

Don't Repeat Yourself

Knowledge can keep us warm and DRY

di *Giuseppe Vincenzi*

La sigla che cito nel titolo di questo articolo è uno dei principi della programmazione informatica: citarlo mi è sembrato opportuno parlando di educazione all'informatica e alla programmazione.

Ma cos'è la programmazione informatica? O ancora più genericamente cos'è l'informatica oggi per un giovane?

Probabilmente la risposta che molti daranno a queste domande, finirà per descrivere l'informatica e soprattutto la programmazione informatica, come **un mondo chiuso e fatto per ingegneri e appassionati quasi patologici dei computer e dell'elettronica**. Non a caso ormai la figura dello sviluppatore è costantemente associata alla parola NERD: letteralmente uno SFIGATO, rinchiuso nella sua cameretta, a scrivere milioni di righe di codice per realizzare qualche suo software che, probabilmente, solo lui capirà a cosa serve realmente.

Ma non è per niente così.

L'informatica è anche fatta di NERD, ma ha soprattutto bisogno di **gente curiosa, con lo sguardo aperto a diverse discipline** (quindi assolutamente non solo ingegneri) e con una spiccata passione per la comunicazione: perché l'informatica è innanzitutto l'arte, passatemi il termine, di gestire l'informazione.

La dimostrazione del fatto che non siamo riusciti a comunicare il "mestiere" dell'informatica ai giovanissimi sta nei dati diffusi sul rapporto tra domanda e offerta di lavoro nel mondo della programmazione: in questo momento di alta disoccupazione,

l'Information Technology, e soprattutto il comparto legato allo sviluppo software, è in netta controtendenza.

C'è gran bisogno di sviluppatori, e sono sempre più rari.

Pochi tra i giovanissimi hanno scelto in questi ultimi 5 anni un percorso professionale di questo tipo.

Certo negli **anni '90** era più facile avvicinarsi ai linguaggi di programmazione: **l'informatica era ancora un piccolo mistero** e, chi aveva una certa dose di curiosità, al di là delle sue reali capacità informatiche, si avvicinava ai computer e al loro linguaggio, per capirne quantomeno le potenzialità.

Chi come me, in quegli anni, era un adolescente o ancora più giovane, vedeva nei computer una forma di gioco nuova rispetto ai Lego, o al Meccano, o ai personaggi della PlayMobil: cominciamo a chiedere come regalo dei computer come il **Commodore64**, con il quale potevamo iniziare ad avere le nostre prime esperienze di giocatori di videogiochi.

Certo la grafica era ancora primordiale, ma la cosa interessante era che ancora **vedevamo bene il "dietro le quinte" dell'informatica**: per far partire un gioco, eravamo costretti a eseguire comandi nelle *"shell di comando"* e quindi, anche senza averlo scelto, ci ritrovavamo a familiarizzare con delle parole chiavi dei linguaggi di programmazione e dei sistemi operativi dell'epoca.

Ognuno di noi poi poteva decidere che tipo di conoscenza voleva del mezzo informatico: c'era chi comprava i giochi, che allora erano contenuti in supporti a nastro (quelle che i più chiamavano *musicassette*), e imparava solo i comandi base per eseguirli; c'era invece chi voleva personalizzare la sua esperienza, e ad esempio andava in edicola a comprare dei mensili nei quali era riportato il codice sorgente dei giochi, che doveva essere riscritto a mano sul proprio computer, avendo

poi possibilità di apportare delle modifiche o delle personalizzazioni al gioco.

Le generazioni **dagli anni 2000 in poi** vivono un rischio che, in realtà, ha contagiato anche una buona parte della mia di generazione: utilizzare **un'informatica fatta per l'utente finale** che è facile da usare, molto più di quella di 20 anni fa, ma che è diventata una **"scatola nera"** per i suoi utenti.

Ora non vediamo più gli ingranaggi del giocattolo: usiamo le nostre applicazioni, sia sul Web, sia sul nostro smartphone, spesso non avendo alcuna idea di come siano fatte, di cosa ci sia *"dietro le quinte"*.

A mio parere **la scuola è proprio in questo vuoto che deve inserirsi**: deve prendere il giocattolo e aprirlo, smontarlo, mostrarne agli studenti gli ingranaggi. La scuola dovrebbe far sapere ai suoi studenti che dietro l'utilizzo di un'applicazione dello smartphone, c'è un codice, un sistema operativo, un hardware: senza per questo obbligare le persone a interessarsi all'informatica nel senso professionale del termine, ma solo per far conoscere meglio cosa c'è dietro un oggetto che quotidianamente è nelle nostre mani.

D'altronde a scuola tutti abbiamo studiato la matematica, ma non tutti siamo diventati dei ragionieri, dei commercialisti, o degli ingegneri.

Però anche i pianisti sappiamo che esiste un'operazione matematica che si chiama radice quadrata: la conoscenza, anche quando ci sembra inutile, accresce inconsciamente le nostre capacità di comprensione del mondo che ci circonda.

Prendiamo ad esempio il browser Chrome e navighiamo su un sito internet a nostro piacimento: se clicchiamo sul tasto F12 vedremo apparire una finestra con dentro il codice HTML che, interpretato dal nostro browser, genera la pagina che vediamo.

Secondo la mia idea di scuola, **in classe bisognerebbe mostrare che dietro un sito c'è un codice, senza soffermarsi sulla sintassi del linguaggio specifico:** l'importante è far capire cosa "fisicamente" c'è dietro quella pagina, e qual è il ruolo di "interprete" di un browser come Chrome.

Credo che a scuola sia importante insegnare "cos'è" un browser, piuttosto che investire il tempo nello spiegare "come si usa" un determinato browser.

Così come credo che a scuola sia importante insegnare "cos'è" un linguaggio di programmazione, piuttosto che soffermarsi sulla sintassi di uno specifico linguaggio.

La scuola, per quanto riguarda altre discipline diverse dall'informatica, si è sempre posta l'obiettivo di dare agli studenti gli "strumenti" che potessero aprirgli poi la strada dell'approfondimento specifico di una materia: io ho frequentato il Liceo Scientifico dove, ad esempio, mi hanno insegnato cosa fosse l'Algebra Lineare, insegnandomi i principi base, che poi sono stati lo strumento che mi ha consentito di approfondire e superare esami di Analisi Matematica nel mio corso di studi in Ingegneria dell'Automazione Industriale.

Ecco, io credo che, allo stesso modo, la scuola debba fornire le **basi "teoriche" all'informatica**, dando in mano al singolo studente la conoscenza e gli strumenti che, qualora volesse

fare dell'informatica la sua professione, gli consentiranno di approfondire, ma con basi che agevolano il suo studio.

Per entrare nello specifico, io ad esempio farei utilizzare a scuola **Linux**, e dei mini-pc come il **Raspberry PI**: questi strumenti "obbligano" gli utilizzatori a guardare gli ingranaggi dell'informatica e a "sporcarsi le mani" con le righe di comando.

Ma d'altronde gli anni in cui siamo a scuola, sono gli anni in cui possiamo "imporci" e "lasciarci imporre" delle scelte di metodo che ci obbligano a utilizzare il tempo che abbiamo per scoprire mondi nuovi: più tardi ognuno di noi avrà sempre meno tempo (e spesso sempre più voglia) di conoscere mondi lontani dai propri interessi del momento e dalle tendenze che crediamo naturali.

Java per me è diventato il mio lavoro quotidiano, nonché una passione che coltivo ogni giorno con piacere: eppure quando l'ho incontrato ero un giovane studente di robotica, appassionato per l'algebra, e amante della musica, della filosofia e del teatro. Mai avrei pensato di avvicinarmi ad un linguaggio di programmazione, fino a che, obbligato appunto dalla scuola, e nello specifico da un esame nel mio corso di studi, ho scoperto la **programmazione orientata agli oggetti**: ho scoperto quanta **creatività** ci potesse essere nell'ingegneria del software, e non sono più riuscito ad allontanarmi da questa disciplina.

Tanto che l'ultimo mio libro, "**Sette grani di caffè contro lo stress – Guida per sospetti programmatori Java**" (disponibile gratuitamente sul web), è proprio un manuale in cui cerco di spiegare i principi della programmazione orientata agli oggetti, nel linguaggio Java specificatamente, usando la filosofia di Platone.

Come si può intuire proprio da questo esempio sulla mia pelle, occuparsi di informatica e di programmazione, non vuol dire affatto occuparsi di bassi “tecnicismi”, ma significa invece riflettere su concetti ben più **complessi**.

Quale miglior modo, quindi, che affrontare una società complessa come la nostra, se non quello di allenarsi a modellizzarla e formalizzarla, non con il fine di controllarla, come faremmo con un computer o un braccio meccanico, ma solo con l'obiettivo di comprenderla e, se siamo bravi, di anticiparla.

Giuseppe Vincenzi