

PENSO DUNQUE SONO: ECCO COME FUNZIONA IL PENSIERO COMPUTAZIONALE



Massimo Nannini*

Quanto è importante oggi conoscere almeno un linguaggio di programmazione? La risposta più ovvia è moltissimo, soprattutto se parliamo delle nuove generazioni che si affacceranno al mondo del lavoro nei prossimi anni, ma anche per chi è già inserito in un qualunque ambito lavorativo visto il particolare periodo storico di transizione verso il digitale.

Quando ero studente mi sentivo spesso dire che la conoscenza di una lingua straniera e più specificatamente dell'inglese mi avrebbe aiutato a trovare un buon posto di lavoro e soprattutto sarebbe stato uno degli elementi distintivi tale da permettermi l'accesso a diverse opportunità. Oggi la conoscenza della lingua inglese è data per scontata in qualunque ambito e spiccatamente in quello tecnico scientifico ove è diventata la lingua universale che ha standardizzato le comunicazioni scritte e orali.

Possiamo paragonare la conoscenza di un linguaggio di programmazione in termini di opportunità alla conoscenza della lingua inglese? A mio modesto parere direi proprio di sì, ovviamente con le dovute accezioni.

Come succede per ogni linguaggio naturale, scritto e parlato, prima di potere esprimere un "concetto" è necessario conoscerne la grammatica e la sintassi, le parole, i costrutti, le regole, i tempi verbali ecc. Tutto questo vale anche nell'informatica perché quello di programmazione è un linguaggio con una propria sintassi, proprie parole chiavi, propri costrutti e proprie regole che

permette di interagire con le macchine invece che con le persone.

Si parte dagli algoritmi

Il "concetto" da esprimere è uno dei possibili algoritmi risolutivi del problema assegnato, che una volta rappresentato mediante l'uso di un determinato codice diventa il programma da eseguire. Senza volersi soffermare più di tanto sulla teoria degli algoritmi e sulle tante variabili in gioco, argomento a cui dedicheremo spazio in futuro in questa rubrica, mi preme in questo contesto focalizzare l'attenzione su un semplice concetto: è importante essere in grado di affrontare il problema attraverso la formulazione di un algoritmo risolutivo e solo dopo averlo definito ci si occuperà di scegliere il linguaggio più idoneo al caso specifico per tradurlo in programma.

Da questa semplice considerazione possiamo cambiare paradigma e così sfatare uno degli equivoci più comuni quando si intende definire quali siano le competenze da acquisire per essere appetibili e proattivi nel contesto della trasformazione digitale. La domanda giusta è dunque sei in grado di sviluppare un "pensiero algoritmico" per risolvere il problema dato? e non tanto quali e quanti linguaggi di programmazione si conoscono.

Il "pensiero algoritmico" è l'insieme dei processi mentali coinvolti nella formulazione di un problema che ammette una soluzione computazionale. La soluzione può essere eseguita da un essere umano, una macchina

calcolatrice, o in generale, da un esecutore generico. L'analisi dei problemi dal punto di vista algoritmico è alla base della creatività e innovazione necessarie per creare nuove tecnologie e fare la differenza in una società sempre più digitale. Il pensiero algoritmico è dunque una parte essenziale del pensiero computazionale: richiede ad una persona di astrarre il problema, specificarne quindi i dati essenziali ed infine formulare, passo per passo, le istruzioni da eseguire per risolvere il problema stesso.

In altri termini, il pensiero algoritmico richiede di:

1. analizzare ed astrarre un problema
2. specificare i dati del problema
3. definire e studiare un algoritmo per risolvere il problema.

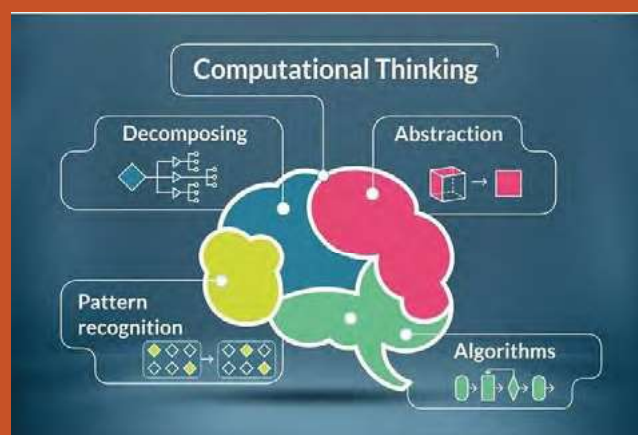
Pensare come una macchina?

Il pensiero computazionale, di cui il pensiero algoritmico è una parte fondamentale, è un concetto coniato nel 2006 dalla scienziata informatica Jeannette Wing, capo del dipartimento di Computer Science alla Carnegie Mellon University di Pittsburgh, in un suo breve articolo di approfondimento pubblicato sulla rivista Communications of the ACM (<https://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf>) dove definisce il pensiero computazionale come il processo mentale che consente di risolvere problemi di varia natura seguendo metodi e strumenti specifici; in poche parole è la capacità di risolvere un problema pianificando una strategia. Si tratta di un processo logico-creativo che consente di scomporre un problema complesso in diverse parti, che sono più gestibili se affrontate una per volta; trovando una soluzione a ciascuna di esse è possibile risolvere il problema generale.

Pensiero computazionale dunque inteso come attitudine mentale a descrivere, analizzare e risolvere problemi (cd. problem solving) attraverso un approccio progettuale scientifico che prescinde dall'uso di un computer, ma proprio delle discipline scientifiche.

Come messo ben in evidenza dalla Prof. Jeannette Wing, imparare a pensare come un informatico significa molto di più che essere capaci di programmare un computer, richiede di pensare a più livelli di astrazione.

Entra in gioco l'astrazione o meglio dire la capacità di astrarre e dunque di creare un modello della realtà ove siano presenti solo i dettagli essenziali rimuovendo perciò tutto il superfluo allo scopo



di rendere questo prototipo più comprensibile, ma anche utilizzabile in un processo di identificazione di schemi ricorrenti, somiglianze e connessioni che va sotto il nome di generalizzazione. Un classico esempio è la mappa della metropolitana dove sono accuratamente riportate tutte le informazioni necessarie ad un viaggiatore per utilizzare la rete metropolitana omettendo informazioni non essenziali allo scopo quali la distanza tra le stazioni, la posizione geografica, i punti di controllo, ecc. L'abilità nell'astrarre sta nello scegliere i dettagli da nascondere in modo che il problema diventi più facile, senza perdere niente di ciò che è importante. Un aspetto fondamentale di questo processo risiede nella scelta di una buona rappresentazione del sistema. Rappresentazioni diverse rendono più semplice operare in contesti diversi.

Conclusioni

In conclusione per rispondere più compiutamente alla domanda posta in apertura a questo articolo possiamo dire che per essere parte attiva nel processo di digitalizzazione delle nostre realtà produttive bisogna imparare ad utilizzare l'approccio di soluzione dei problemi proprio del pensiero computazionale, e dunque non è necessario conoscere uno o più linguaggi di programmazione atti a codificare un algoritmo. Sicuramente questo è il passo fondamentale per capire e dunque essere in grado, a vari livelli, di governare l'evoluzione tecnologica portata in dote dalla nuova rivoluzione industriale 4.0.

***Massimo Nannini**, ingegnere elettronico e libero professionista: si occupa di consulenza informatica, project management e formazione di impresa.

Contatti

Email: info@gemaxconsulting.it
www.gemaxconsulting.it