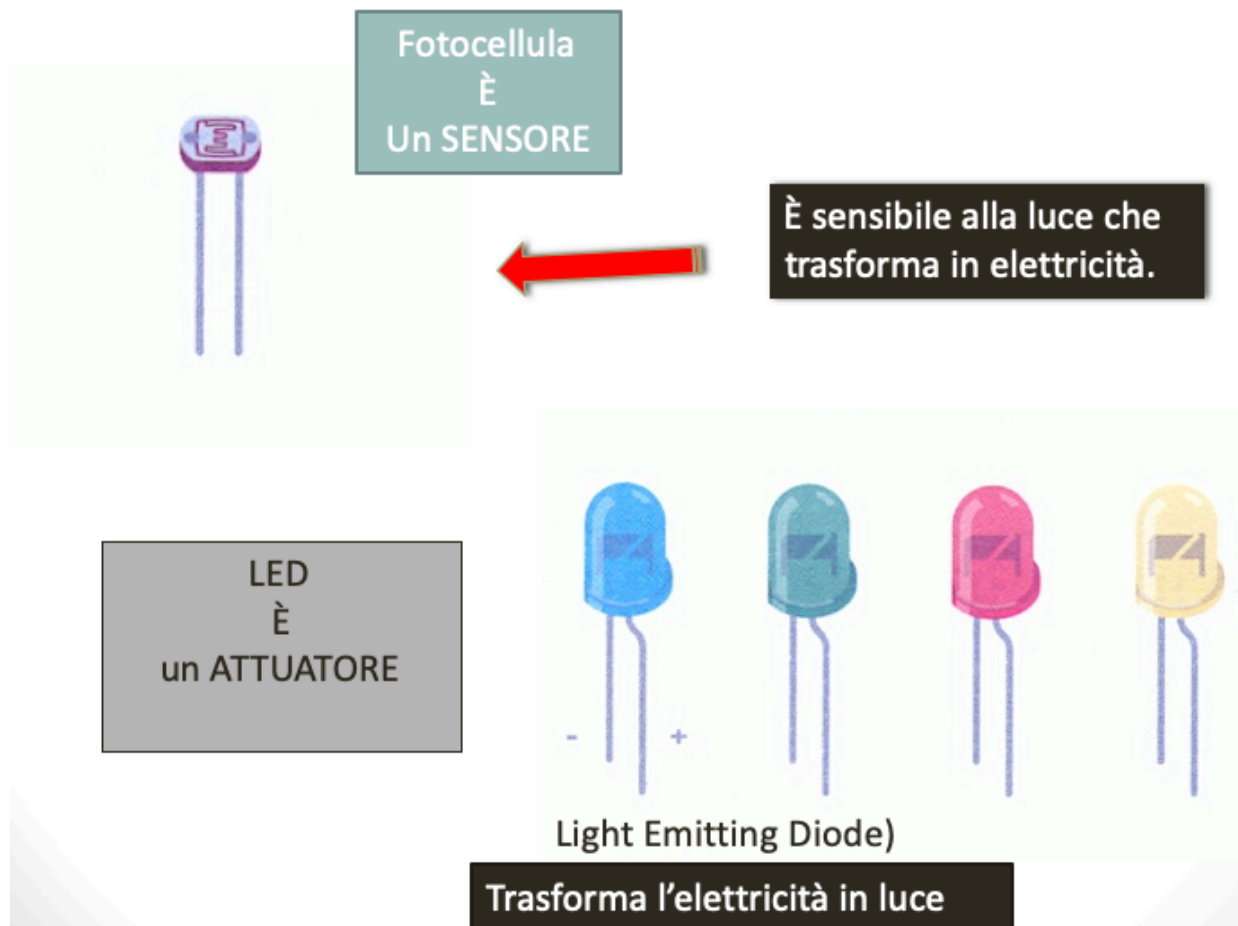


Figura 18 sensori e attuatori

IDE

È l'ambiente di programmazione di Arduino e si presenta come una finestra dove si può scrivere lo sketch (progetto) da interfacciare con Arduino. L'ambiente di sviluppo integrato (IDE) di Arduino è un'applicazione multiplatforma in Java, ed è derivata dall'IDE creato per il linguaggio di programmazione Processing e per il progetto Wiring. È stata creata da Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, e David Mellis. In sostanza è un dialetto C++. In pratica tramite una porta USB³ si collega la scheda ARDUINO, nel frattempo sul computer tramite lo sketch IDE si programma l'azione di Arduino. Ad esempio voglio accendere due LED a vari tempi di intermittenza. Scritto il programma, lo si implementa nella scheda. Lo si attiva ed ecco accendersi e spegnersi i led.

³ USB (acronimo di *Universal Serial Bus*, pronunciato è uno standard industriale di comunicazione seriale che permette di collegare dispositivi diversi (per esempio mouse e tastiera del computer, macchina fotografica, stampante, dispositivi audio e altre periferiche) tramite la stessa interfaccia.

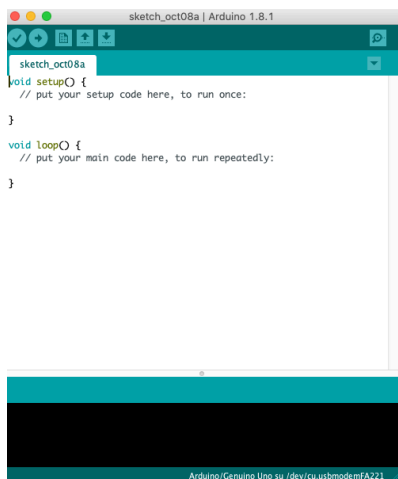


Figura 19 ambiente IDE

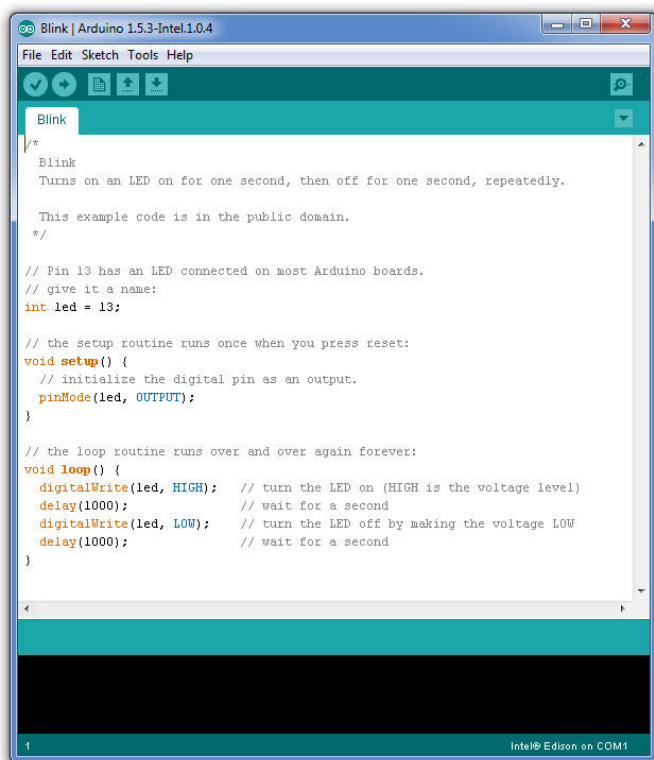


Figura 20 BLINK è il programma di prova di ARDUINO

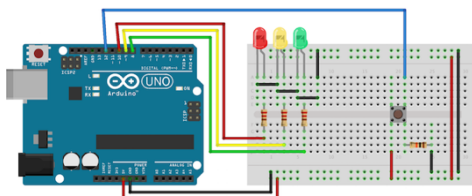


Figura 21 es. di progettazione con FRITZING

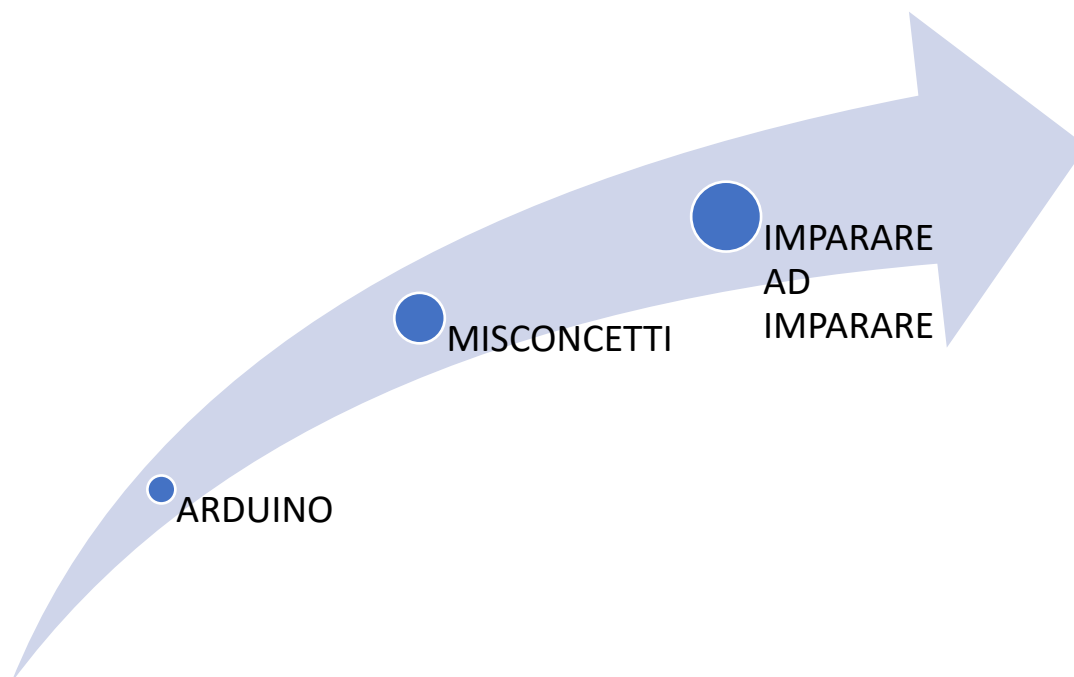
Il programma di prova fa sì che il led sia posizionato sul pin 13 esegua un loop di acceso e spento con un intervallo di 1000 millisecondi (un secondo).

Nel sito www.adalovelace.it c'è un link personalizzato che porta ad un sito realizzato con un allievo di terza media Marco Manfrin dove sono state raccolte 14 lezioni di base per chi volesse cimentare con questa dimensione dell'informatica.

<https://arduinocheapassione.jimdofree.com/>

Articolo “Misconcetti & Arduino”

Il lavoro fatto con i ragazzi in tre anni scolastici articolato in due pomeriggi settimanali è stato descritto in un articolo che ha avuto risonanza mondiale. Di seguito.



MISCONCETTI & ARDUINO

Prof. Sandro Corradini, Dr. Antonio Piva
Istituto Comprensivo
Via dello Stadio 3, 31052 Maserada sul Piave (Treviso)
sandro.corradini7@gmail.com
antonio@piva.mobi

Abstract. È la continuazione della sperimentazione svolta in una scuola secondaria di I grado sull'avvio all'uso di ARDUINO. La prima sperimentazione è stata presentata a DIDAMATICA 2016⁴. Quest'anno, sono stati replicati e rinnovati i progetti ponendo un'attenzione maggiore ai processi di apprendimento. Una volta rilevata la presenza di “misconcetti” è stata elaborata una strategia per superarli e giungere a un “conceptual change” che aiuti a una progettazione più consapevole con ARDUINO.

Keywords: misconcetti, conceptual change, Arduino.

1. Introduzione

Il lavoro rappresenta un'evoluzione di un progetto su ARDUINO proposto nel 2016, nell'ambito del Tempo Prolungato di un Istituto Comprensivo. Ci si pone l'obiettivo primario di correggere l'impostazione dei cosiddetti “misconcetti” luoghi comuni e preconcetti che rappresentano la loro conoscenza ibernata e fungono da ostacolo ai fondamenti della conoscenza corretta dei giovani. Ulteriore obiettivo è quello di valutare la curva di apprendimento degli studenti dopo il corso.

⁴ È il paper 22 con il titolo di “LOGIC GATES verso ARDUINO” presentato a DIDAMATICA 2016

La scelta di Arduino con l'ingresso nel mondo dei maker e del "physical computing" è dovuta alla sentita necessità da parte degli allievi di "learning by doing"⁵. Si ha inoltre la possibilità perseguire i seguenti obiettivi più specifici:

1. Il "physical computing" favorisce il pensiero formativo o computazionale;
2. Consente agli allievi e alle strutture informatiche di "cooperare" nel prendere decisioni e controllare situazioni complesse;
3. Lavorare con precisione e accuratezza per un efficace CODING.

La durata del laboratorio è di un quadrimestre, con due pomeriggi alla settimana con lezioni della durata di due ore. A singolo quadrimestre partecipano due gruppi di circa 15 ragazzi, un gruppo di neofiti (I media) e un gruppo di più esperti, ragazzi che avevano già affrontato l'anno precedente il lab. (II o III media). Ogni gruppo usufruisce mediamente di 14-15 lezioni.

2 Fase iniziale

Nella fase iniziale è proposto agli allievi neofiti un semplice questionario che testa le conoscenze informatiche presentate dai ragazzi. Per brevità è proposto solo il test corredato di una tabella dati. È stato chiarito che se non si sentivano sicuri della risposta, dovevano rispondere "non so".

1. Cosa intendi con la parola informatica?
2. Chi ha inventato l'informatica?
3. La matematica c'entra con l'informatica?
4. Quali sono gli strumenti dell'informatica?
5. Sapresti descrivere, brevemente, gli aspetti dell'informatica nella vita moderna?
6. Chi sono i nativi digitali?
7. Cosa è la "rete"?
8. Cosa sono le "logic gates"?

	coerenza risposte	comprensione domande
buono	41%	12%
suff	18%	35%
scarsa	41%	53%

	non so	
quest 1	35%	Cosa intendi con la parola informatica?
quest 2	88%	Chi ha inventato l'informatica?
quest 3	0%	La matematica c'entra con l'informatica?
quest 4	18%	Quali sono gli strumenti dell'informatica?
quest 5	29%	Sapresti descrivere, brevemente, gli aspetti dell'informatica nella vita moderna?
quest 6	41%	Chi sono i nativi digitali?
quest 7	0%	Cosa è la "rete"?
quest 8	100%	Cosa sono le "logic gates"?

Tabella 1

La percezione che i cosiddetti "nativi digitali" hanno dell'universo informatico e del suo coinvolgimento nella vita di tutti i giorni è molto confuso, con un guazzabuglio di luoghi comuni, sempre comunque mal digeriti.

Emergono dalle risposte misconcetti da considerare come "interpretazioni erranee" o convinzioni specifiche scorrette, altre (le risposte "non so") indicano solo ignoranza dell'argomento. Tutto questo può rappresentare un effettivo "ostacolo all'apprendimento". Sorge quindi la necessità di avviare un processo di "conceptual change" che porti a una corretta formalizzazione gli argomenti trattati.

3 CONCEPTUAL CHANGE MODEL

Il modello "conceptual change" utilizzato è stato il seguente:

⁵ Imparare facendo, imparare attraverso il fare.

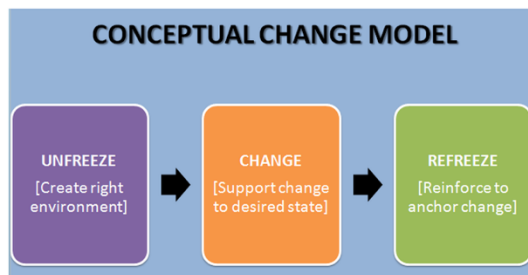


Figura 22 il modello di conceptual change

UNFREEZE (lett. scongelamento) per creare l'ambiente giusto

CHANGE (lett. cambio) un cambiamento in sostegno allo stato desiderato

REFREEZE (lett. ricongelamento) rafforzare per ancorare il cambiamento

UNFREEZE - La prima proposta è stata quella di costruire assieme una **timeline del pensiero informatico**. La timeline è stata incentrata sui personaggi storici che hanno portato alla rivoluzione informatica. Si è partiti da Aristotele (concetto di V/F), Leibnitz (sistema binario 0,1), Boole (AND, OR, NOT), Shannon (switching circuits), fino ad arrivare alla definizione delle LOGIC GATES.

Questa è stata la fase UNFREEZE, che ha permesso lo scongelamento e la creazione dell'ambiente adatto (background) dove innestare le conoscenze e competenze utili poi per lavorare con ARDUINO. Il tutto è poi rifluito in una presentazione.

CHANGE – **costruzione di un circuito su cartoncino**. Si è proceduto alla costruzione manuale⁶, su cartoncini di formato A3, delle principali porte logiche: AND (due interruttori in serie), OR (due interruttori in parallelo), NOT (un interruttore invertitore). È stata una fase importante per l'accrescimento dell'autostima, per fissare i concetti riguardanti le tre porte principali.

REFREEZE – **ARDUINO**. Si procede con la creazione di un circuito, presentazione dell'IDE (Integrated Development Environment) e programmazione di uno sketch, come illustrato nel prossimo paragrafo.

4 Arduino e progettazione sketch

ARDUINO è una scheda "open source" che permette la programmazione di sensori e attuatori come led, fotocellule ecc. Si entra nel "Phisical Computing".

Studio di problema, relativa circuitazione e coding per illustrare come ARDUINO interpreta i segnali dal mondo esterno (sensori) e modula le risposte con gli attuatori. Si realizzano diversi progetti partendo da quelli più semplici come la gestione dei led fino ad arrivare al progetto che prevede la simulazione di un semaforo.

Si costruisce il circuito sulla breadboard e si scrive il Code necessario per farlo funzionare nell'ambiente IDE di Arduino. Questo progetto si presta a successive sofisticazioni che permettono l'introduzione di strutture di controllo: semaforo semplice (semplice sequenza d'istruzioni), semaforo con il giallo lampeggiante in questo caso dalla semplice sequenza si passa alla ripetizione della situazione di lampeggio del giallo e quindi l'introduzione dell'istruzione for (ripetizione di un blocco d'istruzioni), semaforo con richiesta si lascia il semaforo lampeggiante con il giallo introducendo un bottone alla cui pressione si attiva il semaforo vero e questo è possibile con l'introduzione dell'istruzione if (struttura condizionale)

L'esperienza laboratoriale prosegue con lo studio e l'introduzione di altri sensori come ad esempio quelli a ultrasuoni, RGB, TMP, servo motore.....

5 L'apprendimento

Questo laboratorio favorisce la riflessione e la concentrazione stimolando un apprendimento significativo, il "learning by doing" è adatto all'interiorizzazione di una metodologia di apprendimento che rende il soggetto progressivamente autonomo nei processi conoscitivi (deutero apprendimento⁷):

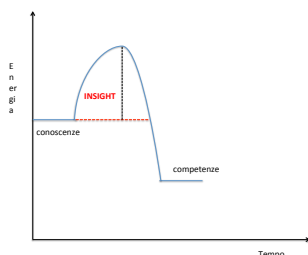
I fase: avvicinamento ai contenuti, fatto con diversi metodi e strategie

II fase: una volta scoperti i contenuti, è naturale il processo d'imitazione con il quale l'allievo cerca di replicare e prova a giocare con i contenuti.

⁶ Questa esperienza manuale, è fondamentale, "learning by doing". Tutti gli allievi hanno gradito la manualità associata alla teoria.

⁷ Imparare a imparare. (Bateson)

III fase: INSIGHT (intuizione) dove ogni processo di apprendimento ha bisogno di una “energia di attivazione”, altrimenti i contenuti restano lì fermi, senza contribuire alla costruzione delle competenze. In questa occasione l’allievo si rende indipendente nella risoluzione dei problemi.



Il catalizzatore di questa reazione, la scintilla che fa partire la reazione, dipende dalla dinamica affettiva e relazionale che si instaura nei gruppi di lavoro e dal fatto “funziona o non funziona?” e quindi dalla continua ricerca degli errori.

Figura 23 INSIGHT

*Quel sol che pria d'amor mi scald'o il petto Di bella verit`a m'avea scoperto **Provando e riprovando**, il dolce aspetto.
Paradiso, canto terzo, 1-3.*

Provando e riprovando è il motto della scienza moderna!

6 Conclusione

Concludendo si può dire che i ragazzi conseguono con questo lab.

- Sicuro possesso di saperi disciplinari
- Padronanza delle metodologie
- Coinvolgimento della dimensione affettiva-relazionale
- Acquisizione di capacità metacognitive

7 Bibliografia

Lewis Carroll, Alice, Mondadori, 2015.

Seth Lloyd, Il programma dell'universo, Einaudi, 2006.

Mario G. Losano, Informatica per le scienze sociali, Piccola Biblioteca Einaudi, 1985.

Maik Schmidt, Il manuale di Arduino, Apogeo, 2015.

Bruno D'A- Silvia Sbaragli, Analisi semantica e didattica dell'idea di “misconcezione”

Mattia Valsania, Corso pratico di Arduino Modulo base, Area 51, 2015.

https://www.google.it/webhp?sourceid=chrome-instant&rlz=1C5MACD_enIT557IT558&ion=1&espv=2&ie=UTF-8#q=ermanno+puricelli+unit%C3%A0+apprendimento, LE UNITA' DI APPRENDIMENTO, di Ermanno Puricelli

<http://www.fisica.uniud.it/GEI/GEIweb/ricerche/res.htm>, IL CONCEPTUAL CHANGE, Michelini Salvi

Edgar Morin, La testa ben fatta. Riforma dell'insegnamento e riforma del pensiero, Milano, Raffaello Cortina Editore, 2000, pp.138, ISBN 88-7078-613-7

Simbiosi uomo-computer di JCR Licklider IRE su Human Factors in Electronics, volume HFE1, pagine 4-11, marzo 1960

Dante Alighieri Paradiso canto 3



Figura 24 Bambina reduce da una rissa

Dopo tutte queste botte è possibile che vi sentiate in questo modo!!!! Norman Rockwell, grande disegnatore americano, ha descritto la sua società con grande sensibilità e occhio acuto e preveggenza. La bambina, reduce da una rissa, rappresenta lo scontro tra il modo di pensare classico e le novità che stanno irrompendo nel mondo.

I LIVELLI DI APPRENDIMENTO

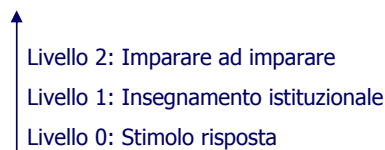
L'apprendimento, in linea generale, consiste in un processo che consente una modificazione durevole del comportamento, per effetto dell'esperienza. Con questa definizione si escludono tutte le modificazioni di breve durata dovute a condizioni temporanee, episodi isolati, eventi occasionali, fatti traumatici, mentre il riferimento all'esperienza esclude modificazioni determinate da fatti innati o dal processo biologico di apprendimento.

Nelle classificazioni dei livelli di apprendimento proposta da Bateson, si distinguono tre differenti livelli di apprendimento (cfr: G. Bateson, *"Verso un'ecologia della mente"* – Adelphi):

Livello zero Si tratta dell'apprendimento associativo, detto anche semplice o meccanico, fondato principalmente sulla reazione stimolo – risposta, che dà origine alla formazione di abitudini.

Livello uno Con questo livello di apprendimento Bateson fa specifico riferimento all'apprendimento di tipo istituzionale classico, quello delle funzioni scolastiche. In questo contesto è l'insegnante a stabilire che cosa deve imparare l'allievo ed è lui a definire i ritmi e la qualità dell'insegnamento da erogare. Questo tipo di apprendimento – detto anche complesso – coinvolge funzioni superiori dell'apparato psichico come la percezione, l'intelligenza e, in generale, i processi cognitivi propri dell'uomo.

Livello due Con questo livello Bateson si riferisce a funzioni di valutazione complesse e sofisticate, quali, ad esempio, la consapevolezza di esistere il senso dell'esistenza, la capacità di provare sentimenti sui propri sentimenti. Si tratta di una forma di apprendimento continuo e sistematico in cui il soggetto "impara ad imparare", facendo esperienza e modificando le risposte in relazione alle esigenze del contesto in cui avviene la relazione. Si tratta pertanto di una forma di apprendimento cognitivo superiore, basato sulla capacità di fornire risposte adeguate agli stimoli ricevuti. Esso rappresenta "la differenza che fa la differenza" nella qualità delle risposte comportamentali messe in atto da una persona.



DARK

Le soddisfazioni avute con i ragazzi sono state immense e gratificanti, l'articolo è conosciuto e apprezzato in tutto il mondo scientifico, ma ha fatto anche emergere un altro aspetto inquietante e significativo della nostra epoca:

I MISCONCETTI

Cosa sono i misconcetti? Nel campo delle scienze e matematica non sono altro che "concetti sbagliati". Come quando un ragazzino ti dice che $\frac{1}{4}$ è sicuramente più grande di $\frac{1}{2}$. Ma a questo misconcetto è facile rimediare, diverso è quando ti dicono che la terra è piatta, ma anche a questo è facile rimediare. Diverso è quando contestano l'evoluzione di Darwin e questo succede negli stati del sud degli Stati Uniti e nella vecchia URSS dove il Darwinismo era considerata una teoria borghese come l'ereditarietà di Mendel. In questo caso il misconcetto assume anche una veste politica molto pericolosa.

A questo proposito cito testuale una affermazione del Nobel per la fisica Giorgio Parisi:

Io penso che la scienza e la religione siano cose completamente diverse. La fisica, e più in generale la scienza, cercano una spiegazione del mondo restando dentro il mondo, mentre le religioni - almeno quelle che conosco io, diciamo le religioni monoteistiche - cercano risposte fuori dal mondo, in qualcosa che trascende il mondo.

Ho incontrato spesso fisici profondamente religiosi, ma questo aspetto non è quasi mai entrato nel nostro lavoro, nelle nostre discussioni. In fondo, essendo la religione un ambito che cerca le risposte al di fuori del mondo, è difficile che avesse punti di contatto con il lavoro come fisico.

Non penso che la religione sia da combattere. Io non sono religioso, ma non ho mai pensato di fare una battaglia contro la religione, tanto più utilizzando la mia autorità di scienziato per esprimermi su quei temi. Mi sembra una follia, senza voler giudicare le persone che lo fanno.

Per quanto riguarda l'"oscurantismo", in questo campo ci sono delle opinioni che si classificano come più politiche - ad esempio "la religione è l'oppio dei popoli".

Posso avere le mie idee politiche sugli influssi di un certo radicalismo islamico nella società, o di un certo pensiero del radicalismo cristiano protestante contro le teorie dell'evoluzione che si vede soprattutto negli Stati Uniti, ma io terrei completamente separato un politico sulla religione da un pensiero scientifico sulla religione.

Negli ultimi anni il peso della religione e la sua influenza sulla storia di questo pianeta - penso a molte guerre di oggi, ai massacri in Jugoslavia, al conflitto tra sunniti e sciiti - sono tornati con una forza che non avevano da tanto tempo. La Seconda Guerra mondiale non era certamente una guerra mossa in primo luogo dalla Germania protestante alla Francia cattolica. Il peso odierno della religione si vede anche perché c'è una ricerca delle persone a valori, significati della vita che la scienza non può dare.

La scienza ha risposte al mondo nel mondo, ma non spiega il perché del mondo. C'è una vecchia affermazione della dottrina cattolica: "La fede è una grazia". Io questa grazia non ce l'ho, e quindi le risposte a quelle domande non me le so dare.

- Giorgio Parisi

La genesi

Mi sono imbattuto nei misconcetti quando ho somministrato un questionario ai miei allievi del corso di ARDUINO: una delle risposte che mi ha lasciato basito è stata la seguente, illustrata da due slides.

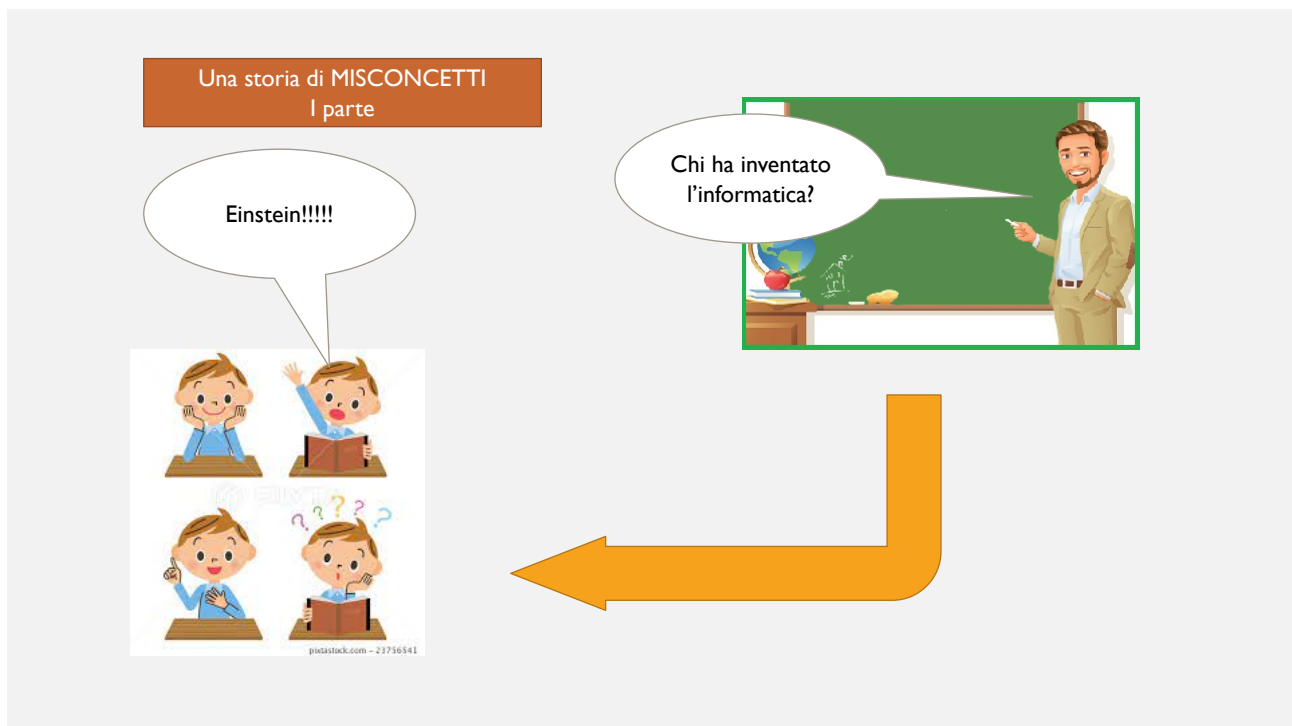


Figura 25 Einstein 1

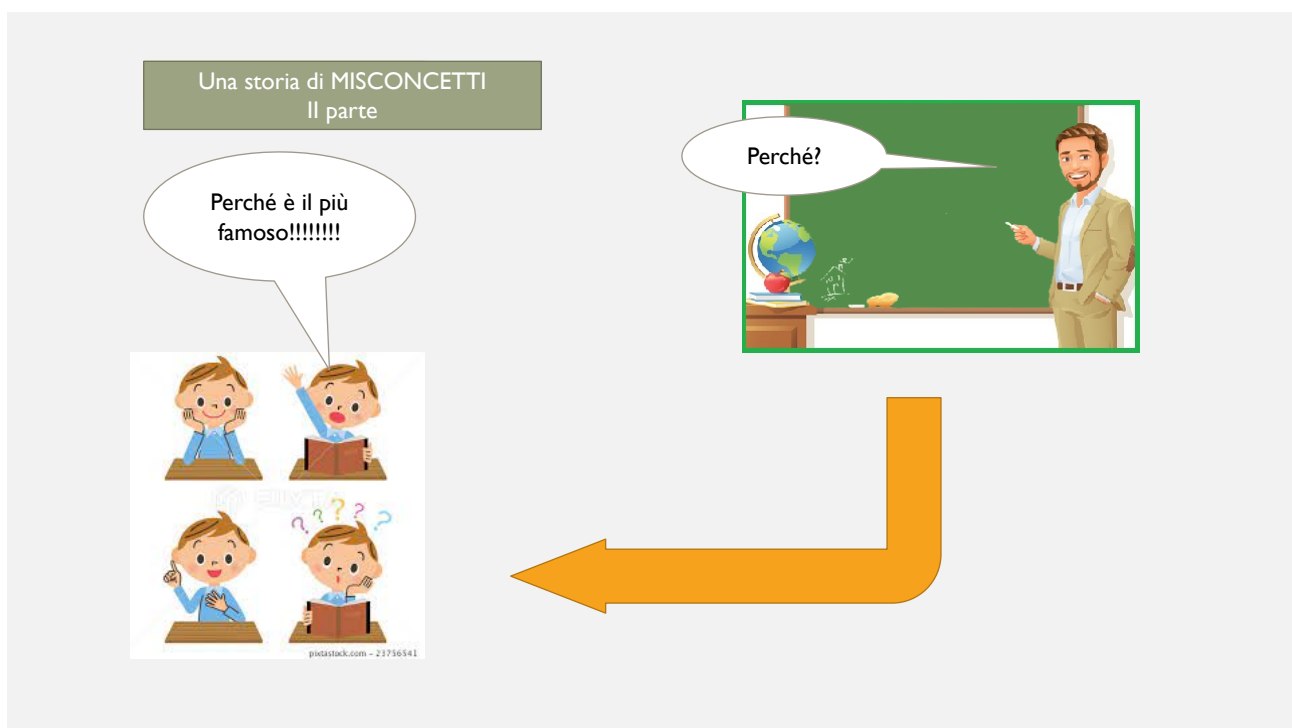


Figura 26 Einstein 2

La domanda era rivolta a ragazzini di I media dell'età di 11 anni e le considerazioni da fare sono numerose. È una risposta ingenua, ma ti dà molte indicazioni sul modo di vedere l'informatica. Gran parte non hanno risposto alla domanda e solo due hanno dato quella risposta che è chiaramente un misconcetto. Per la prima volta ho sentito la necessità di escogitare un metodo per porre rimedio a questo misconcetto. Ci sono state altre riflessioni sul fatto che la gente quando affronta un problema che conosce poco fa ricorso a misconcetti o risposte evasive o ingenuie.

Al giorno d'oggi il problema dei misconcetti è importante e da non sottovalutare.

II CONCEPTUAL CHANGE

Partendo da questi presupposti è un dovere trovare un modo, un metodo per combattere questi misconcetti. Come già accennato nell'articolo il modello da seguire è il:

CONCEPTUAL CHANGE MODEL

Con le sue tre fasi:

1. UNFREEZE (lett. scongelamento) per creare l'ambiente giusto
2. CHANGE (lett. cambio) un cambiamento in sostegno allo stato desiderato
3. REFREEZE (lett. ricongelamento) rafforzare per ancorare il cambiamento

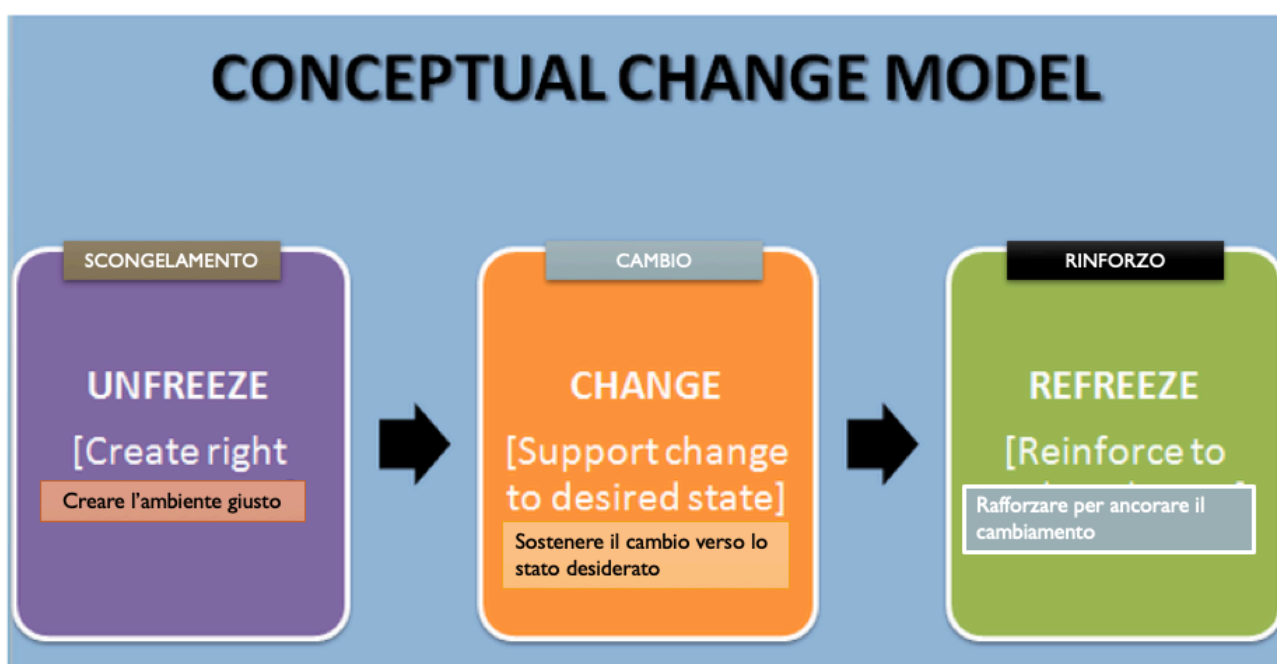


Figura 27 Conceptual Change Model

I fase UNFREEZE

La I fase prevede la creazione del giusto ambiente sul quale operare per iniziare la demolizione del misconcetto. Nel nostro caso ho adoperato/costruito una presentazione che mi raffiguri una Timeline del pensiero informatico dal punto di vista storico.

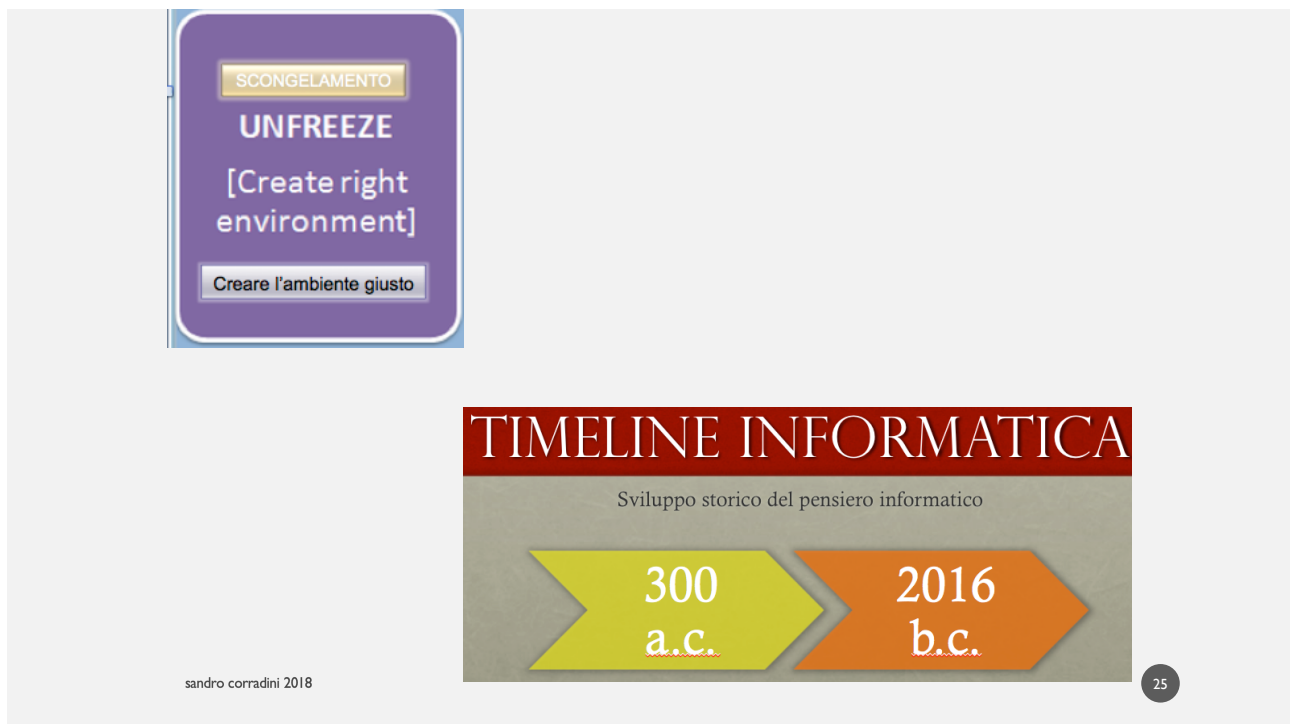


Figura 28 prima fase conceptual change UNFREEZE

Innanzitutto in questo modo ho creato l'ambiente giusto nel quale iniziare a lavorare. Ha appassionato i ragazzi e li ha fatto ricredere: hanno esclamato alla fine "ma l'informatica è una storia lunga..... e interessante, non è come la immaginavamo!"



Figura 29 prima fase UNFREEZE creare l'ambiente giusto

Questa presentazione (18 slides) la si può consultare al seguente link:

<https://www.adalovelace.it/presentazioni-in-pdf/>

II fase CHANGE

La seconda fase prevede il CHANGE o cambiamento. Sempre nell'ambito del "learning by doing" cioè che i ragazzini facessero da soli, abbiamo realizzato le LOGIC GATES manualmente su cartoncini verificando quello che avevamo studiato nella prima fase.

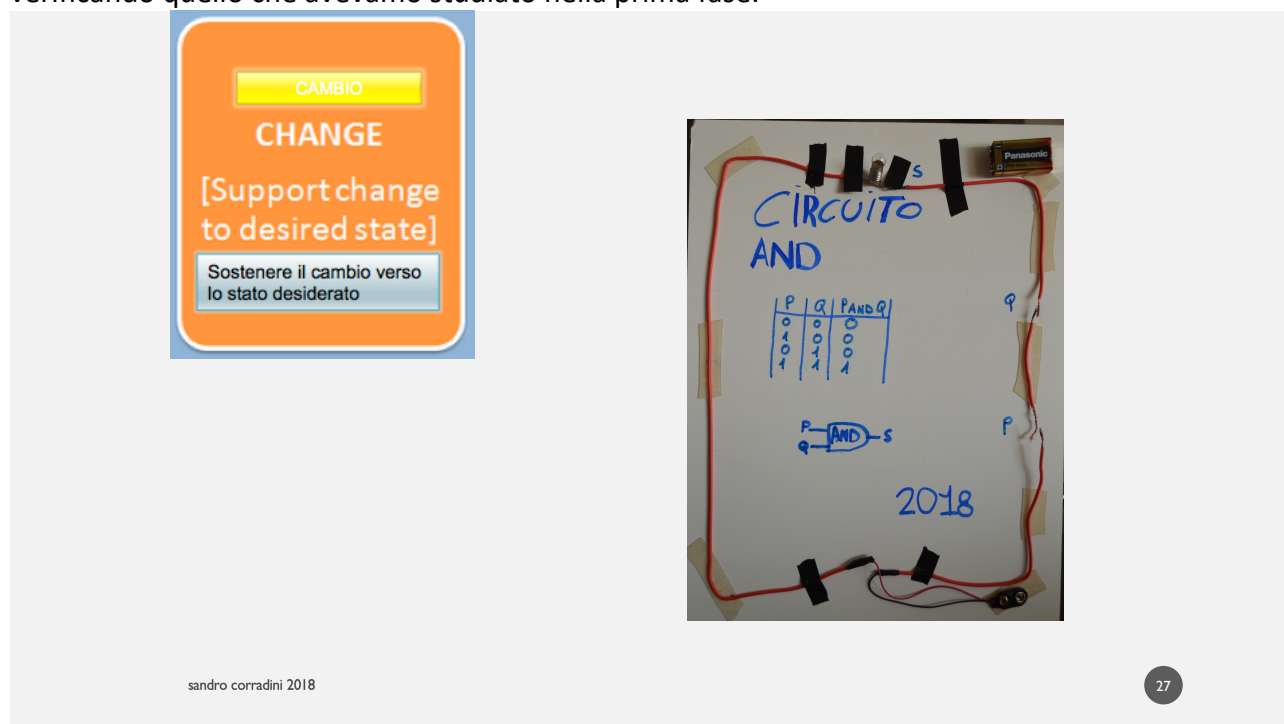


Figura 30 seconda fase CHANGE costruzione delle LOGIC GATES

III fase REFREEZE o rinforzo

Rinforzare per ancorare il cambiamento. Nella terza fase ho finalmente introdotto ARDUINO e il CODING necessario per farlo funzionare. Fase complessa dove i ragazzi hanno risposto in modo eccezionale, presi da sincero entusiasmo! Hanno programmato da soli e come tutti i bambini hanno subito imparato le parole da adoperare per programmare: è stata una esperienza molto gratificante e l'ultima della mia carriera di docente.

C'è stato anche un effetto collaterale! Sono riuscito anche a valutare l'effetto INSIGHT (intuizione). Il programmare per riuscire a realizzare l'effetto richiesto (es. l'accensione di due led) ha stimolato i ragazzi e riporto quanto scritto nell'articolo:

III fase: INSIGHT (intuizione) dove ogni processo di apprendimento ha bisogno di una "energia di attivazione", altrimenti i contenuti restano lì fermi, senza contribuire alla costruzione delle competenze. In questa occasione l'allievo si rende indipendente nella risoluzione dei problemi.

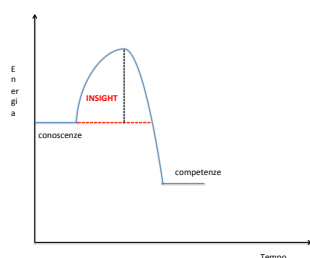


Figura 31 INSIGHT

Il catalizzatore di questa reazione, la scintilla che fa partire la reazione, dipende dalla dinamica affettiva e relazionale che si instaura nei gruppi di lavoro e dal fatto "funziona o non funziona?" e quindi dalla continua ricerca degli errori.



Figura 32terza fase REFREEZE il rinforzo con ARDUINO

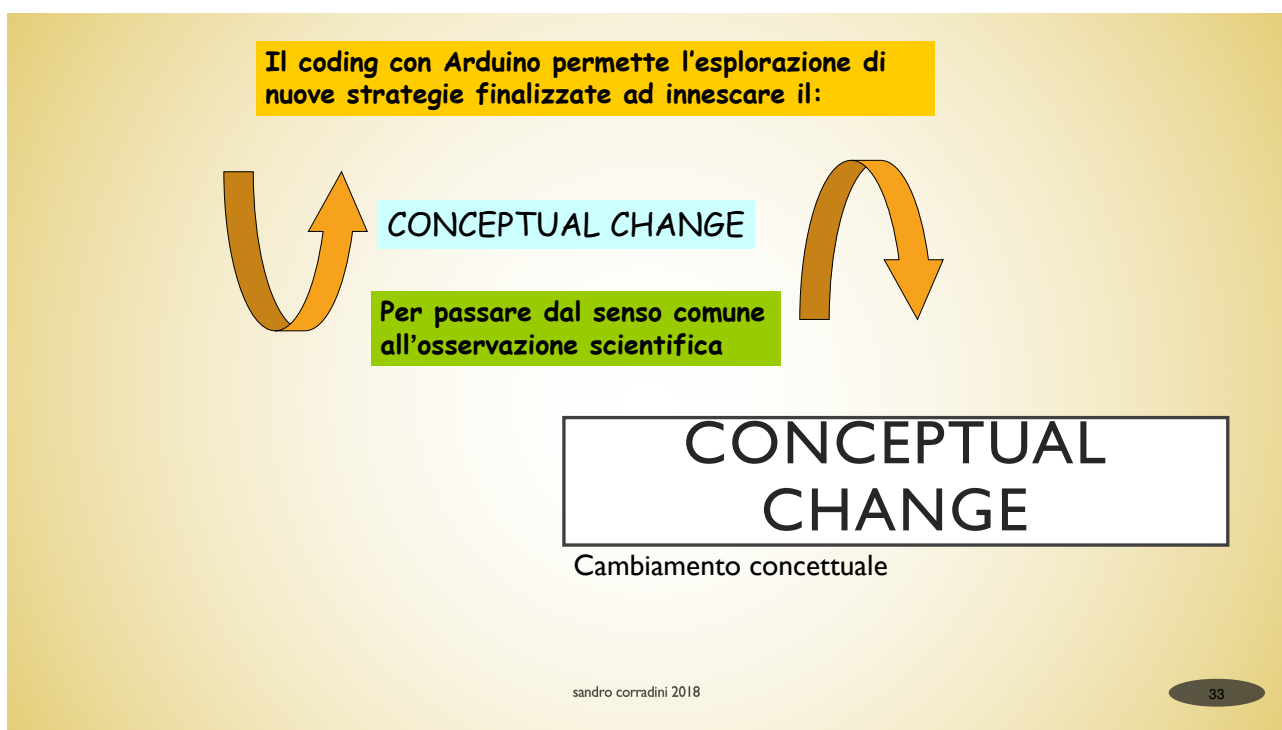


Figura 33 IL CONCEPTUAL CHANGE

Figura 1 Il regolo calcolatore prima calcolatrice RPN della HP	Figura 2 4
Figura 3 slide TIC n°1	7
Figura 4 Slide TIC n° 2	7
Figura 5 Slide TIC n° 3	8
Figura 6 I comandi LOGO per costruire un quadrato	9
Figura 7 Frontespizio del lavoro di Seymour Papert	11
Figura 8 Schermata di LogoBlocks (Andrew Begel) molto vecchia	14
Figura 9 Scratch 3	15
Figura 10 Papert e la prima turtle	16
Figura 11 Seymour Papert e i suoi mattoncini Lego con i quali costruiva i primi Robot	17
Figura 12 Foto del 1971: Marvin Minski, padre dell'AI, assieme a Seymour Papert	18
Figura 13 Children con la LOGO turtle di Papert	19
Figura 14 I primi laptop per il III mondo. La carica era a manovella	20
Figura 15 la scheda ARDUINO	24
Figura 16 Jeannette M. Wing e il pensiero computazionale	24
Figura 17 mappa Physical Computing	25
Figura 18 sensori e attuatori	26
Figura 19 ambiente IDE	27
Figura 20 BLINK è il programma di prova di ARDUINO	27
Figura 21 es. di progettazione con FRITZING	27
Figura 22 il modello di conceptual change.....	30
Figura 23 INSIGHT	31
Figura 24 Bambina reduce da una rissa	32
Figura 25 Einstein 1	35
Figura 26 Einstein 2	35
Figura 27 Conceptual Change Model	36
Figura 28 prima fase conceptual change UNFREEZE	37
Figura 29 prima fase UNFREEZE creare l'ambiente giusto	37
Figura 30 seconda fase CHANGE costruzione delle LOGIC GATES.....	38
Figura 31 INSIGHT	38
Figura 32terza fase REFREEZE il rinforzo con ARDUINO.....	39
Figura 33 IL CONCEPTUAL CHANGE.....	39