

Educare al pensiero computazionale: un'esigenza per i processi di Formazione Professionale oggi.

Prima parte

MICHELE PELLEREY¹

Questo primo contributo sulla tematica del pensiero computazionale intende accostare il lettore, un po' a volo di uccello, a una prima comprensione di quella dimensione del pensiero umano che a partire dagli apporti di Seymour Papert e di Jannette Wing è stata definita computazionale. Una sua attenta analisi metterà in luce come, soprattutto nell'ambito della formazione professionale, sia necessario promuoverne lo sviluppo come elemento fondamentale di un'identità professionale adeguata alle trasformazioni tecnologiche e organizzative che stanno contraddistinguendo i nostri tempi. In un prossimo contributo si approfondiranno alcuni aspetti fondamentali, qui solo accennati, da un punto di vista formativo.

This first article on the subject of computational thinking intends to approach the reader to a first understanding of that dimension of human thought starting from the contributions of Seymour Papert and Jannette Wing. A careful analysis will highlight how, especially in the field of vocational training, it is necessary to promote its development as a fundamental element of a professional identity in line with the technological and organizational changes that are taking place in our times. In a future article some fundamental aspects will be explored from a training point of view.

Introduzione

42 anni fa nel settembre del 1976 il CNOS-FAP organizzò in collaborazione con l'Istituto di Didattica dell'Università Salesiana il primo corso di informatica per operatori della Formazione Professionale. Pietro Chasseur rievocando quell'esperienza scriveva nel 1985: «Quanti frequentarono questo "epico" corso, proseguito per altri due anni, incominciarono poi a socializzarlo, in un primo tempo con i colleghi insegnanti, poi ai primi tentativi di estensione agli allievi del terzo anno, consolidandosi poi nell'attuale impostazione».² Quell'esperienza portò, dopo dieci anni di approfondimenti, alla pubblicazione nel 1986 di un volume dedicato ai fondamenti culturali e tecnologici dell'informatica.³ Già in quell'e-

¹ Professore emerito, già Ordinario di Didattica dell'Università Pontificia Salesiana di Roma.

² P. CHASSEUR, *Esperienza di informatica al CFP San Zeno di Verona*, Rassegna CNOS, 1985, 1, pp. 93-96.

³ M. PELLEREY, *Informatica. Fondamenti culturali e tecnologici*, Torino, SEI, 1986.

sperienza veniva evocata la possibilità di un'introduzione "povera" di tale disciplina, cioè senza il supporto di particolari attrezzature informatiche, puntando soprattutto allo sviluppo dei concetti e dei procedimenti essenziali che la caratterizzavano.

L'attenzione rivolta più al pensiero che sta alla base dell'informatica che all'uso pratico di attrezzature digitali è riemersa nel mondo accademico nel 2006 a causa di un influente intervento dell'allora Direttrice del Dipartimento di Informatica della Carnegie Mellon University degli Stati Uniti Jannette Wing.⁴ Questa studiosa riprendeva una sollecitazione di Seymour Papert degli Anni Sessanta, orientata a considerare prioritaria la formazione a quello che egli aveva denominato "pensiero computazionale". Da allora ha avuto inizio un forte movimento a livello accademico, scolastico e politico nel cercare di inserire nei percorsi educativi scolastici e formativi professionali non solo e non tanto l'uso di strumentazioni digitali, quanto piuttosto lo sviluppo dei suoi fondamenti cognitivi e culturali. Anche in Italia ci si è mossi in questa direzione con le varie iniziative di *coding*. Così anche il CINI (Consorzio Interuniversitario Nazionale per l'Informatica) ha sviluppato una "Proposta di Indicazioni Nazionali per l'Informatica nella Scuola".

Mi è sembrato utile allora riprendere molti degli elementi che avevano caratterizzato le esperienze degli Anni Settanta e Ottanta, riconsiderandoli nella prospettiva oggi dominante, cioè come componenti essenziali del pensiero computazionale, ed esaminandone alcune delle ricadute essenziali sui processi formativi in particolare di natura professionale.

1. L'astrazione come processo essenziale nello sviluppo del pensiero e del pensiero computazionale in particolare

Ricordo la situazione degli anni settanta relativa alla Formazione Professionale nell'ambito della meccanica e degli Anni Ottanta per quanto concerne la grafica: l'introduzione delle macchine a controllo numerico nel primo caso e della fotocomposizione nel secondo, pose seri problemi di natura cognitiva a operatori abituati ad agire direttamente sulle macchine e ora costretti a usare per comunicare con esse linguaggi artificiali. Se avessero avuto un'adeguata formazione matematica, ciò li avrebbe aiutati a gestire rappresentazioni astratte di

⁴ J.M. WING, *Computational Thinking*, Communications of the ACM, march 2006, 49, 3, pp. 33-35.

processi concreti, cioè saper valorizzare modellizzazioni matematiche della realtà. La rappresentazione astratta di situazioni concrete contraddistingue molte delle discipline fondamentali che oggi dovrebbero caratterizzare la formazione sia culturale, sia soprattutto professionale. Sempre più operazioni ripetitive, ben rappresentate nelle parodie di Charlie Chaplin del film *Tempi moderni*, vengono delegate a macchine che le possono eseguire spesso con più precisione e più continuità degli esseri umani. Basti pensare alla liberazione da lavori gravosi avvenuta nelle abitazioni domestiche con l'avvento delle lavatrici automatiche e delle lavastoviglie.

Ma il passaggio da forme dirette di comunicazione a forme mediate da rappresentazioni astratte fa parte della storia umana in maniera determinante. La prima conquista fondamentale, almeno per il mondo occidentale, è stata l'invenzione dell'alfabeto fonetico verso il 600 a.C. Con poco più di venti segni grafici⁵ si potevano rappresentare tutte le parole dette e i pensieri che ne stavano dietro. Ma la cosa fondamentale era che la lettura, cioè la decodificazione delle parole scritte, permetteva di risalire a quanto detto e pensato dallo scrivente. Al processo di rappresentazione astratta corrispondeva un processo di interpretazione o di ritorno alla realtà del contenuto rappresentato, anche se tutto ciò implicava una dinamica in cui entrava in gioco la competenza linguistica e il mondo interiore del lettore. Scrivere e leggere diventavano così competenze fondamentali per poter partecipare consapevolmente alla vita culturale. Acquisire le conoscenze e le abilità coinvolte nel processo di scrittura e in maniera simmetrica quelle necessarie per attivare con successo quello di lettura diventavano l'essenza della cosiddetta alfabetizzazione di base.

Ci sono voluti molti secoli perché l'uomo occidentale, in particolare quello europeo, conquistasse un altro passaggio fondamentale. All'inizio del 1200 veniva pubblicato, ma la stampa non c'era ancora, il *Liber abaci* di Leonardo Pisano detto il Fibonacci. Si veniva a conoscenza di quanto già sviluppato da indiani e arabi: un sistema di rappresentazione astratta del numero più valido e funzionale di quello fino ad allora usato. Ora con dieci segni grafici si poteva non solo scrivere qualsiasi numero, ma soprattutto operare nell'ambito dell'aritmetica in maniera agevole ed efficace. Venivano così anche identificati e denominati alcuni procedimenti di calcolo, in particolare gli algoritmi necessari per eseguire le quattro operazioni fondamentali. La parola "algoritmo" era stata introdotta da Leonardo Pisano come latinizzazione del nome del grande matematico arabo persiano Mu ammad ibn M sà al-Khw rizm. Leonardo scriveva infatti "dixit Algorit-

⁵ Segni grafici detti grafemi corrispondenti ai suoni presi in considerazione detti fonemi.

mus” quando faceva riferimento a un procedimento da lui messo a punto. Anche in questo caso al mondo numerico reale faceva da corrispondente una sua rappresentazione astratta valida ed efficace.

Parole e numeri presenti nel mondo reale, concreto, avevano infine trovato forme rappresentative astratte, segni grafici da utilizzare sia nel percorso di scrittura delle parole, dei numeri e dei calcoli, sia di loro interpretazione ed esecuzione. Rimaneva tuttavia un mondo reale che nella comunicazione svolgeva una funzione fondamentale: la musica. La questione era dunque quella di giungere a forme di rappresentazione astratta, mediante segni grafici, di melodie e armonie sia cantate, sia suonate mediante strumenti. Nel corso dei secoli ci sono stati molti passaggi finché anche in questo caso si è giunti a una forma convenzionale e universale di notazione musicale: il rigo o pentagramma. Il musicista poteva ora scrivere le sue musiche e gli altri leggere e interpretare quanto da lui elaborato.

Era possibile compiere progressi simili nell’ambito di altre realtà umane come lavori ripetitivi e faticosi, le cosiddette opere servili, affidate a schiavi umani, o, ancora, in produzioni più impegnative, che richiedevano particolare attenzione operativa, come costruire qualcosa? In altre parole pensare a congegni capaci di operare su materiali fisici al fine di produrre i risultati desiderati? Un sogno che può essere ricondotto al mito ebraico del Golem, cioè di una realtà costruita dall’uomo, dotata di una straordinaria forza e resistenza in grado di eseguire alla lettera gli ordini del suo creatore, di cui diventava una specie di schiavo, tuttavia incapace di pensare, di parlare e di provare qualsiasi tipo di emozione perché privo di un’anima. In altre parole quello che oggi passa sotto la dizione “automa” esecutivo, oppure di “robot non autonomo”: un congegno che riesce a eseguire operazioni o movimenti propri dell’uomo ma privo di autonomia decisionale.

Il sogno di avere al proprio servizio strumenti fisici in grado di svolgere mansioni spesso ripetitive e noiose portò nel 1600 a progettare strumenti di calcolo aritmetico su base meccanica. Qualche anticipazione c’era stata nei secoli precedenti nell’ambito degli orologi e di altri congegni abbastanza industriosi. Così il diciannovenne Blaise Pascal pensò di aiutare il padre intendente di finanza nel fare calcoli monetari lunghi e complessi realizzando la sua prima calcolatrice meccanica: la cosiddetta Pascalina. G. Leibniz ne progettò una più evoluta. La progettazione di queste macchine esigeva, ormai lo sappiamo, una rappresentazione astratta dei procedimenti richiesti per poterla poi concretizzare nell’ambito meccanico mediante appositi meccanismi, come ruote dentate ingranate tra di loro.

Quando nell’Ottocento si cercò di ideare macchine in grado di svolgere operazioni più complesse, il primo passo fu la prefigurazione astratta di un sistema adatto a tale impresa. Gran parte dei primi passi e poi della nascita concreta di una calcolatrice automatica sono stati compiuti a livello di progettazione astrat-

ta, valorizzando e talora sviluppando ulteriormente concetti e procedimenti propri della logica formale. D'altra parte la progettazione di un sistema di questo tipo implicava una simulazione attenta di come l'uomo procede nell'affrontare elaborazioni complesse. Da una parte occorre mettere in gioco la disponibilità di informazioni opportunamente codificate e strutturate (come numeri e parole scritte), dall'altra, era necessario potersi avvalere di un insieme di procedimenti o algoritmi che permettessero di operare su tali dati per ottenere i risultati attesi. Infine, l'interazione tra insieme dei dati da elaborare e procedimenti da mettere in campo implicava un organo di governo, che su comando umano mettesse in relazione dati e procedimenti per ottenere il risultato richiesto. Se desideravo ottenere il risultato di operazioni aritmetiche specifiche occorreva non solo dire quali numeri prendere in considerazione, ma anche quale procedimento mettere in atto, il tutto in forme comprensibili dal sistema automatico. Di qui la delimitazione di una unità di memoria, di una unità logico-aritmetica e di una unità di governo o esecutiva. Infine, era necessario prevedere come immettere i dati in tale sistema e come ottenere che i risultati delle elaborazioni richieste diventassero disponibili: gli organi di input (ingresso) e di output (uscita). Ne è derivato un modello astratto di sistema ancor oggi valido. Esso comprende: a) Unità di ingresso: serve per comunicare al calcolatore quello che desideriamo che faccia, dandogli anche le informazioni necessarie; b) Unità di uscita: serve al calcolatore per comunicare a noi il risultato del suo lavoro. Può essere un monitor o una stampante; c) Unità di elaborazione: è la parte nella quale si eseguono le operazioni che noi desideriamo; d) Unità di memoria: è la parte nella quale si conservano sia i dati da utilizzare, sia le procedure da eseguire; e) Unità di governo: è la parte fondamentale, quella che garantisce il funzionamento corretto della macchina, un po' come il capostazione di una stazione ferroviaria o la torre di controllo di un aeroporto.

Questa lunga riflessione anche di natura storica è giustificata dall'affermazione di Jannette Wing circa la centralità del processo astrattivo nell'ambito del pensiero computazionale: «[...] cercare di catturare le proprietà essenziali comuni a un insieme di oggetti, mentre si nascondono distinzioni irrilevanti».⁶ Nella creazione del linguaggio fonetico si è cercato di catturare le proprietà essenziali della parola detta, nascondendo intonazione, timbro della voce, ecc. Nella scrittura della musica sul pentagramma non vengono prese in considerazione molte qualità proprie del canto reale o di uno strumento musicale concreto come la tromba o il clarinetto. Nel caso che ora esploreremo più in dettaglio, i pro-

⁶ J. WING, Computational Thinking. What and Why, *The Link*, March 2011, p. 1.

cessi e le situazioni reali vengono esaminati per evidenziare, se ciò è possibile, la loro struttura essenziale dal punto della loro rappresentazione algoritmica o dell'organizzazione logica dei dati.

2. Dalle macchine a controllo numerico alla gestione automatica di biblioteche, musei e magazzini

Parallelamente allo sviluppo dei primi elaboratori a base elettromeccanica, poi elettronica, è iniziata a farsi strada l'idea di macchine utensili controllate da programmi elaborati con opportuni linguaggi comprensibili da tali strumenti meccanici e gestiti da apparati elettronici. Dalla fine degli Anni Quaranta dell'altro secolo agli Anni Cinquanta, grazie anche alla collaborazione-competizione tra John T. Parsons e il Massachusetts Institute of Technology, si arrivò alla progettazione e realizzazione delle prime macchine a controllo numerico. Esse furono sviluppate soprattutto negli Anni Sessanta e negli Anni Settanta iniziarono a diffondersi anche in Italia. I processi di lavorazione dei pezzi meccanici venivano descritti in maniera astratta, tradotti in comandi scritti in codici appositi e quindi messi in atto in maniera solo indirettamente gestita dall'uomo, in quanto egli doveva intervenire solo se il programma esecutivo andava incontro a imprevisti. Anche in questo caso la questione centrale stava nella rappresentazione astratta del processo di fabbricazione di oggetti fisici.

Lo sviluppo delle macchine a controllo numerico, come torni, fresatrici e alesatrici, ha portato progressivamente ad associare il disegno tecnico degli oggetti da realizzare, attuato con il supporto di un computer (*CAD, Computer Aided Design*), a un programma che fosse in grado di interpretare tale disegno tecnico fino a gestirne la realizzazione mediante opportune macchine utensili (*CAM, Computer Aided Manufacturing*). Si è giunti così anche alla progettazione e realizzazione di oggetti in tre dimensioni mediante l'uso di appositi software. Di qui un passo ulteriore verso la progettazione di robot dotati di maggiore autonomia. Ciò è stato possibile inserendo inizialmente nel congegno forme di retroazione, cioè di fronte a un ostacolo o a un nuovo elemento non previsto, la possibilità di cambiare modalità operativa; poi, siamo nei nostri anni, integrando nel sistema elementi propri dell'Intelligenza Artificiale o delle cosiddette "learning machines" o macchine che apprendono.

La questione della rappresentazione astratta di situazioni o processi concreti ha avuto un suo parallelo sviluppo fin dall'antichità nella catalogazione dei libri nelle biblioteche o delle opere nei musei. Qui doveva essere associata la descrizione dell'oggetto alla sua collocazione fisica in appositi reparti specializzati, per poterlo poi reperire quando necessario. L'estensione di questi procedimenti

alla gestione dei magazzini ha portato alla nascita di una nuova disciplina la “logistica”, oggi costituita da un insieme di conoscenze e di abilità che permettono la gestione automatica dei magazzini. È quanto avviene ad esempio per le vendite online da parte di Amazon, ma anche, più in piccolo, in un comune super o ipermercato. A ogni oggetto deve essere associato un codice identificativo preciso che includa anche i parametri di riferimento per poterlo prelevare. Per questo tutti i prodotti commerciali sono ormai contrassegnati da un codice a barre.

Per comprendere meglio come funziona un sistema di questo tipo è utile descrivere più in dettaglio una quotidiana esperienza, che evoca molti elementi di automazione ormai gestiti da computer. Un cliente all’uscita da un supermercato presenta alla cassa ciò che ha nel suo carrello per pagare quanto intende acquistare. L’impiegato alla cassa prende ogni oggetto e fa leggere da un lettore automatico il codice a barre di ogni prodotto. Il codice contiene le informazioni relative al prodotto considerato, a esse sono associate altre informazioni utili (oltre al suo nome, il suo prezzo, l’azienda che lo ha prodotto, la sua collocazione, ecc.). Le casse sono collegate a un computer, o sistema centrale. Questo riceve le informazioni che provengono dalle casse, le memorizza; poi, su richiesta, esegue le operazioni programmate e fa, ad esempio, stampare la ricevuta. Non solo. Conserva nella sua memoria gli incassi, il fatto che un certo prodotto è stato venduto e non è più nel magazzino. Quest’ultimo dato è importante perché segnala al gestore del supermercato che un dato prodotto sta per finire e conviene rifornirsene. A fine giornata, o a fine mese, è possibile sapere l’incasso totale, quali prodotti sono stati più venduti e quali invece no. Ciò aiuta a programmare le prossime forniture.

In tutti questi casi al magazzino, alla biblioteca, al museo, agli impianti reali, vengono associate in maniera astratta tutte le informazioni necessarie alla loro gestione. Le informazioni per poter essere conservate da un sistema informatico, un computer, in maniera appropriata e utile devono essere tradotte in dati, cioè codificate opportunamente, cioè secondo un sistema, o struttura, valido e funzionale alla loro manipolazione. Si vengono così a costituire quelle che in terminologia tecnica si chiamano “basi di dati”.

Così la rappresentazione astratta dei processi concreti mediante algoritmi e delle informazioni mediante strutture o basi di dati viene a costituire il cuore del cosiddetto pensiero computazionale. In tale mondo astratto si cercano le soluzioni ai problemi da affrontare per poi tradurle in codici comprensibili da automi che siano in grado di comprenderle ed eseguirle. Naturalmente i primi problemi affrontati sono stati procedimenti operativi di natura aritmetica, facilmente rappresentabili in maniera astratta perché già noti e denominati in onore del suo massimo inventore come algoritmi. In questo caso le basi di dati erano co-

stituite dal sistema dei numeri naturali, interi, decimali. Ma nel seguito i problemi affrontati si sono moltiplicati e ormai coprono ogni settore del sapere e dell'agire umano. Così è stato possibile affermare che il pensiero computazionale riguarda: «[...] i processi di pensiero coinvolti nel formulare un problema ed esprimere la sua o le sue soluzioni in maniera che un computer, un umano o una macchina, possa effettivamente portarlo a termine».⁷

Queste iniziali osservazioni portano naturalmente ad approfondire due grandi quadri concettuali propri del pensiero computazione: quello degli algoritmi e quello delle basi di dati.

3. Un primo sguardo alla dimensione algoritmica del pensiero computazionale

In generale un algoritmo può essere inizialmente descritto come: «[...] una sequenza finita di passi elementari che portano alla risoluzione di un problema».⁸ Naturalmente esistono alcune condizioni che rendono tale sequenza effettiva: che essa vada bene per una varietà di input, fornendo i risultati richiesti (output); che ogni passo ammetta un'interpretazione unica e sia eseguibile in tempo finito; che la sua esecuzione termini qualunque sia l'input ammesso. Come subito è evidente, la questione centrale sta nell'esame del problema da affrontare e nella possibilità di trovarne una soluzione di tipo algoritmico, cioè che esista effettivamente una successione finita e precisa di operazioni da compiere per ottenerne la soluzione. In linguaggio tecnico si usa dire che in tal caso il problema è di natura *computabile*, cioè calcolabile. Da questo punto di vista esistono moltissimi esempi di algoritmi collegati a problemi di vita quotidiana, anche perché spesso ci imbattiamo in procedure prestabilite che condizionano l'utilizzo di molte apparecchiature o di servizi generali. Queste procedure sono state stabilite, almeno così pensano i loro creatori, per ottimizzarne l'utilizzo. Basta pensare alla prenotazione online di servizi di tipo ospedaliero o all'utilizzazione di macchinari professionali un po' complessi. Più modestamente si possono evocare le ricette di cucina.

Un esempio di problema che implica la ricerca e l'individuazione di una successione precisa di passi da compiere è data da un comune indovinello. "Un contadino ha una capra, un lupo e un cavolo. Se lascia il lupo solo con la capra, il lupo mangia la capra; se lascia la capra sola con il cavolo, la capra mangia il ca-

⁷ J.M. WING, *Computational thinking's influence on research and education for all*, Italian Journal of Educational Technology, 2017, 25(2), pp. 7-14.

⁸ P. FERRAGINA, F. LUCCIO, *Il pensiero computazionale. Dagli algoritmi al coding*, Bologna, Il Mulino, 2017, p. 10.

volò. Giunto alla riva di un fiume, egli può traversarlo solo con una piccola barca, che può portare contemporaneamente solo lui e un'altra cosa: o la capra, o il lupo, o il cavolo. Come fa ad attraversare il fiume, portando tutti salvi all'altra riva?". La soluzione dell'indovinello porta alla scoperta di una precisa sequenza di operazioni. Il contadino: 1) attraversa il fiume con la capra; 2) torna solo; 3) attraversa il fiume con il cavolo; 4) ritorna con la capra; 5) attraversa il fiume con il lupo; 6) ritorna solo; 7) attraversa il fiume con la capra.⁹

Nell'ambito delle macchine a controllo numerico si tratta di individuare una precisa sequenza di operazioni che la macchina deve compiere per produrre il pezzo progettato. Nell'ambito di un servizio di vendite online si tratta di prefigurare la successione di operazioni che portano dall'ordine di un prodotto, al suo reperimento in un magazzino, alla sua impacchettatura, alla sua spedizione, mentre si controlla la carta di credito del cliente e se ne preleva il prezzo stabilito. Per poter esaminare con cura le procedure necessarie si deve ricorrere a forme di loro rappresentazione astratta, soprattutto se esse risultano un po' complicate, ma soprattutto se poi dobbiamo tradurle in comandi che devono essere compresi ed eseguiti dalle macchine automatiche.

La branca di studio propria del pensiero computazionale dedicata all'analisi degli algoritmi ha avuto uno sviluppo assai fecondo. Nel 1966 Corrado Boehm e Giuseppe Jacopini hanno dimostrato uno dei suoi teoremi fondamentali: qualsiasi algoritmo, anche il più complesso, si basa in definitiva su tre schemi o strutture elementari di controllo: la sequenza, la selezione, l'iterazione o ciclo. Vediamo in sintesi di che cosa si tratta.

Il più semplice schema algoritmico è la *sequenza*: una successione ordinata di istruzioni che devono essere eseguite in maniera puntuale per ottenere il risultato previsto. Un esempio è dato dalla soluzione dell'indovinello precedente. Tuttavia non sempre le procedure sono lineari, nel senso che possono essere presenti momenti che implicano la possibile selezione tra due percorsi diversi. Lo schema della *selezione* è dunque costituito dalla scelta che deve essere fatta tra due percorsi possibili in base a una condizione che può risultare vera o falsa. In altre parole emerge una condizione espressa dalla formula "se..., allora...", oppure "se ..., allora ..., altrimenti ...". Un banale esempio è dato dal telefonare: se non ricordo il numero da chiamare, allora lo cerco sulla rubrica. Il terzo schema è detto *iterazione* o *ciclo*: è costituito da una successione di operazioni da ripetere finché non si verifica una precisa condizione.

⁹ Un problema analogo ma più sofisticato ed esaminato con cura dagli studiosi è costituito dalle cosiddette Torri di Hanoi (cfr. P. FERRAGINA, F. LUCCIO, *o.c.*, pp. 212 e ss.).

Per rappresentare in maniera astratta, ma chiaramente definita, i successivi passi di un algoritmo si possono usare diagrammi di flusso o scritture dette “pseudo codici”. Tutte forme che consentono di controllare la qualità delle nostre soluzioni algoritmiche e verificarne la correttezza e complessità. Naturalmente nel corso dei secoli e più recentemente dei decenni passati molti problemi sono stati già affrontati e sono state trovate le relative soluzioni algoritmiche. Ma sono stati individuati anche problemi che non possono essere risolti mediante algoritmi. Problemi di quest’ultimo tipo sono stati definiti non computabili. D’altra parte, in molti casi la soluzione algoritmica di un problema è facilitata dallo spezzare di un problema generale in sotto problemi e così spesso si possono utilizzare soluzioni algoritmiche già individuate. Queste vengono a costituire come un insieme di moduli già pronti per essere valorizzati nello sviluppare la procedura completa: una sorta di moduli componibili. Approfondiremo nel prossimo intervento questa parte del pensiero computazionale cogliendone gli essenziali aspetti formativi.

4. Le basi di dati

L’avvento degli elaboratori elettronici, in qualsiasi modalità oggi si presentano, ha portato a poter disporre di grandi spazi di memoria. Quando compriamo uno smartphone il suo prezzo varia principalmente sulla base dell’ampiezza della sua memoria, oltre che dalla qualità dello schermo e del processore. Accanto, ormai, abbiamo molte memorie dette aggiuntive, ma che sono a nostra disposizione. Alcune, sotto forme di penna, disco, ecc., sono sulla scrivania; altre non si sa dove siano, per questo si parla di *cloud*, di nuvole. Ma sempre a nostra disposizione. Le informazioni sotto forma di numeri, di parole, di immagini fisse e in movimento, di testi più o meno lunghi e multimediali, di app e di programmi di vario tipo riempiono le memorie che sono a nostra disposizione. Ma se passiamo ai cosiddetti social media, alla rete informatica, entriamo facilmente in una foresta immensa nella quale è facile perdersi o girovagare senza meta. Non per nulla si parla sempre più di “big data”, nel senso che le informazioni trasformate in dati attraverso la loro codificazione aumentano giorno dopo giorno a dismisura. Come è stato nel passato ed è oggi possibile muoversi intelligentemente in questo oceano?

Il primo e fondamentale passaggio per impostare una memoria contenente le informazioni prese in considerazione è dato dall’assegnare a ciascuna un codice identificativo. Ma questo codice va progettato sulla base delle possibili necessità di reperibilità e di elaborazione successiva. Un esempio a noi familiare è quello del “codice fiscale”. Alla nascita a ciascun cittadino italiano è automati-

camente associato un codice identificativo. Esso è riportato, ad esempio, sulla propria tessera sanitaria. Tale codice è formato da sedici caratteri, in parte alfabetici, in parte numerici (per questo esso è denominato codice alfanumerico) ed è organizzato secondo quattro campi più un carattere finale chiamato codice di controllo.

I primi sei caratteri (primo e secondo campo, ciascuno di tre caratteri) riguardano il cognome e il nome.¹⁰ I seguenti cinque caratteri (terzo campo) riguardano la data di nascita e il sesso in questo modo: prima l'anno (due cifre finali dell'anno), poi il mese (indicato con le successive lettere dell'alfabeto), poi il giorno (se femmina a esso si aggiunge 40). A questi primi undici caratteri, che riguardano il tuo cognome, nome, data di nascita e sesso, seguono altri quattro caratteri (quarto campo) che riguardano il comune di nascita indicato secondo il codice catastale (da trovare su Internet¹¹). L'ultimo carattere, il sedicesimo, è una lettera che serve come codice di controllo della correttezza di quanto scritto. Per trovarlo si usa un algoritmo un po' complicato. Per semplicità si può trovare ricorrendo a uno dei numerosi siti dedicati al codice fiscale.¹² L'insieme dei codici fiscali e delle dichiarazioni dei redditi ad essi associate vengono a costituire un archivio imponente. La cosiddetta anagrafe tributaria, che comprende oltre ai codici fiscali anche le partite IVA sia delle persone, che delle aziende.

La struttura del codice fiscale permette di associare i vari suoi campi ad altre basi di dati o archivi disponibili. Si possono citare l'anagrafe nazionale della popolazione residente, l'insieme dei comuni in base al loro codice ISTAT. Ma anche possono essere collegati a ciascun codice fiscale altri codici interessanti e sensibili, come quelli relativi ai conti bancari. In particolare va considerato il codice IBAN o International Banking Account Number (Numero di Conto Bancario Internazionale). Esso è stato definito a livello internazionale ed è dal 2008 obbligatorio nelle varie transazioni monetarie tra banche. Ma a poco a poco esso può essere utilizzato anche per altri fini. Ad esempio, l'Agenzia delle Entrate potrebbe accedere ai vari conti bancari delle persone fisiche tramite il loro codice fiscale.

Naturalmente tutto questo comporta sia per le grandi basi di dati, sia per gli archivi più modesti, come quelli delle biblioteche, non solo tecniche di

¹⁰ Vengono prese le consonanti prima del cognome e poi del nome (o dei cognomi e nomi, se più di uno) nel loro ordine. Se le consonanti sono insufficienti, si prelevano anche le vocali, sempre nel loro ordine. Le vocali vengono riportate dopo le consonanti. Per le donne, viene preso in considerazione il solo cognome da nubile. Nel caso del nome, se contiene quattro o più consonanti, si scelgono la prima, la terza e la quarta, altrimenti le prime tre in ordine.

¹¹ <http://www.paginebianche.it/codice-catastale>.

¹² <http://ilmiof.it/>; <http://www.codicefiscale.com/>; <http://www.paginebianche.it/codice-fiscale>.

codificazione valide e funzionali, ma anche quello che viene definito un Sistema di Gestione delle Basi di Dati (DBMS), cioè l'insieme strutturato delle procedure (algoritmi) che consentono di valorizzarle, cioè definire, manipolare, richiamare e gestire i dati. Ad esempio si possono citare modalità di interrogazione delle basi di dati cosiddette relazionali dette *query*.

Una base di dati è quindi un insieme organizzato di informazioni opportunamente indicizzate, cioè trasformate in dati, alle quali si può facilmente accedere e che possono essere opportunamente gestite e aggiornate. Dal punto di vista del pensiero computazionale la questione centrale diventa così la logica che permette di costruire, valorizzare e aggiornare tali insiemi di dati a seconda delle esigenze poste dalle finalità per le quali sono progettate. Nel corso degli anni si sono quindi succedute modalità di progettazione e realizzazione sia differenti dal punto di vista della loro logica interna, sia da quello dell'ambito di loro applicazione. Da quest'ultimo punto di vista esse possono differire dal punto di vista dei loro contenuti: bibliografiche, testuali, numeriche, iconiche. Ma oggi in molti casi si punta a una organizzazione detta a oggetti, nel senso che un elemento della base di dati può avere un contenuto multimediale (testo, immagini, numeri) oppure anche un procedimento algoritmico opportunamente indicizzati. In generale, si parla di basi di dati di tipo relazionale quando si prefigurano sistemi basati su elementi alfanumerici messi in relazione tra di loro, di basi di dati non solo di tipo relazionale se ci si appoggia a modalità più complesse come quelle a oggetti.

5. La programmazione

Sia che l'esecutore sia un essere umano, sia un automa, la procedura progettata deve essere comunicata in maniera chiara e precisa, e ciò richiede una forma linguistica adeguata. Per questo è possibile sviluppare almeno in parte il pensiero computazionale senza l'utilizzo di un elaboratore elettronico. Tuttavia la natura stessa di questi congegni implica spostare l'attenzione dagli aspetti più tipicamente logico-matematici a quelli ingegneristici. La stessa già citata Jannette Wing lo mette ben in chiaro: «Il pensiero computazionale si basa congiuntamente sul pensiero matematico e su quello ingegneristico. Diversamente dalla matematica, tuttavia, i nostri sistemi computazionali sono vincolati dalla fisica delle strutture implicate dai processi di elaborazione delle informazioni e dai loro ambienti operativi».¹³ La differenza fondamentale tra pensiero matema-

¹³ J. WING, o.c., p. 2.

tico e pensiero computazionale sta proprio in questa intima unione che avviene a livello della seconda modalità di pensiero: la formalizzazione matematica deve tener conto delle esigenze poste dal sistema fisico che deve realizzare quanto prefigurato.¹⁴

Lo sviluppo tecnologico ed ingegneristico di tali strutture ha condizionato non solo le possibilità di una loro valorizzazione, bensì anche le modalità di comunicazione con essi. Tale sviluppo progressivo ha permesso di progettare linguaggi sempre più evoluti. Inizialmente, infatti, si usavano i cosiddetti linguaggi macchina, cioè la scrittura di istruzioni direttamente riferite sia alle cosiddette locazioni di memoria (gli elementi delle iniziali basi di dati), sia alle operazioni da eseguire con essi. Poi, parallelamente alla complessificazione delle macchine, si è avuta una più elevata possibilità non solo di elaborazione, ma anche di specificazione dei linguaggi artificialmente prefigurati. Ciò ha permesso di configurare gli elaboratori come macchine universali in quanto l'adattamento alle varie applicazioni poteva avvenire tramite opportuni programmi scritti in linguaggi specifici. Il cosiddetto *hardware*, la parte ingegneristica, poteva essere configurato come specializzato in ambiti particolari tramite il cosiddetto *software*, la parte logico-matematica. Di qui lo sviluppo di software applicativi di vario genere, da quelli specializzati in ambito contabile a quelli di tipo didattico, da quelli statistici a quelli ludici. Le app dei nostri smartphone, abbreviazione della parola di applicazione, sono specifici programmi applicativi. Ad esempio per sapere l'andamento climatico di un posto specifico, basta immettere il nome della località in una delle tante app di natura metereologica.

La scrittura di programmi, cioè di software, in grado di far funzionare elaboratori elettronici più o meno potenti e complessi è l'ambito più specificatamente denominato *coding*. Per cogliere il senso di questa parte del pensiero computazionale e promuoverne lo sviluppo nel tempo sono state proposte varie soluzioni. La più celebre è stata ideata e realizzata come ambiente di apprendimento da Seymour Papert negli Anni Sessanta presso il MIT (Massachusetts Institute of Technology) di Boston. Veniva usato un linguaggio di programmazione assai semplice, il Logo, a prova di bambini. Oggi lo stesso MIT ha messo a disposizione una sua versione più moderna e ricca, ma sempre assai semplice, di linguaggio di programmazione tramite il software denominato Scratch, di cui è disponibile anche una versione italiana gratuitamente scaricabile.¹⁵

¹⁴ J. WING, *Computational thinking*, Communications of the ACM, March 2006, 49, 3, p. 35.

¹⁵ <https://scratch.mit.edu/>

Per capire lo spirito della proposta sviluppata a suo tempo da Seymour Papert, e poi ripresa da varie esperienze, è utile riportare un brano della sua intervista del 1997, nella quale prende lo spunto dalla programmazione di un videogioco da parte dei bambini, tramite un software adatto. «Non è importante fare un videogioco, ma per i bambini il videogioco fa parte della cultura in cui vivono, loro pensano che sia importante, ed è importante per le loro vite. Dunque, il primo cambiamento che arriva quando un bambino può fare un proprio videogioco è che i bambini passano dall'essere consumatori ed essere produttori. Questo è un primo cambiamento nell'approccio e nella mentalità. L'errore della televisione, dei media, persino della scuola, sta nell'offrire la conoscenza ai bambini; in questa prospettiva i bambini consumano, non producono. Il bambino, viceversa, può, ora, realmente realizzare un videogioco, uno veramente bello; e questo è un cambiamento già di per sé importante. Ma facendo questo videogioco, parti realmente centrali della conoscenza entrano nel gioco, e così il bambino è molto motivato ad apprendere bene. Che cosa? Prima di tutto la programmazione: il bambino apprende a programmare il computer per fare il gioco. Abbiamo dei bambini di nove, dieci anni che imparano a programmare ad un livello che normalmente non ci si aspetta neanche da studenti di scuole medie o addirittura da studenti universitari». ¹⁶ E cita la possibilità anche di altri possibili apprendimento nel campo della matematica e della fisica.

La parte del pensiero computazionale più impegnativa che sta al centro di queste proposte sta proprio nel prefigurare il gioco, un gioco che deve seguire regole precise, tener conto delle opportunità e dei vincoli del software (in definitiva della macchina), definire le singole operazioni possibili, avvalendosi di una base di dati in parte esistente, in parte da definire. La programmazione consiste nel tradurre tale progetto in operazioni che esegue il computer sotto l'impulso dell'utente: un programma operativo. In questo modo non è la macchina che programma l'uomo, ma l'uomo che la programma. Ma anche nel caso della programmazione più semplice entra in gioco un processo di rappresentazione astratta. Si tratta di usare un linguaggio artificiale costruito proprio perché quanto comunicato possa essere compreso ed eseguito senza ambiguità e in completezza da un automa.

¹⁶ <http://www.mediamente.rai.it/biblioteca/biblio.asp?id=260&tab=int#link002>. Intervista raccolta il 7 marzo 1997. Sito consultato il 13 luglio 2015.

■ Conclusione

La conclusione più importante di questa prima presentazione di alcuni elementi fondamentali del pensiero computazionale sta nel mettere in evidenza alcune ricadute educative centrali, la cui principale è la necessità di promuovere una capacità di rappresentazione astratta, di modellizzazione, di situazioni o processi reali al fine di affrontare e risolvere problemi sul piano astratto, quello del modello, per poi riportare tali soluzioni sul piano della configurazione o della trasformazione della realtà concreta. Da un punto di vista educativo un passaggio obbligato da valorizzare in maniera adeguata nei percorsi formativi sembra essere costituito dall'apprendimento del disegno tecnico, un linguaggio rappresentativo astratto ormai di carattere universale. Esso implica proprio la capacità di descrivere con completezza, univocità, fedeltà e trasferibilità un oggetto fisico al fine di poter comunicare in maniera efficace e funzionale a un qualsiasi operatore, umano o artificiale, cosa produrre e come farlo.

Un secondo messaggio sta nell'esigere che la soluzione trovata al problema esaminato deve implicare una sua possibile esecuzione pratica o da parte di un essere umano o di una macchina. Gran parte dei problemi affrontati sul piano dell'azione nei vari ambiti della vita sociale, politica, professionale deve trovare prospettive di soluzione che siano praticabili. Cioè, è essenziale che ci si confronti sempre con la fattibilità di quanto proposto e non si rimanga sul piano solo ideale. Un messaggio estremamente importante soprattutto quando si redigono norme e procedure che guidano la vita quotidiana delle persone.