Guide per lo sviluppo professionale degli insegnanti

Collana diretta da
Carlo Petracca

a cura di
Enrico Nardelli

CODING E OLTRE:
l'informatica nella scuola

LISCIANISCUOLA

© Educational S.r.l.
Via Facii, Zona Ind.le S. Atto
64100 Teramo - ITALY

www.liscianiscuola.it

ISBN 978-88-7627-436-7

Prima edizione: settembre 2020

Stampato in Italia

Ristampe							Anno			
0	1	2	3	4	5	6	2020	2021	2022	2023

Indice

Introduzione di *Enrico Nardelli* ..... » 11

Capitolo I • Insegnamento dell'informatica nella scuola: cosa fare e perché
di *Enrico Nardelli*

1 Competenze digitali e competenze informatiche ..... » 16
2 Informatica e *coding*: quale relazione? ..... » 19
3 Informatica e pensiero computazionale ..... » 21
4 Informatica nella scuola: cenni sul panorama internazionale ..... » 26
5 Informatica nella scuola italiana ..... » 28
6 I concetti fondamentali dell'informatica ..... » 31
6.1 *Rappresentazione* ..... » 32
6.2 *Automa e linguaggio* ..... » 33
6.3 *Algoritmo* ..... » 34
6.4 *Sintesi* ..... » 35
7 Una proposta operativa per l'insegnamento dell'informatica nella scuola ..... » 36
7.1 *Ambito algoritmi* ..... » 39
7.2 *Ambito programmazione* ..... » 40
7.3 *Ambito dati e informazioni* ..... » 42
7.4 *Ambito consapevolezza digitale* ..... » 44
7.5 *Ambito creatività digitale* ..... » 45
8 Riflessioni finali ..... » 46
9 Riferimenti bibliografici ..... » 49

Capitolo II • Algomotricità: manipolare i fondamenti dell'informatica
di *Carlo Bellettini, Violetta Lonati, Dario Malchiodi, Mattia Monga, Anna Morpurgo*

1 Introduzione » 53
2 L'ascensore » 55
3 Wikipasta » 60
4 Rettangoli » 65
5 Robot umani » 70
6 Riflessioni meta-cognitive finali » 77
7 Riferimenti bibliografici » 79

Capitolo III • Cittadinanza digitale consapevole
di *Isabella Corradini, Francesco Lacchia, Enrico Nardelli*

1 Introduzione » 81
2 La guida alla cittadinanza digitale » 83
2.1 *Introduzione* » 83
2.2 *Scheda didattica n. 1 – Il mio quartiere digitale* » 85
2.3 *Scheda didattica n. 2 – Il potere delle parole* » 86
2.4 *Scheda didattica n. 3 – Super Cittadino Digitale* » 87
2.5 *Corrispondenza con la proposta CINI* » 88
3 I sistemi informatici » 89
3.1 *Introduzione* » 89
3.2 *Scheda didattica n. 4 – Come funzionano i computer* » 90
3.3 *Corrispondenza con la proposta CINI* » 92
4 Riferimenti bibliografici » 93

Capitolo IV • "Archeologia dell'informazione" nella scuola primaria
di *Diana Bitto, Claudio Mirolo*

1 Introduzione » 95
2 Concetti esplorati » 97
2.1 *Dati e informazioni* » 97
2.2 *Convenzionalità delle rappresentazioni* » 98
2.3 *Sintassi di una rappresentazione* » 99
2.4 *Proprietà operative di una rappresentazione* » 99
2.5 *Algoritmi* » 100
3 Esempi di unità didattiche » 101
3.1 *Elaborare dati quantitativi per mezzo di gettoni (Neolitico nel Medio-oriente)* » 102

3.2 *Memorizzare rispettando le regole sintattiche (Quipu incaici)* » 106
3.3 *Approfondimenti e materiali* » 110
4 Riflessioni conclusive » 110
5 Riferimenti bibliografici » 112

Capitolo V • Informatica senza e con computer nella Scuola Primaria
di *Michael Lodi, Renzo Davoli, Rebecca Montanari, Simone Martini*

1 Introduzione » 113
2 Le attività senza computer » 115
2.1 *I numeri binari con le monete* » 115
2.2 *La trasmissione dei caratteri con le luci (o i cartellini colorati)* » 117
2.3 *La ricerca del massimo con le bilance* » 118
2.4 *Domo: un maggiordomo "computazionale"* » 120
3 Attività con computer » 123
3.1 *Informatica creativa con Scratch* » 123
4 Conclusioni » 128
5 Riferimenti bibliografici » 130

Capitolo VI • Geografia collaborativa online
di *David Leoni, Alberto Montresor*

1 Introduzione » 131
2 Creazione mappe di carta » 134
3 Introduzione in classe » 135
4 Uscita e raccolta punti di interesse » 136
5 Navigazione mappe » 138
5.1 *OpenStreetMap* » 138
5.2 *OsmHydrant* » 141
5.3 *WheelMap* » 142
5.4 *Mappa Sentieri* » 142
6 Inserimento punti su Crowdmap » 143
6.1 *Registrazione in CrowdMap* » 143
6.2 *Creazione della mappa CrowdMap* » 143
6.3 *CrowdMap in classe* » 144
6.4 *Modificare OpenStreetMap* » 147
6.5 *Come proseguire* » 149
6.6 *Criticità e possibili sviluppi* » 150
6.7 *Corrispondenza con la proposta CINI* » 151
Ringraziamenti » 153

Capitolo VII • Due proposte per l'avvio alla programmazione nella scuola primaria
di *G. Barbara Demo*

1 Introduzione ... » 155
2 Attività senza calcolatore: BimboRobot ... » 156
2.1 *Fase 1* ... » 157
2.2 *Fase 2* ... » 159
2.3 *Fase 3* ... » 160
2.4 *Osservazioni sulle attività unplugged* ... » 161
3 Creare storie e raccontarle in Scratch ... » 162
3.1 *Fase 1. Introduzione all'ambiente* ... » 163
3.2 *Fase 2. Casi "di vita quotidiana" e loro trasposizione nell'ambiente di programmazione* ... » 164
3.3 *Fase 3. Storie originali, quiz, schede libro, giochi* » 168
3.3.1 *Evoluzione delle storie iniziali in storie originali* ... » 168
3.3.2 *Dalle storie ai quiz e ai giochi* ... » 169
3.4 *Caratteri trasversali alle fasi* ... » 170
3.4.1 *Interdisciplinarietà* ... » 170
3.4.2 *Imparare ad imparare* ... » 171
3.5 *Fase 4. Le attività digitali collettive sono esercizio di soft skill* ... » 172
3.6 *Non soltanto programmazione* ... » 176
4 Obiettivi della Proposta di Indicazioni Nazionali per l'Informatica ... » 178
5 Riferimenti bibliografici ... » 179
6 APPENDICE 1 - Primo Ciclo classi prima, seconda e terza ... » 180
7 APPENDICE 2 - Primo Ciclo classi quarta e quinta ... » 181

Capitolo VIII • Introdurre il pensiero computazionale nella scuola primaria
di *Antonio Piccinno, Veronica Rossano, Antonietta Lanza*

1 Introduzione ... » 183
2 Descrizione degli obiettivi didattici ... » 184
2.1 *Acquisire il concetto di astrazione* ... » 185
2.2 *Conoscere il computer* ... » 186
2.3 *Acquisire un linguaggio come mezzo di comunicazione* ... » 186
2.4 *Istruire il computer ad eseguire un compito* ... » 187
2.5 *Verificare un programma* ... » 187

3 Descrizione dell'esperienza ... » 188
3.1 *Esperienza condotta nell'a.s. 2015/2016* ... » 189
3.1.1 *Specificità per ciascun grado* ... » 191
3.1.2 *Considerazioni finali sull'esperienza condotta* ... » 191
3.2 *Esperienza condotta nell'a.s. 2016/2017* ... » 192
3.2.1 *Specificità per ciascun grado* ... » 192
3.2.2 *Considerazioni finali sull'esperienza condotta* ... » 193
3.3 *Esperienza condotta nell'a.s. 2017/2018* ... » 193
3.3.1 *Specificità per ciascun grado* ... » 194
3.3.2 *Considerazioni finali sull'esperienza condotta* ... » 195
4 Manuale operativo ... » 196
4.1 *Astrazione* ... » 196
4.1.1 *Classi Prime - Workflow* ... » 197
4.1.2 *Classi Seconde - Workflow* ... » 198
4.2 *Algoritmo e programma* ... » 199
4.2.1 *Classi Prime - Workflow* ... » 199
4.2.2 *Classi seconde e terze - Workflow* ... » 202
4.2.3 *Classi quarte e quinte - Workflow* ... » 202
5 Conclusioni ... » 204
Ringraziamenti ... » 205

Capitolo IX • Il pensiero computazionale in classe prima: un lavoro di équipe multidisciplinare
di *Tullio Vardanega, Barbara Arfè, Elisabetta Pagani*

1 Introduzione ... » 207
2 Progetto formativo ... » 208
3 Il percorso di coding per le classi prime: strumenti e organizzazione ... » 209
4 Scheda tecnica dell'attività didattica ... » 210
4.1 *La selezione degli esercizi* ... » 210
4.2 *Criteri di selezione* ... » 210
4.3 *Organizzazione: strumenti, logistica e programma* ... » 211
5 Rapporto con gli obiettivi di apprendimento del PINI » 216
6 Considerazioni conclusive ... » 217
7 Riferimenti bibliografici ... » 220

Capitolo X • Coding e Robotica

di *Fabio Sorrentino, Lucio Davide Spano, Riccardo Scateni*

1 Obiettivi didattici ... » 221
2 Osservazioni e suggerimenti preliminari ... » 222
2.1 *Dotazioni informatiche scolastiche* ... » 222
2.2 *Omogeneità della classe e disturbi specifici di apprendimento* ... » 222
2.3 *Programmazione individuale (Adatto alla configurazione L1* ... » 224
2.4 *Pair programming o programmazione di coppia (Adatta alle configurazioni L1 e L2)* ... » 224
2.5 *Pause e premi* ... » 225
3 Apprendere con i dispositivi elettronici ... » 226
3.1 *Creazione delle classi e dell'ambiente di apprendimento su code.org* ... » 226
3.2 *Competizione tra bambini (adatta a tutte le configurazioni)* ... » 227
4 Dal coding alla robotica educativa ... » 228
4.1 *Simulazione robot* ... » 229
4.2 *Robot educativi* ... » 230
4.3 *Costruzione robot* ... » 230
4.4 *Primi passi con mBot* ... » 231
4.4.1 *Il primo script* ... » 231
4.4.2 *Il primo collegamento* ... » 232
4.4.3 *Caricare script basati su sensoristica* ... » 233
4.4.4 *Divertirsi con mBlock* ... » 233
5 Conclusioni ... » 234
6 Riferimenti bibliografici ... » 238

Appendice
Traguardi ed obiettivi didattici della proposta CINI per l'insegnamento dell'Informatica nella Scuola ... » 239

Introduzione

Enrico Nardelli *

Il tema dell'inserimento dell'informatica nella scuola è in questi anni ampiamente dibattuto.

Negli USA è attiva dal 2015 l'iniziativa *Computer Science for All*, che sostiene l'inserimento dell'informatica nell'istruzione scolastica alla pari delle altre discipline scientifiche e tecnologiche. Nel Regno Unito già dal 2014-15 la disciplina *Computing* è obbligatoria in tutte le scuole ed altri Paesi, in Europa e nel mondo, si sono avviati su questa strada.

In Italia, il C.I.N.I. (Consorzio Nazionale Interuniversitario per l'Informatica) ha preparato nel dicembre 2017 una proposta per l'insegnamento dell'informatica nella scuola italiana, (CINI 2017), redatta sul modello di analoghi documenti MIUR che riportano indicazioni curricolari. Tale proposta descrive nel modo più articolato possibile la visione culturale e scientifica della comunità universitaria italiana sull'insegnamento dell'informatica nella scuola. L'elenco di dettaglio di traguardi di competenza ed obiettivi di apprendimento della proposta è ripreso, per comodità di lettura, nell'Appendice.

L'obiettivo di questo libro è costituire un quaderno operativo per la scuola che raccoglie e descrive esperienze già sviluppate di insegnamento dell'informatica nella primaria, presentandole nel contesto della suddetta proposta. Si tratta quindi di una prima mappa che offriamo ai docenti per iniziare a navigare il territorio dell'insegnamento dell'informatica nella primaria, costruita sulla base delle esplorazioni che – con i colleghi della comunità universitaria e con l'indispensabile aiuto dei tanti docenti di scuola che hanno collaborato con noi con entusiasmo e dedizione – abbiamo condotto in questi anni.

* Università di Roma "Tor Vergata", Direttore Laboratorio Nazionale C.I.N.I. "Informatica e Scuola", Presidente di Informatics Europe

Ogni capitolo tratteggia un'esperienza concreta e sperimentata sul campo, in stretta collaborazione con gli insegnanti, di cui si presentano gli obiettivi di apprendimento fornendo poi una descrizione operativa (destinatari, materiali, metodologie, riferimenti online, …) che ne renda possibile la sua ripetizione in classe (eventualmente modificata sulla base delle esigenze didattiche). Viene poi offerta un'indicazione analitica di come l'esperienza riportata indirizzi la proposta sopra citata, concludendo il capitolo con una bibliografia di approfondimento.

Il primo capitolo del manuale, a cura del sottoscritto, offre un inquadramento generale del tema della formazione informatica nella scuola. Abbiamo assistito, infatti, in questi ultimi anni, a molte iniziative in merito ed è fondamentale fare chiarezza su quanto proposto.

Il capitolo 2 è stato preparato dai colleghi dell'università di Milano e presenta ben quattro attività di cui due toccano i temi della rappresentazione di dati e informazioni, mentre le altre due sviluppano alcuni concetti elementari relativi agli algoritmi e alla programmazione informatica.

Il terzo capitolo, a cura dell'università di Roma Tor Vergata, affronta il tema di consapevolezza digitale, proponendo tre esperienze relative all'impatto delle tecnologie digitali sulle relazioni personali e sociali, e discutendo brevemente una quarta esperienza relativa alla comprensione del funzionamento dei dispositivi digitali.

Il capitolo 4, a cura dell'università di Udine, si focalizza sulla comprensione dei concetti di base relativi agli algoritmi ed alla rappresentazione dei dati, presentando due esperienze basate su oggetti fisici "provenienti dal passato", che offrono quindi lo spunto anche per un eventuale allargamento multidisciplinare del discorso, e fornendo riferimenti per ulteriori percorsi.

Il quinto capitolo, realizzato dall'università di Bologna presenta diversi percorsi caratterizzati dall'alternanza tra attività svolte senza l'utilizzo di dispositivi digitali (quattro in tutto) e attività da svolgere al computer (tre). I percorsi sono relativi a concetti elementari degli algoritmi, della programmazione informatica e della rappresentazione di dati e informazioni.

Il capitolo 6, curato dall'università di Trento, è un esempio di uso trasversale dell'informatica, applicata in questo caso alla geografia, che al tempo stesso contribuisce allo sviluppo dei traguardi di creatività e di responsabilità. Propone infatti una guida per realizzare unità di apprendimento di geografia basate sulla cartografia liberamente accessibile online.

Il settimo capitolo, realizzato dall'università di Torino presenta in dettaglio due esperienze per l'avvio alla programmazione informatica nella scuola primaria. La prima viene svolta in modalità tradizionale, cioè senza l'utilizzo di dispositivi tecnologici, mentre la seconda richiede l'uso del computer. Sono anche forniti spunti per estensioni interdisciplinari delle attività e per ulteriori approfondimenti, sia all'interno dell'informatica che di carattere trasversale.

Il capitolo 8, preparato dai colleghi dell'università di Bari, discute tre esperienze, svolte senza l'utilizzo di dispositivi tecnologici, mirate allo sviluppo del cosiddetto "pensiero computazionale", cioè di quel modo di pensare sviluppato dallo studio e dall'esercizio dell'informatica, che viene discusso in dettaglio nel corso del capitolo 1 del manuale.

Il capitolo 9, realizzato dall'università di Padova, affronta l'ambito della programmazione informatica (il cosiddetto *coding*) come strumento per lo sviluppo del pensiero computazionale, presentando un'esperienza di facile realizzazione.

Il decimo capitolo, preparato dai colleghi dell'università di Cagliari, è focalizzato su attività elementari di programmazione informatica miranti al controllo di un semplice robot e allo sviluppo delle competenze trasversali dei bambini.

Conclude il volume l'Appendice, che riporta integralmente traguardi di competenza ed obiettivi di apprendimento della proposta per l'insegnamento dell'informatica nella scuola italiana, preparata dal CINI.

La speranza di questo libro è di offrire un contributo utile ad allargare la comunità delle scuole che si sono avviate sulla sperimentazione delle varie componenti di questa proposta e, in tal modo, contribuire al suo miglioramento e alla sua diffusione.

Se avete commenti o suggerimenti potete scrivere al sottoscritto o direttamente agli autori dei singoli capitoli.

Enrico Nardelli*

Capitolo I
Insegnamento dell'informatica nella scuola: cosa fare e perché

Nel 1931 Albert Einstein riteneva che lo scopo primario della scuola fosse quello di «sviluppare in modo bilanciato la persona», affinché avesse «una generale abilità nel pensare in modo indipendente e giudicare criticamente» (Einstein 2006). Così, secondo Einstein, il bambino sarebbe stato in grado di «trovare sicuramente la propria strada e in grado di adattarsi al progresso e ai cambiamenti». Già quasi un secolo fa era chiara la missione che la scuola deve adempiere: preparare i ragazzi ad affrontare l'unica costante della vita, quella del cambiamento, in modo che potessero agire da cittadini consapevoli in qualunque futura società.

Essere cittadini consapevoli è infatti fondamentale per una società democratica e la consapevolezza passa attraverso l'istruzione. Solo con l'istruzione obbligatoria il suffragio universale attua davvero le sue promesse.

È ormai chiaro a tutti che stiamo attraversando un cambiamento sociale di natura epocale. La società industriale, che ha soppiantato la precedente società agricola, sta trasformandosi in una società digitale. Ci sono sempre più dati digitalizzati e una sempre maggior presenza di sistemi informatici che, sulla base di questi dati digitali, automaticamente fanno qualcosa per noi, anche senza che ne siamo a conoscenza.

Si sente spesso dire che, a fronte di questa evoluzione della società, la scuola dovrebbe cambiare radicalmente. Io, personalmente, sono un po' perplesso, ma non perché ritengo che la scuola non

* Università di Roma "Tor Vergata", Direttore Laboratorio Nazionale C.I.N.I. "Informatica e Scuola", Presidente di Informatics Europe.

possa migliorare utilizzando in modo sapiente la tecnologia digitale. Quest'ultima, se ben usata, può certamente offrire ai docenti ed ai ragazzi un più facile accesso alle informazioni e più efficaci forme di comunicazione e collaborazione. Dubito però che siano utili i cambiamenti radicali quando le persone e le relazioni sociali sono sempre le stesse da millenni. Siamo fatti della stessa materia e siamo mossi dagli stessi desideri, aspirazioni, impulsi ed emozioni di quando abbiamo conquistato il linguaggio, centinaia di migliaia di anni fa. Le dinamiche sociali relazionali magari adesso si estrinsecano attraverso dispositivi digitali, ma sono sempre della stessa natura. Per questo è indispensabile non trascurare lo studio della letteratura, della storia e in generale delle materie umanistiche, perché comunque ci aiutano a capire ciò che, in un flusso di cambiamento continuo, non è invece mai cambiato.

Chiarito questo, rimane certamente il problema, per la scuola, di come preparare i ragazzi ad essere cittadini di una società digitale (Nardelli 2019c). Le risposte che vengono date sono diverse e richiedono di essere analizzate.

1 Competenze digitali e competenze informatiche

Una soluzione che viene proposta è far sì che i ragazzi acquisiscano **competenze digitali**. All'altro estremo vi è la proposta di insegnare nella scuola l'**informatica**. Qualcuno potrebbe chiedersi: ma non sono la stessa cosa? La risposta è negativa, anche se molti nella scuola e nella società non ne sono consapevoli. Nel mondo della scuola, in particolare, c'è confusione tra i due aggettivi "informatiche" e "digitali" ed i relativi sostantivi. Ma non è affatto colpa dei docenti, né cercare chi sia il colpevole ha una qualche utilità. Nel corso degli ultimi 30 anni nella scuola si è parlato di competenze informatiche e di competenze digitali intendendo di volta con l'uno o l'altro termine: la capacità di programmare in Pascal, di usare Word, di scrivere e-mail, di fare *coding*, di usare consapevolmente i social, e così via. Non è sorprendente quindi che su questi temi non ci sia chiarezza.

È quindi importante ricordare l'etimologia delle due parole: "digitale" si riferisce alla rappresentazione di un dato mediante un simbolo numerico, mentre "informatico" si riferisce alla capacità

di elaborazione automatica dei dati, resa possibile dai metodi e dalle teorie dell'informatica che è una disciplina scientifica. Rappresentare dati attraverso simboli numerici non è una novità degli ultimi decenni. Lo facevano i Babilonesi per il calcolo delle orbite astronomiche e gli antichi Egizi per il calcolo delle superfici dei terreni, ognuno con il loro sistema di rappresentazione. La novità è la possibilità di elaborare queste rappresentazioni in modo automatico (cioè l'informatica), come se fosse all'opera un gigantesco complicatissimo orologio. È chiaro a tutti che l'orologio è un mero esecutore meccanico di un calcolo pensato e progettato dall'uomo, e che un orologio non "sa" cosa siano e cosa rappresentino e a cosa servano i diversi ingranaggi che lo costituiscono.

Ma la realizzazione dei computer consente di avere, per la prima volta nella storia dell'umanità, un sistema automatico che, manipolando simboli di cui ignora il significato secondo istruzioni di cui ignora il significato, trasforma dati che hanno significato per l'uomo. Si ottiene, quindi, una "**macchina cognitiva**", che realizza, cioè, operazioni di natura cognitiva. Si tratta di una vera e propria rivoluzione, la "rivoluzione informatica", che ho caratterizzato come la **terza rivoluzione dei rapporti di potere**, perché per la prima volta nella storia dell'umanità funzioni cognitive non elementari vengono svolte da macchine (Nardelli 2018a, 2019a). Una macchina cognitiva è "conoscenza pronta ad entrare in azione" (*actionable knowledge*, in inglese). Questa terza rivoluzione rompe il potere dell'intelligenza umana, realizzando artefatti che possono meccanicamente replicare attività cognitive caratteristiche dell'uomo.

Quali competenze servono, quindi, per formare i cittadini di questa società che è pervasa da macchine digitali, o meglio, da macchine cognitive? Quelle digitali o quelle informatiche? Possiamo dare una risposta rifacendoci all'etimologia sopra ricordata. Le competenze digitali sono tutte quelle relative all'uso di tecnologie che manipolano dati in forma numerica, cioè di tutte le tecnologie basate su computer: **sono quindi competenze operative**. Invece le competenze informatiche sono quelle attinenti ai princìpi e ai metodi dell'informatica: **sono pertanto competenze scientifiche**.

Nella scuola quindi servono entrambe, ma con scopi diversi.

I documenti dell'Unione Europea in proposito non sono purtroppo chiari in merito, mescolando gli aspetti operativi e quelli scientifici. Non è una mia opinione. Leggendo, ad esempio, quanto raccomandato dal Consiglio dell'Unione Europea (EurLex 2018), si vede

che la programmazione viene considerata una competenza digitale quando si tratta invece di una parte fondamentale dell'informatica, quindi di una competenza scientifica. Considerare la programmazione informatica come una competenza operativa invece che come uno degli elementi fondamentali dell'**informatica, che è disciplina scientifica,** equivale a degradare l'aritmetica ad una competenza operativa (quella della tabellina pitagorica). Negli Stati Uniti, in UK e in molti altri paesi avanzati la differenza è invece chiara.

In questi paesi si considera, correttamente, l'attuale società digitale come un'evoluzione della società industriale. Quest'ultima fu il risultato della trasformazione della società agricola sotto la spinta della rivoluzione industriale. La società digitale è ancora una società di macchine, ma che manipolano dati digitali.

Da un punto di vista fisico possiamo chiamarle "macchine digitali", perché "digitale" è un termine ombrello, che fa riferimento a tutte le tecnologie, basate su computer, che manipolano dati in forma digitale. Da un punto di vista concettuale, però, è più corretto chiamarle "macchine cognitive", per i motivi sopra esposti.

Per preparare i cittadini alla società industriale, nei due secoli passati, non sono state date agli studenti competenze operative sui macchinari industriali, ma sono state inserite nelle scuole le discipline scientifiche che ne spiegavano i princìpi scientifici alla base (matematica, fisica, chimica, …). Allo stesso modo, per preparare i cittadini a comprendere appieno la società digitale serve inserire nella scuola la formazione sulla disciplina scientifica che la spiega e che l'ha resa possibile: l'informatica (Nardelli 2018b).

Questo non nega che nella scuola servano anche le competenze digitali, il livello operativo. È bene averle, e sono trasversali a qualunque disciplina, perché in ogni materia il docente può avvantaggiarsi nella sua attività didattica e di gestione dal possesso di un buon livello di competenze digitali.

Quindi certamente la scuola deve formare le competenze digitali degli studenti (livello operativo) e deve assicurarsi che le posseggano anche i loro docenti, di ogni materia. Però è assolutamente necessario fornire un'istruzione informatica agli studenti (livello scientifico), per renderli in grado di partecipare in modo attivo e informato alla società digitale.

2 Informatica e *coding*: quale relazione?

Nel dibattito tra le competenze informatiche e quelle digitali si è negli ultimi anni prepotentemente inserito il tema del *coding*, la programmazione informatica, di cui si sente parlare sempre più frequentemente. Nella sfera mediatica molti parlano dell'importanza di insegnarlo a tutti gli studenti, descrivendolo come il linguaggio indispensabile per operare nella società digitale. Si dice: «il *coding* è il nuovo inglese», messaggio certamente semplice, ma che rispetto agli obiettivi di formazione dei cittadini della società digitale è inadeguato e fuorviante (Nardelli 2017a).

È inadeguato perché dovrebbe riferirsi all'apprendimento dell'informatica e non del solo *coding*, cioè della programmazione informatica che è la parte più immediata e operativa. Ed è per questo fuorviante perché, se preso alla lettera, senza collocarlo nel giusto scenario, rischia di creare i nuovi proletari del futuro.

Chiarisco subito che questa mia **osservazione non vuole affatto svilire il *coding*, ma inserirlo in una corretta prospettiva**.

L'informatica è la disciplina scientifica che spiega i meccanismi base del funzionamento e della costruzione di queste macchine digitali che sono ormai parte integrante della nostra vita[1]. Come ben noto, in qualunque settore tecnologico, la realizzazione di un macchinario, tranne il caso di dispositivi semplici o dell'opera di un creatore geniale, non avviene semplicemente cominciando a mettere insieme dei pezzi.

L'esperienza di secoli ha mostrato che è necessario conoscere le basi scientifiche del dominio in cui si vuole creare la macchina, poi sapere come elaborare un progetto ingegneristico che porti alla specifica delle caratteristiche di ciò che si vuole ottenere ed alla definizione di come può essere fabbricato; infine, si passa alla realizzazione. Non c'è proprio altra strada poi, se si vuole industrializzare questo processo produttivo e renderlo ripetibile e di qualità garantita.

Parlare solo di *coding* è come dare scatoloni di sbarre, bulloni e piastre di ferro a qualcuno e poi dirgli di costruire qualcosa. Finché questo «qualcosa» è piccolo e semplice non ci sono problemi, ma

[1] https://www.slideshare.net/enriconardelli1/rivoluzione-informaticacittadinanzadigitale

se si vuole edificare un vero ponte serve qualcosa di più che la semplice abilità nel mettere insieme i pezzi.

Con la programmazione informatica, cioè il *coding*, si hanno a disposizione gli equivalenti digitali di sbarre, bulloni e piastre di ferro e non ci sono problemi nel costruire cose piccole e semplici. Anzi, è molto più facile che per ogni altra macchina, sia perché questi componenti di base sono immateriali, non pesano e non sono ingombranti, sia perché c'è a disposizione un gran numero di componenti anche abbastanza sofisticati che possono essere combinati senza sforzo. Negli ambienti moderni di *coding*, qualunque bambino può - ad esempio - realizzare applicazioni che adesso sono semplici ma che cinquant'anni fa erano alla portata solo di centri di avanguardia tipo la NASA: gli astronauti sono sbarcati sulla luna usando un sistema informatico di complessità inferiore a quelli che ci sono negli oggetti "smart" intorno a noi oggi. Nessun bambino, invece, può fabbricare da solo una Torre Eiffel, neanche oggi, pur avendo a disposizione tutte le sbarre, bulloni e piastre che vuole.

Vediamo adesso perché pensare che basti il *coding* per formare i cittadini della società digitale è fuorviante.

Perché da sempre, nella storia dell'umanità, **il livello operativo della forza lavoro è stato quello più debole nei contrasti sociali e soggetto all'evoluzione tecnologica**. Negli ultimi secoli, successivamente alla transizione da una società agricola ad una industriale, si è capito che l'istruzione è fondamentale per consentire a tutti una vita lavorativa soddisfacente nel tempo. Ed in effetti negli ultimi decenni, in tutti i Paesi avanzati, si insiste su durata e qualità della formazione scolastica e universitaria. Nell'Italia degli anni '50 e '60 i figli delle classi meno abbienti sono stati portati a studiare e laurearsi proprio in quest'ottica, che rimane quella corretta per la crescita sociale ed economica.

Anche nell'odierna fase di transizione da società industriale a società digitale i problemi di base del lavoro rimangono. Anzi, con l'accelerazione del progresso tecnologico stanno già emergendo. La sofisticazione di quelle tecniche informatiche che vanno generalmente sotto il nome di "Intelligenza Artificiale" già rende possibile la produzione automatica di programmi informatici. Per ora questo accade in domìni ancora limitati e in misura molto ridotta. Ma è una tendenza inarrestabile. È chiaro quindi che **indicare solo il *coding* come obiettivo dell'istruzione in materia di competenze digitali rischia di mettere su una cattiva strada chi non conosce questi scenari**. Senza fornire ai ragazzi una formazione

a largo spettro sui vari aspetti dell'informatica, prepareremo solo "operai digitali" che saranno i primi ad essere espulsi dal mercato del lavoro (Nardelli 2018c).

In un quadro in cui questa visione strategica sia chiara si può benissimo parlare del *coding*, perché dopo tutto scrivere programmi informatici è un'attività fondamentale per imparare l'informatica. Così come saper fare le operazioni aritmetiche o calcolare l'area di figure geometriche sono passi fondamentali dell'apprendimento della matematica. Va quindi benissimo "fare *coding*", soprattutto poi se questo aiuta ad avvicinare ragazze e ragazzi ad una materia che può dar loro molte soddisfazioni. Ma facciamo attenzione a come viene comunicato il messaggio!

Inoltre, sempre per rimanere nel paragone con la Matematica, la moltiplicazione è una parte dell'aritmetica e quest'ultima è, insieme a geometria, algebra, probabilità, analisi e via dicendo, una parte di tutta la Matematica. Analogamente il *coding*, cioè la programmazione, è solo una parte del processo di sviluppo del software (che include anche analisi, progetto, verifica e correzione di errori) ed il software è solo una parte dell'Informatica, insieme agli algoritmi, ai linguaggi di programmazione, alle architetture di calcolo, al calcolo distribuito, alla grafica, all'intelligenza artificiale e molto altro.

3 Informatica e pensiero computazionale

In questa discussione sul come preparare nella scuola i futuri cittadini digitali, è emerso negli ultimi tempi il termine "pensiero computazionale", traduzione del termine inglese *computational thinking*. Questa espressione era stata resa nota e già usata negli anni '50 da Seymour Papert nel suo lavoro sull'apprendimento della matematica e da diversi scienziati che realizzavano esperimenti mediante simulazioni al computer dei fenomeni oggetto di studio[2]. È stata però resa estremamente popolare da un citatissimo articolo di Jeannette Wing, la quale sinteticamente esponeva la posizione che «il pensiero computazionale è il modo di pensa-

[2] Per una discussione storica della diffusione del termine si veda (Tedre, Denning 2016).

re di un informatico» (Wing 2006). Dal momento che non aveva fornito una precisa caratterizzazione del pensiero computazionale, molti articoli successivi hanno cercato di formularla. Molti educatori hanno cavalcato questo termine usandolo per denotare le competenze da far acquisire a ragazzi ed insegnanti di una scuola davvero digitale.

Il punto è che un approccio di questo genere è inutile ed errato (Nardelli 2019b). Il pensiero computazionale, in quanto modo di pensare sviluppato da parte di chi ha studiato e pratica l'informatica, ha la stessa natura del "pensiero matematico", posseduto da chi ha studiato e pratica la matematica, o del "pensiero fisico" di un fisico o del "pensiero biologico" di un biologo. Il matematico affronta i suoi problemi in termini dei concetti fondamentali per la sua disciplina (quantità, relazioni, strutture, …) così come il fisico (massa, forza, campi, …) o il biologo (cella, organismo, metabolismo, …) nei loro campi. Per l'informatico i concetti fondamentali sono l'algoritmo, il programma, l'automa, ed è su questi concetti e sulla disciplina che deve essere centrata l'istruzione, non sul "pensiero computazionale". Questo è ciò che accade a scuola per le altre materie: si insegna matematica, fisica e biologia, non "pensiero matematico" o "pensiero fisico" o "pensiero biologico".

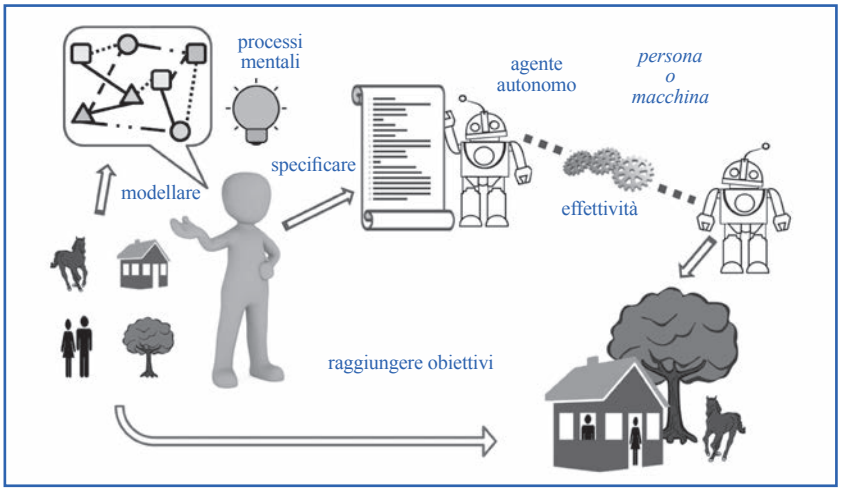
L'espressione "pensiero computazionale" può essere utile per far capire che quello che si vuole insegnare nella scuola non sono tanto le competenze operative e strumentali quanto quelle scientifiche. Nella matematica non si insegna la tavola pitagorica perché il fine è saper fare i conti. Sapere a memoria che 3 x 2 = 6 o 12 : 4 = 3, serve a far capire che, se 3 bambini hanno 2 caramelle ciascuna, il numero totale di caramelle si ottiene con la moltiplicazione, mentre se 12 biscotti devono essere distribuiti a 4 bambini il numero di biscotti per bambino si ottiene con la divisione. Non viene quindi tanto insegnato uno strumento operativo, quanto una chiave di comprensione della realtà ("il pensiero matematico"). Lo stesso deve accadere per l'informatica. Parlare di "pensiero computazionale" serve a sottolineare che stiamo parlando dei princìpi e dei metodi, del nucleo concettuale della disciplina. Non ha senso pensare di insegnare il "pensiero computazionale" mentre ha senso insegnare informatica in modo da far sì che i ragazzi, attraverso il suo studio e la sua pratica, assumano questa prospettiva di pensiero.

A questo proposito è opportuno puntualizzare che una "definizione" assai diffusa del pensiero computazionale lo caratterizza come "un processo mentale per risolvere problemi (*problem solving*)

seguendo metodi e strumenti specifici". Questa definizione omette purtroppo un elemento fondamentale che distingue il pensiero computazionale, cioè il modo con cui gli informatici affrontano i problemi, da quello di altre discipline scientifiche, per esempio la matematica. Anch'essa risolve problemi, da millenni, ed è importante spiegare qual è «la differenza che fa la differenza» (Bateson 1972) tra il *problem solving* della matematica e quello dell'informatica. La differenza che conta è che si ha una soluzione matematica ad un problema *quando la soluzione è una formula che definisce la risposta al problema stesso*. Si ha invece una soluzione informatica ad un problema *quando la soluzione è un processo che computa la risposta*. Il termine "processo" indica un **algoritmo** espresso in un **linguaggio di programmazione** ed eseguito da un **automa** (o **esecutore** o **agente**).

In aggiunta a questa differenza tra matematica e informatica ("formula che definisce la risposta" contro "processo che computa la risposta") vi è anche l'importante differenza nella natura del destinatario dell'attività di risoluzione di problemi. Nel caso della matematica è un altro matematico, o in generale una persona, nel caso dell'informatica è un dispositivo, detto esecutore o agente. Osserviamo che il ruolo dell'agente può essere svolto anche da un essere umano, cui però si richiede di comportarsi in modo completamente meccanico, limitandosi strettamente ad eseguire le istruzioni fornite. Questa differenza, enunciata in (Nardelli 2016a), ripresa in (Nardelli 2016b) ed approfondita in (Lodi et al. 2017) e (Nardelli 2019b) fa sì che l'informatica apporti, rispetto alla matematica, un mutamento radicale di punto di vista. Il cambiamento di paradigma concettuale è costituito dal *passaggio dal risolvere i problemi al far risolvere i problemi*. Senza la presenza dell'esecutore e la sua capacità di agire meccanicamente per eseguire le istruzioni ricevute, non c'è informatica, ma solo matematica, che infatti ha risolto problemi per millenni.

Se proprio si vuole fornire una definizione, con l'avvertenza di usarla solo per spiegare la novità apportata dall'informatica nel panorama delle scienze, si può usare quella che ho introdotto in (Nardelli 2019b) e (Lodi et al. 2017): «l'insieme dei processi mentali usati per modellare scenari e specificare i modi mediante i quali un agente che processa informazioni (esecutore) può operare in modo effettivo all'interno della situazione stessa per raggiungere uno o più obiettivi forniti dall'esterno». Tale definizione è illustrata dalla figura seguente.



(Figura da Lodi et al. 2017)

Questa caratterizzazione mette bene in luce da un lato l'importanza intrinseca della disciplina, affine alla matematica nel fatto che si occupa di risolvere i problemi e che studia leggi definite dall'uomo che possono essere dimostrate invece che leggi di natura della cui esattezza non si può mai essere assolutamente certi (Knuth 1974). Dall'altro indica anche quella che ho chiamato la sua "grande bellezza" (Nardelli 2017b) cioè il fatto che le sue "astrazioni", cioè i suoi "modelli", sono eseguibili in modo meccanico. Questo vuol dire che i modelli informatici possono essere animati, cioè si può "dar loro vita e vedere che accade" senza doverne ogni volta costruire una nuova rappresentazione fisica. Anche nelle altre discipline si può dar vita alle astrazioni mediante oggetti fisici che esprimono i fenomeni modellati, sostanziando così la loro rappresentazione espressa sotto forma di modello logico-matematico. Ma per ogni modello va realizzato ad hoc un insieme di oggetti. Questo valore aggiunto, cioè l'eseguibilità diretta, è posseduto solamente dall'informatica e la rende una disciplina utile, oltre che in sé e per sé, anche per la comprensione delle altre materie, perché offre un nuovo, complementare ed utile punto di vista per la descrizione dei fenomeni naturali ed artificiali (Nardelli 2017c). E ribadisco "complementare", quindi non in sostituzione degli approcci caratteristici di ogni disciplina, ma in aggiunta ad essi.

Concludo con alcune riflessioni relative alle critiche che ogni tanto si levano a proposito dell'espressione "pensiero computazionale" rimandando a (Nardelli 2017d) per ulteriori approfondimenti. La critica più diffusa è "insegnando agli studenti il pensiero computa-

zionale" si spengono le capacità di giudizio degli studenti trasformandoli da teste pensanti in macchine calcolanti. Ma così facendo si dimostra anche una carente conoscenza del reale significato di tale espressione. Formare al pensiero computazionale, infatti, non vuol dire insegnare agli studenti a pensare in modo meccanico ma **sviluppare negli studenti quelle capacità logico-razionali che sono alla base di qualunque attività di critica**. Quindi è esattamente l'opposto di ciò che viene contestato!

Il suo valore formativo è per certi versi assimilabile a quello che ha fatto ritenere, in modo continuativo almeno a partire dal Rinascimento, lo studio della geometria un passo essenziale nella formazione di un cittadino istruito. Per lo studente che avesse più tardi approfondito materie scientifiche o ingegneristiche esso costituiva un utile allenamento preparatorio, ma per coloro che si fossero dedicati a giurisprudenza o medicina la sua rilevanza educativa era basata sulla sua architettura logico-deduttiva.

Studiando la dimostrazione di un teorema di geometria lo studente viene formato ad analizzare quali siano le basi su cui esso è fondato (gli assiomi) e a valutare se assumendo come vere certe premesse (le ipotesi) ne discendono necessariamente certe conclusioni (le tesi). Questo allenamento di base è essenziale per prepararsi a capire, ad esempio, da avvocato o giudice se una persona è colpevole o meno, o da medico di quale malattia sia affetta.

Allo stesso modo, sviluppando un programma informatico, che è l'esempio concreto più immediato di pensiero computazionale, lo studente è costretto a riflettere se questo programma, cioè il piano di lavoro che prepara per l'esecutore, metterà quest'ultimo in grado di raggiungere l'obiettivo desiderato. È importante sottolineare che l'esecutore automatico di un programma informatico è un meccanismo puro, sprovvisto di qualunque intelligenza che possa trarlo d'impaccio in caso di difficoltà che non siano state previste dal suo programmatore. Il fatto che tale esecutore automatico sia, in termini umani, così stupido da dovergli spiegare ogni cosa in dettaglio è una palestra fondamentale per allenare lo studente a sviluppare una capacità utile per qualunque futuro lavoro. Qualunque sia, egli dovrà guidare o comunque interagire con altre persone, che sono esecutori intelligenti, per conseguire i suoi obiettivi. Aver fatto pratica in condizioni più difficili costituisce una preparazione altamente rilevante.

Questo valore formativo dell'educazione al pensiero computazionale è riconosciuto in tutto il mondo, anche al di là dell'immediato

impatto in termini occupazionali (che non è comunque da sottovalutare, visto che la maggior parte dei laureati triennali in Informatica ed Ingegneria Informatica trova subito lavoro).

4 Informatica nella scuola: cenni sul panorama internazionale

Spero di aver argomentato con sufficiente chiarezza il perché nella scuola sia necessario introdurre lo studio dell'informatica come disciplina scientifica. Non del pensiero computazionale, che sarà un beneficio acquisito dallo studio di questa materia, né del *coding*, che è solo una delle sue aree. Ma proprio dell'informatica, in quanto scienza autonoma.

Questo, d'altro canto, è quanto sta accadendo nei paesi più avanzati. Un panorama dettagliato della situazione nei vari stati europei è presente in (CECE 2017).

In Inghilterra, il Ministero dell'Istruzione ha pubblicato[3] nel settembre 2013 il "Programma nazionale di studio per l'Informatica", che ha reso l'informatica una materia obbligatoria sin dalla prima elementare a partire dall'a.s. 2014-15. È interessante osservare che il testo usa il termine pensiero computazionale (senza mai definirlo) solo un paio di volte e proprio nel senso, precedentemente discusso, di ciò che si ottiene con lo studio e la pratica dell'informatica.

Negli USA l'atto legislativo "*Every Student Succeeds Act*" (ESSA) è stato approvato dal Congresso nel 2015, con un supporto bipartisan e ha introdotto l'informatica tra le materie "per un'istruzione a tutto tondo" (*well rounded educational subjects*) che devono essere insegnate nella scuola. Di nuovo, senza mai citare né pensiero computazionale né *coding*. A gennaio 2016 l'allora presidente Obama ha lanciato l'iniziativa "*Computer Science For All*" con l'obiettivo di «mettere tutti gli studenti americani, dall'asilo al liceo, in condizione di imparare l'informatica ed essere così attrezzati con le abilità di pensiero computazionale di cui hanno bisogno …»[4]. Viene nuovamente ribadito che il pensiero computazionale è quello che si

[3] https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study/national-curriculum-in-england-computing-programmes-of-study

[4] http://www.nsf.gov/csforall

ottiene quando si impara l'informatica. Tale iniziativa ha raccolto una grande coalizione, cui partecipano anche moltissime aziende di informatica, grandi e piccole.

In Francia, la *Académie des Sciences*, la più importante istituzione che rappresenta gli scienziati francesi, ha pubblicato nel maggio 2013 il rapporto "L'insegnamento dell'informatica in Francia. È urgente non aspettare oltre". In esso si raccomanda che l'insegnamento dell'informatica «inizi nella scuola primaria, attraverso le nozioni fondamentali dell'informatica e degli algoritmi e … venga ulteriormente sviluppato nella scuola secondaria» (Académie 2013). Anche in questo caso si usa il termine pensiero computazionale per denotare il modo di pensare sviluppato da chi ha studiato e praticato l'informatica. La programmazione, viene aggiunto, è un modo di acquisire i meccanismi di base del pensiero computazionale. A seguito di ciò l'insegnamento degli elementi di informatica è stato introdotto nel 2015 nelle scuole primarie e secondarie inferiori. Nel 2018 è stato introdotto l'esame di stato per qualificare i docenti all'insegnamento dell'informatica nella scuola.

In Israele dal 2016 l'informatica è stata introdotta come materia di studio obbligatoria a partire dalla classe quarta della scuola primaria, con due ore settimanali, dedicate soprattutto agli algoritmi e alla programmazione. Era già presente, ma come materia a scelta, nella scuola secondaria superiore fin dagli anni '90.

In Polonia l'informatica è materia di insegnamento nella scuola superiore fin dal 1985 ed è stata introdotta come obbligatoria a partire dalla prima elementare (con un'ora a settimana) dal 2017. Nella scuola secondaria di secondo grado si aggiungono altre 2 ore settimanali per alcuni indirizzi. Un risvolto interessante dell'impostazione del loro programma è che viene de-enfatizzato l'aspetto tecnologico della programmazione, in modo da sottolineare gli aspetti concettuali della risoluzione dei problemi attraverso un esecutore, più che lo specifico ambiente di programmazione utilizzato. Un ulteriore elemento caratteristico del loro curriculum è la considerazione che viene data agli aspetti sociali dell'utilizzo dell'informatica ed alle problematiche di consapevolezza (privacy, proprietà intellettuale, educazione, sicurezza, etica, …).

Recentemente l'UNESCO ha pubblicato un appello affinché i programmi informatici in codice sorgente siano considerati un patrimonio dell'umanità e vengano conservati e preservati (UNESCO 2018). In tale contesto è stato formulato un invito a tutti gli Stati membri a inserire i fondamenti scientifici dell'informatica nell'i-

struzione di base per tutti i cittadini e a responsabilizzare tutti sulle conseguenze etiche dell'uso dei sistemi informatici. Si tratta di un elemento indispensabile al buon funzionamento di ogni società democratica, in cui il codice informatico sempre di più interviene nella regolazione e realizzazione di ogni attività umana. Questo codice mette a disposizione un enorme potere creativo, ma richiede anche un'accorta regolamentazione che assicuri quella trasparenza e responsabilità senza le quali nessuna democrazia è degna di questo nome. Questo esplicito appello dell'UNESCO è un ulteriore conferma che l'importanza di un'adeguata formazione scientifica nell'informatica sin dalla scuola è un tema sempre più condiviso in tutti i Paesi avanzati.

Infine, è da poco partita in Europa l'iniziativa[5] "*Informatics for All*" che sostiene a livello europeo l'importanza, per un pieno ed equilibrato sviluppo della società digitale, dell'insegnamento dell'informatica fin dalle prime classi.

5 Informatica nella scuola italiana

L'esperienza delle nazioni che si sono avviate prima dell'Italia sulla strada dell'introduzione dell'informatica nella scuola è importantissima. Ci mostra infatti che si tratta di un cambiamento epocale, i cui tempi non possono essere compressi, perché soggetti obbligatoriamente al cambiamento nelle persone e nelle procedure in azione all'interno di un sistema complesso quale quello dell'istruzione obbligatoria.

È altresì chiaro che formare adeguatamente all'informatica è un'attività strategica per lo sviluppo di una nazione, data la sempre maggiore dipendenza di prodotti e servizi dalle tecnologie informatiche in tutte le fasi di progettazione, realizzazione ed erogazione, e considerando che lo stesso tessuto sociale (relazioni e comunicazioni) è ormai fittamente innervato da strumenti digitali. La vitalità ed il successo di un paese democratico in un futuro sempre più digitale dipenderanno in larga misura dal livello di cultura informatica dei suoi cittadini.

Si tratta di un percorso che richiede di affrontare simultaneamente sia la formazione di lavoratori e cittadini, sia il cambiamento del si-

[5] http://www.informaticsforall.org

stema scolastico. Nel seguito ci focalizziamo solo sul sistema scolastico per il quale, a sua volta, va distinta la situazione transitoria (che può durare anche 5-10 anni) da quella a regime.

Nella situazione transitoria vanno sostenute quelle azioni di accompagnamento che nel corso di questi ultimi 3-5 anni hanno mostrato la loro validità, in modo che questa formazione culturale sull'informatica diventi consuetudine diffusa, sia per i docenti che per gli studenti, mentre si avviano e realizzano le attività che consentiranno di avere a regime un'adeguata quantità di didattica dell'informatica nella scuola. Lo scopo è di arrivare a far considerare "naturale" ad ogni cittadino e ad ogni genitore una formazione scientifica di base nell'informatica, così come viene considerata naturale una formazione scientifica di base in fisica e biologia per capire come funzionano l'elettricità e la cellula.

Tra queste azioni va sicuramente ricordato il progetto Programma il Futuro[6]. Realizzato dal CINI (Consorzio Interuniversitario Nazionale per l'Informatica) in collaborazione con il MIUR (Ministero dell'Istruzione, dell'Università e della Ricerca), nel corso dell'intero a.s. 2017-18, ha visto la partecipazione di ben 111.311 classi (il 30% delle classi di tutti gli ordini di scuola) corrispondenti a 2.044.785 studenti che hanno svolto in media 15 ore di formazione informatica[7]. Il progetto, iniziato nell'a.s. 2014-15, aveva l'obiettivo di coinvolgere in un triennio il 40% degli istituti scolastici portandoli a svolgere 10 ore di informatica nel 9% delle classi. Al 15 gennaio 2019 risultano coinvolte complessivamente 6.619 scuole, corrispondenti al più dell'80% degli istituti italiani, superando quindi largamente tutti gli obiettivi inizialmente fissati. Nell'ambito del progetto sono state anche realizzate attività di ricerca sulla didattica dell'informatica pubblicate a convegni scientifici internazionali (Corradini et al. 2017a, 2017b, 2018a, 2018b, 2018c).

Gli elementi necessari affinché il sistema scolastico sia permanentemente in grado di erogare un'adeguata formazione sull'informatica sono la presenza di **indicazioni didattiche ministeriali** e la presenza di un **congruo numero di docenti ben preparati**.

Per quanto riguarda le indicazioni didattiche ministeriali, bisognerà intervenire su quelle Indicazioni Nazionali che costituiscono il quadro di riferimento unico nazionale per lo sviluppo dell'offerta formativa nei singoli istituti. La sfida qui consiste nel riuscire a

[6] https://programmailfuturo.it
[7] https://programmailfuturo.it/progetto/monitoraggio-del-progetto

bilanciare le molte richieste di aggiornamento di tali Indicazioni con la limitatezza del tempo a disposizione per l'insegnamento in classe. Bisognerà riflettere a fondo su come poter conciliare le necessità educative presentate dalla società contemporanea, che ha raggiunto una complessità strutturale e funzionale mai vista prima, con la difesa della specificità della tradizione culturale italiana. Obiettivo non facile che richiederà un approccio interdisciplinare, ma che non può non essere affrontato se si vuole davvero cambiare il Paese per rilanciarne il futuro. Un possibile modo di farlo è quello della "tabula rasa".

Immaginiamo di non avere affatto le Indicazioni Nazionali e di partire adesso ex-novo a costruire dei programmi di insegnamento, come se non ci fosse un passato. Dato l'attuale sviluppo della società e la necessità di formare cittadini in grado di agire in essa con sufficiente conoscenza delle questioni in gioco, su quali discipline dobbiamo formarli? E come? Sarà necessario un approfondito dibattito, un attento esame di tutte le questioni in gioco. È un esercizio che chiamerà in causa la politica al suo più alto livello, perché agiranno certamente molte spinte in direzioni discordanti, ed è la politica che dovrà avere la capacità di trovare una sintesi nell'interesse generale della società.

Il CINI, mediante il Gruppo di Lavoro "Informatica e Scuola" coordinato dall'autore, ha già elaborato una proposta operativa (CINI 2017), sottoposta all'attenzione del Comitato Scientifico per le Indicazioni Nazionali per il curriculo della scuola dell'infanzia e del primo ciclo di istruzione del MIUR, che verrà discussa più in dettaglio nella sezione 7 ed è riportata integralmente in Appendice.

Per quanto riguarda la disponibilità di docenti ben preparati, vale la pena ricordare, in tempi in cui sembra che tutto si possa imparare da soli su Internet, che l'insegnamento, strutturale e permanente, di una materia scolastica non può prescindere dalla presenza diffusa di insegnanti che sappiano il fatto loro.

Il tema della formazione docenti è estremamente complesso e sfaccettato, dal momento che formare un docente di scuola primaria richiede un impegno ben diverso da quello di uno della secondaria di secondo grado. Un recente rapporto, elaborato dalla Royal Society in riferimento alla situazione nel Regno Unito (Royal Society 2017), ha evidenziato che la presenza di un sufficiente numero di insegnanti ben preparati ed adeguatamente supportati è il problema di gran lunga più importante per apportare un cambiamento permanente e significativo. In seguito a ciò il governo inglese ha avviato

nel novembre 2018, con una dotazione di 84 milioni di sterline, un centro nazionale per l'insegnamento dell'informatica a 8.000 insegnanti[8], con un costo medio di circa 10.500 sterline per insegnante.

Il CINI, grazie alla diffusione degli atenei del Consorzio in tutto il territorio italiano ed alla sua rete di circa 1.700 tra docenti e ricercatori universitari, può certamente essere un valido braccio operativo – attraverso il Gruppo di Lavoro "Informatica e Scuola" – per organizzare ed erogare tale formazione in modo coerente e scientificamente valido.

Un intervento di questo tipo ha un'enorme complessità organizzativa e valutazioni estese a tutti i livelli di scuola necessitano di un'individuazione chiara degli obiettivi generali, che non può prescindere da considerazioni relative alla riorganizzazione dei curricula didattici.

Per fare giusto un esempio, che parte da ipotesi molto preliminari e considera solo la formazione dei docenti della scuola primaria, possiamo assumere l'obiettivo di far svolgere in ogni classe della scuola primaria almeno un'ora di informatica a settimana (che è il livello medio in atto nel Regno Unito). Allora dovremmo formare in informatica e nella sua didattica almeno un docente ogni 25, con una didattica sia teorica che pratica e laboratoriale e attività svolte sia in presenza (circa 60 ore di formazione in aula) che a distanza. Considerando i numeri in gioco (circa 10.000 insegnanti per circa 5000 scuole primarie) e che si tratta di docenti in servizio, il processo non potrebbe essere completato in un arco di tempo inferiore ai 3-5 anni.

È chiaro da quanto sopra esposto che sono necessarie volontà politica e risorse economiche. Se c'è la prima le seconde si trovano.

6 I concetti fondamentali dell'informatica

In questa sezione fornisco una sintetica descrizione dei concetti fondamentali dell'informatica che sono alla base dell'intero edificio teorico e sistemistico. È stato già chiarito che l'obiettivo dell'informatica come scienza è la risoluzione automatica di problemi, effettuata da un esecutore o agente. Ovviamente tutto nasce con

[8] https://www.gov.uk/government/news/tech-experts-to-provide-national-centre-for-computing-education